



## Pengembangan Aplikasi Eventify: Sistem Web Terintegrasi antara Main App dan Admin Dashboard untuk Manajemen Event

Dwi Wahyuningtyas<sup>1</sup>, Reihan Setya Banda Syah Putra<sup>2</sup>, Raihan Canggih Panilih<sup>3</sup>,  
Rahmat Nur Shidiq<sup>4</sup>

<sup>1-4</sup> Universitas Duta Bangsa Surakarta, Indonesia

Email : [220101012@mhs.udb.ac.id](mailto:220101012@mhs.udb.ac.id)<sup>1\*</sup>, [220101034@mhs.udb.ac.id](mailto:220101034@mhs.udb.ac.id)<sup>2</sup>,  
[220101033@mhs.udb.ac.id](mailto:220101033@mhs.udb.ac.id)<sup>3</sup>, [220101032@mhs.udb.ac.id](mailto:220101032@mhs.udb.ac.id)<sup>4</sup>

Alamat: Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta, Jl. Bhayangkara No.55,  
Tipes, Kec. Serengan, Kota Surakarta, Jawa Tengah 57154

Korespondensi penulis: [220101012@mhs.udb.ac.id](mailto:220101012@mhs.udb.ac.id)<sup>1\*</sup>

**Abstract.** *The increasing prevalence of competitions and events necessitates efficient management systems to overcome challenges such as limited information dissemination and inefficient data handling. This study developed Eventify, an integrated web-based platform comprising a Main App for participants and organizers and an Admin Dashboard for monitoring and management. The system aims to simplify event administration, enhance transparency, and improve user experience. Built with the Laravel framework, Eventify utilizes a direct multi-database connection approach, streamlining data communication between the two applications. The development followed the SDLC waterfall model, encompassing analysis, design, implementation, and testing. Key findings indicate that Eventify effectively supports event creation, registration, participant data management, and administrative oversight. This integrated solution significantly improves operational efficiency and data transparency for event management.*

**Keywords:** *Event Management, Integrated System, Laravel, Web Application, Workflow Efficiency.*

**Abstrak.** Penyelenggaraan berbagai lomba dan kompetisi yang kian marak menuntut sistem manajemen efisien untuk mengatasi kendala penyebaran informasi dan penanganan data yang tidak efisien. Penelitian ini mengembangkan Eventify, sebuah platform berbasis web terintegrasi yang terdiri dari Aplikasi Utama untuk peserta dan penyelenggara serta Dasbor Admin untuk pemantauan dan pengelolaan. Sistem ini bertujuan menyederhanakan administrasi acara, meningkatkan transparansi, dan memperbaiki pengalaman pengguna. Dibangun dengan framework Laravel, Eventify menggunakan pendekatan koneksi multi-database langsung, yang menyederhanakan komunikasi data antar kedua aplikasi. Pengembangan mengikuti model waterfall SDLC, mencakup analisis, desain, implementasi, dan pengujian. Temuan kunci menunjukkan bahwa Eventify secara efektif mendukung pembuatan acara, pendaftaran, pengelolaan data peserta, dan pengawasan administratif. Solusi terintegrasi ini secara signifikan meningkatkan efisiensi operasional dan transparansi data untuk manajemen acara.

**Kata kunci:** Efisiensi Alur Kerja, Laravel, Manajemen Event, Sistem Terintegrasi, Aplikasi Web.

### 1. LATAR BELAKANG

Dalam beberapa tahun terakhir, dunia pendidikan dan masyarakat umum semakin aktif menyelenggarakan berbagai jenis lomba dan kompetisi, mulai dari bidang olahraga, akademik, seni, teknologi, hingga kreativitas digital. Ajang-ajang seperti ini tidak hanya menjadi sarana untuk menunjukkan kemampuan, tetapi juga sebagai wadah untuk menjalin relasi, mengembangkan potensi, serta meningkatkan daya saing generasi muda. Dengan berkembangnya teknologi informasi yang semakin cepat, khususnya pada hal komunikasi dan informasi. Kini, semua informasi yang kita inginkan dapat diakses dengan mudah melalui platform website internet (Al Qossam, 2024). Website adalah media yang sesuai untuk menyajikan informasi kegiatan karena dapat menampilkan informasi secara detail,

