

Penerapan Database Pengembangan Sistem Penjualan Tiket Acara di Kota Bandung

1st Muhamad Ibnu Fajar Wibawa
Computer Engineering, Faculty of
Electrical Engineering
Telkom University
Bandung, Indonesia
mibunfajarw@student.telkomuniversity
.ac.id

2nd Meta Kallista
Computer Engineering, Faculty of
Electrical Engineering
Telkom University
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

3rd Muhammad Faris Ruriawan
Computer Engineering, Faculty of
Electrical Engineering
Telkom University
Bandung, Indonesia
muhammadfaris@telkomuniversity.ac.id

Abstrak— Acara adalah suatu kejadian yang direncanakan atau diatur dengan tujuan tertentu, seringkali untuk memberikan hiburan, pendidikan, atau pengalaman sosial kepada peserta. Di Kota Bandung, di mana peminat acara sangat banyak, terutama yang menantikan acara favorit mereka, seringkali ditemui kendala dalam pembelian tiket acara. Biasanya, orang harus mengantri di lokasi penjualan tiket fisik, yang tentu saja bisa menjadi tugas yang melelahkan dan memakan waktu.

Untuk mengatasi masalah ini, pengembang memutuskan untuk menggunakan metode pengembangan website berdasarkan Waterfall, yang merupakan pendekatan berjenjang untuk merancang dan mengembangkan solusi penyediaan tiket acara secara online. Bagian database menjadi elemen penting dalam pengembangan ini, dengan penggunaan MySQL sebagai sistem manajemen basis data. Pengelolaan data acara termasuk ke dalam fokus utama, dengan fitur untuk menambahkan dan mengelola informasi acara secara efektif. Pengguna dapat membuat akun, masuk, dan mengelola informasi pribadi mereka. Sistem otentikasi dan otorisasi yang kuat diterapkan untuk memastikan keamanan dan privasi pengguna.

Penelitian pengembang terhadap peminat acara di Kota Bandung menghasilkan wawasan yang berharga, yang membantu pengembang dalam merancang struktur database berdasarkan kebutuhan dan preferensi mereka. Dengan ini, pengembang berharap memberikan kemudahan kepada peminat acara di Kota Bandung untuk memesan tiket secara online, menghilangkan hambatan antrean, dan membuat pengalaman pembelian tiket mereka menjadi lebih cepat dan nyaman.

Kata kunci: Acara, Pembelian Tiket, Pengelolaan Data Acara, Struktur database, MySQL, Waterfall

I. PENDAHULUAN

Pertumbuhan industri hiburan di Kota Bandung telah menjadi sorotan utama dalam beberapa tahun terakhir. Semakin banyaknya acara seperti konser, pameran seni, dan seminar penting menuntut efisiensi dalam penjualan tiket. Namun, penyelenggara acara sering mengalami kesulitan dalam mempromosikan dan mengelola penjualan tiket. Persiapan acara bukanlah tugas yang mudah, seringkali terhambat oleh masalah teknis seperti penyebaran tiket ilegal dengan harga yang tidak sah[9][10].

Kendala dalam penjualan tiket di Kota Bandung mencakup antrean panjang, promosi acara, dan sulitnya memantau data pembeli tiket. Untuk mengatasi tantangan ini, pengembang berencana untuk menciptakan sistem penjualan tiket yang inovatif. Sistem ini akan membantu penyelenggara acara mengelola penjualan tiket dengan efisien, memantau

data pembeli tiket, dan meningkatkan transparansi alokasi tiket. Diharapkan, solusi ini akan memperkuat posisi Kota Bandung sebagai pusat hiburan dan budaya yang berkembang pesat.

Usulan solusi atas permasalahan tersebut adalah dengan mengembangkan Sistem Penjualan Tiket Acara di Kota Bandung, sistem ini akan menggunakan database MySQL yang mampu mengelola data pengguna sehingga sistem pada situs web dapat berkerja dengan semestinya.

II. KAJIAN TEORI

A. Sistem

Sistem adalah gabungan komponen yang terhubung untuk aliran informasi atau energi menuju tujuan. Ini juga mencakup teknologi informasi dan aktivitas orang yang menggunakannya untuk dukungan operasi dan manajemen. Istilah sistem informasi merujuk pada interaksi antara orang, proses, data, dan teknologi. Ini tidak hanya mengacu pada penggunaan teknologi informasi dan komunikasi dalam organisasi, tetapi juga pada cara orang berinteraksi dengan teknologi ini dalam mendukung bisnis [1].

B. Basis data

Basis data adalah kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengkonsolidasi banyak catatan yang sebelumnya disimpan dalam file terpisah. Basis data adalah kumpulan file-file yang saling berelasi, maka dari itu dapat disimpulkan basis data adalah koleksi dari data terkait yang formatnya standar dan dirancang untuk bisa diakses beberapa pengguna [2].

C. MySQL

MySQL adalah salah satu sistem manajemen basis data (DBMS) yang paling terkenal dan digunakan secara luas. Keunggulan MySQL dapat ditelusuri ke penggunaan SQL (Structured Query Language) sebagai bahasa standarnya untuk mengakses dan mengelola basis data. MySQL adalah contoh dari RDBMS (Relational Database Management System) yang mengorganisasi data dalam bentuk tabel yang terdiri dari baris dan kolom.

