

**Studi Komparatif Algoritma Mesin Pembelajaran untuk Deteksi
Anomali pada Mesin Milling**

***Comparative Study of Machine Learning Algorithms for Anomaly
Detection in Milling Machines***

Oleh

Rijal Rahman Zuhri

140910200004

TUGAS AKHIR

**Untuk memperoleh gelar Sarjana Teknik dalam Bidang Ilmu Teknik Elektro
pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran**



**PROGRAM STUDI SARJANA TEKNIK ELEKTRO
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
2025**

**Studi Komparatif Algoritma Mesin Pembelajaran untuk Deteksi
Anomali pada Mesin Milling**

***Comparative Study of Machine Learning Algorithms for Anomaly
Detection in Milling Machines***

Oleh

Rijal Rahman Zuhri

140910200004

TUGAS AKHIR

**Untuk memenuhi mata kuliah Tugas Akhir pada
Program Studi Sarjana Teknik Elektro
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Padjadjaran
Jatinangor, Januari 2025**

Pembimbing Utama,

Co Pembimbing,

**Agus Trisanto Ph.D
NIP. 196808091999031001**

**Dr. Eng. Darmawan Hidayat
NIP. 197412152002121002**

**Mengetahui,
Ketua Program Studi Sarjana Teknik Elektro**

**Dr. Mohammad Taufik
NIP. 197001121995121001**

PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis saya, tugas akhir berjudul “Studi Komparatif Algoritma Mesin Pembelajaran untuk Deteksi Anomali pada Mesin Milling” adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, magister dan/atau doktor) , baik di Universitas Padjadjaran maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya mandiri, tanpa bantuan pihak lain, kecuali arahan Tim Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah disebutkan nama pengarang dan dicantumkan dalam daftar pustaka
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima saksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta saksi lainnya sesuai dengan norma yang berlaku di Universitas Padjadjaran

Jatinangor, 31 Januari 2025

Yang membuat Pernyataan,

Rijal Rahman Zuhri

NPM. 14091020004

KATA PENGANTAR

Puji dan syukur saya panjatkan ke hadirat Allah SWT. berkat rahmat, karunia serta hidayahNya saya dapat menyusun Laporan Tugas Akhir sebagai pengajuan judul dan metode dalam menyelesaikan tugas akhir saya.

Laporan Tugas Akhir ini berjudul **“Studi Komparatif Algoritma Mesin Pembelajaran untuk Deteksi Anomali pada Mesin Milling”**. Laporan Tugas Akhir ini dibuat dalam rangka menyelesaikan mata kuliah Tugas Akhir yang harus diselesaikan oleh Mahasiswa S1 Jurusan Teknik Elektro di Universitas Padjadjaran. Isi dari Laporan Tugas Akhir ini adalah proyek yang akan saya kembangkan serta metode yang saya gunakan berkaitan dengan tugas akhir.

Tujuan dari Tugas Akhir ini adalah untuk memberikan mahasiswa ilmu yang dapat diterapkan diluar dari kampus, ilmu tersebut dapat berupa teori maupun praktek. Mahasiswa harus memantapkan teori yang sudah dipelajari di kampus dan mengaplikasikannya di lapangan. Harapannya materi yang diberikan dapat dikembangkan dengan baik dan dapat membantu masyarakat atau berbagai instansi dalam pekerjaannya.

Dengan segala hormat saya ucapkan terimakasih kepada dosen-dosen di jurusan Teknik Elektro Universitas Padjadjaran atas ilmu yang diberikan selama 7 semester ini dan sangat membantu saya dalam mengerjakan Tugas Akhir.

Dalam penyusunan Laporan Tugas Akhir ini, banyak pihak yang telah membantu penulis. Maka penulis sampaikan rasa terimakasih kepada:

1. Allah SWT. yang telah menjaga dan melindungi penulis dalam mencari materi untuk menyusun laporan tugas akhir
2. Kedua Orang Tua dan adik saya yang telah memberikan dukungan dan semangat selama penulis mengikuti kuliah hingga dapat menyelesaikan laporan.
3. Saudara-saudara saya yang memberikan dukungan baik secara batin dan moral sehingga memberikan semangat tambahan dalam menyelesaikan laporan.

4. Bapak Dr. Mohammad Taufik, M.Si selaku Kepala Program Studi Teknik Elektro Universitas Padjadjaran dan Dosen wali.
5. Bapak Agus Trisanto, Ph.D selaku Dosen Pembimbing 1 yang selalu menerima berbagai diskusi dan memberikan ide dalam membantu pengerjaan tugas akhir saya.
6. Bapak Dr. Eng. Darmawan Hidayat selaku Dosen Pembimbing 2 yang selalu menerima berbagai diskusi dan memberikan ide dalam membantu pengerjaan tugas akhir saya.
7. Bapak/Akang Septian Ari Kurniawan, M.T selaku Dosen yang sudah memberikan motivasi dan semangat selama melakukan aktivitas penyusunan laporan tugas akhir.
8. Para Dosen di program Studi Teknik Elektro, FMIPA, Universitas Padjadjaran yang telah mendidik dan memberikan ide dan inovasi selama penulisan laporan ini.
9. Anita, Adi, Akmal, Fauzan, Galvin, Given, Ikhsan, dan Ihsan yang memberikan dukungan selama pengerjaan tugas akhir.
10. Seluruh teman-teman Teknik Elektro Angkatan 2020.
11. Semua pihak yang telah membantu dalam penyelesaian tugas akhir.

Penulis menyadari bahwa dalam penulisan laporan ini masih banyak terdapat kekurangan. Oleh karena itu, saran dan kritik senantiasa penulis harapkan untuk kesempurnaan laporan ini. Besar harapan penulis agar laporan ini dapat bermanfaat dan memberikan sumbangan bagi kemajuan ilmu pengetahuan dan teknologi di Indonesia, khususnya di Universitas Padjadjaran.

Jatinangor, 31 Januari 2025

Penulis

Rijal Rahman Zuhri

ABSTRAK

Dalam industri modern, mesin milling memainkan peran penting dalam produksi komponen presisi. Namun, tantangan yang dihadapi mencakup kebutuhan untuk menjaga kualitas produk dan mengurangi biaya operasional, di mana kegagalan mesin dapat menyebabkan downtime yang signifikan. Penelitian ini mengusulkan pengembangan sistem deteksi anomali menggunakan algoritma *Machine Learning* (ML) yaitu *Artificial Neural Network* (ANN), *random forest*, *Extreme Gradient Boosting* (XGBoost), dan *Support Vector Machine* (SVM) untuk menganalisis data sintetis yang dihasilkan. Penelitian ini menggunakan data sintetis yang tidak berimbang dari UC Irvine Machine Learning Repository. Proses pra-pemrosesan data menggunakan SMOTE dalam peyeimbangan data. Hasil evaluasi dari metrik yang digunakan, model ANN unggul dibandingkan model lain dengan nilai pada data anomali. Nilai presisi sebesar 81%, *recall* sebesar 91%, F1-score sebesar 85%, *G-Mean* sebesar 95%, dan MCC sebesar 82%.

Kata kunci : Mesin Milling, Deteksi Anomali, Klasifikasi Biner, Artificial Intelligence

ABSTRACT

In the modern industry, milling machines play a crucial role in the production of precision components. However, the challenges faced include the need to maintain product quality and reduce operational costs, as machine failures can result in significant downtime. The goal of this study is to develop a system that can identify anomalies in fake data by utilizing Machine Learning (ML) algorithms such as support vector machine (SVM), random forests, artificial neural network (ANN), and extreme gradient boosting (XGBoost). The study utilizes imbalanced synthetic data from the UC Irvine Machine Learning Repository. The data preprocessing process employs SMOTE for data balancing. The metrics used show that the ANN model does better than others when tested on anomaly data, with an accuracy of 81%, a recall of 91%, an F1-score of 85%, a G-Mean of 95%, and an MCC of 82%.

Keywords: Milling Machine, Anomaly Detection, Binary Classification, Machine Learning

DAFTAR ISI

PERNYATAAN	ii
KATA PENGANTAR	iii
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	1
DAFTAR GAMBAR.....	4
DAFTAR TABEL	6
BAB I PENDAHULUAN.....	7
1.1. Latar Belakang	7
1.2. Rumusan Masalah	9
1.3. Batasan Masalah.....	9
1.4. Tujuan Penelitian	10
1.5. Kegunaan Penelitian.....	10
BAB II KAJIAN PUSTAKA.....	11
2.1. Mesin <i>Milling</i>	13
2.2. <i>Artificial Neural Network</i>	13
2.3 Fungsi Aktivasi <i>Rectified Linear Unit</i>	17
2.4. <i>Random Forest</i>	17
2.5. <i>Extreme Gradient Boosting</i>	18
2.6. <i>Support Vector Machine</i>	19
2.7. <i>Principal Component Analysis</i>	20
2.8. <i>Robustscaler</i>	21
2.9. <i>Synthetic Minority Over-sampling Technique</i>	21

2.10. Metrik Evaluasi	22
2.10.1. Matrik Korelasi	22
2.10.2. Matrik Kebingungan	23
2.10.3. Precision Recall Curve.....	24
2.10.4. Matthew Correlation Coefficient	25
2.10.5. G-Mean	26
BAB III METODOLOGI PENELITIAN	27
3.1. Model Penelitian	27
3.2. Data Penelitian	27
3.3. Perancangan Model.....	28
3.3.1. Pra-pemrosesan Data.....	29
3.3.2. Perancangan Model.....	30
3.3.3. Evaluasi Model.....	33
3.4. Perangkat Penelitian.....	33
3.5. Jadwal Kegiatan	34
BAB IV HASIL DAN PEMBAHASAN	35
4.1. Dataset	35
4.2. Pra-pemrosesan	39
4.2.1. Transformasi Data.....	39
4.2.2. Pembagian Data	40
4.2.3. Penyeimbangan Data.....	40
4.2.4. Pengacakan Data	41
4.3. Pemodelan.....	41
4.3.1. Arsitektur Model	42
4.3.2. Pelatihan Model	43

4.4. Evaluasi Model.....	48
BAB V KESIMPULAN	52
5.1. Kesimpulan	52
5.2. Saran.....	52
DAFTAR PUSTAKA.....	53

DAFTAR GAMBAR

Gambar 2.1 Mesin Milling CNC [14]	13
Gambar 2.2 Arsitektur Artificial Neural Networks.....	14
Gambar 2.3 Fungsi sigmoid [16].....	15
Gambar 2.4 Grafik ReLU[19]	17
Gambar 2.5 SMOTE [27].....	22
Gambar 2.6 Matrik Korelasi [29].....	23
Gambar 2.7 Matrik Kebingungan [30].....	23
Gambar 2.8 Precision-Recall Curve[31]	25
Gambar 3.1 Diagram Alir Penelitian	27
Gambar 3.2 Alur Perancangan Model.....	28
Gambar 3.3 Pra-pemrosesan Data.....	29
Gambar 3.4 Diagram Alur Pemodelan dan Pelatihan	30
Gambar 4.1 Visualisasi Distribusi Data Fitur Numerikal	36
Gambar 4.2 Visualisasi Data Variabel Target.....	36
Gambar 4.3 Matrik Korelasi pada Dataset	37
Gambar 4. 4 Box Plot Fitur Numerikal terhadap Target.....	38
Gambar 4.5 Grafik antar fitur-fitur sebelum dilakukan normalisasi	39
Gambar 4.6 Grafik antar fitur-fitur setelah dilakukan normalisasi	40
Gambar 4.7 Hasil Pembagian Data Latih, Data Validasi, dan Data Uji....	40
Gambar 4.8 Perbedaan Data Sebelum dan Sesudah Diseimbangkan.....	41
Gambar 4.9 Kurva Riwayat Pelatihan atau Kurva Pembelajaran Kesalahan Klasifikasi pada (a) ANN, (b) <i>Random Forest</i> , (c) XGBoost, dan (d) SVM	46
Gambar 4.10 Hasil Pencarian Ambang Batas (a) ANN, (b) <i>Random Forest</i> , (c) XGBoost, dan (d) SVM	47
Gambar 4.11 <i>Confusion matrix</i> pada Model (a) ANN, (b) <i>Random Forest</i> , (c) XGBoost, dan (d) SVM	48
Gambar 4. 12 PRC Keempat Model.....	49

Gambar 4.13 Perbandingan Nilai G-Mean Antar Model	50
Gambar 4.14 Perbandingan Nilai MCC Antar Model.....	51

DAFTAR TABEL

Tabel 3.1 Fitur-fitur dalam dataset	28
Tabel 3.2 Jadwal Kegiatan	34
Tabel 4.1 Karakteristik Data	35
Tabel 4.2 Hasil klasifikasi pada anomali seluruh metode	49

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam era industri modern, banyak perusahaan manufaktur mengadopsi mesin otomatisasi guna meningkatkan efisiensi produksi. Salah satu teknologi kunci adalah mesin *milling*. Mesin tersebut berperan vital dalam memproduksi komponen presisi tinggi, terutama komponen tipis yang rentan mengalami deformasi akibat tegangan mekanis[1]. Mesin ini mengandalkan mekanisme pemotongan rotatif oleh *cutter* untuk membentuk material sesuai desain digital [2]. Namun, operasional mesin *milling* rentan terhadap kegagalan tak terduga (seperti panas berlebih atau kegagalan beban mekanik) yang sering tidak terdeteksi selama pemantauan manual oleh operator. Kegagalan ini tidak hanya menurunkan kualitas produk, tetapi juga memicu tantangan kompleks bagi industri. Tantangan kompleks tersebut seperti peningkatan biaya perawatan, downtime produksi, dan risiko ketidakpatuhan terhadap standar kualitas.

Untuk meminimalkan downtime, pemeliharaan preventif menjadi solusi yang banyak diadopsi oleh industri. Pendekatan ini mengandalkan jadwal perawatan rutin berdasarkan interval waktu atau penggunaan mesin tertentu. Namun, pemeliharaan preventif memiliki kelemahan signifikan: jadwal perawatan yang tetap tidak mempertimbangkan kondisi aktual mesin. Akibatnya, kegagalan yang terjadi di luar jadwal perawatan sering tidak terdeteksi, sehingga mesin tetap berisiko mengalami breakdown mendadak.

Oleh karena itu, pemantauan kondisi mesin perlu adanya pemeliharaan prediktif. Pemeliharaan prediktif digunakan untuk mengantisipasi kerusakan berlebih dengan memantau kondisi mesin secara dini. Pemeliharaan prediktif dapat mengurangi biaya perbaikan dan pemeliharaan terjadwal masing-masing hingga 12% dan 30%, dan secara signifikan mengurangi kerusakan dengan memprediksi sekitar 70% kegagalan [3]. Perawatan prediktif secara pendekatan dibagi menjadi tiga yaitu berbasis data, berbasis pengetahuan, dan berbasis fisika [4]. Namun, tren teknologi industri 4.0 pendekatan yang dilakukan berdasarkan data melalui

pemodelan salah satunya *Machine Learning* (ML) [5], [6]. Prediksi yang dilakukan melalui deteksi anomali. Dengan memanfaatkan ML dalam deteksi dini terhadap potensi kerusakan pada mesin dapat dilakukan terhadap potensi kerusakan pada mesin. Jadi, memungkinkan tindakan perbaikan dilakukan lebih awal dan mengurangi risiko yang tidak diinginkan.

Penerapan ML dalam mendeteksi anomali telah banyak diteliti. Salah satu pendekatan yang digunakan adalah algoritma pembelajaran mesin seperti *random forest* dan *Support Vector Machine* (SVM), yang terbukti efektif dalam memprediksi stabilitas jaringan listrik [7]. Selain itu, analisis komparatif terhadap berbagai algoritma, termasuk *decision tree* dan *neural network*, menunjukkan bahwa beberapa metode dapat mencapai akurasi tertinggi dalam klasifikasi stabilitas jaringan [8]. Untuk mengatasi masalah klasifikasi data tidak seimbang beberapa penelitian mengusulkan penggunaan SVM yang dioptimalkan dengan *genetic algorithm* dan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) yang meningkatkan kinerja dalam mengidentifikasi data minoritas [9]. Di sisi lain, sistem identifikasi cerdas yang mengintegrasikan teknik pembelajaran mesin juga telah dikembangkan untuk mendeteksi dan mengklasifikasikan gangguan dalam jaringan pintar [10]. Dalam konteks manajemen energi terbaru, analitik *big data* dan teknik *machine learning* digunakan untuk memprediksi permintaan dan produksi energi, yang berkontribusi pada efisiensi sistem [11]. Model pembelajaran mesin lainnya, seperti *neural network* dan *decision tree*, juga diterapkan untuk memprediksi konsumsi energi dengan hasil yang menjanjikan [12]. Selain itu, pengembangan model *artificial neural network* (ANN) dengan teknik optimasi menunjukkan potensi besar dalam meningkatkan akurasi prediksi konsumsi energi [13]. Metode pembelajaran mesin berbasis *cascade*, termasuk *random forest* dan *gradient boosting*, telah digunakan untuk klasifikasi prediksi stabilitas jaringan pintar dengan memanfaatkan data real-time dari perangkat IoT [14]. Terakhir, penerapan SVM yang dioptimalkan dengan *simulated annealing* dan teknik generasi data sintetik juga menunjukkan efektivitas dalam klasifikasi data tidak seimbang [15].

