

BUKU PEDOMAN PELAKSANAAN CAPSTONE (REV 1)

Departemen Teknik Komputer
2025



Universitas Diponegoro

Daftar Isi

Daftar Isi	2
BAB I - Penjelasan	3
1. Latar Belakang	3
2. Pengertian.....	3
3. Ruang Lingkup	3
BAB II - Tata Cara Pelaksanaan	4
1. Alur Pelaksanaan	4
2. Jadwal Pelaksanaan	5
3. Syarat.....	5
4. Topik Proyek Desain Capstone	6
5. Luaran Proyek Desain Capstone	6
6. Kelas Proyek Desain Capstone 1	6
7. Kelas Proyek Desain Capstone 2	8
BAB III - Pedoman Penulisan	10
1. C100	10
2. C200	34
3. C300	55
4. C400	70
5. C500	87
Lampiran Dokumen.....	102

BAB I – Penjelasan

1. Latar Belakang

Proyek Desain Capstone merupakan mata kuliah wajib yang ada di Kurikulum Departemen Teknik Komputer yang terdiri dari dua bagian, yaitu Proyek Desain Capstone 1 yang ditawarkan pada Semester 6 dan Proyek Desain Capstone 2 pada Semester 7. Berdasarkan kompetensi yang ada pada IABEE dan Atribut Lulusan dan Kompetensi Profesional yang ada, mahasiswa diharapkan memiliki kompetensi untuk menyelesaikan suatu permasalahan kompleks dalam bidang Teknik. Kompetensi yang diharapkan ini yang digunakan sebagai landasan untuk seluruh mahasiswa Departemen Teknik Komputer untuk wajib mengikuti dan menyelesaikan mata kuliah Proyek Desain Capstone.

2. Pengertian

Proyek Desain Capstone didesain sebagai suatu wadah mahasiswa dalam memenuhi kompetensi penyelesaian permasalahan kompleks yang ada di dalam bidang keteknikan. Kriteria kompleks yang ada diacu dari Atribut Lulusan dan Kompetensi Profesional, yang terdiri dari:

- KR 1 Tidak dapat diselesaikan tanpa pengetahuan teknik yang mendalam pada tingkat satu atau lebih P1 (dasar teknik), P2 (pengetahuan spesialis), P3 (desain dan operasi teknik), P4 (praktik teknik), P5 (literatur penelitian) yang memungkinkan pendekatan analitis prinsip pertama berbasis fundamental.
- KR 2 Melibatkan isu-isu teknis dan non-teknis yang luas dan/atau bertentangan (seperti etika, keberlanjutan, hukum, politik, ekonomi, sosial) dan pertimbangan persyaratan masa depan.
- KR 3 Melibatkan kolaborasi lintas disiplin teknik, bidang lain, dan/atau beragam kelompok pemangku kepentingan dengan kebutuhan yang sangat bervariasi.
- KR 4 Mengatasi masalah tingkat tinggi dengan banyak komponen atau sub-masalah yang mungkin memerlukan pendekatan sistem.

Note:

- KR = Kriteria
- P = Parameter

Kriteria kompleks yang ada diacu dari Atribut Lulusan dan Kompetensi Profesional yang sudah disebutkan digunakan sebagai acuan parameter kualifikasi mahasiswa Departemen Teknik Komputer dalam menyelesaikan Proyek Desain Capstone.

3. Ruang Lingkup

Ruang lingkup dari Proyek Desain Capstone adalah batasan pengerjaan mahasiswa dalam menyelesaikan suatu permasalahan kompleks yang dimiliki oleh *stakeholder* di dalam mata kuliah Proyek Desain Capstone 1 dan Proyek Desain Capstone 2. Permasalahan yang

dihadapi oleh stakeholder yang dapat dipilih oleh mahasiswa harus memenuhi Kriteria 1-4 yang telah disebutkan. Kelompok Proyek Desain Capstone yang terbentuk harus terdiri dari minimal 3 mahasiswa yang memiliki minimal 2 kepeminatan yang berbeda.

BAB II – Tata Cara Pelaksanaan

1. Alur Pelaksanaan

Sesuai dengan alur yang dapat dilihat pada website Capstone Departemen Teknik Komputer ([Tata Cara dan Unduh Formulir Proyek Desain Capstone – Capstone TA Teknik Komputer](#)), alur dari pelaksanaan Proyek Desain Capstone terdiri dari beberapa tahapan, yaitu:

- **Pendaftaran Judul Proyek Desain Capstone dan Pencarian atau Pembagian Dosen Pembimbing**

Dalam tahapan ini mahasiswa membentuk kelompok untuk menentukan judul Proyek Desain Capstone yang akan dikerjakan dan memilih Dosen Pembimbing. Jika mahasiswa tidak memiliki judul sesuai dengan batas waktu yang diberikan, mahasiswa tetap melakukan pendaftaran dan Koordinator Proyek Desain Capstone akan mengelompokkan dan mengalokasikan judul serta Dosen Pembimbing untuk mahasiswa yang sudah mendaftar. Selain itu, mahasiswa juga bisa melakukan pembentukan tim dan pemilihan judul Proyek Desain Capstone melalui mekanisme:

- Proyek yang diberikan oleh Prodi
- Tawaran Capstone Universitas/Fakultas/Departemen lain

Bagi mahasiswa yang memilih dari 2 alur yang ada pada opsi lain, maka bisa mengikuti alur pengerjaan Proyek Desain Capstone sesuai dengan proyeknya masing-masing dan nanti akan dikonversi menyesuaikan dengan mekanisme Proyek Desain Capstone 1 dan Proyek Desain Capstone 2 yang ada di Departemen Teknik Komputer.

Dalam proses ini, mahasiswa bisa mengisi data pada Google Form yang diberikan oleh Koordinator Proyek Desain Capstone pada Website Proyek Desain Capstone (<https://capstone-ta.ce.undip.ac.id>) atau Discord Departemen Teknik Komputer (<https://discord.gg/rQNjs4R>). Mahasiswa juga harus melengkapi Form Pendaftaran Proyek Desain Capstone dan menyerahkan Form tersebut kepada Koordinator Proyek Desain Capstone.

Pemilihan Dosen Pembimbing pada form pendaftaran hanya dilakukan untuk Dosen Pembimbing I. sedangkan untuk Dosen Pembimbing II akan di-*balancing* oleh tim Capstone. Dosen Pembimbing Capstone tidak harus berasal dari home-based Departemen Teknik Komputer, namun salah ada satu Dosen Pembimbing yang berasal dari Departemen Teknik Komputer.

- **Perkuliahan Proyek Desain Capstone 1**

Setelah mahasiswa melakukan pendaftaran dan mendapatkan Dosen Pembimbing, mahasiswa harus mengambil mata kuliah Proyek Desain Capstone 1 dan mengikuti perkuliahan secara aktif di dalam kelas. Selama proses perkuliahan Proyek Desain Capstone 1, mahasiswa harus mengumpulkan Dokumen C100 hingga C300, Form Nilai Dosen Pembimbing Capstone 1, Form nilai ujian dari Dosen Penguji 1 dan 2, Form berita acara dan mengikuti rangkaian sidang Proyek Desain Capstone untuk menguji proses pengerjaan C100-C300 yang sudah dilakukan.

- **Masa Pengerjaan Produk Capstone**

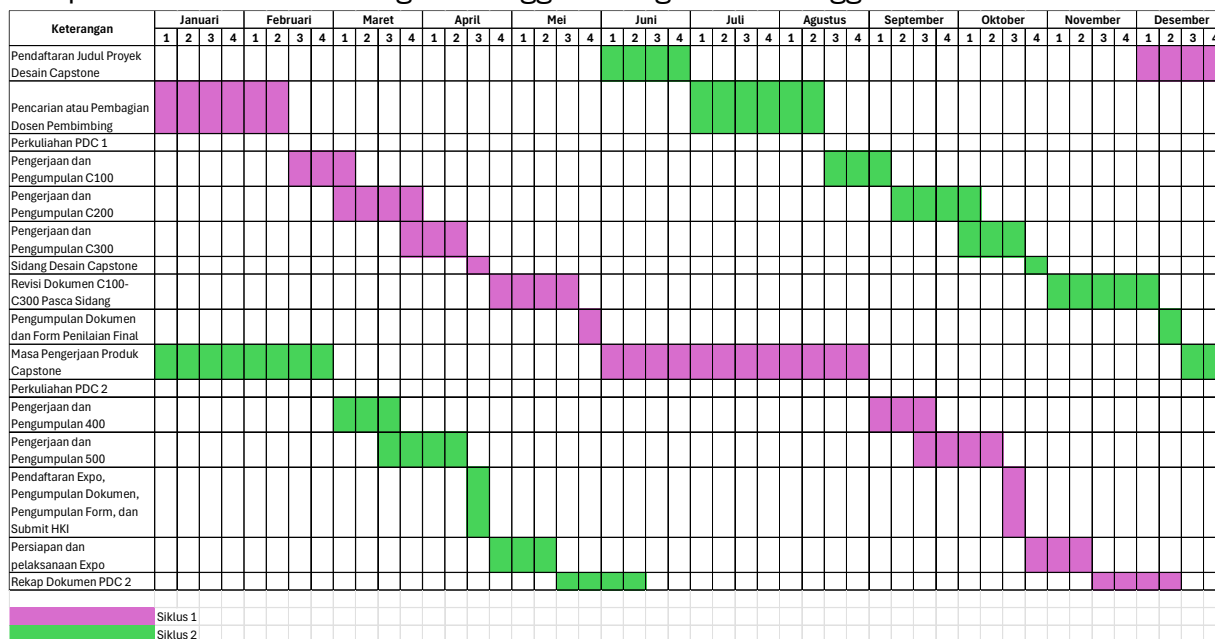
Setelah mahasiswa menyelesaikan perkuliahan Proyek Desain Capstone 1, mahasiswa bisa mulai melakukan implementasi terhadap desain rancangan yang telah disusun selama perkuliahan Proyek Desain Capstone 1.

- **Perkuliahan Proyek Desain Capstone 2**

Setelah mahasiswa menyelesaikan mata kuliah Proyek Desain Capstone 1, mahasiswa harus mengambil mata kuliah Proyek Desain Capstone 2. Untuk perkuliahan Proyek Desain Capstone II, mahasiswa tidak perlu duduk di dalam kelas. Selama proses perkuliahan Proyek Desain Capstone 2, mahasiswa harus mengumpulkan Dokumen C100 hingga C500, Form Nilai Dosen Pembimbing Proyek Desain Capstone 2 dari Dosen Pembimbing 1 dan 2, Form Kesesuaian Milestone Proyek Desain Capstone 2 dari Dosen Pembimbing 1 dan 2, Form Penilaian Expo Proyek Desain Capstone 2 (minimal 5 penilai), dan Form Peer Review Proyek Desain Capstone 2.

2. Jadwal Pelaksanaan

Proyek Desain Capstone dibagi menjadi 2 mata kuliah, yaitu mata kuliah Proyek Desain Capstone 1 yang ada pada semester 6 dan mata kuliah Proyek Desain Capstone 2 yang ada pada semester 7. Pengerjaan Proyek Desain Capstone yang ada pada Departemen Teknik Komputer berlangsung selama 1 tahun dan disediakan dalam 2 siklus. Jadwal pelaksanaan pengerjaan Proyek Desain Capstone dapat dilihat pada Gambar 1. Jadwal pelaksanaan yang ada pada Gambar 1 bisa bergeser hingga kurang-lebih 4 minggu.



Gambar 1 Timeline Proyek Desain Capstone

3. Syarat

Mata kuliah Proyek Desain Capstone 1 bisa diambil oleh mahasiswa yang sudah menempuh 100 sks mata kuliah yang ada pada Kurikulum Departemen Teknik Komputer. Mata kuliah Proyek Desain Capstone 2 bisa diambil oleh mahasiswa yang sudah menyelesaikan Mata kuliah Proyek Desain Capstone 1 dan dinyatakan lulus pada mata kuliah Proyek Desain Capstone 1.

4. Topik Proyek Desain Capstone

Topik Proyek Desain Capstone yang ada pada Departemen Teknik Komputer mengacu pada Rencana Induk Penelitian (RIP) dan Pengabdian Kepada Masyarakat yang ada pada Buku Pedoman Penelitian dan Pengabdian Departemen Teknik Komputer.

Departemen Teknik Komputer memiliki empat buah konsentrasi keilmuan yang meliputi: 1) Sistem Tertanam; 2) Jaringan Komputer dan Keamanan; 3) Rekayasa Perangkat Lunak; 4) dan Multimedia.

Oleh sebab itu, dengan mengacu pada RIP, maka tema Proyek Desain Capstone di lingkungan Departemen Teknik Komputer dalam periode 2020 – 2029 adalah:

“Peningkatan kualitas kehidupan smart society melalui pemanfaatan smart technologies”.

Tema ini selanjutnya dibagi menjadi beberapa topik sebagai berikut:

- 1) *Early warning system*
- 2) *Building/area monitoring or controlling system*
- 3) *Smart business/organization platform/support system*
- 4) *Smart city and transportation*

5. Luaran Proyek Desain Capstone

Dalam pelaksanaan Proyek Desain Capstone, mahasiswa harus menghasilkan beberapa keluaran, yaitu:

- Dokumen C100-C500
- Produk hasil penyelesaian permasalahan *stakeholder*
- Sertifikat HKI

6. Kelas Proyek Desain Capstone 1

Mata kuliah Proyek Desain Capstone 1 terdiri dari 5 CPMK, yaitu:

- CPMK 1 - Mahasiswa mampu menerapkan pengetahuan matematis atau ilmu alam dalam perancangan dan pengembangan solusi proyek berbasis Teknik Komputer ketika menganalisis dan menyelesaikan masalah kompleks di dunia nyata.
- CPMK 2 - Mahasiswa mampu merancang dan mengembangkan komponen, sistem, atau proses berbasis Teknik Komputer untuk memenuhi kebutuhan spesifik dengan mempertimbangkan batasan ekonomi, sosial, lingkungan, Kesehatan, dan keberlanjutan menggunakan perangkat teknis yang relevan, misalnya pemrograman mikrokontroler, desain prototipe IoT, analisis data menggunakan platform/framework modern, atau perancangan UI/UX yang sesuai dengan prinsip desain antarmuka pengguna.
- CPMK 3 - Mahasiswa mampu melakukan riset untuk mengumpulkan, menganalisis, dan menginterpretasi data guna mendukung penilaian teknis dan ilmiah dalam desain proyek.
- CPMK 4 - Mahasiswa mampu mengidentifikasi, merumuskan, menganalisis, dan menyelesaikan permasalahan kompleks di bidang Teknik Komputer.
- CPMK 5 - Mahasiswa mampu memahami dan mengikuti perkembangan teknologi terbaru di bidang Teknik Komputer serta menunjukkan tanggung jawab untuk pembelajaran sepanjang hayat yang berinisiatif sebagai agen perubahan di masyarakat.

Perkuliahan Proyek Desain Capstone 1 mengikuti ketentuan perkuliahan yang ada di Departemen Teknik Komputer terdiri dari 14 pertemuan dengan pertemuan tiap minggunya dapat dilihat pada Tabel 1.

Tabel 1 Rencana Perkuliahan Proyek Desain Capstone 1

Pertemuan	Materi
1	Penjelasan Tentang Proyek Capstone Kesesuaian Latar Belakang Masalah, Rumusan Masalah, dan Tujuan Proyek Capstone
2	Analisis Aspek Ekonomis Analisis Aspek Sustainabilitas Analisis Aspek Manufakturabilitas Solusi Perbandingan Teknik
3	Gambaran Sistem Saat ini Target Sistem yang Dikembangkan
4	Menentukan Fungsi Utama Produk Menggambarkan Karakteristik Pengguna Sistem Perbedaan Lingkungan Pengembangan dan Lingkungan Operasional
5	<i>External Interface</i> <i>Functional Description</i> <i>Data Requirement</i> <i>Functional and Non-Functional</i>
6	Desain dan Verifikasi Produk <i>Testing Plan</i>
7	Penjelasan contoh pembuatan arsitektur sistem dan detilnya Jenis - Jenis diagram yang dapat digunakan dalam pengembangan sistem
8	Tidak Ada UTS, Periode Pengumpulan C100-C300
9	Implementasi Produk Tampilan Produk Demonstrasi Produk
10	Teknik-teknik Pengujian Penjelasan penelitian yang bisa dilakukan dari Produk Capstone yang dihasilkan Penjelasan Pengajuan HKI Penjelasan Pendaftaran Makalah pada Jurnal
11	Periode Sidang PDC
12	Periode Sidang PDC
13	Revisi C100-C300
14	Revisi C100-C300
15	Revisi C100-C300
16	No UAS, Periode Pengumpulan dokumen C100-C300, dan form-form penilaian

Rubrik penilaian yang digunakan pada mata kuliah Proyek Desain Capstone 1 dapat dilihat pada Tabel 2 berikut.

Tabel 2 Rubrik Penilaian Proyek Desain Capstone 1

CPMK	Form Penilaian Dosbing 1	Form Penilaian Dosbing 2	Form Penilaian Penguji 1	Form Penilaian Penguji 2
CPMK 1	15	15	15	15
CPMK 2	25	25	25	25
CPMK 3	25	25	25	25
CPMK 4	15	15	15	15
CPMK 5	20	20	20	20

Bobot untuk total nilai pada keempat Form yang digunakan adalah sama, yaitu masing-masing komponen sebesar 25% pada SIAP.

7. Kelas Proyek Desain Capstone 2

Mata kuliah Proyek Desain Capstone 2 terdiri dari 6 CPMK, yaitu:

- CPMK 1 - Mahasiswa mampu melakukan riset untuk mengumpulkan, menganalisis, dan menginterpretasi data guna mendukung penilaian teknis dan ilmiah dalam desain proyek.
- CPMK 2 - Mahasiswa mampu mengidentifikasi, merumuskan, menganalisis, dan menyelesaikan permasalahan kompleks di bidang Teknik Komputer.
- CPMK 3 - Mahasiswa memiliki keterampilan dalam menerapkan metode dan desain perancangan sistem.
- CPMK 4 - Mahasiswa mampu menyusun laporan dan mempresentasikan hasil proyek secara ilmiah dan profesional di hadapan masyarakat akademik maupun publik.
- CPMK 5 - Mahasiswa mampu melakukan perencanaan dan pengelolaan proyek desain capstone berbasis Teknik Komputer secara komprehensif dengan memperhatikan keterbatasan sumber daya, waktu, dan kebutuhan pengguna yang teridentifikasi.
- CPMK 6 - Mahasiswa mampu bekerja sama secara efektif dalam tim lintas peminatan untuk mengembangkan solusi kreatif berbasis teknologi, serta berperan sebagai anggota maupun pemimpin tim dengan menerapkan etika profesi yang baik.

Pada mata kuliah Proyek Desai Capstone 2, mahasiswa tidak duduk di dalam kelas. Rubrik penilaian yang digunakan pada mata kuliah Proyek Desain Capstone 2 dapat dilihat pada Tabel 2 berikut.

Tabel 2 Rubrik Penilaian Proyek Desain Capstone 2

CPMK	Form Penilaian Dosbing 1	Form Penilaian Dosbing 2	Form Kesesuaian Milestone Dosbing 1	Form Kesesuaian Milestone Dosbing 2	Form Penilaian Expo	Form Peer Review
CPMK 1	30	30				
CPMK 2	30	30				
CPMK 3	40	40				
CPMK 4					100	
CPMK 5			100	100		
CPMK 6						100

Bobot untuk total nilai pada Form Penilaian Dosen Pembimbing 1 dan 2 serta Form Kesesuaian Milestone dari Dosen Pembimbing 1 dan 2 adalah 15% pada SIAP. Form Penilaian Expo memiliki bobot 20% pada SIAP. Form Peer Review memiliki bobot 20% pada SIAP.

BAB III – Pedoman Penulisan

Template dan Dokumen C100

Topik Capstone	Topik Capstone	
Siklus / Tahun	*Gasal atau Genap / (tahun)	
Judul Dokumen	Capstone TA Judul Capstone Proyek kelompok	
Jenis Dokumen	PROPOSAL Catatan: Penggunaan dan penyebaran dokumen ini dikendalikan oleh Departemen Teknik Komputer Universitas Diponegoro	
Nomor Dokumen	C100.[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
Pembimbing 1 (Utama)	Nama		Jabatan Anggota
	NIP.		
	Tanggal		Tanda Tangan
Pembimbing 2	Nama		Jabatan Anggota
	NIP.		
	Tanggal		Tanda Tangan

	Tanggal		
--	---------	--	--

Versi, Tanggal, Oleh	Perbaikan

Daftar Isi

1.	Pendahuluan	15
1.1.	Ringkasan isi dokumen	15
1.2.	Aplikasi Dokumen	15
1.3.	Referensi	15
1.4.	Daftar Singkatan	15
2.	Proposal Pengembangan Produk	16
2.1.	Latar belakang masalah	16
2.2.	Rumusan masalah	17
2.3.	Tujuan	17
2.4.	Pemilihan Solusi dan Teknik	18
2.5.	Skenario pemanfaatan produk oleh stakeholder	26
3.	Usaha pengembangan	27
3.1.	Man-month	27
3.2.	Machine-month	28
3.3.	Development tools	30
3.4.	Test equipment	32
3.5.	Kebutuhan expert	32
4.	Kesimpulan	32

1. Pendahuluan

1.1. Ringkasan isi dokumen

Dokumen ini memaparkan informasi mengenai alasan pembuatan, analisis sistem kebutuhan dan usulan proses pengembangan dari sistem. Dokumen ini juga membahas seluruh aspek dari pembuatan sistem, mulai dari sudut pandang pihak pengguna hingga dari sudut pandang pihak penulis. Dokumen ini sendiri dapat membantu menjelaskan sistem yang dibuat kepada pengguna. Dengan demikian, pembaca diharapkan dapat mengerti proses pengembangan dari solusi yang ditawarkan. Kemudian, dokumen ini dapat digunakan sebagai dasar proses pengembangan dan evaluasi selama tahap pengembangan proyek berlangsung. Selain itu, dokumen ini juga dapat digunakan sebagai acuan teknis untuk

1.2. Aplikasi Dokumen

Dokumen ini berlaku untuk digunakan dalam pengembangan produk proyek capstone:

- (1) Sebagai gambaran umum dari segi teknis maupun non-teknis tugas akhir yang akan dikerjakan.
- (2) Memastikan kelayakan tugas akhir, baik dari segi teknik, waktu, biaya/ekonomis, maupun strategis.
- (3) Menjadi catatan proses pengerjaan dan revisi yang dilakukan.

Proposal ini diajukan kepada dosen pembimbing tugas akhir dan tim capstone tugas akhir Program Studi Sarjana Teknik Komputer Undip sebagai bahan penilaian tugas akhir.

1.3. Referensi

- [1] P. Citra, “Kombinasi Sobel , Canny Dan Otsu Untuk Segmentasi Citra,” *Technologia*, vol. 13, no. 2, hal. 102–107, 2022.
- [2] S. C. Re, K. Daya, D. Tanggung, dan J. Perusahaan, “Keselamatan dan Kesehatan Kerja,” 2013, Diakses: Feb 10, 2023 Tersedia pada: www.ifro.org.
- [3] “Raspberry Pi 3 Model B+,” Diakses: Feb 10, 2023. Tersedia pada: www.raspberrypi.org/products/raspberry.
- [4] IEA Constituent Agreements, 2021. Graduate Attributes and Professional Competences. Available at: <https://www.ieagreements.org/assets/Uploads/IEA-Graduate-Attributes-and-Professional-Competencies-2021.1-Sept-2021.pdf> [Accessed 13 January 2025].

1.4. Daftar Singkatan

Akronim	Pengertian
DoS	Denial of Service
DDoS	Distributed Denial of Service
IDS	Intrusion Detection System

IPS	Intrusion Prevention System
-----	-----------------------------

2. Proposal Pengembangan Produk

2.1. Latar belakang masalah

Masalah dilihat dari kacamata customer dan dapat menunjukkan di mana terjadinya masalah, merupakan complex engineering problem. Definisi complex engineering problem mengacu pada buku pedoman capstone project.

Contoh:

Keamanan jaringan menjadi semakin krusial mengingat perkembangan pesat teknologi informasi dan internet pada era digital saat ini. Keamanan jaringan menjadi fokus utama dalam menghadapi tantangan berbagai ancaman siber yang terus berkembang, termasuk serangan peretasan, virus, *malware*, dan pencurian data sensitif. Dengan semakin kompleksnya infrastruktur jaringan dan ketergantungan masyarakat terhadap teknologi, organisasi, perusahaan, serta lembaga pemerintahan perlu memastikan integritas, kerahasiaan, dan ketersediaan data yang berada dalam jaringan mereka.

Salah satu jenis serangan pada jaringan adalah DoS/DDoS. *Denial of Service* (DoS) atau *Distributed Denial of Service* (DDoS) merupakan serangan yang membanjiri server dengan mengirimkan permintaan yang sangat banyak sehingga menghabiskan sumber daya pada server tersebut sampai server tersebut tidak dapat menjalankan fungsi dan tugasnya dengan benar [1]. Serangan DoS/DDoS mempunyai beberapa bentuk berdasarkan *OSI layer* yang diserang seperti, serangan *slow HTTP DoS*, DRDoS (*Distributed Reflection Denial of Service*), dan *DNS amplification DoS/DdoS*.

Serangan *slow HTTP DoS* adalah salah satu metode serangan DoS yang menargetkan server HTTP. Metode ini menghambat layanan dengan menjenuhkan kumpulan koneksi dengan permintaan yang lambat dan banyak. Laporan bahwa serangan DoS “*slow read*” semacam serangan HTTP DoS dari hanya satu penyerang dapat dicegah secara efektif dengan membatasi jumlah koneksi untuk setiap alamat IP [2]. Sementara itu ada juga serangan DoS lainnya, yaitu *DNS amplification*. Serangan amplifikasi *Domain Name Server* (DNS) adalah bentuk populer dari *Denial Distribution of Service* (DDoS), di mana penyerang menggunakan server DNS terbuka yang dapat diakses publik untuk membanjiri sistem target dengan lalu lintas respons DNS [3]. DRDoS (*Distributed Reflection Denial of Service*) adalah sejenis serangan DoS (*Denial of Service*). Pada serangan ini pihak ketiga ditipu atau dibohongi untuk mengirimkan data dalam jumlah besar ke para korban. Artinya, penyerang menggunakan spoofing IP alamat sumber untuk menyembunyikan identitas mereka dan menyebabkan pihak ketiga mengirim data ke korban.

