Pengembangan Model Machine Learning untuk Deteksi Serangan Siber

Nurdin Andi Baso Daeng Marewa[1)*], **Muhammad Adrinta Abdurrazzaq**[2)]

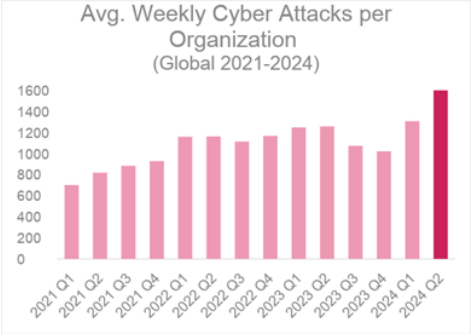
[1,2)]Infromatika, Fakultas Ilmu Komputer dan Desain, Universitas Kalbis
Jalan Pulomas Selatan Kav, 22 Jakarta 132210
Email: nurdinandi123@gmail.com
Email: Muhammad.abdurrazzaq@kalbis.ac.id

Abstract: *This study proposes a deep learning-based Intrusion Detection System (IDS) by combining Convolutional Neural Network (CNN) and Random Forest (RF) to detect network attacks. The CICIDS2018 dataset is used as training data to recognize various types of attacks such as DDoS and Brute Force. CNN acts as a feature extractor, while RF is used for classification. This system is implemented as a web application with an interactive interface for ease of use. Test results show that the hybrid CNN-RF model achieves high accuracy (91%) and is superior to both the single CNN model and CNN-XGBoost. This approach improves attack detection accuracy and provides an adaptive and efficient solution for network security*
Keywords: *convolutional neural networks, network security, random forest, intrusion detection system.*

Abstract: *Penelitian ini mengusulkan Intrusion Detection System (IDS) berbasis deep learning dengan menggabungkan Convolutional Neural Network (CNN) dan Random Forest (RF) untuk mendeteksi serangan jaringan. Dataset CICIDS2018 digunakan sebagai data latih untuk mengenali berbagai jenis serangan seperti DDoS dan Brute Force. CNN berperan sebagai ekstraktor fitur, sedangkan RF digunakan untuk klasifikasi. Sistem ini diimplementasikan sebagai aplikasi web dengan antarmuka interaktif agar mudah digunakan. Hasil pengujian menunjukkan model hybrid CNN-RF mencapai akurasi tinggi (91%) dan lebih unggul dibandingkan model CNN tunggal maupun CNN-XGBoost. Pendekatan ini meningkatkan akurasi deteksi serangan dan memberikan solusi adaptif serta efisien bagi keamanan jaringan.*
Kata Kunci: *convolutional neural networks, keamanan jaringan, random forest, system deteksi intrusi*

I. PENDAHULUAN

Dalam era digital yang semakin berkembang, keamanan jaringan menjadi aspek krusial dalam menjaga integritas, kerahasiaan, dan ketersediaan data. Meningkatnya jumlah perangkat yang terhubung ke internet serta kompleksitas infrastruktur jaringan telah membuka celah bagi berbagai jenis serangan siber. Beberapa jenis serangan siber yang umum terjadi diantaranya adalah Distributed Denial of Service (DDoS), SQL Injection, Man-in-the-Middle (MitM), serta serangan berbasis malware dan ransomware. Serangan tersebut dapat mengakibatkan pencurian data, gangguan layanan, hingga kerugian finansial bagi individu maupun kelompok.



