

## Deteksi Anomali Sinyal Vibrasi pada Mesin Industri Menggunakan Autoencoder di PT. Pusri Palembang

Zainal Umari<sup>1</sup>, Julian Supardi<sup>\*2</sup>

<sup>1,2</sup>Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya, Indonesia  
Email: <sup>1</sup>[09012682327002@student.unsri.ac.id](mailto:09012682327002@student.unsri.ac.id), <sup>2</sup>[julian@unsri.ac.id](mailto:julian@unsri.ac.id)

### Abstrak

Analisa sinyal vibrasi merupakan metode penting untuk mendeteksi kondisi kesehatan mesin untuk mencegah kerusakan lebih lanjut dan mengurangi *downtime*. Pendekatan konvensional sering kali tidak efektif dalam menangkap pola kompleks pada sinyal vibrasi, sehingga dibutuhkan metode yang lebih canggih, seperti pembelajaran mesin. Penelitian ini bertujuan untuk mengembangkan model deteksi anomali pada sinyal vibrasi menggunakan *autoencoder*. Data vibrasi yang digunakan diperoleh dari tiga unit blower di PT. Pusri Palembang. Evaluasi dilakukan dengan membandingkan kemampuan model dalam mendeteksi anomali dan kondisi normal. Hasil penelitian menunjukkan bahwa model *autoencoder* efektif mendeteksi anomali dengan akurasi tinggi serta keseimbangan optimal dalam mengidentifikasi kondisi mesin. Penelitian ini menawarkan solusi praktis bagi industri untuk meningkatkan efisiensi pemeliharaan dan keandalan peralatan.

**Kata kunci:** *autoencoder, deteksi anomali, pembelajaran mesin, sinyal vibrasi*

## *Anomaly Detection of Industrial Machine Vibration Signals Using Autoencoder at PT. Pusri Palembang*

### *Abstract*

*Vibration signal analysis is an essential method for detecting machine health conditions to prevent further damage and reduce downtime. Conventional approaches often fail to capture the complex patterns in vibration signals, necessitating more advanced methods such as machine learning. This study aims to develop an anomaly detection model for vibration signals using an autoencoder. The vibration data used were obtained from three blower units at PT. Pusri Palembang. The evaluation was conducted by comparing its performance in detecting anomalies and normal conditions. The results of the study indicate that the autoencoder model is effective in detecting anomalies with high accuracy and optimal balance in identifying machine conditions. This research offers a practical solution for industries to improve maintenance efficiency and equipment reliability.*

**Keywords:** *anomaly detection, autoencoder, machine learning, vibration signal*

## 1. PENDAHULUAN

Analisa sinyal vibrasi merupakan salah satu metode yang dapat diandalkan untuk memantau kondisi kesehatan mesin dalam berbagai industri, seperti manufaktur, pembangkit listrik, dan proses kimia. Perubahan pada sinyal vibrasi dapat mengindikasikan adanya anomali, seperti kerusakan mekanis atau penurunan kinerja komponen [1]. Deteksi dini anomali sangat penting untuk mencegah kegagalan mesin yang lebih serius dan meminimalkan waktu henti operasional (*downtime*) [2]. Namun, metode konvensional seperti analisis spektral sering kali memiliki keterbatasan dalam menangkap pola kompleks pada sinyal vibrasi, terutama ketika sinyal tersebut memiliki dimensi yang tinggi dan tidak berlabel, sehingga menyulitkan proses analisis secara manual [3], [4].

Berbagai pendekatan telah dikembangkan untuk meningkatkan efektivitas deteksi anomali sinyal vibrasi. Zhou dkk. memanfaatkan metode *transformer* berbasis *self-supervised learning* pada data getaran tanpa label dan menunjukkan performa lebih unggul dibandingkan model konvensional [5]. Sangeetha dkk. mengusulkan model hibrida yang menggabungkan *Support Vector Machine* (SVM) dan *Extreme Gradient Boosting* (XGBoost), mencapai akurasi 94% berdasarkan standar ISO 10816 [6]. Almutairi dkk. menggunakan spektrum komposit poli-koheren untuk menyederhanakan analisis data dan meningkatkan deteksi pada berbagai kecepatan operasional [7]. Sementara itu, Styanidis dkk. menerapkan *autoencoder* untuk meningkatkan akurasi deteksi anomali pada mesin berputar pada radar cuaca [8]. *Autoencoder* merupakan jaringan saraf tiruan yang dirancang

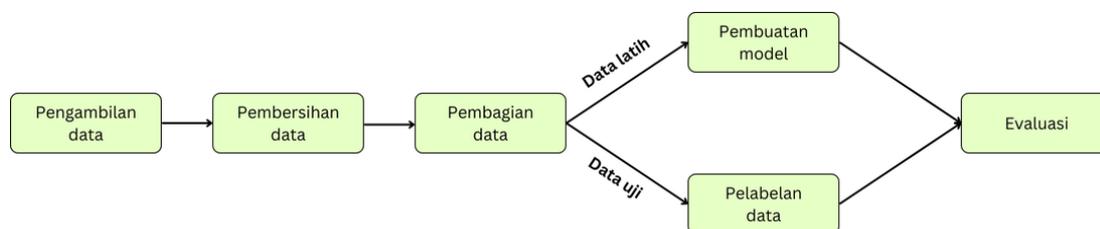
untuk mempelajari representasi tersembunyi dari data [9]. *Autoencoder* efektif dalam menangkap pola nonlinier dan hubungan kompleks dalam data, yang biasanya sulit ditangkap oleh metode konvensional [10]. Studi-studi ini menunjukkan potensi besar teknik *autoencoder* dan pemrosesan sinyal dalam mendeteksi anomali pada mesin industri.