Pada kantor Ditreskrimsus Polda Jateng belum ada sistem yang dapat mengenali serangan DoS/DDoS yang terjadi pada server. Oleh karena itu dibutuhkan sistem yang dapat digunakan untuk melakukan monitoring dan memberikan peringatan ketika terjadi suatu serangan DoS/DDoS pada server tersebut. Sebagai bentuk upaya meningkatkan pengamanan pada server kantor Subdit V Ditreskrimsus Polda Jateng dari serangan *Denial of Service/Distributed Denial of Service (DoS/DDoS)*, maka dapat diterapkan teknologi *Intrusion Detection System/Intrusion Prevention System (IDS/IPS)*. *Intrusion Detection System (IDS)* adalah sebuah aplikasi perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan [5]. Dalam melakukan pendeteksian, IDS menggunakan beberapa metode, diantaranya yaitu, *signatures dan anomaly detection*. Teknik IDS saat ini masih banyak menggunakan metode tradisional, sehingga masih jauh dari sempurna dibandingkan dengan teknik dan alat yang digunakan penyerang, karena IDS dengan metode tradisional hanya menggunakan deteksi berbasis signature atau model deteksi berbasis anomali dan menyebabkan banyak kesalahan [6]. Oleh karena itu *Model-Based Feature Selection* diusulkan sebagai cara untuk menutup kesenjangan dan meningkatkan efektivitas IDS. *Smart Intrusion Detection System (SIDS)* ini akan diintegrasikan dengan *log management system* yang akan mempermudah admin dari *stakeholder* untuk memonitoring perangkat dari serangan DoS/DDoS.

2.2. Rumusan masalah

Berisi permasalahan stakeholder yang dipilih untuk diselesaikan dengan produk yang diusulkan oleh Tim Capstone. Dapat dinyatakan dalam minimal dua kalimat pernyataan tanpa jargon.

Contoh:

Ditreskrimsus Polda Jateng belum memiliki sistem yang dapat mengenali serangan DoS/DDoS yang terjadi pada server sehingga dibutuhkan sistem yang dapat digunakan untuk memonitor dan memberikan peringatan ketika terjadi serangan DoS/DdoS.

Intrusion Detection System (IDS) dengan metode tradisional hanya menggunakan deteksi berbasis *signature* atau model deteksi berbasis anomali dan menyebabkan banyak kesalahan.

2.3. Tujuan

Tujuan umum menjelaskan visi besar atau hasil utama yang ingin dicapai dari proyek diikuti dengan tujuan khusus yang merinci langkah-langkah spesifik atau aspek teknis untuk mendukung pencapaian tujuan umum.

Contoh:

A. Tujuan Umum

"Mengembangkan *Smart Intrusion Detection System* (SIDS) untuk memonitor perangkat dari serangan DoS/DDoS dengan menambahkan *Model-Based Feature Selection* untuk meningkatkan efektivitas IDS tradisional.

B. Tujuan Khusus

Tujuan khusus ini memuat tujuan proyek capstone sesuai dengan pembagian pekerjaan pada masing-masing anggota tim Capstone.

Contoh:

1. "Membuat dan melakukan pelatihan *machine learning Model-Based Feature Selection* untuk mengidentifikasi serangan DoS/DDoS."
2. "Mengintegrasikan *Model-Based Feature Selection* untuk meningkatkan efektivitas IDS pada *Smart Intrusion Detection System* (SIDS)."
3. "Mengintegrasikan *Smart Intrusion Detection System* (SIDS) dengan *log management system* dan *distributed streaming system* yang akan mempermudah admin dari *stakeholder* untuk memonitor perangkat dari serangan DoS/DDoS."

2.4. Pemilihan Solusi dan Analisis

Gambarkan beberapa alternatif solusi yang bisa dipilih untuk menyelesaikan masalah yang ada (minimal 3), kemudian lakukan analisis berikut untuk memilih solusi yang ada. Alternatif solusi yang diberikan merupakan tiga contoh alternatif solusi secara umum yang dapat digunakan untuk menyelesaikan permasalahan yang ada. Jika kasus yang digunakan tidak bisa memberikan alternatif solusi secara umum, maka tim capstone bisa memberikan alternatif terhadap fitur yang ada di dalam sistem. Contoh pemilihan alternatif solusi bisa dipisahkan berdasarkan perangkat keras, perangkat lunak, atau hal lain sesuai dengan proyek yang dikembangkan. Analisis aspek terkait yang dilakukan minimal sebanyak tiga aspek dengan aspek wajib yaitu: ekonomis, manufakturabilitas, dan sustainabilitas.

Contoh:

2.4.1. Alternatif Solusi

A. Alternatif Solusi untuk Deteksi DoS/DDoS

Berdasarkan usulan solusi yang ditunjukkan oleh tabel 1, *machine learning* dipilih untuk menjadi solusi dalam deteksi DoS/DDoS. Hal ini dikarenakan dalam proses deteksi menggunakan pendekatan berbasis *signature*, IDS tidak dapat mendeteksi serangan yang belum dikenal karena basis data *signature* yang telah kadaluwarsa atau karena *signature* memang belum tersedia. Pendekatan kedua adalah deteksi berbasis anomali dimana pada deteksi ini, perlu diciptakan suatu profil perilaku khusus dari fitur aktivitas jaringan dalam taraf tertentu. Profil ini kemudian dijadikan sebagai dasar untuk mendefinisikan aktivitas jaringan normal [10].

Tabel 1. Alternatif solusi untuk deteksi DoS/DDoS

Solusi	Deskripsi
--------	-----------

<i>Signature Based Detection</i>	Signature Based Detection adalah salah satu metode IDS yaitu, dengan cara menyadap paket data kemudian membandingkannya dengan database rule IDS yang berisi signature-signature paket serangan. Berdasarkan perbandingan tersebut, jika paket data mempunyai pola yang sama dengan setidaknya salah satu pola di dalam database rule IDS, maka paket tersebut dapat dianggap sebagai serangan, dan demikian juga sebaliknya. Apabila paket data tersebut sama sekali tidak mempunyai pola yang sama dengan pola di database rule IDS. maka paket data tersebut dianggap bukan serangan [7].
<i>Anomaly Based Detection</i>	IDS jenis ini dapat mendeteksi adanya penyusupan dengan mengamati adanya kejanggalan pada sistem, atau adanya penyimpangan-penyimpangan dari kondisi normal. Sebagai contoh, apabila terdapat penggunaan memori yang melonjak secara terus menerus atau terdapat koneksi paralel dari sebuah IP Address dalam jumlah yang banyak dalam waktu yang bersamaan, maka kondisi tersebut dapat dianggap sebagai sebuah kejanggalan, yang kemudian oleh IDS dianggap sebagai serangan [7].
<i>Machine Learning</i>	Machine learning adalah bidang ilmu komputer yang menggunakan teknik statistika untuk memberi kemampuan sistem komputer agar dapat belajar dari data tanpa diprogram secara eksplisit [8]. Untuk IDS, machine learning dapat digunakan untuk mendeteksi serangan yang diketahui atau serangan yang tidak diketahui jika model telah cukup terlatih [9].

B. Alternatif solusi untuk *Intrusion Detection System/Intrusion Prevention System* (IDS/IPS)

Berdasarkan pada tiga usulan solusi yang ditunjukkan di tabel 2, Snort dipilih untuk menjadi solusi dan teknik untuk *Intrusion Detection System/Intrusion Prevention System* (IDS/IPS). Hal ini dikarenakan Snort lebih mudah untuk diimplementasikan dan memiliki *ruleset* yang dapat dikustomisasi sesuai kebutuhan [13].

Tabel 2. Alternatif solusi untuk *Intrusion Detection System/Intrusion Prevention System* (IDS/IPS)

Solusi	Deskripsi
<i>Suricata</i>	Suricata adalah IDS, IPS dan monitoring engine untuk jaringan yang berkinerja tinggi. Suricata adalah open source dan dimiliki oleh masyarakat yang dikelola yayasan non-profit, Open Information Security Foundation (OISF). Suricata dikembangkan oleh OISF dan vendor pendukungnya [11].

<i>Snort</i>	Snort merupakan salah satu contoh program Network-based Intrusion Detection System, yaitu sebuah program yang dapat mendeteksi suatu usaha penyusupan pada suatu sistem jaringan komputer. Snort bersifat open source dengan lisensi GNU General Purpose License sehingga software ini dapat dipergunakan untuk mengamankan sistem server tanpa harus membayar biaya lisensi [7].
<i>Security Onion</i>	Security Onion merupakan salah satu distro dari sistem operasi berbasis Linux. Security Onion umumnya digunakan sebagai tools atau bundle packet untuk Network Security Monitoring (NSM). Security Onion dapat difungsikan menjadi dua jenis sistem operasi, pertama sebagai sistem operasi standalone dimana Security Onion akan berfungsi sebagai penyaji data sedangkan jenis kedua sebagai server untuk merekam, mengelola dan menyajikan data yang didapat dari sistem sensornya [12].

C. Alternatif solusi untuk *Log Management System*

Dari tiga pilihan solusi yang ditunjukkan dalam tabel 3, ELK Stack dipilih karena *platform Log Management System* lainnya mematok harga yang cukup tinggi untuk penyimpanannya. ELK Stack merupakan *Log Management System* berbasis open source dan gratis serta memiliki kemampuan untuk menjadi *Log Management System* yang baik.

Tabel 3. Alternatif solusi untuk *Log Management System*

Solusi	Deskripsi
ELK Stack	ELK Stack terdiri dari 4 buah aplikasi yang berbeda yaitu Elasticsearch, Logstash, Kibana, dan Beats. Masing-masing aplikasi memiliki fungsi yang berbeda-beda untuk menunjang dalam manajemen log [14]. Elasticsearch Logstash Kibana (ELK Stack) merupakan komponen yang tepat dalam membangun <i>log event management</i> yang dapat memberi insight kepada sistem administrator mengenai tren, statistik, dan anomali yang terjadi [15]. Sebuah <i>platform</i> perangkat lunak yang digunakan untuk memantau, mencari, menganalisis dan memvisualisasikan data yang dihasilkan mesin.
Splunk	Splunk melakukan penangkapan, pengindeksan, dan menghubungkan data secara realtime ke dalam sebuah wadah yang dapat dengan mudah dicari dan menghasilkan suatu grafik, peringatan, <i>dashboard</i> serta visualisasi agar data dapat dengan mudah dibaca dan dianalisis [16]. Menghadirkan pengumpulan dan normalisasi log terpusat, deteksi ancaman dan respons otomatis,

	visualisasi yang intuitif, dan antarmuka pengguna, serta korelasi waktu nyata dan pencarian <i>log</i> untuk mendukung penyelidikan [17].
--	---

D. Alternatif solusi untuk algoritma klasifikasi

Dari usulan solusi untuk algoritma machine learning pada tabel 4, Algoritma Naive Bayes dan KNN dipilih. Pada penelitian ini digunakan dua dataset yaitu, CICDoS2017 dan CICDDoS2019. Algoritma Decision Tree memberikan akurasi 60,23% untuk UNSW-NB-15 dan 99,08% untuk dataset CICDDoS2019 [22]. Sementara itu, pada dataset CICDoS2017 Algoritma KNN memberikan akurasi 99,36% [23].

Tabel 4. Alternatif solusi untuk algoritma klasifikasi

Solusi	Deskripsi
<i>Decision Tree</i>	<i>Decision Tree</i> adalah algoritma yang bisa menghasilkan keputusan dengan cara membentuk pohon keputusan [18]. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target [19].
<i>Naive Bayes</i>	Algoritma <i>Naive Bayes</i> merupakan algoritma yang menggunakan perhitungan probabilitas. Algoritma ini umumnya digunakan untuk menyelesaikan permasalahan prediksi berupa klasifikasi. Algoritma ini juga dikenal sebagai yang memiliki akurasi yang tinggi. Proses klasifikasi pada algoritma ini terdapat fase yakni fase <i>training</i> dan fase <i>testing</i> . Fase <i>training</i> atau bisa disebut sebagai fase <i>learning</i> adalah sebagian data telah diketahui kelas datanya untuk model perkiraan. Selanjutnya fase pengujian atau bisa disebut fase klasifikasi model yang sudah terbentuk diuji dengan sebagian data lainnya agar diketahui akurasi atas model yang sudah terbentuk [20].
<i>KNN</i>	Algoritma KNN bersifat sederhana bekerja berdasarkan jarak terpendek dari data uji ke data latih untuk menentukan kelas dari data tersebut Setelah mengumpulkan data-data pada kelompok k tertentu, kemudian diambil kelas data mayoritas untuk dijadikan sebagai kelas prediksi dari data uji. KNN memiliki beberapa kelebihan yaitu tahan terhadap data yang memiliki <i>noise</i> dan efektif terhadap data latih yang berjumlah besar dan memiliki performa cukup baik. Namun waktu yang digunakan untuk komputasi

	sangatlah lama jika data latihnya besar dan sangat sensitif dengan ciri yang redundan atau relevan [21].
--	--

2.4.2. Analisis Aspek Terkait

A. Analisis dari aspek ekonomis

Analisis aspek ekonomis menggunakan definisi sebagai berikut: (Hoffman, H.F. and Hoffman, H.F., 2014. Engineering and the capstone course (pp. 1-5). Springer International Publishing.)

Economic or financial: effect of this topic on the local economy, savings, possible cost of project development, material cost, labor issues, outsourcing needs, etc. Harus terdapat cost-benefit analysis yang mencakup hal-hal berikut.

Cost:

- *Biaya operasional existing (sebelum adanya proposal solusi yang dikembangkan, jika ada).*
- *Biaya project development, yang juga mencakup setiap material cost.*
- *Biaya kebutuhan tenaga ahli dan operator (labor) dengan adanya solusi tersebut, sekaligus bagaimana efeknya terhadap ekonomi setempat, jika ada.*

Benefit:

- *Saving (berapa ekspektasi pengurangan biaya dengan adanya solusi).*
- *Aspek benefit yang intangible dengan adanya solusi yang ditawarkan. Contoh: solusi dapat memotong ekspektasi waktu pekerjaan tiga hari menjadi tiga jam.*

Financial Metrics (pilih salah satu):

- *Return of Investment (ROI)*
- *Payback period*
- *Net Present Value (NPV)*
- *Internal Rate of Return (IRR)*

Contoh:

Cost:

Kategori	Item	Biaya (Rp)	Keterangan
Biaya Pengembangan Sistem	Tenaga Kerja	0	Tim Capstone
	Lisensi Software: Suricata atau Snort, ElkStack, Elasticsearch, Logstast, dan Kibana), Apache Kafka	0	Menggunakan perangkat <i>open source</i>
	Infrastruktur: Server internal Polda	0	Menggunakan server yang sudah ada

	Dataset	0	Pemanfaatan dataset publik seperti CICDoS2017 dan CICDDoS2019.
Biaya Operasional Tahunan	Pemeliharaan server internal	0	Biaya listrik, pendinginan, dan perawatan server Polda Jateng sudah masuk ke anggaran tahunan Polda.
	Sertifikat SSL/TLS	0	Keamanan enkripsi komunikasi data, sudah masuk ke anggaran tahunan Polda

Benefit:

1. Tangible Benefit

Mengurangi biaya downtime akibat serangan DDoS:

- Asumsi kerugian akibat gangguan sistem Rp20.000.000/jam.
- Jika sistem mencegah downtime 5 jam/bulan:
 - Penghematan: $\text{Rp}20.000.000 \times 5 \times 12 \text{ bulan} = \text{Rp}1.200.000.000/\text{tahun}$.

Kategori	Item	Biaya (Rp)	Keterangan
Penghematan dan Efisiensi	Pencegahan downtime (5 jam/bulan)	1.200.000.000	Mengurangi kerugian akibat gangguan sistem layanan daring.

2. Intangible Benefit

Aspek ekonomis tidak harus selalu diukur dari sesuatu yang *tangible*, tetapi bisa juga dari aspek *intangible*. Misal: kecepatan proses pencarian data, kemampuan *real-time tracking*,

B. Analisis dari aspek manufakturabilitas

Analisis dari aspek manufakturabilitas ini terkait dengan ketersediaan, skalabilitas dan maintainabilitas seperti: material availability, software availability, technology stack, use of commercial-off-the-shelf (COTS) versus custom components, special needs for hostile environments.

Contoh:

1. Material Availability:

Sensor	Sensor kualitas air (pH, suhu, oksigen terlarut, salinitas).
Ketersediaan	Tersedia secara luas di pasar dengan berbagai spesifikasi.

2. Penggunaan *Commercial-Off-The-Shelf* dan *Custom Components*

COTS Components	Sensor, mikrokontroler, modul komunikasi
Custom Components	jalur sensor, tempat sensor, dkk

3. *Special Needs for Hostile Environments*

a. Lingkungan basah dan korosif:

Masalah	Air tambak sering kali memiliki kadar garam yang tinggi, yang dapat mempercepat korosi pada komponen logam.
Solusi	Gunakan bahan tahan korosi seperti stainless steel untuk rangka dan sensor dan melapisi PCB dengan bahan konformal untuk melindungi dari kelembapan dan salinitas.
	Gunakan casing pelindung IP68 yang tahan air dan debu digunakan untuk perangkat elektronik utama.

b. Paparan sinar matahari dan suhu ekstrem:

Masalah	Perubahan suhu yang drastis dapat memengaruhi performa baterai dan sensor.
Solusi	Gunakan baterai lithium-ion yang dirancang untuk suhu tinggi dan menambahkan pelindung UV pada casing untuk mengurangi degradasi material.

4. *Software availability:*

Jenis Perangkat Lunak	Nama Perangkat Lunak	Keterangan
<i>Intrusion Detection System/Intrusion Prevention System (IDS/IPS)</i>	Suricata	<i>Open source</i>
	Snort	<i>Open source</i>
<i>Log Management System</i>	Elk Stack (Elasticsearch, Logstash, dan Kibana)	<i>Open source</i>
<i>Distributed Streaming System</i>	Apache Kafka	<i>Open source</i>

Perangkat lunak yang digunakan yaitu Suricata atau Snort, Elk Stack (Elasticsearch, Logstash, dan Kibana) dan Apache Kafka merupakan perangkat lunak *open source* dan diperbaharui secara berkala. Forum penggunaan pun aktif, sehingga pencarian informasi mengenai perangkat lunak yang digunakan lebih mudah dilakukan. Selain itu aplikasi ini juga berjalan pada sistem operasi Linux, Linux sendiri merupakan sistem operasi yang bersifat *open source* yang cukup populer.

C. Analisis dari aspek sustainabilitas

Dalam membahas aspek sustainabilitas, Tim Capstone perlu memaparkan dampak analisis produk yang dipilih dalam hal: meminimalisasi pengaruh negatif terhadap lingkungan, memastikan kelangsungan ekonomi jangka panjang, penggunaan energi, dll.

Contoh:

1. Efisiensi Energi: bagaimana meminimalisasi penggunaan energi akan produk yang dikembangkan.
2. Manajemen Sumber: bagaimana merancang suatu aplikasi yang meminimalkan penggunaan ruang penyimpanan, bagaimana meminimalkan proses data transfer.
3. Efisiensi desain: bagaimana merancang sistem yang *scalable*.
4. Dll.

D. Analisis aspek tambahan

Bila diperlukan dalam perancangan produk yang dibuat dapat memilih salah satu proses analisis aspek yang ada (ethical, legal, environment, political, societal), yang berhubungan dengan produk yang akan dirancang.

Contoh:

1. Aspek Legal: Aspek legal terkait dengan hukum yang mungkin muncul pada produk yang akan dibuat, terkait dengan privasi data, hukum-hukum yang mengatur pengambilan dan penyimpanan data. Contoh: Bila pada program yang dibuat berisikan tentang data-data kesehatan, maka perlu melakukan analisis program berdasarkan Undang-Undang Nomor 36 Tahun 2009 tentang kerahasiaan data kesehatan, UU Nomor 19 Tahun 2016 tentang informasi dan transaksi elektronis, dll.
2. Aspek Etika: Terkait isu etika yang muncul pada produk yang akan dibuat. Contoh: Isu etika yang dibahas dapat terkait pada tanggung-jawab moral terkait transparansi data, atau pada pembagian hak akses data pada masing-masing *role* dan kerahasiaannya.

2.4.3. Pemilihan Solusi

Berdasarkan usulan solusi yang ditunjukkan oleh tabel 1, *machine learning* dipilih untuk menjadi solusi dalam deteksi DoS/DDoS. Hal ini dikarenakan dalam proses deteksi menggunakan pendekatan berbasis *signature*, IDS tidak dapat mendeteksi serangan yang belum dikenal karena basis data *signature* yang telah kadaluwarsa atau karena *signature* memang belum tersedia. Pendekatan kedua adalah deteksi berbasis anomali dimana pada deteksi ini, perlu diciptakan suatu profil perilaku khusus dari fitur aktivitas jaringan dalam taraf tertentu. Profil ini kemudian dijadikan sebagai dasar untuk mendefinisikan aktivitas jaringan normal [10].

Berdasarkan pada tiga usulan solusi yang ditunjukkan di tabel 2, Snort dipilih untuk menjadi solusi dan teknik untuk *Intrusion Detection System/Intrusion Prevention System*

(IDS/IPS). Hal ini dikarenakan Snort lebih mudah untuk diimplementasikan dan memiliki *ruleset* yang dapat dikustomisasi sesuai kebutuhan [13].

Dari tiga pilihan solusi yang ditunjukkan dalam tabel 3, ELK Stack dipilih karena *platform Log Management System* lainnya mematok harga yang cukup tinggi untuk penyimpanannya. ELK Stack merupakan *Log Management System* berbasis open source dan gratis serta memiliki kemampuan untuk menjadi *Log Management System* yang baik.

Dari usulan solusi untuk algoritma machine learning pada tabel 4, Algoritma Naive Bayes dan KNN dipilih. Pada penelitian ini digunakan dua dataset yaitu, CICDoS2017 dan CICDDoS2019. Algoritma Decision Tree memberikan akurasi 60,23% untuk UNSW-NB-15 dan 99,08% untuk dataset CICDDoS2019 [22]. Sementara itu, pada dataset CICDoS2017 Algoritma KNN memberikan akurasi 99,36% [23].

2.5. Skenario pemanfaatan produk oleh stakeholder

Mendeskripsikan bagaimana produk digunakan (dijelaskan sistem/ sub-sistem serta bagaimana masing-masing stakeholder memanfaatkan produk. Dalam bagian ini, tim capstone harus membuat: 1. Gambar rancangan arsitektur high-level dari produk (bagaimana setiap sistem/subsistem terhubung); 2. Penjelasan bagaimana produk yang dikembangkan digunakan per-stakeholder (misal dihubungkan antara fitur dengan stakeholder).

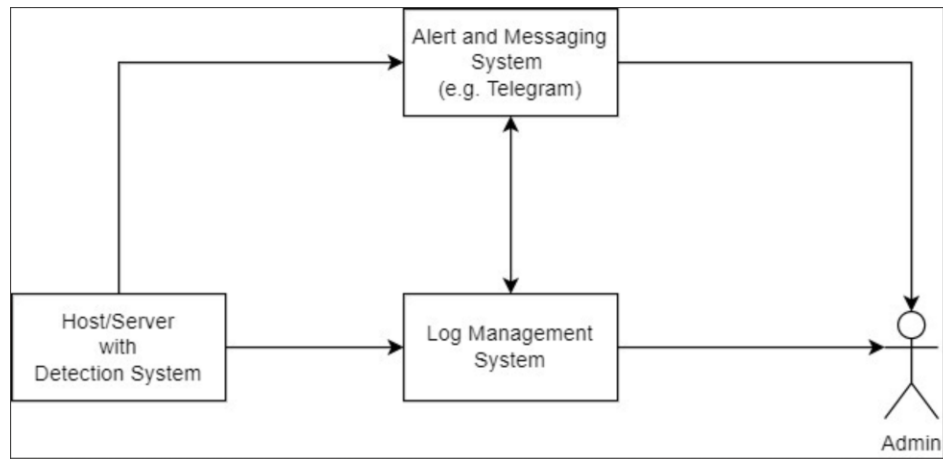
Contoh:

Dalam penelitian ini, dilakukan penelitian untuk mengembangkan sebuah Sistem Monitoring dan Deteksi Serangan DoS/DDoS Menggunakan *Smart Intrusion Detection System* (SIDS) dengan menerapkan pendekatan *Model-Based Feature Selection* ke sistem yang sedang dibangun. Sistem monitoring ini memiliki fungsi utama, yaitu untuk mendeteksi beberapa jenis serangan DoS/DDoS seperti *slow* HTTP DoS, DNS amplification DoS/DDoS, dan DRDoS (*Distributed Reflection Denial of Service*).

Gambar 2.1 menunjukkan arsitektur sistem monitoring ini menggunakan beberapa perangkat lunak seperti IDS Snort, Kafka, dan Elasticsearch Logstash Kibana (ELK Stack). Setiap perangkat lunak memiliki fungsi masing-masing dalam pembangunan sistem ini dan akan dijelaskan pada bagian *development tools*. Berikut merupakan cara kerja Sistem Monitoring dan Deteksi Serangan DoS/DDoS Menggunakan Smart Intrusion Detection System (SIDS) :

1. IDS Snort melakukan Capture/Sniff trafik jaringan pada host/server kemudian menyimpannya dalam bentuk file pcap.
2. File pcap diubah ke bentuk file csv.
3. Setelah itu, dilakukan ekstraksi fitur terhadap file pcap yang sudah diubah ke bentuk file .csv berdasarkan format dari dataset yang digunakan.
4. Melakukan prediksi menggunakan model machine learning yang sudah dikembangkan.

5. Mengirimkan hasil prediksi ke Apache Kafka menggunakan Filebeat.
6. Apache Kafka mendistribusikan log yang dikirim dari Filebeat ke Log Management System.
7. Log Management System (ELK Stack) melakukan klasifikasi terhadap log yang diterima menjadi log trafik berjenis DoS/DDoS atau trafik normal.
8. ElastAlert dapat mengirimkan notifikasi ke aplikasi pesan seperti Whatsapp apabila dibutuhkan.



