

**BUSINESS REQUIREMENTS DOCUMENT (BRD)  
SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**



Dosen Pengampu:

Jefry Sunupurwa, S.Kom.,M.Kom

Disusun Oleh:

Dinda Claudya – 20230801090

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS ESA UNGGUL TANGERANG**

# BUSINESS REQUIREMENT DOCUMENT (BRD)

## Aplikasi Manajemen Tugas Kuliah

### 1. Ringkasan Bisnis

**Nama Proyek:** Aplikasi Manajemen Tugas Kuliah

#### **Deskripsi Singkat:**

Aplikasi ini adalah sebuah platform berbasis web yang dirancang untuk memudahkan mahasiswa dalam mengelola seluruh tugas kuliah mereka. Tugas kuliah terdiri dari tugas teori harian yang sering diberikan setiap pertemuan dan tugas besar atau praktikum yang diberikan untuk jangka waktu tertentu. Dengan sistem ini, mahasiswa dapat memantau, mengedit, dan mengelola setiap tugas yang mereka miliki melalui satu aplikasi yang terpusat.

#### **Latar Belakang Bisnis:**

Tugas yang menumpuk tanpa manajemen yang baik sering kali berakibat pada keterlambatan pengumpulan. Proses pencatatan manual sering kali tidak efektif dan rawan hilang. Dengan adanya sistem berbasis web, mahasiswa dapat mengakses dan mengelola tugas kapan saja dan di mana saja, termasuk melalui perangkat mobile.

Selain itu, proyek ini menggunakan **Laravel boilerplate** untuk mempercepat proses pengembangan dengan struktur yang sudah rapi dan best practice Laravel, sehingga pengembangan menjadi lebih efisien dan mudah dipelihara. API yang dihasilkan akan terdokumentasi menggunakan **L5-Swagger**, memudahkan integrasi dengan aplikasi lain atau layanan mobile. Antarmuka aplikasi dibangun dengan **Bootstrap template**, sehingga tampilan UI responsif, modern, dan mudah digunakan.

#### A. Informasi Umum

Mahasiswa sering menerima berbagai tugas dari mata kuliah yang berbeda. Tugas ini ada yang berupa tugas teori harian dan ada juga praktikum besar. Selama ini banyak yang masih mencatat manual atau bahkan lupa, sehingga dibutuhkan sistem berbasis web yang mempermudah pengelolaan

## 2. Executive Summary

Aplikasi Manajemen Tugas Kuliah adalah sistem berbasis web yang dirancang untuk membantu mahasiswa mengelola tugas teori harian dan tugas besar/praktikum dengan lebih efektif. Dengan antarmuka yang responsif menggunakan Bootstrap, pengelolaan data tugas menjadi lebih sederhana. Penggunaan boilerplate Laravel mempercepat pengembangan sekaligus menjaga konsistensi struktur kode, sedangkan L5-Swagger digunakan untuk mendokumentasikan API yang memungkinkan integrasi dengan aplikasi lain di masa depan. Hasil akhirnya adalah aplikasi yang praktis, modern, dan siap dikembangkan lebih lanjut.

## 3. Objective Summary

Tujuan utama dari proyek ini adalah menyediakan platform tunggal yang:

- Mempermudah mahasiswa dalam mencatat, mengedit, menghapus, dan memantau tugas kuliah.
- Menyediakan dokumentasi API yang jelas melalui L5-Swagger untuk integrasi eksternal.
- Memberikan antarmuka pengguna yang intuitif, cepat, dan responsif menggunakan Bootstrap.
- Memanfaatkan Laravel boilerplate agar proses pengembangan efisien dan standar.

## 4. Goals

- Membangun sistem CRUD untuk Tasks (tugas teori) dan Projects (tugas besar).
- Menghadirkan admin panel dengan Filament untuk kemudahan manajemen.
- Menyediakan API lengkap dengan dokumentasi otomatis (L5-Swagger).
- Memberikan UI yang rapi, responsif, dan mudah digunakan (Bootstrap).
- Memastikan struktur kode rapi dan scalable melalui Laravel boilerplate.

## 5. Cara Mewujudkan

### **Teknologi yang digunakan:**

- Laravel 12 dengan boilerplate untuk kerangka kerja standar.
- Filament v3 sebagai admin panel CRUD.

- Bootstrap sebagai template antarmuka.
- L5-Swagger untuk dokumentasi API otomatis.
- MySQL untuk basis data.

