

Rekontruksi Data Suhu dan Tekanan pada Pembangkit Listrik Tenaga Mesin Gas (PLTMG) menggunakan Recurrent Neural Network

Enggal Nur F Naufaldhianto¹, Berlian Al Kindhi*², Ilham Agung Wicaksono³

^{1,2,3} Departemen Teknik Elektro Otomasi, Institut Teknologi Sepuluh Nopember, Surabaya
Email: ¹2040211002@student.its.ac.id, ²berlian@its.ac.id, ³ilham.agung@its.ac.id

*Penulis Korespondensi

Abstrak

Pembangkit Listrik Tenaga Mesin Gas (PLTMG) memanfaatkan sensor suhu dan tekanan untuk memantau kondisi operasional. Sistem keamanan konvensional yang menggunakan ambang batas tetap sering kali tidak mampu mengidentifikasi gejala awal gangguan sebelum mencapai kondisi kritis. Hal ini berisiko menyebabkan terjadinya *shutdown* tak terduga. Penelitian ini mengusulkan pengembangan sistem deteksi anomali berbasis metode *Long Short-Term Memory (LSTM) Autoencoder* sebagai solusi yang lebih adaptif. Data sensor dikumpulkan melalui protokol MQTT dan disimpan dalam database, kemudian diproses melalui tahap pra-pemrosesan, normalisasi, dan pembentukan data berurutan (*time series*). Model dilatih untuk mengenali pola normal dan mengidentifikasi deviasi melalui nilai *reconstruction loss*. Hasil pengujian menunjukkan bahwa konfigurasi model optimal menghasilkan nilai MSE terendah sebesar 0.00035 dengan kemampuan yang baik dalam mendeteksi anomali pada data yang sebelumnya tidak dikenali. Sistem ini terbukti efektif memberikan peringatan dini dan berpotensi mengurangi risiko kerusakan maupun gangguan operasional.

Kata kunci: deteksi anomali, LSTM *autoencoder*, peringatan dini, PLTMG

Abstract

Gas Engine Power Plants (PLTMG) utilize temperature and pressure sensors to monitor operational conditions. Conventional safety systems that rely on fixed threshold limits often fail to detect early signs of malfunction before reaching critical levels, increasing the risk of unexpected shutdowns. This study proposes the development of an anomaly detection system based on the Long Short-Term Memory (LSTM) Autoencoder method as a more adaptive solution. Sensor data is collected via the MQTT protocol and stored in a database, then processed through stages of preprocessing, normalization, and time series formatting. The model is trained to recognize normal patterns and identify deviations using reconstruction loss values. Test results show that the optimal model configuration yields the lowest Mean Squared Error (MSE) of 0.00035, demonstrating strong performance in detecting anomalies in previously unseen data. The system proves effective in providing early warnings and has the potential to reduce the risk of damage and operational disruptions.

Keywords: anomaly detection, early warning, LSTM *autoencoder*, PLTMG

I. PENDAHULUAN

Salah satu jenis pembangkit listrik yang ada di Indonesia adalah Pembangkit Listrik Tenaga Mesin Gas (PLTMG). Mesin pada PLTMG dirancang beroperasi dengan dua bahan bakar (*dual fuel*), yaitu gas alam dan bahan bakar minyak diesel. Bahan bakar minyak diesel biasanya digunakan untuk dua fungsi, yaitu sebagai bahan bakar utama (*main fuel*) dan bahan bakar awalan (*pilot fuel*). Bahan bakar awalan akan selalu menggunakan bahan bakar minyak diesel pada setiap operasi mesin. Sedangkan bahan bakar utamanya dapat ditukar (*switch-over*) dari bahan bakar minyak diesel ke bahan bakar gas. Sehingga, mesin PLTMG tidak hanya dapat dioperasikan dengan menggunakan 100% bahan bakar minyak diesel [1].

Sistem bahan bakar pada PLTMG dilengkapi dengan sensor-sensor yang dapat mengetahui nilai dari suhu dan tekanan. Nilai tersebut ditampilkan pada sistem *monitoring* untuk dilakukan pemantauan secara *real-time*. Sistem keamanan pada mesin menggunakan ambang batas tetap dari nilai suhu dan tekanan. Nilai suhu dan tekanan yang melebihi ambang batas dapat memicu alarm dan *valve* suplai bahan bakar ke mesin tertutup, sehingga mesin akan mengalami *shutdown* [2]. Sistem keamanan tidak dapat mendeteksi adanya anomali saat terjadi perubahan drastis pada nilai sensor sebelum mencapai ambang

batas. Mesin yang sering mengalami *shutdown* dapat mengakibatkan komponen mesin yang bekerja dalam kondisi tidak stabil akan cepat mengalami kerusakan atau penurunan kinerja mesin [3].

