

# Software Design Document

## "Voices Unheard"



### Kelompok Kapal Karam

Nama Kelompok	NIM	Role
SYAHRIL ARFIAN ALMAZRIL	103032300013	System Analyst
ARDI SETYAPERMANA	103022400108	UI/UX Designer
NAZARIO JOSE VALENTE DA CRUZ	103012350552	Front End
RAKA VALRIZQY AKHDANSYAH	103022330096	Back End
WILDAN BUGI RIVANSYAH	103012400272	Quality Assurance

**Project Manager:** Raygama

LABORATORIUM ADVANCED SOFTWARE ENGINEERING  
FAKULTAS INFORMATIKA  
UNIVERSITAS TELKOM  
2025

## VERSION HISTORY

Versi #	Ditulis Oleh	Tanggal Revisi	Disetujui Oleh	Tanggal Disetujui	Keterangan
V1.0	System Analyst	-	Project Manager	-	Pembuatan dokumen awal berdasarkan analisis kebutuhan.

# Daftar Isi

VERSION HISTORY .....	i
Daftar Isi .....	ii
1. Pendahuluan.....	1
1.1 Tujuan Penulisan Dokumen.....	1
1.2 Lingkup Masalah .....	1
1.3 Definisi dan Istilah .....	1
1.4 Referensi .....	2
2. Perancangan Global .....	3
2.1 Rencana Lingkungan Implementasi.....	3
2.2 Deskripsi Arsitektur Perangkat Lunak .....	4
2.3 Perancangan Rinci .....	4
2.4 Realisasi Use Case .....	5
2.4.1 Use Case #1 <Moderator : Login> .....	5
2.4.1.1 Use Case Scenario #1 <Moderator : Login> .....	5
2.4.1.2 UI Design dan Deskripsi Objek UI #1 <Moderator : Login> .....	7
2.4.1.3 Identifikasi Objek dan Tipe Kelas #1 <Moderator : Login> .....	7
2.4.1.4 Sequence Diagram #1 <Moderator : Login> .....	8
2.4.2 Use Case #2 <Anonymous User : Submit Story> .....	8
2.4.2.1 Use Case Scenario #2 <Anonymous User : Submit Story>.....	8
2.4.2.2 UI Design dan Deskripsi Objek UI #2 <Anonymous User : Submit Story> .....	10
2.4.2.3 Identifikasi Objek dan Tipe Kelas #2 <Anonymous User : Submit Story>.....	12
2.4.2.4 Sequence Diagram #2 <Anonymous User : Submit Story> .....	12
2.4.3 Use Case #3 <Moderator : Review Story> .....	13
2.4.3.1 Use Case Scenario #3 <Moderator : Review Story> .....	13
2.4.3.2 UI Design dan Deskripsi Objek UI #3 <Moderator : Review Story> .....	15
2.4.3.3 Identifikasi Objek dan Tipe Kelas #3 <Moderator : Review Story> .....	16
2.4.3.4 Sequence Diagram #3 <Moderator : Review Story> .....	16
2.4.4 Use Case #4 <Anonymous User : View Stories> .....	17
2.4.4.1 Use Case Scenario #4 <Anonymous User : View Stories> .....	17
2.4.4.2 UI Design dan Deskripsi Objek UI #4 <Anonymous User : View Stories> .....	18
2.4.4.3 Identifikasi Objek dan Tipe Kelas #4 <Anonymous User : View Stories> .....	20
2.4.4.4 Sequence Diagram #4 <Anonymous User : View Stories> .....	20
2.4.5 Use Case #5 <Anonymous User : Delete Story> .....	21

2.4.5.1 Use Case Scenario #5 <Anonymous User : Delete Story> .....	21
2.4.5.2 UI Design dan Deskripsi Objek UI #5 <Anonymous User : Delete Story> .....	22
2.4.5.3 Identifikasi Objek dan Tipe Kelas #5 <Anonymous User : Delete Story> .....	23
2.4.5.4 Sequence Diagram #5 <Anonymous User : Delete Story>.....	23
2.4.6 Use Case #6 <Anonymous User : Quick Exit> .....	24
2.4.5.1 Use Case Scenario #5 <Anonymous User : Quick Exit> .....	24
2.4.6.2 UI Design dan Deskripsi Objek UI #6 <Anonymous User : Quick Exit> .....	25
2.4.6.3 Identifikasi Objek dan Tipe Kelas #5 <Anonymous User : Quick Exit> .....	25
2.4.6.4 Sequence Diagram #5 <Anonymous User : Quick Exit > .....	26
2.6 Perancangan Detail Kelas .....	27
2.7 Perancangan Algoritma dan/atau Query .....	29
Algoritma #1 .....	29
Algoritma #2 .....	30
Algoritma #3 .....	31
Algoritma #4 .....	32
Algoritma #5 .....	33
Query #1 .....	34
Query #2 .....	34
Query #3 .....	34
Query #4 .....	35
Query #5 .....	35
3. Matriks Keruntutan (Requirement Traceability Matrix).....	35
4. Design Pattern dengan Observer Pattern.....	36
Alasan Pemilihan:.....	36
Implementasi Observer Pattern: .....	36
Deskripsi Implementasi: .....	37
Alasan Implementasi: .....	37
Contoh Penggunaan:.....	37
4.1 Jenis Pengujian .....	38
4.2 Prosedur Pengujian .....	39

# 1. Pendahuluan

## 1.1 Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan secara rinci bagaimana perancangan perangkat lunak platform digital "Voices Unheard" yang akan dikembangkan. Dokumen DPPL ini akan digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap perancangan perangkat lunak. Selain itu menjelaskan fitur-fitur aplikasi seperti pengiriman cerita anonim, moderasi konten berbasis AI, sistem keamanan berlapis, dan mekanisme kontrol pengguna. Tujuan utamanya adalah untuk memastikan pemahaman bersama tentang lingkup platform, serta memberikan panduan untuk pengembangan, pengujian, dan pemeliharaan aplikasi. Dokumen ini juga akan digunakan sebagai alat evaluasi kualitas dalam pengembangan proyek, untuk memastikan keberhasilan pelaksanaan proyek, dan sebagai dasar untuk perkembangan dan pemeliharaan masa depan platform Voices Unheard.

## 1.2 Lingkup Masalah

Platform ini dirancang sebagai solusi digital yang aman untuk individu yang terdampak konflik—termasuk korban perang, pengungsi, dan korban diskriminasi—untuk berbagi cerita mereka secara anonim. Platform mengadopsi model "Anonimitas Terkelola" dengan menggabungkan keamanan teknis absolut dan moderasi profesional. Sistem mengintegrasikan kecerdasan buatan (AI) sebagai garda terdepan dalam proses penyaringan konten yang bertugas melakukan pra-moderasi dengan memindai setiap kiriman secara otomatis untuk mendeteksi risiko keamanan sebelum diteruskan ke moderator manusia. Platform dirancang untuk mencegah "Digital Refoulement" dengan memastikan tidak ada pengumpulan data pengguna yang dapat diidentifikasi, sambil tetap memberikan validasi, rasa kebersamaan, dan kenyamanan kepada pengguna dalam lingkungan yang sepenuhnya aman dan terpercaya.

## 1.3 Definisi dan Istilah

Definisi dan istilah berikut digunakan secara konsisten dalam dokumen ini untuk memastikan pemahaman yang seragam.

- **Anonymous User** : Pengguna utama platform yang berinteraksi dengan fitur untuk mengirimkan cerita tanpa identitas yang dapat dilacak
- **Moderator** : Pengguna yang memiliki akses untuk meninjau kiriman cerita dan bertanggung jawab untuk melakukan moderasi konten
- **AI System** : Sistem eksternal yang melakukan pra-moderasi pada semua kiriman cerita sebelum ditinjau oleh moderator manusia
- **Anonimitas Terkelola** : Model hibrida yang menggabungkan keamanan teknis absolut dengan interaksi yang sangat terkontrol dan dimoderasi secara profesional
- **Digital Refoulement** : Konsep yang menggambarkan kegagalan platform digital dalam

melindungi data pengguna

- **EXIF** : Exchangeable Image File Format, metadata yang melekat pada file gambar
- **PII** : Personally Identifiable Information, informasi yang dapat diidentifikasi secara pribadi
- **PTSD** : Post-Traumatic Stress Disorder, gangguan stres pascatrauma
- **OWASP** : Open Web Application Security Project, komunitas online yang berfokus pada keamanan aplikasi web
- **TLS** : Transport Layer Security, protokol kriptografi untuk komunikasi yang aman

## 1.4 Referensi

Dokumen perancangan ini mengacu pada sumber-sumber berikut:

1. Kelompok Kapal Karam. 2025. *Software Requirements Specification Document "Voices Unheard"*. Fakultas Informatika, Universitas Telkom.
2. OWASP Foundation. *OWASP Top Ten*. Diakses pada 15 Juli 2025.
3. UNHCR. *Basic principles of the protection of refugees and the UNHCR's role*. Diakses pada 15 Juli 2025.
4. Kelompok G. *DPPL RPL Group G - Sunkatsu App*. Fakultas Informatika, Universitas Telkom.

## **2. Perancangan Global**

### **2.1 Rencana Lingkungan Implementasi**

Platform Voices Unheard, sebagai solusi digital yang aman untuk berbagi cerita anonim, didesain dengan memperhitungkan arsitektur perangkat lunak yang kokoh dan berfokus pada keamanan. Dari sisi pengembangan, platform ini akan mengadopsi arsitektur berbasis mikroservis dengan fokus pada keamanan berlapis, di mana Back-End akan menjadi pusat logika bisnis, sistem moderasi AI, dan interaksi dengan Database. Back-End akan dikembangkan menggunakan teknologi seperti Node.js dengan Express.js atau Python dengan Django/Flask, memastikan keandalan dalam pemrosesan cerita dan manajemen moderasi.

Platform ini akan menggunakan Database PostgreSQL dengan enkripsi at-rest untuk menyimpan data cerita yang telah dibersihkan dari metadata. Sistem keamanan yang ketat akan diimplementasikan di tingkat Back-End untuk melindungi anonimitas pengguna, termasuk pembersihan otomatis metadata EXIF, tidak mencatat alamat IP pengguna, dan enkripsi end-to-end untuk semua komunikasi.

