# **LAPORAN PROJECT**

# **MANAJEMEN EVENT**



**Oleh:**

|  |  |
| --- | --- |
| **M. AKMAL DZULFIKAR** | **20230040069** |
| **FAJRUL FALAH** | **20230040004** |

## **TEKNIK INFORMATIKA**

## **UNIVERSITAS NUSA PUTRA**

## **2025**

Contents

[**LAPORAN PROJECT** 1](#_Toc188603958)

[**MANAJEMEN EVENT** 1](#_Toc188603959)

[**TEKNIK INFORMATIKA** 1](#_Toc188603960)

[**UNIVERSITAS NUSA PUTRA** 1](#_Toc188603961)

[**2025** 1](#_Toc188603962)

[**BAB I** 3](#_Toc188603963)

[**PENDAHULUAN** 3](#_Toc188603964)

[**1.** **Latar Belakang** 3](#_Toc188603965)

[**2.** **Rumusan Masalah** 3](#_Toc188603966)

[**3.** **Tujuan** 3](#_Toc188603967)

[**BAB II** 4](#_Toc188603968)

[**METODOLOGI** 4](#_Toc188603969)

[**1.** **Desain Sistem** 4](#_Toc188603970)

[**2. Fitur-Fitur Utama** 5](#_Toc188603971)

[**2.** **Perancangan Website** 6](#_Toc188603972)

[a. Endpoint Event 8](#_Toc188603973)

[b. Endpoint Participant 9](#_Toc188603974)

[c. Endpoint Registration 9](#_Toc188603975)

[d. Endpoint Login 10](#_Toc188603976)

[e. Endpoint Additional Queries 10](#_Toc188603977)

[**3.** **Teknologi yang Digunakan** 11](#_Toc188603978)

[**a.** **Express.js** 11](#_Toc188603979)

[**b. MySQL** 12](#_Toc188603980)

[**BAB III** 13](#_Toc188603981)

[**PENGUJIAN** 13](#_Toc188603982)

[**BAB IV** 20](#_Toc188603983)

[**PENUTUP** 20](#_Toc188603984)

[**1.** **Kesimpulan** 20](#_Toc188603985)

[**2.** **Saran** 20](#_Toc188603986)

## 

## **BAB I**

## **PENDAHULUAN**

### **Latar Belakang**

Dalam perkembangan teknologi saat ini, pengelolaan data menjadi salah satu kebutuhan penting untuk meningkatkan efisiensi dan produktivitas, terutama dalam organisasi atau event management. Salah satu teknologi yang mendukung proses pengelolaan data tersebut adalah **Application Programming Interface (API)**. API memungkinkan berbagai sistem untuk saling terhubung dan mempermudah pengelolaan serta akses data secara terpusat.

Sistem ini dirancang untuk membantu pengelolaan data acara (events), peserta (participants), dan registrasi (registrations). Jika dilakukan secara manual, pengelolaan data seringkali memakan waktu yang lama, rentan terhadap kesalahan, dan sulit diakses secara real-time. Dengan adanya API, proses pengelolaan data dapat dilakukan dengan lebih cepat, efisien, dan mudah diintegrasikan dengan berbagai aplikasi atau platform.

### **Rumusan Masalah**

* 1. Bagaimana membangun sistem API yang dapat mendukung pengelolaan data acara, peserta, dan registrasi secara efektif?
  2. Bagaimana memastikan sistem ini memiliki tingkat keamanan yang memadai untuk melindungi data dari akses yang tidak sah?
  3. Bagaimana menyediakan fitur pencarian dan laporan agar data yang diperlukan dapat diakses dengan mudah dan cepat?

### **Tujuan**

* 1. Membangun sistem API untuk mempermudah pengelolaan data acara, peserta, dan registrasi.
  2. Menyediakan fitur pencarian dan laporan yang mendukung analisis data, seperti daftar peserta berdasarkan nama atau acara mendatang.
  3. Menyediakan 20 endpoint API yang fungsional untuk berbagai operasi seperti CRUD (Create, Read, Update, Delete), pencarian, dan laporan.

### 

### 

## **BAB II**

## **METODOLOGI**

### **Desain Sistem**

Sistem manajemen event ini dirancang sebagai platform terintegrasi yang memudahkan perencanaan, pelaksanaan, dan evaluasi berbagai jenis acara. Platform ini berbasis web dengan antarmuka yang user-friendly, memberikan pengalaman pengguna yang optimal.

#### **1.1. Arsitektur Sistem**

* **Platform Berbasis Web:** Sistem dibangun di atas platform web untuk aksesibilitas yang luas dari berbagai perangkat.
* **Desain Responsif:** Antarmuka dirancang untuk menyesuaikan diri dengan berbagai ukuran layar, memastikan tampilan yang optimal.

