BAB III

Metodologi Penelitian

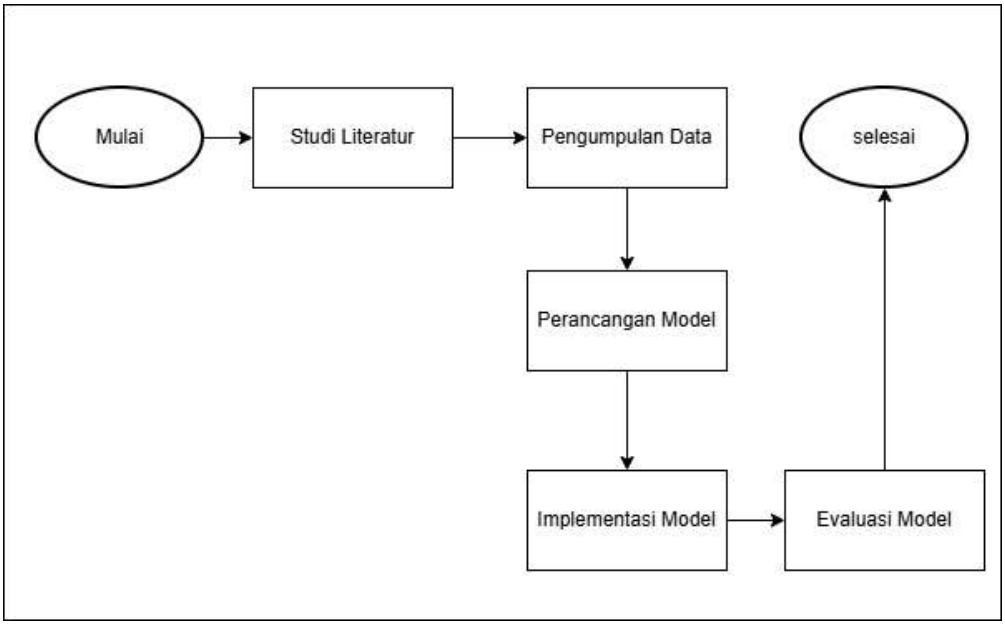
3.1 Spesifikasi Sistem dan Alat Yang Digunakan

Penelitian ini dilakukan dalam lingkungan virtual menggunakan platform VirtualBox. Spesifikasi lingkungan uji yang digunakan pada penelitian ini adalah sebagai berikut.

1. *Operating System* : Ubuntu 24.04
2. *Base Memory* : 6 GB
3. *Processor* : AMD Ryzen 5 4500U with Radeon Graphic
4. *CPU* : 8

3.2 Tahapan Penelitian

Penelitian dilakukan menggunakan metode eksperimen, di mana pengaruh dari kombinasi model IDS berbasis *signature* dan *anomaly* akan diukur. Penelitian ini dilakukan melalui beberapa tahapan, yaitu studi literatur untuk membuat konsep bahasan penelitian, pengumpulan data untuk mencari data yang nantinya akan diolah, perancangan model untuk menghasilkan model yang direncanakan, implementasi model untuk mewujudkan model yang dirancang, dan terakhir evaluasi model yang dilakukan untuk mengevaluasi model yang digunakan pada penelitian. Tahapan penelitian yang dilakukan dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

3.3 Studi Literatur

Pada tahap ini dilakukan pencarian terhadap topik yang relevan dengan penelitian, yaitu IDS, serangan siber*, machine learning,* IDS berbasis *signature,* dan IDS berbasis *anomaly.* Studi literatur menjadi landasan penelitian sekaligus memberikan wawasan untuk merancangan model.

