BAB 2
LANDASAN TEORI

2.1 *Distributed Denial-of-Service Attacks* (DDoS)

Pada awalnya, serangan *Denial of Service* (DoS) berasal dari satu sumber saja. DoS adalah serangan siber di mana penyerang mencoba membuat mesin atau sumber daya jaringan tidak dapat diakses oleh klien yang sah dengan mengganggu layanan dari host yang terhubung ke Internet, baik secara sementara maupun permanen [16]. Serangan DoS membuat sumber daya jaringan tidak tersedia bagi pengguna dengan menghentikan layanan [17].

Serangan DDoS merupakan perkembangan dari serangan DoS yang membuat proses mitigasi dan pencegahannya menjadi lebih sulit, serta menimbulkan dampak yang lebih besar terhadap sistem yang menjadi target. Berbeda dengan serangan DoS yang umumnya berasal dari satu sumber, serangan DDoS memanfaatkan banyak sumber secara bersamaan untuk mengirimkan paket ke target. Hal ini menyebabkan lalu lintas serangan sulit dibedakan dari lalu lintas yang sah. Selain itu, volume lalu lintas yang sangat besar dapat melebihi kapasitas yang mampu ditangani oleh sistem, sehingga mengakibatkan gangguan layanan, penurunan kinerja sistem, hingga kegagalan operasi yang dapat menyebabkan hilangnya sebagian atau seluruh fungsi sistem [18].

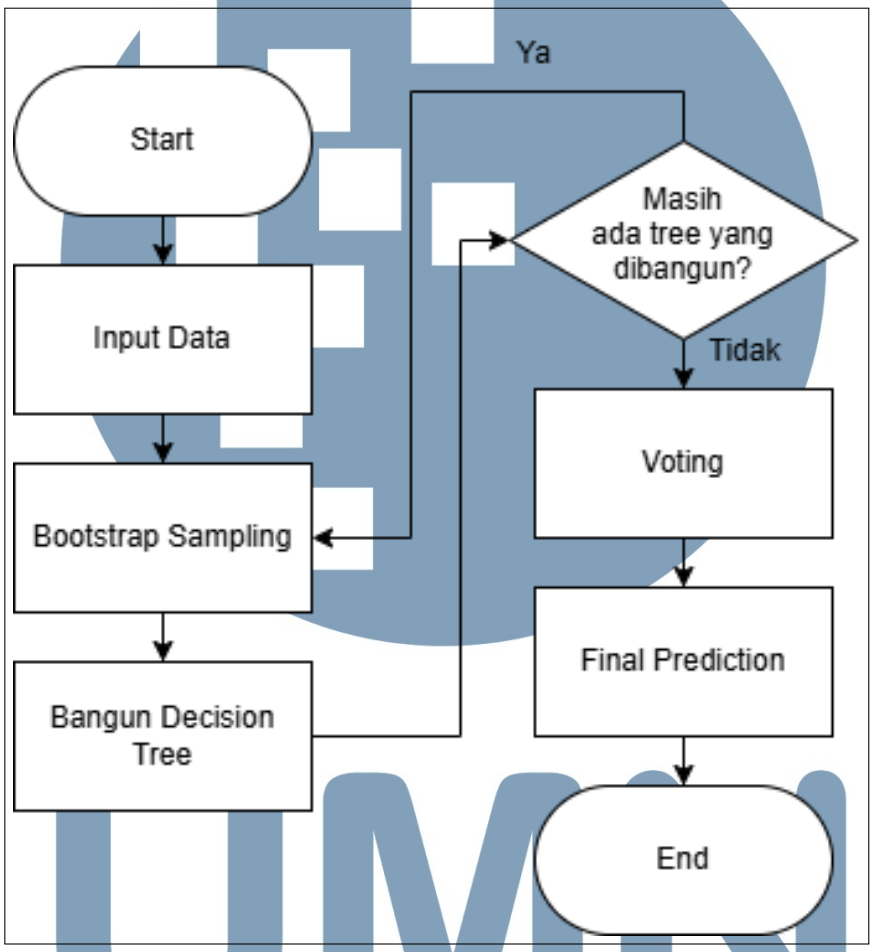
Ada dua metode untuk melancarkan serangan DDoS di Internet. Pertama, penyerang dapat membingungkan protokol atau aplikasi yang berjalan pada mesin korban dengan mengirimkan paket berbahaya (serangan kerentanan). Kedua, penyerang dapat mengirimkan sejumlah besar paket yang tidak berguna untuk menghabiskan semua sumber daya mesin korban, sehingga mencegahnya berinteraksi dengan pengguna yang sah (serangan *bandwidth/flooding*) [16].

