

KANBAN APP

PEMROGRAMAN WEB LANJUT



Oleh:

Nurul Qalbi Zahrani F1D022150
Umam Alparizi F1D02310141
M. Bayu Aji F1D02310144

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MATARAM
2025

DAFTAR ISI

BAB I DESKRIPSI PROJECT	4
A. Latar Belakang.....	4
B. Tujuan <i>Project</i>	4
C. Teknologi yang Digunakan (<i>Backend, Frontend, Database</i>).....	5
BAB II PERANCANGAN SISTEM	6
A. Deskripsi Singkat Sistem	6
B. Desain Database (ERD dan penjelasan singkat tabel)	6
BAB III IMPLEMENTASI	9
A. Implementasi Backend (struktur folder, API, middleware)	9
B. Implementasi Frontend (halaman, routing, konsumsi API)	13
BAB IV PENGUJIAN	18
A. Pengujian API (Postman/Thunder Client).....	18
B. Pengujian Aplikasi (Screenshot fitur utama)	18
LAMPIRAN	19

DAFTAR GAMBAR

Gambar 2.1 ER Diagram Kanban App.....	6
Gambar 2.2 Tabel Kanban App.....	7
Gambar 2.3 Deskripsi Tabel Boards	7
Gambar 2.4 Deskripsi Tabel Cards	7
Gambar 2.5 Deskripsi Tabel Lists	8
Gambar 2.6 Deskripsi Tabel Users.....	8
Gambar 3.1 Struktur Folder	9
Gambar 3. 2 Struktur Folder Frontend.....	13

BAB I

DESKRIPSI PROJECT

A. Latar Belakang

Penggunaan gadget yang semakin intens, khususnya untuk game dan media sosial, telah menyebabkan kecanduan gadget di kalangan mahasiswa. Kondisi ini membuat mahasiswa mudah terdistraksi dan kesulitan mengatur waktu, sehingga banyak tugas perkuliahan yang tertunda dan menumpuk.

Ketidakteraturan dalam mengelola tugas menyebabkan mahasiswa sering kebingungan menentukan tugas yang sudah dikerjakan, sedang dikerjakan, maupun yang memiliki deadline terdekat. Di sisi lain, aktivitas belajar yang membutuhkan fokus tinggi terasa kurang menarik dibandingkan screen time hiburan, sehingga fokus belajar sulit dipertahankan.

Oleh karena itu, dikembangkan website "Kanban App" gabungan Pomodoro dan To-Do List berbasis Kanban sebagai solusi untuk membantu mahasiswa mengatur waktu dan tugas secara lebih terorganisir, nyaman, mudah, dan menarik, sehingga dapat meningkatkan fokus serta produktivitas belajar. Pomodoro adalah teknik manajemen waktu untuk meningkatkan fokus dan produktivitas dengan membagi pekerjaan menjadi interval 25 menit kerja intensif, diikuti istirahat singkat 5 menit, dan setelah empat pomodoro, istirahat lebih panjang (15-30 menit). Sedangkan, Kanban adalah sistem visualisasi alur kerja yang berasal dari Jepang (artinya "kartu tanda") yang digunakan untuk mengelola pekerjaan secara efisien, memvisualisasikan tugas melalui papan dan kartu, serta meningkatkan aliran dan produktivitas tim dengan membatasi pekerjaan yang sedang berjalan (WIP).

B. Tujuan Project

Berikut ini merupakan beberapa tujuan project yang ingin dicapai:

1. Memberikan solusi atas permasalahan kecanduan gadget yang berdampak pada rendahnya fokus dan manajemen waktu mahasiswa.
2. Mengembangkan sebuah website produktivitas yang mengintegrasikan metode Pomodoro dan Kanban untuk membantu pengelolaan waktu dan tugas secara terstruktur.
3. Membantu mahasiswa dalam mengatur, memantau, dan memprioritaskan tugas agar tidak terjadi penumpukan dan keterlambatan deadline.

4. Meningkatkan fokus belajar dengan menciptakan lingkungan kerja yang lebih menarik dibandingkan distraksi dari game dan media sosial.
5. Menyediakan sistem pencatatan tugas yang jelas sehingga pengguna dapat mengetahui tugas yang sudah selesai, sedang dikerjakan, dan yang harus segera diselesaikan.
6. Meningkatkan produktivitas dan kenyamanan pengguna dalam menyelesaikan tugas akademik secara efektif dan efisien.

C. Teknologi yang Digunakan (*Backend, Frontend, Database*)

Berikut ini merupakan teknologi yang kami gunakan dalam *website*, mulai dari *backend*, *frontend*, hingga *database*.

1. *Backend* yang digunakan adalah javascript Modul ECMAScript. Penggunaan ESM memungkinkan struktur kode yang lebih modern dibandingkan standar CommonJS lama. Framework utama yang digunakan adalah Express.js untuk menangani routing API secara efisien.
2. *Frontend* yang digunakan Vue.js. Proyek ini memanfaatkan Vue Router untuk manajemen navigasi halaman (seperti *Dashboard*, *Board View*, dan *Pomodoro*) tanpa melakukan muat ulang halaman secara penuh (*Single Page Application*).
3. Untuk berinteraksi dengan backend, frontend menggunakan library Axios. Terdapat mekanisme *interceptor* yang secara otomatis menyisipkan token JWT (JSON Web Token) ke dalam header setiap permintaan untuk memastikan keamanan akses data pengguna.
4. *Database* yang digunakan MySQL yang berguna dalam *database* yang membutuhkan relasi. *Database* dirancang dengan relasi *Foreign Key* yang memastikan integritas data, misalnya dengan fitur ON DELETE CASCADE, di mana jika sebuah papan dihapus, maka seluruh daftar dan kartu yang ada di dalamnya akan ikut terhapus secara otomatis.

