JURNAL LOCUS: Penelitian & Pengabdian
Volume 5 No. 8 Agustus 2026
E-ISSN 2829-7334| P-ISSN 2829-5439
Hompage: https://locus.rivierapublishing.id/index.php/jl

Perancangan Aplikasi Simulasi Deteksi Serangan DDoS Berbasis Dataset Trafik Menggunakan Metode *Random Forest* yang Dioptimasi *Grey Wolf Optimizer* untuk Meningkatkan Akurasi

Lukas Febrian Laufra* , Somantri, Lusiana Sani Parwati
Universitas Nusa Putra, Indonesia
Email: lukas.febrian_ti22@nusaputra.ac.id* , somantri@nusaputra.ac.id, lusiana.sani@nusaputra.ac.id

Keywords:
Distributed Denial of Service (DDoS); Internet of Things (IoT); Random Forest; Grey Wolf Optimizer; Detection Simulation

Abstract
Distributed Denial of Service (DDoS) attacks remain one of the most critical cybersecurity threats in Internet of Things (IoT) environments due to the rapid growth of interconnected devices and the increasing complexity of network traffic. These conditions require detection mechanisms that are not only accurate but also computationally efficient and suitable for evaluation prior to deployment in real-world environments. This study aimed to develop a traffic dataset-based DDoS detection simulation application by integrating the Random Forest algorithm with Grey Wolf Optimizer (GWO) for hyperparameter optimization. The research employed an engineering approach combining descriptive qualitative and quantitative experimental methods using the Waterfall software development model. The CICIIOT2025 dataset was utilized for model training and testing after preprocessing, while model performance was evaluated using accuracy, precision, recall, and F1-score metrics together with functional testing of the application. The results demonstrated that GWO optimization produced a model with a modest improvement in classification accuracy while reducing model complexity, resulting in a more efficient architecture without sacrificing predictive performance. Functional testing further confirmed that all application features operated according to the specified system requirements and successfully visualized the DDoS detection process through an interactive simulation. Therefore, the proposed application provides a practical and systematic platform for evaluating and simulating DDoS detection models in IoT networks before deployment in real operational environments.

Kata Kunci:
Distributed Denial of Service (DDoS); Internet of Things (IoT); Random Forest; Grey Wolf Optimizer; Simulasi Deteksi.

Abstrak
Serangan *Distributed Denial of Service* (DDoS) merupakan salah satu ancaman keamanan siber yang paling signifikan pada lingkungan *Internet of Things* (IoT) akibat meningkatnya jumlah perangkat yang saling terhubung dan kompleksitas lalu lintas jaringan. Kondisi tersebut menuntut tersedianya mekanisme deteksi yang tidak hanya memiliki tingkat akurasi yang baik, tetapi juga efisien dan mudah diimplementasikan sebagai media evaluasi sebelum diterapkan pada lingkungan nyata. Penelitian ini bertujuan mengembangkan aplikasi simulasi deteksi serangan DDoS berbasis dataset trafik dengan mengintegrasikan algoritma Random Forest dan optimasi *Grey Wolf Optimizer* (GWO). Penelitian menggunakan pendekatan rekayasa dengan metode deskriptif-kualitatif dan eksperimen kuantitatif melalui model pengembangan *Waterfall*. Dataset CICIIOT2025 digunakan sebagai data pelatihan dan pengujian setelah melalui tahapan *preprocessing*, sedangkan evaluasi dilakukan menggunakan metrik akurasi, presisi, *recall*, dan *F1-score*, serta pengujian fungsional aplikasi. Hasil penelitian menunjukkan

bahwa optimasi GWO menghasilkan model dengan akurasi sedikit lebih tinggi dibandingkan model *Random Forest* dasar, sekaligus membentuk arsitektur model yang lebih efisien melalui pengurangan kompleksitas tanpa menurunkan kinerja klasifikasi. Seluruh fitur aplikasi juga berfungsi sesuai kebutuhan sistem dan mampu memvisualisasikan proses deteksi secara interaktif. Dengan demikian, aplikasi yang dikembangkan berpotensi menjadi media pendukung evaluasi dan simulasi deteksi serangan DDoS pada jaringan IoT secara lebih sistematis dan efisien sebelum implementasi pada lingkungan operasional.

