TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIV
Data Storage (API)

Disusun Oleh :

Satria Ariq Adelard

Dompas/2211104033 SE 06 2

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1. **TUGAS PENDAHULUAN**

1. Jenis Utama Web Service yang Sering Digunakan dalam Pengembangan Aplikasi

- **SOAP (Simple Object Access Protocol):**

SOAP adalah protokol berbasis XML yang dirancang untuk memungkinkan komunikasi antara aplikasi yang berbeda menggunakan protokol jaringan seperti HTTP dan SMTP. SOAP memiliki struktur yang ketat dan menyediakan standar untuk pengiriman pesan yang aman dan dapat diandalkan. Karena keamanannya, SOAP sering digunakan dalam lingkungan yang memerlukan tingkat keamanan tinggi seperti perbankan dan layanan pemerintah.

- **REST (Representational State Transfer):**

REST adalah arsitektur yang menggunakan standar HTTP untuk pertukaran data. REST lebih ringan dibandingkan SOAP karena menggunakan format seperti JSON atau XML untuk mentransfer data. REST sangat populer karena kemudahan implementasinya, fleksibilitasnya, dan efisiensi dalam penggunaan sumber daya.

2. Pengertian Data Storage API dan Fungsinya dalam Pengelolaan Data Aplikasi

Data Storage API adalah antarmuka pemrograman aplikasi yang menyediakan cara bagi pengembang untuk menyimpan, mengelola, dan mengambil data dari sistem penyimpanan tanpa harus memahami detail teknis backend penyimpanan tersebut. API ini menyediakan abstraksi yang memungkinkan aplikasi untuk:

1. Mengakses Data Secara Efisien:
Dengan API, aplikasi dapat melakukan operasi seperti membaca, menulis, memperbarui, dan menghapus data secara langsung melalui endpoint yang telah disediakan.

2. Skalabilitas:
API sering dirancang untuk mendukung pertumbuhan volume data yang besar, sehingga dapat digunakan dalam aplikasi dengan kebutuhan data yang terus berkembang.

3. Kemudahan Integrasi:
Data Storage API dapat dengan mudah diintegrasikan ke berbagai jenis aplikasi, baik yang berbasis web maupun mobile, sehingga pengembang tidak perlu membangun solusi penyimpanan data sendiri dari awal.

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

1. Permintaan (Request):
- Klien (misalnya, aplikasi atau browser) mengirimkan permintaan ke server melalui protokol HTTP/HTTPS.
- Permintaan ini biasanya mencakup informasi seperti metode HTTP (GET, POST, PUT, DELETE), URL endpoint, header, dan data (jika ada).

2. Pemrosesan oleh Server:
- Server menerima permintaan, memvalidasi data, dan mengeksekusi logika bisnis yang relevan.

- Server berkomunikasi dengan sumber daya lain seperti basis data atau layanan eksternal jika diperlukan.

3. Tanggapan (Response):
- Server mengirimkan tanggapan kembali ke klien dalam format tertentu, seperti JSON atau XML.
- Tanggapan ini mencakup kode status HTTP (seperti 200 untuk sukses atau 404 untuk data tidak ditemukan) dan data yang diminta.

4. Pengolahan di Klien:
- Klien menerima tanggapan dan menampilkan data kepada pengguna atau memprosesnya lebih lanjut sesuai kebutuhan aplikasi.

4. Pentingnya Keamanan dalam Penggunaan Web Service dan Metode yang Digunakan

Keamanan sangat penting dalam Web Service untuk melindungi data sensitif, mencegah akses yang tidak sah, dan memastikan integritas komunikasi antara klien dan server. Tanpa keamanan yang memadai, data dapat dicuri, dimanipulasi, atau disalahgunakan.

Metode Keamanan yang Diterapkan:

1. Enkripsi:
- Menggunakan protokol seperti HTTPS untuk mengenkripsi data selama transmisi, sehingga hanya pihak yang berwenang yang dapat membaca data tersebut.

2. Autentikasi:
- Memastikan bahwa hanya pengguna atau sistem yang sah yang dapat mengakses layanan, biasanya melalui API key, OAuth, atau JWT (JSON Web Tokens).

3. Otorisasi:
- Menentukan tingkat akses yang dimiliki pengguna atau aplikasi terhadap sumber daya tertentu, misalnya melalui kontrol akses berbasis peran (RBAC).

4. Firewall dan IP Whitelisting:
- Membatasi akses hanya untuk IP tertentu yang telah disetujui sebelumnya.

5. Rate Limiting:
- Membatasi jumlah permintaan yang dapat dilakukan oleh klien dalam waktu tertentu untuk mencegah serangan DoS (Denial of Service).

6. Validasi Input:
- Memastikan bahwa semua data yang diterima dari klien telah divalidasi untuk mencegah serangan seperti SQL injection atau XSS (Cross-Site Scripting).