Gambar 1 Cyber Attack Statistics, 2024, Check Point

Menurut laporan tahun 2024 (Gambar 1) oleh yang didapatkan dari berbagai lembaga keamanan siber, jumlah serangan siber yang terjadi di dunia semakin meningkat[1]. Intrusion Detection System (IDS) hadir sebagai salah satu solusi dalam menghadapi ancaman tersebut. IDS berperan untuk melakukan pemantauan terhadap lalu lintas jaringan dan mengidentifikasi aktivitas mencurigakan yang berpotensi sebagai ancaman keamanan. Di sisi lain, semakin variatifnya pola - pola serangan siber membuat IDS juga harus berkembang. IDS konvensional yang berbasis aturan (rule-based) dan tanda tangan (signature-based) memiliki keterbatasan dalam mengahadapi serangan yang semakin canggih sehingga pendeteksian yang dilakukan semakin tidak akurat, terlihat dari tingginya jumlah false postive dan false negative. Maka dari itu, pengembangan model machine learning yang mampu menghadapi perkembangan serangan menjadi tantangan tersendiri. Perkembangan deep learning yang pesat memungkinkan penerapannya dalam pengembangan sistem deteksi serangan. Deep learning mampu mengenali pola - pola serangan yang sulit di deteksi oleh metode - metode konvensional seperti rule-based dan siganture-based. Salah satu arsitektur deep learning yang lumrah digunakan untuk mengembangkan sistem untuk mendeteksi serangan siber adalah Convolutional Neural Network (CNN).

Convolutional Neural Network (CNN) adalah arsitektur deep learning yang telah terbukti efektif dalam menganalisis berbagai jenis data, termasuk teks, gambar, dan suara. Keunggulan utama CNN terletak pada kemampuannya untuk mengenali pola-pola kompleks dalam data melalui beberapa lapisan pemrosesan, seperti lapisan konvolusi dan pooling [2][3]. Selain CNN, metode Random Forest (RF) juga digunakan dalam sistem deteksi intrusi untuk meningkatkan akurasi klasifikasi. RF adalah algoritma machine learning berbasis ensemble yang terdiri dari banyak pohon keputusan (decision trees) dan bekerja dengan prinsip voting untuk menghasilkan keputusan akhir. Keunggulan utama RF terletak pada kemampuannya dalam menangani data yang kompleks serta mengurangi overfitting dibandingkan dengan model decision tree tunggal [4].

Sebagai respons terhadap tantangan tersebut, pendekatan berbasis deep learning telah berkembang pesat dalam bidang keamanan jaringan. Deep learning memungkinkan sistem untuk mengenali pola-pola anomali yang terdapat pada lalu lintas jaringan. Studi sebelumnya menunjukkan bahwa pendekatan berbasis deep learning ini dapat memberikan akurasi yang lebih tinggi dalam mendeteksi berbagai jenis serangan serta mengurangi terjadinya kesalahan deteksi. Penelitian ini akan mengembangkan model machine learning untuk deteksi serangan siber menggunakan arsitektur CNN dan RF. Model machine learning yang dikembangakan akan bekerja berdasarkan data yang diambil dari file packet capture (PCAP) yang telah dikonversi menjadi format CSV. Di sisi lain, penelitian ini juga akan mengembangkan aplikasi berbasis web untuk memudahkan pengunaan dalam mengunggah file PCAP dan memperoleh hasil deteksi serangan siber.

II. METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini adalah Incremental Model. Metode ini merupakan salah satu model pengembangan sistem yang dilakukan secara bertahap (inkremental). Setiap tahapan atau inkremen dalam pengembangan sistem melewati proses lengkap, mulai dari analisis, desain, implementasi, hingga pengujian, sebelum dilanjutkan ke inkremen berikutnya. Pemilihan metode ini didasarkan pada kebutuhan untuk melakukan evaluasi secara berkala dan integrasi bertahap

terhadap komponen sistem yang dikembangkan, khususnya dalam implementasi model deep learning dan integrasi ke dalam sistem berbasis web.

A. Incremental 1

Tahap Inkremental I dilakukan untuk mengekstrak fitur dari dataset CICIDS2018 yang akan digunakan untuk merancang model untuk mengenali pola serangan.

1. Analisis

Penelitian ini diawali dengan pemilihan dataset CICIDS2018 dalam format Comma-Separated Values (.csv) untuk mempermudah proses pelatihan model machine learning. Dataset tersebut terdiri dari 79 atribut, 15 jenis serangan, dan total 9.625.148 entri data. Gambar 2 menunjukkan distribusi label serangan yang terdapat dalam dataset CICIDS2018.

