Progresif: Jurnal Ilmiah Komputer
https://ojs.stmik-banjarbaru.ac.id/index.php/progresif/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0877

Deteksi Serangan DDoS pada Trafik IoT Menggunakan *Random Forest* dengan Dataset CICIoT2023

DOI: http://dx.doi.org/10.35889/progresif.v22i2.3614

Muchammad Basroil Billah[1], Mohammad Idhom[2*], Hendra Maulana[3]
[1]Informatika, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia
[2]Sains Data, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia
[3]Bisnis Digital, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia
**e-mail Corresponding Author:* idhom@upnjatim.ac.id

Abstract

As the number of Internet of Things (IoT) devices continues to grow, these devices become increasingly vulnerable to Distributed Denial of Service (DDoS) attacks. However, their limited computational capacity makes it difficult to implement conventional security mechanisms. This study proposes a model for detecting DDoS attacks using Random Forest, trained using the CICIoT2023 dataset, which consists of 46 flow-based features collected from 105 real-world IoT devices. The preprocessing stage includes binary classification, normalization using StandardScaler, and handling class imbalance through a combination of 1:10 undersampling and class weighting. Evaluation on 1,154,684 test samples shows excellent performance, achieving 99.99% accuracy, 100% precision, 99.99% recall, and 99.99% F1-score. To ensure reliability, six validation checks are conducted, including overfitting analysis, cross-validation. The results confirm that the model can generalize well beyond the training data. Most attack types are detected perfectly, although application-layer attacks such as DDoS-SlowLoris remain more challenging. Overall, Random Forest proves to be an effective and relatively lightweight approach for DDoS detection in IoT environments.

Keywords: *DDoS; Random Forest; IoT; CICIoT2023; Machine Learning*

Abstrak

Pertumbuhan jumlah perangkat *IoT* menyebabkan peningkatan risiko terhadap berbagai ancaman keamanan terhadap serangan *Distributed Denial of Service* (DDoS). Namun, keterbatasan kapasitas komputasi pada perangkat IoT menyulitkan penerapan mekanisme keamanan konvensional. Penelitian ini mengusulkan model deteksi DDoS berbasis Random Forest yang dilatih menggunakan dataset CICIoT2023, yang terdiri dari 46 fitur berbasis *flow* yang dikumpulkan dari 105 perangkat IoT nyata. Tahap *preprocessing* meliputi klasifikasi biner, normalisasi menggunakan *StandardScaler*, serta penanganan ketidakseimbangan kelas melalui kombinasi *undersampling* (1:10) dan *class weighting*. Hasil evaluasi pada 1.154.684 data uji menunjukkan performa yang sangat tinggi, dengan *accuracy* sebesar 99,99%, *precision* 100%, *recall* 99,99%, dan *F1-score* 99,99%. Untuk memastikan keandalan model, dilakukan enam pengujian validasi, termasuk analisis *overfitting*, *cross-validation*. Hasil penelitian mengonfirmasi bahwa model mampu melakukan generalisasi dengan baik terhadap data di luar data pelatihan. Sebagian besar jenis serangan berhasil dideteksi secara sempurna, meskipun serangan pada lapisan aplikasi seperti DDoS-SlowLoris masih menjadi tantangan. Secara keseluruhan, Random Forest terbukti sebagai pendekatan yang efektif dan relatif ringan untuk deteksi DDoS pada lingkungan IoT

Kata kunci: *DDoS; Random Forest; IoT; CICIoT2023; Machine Learning*

1. Pendahuluan

Keamanan jaringan *Internet of Things* (IoT) menjadi persoalan yang semakin mendesak seiring dengan pertumbuhan jumlah perangkat yang terhubung ke internet. Statista memproyeksikan bahwa jumlah perangkat IoT akan mencapai 29,4 miliar unit pada tahun 2030,

meningkat hampir tiga kali lipat dari 9,7 miliar unit pada tahun 2020 [1]. Lonjakan ini terjadi di berbagai sektor, seperti *smart home*, pertanian presisi, hingga layanan Kesehatan [2]. Namun, setiap perangkat yang terhubung pada dasarnya juga menambah satu titik potensi serangan baru. Kondisi ini diperparah oleh fakta bahwa sebagian besar perangkat IoT dirancang dengan mengutamakan efisiensi biaya dan konsumsi daya rendah, sehingga aspek keamanan sering kali kurang diperhatikan [3]. Keterbatasan pada sisi *hardware* dan *software* membuat perangkat-perangkat ini tidak mampu menjalankan mekanisme proteksi yang lebih canggih [4].

Di antara berbagai ancaman siber, serangan *Distributed Denial of Service* (DDoS) termasuk yang paling meresahkan dalam konteks IoT. Cloudflare pada kuartal ketiga tahun 2024 mencatat serangan DDoS dengan volume mencapai 4,2 Tbps yang memanfaatkan ribuan perangkat IoT yang terinfeksi *malware* [5]. Di Indonesia, laporan AwanPintar.id mencatat lebih dari 133 juta serangan siber pada semester pertama tahun 2025 [6]. Dampak serangan ini tidak hanya berupa gangguan layanan, tetapi juga berpotensi mengancam keselamatan pada sistem kritis seperti perangkat medis dan *industrial control systems* [7].