Dalam MySQL, sebuah basis data dapat berisi satu atau lebih tabel, dengan setiap tabel memiliki berbagai baris yang menyimpan data dan beberapa kolom yang mendefinisikan atribut-atribut dari data tersebut. Pengelolaan basis data MySQL dapat dilakukan melalui berbagai cara, salah satunya adalah melalui prompt DOS atau alat baris perintah

(command-line tool) yang memungkinkan pengguna untuk berinteraksi langsung dengan sistem basis data [3][4]. MySQL memungkinkan pengguna untuk melakukan berbagai tugas, seperti membuat, mengubah, dan menghapus tabel, serta menjalankan kueri SQL kompleks untuk mengambil dan memanipulasi data.

D. Web

Web adalah salah satu layanan yang dinikmati oleh pengguna komputer yang terhubung ke internet. Sebuah situs web, atau juga disebut sebagai website, dapat diartikan sebagai kumpulan halaman yang bertujuan untuk menampilkan berbagai informasi berupa teks, gambar diam atau bergerak, animasi, suara, dan kombinasi dari elemen-elemen tersebut. Konten di dalam situs web bisa bersifat statis (tetap) atau dinamis (berubah-ubah), yang membentuk sebuah struktur yang saling terhubung di mana setiap halaman memiliki keterkaitan dengan halaman lainnya[5].

E. PHP

PHP atau PHP Hypertext Preprocessor, yang dikenal dengan sebutan PHP, merupakan sebuah bahasa pemrograman yang digunakan untuk membangun situs web yang dinamis. PHP terintegrasi dengan kode HTML, yang berarti bahwa keduanya beroperasi dalam konteks yang berbeda. HTML berfungsi sebagai dasar untuk merangkai tampilan visual suatu situs web, sementara PHP digunakan untuk mengelola proses di belakang layar. Dengan adanya PHP, pemeliharaan sebuah situs web menjadi lebih mudah dilakukan. PHP beroperasi di sisi server, menjadikannya sebagai bahasa pemrograman server-side scripting. Ini berarti bahwa untuk menjalankan kode PHP, diperlukan adanya sebuah server[6].

F. Javascript

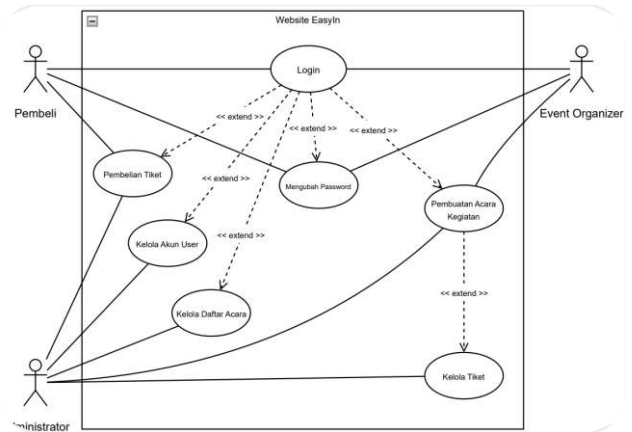
Javascript adalah bahasa skrip yang berjalan dalam dokumen HTML. Sebagai bahasa skrip web pertama, JavaScript memberikan kemampuan ekstra pada HTML dengan mengeksekusi perintah di sisi pengguna, dalam hal ini di browser, bukan di server web. JavaScript tergantung pada browser yang memuat halaman web dengan skrip-skripnya, yang terintegrasi dalam dokumen HTML[7].

G. HTML

HTML, singkatan dari Hyper Text Markup Language, adalah serangkaian kode program yang menjadi dasar dari representasi visual halaman web. Kode HTML ini berisi informasi yang disimpan dalam tag-tag khusus, yang digunakan untuk memformat konten. Meskipun telah ada banyak perkembangan dalam pemrograman web, HTML tetap menjadi dasar bagi bahasa-bahasa web seperti PHP, ASP, dan JSP. Bahkan hingga saat ini, sebagian besar situs web di internet masih menggunakan HTML sebagai teknologi utama. Penggunaan HTML fokus pada membangun tampilan website dengan pendekatan semantik, untuk memudahkan pengembangan dan pemeliharaan [8].

III. PERANCANGAN SISTEM

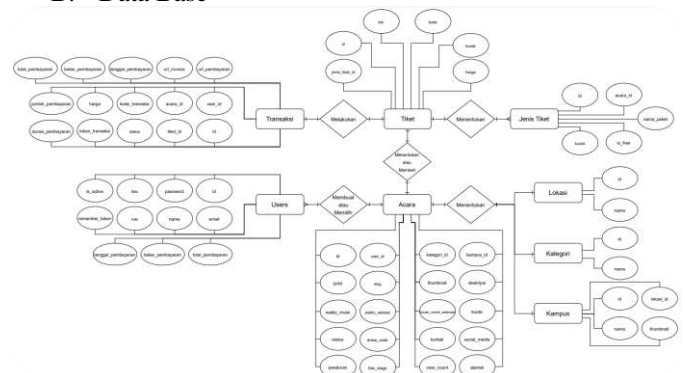
A. Use Case Diagram



Gambar 1 Use Case Diagram

Berdasarkan desain Use Case pada gambar 1 di atas, dapat dijelaskan secara singkat sebagai berikut. Sistem melibatkan beberapa pihak yang berinteraksi, termasuk Pembeli, administrator dan event organizer. Pembeli berperan sebagai pengguna yang dapat melakukan login, mengubah kata sandi, membeli tiket. Event organizer berperan sebagai pengguna yang dapat melakukan login, mengubah kata sandi dan pembuatan acara kegiatan, sementara itu administrator memiliki akses login yang lebih luas, termasuk pembelian tiket, kelola akun user, kelola daftar acara, pembuatan kegiatan acara dan kelola tiket.