PENDAHULUAN

Perkembangan teknologi *Internet of Things* (IoT) telah meningkatkan jumlah perangkat yang saling terhubung melalui jaringan internet dalam berbagai bidang, seperti industri, kesehatan, transportasi, pendidikan, dan rumah pintar (Abiramasundari & Ramaswamy, 2024). Konektivitas yang semakin luas memberikan banyak manfaat dalam pertukaran data dan otomatisasi sistem, tetapi juga diikuti oleh meningkatnya risiko keamanan siber akibat banyaknya titik akses yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab (Prayogi et al., 2024). Salah satu ancaman yang paling sering terjadi pada lingkungan IoT adalah serangan *Distributed Denial of Service* (DDoS) (Hoque et al., 2024), yaitu serangan yang membanjiri target dengan trafik dalam jumlah besar dari banyak sumber secara bersamaan sehingga layanan menjadi lambat, terganggu, bahkan tidak dapat diakses (Kushwaha et al., 2024). Oleh karena itu, diperlukan mekanisme deteksi yang mampu mengidentifikasi serangan DDoS secara cepat dan akurat pada lingkungan IoT.

Permasalahan utama dalam deteksi serangan DDoS pada jaringan IoT terletak pada karakteristik trafik yang bersifat dinamis dan terus berubah, sehingga metode deteksi berbasis aturan (rule-based detection) menjadi kurang efektif dalam menghadapi variasi serangan baru (Al-Shibly et al., 2024). Perangkat IoT juga umumnya memiliki keterbatasan sumber daya komputasi sehingga membutuhkan metode deteksi yang efisien namun tetap akurat (Mualfah et al., 2025). Pemanfaatan metode machine learning menjadi salah satu pendekatan yang banyak digunakan untuk meningkatkan kemampuan deteksi serangan. Penelitian terdahulu menunjukkan bahwa algoritma Random Forest memiliki kemampuan yang baik dalam mendeteksi serangan DDoS karena mampu menangani data berdimensi tinggi serta menghasilkan performa klasifikasi yang stabil (Rachmatullah et al., 2026; Wiratama et al., 2026). Meskipun demikian, performa *Random Forest* sangat dipengaruhi oleh pemilihan parameter model, sehingga parameter yang kurang optimal dapat menyebabkan penurunan akurasi dan kemampuan generalisasi terhadap data baru.

Untuk mengatasi permasalahan tersebut, beberapa penelitian mulai menerapkan pendekatan optimasi berbasis meta-heuristic. *Grey Wolf Optimizer* (GWO) terbukti mampu meningkatkan kualitas proses optimasi pada sistem deteksi serangan jaringan dengan mekanisme pencarian solusi yang sederhana namun efektif (Billah et al., 2026; Sawah et al., 2025), dan efektif digunakan untuk meningkatkan performa klasifikasi pada sistem deteksi intrusi dibandingkan metode optimasi konvensional (Jha & Saini, 2026). Berdasarkan studi literatur, sebagian besar penelitian sebelumnya masih berfokus pada evaluasi model menggunakan dataset statis dan belum banyak mengimplementasikan hasil deteksi ke dalam aplikasi yang mampu mensimulasikan proses klasifikasi secara interaktif dan mudah dipahami

oleh pengguna (Dembele et al., 2024). Penelitian yang mengombinasikan *Random Forest* dan *Grey Wolf Optimizer* pada lingkungan IoT menggunakan dataset CICIIOT2025 juga masih relatif terbatas (Abdulrazzaq & Toker, 2026). Kesenjangan inilah yang menjadi dasar dilaksanakannya penelitian ini.

Penelitian ini bertujuan: (1) merancang aplikasi simulasi deteksi serangan DDoS berbasis dataset trafik menggunakan Python Kivy yang mampu memvisualisasikan proses klasifikasi secara interaktif; (2) mengimplementasikan metode *Random Forest* untuk mendeteksi serangan DDoS menggunakan dataset CICIIOT2025 sebagai data referensi pelatihan model; (3) menerapkan metode *Grey Wolf Optimizer* (GWO) untuk mengoptimasi parameter Random Forest dalam meningkatkan kualitas proses deteksi; dan (4) mengevaluasi kinerja aplikasi berdasarkan fungsi sistem dan hasil klasifikasi terhadap data simulasi dari dataset trafik.

Beberapa penelitian terdahulu telah mengembangkan metode deteksi serangan DDoS menggunakan pendekatan machine learning maupun optimasi metaheuristic, sebagaimana dirangkum pada Tabel 1.

Tabel 1. Ringkasan Penelitian Terkait

No	Peneliti (Tahun)	Metode	Hasil	Celah Penelitian (Gap)
1	Sugeng & Adhliana (2026)	*Random Forest* + PSO	Optimasi meningkatkan performa deteksi DDoS pada jaringan SDN.	Mengimplementasikan optimasi GWO untuk stabilitas pencarian parameter pada ekosistem IoT.
2	Vasin & Kakabian (2025)	ANFIS	ANFIS layak untuk deteksi DDoS pada dataset CIC-DDoS-2019.	Menggunakan dataset CICIIOT2025 yang lebih mutakhir dan mengintegrasikan GWO.
3	Chu et al. (2022)	*Random Forest*	Model RF dapat digunakan untuk mendeteksi serangan DDoS.	Membangun sistem deteksi interaktif melalui aplikasi simulasi berbasis Python Kivy.
4	Rui et al. (2025)	SVM dan *Random Forest*	RF memiliki performa kompetitif untuk deteksi DDoS.	Memaksimalkan stabilitas model dengan pencarian hyperparameter otomatis menggunakan GWO.
5	Dembele et al. (2024)	GWO dan SVM	GWO meningkatkan deteksi DDoS melalui optimasi fitur.	Menggunakan Random Forest sebagai algoritma utama untuk klasifikasi trafik IoT berdimensi tinggi.

