



# Hari 4 (Kamis) – REST API untuk Products, Users, Carts

---



## Tujuan Pembelajaran

- Memahami konsep REST API dan resource
  - Mendesain endpoint untuk products, users, carts
  - Mengimplementasikan API endpoint dan response JSON
  - Menerapkan best practice naming dan struktur RESTful
- 



## Materi Inti (2 Jam)

### 1. Konsep REST API dan Resource

- REST (Representational State Transfer) adalah arsitektur untuk komunikasi antara client-server yang menggunakan HTTP.
- Resource adalah objek yang diakses melalui URI (Uniform Resource Identifier), seperti `/api/products`.
- HTTP methods umum: `GET` untuk membaca data, `POST` untuk membuat data baru, `PUT` untuk memperbarui data, dan `DELETE` untuk menghapus data.
- RESTful API harus stateless, artinya setiap permintaan dari klien harus berisi semua informasi yang diperlukan untuk memahami dan memproses permintaan tersebut.

### 2. Mendesain Endpoint untuk Products, Users, Carts

- Struktur URI yang jelas dan konsisten sangat penting untuk RESTful API:
  - `/api/products` untuk mengakses resource produk
  - `/api/users` untuk mengakses resource pengguna
  - `/api/carts` untuk mengakses resource keranjang
- Contoh endpoint dengan Express.js:

```
const express = require('express');
const app = express();

app.get('/api/products', (req, res) => {
  // Logic untuk mengambil semua produk
});

app.post('/api/products', (req, res) => {
  // Logic untuk menambah produk baru
});

app.put('/api/products/:id', (req, res) => {
  // Logic untuk memperbarui produk berdasarkan ID
});
```

```
app.delete('/api/products/:id', (req, res) => {  
  // Logic untuk menghapus produk berdasarkan ID  
});
```

### 3. Studi Kasus: API Endpoint dan Response JSON

- Implementasi CRUD untuk resource products:
  - **GET** `/api/products`: Ambil semua produk dari database dan kirimkan sebagai response JSON.
  - **POST** `/api/products`: Tambah produk baru ke database dan kirimkan status sukses.
  - **PUT** `/api/products/:id`: Update produk di database berdasarkan ID dan kirimkan status sukses.
  - **DELETE** `/api/products/:id`: Hapus produk dari database berdasarkan ID dan kirimkan status sukses.
- Format response JSON harus konsisten dan informatif:

```
{  
  "status": "success",  
  "data": [...],  
  "message": "Operation completed successfully"  
}
```

### 4. Best Practice Naming dan Struktur RESTful

- Gunakan kata benda untuk URI, bukan kata kerja, misalnya `/api/products` bukan `/api/getProducts`.
- Konsisten dalam penggunaan HTTP methods sesuai dengan operasi yang dilakukan.
- Dokumentasikan setiap endpoint dengan jelas, termasuk method, path, dan contoh request/response.



### Praktik Mandiri (8 Jam)

1. Implementasi endpoint CRUD untuk users dan carts
2. Uji endpoint dengan Postman/Insomnia
3. Tambahkan validasi sederhana pada input data
4. Dokumentasikan seluruh endpoint beserta contoh request/response



### Tips untuk Pemula

- Gunakan tool seperti Postman untuk testing API
- Pastikan setiap endpoint memiliki response yang konsisten
- Validasi input data untuk mencegah error
- Dokumentasikan setiap perubahan pada API



### Referensi

- [REST API Tutorial](#)
- [Express.js Routing](#)
- [Postman Docs](#)