|  |  |  |
| --- | --- | --- |
| Topik Capstone | **Topik Capstone** | |
| Siklus / Tahun | **\*Gasal atau Genap / (tahun)** | |
| Judul Dokumen | **Capstone TA**  Judul Capstone Proyek kelompok | |
| Jenis Dokumen | **PERANCANGAN PRODUK**  Catatan: Penggunaan dan penyebaran dokumen ini dikendalikan oleh Departemen Teknik Komputer Universitas Diponegoro | |
| Nomor Dokumen | **C300.[NoRev]TA[tahun].[1/2].[KodeKelompok]** | |
| Nomor Revisi | **NoRev** | |
| Nama File | **Kode Kelompok.pdf** | |
| Tanggal Penerbitan | **Tanggal Penerbitan** | |
| Unit Penerbit | **Departemen Teknik Komputer Universitas Diponegoro** | |
| Jumlah Halaman | **Jumlah Halaman** | Tidak termasuk sampul |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Pengusul** | | | | |
| Pengusul | Nama |  | Jabatan | Anggota |
|  | NIM |  |  |  |
|  | Tanggal |  | Tanda Tangan |  |
|  | Nama |  | Jabatan | Anggota |
|  | NIM |  |  |  |
|  | Tanggal |  | Tanda Tangan |  |
|  | Nama |  | Jabatan | Anggota |
|  | NIM |  |  |  |
|  | Tanggal |  | Tanda Tangan |  |
| Pembimbing 1 (Utama) | Nama |  | Tanda Tangan |  |
|  | NIP. |  |  |  |
|  | Tanggal |  |  |  |
| Pembimbing 2 | Nama |  | Tanda Tangan |  |
|  | NIP. |  |  |  |
|  | Tanggal |  |  |  |

|  |  |
| --- | --- |
| Versi, Tanggal, Oleh | Perbaikan |
|  |  |

**Daftar Isi**

[1. Pendahuluan 4](#_Toc188012726)

[1.1. Ringkasan isi dokumen 4](#_Toc188012727)

[1.2. Aplikasi Dokumen 4](#_Toc188012728)

[1.3. Referensi 4](#_Toc188012729)

[1.4. Daftar Singkatan 4](#_Toc188012730)

[2. Desain Produk yang Diusulkan 4](#_Toc188012731)

[2.1. Arsitektur Sistem 4](#_Toc188012732)

[2.2. Desain Detail Sistem 6](#_Toc188012733)

[2.2.1. Data description 6](#_Toc188012734)

[2.2.2. Class Diagram/Sequence Diagram atau diagram yang sesuai 7](#_Toc188012735)

[2.2.3. Standar-standar yang dipergunakan 10](#_Toc188012736)

[2.2.4. Method/API 11](#_Toc188012737)

[2.2.5. Kecerdasan Buatan 12](#_Toc188012738)

[2.2.6. Perancangan Alat 14](#_Toc188012739)

[3. Verifikasi Desain Produk 15](#_Toc188012740)

[3.1. Prototipe, atau 15](#_Toc188012741)

[3.2. Hasil Simulasi Awal Produk 15](#_Toc188012742)

# Pendahuluan

## Ringkasan isi dokumen

*Isi Ringkasan Dokumen*

## Aplikasi Dokumen

*Dokumen ini berlaku berfungsi untuk menjelaskan:*

1. *Proses pemilihan desain alat dari beberapa alternatif yang ada.*
2. *Detail desain alat dari level tertinggi sampai terendah*
3. *Menjelaskan standar-standar yang dipergunakan*
4. *Refensi komponen/library yang digunakan*
5. *Verifikasi bahwa hasil rancangan dapat diaplikasikan*
6. *Rencana implementasi dan pengujian*