Deteksi anomali mengacu pada proses menemukan pola dalam data yang tidak sesuai dengan perilaku umum dalam suatu kelompok data. Anomali sering kali mengindikasikan masalah yang perlu perhatian khusus, seperti kesalahan sistem, potensi ancaman keamanan, atau kejadian langka [4]. Anomali dapat dikelompokkan menjadi tiga kategori, yaitu anomali titik (*point anomalies*), anomali kontekstual (*contextual anomalies*), dan anomali kolektif (*collective anomalies*) [5]. Pendekatan dalam deteksi anomali dapat dilakukan menggunakan *machine learning*. *Machine learning* digunakan untuk membangun model yang membedakan kelas normal dan kelas abnormal. Deteksi anomali dapat dibagi menjadi tiga kategori berdasarkan fungsi data pelatihan yang digunakan untuk membangun model, yaitu *supervised anomaly detection*, *semi-supervised detection*, dan *unsupervised anomaly detection* [6].

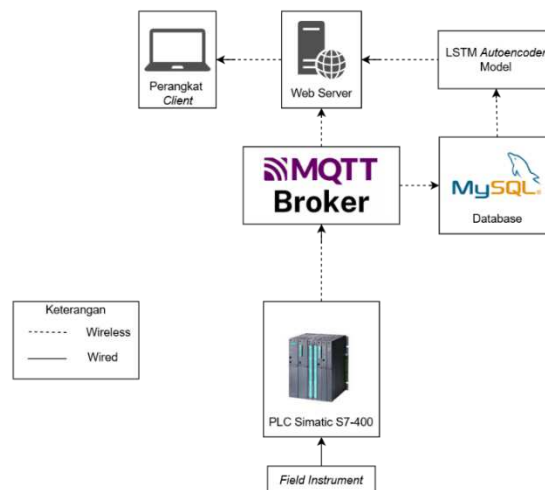
Penelitian untuk mendeteksi adanya sebuah anomali bukan merupakan hal yang baru. Pada penelitian sebelumnya, deteksi anomali dapat menggunakan beberapa metode. Penggunaan metode *Support Vector Data Description* (SVDD) dapat mendeteksi adanya sebuah anomali pada Pembangkit Listrik Tenaga Nuklir (PLTN), akan tetapi pada penelitian tersebut hanya menggunakan satu variabel tunggal tanpa mempertimbangkan hubungan antara variabel-variabel lainnya [7]. Metode *Isolation Forest* (IF) juga dapat digunakan untuk mendeteksi adanya anomali dan memberikan peringatan dini, tetapi tidak cocok digunakan pada data yang kompleks seperti data time series [8]. Sedangkan metode *Long Short-Term Memory* (LSTM) *Autoencoder* dapat mendeteksi adanya sebuah anomali pada data time series, tetapi masih jarang digunakan sebagai peringatan dini pada sistem pembangkit listrik [9].

LSTM *autoencoder* menggunakan gabungan atribut dari LSTM dan *autoencoder*, yang membangun jaringan LSTM berdasarkan operasi *encoding* dan *decoding* dari *autoencoder* [10]. LSTM *autoencoder* terdiri dari *input layer*, *encoder layers*, *repeat vector*, *decoder layers*, dan *output layer*. Pengoperasian LSTM *autoencoder* dapat dibagi menjadi beberapa tahap, yaitu *input sequence data*, LSTM *encoder*, LSTM *decoder*, dan menghitung *reconstruction loss* [11].

Penelitian ini menggunakan pendekatan LSTM *autoencoder* dalam mendeteksi adanya anomali data suhu dan tekanan pada PLTMG. Penelitian ini bertujuan untuk mengembangkan proteksi awal dari sistem keamanan berupa peringatan dini (*early warning*) yang dapat memberikan informasi kepada operator. Peringatan tersebut ditampilkan pada sistem *monitoring* sehingga dapat membantu operator mengetahui potensi terjadinya *shutdown* yang tidak diperlukan.

II. METODE PENELITIAN

2.1. Arsitektur Sistem



Gambar 1. Arsitektur Sistem

Gambar 1 merupakan arsitektur sistem dirancang mampu menghubungkan perangkat keras di lapangan dengan sistem pemrosesan data dan analisis yang berjalan di komputer. Arsitektur sistem ini dimulai dari data yang dikirim oleh sensor-sensor yang ada di lapangan (*field instrument*) ke PLC. PLC berfungsi sebagai pengumpul data utama yang kemudian mengirimkan data tersebut ke sistem pusat menggunakan protokol komunikasi MQTT Broker. Data yang masuk akan disimpan ke dalam *database*

MySQL dan dikirim ke *web server* sebagai bagian dari proses pemantauan. Data historis yang tersimpan pada database akan digunakan untuk melatih model LSTM *Autoencoder*. Model ini bertugas mempelajari pola normal dari data suhu dan tekanan, sehingga mampu mendeteksi adanya anomali ketika menerima data baru. Jika data baru memiliki perbedaan yang signifikan terhadap pola normal, sistem akan menandainya sebagai anomali dan mengirimkan peringatan secara otomatis ke operator melalui sistem *monitoring*.