B. Data Base



Gambar 2 Database

Pada Gambar 3.15, mengenai ERD database yang menunjukkan hubungan setiap database. Dapat dilihat pada diagram bahwa user dapat berperan sebagai Event Organizer atau Pembeli Tiket yang tentu saja user disini dapat membuat/memilih Acara, dan jika user tersebut adalah berperan sebagai Event Organizer, user ini membuat acara, user itu hanya dapat menentukan satu lokasi, satu kategori, satu kampus, dan menentukan tiket dan dapat menentukan beberapa jenis tiket. Sedangkan jika user tersebut adalah berperan sebagai Pembeli, user ini akan langsung menentukan satu tiket dan dapat melakukan satu transaksi tiket tersebut, hal tersebut dapat dikatakan proses user sedang membeli tiket acara.

IV. HASIL DAN PEMBAHASAN

A. Database

a. Tabel User

Dalam database users memiliki beberapa atribut yaitu: id, nama, foto, email, email_verified_at, password, role,

reset_password_token, verify_email_token, is_active, remember_token. Untuk keterangan dapat dilihat dalam Tabel 1 berikut

Tabel 1 Tabel User

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik pengguna
	nama	Varchar(255)	Nama pengguna
	foto	Varchar(255)	Nama file foto profil
	email	Varchar(255)	Alamat email pengguna
	email_verified_at	Timestamp	Waktu verifikasi email
	password	Varchar(255)	Kata sandi pengguna
	role	Enum	Peran pengguna ("admin" atau "user")
	reset_password_token	Varchar(255)	Token reset kata sandi
	verify_email_token	Varchar(255)	Token verifikasi email
	is_active	Tinyint(1)	Status akun aktif (1 atau 0)
	remember_token	Varchar(100)	Token ingat kata sandi

b. Tabel Acara

Dalam database acara memiliki beberapa atribut yaitu: id, kategori_id, user_id, kampus_id, judul, slug, thumbnail, deskripsi, waktu_mulai, waktu_selesai, durasi_menit_estimasi, kuota, status, dress_code, kontak, social_media, peraturan, foto_stage, view_count, dan alamat. Untuk keterangan dapat dilihat dalam Tabel 2 berikut:

Tabel 2 Tabel Acara

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik acara
Foreign Key	kategori_id	BigInt(20) UNSIGNED	ID kategori acara
Foreign Key	user_id	BigInt(20) UNSIGNED	ID pengguna yang membuat acara
	kampus_id	BigInt(20) UNSIGNED	ID kampus terkait (opsional)
	judul	Varchar(255)	Judul acara
	slug	Varchar(255)	Slug acara (URL bersih)
	thumbnail	Varchar(255)	Nama file thumbnail gambar (opsional)

	deskripsi	Text	Deskripsi acara
	waktu_mulai	Datetime	Waktu mulai acara
	waktu_selesai	Datetime	Waktu selesai acara
	durasi_menit_estimasi	Int(11)	Durasi estimasi dalam menit (opsional)
	kuota	Int(11)	Kuota peserta acara
	status	Enum	Status acara ("draft", "published", "rejected")
	dress_code	Varchar(255)	Kode berpakaian untuk acara (opsional)
	kontak	Varchar(255)	Informasi kontak untuk acara (opsional)
	social_media	Varchar(255)	Media sosial acara (opsional)
	peraturan	Text	Peraturan acara (opsional)
	foto_stage	Varchar(255)	Nama file gambar panggung (opsional)
	view_count	Int(11)	Jumlah tampilan acara
	alamat	Text	Alamat acara (opsional)

c. Tabel Lokasi

Dalam database lokasi memiliki beberapa atribut yaitu: id, nama. Untuk keterangan dapat dilihat dalam Tabel 3 berikut:

Tabel 3 Tabel Lokasi

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik lokasi
	nama	Varchar(255)	Nama lokasi

d. Tabel Kategori

Dalam database kategori memiliki beberapa atribut yaitu: id, nama, dan thumbnail. Untuk keterangan dapat dilihat dalam Tabel 4 berikut:

Tabel 4 Tabel Kategori

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik kategori
	nama	Varchar(255)	Nama kategori
	thumbnail	Varchar(255)	Nama file thumbnail gambar

e. Tabel Kampus

Dalam database kampus memiliki beberapa atribut yaitu: id, lokasi_id, thumbnail, dan nama. Untuk keterangan

dapat dilihat dalam Tabel 5 berikut:

Tabel 5 Tabel Kampus

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik kampus
	lokasi_id	BigInt(20) UNSIGNED	ID lokasi terkait (opsional)
	thumbnail	Varchar(255)	Nama file thumbnail gambar (opsional)
	nama	Varchar(255)	Nama kampus

f. Tabel Tiket

Dalam database tiket memiliki beberapa atribut yaitu: id, jenis_tiket_id, kode, tier, kuota, dan harga. Untuk keterangan dapat dilihat dalam Tabel 6 berikut:

Tabel 6 Tabel Tiket

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik jenis tiket
Foreign Key	jenis_tiket_id	BigInt(20) UNSIGNED	ID jenis tiket terkait
	kode	Varchar(255)	Kode tiket
	tier	Varchar(255)	Tier tiket (opsional)
	kuota	Varchar(255)	Kuota tiket (opsional)
	harga	BigInt(20)	Harga tiket

g. Tabel Jenis Tiket

Dalam database jenis_tiket memiliki beberapa atribut yaitu: id, acara_id, nama_paket, kuota, dan is_free. Untuk keterangan dapat dilihat dalam Tabel 7 berikut:

Tabel 7 Tabel Jenis Tiket

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik jenis tiket
Foreign Key	acara_id	BigInt(20) UNSIGNED	ID acara terkait
	nama_paket	Varchar(255)	Nama paket tiket
	kuota	Int(11)	Kuota tiket
	is_free	Tinyint(1)	Status tiket gratis atau tidak (1 atau 0)

h. Tabel Transaksi

Dalam database transaksis memiliki beberapa atribut yaitu: id, tiket_id, acara_id, user_id, kode_transaksi, tanggal_pembayaran, jumlah_dibayar, bukti_transfer, harga, batas_pembayaran, total_pembayaran, dan status. Untuk keterangan dapat dilihat dalam Tabel 8 berikut:

Tabel 8 Tabel Transaksi

Kedudukan	Nama Kolom	Tipe Data	Keterangan
Primary Key	id	BigInt(20) UNSIGNED	Identifikasi unik transaksi
Foreign Key	tiket_id	BigInt(20) UNSIGNED	ID tiket terkait

		D	
Foreign Key	acara_id	BigInt(20) UNSIGNED	ID acara terkait
Foreign Key	user_id	BigInt(20) UNSIGNED	ID pengguna yang melakukan transaksi
	kode_transaksi	Varchar(255)	Kode transaksi
	tanggal_pembayaran	Datetime	Waktu pembayaran transaksi
	url_pembayaran	Varchar(255)	URL pembayaran
	url_invoice	Varchar(255)	URL invoice
	durasi_pembayaran	Varchar(255)	Durasi pembayaran dalam menit
	jumlah_dibayar	Varchar(255)	Jumlah yang harus dibayarkan
	bukti_transfer	Varchar(255)	Nama file bukti transfer
	harga	Varchar(255)	Harga tiket
	batas_pembayaran	Varchar(255)	Batas waktu pembayaran
	total_pembayaran	Varchar(255)	Total pembayaran
	status	Enum	Status transaksi ("MENUNGGU", "DIBAYAR", "DITOLAK", "KADALUWARS A, GAGAL.)

B. Implementasi

a. Koneksi Basis Data

Pada implementasi ini, pengaturan koneksi basis data didefinisikan dalam file (.env). Ini adalah file konfigurasi yang berisi detail koneksi seperti host, port, database name, username, dan password.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nama_database
DB_USERNAME=username
DB_PASSWORD=password
```

Konfigurasi ini memungkinkan website untuk menjalin koneksi yang aman dan terenkripsi dengan basis data yang ditentukan. Setelah konfigurasi ini ditetapkan, Laravel akan menggunakan nilai-nilai ini untuk menghubungkan proyek dengan basis data yang sesuai.

b. Migration

Selain konfigurasi koneksi, implementasi database juga melibatkan penggunaan migrasi, yang dimana ini adalah fitur Laravel yang memungkinkan untuk mendefinisikan struktur tabel dalam basis data menggunakan kode PHP.

c. Tabel users

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel "users" adalah sebagai berikut:

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
```

```

use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('users', function (Blueprint
$table) {
            $table->id();
            $table->string('nama');
            $table->string('foto')->nullable();
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')-
>nullable();
            $table->string('password');
            $table->enum('role', ['admin', 'user'])-
>default('user');
            $table->string('reset_password_token')-
>nullable();
            $table->string('verify_email_token')-
>nullable();
            $table->boolean('is_active')->default(false);
            $table->rememberToken();
            $table->timestamps();
            $table->softDeletes();
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('users');
    }
};

```

Pada sourcecode di atas adalah tentang penciptaan tabel “users”. Setiap atribut seperti “nama”, “email”, dan “password” mewakili kolom dalam tabel dengan tipe dan karakteristik yang telah ditentukan. Metode chaining yang digunakan dalam definisi tabel memberikan kerangka kerja yang terstruktur dan mudah diikuti.

Ketika perintah migrasi dieksekusi, struktur tabel “users” akan terbentuk dalam basis data, memfasilitasi penyimpanan informasi pengguna dengan format yang telah ditentukan.

d. Tabel kampus

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('kampus', function (Blueprint
$table) {
            $table->id();
            $table->bigInteger('lokasi_id')->nullable()-
>unsigned();
            $table->string('thumbnail')->nullable();
            $table->string('nama');
            $table->softDeletes();
            $table->timestamps();

