TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIV

DATA STORAGE

‘API’

Disusun Oleh :

Ade Fatkhul Anam / 2211104051

SE-06-02

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

TUGAS PENDAHULUAN

SOAL

a. Sebutkan dan jelaskan dua jenis utama **Web Service** yang sering digunakan dalam pengembangan aplikasi.
b. Apa yang dimaksud dengan **Data Storage API**, dan bagaimana API ini mempermudah pengelolaan data dalam aplikasi?
c. Jelaskan bagaimana proses kerja komunikasi antara klien dan server dalam sebuah Web Service, mulai dari permintaan (*request*) hingga tanggapan (*response*).
d. Mengapa keamanan penting dalam penggunaan **Web Service**, dan metode apa saja yang dapat diterapkan untuk memastikan data tetap aman?

JAWABAN

a.

1. **SOAP (Simple Object Access Protocol):**

Pengertian: Protokol berbasis XML yang digunakan untuk pertukaran informasi terstruktur antara sistem dalam jaringan.

Keunggulan:

Mendukung keamanan tingkat tinggi (misalnya, WS-Security).

Cocok untuk skenario bisnis yang membutuhkan transaksi yang kompleks.

Kelemahan: Cenderung lebih berat karena format XML yang verbose.

Penggunaan: Digunakan dalam aplikasi yang memerlukan interoperabilitas tingkat tinggi dan kebutuhan keamanan yang kuat.

2. **REST (Representational State Transfer):**

Pengertian: Arsitektur berbasis prinsip HTTP yang memanfaatkan metode HTTP (GET, POST, PUT, DELETE) untuk komunikasi.

Keunggulan:

Ringan dan mudah diimplementasikan.

Mendukung format data yang beragam (JSON, XML, dll.).

Kelemahan: Tidak memiliki standar keamanan bawaan seperti SOAP.

Penggunaan: Umumnya digunakan dalam aplikasi modern seperti API untuk

aplikasi mobile dan web.

b. **Definisi:** Data Storage API adalah antarmuka yang menyediakan cara untuk menyimpan, membaca, mengupdate, dan menghapus data dalam sistem penyimpanan (seperti basis data, cloud storage, atau penyimpanan lokal) melalui program.

Manfaat:

- Abstraksi Data: Mengurangi kompleksitas pengelolaan data dengan menyediakan fungsi standar untuk interaksi dengan sistem penyimpanan.
- Efisiensi: Pengembang dapat fokus pada logika aplikasi tanpa memikirkan detail implementasi penyimpanan data.
- Portabilitas: Memungkinkan aplikasi bekerja dengan berbagai sistem penyimpanan tanpa mengubah kode secara signifikan.

Contoh: Firebase Realtime Database API, Google Cloud Storage API, dan AWS S3 API.

c.

1. Permintaan (Request):

Klien mengirimkan permintaan ke server menggunakan protokol HTTP/HTTPS.

Permintaan ini mencakup:

- URL Endpoint: Lokasi spesifik layanan.
- Metode HTTP: GET (mengambil data), POST (mengirim data), PUT (memperbarui data), DELETE (menghapus data).
- Header: Informasi tambahan seperti otentikasi dan format data (JSON/XML).
- Body: Data tambahan (opsional, biasanya digunakan dalam POST atau PUT).

2. Pemrosesan di Server:

- Server menerima permintaan dan memprosesnya berdasarkan endpoint dan metode HTTP yang diminta.

- Server mungkin berinteraksi dengan basis data, layanan lain, atau logika internal untuk menghasilkan respons.

3. Tanggapan (Response):

- Server mengirimkan tanggapan ke klien dalam format tertentu (misalnya, JSON atau XML).
- Tanggapan ini mencakup:

  1. Status Code: Indikator status permintaan (misalnya, 200 untuk berhasil, 404 untuk tidak ditemukan, 500 untuk kesalahan server).

  2. Body: Data atau pesan hasil dari pemrosesan server.

d.

Pentingnya Keamanan:

- Melindungi data sensitif seperti informasi pengguna, kredensial, dan transaksi keuangan.
- Mencegah serangan seperti pencurian data (data breach), serangan DDoS, atau manipulasi data.
- Menjamin kepercayaan antara penyedia layanan dan pengguna.

Metode Keamanan yang Dapat Diterapkan:

- HTTPS: Menggunakan protokol HTTPS untuk mengenkripsi data selama transmisi antara klien dan server.

Autentikasi dan Otorisasi:

- API Key: Memberikan akses hanya kepada klien yang memiliki kunci API yang valid.
- OAuth: Sistem autentikasi berbasis token yang aman dan sering digunakan.
- Validasi Input: Memastikan data yang diterima dari klien aman dari serangan seperti SQL injection atau XSS (Cross-Site Scripting).
- Rate Limiting: Membatasi jumlah permintaan yang dapat dilakukan oleh klien untuk mencegah penyalahgunaan atau serangan DDoS.
- Pemantauan dan Logging: Memantau aktivitas API untuk mendeteksi dan menangani potensi ancaman secara cepat.

- CORS (Cross-Origin Resource Sharing): Mengatur akses dari domain yang berbeda untuk mencegah eksploitasi melalui browser.

Dengan pendekatan-pendekatan ini, integritas, kerahasiaan, dan ketersediaan data dalam Web Service dapat terjaga.