Kelompok 3

Web ZonaCerdas

Version 1.2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 22/11/2024 | 1.0 | Revisi pemilihan warna dari monokrom ke warna biru | Kelompok |
| 26/11/2024 | 1.1 | Revisi penggunaan profile pic dari user. Sekarang profile picture bersifat unisversal dan sudah ada dalam file web | Eko Dilapanga |
| 30/11/2024 | 1.2 | Revisi penyimpanan video yang awalnya dalam file web, sekarang disimpan di dalam Gdrive untuk efisiensi storage. | Keefa Lasut |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 5

1.4 References 5

1.5 Overview 5

2. Architectural Representation 6

3. Architectural Goals and Constraints 6

4. Use-Case View 7-8

5. Logical View 9-10

6. Process View 11-12

7. Information / Data View 13-14

8. Development View 15-16

9. Physical View 17

10. Size and Performance 18

11. Quality 18

# Introduction

Kemajuan teknologi yang pesat telah merevolusi cara informasi diakses dan dibagikan. Platform e-learning telah muncul sebagai pilar utama pendidikan modern, memungkinkan peserta belajar untuk mengakses konten pendidikan kapan saja dan di mana saja. Dokumen Sistem Analisis dan Desain (SAD) ini menyajikan cetak biru yang komprehensif untuk sebuah situs web e-learning yang dirancang untuk menyimpan dan menyampaikan konten pendidikan berbasis video secara efisien.

## Purpose

Tujuan dari platform e-learning ini adalah menyediakan solusi yang terpusat dan ramah pengguna bagi para tutor dan peserta belajar. Platform ini bertujuan untuk menjembatani kesenjangan antara pendidik dan peserta didik dengan menawarkan sebuah platform di mana:

* **Peserta didik** dapat menjelajahi dan menonton video pendidikan untuk meningkatkan pengetahuan mereka.
* **Admin** dapat mengawasi operasi sistem untuk memastikan fungsionalitas yang lancar dan kepuasan pengguna.

## Scope

Sasaran utama dari situs web e-learning ini adalah:

* **Aksesibilitas**: Memungkinkan pengguna untuk mengakses sumber belajar berbasis video dari perangkat apa pun yang terhubung dengan internet.
* **Pengorganisasian Konten**: Menyediakan alat untuk mengategorikan, mencari, dan memfilter video pendidikan secara efektif.
* **Keterlibatan**: Memfasilitasi interaksi melalui fitur seperti komentar dan umpan balik pada video.
* **Skalabilitas**: Merancang arsitektur sistem yang dapat menangani peningkatan jumlah pengguna dan konten video.

## Definitions, Acronyms, and Abbreviations

* **SAD (System Analysis and Design)**: Analisis dan Desain Sistem.
* **E-learning**: Proses pembelajaran yang menggunakan media elektronik, khususnya internet, untuk menyampaikan materi pendidikan.
* **SDG no. 4:** Sustainable Development Goal ke-empat bertujuan untukmenjamin kualitas pendidikan yang inklusif dan merata serta meningkatkan kesempatan belajar sepanjang hayat untuk semua.
* **PHP:** PHP adalah bahasa pemrograman *server-side* yang digunakan untuk mengembangkan aplikasi web dinamis.
* **HTML:** HTML adalah bahasa markup yang digunakan untuk membuat struktur halaman web.
* **CSS:** CSS digunakan untuk mengatur tampilan dan tata letak elemen yang dibuat dengan HTML.
* **JS:** JavaScript adalah bahasa pemrograman *client-side* yang digunakan untuk membuat halaman web lebih interaktif.
* **DBMS:** DBMS adalah perangkat lunak yang digunakan untuk mengelola, menyimpan, dan mengambil data dalam database.
* **LMS:** LMS adalah sistem manajemen pembelajaran yang digunakan untuk mengelola kursus, materi belajar, peserta didik, dan evaluasi pembelajaran secara online.

## References

* + PHP and MySQL web development (2010)

Oleh: Luke Welling, Laura Thomson.

* + MySQL Database Management System (DBMS) On FTPSite LAPAN Bandung (2021)

Oleh: Bhupesh Rawat, Suryari Purnama, Mulyati

* + arc42 by Example: Software architecture documentation in practice

Oleh: By Dr. Gernot Starke, Michael Simons, Stefan Zörner

* + Collaborative Web Development: Strategies and Best Practices for Web Teams

Oleh: Jessica Burdman

## Overview

Dokumen Sistem Analisis dan Desain (SAD) ini dirancang untuk memberikan pemahaman terstruktur tentang proses perancangan platform e-learning berbasis video.