## Referensi

## Daftar Singkatan

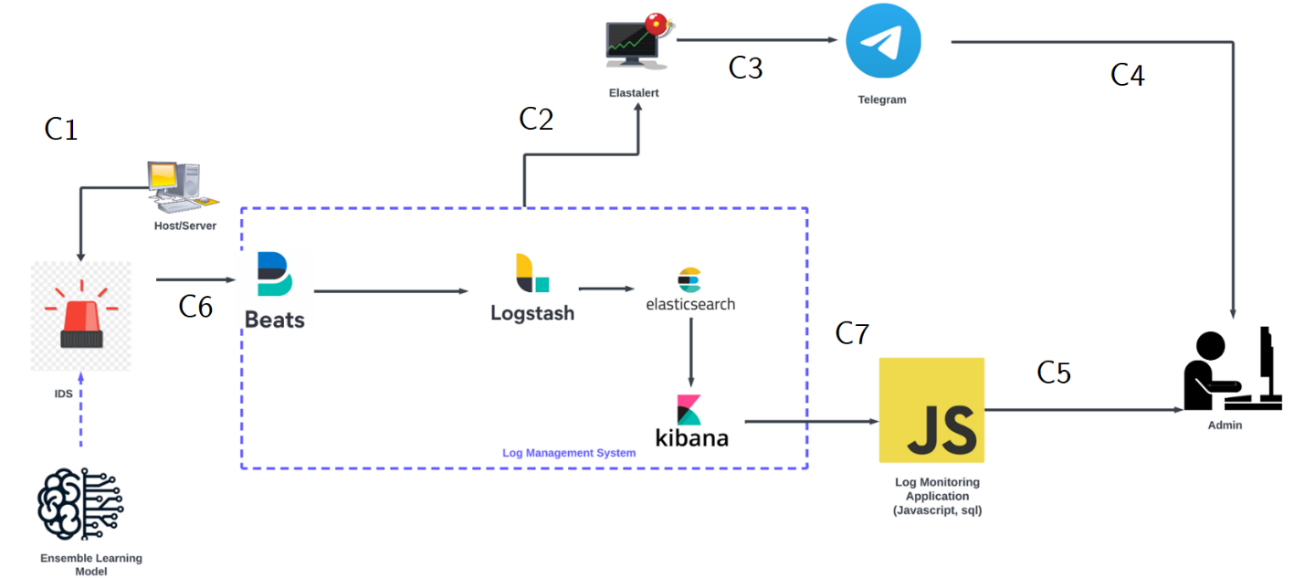
# Desain Produk yang Diusulkan

## Arsitektur Sistem

*Jelaskan arsitektur dari sistem yang diusulkan, yang menunjukkan hubungan antar komponen utama.*

**Contoh:**

Sistem yang kami usulkan yaitu LunaSystem: Sistem Monitoring dan Deteksi Serangan Slow DoS/DRDoS Berbasiskan Ensemble Learning menggunakan ELK Stack. Sistem monitoring ini memiliki fungsi utama, yaitu untuk mendeteksi beberapa jenis serangan Slow DoS/DRDoS. Dapat dilihat pada Gambar 3.1. bahwa arsitektur Luna System menggunakan beberapa perangkat lunak Elasticsearch, Logstash, dan Kibana (ELK Stack) untuk membantu keefektifan sistem melakukan monitoring dan mendeteksi serangan Slow DoS/DRDoS. Setiap perangkat lunak memiliki fungsinya masing-masing dalam pembangunan sistem ini.



Gambar 3.1. Arsitektur Luna System

Berikut cara kerja LunaSystem : Sistem Monitoring dan Deteksi Serangan Slow DoS/DRDoS Berbasiskan *Ensemble Learning* menggunakan ELK Stack:

* 1. TCPdump melakukan penangkapan trafik jaringan pada host/server kemudian menyimpannya dalam berkas pcap
  2. Berkas pcap diubah ke berkas csv
  3. Dilakukan ekstraksi fitur terhadap berkas pcap yang sudah diubah ke bentuk berkas csv berdasarkan format dari dataset yang digunakan.
  4. Dilakukan prediksi menggunakan model pembelajaran ensemble yang sudah dikembangkan dan hasil prediksi disimpan dengan nama log.csv
  5. Hasil prediksi dikirimkan oleh filebeat ke elasticsearch
  6. Dibuatkan dashboard pada kibana berdasarkan hasil prediksi yang diperoleh
  7. Dashboard pada kibana disimpan pada log monitoring application dengan menggunakan iframe
  8. ElastAlert mengirimkan notifikasi telegram apabila serangan Slow DoS/DRDoS terjadi.

Identifikasikan juga bagaimana komunikasi antar komponen.

