

# LAPORAN PROGRES MINGGU 3: IMPLEMENTASI CRUD DASAR

## Proyek Database "Eventify" - Kelompok 9

Disusun oleh:

- Rafael Mahardika Arya Dewamurti (24/536279/PA/22755)
- Bobby Rahman Hartanto (24/539383/PA/22903)

## I. PENDAHULUAN

Sesuai dengan panduan proyek, tugas Minggu 3 adalah mengimplementasikan operasi **CRUD (Create, Read, Update, Delete)** untuk entitas utama. Laporan ini mendokumentasikan progres kami dalam membangun aplikasi web fungsional pertama yang terhubung langsung dengan database eventify\_db yang telah kami rancang di Week 2.

Kami telah berhasil membangun arsitektur 3-tier sederhana:

1. **Database (MySQL):** Pondasi yang menyimpan data, di-setup menggunakan schema\_eventify.sql dari Week 2.
2. **Backend (Node.js):** "Mesin" atau server yang bertindak sebagai perantara, ditulis dalam file backend.js.
3. **Frontend (HTML/CSS/JS):** "Tampilan" atau UI yang dilihat pengguna, ditulis dalam file frontend.html.

Laporan ini akan berfokus pada demo fungsionalitas dan penjelasan singkat arsitektur yang kami gunakan.

## II. ARSITEKTUR & PENJELASAN KONEKSI

Untuk proyek ini, kami tidak menghubungkan frontend langsung ke database karena sangat tidak aman. Kami menggunakan arsitektur *client-server* di mana frontend dan backend adalah dua program terpisah yang "berbicara" satu sama lain.

1. Frontend (frontend.html)

Ini adalah dasbor UI yang kami buat menggunakan HTML dan TailwindCSS. Bagian ini "buta" (agnostic), dia tidak tahu apa-apa soal database. Dia hanya tahu cara "meminta" data ke sebuah alamat URL (yaitu `http://localhost:3001/api/events`) menggunakan JavaScript `fetch`.

2. Backend (backend.js)

Ini adalah "otak" aplikasi. Dia berjalan di `localhost:3001` menggunakan Node.js dan

Express. Backend adalah satu-satunya yang memegang "kunci" (password) ke database. Tugasnya adalah menerima permintaan dari frontend, menerjemahkannya menjadi perintah SQL (misal: `SELECT * FROM Events`), mengeksekusinya ke database, dan mengirimkan hasilnya kembali ke frontend sebagai JSON.

### 3. Database (MySQL/XAMPP)

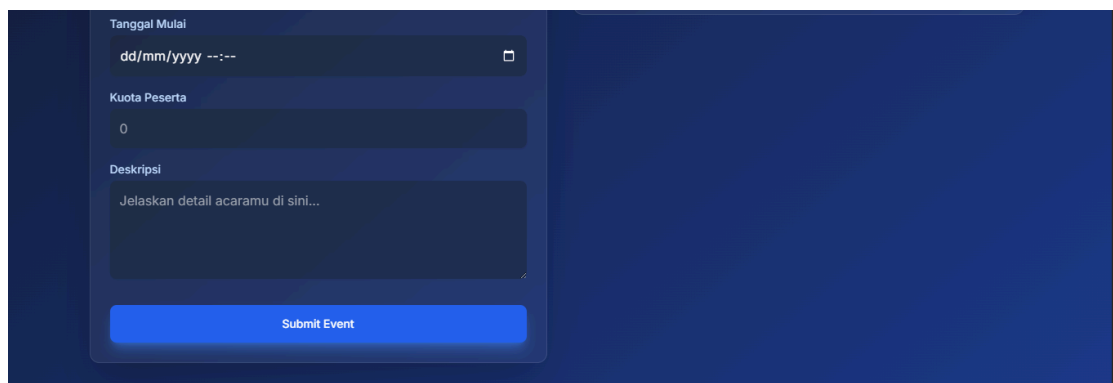
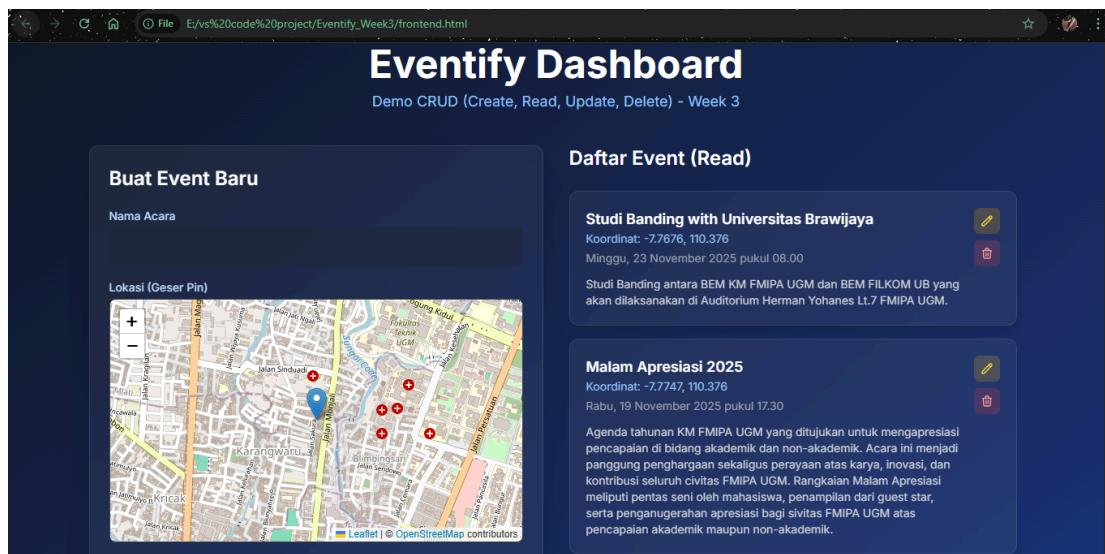
Ini adalah "kulkas" kami, yang berjalan di localhost (biasanya port 3306). Dia hanya menyimpan data dan menjalankan perintah SQL yang diberikan oleh backend.

