JURNAL FASILKOM
Volume 16 No. 2 | Agustus 2026: 352-361

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Ablasi Kelompok Fitur *Multi-View* pada *Random Forest* untuk Deteksi Intrusi *Industrial Internet of Things (IIoT)*

Januar Al Amien[1]*, Bayu Anugrah[2], Fauzan Azim[3], Reny Medikawati Taufiq[4], Syahril[5]
[1,2,3,4]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau
[5]Sistem Informasi, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau
[1]januaralamien@umri.ac.id*, [2]bayuanugerahputra@umri.ac.id, [3]fauzanazim17@umri.ac.id,
[4]renymedikawati@umri.ac.id, [5]syahril@umri.ac.id

Abstract

Internet of Things (IIoT) systems generate heterogeneous multisource telemetry data, including network traffic, host resource usage, and security logs. Intrusion detection studies commonly combine all available feature sources based on the assumption that incorporating more sources (multi-view) will always improve detection performance. This study examines this assumption using the X-IIoTID dataset through two experiments. First, feature selection based on Random Forest Gini importance was evaluated using five feature sizes (K = 10, 20, 30, 45, and 61), with cross-algorithm robustness assessed using Decision Tree, Logistic Regression, and K-Nearest Neighbors. Second, a systematic ablation study was conducted on seven combinations of three feature groups: Network (N), Host (H), and Log (L), with Timestamp excluded from the Network group to ensure consistent feature treatment. Using 299,999 samples, comprising 239,999 training and 60,000 test samples across 19 attack classes and a normal class, the results show that multiclass performance increased with the number of features, achieving an F1-macro of 0.876 at K = 10 and 0.912 at K = 61. The ablation study showed that the Full MultiView (N+H+L) achieved the best performance (F1-macro = 0.912), followed by N+H (0.905) and N+L (0.879). The Log group alone yielded low performance (0.098) but provided additional value when combined with Network features. These findings demonstrate that the effectiveness of multi-view intrusion detection depends on feature-source combinations rather than merely the number of sources, highlighting the importance of feature-group ablation in designing IIoT intrusion detection systems.

Keywords: feature ablation, random forest, industrial internet of things, intrusion detection, multi-view features

Abstrak

Sistem *Industrial Internet of Things* (IIoT) menghasilkan data telemetri multisumber yang heterogen, meliputi lalu lintas jaringan, penggunaan sumber daya *host*, dan log keamanan. Penelitian deteksi intrusi umumnya menggabungkan seluruh sumber fitur dengan asumsi bahwa semakin banyak sumber (*multi-view*) akan selalu meningkatkan performa. Penelitian ini menguji asumsi tersebut menggunakan dataset X-IIoTID melalui dua eksperimen. Pertama, evaluasi seleksi fitur berbasis *Random Forest Gini importance* pada lima jumlah fitur (K = 10, 20, 30, 45, dan 61) serta pengujian lintas algoritma menggunakan *Decision Tree*, *Logistic Regression*, dan *K-Nearest Neighbors*. Kedua, studi ablasi terhadap tujuh kombinasi tiga kelompok fitur, yaitu *Network* (N), *Host* (H), dan *Log* (L), dengan mengecualikan *Timestamp* dari kelompok *Network*. Menggunakan 299.999 data yang terdiri atas 239.999 data latih dan 60.000 data uji, mencakup 19 kelas serangan dan kelas normal, hasil menunjukkan bahwa performa multikelas meningkat seiring penambahan fitur, dengan F1-macro 0,876 pada K = 10 dan 0,912 pada K = 61. Studi ablasi menunjukkan *Full MultiView* (N+H+L) memberikan performa terbaik (F1-macro 0,912), diikuti N+H (0,905) dan N+L (0,879). *Log* secara mandiri berkinerja rendah (0,098), tetapi memberikan kontribusi ketika dikombinasikan dengan *Network*. Temuan menunjukkan bahwa efektivitas *multi-view* bergantung pada kombinasi sumber fitur, bukan sekadar jumlah sumber, sehingga studi ablasi penting dilakukan dalam perancangan sistem deteksi intrusi IIoT.

Kata kunci: ablasi fitur, *Random Forest*, *Industrial Internet of Things*, deteksi intrusi, fitur *multi-view*

1. Pendahuluan

Transformasi *Industry 4.0* telah mendorong konvergensi antara teknologi operasional (*Operational Technology*/OT) dan teknologi informasi (*Information Technology*/IT) melalui paradigma *Industrial Internet of Things* (IIoT), yang menghubungkan sensor [1], aktuator, dan pengendali industri ke jaringan komunikasi modern seperti *MQTT*, *CoAP*, dan *WebSocket* [2]. Konektivitas ini meningkatkan efisiensi dan otomatisasi proses industri, tetapi pada saat yang sama memperluas permukaan serangan (*attack surface*) sistem industri terhadap ancaman siber [3,4]. Berbagai tinjauan pustaka menunjukkan bahwa serangan terhadap sistem IIoT terus meningkat dalam jumlah dan variasi, mulai dari pemindaian (*scanning*), *brute-force*, *denial of service*, hingga serangan lanjutan seperti *command-and-control* (C&C) dan *ransomware* pada infrastruktur kritis [2,3].

Salah satu karakteristik penting sistem IIoT adalah sifat datanya yang multi-sumber (*multi-view*): aktivitas serangan dapat tercermin secara bersamaan pada lalu lintas jaringan (*network traffic*), penggunaan sumber daya perangkat (*host resource*, seperti CPU dan memori), dan catatan log keamanan (*security log*) dari perangkat maupun sistem deteksi intrusi komersial seperti *OSSEC* dan *Zeek/Bro*. Dataset X-IIoTID yang

P-ISSN : 2089-3353
E-ISSN : 2808-9162

diperkenalkan oleh Al-Hawawreh dkk. secara eksplisit dirancang untuk merepresentasikan karakteristik *multi-view* tersebut, dengan fitur yang diekstraksi dari empat sumber berbeda: lalu lintas jaringan, sumber daya *host*, log aplikasi, dan log/peringatan sistem deteksi intrusi. Dataset ini mencakup 19 kelas serangan granular beserta kelas normal, dan pada studi aslinya dievaluasi menggunakan tujuh algoritma *machine learning* dan *deep learning* tanpa melalui tahap seleksi maupun ablasi fitur secara eksplisit per sumber data [2].

Penelitian-penelitian deteksi intrusi berbasis *multi-view* yang lebih baru pada umumnya berasumsi bahwa penggabungan (*fusion*) semakin banyak sumber fitur akan selalu meningkatkan akurasi deteksi. Sebagai contoh, [6] mengusulkan kerangka kerja *federated learning multi-view* yang menggabungkan fitur *auto-encoder* dari berbagai sudut pandang untuk deteksi intrusi, [7] mengusulkan fusi fitur mendalam berbasis *transfer learning* untuk klasifikasi intrusi multikelas, dan [8] mengusulkan fusi *multi-domain* dengan *cross-attention* untuk deteksi intrusi *few-shot*; ketiganya melaporkan bahwa kombinasi seluruh pandangan (*views*) umumnya mengungguli model pandangan tunggal (*single-view*) pada tugas serupa. Pada dataset X-IIoTID secara spesifik, [9] baru-baru ini mengusulkan kerangka kerja *federated learning* berbasis CNN-BiLSTM yang turut memanfaatkan fitur multi-sumber, namun belum mengukur kontribusi tiap kelompok sumber fitur secara terpisah. Sebagian besar studi tersebut tidak secara eksplisit menelaah kontribusi tiap kelompok sumber fitur terhadap tugas klasifikasi granular multikelas pada dataset IIoT yang memiliki ketidakseimbangan kelas ekstrem, sebagaimana dijumpai pada X-IIoTID. Tinjauan taksonomi terbaru terhadap dataset IIoT/ICS, [10] turut menyoroti bahwa ketidakseimbangan kelas dan keterbatasan analisis per sumber fitur masih menjadi celah metodologis pada evaluasi dataset-dataset tersebut.