3.4 Pengumpulan Data

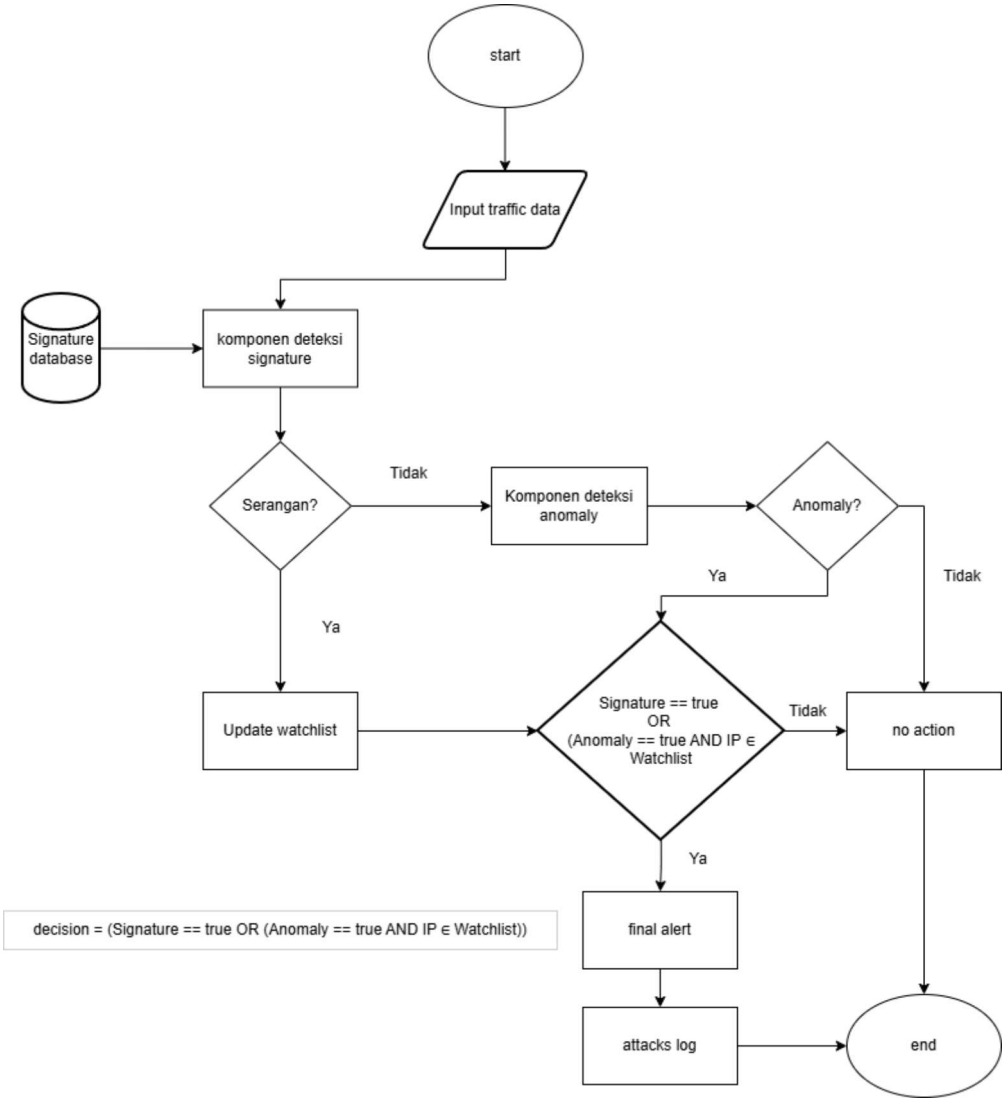
Tahapan ini dilakukan untuk mengumpulkan data yang akan dijadikan bahan utama dalam proses analisis dan pengujian. Data yang digunakan merupakan data dari pihak ketiga bernama CIC-IDS2017. Data ini mencakup berbagai jenis serangan, seperti *Bruteforce, Denial of Service (DoS), Distributed Denial of Service (DDoS), SQL Injection, Cross site scripting* (XSS), dan *Botnet*.

3.5 Perancangan Model

Perancangan model dilakukan untuk menggambarkan alur dan cara kerja model yang akan dirancang. Pada perancangan ini terdapat tiga tahap, yaitu perancangan model hibrida sebagai sistem utama, penyesuaian komponen *signature,* dan terakhir perancangan komponen *anomaly* menggunakan model *Isolation Forest.*

3.5.1 Model Hibrida

Model hibrida terdiri dari dua komponen utama, yaitu komponen deteksi serangan berbasis *signature* dan komponen deteksi anomali berbasis *machine learning.* Selain itu, digunakan juga mekanisme tambahan berupa *watchlist* untuk menyimpan sumber serangan dengan risiko tinggi. Adapun rincian lengkap dari cara kerja model dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Gambaran cara kerja model

Model pada Gambar 3.2 bekerja dengan menerapkan pendekatan hibrida dalam mendeteksi serangan siber. Pada tahap awal, komponen *signature* akan membaca pola data pada trafik dan membandingkannya dengan *signature* serangan yang sebelumnya sudah ada. Hasil perbandingan ini berupa file eve.json yang berisi data dari trafik *alert* dan normal. File tersebut kemudian diproses untuk memperbarui *watchlist* dan memisahkan data antara *alert* dan normal. Jika data merupakan *alert,*

