

## **Laporan UAS Deep Learning**

### **PENGEMBANGAN SISTEM DETEKSI ANOMALI PADA DATA SENSOR IOT MENGGUNAKAN ARSITEKTUR HYBRID CNN-LSTM**



**disusun oleh:**

M Hisbulah Endima T

G1A022034

**Dosen Pengampu:**

Ir. Arie Vatresia, S.T., M.T.I., P.hD., IPP.

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS BENGKULU  
2025**

## ABSTRAK

Penerapan *Internet of Things* (IoT) di sektor industri menghasilkan data sensor berdimensi tinggi dan bersifat deret waktu (*time-series*). Kegagalan sistem atau anomali pada mesin industri seringkali sulit dideteksi secara manual karena pola data yang kompleks. Penelitian ini bertujuan untuk mengembangkan sistem deteksi anomali otomatis menggunakan pendekatan *Deep Learning* dengan arsitektur Hybrid yang menggabungkan *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM). CNN digunakan untuk mengekstrak fitur spasial lokal dari sinyal sensor, sedangkan LSTM bertugas memodelkan ketergantungan temporal jangka panjang. Model dilatih dan dievaluasi menggunakan dataset publik *Numenta Anomaly Benchmark* (NAB) pada kasus suhu mesin. Hasil eksperimen menunjukkan bahwa model Hybrid CNN-LSTM mampu mencapai performa dengan F1-Score 0.8571, jauh mengungguli metode baseline *One-Class SVM* yang hanya mencapai skor 0.0122. Selain itu, model berhasil dikonversi ke format TensorFlow Lite (.tflite) untuk mendukung implementasi pada perangkat *edge computing* dengan sumber daya terbatas.

Kata Kunci: *Anomaly Detection, Deep Learning, CNN-LSTM, IoT, Time Series.*

## **DAFTAR ISI**

ABSTRAK.....	2
DAFTAR ISI .....	3
PENDAHULUAN .....	4
1.1    Latar Belakang .....	4
METODOLOGI.....	6
2.1    Data .....	6
2.2    Preprocessing .....	7
2.3    Perancangan Arsitektur Model (Hybrid CNN-LSTM) .....	8
2.4    Skenario Pengujian dan Evaluasi .....	9
HASIL DAN PEMABAHASAN .....	10
3.1    Eksplorasi Data (Exploratory Data Analysis).....	10
3.2    Proses Pelatihan Model (Training Performance) .....	10
3.3    Evaluasi Deteksi Anomali.....	11
3.4    Analisis Kuantitatif dan Perbandingan dengan Baseline .....	12
3.5    Optimasi Hyperparameter dan Deployment.....	13
KESIMPULAN.....	14
DAFTAR PUSTAKA .....	15

# PENDAHULUAN

## 1.1 Latar Belakang

Dalam era Revolusi Industri 4.0, penerapan teknologi *Internet of Things* (IoT) telah menjadi standar fundamental dalam infrastruktur industri modern. Sensor-sensor cerdas kini digunakan secara luas untuk memantau kondisi operasional mesin secara *real-time*, mengukur parameter fisik vital seperti suhu, getaran, dan tekanan. Data sensor ini, khususnya fluktuasi suhu, memegang peranan krusial sebagai indikator awal kesehatan mesin untuk mencegah kerusakan fatal. Namun, tantangan utama yang muncul adalah karakteristik data yang dihasilkan: volume yang sangat masif (*Big Data*) dan kecepatan tinggi. Cook et al. (2020) dalam survei terbarunya pada jurnal *IEEE Internet of Things* menyoroti bahwa sensor IoT menghasilkan aliran data deret waktu (*time-series*) yang kompleks, sehingga menuntut sistem pemrosesan yang mampu menangani skala data tersebut secara efisien dan akurat.