2.2. Pengumpulan dan Pengambilan Data

Pada tahap penelitian ini, data diambil melalui MQTT broker yang terhubung dengan sensor-sensor yang ada di unit pembangkit. Sistem pemantauan tersebut hanya menampilkan data sesuai dengan parameter secara *real-time* tanpa adanya perekaman data. Sehingga perlu dilakukan perekaman data terlebih dahulu untuk mendapatkan data history dari parameter-parameter tersebut. Perekaman data dilakukan dengan cara mengambil data dari MQTT broker menggunakan Node-Red. Data-data tersebut disimpan pada database MySQL. Data yang telah tersimpan kemudian diambil untuk dikelola menjadi dataset dengan memperhatikan parameter yang akan digunakan dan panjang data berdasarkan waktu yang tersedia pada database. Dataset yang digunakan berjumlah 32920 data untuk setiap parameter dengan rentang waktu 23 hari dan pengambilan data dilakukan setiap menit, sehingga menghasilkan volume data yang besar untuk analisis lebih lanjut. Dataset tersebut akan digunakan untuk membentuk model LSTM *autoencoder*.

2.3. Pra-Pemrosesan Data

Pra-pemrosesan data dilakukan beberapa tahapan untuk mengubah data sensor menjadi data untuk proses pemodelan. Tahap awal data dibersihkan untuk memastikan bahwa data yang digunakan dalam pelatihan dan pengujian model bebas dari kesalahan, tidak lengkap, tidak konsisten, atau nilai-nilai ekstrem yang dapat memengaruhi performa model. Kualitas data yang buruk dapat mengakibatkan model belajar dari pola yang salah dan berdampak pada rendahnya akurasi dalam mendeteksi anomali. Dalam penelitian ini, data yang dikumpulkan berasal dari sensor-sensor pada PLTMG dan disimpan pada database melalui protokol komunikasi MQTT. Dalam proses pengumpulan data terdapat kemungkinan bahwa sensor gagal mengirimkan data pada waktu tertentu, baik karena gangguan komunikasi, gangguan perangkat keras, atau pemeliharaan sistem. Akibatnya, data yang masuk ke dalam database mengandung nilai kosong (*missing values*) atau NaN (*Not a Number*). Proses pembersihan data dilakukan dengan asumsi bahwa setiap nilai NaN menandakan bahwa sensor tidak melakukan pengiriman pada waktu tersebut. Oleh karena itu, penanganan dilakukan dengan cara imputasi data dengan pendekatan *forward filling*, yaitu teknik pengisian nilai yang hilang (*missing values*) berdasarkan nilai sebelumnya yang masih valid [12]. Pendekatan ini dipilih karena dianggap paling representatif dalam konteks data sensor yang bersifat kontinu dan bersambung antar waktu pengukuran. Selain menggunakan imputasi data, penanganannya dapat juga dilakukan dengan cara penghapusan baris data yang tidak lengkap jika jumlah *missing values* terlalu banyak dan dapat mengganggu kestabilan model.

Setelah data dibersihkan, dilakukan proses normalisasi data sebelum data digunakan sebagai input ke dalam proses pembuatan model, khususnya model berbasis jaringan saraf seperti LSTM *autoencoder*. Tujuan utama normalisasi adalah untuk menyamakan skala antar variabel (fitur) agar model tidak mendominasi atau mengabaikan fitur tertentu hanya karena memiliki skala numerik yang lebih besar atau lebih kecil. Perbedaan skala ini dapat mempengaruhi proses pembelajaran karena model cenderung memberikan bobot lebih terhadap fitur dengan nilai numerik yang lebih tinggi. Pada Penelitian ini, normalisasi data menggunakan *Min-Max Scaling* karena data sensor yang digunakan memiliki rentang nilai yang terdefinisi dengan baik dan bersifat *periodic* [13]. Proses normalisasi ini dilakukan terhadap seluruh parameter sensor sebelum data dimasukkan ke dalam model untuk memastikan konsistensi dan meningkatkan kinerja model dalam mengenali pola serta mendeteksi anomali secara akurat. Persamaan *Min-Max Scaling* dapat dituliskan sebagai berikut:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad [1]$$