Berdasarkan penelitian terdahulu, *Random Forest* memiliki performa yang baik dalam deteksi serangan DDoS, sedangkan *Grey Wolf Optimizer* mampu meningkatkan kualitas optimasi pada model klasifikasi. Penelitian ini mengembangkan hal tersebut dengan merancang aplikasi simulasi deteksi DDoS berbasis dataset trafik menggunakan kombinasi *Random Forest* dan GWO pada dataset CICIIOT2025, serta mengimplementasikannya ke dalam antarmuka interaktif berbasis Python Kivy untuk memvisualisasikan proses klasifikasi secara dinamis.

METODE PENELITIAN

Penelitian ini merupakan penelitian rekayasa (*engineering research*) dengan pendekatan kualitatif deskriptif dan kuantitatif eksperimental. Pendekatan kualitatif deskriptif

digunakan untuk mendeskripsikan proses pengembangan aplikasi, kebutuhan pengguna, dan implementasi algoritma, sedangkan pendekatan kuantitatif eksperimental digunakan untuk mengukur kinerja model klasifikasi secara objektif menggunakan metrik akurasi, presisi, recall, dan F1-score. Data diperoleh melalui observasi, wawancara, studi pustaka, dan hasil implementasi sistem. Metode pengembangan perangkat lunak yang digunakan adalah model Waterfall, meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan.

Dataset Penelitian

Dataset yang digunakan adalah CICIIOT2025 yang dikembangkan oleh *Canadian Institute for Cybersecurity, University of New Brunswick*. Dataset ini diposisikan sebagai data referensi dan data pelatihan (*training*) model, bukan sebagai sumber pengujian langsung berupa penangkapan trafik jaringan secara nyata (*live traffic capture*) (Chu et al., 2022; Rui et al., 2025; Sugeng & Adhliana, 2026; Vasin & Kakabian, 2025). Sebelum digunakan dalam pelatihan model, dataset melalui tahap *preprocessing* yang meliputi pembersihan data, penghapusan nilai kosong, transformasi data kategorikal ke bentuk numerik (*label encoding*), normalisasi *Min-Max Scaling*, dan pembagian data menjadi data latih dan data uji menggunakan *stratified train-test split* dengan rasio 80:20.

Dataset awal terdiri dari 20 berkas CSV (10 kategori serangan dan 10 kategori benign) dengan total 685.671 baris data dan 93 fitur, tanpa nilai kosong maupun duplikat (0,00% data dihapus). Sebanyak 71 fitur numerik dinormalisasi menggunakan *Min-Max Scaling* ke rentang [0,1] dengan rumus pada Persamaan (1), untuk mencegah dominasi fitur dengan rentang nilai besar terhadap proses pemisahan simpul (splitting) pada pohon keputusan.

$$X\_norm = (X - X\_min) / (X\_max - X\_min) \quad (1)$$

Rasio kelas dipertahankan sebesar 58,43% Benign dan 41,57% Attack pada data latih maupun data uji melalui stratified split, sehingga model terhindar dari bias akibat ketidakseimbangan kelas (class imbalance). Hasil akhir preprocessing menghasilkan 548.536 sampel data latih dan 137.135 sampel data uji, masing-masing dengan 93 fitur numerik.

Model Klasifikasi Random Forest

Random Forest merupakan algoritma ensemble learning yang membangun banyak decision tree melalui metode bagging (*Bootstrap Aggregating*), di mana setiap pohon dilatih menggunakan sampel acak dari data latih dan subset fitur yang berbeda untuk mengurangi risiko *overfitting* (Gunawan et al., 2026; Impron, 2025; Riansah et al., 2025; Rozam et al., 2026; Suryadewiansyah & Tju, 2022). Keputusan akhir klasifikasi diperoleh melalui mekanisme majority voting sebagaimana dituliskan pada Persamaan (2).

$$\hat{y} = \text{mode}(h_1(x), h_2(x), h_3(x), \ldots, h_n(x)) \quad \ldots(2)$$

dengan $\hat{y}$ adalah hasil prediksi akhir, $h_n(x)$ adalah hasil prediksi dari pohon ke-n, dan mode adalah kelas yang paling sering muncul. Kriteria pemisahan data pada setiap decision tree menggunakan Gini Impurity sebagaimana Persamaan (3), dengan p_i adalah probabilitas kelas ke-i dan c adalah jumlah kelas.

$$Gini = 1 - \Sigma_{i=1}^{c} p_i^2 \quad \ldots(3)$$

Model acuan (*baseline*) dibangun menggunakan parameter *default Scikit-Learn* untuk menetapkan standar performa awal sebelum optimasi, dengan konfigurasi sebagaimana disajikan pada Tabel 2.

