

BAB 3

METODE PENELITIAN

Penelitian ini berfokus pada analisis penerapan algoritma ML untuk membangun model klasifikasi *multi-class* menggunakan model *ensemble learning* berbasis *stacking* untuk mendeteksi anomali pada trafik IoT. Metode yang digunakan melibatkan beberapa tahapan, mulai dari pengumpulan data, *preprocessing* data, *splitting* data, penyeimbangan data, pelatihan dan validasi model, evaluasi sampai dengan visualisasi hasil.

Tahapan awal dimulai dengan pengumpulan data. Data yang digunakan merupakan dataset publik yang berisi trafik jaringan IoT. Dataset ini memiliki label *multi-class* yang menggambarkan jenis trafik normal dan serangan. Data kemudian dikumpulkan untuk dilakukan tahap *preprocessing*. Pada tahap ini dilakukan *cleaning* data dengan menghapus nilai kosong dan menangani duplikasi data agar tidak mempengaruhi hasil analisis, selanjutnya *encoding* label dengan mengubah label kategori menjadi nilai numerik menggunakan *Label Encoding*, selanjutnya dilakukan *feature selection* untuk menyaring atribut yang paling relevan kemudian normalisasi data numerik menggunakan *Min-Max Scaling* agar semua fitur berada dalam skala yang sama.

Setelah tahap *preprocessing*, data dibagi menjadi dua bagian yaitu 80% untuk data *training* dan 20% untuk data *testing*. Pembagian data dilakukan sebelum proses *oversampling* untuk menjaga agar data *testing* tetap merepresentasikan distribusi asli. Untuk mengatasi ketidakseimbangan data dalam data training akan dilakukan teknik SMOTE (*Synthetic Minority Oversampling Technique*) yaitu dengan menghasilkan data sintesis dari titik data minoritas dan tetangga terdekat, agar jumlah data pada kelas minoritas menjadi seimbang. Teknik SMOTE hanya diterapkan pada data training untuk menghindari kebocoran data dan menjaga distribusi asli dalam proses evaluasi.

Pada tahap pelatihan model, digunakan metode *ensemble learning* berbasis *stacking*. Metode ini menggabungkan beberapa algoritma base learner yang terdiri

dari *Decision Tree* (DT), *Random Forest* (RF), *Naive Bayes* (NB), dan *Logistic Regression* (LR). Setiap *base learner* dilatih pada data *training* yang telah diseimbangkan untuk menghasilkan prediksi yang kemudian digunakan sebagai input pada model kedua (*meta learner*) yaitu *Support Vector Machine* (SVM), yang bertujuan untuk menghasilkan prediksi akhir. Seluruh proses pelatihan dan pengujian model dilakukan proses validasi menggunakan *5-Fold Stratified Cross Validation* untuk menghindari *overfitting*. Pada proses ini, data *training* dibagi menjadi lima bagian dimana masing-masing bagian digunakan secara bergantian sebagai data validasi dan empat bagian sisanya sebagai data pelatihan. Hasil evaluasi pada setiap fold kemudian dirata-rata untuk mendapatkan performa keseluruhan model yang lebih stabil dan representatif.

Model kemudian dievaluasi dengan membandingkan hasil prediksi terhadap label sebenarnya menggunakan metrik klasifikasi *multi-class*. Tahapan evaluasi dilakukan dengan pendekatan berbasis *confusion matrix* dan kurva ROC. *Confusion matrix* adalah tabel yang menggambarkan distribusi prediksi model terhadap nilai aktual. Tabel ini terdiri dari:

1. *True Positive* (TP): Data positif yang diprediksi benar.
2. *True Negative* (TN): Data negatif yang diprediksi benar.
3. *False Positive* (FP): Data negatif diprediksi sebagai positif.
4. *False Negative* (FN): Data positif diprediksi sebagai negatif.

