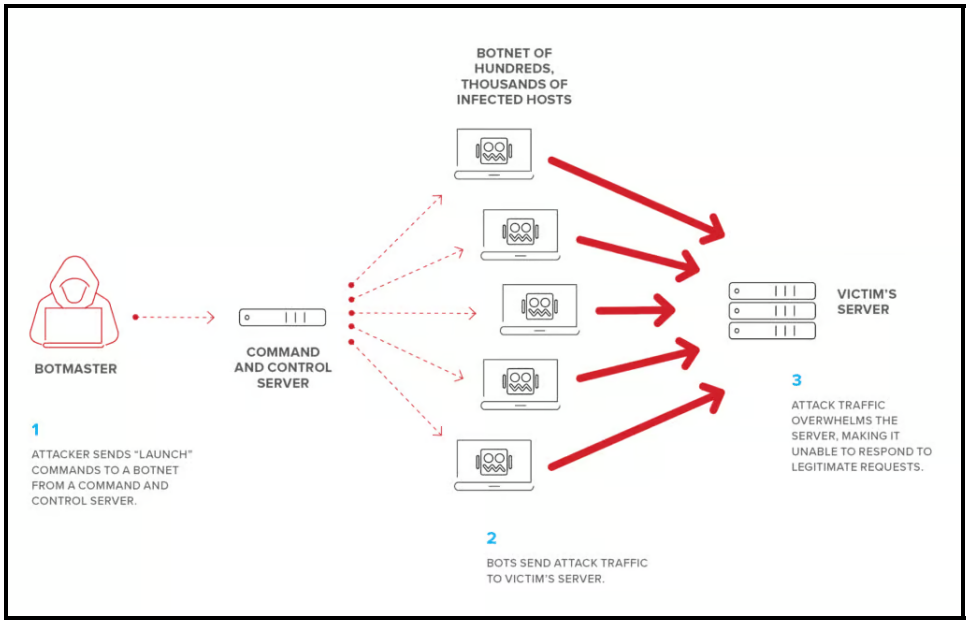
BAB 2
LANDASAN TEORI

2.1 Serangan Distributed Denial of Service (DDoS)

Serangan Distributed Denial of Service (DDoS) merupakan salah satu bentuk ancaman siber yang berbahaya karena dapat menghentikan atau memperlambat layanan jaringan secara signifikan. Serangan ini dilakukan dengan mengirimkan trafik dalam jumlah besar secara bersamaan dari berbagai titik (terdistribusi) ke suatu target, seperti server, sistem, atau layanan tertentu. Tujuannya adalah untuk membuat sumber daya target menjadi kewalahan dan akhirnya tidak dapat diakses oleh pengguna yang sah.

DDoS biasanya memanfaatkan botnet, yakni jaringan perangkat yang telah dikompromi dan dikendalikan oleh penyerang. Dalam konteks infrastruktur jaringan enterprise, serangan ini sangat merugikan karena dapat menimbulkan downtime layanan, hilangnya kepercayaan pelanggan, serta kerugian finansial [7]. Oleh karena itu, deteksi dini terhadap serangan DDoS sangat penting dilakukan secara efektif dan otomatis. Secara umum, serangan DDoS dalam bentuk flow ditunjukkan pada gambar berikut [8]:



Gambar 2.1. Diagram alur DDoS

2.2 Deteksi Serangan Berbasis Machine Learning

Machine Learning (ML) adalah pendekatan yang memungkinkan sistem komputer untuk belajar dari data historis dan membuat prediksi atau keputusan berdasarkan pola yang ditemukan. Dalam deteksi serangan jaringan seperti DDoS, pendekatan supervised learning sering digunakan karena dapat memetakan fitur-fitur dari lalu lintas jaringan (flow) menjadi kelas tertentu, seperti benign (normal) atau malicious (serangan). Algoritma ML bekerja dengan membangun model dari data pelatihan, lalu menggunakan model tersebut untuk mengklasifikasikan data baru. Keunggulan ML dibandingkan metode rule-based konvensional adalah kemampuannya menangani data berskala besar, menangkap pola non-linier, dan memperbarui model secara adaptif ketika ada pola serangan baru [6]. Dua algoritma yang populer dalam deteksi serangan DDoS adalah Random Forest dan CatBoost, yang akan dijelaskan pada bagian berikut.