Tabel 2. Konfigurasi Parameter Model Random Forest Baseline

No	Hyperparameter	Nilai	Fungsi
1	max_depth	15	Membatasi kedalaman maksimum pohon untuk mengontrol kompleksitas model.
2	min_samples_leaf	2	Minimal sampel pada leaf node agar model tidak sensitif terhadap noise.
3	min_samples_split	5	Minimal sampel pada internal node agar simpul dapat dipecah kembali.
4	n_estimators	100	Jumlah pohon keputusan pada ensemble.
5	random_state	42	Mengunci pembangkit bilangan acak agar hasil dapat direproduksi.

Optimasi *Grey Wolf Optimizer* (GWO)

Grey Wolf Optimizer merupakan algoritma optimasi metaheuristic yang meniru hierarki sosial dan perilaku berburu serigala abu-abu, yaitu alpha, beta, delta, dan omega (Fitriani et al., 2025; Junian et al., 2025; Ristyawan et al., 2025; Wolf & Optimizer, n.d.). Posisi alpha merepresentasikan solusi terbaik, beta solusi terbaik kedua, dan delta solusi terbaik ketiga, sedangkan agen omega bergerak mengikuti ketiga posisi terbaik tersebut untuk mencari solusi optimum. Jarak antara agen serigala dan mangsa (solusi terbaik) dihitung menggunakan Persamaan (4)–(5), dan posisi diperbarui menggunakan Persamaan (6).

$$A = 2a \cdot r_1 - a \; ; \; C = 2r_2 \quad \ldots(4)$$

$$D = |C \cdot X_p(t) - X(t)| \quad \ldots(5)$$

$$X(t+1) = X_p(t) - A \cdot D \quad \ldots(6)$$

dengan a adalah parameter yang menurun linier dari 2 ke 0, r_1 dan r_2 adalah bilangan acak antara 0–1, $X_p(t)$ adalah posisi solusi terbaik, dan X(t) adalah posisi agen saat ini. Posisi baru setiap agen ditentukan berdasarkan rata-rata pengaruh tiga solusi terbaik (alpha, beta, delta) sebagaimana Persamaan (7).

$$X(t+1) = (X_1 + X_2 + X_3) / 3 \quad \ldots(7)$$

Dalam penelitian ini, GWO digunakan untuk mencari kombinasi optimal hyperparameter Random Forest, yaitu n_estimators dan max_depth, dengan fungsi fitness berupa (1 − accuracy) sehingga semakin kecil nilai fitness, semakin baik solusi yang diperoleh. Konfigurasi eksperimen GWO disajikan pada Tabel 3.

Tabel 3. Konfigurasi Eksperimen Optimasi GWO

Komponen	Baseline RF	GWO-RF (Rentang Pencarian)
n_estimators	100 (tetap)	[50–200]
max_depth	15 (tetap)	[5–50]
min_samples_split	5 (tetap)	[2–10]
min_samples_leaf	2 (tetap)	[1–5]
Jumlah agen (wolves)	-	30
Maksimum iterasi	-	50
Fungsi fitness	-	1 − Accuracy

Mekanisme integrasi GWO-RF dilakukan melalui tahapan berikut: (1) menentukan parameter Random Forest yang akan dioptimasi; (2) membangkitkan populasi awal solusi GWO; (3) melatih Random Forest untuk setiap kombinasi parameter; (4) menghitung nilai fitness setiap solusi; (5) menentukan posisi alpha, beta, dan delta; (6) memperbarui posisi solusi menggunakan persamaan GWO; (7) mengulang proses hingga iterasi maksimum tercapai; dan (8) menggunakan parameter terbaik untuk membangun model akhir GWO-RF. Model final kemudian disimpan dalam format. pkl (gwo_rf_model.pkl) beserta objek scaler (scaler.pkl) agar dapat dimuat secara instan oleh aplikasi tanpa perlu pelatihan ulang.