Dalam konteks *multiclass*, *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) dihitung untuk setiap kelas menggunakan pendekatan *One-vs-Rest*, masing-masing kelas diperlakukan sebagai kelas positif dan semua kelas lainnya sebagai kelas negatif. Berdasarkan nilai-nilai tersebut, metrik yang digunakan untuk mengevaluasi model adalah:

1. *Accuracy*

- a) Untuk mengukur seberapa banyak prediksi yang benar dibandingkan dengan total prediksi yang dilakukan oleh model.
- b) Untuk klasifikasi *multi-class*, tetap menghitung total prediksi yang benar pada seluruh kelas dan dibandingkan dengan seluruh prediksi yang dilakukan.

c) Formulasi: $Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Total\ Prediksi}$

2. Precision

- a) Untuk mengukur seberapa banyak prediksi positif yang benar benar positif.
- b) Pada klasifikasi *multi-class*, *precision* dihitung untuk setiap kelas secara terpisah dan kemudian dihitung rata-ratanya.

c) Formulasi: $Precision = \frac{TP}{TP + FP}$

3. Recall

- a) Untuk mengukur seberapa banyak data positif yang berhasil dideteksi oleh model.
- b) Pada klasifikasi *multi-class*, *recall* dihitung untuk setiap kelas kemudian dihitung rata-rata keseluruhannya.

c) Formulasi: $Recall = \frac{TP}{TP + FN}$

4. F1-Score

- a) Adalah rata-rata harmonis antara *precision* dan *recall*. Dihitung untuk setiap kelas dan kemudian dihitung rata-rata keseluruhan kelas.

b) Formulasi: $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

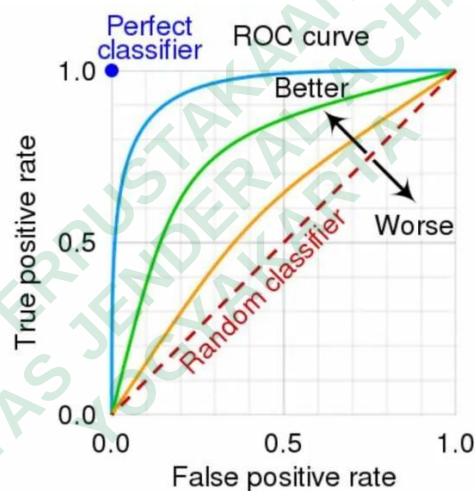
5. ROC AUC

- a) Untuk mengukur kemampuan model dalam membedakan antara kelas positif dan negatif.
- b) Dalam klasifikasi *multi-class*, ROC AUC dihitung dengan pendekatan *One-vs-Rest* (OvR), setiap kelas dianggap sebagai kelas positif sementara kelas lainnya sebagai negatif. Dihitung secara terpisah untuk setiap kelas dan hasilnya dirata-rata.

c) Formulasi: $ROC\ AUC = \frac{1}{N} \sum_{i=1}^N AUC_i$

Dimana N adalah jumlah kelas dalam klasifikasi *multi-class*, dan AUC_i adalah nilai AUC untuk kelas i .

