Analisis Kasus

1. Latar Belakang

Keamanan sistem informasi merupakan aspek krusial dalam pengembangan perangkat lunak, terutama pada aplikasi web yang rentan terhadap berbagai serangan siber seperti SQL Injection, Cross-Site Scripting (XSS), dan Cross-Site Request Forgery (CSRF). Oleh karena itu, perlu adanya pendekatan pengembangan sistem yang tidak hanya fokus pada fungsionalitas, tetapi juga pada aspek keamanan dan ketahanan terhadap potensi ancaman.

Proyek **Tugas-KSI** ini merupakan bagian dari implementasi mata kuliah Keamanan Sistem Informasi (KSI), di mana mahasiswa diminta untuk membangun sebuah aplikasi berbasis Laravel yang berjalan di lingkungan **kontainerisasi** menggunakan Docker. Lingkungan ini mencakup layanan aplikasi (Laravel + PHP), web server (NGINX), serta basis data (MySQL), yang diatur dan dijalankan melalui Docker Compose. Tujuan utamanya adalah untuk membangun sistem yang aman, modular, dan mudah diatur.

2. Permasalahan

Beberapa permasalahan utama yang ingin diselesaikan dalam proyek ini antara lain:

- **Bagaimana mengatur lingkungan pengembangan yang aman dan terisolasi** untuk aplikasi Laravel?
- **Bagaimana cara menerapkan konfigurasi server yang sesuai standar keamanan**, termasuk pengaturan SSL/TLS, reverse proxy, dan manajemen container?
- **Bagaimana membangun fondasi arsitektur aplikasi Laravel yang siap dikembangkan** lebih lanjut dan aman untuk digunakan dalam lingkungan jaringan terbuka?

3. Solusi yang Diterapkan

Untuk menjawab permasalahan tersebut, proyek ini membangun arsitektur sistem berbasis kontainer menggunakan Docker Compose yang terdiri atas tiga layanan utama:

1. **Layanan `app` (PHP-FPM + Laravel)**
   Berisi aplikasi Laravel yang disusun pada direktori `src/`. Laravel dipilih karena framework ini sudah menyediakan berbagai fitur keamanan bawaan, seperti proteksi CSRF, hashing password, dan validasi input.
2. **Layanan `nginx`**
   Berfungsi sebagai reverse proxy yang mengarahkan trafik HTTP dan HTTPS ke layanan

`app`. Sertifikat SSL dummy (`pemweb.test.crt` dan `pemweb.test.key`) disertakan untuk simulasi penggunaan HTTPS secara lokal, mendukung praktik *secure web development*.

3. **Layanan `db` (MySQL)**
   Menyediakan basis data untuk Laravel. Konfigurasi khusus disediakan dalam folder `conf.d/` untuk menyesuaikan keamanan database.

Konfigurasi lengkap berada di dalam file `docker-compose.yml`, dengan masing-masing layanan memiliki `Dockerfile` dan konfigurasi tersendiri. Komponen tambahan seperti `local.ini`, `www.conf`, dan `default.conf` mengatur performa serta keamanan runtime PHP dan NGINX.

4. Analisis Keamanan dan Implementasi

Beberapa poin penting dalam penerapan aspek keamanan antara lain:

- **Isolasi Layanan**: Setiap komponen berjalan dalam kontainernya masing-masing. Hal ini mengurangi kemungkinan satu layanan memengaruhi atau dieksploitasi dari layanan lainnya.
- **SSL/TLS**: NGINX telah dikonfigurasi menggunakan HTTPS untuk menjaga keamanan transmisi data, meskipun masih menggunakan sertifikat lokal.
- **Deployment Local yang Aman**: Dengan tidak mengekspose port layanan aplikasi langsung ke luar, hanya layanan NGINX yang dapat diakses, yang merupakan praktik umum dalam arsitektur berbasis container.

Namun, perlu dicatat bahwa direktori `src/` masih dalam tahap awal dan belum mengandung modul-modul fungsional seperti `routes`, `controllers`, maupun `views` yang menjadi inti dari aplikasi Laravel. Ini menunjukkan bahwa proyek ini masih pada tahap setup infrastruktur dan belum masuk ke tahap pengembangan fitur aplikasi secara menyeluruh.

5. Kesimpulan dan Rekomendasi

Proyek **Tugas-KSI** telah berhasil menyusun arsitektur awal dari sistem informasi yang berbasis Laravel dengan pendekatan DevSecOps sederhana. Struktur folder yang jelas, pemisahan tanggung jawab antar layanan, serta penerapan SSL lokal menunjukkan bahwa sistem ini telah memenuhi beberapa prinsip dasar keamanan dalam pengembangan aplikasi.

Namun, beberapa hal berikut disarankan untuk pengembangan selanjutnya:

- Menyusun dokumentasi (`README.md`) yang menjelaskan langkah menjalankan proyek, struktur sistem, serta tujuan pengamanan yang diimplementasikan.
- Melanjutkan pengembangan aplikasi Laravel pada direktori `src/` agar dapat mendemonstrasikan alur kerja sistem lengkap.

- Menambahkan middleware keamanan Laravel seperti autentikasi, rate limiting, logging, dan input validation.
- Jika ditujukan untuk deployment publik, maka sertifikat SSL perlu diganti dengan sertifikat valid dari otoritas seperti Let's Encrypt.

Dengan menyelesaikan tahapan berikutnya, proyek ini berpotensi menjadi blueprint dari sistem informasi yang aman, modular, dan dapat digunakan dalam pembelajaran maupun praktik pengembangan sistem modern.