Meskipun berbagai metode seperti *transformer*, model hibrida SVM dan XGBoost, serta spektrum komposit poli-koheren telah dikembangkan untuk deteksi anomali sinyal vibrasi, setiap pendekatan masih memiliki keterbatasan dalam penerapannya. Metode *transformer* dan spektrum komposit, misalnya, memiliki kompleksitas perhitungan yang tinggi, sehingga kurang efisien untuk diterapkan pada skala industri besar. Sementara itu, model hibrida membutuhkan optimasi parameter yang rumit, yang menyulitkan implementasi di lingkungan industri yang dinamis. Di sisi lain, meskipun *autoencoder* telah terbukti efektif dalam mendeteksi anomali pada mesin berputar di sistem radar cuaca, penerapannya pada sinyal vibrasi mesin berputar dalam industri masih memerlukan eksplorasi lebih lanjut. Oleh karena itu, diperlukan pengembangan model deteksi anomali yang lebih efektif dan efisien untuk menangkap pola non-linier dan kompleks dalam sinyal vibrasi mesin industri.

Penelitian ini bertujuan untuk mengembangkan model deteksi anomali berbasis *autoencoder* pada sinyal vibrasi mesin berputar di lingkungan industri. Model ini dirancang untuk mampu menangkap pola non-linier dan kompleks pada sinyal vibrasi serta beradaptasi terhadap variasi kondisi operasional, seperti perubahan beban dan tingkat *noise*. Evaluasi kinerja model akan dilakukan menggunakan metrik standar seperti *accuracy*, *precision*, *recall*, dan *F1-score* untuk memastikan efektivitas dan efisiensi deteksi anomali. Dengan pengembangan ini, diharapkan model yang dihasilkan dapat menjadi solusi praktis dalam mendukung pemeliharaan prediktif mesin industri dan meningkatkan efisiensi operasional serta mengurangi risiko *downtime*.

## 2. METODE PENELITIAN

Metode penelitian ini mencakup beberapa tahapan utama seperti ditunjukkan pada gambar 1. Dimulai dengan pengambilan data dan pembersihan data. Kemudian data dibagi menjadi data latih dan data uji, data latih digunakan untuk pembuatan model, sedangkan data uji melalui tahap pelabelan data, yang pada akhirnya digunakan untuk mengevaluasi model.



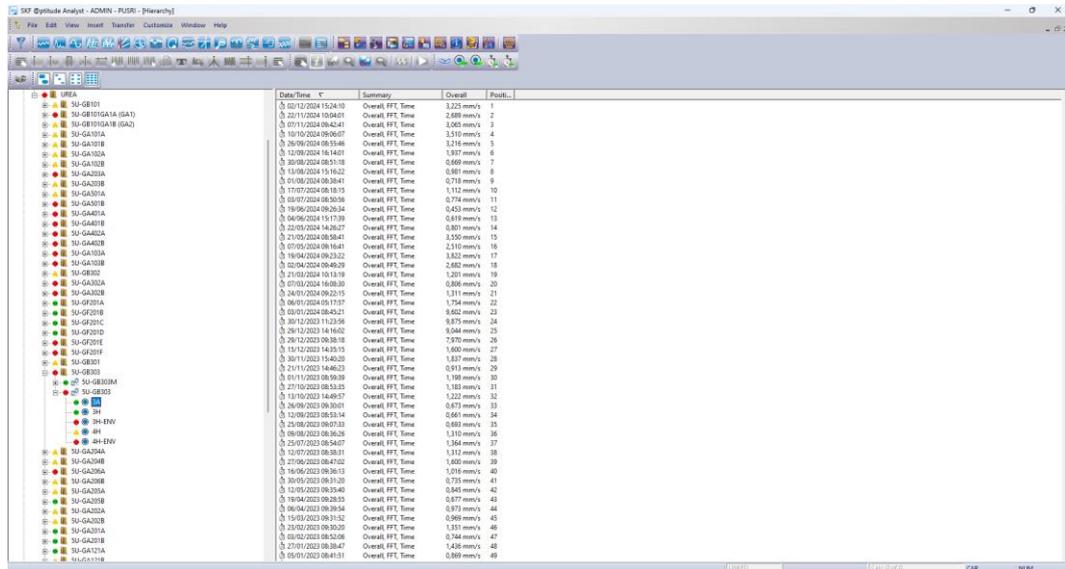
Gambar 1. Alur Tahapan Penelitian

### 2.1. Pengambilan Data

Penelitian ini menggunakan data yang diperoleh dari Departemen Keandalan Pabrik PT. Pusri Palembang. Data tersebut merupakan hasil pengukuran vibrasi pada tiga unit blower identik dengan *tag number* GB303, yang digunakan di Pusri 1B, Pusri 3, dan Pusri 4. Pengukuran secara rutin dilakukan menggunakan alat SKF Microlog Analyzer GX (gambar 2). Hasil pengukuran kemudian disimpan dalam aplikasi berbasis *desktop* bernama SKF @ptitude Analyst (gambar 3).