Di sisi lain, dimensi fitur yang besar pada dataset IIoT juga menimbulkan tantangan komputasi. Penelitian [11] menunjukkan bahwa seleksi fitur berbasis *Random Forest importance* dapat mengurangi jumlah fitur secara signifikan pada dataset IoT sejenis (IoTID20) tanpa mengorbankan akurasi klasifikasi, sejalan dengan temuan [12], yang mengombinasikan *Information Gain* dan *Random Forest importance* (*IGRF-RFE*) pada dataset UNSW-NB15, serta [13], yang mengusulkan seleksi fitur berbasis optimasi merpati (*PIO*) untuk deteksi intrusi IoT berbasis *ensemble learning*. *Random Forest* sendiri dipilih secara luas sebagai basis seleksi fitur karena sifatnya yang tahan terhadap *overfitting* dan mampu menghasilkan skor kepentingan fitur (*feature importance*) berbasis pengurangan impuritas Gini secara langsung dari proses pelatihan model [14]. Selain isu dimensi fitur, dataset serangan siber pada umumnya, termasuk X-IIoTID, memiliki ketidakseimbangan kelas yang ekstrem antara kelas normal/serangan mayoritas dan kelas serangan langka [15]. Berbagai penelitian menangani permasalahan ini melalui teknik *oversampling* seperti *Synthetic Minority Oversampling Technique* (*SMOTE*) [16], baik pada model *tree-boosting* [17], maupun deteksi intrusi IoT berbasis *deep learning* dengan penanganan ketidakseimbangan kelas [18]. Sebagai alternatif terhadap *oversampling* data, sebagian studi menangani ketidakseimbangan kelas melalui pembobotan kelas (*class weighting*) langsung pada proses pelatihan model, yang menjadi pendekatan yang digunakan pada penelitian ini.

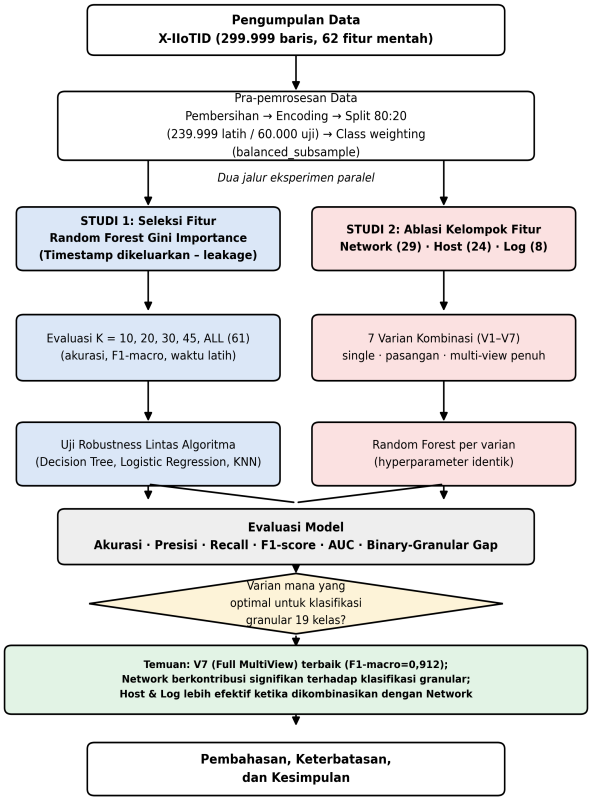
Aspek metodologis lain yang perlu diperhatikan pada dataset dengan komponen waktu, seperti X-IIoTID, adalah potensi kebocoran data (*data leakage*), yaitu kondisi ketika suatu fitur secara tidak sengaja membawa informasi identitas kelas yang tidak akan tersedia pada skenario *deployment* nyata, sehingga menghasilkan estimasi performa yang menyesatkan [19]. Fitur *Timestamp* pada X-IIoTID berpotensi menimbulkan kebocoran semacam ini karena setiap jenis serangan cenderung direkam pada sesi waktu yang terpisah dan hampir tidak tumpang tindih, sehingga fitur waktu dapat bertindak sebagai "kode rahasia" identitas kelas alih-alih sinyal perilaku serangan yang sesungguhnya.

Berdasarkan uraian di atas, penelitian ini menguji secara eksplisit dua pertanyaan penelitian pada dataset X-IIoTID menggunakan algoritma *Random Forest*. Pertama, apakah seleksi fitur berbasis *Random Forest importance* mampu mengurangi dimensi fitur secara signifikan tanpa menurunkan performa klasifikasi biner maupun multikelas granular, sebagaimana dilaporkan pada dataset IoT sejenis [9,10]. Kedua, penelitian ini menguji kontribusi kelompok fitur *Network*, *Host*, dan Log terhadap klasifikasi multikelas granular melalui studi ablasi tujuh kombinasi, untuk menguji asumsi bahwa penggabungan multi-view selalu meningkatkan performa deteksi intrusi IIoT.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan *Random Forest* sebagai model klasifikasi utama. Implementasi dilakukan menggunakan Python dengan pustaka *scikit-learn*, serta pustaka pendukung *pandas*, *NumPy*, *matplotlib* [18–21], dan *joblib*. Penelitian mencakup empat tahap utama: (1) prapemrosesan data, (2) seleksi fitur berbasis *Random Forest importance*, (3) ablasi kelompok fitur *multi-view*, dan (4) evaluasi model. Tahapan tersebut secara ringkas divisualisasikan pada Gambar 1.

P-ISSN : 2089-3353
E-ISSN : 2808-9162



Gambar 1. Diagram Alur Penelitian untuk Evaluasi Seleksi Fitur dan Ablasi *Multi-view* pada Dataset X-IIoTID

2.1. Dataset X-IIoTID

Dataset yang digunakan adalah X-IIoTID, dataset intrusi yang bersifat *connectivity-agnostic* dan *device-agnostic* untuk lingkungan *Industrial Internet of Things* [2]. Dataset ini terdiri atas fitur yang diekstraksi dari empat sumber: lalu lintas jaringan, sumber daya *host*, log aplikasi, dan log/peringatan sistem deteksi intrusi komersial (*OSSEC* dan *Zeek/Bro*). Pada penelitian ini digunakan sampel sebanyak 299.999 baris data dari dataset X-IIoTID, dengan 62 kolom fitur mentah sebelum prapemrosesan. Dataset memiliki dua skema label: label biner (*class3*: *Normal/Attack*) dan label multikelas granular (*class1*) dengan 19 kelas, terdiri atas satu kelas normal dan 18 jenis serangan yang mencakup antara lain *BruteForce*, *C&C*, *Dictionary*, *Discovering_resources*, *Exfiltration*, *Fake_notification*, *False_data_injection*, *Generic_scanning*, *MQTT_cloud_broker_subscription*, *MitM*, *Modbus_register_reading*, *RDOS*, *Reverse_shell*, *Scanning_vulnerability*, *TCP Relay*, *crypto-ransomware*, *fuzzing*, dan *insider_malicious*.