Perancangan Aplikasi Simulasi

Aplikasi simulasi dikembangkan menggunakan Python dengan framework Kivy, yang dipilih karena kemampuannya membangun antarmuka grafis yang responsif dan lintas platform. Aplikasi dilengkapi dashboard simulasi untuk memuat dataset, menjalankan simulasi deteksi melalui mekanisme pembacaan data baris demi baris (*row-by-row replay*), menampilkan hasil klasifikasi beserta tingkat kepercayaan (*confidence*), menampilkan metrik evaluasi model, serta mencatat dan mengekspor log deteksi ke format CSV. Pendekatan simulasi ini dipilih karena sebelum sebuah model diterapkan pada jaringan fisik, diperlukan tahap pembuktian konsep (*proof of concept*) yang kuat menggunakan data historis yang representatif (Erlangga et al., n.d.; Houichi et al., 2025; Isarianto et al., 2025). Pengujian sistem meliputi pengujian fungsional terhadap seluruh fitur aplikasi dan pengujian kinerja model menggunakan metrik akurasi, presisi, recall, dan F1-score pada data uji.

HASIL DAN PEMBAHASAN

Hasil Evaluasi Model Random Forest Baseline

Model Random Forest baseline dilatih menggunakan 548.536 sampel data latih dengan 93 fitur numerik hasil normalisasi Min-Max Scaling. Proses pelatihan diselesaikan dalam waktu 31,23 detik. Evaluasi dilakukan pada 137.135 sampel data uji, terdiri dari 80.135 sampel trafik normal (Benign) dan 57.000 sampel trafik serangan (Attack). Hasil evaluasi disajikan pada Tabel 4.

Tabel 4. Hasil Evaluasi Model Random Forest Baseline

No	Metrik Evaluasi	Nilai
1	Akurasi (*Accuracy*)	95,69%
2	Presisi (*Precision*)	95,97%
3	Recall (*Sensitivitas*)	95,69%
4	F1-Score	95,65%
5	Waktu Pelatihan (Training Time)	31,23 detik

Model baseline menghasilkan akurasi sebesar 95,69%, dengan presisi 95,97% yang menunjukkan bahwa sebagian besar prediksi serangan memang benar-benar serangan. Analisis classification report per kelas menunjukkan bahwa model mencapai recall 100% pada kelas Benign (seluruh trafik normal terdeteksi dengan benar, false negative = 0) dan presisi 100% pada kelas Attack (tidak ada trafik normal yang salah diklasifikasikan sebagai serangan, false positive = 0). Namun, recall pada kelas Attack hanya sebesar 90%, yang berarti sekitar 5.700 dari 57.000 sampel serangan (10%) tidak terdeteksi oleh model. Dalam konteks keamanan jaringan IoT, false negative yang tinggi tergolong kritis karena serangan yang lolos dari deteksi

dapat menyebabkan gangguan layanan yang signifikan. Kondisi ini mengindikasikan bahwa parameter default Scikit-Learn belum sepenuhnya optimal terhadap karakteristik dataset CICIIOT2025, sehingga diperlukan proses optimasi hyperparameter lebih lanjut.

Hasil Optimasi Grey Wolf Optimizer

Proses optimasi GWO dijalankan dengan 30 agen serigala selama 50 iterasi maksimum menggunakan akurasi sebagai fungsi fitness. Hasil observasi menunjukkan konvergensi yang cepat: pada iterasi pertama, akurasi subset pencarian telah melonjak dari 96,11% menjadi 96,15% dan bertahan stabil hingga iterasi terakhir, sementara nilai parameter (seperti n_estimators antara 85–100) terus dieksplorasi pada area ruang pencarian dengan performa setara. Agen alpha akhirnya mengunci kombinasi parameter terbaik berupa n_estimators = 88 dan max_depth = 17, sebagaimana dirangkum pada Tabel 5.

Tabel 5. Parameter Terbaik Hasil Optimasi GWO

Parameter	Nilai GWO	Nilai Default (Baseline)	Analisis Perubahan
n_estimators	88	100	Model 12% lebih ringkas (lebih sedikit pohon)
max_depth	17	Tidak terbatas (None)	Kedalaman dibatasi untuk mencegah overfitting

Temuan ini menunjukkan bahwa GWO berhasil menemukan bahwa model tidak memerlukan 100 pohon keputusan (nilai default) untuk mencapai performa maksimal; dengan hanya 88 pohon, model mampu mencapai akurasi yang setara bahkan cenderung lebih stabil. Pembatasan max_depth menjadi 17 juga menunjukkan adanya mekanisme regularisasi alami yang mencegah pohon tumbuh berlebihan sehingga mengurangi risiko overfitting.

Model GWO-RF final kemudian dibangun ulang menggunakan seluruh data latih dengan parameter optimal tersebut, dan diuji pada 137.135 sampel data uji yang sama dengan model baseline. Perbandingan performa kedua model disajikan pada Tabel 6.