# Architectural Representation

Bagian ini menjelaskan arsitektur perangkat lunak untuk sistem e-learning yang diusulkan dan bagaimana arsitektur tersebut direpresentasikan. Dokumen ini mencakup berbagai pandangan (views) yang diperlukan untuk memahami struktur dan fungsi sistem, yaitu:

* Use case view : menggunakan use case diagram
* Logical view : menggunakan class diagram
* Process view : menggunakan activity diagram
* Information / Data view : menggunakan entity relationship diagram
* Development view : menggunakan component diagram
* Physical view : menggunakan deployment diagram

# Architectural Goals and Constraints

Sistem e-learning ini dirancang untuk memastikan keamanan dan privasi pengguna dengan menerapkan otentikasi yang aman, validasi data, dan perlindungan terhadap akses tidak sah. Selain itu, sistem harus mudah diakses, sederhana, dan efisien.

# Software Requirements:

* 1. Authentication: Sistem login dasar untuk memastikan hanya pengguna terdaftar yang dapat mengakses fitur tertentu.
  2. Database Management: MySQL digunakan untuk menyimpan data pengguna, video, dan komentar.
  3. Web Server: Apache dengan PHP untuk menjalankan backend.
  4. Frontend Requirements:

a. HTML, CSS, dan JavaScript untuk antarmuka pengguna.

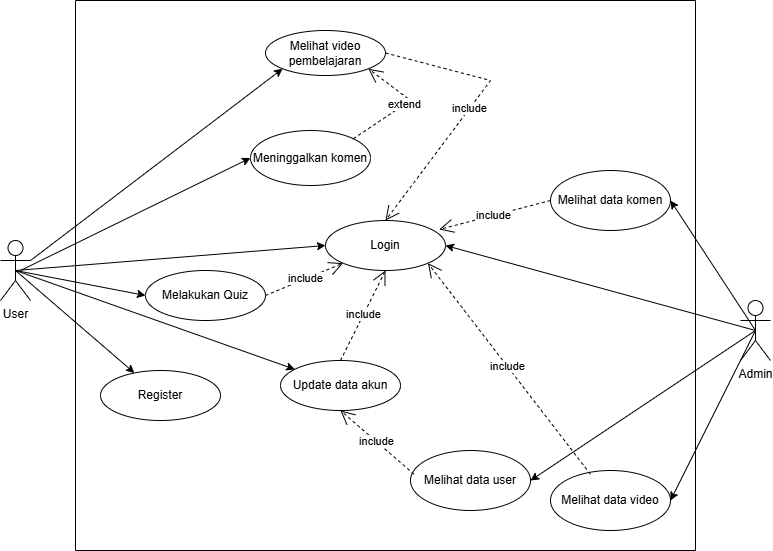
b. Desain responsif untuk mendukung berbagai perangkat.

* 1. Security: Enkripsi kata sandi menggunakan algoritma seperti bcrypt.
  2. Browser Compatibility: Mendukung browser popular seperti chrome dan firefox

# Constraints:

1. Sumber daya terbatas: Hosting website tidak dimungkinkan karena kurangnya akses ke layanan hosting atau infrastruktur yang dapat diandalkan. Karena hamper semua layanan hosting memerlukan subscription dan melebihi budget anggaran yang dimiliki tim.
2. Jangka Waktu Pengembangan: Dengan jangka waktu yang cukup terbatas, maka web tidak dapat dikembengkan dikembangkan secara maksimal sesuai dengan harapan.
3. Pengujian terhadap pengalaman pengguna (UX): uji coba BETA dengan pengguna sesungguhnya mungkin tidak dapat dilakukan dengan baik karena keterbatasan waktu dan sumber daya.

# Use-Case View



**Aktor:**

* User: Aktor utama yang berinteraksi dengan sistem untuk mengakses dan berinteraksi dengan data video pembelajaran dan komentar.
* Admin: Aktor yang memiliki akses lebih untuk mengelola data terkait pengguna dan konten video pembelajaran.