Penelitian ini menggunakan *dataset* dari UC Irvine Machine Learning Repository [16]. Penelitian sebelumnya yang menggunakan dataset tersebut berfokus pada *Explainable Artificial Intelligence* (XAI) [17]. XAI memahami isi *black box* dari ML. Metode yang digunakan pada penelitian tersebut yaitu *decition tree*.

Penelitian ini berfokus pada perbandingan antara metode ANN, *random forest*, *Extreme Gradient Boosting* (XGBoost), dan *Support Vector Machine* (SVM) untuk mendeteksi kondisi mesin *milling*. Metode-metode tersebut didukung oleh penanganan outlier pada data melalui teknik robust scaler. Selain itu, ketidakseimbangan dataset diatasi menggunakan metode SMOTE. Seluruh metode diterapkan untuk mengklasifikasikan kondisi mesin berdasarkan parameter-parameter operasional dengan menggunakan *threshold* berupa *f1-score*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, permasalahan utama yang dihadapi adalah kegagalan dalam mendeteksi kondisi mesin yang dapat mengakibatkan downtime dan kerugian dalam proses produksi. Penelitian ini bertujuan untuk merancang sebuah sistem yang dapat mendeteksi kegagalan kondisi mesin berdasarkan parameter-parameter operasional pada mesin *milling*. Untuk mencapai tujuan tersebut, penelitian ini akan membandingkan efektivitas beberapa metode pembelajaran mesin, yaitu ANN, *random forest*, XGBoost, dan SVM dalam mendeteksi anomali pada kondisi mesin.

1.3. Batasan Masalah

Penelitian ini terdapat beberapa batasan masalah yang sudah ditentukan, di antaranya:

1. Data mesin *milling* diperoleh dari UC Irvine Machine Learning Repository berupa data sintetis
2. Metode yang digunakan yaitu ANN, *random forest*, XGBoost, dan SVM
3. Pembuatan model menggunakan bahasa pemrograman python
4. Model menggunakan klasifikasi biner untuk memprediksi kondisi mesin *milling*

5. Matrik evaluasi yang digunakan pada model ini yaitu presisi, *recall*, *f1-score*, kurva *precision-recall*, *confusion matrix*, *g-mean*, dan *Matthews Correlation Coefficient* (MCC)

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk membandingkan efektivitas metode ANN, *random forest*, XGBoost, dan SVM dalam melakukan klasifikasi biner terhadap kegagalan kondisi mesin *milling* berdasarkan parameter-parameter operasional. Model-model tersebut dirancang untuk memiliki sensitivitas yang lebih tinggi dan akan dievaluasi menggunakan metrik presisi, *recall*, *f1-score*, kurva *precision-recall*, *confusion matrix*, *g-mean*, dan MCC.

1.5. Kegunaan Penelitian

Penelitian ini bermanfaat bagi divisi pemeliharaan dalam melakukan perawatan prediktif terhadap kondisi mesin *milling*. Melalui deteksi anomali berbasis data operasional, divisi pemeliharaan dapat mengidentifikasi potensi kerusakan lebih awal sehingga meminimalkan downtime dan biaya perawatan.

BAB II

KAJIAN PUSTAKA

Dalam penelitian ini, tinjauan literatur mengenai deteksi anomali yang memanfaatkan ML akan digunakan dengan mengacu pada berbagai penelitian terdahulu. Referensi yang dihasilkan dari tinjauan literatur tersebut akan berfungsi sebagai landasan perbandingan dan penunjang bagi penelitian ini. Tabel 2.1 menyajikan ringkasan hasil dari tinjauan literatur terkait penelitian-penelitian sebelumnya.

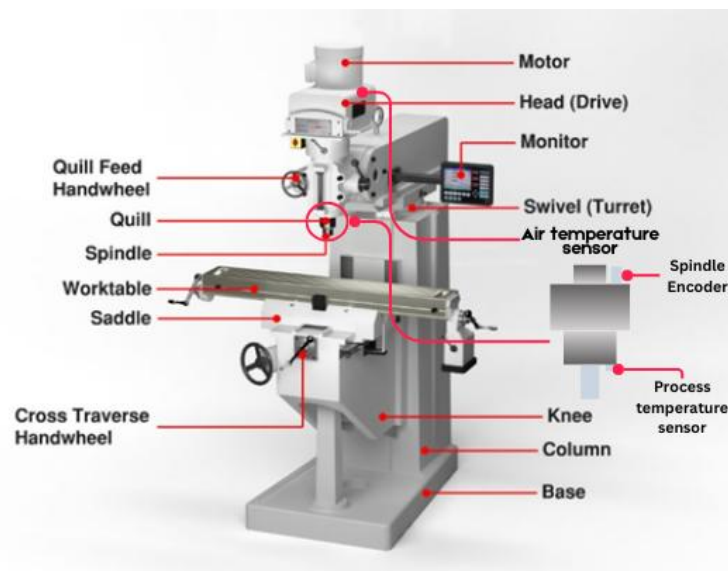
Tabel 2.1 Referensi Hasil Penelitian Sebelumnya

No	Judul Penelitian	Hasil Penelitian	Kelemahan
1	<i>Power-grid stability predictions using transferable machine learning [7]</i>	Penelitian ini menggunakan data yang <i>imbalanced</i> dilakukan teknik penyeimbangan dengan <i>undersampling</i> dan <i>oversampling</i> dalam mengatasi keseimbangan. Metode yang digunakan dalam penelitian ini yaitu <i>random forest</i> , SVM, dan ANN	Penggunaan <i>undersampling</i> menghilangkan informasi dan <i>oversampling</i> tidak menambahkan data baru melainkan menyalin data yang sudah ada
2	<i>Comparative analysis of machine learning algorithms for prediction of smart grid stability [8]</i>	Penelitian menggunakan data yang <i>imbalanced</i> melalui metode SVM, k-NN, LogisticR, <i>Naive bayes</i> , dan <i>decision tree</i>	Tidak menggunakan penyeimbangan data
3	<i>An optimized support vector machine with genetic algorithm for imbalanced data classification [9]</i>	Penelitian ini menggunakan metode SVM dan <i>genetic algorithm</i> dengan penyeimbang data SMOTE	Terbatasnya metode yang digunakan

4	<i>Meticulously intelligent identification system for smart grid network stability to optimize risk management [10]</i>	Penelitian ini menggunakan teknik penyeimbangan dengan <i>undersampling</i> dalam mengatasi keseimbangan. Metode yang digunakan dalam penelitian ini yaitu SVM dan <i>decision tree</i>	Penggunaan <i>undersampling</i> menghilangkan informasi data
5	<i>Renewable energy management in smart grids by using big data analytics and machine learning [11]</i>	Penelitian ini menggunakan teknik penyeimbangan SMOTE. Metode yang digunakan dalam penelitian ini yaitu <i>decision tree</i> dan <i>random forest</i>	Terbatasnya metode yang digunakan.
6	<i>Machine learning-based model for prediction of power consumption in smart grid-smart way towards smart city [12]</i>	Penelitian menggunakan teknik penyeimbangan dengan <i>oversampling</i> . Metode yang digunakan <i>logistic regression</i> , <i>decision tree</i> , dan SVM.	Penggunaan <i>oversampling</i> tidak menambahkan data baru melainkan menyalin data yang sudah ada
7	<i>Imbalanced Data Classification Using SVM Based on Improved Simulated Annealing Featuring Synthetic Data Generation and Reduction [13]</i>	Penelitian menggunakan metode SVM dengan mengoptimalkan <i>simulated annealing algorithm</i>	Data yang digunakan tidak dilakukan <i>imbalanced</i> terlebih dahulu
8	<i>Efficient artificial neural network for smart grid stability prediction [14]</i>	Penelitian menggunakan metode yang digunakan ANN, SVM, dan <i>desition tree</i>	Data yang digunakan tidak dilakukan <i>imbalanced</i> terlebih dahulu
9	<i>Classification of smart grid stability prediction using cascade machine learning methods and the internet of things in smart grid [15]</i>	Penelitian menggunakan metode yang digunakan kernel <i>naive bayes</i> dan SVM	Data yang digunakan tidak dilakukan <i>imbalanced</i> terlebih dahulu
10	<i>A machine learning-based workflow for automatic detection of anomalies in machine tools [18]</i>	Penelitian ini menggunakan penyeimbangan pada data tidak seimbang. Metode yang digunakan yaitu SVM, XGBoost, <i>decition tree</i> , dan <i>random forest</i> .	Evaluasi dalam jurnal tidak menggunakan <i>g-mean</i> yang dapat mengukur keseimbangan performa model.

2.1. Mesin *Milling*

Mesin milling CNC (Computer Numerical Control) memegang peranan penting dalam proses manufaktur modern karena mampu mengotomatisasi pemotongan dengan bantuan komputer. Mesin *milling* CNC memiliki berbagai komponen utama yang bekerja secara sinergis untuk mencapai hasil pemotongan yang optimal. Mesin *milling* yang digunakan dalam penelitian ini tercantum pada Gambar 2.1.



Gambar 2.1 Mesin Milling CNC [19]

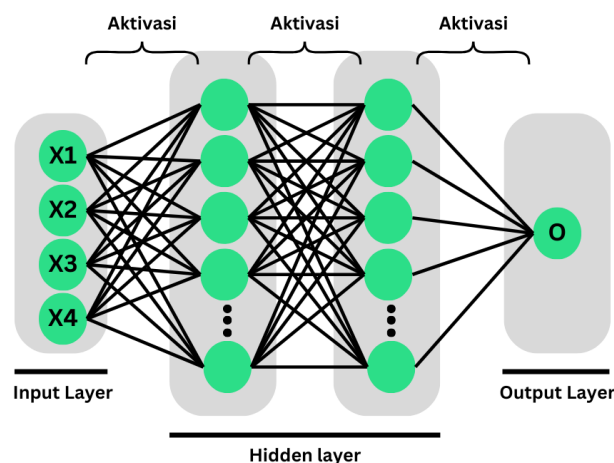
Proses kerja mesin milling CNC dimulai dengan memasukkan G-code untuk mengatur pergerakan mesin. Servo motor mengontrol pergerakan meja pada sumbu X, Y, dan Z untuk posisi benda kerja, sementara spindle berputar pada kecepatan tertentu untuk memotong material sesuai program. Benda kerja dijepit menggunakan *fixture* untuk menjaga kestabilan. Parameter seperti kecepatan spindle dan kecepatan pemakanan dipantau untuk memastikan akurasi pemesinan. Kecepatan rotasi spindle diukur melalui *encoder*, sedangkan torsi dipantau melalui arus motor. Sensor suhu udara dan suhu proses dipasang untuk mencegah overheating dan memantau temperatur di area pemotongan.

2.2. *Artificial Neural Network*

Artificial neural network (ANN) adalah model komputasi yang terinspirasi oleh jaringan saraf biologis, yang terdiri dari elemen-elemen pemrosesan sederhana

yang disebut neuron. Neuron ini saling terhubung dan bekerja secara paralel untuk menyelesaikan berbagai tugas. ANN terdiri dari tiga komponen utama: lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Pada setiap *neuron*, sinyal diterima sebagai *input* diproses. Kemudian hasil proses tersebut dikirimkan ke neuron di lapisan berikutnya. Fungsi dasar dari ANN adalah untuk mempelajari pola-pola dari data dan membuat prediksi berdasarkan pola-pola tersebut [20].

ANN terdiri dari beberapa lapisan yang dirancang untuk menangkap hubungan kompleks dalam data. Gambar 2.2 menunjukkan rancangan lapisan-lapisan dalam model ANN.



Gambar 2.2 Arsitektur Artificial Neural Networks

Struktur dasar ANN melibatkan:

1. Lapisan input untuk menerima data eksternal yang akan diproses.
2. Lapisan tersembunyi yang memproses data dan menerapkan fungsi komputasi guna menemukan pola.
3. Lapisan output menghasilkan keluaran berdasarkan pola yang telah ditemukan.

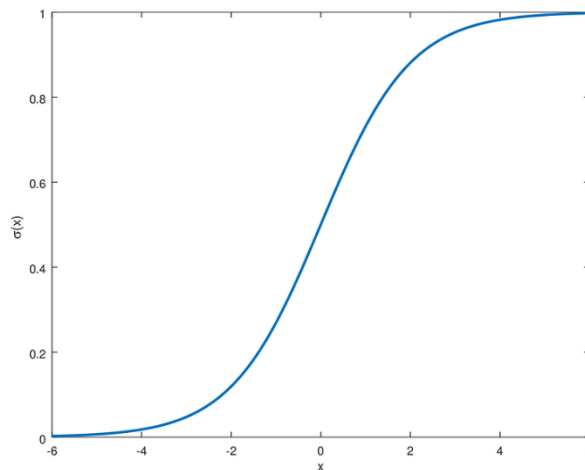
Setiap koneksi antar-neuron diwakili oleh bobot yang menentukan seberapa besar pengaruh neuron sebelumnya terhadap neuron berikutnya[20]. Proses ini dapat dirumuskan sebagai *forward propagation* yang dinyatakan dengan rumusan berikut [20].

$$h_j = f\left(\sum_{i=1}^n w_{ij}x_i + b_j\right) \quad (2.1)$$

Keluaran neuron j dilambangkan sebagai h_j . h_j merupakan hasil perhitungan dalam jaringan neural yang dipengaruhi oleh beberapa komponen utama. Bobot w_{ij} menghubungkan neuron i ke neuron j dan menentukan seberapa besar pengaruhnya terhadap keluaran. Selain itu, neuron j menerima masukan berupa keluaran dari neuron sebelumnya yang dinyatakan sebagai x_i dan berfungsi sebagai dasar dalam proses komputasi. Selanjutnya, bias b_j memberikan fleksibilitas agar neuron dapat menghasilkan keluaran optimal meskipun masukan awal mungkin rendah.

Tugas ANN dalam menjalani klasifikasi biner dalam memprediksi salah satu dari dua kelas. Lapisan *output* umumnya menggunakan fungsi aktivasi *sigmoid*. Aktivasi ini mengubah keluaran jaringan menjadi probabilitas antara 0 dan 1. Fungsi aktivasi tersebut dapat dinyatakan sebagai berikut [21].

$$\hat{y} = \frac{1}{1+e^{-z}} \quad (2.2)$$



Gambar 2.3 Fungsi sigmoid [21]

Output dari *neuron* sebelum aktivasi dilambangkan sebagai z dan merupakan kombinasi linier dari input dan bobot yang terhubung. Nilai ini kemudian diproses melalui fungsi aktivasi untuk menghasilkan probabilitas yang dilambangkan sebagai \hat{y} . Probabilitas ini menunjukkan kemungkinan bahwa input tersebut termasuk dalam kelas 1, yang sering digunakan dalam konteks klasifikasi biner pada jaringan neural.

Perhitungan kesalahan dalam klasifikasi biner menggunakan *binary cross-entropy loss*. Penggunaan *binary cross-entropy loss* menghitung perbedaan antara probabilitas prediksi dan label aktual. Rumus dari penggunaannya sebagai berikut [22].

$$L = -\frac{1}{N} \sum_{i=1}^N [y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)] \quad (2.3)$$

Jumlah sampel dalam dataset dilambangkan sebagai N yang mewakili total data yang digunakan dalam pelatihan atau evaluasi model. Setiap sampel memiliki label sebenarnya y_i yang bernilai 0 atau 1 untuk menunjukkan kategori atau kelas sebenarnya dari data tersebut. Selain itu, probabilitas prediksi untuk kelas 1 dilambangkan sebagai \hat{y}_i yang mencerminkan estimasi model terhadap kemungkinan bahwa sampel tersebut termasuk dalam kelas 1.

Setelah melakukan perhitungan kesalahan menggunakan *binary cross-entropy*, perhitungan gradien dari *loss* terhadap *output* perlu dilakukan. Perhitungan ini dengan mengurangi nilai prediksi dengan nilai sesungguhnya. Hal ini dapat dirumuskan sebagai berikut [20].

$$\frac{\partial E}{\partial z} = y_{pred} - y_{aktual} \quad (2.4)$$

Setelah kesalahan gradien dihitung, backpropagation digunakan untuk menghitung gradien dari *loss* terhadap setiap bobot dalam jaringan. Pembaruan bobot dilakukan menggunakan gradient descent dengan rumus sebagai berikut [23].

$$w_{ij}^{(baru)} = w_{ij}^{(lama)} - \eta \frac{\partial L}{\partial w_{ij}} \quad (2.5)$$

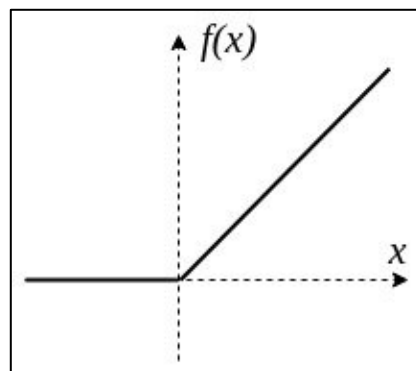
Bobot yang diperbarui dilambangkan sebagai $w_{ij}^{(baru)}$ dan merupakan hasil penyesuaian dari bobot sebelumnya $w_{ij}^{(lama)}$. Penyesuaian ini dipengaruhi oleh laju pembelajaran η yang menentukan seberapa besar perubahan pada bobot setiap iterasi. Gradien dari fungsi kerugian terhadap bobot w_{ij} dirumuskan sebagai $\frac{\partial L}{\partial w_{ij}}$, mengarahkan pembaruan ini dengan menunjukkan arah dan besar perubahan untuk meminimalkan kesalahan pada model. Dengan iterasi berulang-ulang, bobot dalam ANN diperbarui untuk meminimalkan kesalahan sehingga model menjadi lebih baik dalam memprediksi kelas biner. Dengan iterasi berulang-ulang, bobot dalam

jaringan saraf diperbarui untuk meminimalkan kesalahan, sehingga model menjadi lebih baik dalam memprediksi kelas biner.

