Berikut adalah panduan untuk membuat API menggunakan Laravel dan MySQL untuk topik \*\*Registrasi Event Olahraga\*\*, lengkap dengan spesifikasi yang telah Anda berikan. Saya juga akan memberikan gambaran umum tentang cara membuat aplikasi front-end menggunakan React.js.

### 1. Membuat API di Laravel

#### Langkah 1: Setup Laravel Project

1. \*\*Instal Laravel\*\*:

```bash

composer create-project --prefer-dist laravel/laravel event-registration

cd event-registration

```

2. \*\*Buat Database\*\*:

Buat database baru di MySQL, misalnya `event\_registration\_db`.

3. \*\*Konfigurasi Database\*\*:

Ubah file `.env` dengan detail database Anda:

```env

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=event\_registration\_db

DB\_USERNAME=your\_username

DB\_PASSWORD=your\_password

```

#### Langkah 2: Buat Migration dan Model

1. \*\*Buat Migration untuk Tabel `events`\*\*:

```bash

php artisan make:migration create\_events\_table

```

2. \*\*Edit Migration\*\*:

Edit file migration yang baru dibuat di `database/migrations`:

```php

public function up()

{

Schema::create('events', function (Blueprint $table) {

$table->id();

$table->string('participant\_name');

$table->string('event\_name');

$table->date('event\_date');

$table->string('registration\_number')->unique();

$table->string('category');

$table->timestamps();

});

}

```

3. \*\*Jalankan Migration\*\*:

```bash

php artisan migrate

```

4. \*\*Buat Model\*\*:

```bash

php artisan make:model Event

```

#### Langkah 3: Buat Controller

1. \*\*Buat Controller\*\*:

```bash

php artisan make:controller EventController

```

2. \*\*Edit Controller\*\*:

Tambahkan fungsi untuk GET dan POST:

```php

use App\Models\Event;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

class EventController extends Controller

{

// Get all events

public function index()

{

$events = Event::all();

return response()->json([

'code' => 200,

'message' => 'Success',

'data' => $events

]);

}

// Insert new event

public function store(Request $request)

{

$validator = Validator::make($request->all(), [

'participant\_name' => 'required|string|max:255',

'event\_name' => 'required|string|max:255',

'event\_date' => 'required|date',

'registration\_number' => 'required|string|unique:events',

'category' => 'required|string|max:255'

]);

if ($validator->fails()) {

return response()->json(['code' => 400, 'message' => 'Failed'], 400);

}

$event = Event::create($request->all());

return response()->json(['code' => 200, 'message' => 'Success']);

}

}

```

#### Langkah 4: Definisikan Routes

Edit file `routes/api.php` untuk menambahkan route:

```php

use App\Http\Controllers\EventController;

Route::get('/events', [EventController::class, 'index']);

Route::post('/events', [EventController::class, 'store']);

```

### 2. Membuat Aplikasi Front-end dengan React.js

#### Setup React Project

1. \*\*Buat Proyek React\*\*:

```bash

npx create-react-app event-registration-client

cd event-registration-client

```

2. \*\*Instal Axios dan Bootstrap\*\*:

```bash

npm install axios bootstrap

```

3. \*\*Edit `src/index.js` untuk menyertakan Bootstrap\*\*:

```javascript

import 'bootstrap/dist/css/bootstrap.min.css';

```

#### Membuat Komponen untuk Menampilkan dan Menambah Data

1. \*\*Buat Komponen `EventList.js`\*\*:

```javascript

import React, { useEffect, useState } from 'react';

import axios from 'axios';

const EventList = () => {

const [events, setEvents] = useState([]);

useEffect(() => {

const fetchEvents = async () => {

const response = await axios.get('/api/events');

setEvents(response.data.data);

};

fetchEvents();

}, []);

return (

<div className="container">

<h2>Event List</h2>

<ul className="list-group">

{events.map(event => (

<li key={event.id} className="list-group-item">

{event.participant\_name} - {event.event\_name}

</li>

))}

</ul>

</div>

);

};

export default EventList;

```

2. \*\*Buat Komponen `AddEvent.js`\*\*:

```javascript

import React, { useState } from 'react';

import axios from 'axios';

const AddEvent = () => {

const [eventData, setEventData] = useState({

participant\_name: '',

event\_name: '',

event\_date: '',

registration\_number: '',

category: ''

});

const handleChange = (e) => {

setEventData({ ...eventData, [e.target.name]: e.target.value });

};

const handleSubmit = async (e) => {

e.preventDefault();

try {

await axios.post('/api/events', eventData);

alert('Event added successfully!');

} catch (error) {

alert('Failed to add event.');

}

};

return (

<form onSubmit={handleSubmit} className="container">

<h2>Add Event</h2>

<div className="form-group">

<input type="text" name="participant\_name" placeholder="Participant Name" onChange={handleChange} className="form-control" required />

<input type="text" name="event\_name" placeholder="Event Name" onChange={handleChange} className="form-control" required />

<input type="date" name="event\_date" onChange={handleChange} className="form-control" required />

<input type="text" name="registration\_number" placeholder="Registration Number" onChange={handleChange} className="form-control" required />

<input type="text" name="category" placeholder="Category" onChange={handleChange} className="form-control" required />

<button type="submit" className="btn btn-primary mt-2">Add Event</button>

</div>

</form>

);

};

export default AddEvent;

```

3. \*\*Integrasikan Komponen di `App.js`\*\*:

```javascript

import React from 'react';

import EventList from './EventList';

import AddEvent from './AddEvent';

const App = () => {

return (

<div>

<AddEvent />

<EventList />

</div>

);

};

export default App;

```

### 3. Jalankan Proyek

1. \*\*Jalankan Server Laravel\*\*:

```bash

php artisan serve

```

2. \*\*Jalankan Aplikasi React\*\*:

```bash

npm start

```

### Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda telah berhasil membuat API untuk Registrasi Event Olahraga menggunakan Laravel, serta aplikasi front-end menggunakan React.js untuk mengakses API tersebut. Anda dapat mengembangkan lebih lanjut dengan menambahkan fitur-fitur seperti edit dan delete, serta meningkatkan antarmuka pengguna.