Data, penanganan nilai kosong (*missing value*) melalui imputasi, *encoding* variabel kategorikal (*Protocol* dan *Service*), serta penghapusan kolom yang tidak relevan untuk pemodelan (*Date*, *Scr_IP*, *Des_IP*, dan *class2*). Fitur numerik selanjutnya melalui proses penskalaan (*scaling*). Data kemudian dibagi menjadi data latih dan data uji dengan proporsi 80:20, menghasilkan 239.999 baris data latih dan 60.000 baris data uji. Penanganan ketidakseimbangan kelas pada penelitian ini dilakukan melalui pembobotan kelas (*class_weight = 'balanced_subsample'* untuk *Random Forest* dan *'balanced'* untuk algoritma pembanding) pada proses pelatihan model, bukan melalui *oversampling* data seperti *SMOTE* [14,15]. sebagai upaya menjaga distribusi data latih tetap merepresentasikan kondisi lapangan sekaligus menghindari risiko *data leakage* akibat penyisipan sampel sintetis sebelum pembagian data.

Skema *class_weight = 'balanced_subsample'* pada *Random Forest* menetapkan bobot tiap kelas berbanding terbalik dengan frekuensinya pada *bootstrap sample* setiap pohon, mengikuti formulasi umum pembobotan kelas pada *scikit-learn* [18]:

$$w_c = \frac{n}{c \cdot n_c} \quad (1)$$

dengan w_c adalah bobot kelas c, n adalah jumlah sampel (dihitung ulang pada tiap *bootstrap sample* untuk varian *balanced_subsample*, atau pada keseluruhan data latih untuk varian *balanced*), C adalah jumlah kelas ($C = 2$ untuk label biner, $C = 19$ untuk label multikelas *granular*), dan n_c adalah jumlah sampel kelas c. Bobot ini digunakan langsung oleh algoritma pembelajaran pada tahap pemilihan *split* dan penghitungan *loss function*, sehingga kelas minoritas memperoleh pengaruh yang proporsional lebih besar tanpa mengubah distribusi data itu sendiri.

2.2. Studi Seleksi Fitur berbasis *Random Forest Importance*

Studi pertama mengevaluasi efektivitas seleksi fitur berbasis skor kepentingan Gini dari *Random Forest*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

dalam mengurangi dimensi fitur tanpa mengorbankan performa klasifikasi, mengikuti preseden metodologi yang digunakan Hussein dkk [11]. Pada dataset IoTID20 dan Yin dkk. [12] pada dataset UNSW-NB15. Fitur *Timestamp* dikeluarkan dari seluruh eksperimen pada tahap ini karena teridentifikasi berpotensi menimbulkan *data leakage*: setiap jenis serangan pada X-IIoTID cenderung direkam pada sesi waktu yang terpisah dan hampir tidak tumpang tindih, sehingga *Timestamp* berisiko menjadi penanda identitas kelas secara tidak langsung, bukan sinyal perilaku serangan yang sesungguhnya. Setelah pengeluaran *Timestamp*, tersisa 61 fitur kandidat.

Skor kepentingan Gini (*Gini importance*) suatu fitur dihitung berdasarkan rata-rata penurunan impuritas Gini yang dihasilkan oleh fitur tersebut di seluruh pohon penyusun *Random Forest*. Impuritas Gini pada suatu simpul (node) t didefinisikan sebagai:

$$Gini(t) = 1 - \sum_{i=1}^{C} p_i(t)^2 \quad (2)$$

dengan $p_i(t)$ adalah proporsi sampel kelas i pada simpul t, dan C adalah jumlah kelas. Penurunan impuritas pada simpul t akibat pemisahan (*split*) dirumuskan sebagai:

$$\Delta I(t) = Gini(t) - \frac{N_{tL}}{N_t} \cdot Gini(t_L) - \frac{N_{tR}}{N_t} \cdot Gini(t_R) \quad (3)$$

dengan N_t, N_tL, dan N_tR berturut-turut adalah jumlah sampel pada simpul t, anak kiri t_L, dan anak kanan t_R. Skor kepentingan fitur x_j pada seluruh hutan (*forest*) dengan T pohon selanjutnya dihitung sebagai rata-rata penurunan impuritas terbobot di seluruh simpul yang menggunakan x_j sebagai kriteria *split*, mengikuti definisi *mean decrease impurity* (MDI):

$$Imp(x_j) = \frac{1}{T} \cdot \sum_{k=1}^{T} \sum_{t \in T_k:\, s(t)=x_j} \frac{N_t}{N} \cdot \Delta I(t) \quad (4)$$

dengan T_k adalah himpunan simpul internal pada pohon ke-k, $s(t) = x_j$ menandakan simpul t melakukan split menggunakan fitur x_j, dan N adalah jumlah total sampel data latih. Skor $Imp(x_j)$ inilah yang digunakan pada penelitian ini untuk mengurutkan fitur dan memilih subset fitur teratas pada tiap ambang K.

Skor kepentingan fitur dihitung menggunakan *Random Forest* (*n_estimators*=100, *max_depth*=15, *min_samples_leaf*=5, *class_weight*='balanced_subsample', *random_state*=42) yang dilatih pada label multikelas. Fitur kemudian diurutkan berdasarkan skor kepentingan, dan dievaluasi pada lima ambang jumlah fitur teratas (K = 10, 20, 30, 45, dan ALL/61 fitur). Pada setiap nilai K, model *Random Forest* dengan konfigurasi *hyperparameter* yang identik dilatih ulang untuk tugas klasifikasi biner maupun multikelas, dan dicatat akurasi, presisi, *recall*, F1-*score*, serta waktu pelatihan. Sebagai uji *robustness*, subset fitur pada K terbaik selanjutnya diuji ulang menggunakan tiga algoritma pembanding non-*Random Forest*, *Decision Tree*, *Logistic Regression*, dan *K-Nearest Neighbors* untuk memastikan bahwa fitur terpilih valid secara umum dan bukan sekadar artefak dari satu algoritma tertentu.

2.3. Studi Ablasi Kelompok Fitur Multi-View

Studi kedua, yang menjadi fokus utama dan kebaruan penelitian ini, mengevaluasi kontribusi masing-masing kelompok sumber fitur asli pada X-IIoTID terhadap performa klasifikasi. Dari 62 fitur mentah pada dataset, fitur Timestamp tidak disertakan pada studi ablasi ini karena berpotensi menimbulkan data leakage, sejalan dengan perlakuan pada studi seleksi fitur (Subbab 2.2). Sisa 61 fitur kemudian dikelompokkan ke dalam tiga kelompok sumber berdasarkan definisi bawaan dataset, yaitu *Network* (N) sebanyak 29 fitur (mencakup atribut koneksi seperti *Scr_port*, *Des_port*, *Protocol*, *Duration*, *byte_rate*, dan turunan statistik paket/*byte* lainnya), *Host* (H) sebanyak 24 fitur (statistik penggunaan CPU, memori, dan proses seperti *Avg_user_time*, *Avg_kbmemused*, dan *Avg_num_cswch/s*), serta *Log* (L) sebanyak 8 fitur (indikator log keamanan seperti *OSSEC_alert*, *Login_attempt*, dan *is_privileged*).