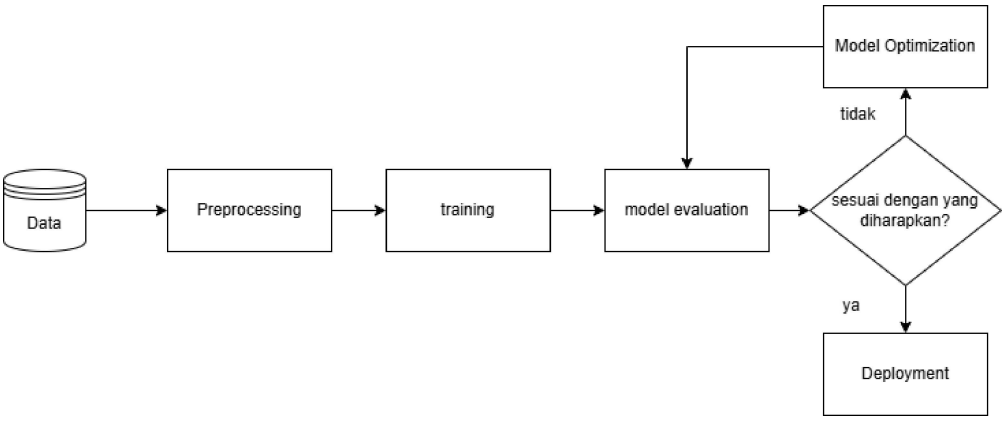
maka data akan disimpan pada sebuah catatan khusus untuk menampung *alert.* Jika data dianggap normal, maka data tersebut akan diproses lebih lanjut oleh komponen deteksi *anomaly* yang dirancang pada Gambar 3.3. Komponen ini akan mengevaluasi trafik yang dialihkan dengan cara membandingkannya dengan pola trafik normal. Apabila pola data pada trafik menyimpang dari perilaku normal, maka trafik tersebut akan dianggap sebagai *anomaly*. *Anomaly* hanya valid ketika *IP* sumbernya merupakan bagian dari *watchlist.* Jika tidak, maka *anomaly* tersebut tidak valid dan tidak akan ada tindakan apa pun. *Anomaly* yang valid kemudian akan dimasukkan ke dalam catatan khusus yang sebelumnya dibuat. Catatan khusus ini berisi data trafik yang memenuhi kondisi *Signature == true OR (Anomaly == true AND IP ∈ Watchlist).* Semua trafik yang memenuhi kondisi akan dianggap sebagai serangan oleh sistem yang dirancang.

3.5.2 Penyesuaian Komponen *Signature*

Langkah penyesuaian komponen *signature* dilakukan untuk mencari dan menganalisis berbagai pola serangan dari trafik jaringan. Pola-pola serangan yang diperoleh kemudian akan disimpan pada *database signature* yang nantinya digunakan untuk mendeteksi serangan dari trafik yang datang. IDS berbasis *signature* yang digunakan dalam penelitian ini adalah Suricata versi 7.0.3. Suricata digunakan karena memiliki akurasi yang cukup baik serta penggunaan sumber daya yang tidak terlalu tinggi (Ernawati dkk., 2019; Fikri dkk., 2024).