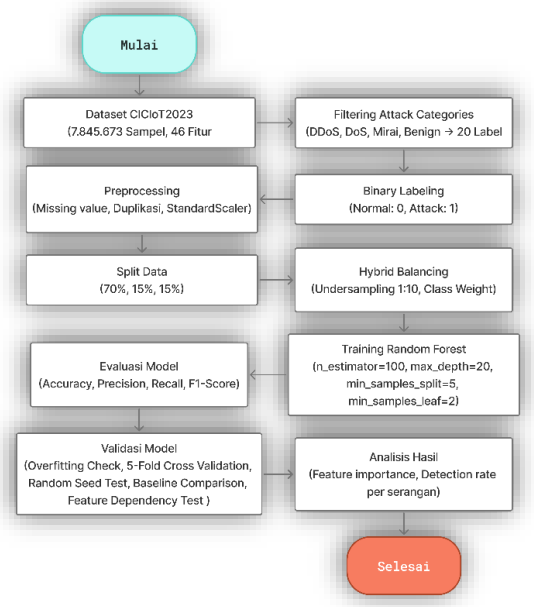
Untuk menghadapi ancaman tersebut, pendekatan deteksi serangan perlu mempertimbangkan karakteristik unik lingkungan IoT. Pendekatan *rule-based* mampu memberikan respons cepat terhadap pola serangan yang sudah dikenal, namun kurang adaptif terhadap varian serangan baru [8]. Sebaliknya, pendekatan berbasis *machine learning* mampu mengenali pola serangan yang lebih kompleks melalui algoritma seperti Random Forest, SVM, dan *Deep Neural Networks* [2]. Meskipun demikian, model *machine learning* dengan akurasi tinggi umumnya membutuhkan komputasi besar dan waktu inferensi yang lebih lama, sehingga kurang sesuai dengan keterbatasan sumber daya pada perangkat IoT [9]. Selain itu, penggunaan dataset yang kurang merepresentasikan kondisi nyata serta proses validasi yang belum menyeluruh sering kali menyebabkan hasil penelitian sulit digeneralisasi [10].

Beberapa penelitian telah mencoba mengatasi permasalahan deteksi DDoS pada jaringan IoT dengan berbagai pendekatan. Alahmadi et al. (2023) melakukan survei terhadap algoritma *machine learning* menggunakan dataset CICIDS2017 dan NSL-KDD, dan menemukan bahwa pendekatan *ensemble* mampu mencapai akurasi di atas 95%, namun belum diuji pada dataset yang merepresentasikan lingkungan IoT (*IoT-native*) [2]. Hussain et al. (2024) mengusulkan arsitektur *hybrid* dengan *response time* 5 ms, tetapi menempatkan model *machine learning* sebagai lapisan awal sehingga berpotensi meningkatkan beban komputasi pada perangkat *edge* [11]. Shakya dan Abbas (2024) membandingkan beberapa algoritma *machine learning* pada dataset CICIoT2023 dan melaporkan akurasi hingga 99,82%, namun belum menyertakan validasi seperti *overfitting* check [12]. Gavric et al. (2024) mencapai akurasi 99,9% melalui *feature engineering*, tetapi belum melakukan analisis per jenis serangan [10], sementara Nawaz et al. (2025) mengembangkan *framework lightweight* dengan akurasi 99,8%, namun validasi masih terbatas pada pendekatan *single split* dan belum diuji pada perangkat *edge* [13]. Berdasarkan tinjauan tersebut, meskipun performa model yang dilaporkan sudah tinggi, masih terdapat celah berupa kurangnya validasi komprehensif serta minimnya analisis performa berdasarkan jenis serangan, sehingga pemahaman terhadap kelemahan model dan kemampuan generalisasinya masih belum optimal.

Berdasarkan celah tersebut, penelitian ini mengusulkan model deteksi DDoS berbasis Random Forest dengan strategi *hybrid balancing* pada dataset CICIoT2023. Random Forest dipilih karena memiliki mekanisme *random feature subset selection* yang dapat mengurangi risiko *overfitting* serta memiliki waktu inferensi yang relatif cepat. Penelitian ini bertujuan untuk fokus pada tiga aspek utama, yaitu: (1) perancangan *preprocessing* yang efisien untuk dataset berskala besar, (2) penerapan validasi model secara komprehensif melalui enam pengujian independen, serta (3) analisis *detection rate* per jenis serangan untuk mengidentifikasi kelemahan model secara spesifik. Dengan pendekatan ini, diharapkan model yang dihasilkan tidak hanya memiliki akurasi tinggi, tetapi juga memiliki tingkat keandalan yang lebih baik untuk implementasi pada lingkungan IoT.

2. Metodologi

Penelitian ini mengikuti pendekatan kuantitatif berbasis eksperimen machine learning. Seluruh proses dari persiapan data hingga evaluasi dirancang untuk menghasilkan luaran yang terukur dan dapat direproduksi. Alur lengkap penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

Secara metodologis, Penelitian ini menerapkan pendekatan kuantitatif melalui eksperimen berbasis *machine learning* untuk menghasilkan hasil yang terukur sebagaimana ditunjukkan pada Gambar 1 (Alur Penelitian). Proses diawali dengan penggunaan dataset CICIoT2023 sebagai sumber data utama yang terdiri dari 7.845.673 sampel dan 46 fitur. Selanjutnya dilakukan *filtering attack categories* untuk menyederhanakan lebih dari 20 label menjadi beberapa kategori utama, kemudian dilanjutkan dengan *binary labeling* yang mengelompokkan data ke dalam dua kelas, yaitu normal (0) dan serangan (1). Setelah proses pelabelan, data selanjutnya diproses melalui tahap *preprocessing* yang mencakup penanganan *missing value*, penghapusan data duplikat, serta proses normalisasi dilakukan menggunakan *StandardScaler* untuk menyeragamkan distribusi fitur agar model dapat belajar secara optimal.

Setelah itu, dilakukan penanganan ketidakseimbangan data melalui pendekatan *hybrid balancing*, yaitu kombinasi *undersampling* dengan rasio 1:10 dan *class weighting*. Tahap berikutnya adalah pembangunan model menggunakan algoritma Random Forest dengan parameter yang telah ditentukan, kemudian dilatih untuk mempelajari pola trafik normal dan serangan. Evaluasi model dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, serta dilengkapi dengan validasi komprehensif melalui *overfitting check*, 5-*fold cross-validation*, *random seed test*, *baseline comparison*, dan *feature dependency test*. Dengan alur ini, model yang dihasilkan tidak hanya memiliki performa tinggi, tetapi juga mampu melakukan generalisasi secara baik.

2.1. Dataset

Dataset yang digunakan sebagai sumber data dalam penelitian ini adalah CICIoT2023 yang dikembangkan oleh Canadian Institute for Cybersecurity [14]. Dataset ini sendiri dibangun dari 105 perangkat IoT nyata dan memuat 33 jenis serangan dalam 7 kategori serta trafik benign, dengan setiap flow direpresentasikan oleh 46 fitur numerik berbasis flow. Total dataset terdiri dari 7.845.673 sampel dengan 47 kolom (46 fitur + 1 label).