2.3 Fungsi Aktivasi *Rectified Linear Unit*

Fungsi aktivasi dalam ANN memiliki peran penting dalam menentukan keluaran dari setiap neuron berdasarkan input yang diterima. Fungsi aktivasi ini memengaruhi bagaimana neuron merespons sinyal dan memungkinkan jaringan untuk menangani masalah non-linear, yang sangat diperlukan untuk tugas pembelajaran kompleks. Salah satunya yaitu *Rectified Linear Unit* (ReLU).

ReLU adalah fungsi aktivasi yang banyak digunakan dalam jaringan saraf modern karena kesederhanaan dan efisiensinya. Fungsi ini didefinisikan sebagai $f(x) = \max(0, x)$ [24]. Input negatif diubah menjadi nol, sementara input positif tetap dipertahankan. Jika digambarkan bentuk grafik yang ditampilkan pada aktivasi tersebut terlihat pada gambar 2.4.



Gambar 2.4 Grafik ReLU[24]

2.4. *Random Forest*

Random forest adalah sebuah metode algoritma ML yang termasuk kedalam kategori yang menggabungkan beberapa pohon keputusan. *Random forest* bekerja dengan membangun sejumlah pohon keputusan selama pelatihan dan mengeluarkan kelas yang merupakan modus dari kelas-kelas yang diprediksi oleh masing-masing pohon [25]. Keunggulan metode untuk menangani data yang besar dan kompleks. Metode ini memiliki *bootstrap aggregating (bagging)* untuk mengurangi varian dan meningkatkan stabilitas prediksi.

Cara kerja metode tersebut dengan pengambilan sampel *bootstrap* dari dataset asli, di mana setiap pohon dibangun menggunakan subset acak dari data yang diambil dengan pengembalian. Untuk setiap node dalam pohon, algoritma memilih subset acak dari fitur dan menghitung impurity menggunakan metrik seperti Gini Impurity atau Entropy. *Gini impurity* dan *entropy* dihitung dengan rumus sebagai berikut [25].

$$Gini = 1 - \sum(p_i^2) \quad (2.6)$$

$$Entropy = - \sum(p_i \log_2(p_i)) \quad (2.7)$$

Di mana p_i adalah proporsi dari kelas i .

Setelah semua pohon dibangun, prediksi dilakukan dengan mengumpulkan hasil dari semua pohon. Pendekatan klasifikasi, prediksi akhir adalah kelas yang paling sering dipilih (modus) oleh semua pohon. Metode ini juga dapat memberikan estimasi ketidakpastian dalam prediksi dan menghitung pentingnya fitur berdasarkan kontribusi mereka terhadap pengurangan impurity di seluruh pohon.

2.5. *Extreme Gradient Boosting*

Extreme gradient boosting adalah sebuah metode algoritma ML dalam kategori *ensemble learning*, khususnya *boosting* untuk meningkatkan kecepatan model. Pendekatan ini memungkinkan *XGBoost* untuk secara signifikan mengurangi kesalahan klasifikasi dan meningkatkan performa, menjadikannya pilihan utama dalam berbagai kompetisi data science dan aplikasi industri [26].

Cara kerja metode ini memulai dengan membuat prediksi awal berdasarkan rata-rata probabilitas dari kelas target. Setiap iterasi melibatkan perhitungan residual dari prediksi saat ini, di mana algoritma kemudian membangun pohon keputusan baru yang fokus pada mengurangi kesalahan klasifikasi tersebut. Dengan menerapkan teknik regularisasi, *XGBoost* mampu mencegah *overfitting*, yang sering menjadi tantangan dalam model yang kompleks. Hal ini meningkatkan kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya [27]. Setelah beberapa iterasi, model akhir dihasilkan, yang dapat digunakan untuk melakukan prediksi klasifikasi. Terdapat beberapa rumus yang melibatkan fungsi

loss untuk mengukur kesalahan antara probabilitas yang diprediksi dan kelas aktual. Berikut rumusan-rumusan yang relevan sebagai berikut [26], [27].

1. Fungsi *loss*

$$L(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2.8)$$

Di mana y_i adalah kelas aktual, \hat{y}_i adalah probabilitas yang diprediksi, dan n adalah jumlah sampel.

2. Fungsi regularisasi

$$R(f) = \lambda \sum_{j=1}^k \|w_j\|_2^2 + \gamma k \quad (2.9)$$

Di mana $R(f)$ adalah fungsi regularisasi, w_j adalah bobot dari pohon keputusan, k adalah jumlah pohon, λ adalah parameter regularisasi untuk bobot, dan γ adalah parameter untuk penalti kompleksitas.

3. *Update* model

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta f_i(x) \quad (2.10)$$

Di mana $\hat{y}^{(t)}$ adalah prediksi pada iterasi t , η adalah learning rate, dan $f_i(x)$ adalah fungsi (pohon) yang ditambahkan pada iterasi tersebut.

2.6. Support Vector Machine

Support Vector Machine adalah sebuah metode algoritma pembelajaran mesin yang berfokus pada pencarian hyperplane optimal yang memisahkan kelas-kelas dalam ruang fitur. SVM bekerja dengan membangun model yang memaksimalkan margin antara dua kelas. Margin didefinisikan sebagai jarak terdekat antara titik data dari masing-masing kelas ke hyperplane. Hyperplane dapat dinyatakan dengan rumus sebagai berikut [28].

$$w \cdot x + b = 0 \quad (2.11)$$

Di mana w adalah vektor bobot yang menentukan arah *hyperplane*, x adalah vektor fitur dari data, dan b adalah bias yang menggeser hyperplane dari asal. Margin dapat dinyatakan dalam rumus berikut [7].

$$Margin = \frac{2}{\|w\|} \quad (2.12)$$

SVM meminimalkan fungsi tujuan yang mengabungkan margin dan kesalahan klasifikasi dapat dinyatakan sebagai berikut [7].

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (2.13)$$

Di mana C merupakan parameter regulasi yang mengontrol *trade-off* antara margin dan kesalahan klasifikasi dan ξ_i merupakan variabel slack yang mengukur kesalahan klasifikasi untuk titik data i. Setelah model dibangun, SVM melakukan prediksi dengan menentukan sisi mana dari hyperplane titik data baru berada.

Dalam data yang tidak dapat dipisahkan secara *linier*, SVM memiliki fungsi kernel yang dapat memetakan data ke ruang dimensi yang lebih tinggi. Fungsi kernel umum yang dapat digunakan sebagai berikut [7].

- Kernel linier

$$K(x_i, x_j) = x_i \cdot x_j \quad (2.14)$$

- Kernel polinomial

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d \quad (2.15)$$

- Kernel *Radial Basis Function* (RBF)

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \quad (2.16)$$

Di mana c adalah konstanta, d adalah derajat polinomial, dan γ adalah parameter yang menentukan seberapa jauh pengaruh satu contoh data terhadap contoh lainnya.

2.7. Principal Component Analysis

Principal Component Analysis (PCA) merupakan teknik statistik yang digunakan untuk mengurangi jumlah variabel dalam dataset dengan mempertahankan informasi paling signifikan. Salah satu fungsinya yaitu memudahkan visualisasi data menjadi 2 atau 3 dimensi dari data semula. Cara kerja PCA akan dilakukan penskalaan dengan standarisasi data karena PCA sensitif terhadap data. Standarisasi data dapat dirumuskan sebagai berikut [29].

$$X' = \frac{X - \mu}{\sigma} \quad (2.17)$$

Di mana X adalah data asli, μ adalah rata-rata, dan σ adalah standar deviasi dari masing-masing fitur. Selanjutnya mengukur hubungan antar fitur melalui matriks kovarians dengan rumus sebagai berikut [29].

$$\Sigma = \frac{1}{n-1} X'^T X' \quad (2.18)$$

Di mana Σ adalah matriks kovarians. Matriks kovarian yang sudah terbentuk dihitung nilai eigen dan vektor eigennya untuk menentukan komponen utama yang memiliki varians terbesar. Komponen dengan eigenvalue terbesar dipilih, lalu data diproyeksikan ke dalam ruang dimensi yang lebih rendah dengan mengalikan data yang distandarisasi dengan eigenvector terpilih. Hasilnya adalah data yang lebih sederhana dengan informasi yang tetap relevan.

2.8. Robustscaler

RobustScaler adalah metode normalisasi data yang bertujuan untuk mengurangi pengaruh nilai ekstrem (*outlier*) [30]. Teknik ini bekerja dengan mengurangi setiap nilai data dengan nilai median, kemudian membaginya dengan rentang antar kuartil (*interquartile range* atau IQR). Rumus yang digunakan adalah sebagai berikut [30].

$$X_{\text{scaled}} = \frac{X - \text{Median}(x)}{\text{IQR}(x)} \quad (2.19)$$

Di mana X adalah nilai asli dari data. Median dari X , yaitu $\text{Median}(X)$ merepresentasikan nilai tengah dari kumpulan data yang telah diurutkan. Dengan menggunakan median sebagai titik referensi.

Sementara itu, $\text{IQR}(x)$ atau rentang antar kuartil merupakan selisih antara kuartil ketiga (Q_3) dan kuartil pertama (Q_1). Kuartil ketiga dan pertama masing-masing mewakili batas atas dan bawah 25% dari data, yang mencakup 50% nilai tengah dalam distribusi. Sebagai berikut merupakan rumusan dari $\text{IQR}(x)$ [30].

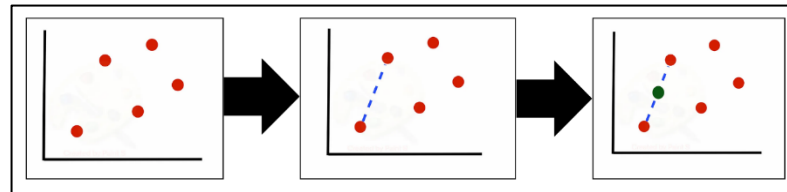
$$\text{IQR}(x) = Q_3(x) - Q_1(x) \quad (2.20)$$

2.9. Synthetic Minority Over-sampling Technique

SMOTE merupakan teknik oversampling yang dirancang untuk meningkatkan jumlah data pada kelas minoritas dengan menghasilkan sampel sintetik melalui interpolasi antara sampel-sampel minoritas yang ada. Teknik ini digunakan untuk mengatasi masalah ketidakseimbangan kelas dalam klasifikasi [31]. Secara matematis, SMOTE dapat dirumuskan sebagai berikut [31].

$$x_{\text{sintetik}} = x_i + \delta \cdot (x_{kNN} - x_i) \quad (2.21)$$

Di mana, sampel x_i dan salah satu tetangga terdekatnya x_{kNN} . Kemudian, δ adalah angka acak yang bernilai antara 0 dan 1, yang digunakan untuk memastikan bahwa sampel sintetis terletak di antara x_i dan x_{kNN} . Dengan demikian, SMOTE menghasilkan sampel sintetis melalui interpolasi linier antara dua sampel minoritas yang dipilih secara acak.



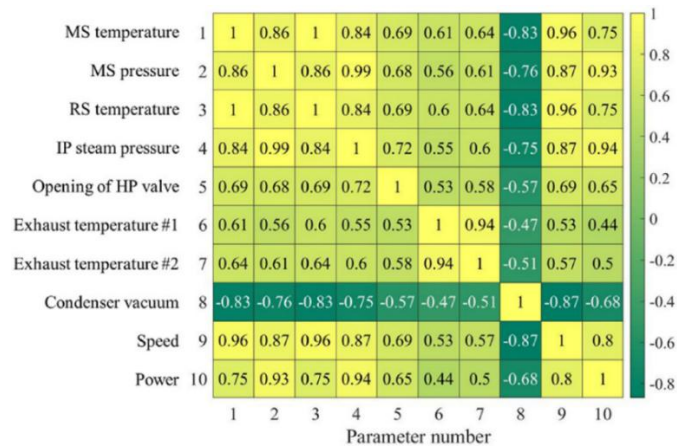
Gambar 2.5 SMOTE [32]

Berdasarkan Gambar 2.5, proses sintesis data menggunakan teknik SMOTE dapat dijelaskan secara lebih rinci. Pertama, diasumsikan terdapat lima data yang tersebar. Konsep dasar SMOTE yaitu melakukan interpolasi antara dua titik data dengan memperhitungkan *k-nearest neighbors*. Seperti yang ditunjukkan pada Gambar 2.5, data pertama memiliki hubungan dengan data lain berdasarkan kedekatan jarak. Selanjutnya, dari dua data yang saling terkait, selisih antara kedua titik dihitung dan dikalikan dengan angka acak antara 0 dan 1. Proses ini menghasilkan data baru yang terletak di sepanjang garis yang menghubungkan kedua titik tersebut. Langkah ini diulangi secara berkelanjutan hingga jumlah data yang disintesis mencapai jumlah yang diinginkan.

2.10. Metrik Evaluasi

2.10.1. Matrik Korelasi

Matrik Korelasi merupakan matriks yang digunakan dalam statistika untuk menggambarkan hubungan antar variabel [33]. Matrik ini menunjukkan korelasi antara setiap variabel dengan variabel lainnya, dengan nilai berkisar antara -1 (negatif) hingga 1 (positif). Nilai 0 menandakan tidak ada hubungan antar variabel. Biasanya, matrik korelasi ditampilkan sebagai matriks yang berisi angka-angka, di mana nilai di atas garis putih adalah positif, nilai di bawah garis putih adalah negatif, dan nilai yang sama dengan 0 menunjukkan tidak memiliki hubungan. Contoh penggunaan matrik korelasi seperti gambar 2.6.



Gambar 2.6 Matrik Korelasi [34]

Rumus untuk menghitung korelasi Pearson antara dua variabel X dan Y sebagai berikut [33].

$$\rho(x, y) = \frac{\sum((x_i - \mu_x)(y_i - \mu_y))}{\sqrt{\sum(x_i - \mu_x)^2 * (\sum(y_i - \mu_y)^2)}} \quad (2.22)$$

Nilai korelasi $\rho(x, y)$ antara variabel x dan y menggambarkan hubungan linear keduanya. Dalam perhitungan ini, x_i dan y_i adalah nilai ke- i dari variabel x dan y. Sementara μ_x dan μ_y adalah rata-rata masing-masing variabel. Rata-rata ini membantu menentukan seberapa jauh setiap nilai x_i dan y_i dari pusat data, yang akhirnya memengaruhi nilai korelasi keseluruhan.

2.10.2. Matrik Kebingungan

Matrik kebingungan adalah tabel yang menampilkan jumlah data uji yang berhasil dan gagal diklasifikasikan, memfasilitasi evaluasi performa sistem klasifikasi [35]. Melalui matrik kebingungan dapat diperoleh gambaran mendetail tentang performa sistem klasifikasi dan dapat diidentifikasi bagian mana kesalahan klasifikasi terjadi. Contoh penggunaan matrik kebingungan seperti Gambar 2.7.

Actual	Predicted		
		Positive (Anomaly)	Negative (Normal)
	Positive (Anomaly)	TP	FN
	Negative (Normal)	FP	TN

Gambar 2.7 Matrik Kebingungan [35]

Pada Gambar 2.7 tersebut perlu dipahami mengenai TP, FP, FN, dan TN. Penjelasan mengenai keempat representasi nilai tersebut yaitu:

1. *True Positive* (TP) adalah jumlah prediksi yang dianggap positif dan dalam realitas juga positif
2. *False Positive* (FP) adalah jumlah prediksi yang dianggap positif, tetapi dalam realitas adalah negatif
3. *False Negative* (FN) adalah jumlah prediksi yang dianggap negatif, namun dalam realitas justru positif
4. *True Negative* (TN) adalah jumlah prediksi yang dianggap negatif dan dalam realitas juga negatif

Tingkat akurasi matriks ini dapat dievaluasi dengan mengindikasikan proporsi dari jumlah prediksi yang benar. Untuk menghitung akurasi, presisi, dan recall, dan *F1-score*, digunakan rumus sebagai berikut [35].

- Akurasi

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.23)$$

- Presisi

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2.24)$$

- *Recall*

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.25)$$

- *F1-Score*

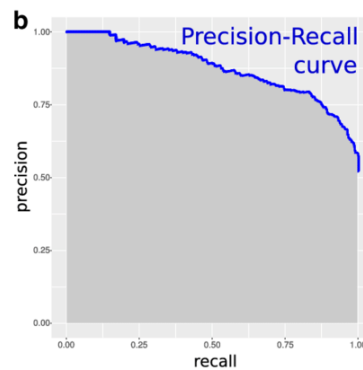
$$\text{F1 - Score} = 2 * \frac{\text{presisi} * \text{recall}}{\text{presisi} + \text{recall}} \quad (2.26)$$

2.10.3. Precision Recall Curve

Precision-Recall Curve (PRC) merupakan alat evaluasi yang umum digunakan dalam ML untuk mengukur efektivitas model klasifikasi biner, terutama ketika menghadapi masalah ketidakseimbangan data. PRC memberikan gambaran mengenai *trade-off* antara *precision* dan *recall*, yang menjadi penting dalam kasus di mana satu kelas data jauh lebih kecil dibandingkan kelas lainnya. PRC memetakan hubungan antara *precision* dan *recall*, yang sangat penting dalam

mengevaluasi model, terutama ketika menghadapi ketidakseimbangan data. Kondisi ini biasanya terjadi ketika satu kelas jauh lebih sedikit dibandingkan kelas lainnya. Kurva ini membantu menilai trade-off antara akurasi prediksi positif dan sensitivitas model terhadap kelas minoritas [36].

