

LAPORAN
MENAMPILKAN DATA DENGAN CHART JS DAN KONEKSI KE
DATABASE MENGGUNAKAN JAVASCRIPT



Mata Kuliah : Pemrograman Web F

Disusun Oleh : Kelompok 2

Daftar Anggota Kelompok :

Rayhan Rizal M	20081010045
Farkhan	20081010060
Faris Syaifullah	20081010078
M. Randy Oktavianus	20081010153

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UPN "VETERAN" JAWA TIMUR
SURABAYA

2022

Koneksi ke Basi Data Menggunakan JavaScript

Pada *project* ini, basis data yang digunakan ialah basis data MySQL. Untuk terhubung ke basis data MySQL diperlukan pemanggilan modul MySQL dengan cara sebagai berikut.

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

Modul MySQL yang dipanggil dimasukkan ke dalam variabel dengan nama `mysql`. Hal ini dilakukan untuk memudahkan proses selanjutnya. Modul berikutnya yang diperlukan dalam *project* ini ialah modul File System. Untuk dapat menggunakannya diperlukan instalasi modul terlebih dulu dengan cara mengetikkan kode berikut pada terminal.

```
farhan@LAPTOP-C065FARJ MINGW64 /d/UPN/Semester/semester4/PEMWEB/Tugas/Tugas4/Kelompok/web (main)
$ npm install fs
```

Modul file system digunakan untuk menampung data dari database dan memasukkannya ke dalam file json. Setelah modulnya terpasang, panggil modul tersebut dan masukkan ke dalam variabel dengan nama `fs`, dapat dilihat kodenya di bawah ini.

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

Untuk menghubungkan ke basis data MySQL dapat menggunakan method `createConnection()` dari modul `mysql`. Agar lebih jelas, dapat dilihat pada kode di bawah ini.

```
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'pemweb_4'
});
```

Penggunaan method `createConnection()` dimasukkan ke dalam variabel `connection` agar memudahkan proses berikutnya. 'localhost' merupakan nama host dari basis data yang digunakan. User dan password diisi dengan user dan password pada basis data yang digunakan. Database diisi dengan nama basis data yang digunakan, dalam hal ini bernama `pemweb_4`. Selanjutnya diperlukan pengecekan apakah basis data sudah terhubung atau belum. Untuk mengeceknya digunakan kode di bawah ini.

```
connection.connect( (err) => {
  if (err) throw err;
  console.log('Connected!');
});
```

Jika mengalami kegagalan ketika menghubungkan, maka akan ditampilkan error dan jika berhasil menghubungkan, maka akan menampilkan pesan Connected! seperti di bawah ini.

```
PS D:\UPN\Semester\semester4\PEMWEB\Tugas\Tugas4\Kelompok\web> node .\mysql.js  
Connected!
```

Jika sudah terhubung, maka selanjutnya akan dilakukan pengambilan data dari basis data dan memasukkannya ke dalam file .json dengan format JSON. Untuk melakukan pengambilan data, digunakan method query() dengan parameter query sql dan luarannya. Luaran yang diberikan berupa pesan error atau hasil datanya. Untuk lebih jelasnya ialah seperti kode berikut.

```
connection.query('SELECT * from mahasiswa', (err, rows) => {  
  if(err) throw err  
  const finished = (error) => {  
    if(error) {  
      console.error(error);  
      return;  
    }  
  }  
  const jsonData = JSON.stringify(rows, null, 2);  
  fs.writeFile('mahasiswa.json', jsonData, finished);  
})
```

Query yang digunakan ialah SELECT * from mahasiswa, yaitu ambil semua data yang ada pada tabel mahasiswa. Jika terjadi error, maka tampilkan errornya. Lalu buat variabel dengan nama finished untuk menampilkan error juga, variabel ini digunakan untuk parameter pada method writeFile(). Method tersebut berasal dari modul file system yang berfungsi untuk menulis data pada file .json. Selanjutnya buat variabel jsonData yang berisi JSON.stringify(). Hal itu berfungsi untuk mengubah data dari basis data ke dalam format JSON. Selanjutnya gunakan method writeFile() dengan tiga parameter. Parameter pertama ialah nama file, parameter kedua ialah data yang dimasukkan, dan parameter ketiga ialah pesan yang ditampilkan jika mengalami error.

Selanjutnya gunakan kode di bawah ini untuk menghentikan proses koneksi ke basis datanya.

```
connection.end();
```

Semua kode di atas ditulis dalam file yang bernama mysql.js. Sehingga secara keseluruhan isi filenya ialah sebagai berikut.

```
const mysql = require('mysql');  
const fs = require('fs');
```

```

const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'pemweb_4'
});

connection.connect( (err) => {
  if (err) throw err;
  console.log('Connected!');
});

connection.query('SELECT * from mahasiswa', (err, rows) => {
  if(err) throw err
  const finished = (error) => {
    if(error) {
      console.error(error);
      return;
    }
  }

  const jsonData = JSON.stringify(rows, null, 2);
  fs.writeFile('mahasiswa.json', jsonData, finished);
})

connection.end();

```

Untuk menjalankan file tersebut dapat dilakukan melalui terminal dengan mengetikkan node .\mysql.js

```

PS D:\UPN\Semester\semester4\PEMWEB\Tugas\Tugas4\Kelompok\web> node .\mysql.js
Connected!

```

Setelah filenya berhasil dijalankan, maka akan muncul file dengan nama mahasiswa.json. Isi dari file mahasiswa.json ialah sebagai berikut.

```

[
  {
    "npm": 20081010045,
    "nama": "Rayhan Rizal Mahendra",

```

```
[{"email": "rayhan@gmail.com",  
  "prodi": "Informatika",  
  "sks": 24  
},  
{  
  "npm": 20081010060,  
  "nama": "Farkhan",  
  "email": "farkhanjayadi@gmail.com",  
  "prodi": "Informatika",  
  "sks": 48  
}]
```

Isi dari file tersebut tentu sesuai dengan data yang terdapat dalam basis data.