2.2 Algoritma Yang Digunakan

Penelitian ini membandingkan algoritma *Random Forest, Decision Tree, Logistic Regression,* dan SVM. Keempat algoritma tersebut merupakan algoritma *supervised learning* untuk klasifikasi, namun memiliki pendekatan yang berbeda dalam melakukan prediksi. Penelitian ini membandingkan algoritma yang berbasis pohon keputusan, algoritma berbasis *ensemble*, algoritma linear probabilistik, dan algoritma linear berbasis margin sehingga dapat diketahui algoritma mana yang

paling efektif untuk mendeteksi serangan DDoS pada dataset BoT-IoT.

2.2.1 *Random Forest*



Gambar 2.1. Diagram Flowchart *Random Forest*

Random Forest adalah algoritma pembelajaran mesin yang digunakan untuk melakukan klasifikasi dan regresi, algoritma ini bekerja dengan cara membangun hutan pohon keputusan selama pelatihan. Hasil *Random Forest* ditentukan oleh gabungan prediksi yang dibuat oleh masing-masing pohon. Untuk klasifikasi, hasil dari algoritma adalah modus dari prediksi yang diberikan oleh setiap pohon, sedangkan untuk regresi, hasilnya adalah rata-rata dari prediksi yang diberikan oleh setiap pohon yang didapat dari rumus 2.1 [2].

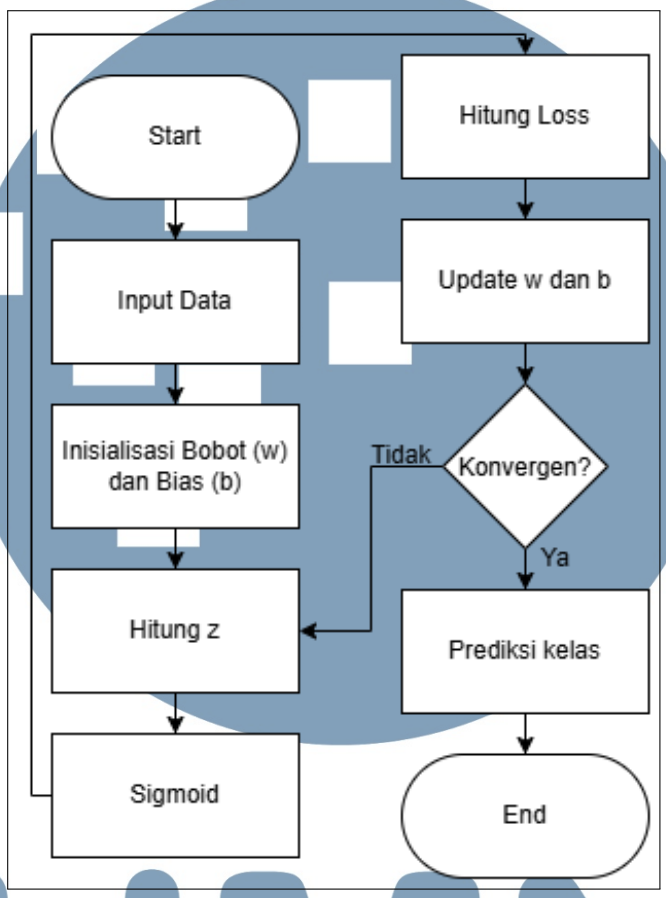
$$H(x) = \frac{1}{N}\sum_{i=1}^{N}(h(x_i, \theta_i)) \tag{2.1}$$

Dalam konteks dimana N adalah jumlah pohon yang dibuat oleh algoritma dan θ_i mewakili parameter dari pohon ke-i.

Random Forest menerapkan metode *Bootstrap Aggregating* (*Bagging*) untuk meningkatkan performa dan stabilitas model. Pada metode ini, sejumlah *dataset* baru dibentuk dengan cara melakukan pengambilan sampel secara acak dari *dataset* asli menggunakan teknik pengambilan dengan penggantian (*sampling with replacement*). Setiap *dataset* hasil pengambilan sampel tersebut disebut sebagai *bootstrap sample* dan digunakan untuk melatih satu pohon keputusan secara independen [19].