Label	
Benign	6876913
DDOS attack-HOIC	686012
DDoS attacks-LOIC-HTTP	576191
DoS attacks-Hulk	461912
Bot	286191
FTP-BruteForce	193360
SSH-Bruteforce	187589
Infilteration	161934
DoS attacks-SlowHTTPTest	139890
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DDOS attack-LOIC-UDP	1730
Brute Force -Web	611
Brute Force -XSS	230
SQL Injection	87

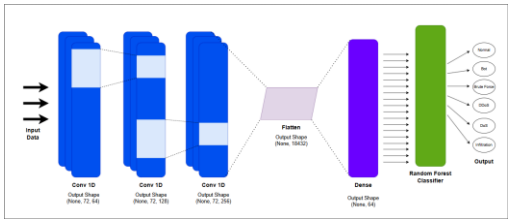
Gambar 2 Label Dataset Jenis Serangan

2. Desain

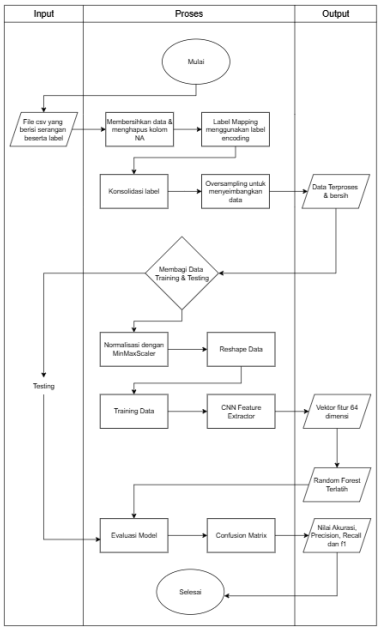
Desain sistem berfokus pada arsitektur *hybrid* yang menggabungkan CNN dan Random Forest untuk mendeteksi serangan jaringan menggunakan dataset CICIDS2018. Data diproses melalui normalisasi (*MinMaxScaler*), *reshaping* ke format 3D untuk CNN, dan konversi label ke numerik.

CNN dirancang dengan tiga lapisan *Conv1D* (64, 128, 256 filter) menggunakan aktivasi ReLU, *Batch Normalization*, *Dropout* (0.3), dan *MaxPooling*. Setelah proses *Flatten* dan lapisan *Dense* (128, 64 neuron), fitur yang dihasilkan diklasifikasikan menggunakan Random Forest (*n_estimators*=100).

Pelatihan dilakukan dengan *optimizer Adam* selama 30 *epoch* pada GPU. Evaluasi menggunakan *confusion matrix*, akurasi, presisi, *recall*, dan F1-score. Alur arsitektur dan proses sistem ditunjukkan pada Gambar 3 dan 4.



Gambar 3 Arsitektur CNN + RF



Gambar 4 Desain alur Inkremental 1

3. Implementasi

Tahap implementasi dimulai dengan pemrosesan dataset CICIDS2018 yang telah dikonversi ke format CSV. File CSV dibaca secara bertahap dengan metode chunking untuk menghindari penggunaan memori berlebih, kemudian digabung

menjadi satu dataset besar. Data yang kosong dihapus agar kualitas dataset tetap terjaga. Selanjutnya, label spesifik dari jenis serangan disederhanakan menjadi enam kategori umum, yaitu Normal, DoS, DDoS, Brute Force, Bot, dan Infiltration.

Proses berikutnya adalah mengatasi *class imbalance* yang terjadi karena perbedaan jumlah data pada setiap kelas. Teknik *undersampling* digunakan agar semua kelas memiliki jumlah data yang seimbang, sehingga model tidak bias terhadap kelas mayoritas. Setelah data seimbang, dilakukan pembagian dataset menjadi data latih dan data uji dengan perbandingan 80:20 untuk memastikan distribusi yang seimbang pada kedua subset.