informatif, dan lebih interaktif (P. S. Hasugian, 2018). Selain itu, digitalisasi sistem informasi terbukti mampu meningkatkan efisiensi operasional dan transparansi data dalam pengelolaan kegiatan publik (Ardiansyah et al., 2021). Penyelenggaraan event sering menghadapi kendala seperti penyebaran informasi yang terbatas, pendaftaran manual yang rawan kesalahan, serta pengelolaan data peserta yang tidak efisien. Hal ini menegaskan bahwa diperlukan sistem digital yang terintegrasi untuk menyederhanakan proses dan meningkatkan pengalaman pengguna, yang menjadi latar belakang dikembangkannya platform Eventify. Untuk mengimplementasikan aplikasi ini, penulis menggunakan framework Laravel guna mempercepat proses pengembangan. Laravel dikenal sebagai salah satu framework PHP yang menawarkan struktur MVC, kemudahan dalam pengelolaan routing, serta integrasi yang kuat dengan database relasional, sehingga cocok untuk pengembangan aplikasi modern secara cepat dan terstruktur (Syahputra & Wahyuni, 2020). Berbeda dari pendekatan umum yang menggunakan REST API dan web service untuk menghubungkan dua sistem, integrasi antara Eventify main app dan Eventify audit dashboard dilakukan melalui koneksi langsung ke basis data yang sama, dengan konfigurasi yang diatur melalui file .env masing-masing aplikasi. File .env memungkinkan pengaturan koneksi database seperti host, nama database, serta kredensial secara fleksibel, sehingga kedua aplikasi dapat saling mengakses dan memperbarui data secara langsung tanpa perlu membuat endpoint API tambahan. Pendekatan ini dipilih untuk menyederhanakan arsitektur sistem dan meningkatkan efisiensi komunikasi data, serupa dengan metode integrasi langsung yang umum digunakan pada sistem internal dengan sumber data bersama (Bagir, 2018).

Aplikasi ini dirancang untuk mempermudah peserta dalam menemukan dan mengikuti lomba yang sesuai dengan minat dan kemampuan mereka, tanpa harus mengalami kendala administratif yang rumit. Selain itu, Eventify juga bertujuan mendukung penyelenggara dalam mengelola event dengan lebih efisien, mulai dari publikasi lomba, proses pendaftaran, hingga pengelolaan data peserta secara sistematis dan aman. Dengan sistem yang terintegrasi ini, transparansi dan kecepatan dalam proses administrasi lomba pun dapat meningkat secara signifikan.

## **2. KAJIAN TEORITIS**

### **2.1 Sistem manajemen event digital**

Sistem manajemen event digital merupakan solusi teknologi yang dirancang untuk mengatasi berbagai permasalahan yang sering muncul dalam penyelenggaraan acara, seperti distribusi informasi yang tidak merata, pendaftaran peserta yang tidak efisien, serta

pengelolaan data yang rentan terhadap kesalahan. Seiring berkembangnya kebutuhan masyarakat akan transparansi dan efisiensi dalam pelaksanaan event, sistem digital menjadi pilihan utama dalam menunjang proses manajemen acara.

Penelitian oleh Ardiansyah et al. (2021) merancang aplikasi manajemen event berbasis web di lingkungan Fakultas Ilmu Administrasi Universitas Brawijaya. Aplikasi ini mengintegrasikan pengajuan event, peminjaman venue, dan kalender digital berbasis Google Calendar, yang secara signifikan menyederhanakan alur administrasi internal.

Rancang bangun sistem informasi event berbasis web juga dikembangkan oleh Pratama dan Yuniarti (2022), dengan fokus pada pemisahan akses pengguna dan pengelola, serta kemampuan untuk menampilkan informasi event secara real-time. Hasil dari penelitian ini menunjukkan bahwa sistem mampu mengurangi beban kerja manual panitia dan mempercepat distribusi informasi kepada peserta.

Berdasarkan berbagai studi tersebut, dapat disimpulkan bahwa sistem manajemen event digital tidak hanya berfungsi sebagai alat pengorganisasian teknis, tetapi juga meningkatkan kualitas interaksi antara peserta dan penyelenggara melalui transparansi, aksesibilitas, dan efisiensi.

## 2.2 Aplikasi Web Terintegrasi

Integrasi aplikasi berbasis web secara langsung melalui koneksi database merupakan pendekatan di mana dua atau lebih aplikasi dihubungkan melalui konfigurasi multi-database tanpa menggunakan REST API atau JSON. Dalam pendekatan ini, masing-masing aplikasi dapat langsung membaca dan menulis data ke database satu sama lain dengan memanfaatkan konfigurasi koneksi di file `.env`

Rizky dan Aditama (2020) menjelaskan bahwa Laravel menyediakan fitur koneksi multi-database yang memungkinkan satu aplikasi terhubung ke lebih dari satu basis data secara simultan, memudahkan pemisahan dan pengelolaan data antara modul utama dan modul pendukung.

Ramadhan dan Hartati (2021) melakukan pengembangan sistem akademik dengan pendekatan koneksi langsung antar database untuk memisahkan data pengguna dan akademik. Pendekatan ini terbukti mempercepat waktu respon sistem karena tidak melalui lapisan API.

Yusuf dan Kurniawan (2022) dalam penelitiannya tentang aplikasi keuangan sekolah menemukan bahwa koneksi langsung antar database melalui file konfigurasi menghasilkan efisiensi waktu pemrosesan dan kemudahan integrasi sistem.