Ketika berfokus pada platform web, Front-End akan dikembangkan menggunakan teknologi seperti React.js atau Vue.js dengan implementasi desain berbasis trauma untuk memastikan antarmuka yang aman dan tidak memicu respons traumatis. Pengguna akan berinteraksi dengan platform melalui antarmuka yang sederhana, empatik, dan dilengkapi dengan fitur "Keluar Cepat" di setiap halaman.

Selain itu, sistem moderasi AI akan diintegrasikan menggunakan layanan seperti Google Cloud Natural Language API atau AWS Comprehend untuk melakukan pra-moderasi otomatis sebelum konten diteruskan ke moderator manusia. Dengan menyatukan semua komponen ini, Voices Unheard diharapkan dapat memberikan pengalaman berbagi cerita yang aman, anonim, dan tervalidasi bagi pengguna.

## 2.2 Deskripsi Arsitektur Perangkat Lunak

No	Nama Komponen	Keterangan
1.	Security Layer	Komponen keamanan berlapis yang menangani enkripsi, pembersihan metadata, dan perlindungan identitas pengguna
2.	Authentication System	Sistem autentikasi khusus untuk moderator dengan enkripsi yang kuat
3.	AI Moderation Engine	Sistem AI yang melakukan pra-moderasi otomatis pada semua kiriman cerita
4.	Story Management System	Komponen inti yang mengelola pengiriman, penyimpanan, dan publikasi cerita
5.	Database Management System	Sistem basis data PostgreSQL dengan enkripsi at-rest untuk menyimpan cerita yang telah dibersihkan
6.	Metadata Cleaner	Modul khusus untuk menghapus semua metadata EXIF dan GPS dari file yang diunggah
7.	Communication API	API yang diamankan dengan HTTPS/TLS untuk komunikasi antara Front-End dan Back-End
8.	Emergency Escalation System	Sistem otomatis untuk menangani kasus darurat seperti ancaman bunuh diri
9.	Trauma-Informed UI Framework	Framework antarmuka pengguna yang dirancang khusus untuk pengguna yang mengalami trauma

## 2.3 Perancangan Rinci



## 2.4 Realisasi Use Case

Berisi TABEL USE CASE sebagai berikut :

No	Nama Use Case	Deskripsi Use Case
#1.	Moderator : Login	Moderator dapat masuk ke sistem untuk mengakses dashboard moderasi
#2.	Anonymous User : Submit Story	Pengguna anonim dapat mengirimkan cerita tanpa membuat akun
#3.	Moderator : Review Story	Moderator dapat meninjau dan memoderasi cerita yang telah dipra-moderasi oleh AI
#4.	Anonymous User : View Stories	Pengguna anonim dapat melihat cerita yang telah dipublikasikan
#5.	Anonymous User : Delete Story	Pengguna dapat menghapus cerita mereka menggunakan token penghapusan
#6.	Anonymous User : Quick Exit	Pengguna dapat mengakses button Quick Exit pada halaman utama

### 2.4.1 Use Case #1 <Moderator : Login>

#### 2.4.1.1 Use Case Scenario #1 <Moderator : Login>

Use case name	Moderator : Login
Description	Untuk memeriksa autentikasi moderator dan memberikan akses ke dashboard moderasi
Pre-condition	Moderator belum masuk ke sistem Voices Unheard
Post-condition	Moderator berhasil masuk dan dapat mengakses dashboard moderasi

### ***Main Scenario***

User	Sistem
1. Mengakses halaman login moderator	2. Menampilkan antarmuka login dengan enkripsi TLS
	3. Meminta moderator mengisi username dan password
4. Mengisi username dan password dengan benar	
	5. Melakukan autentikasi dengan enkripsi yang kuat
	6. Menampilkan sukses login dan redirect ke dashboard
7. Masuk ke dashboard moderasi	

### ***Alternative flow***

User	Sistem
1. Mengakses halaman login moderator	2. Menampilkan antarmuka login dengan enkripsi TLS
	3. Meminta moderator mengisi username dan password
4. Moderator salah mengisi username atau password	
	5. Melakukan autentikasi dan mendeteksi kesalahan
	6. Menampilkan pesan error dan log percobaan login
	7. Kembali ke Step 2 dengan peringatan keamanan

#### 2.4.1.2 UI Design dan Deskripsi Objek UI #1 <Moderator : Login>

##### *A001 - Login Page*

Tabel Deskripsi Objek UI

ID LAYAR	NAMA LAYAR	DESKRIPSI
A001	Moderator Login	Antarmuka login yang aman untuk moderator dengan enkripsi end-to-end

##### *A001 : Moderator Login Page*

Id_Objek	Jenis	Label	Keterangan
Image	Image	Voices Unheard Logo	Menampilkan logo platform
Text 1	Text	Moderator Access	Menampilkan judul halaman
Text 2	Text	Secure Login Portal	Menampilkan subjudul
RTF 1	RTF	Username	Input username moderator yang terenkripsi
RTF 2	RTF	Password	Input password dengan hash security
Button 1	Button	Secure Login	Tombol login dengan validasi keamanan berlapis
Button 2	Button	Emergency Exit	Tombol keluar cepat untuk keamanan

#### 2.4.1.3 Identifikasi Objek dan Tipe Kelas #1 <Moderator : Login>

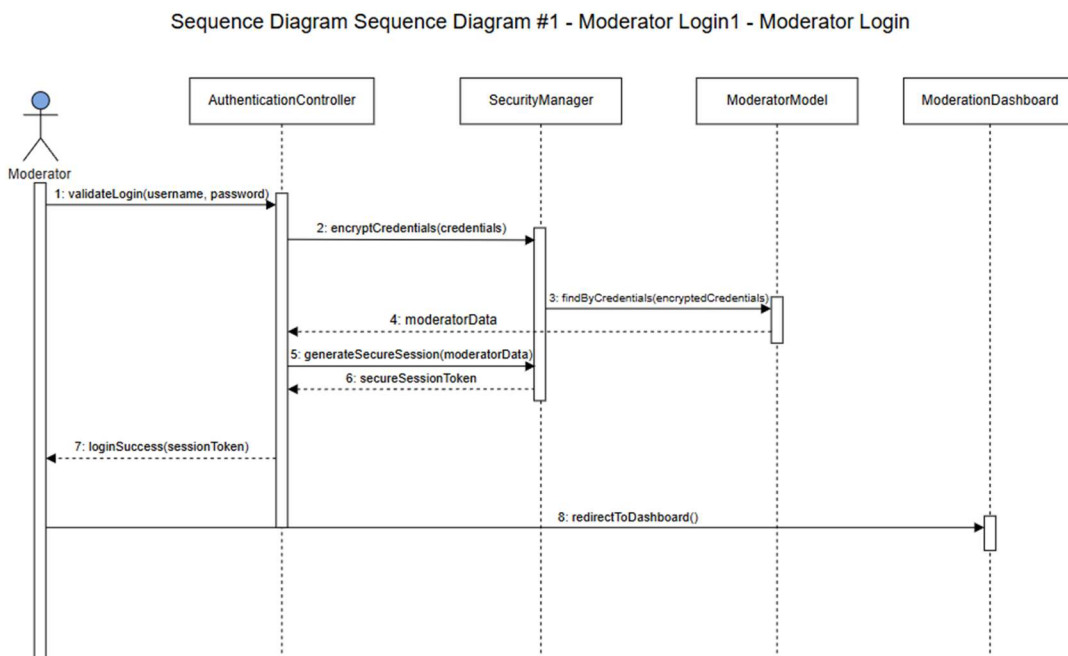
Tabel Objek Perancangan

No	Nama Objek Baru	Jenis / Tipe Kelas
1.	ModeratorLoginView	UI
2.	AuthenticationController	Controller
3.	ModeratorModel	Entity

4.	SecurityManager	Controller
----	-----------------	------------

#### 2.4.1.4 Sequence Diagram #1 <Moderator : Login>

ModeratorLoginView -> AuthenticationController: validateLogin(username, password)  
 AuthenticationController -> SecurityManager: encryptCredentials(credentials)  
 SecurityManager -> ModeratorModel: findByCredentials(encryptedCredentials)  
 ModeratorModel -> AuthenticationController: moderatorData  
 AuthenticationController -> SecurityManager: generateSecureSession(moderatorData)  
 SecurityManager -> AuthenticationController: secureSessionToken  
 AuthenticationController -> ModeratorLoginView: loginSuccess(sessionToken)  
 ModeratorLoginView -> ModerationDashboard: redirectToDashboard()



#### 2.4.2 Use Case #2 <Anonymous User : Submit Story>

##### 2.4.2.1 Use Case Scenario #2 <Anonymous User : Submit Story>

Use case name	Anonymous User : Submit Story
Description	Pengguna anonim dapat mengirimkan cerita tanpa membuat akun atau memberikan informasi identitas

Pre-condition	Pengguna mengakses platform tanpa login
Post-condition	Cerita berhasil dikirim dan masuk ke sistem pra-moderasi AI

### *Main Scenario*

User	Sistem
1. Mengakses halaman submit story	2. Menampilkan form pengiriman cerita yang aman
	3. Tidak meminta informasi identitas apapun
4. Menulis cerita di text area yang disediakan	
5. Mengunggah file gambar (opsional)	
	6. Secara otomatis menghapus metadata EXIF dari file
	7. Tidak mencatat alamat IP pengguna
8. Menekan tombol "Share Anonymously"	
	9. Mengirim cerita ke AI moderation system
	10. Menghasilkan token penghapusan untuk pengguna
	11. Menampilkan konfirmasi pengiriman