Dimana,

X : Nilai asli fitur

X_{min} : Nilai minimum fitur

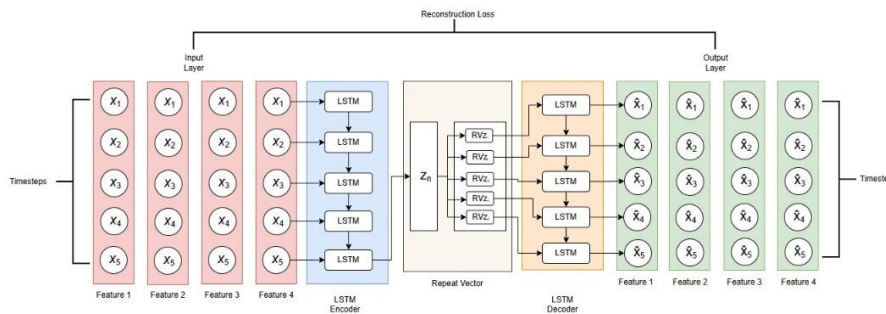
X_{max} : Nilai maksimum fitur

X' : Nilai hasil normalisasi

Setelah proses normalisasi, data sensor hasil akuisisi awal pada umumnya memiliki bentuk dua dimensi (2D) dengan struktur (jumlah baris, jumlah fitur). Untuk dapat diproses oleh model LSTM *autoencoder*, data tersebut harus ditransformasikan ke dalam format 3D. Proses ini dikenal sebagai *reshaping* atau *windowing*, yaitu memotong data menjadi sekumpulan urutan (*sequence*) dengan panjang *time step* tertentu, sehingga terbentuk potongan-potongan data *time series* yang berkesinambungan. Oleh karena itu, model ini memerlukan input dalam format tiga dimensi (3D) dengan struktur (*samples, timesteps, features*) [14]. Dimana, *samples* merupakan jumlah total potongan data time series yang akan digunakan sebagai input model, *timesteps* merupakan panjang urutan waktu, dan *features* adalah jumlah parameter atau variabel pada setiap titik waktu.

Setelah itu, Data yang telah melalui tahap pembersihan, normalisasi, dan pembentukan format 3D, kemudian dibagi menjadi tiga subset utama, yaitu data *training*, *validation*, dan *testing*. Data *training* menggunakan 60% dari total data yang digunakan untuk melatih model agar dapat mengenali pola-pola umum dari kondisi data yang dianggap normal. Data *validation* menggunakan 20% dari total data yang berfungsi sebagai acuan untuk mengontrol performa model saat pelatihan dan mendeteksi *overfitting*. Data *testing* menggunakan 20% dari total data untuk melakukan evaluasi akhir terhadap performa model yang telah dilatih dan divalidasi. Data *testing* ini benar-benar data baru bagi model (tidak pernah dilihat sebelumnya selama pelatihan maupun validasi), sehingga mampu menunjukkan seberapa baik model mampu mengenali pola baru dan mendeteksi anomali secara general.

2.4. Model LSTM Autoencoder



Gambar 2. Arsitektur Model LSTM Autoencoder

Pada penelitian ini model dirancang dengan menggabungkan LSTM dan *Autoencoder* untuk mendeteksi anomali data suhu dan tekanan pada PLTMG. Gambar 2 menunjukkan arsitektur model LSTM *autoencoder* terdiri dari *input layer*, *encoder layers*, *repeat vector*, *decoder layers* dan *output layer*. Pengoperasian LSTM *autoencoder* dapat dibagi menjadi beberapa tahap, yaitu *input sequence data*, LSTM *encoder*, LSTM *decoder*, menghitung *reconstruction loss*. Dalam proses penerapan LSTM *Autoencoder* untuk deteksi anomali, langkah pertama adalah membentuk data historis menjadi urutan-urutan (*sequence*) dengan waktu tetap, lalu mengubahnya ke dalam format tiga dimensi agar sesuai dengan kebutuhan *input* model LSTM. Selanjutnya memasuki tahap *encoder*, pada tahap ini LSTM memproses setiap urutan data dan mencoba mengenali pola-pola hubungan antar waktu. Hasil dari proses ini adalah sebuah vektor laten, yaitu representasi ringkas yang memuat informasi penting dari seluruh urutan data. Tahap berikutnya adalah proses *decoder*. Dengan menggunakan vektor laten dari proses *encoder*, model mencoba membangun kembali urutan data yang asli. Proses ini tidak selalu sempurna, sehingga akan ada selisih antara data hasil rekonstruksi dan data aslinya. Langkah terakhir adalah menghitung *reconstruction loss*. Jika nilai loss rendah maka data dianggap normal. Namun, jika loss melebihi ambang batas yang ditentukan saat validasi maka data diklasifikasikan sebagai anomali. Pendekatan ini memungkinkan deteksi anomali berdasarkan perbedaan antara pola aktual dan pola yang telah dipelajari oleh model.