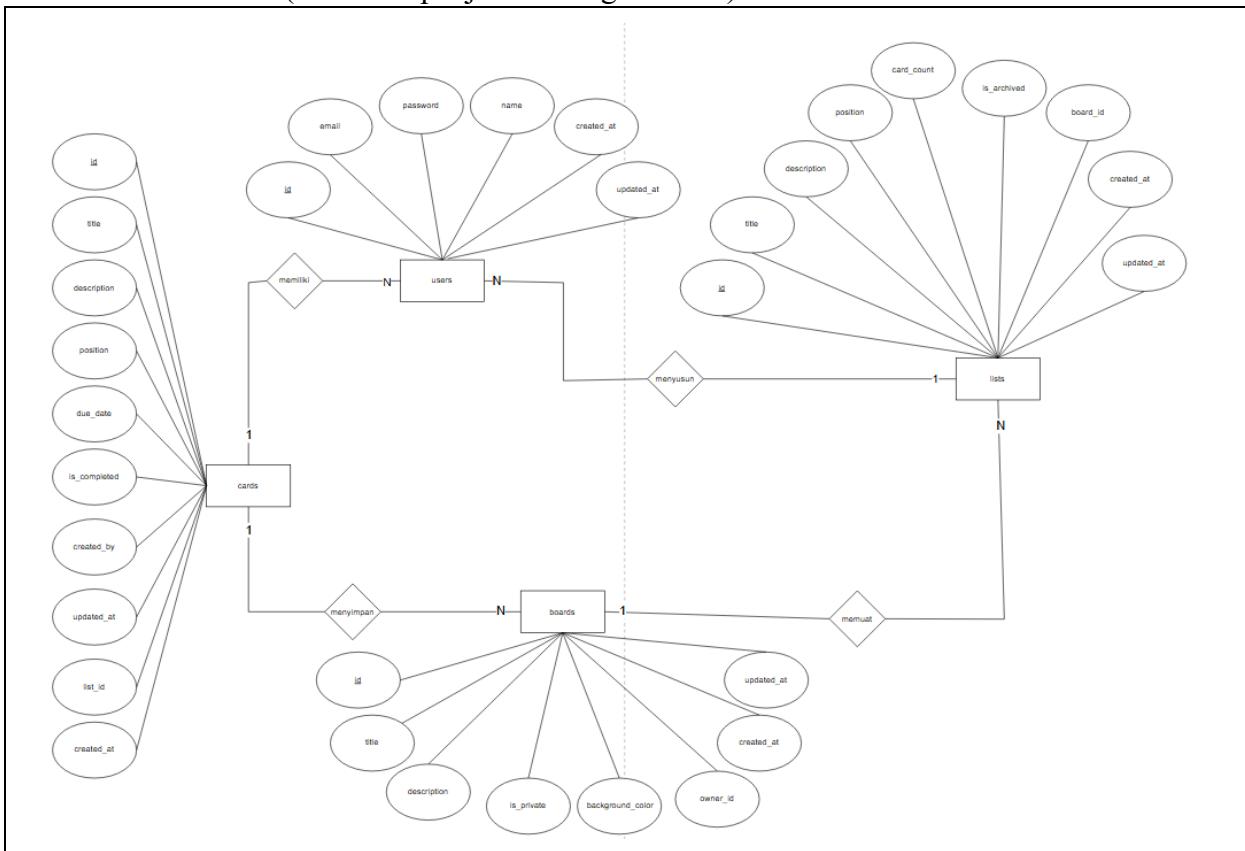
BAB II

PERANCANGAN SISTEM

A. Deskripsi Singkat Sistem

"Kanban App" adalah aplikasi manajemen tugas berbasis web yang mengintegrasikan metode Kanban untuk visualisasi alur kerja dan teknik Pomodoro untuk meningkatkan fokus belajar mahasiswa. Sistem ini memungkinkan pengguna untuk membuat papan kerja (boards), daftar kategori tugas (lists), dan kartu tugas individual (cards) yang dapat dipantau status perkembangannya secara terorganisir. Aplikasi ini dilengkapi dengan sistem autentikasi keamanan untuk melindungi data pribadi pengguna.

B. Desain Database (ERD dan penjelasan singkat tabel)



Gambar 2.1 ER Diagram Kanban App

Berdasarkan Gambar 2.1 didapatkan informasi ER Diagram Kanban App yang terdiri dari 4 entitas, yaitu *users*, *cards*, *lists*, dan *boards* beserta atributnya masing-masing. Terdapat entitas *users* yang digunakan untuk menyimpan informasi users, seperti nama pengguna, email yang dapat dihubungi, dan password untuk masuk ke sistem, serta informasi tanggal akun dibuat dan diperbarui. Entitas *users* memiliki relasi dengan *cards* dan *lists*. *Cards*

```

MariaDB [(none)]> use db_kanban_app
Database changed
MariaDB [db_kanban_app]> show tables;
+-----+
| Tables_in_db_kanban_app |
+-----+
| boards
| cards
| lists
| users
+-----+
4 rows in set (0.001 sec)

```

Gambar 2.2 Tabel Kanban App

Berdasarkan Gambar 2.1 didapatkan informasi berupa desain *database* yang digunakan untuk Kanban App. Database ini memiliki tabel-tabel yang sesuai dengan Entity Relationship Diagram yang telah dibuat sebelumnya, yaitu terdiri dari entitas atau tabel boards, cards, lists, dan users. *Database* ini dinamakan db_kanban_app.

```

MariaDB [db_kanban_app]> desc boards;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| title | varchar(255) | NO | | NULL |
| description | text | YES | | NULL |
| background_color | varchar(255) | NO | | #0079bf |
| is_private | tinyint(1) | NO | | 0 |
| owner_id | int(11) | NO | MUL | NULL |
| created_at | datetime | NO | | NULL |
| updated_at | datetime | NO | | NULL |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.012 sec)

