

Kelompok 1

Nama	NIM
AMALYAH RIZKY KHAIRUNNISA SUTARTO	222212489
HANA HANIFATUL HASANAH	222212635
INTAN AINUN HAYATI	222212668
LAURENTIUS TRISNA YUWANA	222212703
KEVIN ANANDA PUSPITA	222112133
PUTU DIAN SHINTA PRATIVI	222212822
RAFLIANSYAH DWI SETIAWAN TONDAU	222212827

Bahan Diskusi

- Bagaimana cara meminimalisir risiko keamanan sistem di setiap tahapan SDLC?
 1. Tahap Perencanaan
 - Identifikasi kebutuhan keamanan dengan menetapkan kebutuhan kerahasiaan, integritas, ketersediaan, dan akuntabilitas sistem
 - Analisis risiko potensial dari penyelundup eksternal dan internal
 2. Tahap Analisis
 - Lakukan audit data untuk menghapus informasi yang tidak diperlukan
 - Analisis risiko keamanan pada setiap komponen sistem
 3. Tahap Desain
 - Gunakan protokol yang aman seperti HTTPS
 - Gunakan autentikasi seperti dengan enkripsi *password* dan *two-factor authentication*
 - Terapkan enkripsi untuk data yang tersimpan dan lakukan backup secara rutin
 - Gunakan penghalang akses seperti *firewall*, *private network*, *virtual private network*
 4. Tahap Pengembangan
 - Lakukan *defensive programming* (misal menggunakan blok *try-catch*)
 - Lakukan pengujian yang ketat
 5. Tahap Pengujian
 - Lakukan uji penetrasi untuk mengeksplorasi kelemahan keamanan sistem
 - Lakukan pengujian pemulihan perangkat lunak menggunakan data realistis

- Lakukan uji proses backup

6. Tahap Implementasi

- Terapkan backup berkala dan audit secara rutin
- Lakukan pengujian kerentanan secara periodik
- Lakukan pelatihan staf agar memahami prosedur keamanan, tanyakan saran mereka
- Siapkan *disaster plan*: tetapkan prosedur pemulihan, komunikasi kepada pengguna, dan manajemen krisis

7. Tahap Pemeliharaan

- Terapkan *patch* keamanan terbaru pada perangkat lunak
- Lakukan pengujian keamanan jaringan dan pengecekan *password* secara periodik
- Gunakan *software* yang dapat memantau dan mendeteksi ancaman

8. Tahap Penghentian

- Hapus data sensitif menggunakan teknik penghapusan yang aman
- Dokumentasikan proses untuk memastikan tidak ada kerentanan yang tersisa

- Buatlah User Access Matrix dari sistem yang sedang Anda bangun pada project kelompok.

Use Case	Manajer	Supervisor	Karyawan
Login	CR	CR	CR
Presensi	R	R	CR
Perizinan (Ajukan)	X	R	CR
Perizinan (Konfirmasi)	X	RU	R
Gaji (Kelola)	CRU	X	X
Gaji (Slip)	R	X	R
Kasbon (Ajukan)	X	R	CR
Kasbon (Pembayaran)	X	RU	CR
Informasi Karyawan	CRUD	R	X
Dashboard	R	R	R

- Identifikasi potensi Antrian pada sistem yang sedang Anda bangun pada project kelompok.

1. Login

Potensi Bottleneck: Lonjakan akses pengguna (karyawan, supervisor, manajer) pada waktu bersamaan, terutama saat presensi pagi.

Bottleneck pada:

- Authentication Server: Validasi username dan password yang memakan waktu.
- Database Deadlock: Konflik ketika beberapa pengguna mencoba melakukan pendaftaran atau login ulang secara bersamaan.

2. Mengakses Data Karyawan

Potensi Bottleneck: Banyak pengguna (manajer atau supervisor) mengakses data karyawan secara bersamaan, misalnya saat evaluasi bulanan.

Bottleneck pada:

- Query Database: Query tidak dioptimalkan sehingga lambat saat volume data karyawan besar.
- I/O Disk: Akses berulang terhadap data karyawan tanpa caching.

3. Presensi Karyawan

Potensi Bottleneck: Validasi lokasi GPS memerlukan waktu lebih lama jika banyak karyawan melakukan presensi pada waktu yang sama.

Bottleneck pada:

- Sistem GPS: Permintaan validasi lokasi yang tinggi ke server.
- Penulisan ke Database: Banyak permintaan presensi simultan ke database.

4. Pengajuan Kasbon dan Perizinan

Potensi Bottleneck: Lonjakan pengajuan yang dilakukan bersamaan, misalnya saat pengumuman cuti bersama atau ketersediaan dana tambahan untuk kasbon.

Bottleneck pada:

- Sistem Antrian: Banyaknya pengajuan dalam waktu singkat dapat memperlambat server.
- Database Writes: Konflik pada tabel pengajuan saat banyak data baru ditambahkan.

5. Pengunduhan Slip Gaji

Potensi Bottleneck: Banyak karyawan mengunduh slip gaji pada hari pembayaran secara bersamaan.

Bottleneck pada:

- Server Dokumen: Pembuatan file PDF secara real-time.
- Jaringan: Trafik unduhan yang tinggi menyebabkan kemacetan.

6. Dashboard

Potensi Bottleneck: Banyak pengguna mengakses fitur dashboard secara bersamaan untuk melihat data presensi atau performa toko.

Bottleneck pada:

- Query Database: Query yang kompleks, terutama untuk agregasi data.
- Render Data Real-Time: Visualisasi data dalam grafik dapat membebani server.