Label hasil encoding dari *LabelEncoder* kemudian diubah ke bentuk *one-hot encoding* agar dapat digunakan oleh model CNN. Selain itu, fitur yang tidak relevan atau mengandung nilai tidak valid, seperti "Protocol", "PSH Flag Cnt", "Flow Byts/s", dan "Flow Pkts/s", dihapus karena dapat menurunkan akurasi model. Setelah fitur dipilih, dilakukan normalisasi menggunakan *MinMaxScaler* untuk mengubah nilai ke rentang [0,1], yang penting untuk stabilitas dan kecepatan pelatihan model deep learning.

Tahap selanjutnya adalah menyiapkan data untuk CNN dengan mengubah bentuk data (*reshape*) menjadi tiga dimensi agar sesuai dengan input Conv1D. CNN dibangun dengan tiga blok lapisan *Conv1D* yang masing-masing memiliki filter 64, 128, dan 256, disertai *Batch Normalization*, *MaxPooling*, dan *Dropout* untuk mencegah *overfitting*. Lapisan akhir terdiri dari *Flatten* dan dua *Dense layer* yang berfungsi sebagai *feature extractor*. Pelatihan CNN dilakukan selama 30 *epoch* menggunakan *optimizer Adam* dan *mixed precision training* untuk efisiensi komputasi.

Setelah pelatihan selesai, lapisan *Dense* terakhir digunakan untuk mengekstrak fitur yang akan menjadi input bagi algoritma *Random Forest* untuk klasifikasi. Random Forest dipilih karena kemampuannya menangani data kompleks dan mengurangi risiko *overfitting*. Sebagai pembanding, dilakukan pula pelatihan menggunakan XGBoost untuk mengevaluasi perbedaan performa dengan Random Forest. Evaluasi keseluruhan model dilakukan menggunakan metrik akurasi, presisi, *recall*, F1-score, serta *confusion matrix* pada data uji.

4. Pengujian

Tahap pengujian akan menggunakan *confusion matrix* untuk menguji model pada testing dengan evaluasi model menggunakan nilai *accuracy*, *precicsion*, *recall* dan *f1*.

B. Inkremental II

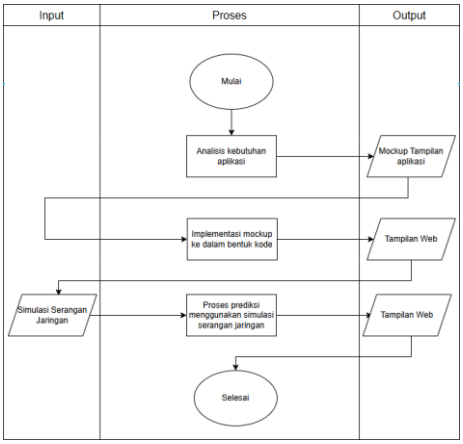
Tahap Inkremental II akan dilakukan untuk membangun UI dari Intrusion Detection System yang sudah dibuat agar dapat digunakan dengan lebih mudah.

1. Analisis

Tahap Incremental II merupakan kelanjutan dari tahap sebelumnya, yang berfokus pada implementasi model machine learning untuk mendeteksi jaringan ke dalam sistem berbasis web. Tujuan dari tahap ini adalah untuk menyediakan antarmuka pengguna (user interface) yang intuitif dan mudah digunakan, serta memungkinkan pengguna untuk melihat hasil deteksi serangan lalu lintas.

