**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.

**2.3. Rekomendasi Arsitektur**

* **Rekomendasi**: Berdasarkan analisis kebutuhan, proyek ini direkomendasikan untuk menggunakan Model-View-Controller. Hal ini karena Memiliki kelebihan,terutama dalam hal struktur, pemisahan tanggung jawab, dan kemudahan pemeliharaan.

**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**: [Sertakan diagram ER yang relevan]

**2.2.2. Tabel-Tabel Utama**

* **Tabel Pengguna**
  + **Kolom**: UserID (PK), Username, PasswordHash, Email, CreatedAt, UpdatedAt
* **Tabel Produk**
  + **Kolom**: ProductID (PK), ProductName, Description, Price, StockQuantity, CreatedAt, UpdatedAt
* **Tabel Pesanan**
  + **Kolom**: OrderID (PK), UserID (FK), OrderDate, TotalAmount, Status
* **Tabel DetailPesanan**
  + **Kolom**: OrderDetailID (PK), OrderID (FK), ProductID (FK), Quantity, Price

**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**: React, Angular, Vue.js
* **Deskripsi**: Framework/library untuk membangun antarmuka pengguna yang interaktif.
* **Kriteria Pemilihan**: [misalnya, performa, komunitas, dokumentasi, kemudahan integrasi dengan backend]

**3.2.2. Teknologi Tambahan**

* **CSS Framework**: Bootstrap,
* **Tooling**: Webpack, Babel

**3.3. Pemilihan Teknologi Backend**

**3.3.1. Bahasa Pemrograman dan Framework**

* **Pilihan**: PHP(Laravel)
* **Deskripsi**: .
* **Kriteria Pemilihan**: Ekosistem luas,flexebilitas dan skabilitas yang tinggi

**3.3.2. Basis Data**

* MySQL
* **Deskripsi**: MySQL dikenal karena kemampuannya menangani data dalam skala besar dengan kinerja yang baik,sehingga ideal untuk aplikasi dengan banyak pengguna.
* **Kriteria Pemilihan**: Skalabilitas yanng tinggi, kompatibilitas dengan banyak platfrom, kecepatan dan efisiensi, Multi-User dan Multi-Threaded, keamanan.

**3.4. Rekomendasi Teknologi**

* **Frontend**: Bootstrap karena menyediakan banyak komponen yang siap digunakan .
* **Backend**: 1. Laravel karena Mudah diintegrasikan dengan database MySQL.

1. MySQL karena skema database yang terstruktur sesuai dengan relasi yang dibutuhkan.
2. Apache karena mudah dikonfigurasi dan banyak modul yang tersedia.