Tabel 6. Perbandingan Performa Random Forest Baseline vs GWO-RF

Metrik	RF Baseline	GWO-RF	Delta
Akurasi	95,69%	95,82%	+0,13%
Recall (Attack)	90%	90%	0,00%
Presisi (Attack)	100%	100%	0,00%
Jumlah Pohon	100	88	-12 (lebih efisien)
Kedalaman Maksimum	Tidak terbatas	17	Lebih terkendali

Model GWO-RF berhasil meningkatkan akurasi dari 95,69% menjadi 95,82% (+0,13%). Meskipun peningkatan tersebut terlihat moderat, kenaikan akurasi tetap bermakna pada dataset berskala besar (137.135 sampel uji) karena mencerminkan pengurangan kesalahan klasifikasi secara absolut. Keunggulan utama GWO-RF terletak pada efisiensi arsitektur: dengan hanya 88 pohon (12% lebih sedikit dari baseline) dan kedalaman dibatasi hingga 17 tingkat, model membutuhkan sumber daya komputasi inferensi yang lebih ringan tanpa mengorbankan akurasi — sebuah karakteristik yang penting untuk implementasi pada lingkungan dengan keterbatasan sumber daya seperti perangkat IoT. Kedua model juga menunjukkan karakteristik zero false alarm yang konsisten, yaitu presisi Attack 100% (tidak

ada trafik normal yang salah dikategorikan sebagai serangan) dan *recall Benign* 100% (seluruh trafik normal dikenali dengan benar), sementara *recall Attack* tetap berada pada 90%, yang menunjukkan model bersifat konservatif dan mengutamakan keandalan (menghindari false alarm) dibandingkan agresivitas deteksi. Hasil ini selaras dengan temuan (Jha & Saini, 2026) dan (Dembele et al., 2024) yang menyatakan bahwa GWO efektif digunakan untuk optimasi parameter pada sistem deteksi intrusi berbasis ensemble learning.

Hasil Pengujian Sistem Aplikasi

Pengujian fungsional dilakukan terhadap seluruh fitur aplikasi, meliputi dashboard monitoring, pelatihan model GWO-RF, simulasi deteksi, detection logs dengan filter tanggal, proses optimasi GWO, evaluasi model, dan ekspor log ke format CSV. Seluruh pengujian menunjukkan status berhasil: dashboard mampu menampilkan informasi trafik dan status sistem secara interaktif; menu simulasi deteksi berhasil menjalankan mekanisme *row-by-row replay* dan menampilkan hasil klasifikasi beserta nilai *confidence*; menu detection logs berhasil menyimpan, menampilkan, dan memfilter riwayat deteksi berdasarkan rentang tanggal, serta mengekspornya ke format CSV untuk analisis lanjutan menggunakan aplikasi spreadsheet. Hasil ini menunjukkan bahwa seluruh kebutuhan fungsional yang dirancang pada tahap analisis kebutuhan telah terpenuhi, sehingga aplikasi layak digunakan sebagai alat evaluasi dan simulasi deteksi serangan DDoS berbasis dataset.

Secara keseluruhan, penelitian ini menjawab keempat rumusan masalah yang diajukan. Pertama, aplikasi simulasi deteksi DDoS berbasis Python Kivy berhasil dirancang dan mampu memvisualisasikan proses klasifikasi secara interaktif melalui mekanisme *row-by-row replay*. Kedua, metode *Random Forest* berhasil diimplementasikan sebagai model klasifikasi utama dengan memanfaatkan dataset CICIIOT2025 sebagai data referensi pelatihan, menghasilkan akurasi baseline 95,69%. Ketiga, penerapan *Grey Wolf Optimizer* berhasil mengotomatisasi pencarian hyperparameter dan menghasilkan model yang lebih ringkas, lebih terstruktur, serta sedikit lebih akurat dibandingkan baseline, tanpa memerlukan proses coba-coba manual. Keempat, hasil pengujian fungsional menunjukkan bahwa seluruh fitur aplikasi berjalan sesuai dengan kebutuhan sistem yang dirancang.

Dibandingkan dengan penelitian sebelumnya yang sebagian besar hanya berfokus pada evaluasi model menggunakan dataset statis tanpa visualisasi, penelitian ini memberikan nilai tambah melalui integrasi model GWO-RF ke dalam aplikasi simulasi interaktif berbasis Kivy yang dapat digunakan sebagai proof of concept sebelum penerapan pada jaringan IoT fisik yang sesungguhnya. Meskipun demikian, penelitian ini memiliki beberapa keterbatasan, yaitu: (1) pengujian hanya dilakukan pada satu dataset (CICIIOT2025); (2) jenis serangan yang dideteksi masih terbatas pada DDoS secara umum tanpa klasifikasi multi-kelas terhadap varian serangan tertentu; dan (3) pengujian dilakukan dalam bentuk simulasi replay dataset, bukan pada trafik jaringan IoT fisik secara langsung (*live capture*). Kelemahan pada recall kelas Attack (90%) juga mengindikasikan bahwa penelitian lanjutan diperlukan untuk lebih meningkatkan sensitivitas model terhadap serangan yang menyerupai pola trafik normal.

KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan aplikasi simulasi deteksi serangan *Distributed Denial of Service* (DDoS) berbasis dataset trafik menggunakan Python Kivy, yang mampu memvisualisasikan proses klasifikasi secara interaktif melalui mekanisme row-by-row replay dataset CICIIOT2025. Model Random Forest baseline yang dilatih pada 548.536 sampel data menghasilkan akurasi 95,69%, presisi 95,97%, recall 95,69%, dan F1-score 95,65% pada 137.135 sampel data uji. Optimasi menggunakan *Grey Wolf Optimizer* berhasil menemukan konfigurasi parameter yang lebih efisien (n_estimators=88, max_depth=17) dan meningkatkan akurasi menjadi 95,82%, sekaligus menghasilkan arsitektur model yang 12% lebih ringkas dibandingkan baseline tanpa mengorbankan performa klasifikasi. Hasil pengujian fungsional menunjukkan seluruh fitur aplikasi — dashboard monitoring, pelatihan model, simulasi deteksi, *detection logs*, dan ekspor CSV — berjalan sesuai dengan kebutuhan sistem yang dirancang. Dengan demikian, kombinasi *Random Forest* dan *Grey Wolf Optimizer* terbukti mampu menghasilkan sistem simulasi deteksi DDoS yang lebih efisien dan dapat menjadi alternatif solusi untuk mendukung evaluasi keamanan jaringan IoT sebelum penerapan pada perangkat fisik. Penelitian selanjutnya disarankan untuk memperluas cakupan deteksi pada berbagai jenis serangan siber lain (*Brute Force, Port Scanning, Botnet, Spoofing*), mengintegrasikan mekanisme mitigasi otomatis, menguji model pada dataset dan lingkungan jaringan IoT yang lebih beragam, serta membandingkan GWO dengan metode optimasi metaheuristic lain seperti *Particle Swarm Optimization* (PSO), *Genetic Algorithm* (GA), atau *Whale Optimization Algorithm* (WOA).

REFERENSI

Abdulrazzaq, M. A., & Toker, S. (2026). Quantifying the Real-Time Gap: A Comparative Study of LLMs vs. Random Forest on the DataSense: CIC-IIoT-2025 Dataset. *2026 5th International Informatics and Software Engineering Conference (IISEC)*, 456–461.

Abiramasundari, S., & Ramaswamy, V. (2024). *Distributed Denial-of-Service Attacks in IoT Networks: A Review*.

Al-Shibly, I., Burgas, L., & Massana, J. (2024). *Interpretable Intrusion Detection Systems*.

Billah, M. B., Idhom, M., & Maulana, H. (2026). Deteksi Serangan DDoS pada Trafik IoT Menggunakan Random Forest dengan Dataset CICIoT2023. *Progresif: Jurnal Ilmiah Komputer*, *22*(2), 388–399.

Chu, T. S., Si, W., Simoff, S., & Nguyen, Q. V. (2022). A Machine Learning Classification Model Using Random Forest for Detecting DDoS Attacks. *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–7.

Dembele, A., Mwangi, E., Ronoh, K. K., & Ataro, E. O. (2024). *A Novel Approach for Detection of DDoS Attacks in Software-Defined Networks Based on Grey Wolf Optimizer and Support Vector Machine*.

Erlangga, R., Pradifta, R. D., & Putra, A. S. (n.d.). *Penerapan Metode Pemrograman Berorientasi Objek pada Bahasa Pemrograman Python dalam Merancang Aplikasi Praktik Mandiri Bidan Rozi Silvana*.

Fitriani, S., Budiman, E., Fadli, M., Surono, M., & Sulistiani, H. (2025). Optimalisasi Metode Random Forest Menggunakan Particle Swarm Optimization dalam Prediksi Prestasi Mahasiswa. *Prosiding Seminar Nasional Teknologi Komputer Dan Sains*, 406–415.