Secara formal, jika F menyatakan himpunan seluruh fitur mentah pada X-IIoTID, ketiga kelompok sumber tersebut membentuk partisi terhadap F:

$$F = N \cup H \cup L, \quad N \cap H = N \cap L = H \cap L = \emptyset \quad (5)$$

dengan $|N| = 30, |H| = 24, dan\ |L| = 8$. Ketujuh varian kombinasi $V1 - V7$ yang diuji pada studi ablasi merupakan seluruh himpunan bagian tak-kosong dari $\{N, H, L\}$:

$$V_i \subseteq \{N, H, L\},\ V_i \neq \emptyset,\ i = 1, 2, \ldots, 2^3 - 1 = 7 \quad (6)$$

Dari ketiga kelompok tersebut disusun tujuh varian kombinasi (V1–V7): (V1) *Network* saja, (V2) *Host* saja, (V3) *Log* saja, (V4) *Network*+*Host*, (V5) *Network*+*Log*, (V6) *Host*+*Log*, dan (V7) kombinasi ketiganya (*multi-view* penuh). Setiap varian dilatih menggunakan *Random Forest* dengan konfigurasi *hyperparameter* yang identik pada seluruh varian (mengacu pada konfigurasi dasar yang sama dengan tahap seleksi fitur pada subbab 2.2) untuk tugas klasifikasi biner dan multikelas granular secara terpisah, sehingga selisih performa antarvarian dapat diatribusikan pada perbedaan komposisi sumber fitur, bukan pada perbedaan *hyperparameter* model. Jumlah fitur pada tiap varian kombinasi ditunjukkan pada Tabel 5.

2.4. Evaluasi Model

Evaluasi performa klasifikasi menggunakan metrik akurasi, presisi, *recall*, F1-*score*, dan *area under the curve* (AUC), yang dihitung dari elemen *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN) pada *confusion matrix* [18]. Akurasi dihitung sebagai proporsi prediksi yang benar terhadap keseluruhan data:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

P-ISSN : 2089-3353
E-ISSN : 2808-9162

sedangkan presisi dan *recall* untuk tiap kelas i dihitung sebagai:

$$Presisi_i = \frac{TP_i}{TP_i + FP_i} \quad (8)$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (9)$$

dan *F1-score* kelas i merupakan rata-rata harmonis antara presisi dan *recall* kelas tersebut:

$$F1_i = 2 \cdot \frac{Presisi_i \cdot Recall_i}{Presisi_i + Recall_i} \quad (10)$$

Untuk klasifikasi multikelas, presisi, *recall*, dan *F1-score* dilaporkan menggunakan skema rata-rata makro (*macro-average*) [13]:

$$Metrik_{macro} = \frac{1}{C} \cdot \sum_{i=1}^{C} Metrik_i \quad (11)$$

dengan CCC adalah jumlah kelas dan $Metrik_i$ adalah nilai presisi, *recall*, atau F1 pada kelas i. Skema ini memberi bobot yang setara pada setiap kelas terlepas dari jumlah sampelnya, sehingga kelas minoritas berkontribusi setara dengan kelas mayoritas terhadap skor akhir—sesuai dengan karakteristik ketidakseimbangan kelas ekstrem pada X-IIoTID (subbab 3.1). Selain metrik standar tersebut, penelitian ini turut menghitung metrik *binary-granular accuracy gap*, yaitu selisih absolut antara akurasi klasifikasi biner dan akurasi klasifikasi multikelas granular pada varian fitur yang sama, sebagai indikator seberapa besar kesenjangan kesulitan antara tugas deteksi serangan sederhana (biner) dan tugas identifikasi jenis serangan secara rinci (*granular*):

$$Gap = |Akurasi_{biner} - Akurasi_{multikelas}| \quad (12)$$

3. Hasil dan Pembahasan

Sampel data yang digunakan berjumlah 299.999 baris, dengan distribusi kelas biner yang relatif seimbang antara kelas *Normal* (154.020 baris, 51,3%) dan kelas *Attack* (145.979 baris, 48,7%). Namun demikian, pada label multikelas granular (19 kelas), distribusi data sangat tidak seimbang, dengan rasio antara kelas terbesar (*Normal*, 154.020 baris) dan kelas terkecil (*Fake_notification*, 10 baris) mencapai lebih dari 15.000:1. Tabel 1 menunjukkan distribusi lengkap kelas multikelas pada sampel data yang digunakan.

Tabel 1. Distribusi Kelas Multikelas (Subtipe Serangan) pada Sampel Data

Kelas (*Subtipe* Serangan)	Jumlah
Normal	154.020
RDOS	51.628
Scanning_vulnerability	19.316
Generic_scanning	18.375
BruteForce	17.266
MQTT_*cloud_broker_subscription*	8.598
Discovering_resources	8.460
Exfiltration	8.090
insider_malicious	6.377
Modbus_register_reading	2.176
False_data_injection	1.862
C&C	1.046
Dictionary	940
TCP Relay	774
fuzzing	480
Reverse_shell	371
crypto-ransomware	167
MitM	43
Fake_notification	10

3.2. Hasil Studi Seleksi Fitur

Tabel 2 menyajikan hasil evaluasi performa klasifikasi pada lima ambang jumlah fitur teratas (K) berdasarkan skor kepentingan Random Forest. Hasil menunjukkan bahwa performa klasifikasi biner relatif stabil pada seluruh nilai K (akurasi berkisar 0,995–0,996), sedangkan performa klasifikasi multikelas granular meningkat secara monoton seiring bertambahnya jumlah fitur, dari F1-macro 0,876 pada K = 10 hingga 0,912 pada seluruh 61 fitur (K = ALL). Pola ini menunjukkan bahwa penambahan fitur memberikan manfaat yang lebih jelas pada tugas klasifikasi granular dibandingkan klasifikasi biner. Hal tersebut mengindikasikan bahwa fitur dengan skor kepentingan lebih rendah secara agregat masih dapat membawa informasi yang relevan untuk membedakan kelas-kelas serangan secara lebih spesifik.

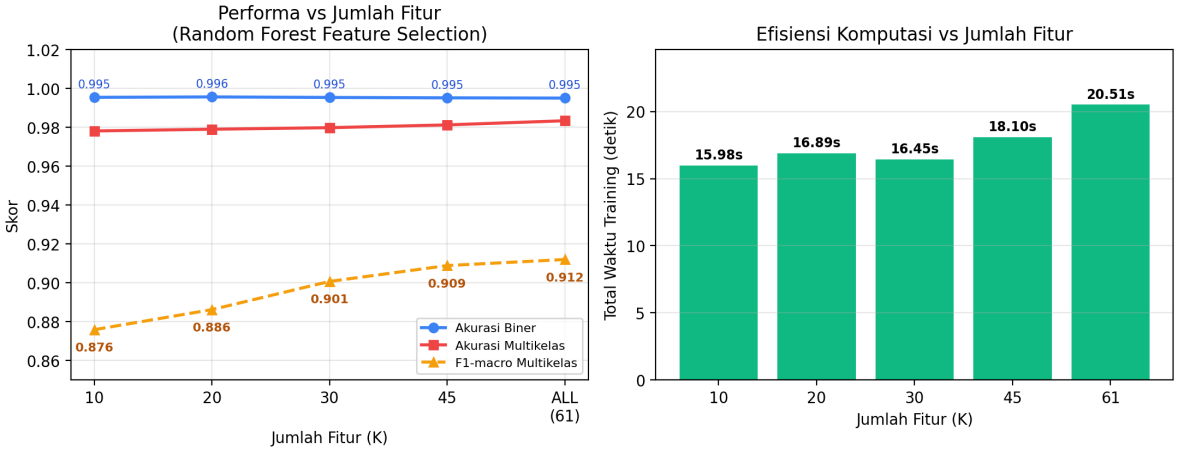
Tabel 2. Perbandingan Performa terhadap Jumlah Fitur (K) Hasil Seleksi Fitur *Random Forest*