**Contoh**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kode** | **Komponen Pengirim** | **Komponen Penerima** | **Media Transmisi** | **Metode Transmisi Data** |
| C1 | Server | IDS | 4G | HTTP Post |
| … | … | … | … | … |
| C3 | Gateway | Server cloud | 4G | HTTP Post |
| C6 | Server Cloud | Client Browser | Internet | API |

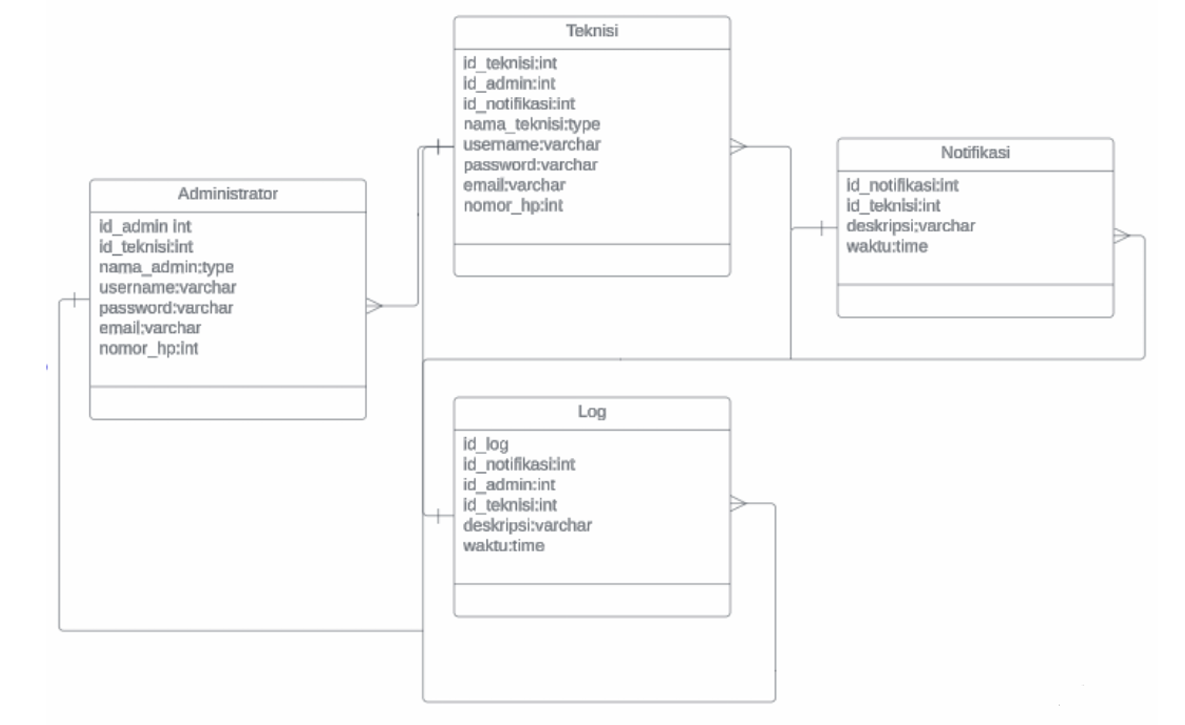
## Desain Detail Sistem

*Jika ada jelaskan tentang poin-poin berikut:*

### **Data description**

*Jelaskan Physical Data Model dan tabel-tabel dalam database beserta definisi domain/tipe data dan struktur tabel. Tunjukkan hubungan tabel-tabel dalam database dengan desain ER yang telah dibuat.*

**Contoh**:



Gambar 3.1. Entity Relationship Diagram

Struktur tabel yang digunakan dalam basis data Luna *System* dijelaskan pada Tabel 3.2 hingga Tabel 3.3. *Dataset* yang digunakan dijelaskan pada Tabel 3.4 hingga Tabel 3.5.

Tabel 3.2 Struktur Tabel *Users*

| **Kolom** | **Tipe Data**  **(Panjang Data)** | **Bawaan** | **Keterangan** |
| --- | --- | --- | --- |
| id | bigint(20) |  | *Primary Key, Auto Increament* |
| nama | VARCHAR |  |  |
| password | VARCHAR | *NULL* |  |
| created\_at | datetime | NOW0 |  |
| updated\_at | datetime | NOW0 |  |

Tabel 3.3 user\_activities

| **Kolom** | **Tipe Data**  **(Panjang Data)** | **Bawaan** | **Keterangan** |
| --- | --- | --- | --- |
| id | * + - 1. bigint(20) |  | *Primary Key, Auto Increament* |
| src\_ip | VARCHAR |  |  |
| src\_port | VARCHAR |  |  |
| dst\_ip | VARCHAR |  |  |
| dst\_port | VARCHAR |  |  |
| result | text |  |  |
| category | text |  |  |

Hubungan tabel-tabel dalam basis data dengan desain ERD yang telah dibuat dapat dilihat pada Tabel 3.4.

