

**Laporan Praktikum**  
**Mata Kuliah Pemograman Web**



**Pertemuan 6**  
**"CRUD"**

Dosen Pengampu :  
Willdan Aprizal Arifin, S.Pd., M.Kom.

Disusun Oleh :  
Anggi Julian Agustin  
2301755

**PROGRAM STUDI SISTEM INFORMASI KELAUTAN**  
**UNIVERSITAS PENDIDIKAN INDONESIA**  
**2024**

## I. PENDAHULUAN

CRUD adalah singkatan dari Create, Read, Update dan Delete yang merupakan empat operasi utama dalam pengolahan data pada sistem database. Konsep ini sangat penting untuk pengembangan aplikasi, baik online, desktop, atau seluler, karena memungkinkan pengguna mengelola data secara terstruktur dan efisien. Dalam pengembangan perangkat lunak, implementasi CRUD menjadi dasar interaksi pengguna dengan database dan membantu dalam pemrosesan data secara dinamis.

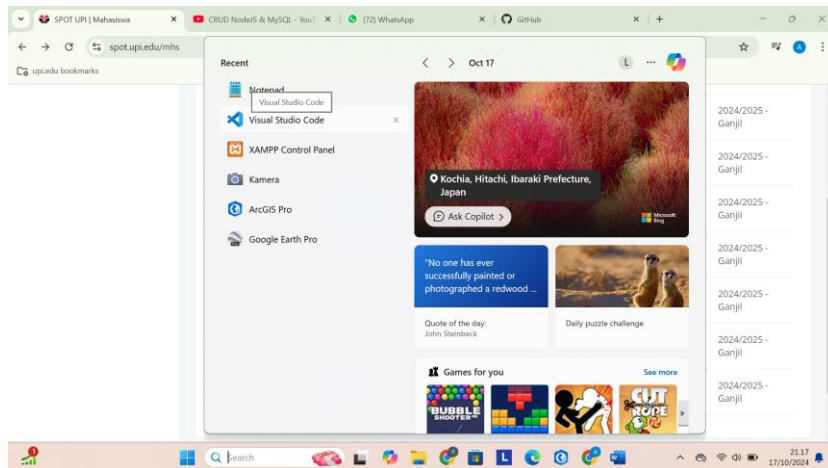
Penggunaan CRUD tidak hanya terbatas pada pengembangan aplikasi, tetapi juga mencakup berbagai aspek bisnis dan manajemen data. Dengan memahami dan mengimplementasikan operasi CRUD, pengembang dapat menciptakan aplikasi yang lebih user-friendly dan responsif terhadap kebutuhan pengguna. Dalam pendahuluan ini, kita akan membahas lebih lanjut mengenai masing-masing komponen dari CRUD, serta pentingnya implementasi yang tepat dalam pengembangan sistem informasi modern.

## II. ALAT DAN BAHAN

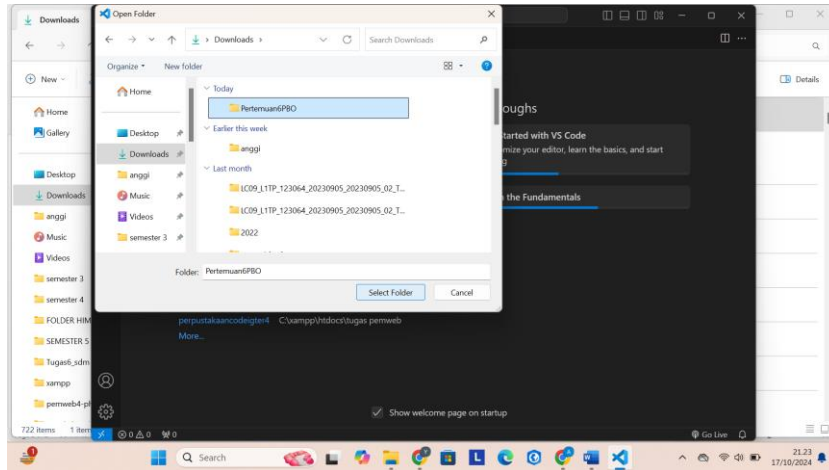
1. Visual Studio Code
2. Node JS
3. XAMPP
4. MySQL

## III. LANGKAH KERJA

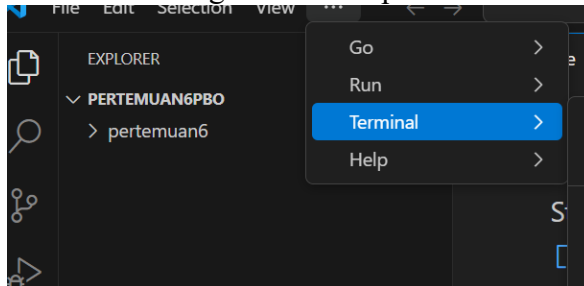
1. .