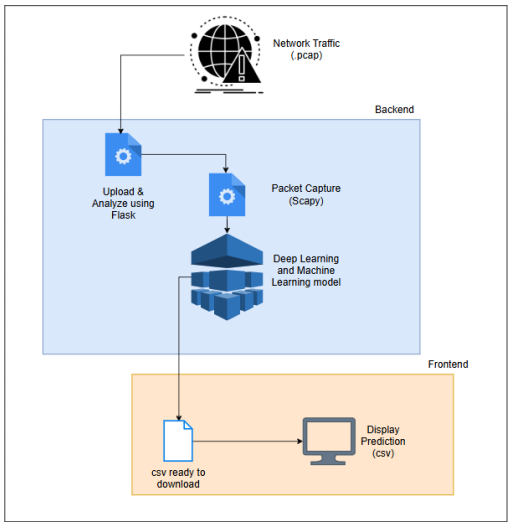
2. Desain

Pembuatan desain web akan dilakukan pada tahapan ini. Desain yang dibuat akan mempermudah user untuk melihat deteksi hasil serangan siber. Gambar 5 menunjukkan tahapan yang akan dilakukan dalam proses pembuatan UI web.



Gambar 5 Desain alur Inkremental II

Gambar 6 menunjukkan arsitektur web yang dimulai dari user meng-input file .pcap sampai kepada tahap menampilkan data pada user dalam bentuk csv.



Gambar 6 Desain arsitektur web

3. Implementasi

Tahap implementasi sistem terdiri dari tiga komponen utama, yaitu antarmuka web, backend menggunakan *Flask*, serta modul pemrosesan file *.pcap* untuk analisis dan prediksi serangan siber. Ketiga komponen ini saling terintegrasi sehingga pengguna dapat mengunggah file, menjalankan analisis, dan mengunduh hasil klasifikasi secara mudah melalui aplikasi berbasis web.

Antarmuka web dirancang sederhana dengan satu input file dan empat tombol utama: *upload*, *analyze*, *clear*, dan *download*. Tombol *download* akan muncul setelah proses analisis selesai. Selain itu, digunakan *JavaScript* untuk mengatur logika interaksi tombol agar proses unggah, analisis, dan unduhan berjalan secara responsif.

Pada sisi backend, Flask digunakan sebagai kerangka kerja untuk menangani permintaan dari pengguna. Proses dimulai ketika pengguna mengunggah file *.pcap*, sistem memverifikasi format file sebelum menyimpannya di server untuk dianalisis. Fungsi analisis kemudian memproses file tersebut dengan memanggil modul yang mengekstrak fitur-fitur penting dari setiap aliran (*flow*) jaringan, seperti IP address, jumlah paket, durasi, dan atribut lainnya.

Pemrosesan file *.pcap* dilakukan melalui serangkaian fungsi yang membaca data jaringan, mengelompokkan paket ke dalam flow berdasarkan kombinasi IP, port, dan protokol, lalu menghitung fitur statistik dari setiap flow. Fitur-fitur ini kemudian disusun dalam bentuk *DataFrame*, dinormalisasi dengan *MinMaxScaler*, dan di-*reshape* agar sesuai dengan input model CNN. CNN digunakan untuk mengekstraksi fitur lanjutan, sementara klasifikasi akhir dilakukan oleh model Random Forest untuk menentukan jenis serangan.

Hasil klasifikasi untuk seluruh flow kemudian digabung menjadi sebuah file CSV yang dapat diunduh pengguna melalui tombol *Download CSV Result*. Pipeline ini memungkinkan analisis dilakukan secara otomatis dan efisien, mulai dari pemrosesan awal file *.pcap*, ekstraksi fitur, prediksi serangan, hingga penyajian hasil dalam format yang siap digunakan untuk evaluasi lebih lanjut.

4. Pengujian

Pengujian web akan menggunakan blackbox testing. Pengujian akan dilakukan terhadap seluruh fitur yang terdapat dalam tampilan user interface

dengan tujuan untuk mengetahui keberhasilan setiap fungsi yang ada.