```

Gambar 2.3 Deskripsi Tabel Boards

Berdasarkan Gambar 2.3 didapatkan informasi berupa deskripsi terkait tabel boards. Tabel boards ini berisi 8 kolom yang sesuai dengan ERD yang dibuat sebelumnya. Tabel boards ini memiliki relasi dengan tabel cards dan lists.

```

MariaDB [db_kanban_app]> desc cards;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| title | varchar(255) | NO | | NULL |
| description | text | YES | | NULL |
| position | int(11) | NO | | 0 |
| due_date | datetime | YES | | NULL |
| is_completed | tinyint(1) | NO | | 0 |
| list_id | int(11) | NO | MUL | NULL |
| created_by | int(11) | YES | MUL | NULL |
| created_at | datetime | NO | | NULL |
| updated_at | datetime | NO | | NULL |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.013 sec)

```

Gambar 2.4 Deskripsi Tabel Cards

Berdasarkan Gambar 2.4 didapatkan informasi berupa deskripsi terkait tabel cards. Tabel cards ini berisi 10 kolom yang sesuai dengan ERD yang dibuat sebelumnya. Tabel boards ini memiliki relasi dengan tabel users dan boards.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
title	varchar(255)	NO		NULL	
description	text	YES		NULL	
position	int(11)	NO		0	
card_count	int(11)	NO		0	
is_archived	tinyint(1)	NO		0	
board_id	int(11)	NO	MUL	NULL	
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	

Gambar 2.5 Deskripsi Tabel Lists

Berdasarkan Gambar 2.5 didapatkan informasi berupa deskripsi terkait tabel cards. Tabel cards ini berisi 10 kolom yang sesuai dengan ERD yang dibuat sebelumnya. Tabel boards ini memiliki relasi dengan tabel users dan boards.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
email	varchar(255)	NO	UNI	NULL	
password	varchar(255)	NO		NULL	
name	varchar(255)	NO		NULL	
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	

Gambar 2.6 Deskripsi Tabel Users

Berdasarkan Gambar 2.6 didapatkan informasi berupa deskripsi tabel users. Tabel users memiliki 6 kolom yang menyimpan informasi seputar pengguna, seperti *email*, *password*, dan nama dari pengguna. *Primary key* dari tabel ini adalah id dengan *auto increment*.

BAB III

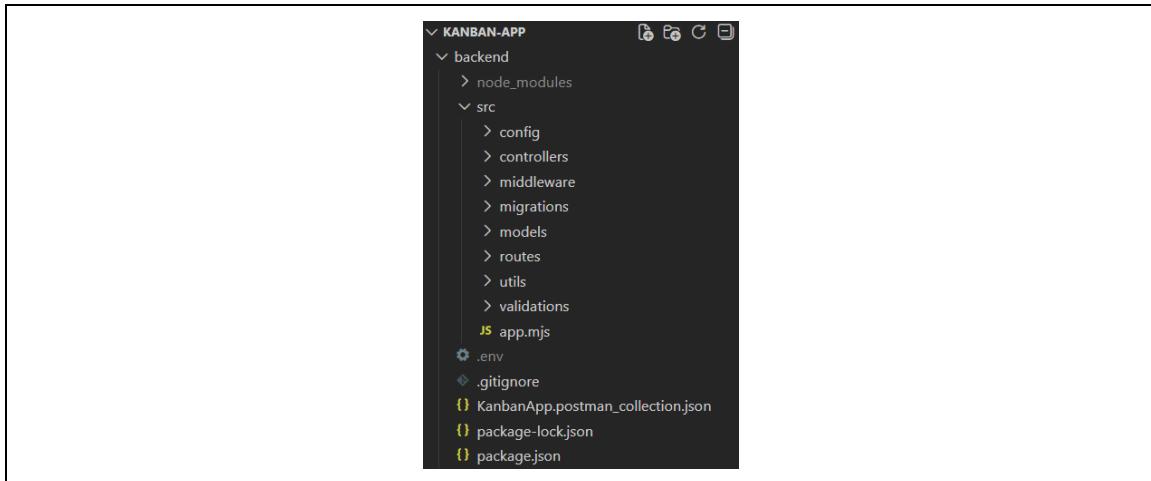
IMPLEMENTASI

A. Implementasi Backend (struktur folder, API, middleware)

Arsitektur sistem menerapkan pola desain *Model-Controller-Route* (MCR) yang memisahkan definisi data, logika bisnis, dan routing untuk menjaga kode tetap modular dan mudah dipelihara. Interaksi dengan basis data MySQL dikelola menggunakan Sequelize ORM, yang memungkinkan manipulasi data dilakukan melalui objek JavaScript.

1. Struktur Folder

Kode sumber *backend* diorganisir secara hierarkis sesuai kebutuhan di dalam direktori `src`. Struktur ini dirancang untuk memisahkan tanggung jawab setiap komponen sistem. Berikut tampilan struktur hierarkies proyek *backend* untuk proyek kanban dalam VS Code Explorer:



Gambar 3.1 Struktur Folder

Berikut adalah rincian struktur folder beserta fungsinya seperti yang terlihat pada **Gambar 3.1**:

- `src/app.mjs`

File ini merupakan titik masuk utama aplikasi (entry point). Berkas ini menginisialisasi server Express, mengatur konfigurasi CORS, parsing JSON, dan memuat seluruh rute utama.

- `src/config/`

Direktori ini menyimpan konfigurasi koneksi basis data (database.mjs) yang memuat kredensial host, pengguna, dan nama basis data yang diambil dari variabel lingkungan.

- `src/controllers/`

Folder Controllers berisi logika bisnis utama. Setiap berkas (seperti authController.mjs, boardController.mjs) menangani permintaan dari klien, memproses data melalui Model, dan mengirimkan respons balik.