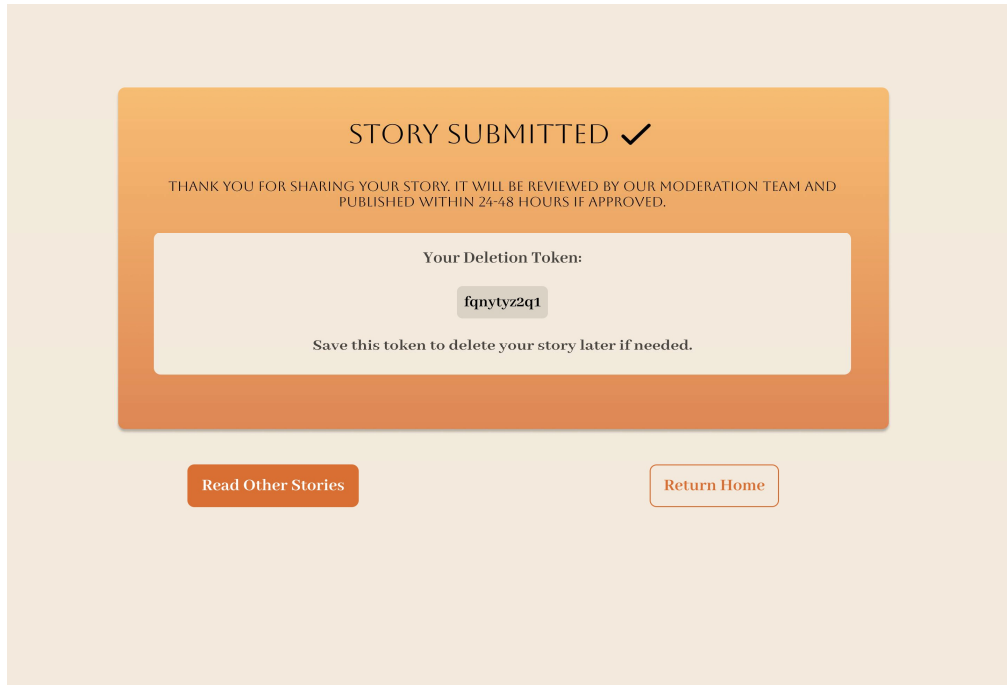
### *Alternative flow*

User	Sistem
1. Mengakses halaman submit story	2. Menampilkan form pengiriman cerita
4. Menulis cerita dengan konten yang terdeteksi berbahaya	
8. Menekan tombol "Share Anonymously"	

	9. AI system mendeteksi konten berbahaya
	10. Menampilkan pesan bahwa cerita perlu review lebih lanjut
	11. Cerita masuk ke queue moderasi darurat

#### 2.4.2.2 UI Design dan Deskripsi Objek UI #2 <Anonymous User : Submit Story>

The image shows a mobile app interface for submitting an anonymous story. The background is a light beige color. At the top left, there is a back arrow icon and the text "Back to home". At the top right, there is a "Quick Exit" button. The main heading is "SHARE YOUR STORY" in a large, bold, sans-serif font. Below the heading, there is a subheading: "YOUR VOICE MATTERS. SHARE YOUR EXPERIENCE ANONYMOUSLY IN A SAFE, SUPPORTIVE SPACE." Below this, there is a light orange box containing a shield icon and the text: "Complete Privacy: No account required, no tracking, no personal data collected. Your story will be anonymous and any metadata will be automatically removed." Below this box is the main form area, which has an orange header bar with the text "Your anonymous story" and an "Image" button with a plus icon. The form area contains a large text input field with the placeholder text: "share your experience, thoughts or feelings. Take your time and write as much or as little as feels right for you." Below the text input field, there is a character count "0/5000 characters" and a note "Stories are reviewed before publication". At the bottom left of the form area, there is a small icon and the text "Please write at least 10 characters to submit your story". At the bottom right of the form area, there is a "Review & Submit" button.



## B001 - Submit Story Page

### *Tabel Deskripsi Objek UI*

ID LAYAR	Nama Layar	Deskripsi
B001	Submit Story	Antarmuka pengiriman cerita yang trauma-informed dan aman

## B001 : Submit Story Page

Id_Objek	Jenis	Label	Keterangan
Text 1	Text	Share Your Voice	Judul halaman yang empatik
Text 2	Text	Your story matters. You are safe here.	Pesan validasi dan keamanan
TextArea	TextArea	Write your story...	Area teks untuk menulis cerita
FileUpload	FileUpload	Add Image (Optional)	Upload gambar dgn pembersihan metadata otomatis
Checkbox 1	Checkbox	I understand this will be reviewed	Persetujuan untuk proses moderasi

Button 1	Button	Share Anonymously	Tombol pengiriman yang aman
Button 2	Button	Quick Exit	Tombol keluar cepat ke situs netral
Text 3	Text	Content Warning Guidelines	Link ke panduan konten

#### 2.4.2.3 Identifikasi Objek dan Tipe Kelas #2 <Anonymous User : Submit Story>

**Tabel Objek Perancangan**

No	Nama Objek Baru	Jenis / Tipe Kelas
1.	StorySubmissionView	UI
2.	StoryController	Controller
3.	StoryModel	Entity
4.	MetadataCleanerService	Service
5.	AIModerationService	Service

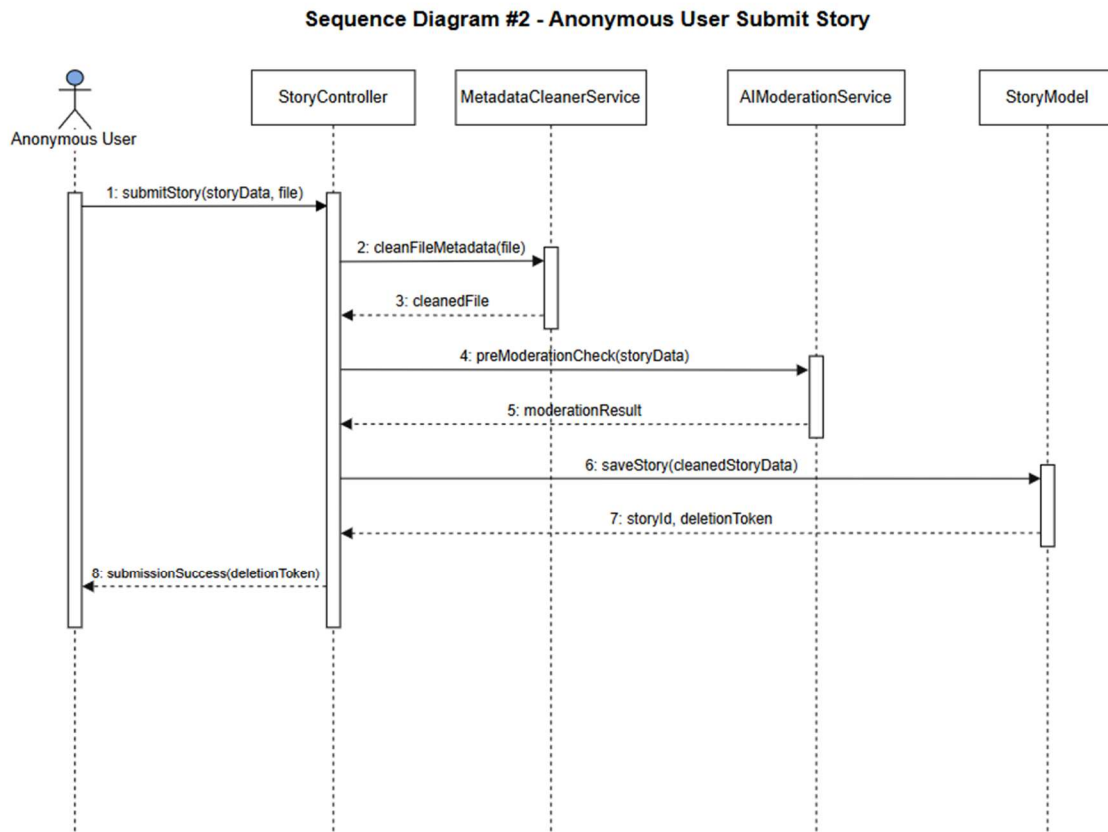
#### 2.4.2.4 Sequence Diagram #2 <Anonymous User : Submit Story>

```

StorySubmissionView -> StoryController: submitStory(storyData, file)
StoryController -> MetadataCleanerService: cleanFileMetadata(file)
MetadataCleanerService -> StoryController: cleanedFile
StoryController -> AIModerationService: preModerationCheck(storyData)
AIModerationService -> StoryController: moderationResult
StoryController -> StoryModel: saveStory(cleanedStoryData)
StoryModel -> StoryController: storyId, deletionToken
StoryController -> StorySubmissionView: submissionSuccess(deletionToken)

```





### 2.4.3 Use Case #3 <Moderator : Review Story>

#### 2.4.3.1 Use Case Scenario #3 <Moderator : Review Story>

Use case name	Moderator : Review Story
Description	Moderator dapat meninjau cerita yang telah melalui pra-moderasi AI dan memutuskan untuk mempublikasikan atau menolak
Pre-condition	Moderator telah login dan terdapat cerita dalam queue moderasi
Post-condition	Cerita telah ditinjau dan status publikasinya ditentukan