3.5.3 Komponen *Anomaly*

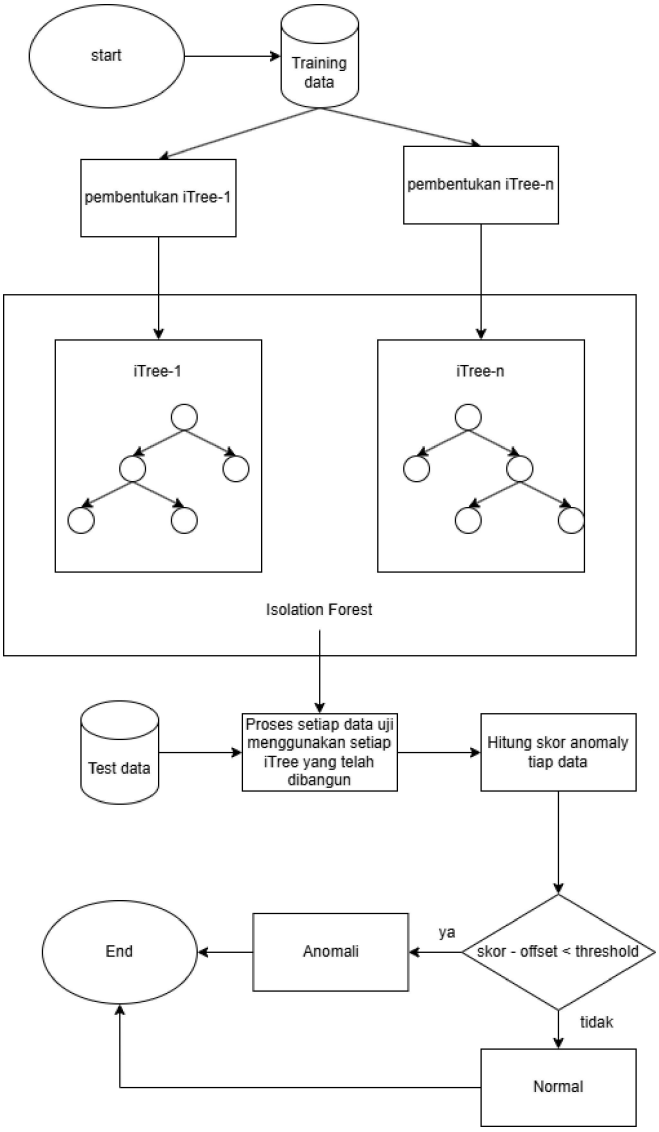
Tahap berikutnya merupakan perancangan model *machine learning* sebagai komponen *anomaly* pada sistem. Adapun langkah dalam merancang model *machine learning* dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Model *machine learning* berbasis *anomaly*

Pada perancangan model *machine learning*, data yang diperoleh pada tahap pengumpulan data akan melalui *preprocessing* terlebih dahulu. *Preprocessing* dilakukan untuk mempersiapkan data, seperti *data cleaning*, pemilihan fitur, dan binarisasi data agar model *machine learning* lebih mudah dalam memproses data tersebut. Setelah data siap untuk diolah, langkah selanjutnya adalah melatih model. Model *machine learning* dilatih hanya menggunakan data trafik normal dengan mengikuti konsep *novelty detection*, sehingga model dapat mempelajari bagaimana trafik normal yang seharusnya dan mengidentifikasi *anomaly* (Ulrich dkk., 2025). Hal ini juga berkaitan dengan IDS, di mana pada skenario nyata trafik normal lebih mudah diperoleh jika dibandingkan dengan trafik serangan yang jarang terjadi (Uddin dkk., 2024). Model yang dilatih di sini adalah *Isolation Forest.* Model ini

dipilih karena efisiensi komputasinya, seperti kebutuhan memori yang rendah serta waktu eksekusi yang relatif singkat. Dengan kemampuan tersebut, *Isolation Forest* mampu menangani *dataset* besar (Cao dkk., 2024; Chabchoub dkk., 2022; Liu dkk., 2008). Selain efisiensi komputasi, model *Isolation Forest* juga baik dalam membedakan antara data normal dan *anomaly* (Soenen dkk., 2021). Cara kerja dari *Isolation Forest* dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Bagan proses *Isolation Forest*

Isolation Forest bekerja dengan cara membentuk sekumpulan *tree* yang disebut *isolation tree* (*iTree)* dari data latih. Pada setiap *tree,* model akan secara acak memilih fitur dan *threshold* sebagai titik pemisah untuk setiap *node.* Kumpulan *tree* yang dibangun ini kemudian akan digunakan untuk memberikan skor *anomaly* pada setiap data yang diproses. *Isolation Forest* akan mencatat panjang *path* yang dilalui oleh setiap data. Rata-rata dari panjang *path* ini kemudian digunakan untuk

menghitung skor *anomaly* pada masing-masing data. Secara teoritis, skor anomali dihitung menggunakan persamaan (1) sampai (3).

$$h(x) = \frac{1}{t}\sum_{i=1}^{t} h_i(x) \quad (1)$$

$$c(n) = 2H(n-1) - \left(\frac{2(n-1)}{n}\right) \quad (2)$$

$$s(x,n) = 2^{\frac{E(h(x))}{c(n)}} \quad (3)$$

Dalam hal ini, *s(x,n)* merupakan skor *anomaly* untuk sampel x dengan ukuran subsampel n; *h(x)* menyatakan panjang jalur sebuah pohon dari *root* sampai akhir; *E(h(x))* merupakan nilai rata-rata dari *h(x)*; $H_{(i)}$ merupakan nilai harmonik yang dapat dihitung dengan *ln(i)* + 0,557; dan $c(n)$ adalah rata-rata dari $h(x)$ untuk nilai *n* tertentu yang digunakan untuk menormalisasi $h(x)$ (Cao dkk., 2024).

