**2. Pemilihan Arsitektur Aplikasi**

**2.1. Tujuan Pemilihan Arsitektur**

* Menyediakan dasar yang solid untuk pengembangan, pemeliharaan, dan skalabilitas aplikasi.
* Memastikan bahwa arsitektur yang dipilih dapat memenuhi kebutuhan fungsional dan non-fungsional proyek.

**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**: Skalabilitas tinggi, memungkinkan pengembangan dan penyebaran terpisah, fleksibel dalam pemilihan teknologi.
* **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**: Berdasarkan analisis kebutuhan, proyek ini direkomendasikan untuk menggunakan Model-View-Controller berbasis Laravel.Hal ini dikarenakan pemeliharaan yang mudah mulai dari pengembangan, perbaikan, dan pembaruan. Framework Laravel yang mendukung pengembangan web yang cepat deng fitur seperti routing dan autentikasi. Responsif dan skalabel, Keamanan Laravel dilengkapi fitur keamanan bawaan, seperti CSRF, encripsi, dan validasi data sehingga memudahkan pengelolaan dan pengambangan lebih lanjut,

**2. Desain Struktur Database Awal**

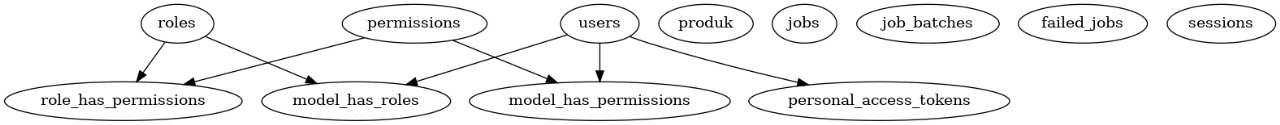
**2.1. Tujuan Desain Struktur Database**

* Memastikan kemudahan akses, pengelolaan dan integritas data.
* Mengorganisasi data toko roti secara efisien untuk mendukung operasional platform online.

**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 Pengguna**
  + **Kolom**: id(PK), name, email, email\_verified\_at, password, remember\_token, created\_at, update\_at, role.
* **Tabel Produk**
  + **Kolom**: id(PK), nama\_produk, deskripsi, harga, stok, foto, created\_at, update\_at, delete\_at.

**2.3. Rekomendasi Struktur Database**

* **Rekomendasi**: Struktur database untuk platform toko roti online “Nindy Bakery” dirancang untuk mendukung operasional yang efisien dan pengelolaan data yang terorganisir.

**3. Pemilihan Teknologi Frontend dan Backend**

**3.1. Tujuan Pemilihan Teknologi**

* Memastikan bahwa platform dapat dikembangkan dengan efisien, mudah dipelihara, dan menawarkan pengalaman pengguna yang optimal.
* Memenuhi kebutuhan fungsionalitas, performa, keamanan, dan skalabilitas, serta dapat dengan mudah diintegrasikan antara front-end dan back-end.

**3.2. Pemilihan Teknologi Frontend**

**3.2.1. Framework/Library**

* **Pilihan**: Laravel
* **Deskripsi**: Framework/library untuk membangun antarmuka pengguna yang interaktif.
* **Kriteria Pemilihan**: Laravel dirancang untuk menangani permintaan HTTP dengan cepat, menggunakan fitur seperti caching bawaan dan optimisasi query melalui Eloquent ORM.

**3.2.2. Teknologi Tambahan**

* **CSS Framework**: Tailwind CSS
* **Tooling**: Visual Studio Code

**3.3. Pemilihan Teknologi Backend**

**3.3.1. Bahasa Pemrograman dan Framework**

* **Pilihan**: PHP (Laravel)
* **Deskripsi**: Bahasa dan framework untuk pengembangan logika aplikasi dan layanan backend.
* **Kriteria Pemilihan**: Kemudahan penggunaan dan pengembangan cepat, Keamanan siap pakai, Struktur dan Organisasi kode yang jelas untuk pengelolaan routing, pengujian, migrasi ke database. Skalabilitas dan performa baik, Fitur bawaan untuk pengembangan web yang canggih yang dibutuhkan dalam pengembangan aplikasi web.

**3.3.2. Basis Data**

* **Pilihan**: MySQL
* **Deskripsi**: Sistem manajemen basis data relational (RDBMS) yang digunakan untuk penyimpanan data.
* **Kriteria Pemilihan**: Performa kinerja yang baik, Mendukung berbagai platform dan Bahasa pemrograman, termasuk PHP dan Laravel, Keamanan yang baik.

**3.4. Rekomendasi Teknologi**

* **Frontend**: **[Tailwind CSS]** karena CSS Framework seperti Tailwind mempercepat styling dengan desain responsif
* **Backend**: **[Laravel (PHP**)**]** karena kemudahan pengelolaan fitur