Gunawan, T., Rivaldi, M. F., Santika, S. F., Sakti, M. H. P., & Permana, Z. S. (2026). Mengatasi Serangan DDoS SYN Flood dengan Metode Rate Filtering untuk Peningkatan Keamanan Jaringan Server. *Prosiding Seminar Nasional Teknologi Informasi,*

Mekatronika, Dan Ilmu Komputer, 9–16.
Hoque, S., Aydeger, A., Zeydan, E., & Liyanage, M. (2024). *A Survey on DDoS Attacks in IoT Networks*.
Houichi, M., Jaidi, F., & Bouhoula, A. (2025). Enhancing Smart City Security: An Intrusion Detection System Using Machine Learning Methods with the UNB CIC IoT 2023 Dataset. *IET Smart Cities*, *7*(1), e70014.
Impron, A. (2025). Bab 5 IP Sebagai Lapisan Jaringan pada IoT. In *Dasar-Dasar dan Aplikasi Internet of Things (IoT)* (p. 74).
Isarianto, I., Zy, A. T., Maulana, D., & Susilo, A. (2025). Analisis Efektivitas Sistem Deteksi Intrusi terhadap Serangan DDoS: Investigasi Berbasis Simulasi. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(4), 6983–6987.
Jha, R. S., & Saini, H. (2026). Meta-Heuristic Intrusion Detection: Grey Wolf Optimization (GWO) for Attack Classification in Network Traffic. *OPSEARCH*, 1–47.
Junian, W. E., Adilla, R. P., & Irawati, S. M. (2025). Implementasi Algoritma Grey Wolf Optimizer (GWO) untuk Pemodelan Inversi 1D Data Magnetotellurik. *Jurnal Geosaintek*, *11*(2), 197–206.
Kushwaha, V. K., Verma, D. K., Yadav, S. P., & Gupta, H. (2024). *DDoS Attack Detection and Mitigation Techniques*.
Mualfah, D., Ardiansyah, R., & Gunawan, R. (2025). Classification of DDoS Attacks Using the Random Forest Method and Class Weight Technique on the CICDDoS2019 Dataset. *Jurnal CoSciTech (Computer Science and Information Technology)*, *6*(3), 530–535.
Prayogi, A., Pane, M. A. S., Dian, R., Siregar, R. M., Sugianto, R. A., & Simbolon, H. F. S. (2024). Penggunaan Random Forest dan Analisis Perilaku untuk Prediksi Serangan DDoS dalam Lingkungan Cloud Computing. *Techno.Com*, *23*(3).
Rachmatullah, M. A., Firdaus, A. S., Ekayana, N. A., Bahri, F. A. S., & Zauzi, M. (2026). Analisis Efektivitas Intrusion Detection System (IDS) untuk Serangan DDoS Menggunakan Komparasi Random Forest dan Decision Tree. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *10*(1), 1536–1541.
Riansah, A., Nurdiawan, O., & Herdiana, R. (2025). Penerapan Algoritma Random Forest dan Decision Tree untuk Meningkatkan Akurasi Klasifikasi Penjualan pada Toko Bangunan. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(3), 4242–4249.
Ristyawan, A., Nugroho, A., & Amarya, T. K. (2025). Optimasi Preprocessing Model Random Forest untuk Prediksi Stroke. *JATISI*, *12*(1).
Rozam, N. F., Sari, T. N., Yudianto, M. R. A., & Rahman, D. F. (2026). Machine Learning-Based Network Traffic Anomaly Detection Using the CIC-IDS2017 Dataset. *UPGRADE: Jurnal Pendidikan Teknologi Informasi*, *3*(2).
Rui, H. Z., Chien, T. Y., Ee, L. X., San, C. W., & Yi, L. T. (2025). Comparison of the Use of Support Vector Machine (SVM) & Random Forest Algorithms (RF) for DDoS Attack Detection. *International Journal of Research and Innovation in Social Science (IJRISS)*, *9*(1), 1126–1138.
Sawah, M. S., Elmannai, H., El-Bary, A. A., Lotfy, K., & Sheta, O. E. (2025). Distributed Denial of Service (DDoS) Classification Based on Random Forest Model with Backward Elimination Algorithm and Grid Search Algorithm. *Scientific Reports*, *15*(1), 19063.
Sugeng, W., & Adhliana, F. (2026). Deteksi Serangan DDoS pada Jaringan SDN Menggunakan Random Forest dan Particle Swarm Optimization. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *13*(1), 39–46.
Suryadewiansyah, M. K., & Tju, T. E. E. (2022). Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *8*(2), 81–88.
Vasin, N. N., & Kakabian, K. S. (2025). Application of Adaptive Neuro-Fuzzy Inference

System for DDoS Attack Detection Based on CIC-DDoS-2019 Dataset. *Proceedings of Telecommunication Universities*, *11*(3), 87–96.

Wiratama, R., Pirdhaus, A., Bintoro, E. R. P., Sari, Z., & Syaifuddin, S. (2026). Random Forest and LLM Synergies Framework for Autonomous DDoS Mitigation. *Jurnal Teknik Informatika (Jutif)*, *7*(1), 515–528.

Wolf, A. O. G., & Optimizer, G. W. (n.d.). *Model Klastering Hybrid Menggunakan Inisialisasi K-Means+*.