

MODUL 01

Definiton

1. Definisi

RESTful API merupakan implementasi dari REST API yang benar-benar mengikuti semua prinsip REST dengan baik. REST (Representational State Transfer) adalah sebuah arsitektur yang digunakan untuk membangun layanan berbasis web yang sederhana, fleksibel, dan skalabel. RESTful API memungkinkan komunikasi antara client dan server melalui HTTP (Hypertext Transfer Protocol) dengan aturan yang jelas dan terstruktur.

Salah satu prinsip utama dalam RESTful API adalah statelessness, yang berarti bahwa setiap permintaan dari client harus mengandung semua informasi yang dibutuhkan untuk memproses permintaan tersebut. Server tidak menyimpan sesi atau informasi status klien antara permintaan, sehingga setiap interaksi menjadi independen dan lebih mudah diatur skalabilitasnya.

RESTful API juga harus mengikuti standar penggunaan HTTP Method dengan baik dan memberikan HTTP Response yang sesuai dengan permintaan yang dikirim oleh client. Dengan demikian, komunikasi antara client dan server menjadi lebih mudah dipahami dan dikelola.

2. HTTP Method dalam RESTful API

HTTP Method digunakan untuk menentukan aksi yang akan dilakukan terhadap suatu resource dalam sistem. RESTful API harus menggunakan HTTP Method sesuai dengan semantik yang benar agar sistem tetap konsisten dan dapat dipahami dengan baik.

Berikut ini merupakan HTTP Method yang akan dipakai dalam praktikum ini:

HTTP Method	Fungsi dalam RESTful API	Deskripsi
GET	Read (Membaca Data)	Mengambil data dari server. Tidak boleh mengubah data.
POST	Create (Membuat Data Baru)	Menambahkan resource baru ke dalam sistem.
PUT	Update (Memperbarui Data)	Memperbarui resource secara keseluruhan berdasarkan ID.
DELETE	Delete (Menghapus Data)	Menghapus resource berdasarkan ID.

Dengan mengikuti standar ini, RESTful API akan lebih mudah untuk diintegrasikan dengan berbagai sistem dan teknologi.

3. HTTP Response dalam RESTful API

Setiap permintaan yang dikirim ke RESTful API harus mendapatkan respon HTTP yang sesuai dengan standar. Respon ini mencakup status kode HTTP yang menunjukkan apakah permintaan berhasil atau gagal. Status kode HTTP memberikan informasi yang jelas kepada client mengenai hasil dari permintaan yang dikirimkan.

Berikut adalah beberapa HTTP Status Code yang umum digunakan dalam RESTful API:

HTTP Status Code	Kategori	Deskripsi
200 OK	Sukses	Permintaan berhasil diproses.
201 Created	Sukses	Resource baru berhasil dibuat.
400 Bad Request	Kesalahan	Permintaan tidak valid (format salah, data kurang, dll).
404 Not Found	Kesalahan	Resource yang diminta tidak ditemukan.
500 Internal Server Error	Kesalahan	Terdapat kesalahan pada server karena kondisi yang tidak terprediksi

Dengan menerapkan HTTP Response yang sesuai, client dapat memahami status permintaan yang dikirim serta mengambil langkah yang tepat berdasarkan respons yang diterima.

4. Karakteristik RESTful API

Untuk memastikan bahwa sebuah API benar-benar RESTful, ada beberapa karakteristik utama yang harus dipenuhi:

a. Client-Server Architecture

API harus memiliki pemisahan yang jelas antara client (pengguna API) dan server (penyedia API). Hal ini memungkinkan pengembangan yang lebih fleksibel dan skalabel.

b. Stateless

Setiap permintaan dari client harus menyertakan semua informasi yang dibutuhkan tanpa mengandalkan sesi atau state yang tersimpan di server.

c. Cacheable

API harus mendukung caching data untuk meningkatkan efisiensi dan performa.

d. Layered System

RESTful API harus mendukung arsitektur bertingkat, di mana client tidak perlu mengetahui apakah ia berkomunikasi langsung dengan server utama atau melalui perantara seperti proxy atau load balancer.

e. Uniform Interface

API harus memiliki antarmuka yang seragam, termasuk penggunaan endpoint yang konsisten, format data standar (seperti JSON atau XML), serta pemanfaatan HTTP Method dan status kode yang benar.

Dengan memahami dan menerapkan prinsip RESTful API, pengembang dapat membuat layanan web yang lebih terstruktur, mudah dikelola, serta dapat diintegrasikan dengan berbagai sistem dan platform lain.

Referensi:

1. Fielding, R. T. (2000). "Architectural Styles and the Design of Network-based Software Architectures." Dissertation, University of California, Irvine.
2. Richardson, L., & Ruby, S. (2007). "RESTful Web Services." O'Reilly Media.
3. Masse, M. (2011). "REST API Design Rulebook." O'Reilly Media.
4. Subramanian, S. (2017). "Hands-On RESTful Web Services with TypeScript 3." Packt Publishing.
5. Dokumentasi resmi API dari berbagai platform seperti:
 - [REST API - Mozilla Developer Network \(MDN\)](#)
 - [REST API - Microsoft Documentation](#)