Gambar 2.1. Arsitektur Sistem Monitoring dan Deteksi Serangan DoS/DdoS

Sistem monitoring DoS/DDoS ini akan dioperasikan di kantor Ditreskrimsus Polda Jateng Subdit V Siber. Pada kantor tersebut diperlukan satu orang administrator yang dapat menjalankan fungsi sebagai berikut :

1. Melakukan Monitoring Log

Sistem monitoring ini dapat menampilkan log dari berbagai sistem seperti sistem operasi, router, switch, server, dan lain-lain. Namun untuk proyek ini dikhususkan untuk dapat menampilkan log yang berasal dari IDS/IPS.

2. Melakukan Monitoring Dashboard

Sistem monitoring ini juga terdapat fitur dashboard yang dapat dimanfaatkan oleh administrator agar dapat melakukan monitoring sistem dengan lebih mudah dibandingkan hanya melakukan monitoring log. Hal ini dikarenakan data pada dashboard sudah divisualisasikan.

3. Menerima Alert

Alert atau peringatan pada sistem monitoring ini dikhususkan hanya untuk peringatan DoS/DDoS saja. Namun dapat diperluas lagi untuk memberikan peringatan lain sesuai kebutuhan. Alert atau peringatan ini akan diintegrasikan dengan E-mail dan Whatsapp sehingga dapat memudahkan Administrator jika sedang berada di luar kantor.

3. Usaha pengembangan

3.1. *Man-month*

Detail dari analisis ekonomis yang sudah disebutkan dalam hal perhitungan biaya pekerja.

Contoh:

Proyek pembuatan Sistem Monitoring dan Deteksi Serangan DoS/DDoS Menggunakan Model-Based Feature Selection ini, akan dikerjakan oleh satu tim Tugas Akhir Capstone program studi S1-Teknik Komputer periode 2023/2024 yang terdiri atas tiga orang mahasiswa Teknik Komputer angkatan 2020. Dua mahasiswa/pengembang akan bekerja menjadi *machine learning engineer* dan satu lainnya akan menjadi *security engineer* dengan tugas yang berbeda-beda. Berikut Gantt Chart pembagian waktu sesuai tugas masing-masing yang ditunjukkan pada Tabel 8 berikut.

Tabel 8. Gantt chart man month

Nama (Jabatan)	Pekerjaan	September				Oktober				November				Desember				Januari			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Aldo Serena Safiola Hafizh Anjar Saputra (Machine Learning Engineer)	Pemahaman DDoS Attack	■																			
	Pengumpulan Data																				
	Pembersihan dan Persiapan Data		■																		
	Ekstraksi Fitur			■	■																
	Pembuatan dan Pelatihan Model					■	■	■	■												
	Validasi dan Evaluasi Model								■												
	Tuning Model									■											
	Pemantauan dan Perbaikan										■										
Rafif Sadid Hamdani (Security Engineer)	Analisis Kebutuhan sistem	■																			
	Pengembangan dan Implementasi Sistem Deteksi DDoS									■											
	Implementasi model ML ke dalam IDS/IPS										■										
	Pemasangan log management system											■									
	Integrasi log management system dengan IDS/IPS.												■								
	Pengujian Sistem													■							
	Pengawasan Lalu Lintas Jaringan														■						
	Analisis Serangan DDoS															■					
	Deteksi dan Respon Kejadian Keamanan																■				
	Penanganan Serangan																	■			
	Peningkatan Keamanan																		■	■	■

3.2. Machine-month

Detail dari analisis ekonomis yang sudah disebutkan dalam hal perhitungan biaya mesin yang digunakan untuk mengembangkan produk. Sistem ini dibuat oleh tiga orang pengembang dengan setiap pengembang yang menggunakan laptop masing-masing. Dari pernyataan tersebut, berarti akan terdapat tiga buah perangkat yang digunakan untuk membuat sistem monitoring DoS/DDoS di mana setiap perangkat ini akan aktif atau digunakan sesuai dengan jam kerja penggunaannya.

Tabel 9. Tabel lama penggunaan mesin dan waktu pengerjaan

Nama (Jabatan)	Mesin	Waktu (Jam)
Aldo Serena Sadiola (Machine Learning Engineer)	Laptop Lenovo G40-45 6410M	264 Jam
Hafizh Anjar Saputra (Machine Learning Engineer)	Laptop Acer Swift 3	324 Jam
Rafif Sadid Hamdani (Security Engineer)	Laptop Lenovo IdeaPad 1 14ALC7	432 Jam
Total Jam		1020 Jam

Tabel 9 menunjukkan total dari waktu dalam jam yang akan digunakan selama pembuatan Sistem Monitoring dan Deteksi Serangan DoS/DDoS menggunakan *Model-Based Feature Selection*.

Machine-Month merupakan metrik yang diperlukan untuk menentukan waktu/usaha yang dibutuhkan untuk menyelesaikan satu proyek tugas berdasarkan waktu maksimal penggunaan suatu mesin. Formula untuk menghitung *Machine-Month* didefinisikan sebagai berikut:

$$\text{Machine - Month} = \frac{(\text{Hari kerja jika 24 jam bekerja})}{(\text{Hari kerja dalam 1 bulan})}$$

Sementara itu, untuk mendapatkan nilai “Hari kerja jika 24 jam bekerja” dapat menggunakan rumus:

$$\text{Hari kerja jika 24 jam bekerja} = \frac{(\text{Waktu pengerjaan dalam jam})}{(24 \text{ jam})}$$

Kemudian, untuk nilai “Hari kerja dalam 1 bulan” diasumsikan sama dengan 20 hari, yang berarti 5 hari bekerja dalam seminggu selama 4 minggu. Dengan asumsi hari kerja dalam 1 bulan adalah 20 hari, maka dapat disimpulkan bahwa waktu kerja dalam 9 bulan ialah $9 \times 20 \text{ hari} = 180 \text{ hari}$. Perhitungan *Machine-Month* dari pembuatan sistem monitoring ini dapat dilihat pada Tabel 10.

Tabel 10. Tabel perhitungan *Machine-Month*

<i>Machine</i>	Hari Kerja dalam 9 Bulan	Hari Kerja dalam 1 Bulan	Waktu Pengerjaan	Hari Kerja jika 24 Jam Bekerja	<i>Machine-Month</i>
Laptop Lenovo G40-45 6410M	$5 \times 36 = 180$ Hari	$180/9 = 20$ Hari	264 Jam	$264/24 = 11$ Hari	$11/20 = 0,55$ <i>Machine-Month</i>
Laptop Acer Swift 3	180 Hari	20 Hari	324 Jam	13,5 Hari	0,675 <i>Machine-Month</i>
Laptop Lenovo IdeaPad 1 14ALC7	180 Hari	20 Hari	432 Jam	18 Hari	0,9 <i>Machine-Month</i>

3.3. *Development tools*

Detail dari perangkat keras dan perangkat lunak yang digunakan dalam proses pengembangan.

3.3.1. Perangkat Keras

A. PC/Komputer

Pada proyek ini, komputer digunakan sebagai antarmuka untuk mengimplementasikan sistem monitoring dan deteksi DoS/DDoS. Hal ini dikarenakan sistem operasi yang digunakan pada server adalah ClearOS.

B. Laptop

Pada sistem monitoring dan deteksi DoS/DDoS ini, laptop digunakan sebagai tempat pemasangan VirtualBox. Setelah itu dilakukan pemasangan sistem operasi kali linux dan ubuntu pada VirtualBox tersebut.

C. Server

Server adalah sebuah sistem komputer yang terdapat pada jaringan komputer untuk menyediakan suatu layanan kepada pengguna yang disebut sebagai client [24]. Server digunakan sebagai tempat pemasangan perangkat lunak yang berkaitan dengan sistem monitoring DoS/DDoS ini. Selain itu semua log serangan DoS/DDoS juga akan disimpan di server.

3.1.1. Perangkat Lunak

A. VirtualBox

VirtualBox adalah perangkat lunak virtualisasi, yang dapat digunakan untuk mengeksekusi sistem operasi tambahan di dalam sistem operasi utama. VirtualBox berfungsi untuk melakukan virtualisasi sistem operasi. VirtualBox juga dapat digunakan untuk membuat virtualisasi jaringan

komputer [25]. Pada sistem yang dibuat, VirtualBox digunakan untuk menjalankan sistem operasi Kali Linux dan Ubuntu.

B. ClearOS

ClearOS adalah Linux yang di kostumasi khusus untuk keperluan server. Dengan berbagai fitur yang *powerfull* dan *setting* yang sederhana, ClearOS menjadi alternatif pilihan, baik untuk pemula yang tidak mengerti Linux sama sekali maupun untuk profesional yang memerlukan kemampuan terbaik dari OS Linux *server* [23].

C. Ubuntu

Ubuntu adalah sistem operasi turunan dari distro Linux jenis Debian *unstable* (sid), Ubuntu merupakan *project* untuk komunitas, yang bertujuan untuk menciptakan sebuah sistem operasi beserta dengan paket aplikasinya yang bersifat *free* dan *open source*, karena Ubuntu mempunyai prinsip untuk selamanya gratis dan tidak ada tambahan untuk versi *enterprise edition* [24]. Ubuntu merupakan salah satu sistem operasi yang digunakan dalam sistem ini. Ubuntu akan berperan sebagai pusat untuk melakukan pemasangan *tools* SIDS dan *log management system*.

D. Kali Linux

Kali Linux merupakan sistem operasi *open source* yang dapat digunakan untuk *penetration testing* terhadap suatu sistem dan jaringan komputer. Terdapat lebih dari 300 *tools* dengan fungsi masing-masing yang dapat digunakan untuk melakukan pengujian keamanan terhadap suatu sistem jaringan. Kali Linux dikembangkan dan didanai oleh Offensive Security [25]. Kali Linux merupakan salah satu sistem operasi yang digunakan dalam sistem, Sistem operasi Kali Linux akan digunakan sebagai *client* atau *attacker* yang akan meluncurkan serangan DoS/DDoS.

E. Snort

Snort adalah sistem pendeteksi dan pencegahan intrusi jaringan sumber terbuka. Ini dapat menganalisis analisis lalu lintas dan aliran data secara real-time dalam jaringan. Snort mampu memeriksa analisis protokol dan dapat mendeteksi berbagai jenis serangan. Dalam NIDS, Snort pada dasarnya memeriksa paket melawan aturan yang ditulis oleh pengguna. Aturan Snort dapat ditulis dalam bahasa apa pun, strukturnya juga baik dan mudah dibaca, serta aturan dapat dimodifikasi juga [26].

F. Apache Kafka

Apache Kafka adalah sebuah *platform open-source* yang digunakan untuk mengelola aliran data secara *real-time*. Apache Kafka pada awalnya dikembangkan di LinkedIn dan sekarang menjadi bagian dari yayasan Apache Software Foundation. Apache Kafka dirancang untuk menangani volume data yang besar dan memungkinkan pertukaran data yang efisien antara *producer* dan *consumer* [27].

G. ELK Stack

Elasticsearch Logstash Kibana (ELK Stack) merupakan komponen yang tepat dalam membangun log event management. ELK Stack dapat memberi informasi kepada sistem

administrator mengenai tren, statistik, dan anomali yang terjadi. ELK Stack merupakan kumpulan dari tiga alat yaitu Elasticsearch, Logstash, dan Kibana [28].

3.4. *Test equipment*

Gambaran pengujian yang akan dilakukan untuk menguji produk yang diusulkan beserta instrumen yang digunakan.

Contoh:

Pengujian pada sistem ini akan dilakukan dengan menggunakan dataset dan pengujian secara langsung. Pengujian secara langsung akan digunakan dataset dari CICDoS2017 dan CICDDoS2019. Sementara itu, Untuk keperluan pengujian secara langsung digunakan beberapa perangkat lunak DoS dan DDoS yang dipasang pada sistem operasi kali linux. Perangkat lunak tersebut akan digunakan untuk menguji sistem yang sudah terpasang dengan *Smart Intrusion Detection System* (IDS). Perangkat lunak ini dibagi sesuai dengan jenis DoS/DDoS yang akan diuji. Berikut beberapa perangkat lunak DoS/DDoS yang akan digunakan untuk melakukan pengujian yang bisa dilihat pada Tabel 11.

Tabel 11. Perangkat lunak untuk pengujian DoS/DDoS

Jenis Serangan	Nama Perangkat Lunak DoS/DDoS
<i>Slow HTTP DoS</i>	- Slowhttptest
<i>DNS Amplification DoS/DDoS</i>	- Ethanwilloner - Saddam - NSREFLECTDNS Flooder v1.1 - DNSDRDOS

3.5. *Kebutuhan pakar*

Jika membutuhkan saran ahli dalam mengembangkan sistem, maka tambahkan di sini judgment dari ahli tersebut.

Contoh:

Pengembangan proyek Sistem Monitoring dan Deteksi Serangan DoS/DdoS menggunakan *Smart Intrusion Detection System* (SIDS) berbasis algoritma *machine learning* yang akan diimplementasikan di Ditreskrimsus Polda Jateng Semarang ini tidak memerlukan kebutuhan pakar dikarenakan pada proyek ini hanya diperlukan koordinasi antara stakeholder dan pengembang mengenai proyek agar sesuai dengan tujuan dan keinginan yang ditetapkan.

4. Kesimpulan

Kesimpulan yang dapat diambil dari dokumen ini adalah:

- a. Snort digunakan sebagai alat Intrusion Detection System (IDS) terhadap serangan DoS/DDoS.
- b. Pendekatan yang digunakan adalah menggunakan algoritma machine learning.
- c. Jenis serangan DoS/DDoS yang akan dideteksi yaitu, slow HTTP DoS, DNS amplification DoS/DDoS, dan DRDoS (Distributed Reflection Denial of Service). Tiga jenis serangan ini akan diuji menggunakan dataset CICDoS2017 dan CICDDoS2019. Selain itu dilakukan juga uji secara langsung menggunakan tools yang dipasang pada sistem operasi Kali Linux.
- d. Perbedaan serangan DoS dan DDoS terletak pada subjek yang melakukan penyerangan. Serangan DoS biasanya dilakukan dari satu sumber atau satu lokasi. Sementara itu, serangan DDoS biasanya dilakukan oleh bot yang telah dikonfigurasi oleh penyerang.
- e. Sistem operasi yang digunakan adalah berbasis Linux.

Template dan Dokumen C200

Topik Capstone	Topik Capstone	
Siklus / Tahun	*Gasal atau Genap / (tahun)	
Judul Dokumen	Capstone TA Judul Capstone Proyek kelompok	
Jenis Dokumen	SPESIFIKASI Catatan: Penggunaan dan penyebaran dokumen ini dikendalikan oleh Departemen Teknik Komputer Universitas Diponegoro	
Nomor Dokumen	(C200.[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
Pembimbing 1 (Utama)	Nama		Tanda Tangan
	NIP.		
Pembimbing 2	Tanggal		
	Nama		Tanda Tangan

	NIP. Tanggal		
--	-----------------	--	--

Versi, Tanggal, Oleh	Perbaikan

Daftar Isi

1.	<u>Pendahuluan</u>	39
1.1.	<u>Ringkasan isi Dokumen</u>	39
1.2.	<u>Aplikasi Dokumen</u>	39
1.3.	<u>Referensi</u>	39
1.4.	<u>Daftar Singkatan</u>	39
2.	<u>Spesifikasi Sistem dan Pengguna</u>	40
2.1.	<u>Gambaran Sistem</u>	40
2.2.	<u>Batasan Sistem</u>	40
2.3.	<u>Kebutuhan Fungsional</u>	41
2.3.1.	<u>Deskripsi Fitur Utama</u>	41
2.3.2.	<u>Alur Proses Utama</u>	42
2.4.	<u>Kebutuhan Data</u>	46
2.5.	<u>Kebutuhan Non Fungsional</u>	47
2.5.1.	<u>Kinerja Sistem</u>	48
2.5.2.	<u>Keamanan Sistem</u>	48
2.5.3.	<u>Keandalan Sistem</u>	49
2.5.4.	<u>Kebutuhan Skalabilitas</u>	49
2.5.5.	<u>Kebutuhan Teknologi dalam Lingkungan Operasional</u>	50
2.6.	<u>Kebutuhan Arsitektur Sistem</u>	51
2.7.	<u>Karakteristik Pengguna / Role & Permission</u>	52
2.8.	<u>Target Sistem</u>	53
3.	<u>Kebutuhan Antarmuka Sistem</u>	53
3.1.	<u>Antarmuka Perangkat Keras</u>	53
3.2.	<u>Antarmuka Perangkat Lunak</u>	53
3.3.	<u>Antarmuka Komunikasi</u>	54
3.4.	<u>Kebutuhan Integrasi dengan Sistem Lain (<i>Optional</i>)</u>	54

Pendahuluan

Ringkasan isi Dokumen

Jelaskan dengan singkat mengapa dokumen ini dibuat. Sertakan pernyataan tujuan spesifik, seperti mendefinisikan ruang lingkup sistem, mendokumentasikan kebutuhan pengguna, dan memastikan pemahaman yang seragam di antara tim pengembang dan pemangku kepentingan.

Contoh:

Dokumen ini disusun untuk memberikan deskripsi mendetail tentang spesifikasi sistem "Sistem Monitoring dan Deteksi Serangan Slow DoS/DRDoS Berbasis Ensemble Learning menggunakan ELK Stack". Sistem ini dirancang untuk Ditreskrimsus Polda Jateng Semarang dengan tujuan:

- Mengidentifikasi dan memitigasi serangan Slow DoS dan DRDoS pada infrastruktur jaringan.
- Memberikan informasi real-time mengenai aktivitas mencurigakan.
- Menghasilkan laporan analitik yang dapat digunakan untuk investigasi lebih lanjut.

Dokumen ini bertujuan untuk memastikan keselarasan antara kebutuhan pengguna dan implementasi sistem melalui spesifikasi yang terperinci.

Aplikasi Dokumen

Dokumen ini berlaku sebagai technical specification design yang berfungsi untuk menjelaskan:

1. Gambaran proses rekayasa atau proses bisnis yang diakomodasi oleh sistem, termasuk ruang lingkup sistem.
2. Target efisiensi/efektivitas yang ingin dicapai.
3. Dokumentasi dari spesifikasi sistem yang dibangun

Referensi

- [5] P. Citra, "Kombinasi Sobel , Canny Dan Otsu Untuk Segmentasi Citra," *Technologia*, vol. 13, no. 2, hal. 102–107, 2022.
- [6] S. C. Re, K. Daya, D. Tanggung, dan J. Perusahaan, "Keselamatan dan Kesehatan Kerja," 2013, Diakses: Feb 10, 2023 Tersedia pada: www.ifro.org.
- [7] "Raspberry Pi 3 Model B+," Diakses: Feb 10, 2023. Tersedia pada: www.raspberrypi.org/products/raspberry.
- [8] IEA Constituent Agreements, 2021. Graduate Attributes and Professional Competences. Available at: <https://www.ieagrements.org/assets/Uploads/IEA-Graduate-Attributes-and-Professional-Competencies-2021.1-Sept-2021.pdf> [Accessed 13 January 2025].

Daftar Singkatan

Akronim	Pengertian
---------	------------

DoS	Denial of Service
DDoS	Distributed Denial of Service
IDS	Intrusion Detection System
IPS	Intrusion Prevention System

Spesifikasi Sistem dan Pengguna

Gambaran Sistem

Jelaskan secara singkat tujuan dari sistem yang akan dibangun. Nyatakan alasan utama mengapa sistem ini dirancang dan apa manfaat utama yang diharapkan dari penerapan sistem.

Contoh:

Sistem Monitoring dan Deteksi Serangan Slow DoS/DRDoS berbasis Ensemble Learning menggunakan ELK Stack dirancang untuk meningkatkan keamanan jaringan Ditreskrimsus Polda Jateng. Sistem ini mengintegrasikan teknologi ELK Stack dengan model machine learning berbasis Ensemble Learning untuk memberikan kemampuan deteksi dini terhadap serangan jaringan. Data log jaringan dikumpulkan dari berbagai sumber, seperti firewall, router, dan server aplikasi, melalui Logstash. Logstash memproses dan memformat data sebelum diteruskan ke Elasticsearch, yang bertugas menyimpan log dalam indeks terstruktur untuk memudahkan pencarian dan analisis data dalam jumlah besar.

Proses deteksi ancaman dilakukan dengan menerapkan model Ensemble Learning yang menggabungkan algoritma seperti Random Forest dan Gradient Boosting. Model ini menganalisis log yang telah disimpan di Elasticsearch untuk mengidentifikasi pola-pola yang mencurigakan, seperti karakteristik serangan Slow DoS atau DRDoS. Hasil analisis ditampilkan melalui Kibana, yang menyediakan visualisasi interaktif dalam bentuk dashboard. Dashboard ini dirancang untuk menampilkan statistik aktivitas jaringan, grafik anomali, serta laporan serangan yang terdeteksi. Selain itu, sistem dilengkapi dengan mekanisme notifikasi real-time yang mengirimkan peringatan kepada admin Ditreskrimsus melalui email atau aplikasi pesan untuk memastikan respons cepat terhadap potensi ancaman.

Keunggulan sistem ini meliputi kemampuan deteksi ancaman dengan akurasi tinggi, skalabilitas yang mendukung pertumbuhan data log yang besar, dan kemudahan integrasi dengan infrastruktur jaringan yang sudah ada di Ditreskrimsus Polda Jateng. Sistem ini dirancang tidak hanya untuk memantau aktivitas jaringan secara real-time, tetapi juga untuk memberikan data analitik yang mendalam guna mendukung investigasi forensik digital. Dengan mengadopsi sistem ini, Ditreskrimsus Polda Jateng diharapkan dapat meningkatkan efektivitas operasional mereka dalam mendeteksi dan mengatasi serangan siber.

Batasan Sistem

Contoh:

1. Pendeteksian DDoS Terbatas pada Layanan Tertentu: Sistem hanya akan mendeteksi serangan DDoS pada perangkat atau layanan IoT yang terhubung dengan jaringan dan tidak mencakup serangan pada infrastruktur jaringan secara umum.
2. Model AI Terbatas pada Jenis Serangan Tertentu: Sistem menggunakan model AI yang terlatih untuk mendeteksi serangan DDoS berbasis pada pola lalu lintas data. Model ini tidak akan mendeteksi jenis serangan lainnya, seperti serangan malware atau phishing.
3. Monitoring Perangkat IoT Tertentu: Hanya perangkat IoT yang terhubung dengan jaringan yang akan dimonitor. Perangkat yang tidak terhubung atau tidak mendukung protokol komunikasi yang kompatibel tidak akan dimonitor.

Kebutuhan Fungsional

Sub-bab ini akan menjelaskan apa yang harus dilakukan oleh sistem untuk memenuhi kebutuhan yang telah dipaparkan pada Gambaran Sistem. Spesifikasi ini berfokus pada fungsi yang akan disediakan oleh sistem, yaitu tindakan atau layanan yang harus mampu dilakukan oleh sistem yang akan dibangun, tanpa membahas secara detail bagaimana fungsi tersebut akan diimplementasikan secara teknis.

Lengkap dengan list tabel Kebutuhan Fungsional.

Tabel 4.9 *Functional requirements*

No	Kategori Pengguna	Kode	Deskripsi Kebutuhan	Prioritas
1	Administrator / Teknisi	LUNA-01	Melakukan Autentikasi	Tinggi
2	Administrator / Teknisi	LUNA-02	Mengakses fitur <i>Dashboard Monitoring</i>	Tinggi
3	Teknisi	LUNA-03	Mengakses fitur <i>Configuration</i>	Tinggi

Deskripsi Fitur Utama

Deskripsikan / gambarkan secara jelas tentang fitur inti pada sistem yang akan dibangun. Lebih baik jika disertai dengan manfaatnya masing-masing.

Contoh:

Luna System digunakan untuk mendeteksi serangan Slow DoS/DRDoS menggunakan pembelajaran ensemble dan Elk Stack. Adapun fungsi utama di dalamnya adalah sebagai berikut

1. Dashboard dan Visualisasi Data

Deskripsi: Dashboard akan disajikan dalam bentuk website. Dashboard ini akan berisi data atau log riwayat serangan DoS/DRDoS serta beberapa visualisasi data yang akan mempermudah administrator dalam melakukan monitoring sistem.

Manfaat: ...

2. Notifikasi real-time

Deskripsi: Sistem ini akan menggunakan aplikasi pesan yaitu Telegram untuk mengirimkan notifikasi kepada administrator. Ketika terjadi serangan Slow DoS/DRDoS, sistem akan mendeteksinya dan akan langsung mengirimkan notifikasi ke aplikasi tersebut.

Manfaat: ...