PRC menunjukkan hubungan antara precision dan recall pada berbagai threshold keputusan. Pada setiap *threshold*, model menghasilkan prediksi berbeda yang memengaruhi tingkat TP, FP, dan FN. Dengan memvariasikan *threshold*, kurva dapat menunjukkan bagaimana precision berubah seiring recall. Visualisasi PRC dapat dilihat pada Gambar 2.8. Semakin nilai PRC mendekati satu maka model dapat membedakan antar kelas dengan baik. Namun, sebaliknya nilai PRC mendekati nol maka model tidak dapat membedakan antar kelas dengan baik.



Gambar 2.8 *Precision-Recall Curve*[36]

2.10.4. Matthew Correlation Coefficient

Matthews Correlation Coefficient (MCC) adalah metrik evaluasi yang digunakan untuk menilai kinerja model klasifikasi, terutama dalam konteks dataset yang tidak seimbang. MCC memberikan gambaran yang lebih komprehensif tentang kinerja model dengan mempertimbangkan semua kategori prediksi, yaitu *true positives* (TP), *true negatives* (TN), *false positives* (FP), dan *false negatives* (FN). Rumus untuk menghitung MCC adalah sebagai berikut [37].

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (2.27)$$

Nilai MCC berkisar antara -1 hingga +1, di mana nilai +1 menunjukkan prediksi yang sempurna, 0 menunjukkan prediksi yang tidak lebih baik dari acak, dan -1 menunjukkan prediksi yang sepenuhnya salah. Penggunaan MCC sangat

penting dalam situasi di mana kelas positif dan negatif tidak seimbang, karena metrik ini dapat memberikan informasi yang lebih akurat tentang kinerja model dibandingkan dengan akurasi sederhana.

2.10.5. G-Mean

G-mean adalah metrik yang digunakan untuk mengevaluasi kinerja model klasifikasi, terutama dalam situasi di mana terdapat ketidakseimbangan kelas. *G-mean* dihitung sebagai akar dari hasil kali sensitivitas (recall) dan spesifisitas sebagai berikut [38].

$$G\text{-Mean} = \sqrt{\text{sensitivitas} \cdot \text{spesifisitas}} \quad (2.28)$$

Di mana nilai sensitivitas dihitung sebagai berikut,

$$\text{sensitivitas} = \frac{TP}{TP+FN} \quad (2.29)$$

Dan nilai spesifisitas dihitung berdasarkan berikut.

$$\text{spesifisitas} = \frac{TN}{TN+FP} \quad (2.30)$$

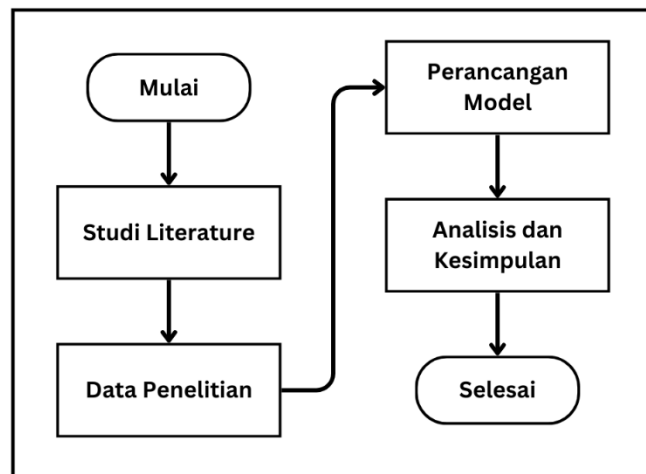
Dengan TP sebagai True Positives, TN sebagai True Negatives, FP sebagai False Positives, dan FN sebagai False Negatives. *G-mean* memberikan gambaran yang lebih komprehensif tentang kinerja model dalam mengklasifikasikan kedua kelas, terutama dalam situasi di mana satu kelas jauh lebih dominan daripada yang lain. Penggunaan *g-mean* memungkinkan peneliti untuk mengevaluasi model tidak hanya berdasarkan akurasi keseluruhan, tetapi juga dengan mempertimbangkan kinerja pada kelas minoritas, yang sering kali diabaikan dalam evaluasi tradisional.

BAB III

METODOLOGI PENELITIAN

3.1. Model Penelitian

Penelitian ini akan dilaksanakan di Pusat Pelayanan Basic Science (PPBS), Gedung C, Laboratorium lantai 3, Universitas Padjadjaran, pada periode April hingga Juli 2024. Fokus utama penelitian ini adalah mengembangkan model deteksi anomali untuk mengidentifikasi kondisi mesin, baik normal maupun anomali. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam pemantauan dan perawatan mesin, sehingga dapat mengurangi risiko kerusakan yang terjadi secara tiba-tiba. Diagram alir yang menggambarkan tahapan penelitian secara keseluruhan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.2. Data Penelitian

Penelitian ini menggunakan data sintetis yang realistis, diperoleh dari UC Irvine Machine Learning Repository [16]. Data yang digunakan terdiri dari 9.859 sampel, yang berisi informasi mengenai parameter operasional dan kondisi mesin pada mesin milling. Kondisi mesin yang dimaksud adalah "normal" dan "anomali". Selain itu, rincian parameter operasional yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Fitur-fitur dalam dataset

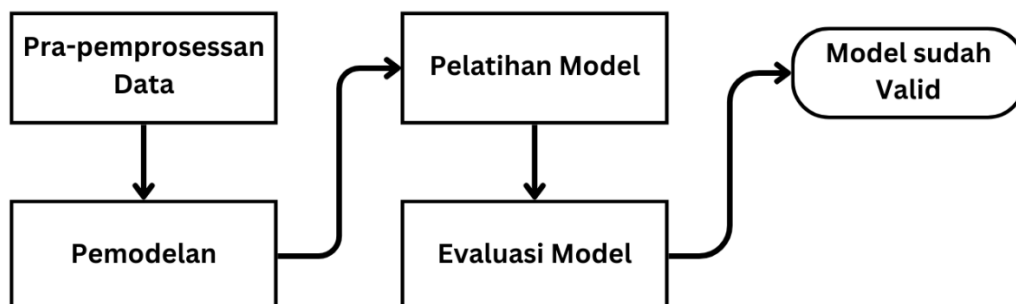
No	Nama fitur	Tipe data	Satuan	Rentang
1	<i>Air temperature</i>	Numerikal	Kelvin	298 K – 302 K
2	<i>Process temperature</i>	Numerikal	Kelvin	309 K – 311 K
3	<i>Rotational speed</i>	Numerikal	rpm	>1380 rpm
4	<i>Torque</i>	Numerikal	Nm	30 Nm – 50 Nm

Berdasarkan tabel tersebut, fitur-fitur dapat dideskripsikan sebagai berikut [16].

1. *Air temperature* [K] dihitung melalui proses nonlinier dan kemudian dinormalisasi ke nilai sekitar 300 K, dengan deviasi standar dua K.
2. *Process temperature* [K] dengan meningkatkan suhu udara sebesar 10 K, suhu proses [K] diciptakan dengan menggunakan proses nonlinier deviasi standar satu K.
3. *Rotational speed* [rpm] dihitung dari *power* 2860 W dengan lapisan noise terdistribusi normal di atas.
4. *Torque* [Nm] tidak bernilai negatif karena nilainya memiliki rata-rata 40 Nm dan deviasi standar 10, maka nilainya terdistribusi secara normal.

3.3. Perancangan Model

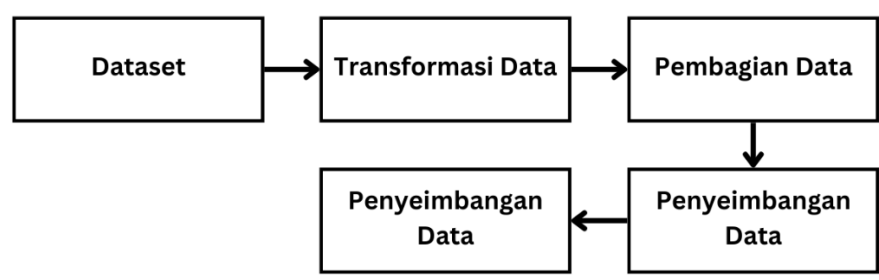
Dalam merancang sebuah model, diperlukan langkah-langkah yang harus dilalui. Secara umum, langkah-langkah ini digunakan dalam penelitian untuk merancang dan membangun model klasifikasi biner. Proses ini mencakup pra-pemrosesan data, pembuatan model (pemodelan), pelatihan model, serta evaluasi menggunakan berbagai metrik kinerja. Langkah-langkah dalam pembuatan model dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur Perancangan Model

3.3.1. Pra-pemrosesan Data

Sebelum memasuki tahap pemodelan, terdapat serangkaian langkah yang harus dilaksanakan. Langkah-langkah ini mencakup tahapan pra-pemrosesan data, yang merupakan prasyarat sebelum inisiasi pelatihan model. Tahapan ini memiliki peran penting dalam memastikan bahwa data yang dimiliki telah lengkap dan terstruktur untuk klasifikasi biner. Oleh karena itu, langkah-langkah pra-pemrosesan data yang akan diimplementasikan dalam penelitian ini, sebagaimana ditampilkan pada Gambar 3.3 menjadi fondasi penting sebelum pelatihan model dapat dimulai.



Gambar 3.3 Pra-pemrosesan Data

Tahap pertama adalah transformasi data, yang bertujuan untuk menyesuaikan skala nilai-nilai fitur agar berada dalam rentang yang sama. Dalam penelitian ini, digunakan *robust scaler* sebagai teknik transformasi data. *Robust scaler* berfungsi untuk penskalaan fitur guna mengatasi *outlier* tanpa menghilangkannya, dengan menggunakan median dan *interquartile range* (IQR). Teknik ini mempertahankan *outlier* tanpa perlu menghilangkannya, sehingga informasi penting tetap terjaga dalam data.

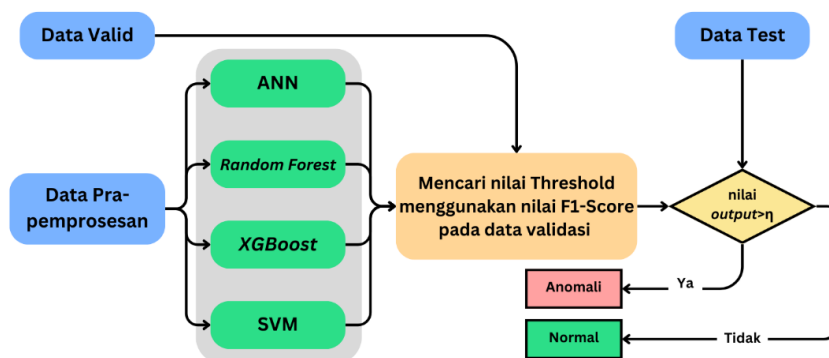
Pembagian data merupakan proses yang cukup vital dalam pra-pemrosesan data. Hal ini disebabkan karena tahap ini menentukan proporsi data yang akan digunakan sebagai data pelatihan, data validasi, dan data pengujian. Sebelum membagi data, langkah pertama yang harus dilakukan adalah memisahkan fitur (X) dan keluaran (y). Bagian X berisi fitur-fitur dari parameter operasional, seperti *air temperature*, *process temperature*, *rotational speed*, dan *torque*. Setelah pemisahan ini dilakukan, dataset kemudian dibagi menjadi beberapa bagian sesuai kebutuhan penelitian. Pada penelitian ini, data dibagi menjadi 70% untuk data latih, 15% untuk

data validasi, dan 15% untuk data uji. Data latih dan data validasi akan digunakan selama pelatihan dan *threshold* untuk mengoptimalkan kinerja model. Setelah itu, data uji digunakan untuk menilai apakah model tersebut memiliki kinerja yang baik pada data baru yang belum pernah dilihat sebelumnya.

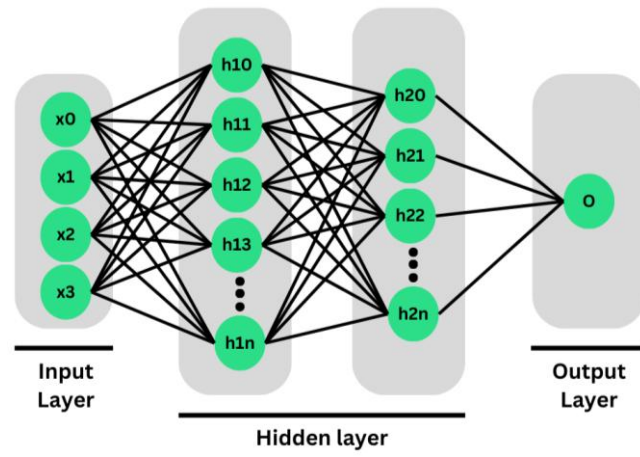
Tahapan berikutnya yang perlu dilakukan dalam deteksi anomali adalah penyeimbangan data, yang bertujuan untuk membantu model agar lebih stabil dalam menangkap pola data. Penyeimbangan data ini hanya dilakukan pada data pelatihan, menggunakan teknik SMOTE. Teknik SMOTE digunakan untuk meningkatkan jumlah sampel pada kelas minoritas, sehingga distribusi data menjadi lebih seimbang dan model dapat belajar secara optimal tanpa bias terhadap salah satu kelas. Kemudian, tahap terakhir dalam pra-pemrosesan adalah pengacakan data. Pengacakan diperlukan untuk mencegah model mempelajari pola yang tidak relevan atau sistematis dalam urutan data. Selain itu, pengacakan membantu mengurangi risiko *overfitting* pada model, sehingga model dapat lebih generalis saat dihadapkan pada data baru.

3.3.2. Perancangan Model

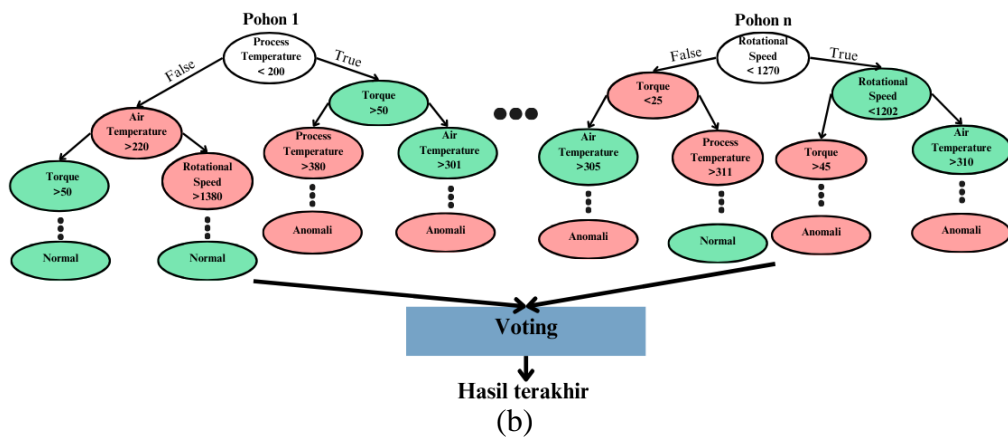
Setelah tahap pra-pemrosesan, data akan diproses menggunakan empat model ML, yaitu ANN, *random forest*, XGBoost, dan SVM. Keempat model dipilih karena model tersebut umum digunakan dan dapat menangani data linier dan nonlinier. Rancangan proses pemodelan secara visual divisualisasikan pada Gambar 3.4, sedangkan detail arsitektur atau struktur masing-masing model ditampilkan pada Gambar 3.5.