2.3 Random Forest

Random Forest merupakan algoritma ensemble learning berbasis pohon keputusan (decision tree) yang diperkenalkan oleh Breiman [9], dengan memanfaatkan pendekatan bagging untuk meningkatkan akurasi dan mengurangi risiko overfitting dalam proses klasifikasi maupun regresi. Algoritma ini bekerja dengan membuat banyak pohon keputusan dari subset acak data pelatihan, dan menggabungkan hasil prediksi masing-masing pohon untuk menghasilkan keputusan akhir secara kolektif.

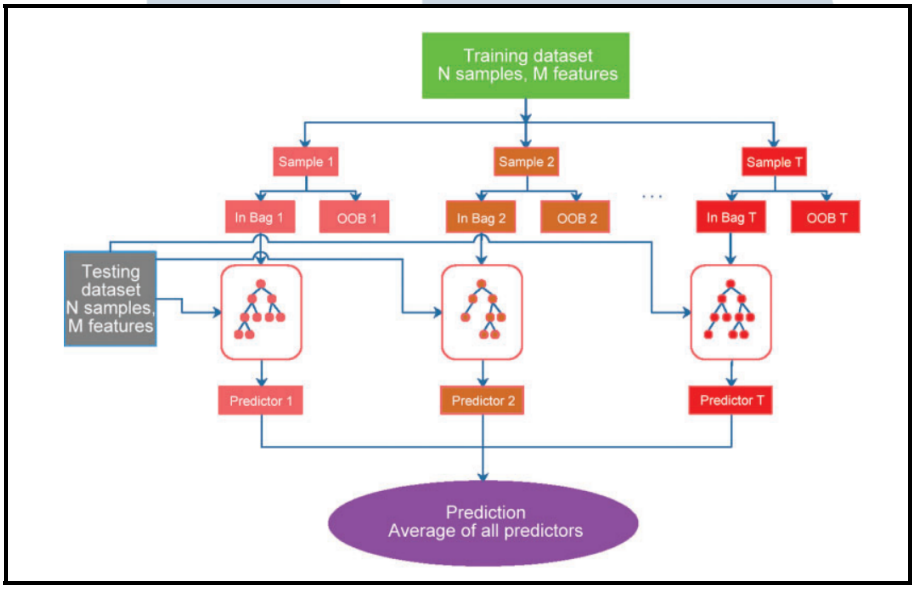
Proses kerja Random Forest diawali dengan melakukan bootstrap sampling, yaitu membentuk beberapa subset data secara acak (disebut In Bag), di mana setiap subset digunakan untuk melatih satu pohon keputusan secara independen. Data yang tidak masuk ke dalam subset pelatihan disebut sebagai Out-of-Bag (OOB) dan digunakan sebagai validasi internal untuk mengevaluasi akurasi model [10].

Setiap pohon dalam Random Forest dibangun dengan memilih subset fitur secara acak pada tiap percabangan (node splitting), biasanya menggunakan kriteria Gini impurity. Semakin rendah nilai impurity, semakin baik pemisahan data pada node tersebut. Struktur CART (Classification and Regression Tree) dalam Random Forest bersifat biner dan simetris, yang memungkinkan setiap node non-daun menghasilkan dua cabang berdasarkan fitur terbaik yang dipilih secara acak [10].

Setelah semua pohon dilatih, proses prediksi dilakukan dengan cara:

- Untuk klasifikasi: menggunakan mayoritas suara (majority voting) dari seluruh pohon.
- Untuk regresi: menggunakan rata-rata dari hasil prediksi seluruh pohon.

Gambar 2.2 menjelaskan alur kerja algoritma Random Forest, mulai dari pembentukan subset data, pelatihan pohon-pohon prediktor, hingga agregasi hasil prediksi akhir [10].



Gambar 2.2. Alur algoritma Random Forest

Keunggulan utama Random Forest terletak pada kemampuannya dalam menangani data berdimensi tinggi, ketahanannya terhadap overfitting, serta efisiensinya dalam pelatihan karena dapat dilakukan secara paralel. Algoritma ini juga sangat andal untuk digunakan dalam sistem deteksi intrusi seperti DDoS karena dapat mengenali pola lalu lintas yang kompleks dan tidak teratur secara akurat.

Secara matematis, prediksi klasifikasi $H(x)$ dalam Random Forest diberikan oleh:

$$H(x) = \text{mode}\{h_1(x), h_2(x), ..., h_n(x)\} \tag{2.1}$$

dengan $h_i(x)$ adalah prediksi dari pohon ke-i, dan mode menyatakan nilai mayoritas dari seluruh prediksi. Hasil prediksi akhir diperoleh dari voting mayoritas (untuk klasifikasi) atau rata-rata (untuk regresi) dari semua pohon [11].