2.2. *Filtering Attack Categories*

Dari 34 label asli pada dataset, dilakukan *filtering* untuk mempertahankan hanya 20 label yang relevan dengan fokus penelitian, yaitu 12 jenis DDoS, 4 jenis DoS, 3 varian Mirai, dan *BenignTraffic*. Label di luar keempat kategori tersebut dihapus karena berada di luar cakupan deteksi DDoS.

2.3. *Binary Labeling*

Seluruh label serangan (DDoS, DoS, Mirai) dipetakan menjadi satu kelas *Attack* dengan nilai 1, sementara *BenignTraffic* dipetakan menjadi kelas *Normal* dengan nilai 0. Pendekatan *binary classification* ini dipilih karena tujuan utama model adalah membedakan antara trafik yang aman dan trafik yang berbahaya, bukan mengklasifikasikan jenis serangan secara spesifik.

2.4. *Preprocessing*

Kualitas data berpengaruh langsung terhadap keandalan model yang dibangun [15]. Proses *preprocessing* dilakukan melalui tiga tahapan utama. Pertama, pengecekan *missing values* dan *infinity values*, di mana fitur yang memiliki nilai kosong diisi menggunakan rata-rata (*mean imputation*). Kedua, identifikasi dan penghapusan baris data duplikat. Ketiga, normalisasi seluruh fitur menggunakan *StandardScaler* dengan formulasi *Z-score* berikut:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Di mana *x* menunjukkan nilai fitur sebelum transformasi, sementara *μ* merepresentasikan rata-rata fitur pada *training set*, dan *σ* adalah standar deviasinya. Metode ini dipilih karena lebih tahan terhadap *outlier* dibandingkan *Min-Max Scaling*. Luaran dari tahap ini berupa dataset yang bersih dengan seluruh fitur memiliki memiliki nilai rata-rata *(mean)* yang mendekati 0 dan standar deviasi yang mendekati 1.

2.5. Split Data

Data Dibagi menjadi 3 subset, 70% sebagai data latih (5.491.971 sampel), 15% sebagai data validasi (1.176.851 sampel), dan 15% sebagai data uji (1.176.851 sampel). Data latih digunakan untuk membangun model, data validasi digunakan untuk memantau performa selama proses pelatihan, sedangkan data uji digunakan untuk mengevaluasi kinerja akhir model secara objektif terhadap data yang belum pernah dilihat sebelumnya.

2.6. *Hybrid Balancing*

Dataset CICIoT2023 memiliki distribusi kelas yang sangat timpang. Untuk mengatasinya, diterapkan strategi *hybrid balancing* yang hanya diterapkan pada data latih. Strategi ini terdiri dari dua komponen, yaitu: (1) *undersampling* pada kelas *Attack* dengan rasio 1:10, dan (2) penerapan *class weighting* pada Random Forest (Normal: bobot 10, *Attack*: bobot 1).

2.7. *Training* Model Random Forest

Random Forest merupakan algoritma *ensemble learning* yang membentuk sejumlah pohon keputusan secara paralel melalui teknik *Bootstrap Aggregating* (*Bagging*) [16]. Setiap pohon dilatih menggunakan subset data yang berbeda hasil dari *bootstrap sampling*. Pada setiap node, hanya sebagian fitur yang dipilih untuk proses pemisahan (√*n_features*) yang dievaluasi untuk menentukan *split* terbaik menggunakan kriteria *Gini Impurity*:

$$Gini = 1 - \Sigma {p_i}^2 \quad (2)$$

Di mana p_i adalah proporsi sampel kelas *i* pada *node*. Prediksi akhir diperoleh melalui *majority voting* dari seluruh pohon. Model dilatih dengan parameter: *n_estimators*=100, *max_depth*=20, *min_samples_split*=5, *min_samples_leaf*=2, *class_weight*={0:10, 1:1}, *n_jobs*=-1, *random_state*=42. Seluruh 46 fitur dipertahankan tanpa dilakukan seleksi fitur secara eksplisit.

2.8. Evaluasi Model

Kinerja model dinilai menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, serta *confusion matrix* pada data latih, data validasi, dan data uji. Pemilihan kombinasi metrik ini mengikuti praktik evaluasi model klasifikasi berbasis *tree* yang telah terbukti efektif [17].

2.9. Validasi Model

Untuk memastikan keandalan hasil, dilakukan enam pengujian validasi independen [18], yaitu: (1) *overfitting check*, (2) 5-*fold cross-validation* pada subsampel 100.000 data, (3) *random seed stability* dengan tiga *seed* berbeda, (4) *baseline comparison* menggunakan *Dummy Classifier*, dan (5) *feature dependency test*.

2.10. Analisis Hasil

Tahap akhir berupa analisis *feature importance* untuk memahami fitur yang paling berkontribusi terhadap keputusan model, serta analisis *detection rate* per jenis serangan untuk mengidentifikasi secara spesifik pola serangan yang masih menjadi tantangan.

3. Hasil dan Pembahasan

3.1. Dataset

Seluruh 46 fitur dipertahankan sebagai variabel input pada algoritma Random Forest tanpa seleksi fitur eksplisit, karena Random Forest sudah memiliki mekanisme *random feature subset selection* yang secara otomatis menangani fitur kurang informatif. Sebagai ilustrasi, Tabel 1 menyajikan sampel data dengan beberapa fitur representatif beserta labelnya.

Tabel 1 . Sampel Data Fitur Representatif

flow_duration	tot_fwd_pkts	flow_pkts_per_sec	syn_count	rst_count	Label
12.4508	1	1.121	1	1	Normal
0.003154	2514	0.001542	0	0	DDoS-SYN_Flood
0.001287	1287	1000	0	0	DDoS-ICMP_Flood
185.2001	20	0.121	0	0	DDoS-SlowLoris

3.2. Hasil *Filtering Attack Categories*

Proses *filtering* berhasil menyaring 20 label yang relevan dari 34 label asli, yang terdiri dari 12 jenis DDoS, 4 jenis DoS, 3 varian Mirai, dan *BenignTraffic*. Sebanyak 14 label lainnya termasuk *Recon*, *Spoofing*, *WebBased*, dan *BruteForce* dihapus karena berada di luar cakupan deteksi DDoS. Setelah proses *filtering*, total sampel yang dipertahankan tetap sebanyak 7.845.673 dari keempat kategori utama tersebut.