#### **1.2. Komponen Utama**

* **Halaman Beranda:**
  + Menyajikan informasi umum tentang platform, fitur utama, dan event unggulan.
  + Menampilkan kalender event, berita terbaru, dan statistik penggunaan.
* **Profil Pengguna:**
  + Menyediakan profil detail untuk setiap pengguna (organizer, peserta, sponsor).
  + Menampilkan riwayat event, preferensi, dan informasi kontak.
* **Manajemen Event:**
  + Membuat dan mengelola detail event secara komprehensif.
  + Mengunggah materi promosi, mengatur jadwal, dan mengelola peserta.
* **Registrasi Peserta:**
  + Sistem pendaftaran online yang mudah dan aman.
  + Pembayaran tiket secara online dan konfirmasi kehadiran.
* **Komunikasi:**
  + Forum diskusi, pesan pribadi, dan notifikasi untuk interaksi pengguna.
* **Evaluasi Event:**
  + Kuesioner kepuasan peserta dan laporan data analitik.
* **Sistem Pencarian dan Rekomendasi:**
  + Pencarian berdasarkan kata kunci dan rekomendasi event yang relevan.
* **Manajemen Konten:**
  + Fitur untuk mengunggah dan mengelola berbagai jenis konten.
* **Panel Admin:**
  + Akses penuh untuk mengelola seluruh aspek platform dan menghasilkan laporan.

#### **1.3. Alur Kerja Sistem**

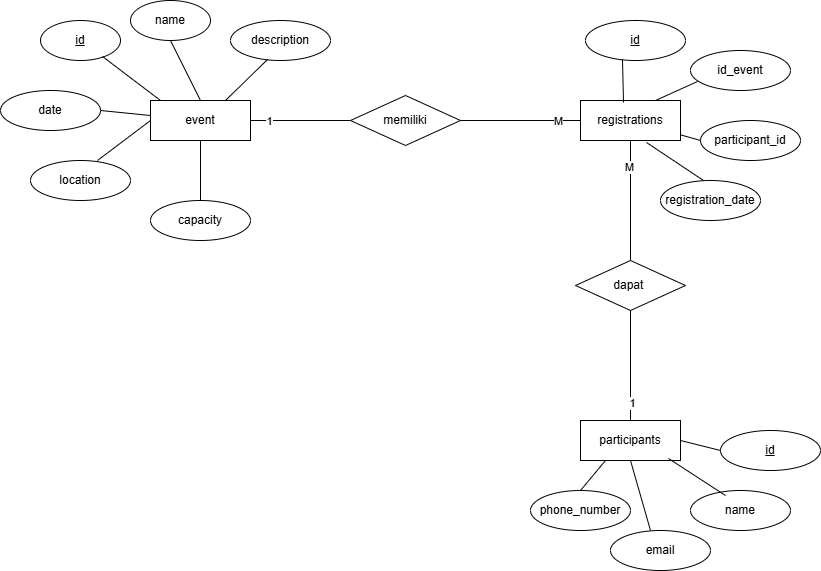
* **Pendaftaran Pengguna:** Pengguna mendaftar dan membuat profil.
* **Penjelajahan Event:** Pengguna menjelajahi event yang tersedia.
* **Registrasi:** Pengguna mendaftar dan melakukan pembayaran.
* **Partisipasi:** Pengguna berpartisipasi dalam event dan memberikan umpan balik.
* **Administrasi:** Admin mengelola event, memantau partisipasi, dan menghasilkan laporan.

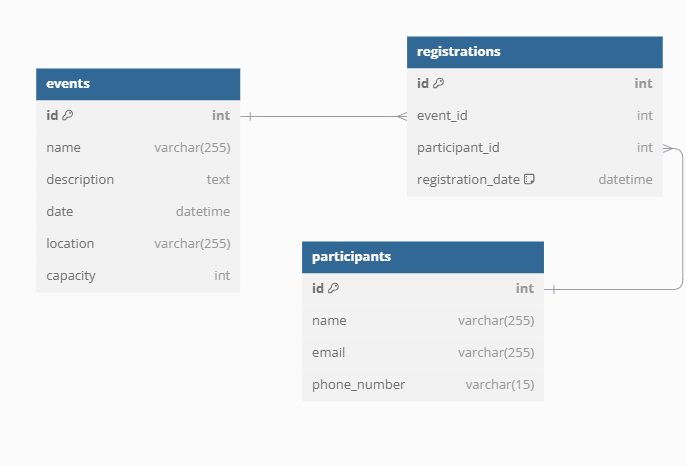
### **2. Fitur-Fitur Utama**

* **Pembuatan Event:**
  + Membuat event baru dengan detail lengkap (judul, tanggal, lokasi, deskripsi, dll.).
  + Mengunggah materi promosi (poster, banner, brosur).
  + Mengatur jadwal acara dan sesi.
  + Mengelola daftar peserta dan speaker.
* **Registrasi Peserta:**
  + Sistem pendaftaran online yang mudah dan aman.
  + Pembayaran tiket secara online.
  + Konfirmasi kehadiran dan pengiriman tiket elektronik.
* **Komunikasi:**
  + Forum diskusi untuk peserta dan penyelenggara.
  + Fitur pesan pribadi untuk komunikasi langsung.
  + Notifikasi terkait pembaruan event.
* **Evaluasi Event:**
  + Kuesioner kepuasan peserta.
  + Laporan data analitik (partisipasi, engagement, dll.).
* **Sistem Pencarian dan Rekomendasi:**
  + Pencarian berdasarkan kata kunci (topik, lokasi, tanggal).
  + Rekomendasi event berdasarkan preferensi pengguna dan riwayat pencarian.

