



---

NAMA	: Praditya Fahlefi
NIM	: 22.01.53.0046
PRODI	: Teknik Informatika

---

## WEB SERVICE

**Mardi Siswo Utomo, S.KOM, M.Cs**

### **Rangkuman Request Method Http**

#### **1. GET**

GET adalah metode HTTP yang digunakan untuk mengambil data dari server tanpa mengubah atau memodifikasi data tersebut. Ini adalah metode yang paling umum digunakan dalam aplikasi web ketika client (misalnya, browser) ingin meminta resource dari server, seperti halaman web, gambar, atau data dari API. Ketika kita mengakses halaman web melalui browser, browser mengirimkan request GET ke server untuk meminta halaman tersebut.

**Fungsi Utama :** Digunakan untuk mengambil atau membaca data dari server tanpa mengubahnya.

**Contoh :** Saat kamu mengetik URL di browser dan menekan Enter, browser mengirimkan permintaan GET untuk mengambil konten dari URL yang kamu masukkan.

#### **Karakteristik :**

- **Idempotent:** Artinya, memanggil GET berkali-kali dengan request yang sama tidak akan mengubah status atau hasil di server. Misalnya, jika kita mengirim request GET ke API yang sama berkali-kali, hasil yang diterima tetap akan sama.
- **Safe Method:** Tidak mengubah data di server, hanya mengambilnya.
- **Cacheable:** Respons dari GET sering kali bisa di-cache oleh browser atau proxy untuk mempercepat akses di masa mendatang.
- **Request Body:** Biasanya tidak mengandung body dalam request-nya, semua parameter dikirim melalui URL (query string).

#### **2. POST**

POST adalah metode HTTP yang digunakan untuk mengirimkan data ke server, biasanya untuk membuat resource baru. Data ini dikirim di dalam body request, bukan melalui URL seperti pada GET. Biasanya, POST digunakan ketika kita mengirimkan form, mengunggah file, atau membuat akun di sebuah aplikasi.

**Fungsi Utama :** Digunakan untuk mengirimkan data ke server dan membuat resource baru.

**Contoh :** Ketika pengguna mendaftar di aplikasi, POST /users digunakan untuk mengirimkan data pengguna baru (nama, email, password), dan server akan membuat akun baru berdasarkan data tersebut. Ketika kamu mengisi form pendaftaran akun dan menekan tombol "Submit", data form tersebut dikirim ke server menggunakan request POST untuk membuat akun baru.

#### **Karakteristik :**

- **Non-idempotent:** Setiap kali POST dijalankan, server dapat membuat resource baru yang berbeda atau menghasilkan hasil yang berbeda. Misalnya, mengirim POST dua kali dapat menghasilkan dua resource baru.
- **Create Operation:** Umumnya digunakan untuk operasi pembuatan data.
- **Request Body:** Data yang dikirim disertakan di dalam body request (misalnya, dalam format JSON atau form-encoded).

- Tidak Cacheable: Biasanya tidak boleh di-cache, karena POST mengubah keadaan di server.

### 3. PUT

PUT adalah metode HTTP yang digunakan untuk mengganti resource yang ada di server dengan yang baru. Jika resource yang dituju oleh URL sudah ada, maka akan diganti sepenuhnya dengan data yang baru. Jika tidak ada resource yang cocok, maka PUT dapat membuat resource baru. PUT sering digunakan untuk operasi pembaruan atau update secara menyeluruh.

**Fungsi Utama :** Digunakan untuk mengupdate resource yang sudah ada atau membuat resource jika belum ada.

**Contoh :** Jika pengguna ingin memperbarui profilnya, request PUT dapat digunakan untuk mengganti informasi pengguna dengan data baru yang dikirimkan di dalam request.

**Karakteristik :**

- Idempotent: Memanggil PUT berulang kali dengan data yang sama akan memberikan hasil yang sama; resource yang ada akan selalu diupdate dengan data yang sama setiap kali.
- Replace Operation: Biasanya digunakan untuk menggantikan seluruh resource dengan data yang baru.
- Request Body: Mengandung seluruh data dari resource yang ingin diperbarui atau diganti.
- Tidak Cacheable: Seperti POST, PUT biasanya tidak di-cache karena mengubah resource.

### 4. DELETE

DELETE adalah metode HTTP yang digunakan untuk menghapus resource di server. Saat request DELETE dikirimkan, server akan menghapus resource yang diminta berdasarkan URL yang diberikan. Resource yang telah dihapus tidak akan tersedia lagi untuk diakses setelahnya.

**Fungsi Utama :** Digunakan untuk menghapus resource dari server.

**Contoh :** Ketika kamu ingin menghapus akunmu dari sebuah aplikasi, server menerima request DELETE untuk menghapus data akun dari basis data.

