**2. Pemilihan Arsitektur Aplikasi**

**2.1. Tujuan Pemilihan Arsitektur**

* Membangun arsitektur yang memungkinkan aplikasi Mie Ayam Gapuro dapat dikembangkan dan dipelihara dengan mudah serta mendukung pertumbuhan bisnis di masa depan.
* Memastikan arsitektur yang dipilih mampu memenuhi kebutuhan fungsional (seperti pemesanan makanan, manajemen menu, dan pengguna) serta kebutuhan non-fungsional (seperti kecepatan, keamanan, dan fleksibilitas).

**2.2. Pilihan Arsitektur**

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

* **Deskripsi:** Laravel menggunakan arsitektur MVC yang membagi aplikasi menjadi tiga komponen utama:
* **Model:** Mengelola data produk, pesanan, dan pengguna.
* **View:** Menyajikan antarmuka aplikasi seperti katalog produk dan status pesanan.
* **Controller:** Mengelola input pengguna dan berinteraksi dengan model untuk memperbarui data, seperti saat pengguna membuat pesanan.
* **Keuntungan**: MVC memisahkan logika bisnis dari tampilan, memudahkan pengelolaan, pemeliharaan, dan pengembangan aplikasi. Laravel menyediakan fitur bawaan untuk ORM (Eloquent) yang memudahkan pengelolaan database.
* **Kekurangan**: Memiliki learning curve bagi pengembang baru yang belum familiar dengan Laravel atau PHP modern.

**2.2.2. Microservices**

* **Deskripsi**: Arsitektur microservices membagi website menjadi serangkaian layanan kecil dan independent di mana setiap layanan bertanggung jawab atas satu fungsi tertentu, seperti pemesanan, pengguna, atau pembayaran. Layanan-layanan ini saling berkomunikasi melalui API (Application Programming Interface), memungkinkan tiap layanan beroperasi secara mandiri tanpa bergantung pada sistem lain.
* **Keuntungan**: Karena layanan terpisah, setiap bagian aplikasi dapat diskalakan secara independen sesuai kebutuhan. Misalnya, layanan pemesanan yang sering digunakan dapat diperbesar kapasitasnya tanpa harus mempengaruhi bagian lain seperti layanan pengguna atau pembayaran.
* **Kekurangan**: Karena setiap layanan berdiri sendiri dan berkomunikasi melalui API, manajemen komunikasi antar layanan menjadi lebih rumit. Tim perlu merancang protokol komunikasi yang efisien untuk memastikan informasi ditransfer dengan benar dan tepat waktu antara layanan.
* **2.2.3. Single Page Application (SPA)**
* **Deskripsi**: Aplikasi web di mana seluruh aplikasi dimuat dalam satu halaman HTML tunggal, kemudian konten di dalamnya diperbarui secara dinamis menggunakan JavaScript tanpa perlu melakukan pemuatan ulang (reload) halaman secara keseluruhan. Pada SPA, setelah halaman utama dimuat, aplikasi berinteraksi dengan server melalui API atau AJAX untuk mengambil data dan menampilkan informasi yang dibutuhkan pengguna, sehingga memberikan pengalaman yang lebih cepat dan mulus.
* **Keuntungan**: Karena konten diperbarui secara dinamis, pengguna tidak perlu menunggu pemuatan ulang seluruh halaman setiap kali melakukan interaksi. Ini menciptakan pengalaman yang lebih cepat dan seamless, seperti yang sering terlihat pada aplikasi modern seperti Google Maps, Facebook, atau Gmail
* **Kekurangan**: Karena sebagian besar konten SPA dihasilkan secara dinamis menggunakan JavaScript setelah halaman dimuat, mesin pencari tradisional yang mengindeks konten halaman bisa kesulitan "melihat" konten yang dihasilkan setelah proses JavaScript berjalan. SEO untuk SPA memerlukan pendekatan khusus, seperti penggunaan server-side rendering (SSR) atau teknologi seperti Prerendering untuk membuat konten lebih mudah diindeks oleh mesin pencari.

**2.3. Rekomendasi Arsitektur**

* **Rekomendasi**: Arsitektur MVC dengan Laravel sangat cocok untuk aplikasi Mie Ayam Gapuro karena struktur yang jelas memisahkan logika bisnis dan tampilan. Laravel juga mendukung pengembangan aplikasi skala kecil hingga menengah, dengan fleksibilitas untuk berkembang di masa depan.