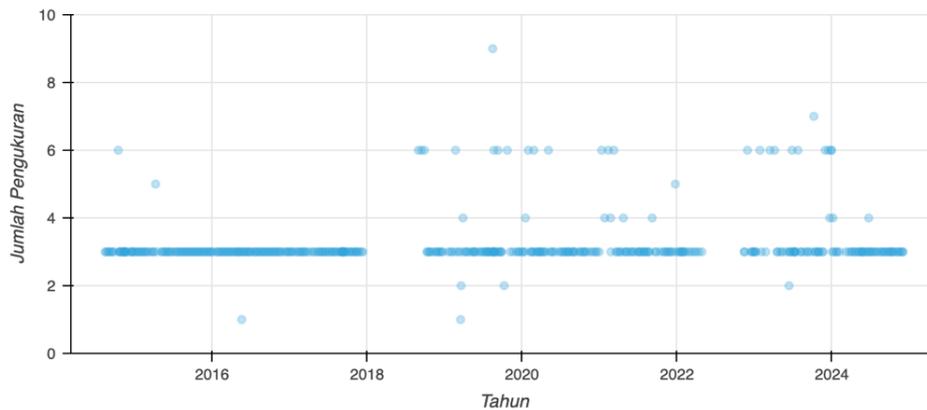


Gambar 2. SKF Microlog Analyzer GX



Gambar 3. SKF @ptitude Analyst

Pengukuran vibrasi dilakukan secara rutin sejak tahun 2014 dengan jumlah data sebanyak 1375 dan dimensi data sebesar 1601. Distribusi pengukuran berdasarkan waktu dapat dilihat pada gambar 4 dengan mayoritas pengukuran sebanyak 3 titik per hari sementara pengukuran yang lebih atau kurang dari 3 titik cenderung bersifat non rutin dan memerlukan pengecekan ekstra.



Gambar 4. Distribusi data terhadap waktu

**2.2. Pembersihan Data**

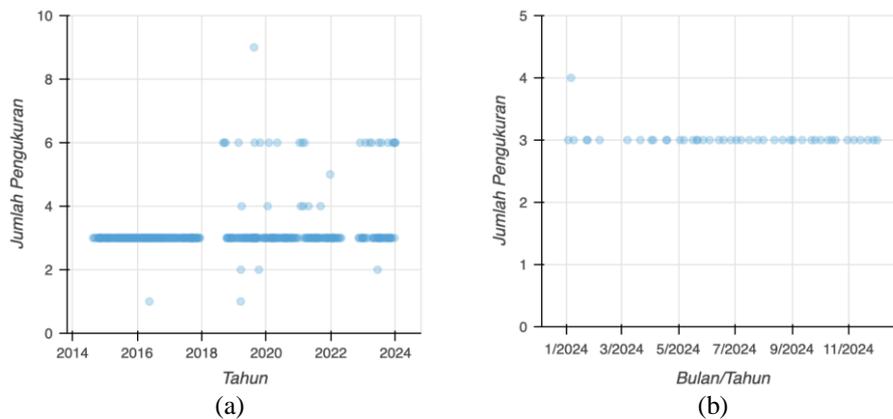
Pembersihan data merupakan langkah penting untuk memastikan data yang digunakan bebas dari *noise*, inkonsistensi, dan kesalahan yang dapat memengaruhi hasil analisis atau performa model. Proses ini bertujuan untuk menghilangkan ketidaksesuaian pada data pengukuran, yang jika dibiarkan dapat mempengaruhi hasil analisis dan menghasilkan kesimpulan yang tidak akurat [11].

Proses pembersihan terdiri dari menghapus data yang dimensi datanya tidak sampai 1601. Pembersihan ini menyebabkan pengurangan jumlah data sebanyak 47, dari 1375 menjadi 1328. Pengurangan ini relatif kecil dibandingkan dengan keseluruhan data, sehingga tidak memengaruhi representativitas data secara signifikan.

**2.3. Pembagian Data**

Pada tahap ini, data dibagi menjadi dua set utama, yaitu data latih dan data uji. Pembagian dilakukan berdasarkan tahun pengambilan data. Pendekatan berbasis waktu ini bertujuan untuk mensimulasikan deteksi anomali pada kondisi yang akan datang. Data latih terdiri dari data yang diambil sebelum tahun 2024 sebanyak 1195 data, yang digunakan untuk melatih model dalam mendeteksi anomali. Sedangkan data uji terdiri dari data yang diambil pada tahun 2024 sebanyak 133 data, yang digunakan untuk menguji performa model dalam

mendeteksi anomali. Distribusi setelah pembagian terhadap waktu dapat dilihat pada gambar 5, dimana gambar 5(a) adalah data latih dan gambar 5(b) adalah data uji.



Gambar 5. Distribusi data terhadap waktu (a) data latih (b) data uji.

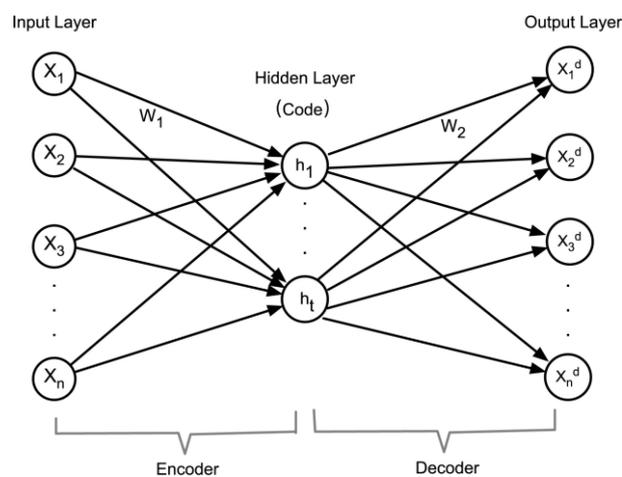
## 2.4. Pembuatan Model

### 2.4.1. Arsitektur *Autoencoder*

Seluruh penjelasan dalam sub-bab ini merujuk pada [12] dan [13]. *Autoencoder* merupakan model yang efektif untuk mempelajari dan menangkap karakteristik penting dari data. *Autoencoder* adalah jaringan saraf tiruan yang dirancang untuk merekonstruksi kembali sinyal input ke bentuk aslinya. Karena proses pelatihan *autoencoder* tidak memerlukan label data, model ini termasuk dalam metode pembelajaran tanpa pengawasan (*unsupervised learning*).