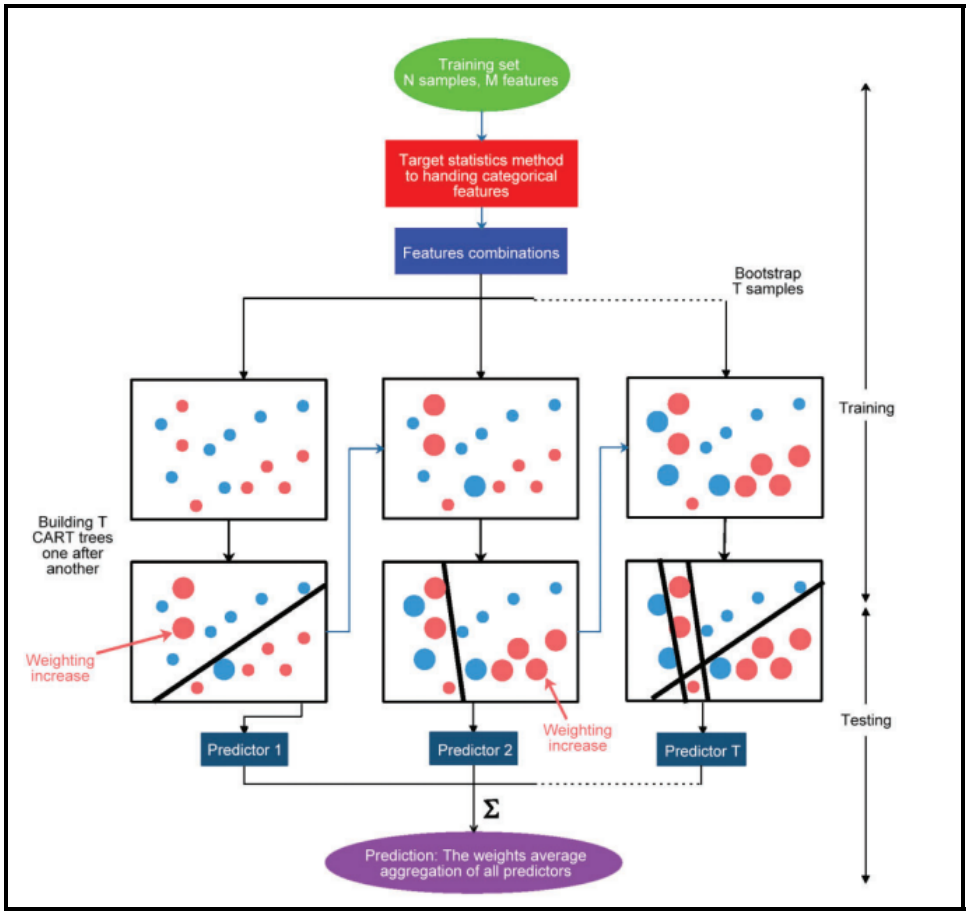
2.4 CatBoost

CatBoost (Categorical Boosting) merupakan algoritma gradient boosting yang dikembangkan oleh Yandex, dan dirancang untuk menangani fitur kategorikal secara efisien tanpa perlu dilakukan encoding eksplisit. CatBoost bekerja dengan cara membangun model secara iteratif menggunakan pendekatan gradient boosting pada pohon keputusan, namun dengan teknik pemrosesan fitur yang unik seperti penggunaan *ordered boosting* dan *target statistics* [12].

Ordered boosting merupakan teknik yang digunakan untuk mencegah *target leakage*, yaitu kebocoran informasi dari data target ke dalam proses pelatihan model [10]. Pada CatBoost, data pelatihan diproses secara sekuensial, di mana nilai rata-rata target untuk setiap kategori dihitung berdasarkan observasi sebelumnya, bukan keseluruhan data. Hal ini menjadikan model lebih andal saat diterapkan pada data nyata.

Selanjutnya, CatBoost memanfaatkan *symmetric tree structure*, di mana setiap pohon dibangun dengan struktur yang seimbang dan konsisten di semua cabang. Hal ini mengurangi kompleksitas model dan mempercepat proses prediksi [13].

Gambar 2.3 berikut menggambarkan alur kerja dari algoritma CatBoost [10]. Dimulai dari data mentah (baik numerik maupun kategorikal), CatBoost melakukan pemrosesan internal terhadap fitur kategorikal, kemudian mengaplikasikan *ordered boosting* secara iteratif hingga mencapai model akhir.



