**Lamansiang, Wayne Stely**

**Backend parallel B**

1. ***Method Web Service***

Web service adalah sebuah metode komunikasi antara dua perangkat lunak melalui jaringan (seperti internet) dengan menggunakan protokol standar untuk pertukaran data. Metode web service umumnya melibatkan penggunaan protokol seperti HTTP dan XML/JSON untuk pertukaran data antara aplikasi yang berbeda. Terdapat beberapa metode atau gaya dalam implementasi web service, dan dua gaya utama yang umum digunakan adalah REST (Representational State Transfer) dan SOAP (Simple Object Access Protocol).

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

- REST adalah gaya arsitektur yang berfokus pada sederhana, skalabilitas, dan keterbacaan.

- Menggunakan metode HTTP (GET, POST, PUT, DELETE) untuk berkomunikasi.

- Data umumnya dikirimkan dalam format seperti JSON atau XML.

- Stateless, artinya setiap permintaan dari klien ke server berisi semua informasi yang dibutuhkan untuk memahami dan memproses permintaan tersebut.

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

- SOAP adalah protokol komunikasi yang lebih kaya fitur dan formal dibandingkan dengan REST.

- Menggunakan XML untuk pertukaran pesan.

- Biasanya menggunakan HTTP, SMTP, atau protokol lain sebagai transport layer.

- Stateful, artinya setiap permintaan dari klien ke server dapat mempertahankan status sesi.

Kedua metode ini memiliki kelebihan dan kekurangan masing-masing, dan pemilihan antara REST dan SOAP tergantung pada kebutuhan proyek dan preferensi pengembang. REST lebih sederhana dan populer untuk pengembangan web modern, sementara SOAP lebih kompleks tetapi dapat memberikan fungsionalitas yang lebih lengkap dalam beberapa kasus bisnis yang spesifik.

**METODE UMUM DALAM WEB SERVICE :**

* **GET:**

- Metode ini digunakan untuk meminta data dari sumber daya tertentu.

- Biasanya digunakan untuk membaca atau mengambil informasi dari server.

- Tidak seharusnya memiliki efek samping terhadap sumber daya yang diminta.

* **POST:**

- Metode ini digunakan untuk mengirimkan data baru ke sumber daya tertentu untuk diproses.

- Biasanya digunakan untuk membuat sumber daya baru atau mengirim data yang perlu diproses oleh server.

* **PUT:**

- Metode ini digunakan untuk memperbarui atau membuat sumber daya tertentu di server.

- Seharusnya dapat digunakan untuk membuat sumber daya baru atau menggantikan yang sudah ada.

* **DELETE:**

- Metode ini digunakan untuk menghapus sumber daya tertentu di server.

- Perlu digunakan dengan hati-hati karena dapat memiliki efek permanen terhadap data.

* **PATCH:**

- Metode ini digunakan untuk memperbarui data di sumber daya tertentu.

- Mirip dengan PUT, tetapi PATCH digunakan untuk memperbarui hanya bagian-bagian tertentu dari sumber daya, tidak seluruhnya.

* **OPTIONS:**

- Metode ini digunakan untuk mendapatkan informasi tentang opsi atau fitur yang tersedia pada sumber daya tertentu.

- Berguna untuk menjelajahi metode-metode yang didukung oleh sumber daya atau server.

* **HEAD:**

- Metode ini mirip dengan GET, tetapi hanya mengambil header respons dan tidak mengirimkan body respons.

- Berguna untuk mendapatkan informasi header tanpa mengunduh seluruh isi sumber daya.

***2 Komponen - komponen penyusun HTTP Request & HTTP Response***

* **Komponen HTTP Request:**

**Method:**

Menentukan tindakan yang harus dilakukan pada sumber daya. Contoh metode: GET, POST, PUT, DELETE.

* **URL (Uniform Resource Locator):**

Menunjukkan lokasi atau alamat sumber daya yang diminta atau yang akan diubah oleh request.