#### *Main Scenario*

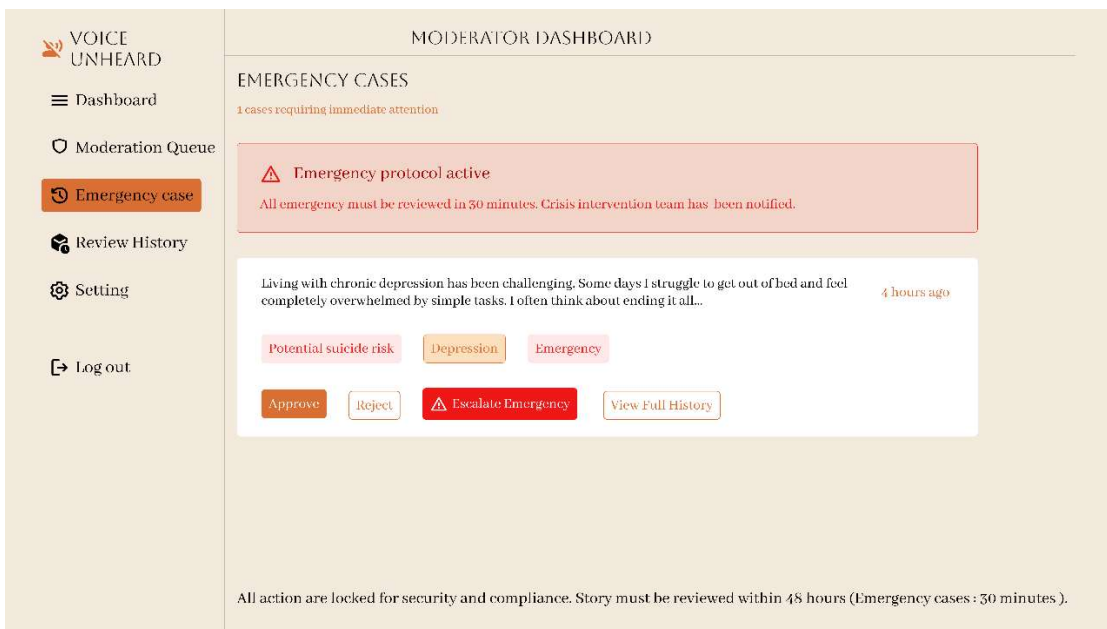
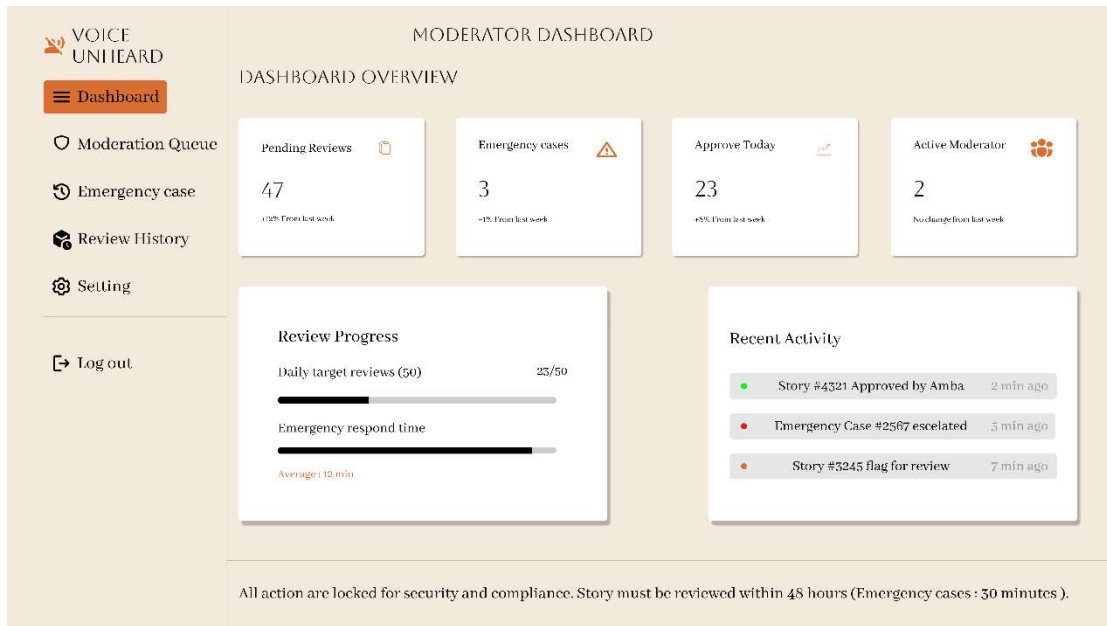
User	Sistem
------	--------

1. Mengakses dashboard moderasi	2. Menampilkan daftar cerita yang perlu ditinjau
	3. Menampilkan hasil pra-moderasi AI untuk setiap cerita
4. Memilih cerita untuk ditinjau	
	5. Menampilkan cerita tanpa informasi identitas pengirim
6. Membaca dan mengevaluasi konten cerita	
7. Memilih "Approve" untuk publikasi	
	8. Memindahkan cerita ke status "Published"
	9. Mengirim notifikasi internal bahwa cerita telah dipublikasi

***Alternative flow - Emergency Escalation***

User	Sistem
4. Memilih cerita dengan flag darurat dari AI	
	5. Menampilkan cerita dengan peringatan darurat
6. Mendeteksi indikasi ancaman bunuh diri	
7. Mengaktifkan protokol eskalasi darurat	
	8. Sistem mencatat insiden tanpa identitas pengguna
	9. Menampilkan resource bantuan untuk ditambahkan ke respon

### 2.4.3.2 UI Design dan Deskripsi Objek UI #3 <Moderator : Review Story>



### C001 - Moderation Dashboard

#### Tabel Deskripsi Objek UI

Layar	Nama Layar	Deskripsi
C001	Moderation Dashboard	Dashboard untuk moderator meninjau

		cerita dengan informasi AI analysis
--	--	-------------------------------------

### **C001 : Moderation Dashboard**

Id_Objek	Jenis	Label	Keterangan
Text 1	Text	Moderation Queue	Judul dashboard
List	ListView	Story Queue	Daftar cerita yang perlu ditinjau
Panel 1	Panel	AI Analysis Results	Panel menampilkan hasil analisis AI
TextArea	TextArea	Story Content	Area menampilkan isi cerita
Button 1	Button	Approve for Publication	Tombol menyetujui publikasi
Button 2	Button	Reject	Tombol menolak cerita
Button 3	Button	Emergency Escalation	Tombol eskalasi darurat
Panel 2	Panel	Emergency Resources	Panel resource bantuan untuk kasus darurat

### **2.4.3.3 Identifikasi Objek dan Tipe Kelas #3 <Moderator : Review Story>**

#### ***Tabel Objek Perancangan***

No	Nama Objek Baru	Jenis / Tipe Kelas
1.	ModerationDashboardView	UI
2.	ModerationController	Controller
3.	StoryModel	Entity
4.	EmergencyEscalationService	Service
5.	AIAnalysisResult	Entity

### **2.4.3.4 Sequence Diagram #3 <Moderator : Review Story>**

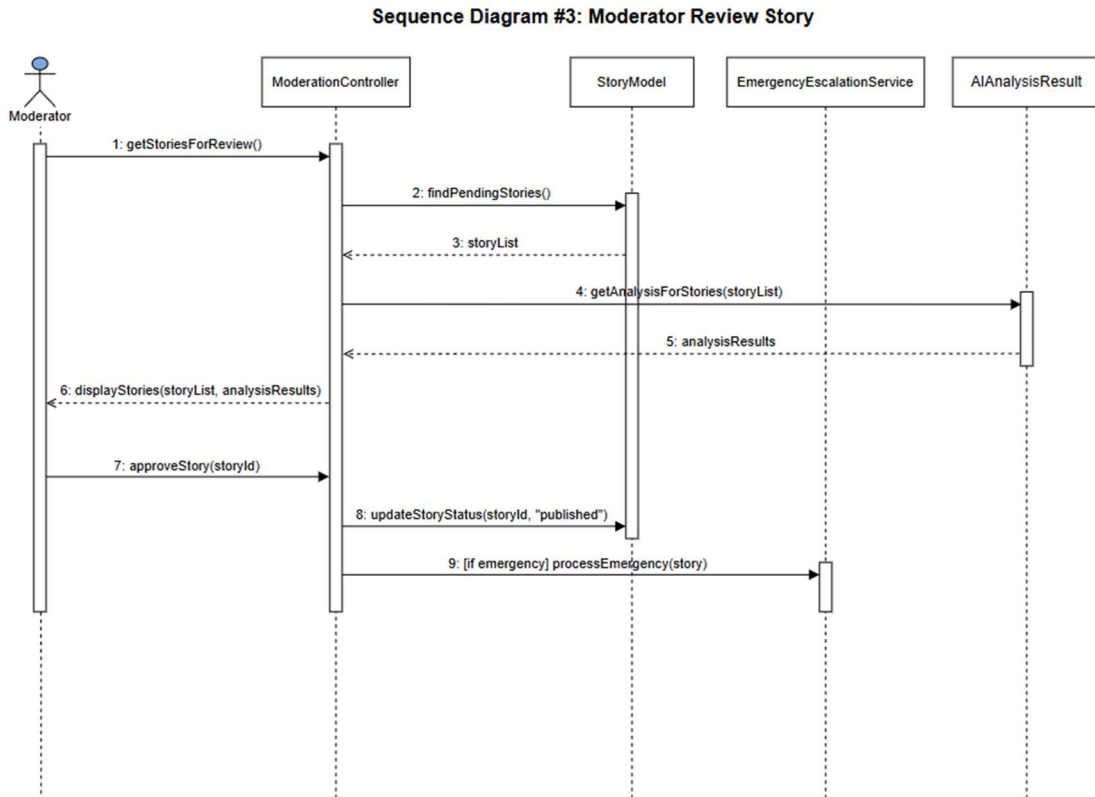
ModerationDashboardView -> ModerationController: getStoriesForReview()

ModerationController -> StoryModel: findPendingStories()

StoryModel -> ModerationController: storyList

ModerationController -> AIAnalysisResult: getAnalysisForStories(storyList)

AIAnalysisResult -> ModerationController: analysisResults  
 ModerationController -> ModerationDashboardView: displayStories(storyList, analysisResults)  
 ModerationDashboardView -> ModerationController: approveStory(storyId)  
 ModerationController -> StoryModel: updateStoryStatus(storyId, "published")