Metode koneksi langsung ini sangat cocok diterapkan dalam sistem yang berjalan pada jaringan lokal atau intranet, karena memberikan performa tinggi dan integrasi yang sederhana, selama aspek keamanan dan otorisasi database dikelola dengan baik..

### **3. METODE PENELITIAN**

Metode yang digunakan dalam Perancangan dan Pengembangan Aplikasi Eventify yaitu metode SDLC dengan model waterfall. System development life cycle (SDLC) adalah proses untuk memahami bagaimana sebuah sistem informasi dapat mendukung kebutuhan bisnis dengan merancang suatu sistem, membangun sistem tersebut dan menyampaikan kepada pengguna” (Dennis, Wixom, & Tegarden, 2015). Metode waterfall atau disebut juga dengan sekuensial linear merupakan metode pengembangan perangkat lunak yang bersifat sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pemeliharaan dan memiliki sifat sequential atau berurut dimulai dari menganalisis, mendesain software, pengkodean, menguji software, dan melakukan pemeliharaan software (Putra, W. A., Fitri, I., & Hidayatullah, D. 2022). Model waterfall ini memiliki beberapa tahapan, yaitu tahapan perencanaan, analisis, perancangan, implementasi, pengujian dan pemeliharaan. Model ini digunakan karena dapat menyesuaikan alur kerja yang jelas dan berurutan sesuai kebutuhan pengembangan aplikasi eventify.

### **4. HASIL DAN PEMBAHASAN**

#### **Analisis Kebutuhan Sistem**

Kebutuhan fungsional merujuk pada proses-proses yang akan dijalankan oleh sistem, serta informasi yang harus tersedia dan dihasilkan oleh sistem. (Setiyani, L., & Tjandra, E. 2024). Sedangkan kebutuhan non fungsional dapat diartikan sebagai perangkat-perangkat yang menunjang dalam perancangan sistem database (Bagir, M. H. 2018).

#### **Analisis Kebutuhan Fungsional**

Analisis kebutuhan fungsional merupakan bagian penting dalam analisis sistem yang menentukan apa saja yang harus dilakukan oleh sistem untuk mendukung proses bisnis atau tujuan aplikasi. Kebutuhan ini langsung berkaitan dengan fungsi-fungsi inti yang dapat digunakan oleh pengguna.

Beberapa kebutuhan fungsional dalam sistem Eventify antara lain:

- a) Pengguna (peserta atau penyelenggara) dapat melakukan registrasi dan login/logout ke dalam sistem.
- b) Penyelenggara dapat membuat, mengedit, dan menghapus data event lomba.

- c) Peserta dapat mencari dan mendaftar lomba secara online melalui form digital.
- d) Admin dapat mengelola akun pengguna, baik peserta maupun penyelenggara.
- e) Admin dapat melihat dan memantau seluruh aktivitas event yang berlangsung di platform.
- f) Sistem mendukung unggah dan unduh dokumen lomba (misalnya syarat pendaftaran atau hasil seleksi).
- g) Penyelenggara dapat melihat dan mengelola data peserta untuk masing-masing event.
- h) Tersedia fitur pencarian dan filter berdasarkan kategori lomba, lokasi, atau tanggal pelaksanaan.
- i) Sistem menyediakan dashboard bagi masing-masing peran (admin, peserta, penyelenggara) untuk memantau data dan aktivitas mereka.

### **Analisis Kebutuhan Non Fungsional**

#### a) Analisis kebutuhan perangkat lunak

Untuk menunjang kinerja hardware maka diperlukan adanya perangkat lunak yang sesuai dengan hardware dan sistem yang dikembangkan. perangkat lunak yang diperlukan yaitu:

Sistem Operasi: Windows 11

Software: Laragon

Database: libmysql - mysqlnd 8.3.8

Bahasa Pemrograman: PHP 8.3.8

Browser: Google Chrome Browser

#### b) Analisis kebutuhan perangkat keras

Perangkat keras harus mendukung proses pengembangan, pengujian, dan penggunaan sistem dengan baik. Berikut adalah kebutuhan perangkat keras:

Satu unit Komputer (Laptop)

Dengan processor minimal Processor Intel Celeron N6210 (1.2 GHz)

Dilengkapi dengan webcam.

Satu unit Ponsel

Dengan sistem operasi Android Nougat (v.7.0+) atau di atasnya,

Dengan spesifikasi RAM minimal 1 GB

Menggunakan RAM minimal 2GB.

Harddisk data dengan kapasitas penyimpanan minimal 500 GB.

### c) Analisis Kebutuhan Operasional

Kebutuhan operasional meliputi kesediaan sumber daya manusia maupun infrastruktur yang mendukung untuk menerapkan sistem yang baru, kebutuhan operasional tersebut yaitu :  
Akses Admin:

Pengguna yang memiliki akses penuh terhadap sistem Eventify. Admin bertanggung jawab untuk mengelola data pengguna, memantau seluruh aktivitas dalam sistem, serta memastikan sistem berjalan dengan baik dan lancar.