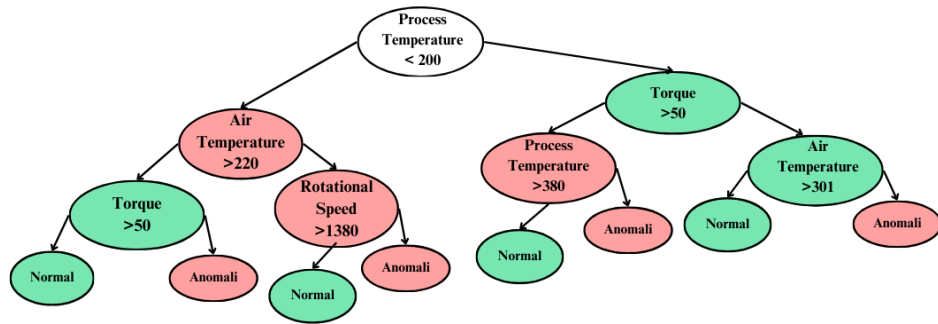
Gambar 3.4 Struktur-struktur metode (a) ANN, (b) Random Forest, (c) XGBoost, dan (d) SVM



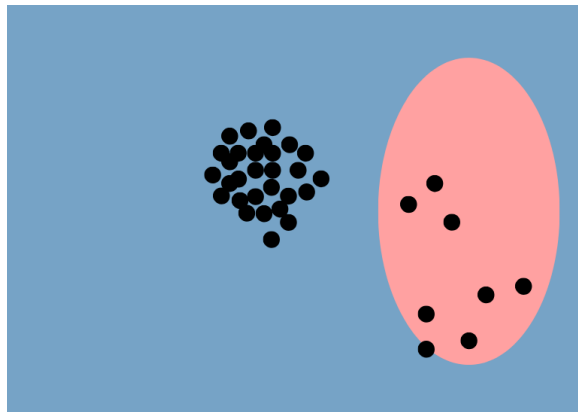
(a)



(b)



(c)



(d)

Gambar 3.5 Diagram Alur Pemodelan dan Pelatihan

Pembuatan model ini menggunakan data latih dalam proses pelatihan yang terlihat pada Gambar 3.4. Data latih setiap model berbeda-beda. Metode *random forest* dan XGBoost menggunakan data latih tanpa melalui transformasi Metode SVM menggunakan data latih melalui transformasi tanpa melalui SMOTE. Metode ANN melalui seluruh tahap pra-pemrosesan. Gambar 3.5a pelatihan ANN akan membentuk 3 lapisan yaitu lapisan input, hidden, dan output. ANN melibatkan forward propagation yaitu data mengalir dari input ke output melalui fungsi aktivasi. Kemudian, dilakukan backpropagation yaitu menyesuaikan bobot berdasarkan error menggunakan optimisasi seperti Adam untuk meminimalkan loss function. Gambar 3.5b pelatihan *random forest* menggunakan teknik bagging dengan membangun banyak decision tree (fitur acak). Hasil dari banyak pohon-pohon digabungkan melalui *voting*/rata-rata untuk meningkatkan akurasi dan mengurangi overfitting. Gambar 3.5c pelatihan XGBoost mengimplementasikan gradient boosting dengan membangun tree secara berurutan (mengoreksi error tree sebelumnya) dan menerapkan regularisasi untuk mengontrol kompleksitas model. Gambar 3.5d pelatihan SVM bertujuan menemukan hyperplane optimal (dengan kernel transformasi) yang memaksimalkan margin. optimisasi fungsi hinge loss dengan batasan regularisasi (parameter C) untuk menyeimbangkan toleransi error.

Setelah model terbentuk, tahap selanjutnya adalah pencarian *threshold*. Penentuan nilai *threshold* ini dilakukan dengan menggunakan data validasi yang

bertujuan untuk memperoleh nilai *f1-score* tertinggi. Pencarian *threshold* ini dilakukan dari 0.0 hingga 1.0.

Setelah nilai *threshold* ditentukan, data uji dimasukkan ke dalam model yang telah dibuat. Kemudian, probabilitasnya dihitung berdasarkan *threshold* yang telah diperoleh. Hasil probabilitas ini akan menentukan apakah data tersebut termasuk anomali atau normal. Jika nilai probabilitas lebih besar dari *threshold*, data akan dianggap sebagai anomali. Sebaliknya, jika nilai probabilitas lebih kecil, data dianggap normal.

3.3.3. Evaluasi Model

Setelah proses pemodelan selesai, langkah berikutnya adalah evaluasi model. Evaluasi ini mencakup penilaian presisi, sensitivitas, *f1-score*, *confusion matrix*, PRC, *g-mean*, dan MCC. Evaluasi model juga mencakup metrik presisi, *recall*, dan *f1-score* yang diperoleh melalui *classification report*. Fokus utama dalam evaluasi ini adalah *recall* pada kelas 1, yang mewakili kategori anomali, karena kemampuan model mendeteksi anomali sangatlah penting. PRC digunakan untuk mengevaluasi model dengan menggambarkan *trade-off* antara presisi dan *recall*. Nilai PRC yang mendekati 0 menunjukkan kinerja model yang buruk, sedangkan nilai yang mendekati 1 menandakan bahwa model mampu membedakan kelas dengan sangat baik. Evaluasi model juga dicek melalui *g-mean* dan MCC. Evaluasi *g-mean* dilakukan untuk mengetahui nilai apakah model memiliki sensitivitas dan spesifisitas yang baik. Evaluasi MCC dilakukan untuk mengetahui apakah model dapat mengklasifikasi dengan baik terhadap data yang tidak seimbang.

3.4. Perangkat Penelitian

Penelitian ini memanfaatkan Google Colab sebagai lingkungan pengembangan berbasis cloud untuk menjalankan program. Spesifikasi perangkat yang digunakan pada Google Colab meliputi prosesor dengan 2 virtual core, memori sebesar 12 GB RAM, serta penyimpanan berkapasitas 5 GB. Sistem operasi yang digunakan adalah Linux, yang berjalan di lingkungan cloud. Google Colab juga menyediakan berbagai perangkat lunak pra-instal, termasuk bahasa

pemrograman seperti Python. Antarmuka pengembangan yang digunakan adalah Jupyter Notebook, yang dapat diakses melalui web.

3.5. Jadwal Kegiatan

Tabel 3. 2 Jadwal Kegiatan

No.	Kegiatan	2024																			
		Maret				April				Mei				Juni				Juli			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur																				
2	Perancangan Sistem Awal																				
3	Penyusunan SUR																				
4	Pra-pemrosesan Data																				
5	Pemodelan ANN																				
6	Perbaikan Model																				
7	Analisis dan pengerjaan laporan tugas akhir																				

BAB IV

HASIL DAN PEMBAHASAN

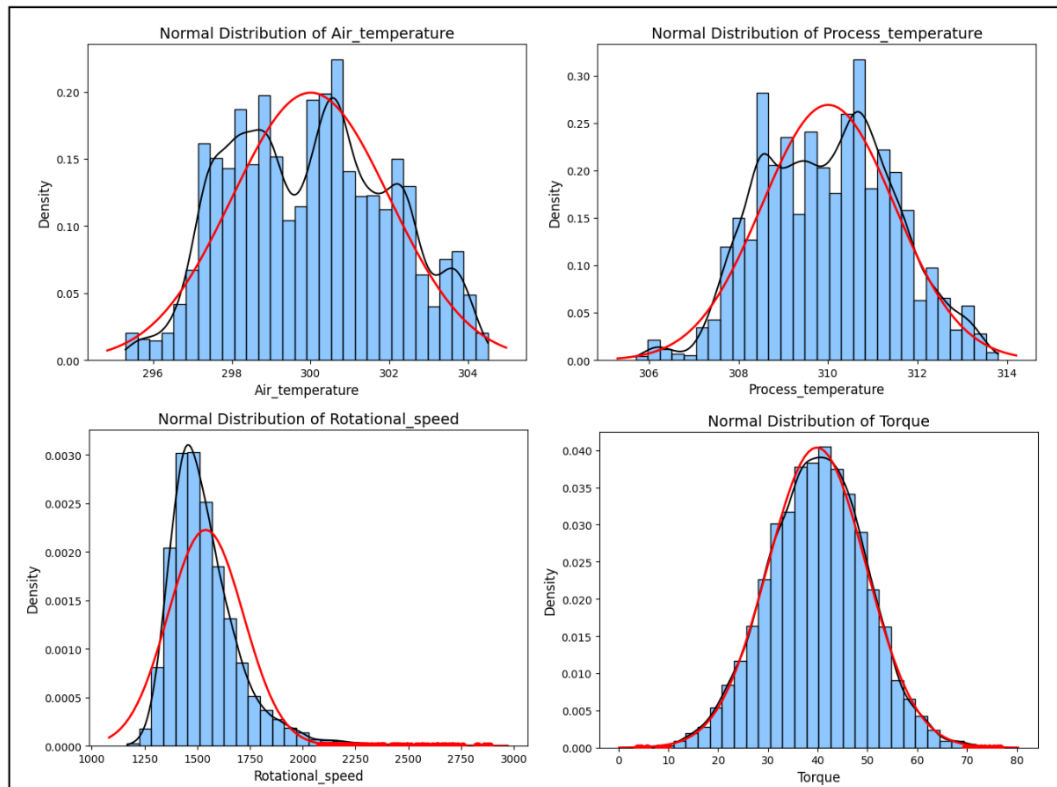
4.1. Dataset

Pada bab ini, dataset dianalisis secara mendalam untuk memahami distribusi data, karakteristik fitur, serta pemanfaatannya dalam pengembangan model klasifikasi. Proses analisis melibatkan visualisasi distribusi variabel-variabel penting, serta pengelompokan ulang tipe kegagalan guna mengidentifikasi pola-pola yang signifikan dalam data. Sebelum masuk ke tahap visualisasi, penting untuk memahami lebih lanjut bagaimana setiap variabel dalam dataset ini dihasilkan dan apa saja karakteristiknya. Karakteristik tersebut terdapat pada tabel 4.1.

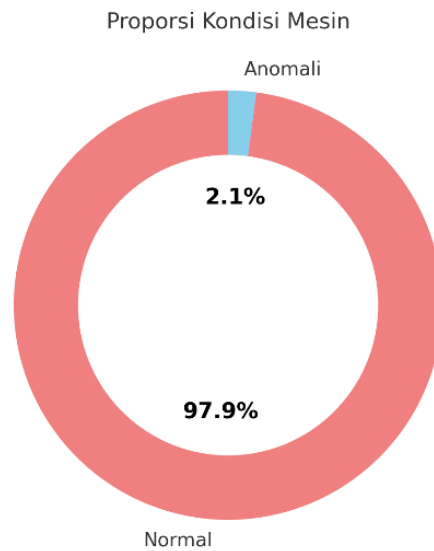
Tabel 4.1 Karakteristik Data

	Air Temperature	Process Temperature	Rotational Speed	Torque
Mean	300	310	1540	39.86
Standar deviasi	2.00	1.48	179.12	9.88
Nilai minimal	295.3	305.7	1168	3.8
Nilai maksimal	304.5	313.8	2886	76.6

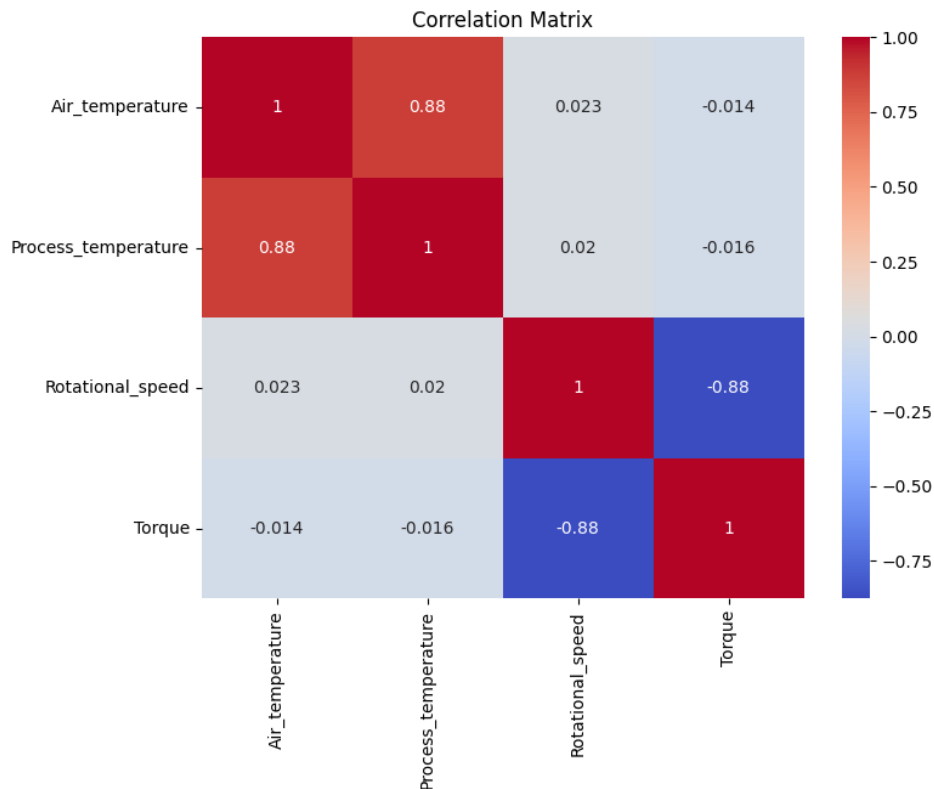
Visualisasi dilakukan pada variabel-variabel seperti *air temperature*, *process temperature*, *rotational speed*, *torque*, dan *target*. Variabel yang bertipe kategorikal yaitu *target* divisualisasikan menggunakan *pie chart*. Variabel yang bertipe *numerical* yaitu *air temperature*, *process temperature*, *rotational speed*, dan *torque* divisualisasikan menggunakan distribusi normal. Visualisasi distribusi untuk *variable numerical* dapat dilihat pada Gambar 4.1, sedangkan visualisasi *bar chart* untuk *variable categorical* dapat dilihat pada Gambar 4.2. Selain itu, terdapat juga hubungan antar fitur-fitur melalui korelasi matriks yang terlihat pada Gambar 4.3.



Gambar 4.1 Visualisasi Distribusi Data Fitur Numerikal



Gambar 4.2 Visualisasi Data Variabel Target



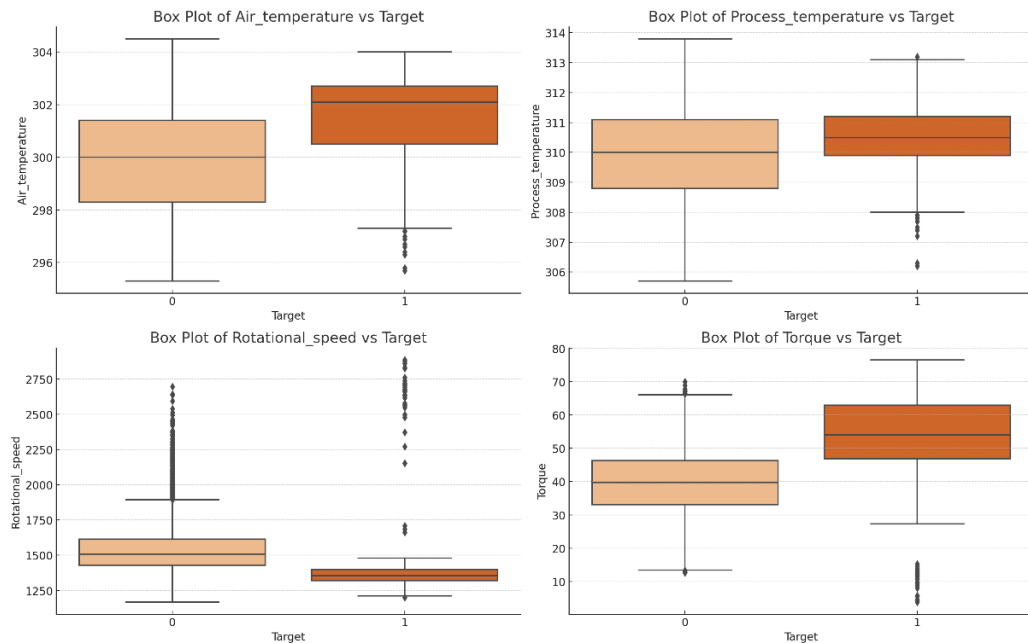
Gambar 4.3 Matrik Korelasi pada Dataset

Selanjutnya, hubungan korelasi antar variabel dalam penelitian ini dapat dilihat pada Gambar 4.3. Berdasarkan gambar tersebut, terdapat dua hubungan utama antar variabel, yaitu hubungan antara *air temperature* dengan *process temperature* serta hubungan antara *torque* dengan *rotational speed*. Hubungan antara *air temperature* dan *process temperature* menunjukkan korelasi positif, yang mengindikasikan bahwa ketika suhu udara meningkat, suhu proses juga cenderung meningkat. Sebaliknya, hubungan antara *torque* dan *rotational speed* menunjukkan korelasi negatif, yang berarti bahwa peningkatan torsi justru diikuti dengan penurunan kecepatan rotasi, atau sebaliknya. Pemahaman mengenai hubungan ini sangat penting untuk analisis lebih lanjut dan untuk pengembangan model yang lebih akurat dalam mendeteksi anomali dalam data yang dianalisis.

Selain visualisasi distribusi normal dan diagram lingkaran (*pie chart*), analisis juga dilakukan menggunakan *box plot* untuk variabel numerikal seperti *air temperature*, *process temperature*, *rotational speed*, *torque*, dan *tool wear* terhadap variabel target (normal = 0 dan anomali = 1). Penggunaan *box plot* sangat berguna

untuk memahami distribusi dan variasi dari masing-masing variabel, serta untuk mendeteksi *outlier* yang mungkin memberikan indikasi adanya anomali dalam data.