## 2.4.4 Use Case #4 <Anonymous User : View Stories>

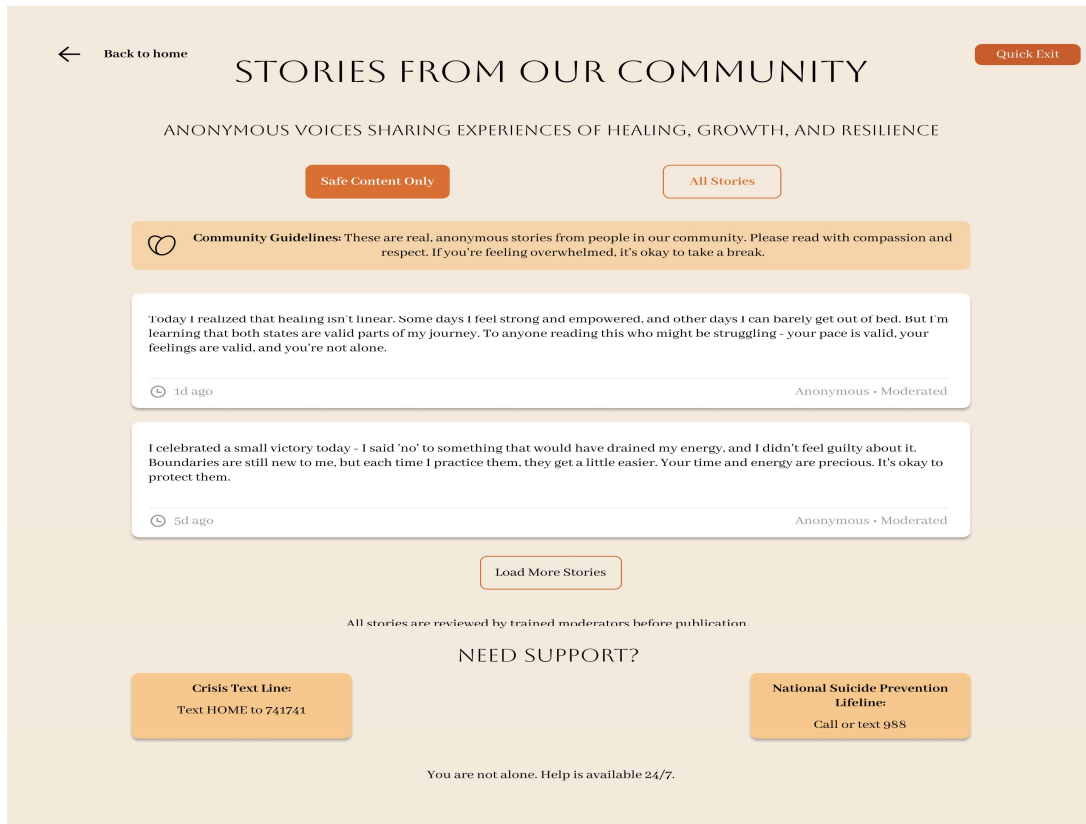
### 2.4.4.1 Use Case Scenario #4 <Anonymous User : View Stories>

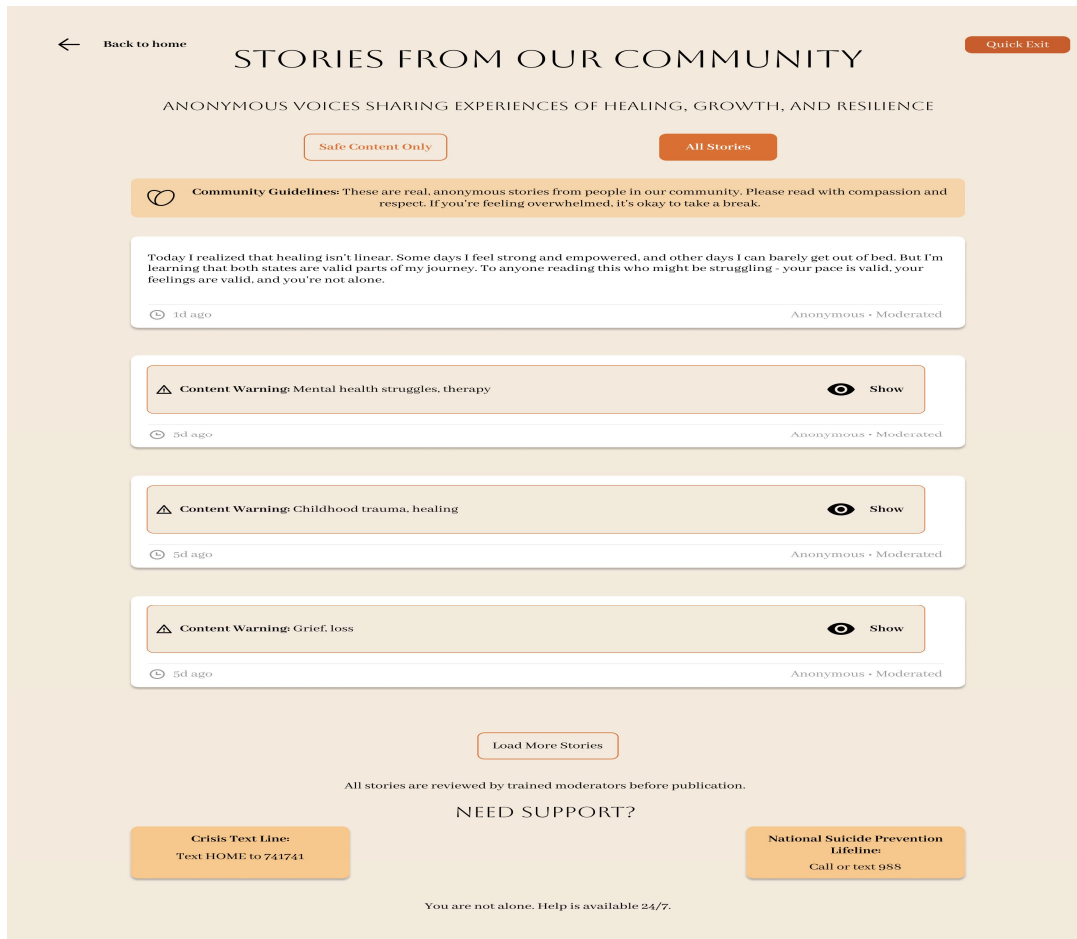
Use case name	Anonymous User : View Stories
Description	Pengguna anonim dapat melihat cerita yang telah dipublikasikan dengan content warning yang sesuai
Pre-condition	Terdapat cerita yang telah disetujui untuk publikasi
Post-condition	Pengguna dapat membaca cerita yang dipublikasikan dengan aman

### Main Scenario

User	Sistem
1. Mengakses halaman cerita publik	2. Menampilkan daftar cerita yang telah dipublikasi
	3. Menampilkan content warning untuk cerita sensitif
4. Memilih cerita untuk dibaca	
	5. Menampilkan peringatan konten jika diperlukan
6. Memberikan persetujuan untuk membaca	
	7. Menampilkan cerita lengkap
	8. Menyediakan tombol "Quick Exit" yang selalu terlihat

#### 2.4.4.2 UI Design dan Deskripsi Objek UI #4 <Anonymous User : View Stories>





## D001 - Story Display Page

### Tabel Deskripsi Objek UI

Layar	Nama Layar	Deskripsi
D001	Public Stories	Halaman menampilkan cerita publik dengan content warning

### D001 : Story Display Page

Id_Objek	Jenis	Label	Keterangan
Text 1	Text	Voices Unheard	Judul platform
Text 2	Text	Stories of Hope and Resilience	Subjudul yang empatik
CardList	Card List	Published Stories	Daftar cerita dalam format card

Modal	Modal	Content Warning	Modal peringatan konten sensitif
Button 1	Button	I Understand, Continue	Tombol persetujuan membaca
Button 2	Button	Quick Exit	Tombol keluar cepat yang selalu terlihat
Panel	Panel	Support Resources	Panel berisi resource bantuan

#### 2.4.4.3 Identifikasi Objek dan Tipe Kelas #4 <Anonymous User : View Stories>

*Tabel Objek Perancangan*

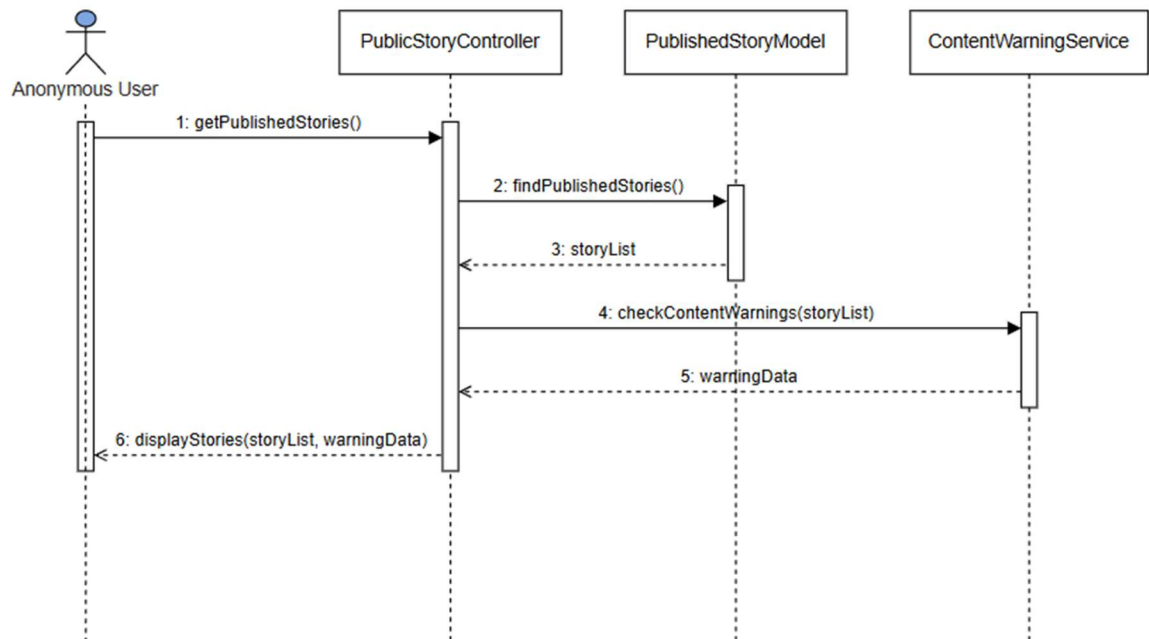
No	Nama Objek Baru	Jenis / Tipe Kelas
1.	StoryDisplayView	UI
2.	PublicStoryController	Controller
3.	PublishedStoryModel	Entity
4.	ContentWarningService	Service