3.3. Hasil *Binary Labeling*

Pemetaan *binary labeling* menghasilkan dua kelas, yaitu *Normal* (0) untuk *BenignTraffic* dan *Attack* (1) untuk seluruh label serangan. Contoh hasil pemetaan ditunjukkan pada Tabel 2. Distribusi kelas menunjukkan dominasi yang sangat tinggi pada kelas *Attack*, yang berpotensi menyebabkan bias model terhadap kelas mayoritas.

Tabel 2. Pemetaan *Binary Labeling*

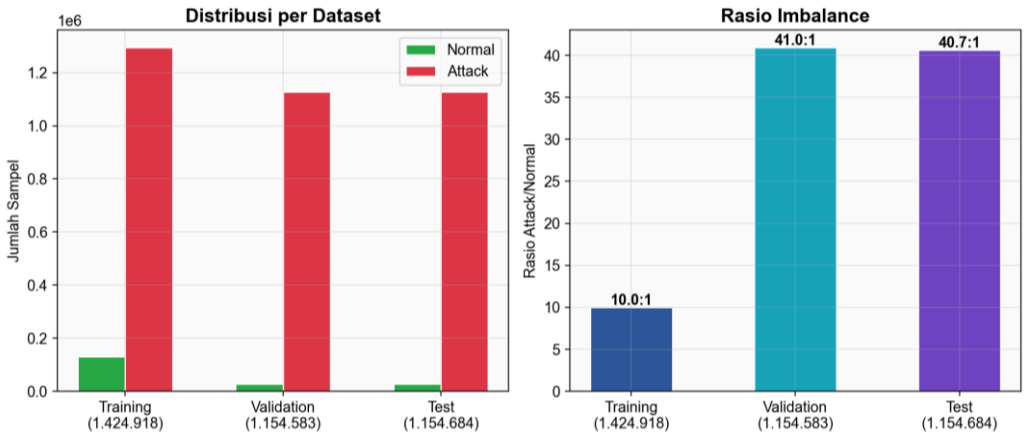
Label Original	Target	Keterangan
BenignTraffic	0	Trafik normal
DDoS-ICMP_Flood	1	Serangan DDoS
DoS-TCP_Flood	1	Serangan DoS
Mirai-greeth_flood	1	Serangan Mirai

3.4. Hasil *Preprocessing*

Pengecekan *missing values* menemukan beberapa fitur dengan nilai kosong dan *infinity* yang kemudian ditangani melalui imputasi *mean*. Proses *deduplication* mengidentifikasi dan menghapus baris data duplikat. Normalisasi menggunakan *StandardScaler* menghasilkan seluruh fitur dengan memiliki nilai *mean* yang mendekati 0 dan standar deviasi sekitar 1, sehingga tidak ada fitur yang mendominasi proses pembelajaran model.

3.5. Hasil Split Data

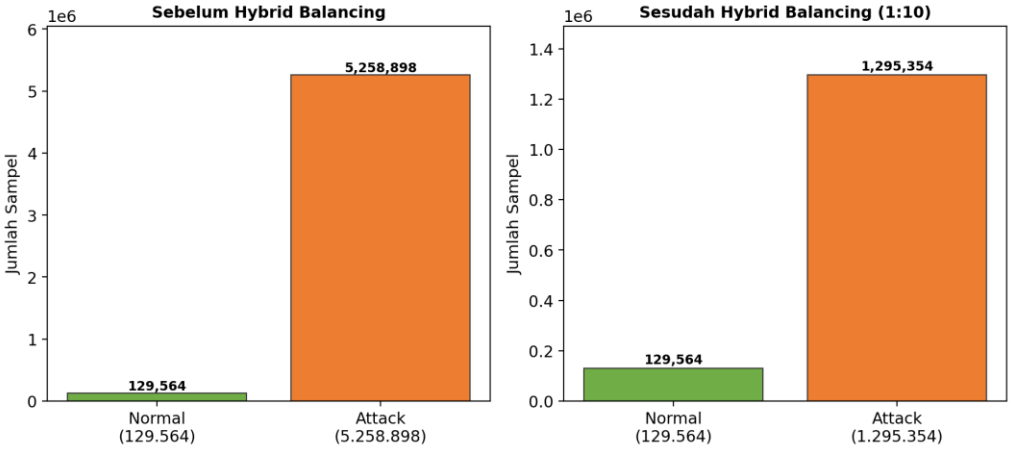
Pembagian dataset menghasilkan tiga subset independen, yaitu data latih sebanyak 5.491.971 sampel (70%), data validasi sebanyak 1.176.851 sampel (15%), dan data uji sebanyak 1.176.851 sampel (15%). Komposisi distribusi kelas pada masing-masing subset serta rasio *imbalance* ditunjukkan pada Gambar 2.



Gambar 2. Komposisi Dataset CICIoT2023

3.6. Hasil *Hybrid Balancing*

Penerapan *hybrid balancing* pada data latih berhasil menurunkan jumlah sampel dari 5.388.462 menjadi 1.424.918 dengan rasio *Normal:Attack* sebesar 1:10. Pendekatan ini menjaga keberagaman pola serangan sekaligus meningkatkan representasi kelas minoritas. Distribusi kelas sebelum dan sesudah *balancing* ditunjukkan pada Gambar 3.