2. Kemudian bikin file baru >> select folder.



3. Lalu klik titik tiga di atas >> pilih terminal.

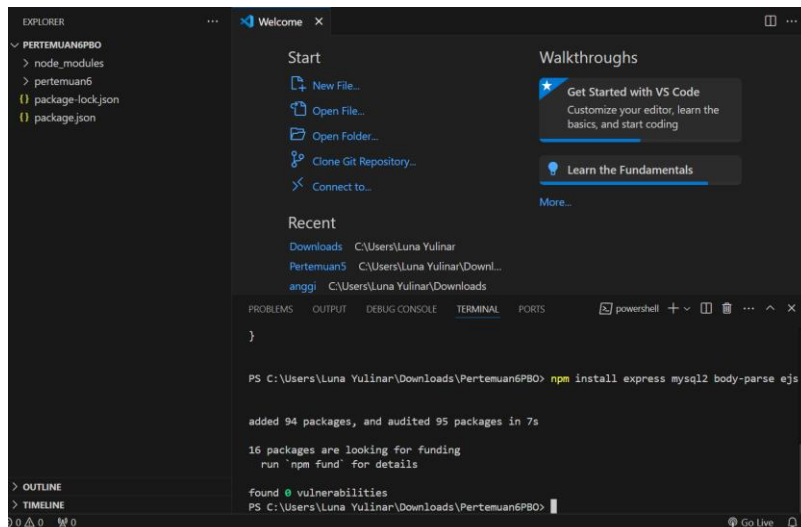


4. Fungsi nya untuk memastikan feked dari Js akan dikemas untuk bisa dibuat >> maka akan muncul packade.jcon.

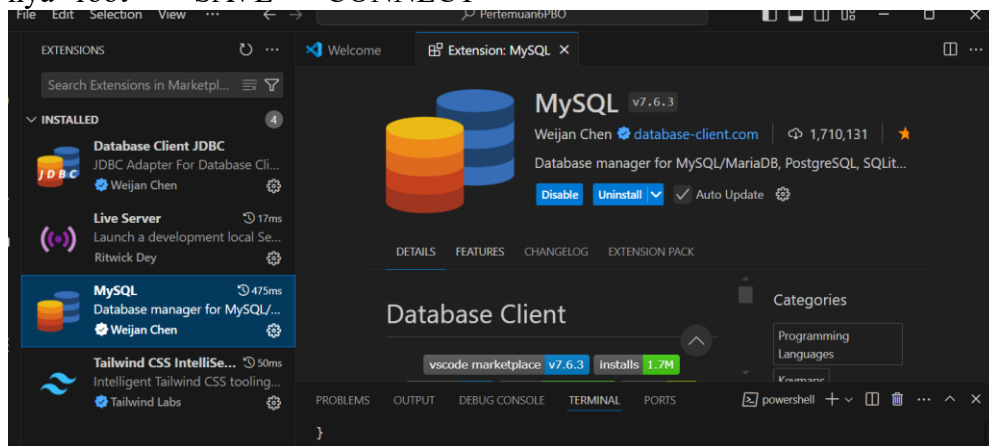
```
PS C:\Users\Luna Yulinar\Downloads\Pertemuan6PBO> npm init -y
Wrote to C:\Users\Luna Yulinar\Downloads\Pertemuan6PBO\package.json:

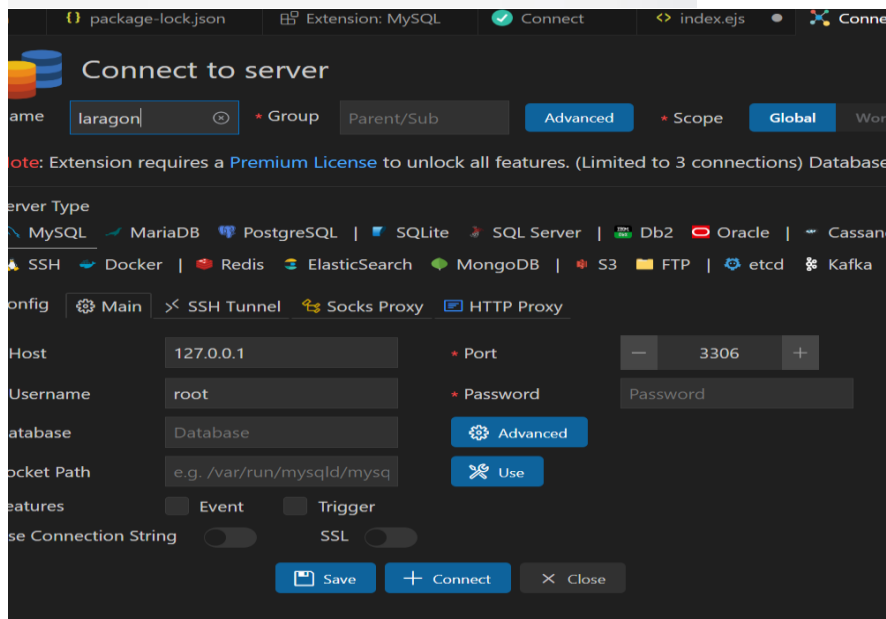
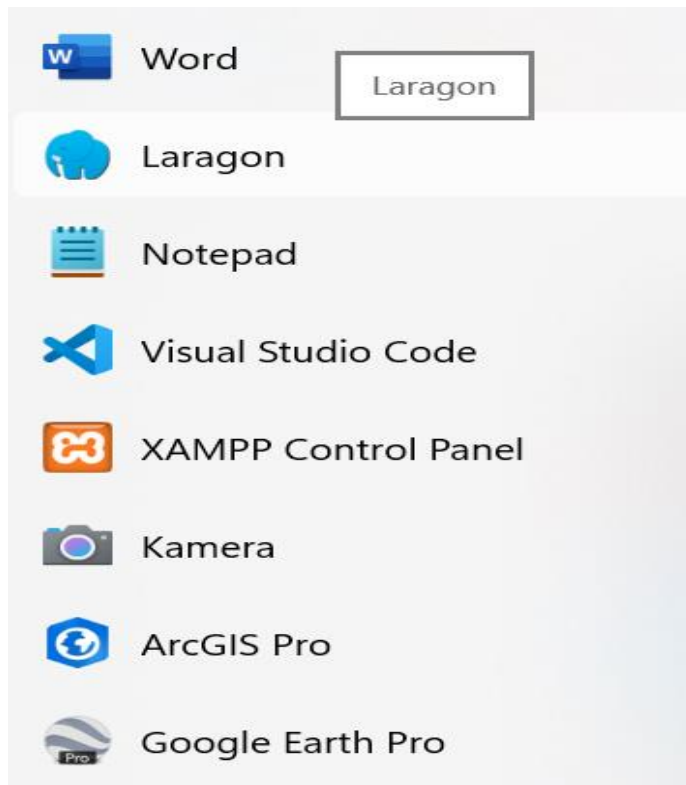
{
  "name": "pertemuan6pbo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
```

5. Kemudian kita perlu instal kebutuhan kita >> npm >> install >> express >> mysql2 >> body-parser >> terus tambah ejs >> dengan cara di spasi >> enter >> akan muncul yang sudah di instal >> ada 16 packages yang telah ditemukan >> di samping kiri akan ada " mode\_modules" dan " package-lock.json.

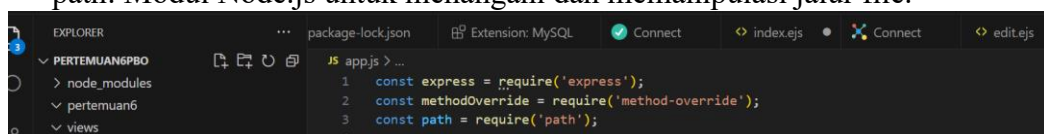


6. Kemudian instal MySQL "Weijan Chen" >> klik data base >> bikin koneksi baru "add connection" >> isi name nya bebas "XAMPP" >> untuk mengisi port nya jangan aktifkan laragon nya "3306" >> host nya "127.0.0.1" >> username nya "root" >> SAVE >> CONNECT





7. Buat di app.js nya >> express: Framework yang digunakan untuk membangun aplikasi web >> method-override: Middleware yang memungkinkan penggunaan metode HTTP seperti PUT dan DELETE melalui formulir HTML >> path: Modul Node.js untuk menangani dan memanipulasi jalur file.