Alur Proses Utama

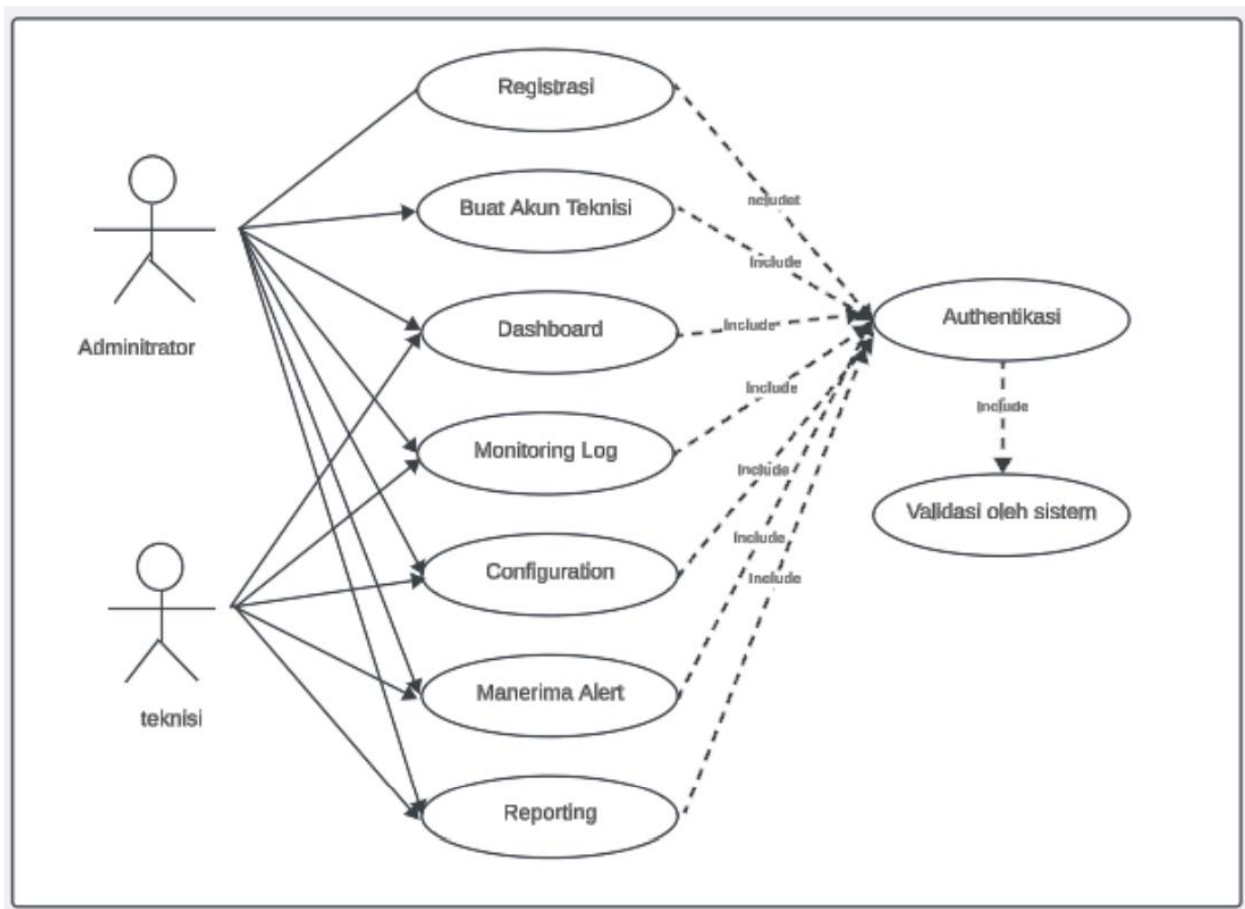
Jelaskan alur bisnis bagaimana sistem bekerja, dilengkapi dengan flowchart secara detail, lebih baik lengkap dengan alur negatifnya. Dapat diperjelas menggunakan Usecase Diagram, Usecase Scenario, dan Activity Diagram/Business Process Model and Notation (BPMN) untuk tiap bisnis proses utama.

Berlaku juga bagi yang memiliki project hardware terkait embedded system / IoT.

Contoh:

Usecase Diagram

Dalam pengembangan Luna System ini terdapat enam Diagram Use Case sesuai dengan enam fitur utama yang dikembangkan yaitu Monitoring Dashboard, Monitoring Log, Configuration Device, Menerima Alert, dan Reporting.



Gambar 1. Usecase Diagram

Usecase scenario

Usecase scenario dari Diagram Use Case Luna System memiliki total 5 bagian untuk setiap fitur utama yaitu Monitoring Dashboard, Monitoring Log, Configuration Device, Menerima Alert, Reporting.

Tabel 1. Tabel *Use Case Scenario* Autentikasi

<i>Use Case ID Number</i>	1	
<i>Use Case Name</i>	Autentikasi	
<i>Use Case Description</i>	Menggambarkan proses autentikasi.	
<i>Primary Actor</i>	Teknisi, Administrator.	
<i>Pre-Condition</i>	<i>User</i> belum <i>login</i> ke sistem.	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> masuk ke halaman	
		2. Sistem menampilkan <i>form login</i>
	3. <i>User</i> memasukkan <i>username</i> dan <i>password</i>	
		4. Sistem melakukan validasi data
	5. Setelah data valid, <i>User</i> melakukan <i>login</i>	
		6. Sistem menampilkan halaman <i>dashboard</i> .
<i>Error Flow of Events</i>	4a. <i>User</i> memberikan data <i>login</i> yang tidak sesuai.	
		4b. Sistem menolak proses <i>login</i> dan menampilkan pesan <i>error</i> .
<i>Post Condition</i>	<i>User</i> berhasil <i>login</i> .	

Tabel 2. Tabel *Use Case Scenario* Membuat Akun Teknisi

<i>Use Case ID Number</i>	2	
<i>Use Case Name</i>	<i>Buat akun teknisi</i>	
<i>Use Case Description</i>	<i>Menggambarkan proses dibuatnya akun teknisi</i>	
<i>Primary Actor</i>	<i>Administrator</i>	
<i>Secondary Actor</i>	<i>Teknisi</i>	
<i>Pre-Condition</i>	<i>Administrator</i> telah <i>login</i> ke sistem.	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. Admin masuk ke menu <i>user management</i>	
		2. Sistem menampilkan menu <i>teknisi</i>
	3. Admin memilih menu “tambahkan <i>teknisi</i> ”	

		4. Sistem mengarahkan ke halaman buat teknisi baru.
	5. Admin memasukkan data teknisi (username, password)	
		6. Sistem menyimpan data dan menambahkan menjadi teknisi baru. Teknisi siap menggunakan akun.
Error Flow of Events	5a. Admin memberikan data yang tidak sesuai.	
		5b. Sistem menolak proses pembuatan akun teknisi dan menampilkan pesan error.
Post Condition	Akun teknisi bisa digunakan	

Tabel 3. Tabel Use Case Scenario Menampilkan Dashboard

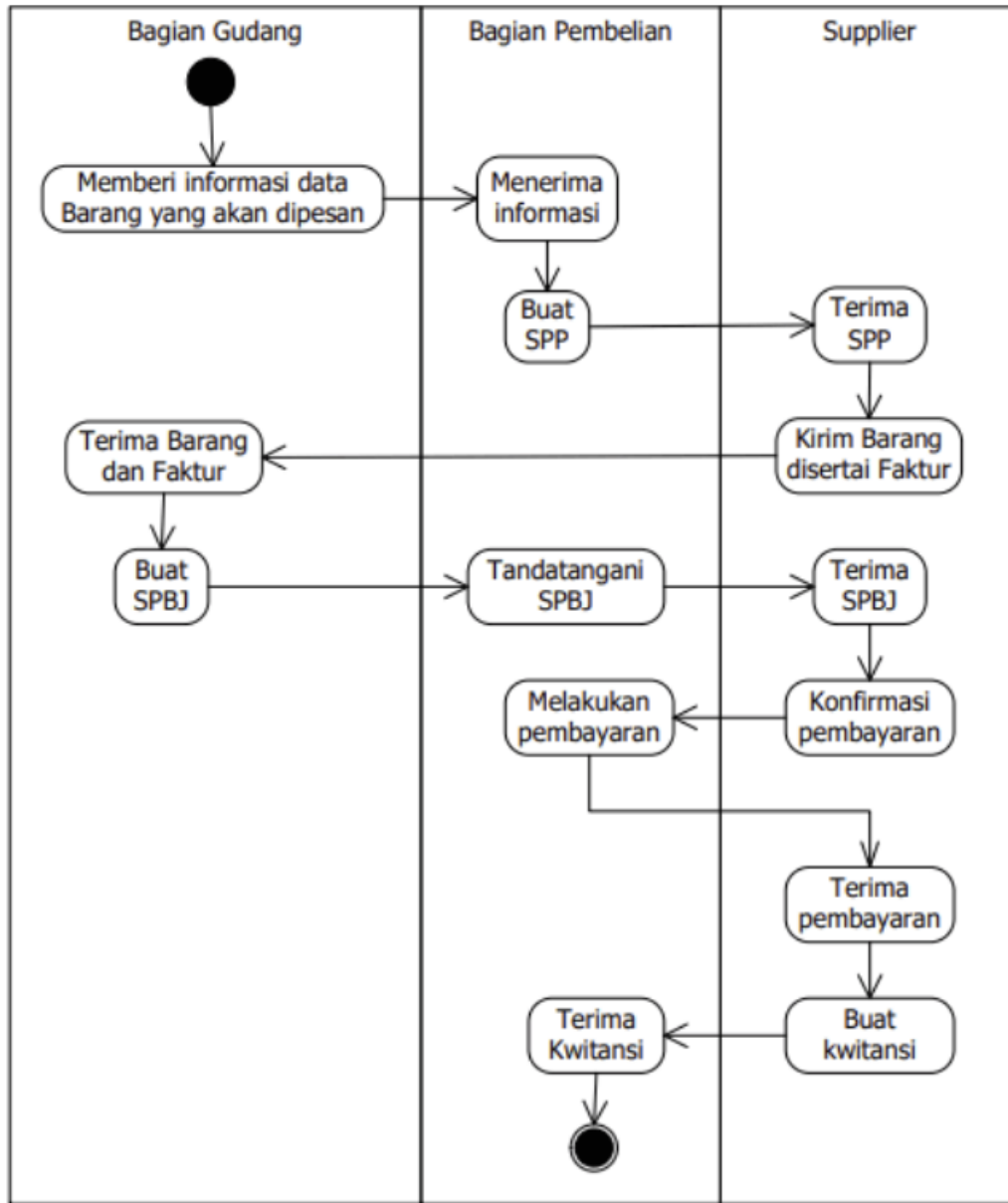
Use Case ID Number	3	
Use Case Name	Menampilkan Dashboard	
Use Case Description	Menggambarkan proses <i>monitoring dashboard</i>	
Primary Actor	Administrator, Teknisi	
Secondary Actor		
Pre-Condition	User telah login ke sistem	
Primary Flow of Events	User Action	System Response
	1. User memilih menu <i>Dashboard</i>	
		2. Sistem menampilkan menu untuk melihat visual dari data-data hasil deteksi seperti grafik dan <i>activity log</i> .
Error Flow of Events		2b. <i>Dashboard</i> tidak menampilkan data karena belum dilakukan deteksi pada <i>traffic</i> jaringan.
Post Condition	User dapat melihat tampilan visual dari data-data hasil deteksi seperti grafik dan <i>activity log</i> .	

Tabel 4. Tabel Use Case Scenario Monitoring Log

Use Case Name	Monitoring Log	
Use Case Description	Meenggambarkan proses <i>monitoring Log</i>	
Primary Actor	Teknisi	
Secondary Actor	Administrator	
Pre-Condition	Teknisi telah login ke sistem	
Primary Flow of Events	User Action	System Response
		1. Sistem melakukan pengamatan pada lalu lintas jaringan, permintaan HTTP, atau kejadian keamanan.

		2. Sistem melakukan Analisis pola lalu lintas <i>log</i>
		3. Sistem menerapkan teknik deteksi anomali untuk membandingkan perilaku lalu lintas saat ini dengan <i>baseline</i> normalitas.
	4. <i>User</i> mengamati proses <i>monitoring</i> dan melakukan penanganan apabila ditemukan serangan keamanan.	
<i>Error Flow of Events</i>		1a. Sistem tidak menampilkan proses pengamatan pada lalu lintas jaringan, permintaan HTTP, atau kejadian keamanan dikarenakan belum berjalannya proses deteksi.
<i>Post Condition</i>	<i>User</i> mengamati proses, melakukan penanganan, dan melakukan perbaikan pada infrastruktur jika diperlukan.	

Activity Diagram



Gambar 2. Contoh Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas.

Jelaskan alur secara deskriptif beserta aturannya jika ada.

Kebutuhan Data

Kebutuhan data merujuk pada semua jenis data yang diperlukan oleh sistem untuk berfungsi dengan baik. Bagian ini biasanya menjelaskan secara rinci tentang data apa saja yang

harus dikumpulkan, diolah, disimpan, dan dikelola oleh sistem untuk memenuhi tujuan dan persyaratan yang ditetapkan.

1. *Jenis Data: Mengidentifikasi tipe data yang akan digunakan, misalnya log jaringan, metadata, atau informasi konfigurasi.*
Contoh: Data log dari firewall, server, dan perangkat jaringan lainnya.
2. *Sumber Data: Menjelaskan dari mana data tersebut berasal.*
Contoh: Data log berasal dari perangkat seperti router, switch, atau server.
3. *Format Data: Format atau struktur data yang harus digunakan untuk memastikan kompatibilitas.*
Contoh: JSON, CSV, atau format log lainnya.
4. *Volume Data: Estimasi jumlah data yang akan dikelola oleh sistem, baik dalam ukuran file (MB, GB) maupun jumlah record.*
Contoh: Sistem harus mampu menangani minimal 1.000 log per detik.
5. *Frekuensi Pengumpulan Data: Menjelaskan seberapa sering data akan diperbarui atau dikumpulkan.*
Contoh: Data dikumpulkan secara real-time atau dalam interval tertentu (misalnya, setiap 5 menit).
6. *Penyimpanan Data: Kebutuhan penyimpanan, termasuk kapasitas penyimpanan dan durasi retensi data.*
Contoh: Data log harus disimpan selama 6 bulan untuk keperluan audit dan analisis.
7. *Keamanan Data: Persyaratan perlindungan terhadap data untuk memastikan kerahasiaan, integritas, dan ketersediaannya.*
Contoh: Semua data harus dienkripsi saat transit dan saat disimpan.
8. *Kebutuhan Pemrosesan Data: Menjelaskan bagaimana data akan diolah, seperti analisis, transformasi, atau agregasi.*
Contoh: Data log diproses melalui Logstash untuk parsing dan dikirim ke Elasticsearch untuk analisis lebih lanjut.

Contoh:

Dalam kasus Sistem Monitoring dan Deteksi Serangan Slow DoS/DRDoS, kebutuhan data mencakup:

- Data log yang mencakup informasi tentang alamat IP sumber, waktu akses, jenis permintaan, dan status respons.
- Data pelatihan untuk model Ensemble Learning, seperti pola serangan sebelumnya atau data normal dari aktivitas jaringan.
- Informasi konfigurasi, seperti aturan firewall atau daftar putih/daftar hitam IP.

Kebutuhan Non Fungsional

Spesifikasi non-fungsional berfokus pada "bagaimana sistem beroperasi" dan kualitas yang harus dimiliki oleh sistem untuk memenuhi kebutuhan pengguna dan stakeholder.

Lengkap dengan list tabel kebutuhan Non Fungsional.

Tabel 4.10 Non-functional requirements

SRS-Id	Parameter	Requirement
LUNA-01	<i>Availability</i>	Aplikasi ini dapat beroperasi 7 hari dalam seminggu dan 24 jam dalam satu hari
LUNA-02	<i>Reliability</i>	System akan menjamin minimalisasi tingkat kegagalan dalam pengoperasian
LUNA-04	<i>Portability</i>	Sistem dapat dioperasikan pada komputer yang memiliki sistem operasi <i>Windows</i> dan <i>web browser</i>
LUNA-06	<i>Response time</i>	Memberikan waktu respon maksimal kira-kira 2 detik
LUNA-07	<i>Safety</i>	Sistem menampilkan dan menulis data sesuai dengan kewenangan yang dimiliki

Contoh:

Spesifikasi non-fungsional berfokus pada bagaimana sistem beroperasi serta kualitas yang harus dimiliki oleh sistem untuk memenuhi kebutuhan pengguna dan stakeholder. Pada sistem monitoring dan deteksi serangan Slow DoS/DRDoS berbasis Ensemble Learning menggunakan ELK Stack, beberapa spesifikasi non-fungsional yang perlu dipertimbangkan adalah:

Kinerja Sistem

Deskripsi seperti waktu respons maksimum, throughput, atau kebutuhan bandwidth.

Contoh:

- Waktu Respons Maksimum: Sistem harus mampu memberikan respons dalam waktu kurang dari 5 detik untuk setiap deteksi serangan yang terjadi pada jaringan yang dipantau. Waktu respons ini penting agar petugas di Ditreskrimsus Polda Jateng dapat segera melakukan tindakan mitigasi terhadap serangan yang terdeteksi.
- Throughput: Sistem diharapkan mampu memproses data log jaringan yang masuk dengan throughput minimal 1.000 log per detik. Hal ini memastikan bahwa sistem dapat menangani volume data yang tinggi dari berbagai sumber perangkat dan server yang ada di lingkungan Ditreskrimsus Polda Jateng.
- Kebutuhan Bandwidth: Sistem monitoring ini memerlukan bandwidth minimal 100 Mbps untuk memastikan data yang dikirimkan dari berbagai perangkat jaringan dapat diterima dan dianalisis secara real-time tanpa menyebabkan latensi atau penurunan kualitas deteksi serangan.

Keamanan Sistem

Persyaratan keamanan seperti enkripsi data atau otentikasi multi-faktor.

Contoh:

Keamanan sistem sangat penting, mengingat sistem ini beroperasi dalam konteks jaringan yang sensitif dan rentan terhadap serangan. Beberapa persyaratan keamanan yang harus dipenuhi oleh sistem adalah:

- **Enkripsi Data:** Semua data yang dikirimkan antara perangkat, server, dan aplikasi monitoring harus dienkripsi menggunakan protokol TLS/SSL untuk menjaga kerahasiaan dan integritas data. Hal ini untuk melindungi data yang dapat berisi informasi sensitif, seperti log dan metadata terkait serangan.
- **Otentikasi Multi-Faktor (MFA):** Untuk meningkatkan keamanan akses ke antarmuka administrasi sistem, sistem harus menggunakan otentikasi multi-faktor, yang mengharuskan pengguna untuk memasukkan kombinasi informasi yang lebih dari sekadar username dan password (misalnya, kode OTP atau autentikasi berbasis biometrik).
- **Pengelolaan Akses Berbasis Peran (RBAC):** Sistem harus mengimplementasikan pengelolaan akses berbasis peran untuk memastikan bahwa hanya pengguna yang memiliki izin yang tepat yang dapat mengakses fungsionalitas sensitif, seperti pengaturan deteksi serangan atau analisis data.

Keandalan Sistem

Tingkat ketersediaan atau waktu kerja (uptime) yang diharapkan.

Contoh:

Keandalan sistem berfokus pada ketersediaan dan waktu operasi sistem yang tinggi, yang penting untuk memastikan sistem dapat berfungsi secara terus-menerus tanpa gangguan signifikan.

- **Uptime:** Sistem harus memiliki tingkat ketersediaan atau uptime sebesar 99.9%, yang berarti sistem hanya boleh mengalami downtime maksimal 8 jam per tahun. Hal ini untuk memastikan bahwa sistem monitoring selalu tersedia untuk mendeteksi dan menangani serangan secara real-time.
- **Pemulihan dari Gangguan (Disaster Recovery):** Sistem harus dilengkapi dengan rencana pemulihan bencana yang memungkinkan pemulihan dalam waktu maksimal 2 jam jika terjadi kegagalan sistem besar atau kehilangan data.

Kebutuhan Skalabilitas

Penjelasan kemampuan sistem untuk berkembang dengan meningkatnya jumlah pengguna.

Contoh:

Sistem ini harus mampu berkembang seiring dengan meningkatnya jumlah perangkat yang terhubung ke jaringan dan volume data yang dihasilkan. Kebutuhan skalabilitas mencakup beberapa aspek:

- **Skalabilitas Horizontal:** Sistem harus dirancang agar dapat melakukan skalabilitas horizontal, memungkinkan penambahan server atau node baru ke dalam kluster ELK Stack

jika volume data meningkat atau jumlah perangkat yang dipantau bertambah. Dengan demikian, sistem dapat menangani beban yang lebih tinggi tanpa penurunan kinerja.

- **Skalabilitas Proses Deteksi:** Sistem harus dapat menyesuaikan jumlah dan jenis model machine learning dalam ensemble learning berdasarkan jumlah data dan kompleksitas serangan. Sebagai contoh, jika data log yang masuk meningkat, maka jumlah model ensemble yang digunakan dapat ditingkatkan untuk menjaga akurasi deteksi.
- **Integrasi dengan Sistem Eksternal:** Sistem harus mampu berintegrasi dengan sistem keamanan eksternal lainnya, seperti firewall, IDS/IPS, dan sistem manajemen jaringan lainnya, agar dapat menangani peningkatan jumlah perangkat atau perubahan kebijakan keamanan tanpa gangguan operasional.

Kebutuhan Teknologi dalam Lingkungan Operasional

*Bab ini menggambarkan berbagai **teknologi** yang dibutuhkan untuk mendukung pengembangan dan operasi sistem yang sedang dirancang pada lingkungan operasional. **Kebutuhan teknologi** mencakup perangkat lunak, perangkat keras, dan lingkungan operasional yang diperlukan agar sistem dapat berfungsi dengan baik.*

Penjelasan lingkungan operasional

Perangkat Lunak

Perangkat lunak yang diperlukan untuk mendukung pengembangan, operasi, dan pemeliharaan sistem. Ini mencakup sistem operasi, perangkat lunak aplikasi, pustaka, framework, serta alat bantu lain yang diperlukan untuk mengembangkan dan menjalankan fungsionalitas sistem.

- **Sistem Operasi:**
 - Linux (Ubuntu, CentOS) digunakan sebagai sistem operasi untuk server pusat yang menjalankan sistem monitoring dan analitik.
 - Windows Server bisa digunakan jika ada kebutuhan tertentu untuk aplikasi berbasis Windows.
- **Perangkat Lunak AI:**
 - TensorFlow atau PyTorch untuk membangun model AI yang digunakan untuk mendeteksi pola serangan DDoS.
 - Scikit-learn untuk implementasi algoritma machine learning lainnya, seperti analisis trafik jaringan.
- **Perangkat Lunak Jaringan:**
 - Wireshark untuk pemantauan dan analisis trafik jaringan.
 - Zabbix atau Nagios untuk pemantauan jaringan secara real-time.
- **Database:**
 - MySQL atau PostgreSQL untuk menyimpan data yang berkaitan dengan status perangkat IoT, serangan DDoS, dan log lainnya.
 - InfluxDB atau Elasticsearch untuk menyimpan data time-series dari perangkat IoT yang terus-menerus mengirim data.

- Alat Analitik & Visualisasi:
 - Grafana untuk visualisasi data real-time dari perangkat IoT dan status DDoS.
 - Kibana (untuk Elasticsearch) untuk menganalisis dan menampilkan log serta data yang relevan dengan serangan DDoS

Perangkat Keras

Perangkat keras yang diperlukan untuk menjalankan dan mendukung perangkat lunak dan fungsi sistem. Ini mencakup server, perangkat IoT, serta perangkat pendukung lain seperti sensor, dan perangkat jaringan.

- Server:
 - Server pusat untuk menjalankan aplikasi monitoring dan model AI. Server ini bisa menggunakan spesifikasi seperti:
 - CPU: Intel Xeon atau AMD EPYC untuk pemrosesan data besar dan pembelajaran AI.
 - RAM: 32GB atau lebih, tergantung pada kebutuhan pemrosesan.
 - Penyimpanan: SSD untuk kecepatan akses data tinggi, 1TB atau lebih.
- Perangkat IoT:
 - Sensor jaringan dan perangkat IoT yang digunakan untuk mengumpulkan data trafik dari perangkat yang terhubung di jaringan, seperti sensor IoT Edge Devices atau IoT Gateways yang terhubung ke perangkat lain di jaringan.
 - Raspberry Pi atau ESP32 yang digunakan untuk mengumpulkan data dari perangkat IoT dan mengirimkannya ke server pusat.
- Perangkat Jaringan:
 - Switch dan Router untuk menghubungkan perangkat IoT ke server pusat dan mengelola lalu lintas data.
 - Firewall hardware atau IDS/IPS (Intrusion Detection/Prevention Systems) yang dapat diintegrasikan dengan sistem monitoring untuk melakukan mitigasi otomatis terhadap serangan DDoS yang terdeteksi.
- Perangkat AI dan Komputasi:
 - GPU (Graphics Processing Unit) untuk mempercepat proses pelatihan model AI jika diperlukan, terutama dalam penggunaan deep learning.
 - Edge computing devices untuk mengolah data lebih dekat ke sumbernya (di tempat perangkat IoT berada), mengurangi latensi.

Kebutuhan Arsitektur Sistem

Menyediakan gambaran umum komponen utama dan interaksi antar komponen sistem secara keseluruhan.

Contoh:

Sistem ini dirancang untuk menyediakan gambaran umum tentang bagaimana sistem monitoring jaringan IoT dengan kemampuan deteksi DDoS. Sistem ini mengintegrasikan

komponen utama jaringan IoT dengan algoritma kecerdasan buatan untuk mendeteksi dan merespons potensi ancaman secara otomatis.

Karakteristik Pengguna / Role & Permission

Penjelasan secara rinci siapa saja pengguna sistem yang diusulkan, peran mereka, dan izin akses yang dimiliki masing-masing peran. Informasi ini penting untuk memastikan bahwa setiap pengguna dapat mengakses fitur sistem sesuai dengan tanggung jawab dan kebutuhan mereka.

- *Identifikasi Pengguna*
Sebutkan jenis-jenis pengguna sistem yang ada, berdasarkan peran atau fungsi mereka dalam organisasi. Jelaskan profil pengguna, seperti tingkat keahlian teknis, frekuensi penggunaan sistem, atau kebutuhan mereka terkait sistem.
- *Peran dan Tanggung Jawab (Role)*
Uraikan peran masing-masing pengguna dalam sistem, termasuk tugas utama mereka. Gunakan istilah yang mudah dipahami untuk mendeskripsikan peran, seperti "Administrator", "Analyst", atau "Operator".
- *Hak Akses dan Izin (Permission)*
Jelaskan hak akses yang diberikan kepada masing-masing peran. Deskripsikan fitur atau modul sistem yang dapat diakses oleh pengguna tertentu, serta batasan yang diterapkan untuk menjaga keamanan dan integritas data.

Contoh:

Tabel 3.1 Karakteristik pengguna

NO	Pengguna	Pekerjaan	Hak Akses
1.	Teknisi	Aktor pengguna untuk melakukan konfigurasi terhadap sistem operasi linux jika terjadi kesalahan pada layanan atau sistem.	mendapatkan akses root agar dapat mengkonfigursi layanan pada sistem operasi linux jika terjadi kesalahan.
2.	Administrator	Aktor yang melakukan <i>monitoring</i> serangan Slow DoS/DRDoS	Melakukan <i>monitoring log</i> , Melakukan <i>monitoring dashboard</i> , Menerima <i>alert</i>

1. Description of User : Pelaku

Role : Teknisi

Prerequisite : Teknisi memerlukan akses root

Task description : Melakukan konfigurasi layanan dan sistem jika terjadi kesalahan

Target Sistem

Jelaskan kondisi system existing seperti apa, lalu jelaskan target dari system yang dibuat, lebih baik jika dilengkapi dengan metric (service time) yang akan di optimalkan.