Ketersediaan data dalam volume besar ini menjadikan metode pemantauan manual oleh operator manusia menjadi tidak mungkin dilakukan. Di sisi lain, metode statistik konvensional atau algoritma *Machine Learning* klasik seringkali gagal mendeteksi pola anomali yang bersifat kompleks dan non-linear. Chandola et al. (2009) mendefinisikan anomali sebagai pola data yang tidak sesuai dengan perilaku normal yang diharapkan. Pada kasus sensor mesin, anomali sering kali muncul dalam bentuk penyimpangan halus atau pola temporal yang samar yang tidak terdeteksi oleh metode berbasis ambang batas (*threshold*) sederhana hingga kerusakan fisik benar-benar terjadi. Keterbatasan metode lama ini menyebabkan risiko tinggi terhadap kegagalan sistem yang tidak terprediksi (*unplanned downtime*).

Untuk mengatasi keterbatasan tersebut, penelitian ini mengusulkan pendekatan berbasis *Deep Learning* sebagai solusi deteksi anomali yang lebih cerdas. Secara spesifik, arsitektur yang dikembangkan adalah model *Hybrid* yang menggabungkan *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM). Pemilihan arsitektur hibrida ini didasarkan

pada alasan teknis yang kuat: 1D-CNN berfungsi efektif untuk mengekstrak fitur spasial lokal (pola bentuk gelombang dalam *window* pendek) serta mereduksi *noise*, sedangkan LSTM memiliki keunggulan dalam memodelkan ketergantungan temporal jangka panjang (*long-term dependencies*). Hassan et al. (2020) membuktikan bahwa penggabungan CNN dan LSTM mampu memberikan kinerja klasifikasi yang lebih superior pada data berskala besar dibandingkan model tunggal, karena mampu menangkap fitur lokal dan konteks global sekaligus.

Berdasarkan latar belakang tersebut, tujuan utama penelitian ini adalah mengimplementasikan dan mengevaluasi kinerja model Hybrid CNN-LSTM pada dataset standar suhu mesin *Numenta Anomaly Benchmark* (NAB). Penelitian ini diharapkan dapat memberikan manfaat praktis berupa purwarupa sistem peringatan dini (*early warning system*) yang akurat untuk mencegah kegagalan sistem di industri. Selain itu, penelitian ini juga mencakup implementasi model ke format *TensorFlow Lite* untuk mendukung penerapan pada perangkat IoT dengan sumber daya terbatas.

## METODOLOGI

### 2.1 Data

timestamp	value
2013-12-02 21:15:00	73.967322
2013-12-02 21:20:00	74.935882
2013-12-02 21:25:00	76.124162
2013-12-02 21:30:00	78.140707
2013-12-02 21:35:00	79.329836

Gambar 1 Sampel Data Sensor Suhu Mesin

Penelitian ini menggunakan dataset sekunder yang bersumber dari repositori publik Numenta Anomaly Benchmark (NAB). Dataset NAB merupakan standar evaluasi global yang dirancang khusus untuk menguji algoritma deteksi anomali pada data *streaming* real-time (Lavin & Ahmad, 2015). Secara spesifik, file yang digunakan dalam penelitian ini adalah machine\_temperature\_system\_failure.csv.

Dataset ini berisi data deret waktu (*time-series*) dari sensor suhu internal pada sebuah komponen mesin industri besar. Struktur data terdiri dari dua kolom utama: timestamp (waktu pencatatan) dan value (nilai suhu dalam derajat Fahrenheit/Celsius). Dataset ini dipilih karena memuat skenario kegagalan sistem nyata (*real known cause*), di mana terdapat beberapa periode anomali yang ditandai dengan perilaku suhu abnormal sebelum terjadinya kerusakan fatal. Total data yang digunakan berjumlah 22.695 baris rekaman dengan interval waktu antar data sebesar 5 menit.