Tabel 3.4 Tabel *Traceable Data*

| **Nama Tabel** | ***Primary Key*** | ***Entity Class*** | ***ER*** | ***Deskripsi Isi*** |
| --- | --- | --- | --- | --- |
| users | id | User | Need | Berisi daftar data pengguna. |
| user\_activities | id | UserActivity | Need | Berisi daftar rekam jejak aktivitas pengguna dalam sistem. |

## Class Diagram/Sequence Diagram atau diagram yang sesuai

*Buat diagram-diagram yang diperlukan untuk menjelaskan sistem yang dibuat, misal class/sequence diagram, state diagram, flow chart, dsb).*

**Contoh**:

Gambar 3.2 menggambarkan struktur sistem dan hubungan antara komponen-komponen utama dalam lunasystem. Berikut merupakan penjelasan komponen-komponen utama dari lunasystem:

1. TCPdump

Fungsi:

* Alat untuk menangkap paket data dari jaringan.

Metode Utama:

* capture(): Menangkap paket data yang melintas di jaringan.
* saveasPCAP(): Menyimpan paket data yang ditangkap dalam format file PCAP.

Hubungan:

* Menghasilkan file PCAP yang akan digunakan oleh komponen berikutnya dalam sistem.

1. CICFlowMeter

Fungsi:

* Alat untuk mengkonversi file PCAP menjadi file CSV.

Metode Utama:

* convertToCSV(): Mengubah file PCAP menjadi file CSV dengan fitur-fitur alur data.

Hubungan:

* Menerima file PCAP dari TCPDUMP dan menghasilkan file CSV

1. luna.service

Fungsi:

* Layanan yang memantau dan menganalisis file CSV.

Metode Utama:

* monitorCSV(): Memantau file CSV untuk deteksi file baru.
* predict(): Melakukan prediksi terhadap data dalam file CSV.
* saveLog(): Menyimpan hasil prediksi dalam file log.csv.

Hubungan:

* Memantau file CSV yang dihasilkan CICFlowMeter dan menghasilkan file log.csv dengan hasil prediksi.

1. ELK Stack

Fungsi:

* Sistem yang mengumpulkan, menyimpan, dan memproses data log.

Hubungan:

* Menerima file log.csv dari luna.service dan memprosesnya sebelum mengirimkan data ke aplikasi pemantauan log.

1. Log Monitoring Application

Fungsi:

* Aplikasi yang menampilkan data log yang telah diproses oleh ELK Stack.

Hubungan:

Menerima data dari ELK Stack dan menampilkan informasi hasil prediksi kepada pengguna.

A black and white diagram

Description automatically generated

Gambar 3.2 *Class Diagram* LunaSystem

### **Standar-standar yang dipergunakan**

*Jelaskan standar yang dipergunakan, misal protokol komunikasi data, sistem security, dsb.*

**Contoh**:

Dalam implementasinya, Luna System menggunakan berbagai macam standar untuk bekerja. Standar-standar tersebut adalah:

1. C1: Koneksi HTTP (Hypertext Transfer Protocol) dari Klien ke Server untuk menginisiasi permintaan dan menerima respons.
2. C2: Koneksi HTTPS (Hypertext Transfer Protocol Secure) dari Server ke Filebeat untuk transmisi data log yang aman.
3. C3: Koneksi Filebeat dari Filebeat ke Logstash untuk mengirim data log.
4. C4: Koneksi Logstash dari Logstash ke Elasticsearch untuk melakukan parsing dan pengindeksan data log.
5. C5: Koneksi Elasticsearch dari Elasticsearch ke Kibana untuk visualisasi data.
6. C6: Koneksi Elasticsearch dari Elasticsearch ke Elastalert untuk deteksi anomali.
7. C7: Notifikasi yang dikirim dari Elastalert ke Pengguna ketika anomali terdeteksi.