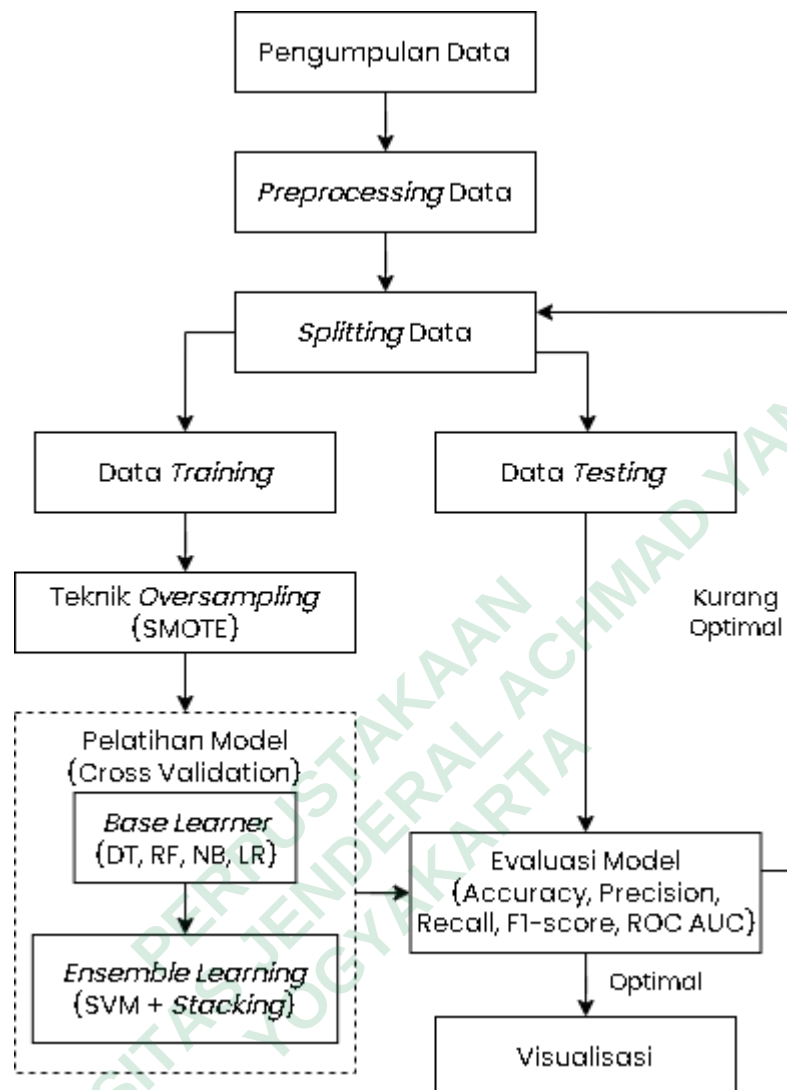
Metrik ini memberikan gambaran menyeluruh mengenai kemampuan model dalam mengklasifikasi data ke dalam masing-masing kelas dengan benar. Jika hasil evaluasi model belum optimal, maka dilakukan proses *recursive*, yaitu kembali ke tahap pembagian data dan mengulang proses pelatihan model. Namun jika hasil evaluasi sudah optimal, maka proses dilanjutkan ke tahap visualisasi menggunakan kurva ROC dan *confusion matrix*. Kurva ROC digambarkan untuk setiap kelas dengan pendekatan One-vs-Rest, untuk menilai kemampuan diskriminatif model per kelas. Sementara itu, *confusion matrix* digunakan untuk menggambarkan distribusi prediksi benar dan salah pada tiap kelas, sehingga mempermudah analisis kesalahan klasifikasi. Contoh gambaran kurva ROC ditunjukkan pada Gambar 3.1 berikut.



Gambar 3. 1 Contoh Kurva ROC

Sumber: https://spotintelligence-com.translate.goog/2024/06/17/roc-auc-curve-in-machine-learning/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=imgs

Berikut adalah gambaran lengkap mengenai tahapan penelitian yang dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Alur Penelitian

3.1 Bahan dan Alat Penelitian

Penelitian ini menggunakan data sekunder yaitu dataset ACI IoT Network Traffic 2023 (ACI-IoT-2023) yang diambil dari *Kaggle.com* (Bastian dkk., 2023). Dataset ini berisi trafik jaringan IoT dalam lingkungan rumah pintar yang mencakup lalu lintas normal dan berbagai jenis serangan siber. Dataset ini dirancang khusus untuk pembelajaran mesin. Dataset ini diproduksi oleh tim *Army Cyber Institute* (ACI) melalui eksperimen di laboratorium *Internet of Things Research Lab* (IoTRL), yang dirancang menyerupai lingkungan rumah pintar. IoTRL mengintegrasikan berbagai perangkat IoT, baik yang terhubung secara kabel

maupun nirkabel, dan dikendalikan menggunakan sistem *Home Assistant* sebagai pusat otomatisasi. Untuk mengumpulkan data, tim peneliti menggunakan network sniffers dan titik pemantauan yang disebar di seluruh jaringan laboratorium. Selama satu minggu, mereka merekam lalu lintas normal dan melakukan serangkaian serangan siber secara terjadwal seperti recon (ping sweep, OS scan), DoS (ICMP Flooding, Slowloris, UDP Flood, dll), brute force, dan spoofing (ARP spoofing). Dataset ini terdiri dari 1.231.411 entri dan 85 atribut, atribut dan penjelasannya dapat dilihat pada tabel lampiran.