**Use Case untuk User:**

* Register: User dapat mendaftar untuk membuat akun baru.
* Login: User perlu login untuk mengakses fitur lain dalam sistem.
* Melihat video pembelajaran: User dapat menonton video pembelajaran yang disediakan oleh sistem.
* Meninggalkan komentar: Setelah menonton video, User dapat meninggalkan komentar sebagai feedback atau diskusi terkait video tersebut.
* Update data akun: User dapat memperbarui data akun mereka setelah login

Relasi dengan use case lainnya:

* Login memiliki hubungan include dengan Register dan Update data akun, yang berarti bahwa untuk melakukan update data akun atau register, User harus terlebih dahulu login.
* Melihat video pembelajaran memiliki hubungan extend dengan Meninggalkan komentar, yang berarti bahwa setelah menonton video, User dapat memilih untuk meninggalkan komentar, namun ini bersifat opsional dan bisa ditambahkan setelah video ditonton.
* Melakukan quiz memiliki hubungan include dengan login. Berarti user harus login untuk melakukan quiz.

**Use Case untuk Admin:**

* Melihat data user: Admin dapat melihat data pengguna yang terdaftar dalam sistem, seperti informasi profil atau aktivitas.
* Melihat data video: Admin dapat mengelola dan melihat data terkait video pembelajaran yang diunggah ke sistem.
* Melihat data komen: Admin dapat melihat komentar yang ditinggalkan oleh User pada video untuk melakukan analisis.

Relasi dengan use case lainnya:

* Login memiliki hubungan include dengan Melihat data komen, Melihat data user, dan melihat data video, yang berarti Admin hanya bisa mengakses data-data ini setelah login.

# Logical View

# 

# Users Class:

**Atribut:**

* id: Integer yang unik untuk mengidentifikasi setiap pengguna.
* username: String yang menyimpan nama pengguna.
* email: String yang menyimpan alamat email pengguna.
* password: String yang menyimpan kata sandi pengguna.
* created\_at: Timestamp yang mencatat waktu pembuatan akun.
* age: Integer yang mencatat usia pengguna.
* gender: String yang mencatat jenis kelamin pengguna.

**Metode:**

* createUser(): Membuat pengguna baru.
* updateUser(): Memperbarui informasi pengguna.
* deleteUser(): Menghapus pengguna dari sistem.
* getUserById(): Mengambil informasi pengguna berdasarkan ID.
* getAllUsers(): Mengambil semua pengguna yang ada dalam sistem.
* checkCredentials(): Memverifikasi kredensial pengguna saat login.

**Videos Class:**

**Atribut:**

* id: Integer yang unik untuk mengidentifikasi setiap video.
* title: String yang menyimpan judul video.
* video\_url: String yang menyimpan URL video.
* description: Text yang menyimpan deskripsi video.
* created\_at: Timestamp yang mencatat waktu pembuatan video.

**Metode:**

* createVideo(): Membuat video baru.
* updateVideo(): Memperbarui informasi video.
* deleteVideo(): Menghapus video dari sistem.
* getVideoById(): Mengambil informasi video berdasarkan ID.
* getAllVideos(): Mengambil semua video yang ada dalam sistem.
* searchVideos(): Mencari video berdasarkan kriteria tertentu.

**Comment Class:**

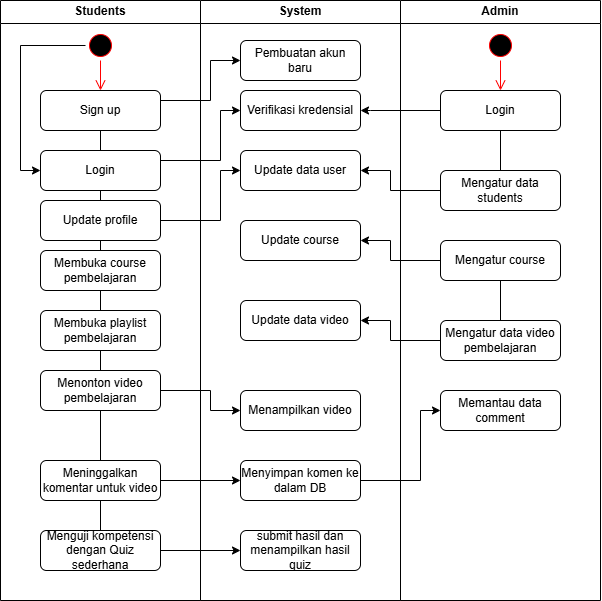
**Atribut:**

* id: Integer yang unik untuk mengidentifikasi setiap komentar.
* user\_id: Integer yang mengacu pada ID pengguna yang meninggalkan komentar.
* video\_id: Integer yang mengacu pada ID video yang dikomentari.
* comment: Text yang menyimpan isi komentar.
* created\_at: Timestamp yang mencatat waktu pembuatan komentar.

