

# Laporan Praktikum WSE #5

Mata Kuliah : Web Service Engineering  
Dosen Pengampu : Muhayat, M.IT  
Praktikum : P5 – Membangun RESTful CRUD API dengan Express  
Nama Mahasiswa : M. RIZAL KURNIAWAN  
NIM : 230104040211  
Kelas : TI23A  
Tanggal Praktikum : 03-11-2025

## A. Tujuan Praktikum

1. Mampu membangun RESTful API sederhana menggunakan Express.js, dengan memahami konsep dasar request dan response dalam arsitektur web service. Mahasiswa dapat membuat server berbasis Node.js yang mampu menangani berbagai rute dan endpoint sesuai standar REST, serta menampilkan data dalam format JSON agar mudah diolah oleh client.
2. Menerapkan operasi dasar CRUD (Create, Read, Update, Delete) pada resource tertentu (misalnya products) secara efektif. Mahasiswa mampu menulis kode yang dapat menambahkan data baru, menampilkan daftar atau detail data, memperbarui data yang sudah ada, serta menghapus data tertentu melalui API menggunakan metode HTTP yang tepat.
3. Menggunakan HTTP method dan status code secara benar, seperti GET untuk pengambilan data, POST untuk penambahan data, PUT untuk pembaruan, dan DELETE untuk penghapusan. Selain itu, mahasiswa memahami makna dan penerapan status code seperti 200 OK, 201 Created, 400 Bad Request, 404 Not Found, serta 500 Internal Server Error dalam konteks respons API yang konsisten dan informatif.
4. Mengetahui struktur dasar server API modular dengan Node.js, termasuk pembagian folder dan file yang rapi seperti routes, controllers, dan data agar proyek lebih mudah dikembangkan dan dipelihara. Mahasiswa juga memahami pentingnya modularisasi kode, penggunaan middleware Express, serta pengelolaan dependensi menggunakan npm.
- 5.

## B. Tools

Node.js 18+ & npm  
Express.js  
VS Code  
Nodemon (dev dependency)  
Postman / Thunder Client  
Git & GitHub (opsional)

## C. Arsitektur Singkat

- Client (Postman / Thunder Client) → mengirim HTTP request.  
API Server (Express) → menerima request dan menentukan route handler.  
Router (products.routes.js) → mendefinisikan endpoint CRUD.  
Data (products.data.js) → menyimpan data produk sementara (array).  
Response JSON → dikembalikan ke client dengan status code yang sesuai.

## D. Langkah Implementasi (ringkas)

1. Membuat folder project bernama praktikum5\_restful\_crud/
2. Inisialisasi project menggunakan npm init -y
3. Menginstal Express dan Nodemon (npm install express & npm install nodemon -save-dev)
4. Membuat struktur folder src/routes/, src/data/, src/app.js
5. Menulis kode CRUD (GET, POST, PUT, DELETE) di products.routes.js
6. Menghubungkan route ke app.js menggunakan app.use('/api/products', productRoutes)
7. Menjalankan server dengan npm run dev dan menguji endpoint di Postman

## E. Hasil & Bukti

Lampirkan **Screenshot Hasil Uji Endpoint** di Postman

- GET ALL→ Menampilkan seluruh produk (200 OK)

The screenshot shows the Postman interface with a collection named 'Rizal Kurniawan's Workspace'. A specific test case titled 'P5-CRUD-REST-230104040211 M RIZAL ...' is selected. The 'GET ALL' test is highlighted. The request URL is set to 'http://localhost:3000/products'. The response status is '200 OK' with a response time of 11 ms and a size of 524 B. The response body is displayed as JSON:

```
1 [
2   {
3     "status": "success",
4     "message": "Daftar produk berhasil diambil",
5     "data": [
6       {
7         "id": 1,
8         "name": "Leptop Gaming",
9         "price": 1500000
10      },
11      {
12        "id": 2,
13        "name": "Mouse",
14        "price": 150000
15      },
16      {
17        "id": 3,
18        "name": "Keyboard",
19        "price": 300000
20      },
21      {
22        "id": 4,
23        "name": "Add your name in the body"
24      }
25    ]
26  }
```