Hasil visualisasi menggunakan *box plot* ini ditampilkan pada Gambar 4.4



Gambar 4. 4 *Box Plot* Fitur Numerikal terhadap Target

Beberapa temuan penting dari visualisasi ini adalah sebagai berikut:

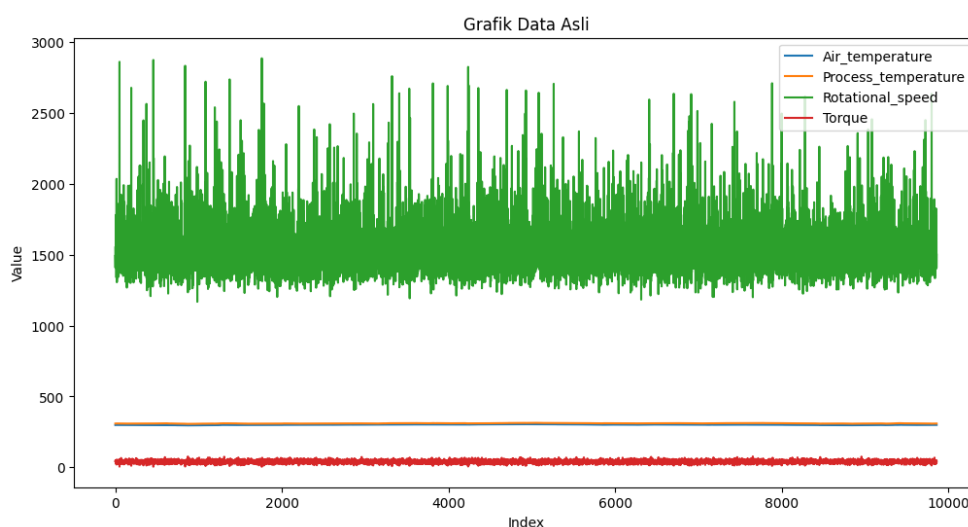
1. *Air temperature* dan *process temperature* terlihat cukup stabil baik di kategori normal maupun anomali. Tidak terdapat *outlier* atau nilai ekstrem yang mencolok. Dengan demikian, dapat disimpulkan bahwa *air temperature* dan *process temperature* mungkin bukan faktor utama penyebab anomali dalam sistem.
2. *Rotational speed* mulai menunjukkan sesuatu yang menarik. Variabel ini memiliki beberapa *outlier* pada kategori normal yang artinya ada kondisi operasi ekstrem namun tidak memicu anomali. Namun, dalam kategori anomali, nilai median lebih rendah. Ini bisa menjadi tanda bahwa penurunan kecepatan rotasi mungkin terkait dengan anomali.
3. *Torque* juga menunjukkan *outlier* pada kategori normal yang berarti ada variabilitas torsi yang tinggi dalam operasi tanpa kegagalan. Di kategori anomali, nilai median torsi lebih tinggi yang menunjukkan bahwa torsi tinggi bisa menjadi salah satu faktor yang memicu terjadinya anomali.

4.2. Pra-pemrosesan

Setelah pengelompokan dataset selesai dilakukan, tahapan selanjutnya adalah pra-pemrosesan. Pengolahan ini penting untuk memastikan bahwa *dataset* siap digunakan dalam pelatihan model klasifikasi. Proses pra-pemrosesan mencakup beberapa tahapan penting, antara lain pelabelan data, normalisasi data, serta pembagian data menjadi data latih, data validasi, dan data uji. Pelabelan data bertujuan untuk memberikan identitas yang jelas pada setiap kategori dalam dataset, sementara normalisasi data dilakukan untuk menyeimbangkan skala variabel-variabel numerik agar model dapat dilatih dengan lebih efektif. Tahapan pembagian data kemudian memastikan bahwa *dataset* dapat digunakan secara optimal untuk melatih dan menguji performa model klasifikasi yang akan dikembangkan.

4.2.1. Transformasi Data

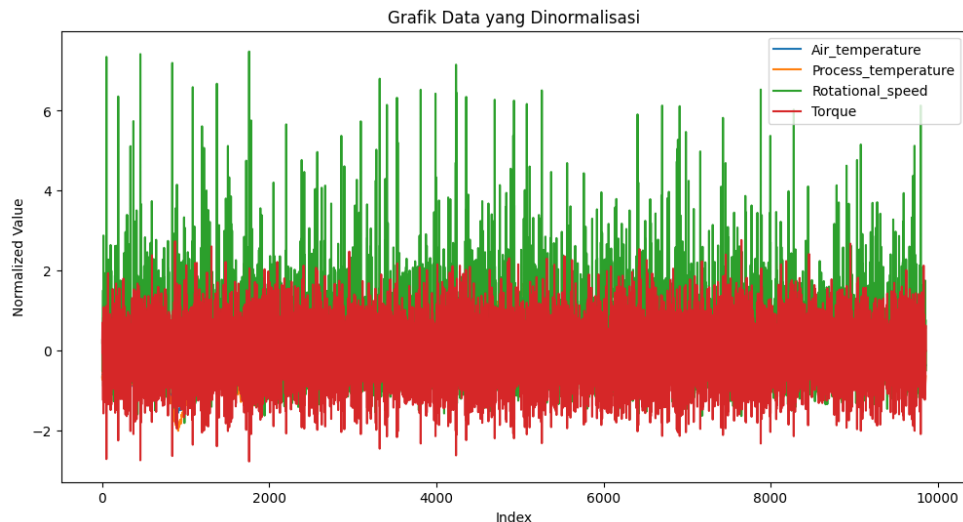
Pada tahap ini, transformasi data dilakukan menggunakan *robustscaler* untuk menangani data yang mengandung *outliers*. Variabel yang dilalui tahap ini yaitu *air temperature*, *process temperature*, *rotational speed*, dan *torque*. *Range* variabel-variabel terlihat pada Gambar 4.5. Rentang nilai antara variabel ini masih berjauhan tidak berseragam.



Gambar 4.5 Grafik antar fitur-fitur sebelum dilakukan normalisasi

Hasil tahapan ini memiliki rentang yang lebih seragam. Hal ini terlihat setelah dilakukan normalisasi terlihat bahwa data tersebut memiliki *range* yang

jelas dan terpola seperti Gambar 4.6. rentang yang terlihat pada gambar tersebut dari -2 hingga 6.



Gambar 4.6 Grafik antar fitur-fitur setelah dilakukan normalisasi

4.2.2. Pembagian Data

Setelah data melalui proses normalisasi, data tersebut dibagi menjadi dua bagian, yaitu X dan y . Data X merupakan fitur-fitur yang merepresentasikan target dan dijadikan output model. Fitur-fitur dalam data X meliputi *air temperature*, *process temperature*, *torque*, dan *rotational speed*. Sementara itu, data y berfungsi sebagai target kelas dalam klasifikasi, yang berisi kategori-kategori pada variabel yaitu normal dan anomali.

Kemudian dilakukan pembagian porsi data latih, data validasi, dan data pengujian. Pembagian data tersebut terinci sebagai berikut: 70% data latih; 15% data validasi; dan 15% data uji. Implementasi pembagian data dapat dilihat pada Gambar 4.7.

```

Ukuran Data latih X dan y : (6901,4), (6901,)
Ukuran Data validasi X dan y : (1479,4), (1479,)
Ukuran Data uji X dan y : (1479,4), (1479,)

```

Gambar 4.7 Hasil Pembagian Data Latih, Data Validasi, dan Data Uji

4.2.3. Penyeimbangan Data

Pada tahap ini, dilakukan penyeimbangan data untuk mengatasi permasalahan ketidakseimbangan kelas pada data pelatihan dan validasi. Data

pelatihan dan validasi yang digunakan dalam penelitian ini memiliki ketidakseimbangan antara kelas mayoritas (dengan target 0) dan kelas minoritas (dengan target 1). Untuk mengatasi hal tersebut, digunakan teknik *oversampling* pada kelas minoritas.

SMOTE dipilih menjadi teknik *oversampling* karena menciptakan data sintetik baru dengan interpolasi antara contoh-contoh yang ada, sehingga menambah variasi pada kelas minoritas. Dengan menerapkan SMOTE, jumlah data pada kelas minoritas menjadi sebanding dengan jumlah data pada kelas mayoritas. Hasil dari proses ini adalah data pelatihan yang lebih seimbang antara kelas mayoritas dan minoritas, yang diharapkan dapat meningkatkan performa model dalam mengklasifikasikan kedua kelas dengan lebih baik. Hasil tersebut dapat terlihat pada Gambar 4.8.

```
Ukuran Data latih X dan y : (6901,4), (6901,)
Ukuran Data latih X dan y : (13498,4), (13498,)
Nilai y pada train kelas 0 : 6763
Nilai y pada train kelas 1 : 237
Nilai y pada train setelah penyeimbangan kelas 0 : 6763
Nilai y pada train setelah penyeimbangan kelas 1 : 6763
```

Gambar 4.8 Perbedaan Data Sebelum dan Sesudah Diseimbangkan

4.2.4. Pengacakan Data

Proses terakhir yaitu pengolahan data dengan teknik mengacak (*shuffling*). Data pelatihan yang sudah dilakukan penyeimbangan diacak secara serentak. Pengacakan ini menggunakan *library* scikit-learn. Hasil dari pengacakan ini yang dilatih oleh model.

4.3. Pemodelan

Setelah tahapan pengolahan data selesai, proses selanjutnya adalah pemodelan. Pemodelan ini mencakup perancangan dan pelatihan model yang dirancang untuk mengklasifikasikan data menjadi dua kelompok, yaitu “normal” dan “anomali”. Model klasifikasi yang digunakan adalah ANN, *random forest*, XGBoost, dan SVM yang dibangun menggunakan *library tensorflow* dan *keras*.

4.3.1. Arsitektur Model

Setelah melakukan pra-pemrosesan data, langkah berikutnya perancangan model. Perancangan model yang digunakan dalam pemodelan memiliki arsitektur terlihat pada tabel 4.2.

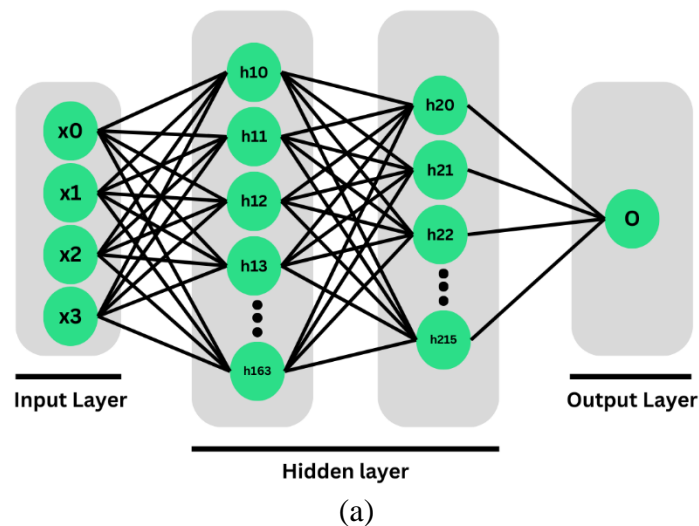
Tabel 4.2 Hyperparameter model

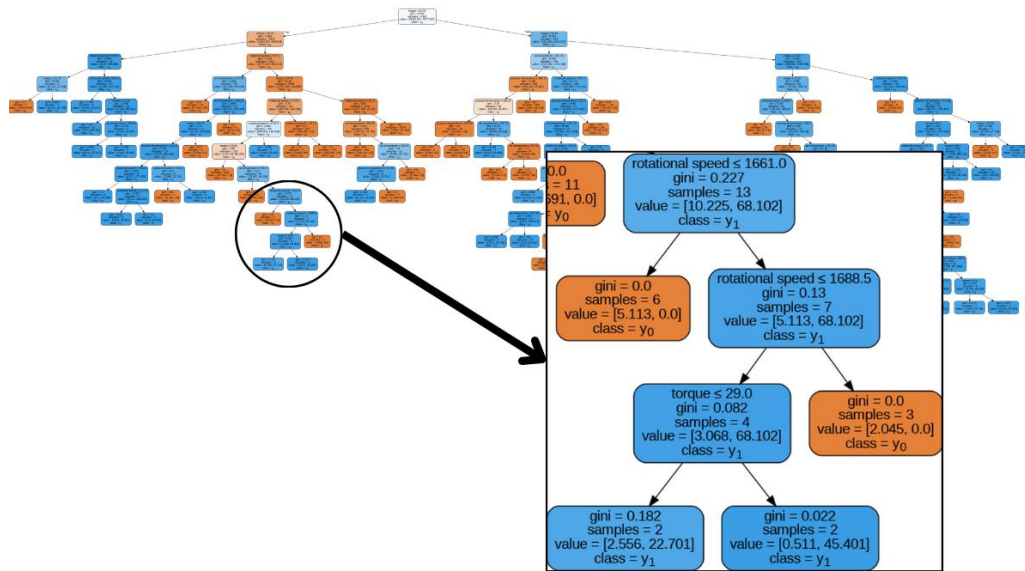
No	Model	Hyperparameter
1	ANN	1. Input 4 neuron 2. Hidden layer terdapat 2 layer 3. Hidden layer pertama memiliki 64 neuron dengan aktivasi relu 4. Hidden layer Kedua memiliki 16 neuron dengan aktivasi relu 5. Terdapat dropout bernilai 0,2 diletakan setelah hidden layer pertama dan kedua 6. Output memiliki 1 neuron dengan aktivasi sigmoid 7. optimasi : Adam 8. <i>epoch</i> : 200 9. <i>batchsize</i> : 64 10. <i>random state</i> : 42 11. <i>loss</i> : <i>binary cross entropy</i>
2	<i>Random Forest</i>	1. <i>n estimator</i> : 100 2. <i>criterion</i> : gini 3. <i>min_samples_split</i> : 2 4. <i>min_samples_leaf</i> : 2 5. <i>max_features</i> : <i>sqrt</i> 6. <i>max_depth</i> : <i>none</i> 7. <i>bootstrap</i> : <i>True</i> 8. <i>class weight</i> : <i>balanced</i> 9. <i>random state</i> : 42
3	XGBoost	1. <i>learning rate</i> : 0,22 2. <i>n estimator</i> : 600 3. <i>gamma</i> : 2 4. <i>max_depth</i> : 4 5. <i>min_child_weight</i> : 3 6. <i>reg_lambda</i> : 1 7. <i>reg_alpha</i> : 2 8. <i>subsample</i> : 0.7 9. <i>colsample_bytree</i> : 0.7 10. <i>scale_pos_weight</i> : <i>balanced</i> 11. <i>random state</i> : 42
4	SVM	1. <i>c</i> : 99,78 2. <i>gamma</i> : 0,99 3. <i>probability</i> : <i>True</i>

		4. <i>class weight : balanced</i> 5. <i>kernel : rbf</i> 6. <i>random state : 42</i>
--	--	--

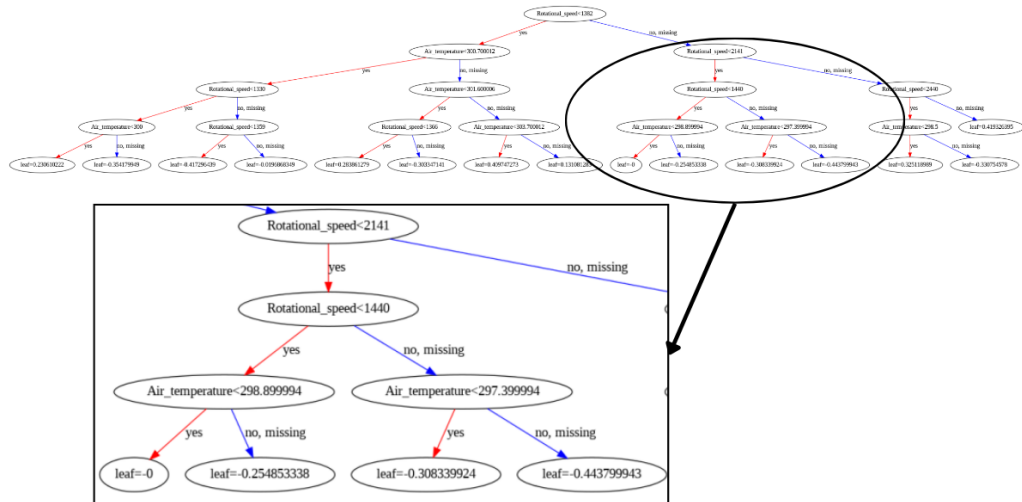
4.3.2. Pelatihan Model

Pada tahap pelatihan model, metode ANN, *XGBoost*, *random forest*, dan SVM diterapkan untuk melatih model menggunakan data yang telah dipersiapkan sebelumnya. Hasil pelatihan model mencakup struktur model dan riwayat pelatihan yang berupa kurva. Struktur model menggambarkan bentuk model yang telah dihasilkan dan dapat dilihat pada Gambar 4.9. Riwayat pelatihan menunjukkan hasil yang memperlihatkan nilai loss dari model-model yang dilatih. Model ANN menggunakan kurva riwayat pelatihan, sementara model *XGBoost*, *random forest*, dan SVM menggunakan kurva pembelajaran kesalahan klasifikasi. Kurva-kurva tersebut dapat dilihat pada Gambar 4.10.