2.5. Evaluasi Model

Pada tahap ini model LSTM *autoencoder* akan dievaluasi untuk melihat model terbaik. Pengujian model dilakukan dengan cara mencari *hyperparameter* terbaik dalam menghasilkan MSE (*Mean squared error*) terkecil sebagai parameter untuk menyimpulkan model LSTM *autoencoder* dapat merekonstruksi data dengan sangat baik. Kemudian setelah mendapatkan *reconstruction error* terkecil dari setiap pengujian, akan dilakukan validasi model pada data baru untuk menguji apakah model tetap dapat mampu untuk merekonstruksi data yang belum pernah dilihat [15]. Nilai MSE yang dihasilkan pada

tahap validasi model akan digunakan sebagai *threshold* untuk mendeteksi anomali pada tahap pengujian, apabila nilai error dari data pengujian melebihi dari error yang dijadikan sebagai *threshold*, maka data tersebut dianggap sebagai anomali. Persamaan MSE dapat dituliskan sebagai berikut:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - y'_i)^2 \quad [2]$$

Dimana,

y_i : Data asli

y'_i : Rekonstruksi data

N : Nilai rata-rata dari setiap data

III. HASIL DAN PEMBAHASAN

3.1. Hasil Pengumpulan dan Pembersihan Data

Tabel 1. Hasil Pengumpulan Data

Time	FO_BM_FOP	FO_BM_FODP	FO_BM_FOT1	FO_BM_FOT2
2024-11-08 00:00:00	103.0	NaN	NaN	NaN
2024-11-08 00:01:00	NaN	6776.0	NaN	NaN
2024-11-08 00:02:00	103.0	5592.0	NaN	NaN
2024-11-08 00:03:00	103,3	5823,3	NaN	128
2024-11-08 00:04:00	103,34	5865,9	NaN	128

Data yang digunakan merupakan hasil dari pengumpulan data historis dari parameter suhu dan tekanan yang telah disimpan pada database lokal. Panjang data setiap parameter adalah 32920 dengan rentang waktu 23 hari. Pada Tabel 1 setiap baris menunjukkan waktu pengambilan data dalam format *timestamp*, sedangkan setiap kolom setelahnya menunjukkan nilai sensor dari parameter terkait. Terlihat bahwa dalam beberapa waktu tertentu terdapat nilai NaN (*Not a Number*) yang menunjukkan bahwa sensor gagal mengirimkan data atau terjadi gangguan komunikasi data pada saat data dikirimkan. Oleh karena itu, dilakukan imputasi data dengan pendekatan *forward filling* dan juga dilakukan penghapusan baris data yang tidak lengkap jika jumlah *missing values* terlalu banyak dan dapat mengganggu kestabilan model. Hasil dari pembersihan data dari beberapa parameter dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pembersihan Data

Time	FO_BM_FOP	FO_BM_FODP	FO_BM_FOT1	FO_BM_FOT2
2024-11-08 00:06:00	103.0	5850.17	108.0	127.5
2024-11-08 00:07:00	103.23	5835.73	108.0	127.0
2024-11-08 00:08:00	103.16	5824.55	108.0	127.0
2024-11-08 00:09:00	103.16	5824.55	108.0	127.0
2024-11-08 00:10:00	103.16	5824.55	108.0	127.0

Berdasarkan Tabel 2, dapat dilihat bahwa parameter telah memiliki nilai yang lengkap dan tidak lagi mengandung nilai kosong. Proses imputasi menggunakan metode *forward filling* berhasil mempertahankan kontinuitas data tanpa mengubah pola asli dari tren pengukuran. Selain itu, penghapusan baris-baris yang memiliki *missing values* berlebih turut membantu meningkatkan konsistensi data secara keseluruhan. Dengan kondisi data yang sudah bersih dan terstruktur seperti ini, pelatihan model deteksi anomali dapat dilakukan dengan lebih optimal dan meminimalisasi adanya gangguan akibat data yang tidak teratur.

3.2. Hasil Normalisasi dan Pembagian Data

Sebelum data digunakan untuk melatih model LSTM *autoencoder*, perlu dilakukan normalisasi data untuk menstandarisasi nilai-nilai fitur dengan tujuan menjadikan nilai pada seluruh variabel dalam rentang nilai yang sama. Hal ini dilakukan untuk menjadikan variabel memiliki standar deviasi pada skala yang sama ketika dimasukkan dalam pembuatan model LSTM *autoencoder*. Hasil dari dataset yang telah dinormalisasi bisa dilihat pada Tabel 3.