Dataset ini relevan dengan penelitian yang akan dilakukan karena berfokus pada analisis penerapan algoritma ML untuk deteksi anomali pada trafik IoT. Adapun beberapa alasan yang mendukung pemilihan dataset ini yaitu:

1. Menampilkan Perangkat dan Pola Lalu Lintas yang Beragam

Dataset ini mencerminkan aktivitas jaringan IoT dengan berbagai perangkat yang terhubung, baik melalui koneksi kabel maupun nirkabel, serta beragam pola komunikasi yang terjadi dalam lingkungan rumah pintar.

2. Kombinasi Lalu Lintas Normal dan Berbahaya

Dataset ini mencakup berbagai jenis serangan dan lalu lintas normal. Keberagaman ini penting untuk melakukan deteksi anomali.

3. Format Data yang Terstruktur

Dataset ini disusun dalam format yang mudah diolah, dengan atribut yang relevan seperti waktu, alamat IP sumber dan tujuan, protokol yang digunakan, serta label yang menunjukkan jenis serangan atau lalu lintas normal. Struktur ini memungkinkan penerapan metode *ensemble learning* secara optimal.

4. Tersedia Data Berlabel untuk Klasifikasi *multi-class*

Setiap entri dalam dataset telah diberi label untuk mendukung klasifikasi. Label ini memungkinkan model untuk tidak hanya membedakan antara lalu lintas normal dan anomali saja, namun juga mengelompokkan jenis anomalnya.

Penelitian ini memanfaatkan hardware dan software yang digunakan pada proses analisis dan pengolahan data untuk memastikan keberhasilan penelitian. Perangkat keras yang digunakan mencakup komputer dengan spesifikasi yang

cukup memadai untuk menjalankan sistem operasi dan perangkat lunak pendukung yang terhubung dengan internet.

Sistem Operasi dan perangkat lunak yang digunakan dalam penelitian ini adalah:

1. Sistem Operasi: Windows 10
2. Processor: Intel Core i5-6200U CPU @2.30 Ghz
3. RAM: 4 GB
4. Storage: HDD 500 GB
5. Software: Google Colab

3.2 Jalan Penelitian

Penelitian ini terdiri dari beberapa tahapan, mulai dari pengumpulan data, persiapan data, pengujian model dan yang terakhir evaluasi performa model. Setiap tahapan dilakukan secara berurutan untuk memastikan model bekerja secara optimal dalam mendeteksi anomali pada trafik IoT.

Berikut adalah beberapa tahapan yang dilakukan:

1. Pengumpulan Data
Menggunakan data sekunder yaitu dataset ACI IoT 2023 yang tersedia pada platform *kaggle.com*.
2. *Preprocessing* Data
Meliputi pembersihan data, encoding label, pemilihan fitur dan normalisasi data numerik.
3. *Splitting* Data
Data dibagi menjadi dua bagian yaitu data *training* sebesar 80% untuk melatih model dan data *testing* sebesar 20% untuk menguji performa model.
4. Penyeimbangan Data
Menggunakan teknik SMOTE (*Synthetic Minority Over-sampling Technique*) pada data training, untuk mengatasi ketidakseimbangan kelas.
5. Pelatihan Model
Dilakukan pelatihan dan pengujian model dengan menggunakan teknik Stacking Ensemble Learning.

- a) *Base Learner*: Model dasar yang digunakan terdiri dari *Decision Tree* (DT), *Naive Bayes* (NB), *Random Forest* (RF), dan *Logistic Regression* (LR).
- b) *Meta Learner*: Hasil prediksi dari *base learner* digunakan sebagai input tambahan untuk *Support Vector Machine* (SVM) yang berperan sebagai *meta learner*.

6. Validasi Model

Menggunakan *5-Fold Cross Validation* untuk menghindari *overfitting* dan meningkatkan generalisasi model.

7. Evaluasi Model

Menggunakan Confusion Matrix. Confusion Matrix dan metrik klasifikasi *multi-class* yaitu *Accuracy*, *Precision*, *Recall*, *F1-Score* dan ROC AUC.

8. Perbaikan Model

Jika hasil evaluasi belum optimal maka dilakukan pengulangan proses (*recursive*) dari tahapan pembagian data. Jika sudah optimal maka dilanjutkan ke tahapan selanjutnya.

9. Visualisasi Hasil

Menampilkan kurva ROC untuk masing-masing kelas dan *confusion matrix* sebagai interpretasi visual hasil klasifikasi.