            $table->foreign('lokasi_id')
                ->references('id')
                ->on('lokasi')
                ->onDelete('cascade');
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('kampus');
    }
};

```

Pada sourcecode di atas menjelaskan pembuatan tabel

“kampus” yang memiliki atribut-atribut seperti id, lokasi_id, thumbnail, dan nama. Penggunaan metode chaining pada objek Blueprint memudahkan pendefinisian struktur tabel secara terstruktur dan ringkas. Selain itu, pengaturan foreign-key menunjukkan hubungan antara tabel “kampus” dengan tabel “lokasi”.

Migrasi ini akan dijalankan saat perintah migrasi dieksekusi, sehingga struktur tabel “kampus” akan tercipta di basis data situs.

e. Tabel acara

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('acara', function (Blueprint
$table) {
            $table->id();
            $table->bigInteger('kategori_id')-
>unsigned();
            $table->bigInteger('user_id')->unsigned();
            $table->bigInteger('kampus_id')->unsigned()-
>nullable();
            $table->string('judul');
            $table->string('slug');
            $table->string('thumbnail')->nullable();
            $table->text('deskripsi');
            $table->dateTime('waktu_mulai');
            $table->dateTime('waktu_selesai');
            $table->integer('durasi_estimasi')-
>nullable();
            $table->integer('kuota');
            $table->enum('status', ['draft', 'published']);
            $table->string('dress_code')->nullable();
            $table->string('kontak')->nullable();
            $table->string('social_media')->nullable();
            $table->text('peraturan')->nullable();
            $table->string('foto_stage')->nullable();
            $table->integer('view_count')->default(0);
            $table->text('alamat')->nullable();
            $table->string('latitude')->nullable();
            $table->string('longitude')->nullable();
            $table->softDeletes();
            $table->timestamps();

            $table->foreign('kategori_id')
                ->references('id')
                ->on('kategori');

            $table->foreign('user_id')
                ->references('id')
                ->on('users');

            $table->foreign('kampus_id')
                ->references('id')
                ->on('kampus');
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('acara');
    }
};

```

Pada sourcecode di atas menjelaskan, bahwa dalam aspek migrasi basis data berfokus pada penciptaan struktur tabel “acara”. Atribut-atribut seperti “judul”, “waktu_mulai”, dan “kuota” merepresentasikan informasi yang tersimpan dalam kolom-kolom tabel. Penggunaan metode chaining digunakan untuk membangun hubungan antara tabel “acara” dengan tabel “kategori” dan

tabel “users”.

Implementasi migrasi ini, ketika dieksekusi, akan menciptakan sebuah entitas tabel “acara” yang memfasilitasi penyimpanan informasi tentang berbagai acara yang diadakan di situs.

f. Tabel kategori

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('kategori', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('thumbnail')->nullable();
            $table->softDeletes();
            $table->timestamps();
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('kategori');
    }
};
```

Pada sourcecode di atas menjelaskan penciptaan struktur tabel “kategori”. Atribut-atribut seperti “nama” dan “thumbnail” merepresentasikan data yang akan disimpan dalam bentuk kolom dalam tabel. Pada saat migrasi dieksekusi, struktur tabel “kategori” akan tercipta, memungkinkan penyimpanan informasi tentang berbagai kategori acara.

g. Tabel tiket

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('tiket', function (Blueprint $table) {
            $table->id();
            $table->bigInteger('jenis_tiket_id')->unsigned();
            $table->string('kode');
            $table->string('tier')->nullable();
            $table->string('kursi')->nullable();
            $table->bigInteger('harga')->default(0);
            $table->softDeletes();
            $table->timestamps();

            $table->foreign('jenis_tiket_id')
                ->references('id')
                ->on('jenis_tiket');
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('tiket');
    }
};
```

Pada sourcecode di atas, dalam bagian ini diperlihatkan

proses penciptaan struktur tabel “tiket”. Atribut-atribut seperti “kode”, “tier”, dan “harga” mendefinisikan data yang akan disimpan dalam tabel. Pengaturan foreign-key dihubungkan dengan tabel “jenis_tiket” untuk membentuk relasi antara jenis tiket dan tiket-tiket yang tersedia.

h. Tabel jenis_tiket

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('jenis_tiket', function (Blueprint $table) {
            $table->id();
            $table->bigInteger('acara_id')->unsigned();
            $table->string('nama_paket');
            $table->integer('kuota')->default(0);
            // $table->integer('harga')->nullable();
            $table->boolean('is_free');
            $table->softDeletes();
            $table->timestamps();

            $table->foreign('acara_id')
                ->references('id')
                ->on('acara');
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('jenis_tiket');
    }
};
```

Pada sourcecode di atas menciptakan tabel “jenis_tiket” yang memiliki kolom-kolom penting untuk menggambarkan beberapa jenis tiket yang terkait dengan berbagai acara dalam sistem. Kolom “id” berperan sebagai primary-key untuk setiap entri dalam tabel ini, memastikan keunikan data. Selanjutnya, kolom “acara_id” adalah foreign-key yang menghubungkan jenis tiket dengan entri acara terkait. Ini memungkinkan hubungan antara jenis tiket dan acara yang ditentukan. Kolom “is_free” merupakan kolom boolean yang menandakan apakah jenis tiket ini gratis atau tidak.

Dengan struktur tabel yang ditetapkan melalui migrasi ini, informasi penting seperti nama jenis tiket, kuota, status gratis atau tidak, dan kaitannya dengan acara dapat terlacak dengan jelas. Ketika perintah migrasi dijalankan, struktur tabel “jenis_tiket” akan dibuat secara konsisten sesuai definisi dalam kode migrasi.

i. Tabel keuntungan

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('keuntungan', function (Blueprint $table) {
            $table->id();
```

```

        $table->bigInteger('tiket_id')->unsigned();
        $table->foreign('tiket_id')
            ->references('id')
            ->on('tiket');
        $table->text('keuntungan');
        $table->softDeletes();
        $table->timestamps();
    });
}
public function down(): void
{
    Schema::dropIfExists('keuntungans');
}
};