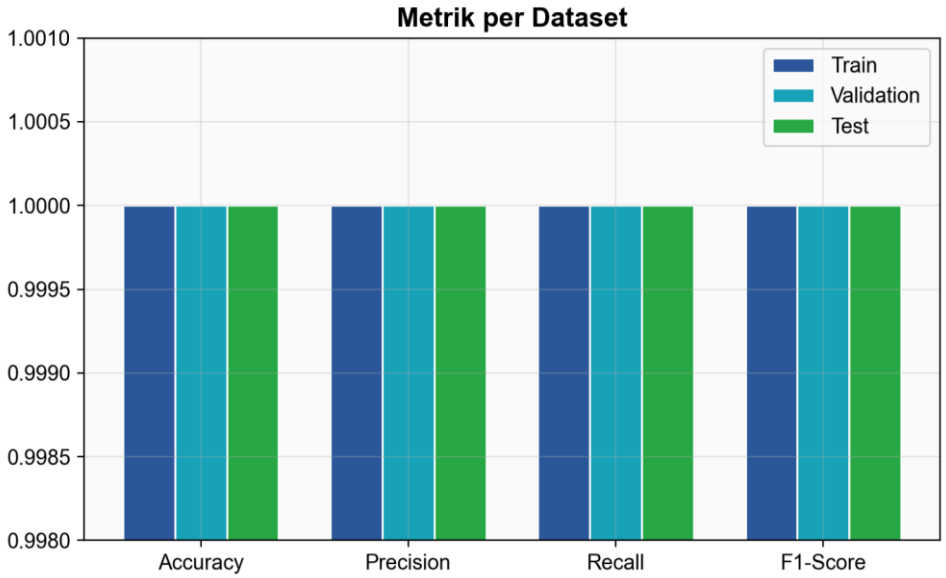
Gambar 3. Distribusi Kelas Sebelum dan Sesudah *Hybrid Balancing*

3.7. Hasil *Training* Model Random Forest

Hasil pelatihan model *Random Forest* menunjukkan performa yang sangat tinggi pada berbagai metrik evaluasi. Model mampu mencapai nilai akurasi, presisi, *recall*, dan *F1-Score* yang hampir sempurna pada *training*, *validation*, maupun *test set*. Konsistensi ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik dalam membedakan *traffic* normal dan serangan. Dan juga tidak ditemukan indikasi *overfitting* yang signifikan karena selisih performa antar *dataset* relatif sangat kecil. Hal ini menunjukkan bahwa model tidak hanya menghafal pola pada data latih, tetapi juga mampu mengenali pola serangan pada data yang belum pernah dilihat sebelumnya

3.8. Hasil Evaluasi Model (Pelatihan & Validasi)

Model *Random Forest* dilatih menggunakan data latih (70%) yang telah melalui proses *balancing*, kemudian divalidasi menggunakan data validasi (15%) dengan distribusi asli. Hasil evaluasi pada kedua *dataset* menunjukkan nilai metrik yang sangat konsisten, sebagaimana ditampilkan pada Gambar 4 dan Tabel 3. Hal ini mengindikasikan bahwa model mampu mempertahankan performa pada data baru tanpa mengalami penurunan kinerja yang berarti.



Gambar 4 *Perbandingan Metrik pada Training, Validation, dan Test Set*

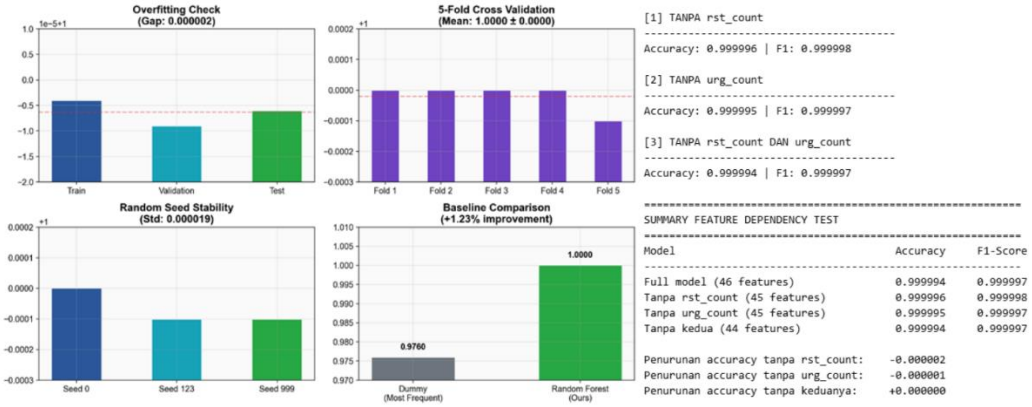
Konsistensi performa antara data latih dan data validasi mengindikasikan bahwa model tidak mengalami *overfitting*. Hal ini menunjukkan bahwa model tidak hanya menghafal pola pada data latih, tetapi juga mampu menangkap pola umum yang dapat diterapkan pada data baru. Stabilitas ini juga dipengaruhi oleh karakteristik Random Forest yang menggunakan teknik *bagging* dan pemilihan subset fitur secara acak, sehingga mampu mengurangi variansi model dan meningkatkan kemampuan generalisasi.

Tabel 3. Performa Model Random Forest pada Pelatihan & Validasi

Dataset	Accuracy	Precision	Recall	F1-Score
Training (70%)	0.999996	1.000000	0.999995	0.999998
Validation (15%)	0.999991	1.000000	0.999991	0.999996

3.9. Hasil Validasi Model

Untuk memastikan bahwa performa tinggi bukan disebabkan oleh bias data atau konfigurasi eksperimen, dilakukan enam pengujian validasi independen. Hasil lengkap disajikan pada Tabel 4 dan Gambar 5.



Gambar 5 *Hasil Validasi Model*

Gap antara performa *training* dan *test* sebesar 0,000002 menunjukkan bahwa model tidak mengalami *overfitting*. *Cross-validation* menghasilkan *F1-score* yang identik di seluruh *fold*,

yang menandakan stabilitas model. *Feature dependency test* menunjukkan bahwa penghapusan fitur dominan tidak menurunkan performa model secara signifikan.