### **Method/API**

*Jika ada, jelaskan secara detail proses CRUD dan data Query, termasuk yang berhubungan dengan komunikasi antar komponen di Section 3.1 (kode komunikasi C3, C6). Termasuk konfigurasi server bila ada (misal load balancing, dsb).*

**Contoh**:

Luna System memiliki beberapa *method* yang dibuat untuk memenuhi kebutuhan fungsional sistem yang dapat dilihat pada Tabel 3.6.

Tabel 3.6 Deskripsi Method/API

| **SRS-Id** | **Nama *Method*** | ***Keterangan*** |
| --- | --- | --- |
| LUNA-01 | capture() | *Script* melakukan penangkapan paket data yang melintas di jaringan |
| LUNA-02 | saveasPCAP() | *Script* menyimpan paket data yang ditangkap dalam format file PCAP |
| LUNA-03 | convertToCSV() | *Script* mengubah file PCAP menjadi file CSV dengan fitur-fitur alur data |
| LUNA-04 | monitorCSV() | *Script* memantau file CSV untuk deteksi file baru. |
| LUNA-05 | predict() | *Script* melakukan prediksi terhadap data dalam file CSV |
| LUNA-06 | saveLog() | Berfungsi untuk menyimpan hasil prediksi dalam file log.csv |
| LUNA-07 | login() | Admin melakukan autentikasi ke dalam sistem. |
| LUNA-08 | getDashboard() | Admin mendapatkan data ringkasan yang ditampilkan dalam bentuk grafik dan data tabular pada halaman *dashboard*. |
| LUNA-09 | getAlert() | Admin mendapatkan peringatan melalui aplikasi telegram |
| LUNA-10 | logout() | Admin mengakhiri sesi *login* admin yang aktif, sehingga memastikan admin keluar dari sistem dengan aman |

### **Kecerdasan Buatan**

Apabila terdapat implementasi kecerdasan buatan, tuliskan secara spesifik algoritma apa saja yang akan dijadikan perbandingan untuk pada akhirnya diimplementasikan di sistem. Jelaskan secara spesifik *hyperparameter* apa saja yang akan digunakan. Apabila algoritma berbasis pembelajaran (machine learning atau deep learning), jelaskan data yang digunakan untuk melakukan *training, validation,* dan *testing.*

**Contoh**:

* + - 1. **Data Pelatihan**

Berikut adalah daftar dataset yang digunakan dalam implementasi sistem.Tabel 3.4 berisi dataset CIC-IDS2017 dan 2018, sedangkan Tabel 3.5 berisi deskripsi 2019.

Tabel 3.4 *Dataset* CIC-IDS2017 dan CSE-CIC-IDS2018

|  |  |
| --- | --- |
| **Jenis Serangan** | **Jumlah** |
| *Benign* | 1,596,744 |
| DoS attacks-Slowloris | 15,293 |
| DoS attacks-Slowhttptest | 5,283 |

Tabel 3.5 Deskripsi *Dataset* CIC-DDoS2019

|  |  |
| --- | --- |
| Jenis DRDoS | Jumlah |
| DRDoS DNS | 108,119 |
| DRDoS LDAP | 28,871 |
| DRDoS MSSQL | 193,346 |
| DRDoS NetBIOS | 17,925 |
| DRDoS NTP | 111,269 |
| DRDoS SNMP | 112,061 |
| DRDoS SSDP | 890,292 |

* + - 1. **Model**

Untuk implementasi model pembelajaran mesin berbasis ensemble, kami menggunakan metode Random Forest. Random Forest adalah salah satu metode ensemble yang populer dan efektif, yang mengkombinasikan hasil dari beberapa pohon keputusan (decision trees) untuk meningkatkan akurasi dan mengurangi overfitting.