## POST CREATE → Menambah produk baru (201 Created)

The screenshot shows the Postman interface with a collection named "P5-CRUD-REST-230104040211 M RIZAL KURNIAWAN". A POST request is selected for the "products" endpoint at `http://localhost:3000/products`. The "Body" tab contains a JSON payload:

```
1 {
2     "status": "success",
3     "message": "Produk baru berhasil ditambahkan",
4     "data": {
5         "id": 8,
6         "name": "Headset",
7         "price": 250000
8     }
9 }
```

The response status is 201 Created, and the response body is:

```
1 {
2     "status": "success",
3     "message": "Produk berhasil ditambahkan",
4     "data": {
5         "id": 8,
6         "name": "Headset",
7         "price": 250000
8     }
9 }
```

## PUT UPDATE → Memperbarui data produk (200 OK / 404 Not Found)

The screenshot shows the Postman interface with the same collection. A PUT request is selected for the "products/1" endpoint at `http://localhost:3000/products/1`. The "Body" tab contains a JSON payload:

```
1 {
2     "name": "Laptop Gaming",
3     "price": 15000000
4 }
5 
```

The response status is 200 OK, and the response body is:

```
1 {
2     "status": "success",
3     "message": "Produk berhasil diperbarui",
4     "data": {
5         "id": 1,
6         "name": "Laptop Gaming",
7         "price": 15000000
8     }
9 }
```

## DELETE → Menghapus produk (200 OK / 404 Not Found)

The screenshot shows the Postman interface with the following details:

- Collection:** P5-CRUD-REST-230104040211 M RIZAL KURNIAWAN
- Request Type:** DELETE
- URL:** <https://localhost:3000/products/2>
- Pre-request Script:**

```
pm.test('Successful DELETE request', function () {  
    pm.expect(pm.response.code).to.be.oneOf([200, 202, 204]);  
});
```
- Post-response Script:**

```
{  
  "status": "success",  
  "message": "Produk berhasil dihapus",  
  "data": [  
    {"id": 2,  
     "name": "Mouse",  
     "price": 150000  
  }]  
}
```
- Status:** 200 OK
- Headers:** Content-Type: application/json
- Body:** JSON (Preview shows the response object)

## F. Analisis

API yang telah dibuat menunjukkan bahwa fungsionalitas CRUD (Create, Read, Update, Delete) berjalan dengan baik sesuai konsep RESTful. Setiap endpoint mampu merespons permintaan klien menggunakan metode HTTP yang tepat, seperti GET, POST, PUT, dan DELETE, serta memberikan status code yang sesuai seperti 200 OK atau 201 Created. Struktur data yang dikirim dan diterima menggunakan format JSON juga sudah konsisten dengan standar API modern.

Meskipun demikian, sistem masih bersifat dasar karena belum dilengkapi dengan validasi input yang ketat dan mekanisme penanganan error yang komprehensif. Selain itu, data masih disimpan secara statis di dalam array dan belum menggunakan basis data eksternal. Hal-hal tersebut direncanakan akan dikembangkan lebih lanjut pada Praktikum 6, yang berfokus pada penerapan RESTful API Best Practices dan peningkatan keamanan serta efisiensi pengelolaan data.

## **G. Kesimpulan**

Dari hasil praktikum, dapat disimpulkan bahwa mahasiswa telah berhasil membangun sebuah RESTful CRUD API sederhana menggunakan Express.js. Implementasi yang dibuat mampu menjalankan operasi dasar seperti menambah, menampilkan, memperbarui, dan menghapus data melalui endpoint yang dirancang sesuai standar REST. Hasil pengujian menggunakan Postman menunjukkan bahwa setiap operasi menghasilkan respons dan status code yang benar.

Melalui praktikum ini, mahasiswa memperoleh pemahaman lebih mendalam mengenai cara kerja server berbasis Node.js, konsep HTTP request-response, serta struktur dasar pengembangan web service yang modular. Praktikum ini menjadi dasar penting untuk tahap selanjutnya, yaitu pengembangan API yang lebih kompleks, terintegrasi dengan database, dan menerapkan prinsip best practices dalam pengembangan RESTful API.

## **H. Checklist Praktikum**

- Struktur project Express sudah sesuai
- CRUD endpoint berjalan dengan benar
- Status code sesuai dengan operasi
- Response JSON terstruktur rapi
- Pengujian berhasil di Postman