Tabel 3. Hasil Normalisasi Data

FO_BM_FOP	FO_BM_FOT1	FO_BM_FOP Normalisasi	FO_BM_FOT1 Normalisasi
103.0	108.0	0.94	0.75
103.23	108.0	0.95	0.75
103.16	108.0	0.94	0.75
103.16	108.0	0.95	0.75
103.16	108.0	0.96	0.75

Pada Tabel 3 menunjukkan hasil normalisasi data dari beberapa parameter yang digunakan dalam penelitian. Proses normalisasi dilakukan dengan metode *Min-Max Scaling*, yang mengubah nilai asli pada masing-masing fitur ke dalam rentang [0, 1] berdasarkan nilai minimum dan maksimum dari setiap fitur. Tujuan dari normalisasi ini adalah untuk memastikan bahwa setiap fitur memiliki kontribusi yang seimbang selama proses pelatihan model, serta menghindari dominasi fitur yang memiliki skala numerik lebih besar. Dengan skala yang seragam, model LSTM *autoencoder* dapat mempelajari pola dan struktur data dengan lebih akurat tanpa bias terhadap fitur tertentu.

Setelah proses normalisasi, total data sebanyak 32.910 baris dibagi secara proporsional ke dalam tiga subset utama, yaitu data pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*). Sebanyak 60% dari keseluruhan data, yaitu 19.746 baris, digunakan untuk melatih model LSTM *autoencoder* agar dapat mempelajari pola normal dari data sensor. Selanjutnya, 20% data (6.582 baris) digunakan untuk validasi selama proses pelatihan guna memantau kinerja model dan mencegah *overfitting*. 20% sisanya (6.582 baris) dipisahkan sebagai data pengujian untuk mengevaluasi performa akhir model setelah pelatihan selesai.

3.3. Hasil Pengujian Rekonstruksi Data

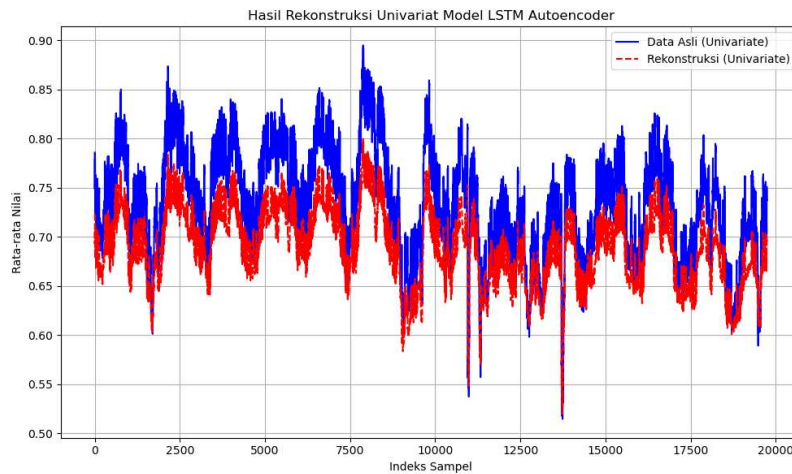
Pada pengujian ini dilakukan dengan berbagai konfigurasi untuk menemukan pengaturan terbaik yang memberikan hasil MSE (*Mean Squared Error*) terkecil pada proses *training* dan *validation*. Pada skema pengujian yang dilakukan adalah mengubah fungsi aktivasi, jumlah *layer*, *epoch*, *batch size*, dan *time step* untuk melihat pengaruh terhadap performa dari model dengan membandingkan nilai dari matrik MSE (*Mean Squared Error*), model yang menghasilkan MSE terkecil diasumsikan dapat merekonstruksi pola data normal pada tahap *training* dan *validation*. Konfigurasi lengkap dari skema pengujian ditunjukkan pada Tabel 4.

Tabel 4. Konfigurasi *Hyperparameter Tuning*

Fungsi Aktivasi	Jumlah Layer	Epoch	Batch Size	Time Step	MSE
Tanh	4	50	64	10	0.00080
Tanh	4	100	64	10	0.00047
Tanh	4	100	64	50	0.00167
Tanh	6	100	64	10	0.00157
Tanh	4	100	32	10	0.00035
ReLU	4	100	64	10	0.0068