#### Pendaftar (Peserta)

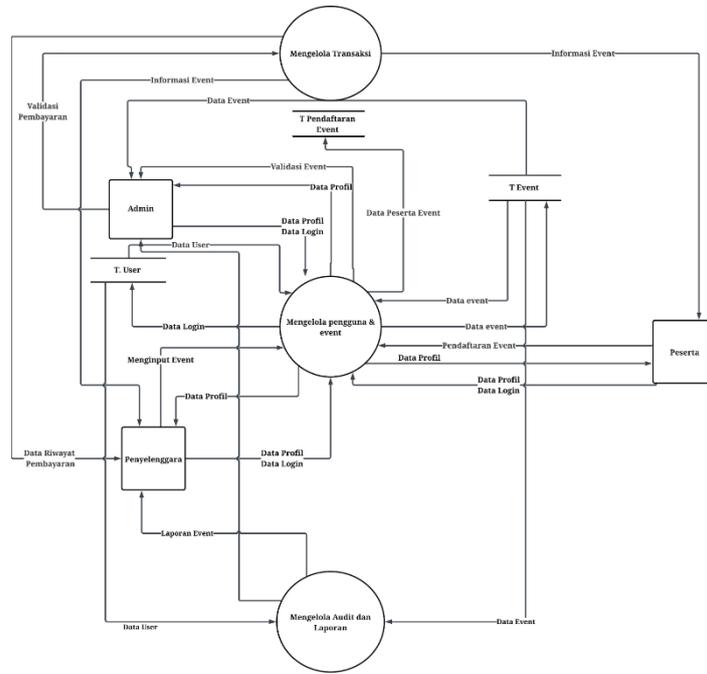
Pengguna yang menggunakan sistem untuk mencari dan mendaftar lomba. Peserta dapat melihat detail event, mengunggah dokumen pendaftaran, serta menerima notifikasi dan pengumuman melalui platform Eventify.

#### Penyelenggara

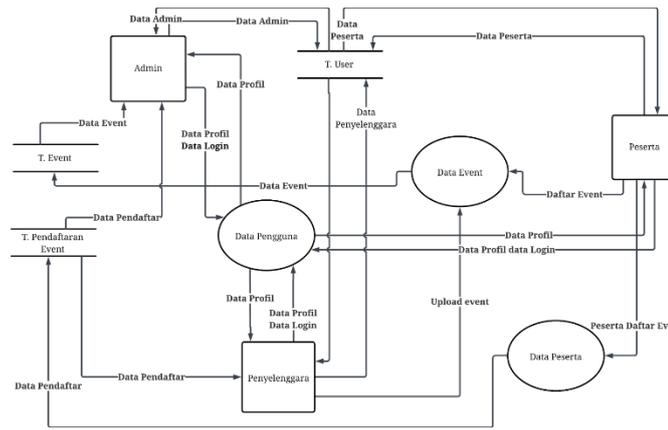
Pengguna yang membuat dan mengelola event lomba. Penyelenggara dapat memasukkan informasi lomba, memverifikasi data peserta, mengelola jadwal, serta mengumumkan hasil lomba kepada peserta.

### **Perancangan Sistem**

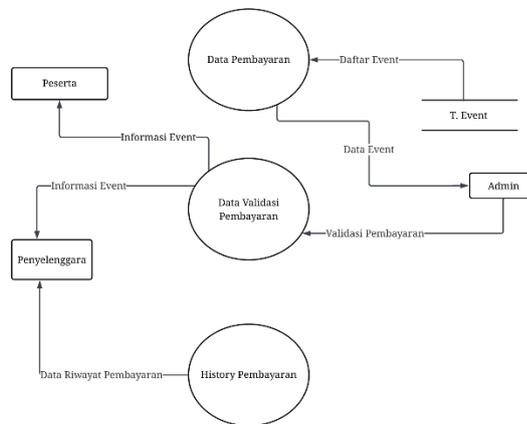
Data Flow Diagram (DFD) berfungsi sebagai model logis untuk memvisualisasikan perjalanan data dalam sistem, mengidentifikasi asal dan tujuan data yang keluar, lokasi penyimpanan data, proses yang menghasilkan data, serta bagaimana data tersimpan berinteraksi dengan proses yang menggunakannya (Paillina & Widiatmoko, 2021). Berikut adalah data flow diagram dari aplikasi eventify.