Misalkan tersedia *dataset* D yang terdiri atas N data. Melalui proses *bootstrap*, dibentuk sejumlah subset D_i yang masing-masing juga berukuran N. Karena pengambilan dilakukan dengan penggantian, suatu data dapat terpilih lebih dari satu kali, sedangkan data lainnya mungkin tidak terpilih sama sekali. Proses ini diulang sebanyak jumlah pohon yang akan dibangun dalam model *Random Forest* [2].

2.2.2 *Logistic Regression*



Gambar 2.2. Diagram Flowchart *Logistic Regression*

Logistic Regression sering digunakan untuk masalah klasifikasi biner. Model ini memperkirakan probabilitas hasil biner dengan menggunakan fungsi logistik seperti yang dijelaskan dalam rumus 2.2.

$$P_{(y=1)} = \frac{1}{(1+e^{-(\beta_0+\beta_1*x_1+...+\beta_N*x_N)}} \quad (2.2)$$

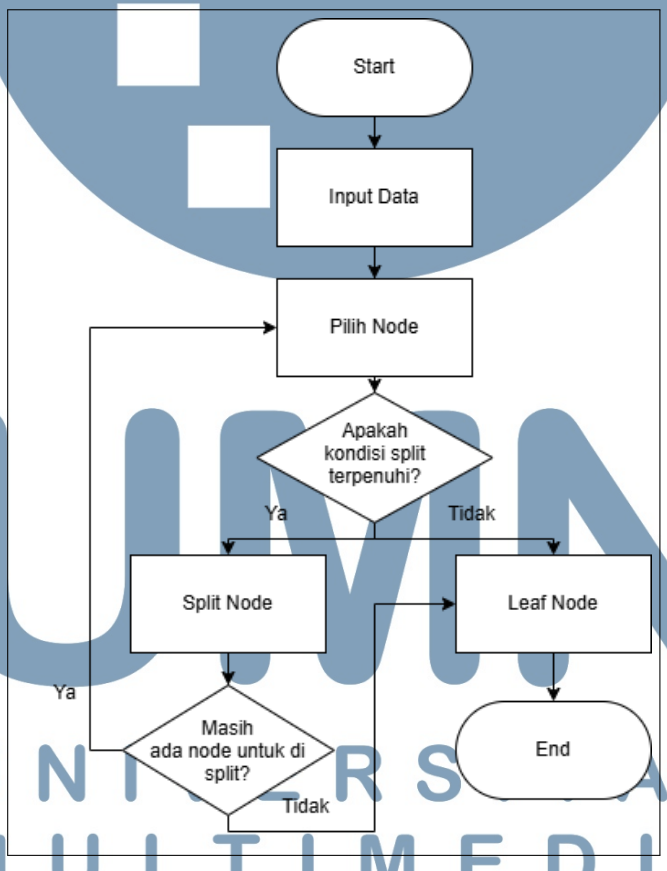
Dimana $P_{(y=1)}$ adalah probabilitas hasil biner, $\beta_0,\beta_1,...\beta_N$, adalah koefisien regresi dan $x_0,x_1,...,x_N$ adalah variabel fitur [2].

Seberapa baik model dapat memprediksi dapat ditentukan menggunakan rumus kerugian *binary cross-entropy* 2.3

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] \quad (2.3)$$

Dimana m adalah jumlah contoh pelatihan, $y^(i)$ adalah label aktual untuk pelatihan ke-*i*, $h_\theta(x^{(i)})$ probabilitas prediksi bahwa labelnya adalah 1 (kelas positif) untuk contoh pelatihan ke-*i*, dan θ merepresentasikan parameter dari model *logistic regression*.

2.2.3 *Decision Tree*



Gambar 2.3. Diagram Flowchart *Decision Tree*

Decision Tree merupakan algoritma pembelajaran terawasi (*supervised learning*) yang bersifat non-parametrik dan dapat digunakan untuk menyelesaikan permasalahan klasifikasi maupun regresi. Algoritma ini merepresentasikan proses pengambilan keputusan dalam bentuk struktur pohon, di mana setiap simpul (*node*)

