TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIV

DATA STORAGE
‘API’

Disusun Oleh :

Zivana Afra Yulianto/2211104039

S1SE-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.

1. SOAP (Simple Object Access Protocol):
   SOAP adalah protokol berbasis XML yang digunakan untuk pertukaran data terstruktur antara aplikasi. SOAP menggunakan protokol seperti HTTP atau SMTP untuk mengirim pesan. Karakteristik utama SOAP:
   - Standar ketat: Menggunakan WSDL (Web Services Description Language) untuk mendeskripsikan layanan.
   - Keamanan: Mendukung keamanan tingkat tinggi dengan WS-Security.
   - Kompleksitas: Protokol yang lebih kompleks, cocok untuk aplikasi besar dengan kebutuhan integrasi yang rumit.
2. REST (Representational State Transfer):
   REST adalah arsitektur yang memanfaatkan standar HTTP untuk pertukaran data. REST menggunakan format data ringan seperti JSON atau XML. Karakteristik utama REST:
   - Sederhana dan fleksibel: Menggunakan HTTP metode seperti GET, POST, PUT, dan DELETE.
   - Efisien: Mendukung cache untuk meningkatkan kinerja.
   - Popularitas: Lebih banyak digunakan dalam pengembangan aplikasi modern karena kesederhanaannya.

b. Apa yang dimaksud dengan **Data Storage API**, dan bagaimana API ini mempermudah pengelolaan data dalam aplikasi?

Data Storage API adalah antarmuka pemrograman aplikasi yang memungkinkan pengembang untuk menyimpan, mengakses, dan mengelola data dalam penyimpanan yang disediakan oleh platform tertentu (misalnya database, cloud storage, atau file system).

Manfaat Data Storage API:

1. Abstraksi: Pengembang tidak perlu memahami detail teknis penyimpanan data. API menyediakan fungsi-fungsi sederhana untuk operasi seperti `read`, `write`, dan `update`.
2. Efisiensi: API sering kali dioptimalkan untuk performa tinggi, termasuk caching atau kompresi data.
3. Kemudahan integrasi: Mempermudah aplikasi untuk terhubung dengan berbagai layanan penyimpanan seperti Firebase, AWS S3, atau database SQL/NoSQL.
4. Keamanan: Mendukung fitur seperti enkripsi dan kontrol akses untuk menjaga data tetap aman.

c. Jelaskan bagaimana proses kerja komunikasi antara klien dan server dalam sebuah Web Service, mulai dari permintaan (*request*) hingga tanggapan (*response*).

- Permintaan (Request):
  - Klien mengirimkan permintaan ke server melalui protokol seperti HTTP. Permintaan ini mencakup URL endpoint, metode HTTP (GET, POST, dll.), dan data yang diperlukan (jika ada).
  - Contoh: Klien mengirimkan permintaan GET ke `https://api.example.com/users`.
- Pemrosesan di Server:
  - Server menerima permintaan dan memprosesnya sesuai dengan logika bisnis.
  - Server dapat mengakses database atau sistem lain untuk mengambil atau memperbarui data yang diminta.
- Tanggapan (Response):
  - Setelah selesai memproses, server mengirimkan respons kembali ke klien.
  - Respons biasanya dalam format seperti JSON atau XML, yang mencakup kode status HTTP (misalnya 200 untuk sukses, 404 untuk tidak ditemukan).
- Penanganan oleh Klien:
  - Klien menerima respons dan menampilkan data kepada pengguna atau memprosesnya lebih lanjut.

d. Mengapa keamanan penting dalam penggunaan **Web Service**, dan metode apa saja yang dapat diterapkan untuk memastikan data tetap aman?

Keamanan penting untuk melindungi data sensitif, menjaga integritas aplikasi, dan mencegah serangan berbahaya seperti pencurian data, peretasan, atau manipulasi data.

Metode untuk Menjamin Keamanan Web Service:

1. Enkripsi: Menggunakan HTTPS untuk mengenkripsi data selama transmisi.
2. Autentikasi: Memastikan hanya pengguna atau aplikasi yang sah yang dapat mengakses layanan. Contoh: OAuth, API key, JWT (JSON Web Token).
3. Validasi Input: Mencegah serangan seperti SQL Injection dengan memvalidasi data yang masuk.
4. Firewall API: Membatasi akses berdasarkan lokasi IP atau pola penggunaan tertentu.
5. Rate Limiting: Mencegah penggunaan berlebihan atau serangan DDoS dengan membatasi jumlah permintaan dari klien.
6. Audit Log: Mencatat aktivitas untuk mendeteksi anomali dan mendukung analisis forensik.
7. Update Rutin: Memastikan server dan perangkat lunak terus diperbarui untuk mengatasi kerentanan terbaru.