UTS Keamanan Sistem Informasi - KJ002

Nama : Bryan Frisco Peraira
NIM : 20230801224

---

1. Pendahuluan

Sebuah universitas membangun sistem informasi akademik berbasis web menggunakan framework Laravel. Sistem ini digunakan untuk mengelola data mahasiswa, sesi login pengguna, sistem cache, dan pengelolaan antrian tugas latar belakang (jobs).

2. Deskripsi Sistem

Aplikasi yang dikembangkan menggunakan framework Laravel terdiri dari empat modul utama:

1. **Modul Pengguna dan Sesi**
    - Tabel users: Menyimpan data pengguna (nama, email, password, verifikasi email).
    - Tabel password_reset_tokens: Menyediakan mekanisme reset password.
    - Tabel sessions: Menangani sesi aktif pengguna.
2. **Modul Cache**
    - Tabel cache: Menyimpan data cache aplikasi.
    - Tabel cache_locks: Mengatur mekanisme penguncian cache.
3. **Modul Antrian Pekerjaan**
    - Tabel jobs: Menyimpan pekerjaan terjadwal atau yang sedang antre.
    - Tabel job_batches: Menangani batch jobs.
    - Tabel failed_jobs: Log pekerjaan yang gagal.
4. **Modul Data Mahasiswa**
    - Tabel students: Menyimpan data mahasiswa (NIM, nama, fakultas).

3. Identifikasi Aset

Aset	**Deskripsi**
Data Pengguna	Nama, email, password (hash), status verifikasi
Token Reset Password	Token unik untuk reset password
Sesi Aktif	Informasi sesi, user agent, IP, payload

Data Cache	Potongan data aplikasi yang di-cache
Lock Cache	Informasi kunci untuk menghindari race conditions pada cache
Antrian Pekerjaan	Payload dan status pekerjaan asynchronous
Log Gagal Pekerjaan	Detail exception untuk debugging dan audit
Data Mahasiswa	NIM, nama, fakultas

4. Analisis Ancaman dan Kerentanan

Komponen	Ancaman	Kerentanan
Autentikasi	- Brute force login- Credential stuffing	- Kebijakan password lemah- Tidak ada rate-limiting
Reset Password	- Token hijacking- Replay attack	- Token tidak terenkripsi- Validitas token lama
Sesi	- Session hijacking- Session fixation	- ID sesi prediktif- Payload sensitif tanpa enkripsi
Cache	- Cache poisoning- Race condition lock bypass	- Data sensitif di-cache- Lock expiration lama
Antrian Pekerjaan	- Denial of Service (DoS) antrean- Pekerjaan berbahaya	- Payload tidak tervalidasi- Tidak ada pembatasan
Data Mahasiswa	- Kebocoran data pribadi mahasiswa- SQL injection	- Input user tidak di-sanitasi- Akses kontrol lemah

5. Rekomendasi Kontrol Keamanan

1. **Autentikasi & Password**
    - Terapkan kebijakan password kuat (minimal 12 karakter, kombinasi huruf, angka, simbol).
    - Implementasi rate-limiting pada endpoint login.
    - Gunakan hashing modern (bcrypt/Argon2) dan salt.
2. **Reset Password**
    - Enkripsi token reset di database.
    - Batasi masa berlaku token (misalnya 15 menit).
    - Implementasikan one-time token dan hentikan penggunaan ulang.
3. **Manajemen Sesi**
    - Gunakan sesi aman (HTTP-only cookie, Secure flag).

- Regenerasi ID sesi setelah login.
- Batasi masa berlaku sesi dan implementasi idle timeout.

4. **Cache & Lock**
    - Simpan hanya data yang tidak sensitif di cache.
    - Terapkan integrity checks pada data cache (misalnya HMAC).
    - Pastikan expiration lock sesuai waktu maksimum operasi.
5. **Antrian Pekerjaan**
    - Validasi dan sanitasi payload sebelum dijalankan.
    - Batasi jumlah retry dan implementasikan circuit breaker.
    - Monitor antrean untuk mendeteksi lonjakan abnormal.
6. **Data Mahasiswa**
    - Sanitasi dan validasi input user (ORM Laravel mencegah SQLi).
    - Terapkan kontrol akses (hanya admin dapat mengelola data mahasiswa).
    - Enkripsi data sensitif jika diperlukan.

6. Kesimpulan

Studi kasus ini menggambarkan pentingnya penerapan prinsip keamanan pada setiap lapisan aplikasi: mulai dari autentikasi, manajemen sesi, cache, antrian pekerjaan, hingga penyimpanan data mahasiswa. Dengan mengidentifikasi ancaman dan kerentanan, serta menerapkan kontrol yang tepat, diharapkan integritas, kerahasiaan, dan ketersediaan sistem dapat terjaga.