Contoh:

Target service time pada proyek Luna System yang sedang dikembangkan yaitu system monitoring yang dapat mengidentifikasi serangan Denial of Service/Distributed Denial of Service (Slow DoS/DRDoS) di Ditreskrimsus Polda Jateng yaitu:

1. Terbuatnya sistem untuk mendeteksi serangan Slow DoS/DRDoS yang sebelumnya belum ada pada Ditreskrimsus Polda Jateng.
2. Hasil deteksi serangan Slow DoS/DRDoS ke dashboard sistem dapat dilakukan dalam waktu kurang dari 10 detik dari waktu inisial serangan terjadi.
3. Dashboard sistem dapat memonitor log serangan Slow DoS/DRDoS secara real-time.
4. Dapat memberikan notifikasi serangan yang terjadi ke administrator secara real-time.

Kebutuhan Antarmuka Sistem

Merujuk pada deskripsi detail tentang bagaimana sistem akan berinteraksi dengan perangkat keras (hardware) dan perangkat lunak (software) lainnya. Bagian ini penting untuk memastikan bahwa sistem dapat diintegrasikan dengan lancar ke dalam lingkungan operasional yang ada.

Antarmuka Perangkat Keras

Pada sistem deteksi serangan Slow DoS/DRDoS dari Luna System yang diajukan, proses pengembang telah dirancang dengan mempertimbangkan beberapa aspek antarmuka perangkat keras yang diperlukan untuk memfasilitasi interaksi pengguna dengan sistem. Daftar beberapa hardware interface yang telah dipertimbangkan dalam perancangan sistem ini dijelaskan dalam Tabel dibawah

Tabel 3. Antarmuka Perangkat Keras

No	Antarmuka Pengguna	Fungsi
1.	<i>Keyboard</i>	Antarmuka <i>keyboard</i> digunakan untuk memasukkan data ke dalam sistem.
2.	<i>Mouse</i>	Antarmuka <i>mouse</i> digunakan untuk memindahkan objek dari satu tempat ke tempat lain.
3.	<i>Monitor</i>	Antarmuka <i>monitor</i> digunakan untuk melihat tampilan dari sistem informasi.

Antarmuka Perangkat Lunak

Fokus pada kebutuhan yang berkaitan dengan interaksi antar perangkat lunak, library, atau database yang digunakan untuk pertukaran data antara aplikasi atau modul dalam sistem.

Contoh:

1. API RESTful untuk mengakses data log.
2. Integrasi dengan database (misalnya MySQL, Elasticsearch).
3. Kompatibilitas dengan system operasi tertentu (Linux, Windows).
4. Format data yang digunakan (JSON, XML).

Antarmuka Komunikasi

Fokus pada kebutuhan komunikasi di tingkat jaringan, seperti protokol, bandwidth, dan keamanan dalam pertukaran data antar perangkat atau sistem melalui jaringan.

Contoh:

1. Protokol komunikasi yang digunakan, seperti HTTP, HTTPS, MQTT, atau WebSocket.
2. Keamanan komunikasi, seperti penggunaan TLS/SSL untuk enkripsi.
3. Kebutuhan bandwidth untuk mengelola volume data yang besar (misalnya, 1.000 log/detik).
4. Konektivitas antar perangkat melalui jaringan lokal (LAN) atau internet.

Kebutuhan Integrasi dengan Sistem Lain (Optional)

Merujuk pada kemampuan sistem yang sedang dikembangkan untuk berinteraksi, berbagi data, atau bekerja sama dengan sistem atau perangkat lain dalam sebuah ekosistem teknologi. Tujuannya adalah memastikan interoperabilitas dan efisiensi dalam mencapai tujuan fungsional dan operasional.

Contoh:

Dalam konteks *Sistem Monitoring dan Deteksi Serangan Slow DoS/DRDoS Berbasis Ensemble Learning*:

1. Integrasi dengan Sistem SIEM:
 - o Sistem harus dapat mengirimkan data deteksi serangan ke platform SIEM (misalnya, Splunk atau QRadar) untuk korelasi dan pelaporan yang lebih luas.
2. Integrasi dengan Firewall atau IDS/IPS:
 - o Sistem dapat mengirimkan perintah mitigasi serangan ke perangkat firewall untuk memblokir alamat IP yang mencurigakan.
3. Integrasi dengan Dashboard atau Platform Monitoring:
 - o Hasil analisis dari sistem harus dapat diakses oleh pengguna melalui dashboard pihak ketiga, seperti Grafana atau Kibana.
4. Integrasi dengan Sistem Notifikasi:
 - o Sistem perlu mengirimkan peringatan ke email gateway, SMS gateway, atau aplikasi pesan instan (misalnya, Slack atau WhatsApp).
5. Integrasi dengan Database:
 - o Sistem menyimpan data log dan analisis di database yang ada, seperti Elasticsearch, MongoDB, atau PostgreSQL.

Template dan Dokumen C300

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
Pembimbing 1 (Utama)	Nama		Jabatan Anggota
	NIM		
	Tanggal		Tanda Tangan
Pembimbing 2	Nama		Jabatan Anggota
	NIP.		
	Tanggal		Tanda Tangan

Versi, Tanggal, Oleh	Perbaikan

Daftar Isi

1.	Pendahuluan	59
1.1.	Ringkasan isi dokumen	59
1.2.	Aplikasi Dokumen	59
1.3.	Referensi	59
1.4.	Daftar Singkatan	59
2.	Desain Produk yang Diusulkan	60
2.1.	Arsitektur Sistem	60
2.2.	Desain Detail Sistem	61
2.2.1.	Data description	61
2.2.2.	Class Diagram/Sequence Diagram atau diagram yang sesuai	62
2.2.3.	Standar-standar yang dipergunakan	64
2.2.4.	Method/API	65
2.2.5.	Kecerdasan Buatan	66
2.2.6.	Perancangan Alat	67
3.	Verifikasi Desain Produk	68
3.1.	Prototipe, atau	68
3.2.	Hasil Simulasi Awal Produk	68

5. Pendahuluan

5.1. Ringkasan isi dokumen

Isi Ringkasan Dokumen

5.2. 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*

5.3. Referensi

5.4. Daftar Singkatan

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

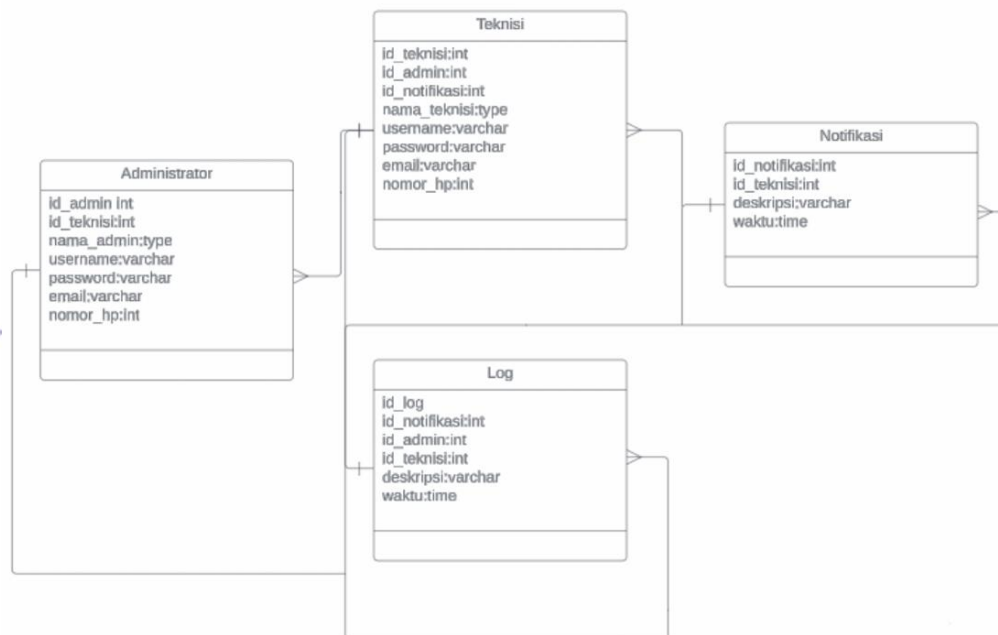
6.2. Desain Detail Sistem

Jika ada jelaskan tentang poin-poin berikut:

6.2.1. 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)		<i>Primary Key, Auto Increment</i>
nama	VARCHAR		
password	VARCHAR	<i>NULL</i>	

Kolom	Tipe Data (Panjang Data)	Bawaan	Keterangan
created_at	datetime	NOW0	
updated_at	datetime	NOW0	

Tabel 3.3 user_activities

Kolom	Tipe Data (Panjang Data)	Bawaan	Keterangan
id	A. bigint(20)		<i>Primary Key, Auto Increment</i>
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	<i>Primary Key</i>	<i>Entity Class</i>	<i>ER</i>	<i>Deskripsi Isi</i>
users	id	User	Need	Berisi daftar data pengguna.
user_activities	id	UserActivity	Need	Berisi daftar rekam jejak aktivitas pengguna dalam sistem.

6.2.2. 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.

2. 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

3. 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.

4. 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.

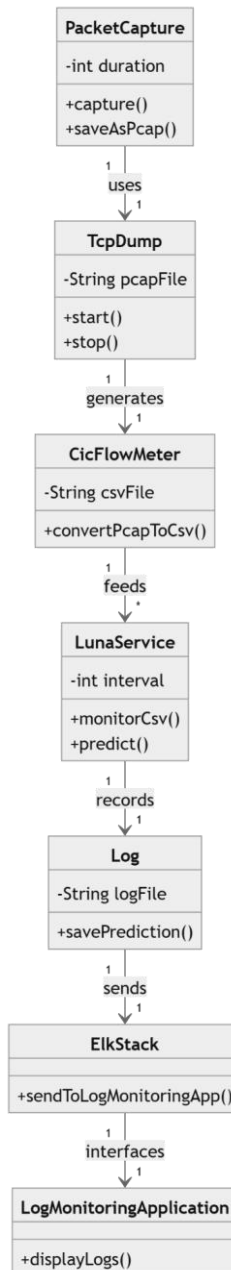
5. Log Monitoring Application

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.



Gambar 3.2 *Class Diagram* LunaSystem

6.2.3. 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:

- C1: Koneksi HTTP (Hypertext Transfer Protocol) dari Klien ke Server untuk menginisiasi permintaan dan menerima respons.
- C2: Koneksi HTTPS (Hypertext Transfer Protocol Secure) dari Server ke Filebeat untuk transmisi data log yang aman.

- c. C3: Koneksi Filebeat dari Filebeat ke Logstash untuk mengirim data log.
- d. C4: Koneksi Logstash dari Logstash ke Elasticsearch untuk melakukan parsing dan pengindeksan data log.
- e. C5: Koneksi Elasticsearch dari Elasticsearch ke Kibana untuk visualisasi data.
- f. C6: Koneksi Elasticsearch dari Elasticsearch ke Elastalert untuk deteksi anomali.
- g. C7: Notifikasi yang dikirim dari Elastalert ke Pengguna ketika anomali terdeteksi.

6.2.4. 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 <i>dashboard</i> .
LUNA-09	getAlert()	Admin mendapatkan peringatan melalui aplikasi telegram
LUNA-10	logout()	Admin mengakhiri sesi <i>login</i> admin yang aktif, sehingga memastikan admin keluar dari sistem dengan aman

6.2.5. 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:

A. 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
<i>Benign</i>	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

B. 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:

- a. **n_estimators**: Jumlah pohon keputusan dalam hutan. Pada penelitian ini, kami menggunakan 100 pohon untuk mendapatkan keseimbangan antara kinerja dan waktu komputasi.
- b. **max_depth**: Kedalaman maksimum pohon keputusan. Kami membatasi kedalaman pohon pada 10 untuk mencegah overfitting.
- c. **min_samples_split**: Jumlah minimum sampel yang dibutuhkan untuk memisahkan suatu node. Nilai yang digunakan adalah 2.
- d. **min_samples_leaf**: Jumlah minimum sampel yang dibutuhkan pada setiap daun pohon. Nilai yang digunakan adalah 1.
- e. **criterion**: Fungsi yang digunakan untuk mengukur kualitas pemisahan. Kami menggunakan "gini" untuk indeks Gini.
- f. **bootstrap**: Apakah akan menggunakan bootstrap samples saat membangun pohon. Kami menggunakan nilai True.

6.2.6. Perancangan Alat

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

B. 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:

- a. *Diagram Blok: Menunjukkan fungsi utama dari setiap komponen dan bagaimana mereka terhubung satu sama lain.*
- b. *Diagram Skematik: Diagram yang lebih rinci yang menunjukkan semua koneksi listrik antara komponen.*
- c. *Diagram PCB (Printed Circuit Board): Diagram layout fisik dari komponen pada papan sirkuit.*

C. Deskripsi Komponen

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

- a. *Mikrokontroler: Spesifikasi teknis dan fungsi dalam sistem.*
- b. *Sensor: Jenis sensor, data yang diukur, dan cara kerja.*
- c. *Aktuator: Jenis aktuator dan fungsinya dalam sistem.*

D. Pengujian Perangkat Keras

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

- a. *Pengujian Fungsional: Menguji apakah setiap komponen bekerja sesuai dengan yang diharapkan.*
- b. *Pengujian Integrasi: Menguji bagaimana komponen bekerja bersama sebagai sebuah sistem.*
- c. *Pengujian Kinerja: Mengukur kinerja sistem dalam kondisi operasi yang berbeda.*

E. Prosedur Kalibrasi

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

7. Verifikasi Desain Produk

Pilih yang digunakan dalam pengembangan produk:

7.1. Prototipe, atau

7.2. 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.

8. Rencana Implementasi dan Pengujian

8.1. Gaant Chart

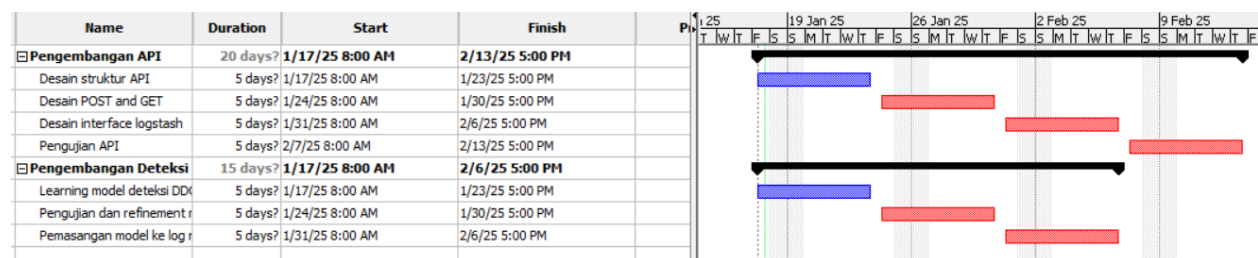
Harus menggunakan Software Project Libre untuk menunjukkan gambaran detail tahapan implementasi dan pengujian yang akan dilakukan. Gantt Chart harus mencakup:

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

Contoh:

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

Tabel 5. 1 Gantt Chart jadwal pengerjaan



8.2. Tujuan Pengujian

8.3. Lingkup Pengujian

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

8.4. Metode Pengujian

Menunjukkan metode pengujian yang akan digunakan ketika produk jadi.

Contoh:

Metode pengujian yang akan digunakan ketika produk jadi meliputi:

a. 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.

b. 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.

c. 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.

d. 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.

e. Pengujian Keamanan:

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

f. Pengujian Kompatibilitas:

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

8.5. Kriteria Pengujian

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

a. Fungsi:

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

b. Kinerja:

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

c. Keamanan:

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

d. Keandalan:

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

Template dan Dokumen C400

Topik Capstone	Topik Capstone	
Siklus / Tahun	Gasal atau Genap / (tahun)	
Judul Dokumen	Capstone TA Judul Capstone Proyek kelompok	
Jenis Dokumen	IMPLEMENTASI PRODUK Catatan: Penggunaan dan penyebaran dokumen ini dikendalikan oleh Departemen Teknik Komputer Universitas Diponegoro	
Nomor Dokumen	C400.[NoRev]TA[tahun].[1/2].[KodeKelompok]	
Nomor Revisi	NoRev	
Nama File	KodeKelompok.doc	
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	NIP.	Tanda Tangan
	Tanggal		
Pembimbing 2	Nama	NIP.	Tanda Tangan
	Tanggal		

Versi, Tanggal, Oleh	Perbaikan

Daftar Isi

<u>1.</u>	<u>Pendahuluan</u>	75
1.1.	<u>Ringkasan isi dokumen</u>	75
1.2.	<u>Aplikasi Dokumen</u>	75
1.3.	<u>Referensi</u>	75
1.4.	<u>Daftar Singkatan</u>	75
<u>2.</u>	<u>Implementasi</u>	75
2.1.	<u>Implementasi Produk</u>	75
2.2.	<u>Tampilan Produk</u>	82
2.3.	<u>Demonstrasi Produk</u>	86
2.4.	<u>Kesimpulan Realisasi Desain Produk dengan Implementasi Produk</u>	86

9. Pendahuluan

9.1. Ringkasan isi dokumen

Isi Ringkasan Dokumen

9.2. Aplikasi Dokumen

Dokumen ini berlaku berfungsi untuk menjelaskan:

- 7) Tahap implementasi dari rancangan pada Dokumen C300
- 8) Acuan tahap pengujian pada dokumen selanjutnya
- 9) Catatan proses pengerjaan dan revisi yang sedang berlangsung

9.3. Referensi

9.4. Daftar Singkatan

10. Implementasi

10.1. Implementasi Produk

Jelaskan dengan singkat gambaran implementasi produk yang dibuat.

Contoh:

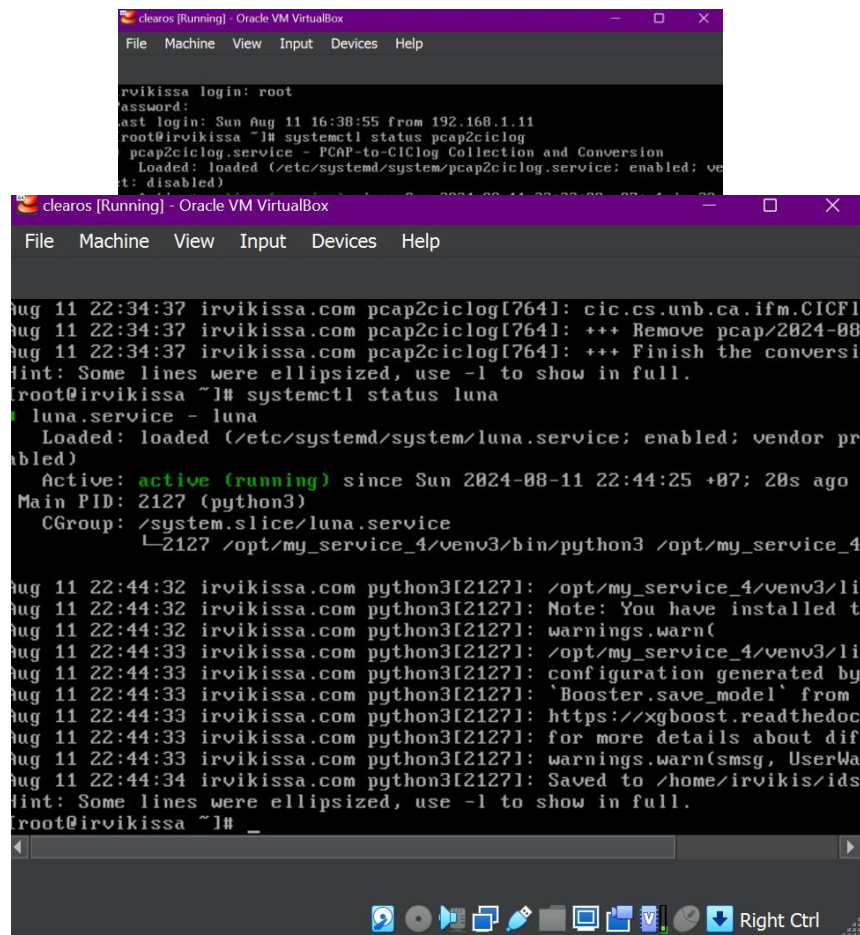
Implementasi LunaSystem Sistem *Monitoring* dan Deteksi Serangan Slow DoS/DRDoS Berbasiskan *Ensemble Learning* menggunakan ELK Stack pada Ditreskrimsus Polda Jateng Semarang dilakukan dengan pembuatan sistem yang dibagi berdasarkan 2 bagian, yaitu pembuatan model ensemble dengan menggunakan dataset CIC-IDS2017, CICCSEIDS2018, dan CICDDOS2019 setelah itu dilanjut dengan pembuatan *Log Management System* yang terintegrasi dengan *Web Application*.

Pembuatan model ensemble dibagi menjadi beberapa tahap yaitu, tahap *preprocessing data*, tahap *feature engineering*, dan yang terakhir adalah tahap pelatihan dan pengujian. Tahap *preprocessing* adalah langkah yang paling awal dilakukan, data mentah yang diambil dari dataset diubah dilakukan beberapa langkah, seperti menghapus data *redundant* yang tidak perlu dan menormalisasi data, sehingga data tersebut siap digunakan untuk dilatih dan diuji. Tahap berikutnya yaitu melakukan *feature engineering*. Pada tahap ini dilakukan dua tahap *feature engineering* yaitu melakukan *feature selection* dengan menggunakan *information gain* dan *Fast Correlation Base Filter*. Feature selection dengan menggunakan *information gain* adalah metode

untuk memilih fitur-fitur yang paling relevan dalam dataset berdasarkan seberapa baik fitur-fitur tersebut memisahkan data ke dalam kategori-kategori yang berbeda. *Fast Correlation-Based Filter (FCBF)* adalah metode *feature selection* yang efisien untuk memilih fitur-fitur paling relevan dari dataset. FCBF mengurangi redundansi dengan mempertahankan hanya fitur-fitur yang memiliki korelasi signifikan dengan target (label) yang ingin diprediksi. FCBF menghitung *C-correlation* (korelasi antara fitur dan kelas) dan *F-correlation* (korelasi antara dua fitur), kemudian menyaring fitur-fitur yang *C-correlation*-nya kurang dari ambang batas yang telah ditentukan dan menghapus fitur-fitur redundant dengan menggunakan Markov blanket yang telah diidentifikasi[1]. Masuk ke tahap pelatihan dan pengujian. Pada tahap ini dataset dibagi menjadi 70 persen untuk data latih dan 30 persen untuk data uji. Hasil atau model dari tahap pelatihan kemudian disimpan dalam bentuk joblib. Model ini akan digunakan untuk memprediksi trafik secara *real time*.

Gambar 2.1. *Service pcap2ciclog*

Gambar 2.1. menunjukkan sebuah *service* pcap2ciclog yang berguna untuk melakukan penangkapan trafik menggunakan TCPDump. *Service* tersebut akan berjalan selama t detik,



```
clearos [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

irvikissa login: root
password:
Last login: Sun Aug 11 16:38:55 from 192.168.1.11
root@irvikissa ~]# systemctl status pcap2ciclog
pcap2ciclog.service - PCAP-to-CIClog Collection and Conversion
Loaded: loaded (/etc/systemd/system/pcap2ciclog.service; enabled; vendor preset: disabled)
Active: active (running) since Sun 2024-08-11 22:44:25 +07; 20s ago
Main PID: 2127 (python3)
CGroup: /system.slice/luna.service
└─2127 /opt/my_service_4/venv3/bin/python3 /opt/my_service_4/venv3/li

Aug 11 22:34:37 irvikissa.com pcap2ciclog[764]: cic.cs.unb.ca.ifm.CICFI
Aug 11 22:34:37 irvikissa.com pcap2ciclog[764]: +++ Remove pcap/2024-08
Aug 11 22:34:37 irvikissa.com pcap2ciclog[764]: +++ Finish the conversi
Hint: Some lines were ellipsized, use -l to show in full.
root@irvikissa ~]# systemctl status luna
luna.service - luna
Loaded: loaded (/etc/systemd/system/luna.service; enabled; vendor preset: disabled)
Active: active (running) since Sun 2024-08-11 22:44:25 +07; 20s ago
Main PID: 2127 (python3)
CGroup: /system.slice/luna.service
└─2127 /opt/my_service_4/venv3/bin/python3 /opt/my_service_4/venv3/li

Aug 11 22:44:32 irvikissa.com python3[2127]: /opt/my_service_4/venv3/li
Aug 11 22:44:32 irvikissa.com python3[2127]: Note: You have installed t
Aug 11 22:44:32 irvikissa.com python3[2127]: warnings.warn(
Aug 11 22:44:33 irvikissa.com python3[2127]: /opt/my_service_4/venv3/li
Aug 11 22:44:33 irvikissa.com python3[2127]: configuration generated by
Aug 11 22:44:33 irvikissa.com python3[2127]: 'Booster.save_model' from
Aug 11 22:44:33 irvikissa.com python3[2127]: https://xgboost.readthedoc
Aug 11 22:44:33 irvikissa.com python3[2127]: for more details about dif
Aug 11 22:44:33 irvikissa.com python3[2127]: warnings.warn(msg, UserWa
Aug 11 22:44:34 irvikissa.com python3[2127]: Saved to /home/irvikis/ids
Hint: Some lines were ellipsized, use -l to show in full.
root@irvikissa ~]#
```

kemudian menyimpan trafik tersebut dalam bentuk berkas .pcap. *Service* tersebut juga akan mengkonversi berkas pcap yang sudah dihasilkan ke bentuk berkas csv berdasarkan format *dataset*

yang digunakan.