Struktur *autoencoder* umumnya terdiri dari dua bagian utama:

1. *Encoder*, yang berfungsi untuk mempelajari karakteristik penting dari data input dengan mereduksi dimensi dan memetakan sinyal input ke ruang laten agar dapat direpresentasikan dalam bentuk yang lebih representatif.
2. *Decoder*, yang berfungsi untuk mengembalikan sinyal yang telah dikonversi ke ruang laten kembali ke ruang aslinya dan merekonstruksi data dalam bentuk semula.



Gambar 6. Arsitektur sederhana *autoencoder*

Gambar 6 menampilkan arsitektur sederhana *autoencoder* yang terdiri dari tiga lapisan utama yaitu lapisan input, lapisan tersembunyi, dan lapisan output. Jumlah neuron di setiap lapisan masing-masing adalah  $n$ ,  $t$ , dan  $n$ . Jumlah neuron pada lapisan input sama dengan jumlah di lapisan output, sedangkan jumlah neuron pada lapisan tersembunyi dapat bervariasi. Biasanya, jumlah neuron di lapisan tersembunyi lebih sedikit dibandingkan dengan lapisan input ( $t < n$ ), yang bertujuan untuk mengurangi dimensi data.

Lapisan input dan lapisan tersembunyi berperan sebagai *encoder*, sedangkan lapisan tersembunyi dan lapisan output berperan sebagai *decoder*. Data asli  $X$  dimasukkan ke dalam model, kemudian menghasilkan data

$h$  yang diperoleh setelah tahap *encoding*. Proses *encoding* dijelaskan dengan persamaan (1), di mana  $W_1$  merupakan matriks bobot antara lapisan input dan lapisan tersembunyi, dan  $b_1$  adalah vektor bias.  $f(x)$  adalah fungsi aktivasi yang digunakan untuk transformasi non-linier.

$$h = f(x) = f(W_1X + b_1) \tag{1}$$

Proses *decoding* bertujuan untuk mengembalikan data  $h$  ke bentuk asli dan merekonstruksi  $X^d$ . Proses decoding dilakukan dengan persamaan (2), di mana  $W_2$  adalah matriks bobot antara lapisan tersembunyi dan lapisan output, dan  $b_2$  adalah vektor bias. Fungsi  $g(x)$  dapat berupa transformasi non-linier atau fungsi transformasi *afine*. Beberapa fungsi aktivasi non-linier yang umum digunakan adalah fungsi sigmoid dan fungsi tanh.

$$X^d = g(x) = g(W_2h + b_2) \tag{2}$$

*Autoencoder* dilatih untuk menemukan parameter (matriks bobot  $W$  dan vektor bias  $b$ , yang diwakili oleh  $\theta$ ) dengan meminimalkan kesalahan rekonstruksi antara  $X^d$  dan  $X$ . Untuk regresi, kesalahan rekonstruksi  $L$  dan fungsi kerugian  $J_{AE}(\theta)$  yang sering digunakan adalah *square error*, seperti pada persamaan (3) dan (4).

$$L(X, X^d) = \| x^d - x \|^2 \tag{3}$$

$$J_{AE}(\theta) = \frac{1}{n} \sum_{i=1}^n \| x_i^d - x_i \|^2 \tag{4}$$

Sedangkan untuk klasifikasi kesalahan rekonstruksi  $L$  dan fungsi kerugian  $J_{AE}(\theta)$  yang sering digunakan yaitu *cross-entropy*, seperti pada persamaan (5) dan (6).

$$L(X, X^d) = - (X \log X^d + (1 - X) \log(1 - X^d)) \tag{5}$$

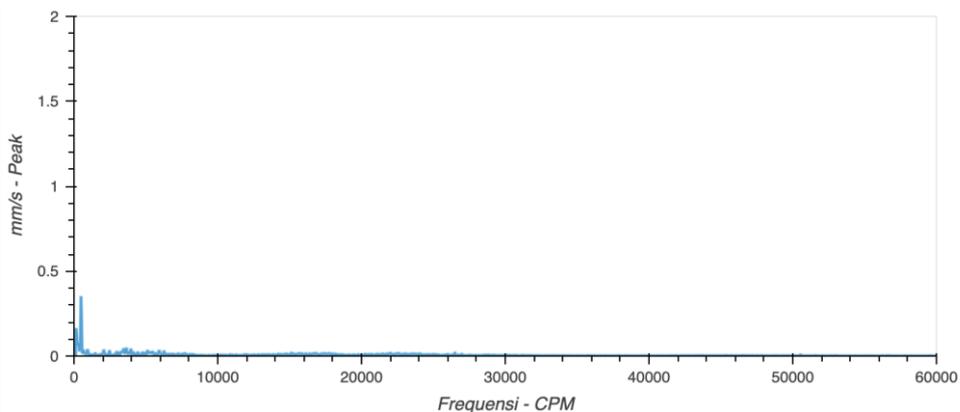
$$J_{AE}(\theta) = - \sum_{i=1}^n (x_i \log x_i^d + (1 - x_i) \log(1 - x_i^d)) \tag{6}$$

di mana input data  $X = \{x_1, x_2, \dots, x_n\}$ , dan rekonstruksi  $X^d = \{x_1^d, x_2^d, \dots, x_n^d\}$ .