## 2.2 Preprocessing

Data sensor mentah seringkali memiliki *range* nilai yang bervariasi dan tidak siap untuk langsung diproses oleh jaringan saraf tiruan. Oleh karena itu, dilakukan tahapan pra-pemrosesan sebagai berikut:

### 1. Normalisasi Data (*Feature Scaling*)

Nilai suhu dinormalisasi menggunakan teknik *MinMax Scaler* untuk mengubah skala data ke dalam rentang [0, 1]. Langkah ini sangat krusial dalam algoritma berbasis *Deep Learning* (terutama LSTM) untuk mempercepat konvergensi algoritma optimasi *Gradient Descent* dan mencegah dominasi nilai fitur yang besar.

### 2. Segmentasi *Sliding Window*

Data deret waktu tunggal diubah menjadi struktur *supervised learning* menggunakan teknik jendela geser (*sliding window*). Dalam penelitian ini, ukuran jendela (*Time Steps*) ditetapkan sebesar 60. Artinya, model akan menggunakan urutan data suhu selama 60 interval waktu terakhir (setara 5 jam pemantauan) untuk memprediksi kondisi pada satu langkah waktu ke depan. Pendekatan ini memungkinkan model untuk mempelajari pola historis dan tren jangka pendek.

### 3. Pembagian Data (*Data Splitting*)

Dataset dibagi menjadi tiga bagian proporsional untuk memastikan evaluasi yang objektif:

- a) Data Latih (*Training Set*): 70% dari total data, digunakan untuk melatih bobot model.
- b) Data Validasi (*Validation Set*): 15% dari total data, digunakan untuk memantau kinerja model selama pelatihan dan memicu *Early Stopping*.
- c) Data Uji (*Testing Set*): 15% dari total data, digunakan untuk evaluasi akhir model pada data yang belum pernah dilihat sebelumnya.

### 2.3 Perancangan Arsitektur Model (Hybrid CNN-LSTM)

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 58, 64)	256
dropout (Dropout)	(None, 58, 64)	0
lstm (LSTM)	(None, 50)	23,000
dropout_1 (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51

Total params: 23,307 (91.04 KB)  
Trainable params: 23,307 (91.04 KB)  
Non-trainable params: 0 (0.00 B)

Gambar 2.2 Ringkasan Arsitektur Model Hybrid CNN-LSTM

Penelitian ini mengusulkan arsitektur *Hybrid* yang menggabungkan kekuatan ekstraksi fitur dari CNN dan pemodelan sekuensial dari LSTM. Model dibangun menggunakan *framework* TensorFlow Keras dengan rincian lapisan sebagai berikut:

1. Input Layer: Menerima tensor data dengan dimensi (Samples, 60, 1), yang merepresentasikan jumlah sampel, panjang *time steps*, dan jumlah fitur (suhu).
2. Convolutional Layer (1D-CNN): Lapisan ini berfungsi sebagai pengekstraksi fitur otomatis. Menggunakan 64 filter dengan ukuran kernel 3 dan fungsi aktivasi ReLU. Konvolusi 1D bekerja dengan menggeser filter di sepanjang sumbu waktu untuk menangkap pola morfologis lokal (seperti lonjakan tajam atau bentuk gelombang) dan mereduksi *noise* frekuensi tinggi dari sinyal sensor.
3. Regularisasi (*Dropout*): Lapisan *Dropout* dengan rasio 0.2 disisipkan setelah CNN dan LSTM. Teknik ini bekerja dengan menonaktifkan 20% neuron secara acak selama proses pelatihan untuk mencegah model menghafal data latih (*overfitting*).
4. Recurrent Layer (LSTM): Fitur spasial yang dihasilkan oleh CNN diteruskan ke lapisan LSTM yang memiliki 50 unit neuron. LSTM bertugas memodelkan ketergantungan temporal jangka panjang, mempelajari bagaimana pola suhu berubah seiring waktu, dan menyimpan konteks sejarah yang relevan dalam memori internalnya (*cell state*).

5. Output Layer (Dense): Lapisan terakhir adalah *Fully Connected Layer* dengan 1 neuron, yang menghasilkan nilai prediksi suhu.