#### 2.4.4.4 Sequence Diagram #4 <Anonymous User : View Stories>

StoryDisplayView -> PublicStoryController: getPublishedStories()  
 PublicStoryController -> PublishedStoryModel: findPublishedStories()  
 PublishedStoryModel -> PublicStoryController: storyList  
 PublicStoryController -> ContentWarningService: checkContentWarnings(storyList)  
 ContentWarningService -> PublicStoryController: warningData  
 PublicStoryController -> StoryDisplayView: displayStories(storyList, warningData)



**Sequence Diagram #4: Anonymous User View Stories**



## 2.4.5 Use Case #5 <Anonymous User : Delete Story>

### 2.4.5.1 Use Case Scenario #5 <Anonymous User : Delete Story>

Use case name	Anonymous User : View Stories
Description	Pengguna dapat menghapus cerita yang telah mereka kirim menggunakan token penghapusan
Pre-condition	Pengguna memiliki token penghapusan yang valid
Post-condition	Cerita berhasil dihapus dari sistem

#### *Main Scenario*

User	Sistem
1. Mengakses halaman delete story	2. Menampilkan form input token penghapusan
3. Memasukkan token penghapusan	

	4. Memvalidasi token penghapusan
	5. Menampilkan preview cerita yang akan dihapus
6. Mengkonfirmasi penghapusan	
	7. Menghapus cerita dari database
	8. Menampilkan konfirmasi penghapusan berhasil

#### 2.4.5.2 UI Design dan Deskripsi Objek UI #5 <Anonymous User : Delete Story>

##### E001 - Delete Story Page

##### *Tabel Deskripsi Objek UI*

Layar	Nama Layar	Deskripsi
E001	Delete Story	Halaman untuk menghapus cerita menggunakan token penghapusan

##### E001 : Delete Story Page

Id_Objek	Jenis	Label	Keterangan
Text 1	Text	Delete Your Story	Judul halaman
Text 2	Text	Enter your deletion token to remove your story	Instruksi penggunaan
RTF 1	RTF	Deletion Token	Input field untuk token penghapusan
Button 1	Button	Find Story	Tombol untuk mencari cerita berdasarkan keyword
Panel	Panel	Story Preview	Panel menampilkan preview cerita yang akan dihapus
Button 2	Button	Confirm Delete	Tombol konfirmasi penghapusan

Button 3	Button	Cancel	Tombol membatalkan penghapusan
Button 4	Button	Quick Exit	Tombol keluar cepat

#### 2.4.5.3 Identifikasi Objek dan Tipe Kelas #5 <Anonymous User : Delete Story>

**Tabel Objek Perancangan**

No	Nama Objek Baru	Jenis / Tipe Kelas
1.	DeleteStoryView	UI
2.	StoryDeletionController	Controller
3.	StoryModel	Entity
4.	TokenValidationService	Service

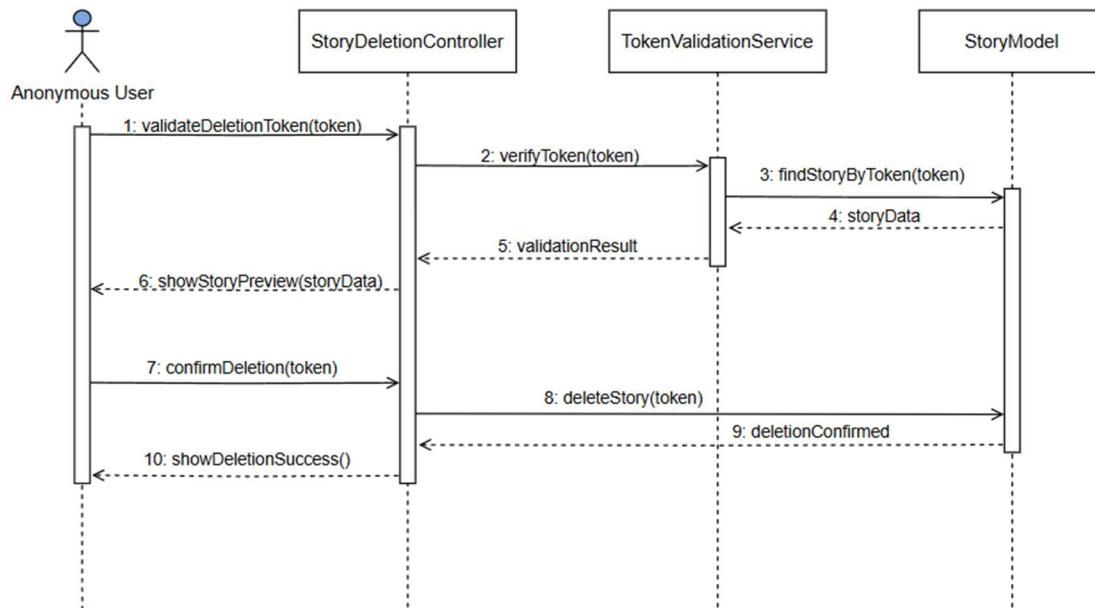
#### 2.4.5.4 Sequence Diagram #5 <Anonymous User : Delete Story>

```

DeleteStoryView -> StoryDeletionController: validateDeletionToken(token)
StoryDeletionController -> TokenValidationService: verifyToken(token)
TokenValidationService -> StoryModel: findStoryByToken(token)
StoryModel -> TokenValidationService: storyData
TokenValidationService -> StoryDeletionController: validationResult
StoryDeletionController -> DeleteStoryView: showStoryPreview(storyData)
DeleteStoryView -> StoryDeletionController: confirmDeletion(token)
StoryDeletionController -> StoryModel: deleteStory(token)
StoryModel -> StoryDeletionController: deletionConfirmed
StoryDeletionController -> DeleteStoryView: showDeletionSuccess()

```

**Sequence Diagram #5: Anonymous User : Delete Story**



## 2.4.6 Use Case #6 <Anonymous User : Quick Exit>

### 2.4.5.1 Use Case Scenario #5 <Anonymous User : Quick Exit>

Use case name	Anonymous User : Quick Exit
Description	Pengguna dapat dengan cepat keluar dari platform dan diarahkan ke situs netral untuk keamanan dengan pembersihan jejak browsing
Pre-condition	Pengguna sedang mengakses platform Voices Unheard
Post-condition	Pengguna berhasil diarahkan ke situs netral dengan jejak browsing dibersihkan

#### *Main Scenario*

User	Sistem
1. Mengklik tombol "Quick Exit" yang selalu terlihat di setiap halaman	2. Mendeteksi permintaan quick exit dan mengaktifkan mode keamanan
	3. Membersihkan browser history,

	cookies, dan session storage
	4. Menutup semua tab terkait platform
	5,. Mengarahkan ke situs netral (google.com) dengan query pencarian umum

#### 2.4.6.2 UI Design dan Deskripsi Objek UI #6 <Anonymous User : Quick Exit>

##### F001 - Quick Exit Interface

##### *Tabel Deskripsi Objek UI*

Layar	Nama Layar	Deskripsi
F001	Quick Exit Interface	Halaman untuk menghapus cerita menggunakan token penghapusan

##### F001 : Quick Exit Interface

Id_Objek	Jenis	Label	Keterangan
Button 1	Button	Quick Exit	Tombol floating yang selalu terlihat, posisi konsisten di semua halaman
Modal 1	Modal	Exit Confirm	Modal dengan pilihan confirm exit (ya/tidak)

#### 2.4.6.3 Identifikasi Objek dan Tipe Kelas #5 <Anonymous User : Quick Exit>

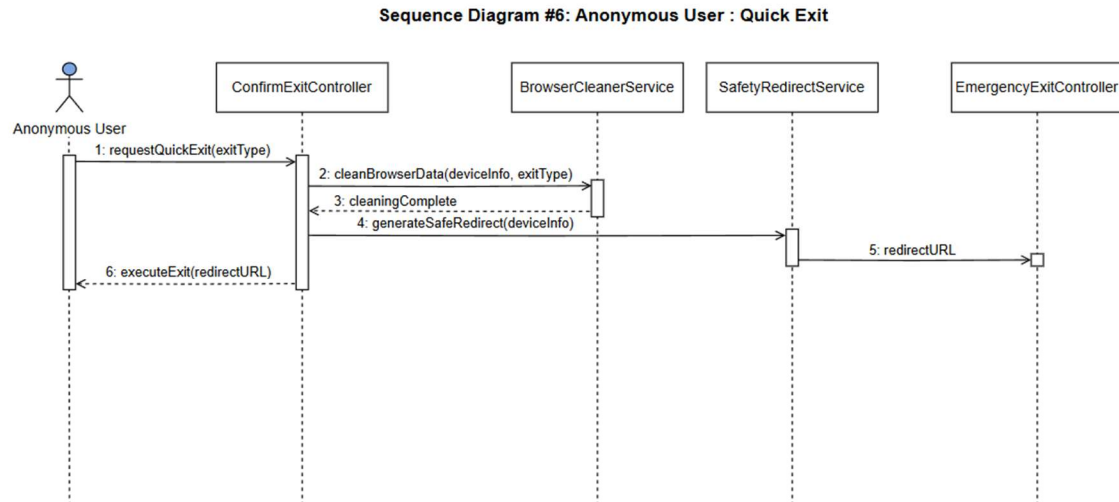
##### **Tabel Objek Perancangan**

No	Nama Objek Baru	Jenis / Tipe Kelas
1.	QuickExitView	UI
2.	ConfirmExitController	Controller