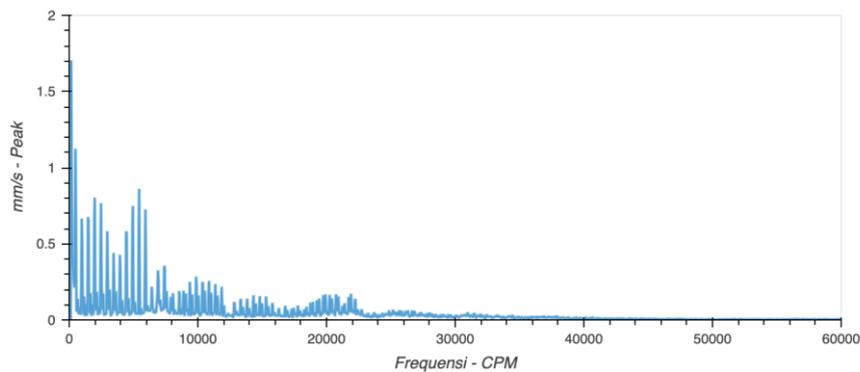
**2.4.2. Pemilihan Hyperparameter**

Kingma (2014) menunjukkan bahwa Adam optimizer memiliki keunggulan dalam mencapai konvergensi yang cepat dan stabil pada model deep learning [14]. Optimizer ini dipilih karena sifat adaptifnya yang mampu menangani gradien tidak stabil dan parameter kompleks dengan efisien, sehingga cocok untuk digunakan pada model *autoencoder*. *Learning rate* 0.001 merupakan nilai default yang secara empiris terbukti optimal untuk berbagai arsitektur jaringan saraf [15].

**2.5. Pelabelan Data**



Gambar 7. Sampel data normal



Gambar 8. Sampel data anomali

Pelabelan data uji dilakukan oleh seorang ahli vibrasi untuk memberikan label pada setiap sampel, apakah anomali atau tidak, dengan tujuan untuk mengevaluasi performa model deteksi anomali. Proses ini dilakukan secara manual dengan memperhatikan pola kelainan yang menunjukkan kerusakan atau gangguan pada peralatan. Dari 133 data uji yang diberi label, terdapat 97 data normal dan 36 data anomali. Sampel data normal menunjukkan pola vibrasi tanpa gangguan (gambar 7), sementara data anomali menunjukkan indikasi kerusakan (gambar 8).

### 2.6. Evaluasi

Evaluasi model deteksi anomali dilakukan untuk mengukur seberapa baik model membedakan antara data normal dan anomali. Berbagai metrik digunakan untuk menilai performa model, di antaranya adalah *confusion matrix*, *accuracy*, *precision*, *recall*, dan *F1-score*, yang memberikan gambaran menyeluruh tentang kemampuan model dalam mendeteksi anomali dan meminimalkan kesalahan klasifikasi.

*Confusion matrix* adalah tabel yang menggambarkan performa model klasifikasi dengan mengklasifikasikan hasil prediksi ke dalam empat kategori [16]:

- *True positive* (TP): Data anomali yang terdeteksi dengan benar sebagai anomali.
- *True negative* (TN): Data normal yang terdeteksi dengan benar sebagai normal.
- *False positive* (FP): Data normal yang salah terdeteksi sebagai anomali.
- *False negative* (FN): Data anomali yang salah terdeteksi sebagai normal.

*Accuracy* mengukur seberapa sering model menghasilkan prediksi yang benar secara keseluruhan. Nilai *accuracy* dihitung dengan persamaan (7) [17].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

*Precision* mengukur ketepatan model dalam mengklasifikasikan data sebagai anomali. *Precision* yang tinggi berarti model jarang salah mengklasifikasikan data normal sebagai anomali. *Precision* dihitung dengan persamaan (8).

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

*Recall*, atau sensitivitas, mengukur kemampuan model dalam mendeteksi anomali yang sebenarnya ada. *Recall* yang tinggi menunjukkan model mampu mendeteksi sebagian besar anomali. *Recall* dihitung dengan persamaan (9).

