TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIV

DATA STORAGE
'API'

Disusun Oleh :

SALMAN ALFA

RIZZI

2211104056

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. Dua Jenis Utama Web Service

1. **RESTful Web Service (REST):**

   • REST (Representational State Transfer) adalah jenis web service yang menggunakan protokol HTTP untuk komunikasi antara klien dan server.

   • REST menggunakan metode HTTP seperti GET, POST, PUT, DELETE untuk operasi CRUD.

   • Data dikirim dan diterima dalam format ringan seperti JSON atau XML.

   • REST terkenal karena kesederhanaan, skalabilitas, dan fleksibilitasnya dalam integrasi dengan aplikasi modern, termasuk aplikasi web dan mobile.

2. **SOAP Web Service (Simple Object Access Protocol):**

   • SOAP adalah protokol yang lebih ketat dan berstruktur, menggunakan XML untuk komunikasi antara klien dan server.

   • SOAP menyediakan fitur keamanan bawaan seperti WS-Security, yang membuatnya cocok untuk aplikasi enterprise yang memerlukan transaksi kompleks dan aman.

   • SOAP mendukung berbagai protokol selain HTTP, seperti SMTP atau TCP.

b. Apa itu Data Storage API dan Fungsinya

Data Storage API adalah antarmuka pemrograman yang menyediakan cara bagi aplikasi untuk menyimpan, mengakses, dan mengelola data secara efisien, baik di cloud maupun lokal. Contoh API ini termasuk Firebase Realtime Database, SQLite, atau AWS DynamoDB.

Bagaimana Data Storage API Mempermudah Pengelolaan Data:

1. **Abstraksi Kompleksitas:** API ini menyederhanakan proses penyimpanan dan pengambilan data tanpa pengembang harus mempelajari detail teknis seperti query database yang rumit atau manajemen infrastruktur.

2. **Sinkronisasi Real-Time:** Banyak Data Storage API (seperti Firebase) mendukung sinkronisasi data secara real-time antara klien dan server, mempermudah pengelolaan data pada aplikasi dengan banyak pengguna.

3. **Keamanan dan Otorisasi:** Data Storage API biasanya dilengkapi fitur keamanan seperti autentikasi pengguna dan kontrol akses untuk melindungi data.

c. Proses Kerja Komunikasi antara Klien dan Server dalam Web Service

1. **Permintaan (Request):**
   - Klien mengirimkan permintaan ke server menggunakan protokol HTTP/HTTPS.
   - Permintaan mencakup informasi seperti URL, metode HTTP (GET, POST, dll.), header, dan body (jika ada data yang dikirim).
2. **Pemrosesan di Server:**
   - Server menerima permintaan dan memprosesnya berdasarkan endpoint dan metode HTTP yang diminta.
   - Server dapat berinteraksi dengan database atau layanan lain untuk memproses data.
3. **Tanggapan (Response):**
   - Server mengirimkan tanggapan kembali ke klien.
   - Tanggapan mencakup **status kode HTTP** (seperti 200 OK atau 404 Not Found), header, dan body (biasanya data dalam format JSON atau XML).
4. **Render di Klien:**
   - Klien menerima tanggapan dan menampilkan data atau menjalankan logika bisnis berdasarkan respons tersebut.

d. Pentingnya Keamanan dalam Penggunaan Web Service

Keamanan penting dalam penggunaan web service untuk melindungi data sensitif, mencegah serangan seperti pencurian data, spoofing, dan manipulasi data, serta menjaga kepercayaan pengguna.

Metode untuk Memastikan Keamanan:

1. **HTTPS (Hypertext Transfer Protocol Secure):**
   - Mengamankan komunikasi antara klien dan server dengan enkripsi melalui SSL/TLS.
2. **Autentikasi:**
   - Menggunakan token seperti **OAuth2**, **API Key**, atau **JWT (JSON Web Token)** untuk memastikan hanya pengguna yang berwenang dapat mengakses layanan.
3. **Validasi Input dan Sanitasi Data:**
   - Mencegah serangan seperti SQL Injection dan Cross-Site Scripting (XSS) dengan memvalidasi input dari pengguna.
4. **Rate Limiting dan Throttling:**
   - Membatasi jumlah permintaan ke server dalam periode tertentu untuk mencegah serangan DDoS.
5. **Enkripsi Data:**

- Menggunakan enkripsi pada data yang disimpan dan data yang dikirim untuk memastikan kerahasiaan.

6. **Pengelolaan CORS (Cross-Origin Resource Sharing):**

- Membatasi domain mana yang dapat mengakses API untuk mencegah permintaan yang tidak sah.