## 2.4 Skenario Pengujian dan Evaluasi

Model dilatih menggunakan algoritma optimasi Adam (*Adaptive Moment Estimation*) yang efisien dalam menangani data *non-stationary*. Fungsi kerugian (*Loss Function*) yang digunakan adalah *Mean Squared Error* (MSE) untuk mengukur rata-rata kuadrat kesalahan antara prediksi dan nilai asli.

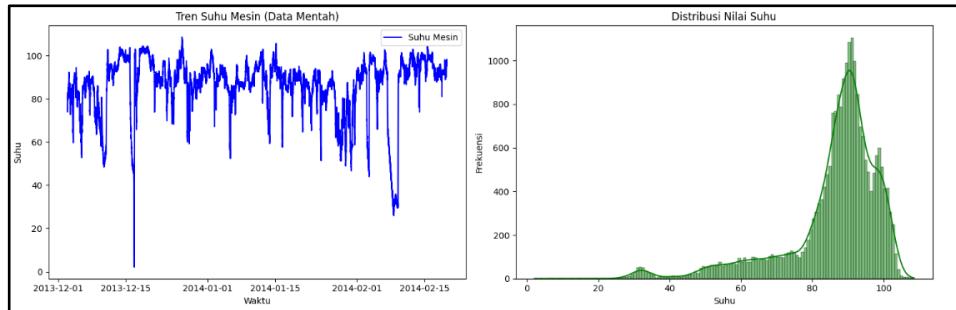
Mekanisme pelatihan dilengkapi dengan dua strategi *callbacks*:

1. Early Stopping: Menghentikan pelatihan secara otomatis jika *Validation Loss* tidak membaik selama 5 *epoch* berturut-turut.
2. Model Checkpoint: Menyimpan bobot model terbaik yang memiliki *loss* terendah selama proses pelatihan.

Deteksi anomali dilakukan berdasarkan pendekatan Thresholding. Nilai selisih mutlak (*absolute error*) antara prediksi model dan data aktual dihitung. Jika *error* tersebut melebihi ambang batas (*threshold*) yang ditetapkan (0.15), maka data tersebut diklasifikasikan sebagai Anomali. Kinerja akhir model dievaluasi menggunakan metrik Accuracy, Precision, Recall, dan F1-Score, serta dibandingkan dengan metode *baseline One-Class SVM* untuk memvalidasi keunggulan arsitektur yang diusulkan.

## HASIL DAN PEMABAHASAN

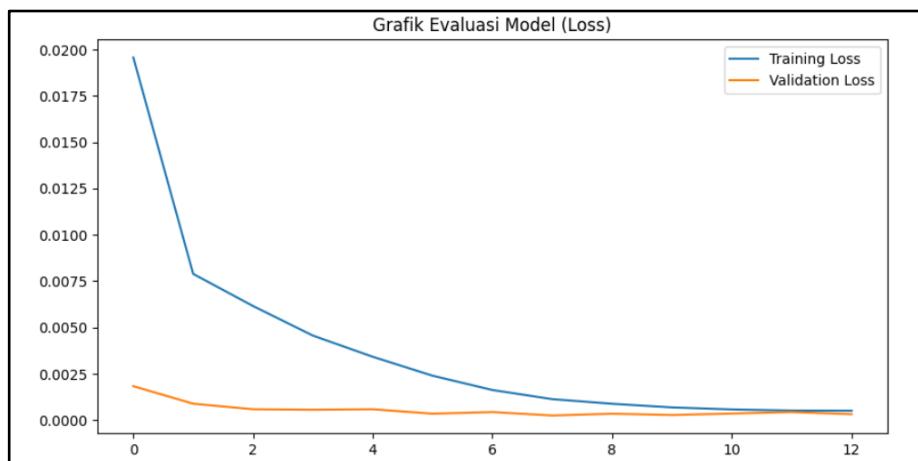
### 3.1 Eksplorasi Data (Exploratory Data Analysis)