## III. DEMO FUNGSIONALITAS APLIKASI

Untuk membuktikan bahwa koneksi ini berjalan, kami mendemokan 4 (empat) operasi CRUD pada entitas utama kami, yaitu **Events**.

### A. Tampilan Antarmuka (UI)

Kami mendesain UI profesional dengan tema *dark mode* dan efek *glassmorphism* (kaca buram) menggunakan TailwindCSS. Layout dibagi dua: formulir di kiri, dan daftar data di kanan.



- **Penjelasan Gambar:** Screenshot ini menunjukkan tampilan utama dasbor "Eventify". Kolom kiri adalah formulir CREATE/UPDATE, dan kolom kanan adalah panel READ/DELETE.

## B. Operasi CREATE (Membuat Event)

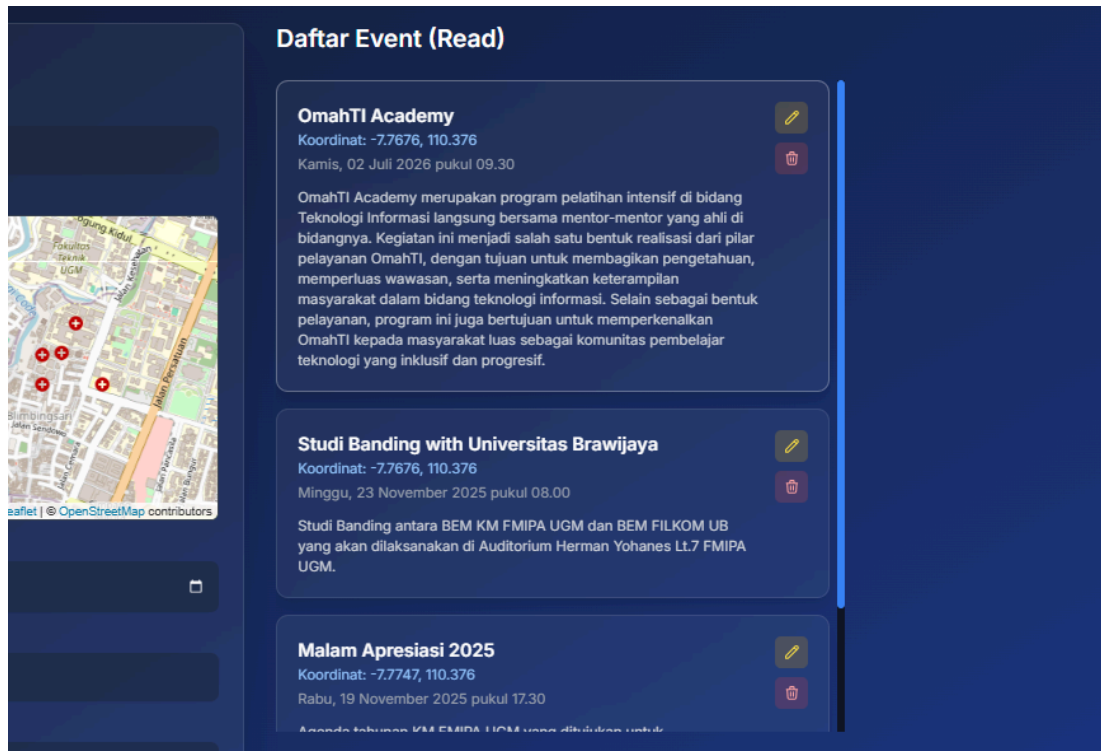
Pengguna mengisi formulir di sebelah kiri (termasuk memilih lokasi di peta interaktif Leaflet.js) dan menekan "Submit Event".