- `src/models/`

Direktori ini mendefinisikan skema tabel basis data dan relasi antar tabel (asosiasi). Berkas index.mjs di dalamnya berfungsi untuk menginisialisasi Sequelize dan menyatukan seluruh model.

- `src/routes/`

Direktori ini menangani definisi endpoint URL dan metode HTTP (GET, POST, PUT, DELETE). Setiap rute dipetakan ke fungsi pengendali (controller) yang sesuai.

- `src/middleware/`

Direktori Middleware menyimpan fungsi perantara (interceptors) yang dijalankan sebelum permintaan mencapai controller, seperti verifikasi login dan penanganan error global.

- `src/validations/`

Direktori ini berisi aturan validasi data masukan menggunakan express-validator untuk memastikan integritas data sebelum diproses.

- `src/migrations/`

Direktori ini digunakan untuk menyimpan skrip SQL atau file migrasi untuk manajemen versi skema basis data.

2. API

Aplikasi menyediakan antarmuka RESTful API yang terstruktur berdasarkan entitas sumber daya. Setiap endpoint dirancang untuk menerima input dalam format JSON dan mengembalikan respons JSON yang standar. Berikut adalah rincian endpoint yang diimplementasikan:

a. Modul Autentikasi (Auth)

Modul ini menangani identifikasi dan akses pengguna menggunakan JSON Web Token (JWT).

1. POST /auth/register: Mendaftarkan pengguna baru dengan memvalidasi nama, email, dan enkripsi kata sandi.
 2. POST /auth/login: Memverifikasi kredensial pengguna dan menerbitkan token JWT jika valid.
 3. GET /auth/me: Mengambil profil data pengguna yang sedang login berdasarkan token autentikasi.
- b. Modul Papan (Boards)
- Modul ini mengelola wadah utama proyek atau papan kerja Kanban.
1. GET /boards: Menampilkan seluruh papan yang dimiliki oleh pengguna.
 2. GET /boards/public: Menampilkan papan yang diatur dengan visibilitas publik.
 3. GET /boards/:id: Mengambil detail lengkap satu papan beserta daftar (lists) dan kartu (cards) di dalamnya.
 4. POST /boards: Membuat papan baru dengan judul dan deskripsi.
 5. PUT /boards/:id: Memperbarui informasi papan.
 6. DELETE /boards/:id: Menghapus papan beserta seluruh kontennya secara permanen.
- c. Modul Daftar (Lists)
- Modul ini mengelola kolom status dalam papan Kanban. Modul ini memiliki logika khusus untuk pengurutan posisi.
1. GET /lists/:boardId: Mengambil semua daftar yang terkait dengan ID papan tertentu.
 2. POST /lists: Menambahkan daftar baru ke dalam papan.
 3. PUT /lists/:id: Memperbarui judul daftar atau mengubah posisi urutan (reordering) daftar dalam papan.
 4. DELETE /lists/:id: Menghapus daftar dan semua kartu yang ada di dalamnya.
- d. Modul Kartu (Cards)
- Modul terakhir yaitu Cards mengelola unit tugas terkecil. Modul ini adalah yang paling kompleks karena menangani pemindahan antar daftar dan status penyelesaian.
1. GET /cards/:listId: Mengambil semua kartu dalam daftar tertentu.
 2. POST /cards: Membuat kartu baru dengan atribut judul, deskripsi, dan tenggat waktu.

3. PUT /cards/:id: Memperbarui detail informasi kartu.
4. PATCH /cards/:id/completion: Mengubah status kartu (toggle) antara selesai atau belum selesai.
5. PUT /cards/:id/move: Menangani logika pemindahan kartu, baik pemindahan posisi dalam satu daftar (sorting) maupun pemindahan antar daftar yang berbeda.
6. DELETE /cards/:id: Menghapus kartu.

3. Middleware

Middleware digunakan untuk meningkatkan keamanan dan stabilitas aplikasi dengan menangani tugas-tugas yang bersifat lintas sektoral (cross-cutting concerns).

Middleware Autentikasi (auth.mjs)

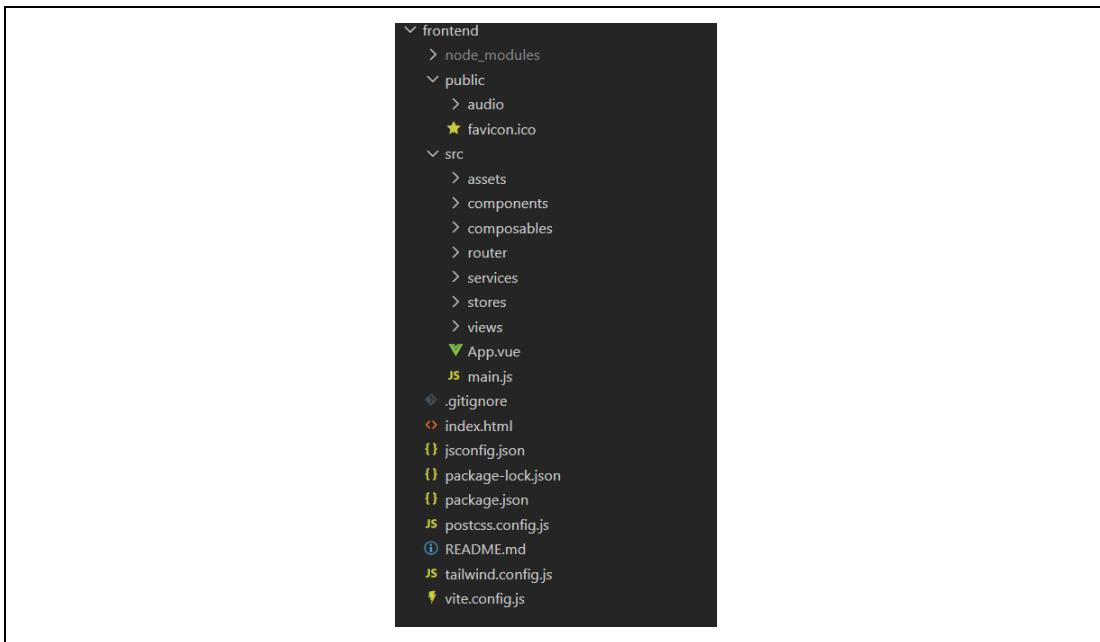
Middleware ini bertugas melindungi rute privat. Logika yang diterapkan adalah memeriksa header Authorization pada setiap permintaan masuk. Sistem akan mengekstrak token Bearer dan memverifikasinya menggunakan kunci rahasia (secret key). Jika token valid, informasi pengguna (seperti ID dan email) akan disisipkan ke dalam objek req.user agar dapat digunakan oleh controller. Jika token tidak valid atau kadaluwarsa, middleware akan langsung memutus permintaan dan mengembalikan respons 401 Unauthorized.