Tabel 4. Ringkasan Hasil Validasi Model

Pengujian	Hasil	Kesimpulan
Overfitting Check	Gap: 0,000002	Tidak ada overfitting
5-Fold Cross Validation	F1: 1,0000 ± 0,0000	Model stabil
Random Seed Test	Std Dev: 0,000019	Stabil terhadap inisialisasi
Baseline Comparison	Baseline: 97,60%	Model lebih baik +1,23%
Feature Dependency	Akurasi tetap 1,0000	Tidak bergantung pada satu fitur

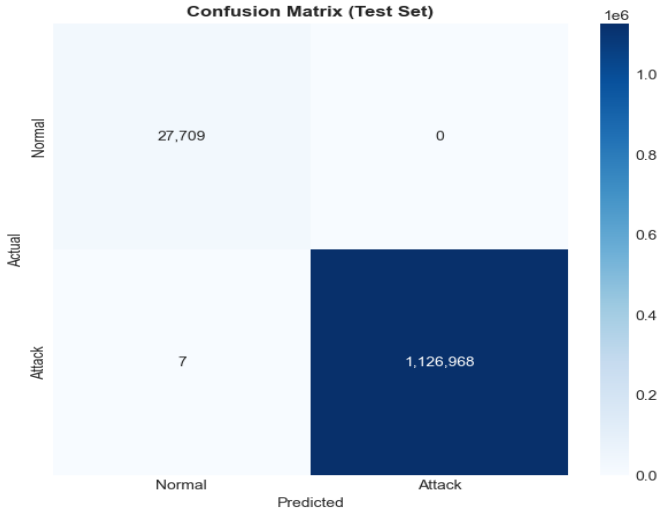
3.10. Hasil Pengujian

Model diuji menggunakan data uji (15%) yang tidak pernah terlibat dalam proses pelatihan maupun validasi. Performa pada *test set* tetap konsisten, sebagaimana ditunjukkan pada Tabel 5.

Tabel 5. Performa Model Random Forest pada Data Uji

Dataset	Accuracy	Precision	Recall	F1-Score
Test (15%)	0.999994	1.000000	0.999994	0.999997

Dari 1.154.684 sampel pada *test set*, *confusion matrix* (Gambar 6) menunjukkan bahwa model berhasil mengklasifikasikan seluruh 27.709 trafik normal tanpa kesalahan (*zero false positive*), serta mendeteksi 1.126.968 trafik serangan dengan hanya 7 kesalahan klasifikasi. Hasil ini menunjukkan bahwa model memiliki sensitivitas dan spesifisitas yang sangat tinggi, yang penting dalam konteks keamanan jaringan untuk meminimalkan kesalahan deteksi, baik berupa *false positive* maupun *false negative*.



Gambar 6 *Confusion Matrix pada Test Set*

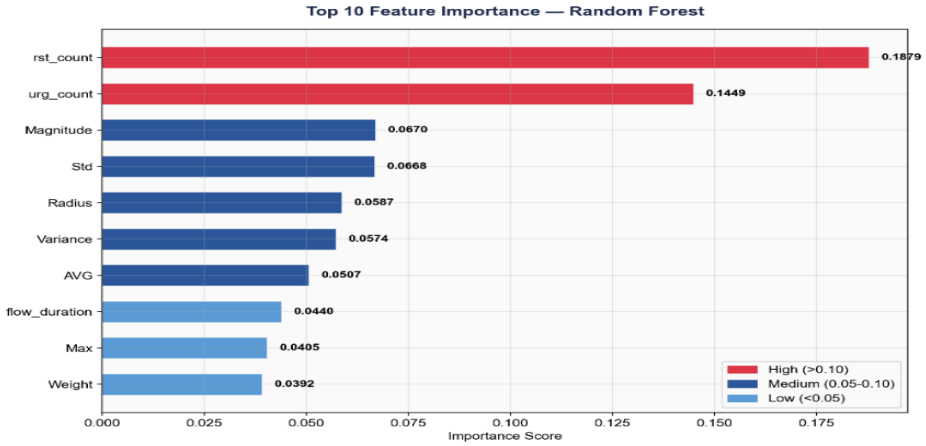
Sampel yang gagal dikenali memiliki pola statistik yang sangat mirip trafik normal, sehingga sulit dibedakan oleh fitur berbasis *flow*. Temuan ini mengarahkan pengembangan selanjutnya pada penambahan fitur yang lebih sensitif terhadap perilaku *application-layer attack*.

Tabel 6. *Karakteristik Sampel Berhasil vs Gagal Dikenali*

Karakteristik	Berhasil Dikenali	Gagal Dikenali
flow_duration	< 1 detik	> 100 detik
flow_pkts_per_sec	> 10.000	< 5
Pola paket	Searah, volume tinggi	Bidirectional, mirip normal
Jumlah sampel	1.126.968	7

3.11. Hasil Analisis: *Feature Importance*

Analisis *feature importance* (Gambar 8) menunjukkan bahwa fitur *rst_count* dan *urg_count* memiliki kontribusi paling tinggi dengan total *importance* sebesar 33,28%. Kedua fitur ini berkaitan dengan perilaku terminasi koneksi TCP, yang secara signifikan berbeda antara trafik normal dan trafik serangan. Pada trafik normal, koneksi cenderung ditutup secara teratur sehingga nilai fitur lebih tinggi, sedangkan pada trafik serangan, koneksi sering kali tidak diselesaikan dengan benar. Perbedaan pola ini menjadikan kedua fitur tersebut sangat efektif dalam membedakan karakteristik trafik, sehingga memberikan kontribusi signifikan dalam proses pengambilan keputusan pada model Random Forest.

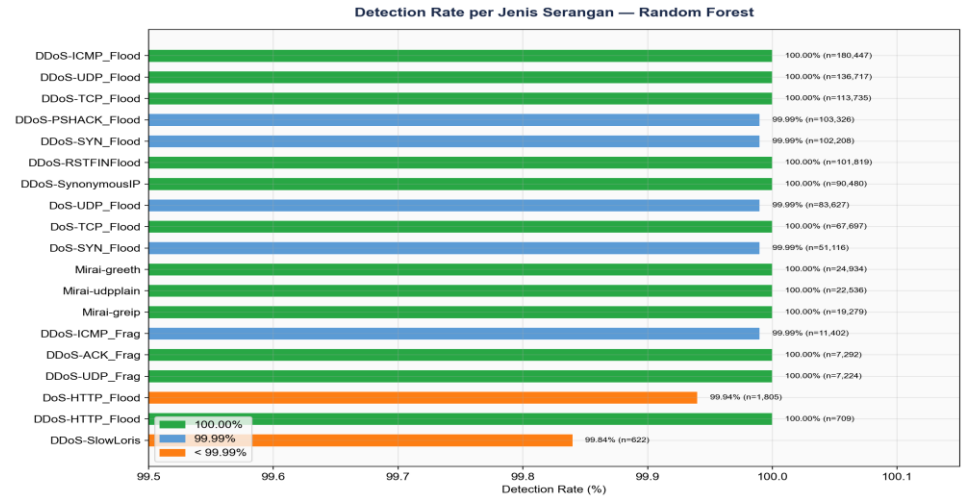


***Gambar 7** Feature Importance*

Selain itu, fitur-fitur lain dengan tingkat kepentingan menengah menunjukkan bahwa model tidak hanya mengandalkan satu indikator, tetapi memanfaatkan kombinasi berbagai karakteristik trafik. Hal ini memperkuat kemampuan model dalam mendeteksi berbagai jenis serangan dengan pola yang berbeda-beda.