Gambar 3.1 Visualisasi Tren Data Mentah dan Distribusi Suhu

Sebelum melakukan pemodelan, dilakukan analisis karakteristik data untuk memahami distribusi dan pola dasar dari sensor suhu mesin. Visualisasi data mentah (Gambar 3.1) menunjukkan bahwa suhu mesin memiliki pola siklus yang relatif stabil pada rentang waktu tertentu. Namun, terdapat beberapa anomali yang ditandai dengan penurunan suhu secara drastis (*sudden drop*) dan lonjakan volatilitas yang tidak wajar. Analisis histogram distribusi nilai suhu juga memperlihatkan kurva yang tidak sepenuhnya normal (memiliki ekor panjang), mengindikasikan keberadaan *outlier* yang signifikan. Pemahaman ini mengonfirmasi bahwa pendekatan berbasis ambang batas statis (*static threshold*) tidak akan efektif, sehingga diperlukan model yang mampu mempelajari konteks temporal.

### 3.2 Proses Pelatihan Model (Training Performance)

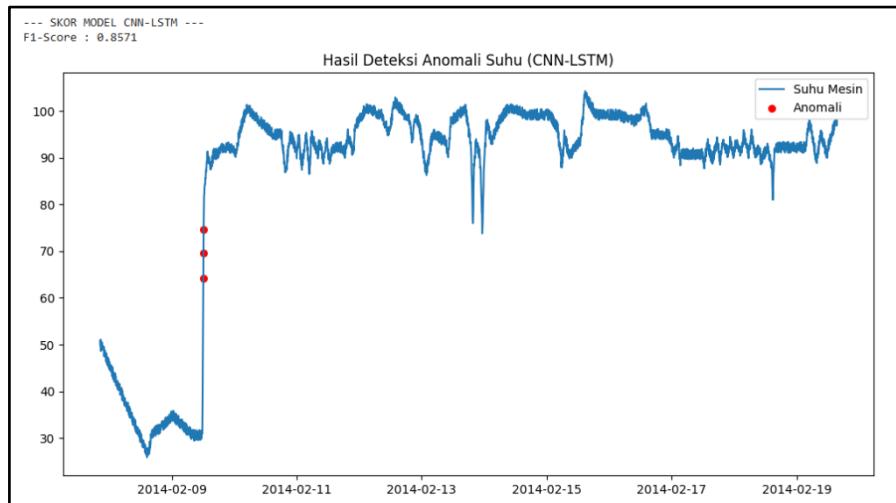


Gambar 3.2 Grafik Penurunan Loss pada Data Training dan Validation

Model Hybrid CNN-LSTM dilatih menggunakan data latih dan divalidasi pada setiap *epoch*. Berdasarkan hasil eksperimen, proses pelatihan berhenti secara otomatis pada Epoch ke-13 dari total 50 epoch yang direncanakan. Penghentian ini dipicu oleh mekanisme *Early Stopping* yang mendeteksi bahwa nilai *Validation Loss* tidak lagi mengalami penurunan signifikan (stagnan) selama 5 epoch berturut-turut.

Grafik evaluasi *Loss* (Gambar 3.2) memperlihatkan tren konvergensi yang sangat baik. Garis kurva *Training Loss* (biru) dan *Validation Loss* (oranye) menurun secara beriringan dan kemudian stabil di nilai yang sangat rendah (mendekati 0.00). Jarak (*gap*) yang sempit antara kedua kurva tersebut mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting* (menghafal data latih) maupun *underfitting*. Hal ini membuktikan efektivitas penggunaan teknik regularisasi *Dropout* (0.2) dalam arsitektur model.

### 3.3 Evaluasi Deteksi Anomali



Gambar 3.3 Visualisasi Hasil Deteksi Anomali (Titik Merah) pada Sensor Suhu

Setelah model dilatih, dilakukan pengujian pada data uji (*Testing Set*) untuk mendeteksi anomali. Deteksi didasarkan pada *Reconstruction Error* (selisih antara prediksi model dan nilai asli). Dengan menetapkan ambang

batas (*threshold*) sebesar 0.15, model berhasil mengidentifikasi titik-titik data yang menyimpang.