### **Perancangan Website**

* 1. **ERD (Entity-Relationship Diagram) database**
     1. **Penjelasan**ERD menunjukkan struktur basis data untuk sistem ini, termasuk entitas, atribut, dan hubungan antar entitas. Berikut adalah entitas utama dan hubungannya :

****

****

* 1. **Hubungan Antar Entitas**
     1. Event memiliki hubungan 1:N dengan Registration, karena satu event dapat memiliki banyak pendaftar.
     2. Participant memiliki hubungan 1:N dengan Registration, karena satu peserta dapat mengikuti banyak event.
     3. Registration memiliki hubungan N:1 dengan Event dan Participant, karena satu registrasi terkait dengan satu event dan satu peserta.
  2. **Endpoint Autentikasi**

a. Endpoint Event

i. **GET /events**

1. **Deskripsi**: Mengambil daftar semua event.
2. **Input**: Tidak ada.
3. **Output**: Daftar semua event.

ii. **GET /events/:id**

1. **Deskripsi**: Mengambil detail event berdasarkan ID.
2. **Input**: ID event.
3. **Output**: Detail event berdasarkan ID.

iii. **POST /events**

1. **Deskripsi**: Menambahkan event baru.
2. **Input**: nama, deskripsi, tanggal, lokasi, kapasitas.
3. **Output**: Konfirmasi bahwa event berhasil ditambahkan.

iv. **PUT /events/:id**

1. **Deskripsi**: Memperbarui event berdasarkan ID.
2. **Input**: ID event, nama, deskripsi, tanggal, lokasi, kapasitas.
3. **Output**: Konfirmasi bahwa event berhasil diperbarui.

v. **DELETE /events/:id**

1. **Deskripsi**: Menghapus event berdasarkan ID.
2. **Input**: ID event.
3. **Output**: Konfirmasi bahwa event berhasil dihapus.

b. Endpoint Participant

i. **GET /participants**

1. **Deskripsi**: Mengambil daftar semua peserta.
2. **Input**: Tidak ada.
3. **Output**: Daftar semua peserta.

ii. **GET /participants/:id**

1. **Deskripsi**: Mengambil detail peserta berdasarkan ID.
2. **Input**: ID peserta.
3. **Output**: Detail peserta berdasarkan ID.

iii. **POST /participants**

1. **Deskripsi**: Menambahkan peserta baru.
2. **Input**: nama, email, nomor telepon, password.
3. **Output**: Konfirmasi bahwa peserta berhasil ditambahkan.

iv. **PUT /participants/:id**

1. **Deskripsi**: Memperbarui detail peserta berdasarkan ID.
2. **Input**: ID peserta, nama, email, nomor telepon.
3. **Output**: Konfirmasi bahwa peserta berhasil diperbarui.

v. **DELETE /participants/:id**

1. **Deskripsi**: Menghapus peserta berdasarkan ID.
2. **Input**: ID peserta.
3. **Output**: Konfirmasi bahwa peserta berhasil dihapus.

c. Endpoint Registration

i. **GET /registrations**

1. **Deskripsi**: Mengambil daftar semua registrasi.
2. **Input**: Tidak ada.
3. **Output**: Daftar semua registrasi.

ii. **GET /registrations/:id**

1. **Deskripsi**: Mengambil detail registrasi berdasarkan ID.
2. **Input**: ID registrasi.
3. **Output**: Detail registrasi berdasarkan ID.

iii. **POST /registrations**

1. **Deskripsi**: Menambahkan registrasi baru.
2. **Input**: event\_id, participant\_id.
3. **Output**: Konfirmasi bahwa registrasi berhasil ditambahkan.

iv. **DELETE /registrations/:id**

1. **Deskripsi**: Menghapus registrasi berdasarkan ID.
2. **Input**: ID registrasi.
3. **Output**: Konfirmasi bahwa registrasi berhasil dihapus.

d. Endpoint Login

i. **POST /participants/login**

1. **Deskripsi**: Login peserta.
2. **Input**: email, password.
3. **Output**: Token JWT untuk autentikasi.

e. Endpoint Additional Queries

i. **GET /events/registrations/:eventId**

1. **Deskripsi**: Mengambil daftar registrasi untuk event tertentu.
2. **Input**: eventId (ID event).
3. **Output**: Daftar registrasi peserta untuk event tersebut.

ii. **GET /participants/registrations/:participantId**

1. **Deskripsi**: Mengambil daftar registrasi untuk peserta tertentu.
2. **Input**: participantId (ID peserta).
3. **Output**: Daftar event yang diikuti oleh peserta tersebut.

iii. **PUT /events/:id/capacity**

1. **Deskripsi**: Memperbarui kapasitas event.
2. **Input**: ID event, kapasitas.
3. **Output**: Konfirmasi bahwa kapasitas event berhasil diperbarui.

iv. **GET /registrations/date**

1. **Deskripsi**: Mengambil daftar registrasi berdasarkan tanggal tertentu.
2. **Input**: Tanggal (query parameter date).
3. **Output**: Daftar registrasi yang sesuai dengan tanggal.

v. **GET /events/upcoming**

1. **Deskripsi**: Mengambil daftar event yang akan datang.
2. **Input**: Tidak ada.
3. **Output**: Daftar event yang belum berlangsung.