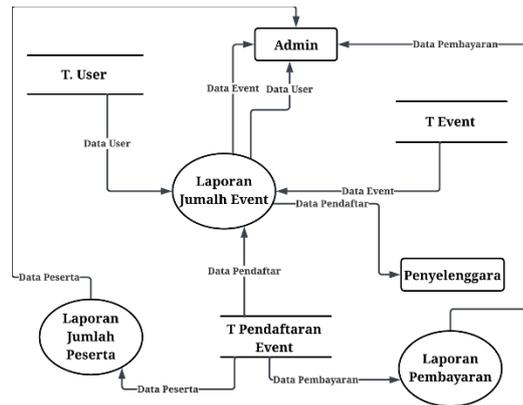
### DFD Level 1 Mengelola Pengguna & Event



### DFD Level 1 Mengelola Transaksi



## DFD Level 1 Mengelola Audit dan Laporan



## Implementasi Sistem

Implementasi sistem merupakan tahap akhir dalam proses pengembangan sistem, di mana hasil dari analisis, dan perancangan diterapkan dalam bentuk sistem yang dapat digunakan dengan baik oleh pengguna. Dalam pengembangan Aplikasi Eventify, implementasi dilakukan secara bertahap berdasarkan hasil dari tahapan sebelumnya yang telah dilakukan menggunakan metode SDLC model Waterfall. Tahap ini menjadi bagian penting dari proses pengembangan sistem yang melibatkan pengintegrasian komponen-komponen sistem yang telah dirancang sebelumnya ke dalam bentuk aplikasi yang fungsional. Pada tahap ini, aplikasi Eventify dibangun dengan menggunakan framework Laravel dan pendekatan integrasi langsung ke database melalui konfigurasi multi-connection di file .env

## Arsitektur Sistem

Platform Eventify terdiri dari dua aplikasi utama yang saling terintegrasi:

- Eventify Main App: aplikasi utama yang digunakan oleh peserta dan penyelenggara untuk melihat, mendaftar, dan mengupload event.
- Eventify Audit Dashboard: sistem backend yang digunakan oleh admin untuk memantau aktivitas sistem, mengelola data pengguna, dan melakukan konfirmasi pada event yang di upload penyelenggara

Kedua aplikasi ini berjalan secara independen namun terhubung ke dua database berbeda yang disimpan dalam satu server MySQL. Integrasi data dilakukan dengan memanfaatkan fitur multi-database connection pada Laravel.

### 1. Konfigurasi Langsung ke Database

Dalam proses pengembangan sistem Eventify, digunakan dua aplikasi terpisah: Eventify Main App dan Eventify Admin Dashboard. Keduanya dikembangkan menggunakan framework Laravel dan membutuhkan akses ke dua database yang berbeda, yaitu eventify

sebagai database utama, dan eventify\_admin sebagai database untuk keperluan administrasi dan audit.

Untuk mengintegrasikan kedua sistem tersebut, digunakan pendekatan koneksi langsung antar database. Pendekatan ini dipilih karena dinilai lebih sederhana dan efisien untuk sistem yang masih berada dalam satu server atau satu jaringan lokal (intranet).

#### a) Konfigurasi pada file .env

```
'connections' => [
    'mysql' => [ // default
        'driver' => 'mysql',
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'eventify'),
        'username' => env('DB_USERNAME', 'root'),
        'password' => env('DB_PASSWORD', ''),
    ],
    'admin_db' => [
        'driver' => 'mysql',
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('ADMIN_DB_DATABASE', 'eventify_admin'),
        'username' => env('ADMIN_DB_USERNAME', 'root'),
        'password' => env('ADMIN_DB_PASSWORD', ''),
    ],
    'mariadb' => [
        'driver' => 'mariadb',
        'url' => env('DB_URL'),
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'laravel'),
        'username' => env('DB_USERNAME', 'root'),
        'password' => env('DB_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => env('DB_CHARSET', 'utf8mb4'),
        'collation' => env('DB_COLLATION', 'utf8mb4_unicode_ci'),
        'prefix' => ''
    ]
]
```

Variabel-variabel ini nantinya akan dipanggil dalam konfigurasi Laravel agar dapat mengakses database secara langsung sesuai peran masing-masing aplikasi.

#### b) Konfigurasi pada config/database.php

Setelah mendefinisikan variabel lingkungan di file .env, langkah selanjutnya adalah mengonfigurasi koneksi database di file config/database.php. Laravel menyediakan fungsionalitas untuk mendefinisikan beberapa koneksi database. Dengan ini, kita dapat membuat entri terpisah untuk database eventify dan eventify\_admin, masing-masing mengambil nilai dari variabel .env yang relevan.

```
'mariadb' => [
  'driver' => 'mariadb',
  'url' => env('DB_URL'),
  'host' => env('DB_HOST', '127.0.0.1'),
  'port' => env('DB_PORT', '3306'),
  'database' => env('DB_DATABASE', 'laravel'),
  'username' => env('DB_USERNAME', 'root'),
  'password' => env('DB_PASSWORD', ''),
  'unix_socket' => env('DB_SOCKET', ''),
  'charset' => env('DB_CHARSET', 'utf8mb4'),
  'collation' => env('DB_COLLATION', 'utf8mb4_unicode_ci'),
  'prefix' => '',
  'prefix_indexes' => true,
  'strict' => true,
  'engine' => null,
  'options' => extension_loaded('pdo_mysql') ? array_filter([
    PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
  ]) : [],
],

'pgsql' => [
  'driver' => 'pgsql',
  'url' => env('DB_URL'),
  'host' => env('DB_HOST', '127.0.0.1'),
  'port' => env('DB_PORT', '5432'),
  'database' => env('DB_DATABASE', 'laravel'),
  'username' => env('DB_USERNAME', 'root'),
  'password' => env('DB_PASSWORD', ''),
  'charset' => env('DB_CHARSET', 'utf8'),
  'prefix' => '',
  'prefix_indexes' => true,
  'search_path' => 'public',
  'sslmode' => 'prefer',
],
```