Gambar 2.3. Alur algoritma CatBoost

Dengan pendekatan ini, CatBoost menunjukkan performa tinggi dalam berbagai tugas klasifikasi dan regresi, terutama pada data yang mengandung banyak fitur kategorikal. Dalam konteks penelitian ini, CatBoost digunakan sebagai model klasifikasi untuk membedakan antara trafik benign dan serangan DDoS secara akurat dan efisien.

Rumus umum gradient boosting adalah:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{2.2}$$

dengan:

- $F_m(x)$ adalah model gabungan pada iterasi ke-m,
- $F_{m-1}(x)$ adalah model sebelumnya,
- $h_m(x)$ adalah model weak learner (pohon keputusan kecil),

- γ_m adalah koefisien pembobot hasil training.

CatBoost memperbaiki kelemahan boosting tradisional dengan mengintroduksi:

- *Ordered boosting* untuk menghindari data leakage,
- Penggunaan *symmetric trees* untuk mempercepat pelatihan,
- Optimasi pemrosesan data kategorikal.

2.5 Metode Evaluasi Kinerja

Untuk mengevaluasi kinerja model klasifikasi dalam mendeteksi serangan DDoS, digunakan tiga metrik utama yaitu Precision, Recall, dan F1-Score. Ketiga metrik ini relevan untuk sistem deteksi karena sering kali data bersifat tidak seimbang.

2.5.1 Precision

Precision adalah ukuran yang menunjukkan seberapa banyak dari semua prediksi positif yang dibuat oleh model benar-benar merupakan kelas positif.

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

Keterangan:

TP (True Positive): Jumlah data positif yang berhasil diklasifikasikan dengan benar sebagai positif.

FP (False Positive): Jumlah data negatif yang salah diklasifikasikan sebagai positif.

Semakin tinggi nilai precision, semakin sedikit kesalahan positif palsu (false positive) yang dilakukan oleh model.

2.5.2 Recall

Recall atau disebut juga *sensitivity* atau *true positive rate*, mengukur seberapa banyak dari semua data positif yang berhasil diidentifikasi dengan benar oleh model.

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

Keterangan:

FN (False Negative): Jumlah data positif yang salah diklasifikasikan sebagai negatif.

Recall sangat penting ketika kesalahan dalam mengklasifikasikan data positif sebagai negatif (false negative) dapat menimbulkan dampak besar, seperti pada kasus deteksi serangan siber seperti DDoS.

2.5.3 F1-Score

F1-Score adalah ukuran rata-rata harmonis dari precision dan recall. F1-Score berguna untuk menyeimbangkan antara precision dan recall, terutama ketika distribusi kelas tidak seimbang.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.5}$$

Keterangan:

F1-Score memberikan nilai yang tinggi hanya ketika precision dan recall keduanya tinggi.

Cocok digunakan pada kasus seperti deteksi anomali atau deteksi serangan DDoS, di mana ketepatan dan sensitivitas sama-sama penting.

2.6 Feature Extraction

Feature extraction merupakan tahapan penting dalam prapemrosesan data pada machine learning, dimana data mentah diubah menjadi representasi fitur yang lebih bermakna dan informatif untuk proses klasifikasi atau regresi. Dalam konteks deteksi serangan DDoS, fitur-fitur yang diambil berhubungan langsung dengan lalu lintas jaringan, seperti *Fwd Packet Length Mean*, *Packet Length Variance*, dan *Total Fwd Packets*. Tujuan utamanya adalah memisahkan fitur yang relevan dari data noise atau redundan, demi efisiensi dan akurasi model [14].

Pada penelitian ini, dataset CIC-DDoS2019 menyediakan lebih dari 80 fitur statistik jaringan. Namun tidak semua fitur tersebut memberikan kontribusi

signifikan terhadap klasifikasi trafik benign dan DDoS. Oleh karena itu dilakukan seleksi manual berdasarkan pemahaman domain dan hasil eksplorasi statistik awal, dengan mempertahankan fitur yang informatif dan menghapus fitur yang bersifat konstan atau berisi banyak nilai nol.

Pendekatan ini memastikan fitur yang digunakan relevan terhadap pola serangan DDoS sekaligus mengurangi kompleksitas komputasi. Dengan memilih fitur yang tepat, proses pelatihan model Random Forest dan CatBoost dapat berjalan lebih efisien tanpa mengurangi ketepatan deteksi.