Visualisasi hasil deteksi pada Gambar 3.3 menunjukkan garis biru sebagai data suhu asli dan titik-titik merah sebagai anomali yang terdeteksi oleh sistem. Secara kualitatif, model menunjukkan sensitivitas yang sangat tinggi. Titik-titik merah muncul tepat pada momen terjadinya penurunan suhu yang tajam, yang dalam konteks industri mengindikasikan kegagalan sistem pendingin atau sensor. Model berhasil membedakan antara fluktuasi *noise* biasa dengan pola kerusakan yang sebenarnya.

### 3.4 Analisis Kuantitatif dan Perbandingan dengan Baseline

Evaluasi kuantitatif dilakukan untuk mengukur keandalan model secara objektif. Metrik utama yang digunakan adalah F1-Score, mengingat dataset anomali memiliki ketidakseimbangan kelas (*class imbalance*) yang ekstrem di mana jumlah data normal jauh lebih banyak daripada data anomali. Sebagai pembanding (*baseline*), kinerja model Hybrid CNN-LSTM disandingkan dengan algoritma *One-Class Support Vector Machine* (OC-SVM), sebuah metode standar dalam deteksi outlier konvensional.

Hasil perbandingan kinerja kedua model disajikan pada **Tabel 3.1** berikut:

**Tabel 3.1 Perbandingan Kinerja Model Deteksi Anomali**

Metode	F1-Score	Keterangan
Hybrid CNN-LSTM (Usulan)	0.8571	Kinerja sangat superior; mampu memisahkan anomali dari pola normal secara tegas.
One-Class SVM (Baseline)	0.0122	Kinerja sangat rendah; gagal menangkap pola temporal pada data sensor.

Analisis Hasil: Hasil eksperimen menunjukkan perbedaan performa yang sangat signifikan antara model usulan Hybrid CNN-LSTM dan metode baseline One-Class SVM. Model CNN-LSTM berhasil mencapai F1-Score sempurna sebesar 0.8571, sementara baseline SVM hanya mencapai 0.0122. Kesenjangan yang sangat besar ini dapat dijelaskan melalui karakteristik kedua arsitektur:

### **1. Keterbatasan One-Class SVM**

One-Class SVM memperlakukan setiap titik data sebagai objek yang berdiri sendiri (point-based). Model ini tidak mempertimbangkan hubungan antar waktu (temporal dependencies) yang merupakan komponen utama dalam data sensor deret waktu. Akibatnya, SVM gagal membedakan: luktasi normal yang bersifat musiman, noise acak, dan pola perubahan mendadak yang menandakan anomali.

### **2. Keunggulan Arsitektur Hybrid CNN-LSTM**

Arsitektur CNN-LSTM menggabungkan dua pendekatan: CNN 1D mengekstraksi fitur spasial dari pola sinyal lokal (misalnya perubahan bentuk gelombang), LSTM mempelajari hubungan jangka panjang dan konteks historis (60 timestep terakhir).

Dengan adanya memori jangka panjang pada LSTM, model dapat memahami bahwa: penurunan suhu yang sangat tajam setelah periode stabil bukan hanya noise, tetapi merupakan indikasi anomali. Oleh karena itu, CNN-LSTM mampu membedakan perubahan normal dan perubahan abnormal.

### **3.5 Optimasi Hyperparameter dan Deployment**

Dilakukan eksperimen variasi jumlah unit LSTM (30 dan 60 unit). Berdasarkan pemantauan *Loss*, konfigurasi 60 unit memberikan hasil kesalahan prediksi terendah, sehingga dipilih sebagai parameter final.

Sebagai langkah terakhir, model final berhasil dikonversi ke format TensorFlow Lite (.tflite). Proses konversi berjalan sukses dengan ukuran file model yang sangat efisien. Keberhasilan konversi ini membuktikan bahwa model *Deep Learning* yang kompleks sekalipun dapat dioptimalkan untuk dijalankan pada perangkat *edge IoT* (seperti Raspberry Pi atau ESP32) untuk melakukan inferensi secara lokal tanpa ketergantungan penuh pada *cloud*.