K	n_fitur	Akurasi Biner	F1 Biner	Waktu Latih Biner (s)	Akurasi Multikelas	F1-macro Multikelas	Waktu Latih Multi (s)	Total Waktu (s)
10	10	0,995	0,996	8,03	0,978	0,876	7,95	15,98
20	20	0,996	0,996	8,34	0,979	0,886	8,55	16,89
30	30	0,995	0,996	8,05	0,980	0,901	8,40	16,45
45	45	0,995	0,995	8,98	0,981	0,909	9,12	18,10
ALL	61	0,995	0,995	9,70	0,983	0,912	10,81	20,51

Tabel 3. Hasil Uji *Robustness* Lintas Algoritma pada K Terbaik (*K=ALL*)

Algoritma	K	Akurasi Biner	F1 Biner	Akurasi Multikelas	F1-macro Multikelas
Decision Tree	ALL	0,990	0,991	0,969	0,897
Logistic Regression	ALL	0,957	0,959	0,845	0,703
K-Nearest Neighbors	ALL	0,982	0,982	0,980	0,867

P-ISSN : 2089-3353
E-ISSN : 2808-9162



Gambar 2. Performa vs Jumlah Fitur (kiri) dan Efisiensi Komputasi vs Jumlah Fitur (kanan) pada Studi Seleksi Fitur (label nilai ditampilkan pada tiap titik/batang)

Temuan ini berbeda dengan preseden pada literatur seleksi fitur berbasis Random Forest pada dataset IoT sejenis, yang berhasil mengurangi jumlah fitur secara signifikan pada dataset IoTID20 tanpa penurunan akurasi berarti. Pada dataset X-IIoTID dengan target klasifikasi 19 kelas granular, pengurangan jumlah fitur secara konsisten menurunkan performa F1-macro, mengindikasikan bahwa informasi yang dibawa oleh fitur-fitur dengan skor kepentingan yang lebih rendah tetap memberikan kontribusi diskriminatif bagi kelas-kelas minoritas, meskipun kontribusinya terhadap keseluruhan skor kepentingan Random Forest relatif kecil. Dengan demikian, pada tugas klasifikasi granular multikelas X-IIoTID, seleksi fitur berbasis ambang top-K tidak memberikan manfaat reduksi dimensi tanpa kompromi performa, kontras dengan efektivitasnya pada tugas klasifikasi biner atau pada dataset dengan jumlah kelas yang lebih sedikit.

Gambar 2 memperlihatkan bahwa peningkatan performa multikelas melambat (mengalami *diminishing returns*) setelah K = 30, namun tetap menunjukkan tren naik hingga K = ALL, sementara waktu komputasi total meningkat secara konsisten seiring bertambahnya jumlah fitur, dari 15,98 detik (K = 10) hingga 20,51 detik (K = ALL). *Trade-off* ini mengindikasikan bahwa pemilihan jumlah fitur pada *dataset* X-IIoTID perlu mempertimbangkan konteks operasional: jika prioritas utama adalah efisiensi komputasi pada skenario *deployment real-time* dengan sedikit toleransi terhadap penurunan F1-macro, subset K = 30 hingga K = 45 dapat menjadi kompromi yang wajar; namun jika prioritas utama adalah akurasi identifikasi granular jenis serangan, seluruh fitur (K = ALL) tetap menjadi pilihan optimal berdasarkan hasil penelitian ini.

Sebagai uji *robustness*, subset K = ALL (fitur terbaik berdasarkan F1-macro tertinggi) diuji ulang menggunakan tiga algoritma pembanding non-*Random Forest*. Tabel 3 menunjukkan bahwa performa relatif konsisten lintas algoritma, dengan Decision Tree dan K-*Nearest Neighbors* mencapai F1-macro multikelas masing-masing 0,897 dan 0,867, sementara Logistic *Regression* tertinggal signifikan (F1-macro = 0,703), mengindikasikan bahwa hubungan antara fitur dan kelas serangan pada X-IIoTID cenderung bersifat non-linear sehingga kurang cocok dimodelkan oleh algoritma linear seperti *Logistic Regression.*

3.3. Hasil Studi Ablasi Kelompok Fitur Multi-*View*

Bagian ini menyajikan temuan utama penelitian, yaitu kontribusi masing-masing kelompok sumber fitur (*Network, Host, Log)* terhadap performa klasifikasi. Definisi jumlah fitur pada tiap kelompok ditunjukkan pada Tabel 4, sedangkan hasil lengkap ketujuh varian kombinasi ditunjukkan pada Tabel 5.

Tabel 4. Definisi Kelompok Sumber Fitur pada X-IIoTID

Kelompok	Jml. Fitur	Contoh Fitur
Network (N)	29	*Scr_port, Des_port, Protocol, Duration, byte_rate, total_packet,* dst. (Timestamp dikeluarkan karena data leakage)
Host (H)	24	*Avg_user_time, Avg_kbmemused, Avg_tps, Avg_num_cswch/s*, dst.
Log (L)	8	*OSSEC_alert, Login_attempt, File_activity, is_privileged,* dst.

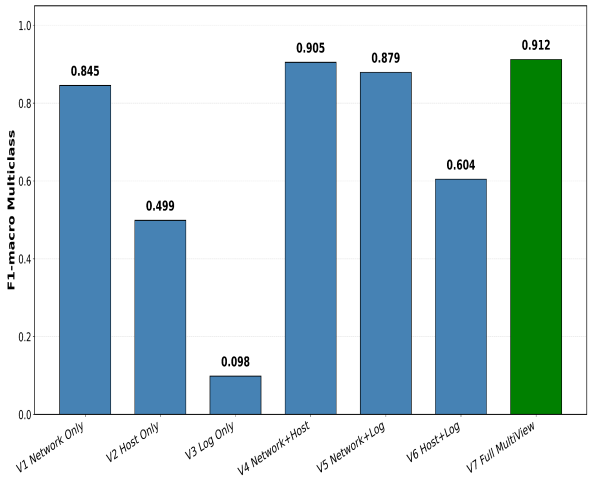
Tabel 5. Perbandingan Performa Tujuh Varian Kombinasi Kelompok Fitur

Varian	n_fitur	Akurasi Biner	F1 Biner	Akurasi Multikelas	Presisi-macro Multikelas	Recall-macro Multikelas	F1-macro Multikelas	Gap Biner-Granular
V7 *Full MultiView*	61	0,995	0,995	0,983	0,910	0,961	0,912	0,0117
V4 *Network+Host*	53	0,993	0,993	0,977	0,897	0,960	0,905	0,0156
V5 *Network+Log*	37	0,996	0,996	0,963	0,866	0,949	0,879	0,0328
V1 *Network Only*	29	0,988	0,988	0,946	0,825	0,930	0,845	0,0420

