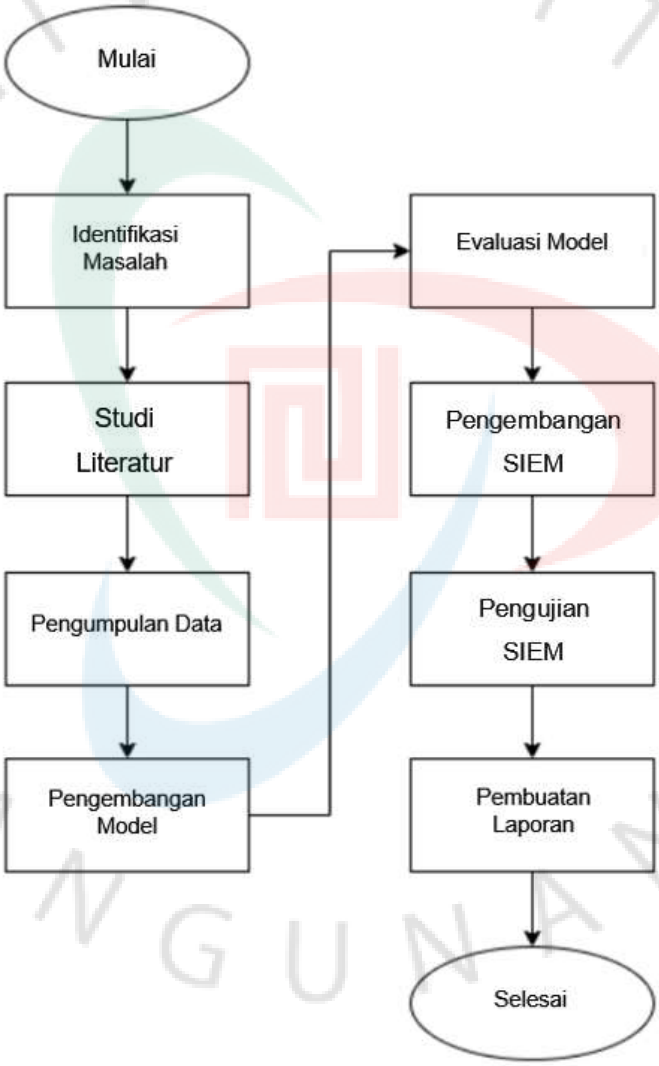
BAB III
TAHAPAN PELAKSANAAN

3.1 Langkah-langkah Pelaksanaan

Proses pelaksanaan ini akan membahas proses penyelesaian tugas akhir. Ini mencakup prosedur pelaksanaan dan teknik pengujian yang digunakan.



Gambar 3.1 Langkah-Langkah Perancangan

Berikut ini adalah penjelasan dari Gambar diatas:

a. Identifikasi Masalah:

Tahap yang berisi apa masalah yang ingin dibahas, mencari permasalahan yang relevan, menentukan tujuan penelitian dan memberikan solusi yang sesuai dengan masalah yang diteliti.

b. Studi Literatur:

Tahap studi literatur dilakukan untuk mencari penelitian terdahulu yang relevan pada penelitian ini, mulai dari penelusuran jurnal, pemilahan jurnal dan sitasi terhadap jurnal yang memiliki latar belakang atau judul yang relevan.

c. Pengumpulan Data:

Proses pengumpulan data terdiri dari pencarian sumber data dan pra-pemrosesan data. Pencarian sumber data yaitu mengumpulkan dataset yang diperlukan pada penelitian ini, yakni adalah kumpulan dataset URL yang diketahui aman dan berbahaya. Pra-pemrosesan data dilakukan setelah selesai mengumpulkan dataset yang diperlukan, kemudian dataset yang dikumpulkan akan dipilah, diekstrasi fitur dan data dinormalisasi.

d. Pengembangan Model :

Pengembangan model yang dilakukan adalah, merancang algoritma *Large Language Model* yang akan digunakan untuk mendeteksi *Malicious* URL dengan bahasa pemrograman Python dan melakukan pelatihan/training menggunakan dataset yang sudah dikumpulkan.

e. Evaluasi Model:

Evaluasi pada pengujian kinerja pada algoritma Large Language Model yaitu; akurasi, presisi dan analisis hasil pengujian untuk menilai efektivitas algoritma-nya dalam melakukan deteksi URL *berbahaya*.

f. Pengembangan *Security Information & Event Management (*SIEM*)*:

Pengembangan *Security Information & Event Management (*SIEM*)* akan dimulai dari mengintegrasikan model *machine learning Large Language Model* yang sudah dilatih untuk melakukan deteksi URL *berbahaya* dan diuji untuk melakukan monitoring pada sistem yang sudah diintegrasikan dengan algoritma *machine learning Large Language Model*.

g. Pengujian *Security Information & Event Management (*SIEM*)*:

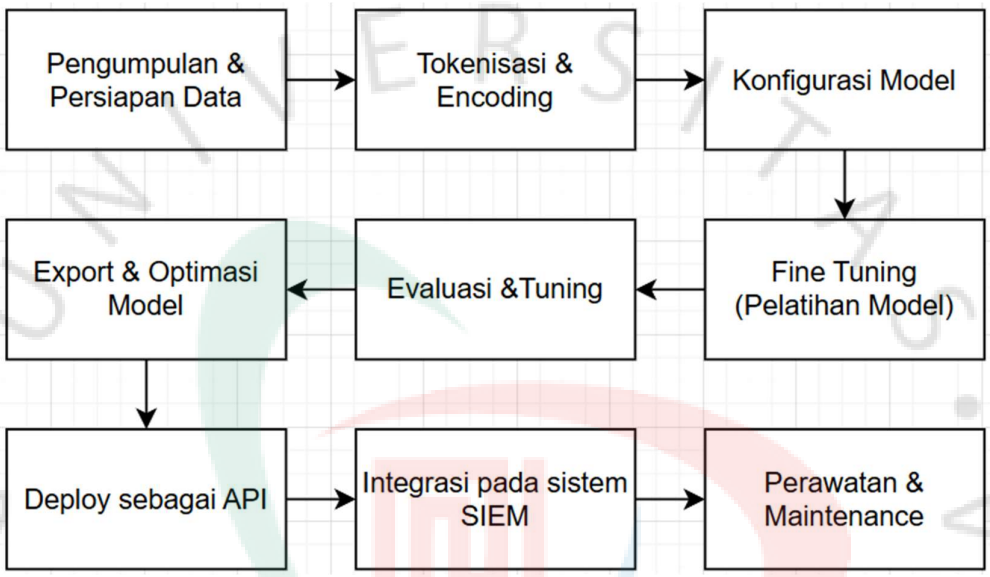
Pengujian *Security Information & Event Management (*SIEM*)* dilakukan dalam skala bertahap, dimulai dari lingkup menggunakan mesin virtual dan komputer pribadi, hingga pengujian pada skala yang lebih besar.

h. Pembuatan Laporan:

Segala aktivitas yang berhubungan pengujian aplikasi, hasil yang dicapai, analisis hingga implementasi aplikasi akan ditulis pada laporan akhir.

3.2 Tahap Penerapan Algoritma

Penerapan Algoritma pada penelitian ini menggunakan algoritma *Large Language Model* untuk melakukan deteksi URL yang berbahaya dengan cara mengelompokkannya berdasarkan karakteristik tertentu. Berikut ini adalah Gambar proses beserta penjelasan untuk mencapai fungsionalitas deteksi URL yang berbahaya:



Gambar 3.2 Tahap Penerapan Algoritma

a) *Pengumpulan & Persiapan Data*:

Langkah pertama adalah mengumpulkan kumpulan data yang berisi daftar URL dan label yang menunjukkan apakah URL tersebut aman (aman) atau *malicious*/*phishing* (berbahaya). Setelah data dikumpulkan, dilakukan proses pembersihan, yang meliputi penghapusan protokol HTTP atau https://, penghapusan spasi, dan pengaturan penulisan yang standar. Label menggunakan format numerik, biasanya 0 untuk *benign* dan 1 untuk *malicious*.

b) Tokenisasi & *Encoding*:

Setelah data selesai, tokenizer, seperti BERT, digunakan untuk mengubah setiap URL menjadi format yang dapat dipahami oleh model

LLM. Proses tokenisasi memecah URL menjadi potongan kecil, yang dikenal sebagai token, yang kemudian diubah menjadi angka, yang dikenal sebagai input ID. Bersamaan dengan itu, *attention mask* dibuat untuk menunjukkan input mana yang harus diproses oleh model.

c) Konfigurasi Model:

Model LLM, seperti BERT untuk klasifikasi, diatur dengan menambahkan lapisan klasifikasi pada bagian akhir. Model ini kemudian disiapkan untuk mengklasifikasikan input menjadi dua kelas (benign atau *malicious*). Hyperparameter seperti *learning rate, batch size,* dan jumlah *epoch* diatur untuk mendukung proses pelatihan yang optimal.

d) *Fine-Tuning* (Pelatihan Model):