```

Pada sourcecode di atas, menciptakan tabel “keuntungans” yang berkaitan dengan keuntungan dari setiap jenis tiket. Kolom “id” berperan sebagai primary-key untuk setiap entri dalam tabel ini, memastikan keunikan data. Selanjutnya, kolom “tiket_id” adalah foreign-key yang menghubungkan keuntungan dengan jenis tiket terkait. Pengaturan foreign-key ini memungkinkan hubungan secara spesifik, dan kolom “keuntungan” adalah kolom teks yang akan menyimpan deskripsi atau informasi tentang keuntungan-keuntungan yang diperoleh dari jenis tiket tertentu.

Saat perintah migrasi dijalankan, struktur tabel “keuntungans” akan terbentuk sesuai definisi dalam kode migrasi, dan informasi tentang keuntungan tiket akan tersimpan dengan rapi dalam basis data.

j. Tabel sliders

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('sliders', function (Blueprint $table) {
            $table->id();
            $table->string('judul');
            $table->string('deskripsi');
            $table->string('thumbnail')->nullable();
            $table->softDeletes();
            $table->timestamps();
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('sliders');
    }
};

```

Pada sourcecode di atas, bagian migrasi ini dapat dilihat ada beberapa langkah dalam penciptaan struktur tabel “sliders”. Tabel ini bertujuan untuk menyimpan informasi mengenai elemen-elemen tampilan yang disebut “sliders”. Atribut-atribut yang didefinisikan dalam struktur tabel ini adalah bagian integral dari data yang akan dikelola dalam tabel.

Pertama, terdapat atribut “judul” yang merujuk pada judul atau keterangan singkat terkait elemen slider. Atribut “deskripsi” digunakan untuk menguraikan lebih lanjut tentang elemen slider tersebut. Selain itu, atribut “thumbnail”

memiliki kemampuan untuk menyimpan gambar kecil yang berhubungan dengan elemen slider, dan atribut ini bersifat opsional karena diberi opsi “nullable”.

k. Tabel transaksis

Sourcecode dibawah ini merupakan implementasi migrasi dari tabel “kampus” adalah sebagai berikut:

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('transaksis', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('tiket_id');
            $table->unsignedBigInteger('acara_id');
            $table->unsignedBigInteger('user_id');

            $table->string('kode_transaksi')->unique();
            $table->string('jenis_pembayaran')->nullable();
            $table->string('token_pembayaran')->nullable();
            $table->dateTime('tanggal_pembayaran')->nullable();
            $table->string('url_pembayaran')->nullable();
            $table->string('url_invoice')->nullable();
            $table->string('durasi_pembayaran')->nullable();
            $table->comment('menit'); //menit
            $table->string('diskon')->nullable();
            $table->string('jumlah_dibayar')->nullable();
            $table->string('jumlah_diterima_disesuaikan')->nullable();
            $table->string('bukti_transfer')->nullable();
            $table->string('harga');
            $table->string('batas_pembayaran');
            $table->string('total_pembayaran');
            $table->enum('status', ['MENUNGGU', 'DIBAYAR', 'DITOLAK', 'KADALUARSA', 'GAGAL']);
            $table->timestamps();
            $table->softDeletes();

            $table->foreign('acara_id')
                ->references('id')
                ->on('acara');
            $table->foreign('tiket_id')
                ->references('id')
                ->on('tiket');
            $table->foreign('user_id')
                ->references('id')
                ->on('users');
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('transaksis');
    }
};

```

Pada sourcecode di atas ini, dapat dilihat langkah-langkah untuk membuat struktur tabel “transaksis”. Tabel ini akan merekam berbagai detail transaksi di situs. Dalam basis data ini foreign-key yang menghubungkan tabel ini dengan tabel lain, seperti “tiket_id”, “acara_id”, dan “user_id”, membentuk hubungan dengan tiket, acara, dan pengguna terkait. Tabel ini juga mencatat status transaksi melalui atribut “status”, seperti “MENUNGGU”, “DIBAYAR”, “DITOLAK”, “KADALUARSA”, dan “GAGAL”.

Semua langkah ini membentuk struktur tabel “transaksis” yang penting dalam mengatur data transaksi, memberikan pandangan terinci pada setiap transaksi yang terjadi di situs.

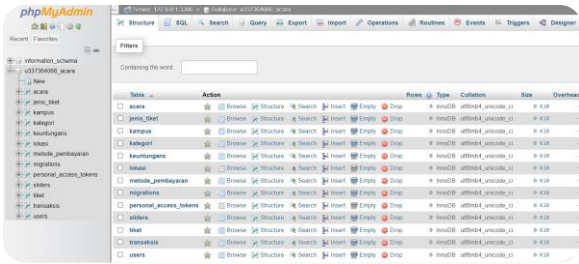
C. Pengujian

Pengujian database yang dilakukan pertama kali yaitu pengujian berfokus kepada kemampuan sistem untuk menghubungkan ke database yang telah dikonfigurasi dengan benar dan memproses login pengguna dengan tepat, melalui simulasi login, verifikasi kredensial, dan pengujian respons sistem dengan database dalam skenario yang berbeda, pengujian ini memastikan bahwa aplikasi dapat mengakses database dan mengelola informasi pengguna dengan benar.