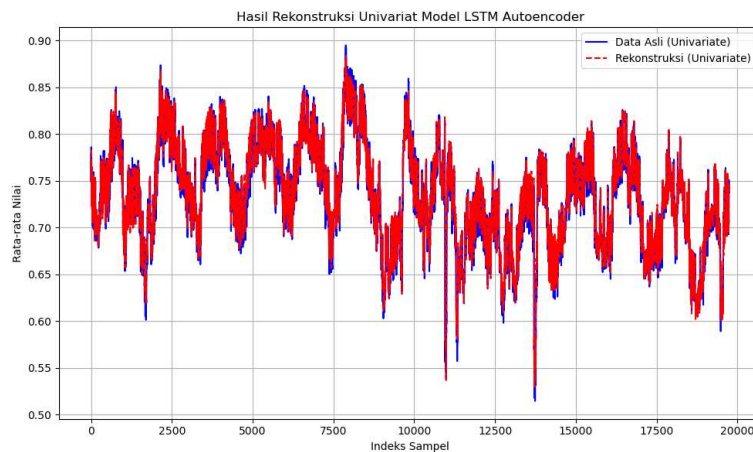
Berdasarkan Tabel 4 model dengan konfigurasi fungsi aktivasi Tanh, 4 *layer*, 100 *epoch*, 32 *batch size*, dan 10 *time step* menghasilkan nilai MSE sebesar 0.00035, nilai tersebut lebih kecil dibandingkan dengan nilai MSE model lainnya. Semakin kecil nilai MSE, semakin baik kemampuan model dalam merekonstruksi data dengan akurat. Nilai MSE meningkat secara signifikan pada konfigurasi *time step* yang lebih panjang, seperti 0.001677 untuk *time step* 50. Hal ini mengonfirmasi bahwa penambahan panjang urutan waktu secara berlebihan tidak membuat performa model meningkatkan, melainkan dapat memperburuk kemampuan generalisasi akibat kompleksitas arsitektur yang tidak sebanding dengan

kapasitas pembelajaran. Peningkatan jumlah *epoch* dari 50 menjadi 100 memberikan dampak positif terhadap kualitas rekonstruksi data. Hasil ini memperkuat bahwa jumlah *epoch* yang lebih tinggi dalam batas wajar mampu meningkatkan kapasitas representasi model LSTM *autoencoder* dalam mendeteksi ketidaksesuaian kecil.



Gambar 3. Grafik Hasil Rekonstruksi Data

Pada Gambar 3 menunjukkan grafik hasil rekonstruksi dari model dengan fungsi aktivasi ReLU. Model ini menghasilkan pola rekonstruksi yang kurang akurat terhadap data asli. Kurva hasil rekonstruksi tampak lebih datar dan tidak mengikuti variasi data dengan baik, khususnya pada puncak dan lembah yang mencerminkan dinamika utama sinyal. Hal ini merupakan sifat dari fungsi ReLU yang hanya mengaktifkan neuron pada nilai positif dan mengabaikan input nilai negatif yang dapat menyebabkan hilangnya sebagian informasi penting dalam proses pembelajaran. Ketidakesuaian tersebut juga dapat dilihat melalui nilai *Mean Squared Error* (MSE) sebesar 0.0068 yang cukup tinggi dalam konteks rekonstruksi data normalisasi. Nilai ini menandakan bahwa rata-rata kesalahan kuadrat antara data asli dan hasil rekonstruksi masih cukup besar yang menunjukkan bahwa model belum mampu mempelajari pola sinyal secara optimal. Meskipun rekonstruksi yang kurang presisi ini mungkin bermanfaat untuk mendeteksi anomali besar, performa model dengan ReLU cenderung tidak cocok untuk aplikasi yang membutuhkan ketelitian tinggi.



Gambar 4. Grafik Hasil Rekonstruksi Data

Sedangkan pada Gambar 4, model LSTM *autoencoder* dengan fungsi aktivasi tanh menunjukkan performa rekonstruksi yang sangat baik. Kurva hasil rekonstruksi sangat mendekati data asli dengan perbedaan yang hampir tidak terlihat secara visual. Fungsi tanh dengan rentang output -1 hingga 1 memiliki kelebihan dalam menangani input positif maupun negatif, sehingga dapat mempertahankan informasi secara lebih menyeluruh sepanjang proses *encoding* dan *decoding*. Keunggulan ini tercermin dari nilai

Mean Squared Error (MSE) yang jauh lebih rendah sebesar 0.00047. Nilai tersebut menunjukkan bahwa rata-rata kesalahan rekonstruksi sangat kecil. Sehingga, fungsi aktivasi tanh terbukti lebih baik dalam rekonstruksi yang akurat dari data dan menjadikannya pilihan yang lebih tepat untuk sistem deteksi anomali yang memerlukan sensitivitas terhadap perubahan kecil pada data. Grafik dari hasil rekonstruksi model dengan nilai MSE terkecil membuktikan bahwa model mampu mengikuti pola data asli dengan sangat baik. Perubahan kecil pada data seperti kenaikan atau penurunan nilai parameter yang tajam dapat direkonstruksi dengan akurat. Dari hasil tersebut dapat disimpulkan bahwa model tersebut memberikan keunggulan dalam menangkap pola data normal secara lebih rinci, sehingga lebih sesuai untuk mendeteksi anomali dalam penelitian ini.