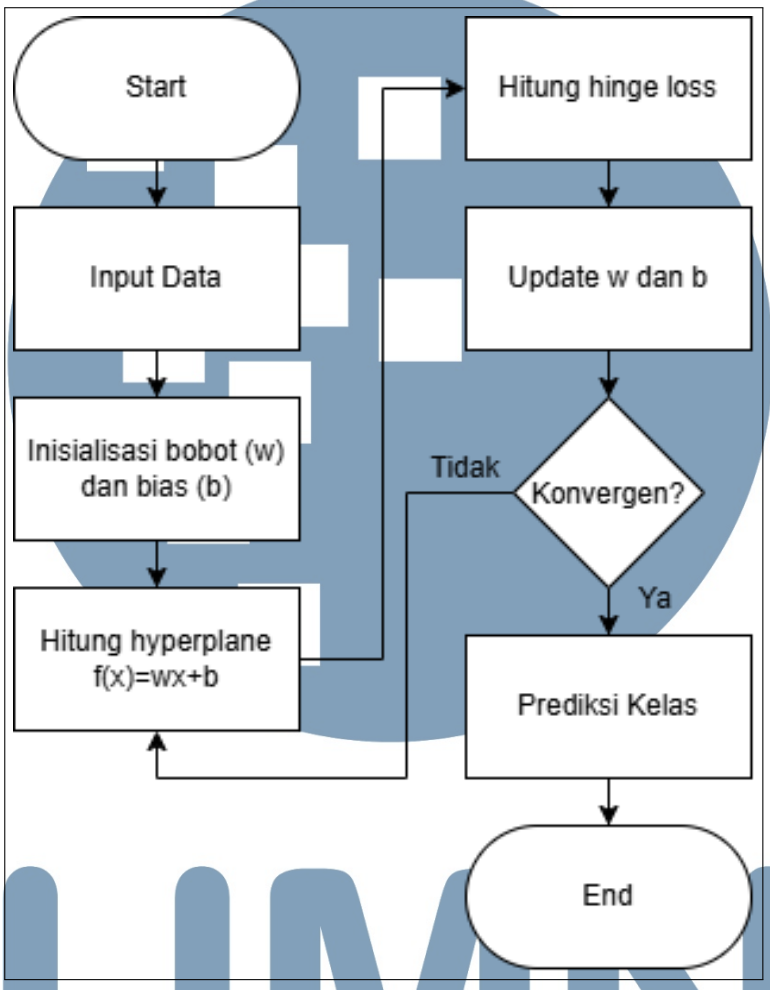
melakukan pemisahan data berdasarkan nilai suatu fitur. Proses pemisahan tersebut dilakukan secara berulang (*recursive splitting*) hingga terbentuk cabang-cabang yang menghasilkan prediksi akhir. Dalam menentukan pemisahan terbaik pada setiap simpul, metrik yang umum digunakan antara lain Ketidakmurnian Gini (*Gini Impurity*) dan Gain Informasi (*Information Gain*) yang dihitung berdasarkan nilai Entropi. Konsep ini digambarkan melalui rumus (2.4-2.5).

$$Gini = 1 - \sum_{k=1}^{N} (p_k)^2 \tag{2.4}$$

$$H(S) = -\sum_{k=1}^{N} (p_k \log_2(p_k)) \tag{2.5}$$

Di mana p_k adalah proporsi sampel yang termasuk dalam kelas k dalam himpunan S [2].

2.2.4 *Support Vector Machine* (SVM)



Gambar 2.4. Diagram Flowchart *Support Vector Machine*

SVM adalah algoritma pembelajaran mesin yang sangat efektif untuk tugas klasifikasi. Algoritma ini bekerja dengan mengidentifikasi *hyperplane* optimal yang memisahkan pengamatan yang termasuk dalam satu kelas dari yang lain berdasarkan pola informasi yang disebut fitur. Fitur-fitur ini biasanya bukan data mentah tetapi data hasil turunan dari proses interpolasi selama proses pemilihan fitur [20].

2.3 *Internet of Things* (IoT)

Lingkungan IoT menggabungkan berbagai entitas jaringan, termasuk perangkat lunak, sensor, dan teknologi lainnya. Komponen-komponen ini bekerja

bersama untuk menghubungkan sistem dan perangkat melalui Internet serta memungkinkan pertukaran data. Dalam IoT, terdapat berbagai entitas yang terintegrasi seperti sensor, manusia, atau layanan. Teknologi ini memperkenalkan arsitektur komputasi baru yang secara mendasar mengubah kehidupan sehari-hari. IoT banyak digunakan dalam manajemen rantai pasokan, layanan kesehatan, transportasi, produksi industri, dan rumah pintar [3].

Koneksi antar mesin dan perangkat melalui internet memungkinkan terciptanya data yang dapat digunakan untuk memberikan wawasan tentang kinerja analitik dan mendukung teknologi baru [21]. Sekelompok objek statis dan/atau bergerak yang saling terhubung, seperti perangkat yang dilengkapi dengan modul komunikasi, sensor, dan aktuator yang terhubung melalui internet [22].