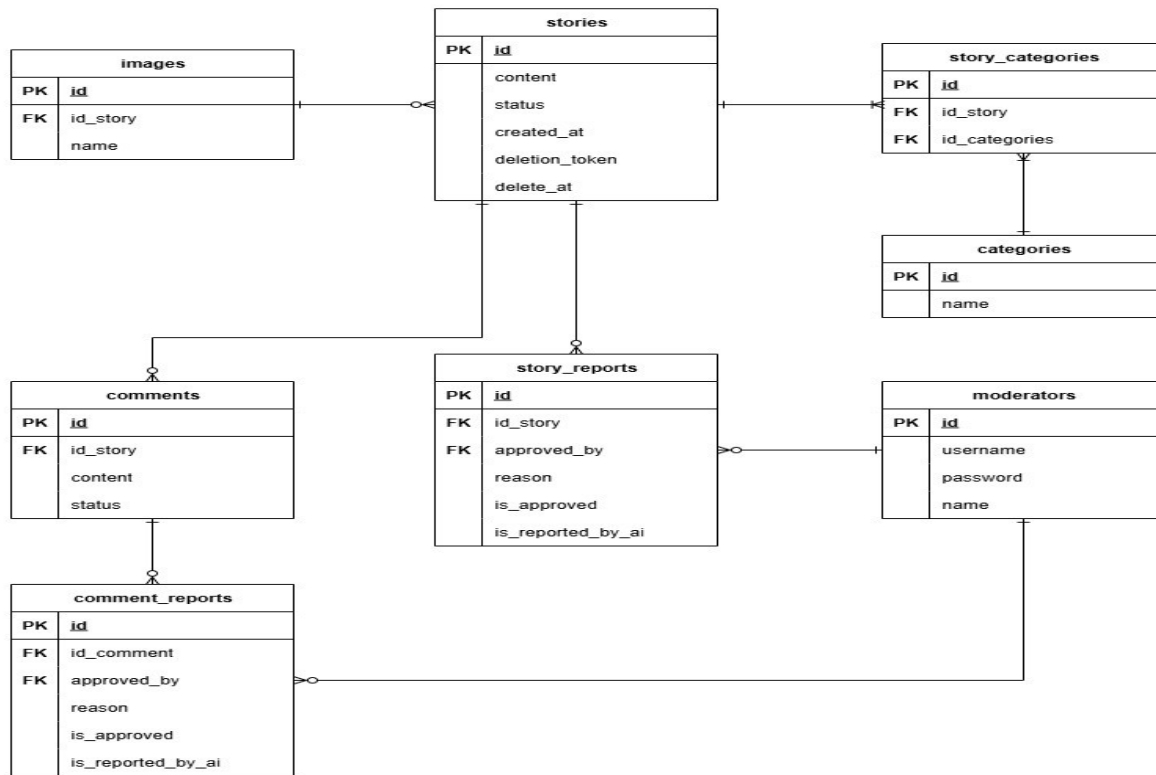
3.	BrowserCleanerService	Service
4.	SafetyRedirectService	Service

#### 2.4.6.4 Sequence Diagram #6 <Anonymous User : Quick Exit >

QuickExitView -> ConfirmExitController: requestQuickExit(exitType)  
ConfirmExitController -> BrowserCleanerService: cleanBrowserData(deviceInfo, exitType)  
BrowserCleanerService -> ConfirmExitController: cleaningComplete  
ConfirmExitController -> SafetyRedirectService: generateSafeRedirect(deviceInfo)  
SafetyRedirectService -> EmergencyExitController: redirectURL  
ConfirmExitController -> QuickExitView: executeExit(redirectURL)



## 2.5 Diagram Kelas Keseluruhan



## 2.6 Perancangan Detail Kelas

### TABEL KELAS :

ID Kelas	Nama Kelas	Perancangan Atribute (visibility)	Method / Operation
1.	AnonymousUser	- sessionId : String	+ submitStory(content: String, file: File) : String + viewStories() : List<Story> + deleteStory(token: String) : boolean + requestQuickExit() : void
2.	Moderator	- id : int - username : String - password : String - role : String -	+ login(credentials: LoginCredentials) : boolean + reviewStory(storyId: int) : Story + approveStory(storyId: int) : void +

		loginTimestamp : DateTime	rejectStory(storyId: int, reason: String) : void + escalateEmergency(storyId: int) : void + logout() : void
3.	Story	- id : int - content : String - attachmentPath : String - status : String - deletionToken : String - submissionTimestamp : DateTime - aiAnalysisResult : String - emergencyFlag : boolean	+ save() : void + delete() : void + updateStatus(newStatus: String) : void + generateDeletionToken() : String + isExpired() : boolean + getAnonymizedContent() : String
4.	AISystem	- apiEndpoint : String - confidenceThreshold : double - riskCategories : List<String>	+ analyzeContent(content: String) : AnalysisResult + detectRisks(content: String) : List<Risk> + classifyContent(content: String) : ContentCategory + checkEmergencyIndicators(content: String) : boolean
5.	SecurityService	- encryptionKey : String - metadataCleanerConfi g : Config	+ cleanMetadata(file: File) : File + generateDeletionToken() : String + encryptData(data: String) : String + anonymizeIPAddress() : void + validateToken(token: String) : boolean + secureDataTransmission(data: Objek) : Objek
6.	StoryController	- aiService : AISystem - securityService : SecurityService - storyRepository : StoryRepository	+ submitStory(storyData: StorySubmissionData) : SubmissionResult + getStoriesForModeration() : List<Story> + reviewStory(storyId: int, decision: String) : void + getPublishedStories() : List<Story> +



			deleteStoryByToken(token: String) : boolean
7.	AuthenticationController	- sessionManager : SessionManager - encryptionService : EncryptionService	+ authenticateModerator(credentials: LoginCredentials) : AuthResult + validateSession(sessionId: String) : boolean + createSecureSession(moderator: Moderator) : String + logout(sessionId: String) : void
8.	EmergencyEscalationService	-emergencyContacts : List<Contact> - escalationRules : List<Rule>	+ processEmergency(story: Story) : void + notifyEmergencyTeam(incident: Incident) : void + generateEmergencyReport(storyId: int) : Report + provideResourceLinks() : List<Resource>
9.	ContentWarningService	- warningCategories : Map<String, String> - sensitivityLevels : List<String>	+ generateWarning(story: Story) : ContentWarning + checkSensitiveContent(content: String) : boolean + getWarningMessage(category: String) : String
10.	TraumaInformedUI	- exitButtonConfig : Config - colorScheme : ColorScheme - languageSettings : LanguageConfig	+ displayQuickExitButton() : void + showEmpathicMessage(message: String) : void + applyTraumaInformedDesign() : void + handleTriggerWarnings() : void

## 2.7 Perancangan Algoritma dan/atau Query

Bagian ini berisi kerangka algoritma untuk proses-proses yang dianggap cukup penting dalam sistem Voices Unheard.

### Algoritma #1

Nama Kelas : SecurityService

Nama Operasi : cleanMetadata()

Algoritma :

```

def cleanMetadata(self, file):
    # Validasi file yang diunggah
    if not self.isValidFile(file):
        raise InvalidFileException("File format not supported")

    # Gunakan exiftool untuk menghapus semua metadata
    try:
        # Hapus semua EXIF data, GPS coordinates, dan metadata lainnya
        cleaned_file = exiftool.execute("-all=", file)

        # Verifikasi bahwa metadata telah dihapus
        if self.hasMetadata(cleaned_file):
            raise MetadataCleaningException("Failed to remove all metadata")

        # Log pembersihan metadata tanpa menyimpan identitas
        self.logMetadataCleaning(file.type, file.size)

        return cleaned_file

    except Exception as e:
        self.logSecurityIncident("Metadata cleaning failed", e)
        raise SecurityException("File processing failed")

```

## Algoritma #2

Nama Kelas : AISystem

Nama Operasi : analyzeContent()

Algoritma :

```

def analyzeContent(self, content):
    # Pra-pemrosesan konten
    processed_content = self.preprocessText(content)

    # Analisis sentimen dan deteksi risiko
    sentiment_analysis = self.analyzeSentiment(processed_content)
    risk_indicators = self.detectRiskIndicators(processed_content)

    # Klasifikasi konten
    content_category = self.classifyContent(processed_content)

    # Deteksi indikator darurat
    emergency_indicators = self.checkEmergencyIndicators(processed_content)

```

```

# Buat hasil analisis
analysis_result = AnalysisResult(
    sentiment=sentiment_analysis,
    risks=risk_indicators,
    category=content_category,
    emergency_flag=len(emergency_indicators) > 0,
    confidence_score=self.calculateConfidence(),
    recommended_action=self.getRecommendedAction()
)

# Log analisis tanpa menyimpan konten asli
self.logAnalysis(analysis_result.category, analysis_result.confidence_score)

return analysis_result

```

Algoritma #3

Nama Kelas : StoryController

Nama Operasi : submitStory()

Algoritma :

```

def submitStory(self, story_data, file=None):
    try:
        # Anonimisasi request (hapus IP tracking)
        self.securityService.anonymizeRequest()

        # Bersihkan metadata dari file jika ada
        cleaned_file = None
        if file:
            cleaned_file = self.securityService.cleanMetadata(file)

        # Pra-moderasi dengan AI
        ai_analysis = self.aiService.analyzeContent(story_data.content)

        # Buat objek story
        story = Story(
            content=story_data.content,
            attachment=cleaned_file,
            ai_analysis_result=ai_analysis,
            status="pending_moderation",
            deletion_token=self.securityService.generateDeletionToken()
        )

```

```

# Simpan story ke database dengan enkripsi
story.save()

# Jika terdeteksi emergency, eskalasi otomatis
if ai_analysis.emergency_flag:
    self.emergencyService.processEmergency(story)

return SubmissionResult(
    success=True,
    deletion_token=story.deletion_token,
    message="Story submitted successfully for review"
)

except Exception as e:
    self.logSecurityIncident("Story submission failed", e)
    return SubmissionResult(
        success=False,
        message="Submission failed. Please try again."
    )

```

#### Algoritma #4

Nama Kelas : EmergencyEscalationService

Nama Operasi : processEmergency()

Algoritma :

```

def processEmergency(self, story):
    # Buat incident report tanpa informasi identitas
    incident = EmergencyIncident(
        story_id=story.id,
        urgency_level=self.assessUrgencyLevel(story),
        detected_risks=story.ai_analysis_result.risks,
        timestamp=datetime.now()
    )

    # Prioritaskan review manual
    story.updateStatus("emergency_review")

    # Notifikasi tim moderasi darurat
    self.notifyEmergencyModerators(incident)

    # Siapkan resource bantuan
    support_resources = self.getSupportResources(incident.detected_risks)

```

```
# Log incident untuk tracking tanpa identitas
self.logEmergencyIncident(incident)

return incident
```

