

Slide 8: Hands-on: Membuat API CRUD untuk User

- **Praktik Langsung:**
 - Membuat project baru.
 - Implementasi CRUD API (Create, Read, Update, Delete) untuk user.
 - Langkah-langkah:
 1. Inisialisasi project.
 2. Buat model user.
 3. Buat route dan controller untuk setiap operasi CRUD.

Hands-on: Membuat API CRUD untuk User

Langkah 1: Inisialisasi Project

Inisialisasi project dengan npm:

Contoh kode:

```
npm init -y
```

Langkah 2: Instalasi Express

Instal Express dan body-parser:

Contoh kode:

```
npm install express body-parser
```

Langkah 3: Buat Server

Buat file `server.js` dan tambahkan kode berikut:

Contoh kode:

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const PORT = 3000;

app.use(bodyParser.json());

app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

Langkah 4: Buat Model User

Buat array sederhana untuk menyimpan data user:

javascript

Contoh kode:

```
let users = [];
```

Langkah 5: Implementasi Endpoint CRUD

GET: Mendapatkan Daftar User

Contoh kode:

```
app.get('/users', (req, res) => {  
  res.json(users);  
});
```

POST: Menambahkan User Baru

Contoh kode:

```
app.post('/users', (req, res) => {  
  const user = req.body;  
  users.push(user);  
  res.status(201).json(user);  
});
```

PUT: Memperbarui User

Contoh kode:

```
app.put('/users/:id', (req, res) => {  
  const { id } = req.params;  
  const user = req.body;  
  users = users.map(u => (u.id === id ? user : u));  
  res.json(user);  
});
```

DELETE: Menghapus User

Contoh kode:

```
app.delete('/users/:id', (req, res) => {  
  const { id } = req.params;  
  users = users.filter(u => u.id !== id);  
  res.status(204).end();  
});
```

