**2. Pemilihan Arsitektur Aplikasi**

**2.1. Tujuan Pemilihan Arsitektur**

* Menyediakan fondasi yang kokoh untuk pengembangan, pemeliharaan, dan skalabilitas aplikasi laundry. Aplikasi harus dapat menangani pemesanan layanan, pelacakan status laundry, dan pengelolaan pengguna (Admin, Operator Laundry, dan Pelanggan)

**2.2. Pilihan Arsitektur**

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

* **Deskripsi**:

MVC membagi aplikasi laundry menjadi Model (data pelanggan, pesanan, dan status laundry), View (menyajikan antarmuka pelanggan dan admin), dan Controller (menghubungkan input pelanggan dengan model).

* **Keuntungan**:

1. Struktur modular memudahkan pengembangan fitur baru seperti integrasi pembayaran atau laporan pelanggan. Cocok untuk sistem laundry dengan alur pemesanan dan pelacakan sederhana.
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 pemesanan, pelacakan, dan manajemen pengguna yang dapat berjalan mandiri.

* **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 laundry memuat satu halaman HTML dengan konten yang diupdate secara dinamis.

* + **Keuntungan**:

Pengalaman pengguna yang lebih interaktif dengan respon cepat saat melakukan pemesanan atau pelacakan laundry.

* + **Kekurangan**:

SEO mungkin terbatas, namun aplikasi laundry lebih fokus pada pengalaman pengguna terdaftar.

**2.3. Rekomendasi Arsitektur**

Untuk aplikasi laundry, MVC adalah pilihan yang paling cocok. Struktur ini memudahkan pengelolaan data pelanggan, pesanan, dan status Pesanan, serta integrasi dengan fitur pembayaran.

**2. Desain Struktur Database Awal**

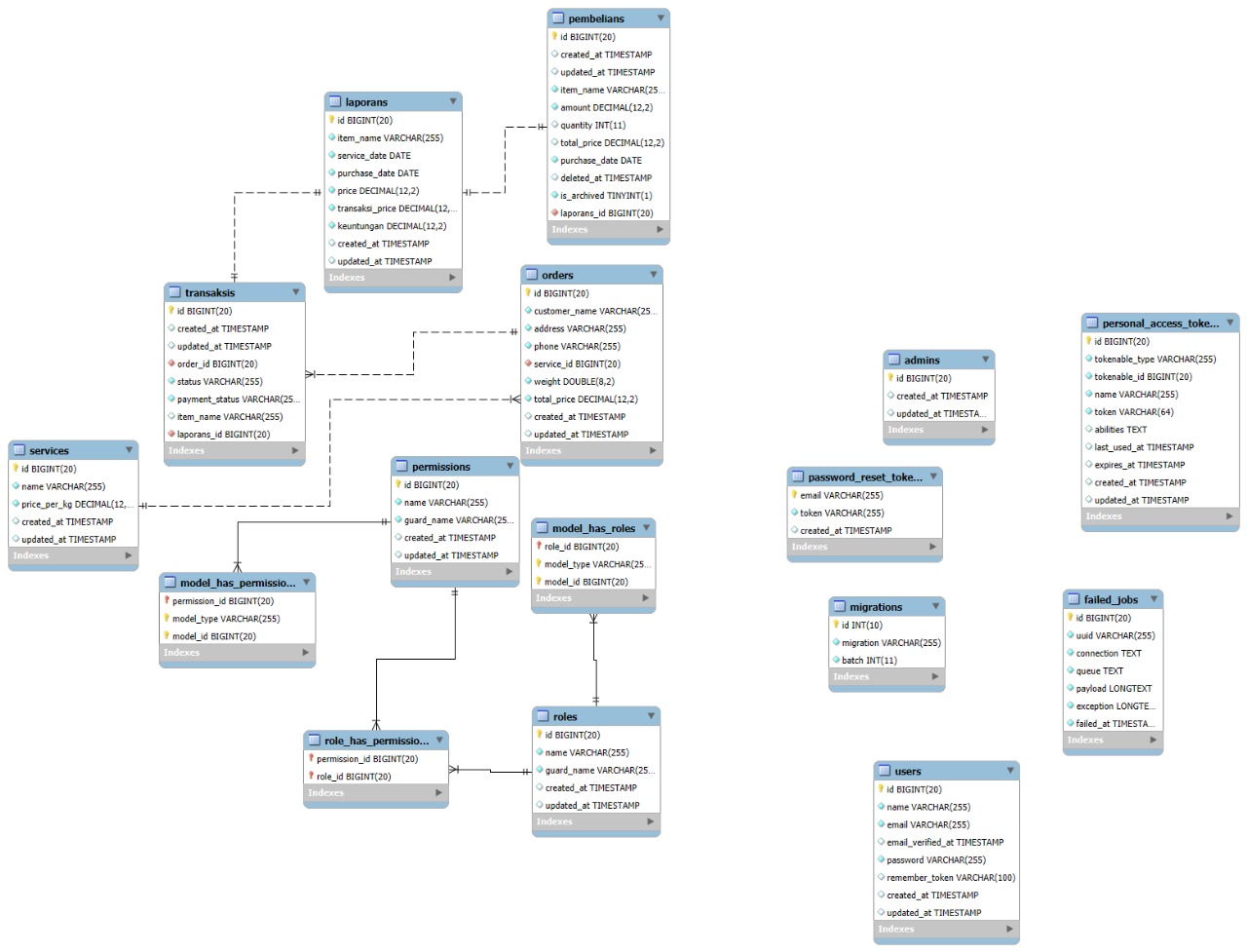
**2.1. Tujuan Desain Struktur Database**

* Memastikan penyimpanan data yang efisien dan terstruktur, sehingga pesanan dan informasi pelanggan dapat dikelola dengan baik, termasuk status laundry dan manajemen pembayaran.

**2.2. Desain Struktur Database**

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

* **Deskripsi Diagram**:



* + Entitas utama termasuk User, Order, Transaksi, Layanan , Pembelian,dan Laporan.

**2.2.2. Tabel-Tabel Utamas**

* **Tabel User** 
  + **Kolom**: UserID (PK), Username, PasswordHash, Email, CreatedAt, UpdatedAt
* **Tabel Pembelian**
  + **Kolom**: ID (PK), ItemsName, Amuont, Quantity, Servis, TotalPrice, ID,Name, Price per KG.
* **Tabel Order**
  + **Kolom**: ID (PK), CustomerName (FK), Addres, Phone, ServisID, Wight, TotalPrice.
* **Tabel Transaksi** 
  + **Kolom**: ID (PK), OrderID (FK), Status (FK), Payment\_Status

**2.3. Rekomendasi Struktur Database**

* Struktur database ini dirancang untuk mengelola semua data penting dari pelanggan, admin, dan pesanan. Relasi antar tabel memastikan data dan efisiensi dalam pengelolaan transaksi dan status laundry.

**3. Pemilihan Teknologi Frontend dan Backend**

**3.1. Tujuan Pemilihan Teknologi**

* Memilih teknologi yang tepat untuk memastikan kinerja sistem laundry 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 laundry, termasuk fitur manajemen pesanan, pembayaran, dan pengguna.
* **Kriteria Pemilihan**: Laravel mendukung pengelolaan transaksi dengan baik, skalabilitas, serta keamanan.

**3.3.2. Basis Data**

* **Pilihan**: MySQL
* **Deskripsi**: MySQL cocok untuk aplikasi laundry yang memerlukan penyimpanan data pelanggan dan pesanan.
* **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