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Varian	n_fitur	Akurasi Biner	F1 Biner	Akurasi Multikelas	Presisi-macro Multikelas	Recall-macro Multikelas	F1-macro Multikelas	Gap Biner-Granular
V6 *Host+Log*	32	0,902	0,909	0,640	0,576	0,837	0,604	0,2621
V2 *Host Only*	24	0,877	0,887	0,641	0,458	0,771	0,499	0,2359
V3 Log Only	8	0,601	0,542	0,090	0,248	0,294	0,098	0,5114

Tabel 5 menunjukkan bahwa kontribusi setiap kelompok fitur terhadap performa klasifikasi granular bergantung pada kombinasi yang digunakan, dan setelah fitur *Timestamp* dikeluarkan dari kelompok *Network*, penambahan kelompok lain tidak lagi secara konsisten menurunkan F1-macro multikelas. F1-macro meningkat dari 0,845 pada V1 (*Network Only*) menjadi 0,905 ketika *Host* ditambahkan pada V4 (*Network+Host)*, dan menjadi 0,879 ketika Log ditambahkan pada V5 (*Network+Log*); penggabungan ketiga kelompok pada V7 (*Full MultiView*) menghasilkan nilai tertinggi sebesar 0,912. Sebaliknya, penggunaan *Host* tanpa *Network* menghasilkan performa jauh lebih rendah, yaitu 0,499 pada V2 (*Host Only*) dan 0,604 pada V6 (*Host+Log*). Hasil ini menunjukkan bahwa kelompok Network memiliki kontribusi penting terhadap klasifikasi granular, sementara kontribusi *Host* dan *Log* menjadi jauh lebih efektif ketika dikombinasikan dengan *Network*; dengan demikian, tidak terdapat bukti bahwa kelompok *Host* secara konsisten bertindak sebagai sumber derau (*noise*) pada konfigurasi penelitian ini.



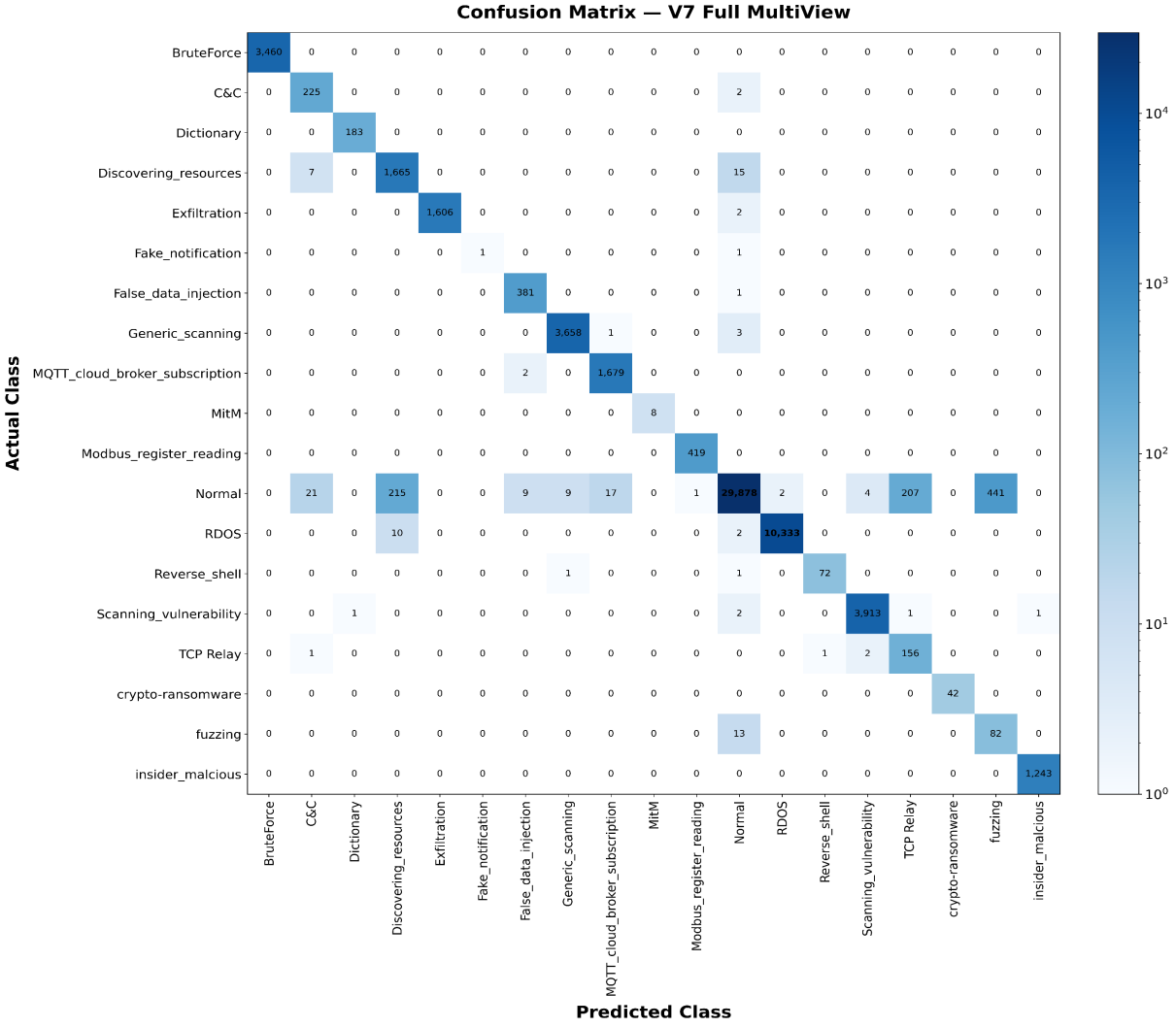
Gambar 3. Perbandingan F1-macro Multikelas Antarvarian Kombinasi Kelompok Fitur (label nilai ditampilkan pada tiap batang; batang hijau = performa terbaik)

Analisis lebih lanjut terhadap Tabel 5 dan Gambar 3 menunjukkan bahwa kontribusi kelompok fitur tidak bersifat aditif secara sederhana. Kelompok Log ketika digunakan secara mandiri pada V3 menghasilkan F1 macro sebesar 0,098, yang menunjukkan bahwa informasi Log saja belum memadai untuk membedakan 19 kelas secara granular. Ketika dikombinasikan dengan *Network* pada V5, F1-macro meningkat menjadi 0,879. Selanjutnya, kombinasi seluruh kelompok fitur pada V7 menghasilkan F1-*macro* yang lebih tinggi, yaitu 0,912. Pola tersebut menunjukkan bahwa informasi Log menjadi lebih bermanfaat ketika dikombinasikan dengan kelompok fitur lainnya, sedangkan integrasi ketiga kelompok fitur pada V7 menghasilkan performa klasifikasi granular tertinggi pada konfigurasi penelitian ini.