Model LLM akan dikonfigurasi dan dilatih menggunakan dataset yang telah disiapkan. Tujuan dari proses ini adalah untuk mengubah bobot internal model berdasarkan pola-pola yang ditemukan dalam URL dan labelnya, sehingga model dapat mengidentifikasi fitur URL yang berbahaya.

e) Evaluasi dan Tuning:

Setelah pelatihan selesai, model dievaluasi menggunakan data uji untuk mengukur kinerjanya dengan metrik seperti akurasi, ketepatan, recall, dan skor F1.

f) Export dan Optimasi Model:

Model yang sudah terlatih disimpan (dieksport) dalam bentuk file dan dapat dioptimalkan untuk kebutuhan pada saat monitoring.

g) Deploy sebagai API

Model yang telah disimpan dijalankan dalam bentuk layanan (API) menggunakan framework Flask. API ini memungkinkan sistem lain untuk mengirimkan URL dan menerima hasil klasifikasi dari model.

h) Integrasi ke dalam sistem SIEM

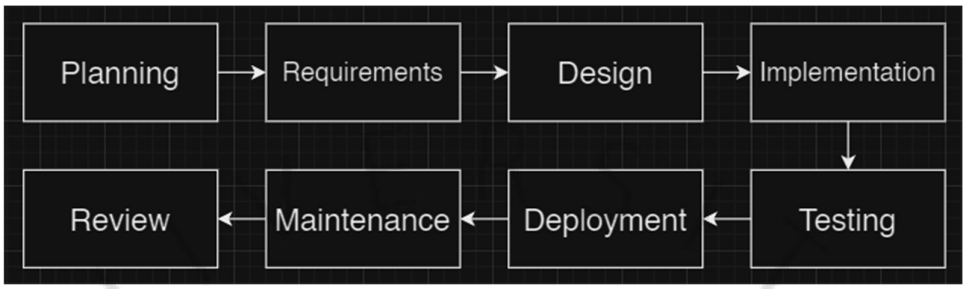
Setelah itu, skrip respons aktif digunakan untuk menghubungkan API prediksi LLM ke sistem Wazuh. Ketika Wazuh menemukan URL yang mencurigakan dalam log, ia memanggil API tersebut dan menerima hasil klasifikasi. Jika URL terindikasi berbahaya, Wazuh dapat melakukan aksi otomatis seperti memblokir URL dan alamat IP.

i) Perawatan dan Maintenance

Setelah sistem berjalan, proses pemantauan dilakukan untuk memastikan model tetap akurat. Ini memungkinkan set data diperbarui secara berkala dengan data terbaru, dan model dilatih ulang untuk menangani jenis serangan baru, yang menghasilkan siklus berkelanjutan untuk menjaga efektivitas deteksi.

3.3 Metode Pengembangan Perangkat Lunak

Tahap pengembangan aplikasi dibuat berdasarkan *Secure Software Development Life Cycle*(SSDLC) untuk penelitian ini, dan akan dijelaskan pada gambar serta penjelasan berikut;



Gambar 3.3 Secure Software Development Life Cycle

a) *Planning*/Perencanaan

Tahap awal untuk perancangan, dimulai dari pelatihan algoritma *Large Language Model* untuk deteksi URL *berbahaya* menggunakan dataset yang sudah diambil, pembuatan topologi jaringan yang akan digunakan beserta lingkupnya, pengembangan dan implementasi algoritma ke dalam aplikasi web yang akan dibuat.

b) *Requirements*/Analisis Kebutuhan

Analisis kebutuhan yang diperlukan adalah pelatihan algoritma *Large Language Model* untuk deteksi URL *berbahaya* dari dataset yang diambil, proses pemilahan dataset yang diuji dan hasil deteksi dari dataset yang duji, merancang topologi jaringan untuk skala pengujian dan merancang aplikasi web yang akan dibuat.

c) *Design*/Desain

Tahap desain dilakukan untuk merancang arsitektur pada aplikasi web yang akan dibuat. Mengintegrasikan algoritma *Large Language Model* sebagai detektor URL *berbahaya*, menggunakan *Security Information &*

*Event Management (*SIEM*)* Wazuh sebagai aplikasi antarmuka untuk menampilkan hasil deteksi.

d) *Implementation*/Implementasi

Implementasi pengembangan *Security Information & Event Management (*SIEM*)* akan menggunakan Wazuh dan algoritma *Large Language Model* yang sudah dilatih. Kemudian model data digunakan untuk menyimpan URL dan hasil analisis yang sudah dilakukan, serta membuat fungsi untuk memproses *input* URL dan menerapkan algoritma *Large Language Model*.