Pada konfigurasi di atas, kami menambahkan entri baru `admin_db` di bawah bagian `'connections'`. Ini akan menjadi nama koneksi yang dapat dirujuk saat melakukan operasi database untuk database `eventify_admin`

### c) Implementasi model multi database

Setelah konfigurasi koneksi selesai, kita dapat mengarahkan model Eloquent untuk menggunakan koneksi database tertentu. Ini dilakukan dengan mendefinisikan properti `$connection` pada model.

Untuk model `User`, karena ia akan menggunakan database utama (`eventify`) yang merupakan koneksi default (yaitu `'mysql'`), Anda tidak perlu secara eksplisit mendefinisikan properti `$connection`. Laravel akan secara otomatis menggunakan koneksi default.

```

pp > Models > User.php
1 </php
2
3 namespace App\Models;
4
5 // use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9
10 class User extends Authenticatable
11 {
12     /** @use HasFactory<Database\Factories\UserFactory> */
13     use HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var list<string>
19      */
20     protected $fillable = [
21         'name',
22         'email',
23         'password',
24         'role',
25         'profile_photo_path',
26     ];
27
28     /**
29      * The attributes that should be hidden for serialization.
30      *
31      * @var list<string>
32      */
33     protected $hidden = [
34         'password',
35         'remember_token',
36     ];
37

```

Untuk model User yang digunakan di Eventify Admin Dashboard, penting untuk secara eksplisit mengarahkan model ke koneksi database admin\_db. Meskipun bisa menggunakan sintaks database.table seperti `protected $table = 'eventify_admin.users';`, praktik terbaik Laravel adalah dengan mendefinisikan properti `$connection` pada model. Ini membuat kode lebih bersih dan fleksibel.

```

Windsurf: Refactor | Explain
10 class User extends Authenticatable
11 {
12     /** @use HasFactory<Database\Factories\UserFactory> */
13     use HasFactory, Notifiable;
14
15     // --- TAMBAHKAN BARIS DI BAWAH INI ---
16     /**
17      * Memberitahu Eloquent secara eksplisit nama database dan tabel untuk model ini.
18      * Ini akan menyelesaikan masalah relasi antar-database.
19      */
20     protected $table = 'eventify_admin.users';
21     // -----
22
23     /**
24      * The attributes that are mass assignable.
25      *
26      * @var list<string>
27      */
28     protected $fillable = [
29         'name',
30         'email',
31         'password',
32     ];
33
34     /**
35      * The attributes that should be hidden for serialization.
36      *
37      * @var list<string>
38      */
39     protected $hidden = [
40         'password',

```

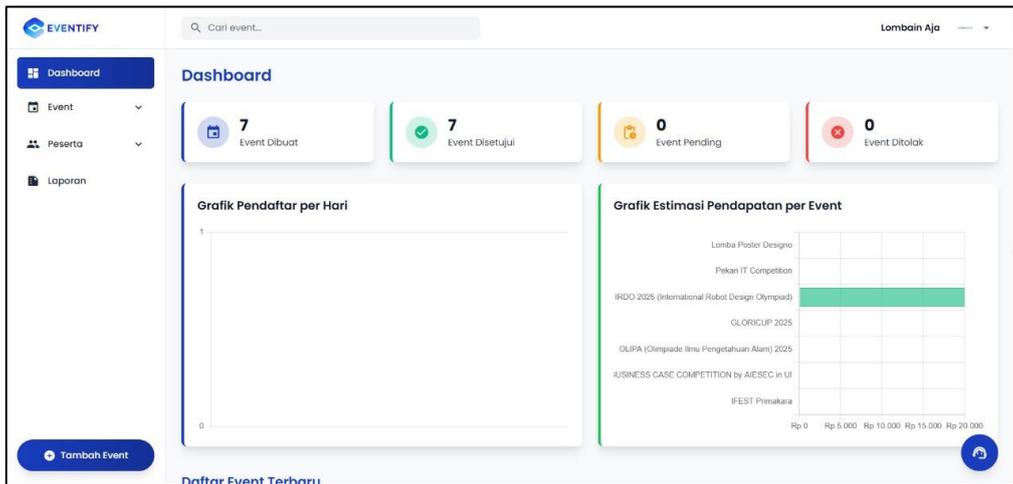
Dengan pendekatan ini, setiap model secara otomatis akan menggunakan koneksi database yang ditentukan, memastikan bahwa operasi CRUD (Create, Read, Update, Delete) diarahkan ke database yang benar. Ini memungkinkan kedua aplikasi untuk berbagi server database fisik yang sama, tetapi secara logis terisolasi ke skema database masing-masing, memfasilitasi manajemen data yang terorganisir dan efisien.

