**2. Pemilihan Arsitektur Aplikasi**

**2.1. Tujuan Pemilihan Arsitektur**

* Menyediakan dasar yang solid untuk pengembangan, pemeliharaan, dan skalabilitas aplikasi.
* Memastikan arsitektur yang dipilih dapat mendukung kebutuhan fungsional (CRUD resep) dan non-fungsional (keamanan, performa).

**2.2. Pilihan Arsitektur**

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

* **Deskripsi**: Arsitektur MVC membagi aplikasi menjadi tiga komponen utama: Model (mengelola data dan logika bisnis), View (menyajikan data kepada pengguna), dan Controller (menangani input dan memperbarui model).
* **Keuntungan**: Pemisahan yang jelas antara logika bisnis, tampilan, dan kontrol. Memudahkan pemeliharaan dan pengembangan fitur baru.
* **Kekurangan**: Bisa menjadi kompleks jika tidak dikelola dengan baik, terutama dalam aplikasi yang sangat besar.

**2.2.2. Microservices**

* **Deskripsi**: Pendekatan arsitektur di mana aplikasi dibagi menjadi serangkaian layanan kecil dan independen yang berkomunikasi melalui API.
* **Keuntungan**: Pemisahan yang jelas antara logika bisnis, tampilan, dan control, Kemudahan dalam pengembangan dan pemeliharaan aplikasi.
* **Kekurangan**: Kompleksitas tinggi dalam hal pengelolaan layanan dan komunikasi antar layanan, memerlukan manajemen distribusi yang cermat.

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

* **Deskripsi**: Aplikasi web yang memuat satu halaman HTML dan mengupdate konten secara dinamis dengan JavaScript.
* **Keuntungan**: Pengalaman pengguna yang lebih mulus dan interaktif, pengurangan waktu muat halaman.
* **Kekurangan**: SEO bisa menjadi tantangan, lebih kompleks dalam hal pengelolaan status dan rute.

**2.3. Rekomendasi Arsitektur**

* **Rekomendasi**: Menggunakan **MVC** karena Laravel telah mengintegrasikan arsitektur ini dengan baik, memberikan alat dan dokumentasi yang memadai untuk kebutuhan pengembangan aplikasi **KreasiDapur**.

**2. Desain Struktur Database Awal**

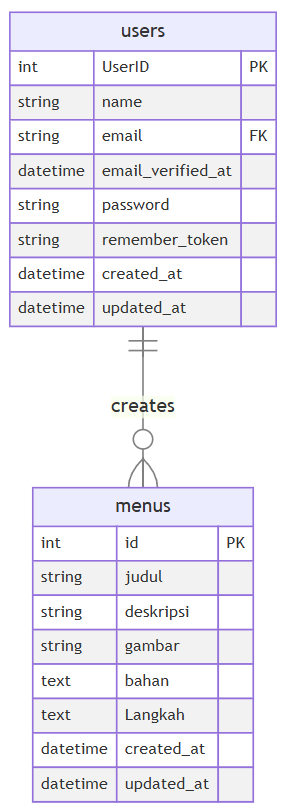
**2.1. Tujuan Desain Struktur Database**

* Memastikan penyimpanan data yang efisien dan terstruktur dengan baik.
* Mempermudah query data untuk operasi CRUD dan pengelolaan data lainnya.

**2.2. Desain Struktur Database**

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

* **Deskripsi**: Diagram ER menggambarkan entitas dalam sistem, atribut mereka, dan hubungan antar entitas.
* **Diagram**:



**2.2.2. Tabel-Tabel Utama**

* **Tabel users**
  + **Kolom**: UserID (PK), name, email(FK), email\_verified\_at, password, remember\_token, created\_at, update\_at
* **Tabel menus**
  + **Kolom**: id (PK), judul, deskripsi, gambar, bahan, langkah, create\_at, update\_at

**2.3. Rekomendasi Struktur Database**

* **Rekomendasi**: Struktur database ini dirancang untuk memastikan integritas data dan mendukung fungsionalitas seperti pencarian resep dan unggah resep

**3. Pemilihan Teknologi Frontend dan Backend**

**3.1. Tujuan Pemilihan Teknologi**

* Memastikan teknologi yang dipilih dapat memenuhi kebutuhan fungsional dan teknis aplikasi.
* Mempertimbangkan aspek performa, skalabilitas, dan kemudahan pemeliharaan.

**3.2. Pemilihan Teknologi Frontend**

**3.2.1. Framework/Library**

* **Pilihan**: Bootstrap.
* **Deskripsi**: Framework CSS untuk mendesain antarmuka pengguna yang responsif dan interaktif.
* **Kriteria Pemilihan**: Mudah digunakan, didukung komunitas luas, dan kompatibel dengan Laravel Blade untuk rendering UI.

**3.2.2. Teknologi Tambahan**

* HTML dan JavaScript digunakan untuk elemen dinamis.
* **Icon Library**: FontAwesome untuk ikon tambahan.

**3.3. Pemilihan Teknologi Backend**

**3.3.1. Bahasa Pemrograman dan Framework**

* **Pilihan**: Laravel (PHP).
* **Deskripsi**: Framework backend dengan arsitektur MVC yang terintegrasi.
* **Kriteria Pemilihan**: Menyediakan fitur bawaan seperti autentikasi, validasi, dan ORM (Eloquent), Dokumentasi lengkap dan komunitas luas.

**3.3.2. Basis Data**

* **Pilihan**: MySQL
* **Deskripsi**: Sistem manajemen basis data relasional yang mendukung transaksi dan query yang kompleks.
* **Kriteria Pemilihan**: Kompatibilitas tinggi dengan Laravel, Mendukung skalabilitas dan integritas data.

**3.4. Rekomendasi Teknologi**

* **Frontend**: Bootstrap dan Blade Laravel untuk UI responsif.
* **Backend**: Laravel dengan MySQL sebagai database, karena mendukung kebutuhan proyek secara fleksibel dan efisien.