Middleware Penanganan Kesalahan (errorHandler.mjs)

Untuk menjaga konsistensi respons error dan mencegah aplikasi berhenti mendadak (crash), diterapkan middleware penanganan kesalahan terpusat. Middleware ini menangkap setiap *exception* yang terjadi di controller (menggunakan blok try-catch). Berdasarkan jenis error yang ditangkap (misalnya error validasi Sequelize, error 404, atau error internal server), middleware ini akan memformat pesan kesalahan menjadi objek JSON yang standar sebelum dikirimkan ke klien.

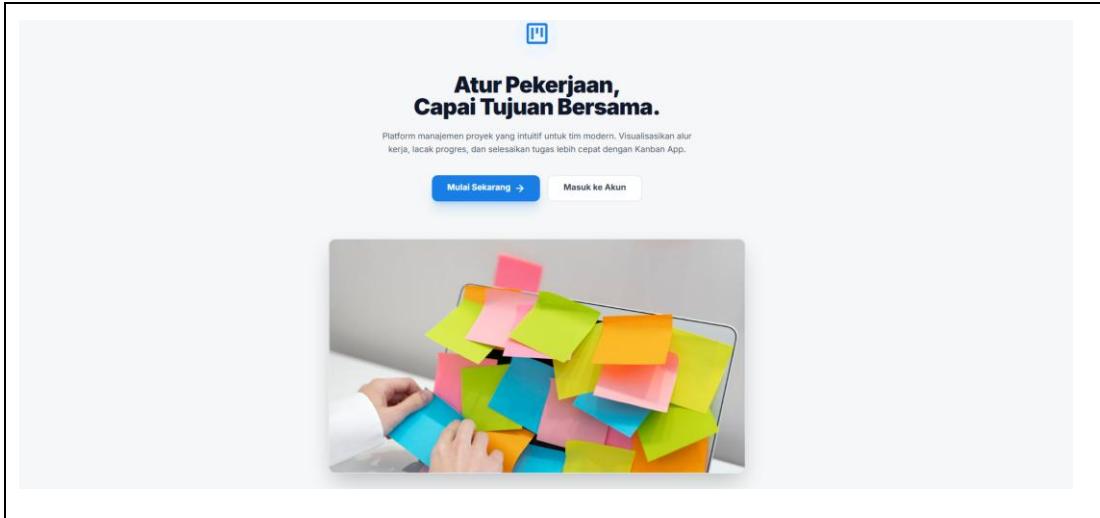
B. Implementasi Frontend (halaman, routing, konsumsi API)

1. Struktur Folder Frontend



Gambar 3.2 Struktur Folder Frontend

2. Screenshot Implementasi *Interface*



Gambar 3.3 Tampilan Landing Page

Kanban App

Buat Akun Baru

Mulai kelola proyek Anda dengan mudah.

Nama
Nurul Qalbi Zahrani

Email
nurulzahrani151@sma.belajar.id

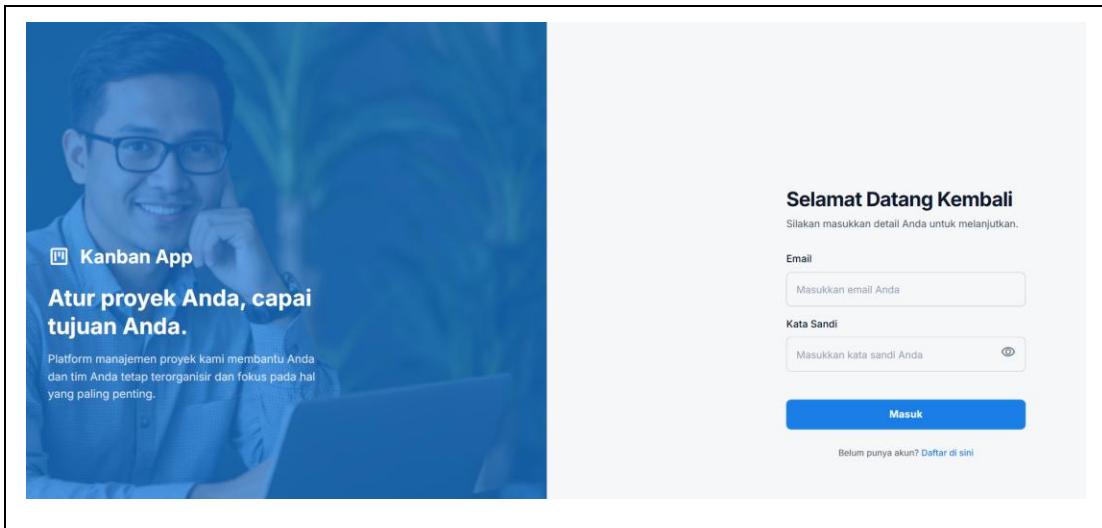
Kata Sandi
151zahrani

Konfirmasi Kata Sandi
151zahrani

Daftar

Sudah punya akun? [Masuk di sini](#)

Gambar 3.4 Halaman Registrasi Akun



Gambar 3.5 Halaman Login Website

Nurul Qalbi Zahrani
nurulzahrani151@sma.belajar.id

Dashboard

Selamat datang kembali, Nurul Qalbi Zahrani!