Konfigurasi ini berfungsi untuk digunakan menyimpan variabel lingkungan yang penting, termasuk informasi koneksi database, dan dilakukan melalui file (.env).

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nama_database
DB_USERNAME=username
DB_PASSWORD=password
```

Dari file (.env), pengembang melakukan konfigurasi yang valid berdasarkan pada phpMyAdmin, seperti host, nama database, username, dan password. Kemudian menjalankan perintah `php artisan migrate` pada terminal sourcecode situs, untuk memastikan bahwa struktur tabel yang diperlukan telah dibuat di database.



Gambar 3 phpMyAdmin

Pada Gambar 3, dapat dilihat bahwa tabel-tabel yang telah diartikan seperti "users", "acara", "kategori", dan tabel lainnya berhasil terbentuk di dalam database yang sesuai. Setiap tabel menyajikan kolom-kolom yang telah ditetapkan sebelumnya sesuai dengan definisi yang ditetapkan. Hal ini menciptakan kerangka yang terstruktur untuk menyimpan serta mengambil data dengan ketertiban.

Hasil analisis dari tabel di atas adalah setiap aspek yang di uji berhasil berjalan. User dapat melakukan daftar, masuk, membuat kegiatan, dan membeli tiket kegiatan acara pada table 9.

Tabel 9 Analisis Database

Skenario	Hasil yang diharapkan	Hasil yang sebenarnya	Status
Registrasi & verifikasi email	Setelah pengguna melakukan registrasi, pengguna melanjutkan dengan melakukan verifikasi pada email yang di daftarkan sebelumnya.	Setelah pengguna melakukan registrasi, pengguna melanjutkan dengan melakukan verifikasi pada email yang di daftarkan sebelumnya	Berhasil
Login & verifikasi kredensial	Pada saat login akan dilakukan verifikasi	Pada saat login akan dilakukan verifikasi kredensial pada	Berhasil

	kredensial pada aspek e-mail, dan password.	aspek e-mail, dan password.	
Buat kegiatan	Pengguna berhasil menambahkan data kegiatan acara baru.	Pengguna berhasil menambahkan data kegiatan acara baru.	Berhasil
Pembelian tiket	Pengguna berhasil melakukan pemesanan tiket sampai dengan pembayaran.	Pengguna berhasil melakukan pemesanan tiket sampai dengan pembayaran.	Berhasil

Berikut adalah langkah pengujian pada database:

- a. Pengujian Koneksi Database
 - Mencoba terhubung ke database EasyIn.site.
 - Memeriksa apakah koneksi berhasil atau gagal.
- b. Pengujian Integritas Data
 - Memeriksa apakah tipe data yang dimasukkan sesuai dengan aturan integritas data yang telah ditetapkan.
 - Memeriksa penggunaan kunci asing (foreign-key) untuk menjaga integritas referensial.
 - Memeriksa apakah pembatasan NULL diterapkan sesuai dengan kebutuhan.
 - Pengujian Pengelolaan Tiket.
 - Menambahkan tiket baru ke dalam database.
 - Memperbarui status tiket (misalnya: dari "tersedia" menjadi "terjual").
 - Menghapus tiket yang sudah tidak berlaku atau digunakan lagi.
 - Pengujian Data Pelanggan
 - Menambahkan data pelanggan baru ke dalam database.
 - Memeriksa apakah data pelanggan disimpan dengan benar.
 - Memeriksa riwayat transaksi pelanggan.
- c. Pengujian Transaksi
 - Melakukan transaksi pembelian tiket dan menyimpannya dalam database.
 - Memeriksa apakah transaksi berhasil tercatat dalam database dengan benar.

Hasil langkah pengujian :

- a. Pengujian Koneksi Database

Tabel 10 Hasil Pengujian Koneksi Database

Skenario	Hasil yang di harapkan	Hasil yang sebenarnya	Status
Mencoba terhubung ke database EasyIn.site.	Diharapkan website dapat terhubung ke database tanpa masalah.	Saat mencoba terhubung, tidak ada pesan kesalahan atau hambatan.	Berhasil
Memeriksa apakah koneksi berhasil atau gagal.	Harapannya, koneksi harus berhasil tanpa kesalahan.	Koneksi berhasil dengan cepat, tidak ada pesan kesalahan, dan respons website tampak normal.	Berhasil

b. Pengujian Integritas Data

Tabel 11 Hasil Pengujian Integritas Data

Skenario	Hasil yang di harapkan	Hasil yang sebenarnya	Status
Memeriksa apakah tipe data yang dimasukkan sesuai dengan aturan integritas data yang telah ditetapkan.	Harapannya, data yang dimasukkan harus sesuai dengan aturan integritas seperti tipe data yang benar.	Semua data yang dimasukkan sesuai dengan tipe data yang diharapkan tanpa kesalahan.	Berhasil
Memeriksa penggunaan kunci asing (foreign key) untuk menjaga integritas referensial.	Harapannya, penggunaan kunci asing harus menjaga integritas referensial antara tabel dengan benar.	Semua kunci asing digunakan secara benar untuk menjaga integritas referensial.	Berhasil