#### **Langkah pengembangan:**

1. Inisialisasi proyek dengan boilerplate Laravel.
2. Rancang database untuk tabel tasks dan projects.
3. Buat CRUD lengkap melalui Filament dan uji fungsionalitas.
4. Integrasikan L5-Swagger untuk dokumentasi API.
5. Implementasikan UI berbasis Bootstrap yang responsif.
6. Lakukan pengujian dan perbaikan performa.

#### **6. Role-Based Access Control (RBAC)**

##### **Admin (mahasiswa):**

- Mengakses panel Filament.
- Mengelola data Tasks (tugas teori).
- Mengelola data Projects (tugas besar).
- Mengakses dokumentasi API.

*(Tahap ini hanya mendukung satu peran — admin. Di masa depan, RBAC dapat diperluas untuk multi-user seperti dosen atau asisten.)*

#### **7. Scope of Work**

##### **Fitur yang akan dikerjakan dalam lingkup proyek ini:**

- Desain database untuk tasks dan projects.
- Implementasi CRUD lengkap untuk Tasks dan Projects.
- Integrasi panel admin Filament untuk manajemen data.
- Pembuatan API endpoint untuk Tasks dan Projects.
- Dokumentasi API menggunakan L5-Swagger.
- Implementasi antarmuka Bootstrap responsif untuk tampilan frontend dasar.
- Uji coba fungsi, dokumentasi, dan performa.

## 8. Analisis Kebutuhan

### *A. Kebutuhan Fungsional*

- CRUD Tasks: Admin dapat menambah, mengedit, menghapus, dan melihat data tugas teori (Tasks).
- Admin dapat menandai status tugas selesai atau belum.
- CRUD Projects: Admin dapat menambah, mengedit, menghapus, dan melihat data tugas besar atau praktikum (Projects).
- Admin dapat menentukan deadline dan deskripsi untuk setiap project.
- Admin Panel Filament: Menyediakan antarmuka manajemen data yang ramah pengguna dan cepat diakses.
- API dengan Dokumentasi L5-Swagger: Semua endpoint Tasks dan Projects terdokumentasi otomatis serta memungkinkan integrasi dengan aplikasi atau layanan lain di masa depan.
- UI Responsif: Halaman frontend sederhana dengan Bootstrap, menampilkan daftar tugas dan proyek dengan tampilan modern.

### *B. Kebutuhan Non-Fungsional*

- Performa dan Skalabilitas: Sistem mampu menangani sejumlah data tugas dan proyek tanpa penurunan performa.
- Kemudahan Pemeliharaan: Dengan boilerplate Laravel, struktur kode rapi dan standar, memudahkan tim lain untuk memahami dan memelihara.
- Antarmuka Responsif: Menggunakan Bootstrap agar tampilan adaptif di berbagai ukuran layar seperti desktop, tablet, dan smartphone.
- Keamanan Dasar: Hanya admin yang dapat mengakses panel manajemen data, dan API hanya digunakan untuk integrasi yang sah.
- Ketersediaan Dokumentasi: Dokumentasi API selalu tersedia dan up to date melalui L5-Swagger.

## 9. Stakeholder

Stakeholder	Peran	Kepentingan
Mahasiswa (Admin)	Pengguna utama	Menggunakan aplikasi untuk mencatat, memantau, dan mengelola tugas kuliah dengan mudah.
Tim Pengembang (Developer)	Pengelola teknis	Mengembangkan, memelihara, dan menambahkan fitur baru ke dalam aplikasi.
Dosen/Pengelola Akademik (opsional di masa depan)	Pengamat	Dapat memantau progres mahasiswa jika fitur tambahan diimplementasikan di masa mendatang.

# SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

## Aplikasi Manajemen Tugas Kuliah

### 1. Pendahuluan

#### 1.1 Tujuan

Dokumen ini bertujuan untuk memberikan spesifikasi lengkap mengenai Aplikasi Manajemen Tugas Kuliah. Sistem ini membantu mahasiswa mengelola tugas teori harian dan tugas besar dengan lebih terstruktur melalui antarmuka berbasis web, API terdokumentasi, serta manajemen data yang mudah digunakan.