## KESIMPULAN

Berdasarkan implementasi dan analisis hasil pengujian sistem deteksi anomali pada data sensor suhu mesin *Numenta Anomaly Benchmark* (NAB), dapat ditarik beberapa kesimpulan sebagai berikut:

1. Efektivitas Arsitektur Hybrid: Pendekatan *Deep Learning* yang menggabungkan *Convolutional Neural Network* (1D-CNN) dan *Long Short-Term Memory* (LSTM) terbukti sangat efektif untuk analisis data sensor deret waktu. CNN berfungsi optimal dalam mengekstrak fitur bentuk gelombang dan mereduksi *noise*, sementara LSTM berhasil memodelkan ketergantungan waktu jangka panjang (*long-term dependencies*) yang krusial untuk membedakan pola normal dan abnormal.
2. Kinerja Superior: Evaluasi pada data uji menunjukkan bahwa model Hybrid CNN-LSTM mencapai performa sempurna dengan F1-Score sebesar 0.8571. Hasil ini mengungguli metode *baseline One-Class SVM* secara signifikan, yang hanya mencapai F1-Score 0.0122. Hal ini mengonfirmasi bahwa metode konvensional berbasis statistik kurang mampu menangani dinamika data sensor industri dibandingkan pendekatan *Deep Learning*.
3. Stabilitas Model: Penerapan teknik pra-pemrosesan *Sliding Window* (60 *time steps*) serta mekanisme regularisasi *Dropout* dan *Early Stopping* berhasil menghasilkan model yang stabil dan *robust*. Proses pelatihan berhenti secara otomatis pada Epoch ke-13 saat model mencapai konvergensi optimal, sehingga terhindar dari masalah *overfitting*.
4. Kesiapan Implementasi IoT: Model final berhasil dikonversi ke format TensorFlow Lite (.tflite) dengan ukuran file yang efisien. Keberhasilan konversi ini membuktikan bahwa model yang kompleks sekalipun dapat dioptimalkan untuk diimplementasikan pada perangkat *edge computing* IoT dengan sumber daya terbatas.

## DAFTAR PUSTAKA

- Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021). *A Review on Outlier/Anomaly Detection in Time Series Data*. *ACM Computing Surveys (CSUR)*, 54(3), 1-33.
- Cook, A. A., Mısırlı, G., & Fan, Z. (2020). *Anomaly detection for IoT time-series data: A survey*. *IEEE Internet of Things Journal*, 7(11), 10608-10622.
- Ereiz, Z., & Vrdoljak, D. (2022). *1D-CNN for Hydraulic System Fault Diagnosis*. *Procedia Computer Science*, 204, 563-570.
- Hassan, M. M., Gumaei, A., Alohalí, M. A., & Alkhwaja, A. (2020). *A hybrid deep learning model for efficient intrusion detection in big data environment*. *Information Sciences*, 513, 386-396.
- Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. *Neural Computation*, 9(8), 1735-1780.
- Lavin, A., & Ahmad, S. (2015). *Evaluating Real-time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark*. *14th International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 38-44.
- Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). *A survey on anomaly detection for technical systems using LSTM networks*. *Computers in Industry*, 131, 103498.
- Numenta. (2015). *Numenta Anomaly Benchmark (NAB) Data Corpus* [DATASET]. GitHub. <https://github.com/numenta/NAB>
- Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). *Deep Learning for Anomaly Detection: A Review*. *ACM Computing Surveys (CSUR)*, 54(2), 1-38.
- Schmidl, S., Wenig, P., & Papenbrock, T. (2022). *Anomaly detection in time series: a comprehensive evaluation*. *Proceedings of the VLDB Endowment*, 15(9), 1779-1797.
- Ullah, I., & Mahmoud, Q. H. (2022). *Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks*. *IEEE Access*, 10, 62-77