### **Teknologi yang Digunakan**

Berikut adalah penjelasan dan kegunaan teknologi yang disebutkan dalam membangun program berbagi keterampilan dan pengalaman:

### **Express.js**

#### **Penjelasan**

Express.js adalah framework backend untuk Node.js yang ringan, fleksibel, dan cepat. Framework ini mempermudah pembuatan aplikasi web dan API dengan menyediakan berbagai fitur bawaan untuk menangani routing, middleware, dan interaksi dengan database.

#### **Kegunaan dalam Program**

* + - 1. **Routing**: Mengatur jalur endpoint API seperti /register, /posts, atau /comments.
      2. **Middleware**: Menyediakan mekanisme untuk validasi data, autentikasi, dan logging.
      3. **Manajemen API**: Mengelola interaksi antara frontend dan database melalui RESTful API.
      4. **Skalabilitas**: Mendukung pengembangan aplikasi yang besar dan modular dengan struktur kode yang terorganisasi.

### **b. MySQL**

#### **i. Penjelasan**

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang populer, open-source, dan cocok untuk aplikasi web karena kecepatan, skalabilitas, dan kemudahan integrasinya dengan backend.

#### **ii. Kegunaan dalam Program**

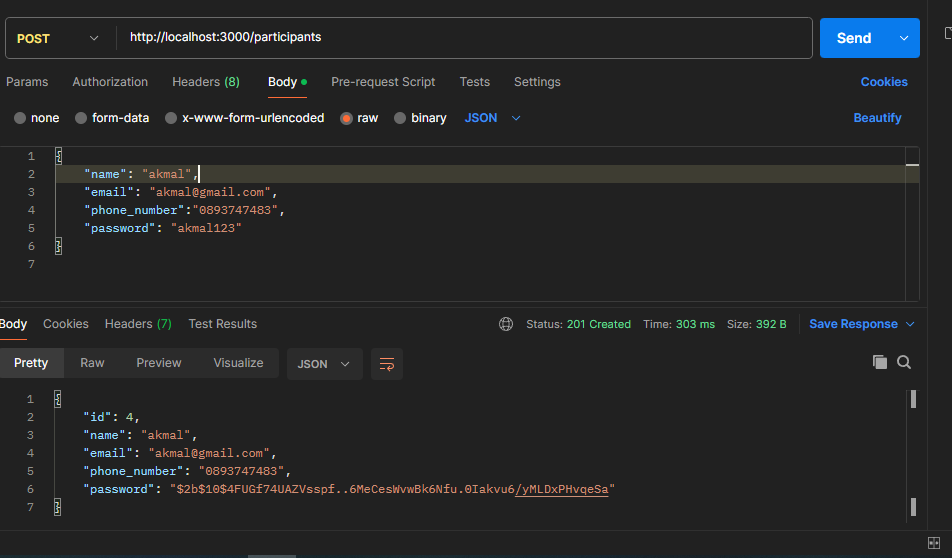
* + 1. **Penyimpanan Data**: Menyimpan data penting, seperti informasi pengguna, postingan, komentar, dan laporan.
    2. **Relasi Antar Data**: Mengelola hubungan antar entitas, misalnya hubungan antara pengguna dan postingan (satu ke banyak).
    3. **Keamanan Data**: Mendukung fitur keamanan seperti enkripsi password dan pembatasan akses data.

## **BAB III**

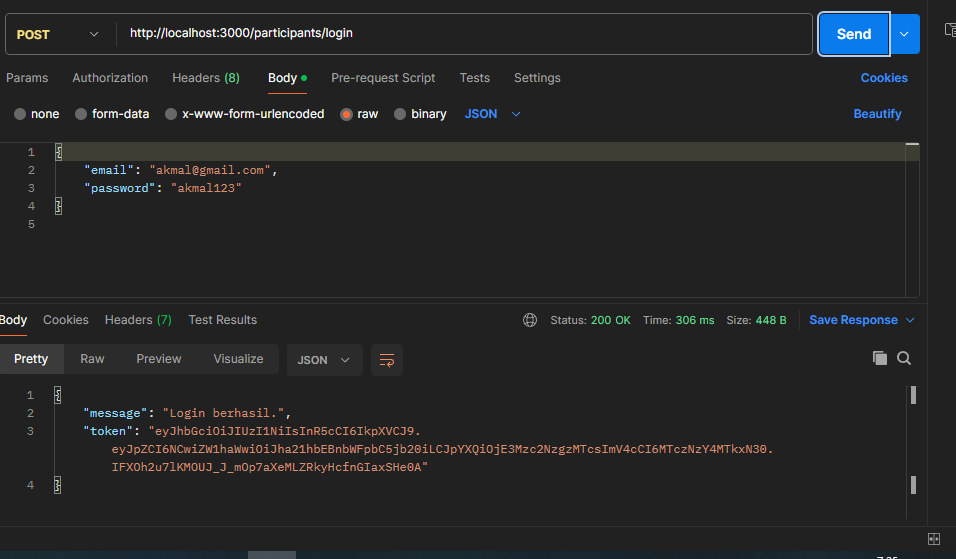
## **PENGUJIAN**

Berisi pengujian API testing dari aplikasi :