Gambar 2.2. *Service Luna*

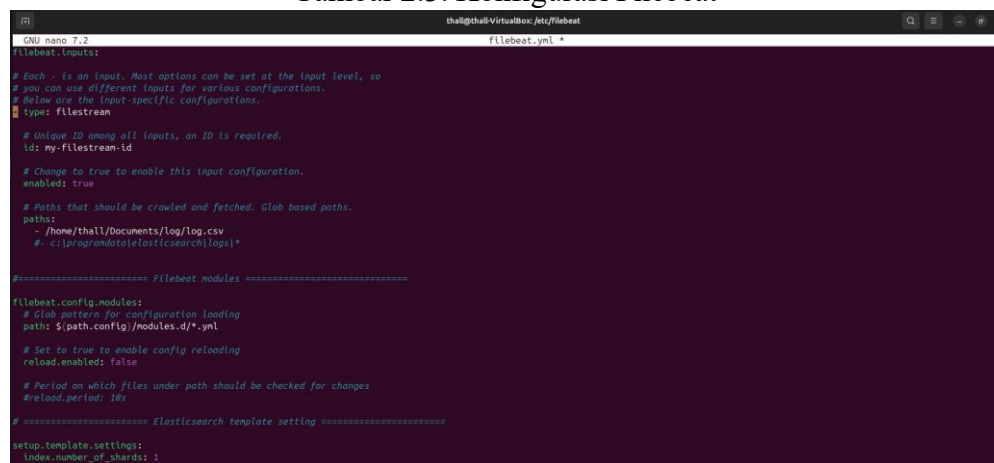
Setelah berkas pcap berhasil dikonversi ke bentuk csv, berkas csv yang berisi trafik tersebut akan dilakukan data *preprocessing* dan *feature engineering* oleh *service Luna*, sama halnya seperti yang dilakukan pada tahap pembuatan model. Setelah itu *service Luna* akan melakukan prediksi terhadap trafik tersebut. Hasil prediksi akan menghasilkan berkas berbentuk csv dengan nama log.csv.

Konfigurasi Filebeat untuk memantau log dari file CSV yang berisi data Slow DoS dan mengirimkannya ke Logstash. Ini memungkinkan data yang relevan untuk dianalisis lebih lanjut atau diproses oleh sistem lain dalam infrastruktur.

Lakukan instalasi pada server yang akan dipantau, dan persiapkan file CSV yang berisi data DoS yang akan dimonitor. Selanjutnya melakukan konfigurasi pada Filebeat, lakukan konfigurasi pada file filebeat.yml. Di dalam filebeat.yml masukkan konfigurasi yang diperlukan seperti melakukan *setting paths* agar memungkinkan user untuk menentukan sumber data log yang akan dipantau dengan fleksibilitas, sehingga memastikan bahwa data log yang relevan dapat diakses dan diproses secara efisien oleh sistem pengolahan data. Konfigurasi dilakukan dengan tujuan untuk menentukan *input log* dari file CSV.

Setelah selesai melakukan konfigurasi pada Filebeat, simpan file konfigurasi dan lakukan *restart* pada layanan Filebeat. Periksa log Filebeat untuk memastikan tidak ada kesalahan dan data dari file CSV telah dikirimkan ke Elasticsearch dengan sukses.

Gambar 2.3. Konfigurasi Filebeat



```
GNU nano 7.2
filebeat.yml
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input-specific configurations.
type: filestream

# Unique ID among all inputs, an ID is required.
id: my-filestream-id

# Change to true to enable this input configuration.
enabled: true

# Paths that should be crawled and fetched. Glob based paths.
paths:
  - /home/thall/Documents/log/log.csv
  #- c:\programdata\elasticsearch\logs\*

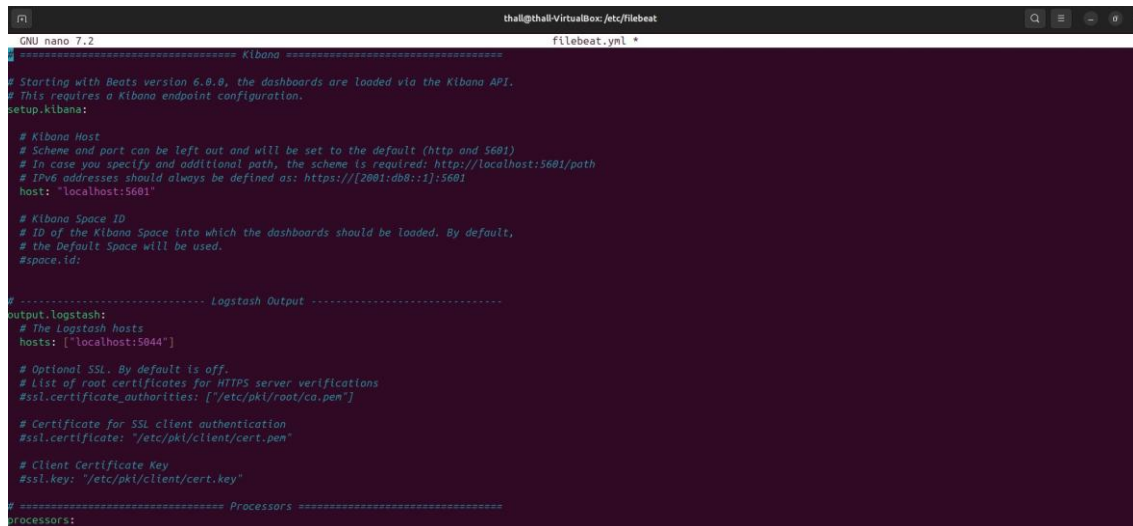
# ===== Filebeat modules =====
filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml

  # Set to true to enable config reloading
  reload.enabled: false

  # Period on which files under path should be checked for changes
  #reload.period: 10s

# ===== Elasticsearch template setting =====
setup.template.settings:
  index.number_of_shards: 1
```

Dalam menggunakan Filebeat, kita harus melakukan konfigurasi terlebih dahulu pada beberapa bagian. Agar log file dapat terbaca oleh Filebeat, lakukan konfigurasi pada bagian *path* dan sesuaikan dengan lokasi log file. Pada bagian *enabled* atur menjadi true agar input filebeat dapat terbaca dan bisa dieksekusi oleh sistem.



```
thall@thall-VirtualBox: /etc/filebeat
GNU nano 7.2 filebeat.yml
# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

# Kibana Host
# Scheme and port can be left out and will be set to the default (http and 5601)
# In case you specify and additional path, the scheme is required: http://localhost:5601/path
# IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
host: "localhost:5601"

# Kibana Space ID
# ID of the Kibana Space into which the dashboards should be loaded. By default,
# the Default Space will be used.
#space.id:

# ----- Logstash Output -----
output.logstash:
# The Logstash hosts
hosts: ["localhost:5044"]

# Optional SSL. By default is off.
# List of root certificates for HTTPS server verifications
#ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

# Certificate for SSL client authentication
#ssl.certificate: "/etc/pki/client/cert.pem"

# Client Certificate Key
#ssl.key: "/etc/pki/client/cert.key"

# ----- Processors -----
processors:
```

Gambar 2.4. Konfigurasi Koneksi ke Logstash

Pada gambar tersebut menjelaskan konfigurasi pada filebeat.yml. Konfigurasi *endpoint* Kibana dan *output* Logstash merupakan bagian penting dalam pengaturan sistem *monitoring* dan manajemen *log* menggunakan ELK stack (Elasticsearch, Logstash, dan Kibana). Pada konfigurasi Kibana, *host* Kibana diatur untuk berjalan di 'localhost' pada *port default* '5601', yang menunjukkan bahwa *dashboard* dan antarmuka pengguna Kibana akan diakses melalui `http://localhost:5601`. ID ruang Kibana tidak diatur, sehingga ruang *default* digunakan untuk mengelompokkan dan mengelola *dashboard* serta visualisasi.

Pada bagian *output* Logstash, *host* Logstash diatur untuk menerima *input* dari 'localhost' pada *port* '5044', memungkinkan Beats seperti Filebeat untuk mengirim data *log* ke Logstash. Konfigurasi ini juga mencakup pengaturan opsional untuk mengaktifkan SSL demi keamanan, dengan opsi untuk menentukan sertifikat otoritas *root*, sertifikat SSL klien, dan kunci sertifikat klien untuk memastikan komunikasi yang aman antara Beats dan Logstash. Selain itu, bagian prosesor mengatur berbagai pemrosesan yang dapat diterapkan pada data sebelum dikirim ke *output*. Prosesor ini berguna untuk memodifikasi, menghapus, atau menambahkan data dalam *log* sesuai kebutuhan, meskipun dalam contoh ini tidak ada pemrosesan spesifik yang diatur. Keseluruhan konfigurasi ini mengarahkan data dari Beats ke Kibana dan Logstash, di mana Kibana digunakan untuk visualisasi data dan Logstash digunakan untuk memproses serta mengirim data ke Elasticsearch untuk penyimpanan lebih lanjut. Pengaturan SSL yang opsional memberikan opsi keamanan tambahan untuk komunikasi antar komponen.

A. Konfigurasi Logstash

Konfigurasi Logstash bertujuan untuk mengonsumsi data yang dikirim dari Elasticsearch dan melakukan pemrosesan data yang sesuai dengan kebutuhan, seperti *parsing*, pengubahan format, atau penyimpanan ke penyimpanan data yang lain. Ini memungkinkan integrasi yang

mulus antara Elasticsearch dan Logstash, memungkinkan aliran data yang lancar dan analisis yang efektif.

Logstash berfungsi sebagai titik tengah dalam arsitektur pengolahan data *log* yang melibatkan Filebeat dan Elasticsearch. Dengan menerima, memproses, dan mengirimkan data *log*, Logstash memainkan peran penting dalam mengelola aliran data dari sumber ke penyimpanan dan analisis. Fungsi utamanya meliputi:

1. **Menerima Data** : Logstash menerima data *log* yang dikirim oleh Filebeat.
2. **Pemrosesan** : Data *log* diproses menggunakan *filter* yang ditentukan, seperti pemetaan, *parsing*, pengubahan format, atau pengayaan data.
3. **Pengiriman** : Data *log* yang telah diproses dikirim ke Elasticsearch untuk penyimpanan dan analisis lebih lanjut.
4. **Integrasi** : Logstash menyediakan integrasi yang kuat dengan berbagai sumber data dan penyimpanan, memungkinkan pengolahan data yang fleksibel dan skalabel.

Dengan mengkonfigurasi Logstash untuk memproses data *log*, kita memastikan bahwa data yang dikumpulkan dari berbagai sumber dapat diproses, diperkaya, dan disimpan dengan efisien sebelum disajikan untuk analisis lebih lanjut. Ini membantu dalam pengelolaan aliran data *log* dalam lingkungan yang kompleks dan memfasilitasi pemahaman yang lebih baik tentang sistem dan aplikasi yang dipantau.

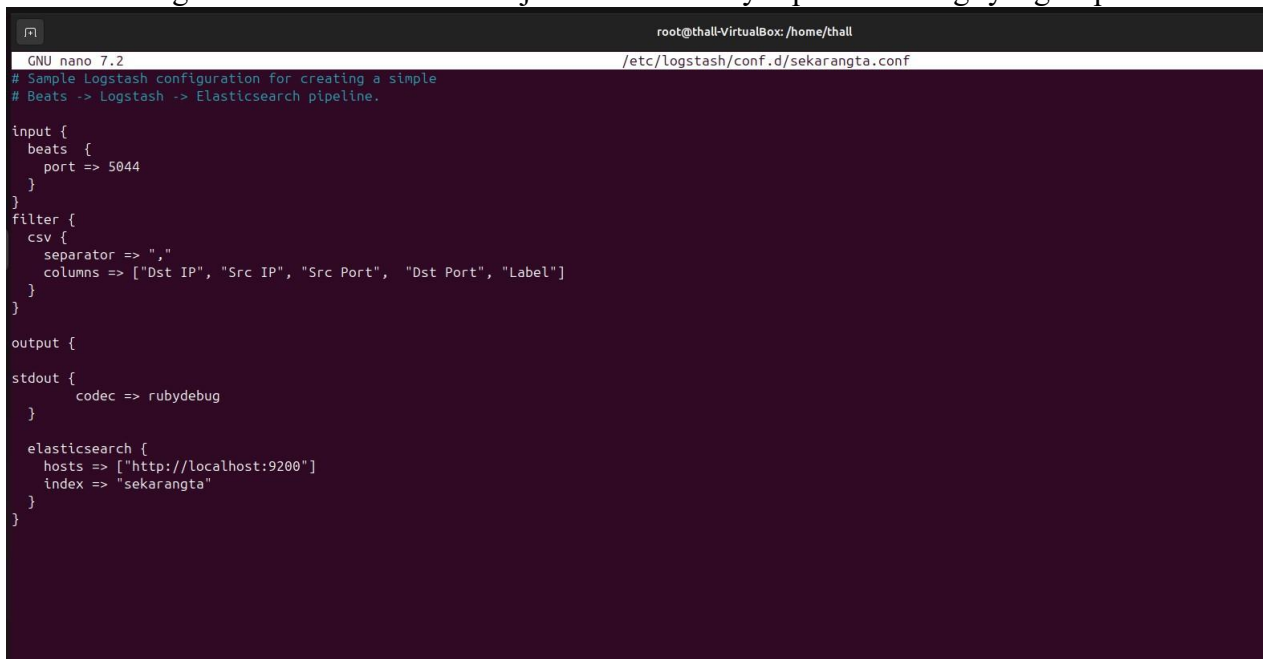
Konfigurasi Logstash ini dirancang untuk menerima, memproses, dan mengirim data ke Elasticsearch sambil juga mencetak data tersebut ke stdout untuk keperluan *debugging*. Pada bagian *input*, Logstash dikonfigurasi untuk menerima data dari *plugin* Beats pada *port* 5044. Beats adalah kumpulan pengirim data ringan seperti Filebeat dan Metricbeat yang mengirim data dari berbagai sumber ke Logstash.

Pada bagian *filter*, Logstash memproses data yang diterima dalam format CSV. *Filter* CSV digunakan untuk memisahkan data berdasarkan koma dan memetakan nilai ke kolom yang telah didefinisikan seperti "Dst IP", "Src IP", "Src Port", "Dst Port", dan "Label". Setelah data diproses, bagian *output* mengirim data ke dua tujuan: stdout dan Elasticsearch. *Output* stdout digunakan untuk mencetak data ke konsol dengan format rubydebug yang mudah dibaca untuk tujuan *debugging*. Sementara itu, *output* Elasticsearch mengirim data yang telah diproses ke *instance* Elasticsearch yang berjalan pada *localhost* pada *port* 9200 dan menyimpannya di indeks bernama "sekarangta". Konfigurasi ini memungkinkan data yang diterima dapat dengan mudah dianalisis dan divisualisasikan menggunakan Elasticsearch dan Kibana, memberikan wawasan yang lebih dalam tentang data tersebut.

Gambar 2.5. Konfigurasi Logstash

B. Konfigurasi Elastisearch

Konfigurasi Elasticsearch bertujuan untuk menyimpan data log yang diproses oleh



```
GNU nano 7.2 /etc/logstash/conf.d/sekarangta.conf
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
  beats {
    port => 5044
  }
}
filter {
  csv {
    separator => ","
    columns => ["Dst IP", "Src IP", "Src Port", "Dst Port", "Label"]
  }
}
output {
  stdout {
    codec => rubydebug
  }
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "sekarangta"
  }
}
```

Logstash dan memfasilitasi penganalisisan data log menggunakan Kibana. Ini membentuk bagian penting dari ELK Stack (Elasticsearch, Logstash, Kibana) yang digunakan untuk pemantauan dan analisis *log*.

Elasticsearch berperan sebagai mesin penyimpanan dan pencarian yang kuat dalam arsitektur ELK Stack. Fungsinya meliputi:

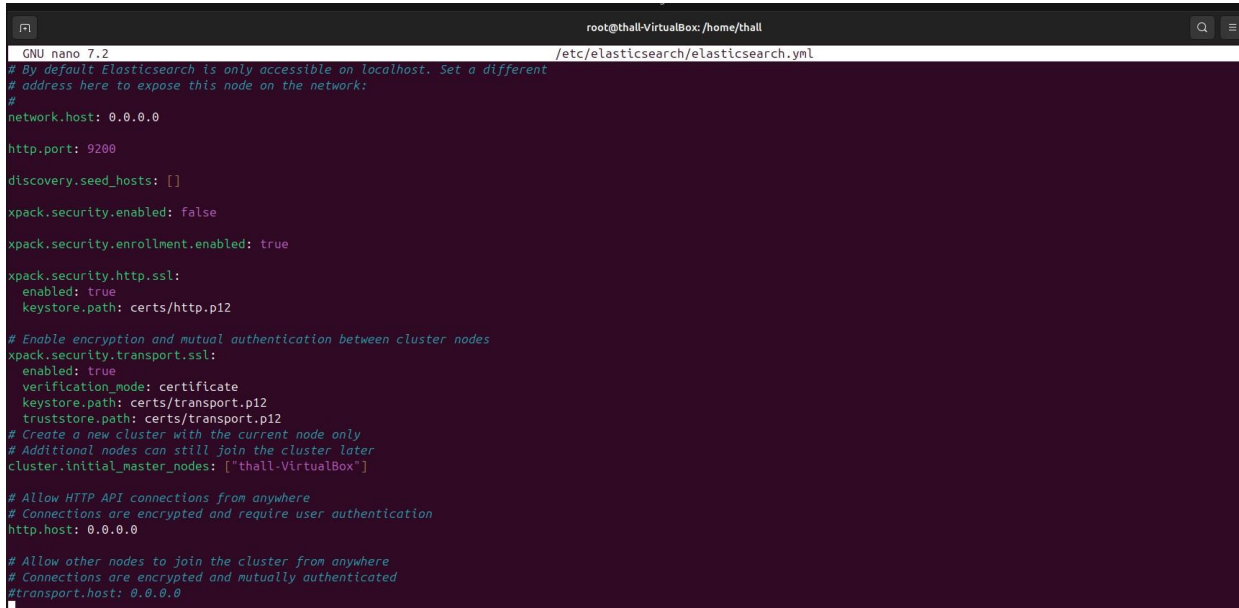
1. **Penyimpanan Data:** Elasticsearch menyimpan data *log* yang diproses oleh Logstash dalam bentuk indeks yang terstruktur dan terdistribusi.
2. **Pencarian dan Penganalisisan:** Elasticsearch menyediakan API pencarian dan penganalisisan yang kuat untuk melakukan kueri dan analisis data *log* dengan cepat dan efisien.

3. **Skalabilitas:** Elasticsearch dirancang untuk skalabilitas horizontal, memungkinkan penambahan *node* dan partisi untuk menangani volume data yang besar.
4. **Integrasi dengan Kibana:** Elasticsearch berintegrasi langsung dengan Kibana, memungkinkan visualisasi dan analisis data *log* yang mudah dan intuitif.

Dengan mengkonfigurasi Elasticsearch untuk menyimpan data *log* dan memfasilitasi pencarian dan analisis data, kita menciptakan fondasi yang kuat untuk pemantauan dan analisis *log* yang efektif. Elasticsearch memainkan peran sentral dalam infrastruktur ELK Stack, memungkinkan pengolahan dan pemahaman yang lebih baik tentang data *log* dari berbagai sumber. Dengan analisis yang kuat dan cepat, kita dapat dengan cepat mengidentifikasi masalah, mendiagnosa penyebabnya, dan mengambil tindakan yang sesuai untuk meningkatkan kinerja dan keamanan sistem.

Konfigurasi Elasticsearch dalam *file* `elasticsearch.yml` mengatur beberapa aspek penting dari *server* Elasticsearch. Pengaturan `network.host: 0.0.0.0` menginstruksikan Elasticsearch untuk mendengarkan koneksi pada semua antarmuka jaringan yang tersedia, memungkinkan akses dari semua alamat IP yang terikat pada mesin tersebut. Ini memberikan fleksibilitas untuk menerima koneksi dari berbagai jaringan atau perangkat. Pengaturan `http.port: 9200` menetapkan *port* 9200 sebagai *port* di mana Elasticsearch akan mendengarkan koneksi HTTP, yang merupakan *port default* untuk layanan Elasticsearch dan digunakan untuk operasi pencarian, pengindeksan, dan administrasi data. Selanjutnya, `discovery.seed_hosts: []` menunjukkan bahwa tidak ada *host seed* yang ditentukan untuk penemuan kluster, yang mungkin berarti *instance* Elasticsearch ini beroperasi sebagai *node* tunggal atau menggunakan metode lain untuk penemuan *node* dalam kluster. Terakhir, `http.host: 0.0.0.0` memastikan bahwa Elasticsearch mendengarkan koneksi HTTP pada semua antarmuka jaringan, memberikan aksesibilitas yang luas untuk layanan HTTP-nya.

Meskipun pengaturan ini memberikan fleksibilitas tinggi dan kemudahan akses, terutama dalam lingkungan pengembangan atau pengujian, penting untuk mempertimbangkan langkah-



```
root@thall-VirtualBox: /home/thall
GNU nano 7.2 /etc/elasticsearch/elasticsearch.yml
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0

http.port: 9200

discovery.seed_hosts: []

xpack.security.enabled: false
xpack.security.enrollment.enabled: true

xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
cluster.initial_master_nodes: ["thall-VirtualBox"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
http.host: 0.0.0.0

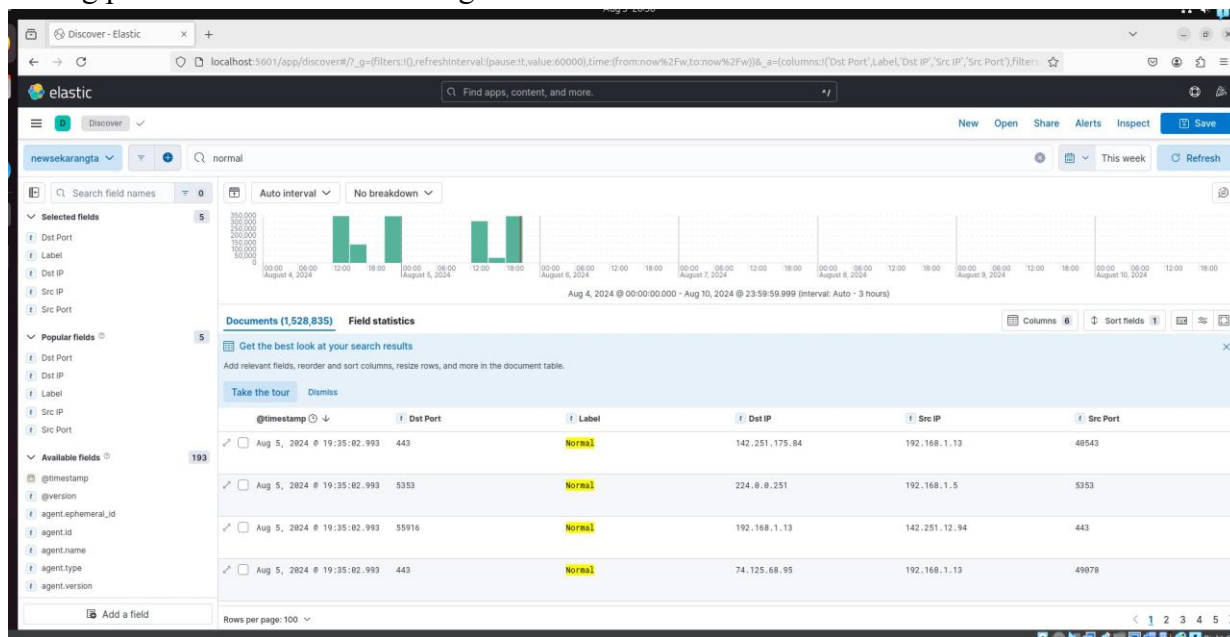
# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
#transport.host: 0.0.0.0
```

langkah keamanan tambahan, seperti konfigurasi Firewall dan penggunaan autentikasi serta enkripsi TLS/SSL, terutama dalam lingkungan produksi, untuk melindungi *instance* Elasticsearch dari akses yang tidak sah.

Gambar 2.6. Konfigurasi Elasticsearch

C. Konfigurasi dan Visualisasi menggunakan Kibana

Proses visualisasi *log* serangan Slow DoS di Kibana dalam ELK Stack dimulai dengan menyiapkan data *log* yang telah dikirim dan tersimpan dalam Elasticsearch, dilanjutkan dengan pembuatan *index pattern* di Kibana untuk mengakses data *log* tersebut, kemudian pengguna dapat membuat berbagai jenis visualisasi seperti histogram, grafik batang, atau peta untuk menampilkan informasi penting seperti pola serangan, jumlah serangan per IP sumber, atau distribusi geografis, yang kemudian disusun dalam *dashboard* untuk memungkinkan pemantauan serangan Slow DoS secara *real-time*, sehingga dengan visualisasi ini, pengguna dapat mengidentifikasi tren serangan, memahami dampaknya terhadap sistem, serta mengambil tindakan yang tepat untuk meningkatkan keamanan dan kinerja sistem secara keseluruhan dengan memberikan pemahaman yang lebih baik tentang pola dan karakteristik serangan Slow DoS.



Gambar 2.7. Visualisasi *Discover* Kibana

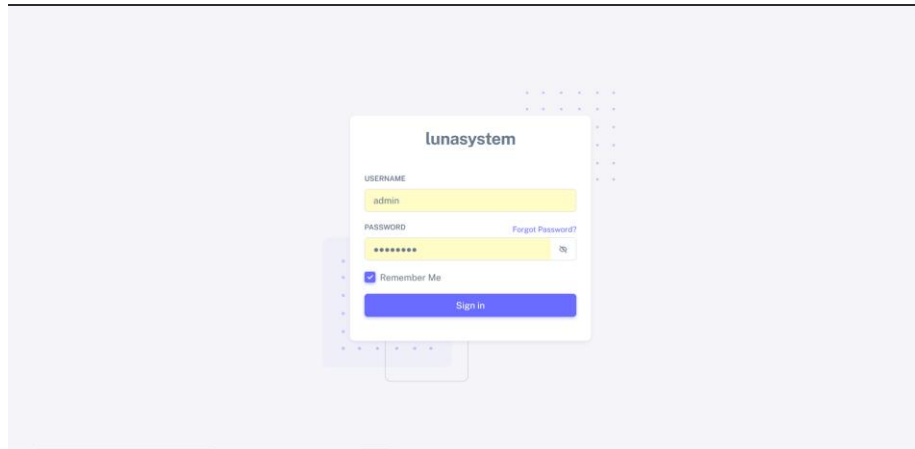
10.2. Tampilan Produk

Tampilkan screenshot dari produk yang dibuat untuk masing-masing role/fungsi yang didesain untuk produk tersebut. Tampilan bisa dibuat per fungsi atau per kelompok user.