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

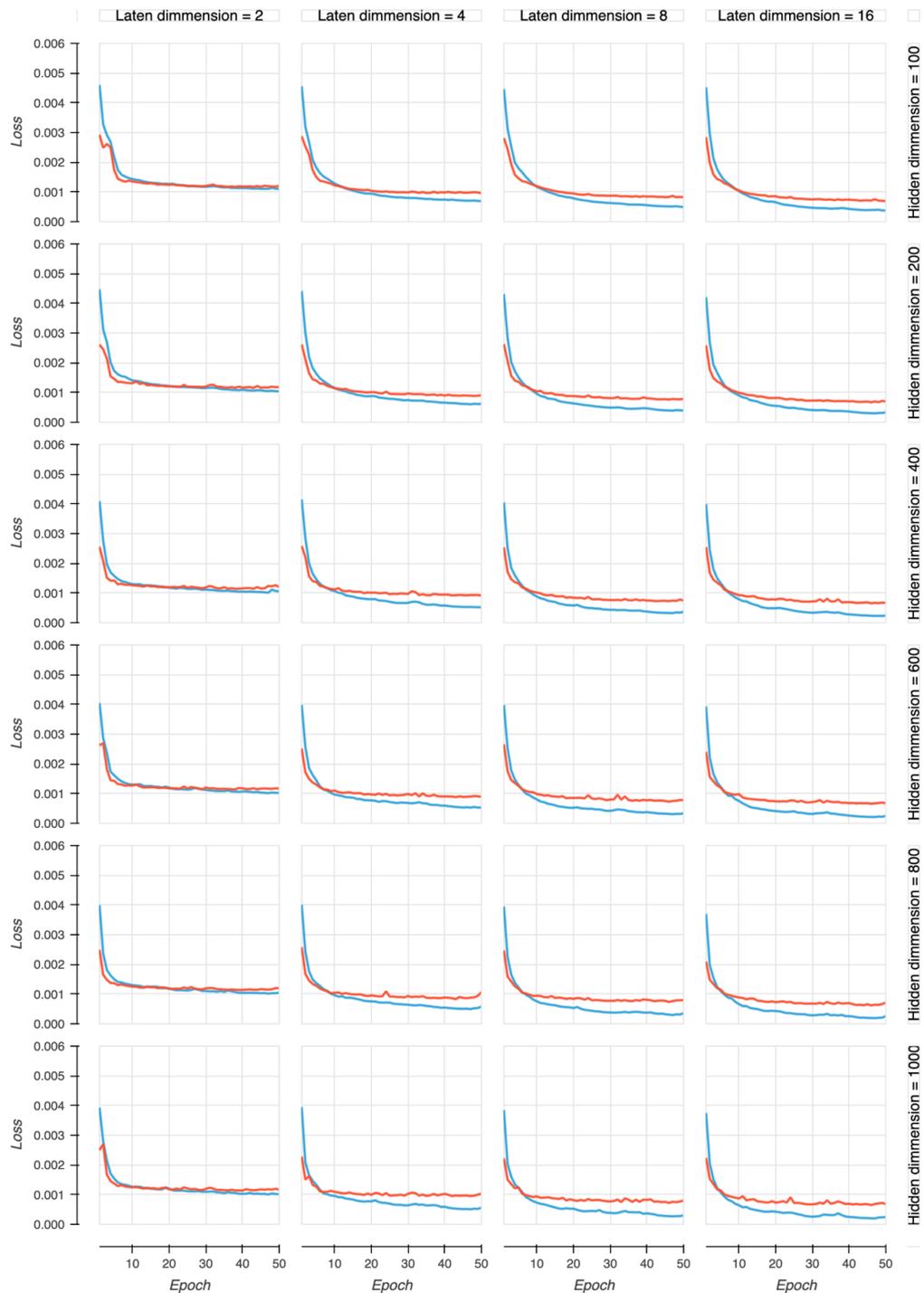
*F1-score* adalah rata-rata harmonik antara *precision* dan *recall*, memberikan keseimbangan antara keduanya, terutama ketika terdapat ketidakseimbangan antara kelas normal dan anomali. *F1-score* yang tinggi menunjukkan keseimbangan yang baik antara mendeteksi anomali dan meminimalkan kesalahan klasifikasi. *F1-score* dihitung dengan persamaan (10).

$$"F1 - score" = \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pembuatan Model

Arsitektur model dirancang dengan mengkombinasikan berbagai jumlah dimensi *hidden layer* dan dimensi ruang laten. Seperti dijelaskan pada subbab 2.4.1, proses *training* dilakukan menggunakan *optimizer* Adam dengan *learning rate* sebesar 0.001. Jumlah *epoch* yang digunakan adalah 50, karena pada titik tersebut nilai *loss* telah menunjukkan kestabilan. Sementara itu, *loss function* yang diterapkan adalah *mean square error* (MSE).



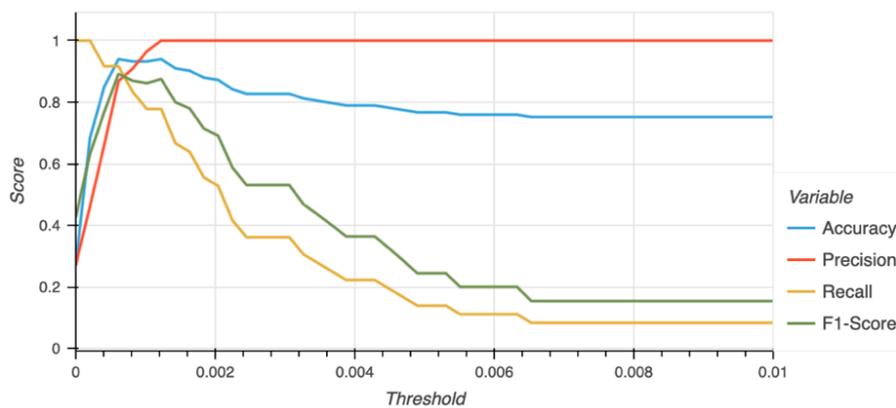
Gambar 9. *Loss function* dari data latih dan data uji pada berbagai kombinasi dimensi *hidden layer* dan dimensi ruang laten

Hasil pengujian kombinasi arsitektur ditunjukkan pada gambar 9. Secara umum, semua kombinasi menunjukkan pola penurunan *loss function* yang konsisten seiring bertambahnya *epoch*. Dimensi laten yang lebih tinggi, khususnya pada nilai 16, cenderung menghasilkan kurva *loss* yang lebih stabil dengan nilai *loss* yang lebih rendah dibandingkan dimensi laten yang lebih kecil seperti 2 atau 4. Selain itu, jumlah dimensi *hidden layer* juga memengaruhi performa model.