**Parameter Model**

Berikut adalah parameter-parameter yang digunakan dalam model Random Forest:

1. **n\_estimators**: Jumlah pohon keputusan dalam hutan. Pada penelitian ini, kami menggunakan 100 pohon untuk mendapatkan keseimbangan antara kinerja dan waktu komputasi.
2. **max\_depth**: Kedalaman maksimum pohon keputusan. Kami membatasi kedalaman pohon pada 10 untuk mencegah overfitting.
3. **min\_samples\_split**: Jumlah minimum sampel yang dibutuhkan untuk memisahkan suatu node. Nilai yang digunakan adalah 2.
4. **min\_samples\_leaf**: Jumlah minimum sampel yang dibutuhkan pada setiap daun pohon. Nilai yang digunakan adalah 1.
5. **criterion**: Fungsi yang digunakan untuk mengukur kualitas pemisahan. Kami menggunakan "gini" untuk indeks Gini.
6. **bootstrap**: Apakah akan menggunakan bootstrap samples saat membangun pohon. Kami menggunakan nilai True.

### **Perancangan Alat**

*Dalam bagian ini, jelaskan secara detail desain dan pengujian perangkat keras yang digunakan dalam proyek.*

* + - 1. **Diagram Perangkat Keras**

*Buat diagram yang menunjukkan komponen-komponen utama dalam sistem perangkat keras, termasuk hubungan antar komponen. Diagram-diagram yang bisa disertakan antara lain:*

1. *Diagram Blok: Menunjukkan fungsi utama dari setiap komponen dan bagaimana mereka terhubung satu sama lain.*
2. *Diagram Skematik: Diagram yang lebih rinci yang menunjukkan semua koneksi listrik antara komponen.*
3. *Diagram PCB (Printed Circuit Board): Diagram layout fisik dari komponen pada papan sirkuit.*
   * + 1. **Deskripsi Komponen**

*Deskripsikan setiap komponen yang digunakan dalam sistem, termasuk spesifikasi teknis dan peran mereka dalam sistem. Contoh:*

1. *Mikrokontroler: Spesifikasi teknis dan fungsi dalam sistem.*
2. *Sensor: Jenis sensor, data yang diukur, dan cara kerja.*
3. *Aktuator: Jenis aktuator dan fungsinya dalam sistem.*
   * + 1. **Pengujian Perangkat Keras**

*Jelaskan metode pengujian perangkat keras yang akan dilakukan untuk memastikan sistem bekerja sesuai dengan spesifikasi. Ini dapat mencakup:*

1. *Pengujian Fungsional: Menguji apakah setiap komponen bekerja sesuai dengan yang diharapkan.*
2. *Pengujian Integrasi: Menguji bagaimana komponen bekerja bersama sebagai sebuah sistem.*
3. *Pengujian Kinerja: Mengukur kinerja sistem dalam kondisi operasi yang berbeda.*
   * + 1. **Prosedur Kalibrasi**

*Jelaskan prosedur kalibrasi yang diperlukan untuk memastikan akurasi dan konsistensi data yang dihasilkan oleh sistem.*

# Verifikasi Desain Produk

*Pilih yang digunakan dalam pengembangan produk:*

## Prototipe, atau

## Hasil Simulasi Awal Produk

*Boleh dalam bentuk referensi, simulator atau pengujian modular. Tunjukkan hasil simulasi awal desain produk yang dikembangkan. Contoh, keberhasilan pengiriman data dari mobile phone ke server di cloud, simulator packet tracer, simulasi di matlab, dll.*

# Rencana Implementasi dan Pengujian

## Gaant Chart

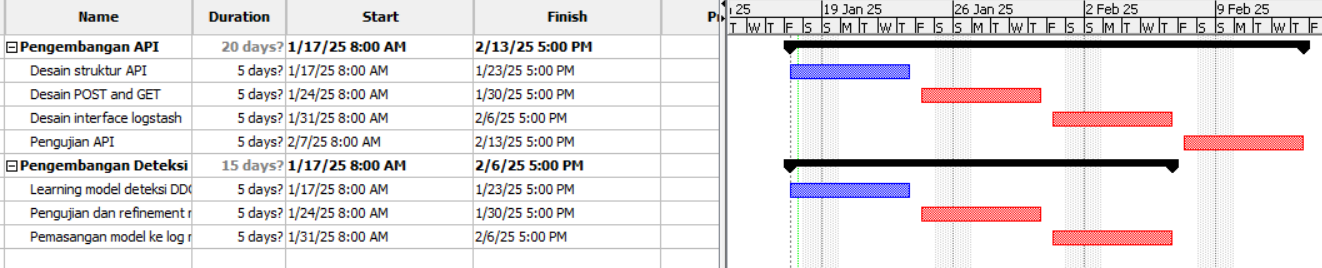
*Harus menggunakan Software Project Libre untuk menunjukkan gambaran detil tahapan implementasi dan pengujian yang akan dilakukan. Gantt Chart harus mencakup:*