**Metode:**

* createComment(): Membuat komentar baru.
* updateComment(): Memperbarui informasi komentar.
* deleteComment(): Menghapus komentar dari sistem.
* getCommentsByUser(): Mengambil semua komentar yang ditinggalkan oleh pengguna tertentu.
* getCommentsByVideo(): Mengambil semua komentar untuk video tertentu.
* getCommentById(): Mengambil informasi komentar berdasarkan ID.

# Process View



**Penjelasan Alur**

**Students (Siswa):**

* **Sign up:** Siswa mendaftar untuk membuat akun baru.
* **Login:** Siswa masuk ke dalam sistem menggunakan kredensial mereka.
* **Update profile:** Siswa memperbarui profil mereka.
* **Membuka course pembelajaran:** Siswa membuka kursus pembelajaran yang tersedia.
* **Membuka playlist pembelajaran:** Siswa membuka daftar putar pembelajaran.
* **Menonton video pembelajaran:** Siswa menonton video pembelajaran yang tersedia.
* **Meninggalkan komentar untuk video:** Siswa meninggalkan komentar pada video yang telah ditonton.
* **Menguji kompetensi dengan Quiz sederhana:** Siswa mengikuti kuis sederhana untuk menguji kompetensi mereka.

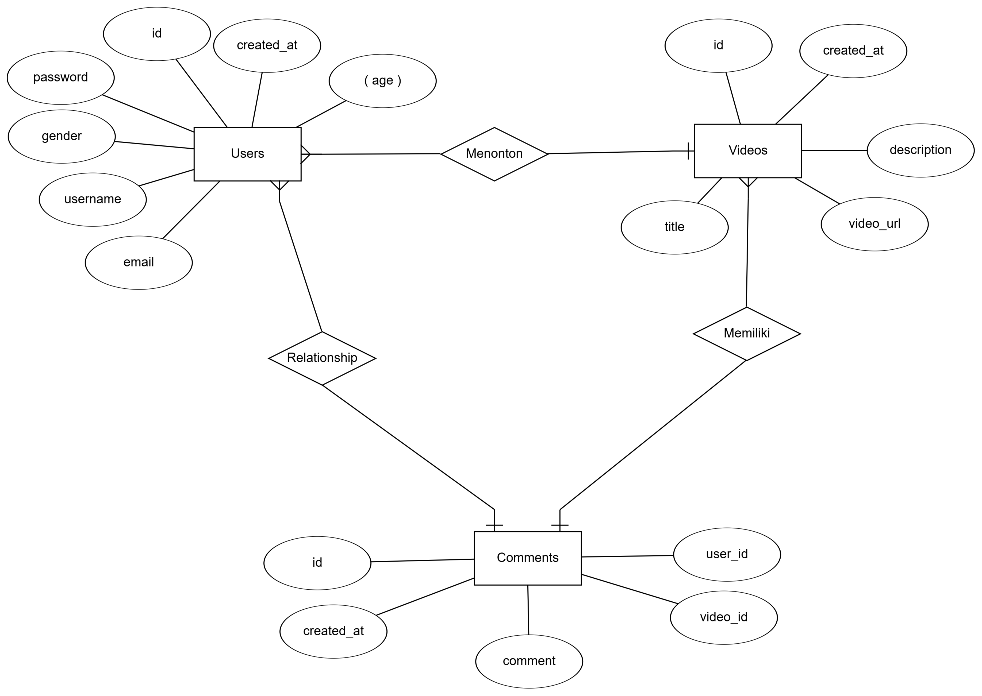
**System (Sistem):**

* **Pembuatan akun baru:** Sistem membuat akun baru untuk siswa yang mendaftar.
* **Verifikasi kredensial:** Sistem memverifikasi kredensial siswa saat login.
* **Update data user:** Sistem memperbarui data pengguna.
* **Update course:** Sistem memperbarui data kursus.
* **Update data video:** Sistem memperbarui data video.
* **Menampilkan video:** Sistem menampilkan video pembelajaran kepada siswa.
* **Menyimpan komen ke dalam DB:** Sistem menyimpan komentar siswa ke dalam basis data.
* **Submit hasil dan menampilkan hasil quiz:** Sistem mengirimkan hasil kuis dan menampilkan hasilnya kepada siswa.