The screenshot displays the Eventify dashboard. On the left, the 'Buat Event Baru' (Create New Event) form is active. It includes fields for 'Nama Acara' (Event Name) with the value 'OmahTI Academy', a map for 'Lokasi (Geser Pin)' (Location) showing a blue pin near Universitas Gadjah Mada, 'Tanggal Mulai' (Start Date) set to '02/07/2026 09:30', 'Kuota Peserta' (Participant Quota) set to '250', and a 'Deskripsi' (Description) field containing text about the OmahTI Academy program. On the right, the 'Daftar Event (Read)' panel lists two events: 'Studi Banding with Universitas Br...' and 'Malam Apresiasi 2025', each with its coordinates and date.

- **Penjelasan Gambar:** Data event baru sedang diisi.
- **Alur:** JavaScript di frontend akan mengirim data ini menggunakan fetch dengan method POST ke backend.js. Backend kemudian menjalankan perintah INSERT INTO Events ... ke database. Data yang baru dibuat akan langsung muncul di panel kanan.

## C. Operasi READ (Membaca Event)

Saat halaman frontend.html pertama kali dibuka, JavaScript otomatis memanggil fungsi `loadEvents()`.



- **Penjelasan Gambar:** Data event berhasil ditampilkan di panel kanan.
- **Alur:** Fungsi `loadEvents()` melakukan fetch dengan method GET ke `backend.js`. Backend menjalankan `SELECT * FROM Events`, dan data yang diterima dari database ditampilkan sebagai daftar di UI.

## D. Operasi UPDATE (Mengedit Event)

Saat pengguna mengklik tombol "Edit" (ikon pensil), data event tersebut akan dimuat kembali ke formulir di sebelah kiri.

**Studi Banding with**  
Koordinat: -7.7676, 110.37  
Minggu, 23 November 2025

Studi Banding antara BEI yang akan dilaksanakan UGM.

**Malam Apresiasi 20**  
Koordinat: -7.7747, 110.37  
Rabu, 19 November 2025  
Academy KM CM

Tanggal Mulai  
02/07/2026 09:30

Kuota Peserta  
36

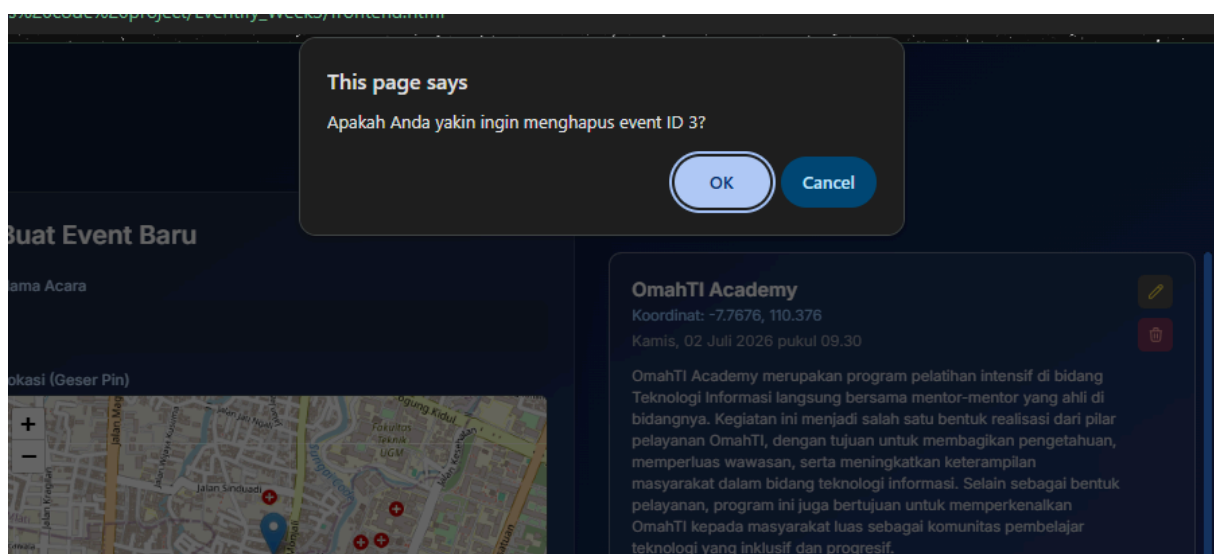
Deskripsi  
OmahTI Academy merupakan program pelatihan intensif di bidang Teknologi Informasi langsung bersama mentor-mentor yang ahli di bidangnya. Kegiatan ini menjadi salah satu bentuk realisasi dari pilar pelayanan OmahTI, dengan tujuan untuk membagikan pengetahuan, memperluas wawasan, serta meningkatkan keterampilan masyarakat dalam bidang teknologi informasi. Selain sebagai bentuk pelayanan, program ini juga bertujuan untuk memperkenalkan OmahTI kepada masyarakat luas sebagai komunitas pembelajar teknologi yang inklusif dan progresif.