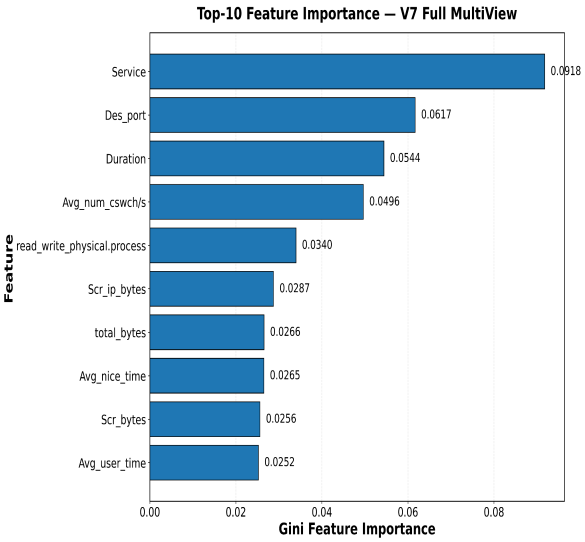
Perbedaan antara performa biner dan multikelas juga terlihat dari nilai Gap Biner-Granular. Di antara varian yang melibatkan kelompok *Network*, V7 memiliki gap terendah sebesar 0,0117, diikuti oleh V4 sebesar 0,0156, V5 sebesar 0,0328, dan V1 sebesar 0,0420. Nilai tersebut menunjukkan bahwa kombinasi *Full MultiView* tidak hanya menghasilkan F1-*macro* multikelas tertinggi, tetapi juga mempertahankan kesenjangan akurasi yang relatif kecil antara tugas klasifikasi biner dan granular. Sebaliknya, varian yang hanya menggunakan *Log* atau *Host* memiliki gap yang lebih besar, masing-masing sebesar 0,5114 pada V3 dan 0,2359 pada V2, yang menunjukkan bahwa penggunaan kedua kelompok fitur tersebut secara mandiri kurang memadai untuk mempertahankan performa ketika tugas diperluas dari deteksi biner menjadi identifikasi 19 kelas secara granular.

Gambar 4 menyajikan *confusion matrix* klasifikasi multikelas pada varian terbaik, yaitu V7 (*Full Multi View)*, berdasarkan hasil prediksi pada 60.000 data uji. *Confusion matrix* digunakan untuk memperlihatkan distribusi prediksi benar dan kesalahan klasifikasi pada 19 kelas secara lebih rinci. Penyajian dengan skala warna logaritmik digunakan untuk mempertahankan keterbacaan perbedaan jumlah prediksi antar-kelas yang memiliki dukungan data berbeda. Hasil *confusion matrix* memberikan gambaran mengenai kemampuan V7 dalam mempertahankan klasifikasi granular serta pola kesalahan prediksi yang masih terjadi antar-kelas.

P-ISSN : 2089-3353
E-ISSN : 2808-9162



Gambar 4. *Confusion Matrix* Klasifikasi Multikelas pada Varian Terbaik (V7: *Full MultiView*), dengan angka aktual per sel dan skala warna logaritmik (n = 60.000 data uji).



Gambar 5. Kepentingan Fitur (*Feature Importance*) pada Varian Terbaik (V7: *Full MultiView*)

Gambar 5 menunjukkan sepuluh fitur dengan kontribusi tertinggi pada varian V7 (*Full MultiView*). Karena V7 menggabungkan kelompok *Network, Host*, dan *Log, feature importance* pada konfigurasi ini digunakan untuk mengidentifikasi fitur yang paling berkontribusi terhadap keputusan klasifikasi *Random Forest* dalam representasi *multi-view*. Hasil tersebut memberikan gambaran mengenai sumber informasi yang paling dominan dalam membedakan subtipe serangan IIoT pada tugas klasifikasi granular 19 kelas.

3.4. Pembahasan dan Keterbatasan Penelitian

Secara keseluruhan, hasil penelitian ini memberikan dua kontribusi utama. Pertama, pada aspek seleksi fitur, hasil menunjukkan bahwa strategi seleksi fitur top-K berbasis skor kepentingan agregat kurang efektif pada tugas klasifikasi granular multikelas dengan ketidakseimbangan kelas ekstrem, karena pengurangan jumlah fitur tidak mampu mempertahankan performa terbaik yang diperoleh ketika seluruh fitur kandidat digunakan. Kedua, studi ablasi kelompok fitur menunjukkan bahwa kontribusi sumber fitur bergantung pada kombinasi yang digunakan. Setelah fitur *Timestamp*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

dikeluarkan dari kelompok *Network*, kombinasi Full *MultiView* (V7) yang menggabungkan Network, Host, dan Log menghasilkan performa multikelas granular tertinggi dengan F1-macro sebesar 0,912, diikuti oleh V4 (*Network+Host*) sebesar 0,905 dan V5 (*Network+Log*) sebesar 0,879. Temuan ini menunjukkan bahwa pada konfigurasi penelitian ini, penggabungan ketiga kelompok fitur memberikan representasi paling efektif untuk klasifikasi granular 19 kelas. Dengan demikian, evaluasi kontribusi setiap kelompok sumber fitur tetap penting sebelum menentukan konfigurasi fitur yang digunakan pada sistem deteksi intrusi IIoT.

Penelitian ini memiliki sejumlah keterbatasan yang perlu menjadi perhatian pada penelitian lanjutan. Pertama, perbedaan perlakuan terhadap fitur *Timestamp* antara studi seleksi fitur dan studi ablasi telah dikoreksi dengan mengeluarkan *Timestamp* dari kelompok *Network* pada evaluasi ablasi. Dengan demikian, varian yang melibatkan *Network* tidak lagi menggunakan *Timestamp* sebagai sumber informasi, sehingga perbandingan antara studi seleksi fitur dan studi ablasi menjadi lebih konsisten. Kedua, penelitian ini menggunakan sampel sebanyak 299.999 baris dari dataset X-IIoTID yang berukuran lebih besar dan belum membandingkan hasil ablasi secara menyeluruh dengan algoritma *non-tree-based* atau pendekatan pembelajaran mendalam. Ketiga, penelitian ini belum menguji mekanisme fusi fitur yang lebih adaptif, seperti pembobotan kelompok fitur secara otomatis (*feature-group-aware fusion* atau *adaptive feature weighting*), yang dapat memberikan mekanisme lebih fleksibel dalam menentukan kontribusi masing-masing sumber fitur pada klasifikasi granular.

Keterbatasan lain berkaitan dengan aspek kompleksitas waktu dan komputasi pada implementasi di perangkat *edge* IIoT berdaya rendah, khususnya untuk varian V5 (*Network+Log*). Setelah *Timestamp* dikeluarkan dari kelompok *Network*, V5 menggunakan 37 fitur, sehingga dimensi inputnya lebih rendah dibandingkan V7 yang menggunakan 61 fitur. Kondisi tersebut berpotensi mengurangi beban pemrosesan dibandingkan konfigurasi dengan jumlah fitur yang lebih besar. Namun, V5 tetap menggunakan *Random Forest* sehingga kebutuhan komputasi dan memori tidak hanya ditentukan oleh jumlah fitur, tetapi juga oleh struktur dan jumlah pohon keputusan yang digunakan dalam model. Pada perangkat *edge* dengan sumber daya terbatas, proses inferensi dan penyimpanan model perlu mempertimbangkan keterbatasan CPU, memori, serta konsumsi energi. Penelitian ini belum melakukan pengukuran langsung terhadap latensi inferensi, penggunaan CPU, penggunaan memori, dan konsumsi energi V5 pada perangkat *edge* IIoT berdaya rendah. Oleh karena itu, hasil penelitian ini belum dapat digunakan untuk menyatakan bahwa V5 telah memenuhi batasan komputasi perangkat *edge* secara operasional. Evaluasi langsung pada perangkat *edge* dengan pengukuran latensi, penggunaan memori, CPU, dan energi diperlukan pada penelitian lanjutan untuk menentukan kelayakan deployment model secara nyata.

