

Laporan Final Project: Forum & Blog Platform

Deskripsi

Aplikasi web full-stack untuk berbagi artikel, diskusi, dan kolaborasi dengan sistem autentikasi yang aman.

Setup & Instalasi Prerequisites

- Node.js (v18 atau lebih tinggi)
- npm atau yarn
- Git

Instalasi Backend

```
# Navigate ke folder backend
cd backend

# Install dependencies
npm install

# Setup environment variables
# Copy .env.example ke .env dan sesuaikan konfigurasi

# Generate Prisma Client
npx prisma generate

# Jalankan migration database
npx prisma migrate dev

# (Optional) Seed database dengan data awal
npm run prisma:seed

# Jalankan development server
npm run start:dev
```

Backend akan berjalan di `http://localhost:5000`

Instalasi Frontend

```
# Navigate ke folder frontend
```

```
cd frontend
```

```
# Install dependencies
```

```
npm install
```

```
# Jalankan development server
```

```
npm run dev
```

Frontend akan berjalan di `http://localhost:3000`

Testing

```
# Backend testing
```

```
cd backend
```

```
# Unit tests
```

```
npm test
```

```
# E2E tests (menggunakan database terpisah test.db)
```

```
npm run test:e2e
```

```
# Coverage report
```

```
npm run test:cov
```

Database Schema

Menggunakan Prisma dengan SQLite database yang memiliki 3 model utama:

- User: Menyimpan informasi user, authentication, dan relasi ke posts dan replies
- Post: Menyimpan artikel/postingan dengan file upload support
- Reply: Menyimpan komentar/balasan pada posts dengan tracking author

Fitur proyek

1. Sistem autentikasi & keamanan

Aplikasi ini pakai sistem login yang aman dengan JWT (JSON Web Tokens). Password user di-hash pakai bcrypt biar aman. Setelah login, sistem kasih 2 token: access token yang expire dalam 15 menit, sama refresh token yang tahan sampai 7 hari. Jadi user gak perlu login terus-terusan. Semua endpoint yang penting dilindungi JWT Guard yang cek token di setiap request, jadi cuma user yang udah login yang bisa akses fitur-fitur tertentu.

2. Manajemen posts dengan file upload

Ini fitur utama aplikasi - user bisa bikin, baca, edit, dan hapus post mereka sendiri. Tiap post bisa isi judul, konten, plus file (gambar atau dokumen). File upload pakai Multer yang otomatis kasih nama unik buat setiap file. Di frontend, gambar langsung keliatan preview-nya, kalau file lain ada tombol download. Post bisa disimpan sebagai draft dulu atau langsung dipublish. Ada pagination buat loading post yang banyak, plus fitur search buat cari post berdasarkan judul atau isi.

3. Reply system dengan author tracking

Fitur baru yang keren - user bisa komen di post, dan sistem otomatis track siapa yang komen. Tampilannya bagus dengan avatar (inisial nama), nama lengkap, dan waktu komen lengkap. Yang paling berguna adalah user bisa delete reply mereka sendiri kalau salah ketik atau mau narik kembali komennya. Tombol delete cuma muncul di komen sendiri dan ada konfirmasi sebelum hapus. Backend juga ngecek authorization biar cuma yang punya komen yang bisa delete, jadi aman dari user lain.

4. Database terpisah untuk testing

Biar data development gak kehapus waktu testing, sistemnya pakai 2 database terpisah: `dev.db` buat development dan `test.db` buat testing. Soalnya test suka hapus semua data sebelum dan sesudah jalan. Ada safety check di test file yang otomatis stop kalau gak pakai database test, jadi data development aman. Setup ini bikin testing lebih tenang tanpa worry data hilang.

5. Email notification system

Pakai Nodemailer buat kirim email otomatis. Welcome email langsung dikirim pas user baru daftar, terus ada email konfirmasi juga pas bikin post baru. Email-nya jalan async jadi gak ngeblok aplikasi dan tetap cepat. Konfigurasi SMTP bisa gampang diganti

lewat environment variables, jadi bisa pakai Mailtrap buat development atau SendGrid/AWS SES buat production.

6. Type-safe development dengan TypeScript & Prisma

Semua kode pakai TypeScript biar type-safe dan ngurangin error. Prisma sebagai ORM otomatis generate TypeScript types dari database schema, jadi query database jadi lebih aman. IDE kasih autocomplete dan error checking waktu coding. DTO validation pakai class-validator buat validasi input di setiap endpoint. Type safety ini bikin maintenance jangka panjang jadi lebih mudah.

7. Responsive & Modern UI design

UI dibangun pakai Tailwind CSS dengan design modern yang responsive di semua device - dari HP sampai desktop. Component-based architecture pakai React bikin code gampang di-reuse. Protected Route otomatis redirect ke login kalau belum login. Form handling pakai controlled components dengan state management yang proper. Ada loading states, error handling, hover effects, dan transitions yang bikin UX lebih smooth. Color scheme pakai blue sebagai warna utama dengan gray tones buat hierarchy.

8. Error handling & Data validation

Error handling di-implement di semua layer. Backend pakai Prisma error handler yang transform database errors jadi HTTP exceptions yang jelas. DTOs validasi input sebelum masuk ke business logic. Frontend tangkap API errors dan tampilkan error messages yang user-friendly. Validation errors dikasih tau spesifik field mana yang bermasalah. Network errors dan timeout di-handle dengan baik biar aplikasi tetap stabil.

Struktur database

User model

```
model User {  
  id          String    @id @default(uuid())  
  email       String    @unique  
  password    String  
  name        String
```

```

    refreshToken String?
    createdAt     DateTime @default(now())
    updatedAt     DateTime @updatedAt
    posts         Post[]
    replies       Reply[]
}

```

Post model

```

model Post {
  id          String    @id @default(uuid())
  title       String
  content     String
  published   Boolean   @default(false)
  imageUrl    String?
  authorId    String
  author      User      @relation(fields: [authorId], references: [id])
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
  replies     Reply[]
}

```

Reply model

```

model Reply {
  id          String    @id @default(uuid())
  content     String
  postId      String
  post        Post      @relation(fields: [postId], references: [id])
  authorId    String
  author      User      @relation(fields: [authorId], references: [id])
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}

```

API endpoints

Authentication

- `POST /api/auth/register` - Registrasi user baru
- `POST /api/auth/login` - Login dan dapatkan tokens

- `POST /api/auth/refresh` - Refresh access token

Posts

- `GET /api/posts` - List semua posts (dengan pagination & search)
- `GET /api/posts/:id` - Detail post dengan replies
- `POST /api/posts` - Create post baru (protected)
- `PATCH /api/posts/:id` - Update post (protected, author only)
- `DELETE /api/posts/:id` - Delete post (protected, author only)

Replies

- `POST /api/posts/:id/reply` - Create reply pada post (protected)
- `DELETE /api/posts/:postId/reply/:replyId` - Delete reply (protected, author only)

Fitur keamanan

1. Password Hashing: Bcrypt dengan salt rounds
2. JWT Authentication: Access & Refresh tokens
3. Authorization Checks: Owner-only operations
4. Input Validation: DTO validation dengan class-validator
5. CORS Configuration: Proper CORS setup untuk development
6. Database Isolation: Separate databases untuk dev dan test
7. SQL Injection Protection: Prisma ORM query parameterization