3.12. Hasil Analisis: Deteksi Per Jenis Serangan

Analisis *detection rate* per jenis serangan menunjukkan bahwa 11 dari 19 jenis serangan dapat dideteksi dengan akurasi sempurna (100%). Serangan volumetrik seperti *ICMP Flood*, *UDP Flood*, dan *TCP Flood* memiliki pola trafik yang sangat mencolok, seperti lonjakan jumlah paket dan durasi koneksi yang tidak wajar, sehingga relatif mudah dikenali oleh model.



***Gambar 8** Detection Rate Per Jenis Serangan*

Sebaliknya, serangan seperti DDoS-SlowLoris (99,84%) dan DoS-HTTP_Flood (99,94%) menunjukkan tingkat deteksi yang sedikit lebih rendah. Kedua jenis serangan ini termasuk dalam kategori *application-layer attack* yang menggunakan laju pengiriman rendah serta pola yang menyerupai trafik normal. Hal ini menyebabkan fitur berbasis *flow* kurang mampu membedakan secara jelas antara trafik normal dan serangan, sehingga menjadi tantangan bagi model berbasis statistik jaringan.

Tabel 7. *Ringkasan Detection Rate Per Jenis Serangan*

Jenis Serangan	Jumlah	Terdeteksi	Akurasi
DDoS-ICMP_Flood	180.447	180.447	100,00%
DDoS-UDP_Flood	136.717	136.717	100,00%
DDoS-TCP_Flood	113.735	113.735	100,00%
DDoS-SYN_Flood	102.208	102.207	99,99%
DoS-HTTP_Flood	1.805	1.804	99,94%
DDoS-SlowLoris	622	621	99,84%
13 jenis lainnya	—	—	99,99-100%

3.13. Pembahasan

Temuan penelitian ini memberikan kontribusi yang memperkuat dan melengkapi hasil-hasil riset terdahulu dalam bidang deteksi DDoS pada jaringan IoT. Performa Random Forest yang dicapai (*accuracy* 99,99%, *precision* 100%) sejalan dengan temuan Gavric et al. [10] yang juga melaporkan akurasi 99,9% menggunakan Random Forest pada dataset CICIoT2023, serta konsisten dengan hasil Shakya dan Abbas [12] yang menemukan algoritma *ensemble* sebagai pendekatan paling efektif di antara enam algoritma yang diuji. Temuan ini semakin menegaskan posisi *Random Forest* merupakan salah satu algoritma yang sangat andal untuk tugas klasifikasi trafik jaringan IoT, sekaligus mengonfirmasi observasi dari survei Alahmadi et al. [2] bahwa pendekatan *ensemble* secara konsisten melampaui akurasi 95% pada deteksi DDoS.

Namun, kontribusi yang lebih substansial dari penelitian ini terletak pada aspek validasi. Penelitian-penelitian terdahulu, termasuk Shakya dan Abbas [12] serta Nawaz et al. [13], umumnya mengevaluasi model hanya melalui *single split* tanpa menguji kemungkinan *overfitting* atau *cross validation* secara eksplisit. Dengan melakukan lima pengujian validasi independen, penelitian ini menambahkan lapisan kepercayaan yang selama ini belum diberikan oleh studi-studi sebelumnya. Hasil *overfitting check* (gap 0,000002) dan c*ross-validation yang menghasilkan F1-score konsisten di seluruh fold memberikan* bukti empiris bahwa performa tinggi pada dataset CICIoT2023 mencerminkan kemampuan generalisasi model, bukan sekadar bukan sekadar akibat bias atau karakteristik spesifik dataset. Pendekatan validasi semacam ini diharapkan dapat menjadi standar yang diadopsi oleh penelitian-penelitian sejenis di masa mendatang, mengingat urgensi validasi yang komprehensif untuk membangun keandalan model klasifikasi telah disoroti dalam berbagai konteks [18].

Analisis per jenis serangan juga menghasilkan temuan yang melengkapi literatur yang ada. Studi Hussain et al. [11] menyebutkan bahwa arsitektur *hybrid* mampu menangkap serangan dengan *response time* rendah, namun tidak merinci jenis serangan mana yang masih menjadi tantangan. Penelitian ini mengidentifikasi secara spesifik bahwa serangan *application-layer*, khususnya DDoS-SlowLoris dan DoS-HTTP_Flood, masih menjadi titik lemah pendekatan berbasis fitur *flow*. Temuan ini memperkaya pemahaman komunitas riset mengenai batasan pendekatan *flow-based* dan dapat menjadi landasan bagi pengembangan fitur tambahan yang lebih sensitif terhadap karakteristik serangan pada lapisan aplikasi.

Strategi *hybrid balancing* yang diterapkan juga menunjukkan implikasi praktis yang penting. Berbeda dengan penggunaan SMOTE yang dominan pada penelitian terdahulu [2][13], kombinasi *undersampling* dengan rasio 1:10 dan *class weighting* berhasil menghasilkan *zero false positive*, yang merupakan capaian penting dalam konteks *security gateway*, di mana kesalahan dalam memblokir trafik normal (*false alarm*) dapat mengganggu operasional perangkat IoT. Temuan ini memberikan alternatif yang lebih tepat dalam menangani *class imbalance* pada dataset keamanan jaringan berskala besar.

Secara keseluruhan, hasil penelitian ini tidak hanya mengonfirmasi efektivitas Random Forest dalam deteksi DDoS pada lingkungan IoT, tetapi juga menunjukkan bahwa pendekatan yang menggabungkan *preprocessing* yang tepat, *balancing* yang terkontrol, serta validasi yang

komprehensif mampu menghasilkan model yang tidak hanya memiliki tingkat akurasi tinggi, tetapi juga andal untuk implementasi nyata.