4. Kesimpulan

Penelitian ini mengevaluasi dua strategi optimasi fitur pada klasifikasi subtipe serangan IIoT granular (19 kelas) menggunakan dataset X-IIoTID dan algoritma *Random Forest*. Pada studi seleksi fitur berbasis *Random Forest importance*, performa klasifikasi multikelas granular meningkat secara monoton seiring bertambahnya jumlah fitur, sehingga tidak ditemukan subset fitur yang lebih kecil dari keseluruhan 61 fitur yang mampu mempertahankan performa F1-*macro* terbaik (0,912). Pada studi ablasi kelompok fitur *multi-view*, setelah fitur *Timestamp* dikeluarkan dari kelompok *Network*, kombinasi *Full MultiView* (V7) yang menggunakan 61 fitur mencapai performa multikelas granular terbaik dengan F1-*macro* = 0,912, diikuti V4 (*Network+Host*) sebesar 0,905, V5 (*Network+Log*) sebesar 0,879, dan V1 (*Network Only*) sebesar 0,845. Fitur *Log* menunjukkan kontribusi yang lebih kuat ketika dikombinasikan dengan *Network* dibandingkan ketika digunakan secara mandiri, sedangkan fitur *Host* menunjukkan performa yang terbatas ketika digunakan sendiri tetapi memberikan kontribusi positif ketika dikombinasikan dengan *Network* dan *Log*.

Temuan tersebut menegaskan pentingnya evaluasi kontribusi tiap kelompok sumber fitur secara eksplisit sebelum perancangan sistem deteksi intrusi IIoT berbasis multi-sumber data. Hasil studi ablasi menunjukkan bahwa kontribusi suatu kelompok fitur bergantung pada kombinasi sumber fitur yang digunakan, sehingga penggabungan multi-view perlu dievaluasi berdasarkan kontribusi aktual masing-masing kelompok, bukan hanya berdasarkan jumlah sumber fitur. Sebagai arah penelitian selanjutnya (*future work*), hasil penelitian ini dapat dikembangkan dengan merancang mekanisme pemilihan dan pembobotan kelompok fitur secara adaptif, sehingga kontribusi setiap sumber fitur (*Network, Host,* dan *Log*) dapat disesuaikan secara dinamis terhadap karakteristik data maupun perubahan pola serangan. Selain itu, perlu dilakukan evaluasi pada berbagai dataset IIoT/IoT lainnya, algoritma *non-tree-based* maupun *deep learning*, serta skenario *streaming* dan *concept drift* untuk menguji konsistensi temuan dan meningkatkan generalisasi pendekatan yang diusulkan pada lingkungan IIoT nyata.

Daftar Rujukan

[1] F. Apri Wenando, "Peran Penggunaan IoT dengan Machine Learning dalam Penanganan Pandemi COVID-19: Systematic Literatur Review," *J. FASILKOM*, vol. 13, no. 02, pp. 318–325, Aug. 2023, doi: 10.37859/jf.v13i02.5651.

[2] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-IIoTID: A Connectivity-Agnostic and Device-Agnostic Intrusion Data Set for Industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3962–3977, 2022, doi: 10.1109/JIOT.2021.3102056.

[3] A. Alnajim, S. Habib, M. Islam, S. Thwin, and F. Alotaibi, "A Comprehensive Survey of Cybersecurity Threats, Attacks, and

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Effective Countermeasures in Industrial Internet of Things," *Technologies*, vol. 11, no. 6, p. 161, Nov. 2023, doi: 10.3390/technologies11060161.

[4] S. Soni, Afdhil Hafid, and Didik Sudyana, "ANALISIS KESADARAN MAHASISWA UMRI TERKAIT PENGGUNAAN TEKNOLOGI & MEDIA SOSIAL TERHADAP BAHAYA CYBERCRIME," *J. FASILKOM*, vol. 9, no. 3, pp. 28–34, Nov. 2019, doi: 10.37859/jf.v9i3.1664.

[5] B. Alotaibi, "A Survey on Industrial Internet of Things Security: Requirements, Attacks, AI-Based Solutions, and Edge Computing Opportunities," *Sensors*, vol. 23, no. 17, p. 7470, Aug. 2023, doi: 10.3390/s23177470.

[6] J. Yu, G. Wang, N. Shi, R. Saxena, and B. Lee, "A Multi-View-Based Federated Learning Approach for Intrusion Detection," *Electronics*, vol. 14, no. 21, p. 4166, Oct. 2025, doi: 10.3390/electronics14214166.

[7] S. Lee, D. Roh, J. Yu, D. Moon, J. Lee, and J.-H. Bae, "Deep Feature Fusion via Transfer Learning for Multi-Class Network Intrusion Detection," *Appl. Sci.*, vol. 15, no. 9, 2025, doi: 10.3390/app15094851.

[8] C. Xu, D. Li, Z. Liu, J. Yang, Q. Shen, and N. Tong, "Few-shot network intrusion detection method based on multi-domain fusion and cross-attention," *PLoS One*, vol. 20, no. 7, p. e0327161, Jul. 2025, doi: 10.1371/journal.pone.0327161.

[9] R. W. Anwer, M. Abrar, M. Ullah, A. Salam, and F. Ullah, "Advanced intrusion detection in the industrial Internet of Things using federated learning and LSTM models," *Ad Hoc Networks*, vol. 178, p. 103991, Nov. 2025, doi: 10.1016/j.adhoc.2025.103991.

[10] A. Termanini, H. Bourdoucen, D. Al-Abri, and A. Al Maashri, "Intrusion Detection Datasets for IIoT and ICS: A Taxonomic Review with a Decision-Aid Scoring Rubric," *Sensors*, vol. 26, no. 13, p. 4099, Jun. 2026, doi: 10.3390/s26134099.

[11] A. Y. Hussein, P. Falcarin, and A. T. Sadiq, "IoT Intrusion Detection Using Modified Random Forest Based on Double Feature Selection Methods," 2022, pp. 61–78. doi: 10.1007/978-3-030-97255-4_5.

[12] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *J. Big Data*, vol. 10, no. 1, p. 15, Feb. 2023, doi: 10.1186/s40537-023-00694-8.

[13] O. Abu Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, "An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning," *Expert Syst. Appl.*, vol. 213, p. 118745, Mar. 2023, doi: 10.1016/j.eswa.2022.118745.

[14] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[15] V. Shanmugam, R. Razavi-Far, and E. Hallaji, "Addressing Class Imbalance in Intrusion Detection: A Comprehensive Evaluation of Machine Learning Approaches," *Electronics*, vol. 14, no. 1, p. 69, Dec. 2024, doi: 10.3390/electronics14010069.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[17] L.-H. Li, R. Ahmad, R. Tanone, and A. K. Sharma, "STB: synthetic minority oversampling technique for tree-boosting models for imbalanced datasets of intrusion detection systems," *PeerJ Comput. Sci.*, vol. 9, p. e1580, Nov. 2023, doi: 10.7717/peerj-cs.1580.

[18] S. A. Wahab, S. Sultana, N. Tariq, M. Mujahid, J. A. Khan, and A. Mylonas, "A Multi-Class Intrusion Detection System for DDoS Attacks in IoT Networks Using Deep Learning and Transformers," *Sensors*, vol. 25, no. 15, p. 4845, Aug. 2025, doi: 10.3390/s25154845.

[19] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, Sep. 2023, doi: 10.1016/j.patter.2023.100804.

[20] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[21] W. McKinney, "Data Structures for Statistical Computing in Python," 2010, pp. 56–61. doi: 10.25080/Majora-92bf1922-00a.

[22] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[23] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.