c. Pengujian Pengelolaan Tiket

Tabel 12 Hasil Pengujian Pengelolaan Tiket

Skenario	Hasil yang di harapkan	Hasil yang sebenarnya	Status
Menambahkan tiket baru ke dalam database.	Harapannya, tiket baru dapat ditambahkan tanpa masalah.	Tiket berhasil ditambahkan ke database tanpa kesalahan.	Berhasil
Memperbarui status tiket (misalnya: dari "tersedia" menjadi "terjual").	Diharapkan status tiket dapat diperbarui dengan benar.	Status tiket berhasil diperbarui sesuai yang diinginkan.	Berhasil
Menghapus tiket yang sudah tidak berlaku atau digunakan lagi.	Harapannya, tiket yang sudah tidak berlaku dapat dihapus tanpa masalah.	Tiket yang tidak berlaku berhasil dihapus dari database.	Berhasil

d. Pengujian Data Pelanggan

Tabel 13 Hasil Pengujian Data Pelanggan

Skenario	Hasil yang di harapkan	Hasil yang sebenarnya	Status
Menambahkan data pelanggan	Diharapkan data	Data pelanggan	Berhasil

baru ke dalam database.	pelanggan baru dapat ditambahkan tanpa masalah.	baru berhasil ditambahkan ke database tanpa kesalahan.	
Memeriksa apakah data pelanggan disimpan dengan benar, termasuk data pribadi.	Harapannya, data pelanggan harus disimpan dengan benar, termasuk data pribadi.	Semua data pelanggan disimpan dengan benar dan sesuai dengan yang diharapkan.	Berhasil
Memeriksa riwayat transaksi pelanggan.	Diharapkan riwayat transaksi pelanggan dapat diakses dan diperiksa dengan benar.	Riwayat transaksi pelanggan dapat diakses dan diperiksa tanpa kesalahan.	Berhasil

e. Pengujian Transaksi

Tabel 14 Hasil Pengujian Transaksi

Skenario	Hasil yang di harapkan	Hasil yang sebenarnya	Status
Melakukan transaksi pembelian tiket dan menyimpannya dalam database.	Harapannya, transaksi pembelian tiket dapat berhasil tercatat dengan benar.	Transaksi pembelian tiket berhasil tercatat dalam database tanpa kesalahan.	Berhasil

V. KESIMPULAN

Kesimpulan dari hasil analisis adalah bahwa database yang digunakan dalam website "EasyIn.site" menunjukkan kinerja yang baik dalam semua aspek yang diuji. Ini termasuk menjaga integritas data, mengelola tiket dan data pelanggan dengan efisien, serta mencatat transaksi dengan akurat. Temuan ini memberikan keyakinan bahwa sistem database yang digunakan dalam situs web tersebut berfungsi dengan baik.

Namun, penting untuk diingat bahwa pemeliharaan dan pemantauan berkala terhadap database tetap diperlukan. Tujuannya adalah memastikan kinerjanya tetap optimal dan mengidentifikasi serta mengatasi potensi kerentanan keamanan yang mungkin muncul. Analisis hasil pengujian ini dapat digunakan sebagai panduan untuk perbaikan dan peningkatan kualitas serta kinerja database di masa mendatang. Dengan demikian, website "EasyIn.site" dapat terus memberikan layanan yang andal dan efisien kepada pengguna.

VI. REFERENSI

- [1] Nestary, N. (2020). Perancangan Sistem Informasi Penjualan pada Toko Stock Point Lily berbasis PHP MySQL. *Jurnal Ilmu Komputer Dan Bisnis*, 11(1). <https://doi.org/10.47927/jikb.v11i1.195> .
- [2] Cahyono,Setiyo.(2009),Pemrograman Database Menggunakan MySQL Dan Java.Bandung
- [3] Connolly,Thomas dan Carolyn Begg,(2010), Database Systems A Practical Approach to Design, Implementation, and Management 4 th Edition, Pearson Education Limited,Harlow
- [4] Yu Ping, Hu Hong-Wei and Zhou Nan, "Design and implementation of a MySQL database backup and recovery system," *Intelligent Control and Automation (WCICA)*, 2014 11th World Congress on, Shenyang, 2014, pp. 5410-5415.
- [5] Raghil Nuruddin. "Pengertian & Definisi Web". 2013
- [6] Saputra Agus. "*Trik dan Solusi Jitu Pemrograman PHP*". PT. Elex Media Komputindo, Jakarta. --- .2013.*menyelesaikan website 12 juta secara profesional*". Jakarta.2011.
- [7] A. Sahi, "APLIKASI TEST POTENSI AKADEMIK SELEKSI SARINGAN MASUK LP3I," *TEMATIK - Jurnal Teknologi Informasi Dan Komunikasi*, vol. 7, no. 1, pp. 120-129, 2020.
- [8] N.Budi N., dan B.Anwar, 2008. Desain Web Menggunakan HTML dan JAVASCRIPT, *Jurnal SAINTIKOM*
- [9] Tarantang, J., Awwaliyah, A., Astuti, M., & Munawaroh, M. (2019). Perkembangan Sistem Pembayaran Digital Pada Era Revolusi Industri 4.0 Di Indonesia. *Jurnal Al-Qardh*, 4(1), 60–75. <https://doi.org/10.23971/JAQ.V4I1.1442>
- [10] Indah Suciani, N., Hernawati, E., & Arik Tridalestari, F. S. (2017). Aplikasi Pengelolaan Penyalur Jasa Tenaga Kerja Berbasis Web. *EProceedings of Applied Science*, 3(3). <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/5297>