Tabel 1 Skenario Blackbox Testing

Fungsi yang akan diuji	Data Input	Hasil yang diharapkan
Tombol "choose file" untuk mengunggah file	Data berupa file dengan .pcap extension	User dapat mengunggah file dengan extension .pcap
Tombol Uplaod	-	File .pcap akan disimpan pada server agar dapat diproses oleh machine learning
Tombol Analyze	-	Machine learning menganalisa file .pcap dan mengubah file dari extension .pcap menjadi .csv
Tombol Download	-	Mengunduh file dengan extension .csv

III. HASIL DAN PEMBAHASAN

A. Inkremental I

Tahap Inkremental I menghasilkan model untuk mendeteksi serangan jaringan melalui kombinasi ekstraksi fitur berbasis CNN dan klasifikasi menggunakan algoritma machine learning. Tiga pendekatan diuji, yaitu CNN murni, CNN + Random Forest (dua varian), dan CNN + XGBoost.

1. Evaluasi Model CNN

Model CNN dilatih selama 30 epoch dengan hasil akurasi akhir 87,66%. Pada awal pelatihan, akurasi meningkat cepat, dari 83,54% hingga 87,01% pada epoch ke-5, kemudian stabil hingga akhir pelatihan. CNN mampu mengekstraksi pola trafik jaringan dengan baik, tetapi performanya menurun pada kelas minoritas.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	32000
1	0.89	0.94	0.92	32000
2	0.99	1.00	1.00	32000
3	0.94	0.88	0.91	32000
4	0.80	0.50	0.61	32000
5	0.74	0.92	0.82	50000
accuracy			0.88	210000
macro avg	0.89	0.87	0.88	210000
weighted avg	0.88	0.88	0.87	210000

Gambar 7 Classification Report Model CNN

2. Evaluasi Model CNN + Random Forest

Model ini memberikan hasil terbaik dengan akurasi 90,68%, lebih tinggi 3% dibanding CNN. Kombinasi CNN sebagai feature extractor dengan Random Forest memberikan keunggulan karena RF dapat memanfaatkan representasi fitur dari CNN secara optimal dan mengurangi overfitting. Nilai precision dan recall lebih merata di semua kelas, menunjukkan stabilitas dalam mendeteksi berbagai jenis serangan.

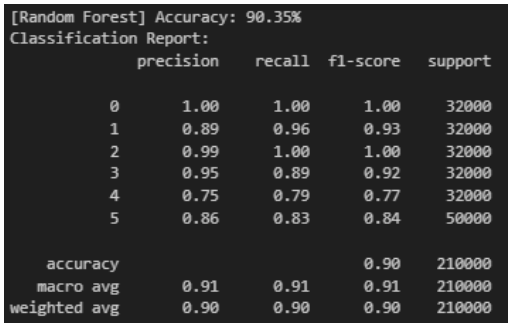
[Random Forest] Accuracy: 90.68%
Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	32000
1	0.89	0.96	0.93	32000
2	0.99	1.00	1.00	32000
3	0.95	0.89	0.92	32000
4	0.76	0.80	0.78	32000
5	0.87	0.84	0.85	50000
accuracy			0.91	210000
macro avg	0.91	0.91	0.91	210000
weighted avg	0.91	0.91	0.91	210000

Gambar 8 Classification Report Model CNN + RF

3. Evaluasi Model CNN + Random Forest (Penelitian Terdahulu)

Pendekatan ini menggunakan arsitektur CNN dengan konfigurasi filter yang berbeda (128, 64, 32) seperti pada penelitian sebelumnya. Akurasi yang diperoleh adalah 90,35%, sedikit lebih rendah dibanding arsitektur yang digunakan. Hal ini menunjukkan bahwa perancangan arsitektur CNN memengaruhi kualitas fitur yang dihasilkan, yang pada akhirnya berdampak pada akurasi klasifikasi.