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem deteksi anomali berbasis *Long Short-Term Memory* (LSTM) *autoencoder* yang adaptif pada Pembangkit Listrik Tenaga Mesin Gas (PLTMG). Sistem ini menunjukkan keunggulan dibandingkan pendekatan konvensional berbasis ambang batas tetap, dengan kemampuan mendeteksi pola-pola penyimpangan lebih dini. Hasil pengujian menunjukkan bahwa konfigurasi optimal model, yaitu fungsi aktivasi Tanh, 4 layer, 100 epoch, batch size 32, dan time step 10 dapat menghasilkan nilai Mean Squared Error (MSE) terendah sebesar 0.00035. Nilai tersebut menunjukkan bahwa model mampu merekonstruksi data normal secara akurat. Pengujian pada data baru yang tidak dikenali sebelumnya juga menunjukkan bahwa sistem memiliki sensitivitas tinggi terhadap penyimpangan perilaku operasional yang tidak sesuai dengan pola yang telah dipelajari, serta mampu memberikan peringatan dini sebelum terjadi kondisi kritis atau *shutdown*.

Dengan demikian, sistem yang dikembangkan dapat diandalkan sebagai alat bantu untuk meningkatkan keandalan dan stabilitas operasi PLTMG. Penelitian ini memiliki potensi untuk dikembangkan lebih lanjut melalui penambahan parameter pemantauan seperti getaran, arus, dan tegangan, serta integrasi dengan pemantauan secara *real-time*. Selain itu, implementasi metode pembelajaran adaptif seperti *online learning* juga dapat menjadi pengembangan yang relevan untuk meningkatkan kemampuan generalisasi model terhadap dinamika nilai sensor yang terus berubah.

REFERENSI

- [1] "71-80 ANALISIS EFEKTIVITAS MESIN DIESEL WARTSILA W20V34DF BERBASIS PERFORMANCE TEST (PT) PADA PEMBANGKIT LISTRIK TENAGA MESIN GAS (PLTMG) AMBON (3)".
- [2] "Technical Documentation Operating Manual Control and Monitoring System."
- [3] R. Nur Robihah *et al.*, "Upaya Peningkatan Daya Mampu Mesin Diesel Mitsubishi S16R-PTA-S di UPDK Tarakan PLTD Malinau," 2022. [Online]. Available: <http://prosiding.pnj.ac.id>
- [4] N. R. Prasad, S. Almanza-Garcia, and T. T. Lu, "Anomaly detection," *Computers, Materials and Continua*, vol. 14, no. 1, pp. 1–22, 2009, doi: 10.1145/1541880.1541882.
- [5] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," 2021, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2021.3083060.
- [6] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS One*, vol. 11, no. 4, Apr. 2016, doi: 10.1371/journal.pone.0152173.
- [7] *2014 IEEE Students' Technology Symposium : February 28 - March 02, 2014*. IEEE Student Branch, IIT Kharagpur, IEEE Kharagpur Section, 2014.
- [8] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep Isolation Forest for Anomaly Detection," Jun. 2022, doi: 10.1109/TKDE.2023.3270293.
- [9] H. Pariaman, G. M. Luciana, M. K. Wisyaldin, and M. Hisjam, "Anomaly detection using lstm-autoencoder to predict coal pulverizer condition on coal-fired power plant," *Evergreen*, vol. 8, no. 1, pp. 89–97, 2021, doi: 10.5109/4372264.
- [10] R. Shrestha *et al.*, "Anomaly detection based on LSTM and autoencoders using federated learning in smart electric grid," *J Parallel Distrib Comput*, vol. 193, Nov. 2024, doi: 10.1016/j.jpdc.2024.104951.
- [11] "View of Unsupervised marine vessel trajectory prediction using LSTM network and wild bootstrapping techniques _ Nonlinear Analysis_ Modelling and Control".

-
- [12] J. Wu, N. Li, and Y. Zhao, "Missing data filling based on the spectral analysis and the Long Short-Term Memory network," in *Proceedings - 2021 International Symposium on Computer Technology and Information Science, ISCTIS 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 198–202. doi: 10.1109/ISCTIS51085.2021.00049.
- [13] S. Gopal, K. Patro, and K. Kumar Sahu, "Normalization: A Preprocessing Stage." [Online]. Available: www.kiplinger.com,
- [14] A. Alos and Z. Dahrouj, "Using MLSTM and Multioutput Convolutional LSTM Algorithms for Detecting Anomalous Patterns in Streamed Data of Unmanned Aerial Vehicles," Jun. 01, 2022, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/MAES.2021.3053108.
- [15] J. Terven, D. M. Cordova-Esparza, J. A. Romero-González, A. Ramírez-Pedraza, and E. A. Chávez-Urbiola, "A comprehensive survey of loss functions and metrics in deep learning," *Artif Intell Rev*, vol. 58, no. 7, Jul. 2025, doi: 10.1007/s10462-025-11198-7.