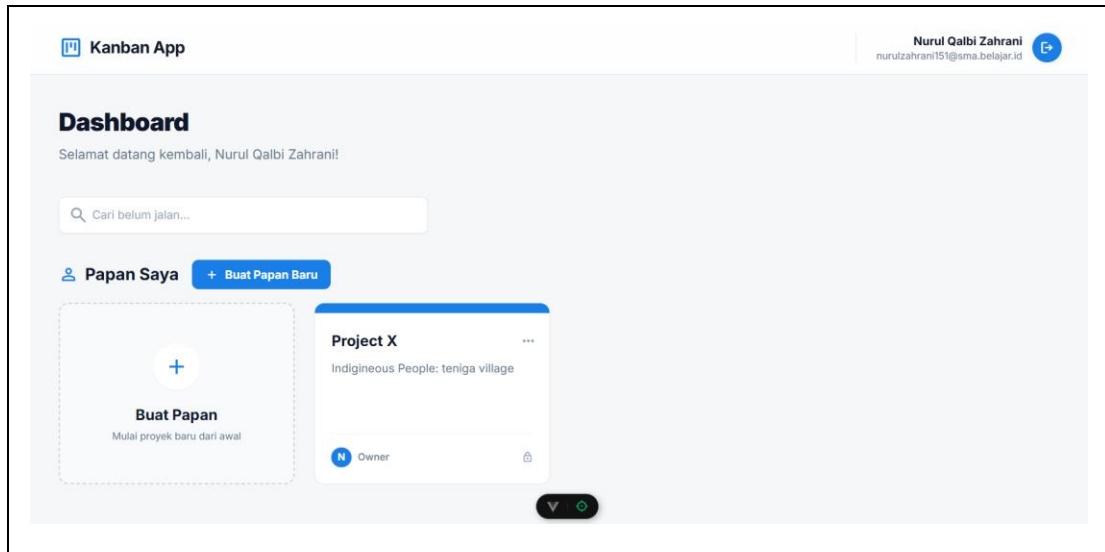
Cari belum jalan...

Papan Saya + Buat Papan Baru

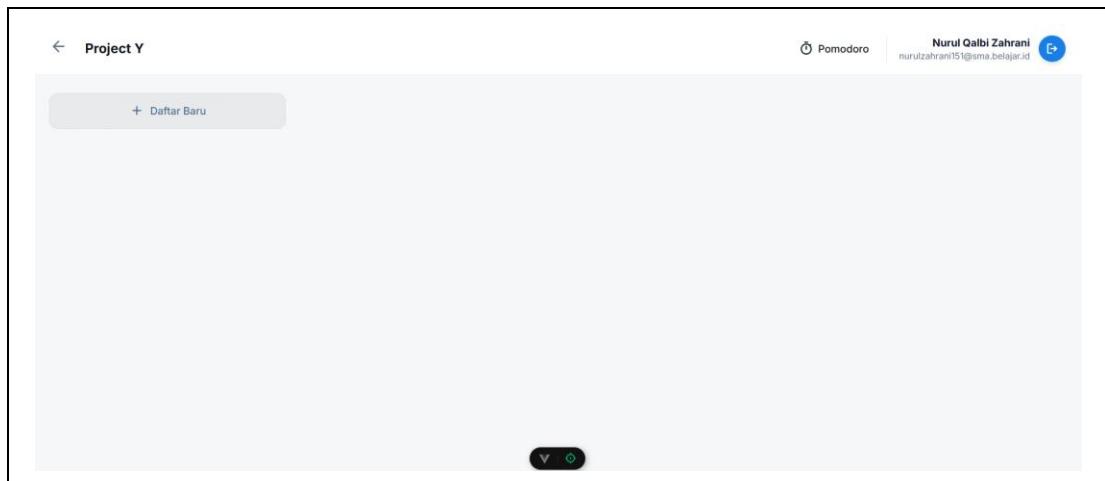
Anda belum memiliki papan proyek.