Contoh:

2.2.1 Web Application

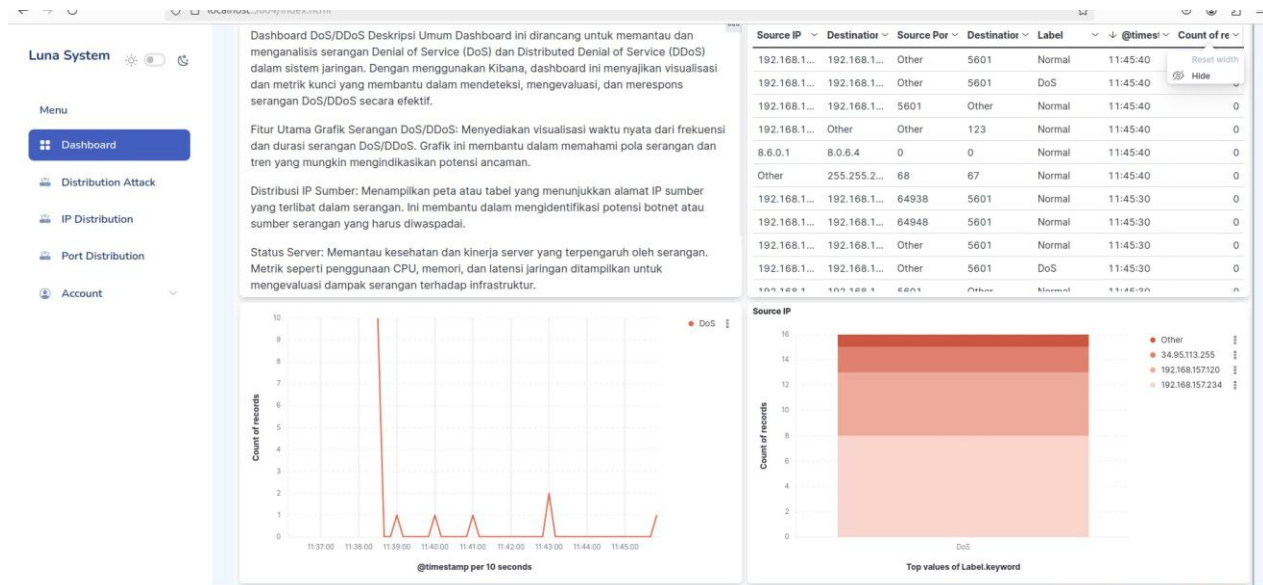
A. Halaman Login



Gambar 2.8. Tampilan Login

Gambar 2.8 merupakan tampilan dari masuk atau *login* web *application* Luna System, halaman ini berisikan *text field* dan button *login* untuk masuk pada halaman *admin*. Admini diminta memasukkan *username* dan *password* yang telah tersedia.

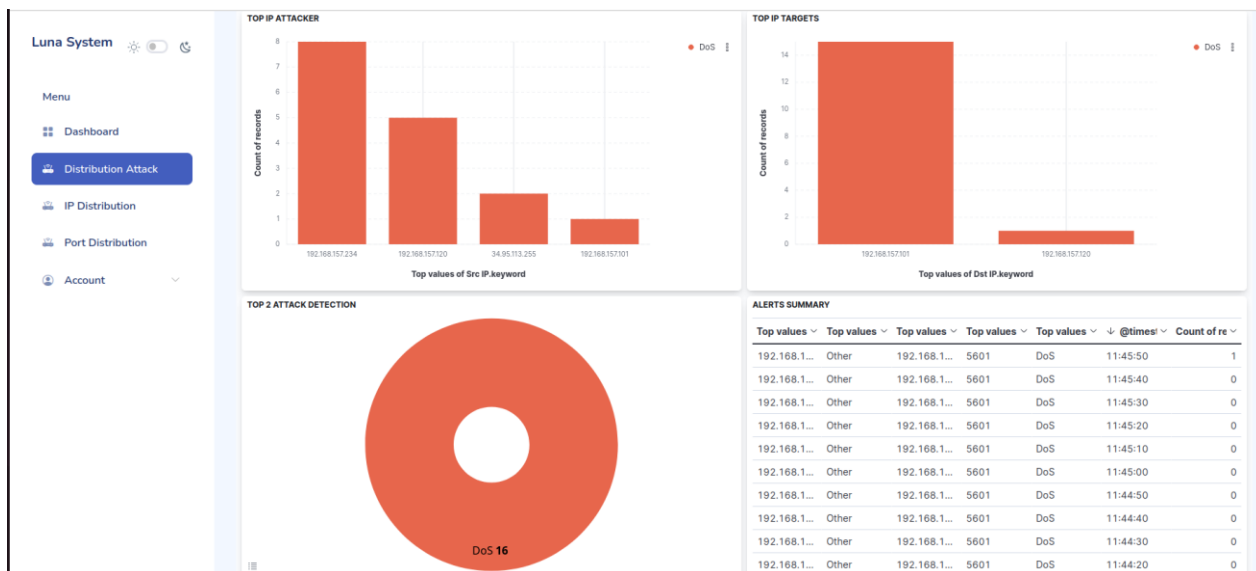
B. Demonstrasi Produk



Gambar 2.9. Tampilan *Dashboard* LunaSystem

Pada gambar 2.9 merupakan tampilan dari *dashboard* setelah sebelumnya *user* melakukan *login* ke dalam sistem. Di dalam menu *Dashboard* terdapat beberapa *chart* dan *table* informasi dari

serangan siber yang terdeteksi. Pada *line chart* terdapat hasil deteksi serangan dalam kurun waktu tiap 10 detik, ini akan membuat perubahan secara *realtime*. Selanjutnya terdapat *table summary* dari hasil serangan siber yang terdeteksi, tabel ini akan berisi *source IP*, *destination IP*, dan label jenis serangan. Kemudian pada *chart Source IP* berfungsi untuk menampilkan IP dari *server* yang mengalami serangan siber.



Gambar 2.10. Tampilan Menu *Distribution Attack*

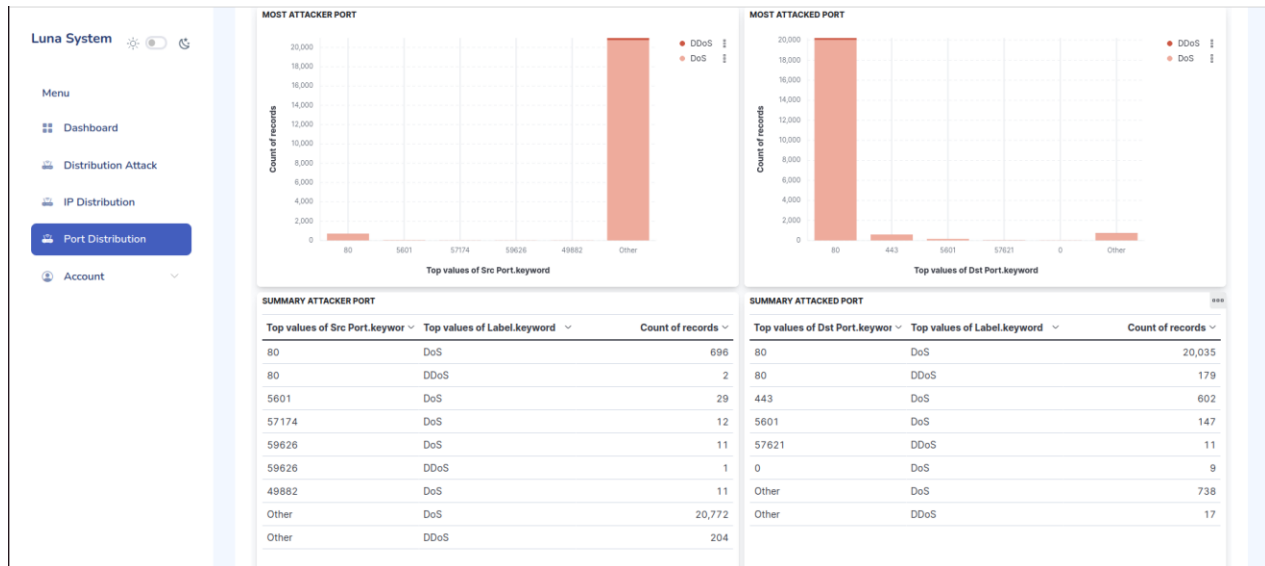
Pada gambar 2.10 tersebut, merupakan tampilan dari menu *distribution attack*, dalam menu ini akan terdapat 3 chart dan 1 table. Pada *chart top IP attacker* akan menampilkan total dan IP dari IP si penyerang. Selanjutnya terdapat *chart top IP targets*, *charts* ini berfungsi untuk menampilkan total dari IP yang menerima serangan dari IP *attacker*. Kemudian terdapat *donut charts*, *donut charts* ini untuk menampilkan total data dari jenis serangan yang telah terdeteksi. Terakhir terdapat *alerts summary*, dimana tabel ini akan menampilkan hasil serangan siber yang terdeteksi, tabel ini akan berisi *source IP*, *destination IP*, dan label jenis serangan.

Gambar 2.11. Tampilan menu *IP Distribution*



Gambar 2.11 diatas merupakan tampilan menu dari *IP Distribution*, kemudian terdapat *chart* untuk menampilkan IP sumber dari mana serangan berasal. Selain itu juga terdapat *chart IP destination*, ini akan menunjukkan IP tujuan yang paling sering diserang. Ini dapat mengidentifikasi *server* atau perangkat mana yang menjadi target utama.

Gambar 2.12. Tampilan menu Port Distribution



Pada gambar 2.12 tersebut, merupakan tampilan dari menu *port distribution*, pada menu ini terdapat 2 chart dan 2 table. *chart* ini untuk menunjukkan *port* sumber yang paling sering digunakan dalam serangan. Selain itu juga terdapat *chart port destination*, ini akan menunjukkan *port* tujuan yang paling sering diserang. Ini dapat mengidentifikasi *server* atau perangkat mana yang menjadi target utama. Selanjutnya terdapat *table* untuk menampilkan kesimpulan dari *port* penyerang dan *port* yang diserang, tabel ini akan mencakup *port*, label serangan dan total serangan yang terjadi.

10.3. Demonstrasi Produk

Sediakan link dan QR code yang merujuk ke video demonstrasi produk.

Contoh:

Link : https://drive.google.com/drive/folders/13rugbQGNDQa_HG53q4kCHoVBwJvy04YO?usp=sharing

10.4. Kesimpulan Realisasi Desain Produk dengan Implementasi Produk

Di dalam sub bab ini, Tim Capstone harus membuat tabel kesimpulan untuk melakukan pelacakan dari list desain yang ada apakah ada proses implementasi yang gagal dari dokumen desain yang ada serta melakukan menjabarkan analisis dari kegagalan yang ada.

Template dan Dokumen C500

Topik Capstone	Topik Capstone	
Siklus / Tahun	Gasal (Nov) atau Genap (Mei) / 2022	
Judul Dokumen	Capstone TA Judul Capstone Proyek kelompok	
Jenis Dokumen	PENGUJIAN PRODUK Catatan: Penggunaan dan penyebaran dokumen ini dikendalikan oleh Departemen Teknik Komputer Universitas Diponegoro	
Nomor Dokumen	C500.[NoRev]TA[tahun].[1/2].[KodeKelompok]	
Nomor Revisi	NoRev	
Nama File	KodeKelompok.doc	
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
Pembimbing 1 (Utama)	Nama	NIP.	Jabatan Anggota
	Tanggal		Tanda Tangan
Pembimbing 2	Nama	NIP.	Jabatan Anggota
	Tanggal		Tanda Tangan

Versi, Tanggal, Oleh	Perbaikan

Daftar Isi

1. Pendahuluan	92
1.1. Ringkasan isi dokumen	92
1.2. Aplikasi Dokumen	92
1.3. Referensi	92
1.4. Daftar Singkatan	92
2. Pengujian	92
2.1. Pengujian model	92
2.2. Pengujian Aplikasi	93

11. Pendahuluan

11.1. Ringkasan isi dokumen

Isi Ringkasan Dokumen

11.2. Aplikasi Dokumen

Dokumen ini berlaku berfungsi untuk menjelaskan:

10) Tahap pengujian produk yang telah dibuat dalam dokumen C300 dan C400

11) Acuan keberhasilan produk sesuai spesifikasi yang telah dijabarkan dalam dokumen C200

12) Menjadi catatan pengerjaan dan proses pengujian produk yang dilakukan

13) Catatan proses pengerjaan dan revisi yang sedang berlangsung

11.3. Referensi

11.4. Daftar Singkatan

12. Pengujian

12.1. Pengujian model

Jelaskan hasil pengujian tentang bagaimana model yang digunakan diuji, rumus pengujian, dan perhitungan presisinya.

Contoh:

Pengujian dilakukan hanya menggunakan dataset yang telah ada. Pengujian algoritma deteksi menggunakan dataset CIC-IDS-2017, CSE-CIC-IDS2018, dan CICDDoS2019. Dataset ini mencakup trafik normal dan serangan Slow DoS dan DRDoS.

Recall atau sensitivitas (juga disebut True Positive Rate) adalah metrik yang mengukur proporsi serangan yang diidentifikasi dengan benar relatif terhadap serangan yang sebenarnya dalam lalu lintas sampel. Hal ini dicapai melalui penerapan persamaan berikut:

$$Recall = TP / TP + FN$$

Precision atau ketepatan adalah kemampuan sistem dalam memastikan bahwa apa yang diidentifikasi sebagai serangan benar-benar adalah serangan. Precision dihitung dengan menggunakan persamaan berikut:

$$Precision = TN / TN + FP$$

F1 Score adalah ombinasi dari precision dan recall, memberikan gambaran keseluruhan tentang keseimbangan antara keduanya. Precision dihitung dengan menggunakan persamaan berikut:

$$F1\ Score = 2 * Precision + Recall / Precision * Recall$$

Akurasi sistem deteksi didefinisikan sebagai persentase deteksi akurat yang dilakukan pada aliran lalu lintas sampel. Metrik ini dihitung dengan menggunakan persamaan berikut:

$$Accuracy = TP + TN / TP + FN + FP + FN$$

Tabel 2.9 Hasil pengujian sistem deteksi DoS dengan Ensemble Stacking menggunakan Test Set

Dataset	Recall	Precision	F1-Score	Akurasi
CIC-IDS2017	99,98804%	99,98801%	99,988013%	99,988041%

CSE-CIC-IDS2018				
CIC-DDOS2019	99,97972948%	99,97972952%	99,97972948%	99,97972946%

Tabel 2.9 menunjukkan hasil dari pengetesan sistem deteksi DoS berdasarkan Test Set Performa ensemble stacking menunjukkan performa yang baik dengan akurasi sebesar 99,9880413%, presisi sebesar 99,9880193%, F1 score sebesar 99,9880139%, recall sebesar 99,9880413%.

Hasil dari pengetesan sistem deteksi reflection-based DDoS berdasarkan Test Set Performa ensemble stacking menunjukkan performa yang baik dengan akurasi sebesar 99,97972948% presisi sebesar 99,97972952%, F1 score sebesar 99,97972948%, recall sebesar 99,97972946%.

12.2. Pengujian Aplikasi

Jelaskan hasil pengujian aplikasi yang meliputi aspek fungsional, keamanan, waktu respon, dan portabilitas (jika memungkinkan) dengan skenario yang telah direncanakan.

Contoh:

Melakukan pengujian aplikasi setelah melakukan implementasi dalam pengembangan perangkat lunak React Js, Html dan Javascript. Pada tahapan ini memiliki fungsi dan tujuan untuk memastikan bahwa aplikasi yang dikembangkan bisa berjalan sesuai dengan fungsinya dan tidak mengalami kegagalan selama sistem berjalan. Sehingga bisa dipastikan bahwa sistem berjalan sesuai dengan kebutuhan pengguna. Metode pengujian aplikasi yang digunakan adalah Black Box Testing.

2.2.1. Pengujian Fungsional Sistem dengan *Black Box Testing*

Pengujian dengan metode Black Box testing adalah suatu pendekatan di mana pengujian dilakukan tanpa memerhatikan struktur internal atau logika implementasi dari suatu sistem atau perangkat lunak [3]. Fokus utama dari Black Box testing adalah pada input yang diberikan dan output yang dihasilkan, tanpa memperhatikan cara sistem mencapai hasil tersebut. Tujuannya adalah untuk mengevaluasi fungsionalitas dan respons sistem tanpa memerlukan pengetahuan detail tentang kode sumbernya. Black Box testing efektif dalam mengidentifikasi bug atau kesalahan yang mungkin terjadi pada tingkat fungsionalitas atau antarmuka pengguna.

A. Pengujian Autentikasi

Pada pengujian fitur autentikasi untuk Administrator menjelaskan tentang hasil pengujian pada halaman Sign In, dan Sign Out pada sistem. Penjelasan terkait hasil pengujian ini terdapat pada Tabel 2.1.

Tabel 2.1. Pengujian Fitur Autentikasi

SRS-id	Kode Pengujian	Nama Pengujian	Bentuk Pengujian	Hasil yang Diharapkan	Hasil Pengujian
(dipetakan dengan SRS-id yang ada di C-300)	A101	<i>Sign in</i> ke sistem.	Memasukkan nama pengguna dan kata sandi akun Administrator.	Sistem menampilkan halaman <i>dashboard</i> sistem.	Berhasil
	A102	Validasi data pada <i>form sign in</i> .	Memasukkan <i>username</i> dan <i>password</i> yang	Sistem membatalkan proses <i>sign in</i> dan	Berhasil

			tidak terdaftar dalam sistem.	memberikan pesan peringatan/ <i>invalid credentials</i> kepada pengguna.	
	A103	<i>Sign out</i> dari sistem.	Melakukan <i>logout</i> dari menu <i>account</i> yang terdapat halaman <i>dashboard</i>	Sistem menampilkan halaman <i>login</i>	Berhasil

Berdasarkan Tabel 2.1 yang memuat pengujian black box pada fitur Autentikasi untuk Administrator didapatkan hasil yang fungsional. Fungsi tersebut sudah berjalan dengan baik sesuai dengan hasil yang diharapkan.

B. Pengujian Fitur

Pada pengujian aplikasi untuk Administrator menjelaskan tentang hasil pengujian setiap halaman pada sistem. Penjelasan terkait hasil pengujian ini terdapat pada Tabel 2.8.

Tabel 2.8 Pengujian aplikasi

SRS-id	Kode Pengujian	Nama Pengujian	Bentuk Pengujian	Hasil yang Diharapkan	Hasil Pengujian
(dipetakan dengan SRS-id yang ada di C-300)	A201	Melihat data grafik pada <i>Dashboard</i>	Administrator dapat melihat data grafik pada halaman <i>Dashboard</i>	Halaman <i>Dashboard</i> menampilkan data grafik sesuai yang diharapkan	Berhasil
	A202	Melihat <i>table summary</i> pada <i>Dashboard</i>	Administrator dapat melihat tabel <i>summary</i> dari serangan siber yang terdeteksi	Halaman <i>Dashboard</i> menampilkan tabel <i>summary</i> sesuai yang diharapkan	Berhasil
	A203	Melihat grafik serangan siber dalam satuan waktu pada <i>Dashboard</i>	Administrator dapat melihat grafik serangan siber dalam satuan waktu	Halaman <i>Dashboard</i> menampilkan data grafik serangan siber dalam satuan waktu sesuai yang diharapkan	Berhasil
	A204	Melihat <i>bar chart</i> pada <i>Dashboard</i>	Administrator dapat melihat <i>bar chart</i> dari IP yang terkena serangan siber	Halaman <i>Dashboard</i> menampilkan data <i>bar chart</i> sesuai yang diharapkan	Berhasil
	A205	Melihat data <i>bar chart Top IP Attacker</i> pada menu <i>Distribution Attack</i>	Administrator dapat melihat <i>bar chart Top IP Attacker</i>	Halaman menu <i>Distribution Attack</i> menampilkan data <i>top ip attackers</i> sesuai dengan yang diharapkan	Berhasil

	A206	Melihat data <i>bar chart Top IP Targets</i> pada menu <i>Distribution Attack</i>	Administrator dapat melihat bar chart <i>Top IP Targets</i>	Halaman menu <i>Distribution 1 Attack</i> menampilkan data <i>top ip targets</i> sesuai dengan yang diharapkan	Berhasil
	A207	Melihat data <i>pie chart Top 2 Attack Detection</i> pada menu <i>Distribution Attack</i>	Administrator dapat melihat pie chart <i>Top 2 Attack Detection</i>	Halaman menu <i>Distribution 1 Attack</i> menampilkan data <i>top 2 attack detection</i> sesuai dengan yang diharapkan	Berhasil
	A208	Melihat tabel <i>Alerts Summary</i> pada menu <i>Distribution Attack</i>	Administrator dapat melihat tabel <i>Alerts Summary</i>	Halaman menu <i>Distribution 1 Attack</i> menampilkan data <i>top 2 attack detection</i> sesuai dengan yang diharapkan	Berhasil
	A209	Melihat data pada <i>chart bar Source IP Distribution</i> menu <i>IP Distribution</i>	Administrator dapat melihat <i>chart bar Source IP Distribution</i>	Halaman menu <i>Distribution Attack</i> menampilkan data <i>source ip distribution</i> sesuai dengan yang diharapkan	Berhasil
	A210	Melihat data <i>chart bar Destination IP Distribution</i> pada menu <i>IP Distribution</i>	Administrator dapat melihat <i>chart bar Destination IP Distribution</i>	Halaman menu <i>destination ip distribution</i> menampilkan data <i>top ip targets</i> sesuai dengan yang diharapkan	Berhasil
	A211	Melihat data pada menu <i>Port Distribution</i>	Administrator dapat melihat <i>bar chart Most Attacker Port</i>	Halaman menu <i>Port Distribution</i> menampilkan data <i>most attacker port.</i> sesuai dengan yang diharapkan	Berhasil
	A212	Melihat data <i>bar chart Most Attacked Port</i> pada menu <i>Port Distribution</i>	Administrator dapat melihat <i>bar chart Most Attacked Port</i>	Halaman menu <i>Port Distribution</i> menampilkan data <i>most attacked port</i> sesuai dengan yang diharapkan	Berhasil
	A213	Melihat data tabel <i>Summary Most Attacker Port</i> pada menu <i>Port</i>	Administrator dapat melihat tabel <i>Summary Most Attacker Port</i>	Halaman menu..... <i>Port Distribution</i> menampilkan data <i>summary most attacker</i>	Berhasil

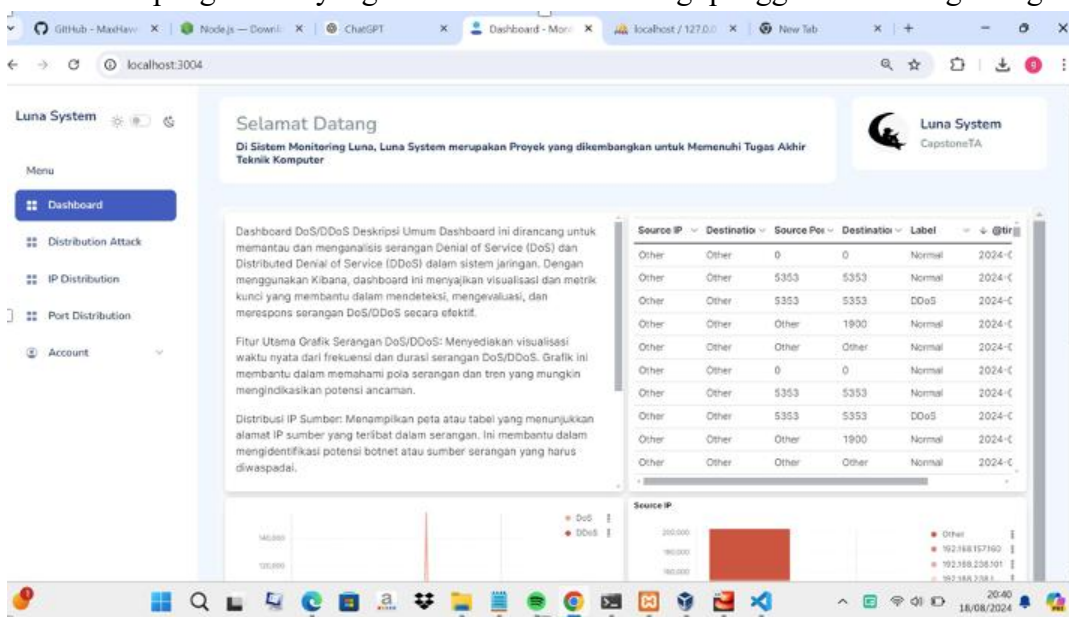
		Distribution		ports Sesuai dengan yang diharapkan	
	A214	Melihat data tabel <i>Summary Most Attacked Port</i> pada menu <i>Port Distribution</i>	Administrator dapat melihat tabel <i>Summary Most Attacked Port</i>	Halaman menu <i>Port Distribution</i> menampilkan data <i>summary most attacked port</i> . Sesuai dengan yang diharapkan	Berhasil

Berdasarkan Tabel 2.8 yang memuat pengujian *black box* pada fitur Dashboard untuk Administrator didapatkan hasil yang fungsional. Fungsi tersebut sudah berjalan dengan baik sesuai dengan hasil yang diharapkan.

