**Analisis Sosial dan Komunitas dalam Aplikasi Elpun Community**

**1. Pengertian Sosial dan Komunitas**

**Sosial** mengacu pada interaksi antara individu atau kelompok dalam suatu masyarakat atau lingkungan. Secara lebih luas, sosial berhubungan dengan hubungan antar manusia yang membentuk struktur atau pola dalam masyarakat. Dalam konteks aplikasi, interaksi sosial sering melibatkan bentuk komunikasi atau pertukaran informasi antar pengguna, seperti komentar, likes, atau berbagi pengalaman.

**Komunitas**, di sisi lain, adalah kelompok orang yang memiliki kesamaan minat, tujuan, atau aktivitas yang sama. Komunitas bisa dibentuk oleh berbagai alasan, seperti minat terhadap topik tertentu (misalnya game), tujuan bersama, atau kebutuhan untuk berinteraksi dengan orang lain yang memiliki kesamaan perspektif. Dalam dunia digital, komunitas sering kali dibangun berdasarkan platform tertentu seperti YouTube, yang membentuk kelompok pengikut, gamer, atau hobi yang berfokus pada konten yang saling berbagi dan berinteraksi.

**2. Hubungan Sosial dan Komunitas dalam Aplikasi Anda**

Untuk aplikasi Anda, **Elpun Community**, yang bertujuan untuk menampilkan jadwal live streaming, meskipun tidak memiliki fitur sosial interaktif (seperti komentar atau feedback), tetap dapat memfasilitasi hubungan sosial dalam bentuk lain. Meskipun pengguna tidak berinteraksi secara langsung dengan jadwal, mereka tetap dapat melihat, memahami, dan mengingat acara yang akan datang. Dalam hal ini, aplikasi berfungsi untuk memperkuat ikatan antar anggota komunitas yang memiliki minat yang sama terhadap game dan acara live yang Anda selenggarakan.

**Interaksi sosial dalam konteks ini** lebih fokus pada konsistensi dan rutinitas—jadwal yang pasti membantu pengikut mengetahui kapan mereka dapat bergabung dan berinteraksi dengan komunitas. Ini menciptakan pola sosial yang memudahkan anggota untuk terlibat dalam acara yang ada, meskipun mereka tidak dapat melakukan interaksi langsung melalui aplikasi.

**3. Pemilihan Algoritma yang Cocok**

Mengingat kebutuhan aplikasi yang relatif sederhana dan berfokus pada penjadwalan, Anda tidak memerlukan algoritma yang kompleks atau canggih. Berikut adalah beberapa pilihan algoritma dan teknologi yang dapat digunakan dalam pengembangan aplikasi Anda:

**a. Pemrograman CRUD (Create, Read, Update, Delete)**

* **Penjelasan**: Aplikasi Anda berfokus pada penyimpanan dan pengelolaan jadwal acara. Oleh karena itu, **CRUD**(Create, Read, Update, Delete) adalah algoritma dasar yang digunakan untuk membuat, membaca, mengedit, dan menghapus jadwal.
* **Penerapan dalam Aplikasi**:
  + **Create**: Admin dapat menambah jadwal acara live baru.
  + **Read**: Jadwal yang telah dimasukkan akan ditampilkan kepada pengguna.
  + **Update**: Admin dapat mengubah jadwal yang sudah ada.
  + **Delete**: Admin dapat menghapus jadwal yang tidak relevan atau tidak ada lagi.
* **Keuntungan**: Mudah untuk diimplementasikan dengan framework Laravel, memberikan kontrol penuh kepada admin tanpa perlu fitur interaktif lainnya seperti pencarian atau notifikasi.

**b. Algoritma Penyimpanan dan Pengelolaan Data**

* **Penjelasan**: Dalam aplikasi ini, data jadwal live disimpan dalam database MySQL yang di-hosting menggunakan Docker. Proses penyimpanan data mengikuti pola biasa untuk aplikasi berbasis **relational database**.
* **Penerapan dalam Aplikasi**:
  + Penggunaan **Laravel Eloquent ORM** untuk memudahkan interaksi dengan database.
  + Setiap jadwal live disimpan dalam tabel schedules yang mencatat hari, waktu mulai, waktu selesai, dan tema game.
* **Keuntungan**: Penyimpanan data dalam MySQL memastikan integritas data dan memungkinkan pengelolaan jadwal yang efisien dengan query sederhana. Dengan Docker, Anda bisa mengisolasi aplikasi dan database secara terpisah, memudahkan manajemen dan skalabilitas.

**c. Algoritma Pengelolaan Akses (Role-based Access Control / RBAC)**

* **Penjelasan**: Anda hanya membutuhkan kontrol akses bagi admin untuk mengelola jadwal. Laravel sudah menyediakan sistem autentikasi bawaan yang dapat dikustomisasi untuk menambah kontrol akses berdasarkan **role**.
* **Penerapan dalam Aplikasi**:
  + **Admin**: Hanya admin yang dapat menambah, mengedit, atau menghapus jadwal. Akses untuk mengubah data dikendalikan oleh sistem autentikasi Laravel.
  + **Pengguna biasa**: Hanya dapat melihat jadwal yang telah dimasukkan admin, tanpa akses untuk melakukan perubahan.
* **Keuntungan**: Penggunaan **middleware** untuk mengatur akses admin mengamankan aplikasi dari perubahan yang tidak sah. Ini sederhana dan efektif untuk aplikasi Anda yang memiliki pengguna terbatas.