Buat Papan Pertama

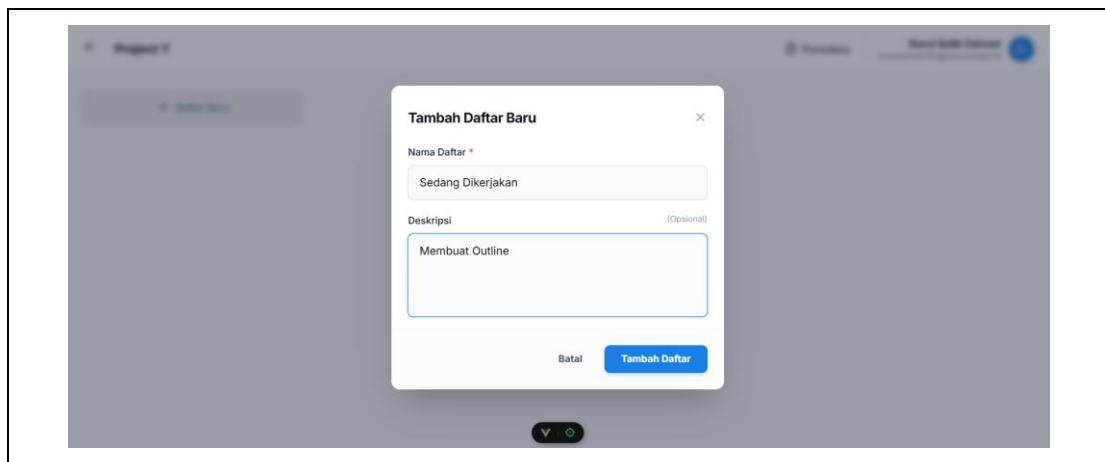
Gambar 3.6 Dashboard Sistem



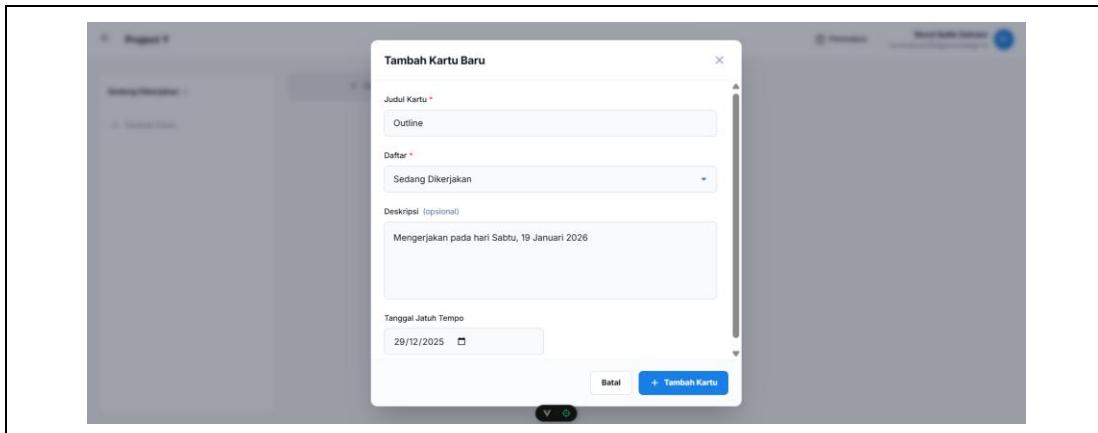
Gambar 3.7 Halaman Dashboard Tambahkan Papan



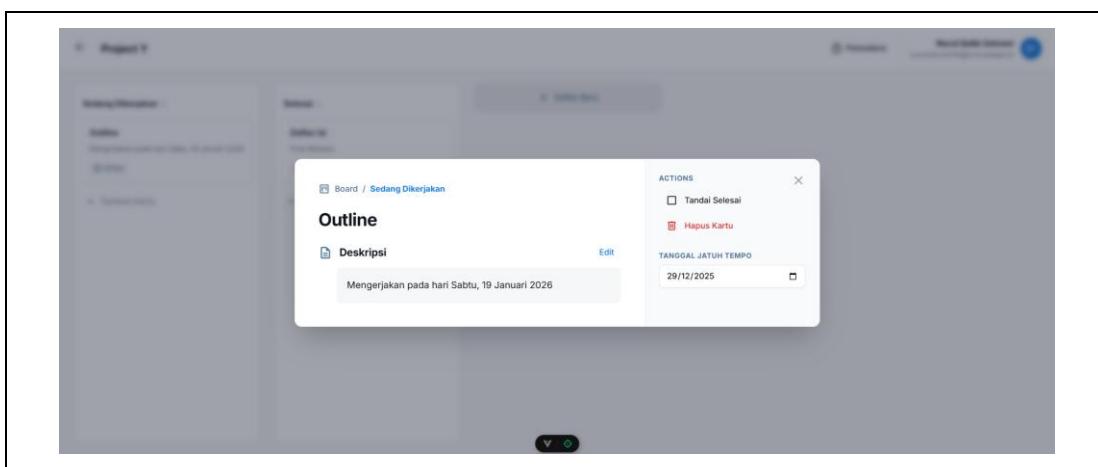
Gambar 3.8 Halaman Utama Project



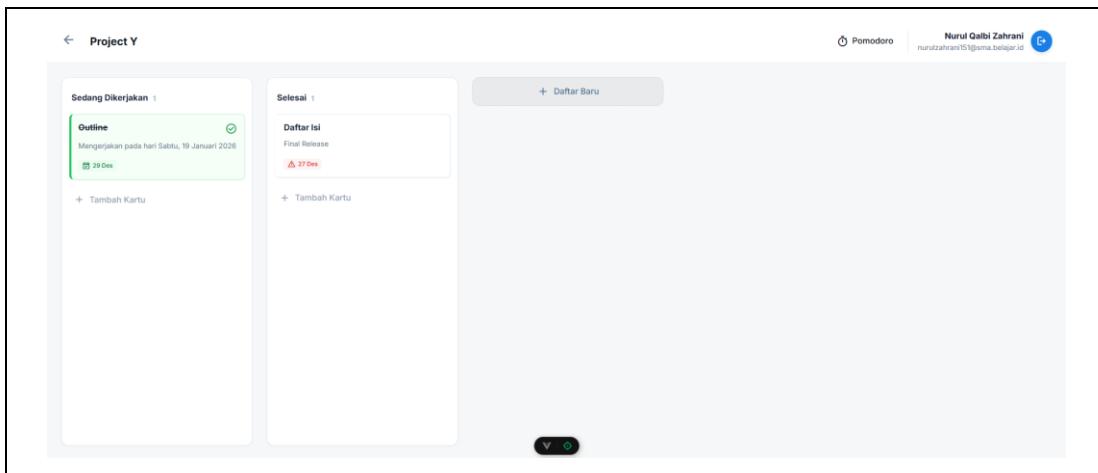
Gambar 3.9 Halaman Tambahkan Daftar/Lists Baru