1. *Fase Perencanaan: Menentukan tujuan dan cakupan proyek, pemilihan alat dan teknologi, dan pembuatan jadwal kerja.*
2. *Fase Desain: Desain sistem secara rinci, pembuatan diagram perangkat keras dan perangkat lunak, serta persiapan data untuk pengujian.*
3. *Fase Implementasi: Konstruksi, integrasi komponen, dan pengujian unit.*
4. *Fase Pengujian: Pengujian sistem secara keseluruhan, validasi hasil pengujian, dan perbaikan bug.*
5. *Fase Peluncuran: Implementasi akhir, pelatihan pengguna, dan evaluasi proyek.*

**Contoh:**

*Catatan: Contoh ini harus dikembangkan lagi supaya mencakup dependensi antar sistem yang dikembangan serta kapan komponen tersebut dikerjakan.*

Tabel 5. 1 Gantt Chart jadwal pengerjaan



## Tujuan Pengujian

## Lingkup Pengujian

*Lingkup pengujian mencakup semua aspek dari sistem yang dikembangkan, termasuk perangkat keras dan perangkat lunak.*

## Metode Pengujian

*Menunjukkan metode pengujian yang akan digunakan ketika produk jadi.*

**Contoh**:

Metode pengujian yang akan digunakan ketika produk jadi meliputi:

1. Pengujian Unit:

Pengujian individual untuk setiap komponen, seperti memastikan Filebeat dapat mengirim data log dengan benar, Logstash dapat mem-parsing data dengan benar, dan Kibana dapat menampilkan visualisasi data yang benar.

1. Pengujian Integrasi:

Menguji bagaimana komponen-komponen seperti Filebeat, Logstash, Elasticsearch, Kibana, dan Elastalert bekerja bersama sebagai sebuah sistem. Contoh: Memastikan Filebeat dapat mengirim data log ke Logstash, Logstash dapat mem-parsing data dan mengirimkannya ke Elasticsearch, dan data tersebut dapat divisualisasikan di Kibana.

1. Pengujian Sistem:

Menguji sistem secara keseluruhan untuk memastikan bahwa semua fungsi bekerja sesuai dengan persyaratan. Contoh: Memastikan sistem mampu mendeteksi dan memberikan notifikasi ketika terdeteksi ancaman DoS/DDoS.

1. Pengujian Kinerja:

Menguji kinerja sistem di bawah berbagai kondisi operasi, termasuk beban tinggi. Contoh: Memastikan sistem tetap responsif ketika menerima volume log yang tinggi selama serangan DoS/DDoS.

1. Pengujian Keamanan:

Mengidentifikasi dan memperbaiki kerentanan keamanan dalam sistem. Contoh: Memastikan data yang ditransmisikan melalui HTTPS (C2) aman dan tidak dapat disadap.

1. Pengujian Kompatibilitas:

Menguji kompatibilitas sistem dengan perangkat dan lingkungan lain. Contoh: Memastikan sistem dapat beroperasi dengan baik pada berbagai server.

## Kriteria Pengujian

*Kriteria pengujian digunakan untuk menentukan apakah sistem telah lulus pengujian atau tidak.*

1. *Fungsi:*

*Sistem harus memenuhi semua persyaratan fungsional yang telah ditetapkan. Contoh: Sistem mampu mendeteksi dan memberikan notifikasi ancaman DoS/DDoS.*

1. *Kinerja:*

*Sistem harus mencapai kinerja yang diharapkan dalam hal kecepatan, responsivitas, dan efisiensi. Contoh: Sistem tetap responsif di bawah beban log yang tinggi.*

1. *Keamanan:*

*Sistem harus bebas dari kerentanan yang dapat dieksploitasi. Contoh: Data yang ditransmisikan menggunakan HTTPS harus aman dari penyadapan.*

1. *Keandalan:*

*Sistem harus bekerja dengan stabil tanpa kesalahan atau gangguan selama periode pengujian. Contoh: Sistem tidak mengalami downtime selama pengujian intensif.*