Secara teoritis, keputusan dari model *Isolation Forest* didasarkan pada skor *anomaly* seperti yang ditunjukkan pada persamaan (3). Semakin tinggi skor yang didapat, maka semakin tinggi kemungkinan sebuah data merupakan *anomaly.* Berdasarkan literatur, model *Isolation forest* memiliki *threshold default* sebesar 0,5 (Liu dkk., 2008). Pada penelitian ini, keputusan model *Isolation Forest* yang digunakan mengikuti perhitungan yang dimiliki oleh pustaka Scikit-Learn. Berdasarkan pustaka tersebut, keputusan model *Isolation Forest* yang digunakan adalah *decision function* = skor *anomaly – offset.* Pada Scikit-Learn, semakin kecil skor *decision function* yang didapat, maka semakin besar kemungkinan sebuah data merupakan *anomaly.* Keputusan akhir model yang digunakan *Isolation Forest*

adalah *decision function* < *threshold,* yang dimana *threshold default*nya 0 (Pedregosa dkk., 2011).

Setelah model selesai dilatih, langkah selanjutnya adalah evaluasi model *machine learning.* Evaluasi model dilakukan menggunakan data uji yang sebelumnya telah disiapkan. Berdasarkan evaluasi yang dilakukan, jika model belum mampu mendeteksi sebagian besar serangan, maka model akan dioptimasi terlebih dahulu. Ruang pencarian yang digunakan saat optimasi dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Ruang pencarian

Hyperparameter	Nilai Minimum	Nilai Maksimum
n_estimators	150	600
Max_samples	0,01	1,0
Max_features	0,01	1,0
Contamination	0,001	0,5

Dimana *n_estimators* merupakan nilai estimator dasar pada model, *max samples* merupakan jumlah sampel pada data yang digunakan untuk melatih model, *max features* merupakan banyaknya fitur pada data untuk melatih model, dan terakhir *contamination* merupakan jumlah kontaminasi pada data. Semua eksperimen dijalankan dengan *seed* 42 untuk mengunci *randomness* pada model. Berdasarkan optimasi yang dilakukan, hasil terbaik kemudian akan dikombinasikan dengan model IDS berbasis *signature.*

3.6 Implementasi Model

Tahapan selanjutnya yang dilakukan adalah implementasi terhadap model. Tahapan ini dilakukan untuk mewujudkan model yang dirancang pada tahap sebelumnya.

3.7 Evaluasi Model

Langkah terakhir yang dilakukan adalah evaluasi terhadap model. Sebelum evaluasi dilakukan, lima elemen pada bagian *signature* seperti *source IP, source port, destination IP, destination port,* dan *protocol* akan dibandingkan terlebih dahulu dengan *ground truth*. Selain kelima elemen tersebut, *timestamp* pada *signature* dan *ground truth* juga harus cocok satu sama lain. Hal tersebut dilakukan untuk memastikan trafik yang terdeteksi oleh *signature* memiliki pasangan yang sesuai pada *ground truth* (Ficke dkk., 2019). Dataset CIC-IDS2017 memiliki *timestamp* yang sedikit rusak. Oleh karena itu, dilakukan juga perbaikan terhadap *timestamp*. Setelah memastikan trafik memiliki pasangan yang sesuai, selanjutnya evaluasi IDS dilakukan melalui persamaan (4) sampai (8) (Tamara Saad Mohamed & Saad Mohammed Khalifah, 2024).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 - score = 2x \frac{recall\ x\ precision}{recall + precision} \quad (7)$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \quad (8)$$

Dalam hal ini, *true positive* (TP) adalah serangan yang terdeteksi sebagai serangan, *true negative* (TN) adalah normal yang terdeteksi sebagai normal, *false positive* (FP) adalah normal yang terdeteksi sebagai serangan, dan terakhir *false negative* (FN) adalah serangan yang terdeteksi sebagai normal.