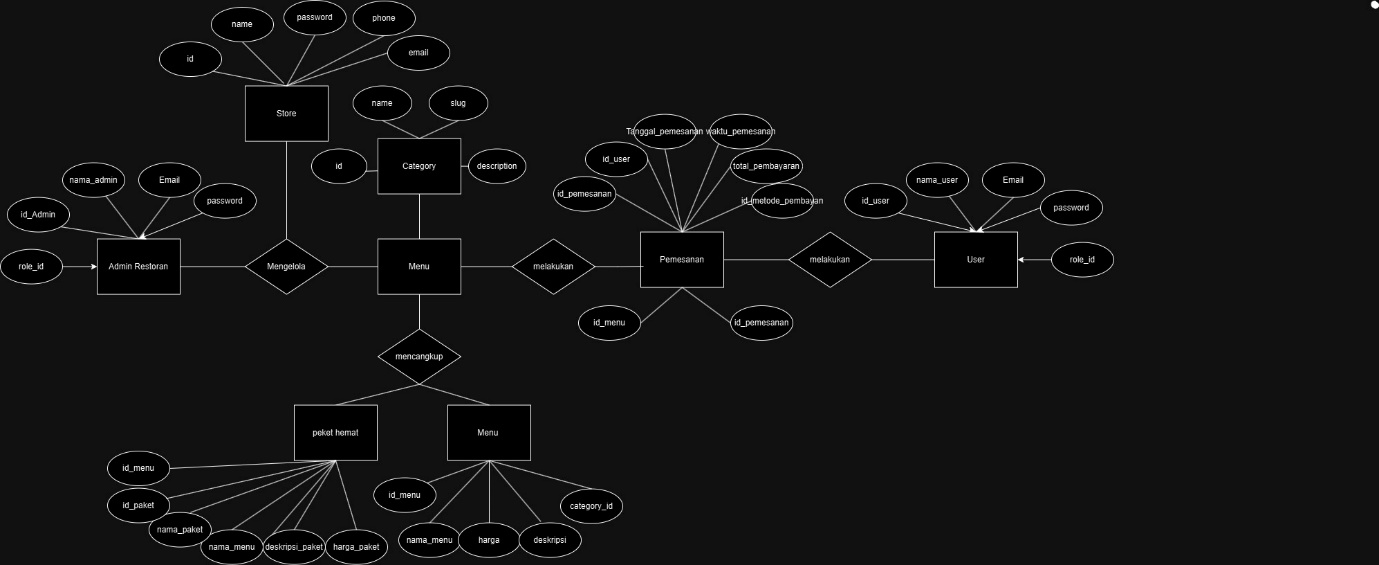
**2. Desain Struktur Database Awal**

**2.1. Tujuan Desain Struktur Database**

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

**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 Admin (Pemilik Restoran)**
  + **Kolom**: id\_admin (PK), nama\_admin, email, password, role\_id
* **Tabel User**
  + **Kolom**: id\_user (PK), nama\_user, email, password, role\_id
* **Tabel Menu**
  + **Kolom**: id\_menu (PK), nama\_menu, harga, deskripsi, category\_id
* **Tabel Paket\_Hemat**
  + **Kolom**: id\_paket (PK), id\_menu, nama\_paket, nama\_menu (FK), deskripsi\_paket, harga\_paket
* **Tabel Pemesanan**
  + **Kolom**: id\_pemesanan (PK), id\_user (FK), tanggal\_pemesanan, waktu\_pemesanan, id\_metode\_pembayaran (FK), total\_pembayaran
* **Tabel Category**
  + **Kolom**: id (PK), name, slug, description
* **Tabel Store**
  + **Kolom**: id (PK), name, password, phone, email

**2.3. Rekomendasi Struktur Database**

* **Rekomendasi**: Struktur database yang diusulkan ini dirancang untuk memastikan integritas data dan efisiensi operasi. Tabel-tabel utama telah diidentifikasi dan hubungan antar tabel diatur untuk mendukung fungsionalitas aplikasi yang diinginkan.

**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**: Blade Template (Laravel)
* **Deskripsi**: Blade adalah engine template bawaan Laravel yang ringan dan fleksibel.
* **Kriteria Pemilihan**: Laravel Breeze memberikan integrasi yang seamless dengan backend

**3.2.2. Teknologi Tambahan**

* **CSS Framework**: Tailwind CSS untuk desain modern dan responsif.
* **Tooling**: Laravel Mix untuk pengelolaan bundling asset (CSS dan JS).

**3.3. Pemilihan Teknologi Backend**

**3.3.1. Bahasa Pemrograman dan Framework**

* **Pilihan**: Laravel (PHP Framework)
* **Deskripsi**: Laravel menyediakan fitur komprehensif untuk pengelolaan logika aplikasi, manajemen database (ORM), dan keamanan.
* **Kriteria Pemilihan**: Laravel terkenal akan kemudahan pengembangan, dokumentasi yang lengkap, dan komunitas yang luas.

**3.3.2. Basis Data**

* **Pilihan**: MySQL
* **Deskripsi**: Sistem manajemen basis data yang digunakan untuk penyimpanan data.
* **Kriteria Pemilihan**: MySQL mendukung operasi transaksi, memiliki skalabilitas baik, dan terintegrasi dengan Laravel melalui Eloquent ORM.

**3.4. Rekomendasi Teknologi**

* **Frontend**: **[pilihan teknologi frontend yang dipilih]** Laravel Breeze dan untuk memberikan pengalaman pengguna yang interaktif dan responsif.
* **Backend**: **[pilihan teknologi backend yang dipilih]** Laravel (PHP) karena fleksibilitas, dukungan komunitas yang luas, dan kemudahan integrasi dengan MySQL.