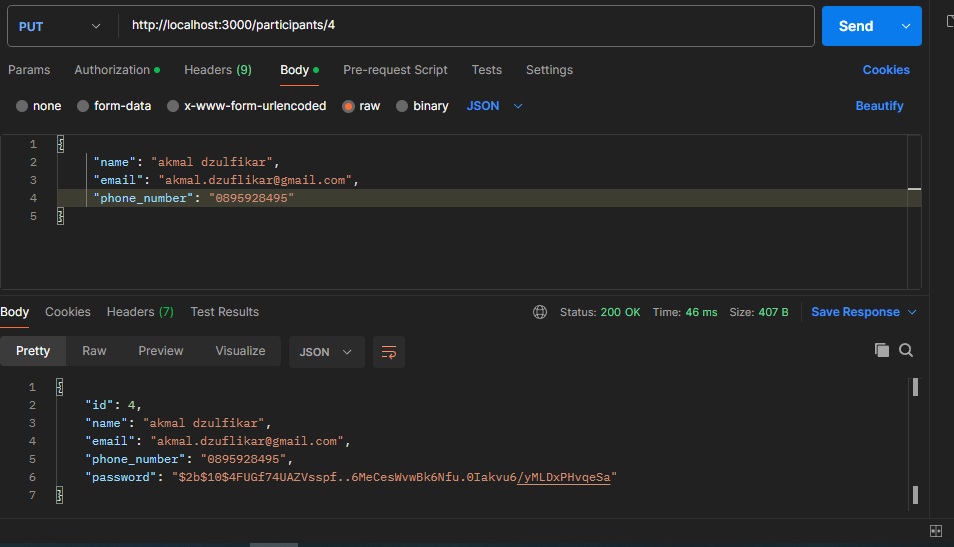
1. **POST /participants**



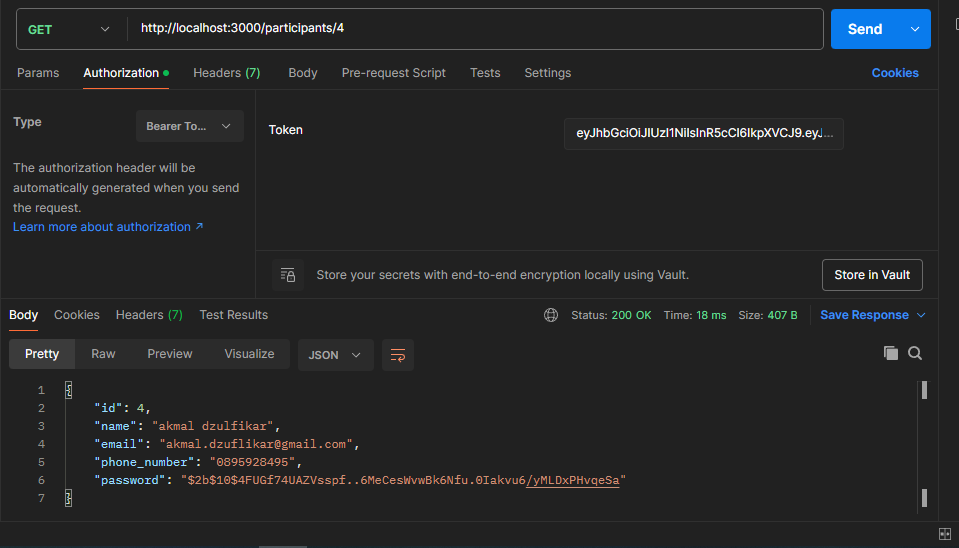
1. **POST /participants/login**



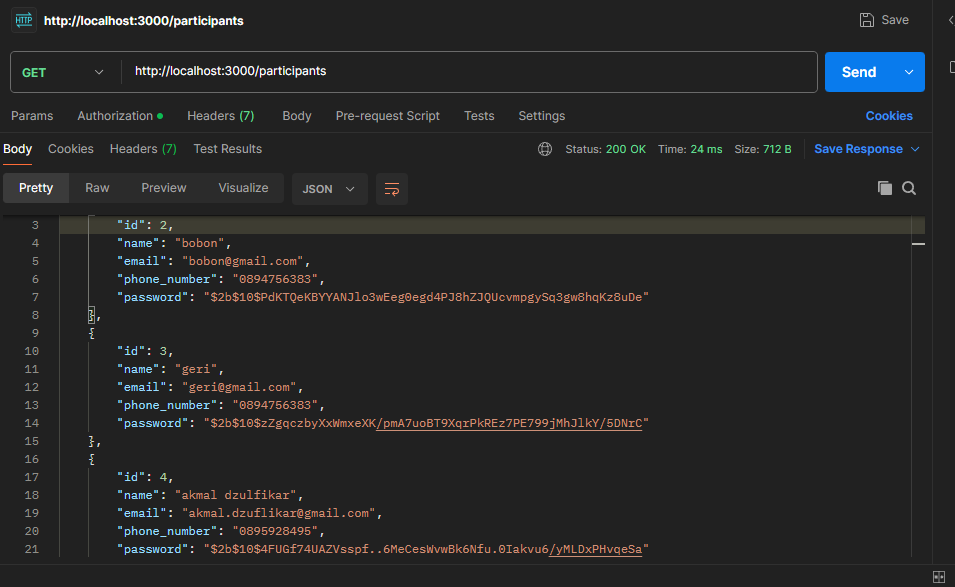
1. **PUT /participant/(id)**



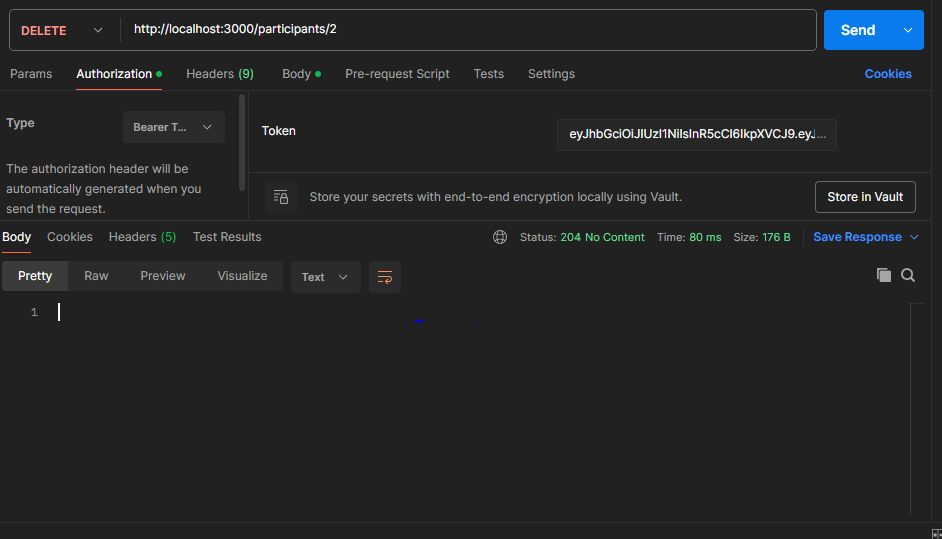
1. **GET /participants/(id)**

****

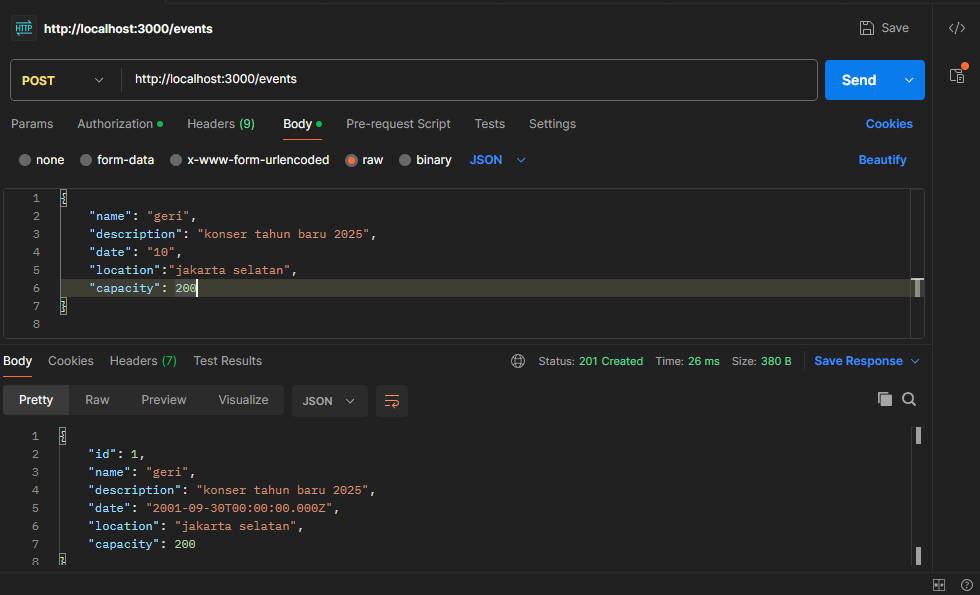
1. **GET /all participants**

