**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**: Menggunakan MVS karena arsitektur ini telah terintergrasikan dengan cukup baik, memberikan alat dan dokumentasi yang memadai untuk pengembangan web Spatuan

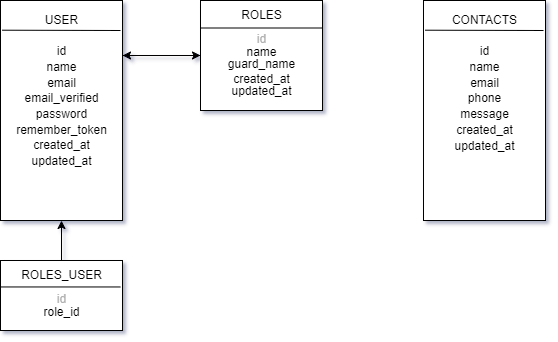
**2. Desain Struktur Database Awal**

**2.1. Tujuan Desain Struktur Database**

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

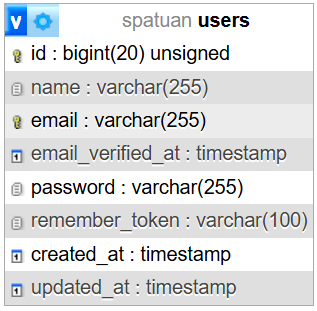
**2.2. Desain Struktur Database**

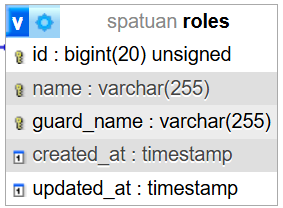
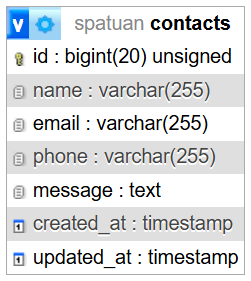
**2.2.1. Diagram Entity-Relationship (ER)**

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

**2.2.2. Tabel-Tabel Utama**

* **Tabel Users**
  + **Kolom**: id, name, email, email\_verified, password, remember\_token, created\_at, updated\_at



* **Tabel Roles**
  + **Kolom**: id, name, guard\_name, created\_at, update\_at
* **Tabel Contacts**
  + **Kolom**: Id, name, email, phone, message, created\_at, update\_at

**2.3. Rekomendasi Struktur Database**

* **Rekomendasi**: Struktur database yang diusulkan ini dirancang untuk memastikan integritas data dan fungsionaltitas pada review

**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**: Boostrap, js
* **Deskripsi**: Framework CSS untuk mendesain web dengan baik
* **Kriteria Pemilihan**: Mudah diterapkan

**3.2.2. Teknologi Tambahan**

* **HTML dan JavaScript**

**3.3. Pemilihan Teknologi Backend**

**3.3.1. Bahasa Pemrograman dan Framework**

* **Pilihan**: Laravel (PHP)
* **Deskripsi**: Menggunakan framework dengan dasaran MVC
* **Kriteria Pemilihan**: Menyediakan fitur bawaan seperti autentikasi, validasi, dan ORM (Eloquent), Dokumentasi lengkap

**3.3.2. Basis Data**

* **Pilihan**: MySQL
* **Deskripsi**: Sistem manajemen basis data yang digunakan untuk penyimpanan data review jasa.
* **Kriteria Pemilihan:** Kompatibilitas tinggi dengan Laravel

**3.4. Rekomendasi Teknologi**

* **Frontend**: **Boostrap dan Blade Laravel**
* **Backend** Laravel dengan MySQL sebagai database