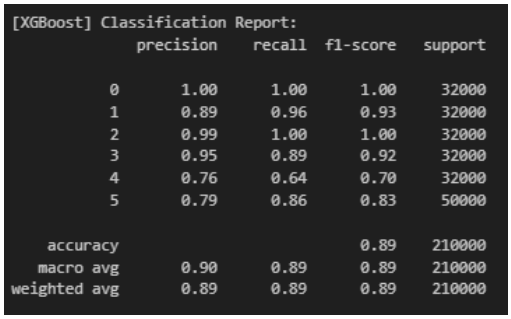
[Random Forest] Accuracy: 90.35%
Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	32000
1	0.89	0.96	0.93	32000
2	0.99	1.00	1.00	32000
3	0.95	0.89	0.92	32000
4	0.75	0.79	0.77	32000
5	0.86	0.83	0.84	50000
accuracy			0.90	210000
macro avg	0.91	0.91	0.91	210000
weighted avg	0.90	0.90	0.90	210000

Gambar 9 Classification Report Model CNN + RF menggunakan arsitektur Penelitian Terdahulu

4. Evaluasi Model CNN + XGBoost

Model ini memperoleh akurasi 89,00%, lebih baik dari CNN murni, tetapi masih di bawah CNN + RF. XGBoost efektif dalam memproses fitur hasil CNN, namun memiliki waktu komputasi lebih tinggi. XGBoost bekerja efektif dalam memproses fitur kompleks, namun memerlukan waktu pelatihan lebih lama dan sedikit menurun performanya pada kelas dengan jumlah data kecil.



[XGBoost] Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	32000
1	0.89	0.96	0.93	32000
2	0.99	1.00	1.00	32000
3	0.95	0.89	0.92	32000
4	0.76	0.64	0.70	32000
5	0.79	0.86	0.83	50000
accuracy			0.89	210000
macro avg	0.90	0.89	0.89	210000
weighted avg	0.89	0.89	0.89	210000

Gambar 10 Classification Report model XGBoost

Tabel 2 Perbandingan Model

Model	**Akurasi (%)**
CNN	87,66
CNN + Random Forest	90,68
CNN + Random Forest (Penelitian Terdahulu)	90,35
CNN + XGBoost	89,00

Dari hasil pengujian, dapat disimpulkan bahwa penggunaan CNN untuk ekstraksi fitur yang dikombinasikan dengan algoritma klasifikasi seperti Random Forest atau XGBoost memberikan hasil yang lebih baik dibanding CNN murni. Model CNN + Random Forest dengan arsitektur usulan memberikan performa terbaik dengan akurasi tertinggi yaitu 90,68%, diikuti oleh CNN + Random Forest pada arsitektur penelitian terdahulu (90,35%), CNN + XGBoost (89,00%), dan CNN murni (87,66%). Hal ini menunjukkan bahwa penggunaan Random Forest setelah ekstraksi fitur CNN lebih optimal dibandingkan dengan XGBoost maupun CNN murni dalam mendeteksi serangan jaringan.

B. Inkremental II

Tahap Inkremental II difokuskan pada implementasi model machine learning untuk mendeteksi serangan siber melalui aplikasi web yang telah dibangun. Aplikasi web ini dirancang agar pengguna dapat mengunggah file berformat *.pcap*, menganalisisnya menggunakan model yang sudah dilatih, dan mengunduh hasil analisis dalam bentuk file *.csv*. Tahap ini mencakup dua aspek utama, yaitu implementasi antarmuka web dan pengujian fungsionalitas.

1. Implementasi

Aplikasi web memiliki antarmuka sederhana dengan satu kolom unggah file (choose file) dan empat tombol utama: Upload, Analyze, Clear, dan Download. Tombol Choose File digunakan untuk memilih file .pcap dari penyimpanan lokal, sedangkan tombol Upload berfungsi mengirim file tersebut ke server. Setelah berhasil diunggah, sistem menampilkan notifikasi bahwa file sudah siap untuk dianalisis.

Proses analisis dilakukan dengan menekan tombol *Analyze*. Pada tahap ini, aplikasi memanggil modul pemrosesan untuk mengekstraksi fitur dari file .pcap, mengubah data menjadi format *.csv*, melakukan normalisasi, kemudian menerapkan model CNN + *Random Forest* untuk melakukan klasifikasi serangan. Jika analisis selesai, sistem menampilkan tombol *Download CSV*

Result yang memungkinkan pengguna mengunduh hasil klasifikasi.