#### 1.2 Ruang Lingkup Sistem

Aplikasi Manajemen Tugas Kuliah dibangun menggunakan Laravel 12 dengan boilerplate untuk struktur yang konsisten. Sistem menyediakan fitur CRUD untuk entitas Tasks (tugas teori) dan Projects (tugas besar) melalui panel admin Filament. API terdokumentasi menggunakan L5-Swagger, dan antarmuka dirancang responsif menggunakan Bootstrap.

#### 1.3 Akronim dan Definisi

- **CRUD:** Create, Read, Update, Delete.
- **Tasks:** Tugas teori harian atau mingguan.

- **Projects:** Tugas besar atau praktikum yang memiliki deadline.
- **Bootstrap:** Framework CSS untuk membuat antarmuka web responsif.
- **L5-Swagger:** Paket Laravel untuk dokumentasi API otomatis.
- **Boilerplate Laravel:** Template dasar Laravel dengan struktur siap pakai.

## 2. Deskripsi Umum

### 2.1 Perspektif Produk

Produk ini adalah aplikasi mandiri berbasis web yang dirancang untuk digunakan secara internal oleh mahasiswa atau admin. Aplikasi dibangun di atas Laravel boilerplate dengan struktur MVC yang jelas. Panel admin menggunakan Filament untuk manajemen data, sedangkan API disediakan dan terdokumentasi melalui L5-Swagger.

### 2.2 Fungsi Utama

- Mengelola daftar Tasks (tugas teori) termasuk status selesai atau belum.
- Mengelola daftar Projects (tugas besar) termasuk deskripsi dan deadline.
- Menampilkan ringkasan data pada panel admin.
- Menyediakan API untuk integrasi eksternal dengan dokumentasi otomatis.
- Menyediakan antarmuka web berbasis Bootstrap yang mudah digunakan.

### 2.3 Karakteristik Pengguna

Pengguna utama adalah mahasiswa yang bertindak sebagai admin. Mereka memiliki akses penuh untuk menambah, mengedit, menghapus, dan melihat data tugas. Pengguna diharapkan terbiasa menggunakan aplikasi web dasar.

## 12. Kebutuhan Fungsional

- Admin dapat menambah, mengedit, menghapus, dan melihat data Tasks.
- Admin dapat menambah, mengedit, menghapus, dan melihat data Projects.
- Panel admin menampilkan daftar tugas dan proyek dengan fitur pencarian dan filter dasar.
- API menyediakan endpoint CRUD untuk Tasks dan Projects, lengkap dengan dokumentasi L5-Swagger.

### 13. Kebutuhan Non-Fungsional

- Sistem responsif di berbagai perangkat berkat Bootstrap.
- Struktur kode rapi, mudah dirawat, dan dapat dikembangkan melalui boilerplate Laravel.
- Kinerja cepat dalam memuat dan memproses data.
- Keamanan dasar dengan akses admin terbatas.
- Dokumentasi API selalu tersedia dan up-to-date melalui L5-Swagger.

### 14. Desain Basis Data

**Tabel: tasks**

Kolom	Tipe Data	Keterangan
id	bigint (auto increment)	Primary key
title	varchar	Judul tugas teori
description	text	Deskripsi tugas teori
is_done	boolean	Status selesai atau belum
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

**Tabel: projects**

Kolom	Tipe Data	Keterangan
id	bigint (auto increment)	Primary key
name	varchar	Nama project atau praktikum
description	text	Deskripsi project
deadline	date	Batas waktu penyelesaian
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui



## 15. Arsitektur Sistem

Aplikasi menggunakan arsitektur **Model-View-Controller (MVC)**:

- **Model**: Task dan Project berinteraksi dengan database.
- **View**: Blade template dengan Bootstrap untuk menampilkan data.
- **Controller**: Mengatur alur data dari model ke view dan sebaliknya.
- **Filament**: Digunakan sebagai admin panel untuk CRUD data.
- **L5-Swagger**: Digunakan untuk mendokumentasikan semua endpoint API.

Server menggunakan **Nginx** dengan **PHP 8.2** di atas sistem operasi Ubuntu. Database menggunakan **MySQL**.

## 16. Pengembangan Masa Depan

- Penambahan fitur autentikasi multi-user dengan role berbeda (misalnya dosen dan asisten).
- Penambahan notifikasi otomatis melalui email atau WhatsApp untuk deadline yang mendekat.
- Fitur export/import data tugas dalam format Excel atau PDF.
- Integrasi dengan aplikasi mobile untuk akses lebih luas.