e) *Testing*/Pengujian

Pengujian aplikasi meliputi; *functional testing*/uji fungsionalitas untuk memastikan fitur-fitur yang ada bekerja sesuai spesifikasi. *Performance testing*/uji performa untuk menguji kinerja aplikasi, seperti waktu respons dan penggunaan sumber daya. *Security testing*/Uji keamanan untuk memastikan tidak ada kerentanan terhadap serangan keamanan. *Usability testing*/Uji kemudahan penggunaan aplikasi untuk memastikan antarmuka sudah sesuai kebutuhan dan mudah digunakan.

f) *Deployment*/Penerapan

Tahap penerapan akan dilakukan pada lingkungan, lingkup dan topologi jaringan yang sudah ditentukan, serta melakukan pemantauan untuk memastikan kinerja dan mendeteksi aktivitas yang mencurigakan.

g) *Maintenance*/Pemeliharaan

Pemeliharaan sistem diperlukan untuk memastikan pembaharuan keamanan tetap terjaga untuk menangkal ancaman, pemeliharaan secara berkala untuk memastikan kinerja tetap optimal dan respon insiden keamanan untuk melakukan pemulihan jika ada terjadi penyerangan.

h) *Review*/Tinjauan

Tinjauan dilakukan secara berkala, untuk memastikan agar semua prosedur dapat diikuti dan lebih mudah untuk melakukan perbaikan jika ada masalah.

3.4 Metode Pengujian

Untuk memastikan kinerja sistem sudah optimal, akan dilakukan pengujian sistem aplikasi web untuk mencegah akses URL *berbahaya* menggunakan algoritma *Large Language Model* menggunakan dua metode pengujian; pertama, pembuatan skenario topologi jaringan dan pengembangan *Security Information & Event Management* (SIEM) yakni yang digunakan adalah Wazuh. Berikut adalah penjelasan tentang metode pengujian yang digunakan.

3.4.1 Pengujian Skenario Topologi Jaringan

Pengujian skenario topologi jaringan dilakukan secara real-time. Pada topologi jaringan yang digunakan, prerangkat terdiri dari laptop/komputer fisik dengan perangkat lunak VirtualBox, yang berisi mesin virtual Server dengan *Security Information & Event Management* (SIEM) dan Client fisik berupa komputer/laptop.

Gambar 3.4 Topologi Jaringan Pengujian

3.4.2 **Pengujian** SIEM

Pengujian *Security Information & Event Management* (SIEM) dilakukan menggunakan metode white box dan metode black box, untuk memastikan tidak ada kendala fungsionalitas.

3.4.2.1 White Box

Metode White Box diterapkan untuk memeriksa dan menguji logika internal serta struktur kode algoritma yang membentuk sistem deteksi dan pencegahan akses URL berbahaya. Pengujian ini dilakukan dengan menganalisis kode dari model LLM, API Flask yang digunakan untuk menerima dan memproses URL, serta skrip Active Response yang ada di Wazuh. Dengan metode ini, penguji dapat memastikan bahwa setiap fungsi dalam proses pelatihan model, prediksi URL, hingga pemicu tindakan pemblokiran oleh Wazuh berjalan sesuai dengan rancangan. Selain itu, pengujian White Box juga memungkinkan identifikasi kesalahan logika, bug, atau alur komunikasi yang tidak efisien antara komponen-komponen sistem, termasuk integrasi antara Wazuh dan model deteksi berbasis LLM.

3.4.2.2 Black Box

Metode Black Box dilakukan dengan fokus pada evaluasi fungsionalitas sistem secara menyeluruh dari perspektif pengguna eksternal tanpa melihat struktur internal sistem. Pengujian dilakukan dengan cara mengirimkan berbagai URL secara langsung dari sisi client, baik URL benign, *phishing*, maupun *malicious*, untuk mengamati bagaimana sistem merespons input tersebut. Tujuan dari pengujian ini untuk memastikan bahwa sistem mampu mendeteksi ancaman dengan benar, mengklasifikasikannya melalui LLM, serta menjalankan Active Response secara otomatis jika URL terdeteksi sebagai berbahaya. Pengujian ini juga mencakup pengujian keamanan dan performa, seperti bagaimana sistem menangani akses dalam kondisi lalu lintas jaringan yang tinggi atau ketika menerima input URL yang tidak biasa (misalnya dengan encoding atau domain baru). Hasil dari pengujian ini digunakan untuk menilai apakah sistem telah memenuhi kebutuhan fungsional dan keamanan dari sisi pengguna.