TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIV
DATA STORAGE ‘API’

Disusun Oleh :

Namirah Salsabila

S1SE0601

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?

JAWAB

a. **Simple Object Access Protocol (SOAP)** : yaitu protocol yang menggunakan XML (extensible Markup Language) untuk penukaran data antar aplikasi yang memiliki struktur yang rigid dan kompleks, sehingga keamanan dan integritas data lebih terjamin.

   Representational State of Resource (REST) : arsitektur yang menggunakan HTTP (Hypertext Transfer Protocol) seperti GETS, POST, PUT, dan DELETE untuk mengakses dan memanipulasi.

b. Antarmuka pemrograman aplikasi yang memungkinkan pengembang mengakses, mengelola data yang tersimpan di server dan mempermudah dalam mengelola data. Cara API dalam mempermudah mengelola data adalah :
   - Menghindari kompleksitas pengelolaan basis data
   - Meningkatkan keamanan dan integritas data
   - Menyediakan metode standar untuk mengakses dan memanipulasi data

c. Proses Komunikasi Antara Klien dan Server dalam Web Service

- **Permintaan (Request) :**

  Klien, seperti aplikasi atau browser, mengirimkan permintaan ke server melalui protokol HTTP. Permintaan ini berupa HTTP request yang terdiri dari :

  - Metode HTTP: seperti GET, POST, PUT, atau DELETE.
  - URL/URI: alamat yang menunjukkan sumber daya yang diminta.

- Headers: informasi tambahan, seperti tipe konten atau token autentikasi.
- Body: bagian opsional yang memuat data tambahan, misalnya data formulir pada metode POST.

- **Penerimaan Permintaan oleh Server :**

  Server menerima permintaan klien melalui endpoint yang telah ditentukan. Server kemudian memproses permintaan tersebut sesuai dengan logika bisnis yang telah diimplementasikan.

- **Pemrosesan di Server :**

  Server mengolah data yang diterima dari klien, seperti:

  - Mengakses database untuk membaca atau menyimpan informasi.
  - Menjalankan fungsi atau logika tertentu untuk menghasilkan tanggapan yang sesuai.
  - Setelah pemrosesan selesai, server mempersiapkan data untuk dikirimkan kembali kepada klien.

- **Pengiriman Tanggapan (Response) :**

  Server mengirimkan tanggapan berupa HTTP response kepada klien. Tanggapan ini mencakup:

  - Kode Status HTTP: misalnya 200 OK untuk permintaan yang berhasil, 404 Not Found jika sumber daya tidak ditemukan, atau 500 Internal Server Error jika ada masalah di server.
  - Headers: informasi tambahan seperti tipe konten.
  - Body: data yang diminta, biasanya dalam format JSON atau XML.
  - Klien menerima tanggapan ini untuk ditampilkan atau diproses lebih lanjut.

d. Karena :

- Perlindungan Data Sensitif : Web Service sering menangani data penting seperti informasi pribadi, keuangan, atau data rahasia bisnis. Keamanan mencegah akses tidak sah ke data tersebut.
- Mencegah Penyadapan : Tanpa pengamanan, data yang dikirimkan dapat disadap oleh pihak ketiga melalui serangan seperti man-in-the-middle.
- Mencegah Serangan Siber : Web Service yang tidak aman rentan terhadap berbagai serangan seperti injeksi SQL, DDoS, atau pencurian identitas pengguna.
- Memenuhi Regulasi : Banyak industri diwajibkan untuk mematuhi standar keamanan tertentu (contohnya GDPR, HIPAA, atau PCI DSS). Keamanan

membantu memenuhi regulasi ini.

Metode yang bisa diterapkan :

- Menggunakan HTTPS : Protokol HTTPS mengenkripsi komunikasi antara klien dan server, sehingga data yang dikirimkan tidak dapat diintip oleh pihak luar.
- Autentikasi dan Otorisasi : Autentikasi memastikan hanya pengguna yang valid yang dapat mengakses layanan, menggunakan metode seperti OAuth, API Key, atau JWT.
- Validasi Input : Memeriksa semua data yang dikirimkan oleh klien untuk mencegah serangan seperti injeksi SQL atau XSS.