2. Pengujian (Blackbox Testing)

Pengujian dilakukan untuk memastikan setiap fitur aplikasi berjalan sesuai harapan. Hasil pengujian ditunjukkan pada Tabel 4.1. Semua fungsi utama, termasuk pemilihan file, pengunggahan, analisis, dan pengunduhan hasil, berhasil berjalan dengan baik. Selain itu, pengujian juga mencakup skenario kesalahan, seperti mencoba mengunggah tanpa memilih file atau menganalisis tanpa file yang diunggah, di mana sistem memberikan peringatan agar pengguna melakukan langkah yang benar.

Tabel 3 Hasil Blackbox Testing

Fungsi yang akan diuji	***Data Input***	***Hasil yang diharapkan***	***Berhasil/Gagal***
Tombol "choose file" untuk mengunggah file	*Data berupa file dengan .pcap extension*	*User dapat mengunggah file dengan extension .pcap*	*Berhasil*
Tombol Uplaod	*-*	*File .pcap akan disimpan pada server agar dapat diproses oleh machine learning*	*Berhasil*
Tombol Analyze	*-*	*Machine learning menganalisa file .pcap dan mengubah file dari extension .pcap menjadi .csv*	*Berhasil*
Tombol Download	*-*	*Mengunduh file dengan extension .csv*	*Berhasil*

Untuk menguji sistem secara menyeluruh, digunakan file .pcap berukuran 27 MB yang mengandung trafik serangan DoS. Setelah file diunggah dan dianalisis, hasilnya berupa file .csv dengan 29.016 baris data. Label yang dihasilkan menunjukkan dua kategori, yaitu Normal dan DoS attack. Hal ini membuktikan bahwa sistem dapat mengidentifikasi pola serangan dari data lalu lintas jaringan yang kompleks dan mengonversinya ke format yang lebih mudah dianalisis.

IV. SIMPULAN

- Serangan *Infiltration* mendapatkan jumlah *false positive* dan *false negative* paling banyak yaitu FP 8,084 dan 6,400, hal ini disebabkan karena uniknya serangan infiltration yang mempunyai pola serangan yang berasal dari dalam ruang lingkup jaringan.
- Model yang dibangun menggunakan CNN untuk ekstraksi fitur dan Random Forest untuk klasifikasi dengan pelatihan 30 *epoch* menghasilkan rata-rata akurasi 91%, dengan *precision*, *recall*, dan F1-score juga sebesar 91%.
- Seluruh tombol dan fitur pada antarmuka pengguna (*UI website*) dapat berfungsi dengan baik sesuai perancangan.
- Proses analisis pada website memerlukan waktu cukup lama ketika menekan tombol *Analyze*. Hal ini disebabkan oleh besarnya jumlah paket jaringan yang dianalisis oleh model serta proses ekstraksi fitur dan konversi data ke format Excel yang memakan waktu.

DAFTAR RUJUKAN

[1] Check Point Team, "Check Point Research Reports Highest Increase of Global Cyber Attacks seen in last two years – a 30% Increase in Q2 2024 Global Cyber Attacks," https://blog.checkpoint.com/research/check-point-research-reports-highest-increase-of-

global-cyber-attacks-seen-in-last-two-years-a-30-increase-in-q2-2024-global-cyber-attacks/.

[2] Purwono, A. Ma'arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. U. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, 2022, doi: 10.31763/ijrcs.v2i4.888.

[3] M. A. Abdurrazzaq, "Sneakers scoring using image regression with transfer learning," *AIP Conf Proc*, vol. 2987, no. 1, p. 020028, Apr. 2024, doi: 10.1063/5.0199373.

[4] A. Oyetoro, J. Mart, and U. Amah, "Using Machine Learning Techniques Random Forest and Neural Network to Detect Cyber Attacks ," Apr. 10, 2023. doi: 10.14293/PR2199.000059.v1.