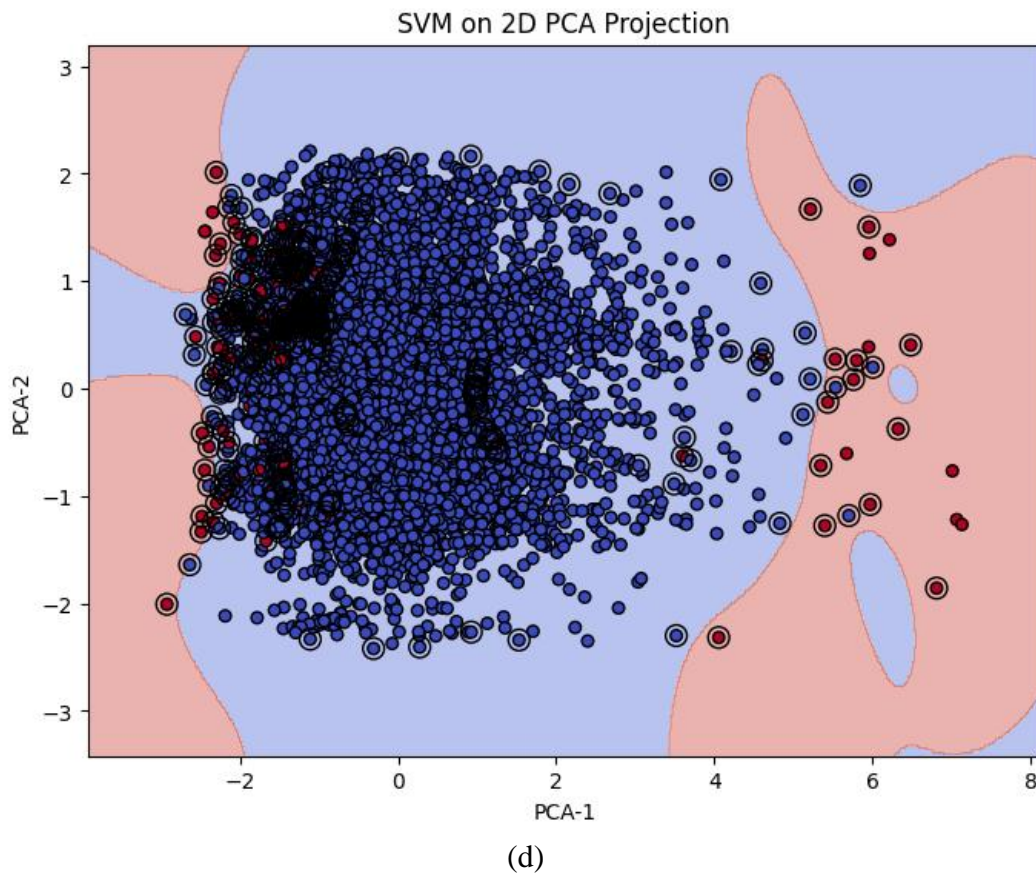




(b)



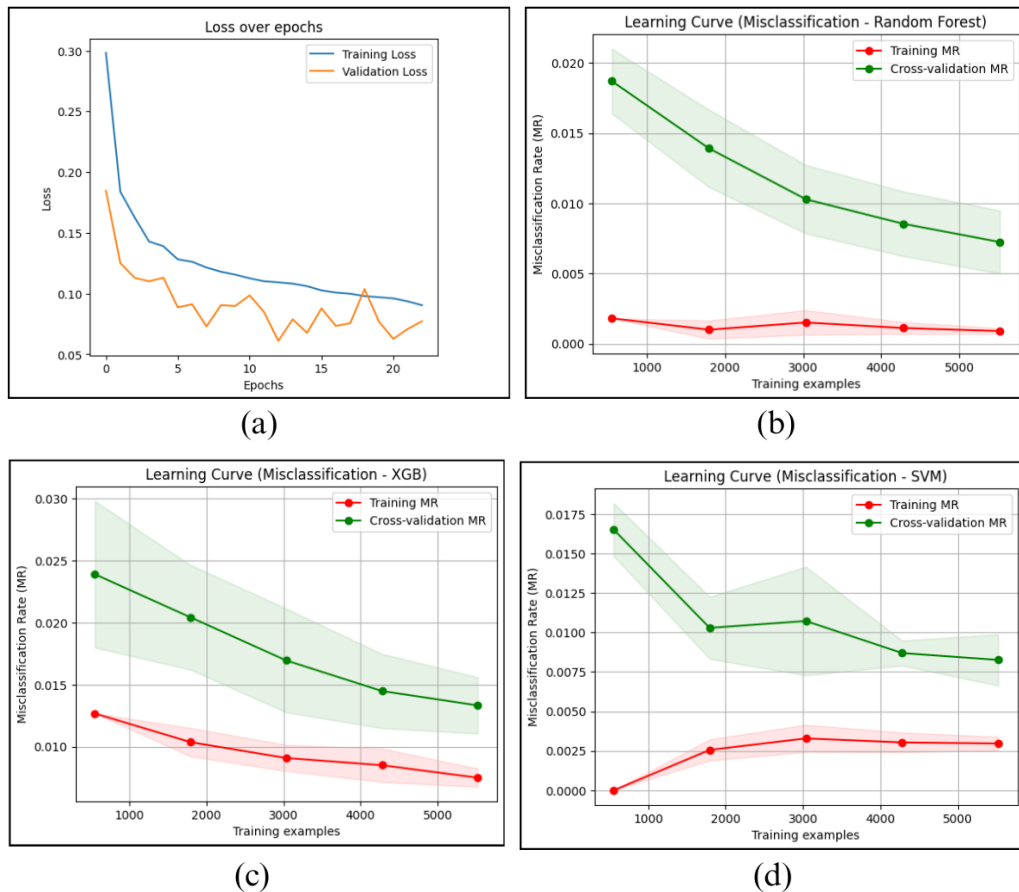
(c)



Gambar 4.9 Struktur model pada (a) ANN, (b) *Random Forest* pohon ke-100, (c) XGBoost, dan (d) SVM

Berdasarkan Gambar 4.9 menggambarkan berbagai model dalam ML. Gambar 4.9a menunjukkan ANN, di mana data mengalir dari lapisan input melalui lapisan tersembunyi ke lapisan output dan sebaliknya. Gambar 4.9b dan 4.9c menggambarkan pohon keputusan yang membagi data secara rekursif berdasarkan kondisi fitur seperti suhu atau kecepatan rotasi, menghasilkan prediksi di node terminal. Gambar 4.9b diambil dari pohon ke-1 dari 100 pohon yang dilatih. Gambar 4.9d memvisualisasikan data yang direduksi dimensinya menggunakan PCA, di mana titik-titik data dipisahkan dalam dua komponen utama (PCA-1 dan PCA-2), dengan batasan yang menunjukkan pemisahan antar kelas, membantu memahami struktur data dalam dimensi yang lebih rendah. Dalam hal ini, titik yang berada di luar batasan (misalnya, berwarna merah) bisa dianggap sebagai anomali atau titik yang berbeda dari pola umum. Namun, titik yang berada di dalam batasan (misalnya, berwarna biru) dapat dianggap sebagai data normal. Batasan ini

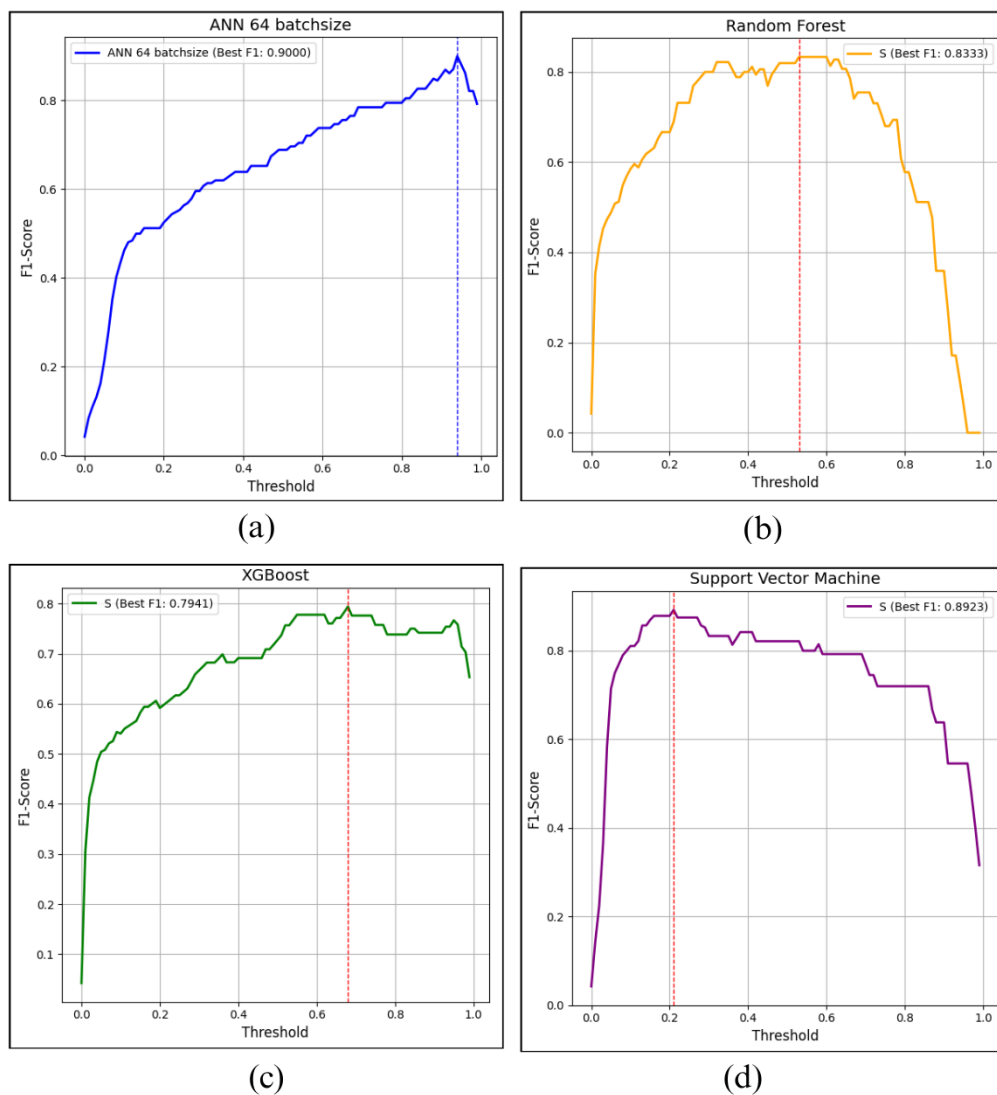
menunjukkan bagaimana model memisahkan data yang berbeda, dengan titik merah yang mungkin menandakan data yang tidak sesuai dengan pola yang dipelajari model.



Gambar 4.10 Kurva Riwayat Pelatihan atau Kurva Pembelajaran Kesalahan Klasifikasi pada (a) ANN, (b) *Random Forest*, (c) XGBoost, dan (d) SVM

Berdasarkan gambar diatas, Gambar 4.10a terlihat bahwa nilai loss pada khususnya *training* menurun, tetapi validasi terdapat fluktuatif. Namun, secara siklus seharusnya bisa hingga 200 *epoch* dirasakan terlalu fluktuatif sehingga berhenti sekitar 20-25 *epoch*. Gambar 4.10b terlihat *training* MR tetap rendah sedangkan *cross-validation* MR menurun seiring bertambahnya data pelatihan. Gambar 4.10c terlihat seiring bertambahnya jumlah data pelatihan, baik error pada data latih maupun validasi cenderung menurun. Kemudian, Gambar 4.10d terlihat awal kondisi berjauhan namun menjadi mendekat antara garis *training* dan garis *cross validation* setelah beberapa sampel.

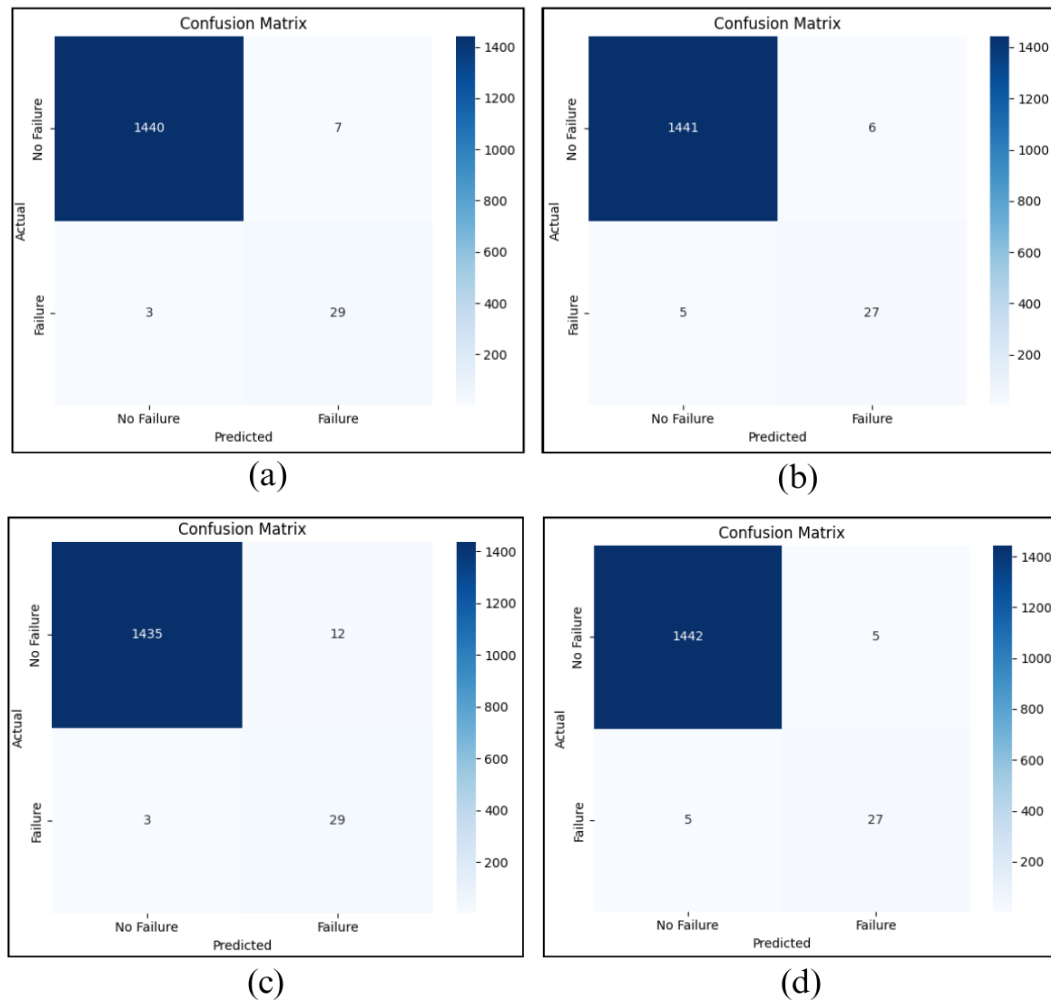
Setelah itu, pengoptimalan model menggunakan *f1-score* untuk menentukan *threshold* dari klasifikasi. Masing-masing model dicari ambang batas melalui data validasi yang sudah melalui tahap probabilistik dari model. Pencarian ambang batas mulai dari 0 hingga 1 dengan 0,01. Hasil dari pencarian tersebut terdapat pada Gambar 4.11. Gambar 4.10a model ANN memiliki *threshold* bernilai 0,94. Gambar 4.11b model *random forest* memiliki *threshold* bernilai 0,53. Gambar 4.11c model XGBoost memiliki ambang batas bernilai 0,68. Gambar 4.11d model SVM memiliki ambang batas bernilai 0,21.



Gambar 4.11 Hasil Pencarian Ambang Batas (a) ANN, (b) *Random Forest*, (c) XGBoost, dan (d) SVM

4.4. Evaluasi Model

Evaluasi model dilakukan dengan pertama dinilai *classification report*. Performa model meliputi akurasi, presisi, *recall*, dan *f1-score*. Fokus utama pada data minoritas yang dilihat berdasarkan *recall*, presisi, dan *f1-score*. Evaluasi terhadap *classification report* tercantum pada table 4.3. Evaluasi tersebut *classification report* ditindak lanjuti menggunakan *confusion matrix* menunjukkan distribusi prediksi keempat model. Hal ini bisa dilihat pada Gambar 4.12.