* **Headers (Header Fields):**

Menyediakan informasi tambahan tentang request atau client. Misalnya, jenis konten yang diterima atau dikirim (Content-Type), tipe bahasa yang diinginkan (Accept-Language), dll.

* **Body:**

Hanya digunakan oleh metode yang mendukung pengiriman data, seperti POST atau PUT.

Berisi data atau payload yang dikirim ke server. Format datanya dapat berupa form data, JSON, XML, atau jenis data lainnya tergantung pada kebutuhan.

* **Parameters:**

Parameter-query (query string) dalam URL untuk menyampaikan informasi tambahan ke server. Contoh: ?key1=value1&key2=value2.

* **Authentication Credentials:**

Dibutuhkan jika sumber daya yang diminta memerlukan autentikasi. Biasanya berupa username dan password.

* **Komponen HTTP Response:**
* **Status Line:**

Menyertakan tiga informasi: versi protokol HTTP, kode status numerik (misalnya 200 OK, 404 Not Found), dan pesan status yang bersifat deskriptif.

* **Headers (Header Fields):**

Memberikan informasi tambahan tentang response atau server. Misalnya, jenis konten yang dikirim (Content-Type), waktu kedaluwarsa (Expires), dll.

* Body:

Berisi data yang dikirimkan dari server ke client sebagai hasil dari permintaan. Formatnya dapat berupa HTML, JSON, XML, atau jenis data lainnya tergantung pada sifat tanggapan.

* **Cookies:**

Informasi yang disimpan oleh browser dan dikirimkan kembali ke server dalam setiap permintaan. Digunakan untuk melacak sesi dan menyimpan informasi pengguna.

* **Authentication Credentials**:

Jika server memerlukan autentikasi, tanggapan dapat mencakup header yang memberikan izin atau token autentikasi.

* **Redirect Information:**

Jika respons adalah tautan redirect, maka informasi redirect akan disertakan dalam header.

1. **HTTP Response Code**

* **HTTP Response Code**

HTTP Response Code, juga dikenal sebagai status code, adalah tiga digit angka yang diberikan oleh server sebagai bagian dari status line dalam respons HTTP. Mereka memberikan informasi tentang hasil dari permintaan HTTP yang dilakukan oleh klien. Setiap kode status memiliki makna khusus dan dapat memberi petunjuk tentang apakah permintaan berhasil, gagal, atau memerlukan tindakan lebih lanjut.

* **1xx (Informational):**

00 Continue: Server menerima header permintaan dan klien seharusnya melanjutkan mengirimkan badan permintaan.

* **2xx (Successful):**

- 200 OK: Permintaan berhasil.

- 201 Created: Sumber daya berhasil dibuat sebagai hasil dari permintaan.

- 204 No Content: Permintaan berhasil, tetapi tidak ada konten yang dikembalikan.

* **3xx (Redirection)**

- 300 Multiple Choices: Terdapat beberapa sumber daya yang cocok, dan klien seharusnya memilih satu.

- 301 Moved Permanently: Sumber daya yang diminta telah dipindahkan secara permanen ke lokasi lain.

- 302 Found: Sumber daya yang diminta sementara berada di lokasi lain. Bergantung pada implementasi, ini bisa dianggap sebagai redirect sementara atau permanen.

* **4xx (Client Error):**

- 400 Bad Request: Permintaan tidak dapat dipahami oleh server karena sintaks yang salah atau permintaan tidak valid.

- 401 Unauthorized: Klien tidak memiliki otorisasi yang diperlukan untuk mengakses sumber daya.

- 403 Forbidden: Server memahami permintaan, tetapi menolak untuk memenuhinya karena batasan akses.

* **5xx (Server Error):**

- 500 Internal Server Error: Terjadi kesalahan server yang tidak dapat diatasi, menyebabkan kegagalan permintaan.

- 502 Bad Gateway: Server bertindak sebagai gateway atau proxy dan menerima respons yang tidak valid dari server upstream.

- 503 Service Unavailable: Server tidak dapat menangani permintaan karena sedang tidak tersedia untuk sementara waktu.