#### Algoritma #5

Nama Kelas : AuthenticationController  
Nama Operasi : authenticateModerator()  
Algoritma :

```
def authenticateModerator(self, credentials):
    try:
        # Enkripsi kredensial
        encrypted_credentials = self.encryptionService.encrypt(credentials)

        # Validasi dengan database
        moderator = self.moderatorRepository.findByCredentials(encrypted_credentials)

        if moderator and self.validatePassword(credentials.password, moderator.password_hash):
            # Buat secure session
            session_token = self.createSecureSession(moderator)

            # Log login berhasil
            self.logSuccessfulLogin(moderator.id)

            return AuthResult(
                success=True,
                session_token=session_token,
                moderator=moderator
            )
        else:
            # Log percobaan login gagal
            self.logFailedLogin(credentials.username)

            return AuthResult(
                success=False,
                message="Invalid credentials"
            )

    except Exception as e:
        self.logSecurityIncident("Authentication error", e)
```

```
return AuthResult(  
    success=False,  
    message="Authentication failed"  
)
```

#### Query #1

Query :

```
SELECT id, content, ai_analysis_result, submission_timestamp  
FROM stories  
WHERE status = 'pending_moderation'  
ORDER BY  
    CASE WHEN ai_analysis_result LIKE '%emergency%' THEN 1 ELSE 2 END,  
    submission_timestamp ASC;
```

**Keterangan :** Query untuk mendapatkan daftar cerita yang perlu dimoderasi, dengan prioritas cerita darurat

#### Query #2

Query :

```
SELECT id, content, attachment_path, submission_timestamp  
FROM stories  
WHERE status = 'published'  
AND submission_timestamp >= NOW() - INTERVAL '30 days'  
ORDER BY submission_timestamp DESC;
```

**Keterangan :** Query untuk mendapatkan cerita yang dipublikasi dalam 30 hari terakhir

#### Query #3

Query :

```
DELETE FROM stories  
WHERE deletion_token = ?  
AND status IN ('pending_moderation', 'published');
```

**Keterangan :** Query untuk menghapus cerita berdasarkan token penghapusan

#### Query #4

Query :

```
UPDATE stories
SET status = ?,
    moderation_timestamp = NOW(),
    moderator_id = ?
WHERE id = ? AND status = 'pending_moderation';
```

**Keterangan :** Query untuk mengupdate status cerita setelah moderasi

#### Query #5

Query :

```
SELECT COUNT(*) as incident_count,
    DATE(submission_timestamp) as incident_date
FROM stories
WHERE ai_analysis_result LIKE '%emergency%'
AND submission_timestamp >= NOW() - INTERVAL '7 days'
GROUP BY DATE(submission_timestamp);
```

**Keterangan :** Query untuk mendapatkan statistik insiden darurat dalam 7 hari terakhir

### 3. Matriks Keruntutan (Requirement Traceability Matrix)

Mapping requirement dengan Use Case yang direalisasikan

Kode FR	Nama Functional Requirement	Nama Use Case
FR-SUB-01	Sistem harus memungkinkan pengguna mengirimkan cerita tanpa membuat akun atau login (Anonim)	Anonymous User : Submit Story
FR-SUB-02	Sistem harus secara otomatis menghapus semua metadata (EXIF, GPS) dari file yang diunggah di sisi server	Anonymous User : Submit Story
FR-SUB-03	Sistem tidak boleh menyimpan atau mencatat alamat IP asli pengguna	Anonymous User : Submit Story
FR-MOD-01	Sistem harus menyediakan dasbor moderasi	Moderator : Review Story

	yang tidak menampilkan informasi identitas pengirim	
FR-MOD-02	Semua kiriman harus melalui proses pra-moderasi oleh moderator tools berbasis AI sebelum dipublikasikan	Moderator : Review Story
FR-MOD-03	Sistem harus memiliki alur kerja eskalasi untuk kasus darurat (misalnya, ancaman bunuh diri)	Moderator : Review Story
FR-CTL-01	Sistem harus menyediakan mekanisme "token penghapusan" agar pengguna dapat menghapus cerita mereka	Anonymous User : Delete Story
FR-CTL-02	Sistem harus menyediakan alur persetujuan (consent) yang jelas dan berlapis untuk penggunaan cerita	Anonymous User : View Stories
FR-CTL-03	Sistem harus menyediakan autentikasi yang aman untuk moderator	Moderator : Login

## 4. Design Pattern dengan Observer Pattern

### Alasan Pemilihan:

Dalam konteks platform Voices Unheard, terdapat berbagai komponen yang perlu dinotifikasi ketika terjadi perubahan status pada cerita (story). Penggunaan Observer Pattern memungkinkan sistem untuk secara otomatis memberitahu berbagai komponen seperti sistem moderasi, layanan eskalasi darurat, dan sistem notifikasi ketika ada perubahan tanpa membuat coupling yang kuat antar komponen. Ini memberikan fleksibilitas dalam menambahkan atau mengubah observer tanpa mempengaruhi struktur utama aplikasi.

### Implementasi Observer Pattern:

#### Observer untuk Monitoring Status Cerita:

```
# Subject: Story
class Story:
    def __init__(self):
        self._observers = []
        self._status = "pending"

    def attach(self, observer):
        self._observers.append(observer)

    def detach(self, observer):
```



```

        self._observers.remove(observer)

    def notify(self):
        for observer in self._observers:
            observer.update(self)

    def set_status(self, status):
        self._status = status
        self.notify()

# Observer: ModerationService
class ModerationService:
    def update(self, story):
        if story.status == "pending_moderation":
            self.add_to_moderation_queue(story)

# Observer: EmergencyService
class EmergencyService:
    def update(self, story):
        if story.status == "emergency" or story.ai_analysis_result.emergency_flag:
            self.escalate_emergency(story)

# Observer: NotificationService
class NotificationService:
    def update(self, story):
        if story.status == "published":
            self.notify_publication_success()

```

### Deskripsi Implementasi:

- Membuat sistem **Story** sebagai Subject yang dapat diobservasi oleh berbagai layanan seperti **ModerationService**, **EmergencyService**, dan **NotificationService**.

### Alasan Implementasi:

- Ketika status cerita berubah (misalnya dari "pending" ke "emergency" atau "published"), semua observer yang relevan akan secara otomatis di notifikasi tanpa Story perlu mengetahui detail implementasi setiap observer. Ini memudahkan penambahan layanan monitoring baru atau modifikasi dalam proses notifikasi yang sudah ada.

### Contoh Penggunaan:

- Misalnya, saat AI mendeteksi indikator darurat dalam cerita, **Story** akan mengubah statusnya menjadi "emergency". **Emergency Service** akan otomatis mendapat notifikasi dan mengaktifkan protokol eskalasi darurat, sementara **ModerationService** akan memindahkan cerita ke queue prioritas tinggi.

Implementasi Observer Pattern ini memberikan fleksibilitas dalam penanganan berbagai event dan

status change dalam sistem tanpa harus mengubah kode inti aplikasi. Dengan menggunakan pola desain ini, platform dapat mengakomodasi kebutuhan untuk menambahkan sistem monitoring atau notifikasi baru dengan lebih mudah dan terstruktur.

#### 4.1 Jenis Pengujian

No.	Jenis Pengujian	Tujuan Pengujian	Kebutuhan Terkait	Deskripsi Singkat
1	Functional Testing	Memastikan seluruh fungsi berjalan sesuai spesifikasi kebutuhan.	FR-SUB-01, FR-SUB-02, FR-SUB-03, FR-MOD-01, FR-MOD-02, FR-MOD-03, FR-CTL-01, FR-CTL-02	Menguji fitur inti seperti pengiriman cerita anonim, penghapusan cerita dengan token, pra-moderasi AI, dan persetujuan penggunaan.
2	Security Testing	Memastikan keamanan data, privasi, dan ketahanan terhadap serangan.	NFR-SEC-01, NFR-SEC-02	Uji TLS 1.3+, AES-256, penghapusan metadata file, pencegahan pencatatan IP
3	Usability Testing	Memastikan antarmuka aman, ramah pengguna, dan sesuai prinsip desain berbasis trauma.	NFR-UX-01	Menguji kemudahan penggunaan, aksesibilitas, bahasa empatik, serta alur navigasi sederhana.
4	Performance Testing	Mengukur kecepatan, kestabilan, dan skalabilitas sistem.	NFR-PERF-01, NFR-PERF-02, NFR-SCALE-01	Melakukan load test, stress test, serta mengukur waktu muat halaman < 3 detik pada koneksi 3G dan kemampuan scaling otomatis.
5	Integration Testing	Memastikan modul saling terhubung dan alur kerja sistem sesuai harapan.	FR-SUB-02 -> FR-MOD-02 -> FR-MOD-03	Contoh: pengiriman cerita -> pembersihan metadata -> pra-moderasi AI -> publikasi.
6	Regression Testing	Memastikan perubahan kode tidak merusak	Semua FR/NFR yang terdampak update	Dijalankan setiap kali ada update atau bug fix untuk memverifikasi stabilitas

		fungsi yang sudah berjalan.		sistem.
7	Acceptance Testing	Memastikan sistem memenuhi kriteria penerimaan oleh pengguna akhir.	Semua FR/NFR	Dilakukan melalui UAT dengan skenario nyata sesuai kebutuhan di SRS.

## 4.2 Prosedur Pengujian

### - Persiapan Lingkungan Uji

Menyediakan staging server, Menggunakan data dummy (teks cerita, gambar) sesuai kebutuhan pengujian.

### - Pembuatan Test Case

Menyusun test case berdasarkan *Requirement Traceability Matrix* (RTM) mencakup skenario positif, negatif, dan edge case.

### - Pelaksanaan Pengujian

Menjalankan functional, usability, security, dan performance testing sesuai test case yang telah dibuat.

### - Pencatatan Bug/Issue

Mencatat bug yang ditemukan, lengkap dengan langkah reproduksi, screenshot, dan prioritas.

### - Perbaikan & Retest

Developer memperbaiki bug, QA melakukan pengujian ulang pada modul yang terdampak.

### - Evaluasi & Laporan Akhir

Memastikan semua pengujian selesai, menyusun laporan hasil uji, dan menyerahkan ke Project Manager.