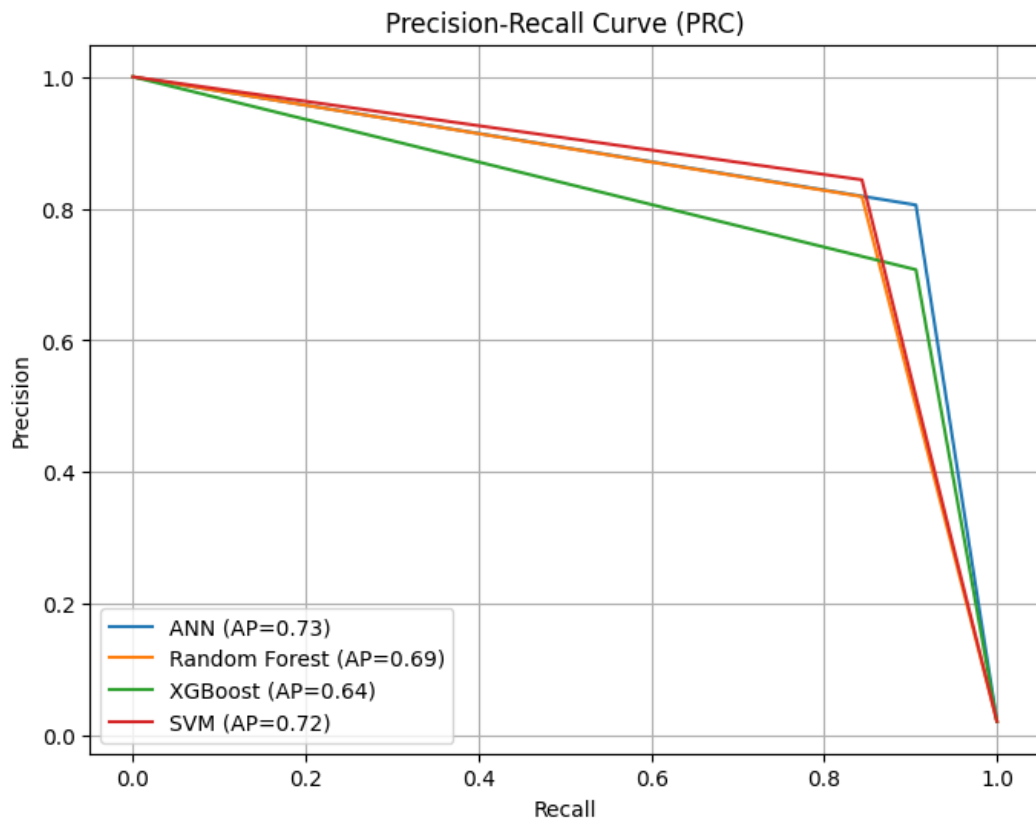


Gambar 4.12 *Confusion matrix* pada Model (a) ANN, (b) *Random Forest*, (c) XGBoost, dan (d) SVM

Tabel 4.3 Classification report pada anomali seluruh metode

Model	Presisi	Recall	F1-Score
ANN	0,81	0,91	0,85
<i>Random Forest</i>	0,82	0,84	0,83
XGBoost	0,71	0,91	0,79
SVM	0,84	0,84	0,84

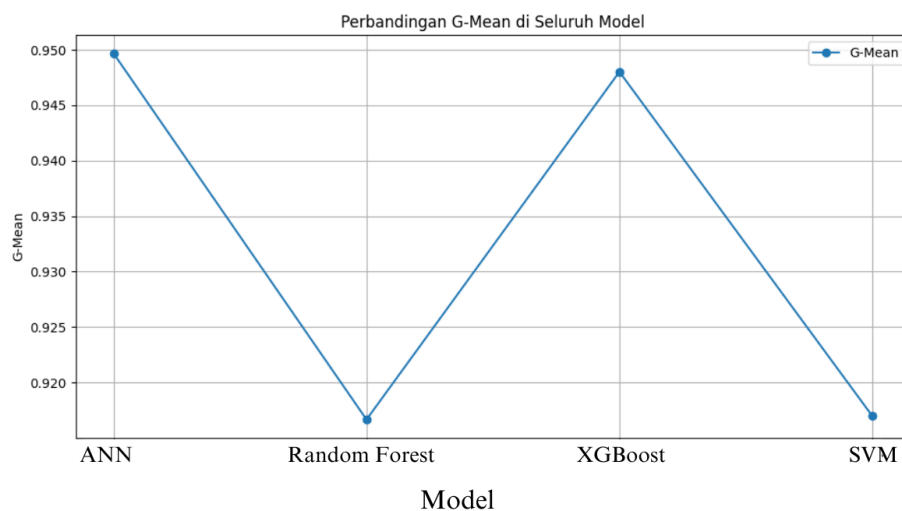
Evaluasi ketiga dinilai dari PRC. PRC dilihat berdasarkan *trade-off* antara *recall* dan presisi pada seluruh model, seperti terlihat pada Gambar 3.3. Berdasarkan Gambar 3.3, model ANN, SVM, dan *random forest* terlihat lebih baik dibandingkan model XGBoost. Model XGBoost tersebut jika dilihat berdasarkan nilai AUC-PRC bernilai 0,64 yang menunjukkan model paling rendah dari ketiga model yang lain. Dari ketiga model yang lebih baik, model ANN menjadi terbaik dari ketiganya.



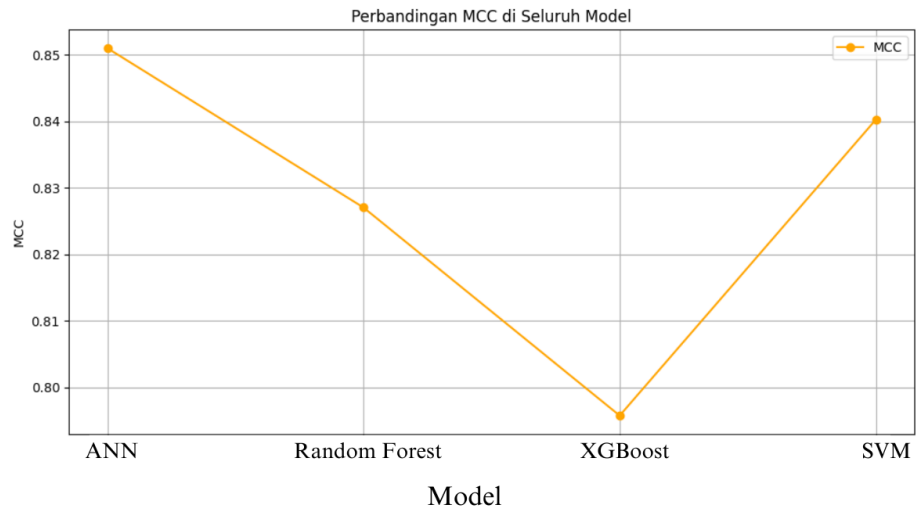
Gambar 4.13 PRC Keempat Model

Evaluasi berikutnya yaitu *g-mean* dan MCC. Gambar 4.14 menunjukkan perbandingan *g-mean* di antara empat model. *G-mean* dilakukan untuk menilai keseimbangan sensitivitas dan spesifitas terutama pada *dataset* yang tidak seimbang. Seluruh model memiliki nilai sensitivitas dan spesifitas yang tinggi yang berada diatas 0,91. Dari seluruh model memiliki keseimbangan yang baik. Nilai masing-masing model ini memiliki nilai-nilai yang berdekatan dan tidak terlalu jauh antar nilainya. Model ANN memiliki nilai *g-mean* tertinggi dengan nilai 0,95. Hal tersebut menunjukkan performa yang lebih baik dalam menjaga keseimbangan antara sensitivitas dan spesifisitas. Sebaliknya, model SVM menunjukkan performa yang lebih rendah dari seluruh model dengan nilai *g-mean* sekitar 0,91.

Gambar 4.15 merupakan nilai MCC dari masing-masing model untuk mengevaluasi performa model biner. Model ini berurutan dari ANN, SVM, *random forest*, dan XGBoost. Nilai MCC pada model ANN yaitu 0,85. Model SVM memiliki nilai MCC sebesar 0,84. Performa model *random forest* memiliki nilai MCC sebesar 0,83. Model XGBoost menunjukkan nilai 0,80 pada evaluasi MCC yang merupakan nilai terendah dari ketiga model lainnya.



Gambar 4.14 Perbandingan Nilai G-Mean Antar Model



Gambar 4.15 Perbandingan Nilai MCC Antar Model

Berdasarkan hasil seluruh evaluasi pada model-model yang telah dilakukan, dapat dirangkum di tabel 4.4.

Tabel 4.4 Hasil evaluasi keempat model

No	Model	Presisi	Recall	F1-score	PRC	G-mean	MCC
1	ANN	0,81	0,91	0,85	0,73	0,95	0,85
2	Random Forest	0,82	0,84	0,83	0,69	0,92	0,83
3	XGBoost	0,71	0,91	0,79	0,64	0,94	0,80
4	SVM	0,84	0,84	0,84	0,72	0,91	0,84

BAB V

KESIMPULAN

5.1. Kesimpulan

Penelitian ini membandingkan keempat metode yaitu ANN, *random forest*, XGBoost, dan SVM dalam mendeteksi anomali melalui klasifikasi biner pada kondisi mesin. Hasil penelitian mengungkapkan bahwa model ANN unggul dengan nilai-nilai yang terbaik berdasarkan evaluasi dibandingkan metode lain. Hal ini terlihat dari evaluasi *classification report* pada kelas anomali yaitu nilai presisi 0,81, nilai recall 0,91, dan nilai f1-score 0,85. Jika melihat kurva PRC, nilai ANN lebih unggul. Selain itu, nilai *g-mean* dan MCC unggul yaitu masing-masing bernilai 0,95 dan 0,85.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan terdapat saran-saran yang dapat dikembangkan selanjutnya, diantara lain:

1. Eksplorasi penggunaan teknik penyeimbangan data seperti ADASYN yang memiliki kemampuan adaptif terhadap data anomali yang sulit diklasifikasi.
2. Pertimbangkan juga *tuning* parameter untuk meningkatkan efektivitasan model
3. Implementasikan model dengan data mesin di lapangan sehingga dapat menilai kinerja model dalam kondisi nyata

DAFTAR PUSTAKA

- [1] Z. Zhang, Y. Qi, Q. Cheng, Z. Liu, Z. Tao, dan L. Cai, "Machining accuracy reliability during the peripheral milling process of thin-walled components," *Robot Comput Integr Manuf*, vol. 59, 2019, doi: 10.1016/j.rcim.2019.04.002.
- [2] M. Assafo, J. P. Städter, T. Meisel, dan P. Langendörfer, "On the Stability and Homogeneous Ensemble of Feature Selection for Predictive Maintenance: A Classification Application for Tool Condition Monitoring in Milling," *Sensors*, vol. 23, no. 9, 2023, doi: 10.3390/s23094461.
- [3] Y. Peng, M. Dong, dan M. J. Zuo, "Current status of machine prognostics in condition-based maintenance: A review," *International Journal of Advanced Manufacturing Technology*, vol. 50, no. 1–4, 2010, doi: 10.1007/s00170-009-2482-0.
- [4] A. Bousdekis, K. Lepenioti, D. Apostolou, dan G. Mentzas, "A review of data-driven decision-making methods for industry 4.0 maintenance applications," 2021. doi: 10.3390/electronics10070828.
- [5] Z. Xu dan J. H. Saleh, "Machine learning for reliability engineering and safety applications: Review of current status and future opportunities," 2021. doi: 10.1016/j.res.2021.107530.
- [6] M. Pech, J. Vrchota, dan J. Bednář, "Predictive maintenance and intelligent sensors in smart factory: Review," 2021. doi: 10.3390/s21041470.
- [7] S. G. Yang, B. J. Kim, S. W. Son, dan H. Kim, "Power-grid stability predictions using transferable machine learning," *Chaos*, vol. 31, no. 12, 2021, doi: 10.1063/5.0058001.
- [8] A. K. Bashir dkk., "Comparative analysis of machine learning algorithms for prediction of smart grid stability†," dalam *International Transactions on Electrical Energy Systems*, 2021. doi: 10.1002/2050-7038.12706.

- [9] H. Shamsudin, U. K. Yusof, Y. Haijie, dan I. S. Isa, "AN OPTIMIZED SUPPORT VECTOR MACHINE WITH GENETIC ALGORITHM FOR IMBALANCED DATA CLASSIFICATION," *J Teknol*, vol. 85, no. 4, 2023, doi: 10.11113/jurnalteknologi.v85.19695.
- [10] Q. A. Al-haija, A. A. Smadi, dan M. F. Allehyani, "Meticulously intelligent identification system for smart grid network stability to optimize risk management," *Energies (Basel)*, vol. 14, no. 21, 2021, doi: 10.3390/en14216935.
- [11] N. Mostafa, H. S. M. Ramadan, dan O. Elfarouk, "Renewable energy management in smart grids by using big data analytics and machine learning," *Machine Learning with Applications*, vol. 9, 2022, doi: 10.1016/j.mlwa.2022.100363.
- [12] S. Tiwari dkk., "Machine learning-based model for prediction of power consumption in smart grid- smart way towards smart city," *Expert Syst*, vol. 39, no. 5, 2022, doi: 10.1111/exsy.12832.
- [13] H. I. Hussein, S. A. Anwar, dan M. I. Ahmad, "Imbalanced Data Classification Using SVM Based on Improved Simulated Annealing Featuring Synthetic Data Generation and Reduction," *Computers, Materials and Continua*, vol. 75, no. 1, 2023, doi: 10.32604/cmc.2023.036025.
- [14] S. Mohsen, M. Bajaj, H. Kotb, M. Pushkarna, S. Alphonse, dan S. S. M. Ghoneim, "Efficient Artificial Neural Network for Smart Grid Stability Prediction," *International Transactions on Electrical Energy Systems*, vol. 2023, 2023, doi: 10.1155/2023/9974409.
- [15] M. Önder, M. U. Dogan, dan K. Polat, "Classification of smart grid stability prediction using cascade machine learning methods and the internet of things in smart grid," *Neural Comput Appl*, vol. 35, no. 24, 2023, doi: 10.1007/s00521-023-08605-x.

- [16] S. Matzka, “AI4I 2020 Predictive Maintenance Dataset”, www.explore.ai/dataset/predictiveMaintenanceDataset.csv, submitted to UCI Machine Learning Repository, 2020.
- [17] S. Matzka, “Explainable Artificial Intelligence for Predictive Maintenance Applications,” dalam *Proceedings - 2020 3rd International Conference on Artificial Intelligence for Industries, AI4I 2020*, 2020. doi: 10.1109/AI4I49448.2020.00023.
- [18] M. Züfle, F. Moog, V. Lesch, C. Krupitzer, dan S. Kounev, “A machine learning-based workflow for automatic detection of anomalies in machine tools,” *ISA Trans*, vol. 125, 2022, doi: 10.1016/j.isatra.2021.07.010.
- [19] B. Roberson, “What Are the Vital Parts of a CNC Milling Machine? | CNC Machines | Roberson Machine Company,” Roberson Tool, Nov. 07, 2022. <https://robersontool.com/what-are-the-vital-parts-of-a-cnc-milling-machine/>.
- [20] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning,” *Genet Program Evolvable Mach*, vol. 19, no. 1–2, 2018, doi: 10.1007/s10710-017-9314-z.
- [21] D. B. Mulindwa dan S. Du, “An n-Sigmoid Activation Function to Improve the Squeeze-and-Excitation for 2D and 3D Deep Networks,” *Electronics (Switzerland)*, vol. 12, no. 4, 2023, doi: 10.3390/electronics12040911.
- [22] D. Ramos, J. Franco-Pedroso, A. Lozano-Diez, dan J. Gonzalez-Rodriguez, “Deconstructing cross-entropy for probabilistic binary classifiers,” *Entropy*, vol. 20, no. 3, 2018, doi: 10.3390/e20030208.
- [23] D. Wang, X. Liu, dan J. Zhang, “Improved Vanishing Gradient Problem for Deep Multi-layer Neural Networks,” dalam *Communications in Computer and Information Science*, 2023. doi: 10.1007/978-981-99-0617-8_12.

- [24] X. Jin, C. Xu, J. Feng, Y. Wei, J. Xiong, dan S. Yan, “Deep learning with S-shaped rectified linear activation units,” dalam *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016. doi: 10.1609/aaai.v30i1.10287.
- [25] L. Breiman, “Random Forest,” *Mach Learn*, vol. 45, no. 1, hlm. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [26] T. Chen dan C. Guestrin, “XGBoost: A scalable tree boosting system,” dalam *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. doi: 10.1145/2939672.2939785.
- [27] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann Stat*, vol. 29, no. 5, 2001, doi: 10.1214/aos/1013203451.
- [28] C. Cortes dan V. Vapnik, “Support-vector networks,” *Mach Learn*, vol. 20, no. 3, hlm. 273–297, Sep 1995, doi: 10.1007/BF00994018.
- [29] J. Shlens, “A Tutorial on Principal Component Analysis,” arXiv:1404.1100 [cs, stat], Apr. 2014, Available: <https://arxiv.org/abs/1404.1100>.
- [30] “RobustScaler,” scikit-learn, 2024. <https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.RobustScaler.html>.
- [31] N. V. Chawla, K. W. Bowyer, L. O. Hall, dan W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, 2002, doi: 10.1613/jair.953.
- [32] C. Maklin, “Synthetic Minority Over-sampling TEchnique (SMOTE),” Medium, May 14, 2022. <https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c>.
- [33] E. N. R. Khakim, A. Hermawan, dan D. Avianto, “IMPLEMENTASI CORRELATION MATRIX PADA KLASIFIKASI DATASET WINE,” *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 1, 2023, doi: 10.26798/jiko.v7i1.771.

- [34] L. Sun, T. Liu, Y. Xie, D. Zhang, dan X. Xia, "Real-time power prediction approach for turbine using deep learning techniques," *Energy*, vol. 233, 2021, doi: 10.1016/j.energy.2021.121130.
- [35] H. Tyagi dan R. Kumar, "Attack and anomaly detection in IoT networks using supervised machine learning approaches," *Revue d'Intelligence Artificielle*, vol. 35, no. 1, 2021, doi: 10.18280/ria.350102.
- [36] T. Saito dan M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, 2015, doi: 10.1371/journal.pone.0118432.
- [37] D. Chicco dan G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, 2020, doi: 10.1186/s12864-019-6413-7.
- [38] H. He dan E. A. Garcia, "Learning from imbalanced data," *IEEE Trans Knowl Data Eng*, vol. 21, no. 9, 2009, doi: 10.1109/TKDE.2008.239.