**Admin (Administrator):**

* **Login:** Admin masuk ke dalam sistem menggunakan kredensial mereka.
* **Mengatur data students:** Admin mengatur data siswa.
* **Mengatur course:** Admin mengatur kursus pembelajaran.
* **Mengatur data video pembelajaran:** Admin mengatur data video pembelajaran.
* **Memantau data comment:** Admin memantau komentar yang ditinggalkan oleh siswa.

# Information / Data View



**Entitas dan Atribut**

**Users** (Tabel Pengguna):

* + **Atribut:**
    - id (Primary Key): Identitas unik setiap pengguna.
    - email: Alamat email pengguna.
    - username: Nama pengguna.
    - gender: Jenis kelamin pengguna.
    - password: Kata sandi pengguna.
    - created\_at: Waktu pembuatan akun.
    - age: Usia pengguna.

**Videos** (Tabel Video):

* + **Atribut:**
    - id (Primary Key): Identitas unik setiap video.
    - title: Judul video.
    - description: Deskripsi video.
    - video\_url: URL tempat video diunggah.
    - created\_at: Waktu pembuatan video.

**Comments** (Tabel Komentar):

* + **Atribut:**
    - id (Primary Key): Identitas unik komentar.
    - user\_id (Foreign Key): Mengacu ke Users.
    - video\_id (Foreign Key): Mengacu ke Videos.
    - comment: Isi komentar.
    - created\_at: Waktu pembuatan komentar.

(**Relasi antara Users dan Videos):**

* Tipe Relasi: One-to-Many.
* Penjelasan: Seorang user bisa memiliki banyak video, tetapi satu video hanya dimiliki oleh satu pengguna.

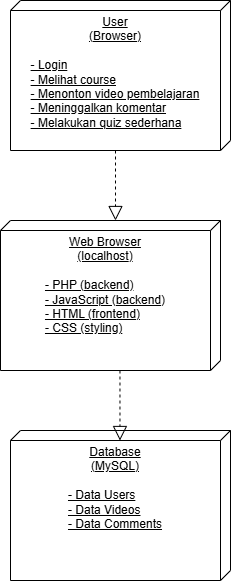
**(Relasi antara Users dan Comments):**

* Tipe Relasi: One-to-Many.
* Penjelasan: Seorang pengguna bisa membuat banyak komentar.

**(Relasi antara Comments dan Videos):**

* Tipe Relasi: Many-to-One.
* Penjelasan: Banyak komentar bisa terkait dengan satu video.

# Development View



**Komponen Utama dalam Development View**

User (Browser)

* Ini adalah sisi klien tempat pengguna mengakses aplikasi melalui web browser.
* Tindakan yang dilakukan pengguna mencakup:
  + - Login: Autentikasi pengguna.
    - Melihat course: Menampilkan daftar kursus yang tersedia.
    - Menonton video pembelajaran: Streaming atau menonton video dari sistem.
    - Meninggalkan komentar: Menambahkan komentar terkait video pembelajaran atau course.
    - Melakukan quiz sederhana: Interaksi kuis sebagai bagian dari aktivitas pembelajaran.

Web Browser (Localhost)

* Ini adalah bagian pengembangan di sisi server (backend) dan sisi klien (frontend).
* Komponen di dalamnya:
  + PHP (backend): Digunakan untuk menangani logika aplikasi di server, seperti proses autentikasi, pengambilan data dari database, dan pengelolaan sesi pengguna.
  + JavaScript (backend): Berperan untuk logika tambahan di sisi klien.
  + HTML (frontend): Digunakan untuk menampilkan antarmuka pengguna (UI).
  + CSS (styling): Menentukan tampilan dan tata letak antarmuka agar lebih menarik dan responsif.
* Localhost mengindikasikan bahwa pengembangan ini dilakukan secara lokal di komputer pengembang sebelum diterapkan ke lingkungan produksi.