**Karakteristik :**

- Idempotent: Memanggil DELETE berulang kali pada resource yang sama akan menghasilkan hasil yang sama, karena resource hanya bisa dihapus satu kali. Jika resource sudah dihapus, panggilan selanjutnya tidak akan mengubah apapun.
- Permanent Operation: Operasi ini bersifat menghapus data secara permanen (meskipun server bisa mengimplementasikan "soft delete" tergantung pada kebutuhan).
- Tidak Cacheable: Respons dari DELETE biasanya tidak di-cache.

### 5. OPTIONS

OPTIONS adalah metode HTTP yang digunakan untuk meminta informasi tentang metode HTTP yang diizinkan di suatu URL atau resource. Ini biasanya digunakan oleh client untuk mengetahui apa yang diizinkan oleh server (misalnya, apakah server mendukung GET, POST, atau metode lainnya). OPTIONS sering digunakan dalam skenario lintas-domain (CORS).

**Fungsi Utama :** Digunakan untuk memeriksa kemampuan server, seperti metode HTTP apa saja yang didukung oleh server di endpoint tertentu.

**Contoh :** Jika aplikasi client ingin tahu metode HTTP apa saja yang diizinkan pada resource tertentu sebelum melakukan permintaan, client bisa menggunakan OPTIONS untuk memeriksa metode-metode yang tersedia. Request OPTIONS juga dapat memberikan informasi tentang metode HTTP apa saja yang diperbolehkan pada resource pengguna (misalnya GET, PUT, DELETE).

### **Karakteristik :**

- No Side Effects: Tidak memodifikasi resource, hanya digunakan untuk komunikasi metadata antara client dan server.
- Used in CORS: Sangat penting dalam implementasi Cross-Origin Resource Sharing (CORS), di mana browser menggunakan OPTIONS untuk memeriksa apakah permintaan lintas domain diizinkan sebelum mengirim request sebenarnya.

## **6. PATCH**

PATCH adalah metode HTTP yang digunakan untuk mengupdate sebagian resource yang sudah ada. Berbeda dengan PUT yang menggantikan seluruh resource, PATCH hanya mengubah bagian tertentu dari resource tanpa memodifikasi keseluruhan data. PATCH cocok digunakan ketika hanya sebagian kecil dari data yang ingin diubah.

**Fungsi Utama :** Digunakan untuk mengupdate sebagian resource, tidak seperti PUT yang menggantikan seluruh resource.

**Contoh :** Jika pengguna hanya ingin memperbarui alamat email mereka, PATCH dapat digunakan untuk mengirimkan hanya data email yang baru, tanpa mengirim seluruh informasi pengguna. Misalnya, jika kamu ingin mengubah hanya alamat emailmu di profil tanpa mengubah data lain, request PATCH bisa dikirim untuk memperbarui hanya email tanpa mengganti seluruh profil.

### **Karakteristik :**

- Non-idempotent: Panggilan PATCH tidak selalu idempotent. Misalnya, memanggil PATCH dua kali dengan data yang sama bisa saja menghasilkan hasil yang berbeda jika pengubahannya bersifat inkremental.
- Partial Update: Hanya mengupdate sebagian dari resource yang ada.
- Request Body: Hanya berisi bagian-bagian yang ingin diubah, bukan seluruh data resource.

## **7. HEAD**

HEAD adalah metode HTTP yang mirip dengan GET, tetapi hanya mengambil header dari response tanpa body atau konten. Ini berguna ketika client ingin memeriksa status resource (misalnya, apakah resource ada) tanpa benar-benar mengambil datanya. HEAD digunakan dalam monitoring atau untuk memeriksa metadata dari resource.

**Fungsi Utama :** Mirip dengan GET, namun hanya mengambil header dari respons tanpa mengambil body atau konten sebenarnya.

**Contoh :** Jika kamu ingin mengetahui apakah halaman web tersedia tanpa mendownload kontennya, request HEAD bisa digunakan untuk mendapatkan informasi tersebut melalui header response, seperti status HTTP, ukuran konten, atau tipe konten. HEAD /index.html akan memberi informasi apakah file index.html tersedia tanpa mengunduh file tersebut. Browser atau alat monitoring menggunakan metode ini untuk memeriksa status file.

### **Karakteristik :**

- Idempotent: Seperti GET, memanggil HEAD berulang kali tidak akan mengubah status resource.
- No Body: Respons dari HEAD hanya berisi header, tidak ada konten body.
- Used for Checking Availability: Berguna untuk memeriksa apakah resource tersedia atau untuk mendapatkan metadata seperti ukuran konten atau tipe konten sebelum mengambil file yang besar dengan GET.