Kombinasi dimensi *hidden layer* yang lebih sederhana, seperti pada nilai 100 atau 200, menghasilkan grafik *loss* yang lebih konsisten tanpa tanda-tanda *overfitting*. Sebaliknya, arsitektur dengan dimensi *hidden layer* lebih besar dari 800 menunjukkan fluktuasi kecil pada data uji, terutama di *epoch* akhir, yang mengindikasikan potensi *overfitting*. Berdasarkan analisis ini, kombinasi terbaik adalah arsitektur dengan dimensi laten 16 dan *hidden dimension* 100, karena memberikan stabilitas grafik *loss* serta nilai *loss* yang rendah untuk data latih dan data uji.

### 3.2. Evaluasi

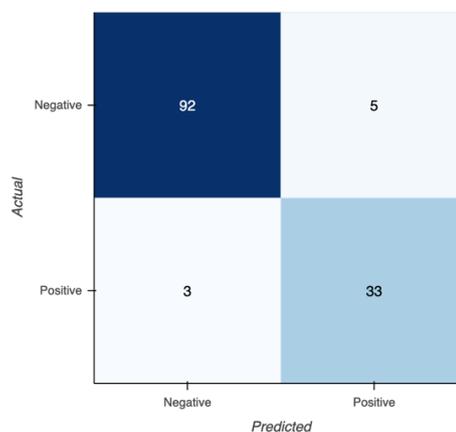
Pada tahap evaluasi model, pemilihan *threshold* yang optimal sangat penting untuk menentukan kinerja model dalam mendeteksi anomali. *Threshold* mengacu pada nilai ambang batas probabilitas yang digunakan untuk mengklasifikasikan data sebagai normal atau anomali. Untuk menemukan *threshold* yang paling tepat, dilakukan evaluasi berdasarkan beberapa metrik, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*, yang dihitung pada berbagai nilai *threshold* yang bervariasi dari 0 hingga 0.01.



Gambar 10. Perbandingan nilai *accuracy*, *precision*, *recall*, dan *F1-score* terhadap *threshold*

Gambar 10 menunjukkan bahwa *threshold* berperan penting dalam mempengaruhi keseimbangan antara *precision* dan *recall*. *Threshold* optimal untuk mencapai performa seimbang adalah sekitar 0.0006, di mana *F1-score* berada pada nilai tertinggi. Oleh karena itu, nilai *threshold* ini yang akan digunakan untuk menentukan apakah data termasuk anomali atau tidak.

Selanjutnya, untuk mengevaluasi performa model secara lebih mendalam, dilakukan analisis menggunakan *confusion matrix* untuk memperlihatkan jumlah prediksi yang benar dan salah pada masing-masing kelas (normal dan anomali).



Gambar 11. *Confusion matrix* hasil evaluasi model pada *threshold* 0.0006

Berdasarkan *confusion matrix* pada gambar 11, performa model klasifikasi menunjukkan hasil yang baik. Dari total 133 data, model berhasil mengklasifikasikan 92 data negatif dengan benar (*true negative*) dan 33 data positif dengan benar (*true positive*). Namun, terdapat 5 data yang salah diprediksi sebagai positif padahal sebenarnya negatif (*false positive*) dan 3 data yang salah diprediksi sebagai negatif padahal sebenarnya positif (*false negative*).

Menggunakan rumus yang dijelaskan pada subbab 2.6, diperoleh nilai *accuracy* sebesar 94% menunjukkan bahwa model mampu memprediksi dengan tepat sebagian besar data. Nilai *precision* sebesar 87% berarti bahwa semua prediksi positif yang diberikan, 87% di antaranya benar. Nilai *recall* sebesar 92% mengindikasikan bahwa model berhasil mendeteksi 92% dari semua kasus positif yang sebenarnya. Nilai *F1-score* sebesar 89% menunjukkan keseimbangan yang baik antara *precision* dan *recall*.

### 3.3. Diskusi

Model memiliki performa yang sangat baik sebagaimana ditunjukkan oleh metrik pada subbab sebelumnya. Tingginya *recall* menunjukkan kemampuan model dalam mendeteksi kasus positif secara efektif, yang menjadikannya sangat berguna untuk sistem pemantauan atau peringatan dini yang memprioritaskan deteksi anomali. Namun, adanya *false positive* menunjukkan bahwa masih ada ruang untuk perbaikan, terutama untuk meningkatkan akurasi prediksi positif sehingga hasil yang diberikan lebih akurat dan andal.

Keunggulan ini dapat dikaitkan dengan kemampuan *autoencoder* dalam menangkap pola non-linier dan kompleks pada sinyal vibrasi berdimensi tinggi dan tidak berlabel. *Autoencoder* mampu mempelajari representasi tersembunyi dari data secara otomatis, sehingga lebih efektif dalam mengenali perbedaan signifikan antara kondisi normal dan anomali. Hal ini membuktikan bahwa pendekatan unsupervised learning yang digunakan *autoencoder* dapat mengatasi keterbatasan dalam menangani data tanpa label yang sering ditemukan di industri.

Jika dibandingkan dengan penelitian sebelumnya, hasil ini konsisten dengan studi Styanidis dkk. [9] yang menerapkan *autoencoder* untuk mendeteksi anomali pada radar cuaca. Selain itu, model ini juga lebih efisien dibandingkan pendekatan berbasis *transformer* yang memiliki kompleksitas komputasi tinggi atau model hibrida SVM dan XGBoost yang memerlukan optimasi parameter yang rumit.

Penelitian ini menunjukkan bahwa *autoencoder* merupakan solusi yang efektif dan efisien untuk deteksi anomali sinyal vibrasi mesin industri. Dengan *accuracy* sebesar 94%, model ini berpotensi meningkatkan efektivitas deteksi anomali mesin industri dan mengurangi kerugian akibat *downtime*.