Pendekatan koneksi langsung ini sangat cocok untuk sistem yang berjalan dalam satu lingkungan server, menyediakan solusi yang ringkas dan mudah dikelola untuk arsitektur multi-database.

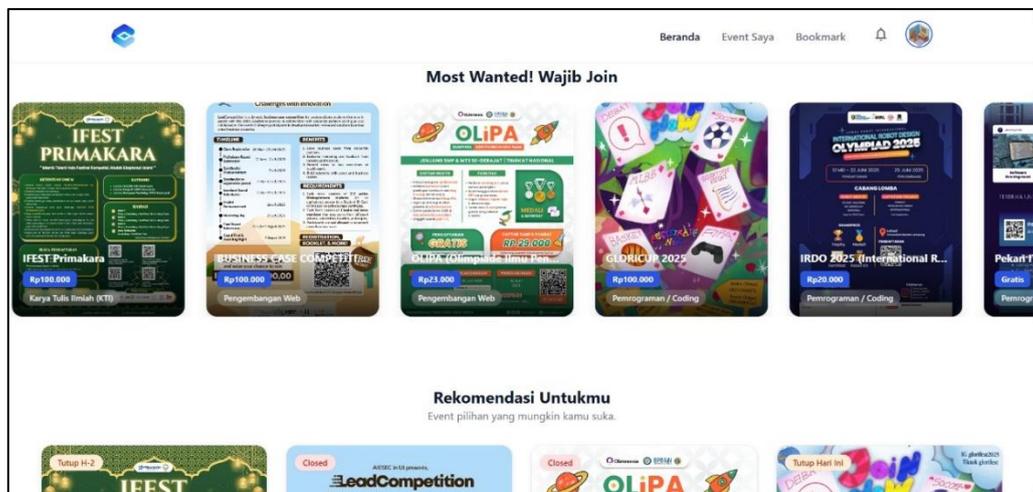
## Tampilan Antarmuka Pengguna

Antarmuka pengguna dirancang responsif dan mudah digunakan, dengan pendekatan mobile-first. Beberapa antarmuka utamanya adalah :

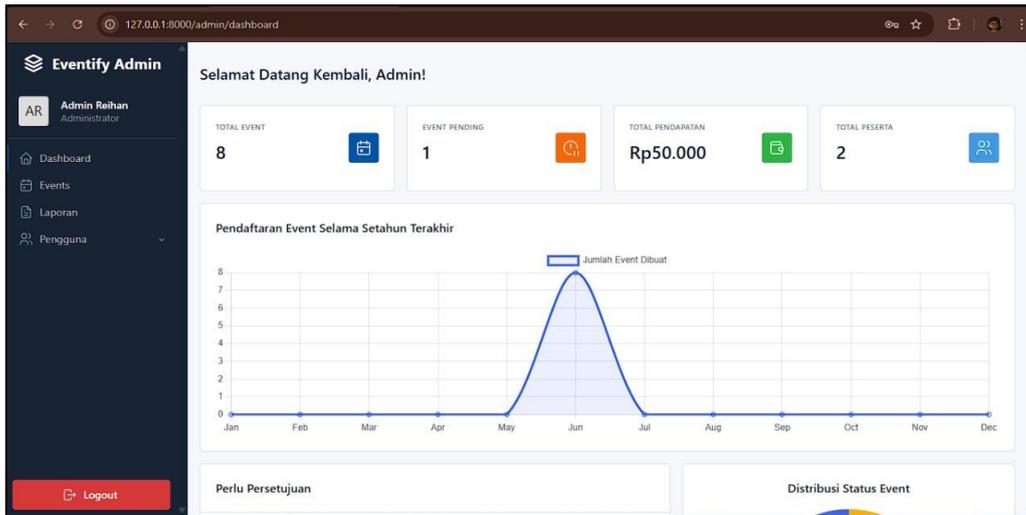
- Halaman dashboard organizer



- Halaman Dashboard Peserta



- Halaman Dashboard Admin



- Halaman Pendaftaran



- Halaman upload event

The 'Buat Event Baru Anda' form is divided into two main sections:

- Informasi Utama Event:** Includes fields for 'Judul Event\*' (with a placeholder example), 'Deskripsi Event\*' (with a placeholder instruction), and 'Poster Event\*' (with an upload icon and instructions: 'Klik untuk memilih atau seret file ke sini', 'JPG, PNG, Ukuran maksimal 2MB').
- Detail Logistik & Pendaftaran:** Includes a 'Kategori\*' dropdown, 'Jenjang Peserta\*' (with options: SD, SMP, SMA, Mahasiswa, Umum), 'Tanggal Event\*' and 'Buka Pendaftaran\*' date pickers, and 'Tutup Pendaftaran\*' date picker.