**4. Faktor Keamanan dan Pengelolaan Data**

* **Keamanan dan Penyimpanan Data**: Karena aplikasi ini hanya untuk admin dan beroperasi di server pribadi, faktor keamanannya dapat lebih sederhana. Anda hanya perlu memastikan autentikasi admin yang kuat dan mengonfigurasi server agar aman.
  + **Autentikasi**: Gunakan autentikasi Laravel untuk mengelola login admin.
  + **Server Keamanan**: Menyimpan data di server pribadi mengurangi risiko akses yang tidak sah.
* **Backup Data**: Akan ada backup manual secara berkala.

**5. Skalabilitas dan Potensi Pengembangan**

* **Skalabilitas**: Meskipun aplikasi ini didesain untuk komunitas kecil, masih dapat membuatnya lebih skalabel di masa depan. Misalnya, jika aplikasi ingin diperkenalkan ke publik, Anda bisa menambahkan fitur seperti pencarian jadwal, integrasi YouTube, dan mungkin sistem notifikasi.
* **Pengembangan di Masa Depan**: Jika Anda berencana mengembangkan aplikasi ini, beberapa fitur yang dapat ditambahkan meliputi:
  + **Integrasi YouTube** untuk menautkan jadwal live ke video YouTube.
  + **Pencarian jadwal** dan **filter berdasarkan hari** untuk memudahkan pengguna melihat jadwal tertentu.
  + **Notifikasi dan Pengingat** kepada pengguna jika jadwal acara mendekati.

**6. Masalah yang Dipecahkan oleh Aplikasi Elpun Community**

* **Kesulitan Menemukan Jadwal Acara Live Streaming**  
  Sebelumnya, pengguna mungkin kesulitan menemukan jadwal acara live streaming yang mereka minati. Dengan aplikasi ini, jadwal acara game yang terorganisir dan terpusat membantu pengguna untuk melihat semua acara yang akan datang dalam satu platform.
* **Kurangnya Keterlibatan Sosial dalam Komunitas**  
  Aplikasi ini memfasilitasi keterlibatan sosial melalui rutinitas yang terstruktur, meskipun tanpa fitur interaktif seperti komentar. Pengguna dapat melihat jadwal acara secara konsisten, yang membangun keterhubungan di antara anggota komunitas yang memiliki minat serupa.
* **Pengelolaan Jadwal oleh Admin yang Tidak Efisien**  
  Admin sering kesulitan mengelola jadwal acara secara manual. Dengan menggunakan sistem **CRUD**, aplikasi ini memungkinkan admin untuk dengan mudah membuat, mengedit, dan menghapus jadwal acara, meningkatkan efisiensi pengelolaan jadwal.
* **Kesulitan dalam Menjaga Keamanan Data**  
  Aplikasi ini mengatasi masalah keamanan dengan sistem kontrol akses berbasis peran (RBAC), memastikan hanya admin yang dapat mengelola data jadwal, serta mengurangi risiko akses yang tidak sah.
* **Keterbatasan Skalabilitas**  
  Meskipun dirancang untuk komunitas kecil, aplikasi ini dirancang agar mudah untuk diubah dan skalabel di masa depan, dengan kemungkinan menambah fitur seperti pencarian jadwal, integrasi dengan YouTube, dan sistem pengingat.

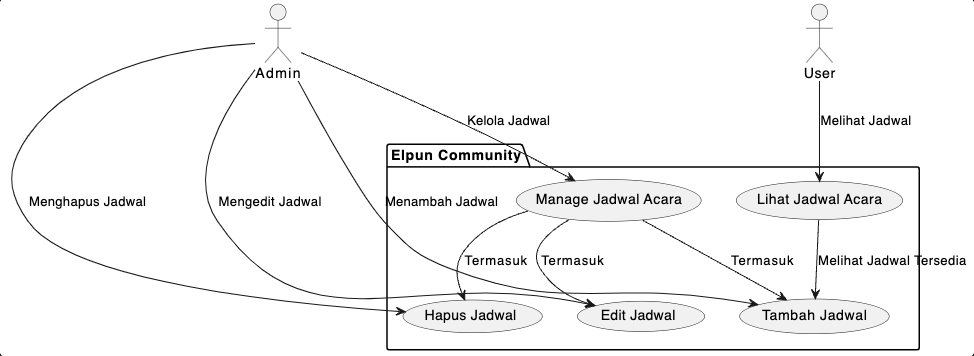
**Kesimpulan**

Aplikasi ini akan memiliki fitur yang sederhana namun cukup efektif untuk menampilkan jadwal acara live kepada pengguna. **Sosial dan komunitas** di sini lebih berfungsi sebagai bagian dari interaksi sosial yang terstruktur, pengguna dapat melihat dan mengikuti jadwal yang sudah ada tanpa perlu interaksi sosial intensif seperti komentar atau feedback.

Pemilihan algoritma yang digunakan cukup sederhana dan langsung sesuai dengan kebutuhan aplikasi, termasuk CRUD untuk manajemen jadwal, kontrol akses berbasis peran, dan penggunaan waktu sesuai dengan zona WIB. Sistem ini akan sangat efisien dengan database MySQL dan penyimpanan data yang dikelola dengan Docker, serta memastikan keamanan aplikasi dengan autentikasi admin.

Apabila di masa depan aplikasi perlu dikembangkan atau ditingkatkan skalabilitasnya, dapat dengan mudah menambahkan fitur-fitur seperti pencarian jadwal, integrasi YouTube, dan lainnya.

**Use Case**



**Flowcart**

**A diagram of a program

AI-generated content may be incorrect.**