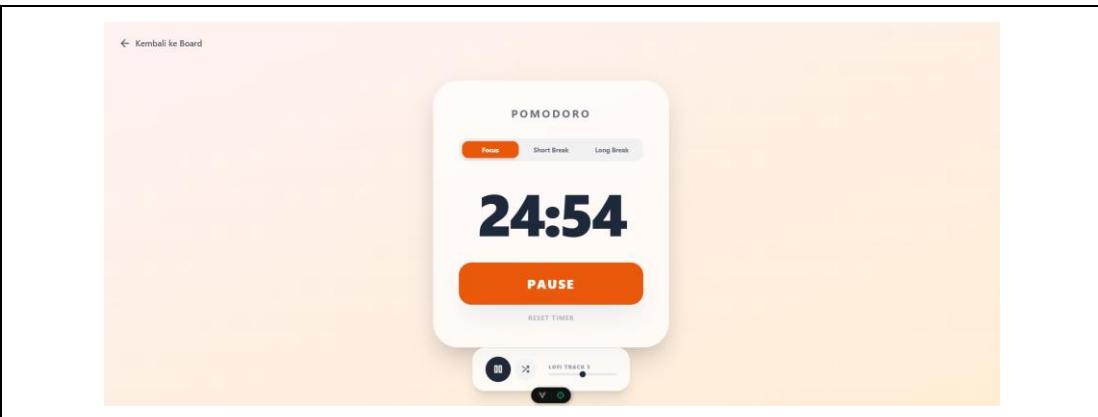
Gambar 3.10 Menambahkan Kartu Baru



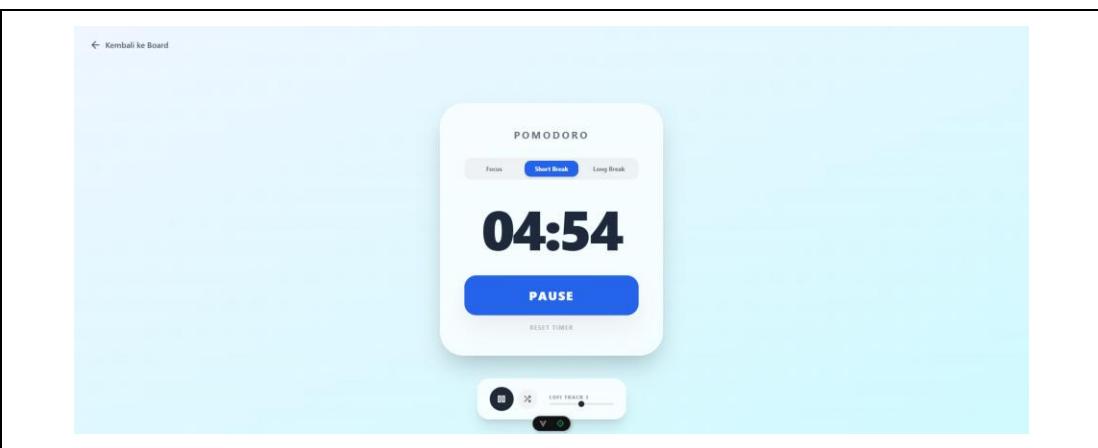
Gambar 3.11 Mengubah Status Board



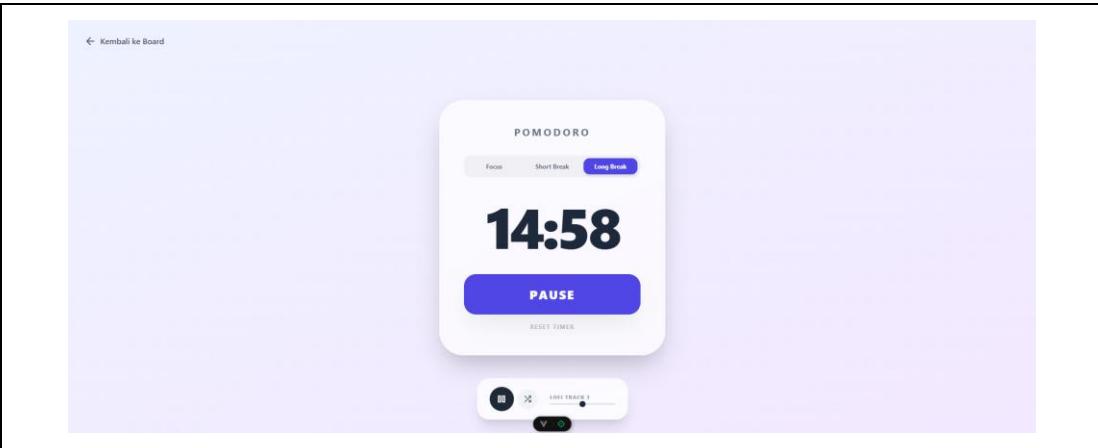
Gambar 3.12 Tampilan Project Selesai dan Melewati Tenggat Waktu



Gambar 3.13 Halaman Pomodoro untuk Fokus Time



Gambar 3.14 Halaman Pomodoro untuk Short Break



Gambar 3.15 Halaman Pomodoro untuk Long Break

3. Frontend dibangun dengan menggunakan komponen Vue.js, seperti login, register, dashboard, pomodoro, dan kanban list. Untuk proses routing, project ini memanfaatkan vue-router sebagai navigator. Sedangkan, untuk API dikonsumsi menggunakan Axios dengan cara

BAB IV

PENGUJIAN

A. Pengujian API (Postman/Thunder Client)

Pengujian dilakukan terhadap endpoint utama yang didefinisikan dalam KanbanApp.postman_collection.json, meliputi:

- Auth: Registrasi pengguna baru dan login untuk mendapatkan token.
- Boards: Pengujian CRUD (Create, Read, Update, Delete) pada papan proyek.
- Lists & Cards: Pengujian penambahan tugas baru dan pemindahan tugas antar list.

B. Pengujian Aplikasi (Screenshot fitur utama)

- Sistem autentikasi (Login/Logout).
- Pembuatan dan manajemen Papan Kanban.
- Fitur timer Pomodoro untuk membantu manajemen waktu fokus pengguna.
- Proteksi halaman agar data pribadi hanya bisa diakses oleh pemilik akun yang sah.

LAMPIRAN

Link Github Repository

<https://github.com/gper00/kanban-app>

Link Deployment (jika ada)

Screenshot Tambahan