4. Simpulan

Penelitian ini menunjukkan bahwa Random Forest, dengan strategi *hybrid balancing* dan validasi komprehensif, mampu mendeteksi serangan DDoS pada trafik IoT dengan *accuracy* 99,99%, *precision* 100%, dan *recall* 99,99% pada *test set* berjumlah 1.154.684 sampel. Enam pengujian validasi independen mengonfirmasi bahwa performa ini mencerminkan kemampuan generalisasi model yang sesungguhnya. Dari 19 jenis serangan, 11 di antaranya terdeteksi secara sempurna, sementara serangan *application-layer* seperti DDoS-SlowLoris (99,84%) dan DoS-HTTP_Flood (99,94%) masih menyisakan ruang untuk perbaikan.

Penelitian selanjutnya dapat diarahkan pada pengujian model menggunakan dataset dari sumber yang berbeda, pengembangan fitur tambahan untuk meningkatkan deteksi serangan *application-layer*, serta evaluasi proses inferensi pada perangkat *edge* dalam skenario *real-time*.

Daftar Referensi

[1] Petroc Taylor, “Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2034,” statista. Accessed: Dec. 01, 2025. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] A. A. Alahmadi *et al.*, “DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions,” *Electronics (Switzerland)*, vol. 12, no. 14, pp. 1–24, Jul. 2023, doi: 10.3390/electronics12143103.

[3] N. M. Karie, N. M. Sahri, W. Yang, C. Valli, and V. R. Kebande, “A Review of Security Standards and Frameworks for IoT-Based Smart Environments,” *IEEE Access*, vol. 9, pp. 121975–121995, 2021, doi: 10.1109/ACCESS.2021.3109886.

[4] M. Aziz Al Kabir, W. Elmedany, and M. S. Sharif, “Securing IoT Devices Against Emerging Security Threats: Challenges and Mitigation Techniques,” *Journal of Cyber Security Technology*, vol. 7, no. 4, pp. 199–223, 2023, doi: 10.1080/23742917.2023.2228053.

[5] J. P. Omer Yoachimik, “4.2 Tbps of bad packets and a whole lot more: Cloudflare’s Q3 DDoS report,” cloudflare. Accessed: Dec. 01, 2025. [Online]. Available: https://blog.cloudflare.com/ddos-threat-report-for-2024-q3/

[6] awanpintar.id, “Laporan Ancaman Siber Indonesia Semester 1 Tahun 2025,” AwanPintar.id. Accessed: Nov. 30, 2025. [Online]. Available: https://www.awanpintar.id/publikasi/

[7] C. Singh and A. K. Jain, “A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network,” *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 8, pp. 1–17, Jun. 2024, doi: 10.1016/j.prime.2024.100543.

[8] S. H. Lee, Y. L. Shiue, C. H. Cheng, Y. H. Li, and Y. F. Huang, “Detection and Prevention of DDoS Attacks on the IoT,” *Applied Sciences (Switzerland)*, vol. 12, no. 23, pp. 1–18, Dec. 2022, doi: 10.3390/app122312407.

[9] N. Tekin, A. Acar, A. Aris, A. S. Uluagac, and V. C. Gungor, “Energy consumption of on-device machine learning models for IoT intrusion detection,” *Internet of Things (Netherlands)*, vol. 21, pp. 1–13, Apr. 2023, doi: 10.1016/j.iot.2022.100670.

[10] N. Gavric, G. Prasad Bhandari, and A. Shalaginov, “Towards Resource-Efficient DDoS Detection in IoT: Leveraging Feature Engineering of System and Network Usage Metrics,” *Journal of Network and Systems Management*, vol. 32, no. 4, pp. 1–21, Oct. 2024, doi: 10.1007/s10922-024-09848-2.

[11] A. Hussain, E. Marin Tordera, X. Masip-Bruin, and H. C. Leligou, “Rule-Based With Machine Learning IDS for DDoS Attack Detection in Cyber-Physical Production Systems (CPPS),” *IEEE Access*, vol. 12, pp. 114894–114911, 2024, doi: 10.1109/ACCESS.2024.3445261.

[12] S. Shakya and R. Abbas, “A Comparative Analysis of Machine Learning Models for DDoS Detection in IoT Networks,” *arXiv preprint*, Nov. 2024, [Online]. Available: http://arxiv.org/abs/2411.05890

[13] M. Nawaz, S. Tahira, D. Shah, S. Ali, and M. Tahir, “Lightweight machine learning framework for efficient DDoS attack detection in IoT networks,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–24, Dec. 2025, doi: 10.1038/s41598-025-10092-0.

[14] N. Thereza and K. Ramli, “Development of Intrusion Detection Models for IoT Networks Utilizing CICIoT2023 Dataset,” in *IEEE*, 2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), 2023, pp. 66–72.
[15] B. Nugroho, H. Maulana, and A. Yuniarti, “Performance of Contrast Adjustment Techniques on The Face Recognition Method with Test Data Under Varying Lighting Conditions,” *Network Security and Information System (IJCONSIST)*, vol. 6, no. 2, pp. 66–72.
[16] M. Mahendra Alvanof, Bustami, and R. Kesuma Dinata, “Penerapan Algoritma Random Forest dalam Deteksi dan Klasifikasi Ransomware,” *Jurnal Elektronika dan Teknologi Informasi*, vol. 5, no. 2, pp. 23–31, 2024.
[17] M. Idhom, A. Fauzi, A. Muhaimin, and W. Caesarendra, “Evaluation of CART and XGBoost Methods on Customer Loan Risk Prediction Based on Consumer Behavior,” *TEM Journal*, vol. 14, no. 3, pp. 2624–2630, Jan. 2025, doi: 10.18421/TEM143-64.
[18] A. R. Dianto, F. T. Anggraeny, and H. Maulana, “Analisis Efektifitas Algoritma Mobilenetv3-Large Dan Efficientnet-B0 Untuk Klasifikasi Citra Penyakit Daun Jeruk,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3, pp. 697–705, Jul. 2025, doi: 10.23960/jitet.v13i3.6956.