****

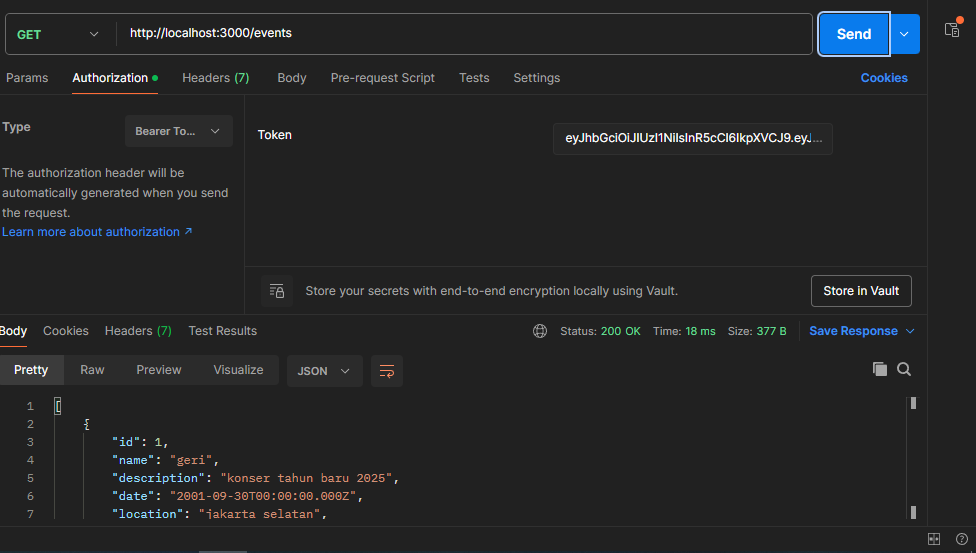
1. **DELETE /participants/(id)**

****

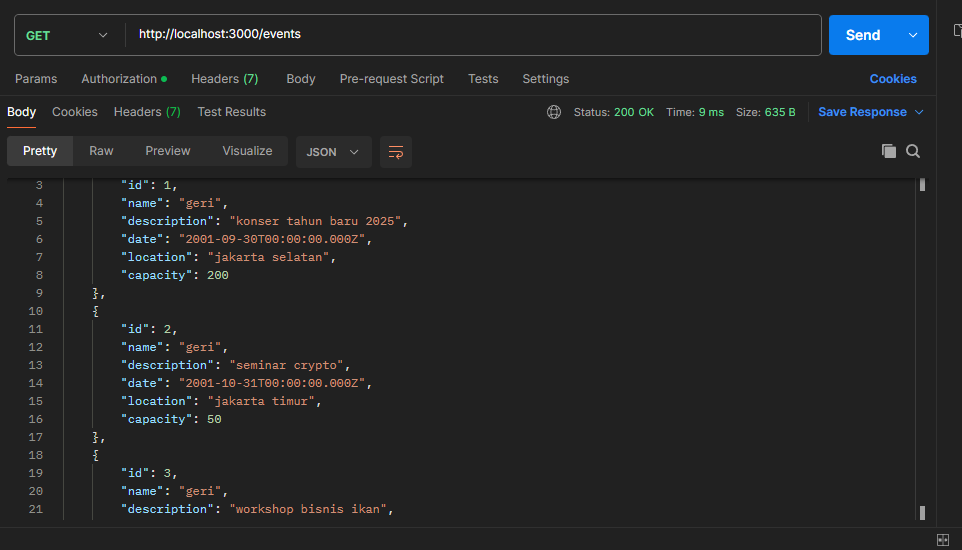
1. **POST /create events**

****

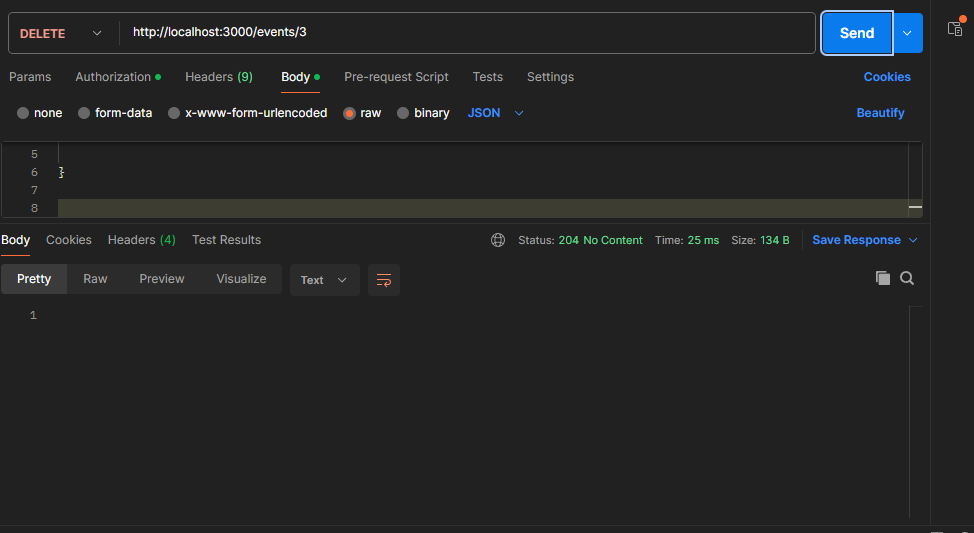
1. **GET /get events**

****

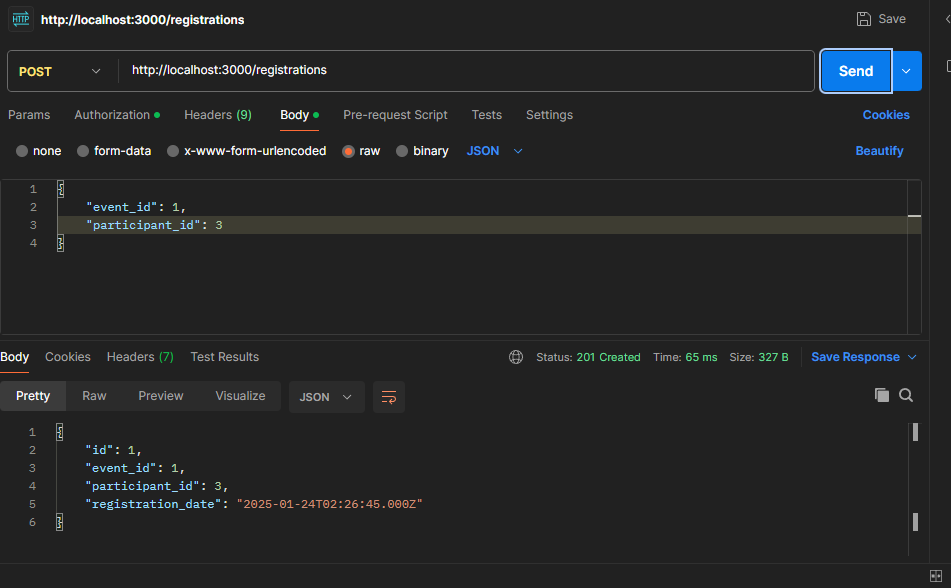
1. **GET /all events**

****

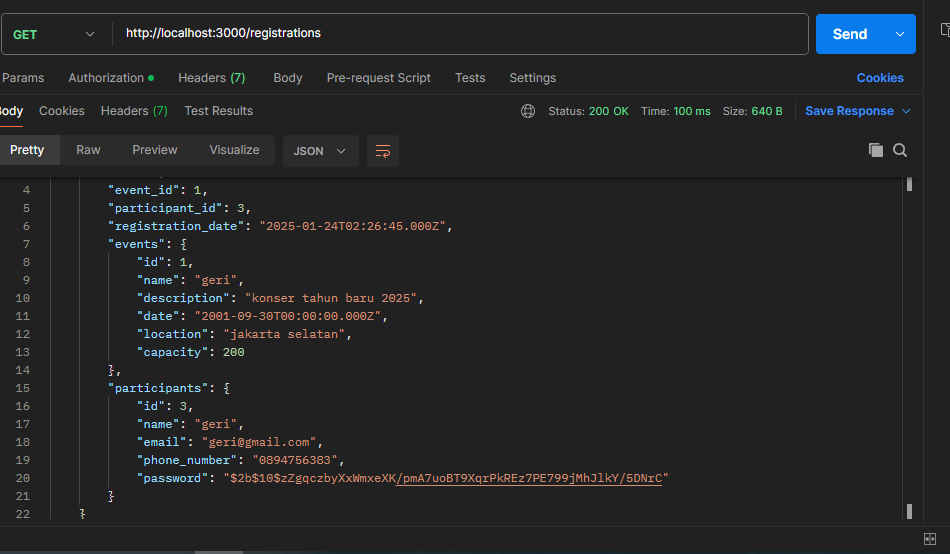
1. **DELETE /events/(id)**

****

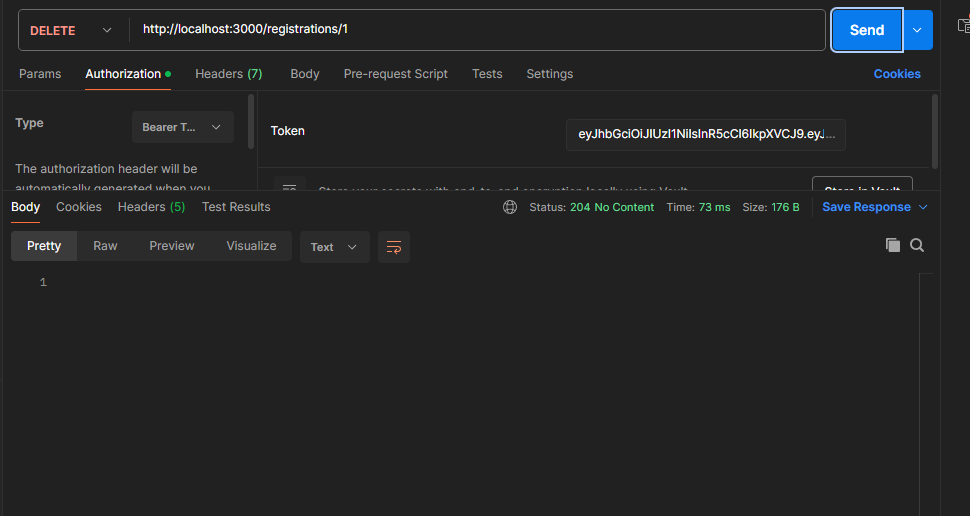
1. **POST /registrations**

****

1. **GET / all registrations**



1. **DELETE /registrations**



## 

## 

## 

## 

## 

## 

## **BAB IV**

## **PENUTUP**

### **Kesimpulan**

Sistem ini memungkinkan pengelolaan data event dan peserta dengan mudah. Pengguna bisa mendaftar untuk mengikuti event, dan admin bisa menambah, mengubah, atau menghapus event dan peserta. Sistem juga menyediakan fitur login untuk peserta yang terdaftar dan mengelola registrasi mereka. Selain itu, ada fitur untuk memperbarui kapasitas event dan mencari event yang akan datang. Secara keseluruhan, sistem ini memberikan solusi yang efisien untuk manajemen event dan peserta dengan keamanan yang terjamin.

### **Saran**

Rekomendasi untuk pengembangan lebih lanjut.