2.7 Feature Importance

Feature importance merupakan teknik evaluasi yang digunakan untuk mengukur kontribusi masing-masing fitur terhadap performa model *machine learning* dalam tugas klasifikasi atau regresi. Teknik ini berperan penting dalam proses *feature selection* karena mampu mengidentifikasi fitur-fitur yang paling relevan terhadap target prediksi, serta meningkatkan efisiensi dan akurasi model.

Random Forest secara otomatis menghitung tingkat kepentingan fitur menggunakan metode *mean decrease in impurity*, yaitu berdasarkan penurunan rata-rata impurity (seperti Gini) yang dihasilkan setiap kali fitur digunakan dalam proses pemisahan (*splitting*) pada pohon keputusan. Nilai feature importance ini bersifat relatif dan dijumlahkan hingga bernilai satu [15]. Sementara itu, **CatBoost** menyediakan berbagai metode untuk menghitung feature importance, antara lain *PredictionValuesChange*, *LossFunctionChange*, dan *Permutation Importance*. Selain itu, interpretasi model dapat diperdalam dengan metode *SHAP (SHapley Additive exPlanations)* yang menghitung kontribusi setiap fitur terhadap prediksi individual [16].

Penelitian ini menggunakan pendekatan feature importance pada algoritma Random Forest dan CatBoost untuk menganalisis 12 fitur penting yang telah diseleksi dari dataset CIC-DDoS2019 (versi sampel). Analisis ini bertujuan untuk:

1. Mengidentifikasi fitur yang paling memengaruhi klasifikasi antara trafik benign dan serangan DDoS,
2. Menyederhanakan kompleksitas model melalui pengurangan fitur tanpa menurunkan performa secara signifikan,
3. Memberikan wawasan terhadap karakteristik penting dalam pola serangan DDoS yang dapat digunakan dalam pengembangan sistem keamanan

jaringan.

2.8 5-Fold Cross-Validation

Cross-validation merupakan salah satu metode evaluasi performa model yang umum digunakan dalam *machine learning*. Salah satu jenis cross-validation yang populer digunakan adalah **5-fold cross-validation**. Metode ini membagi data menjadi lima bagian atau subset yang kurang lebih berukuran sama. Pada setiap iterasi, empat subset digunakan sebagai data pelatihan, sedangkan subset sisanya digunakan sebagai data validasi atau pengujian. Proses ini diulang sebanyak lima kali hingga masing-masing subset pernah digunakan sebagai data validasi satu kali, dan hasil akhir diperoleh dari rata-rata performa semua iterasi [17, 18].

5-fold cross-validation banyak digunakan karena dapat memberikan estimasi performa model yang stabil dengan varians rendah serta mampu meminimalisasi risiko *overfitting*. Menurut studi yang dilakukan oleh Refaeilzadeh et al. (2023), penggunaan 5-fold cross-validation secara signifikan meningkatkan reliabilitas evaluasi model karena setiap data digunakan untuk pelatihan sekaligus validasi, sehingga menghasilkan estimasi akurasi yang lebih konsisten dibandingkan metode validasi tunggal seperti hold-out validation [19].

Selain itu, menurut penelitian Jain et al. (2020), 5-fold cross-validation juga dinilai efektif dalam evaluasi model deteksi intrusi jaringan seperti serangan DDoS, karena metode ini mampu menangkap variasi pola data secara lebih representatif dan memberikan gambaran yang realistis tentang kemampuan model dalam mengklasifikasikan data yang belum pernah dilihat sebelumnya [20].

2.9 Dataset CIC-DDoS2019

Dataset CIC-DDoS2019 merupakan kumpulan data yang dikembangkan oleh Canadian Institute for Cybersecurity untuk menyimulasikan skenario nyata serangan Distributed Denial of Service (DDoS) modern [21]. Dataset ini dirancang dengan pendekatan realistis terhadap lalu lintas jaringan dan memuat berbagai jenis serangan berbasis amplifikasi maupun refleksi. Dataset bersifat flow-based, di mana setiap baris mewakili satu sesi komunikasi atau aliran trafik (flow) yang terdiri atas fitur-fitur statistik jaringan. Meskipun data memiliki komponen waktu, dataset ini tidak digunakan sebagai time series karena penelitian difokuskan pada klasifikasi pola trafik, bukan prediksi berurutan berdasarkan waktu.

Dalam penelitian ini, digunakan versi *sample ringan* dari dataset CIC-DDoS2019 yang terdiri dari lima file utama, yaitu:

- `LDAP_sample.csv`
- `MSSQL_sample.csv`
- `NetBIOS_sample.csv`
- `Portmap_sample.csv`
- `Syn_sample.csv`