2.4 Uji Sistem

Uji sistem dilakukan dengan cara menghitung akurasi, presisi, *recall*, skor F1, *macro* dan *weighted average* dari masing-masing algoritma. Hal ini dilakukan menggunakan rumus (2.6-2.9) [2].

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{2.6}$$

Accuracy adalah perbandingan dari data yang diklasifikasikan positif ataupun negatif secara benar dengan total data yang diklasifikasikan [23].

$$Precision = \frac{TP}{(TP+FP)} \tag{2.7}$$

Precision adalah perbandingan dari data yang diklasifikasikan positif secara benar dengan total data yang diklasifikasikan positif [23].

$$Recall = \frac{TP}{(TP+FN)} \tag{2.8}$$

Recall adalah perbandingan dari data yang diklasifikasikan positif secara benar dengan semua data yang positif [23].

$$F1 = \frac{2(Precision * Recall)}{(Precision + Recall)} \tag{2.9}$$

F1 *Score* adalah metrik evaluasi yang menggabungkan *precision* dan *recall* secara harmonis untuk menilai kinerja model klasifikasi, terutama pada data yang tidak seimbang [24].

Dimana TP adalah *True Positive*, TN adalah *True Negative*, FP adalah *False Positive*, dan FN adalah *False Negative*.

Perbedaan utama dari *macro* dan *weighted average* adalah *macro average* merupakan rata-rata matematis dari skor setiap kelas, sedangkan *weighted average* memperhitungkan jumlah sampel dari masing-masing kelas. *Macro average* dihitung menggunakan rata-rata tanpa bobot, sehingga setiap kelas dianggap memiliki tingkat kepentingan yang sama, tanpa memandang *support* yang dimilikinya. Akibatnya, jika performa model pada kelas minoritas rendah, nilai rata-rata makro dapat menurun secara signifikan. Sebaliknya, *weighted average* memperhitungkan jumlah instans sebenarnya pada setiap kelas. Dengan demikian, kelas yang memiliki lebih banyak data akan memberikan pengaruh yang lebih besar terhadap nilai akhir. Pendekatan ini membantu mengatasi permasalahan ketidakseimbangan data [25].

2.5 Penelitian Terdahulu

Penelitian yang dilakukan oleh G. Airlangga membandingkan algoritma *Random Forest*, *Logistic Regression*, dan *Decision Tree* untuk mengklasifikasikan serangan DDoS pada jaringan *network*. Hasil penelitian tersebut menunjukkan bahwa algoritma *Random Forest* memiliki performa yang lebih baik dibandingkan dengan kedua algoritma lainnya, meskipun membutuhkan sumber daya komputasi yang lebih tinggi. *Dataset* yang digunakan dalam penelitian ini berasal dari platform Kaggle dengan nama "*DDoS Attack Network Logs*" [2].

Penelitian yang dilakukan oleh S. Abiramasundari, dkk. membandingkan beberapa algoritma pembelajaran mesin, yaitu *Random Forest*, *Logistic Regression*, *Decision Tree*, *K-Nearest Neighbor* (KNN), dan *Support Vector Machine* (SVM), untuk mendeteksi serangan *Distributed Denial of Service* (DDoS) pada jaringan *network*. Penelitian ini menggunakan tiga *dataset*, yaitu CICIDS2018, CICIDS2017, dan CICDDoS2019. Berdasarkan hasil pengujian, pada *dataset* CICIDS2017, algoritma *Random Forest* memperoleh akurasi tertinggi sebesar 98.9%. Pada *dataset* CICIDS2018, algoritma SVM dan *Decision Tree* masing-masing mencapai akurasi sebesar 98.7% dan 98.6%. Sementara itu, pada *dataset*

CICDDoS2019, algoritma KNN dan *Random Forest* menunjukkan akurasi tertinggi sebesar 98.7% [6].

Penelitian yang dilakukan oleh R.P. Janivasya, dkk. mengevaluasi efektivitas delapan algoritma pembelajaran mesin, yaitu *Logistic Regression*, KNN, SVM, *Random Forest*, *Decision Tree*, *Long Short-Term Memory* (LSTM), *Multilayer Perceptron* (MLP), dan *Gated Recurrent Unit* (GRU), dengan menggunakan *dataset* UNSW-NB15. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memiliki efektivitas terbaik, sementara algoritma *Decision Tree* menunjukkan performa yang tidak jauh berbeda [5].