**2. Pemilihan Arsitektur Aplikasi**

**2.1. Tujuan Pemilihan Arsitektur**

* Menyediakan fondasi yang kokoh untuk pengembangan, pemeliharaan, dan skalabilitas aplikasi membership GYM. Aplikasi harus dapat menangani pembuatan member, dan pengelolaan pengguna (Admin,User)

**2.2. Pilihan Arsitektur**

**2.2.1. Model-View-Controller (MVC)**

* **Deskripsi**:

MVC membagi aplikasi Membership GYM menjadi Model (data Member), View (menyajikan antarmuka pelanggan dan admin), dan Controller (menghubungkan input pelanggan dengan model).

* **Keuntungan**:

1. Struktur modular memudahkan pengembangan fitur membuat membership secara online. Cocok untuk sistem Membership yang anggotanya berada dijauh lokasi.
2. Memanfaatkan berbagai fitur bawaan Laravel seperti routing, autentikasi, dan Eloquent ORM untuk pengelolaan database.

* **Kekurangan**:

Dalam aplikasi besar, MVC bisa lebih kompleks dalam pengelolaan.

**2.2.2. Microservices**

* **Deskripsi**:

Aplikasi dibagi menjadi layanan-layanan kecil seperti pembuatan membership secara online

* **Keuntungan**:

Setiap layanan dapat dikembangkan dan dipelihara secara independen.

* **Kekurangan**:

Memerlukan komunikasi dan manajemen yang baik antara layanan-layanan ini.

**2.2.3. Single Page Application (SPA)**

* + **Deskripsi**:

Website membership memuat satu halaman HTML dengan konten yang diupdate secara dinamis.

* + **Keuntungan**:

Pengalaman pengguna yang lebih interaktif dengan respon cepat saat melakukan pembuatan members

* + **Kekurangan**:

SEO mungkin terbatas, namun aplikasi membership hanya focus pada pembuatan member saja

**2.3. Rekomendasi Arsitektur**

Untuk aplikasi laundry, MVC adalah pilihan yang paling cocok. Struktur ini memudahkan pengelolaan data user members.

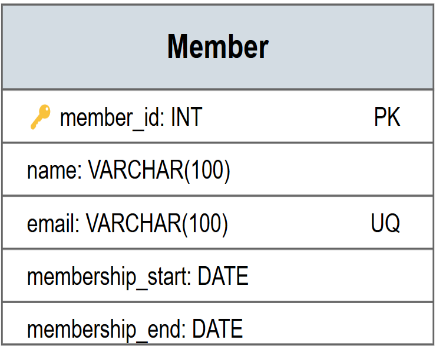
**2. Desain Struktur Database Awal**

**2.1. Tujuan Desain Struktur Database**

* Memastikan penyimpanan data yang efisien dan terstruktur, sehingga data member dapat dikelola dengan baik.

**2.2. Desain Struktur Database**

**2.2.1. Diagram Entity-Relationship (ER)**

* **Deskripsi Diagram**:

**2.2.2. Tabel-Tabel Utama**

* Tabel Member
* **Kolom**:Member\_id(PK), Nama, Email (UQ), Membership\_start, Membership\_end.

**2.3. Rekomendasi Struktur Database**

* Struktur database ini dirancang untuk mengelola semua data penting dari user member.

**3. Pemilihan Teknologi Frontend dan Backend**

**3.1. Tujuan Pemilihan Teknologi**

* Memilih teknologi yang tepat untuk memastikan kinerja sistem membership yang baik, serta kemudahan integrasi antara frontend dan backend.

**3.2. Pemilihan Teknologi Frontend**

**3.2.1. Framework/Library**

* **Pilihan** : Laravel
* **Deskripsi** : Laravel
* **Kriteria Pemilihan** Performa, integrasi dengan baik dengan backend dan dokumentasi yang baik

**3.2.2. Teknologi Tambahan**

* **CSS Framework**: Bootstrap

**3.3. Pemilihan Teknologi Backend**

**3.3.1. Bahasa Pemrograman dan Framework**

* **Pilihan**: PHP dengan Laravel
* **Deskripsi**: Laravel menyediakan solusi pengembangan backend yang solid untuk aplikasi membership, termasuk fitur pembuatan member.
* **Kriteria Pemilihan**: Laravel mendukung pengelolaan transaksi dengan baik, skalabilitas, serta keamanan.

**3.3.2. Basis Data**

* **Pilihan**: MySQL
* **Deskripsi**: MySQL cocok untuk aplikasi membership yang memerlukan penyimpanan data user members
* **Kriteria Pemilihan**: Skalabilitas dan fitur transaksi yang kuat.

**3.4. Rekomendasi Teknologi**

* **Frontend**: HTML, CSS (Bootstrap), Laravel untuk build proses.
* **Backend**:
* Laravel dengan PHP untuk server-side logic dan manajemen data menggunakan MySQL.
* Laravel Eloquent ORM untuk interaksi dengan database
* Laravel Sanctum untuk autentikasi API
* **Database**:
* MySQL dengan Eloquent ORM untuk pengelolaan relasi dan query.
* MySQL dengan Laravel Migrations untuk version control schema database