8. Kemudian tambah kan `const app = express();` untuk **app**: Membuat instance dari aplikasi Express >> dan `PORT = process.env.PORT` untuk Menentukan port yang digunakan untuk menjalankan server. Jika tidak ada variabel lingkungan PORT, maka akan default ke 3000.

```
const path = require( 'path' );

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
```

9. Kemudian juga `express.urlencoded`: Middleware untuk menguraikan data yang dikirim melalui formulir. `extended: true` memungkinkan untuk menguraikan data yang lebih kompleks >> `methodOverride`: Mengizinkan penggunaan metode HTTP yang tidak didukung secara langsung oleh HTML (seperti PUT dan DELETE) >> `set('view engine', 'ejs')`: Mengatur EJS sebagai mesin templating untuk merender tampilan >> `set('views', path.join(__dirname, 'views'))`: Menentukan folder views sebagai tempat penyimpanan file tampilan EJS.

```
// Middleware
app.use(express.urlencoded({ extended: true }));
app.use(methodOverride('_method'));
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));
```

10. Dan tambahkan `laptops` untuk Array untuk menyimpan data laptop. Dalam aplikasi nyata, data ini sebaiknya disimpan di database.

```
// Data Dummy (seharusnya menggunakan database)
let laptops = [
  { id: 0, brand: 'Asus', model: 'ZenBook', price: 15000 },
  { id: 1, brand: 'Acer', model: 'Predator', price: 20000 },
];
```

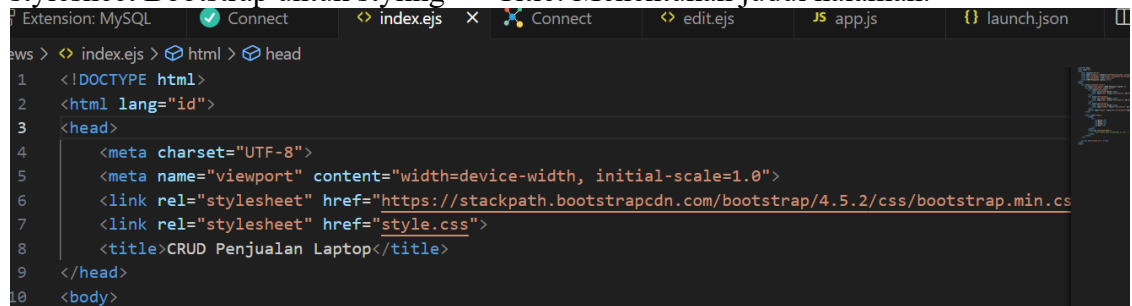
11. Lalu tambahkan `put`. Rutenya untuk memperbarui laptop >> dan di kunci dengan `app.listen` untuk menjalankan server.

```
// Rute untuk memperbarui laptop
app.put('/laptops/:id', (req, res) => {
  const { brand, model, price } = req.body;
  laptops[req.params.id] = { id: req.params.id, brand, model, price };
  res.redirect('/laptops');
});

// Rute untuk menghapus laptop
app.delete('/laptops/:id', (req, res) => {
  laptops.splice(req.params.id, 1);
  res.redirect('/laptops');
});

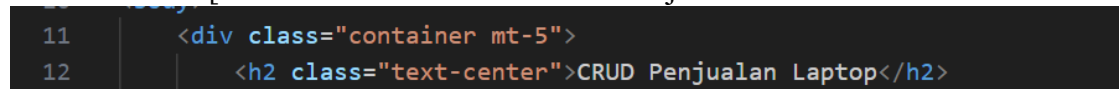
// Jalankan server
app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});
```

12. Pertama tama untuk membuat index html >> klik tambah file >> kasih nama index.html >> klik html : 5 >> Meta Tags: Mengatur karakter set dan viewport untuk responsivitas di perangkat seluler >> Link CSS: Menghubungkan ke stylesheet Bootstrap untuk styling >> Title: Menentukan judul halaman.



```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
7   <link rel="stylesheet" href="style.css">
8   <title>CRUD Penjualan Laptop</title>
9 </head>
10 <body>
```

13. Kemudian untuk membuat container di awali dengan “ Di Kontainer” >> tambahkan <h2[ class=”text” untuk menambahkan judul.



```
11 <div class="container mt-5">
12   <h2 class="text-center">CRUD Penjualan Laptop</h2>
```

#### IV. KESIMPULAN

Setelah melakukan praktikum ini sara sedikit mengerti tentang CRUD adalah konsep inti untuk mengelola data dalam berbagai aplikasi dan platform. Semua aplikasi yang bekerja dengan data (seperti web, seluler, atau desktop) harus menyertakan kemampuan CRUD agar dapat berinteraksi secara efektif dengan pengguna dan data. CRUD memastikan pengelolaan data yang mudah dan memungkinkan pengembangan dan pembaruan sistem yang dinamis..