Database (MySQL)

* Database MySQL digunakan sebagai penyimpanan data utama.
* Tabel-tabel yang disimpan mencakup:
  + Data Users: Menyimpan informasi pengguna, seperti username, password, dan data profil lainnya.
  + Data Videos: Menyimpan metadata video, seperti judul, deskripsi, URL video, dan informasi terkait.
  + Data Comments: Menyimpan komentar yang ditinggalkan oleh pengguna terkait video atau kursus.

**Alur Kerja dalam Development View**

1. Pengguna (User) membuka aplikasi melalui web browser.
2. Web browser (local server) menangani permintaan pengguna dengan:

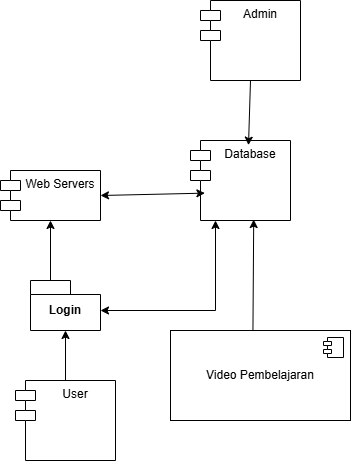
* Menampilkan antarmuka menggunakan HTML dan CSS.
* Menjalankan logika interaktif dengan JavaScript di frontend.
* Mengirim permintaan ke backend (PHP) untuk memproses data seperti login, menampilkan video, atau menyimpan komentar.

1. Backend (PHP) berkomunikasi dengan Database (MySQL):

* Mengambil atau menyimpan data pengguna, video, dan komentar.

1. Hasil dari database dikirim kembali ke web browser, di mana data ditampilkan kepada pengguna.

# Physical View



**Autentikasi**:

* User masuk melalui Login, Data dikirim ke Database untuk validasi.

**Interaksi Konten**:

* User berinteraksi dengan Web Servers untuk mengakses konten.
* Web server mengambil data dari Database.

**Konten Video**:

* Video pembelajaran tersedia melalui komponen Video Pembelajaran, yang mengambil datanya dari database.

**Manajemen Sistem**:

* Admin bertugas memelihara dan mengatur data dalam Database.

# Size and Performance

Tidak ada target untuk performa sistem. Sistem e-learning yang akan dibuat dirancang tanpa batasan khusus terkait kinerja, seperti waktu respon maksimum atau penggunaan sumber daya. Pengembangan akan difokuskan pada memastikan fungsionalitas dasar dapat berjalan dengan baik tanpa mengkhususkan pada kebutuhan performa tinggi atau pengolahan data berskala besar.

# Quality

Sistem yang akan dibangun dirancang untuk memenuhi beberapa karakteristik kualitas utama guna memastikan kehandalan, skalabilitas, dan kemudahan adaptasi di masa mendatang.

Bagaimana sistem yang akan dibangun memenuhi karakteristik-karakteristik seperti:

* Extensibility:
  + Sistem akan dibangun dengan struktur kode yang sederhana namun modular. Hal ini memungkinkan penambahan fitur baru, seperti halaman informasi tambahan atau fungsi pendaftaran pengguna, tanpa memengaruhi bagian lain dari sistem.
  + Menggunakan pendekatan file terpisah untuk setiap fitur, sehingga memudahkan pengembangan lebih lanjut.
* Reliability:
  + Sistem akan dilengkapi dengan validasi input sederhana untuk mengurangi kesalahan pengguna saat memasukkan data.
  + Menggunakan database yang stabil dengan pemantauan secara berkala untuk menjaga keamanan data tanpa memerlukan mekanisme yang rumit.
* Portability:
  + Antarmuka pengguna akan dirancang responsif untuk mendukung akses dari perangkat desktop dan ponsel tanpa memerlukan perangkat lunak tambahan.
* Safety:
  + Memastikan data yang dimasukkan pengguna divalidasi untuk mencegah kesalahan atau serangan berbahaya.
* Security:
  + Pengguna hanya dapat mengakses halaman yang sesuai dengan peran mereka, seperti admin atau user biasa.
  + Data sensitif seperti kata sandi dienkripsi menggunakan algoritma seperti bcrypt sebelum disimpan di database.
* Privacy:
  + Memberikan kendali kepada pengguna untuk melihat, mengubah, atau menghapus data pribadi mereka sesuai kebijakan privasi.

Jika ada dijelaskan, jika tidak tuliskan “tidak ada”