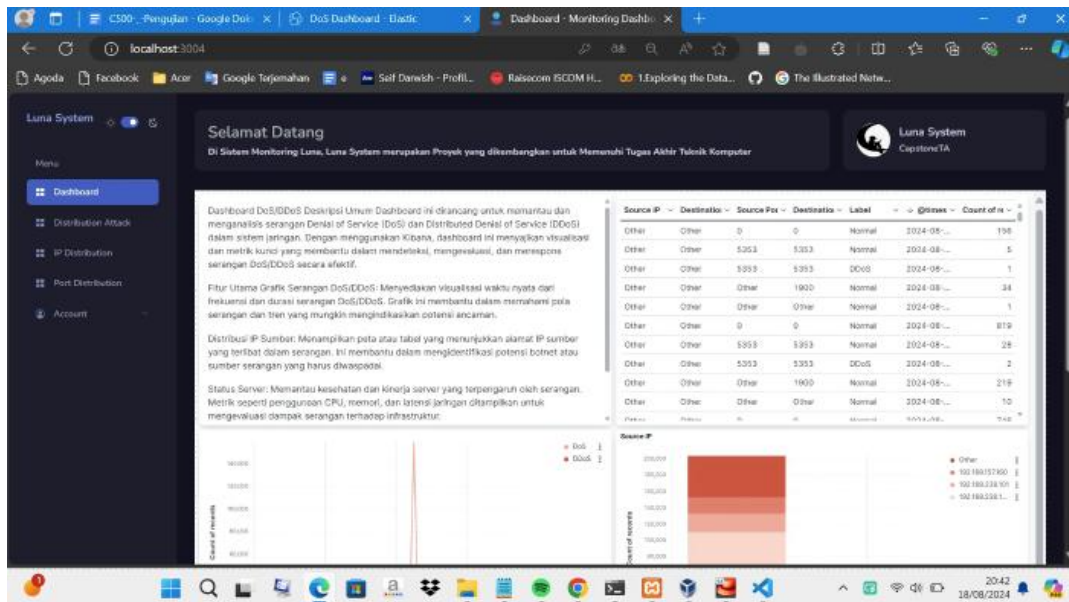
2.2.2. Pengujian Non-Fungsional

A. Pengujian *Portability*

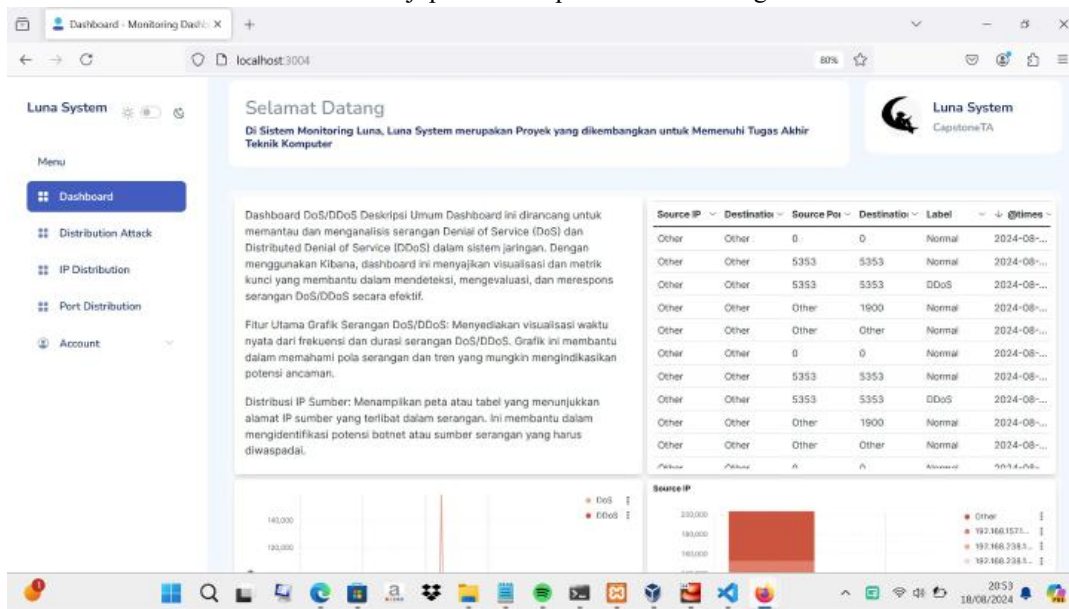
Pada proyek ini, dilakukan suatu pengujian Portabilitas terhadap sistem Sistem Monitoring dan Deteksi Serangan DoS/DRDoS Berbasis Ensemble Learning menggunakan ELK Stack. Tujuannya adalah untuk memastikan bahwa aplikasi ini dapat berfungsi dengan baik di berbagai sistem operasi, khususnya Windows dan Ubuntu, serta diakses melalui web browser yang umum digunakan seperti Google Chrome, Mozilla Firefox, dan Microsoft Edge. Pengujian ini akan mengidentifikasi apakah ada kendala teknis atau perbedaan performa yang signifikan yang dapat mempengaruhi pengalaman pengguna saat mengoperasikan aplikasi di platform yang berbeda. Dengan demikian, pengujian ini tidak hanya bertujuan untuk menilai kompatibilitas teknis tetapi juga untuk memastikan bahwa aplikasi mampu memberikan pengalaman yang konsisten dan aman bagi pengguna di berbagai lingkungan.



Gambar 2.1 Hasil uji portabilitas pada Chrome Windows



Gambar 2.2 Hasil uji portabilitas pada Microsoft Edge Windows

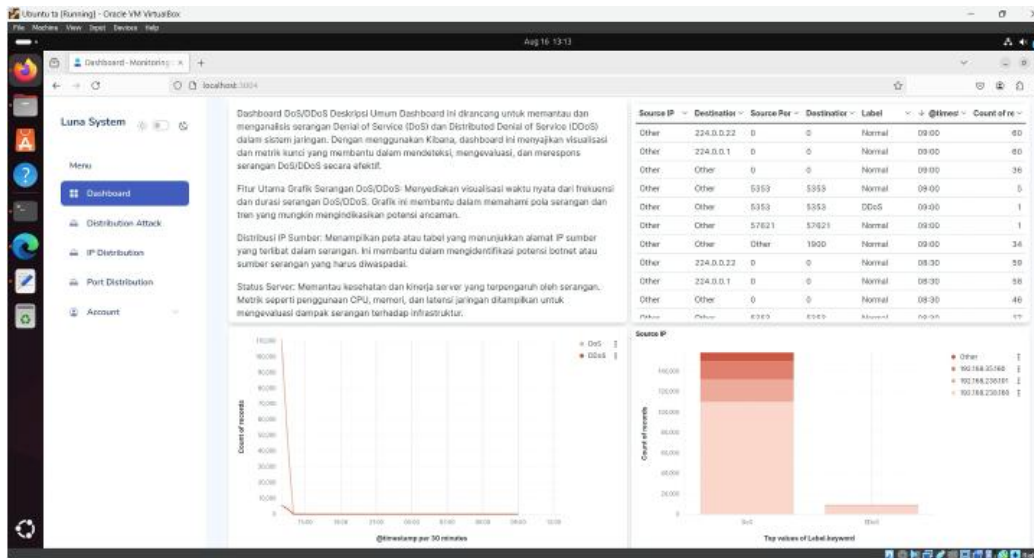


Gambar 2.3 Hasil uji portabilitas pada Mozilla Firefox Windows

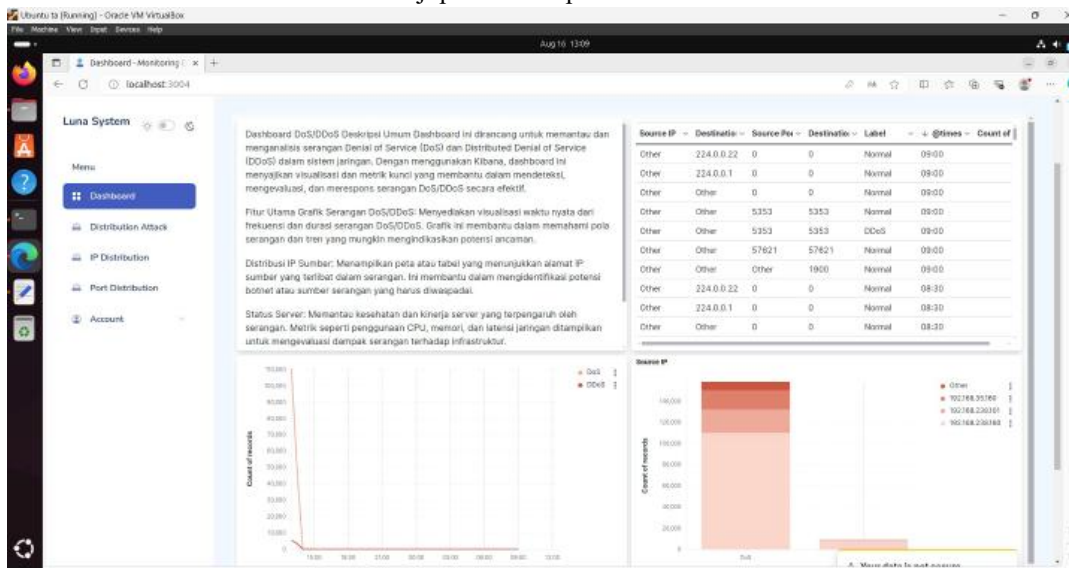
Tabel 2.3 Pengujian Portabilitas pada Windows

Komponen	Windows 10 – Chrome	Windows 10 – Firefox	Windows 10 – Edge	Windows 11 - Edge
Tampilan UI	Konsisten	Konsisten	Konsisten	Konsisten
Fungsi <i>Monitoring</i>	Berfungsi	Berfungsi	Berfungsi	Berfungsi
Grafik Visualisasi	Ditampilkan dengan benar	Ditampilkan dengan benar	Ditampilkan dengan benar	Ditampilkan dengan benar
CORS	Tidak ada masalah	Tidak ada masalah	Tidak ada masalah	Tidak ada masalah

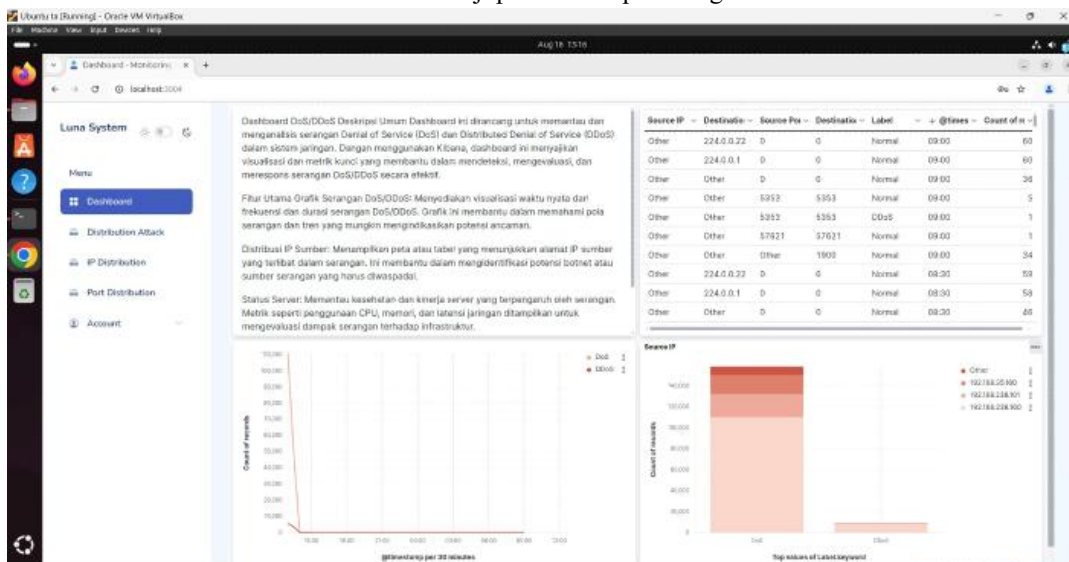
Selanjutnya dilakukan uji portabilitas pada sistem operasi Ubuntu.



Gambar 2.4 Hasil uji portabilitas pada Mozilla Firefox Ubuntu



Gambar 2.5 Hasil uji portabilitas pada Edge Ubuntu



Tabel 2.4 Pengujian portabilitas pada Ubuntu

Komponen	Ubuntu 24.04 – Chrome	Ubuntu 24.04 – Firefox	Ubuntu 24.04 – Edge	Ubuntu 24.04 – Chrome
Tampilan UI	Konsisten	Konsisten	Konsisten	Konsisten
Fungsi <i>Monitoring</i>	Berfungsi	Berfungsi	Berfungsi	Berfungsi
Grafik Visualisasi	Ditampilkan dengan benar	Ditampilkan dengan benar	Ditampilkan dengan benar	Ditampilkan dengan benar
CORS	Tidak ada masalah	Tidak ada masalah	Tidak ada masalah	Tidak ada masalah

Pengujian yang dilakukan pada Sistem ini menunjukkan bahwa aplikasi memiliki tingkat portability yang baik ketika diakses melalui sistem operasi Windows dan Ubuntu, serta di berbagai web browser seperti Google Chrome, Mozilla Firefox, dan Microsoft Edge. Secara umum, tidak ada masalah signifikan yang menghambat fungsionalitas inti dari sistem, dan aplikasi berhasil berfungsi sesuai dengan harapan di semua kombinasi sistem operasi dan browser yang diuji.

B. Pengujian Response Time

Pada pengujian ini, fokus utama adalah untuk mengevaluasi kinerja Sistem dalam merespons berbagai interaksi pengguna. Pengujian dilakukan dengan mengukur waktu muat halaman, dan waktu render grafik pada berbagai kombinasi sistem operasi (Windows dan Ubuntu) dan web browser (Google Chrome, Mozilla Firefox, dan Microsoft Edge). Hasil pengujian ini akan memberikan wawasan tentang seberapa efisien aplikasi dalam menyajikan data dan memberikan pengalaman yang optimal kepada pengguna.

Tabel 2.5 Pengujian *Response Time*

Pengujian	Ubuntu 24.04 – Chrome	Ubuntu 24.04 – Firefox	Windows 10 - Chrome	Windows 10 – Edge	Windows 10 – Firefox
Waktu muat halaman	4 detik	4,5 detik	4 detik	5 detik	4 detik
Waktu <i>render</i> grafik	1 detik	900 milidetik	1,3 detik	1 detik	960 milidetik
Waktu respon Kibana	3 detik	3 detik	3,3 detik	2,5 detik	2,5 detik

Berdasarkan pengujian yang dilakukan, sistem ini menunjukkan kinerja waktu respons yang memuaskan di berbagai platform yang diuji. Secara umum, waktu muat halaman berada dalam kisaran 3 hingga 4 detik, menunjukkan bahwa aplikasi mampu memuat konten dengan cepat di semua kombinasi sistem operasi dan browser.

C. Pengujian Safety

Pengujian ini dirancang untuk mengidentifikasi dan mengatasi potensi kelemahan yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab. Selain itu, pengujian juga mengevaluasi bagaimana aplikasi menangani konfigurasi kritis seperti penggunaan CORS dan protokol HTTP di lingkungan pengembangan, serta bagaimana kesiapan aplikasi ini ketika dipindahkan ke lingkungan

produksi. Langkah-langkah pengujian ini akan memberikan gambaran yang jelas tentang tingkat keamanan yang telah dicapai oleh aplikasi, serta rekomendasi untuk peningkatan lebih lanjut.

Tabel 2.6 Pengujian *safety*

Pengujian	Ubuntu 24.04 – Chrome	Ubuntu 24.04 – Firefox	Windows 10 – Chrome	Windows 10 - Firefox
CORS <i>Configuration</i>	CORS dikonfigurasi dengan benar, akses hanya dari domain yang sah	CORS dikonfigurasi dengan benar, akses hanya dari domain yang sah	CORS dikonfigurasi dengan benar, akses hanya dari domain yang sah	CORS dikonfigurasi dengan benar, akses hanya dari domain yang sah
Proteksi Data (HTTP)	HTTP digunakan pada <i>localhost</i> , risiko rendah di pengembangan	HTTP digunakan pada <i>localhost</i> , risiko rendah di pengembangan	HTTP digunakan pada <i>localhost</i> , risiko rendah di pengembangan	HTTP digunakan pada <i>localhost</i> , risiko rendah di pengembangan
<i>Vulnerability Scanning</i>	Tidak ada kerentanan serius	Tidak ada kerentanan serius	Tidak ada kerentanan serius	Tidak ada kerentanan serius
<i>Incident Response</i>	Deteksi cepat, <i>logging</i> lengkap	Deteksi cepat, <i>logging</i> lengkap	Deteksi cepat, <i>logging</i> lengkap	Deteksi cepat, <i>logging</i> lengkap

Berdasarkan hasil pengujian keamanan, aplikasi ini menunjukkan tingkat kesiapan yang baik dalam menghadapi ancaman keamanan. Meskipun aplikasi saat ini beroperasi di *localhost* dengan HTTP, pengujian menunjukkan bahwa konfigurasi CORS telah diatur dengan baik, membatasi akses hanya pada domain yang sah. Selain itu, hasil pengujian menunjukkan bahwa tidak ada kerentanan serius yang terdeteksi, dan aplikasi mampu menangani insiden keamanan dengan respons yang cepat dan efisien.

D. Pengujian *Security*

Pengujian ini bertujuan untuk mengidentifikasi kerentanan potensial, memastikan konfigurasi keamanan yang tepat, dan menguji respons sistem terhadap berbagai serangan dan ancaman. Dengan hasil pengujian ini, kita dapat menilai efektivitas langkah-langkah keamanan yang diimplementasikan dan memberikan rekomendasi untuk perbaikan lebih lanjut.

Tabel 2.7 Pengujian *security*

Pengujian	Ubuntu 24.04 – Chrome	Ubuntu 24.04 – Firefox	Windows 10 – Chrome	Windows 10 - Firefox
Autentikasi dan Otorisasi	Akses dan autentikasi terkontrol	Akses dan autentikasi terkontrol	Akses dan autentikasi terkontrol	Akses dan autentikasi terkontrol
Konfigurasi keamanan	Konfigurasi aman	Konfigurasi aman	Konfigurasi aman	Konfigurasi aman
<i>Vulnerability Scanning</i>	Tidak ada kerentanan serius	Tidak ada kerentanan serius	Tidak ada kerentanan serius	Tidak ada kerentanan serius

Berdasarkan hasil pengujian keamanan, Sistem ini menunjukkan tingkat keamanan yang sangat baik di berbagai platform. Aplikasi ini bebas dari kerentanan serius, memiliki mekanisme autentikasi yang kuat, dan melindungi data sensitif dengan enkripsi yang memadai. Pengujian penetrasi mengonfirmasi bahwa aplikasi cukup tangguh terhadap serangan yang disimulasikan, dan konfigurasi keamanan telah diterapkan dengan benar untuk melindungi terhadap potensi ancaman.

Lampiran Dokumen

Form Pendaftaran Proyek Desain Capstone

Nama mahasiswa 1 :
NIM :
Nama mahasiswa 2 :
NIM :
Nama mahasiswa 3 :
NIM :

Mengajukan Proyek Desain Capstone dengan judul “(judul proyek capstone yang akan dikerjakan)”

Semarang, (tanggal)

Dosen Pembimbing Capstone 2

Dosen Pembimbing Capstone 1

(Nama)
NIP. (NIP)

(Nama)
NIP. (NIP)

Form Nilai Dosen Pembimbing

Proyek Desain Capstone 1

Nama mahasiswa 1 :
NIM :
Nama mahasiswa 2 :
NIM :
Nama mahasiswa 3 :
NIM :

Parameter yang dinilai	Nilai Dosen Pembimbing I*	Nilai Dosen Pembimbing 2*
(CPMK 1 - Bobot 15) Kemampuan mahasiswa dalam menerapkan pengetahuan matematis atau ilmu alam dalam perancangan dan pengembangan solusi proyek ketika menganalisis dan menyelesaikan masalah kompleks di dunia nyata (C100).		
(CPMK 2 - Bobot 25) Kemampuan mahasiswa dalam melakukan perancangan dan pengembangan komponen, sistem, atau proses berdasarkan aspek ekonomi, manufakturabilitas, sustainabilitas, atau aspek lainnya seperti aspek lingkungan dan legal (C100, C200, C300).		
(CPMK 3 - Bobot 25) Kemampuan mahasiswa dalam melakukan riset dalam proses perancangan desain (C100, C200, C300).		
(CPMK 4 - Bobot 15) Kemampuan mahasiswa dalam melakukan identifikasi, perumusan, dan analisis permasalahan kompleks. (C100).		
(CPMK 5 - Bobot 20) Kemampuan mahasiswa dalam memahami dan mengikuti perkembangan teknologi (C100, C200, C300).		

Semarang, (tanggal)
Koordinator Proyek Desain Capstone

(Nama)
NIP. (NIP)

Note:

Form ini dicetak dan digunakan untuk 1 tim capstone

*Berikan nilai antara 0-100 untuk masing-masing parameter yang ada

Form Nilai Dosen Penguji

Proyek Desain Capstone 1

Nama mahasiswa 1 :
NIM :
Nama mahasiswa 2 :
NIM :
Nama mahasiswa 3 :
NIM :

Parameter yang dinilai	Nilai Dosen Penguji I*	Nilai Dosen Penguji 2*
(CPMK 1 - Bobot 15) Kemampuan mahasiswa dalam menerapkan pengetahuan matematis atau ilmu alam dalam perancangan dan pengembangan solusi proyek ketika menganalisis dan menyelesaikan masalah kompleks di dunia nyata (C100).		
(CPMK 2 - Bobot 25) Kemampuan mahasiswa dalam melakukan perancangan dan pengembangan komponen, sistem, atau proses berdasarkan aspek ekonomi, manufakturabilitas, sustainabilitas, atau aspek lainnya seperti aspek lingkungan dan legal (C100, C200, C300).		
(CPMK 3 - Bobot 25) Kemampuan mahasiswa dalam melakukan riset dalam proses perancangan desain (C100, C200, C300).		
(CPMK 4 - Bobot 15) Kemampuan mahasiswa dalam melakukan identifikasi, perumusan, dan analisis permasalahan kompleks. (C100).		
(CPMK 5 - Bobot 20) Kemampuan mahasiswa dalam memahami dan mengikuti perkembangan teknologi (C100, C200, C300).		

Semarang, (tanggal)
Koordinator Proyek Desain Capstone

(Nama)
NIP. (NIP)

Note:

Form ini dicetak dan digunakan untuk 1 tim capstone

*Berikan nilai antara 0-100 untuk masing-masing parameter yang ada

Form Nilai Dosen Pembimbing

Proyek Desain Capstone 2

Nama mahasiswa 1 :
NIM :
Nama mahasiswa 2 :
NIM :
Nama mahasiswa 3 :
NIM :

Parameter yang dinilai	Nilai Dosen Pembimbing I*	Nilai Dosen Pembimbing 2*
(CPMK 1 - Bobot 40) Kemampuan mahasiswa dalam melakukan riset untuk mengumpulkan, menganalisis, dan menginterpretasi data guna mendukung penilaian teknis dan ilmiah dalam proses perancangan desain (C400, C500).		
(CPMK 2 - Bobot 40) Kemampuan mahasiswa dalam melakukan identifikasi, perumusan, dan analisis permasalahan kompleks (C400, C500).		
(CPMK 3 – Bobot 20) Kemampuan mahasiswa dalam menerapkan metode dan desain perancangan sistem (C400)		

Semarang, (tanggal)
Koordinator Proyek Desain Capstone

(Nama)
NIP. (NIP)

Note:

Form ini dicetak dan digunakan untuk 1 tim capstone

*Berikan nilai antara 0-100 untuk masing-masing parameter yang ada

Form Kesesuaian Milestone Proyek Desain Capstone 2

Nama mahasiswa 1 :
NIM :
Nama mahasiswa 2 :
NIM :
Nama mahasiswa 3 :
NIM :

Parameter yang dinilai	Nilai Dosen Pembimbing I*	Nilai Dosen Pembimbing 2*
(CPMK 5 - Bobot 30) Kemampuan mahasiswa dalam melakukan perencanaan dan pengelolaan proyek desain capstone sesuai dengan waktu yang sudah ditentukan.		
(CPMK 5 - Bobot 30) Kemampuan mahasiswa dalam melakukan perencanaan dan pengelolaan proyek desain capstone sesuai dengan dana yang diusulkan.		
(CPMK 5 - Bobot 40) Kemampuan mahasiswa dalam melakukan perencanaan dan pengelolaan proyek desain capstone sesuai dengan kebutuhan identifikasi desain.		

Semarang, (tanggal)
Koordinator Proyek Desain Capstone

(Nama)
NIP. (NIP)

Note:

Form ini dicetak dan digunakan untuk 1 tim capstone

*Berikan nilai antara 0-100 untuk masing-masing parameter yang ada

Form Penilaian Expo Proyek Desain Capstone 2

Nama Penilai :
Afiliasi :

Berikan nilai kepada peserta expo berdasarkan parameter yang ada, nilai maksimal tertera pada bobot yang ada pada tabel.

Parameter yang dinilai (CPMK 4)	Nilai*
(Bobot 20) Brosur dan standing banner informatif menarik	
(Bobot 20) Video presentasi yang ditampilkan informatif dan menarik	
(Bobot 20) Presentasi produk jelas	
(Bobot 20) Penjelasan yang diberikan jelas	
(Bobot 20) Jawaban pertanyaan jelas	
Total Nilai	

Semarang, (tanggal)

(Nama Pemberi Nilai)

Note:

Form ini dicetak 5x dan digunakan untuk 1 tim capstone

*Berikan nilai antara 0-100 untuk masing-masing parameter yang ada

Form Peer Review Proyek Desain Capstone 2

Judul Proyek Desain Capstone :
 Nama :
 NIM :

(CPMK 6) Berilah nilai sesuai dengan kontribusi teman dalam kelompok Proyek Desain Capstone. Tulis angka 1 – 4 sesuai dengan keterangan setiap kriteria.

No	Kriteria yang dinilai	Skor				Nama Anggota	
		1	2	3	4		
1	Keterlibatan dalam perancangan komponen	Kurang	Cukup	Baik	Sangat baik		
2	Keterlibatan dalam implementasi komponen	Kurang	Cukup	Baik	Sangat baik		
3	Keterlibatan dalam penyusunan laporan	Kurang	Cukup	Baik	Sangat baik		
4	Keterlibatan dalam pembuatan slide	Kurang	Cukup	Baik	Sangat baik		
5	Komunikasi	Sulit dihubungi	Cukup	Baik	Sangat baik & konsisten		
6	Kerjasama	Kurang	Cukup	Baik	Sangat baik		
7	Penyelesaian masalah	Tidak efektif	Cukup	Baik	Sangat baik		
8	Penyelesaian tugas	Tidak selesai	Jarang tepat waktu	Sering tepat waktu	Selalu tepat waktu		
9	Tanggung jawab	Tidak bertanggung jawab	Agak bertanggung jawab	Bertanggung jawab	Sangat bertanggung jawab		
10	Kontribusi keseluruhan	Sedikit	Cukup	Baik	Sangat baik		

*Penilaian ini rahasia, tuliskan sesuai dengan pengamatan anda dalam kelompok.