Simpan Perubahan Batal Edit

- **Penjelasan Gambar:** Formulir otomatis terisi dengan data event yang dipilih, dan tombol berubah menjadi "Simpan Perubahan".
- **Alur:** Setelah diubah dan disubmit, JavaScript mengirim data dengan method PUT. Backend menjalankan UPDATE Events SET ... WHERE event\_id = ?. Daftar di sebelah kanan akan otomatis me-refresh data yang sudah ter-update.

## E. Operasi DELETE (Menghapus Event)

Saat pengguna mengklik tombol "Hapus" (ikon tong sampah), sebuah *pop-up* konfirmasi akan muncul.



- **Penjelasan Gambar:** Konfirmasi sebelum data dihapus permanen.
- **Alur:** Jika dikonfirmasi, JavaScript mengirim permintaan dengan method DELETE ke backend. Backend menjalankan DELETE FROM Events WHERE event\_id = ?. Data akan hilang dari daftar.

## IV. PENUTUP (Kesimpulan)

Pada Minggu 3 ini, kami telah berhasil membangun aplikasi web 3-tier fungsional dari nol. Kami telah membuktikan bahwa arsitektur backend Node.js dan frontend HTML/JS kami dapat terhubung dan melakukan empat operasi CRUD (Create, Read, Update, Delete) ke database MySQL (eventify\_db) yang kami rancang di Week 2.

Struktur ini sudah siap untuk dikembangkan lebih lanjut di Week 4, yaitu untuk mengintegrasikan tabel-tabel lain seperti Users (untuk login) serta Organizers dan Categories (untuk data *dropdown* di formulir).

# LAMPIRAN: Petunjuk Menjalankan Demo

Berikut adalah langkah-langkah ringkas untuk menjalankan demo ini di komputer lokal.

## 1. Prasyarat

- **XAMPP:** Pastikan Apache dan MySQL sedang berjalan.
- **Node.js:** Pastikan Node.js (termasuk npm) sudah terinstal di komputer.

## 2. Langkah 1: Setup Database

1. Buka phpMyAdmin (<http://localhost/phpmyadmin>).
2. Buat database baru dengan nama eventify\_db.
3. Klik database eventify\_db, lalu pilih tab "**Import**".
4. Pilih file schema\_eventify.sql (dari Week 2) dan klik "Go".  
(Pastikan 6 tabel sudah muncul: Users, Events, Organizers, dll).

## 3. Langkah 2: Setup Backend (Server)

1. Buka folder proyek (Eventify\_Week3) di Terminal/CMD.
2. Install semua library yang dibutuhkan (hanya sekali):  
`npm install`
3. Buka file backend.js dan **edit password database** (di baris 23) sesuaikan dengan XAMPP Anda (biasanya password: ").
4. Jalankan server:  
`node backend.js`
5. Biarkan terminal ini tetap berjalan (akan muncul pesan Server backend "Eventify" berjalan...).

## 3. Langkah 3: Menjalankan Frontend (Demo)

1. **Buka terminal BARU** (jangan tutup terminal backend).
2. Buka file frontend.html langsung di browser Anda (klik dua kali filenya).
3. Aplikasi CRUD siap didemokan.