## 4. KESIMPULAN

Penelitian ini berhasil mengembangkan model deteksi anomali pada sinyal vibrasi menggunakan pendekatan *autoencoder*. Hasil pengujian menunjukkan bahwa dimensi laten yang lebih tinggi mampu menghasilkan kurva *loss function* yang lebih stabil dengan nilai *loss* yang lebih rendah. Selain itu, jumlah dimensi hidden layer juga memengaruhi performa model. Arsitektur sederhana, seperti pada hidden layer dengan dimensi 100, menunjukkan grafik *loss* yang lebih konsisten tanpa indikasi *overfitting*.

Model *autoencoder* yang dikembangkan memberikan hasil yang andal dengan nilai *accuracy* sebesar 94%, *precision* sebesar 87%, *recall* sebesar 92%, dan *F1-score* sebesar 89%. Hasil ini membuktikan efektivitas model dalam mendeteksi anomali pada sinyal vibrasi.

Lebih lanjut, hasil ini menunjukkan potensi penerapan model dalam sistem pemantauan kondisi mesin secara *real-time* untuk mencegah kerusakan yang lebih serius dan mengurangi *downtime* produksi. Selain itu, integrasi dengan teknologi *Industrial Internet of Things* (IIoT) dapat dieksplorasi untuk mengotomatisasi proses pengambilan data dan pemantauan anomali secara *real-time*. Dengan pendekatan ini, performa pemeliharaan prediktif di lingkungan industri dapat semakin ditingkatkan.

## DAFTAR PUSTAKA

- [1] M. H. Mohd Ghazali and W. Rahiman, "Vibration Analysis for Machine Monitoring and Diagnosis: A Systematic Review," *Shock and Vibration*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/9469318.
- [2] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the Industry 4.0: A systematic literature review," *Comput Ind Eng*, vol. 150, p. 106889, Dec. 2020, doi: 10.1016/j.cie.2020.106889.
- [3] Y. Liao, H. Li, L. Zhang, Q. Yan, Y. Yang, and B. Li, "Anomaly Transformer for Vibration Data Analysis: Case Study on CWRU Dataset," in *2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, IEEE, Nov. 2023, pp. 785–788. doi:

- 10.1109/ICICML60161.2023.10424762.
- [4] N. Raj, "An Improved Accuracy and Efficiency Based Defect Detection Method for Industrial Signal Processing," in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, IEEE, Aug. 2023, pp. 1–7. doi: 10.1109/ASIANCON58793.2023.10270706.
- [5] X. Zhou, Y. Wu, J. Li, and C. Song, "Anomaly Detection in Rotating Machinery Vibration Signals Based on Domain Transformation and Transformer," in *2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, Sep. 2024, pp. 163–167. doi: 10.1109/ITNEC60942.2024.10732926.
- [6] S. V. T. K., V. C., V. S. D., T. R., and N. A. Patil, "Anomaly Detection in Rotary Equipment using hybrid model with HGSO Optimization," in *2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, IEEE, Jul. 2024, pp. 1–6. doi: 10.1109/CONECCT62155.2024.10677203.
- [7] K. Almutairi, J. K. Sinha, and H. Wen, "Fault Detection of Rotating Machines Using poly-Coherent Composite Spectrum of Measured Vibration Responses with Machine Learning," *Machines*, vol. 12, no. 8, p. 573, Aug. 2024, doi: 10.3390/machines12080573.
- [8] A. Styanidis, M. Loufakis, P. Symeonidis, D. Ioannidis, D. Tzovaras, and I. Kourmpetis, "Vibration-based Anomaly Detection on Weather Radar Rotating Machinery using One Dimensional Convolutional Autoencoder," in *2024 2nd International Conference on Intelligent Perception and Computer Vision (CIPCV)*, IEEE, May 2024, pp. 38–45. doi: 10.1109/CIPCV61763.2024.00017.
- [9] K. Pawar and V. Z. Attar, "Assessment of Autoencoder Architectures for Data Representation," 2020, pp. 101–132. doi: 10.1007/978-3-030-31756-0\_4.
- [10] T. Gao, J. Yang, and S. Jiang, "A Novel Fault Detection Model Based on Vector Quantization Sparse Autoencoder for Nonlinear Complex Systems," *IEEE Trans Industr Inform*, vol. 19, no. 3, pp. 2693–2704, Mar. 2023, doi: 10.1109/TII.2022.3174715.
- [11] D. Dasari and P. S. Varma, "Employing Various Data Cleaning Techniques to Achieve Better Data Quality using Python," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, IEEE, Dec. 2022, pp. 1379–1383. doi: 10.1109/ICECA55336.2022.10009079.
- [12] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Appl Soft Comput*, vol. 138, p. 110176, May 2023, doi: 10.1016/j.asoc.2023.110176.
- [13] A. A. Nelay and M. Turgeon, "A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs," *Machine Learning with Applications*, vol. 17, p. 100572, Sep. 2024, doi: 10.1016/j.mlwa.2024.100572.
- [14] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, doi: 10.48550/arXiv.1412.6980.
- [15] I. Goodfellow, Y. Bengio, and A. Courville, "Optimization for Training Deep Models," in *Deep Learning*, MIT Press, 2016.
- [16] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, "Confusion Matrix-based Feature Selection," in *Proceedings of the Twentysecond Midwest Artificial Intelligence and Cognitive Science Conference*, Madison: Omnipress, 2011, pp. 120–127.
- [17] H. M and S. M.N, "A Review on Evaluation Metrics for Data Classification Evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.