At the bottom right, there is a link for 'Informasi Pembayaran'.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan Aplikasi Eventify, sebuah sistem web terintegrasi antara Main App dan Admin Dashboard, yang bertujuan untuk mempermudah manajemen event lomba dan kompetisi. Sistem ini dirancang untuk mengatasi kendala penyebaran informasi, pendaftaran manual, dan pengelolaan data yang tidak efisien, dengan menyediakan fitur-fitur fungsional bagi peserta, penyelenggara, dan admin. Implementasi dilakukan menggunakan framework Laravel dengan pendekatan koneksi langsung antar database, yang terbukti menyederhanakan arsitektur sistem dan meningkatkan efisiensi komunikasi data.

Meskipun sistem ini telah meningkatkan transparansi dan kecepatan dalam administrasi lomba, terdapat beberapa keterbatasan. Penelitian ini berfokus pada integrasi database dalam satu server atau jaringan lokal. Untuk pengembangan selanjutnya, disarankan untuk mengeksplorasi integrasi dengan REST API atau layanan web untuk skenario di mana aplikasi tersebar di server yang berbeda atau membutuhkan interaksi dengan sistem eksternal. Selain itu, pengembangan fitur notifikasi yang lebih komprehensif, seperti notifikasi email otomatis atau integrasi dengan aplikasi pesan instan, dapat lebih meningkatkan pengalaman pengguna. Penelitian lebih lanjut juga dapat dilakukan untuk menguji skalabilitas sistem dengan jumlah pengguna dan event yang sangat besar.

## DAFTAR REFERENSI

- Al Qossam, F. (2024). *Rancang bangun aplikasi web seminar berbasis CMS dengan email reminder menggunakan framework Laravel* (Skripsi, Sekolah Tinggi Teknologi Terpadu Nurul Fikri).
- Ardiansyah, D., Suryana, A., & Fatimah, A. (2021). Perancangan sistem informasi manajemen event berbasis web pada Fakultas Ilmu Administrasi. *Jurnal Teknologi Informasi dan Komputer*, 7(2), 115–123.
- Bagir, M. H. (2018). Analisis perancangan sistem informasi pergudangan di CV. Karya Nugraha. *JMTSI Jurnal Media Teknik dan Sistem Industri*, 2(1), 20–29. <https://doi.org/10.35194/jmtsi.v2i1.274>
- Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *System analysis and design: An object-oriented approach with UML* (5th ed.). Journal of Chemical Information and Modeling. <https://doi.org/10.1021/la061259j>
- Hasugian, P. S. (2018). Perancangan website sebagai media promosi dan informasi. *Journal of Informatic Pelita Nusantara*, 3(1), 82–86.
- Paillina, D. B., & Widiatmoko, Y. (2021). Rancangan aplikasi monitoring online untuk meningkatkan pemeliharaan prediktif pada PLTD. *Jurnal Sistem Informasi Bisnis*, 11(1), 9–17.

- Pratama, R. A., & Yuniarti, D. R. (2022). Rancang bangun sistem informasi manajemen event berbasis web. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK)*, 6(2), 8576–8583. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/8576>
- Putra, W. A., Fitri, I., & Hidayatullah, D. (2022). Implementasi waterfall dan agile dalam perancangan e-commerce alat musik berbasis website. *JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 6(1), 56–62. <https://doi.org/10.35870/jtik.v6i1.380>
- Ramadhan, I., & Hartati, S. (2021). Sistem informasi akademik berbasis web dengan koneksi langsung antar database. *Jurnal Teknologi dan Sistem Informasi*, 10(1), 34–42.
- Rizky, A., & Aditama, D. (2020). Implementasi koneksi multi database pada Laravel framework untuk sistem terintegrasi. *Jurnal Teknologi Informasi dan Komputer*, 8(2), 76–84.
- Setiyani, L., & Tjandra, E. (2024). Analisis kebutuhan fungsional aplikasi penanganan keluhan mahasiswa (Studi kasus: STMIK Rosma Karawang). *Jurnal Inovasi Pendidikan dan Teknologi Informasi*. <https://ejournal.ummuba.ac.id/index.php/JIPTI/article/view/465/264>
- Syahputra, R., & Wahyuni, I. (2020). Analisis pemanfaatan framework Laravel dalam pengembangan aplikasi web. *Jurnal Ilmiah Teknologi dan Rekayasa*, 25(1), 12–18. <https://doi.org/10.31294/jtr.v25i1.7891>
- Yusuf, M., & Kurniawan, R. (2022). Integrasi aplikasi keuangan sekolah menggunakan pendekatan multi database connection. *Jurnal Rekayasa Sistem dan Teknologi Informasi*, 4(1), 57–64.