

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



Mata Kuliah : Pemrograman Web F

Disusun Oleh : Kelompok 2

Daftar Anggota Kelompok :

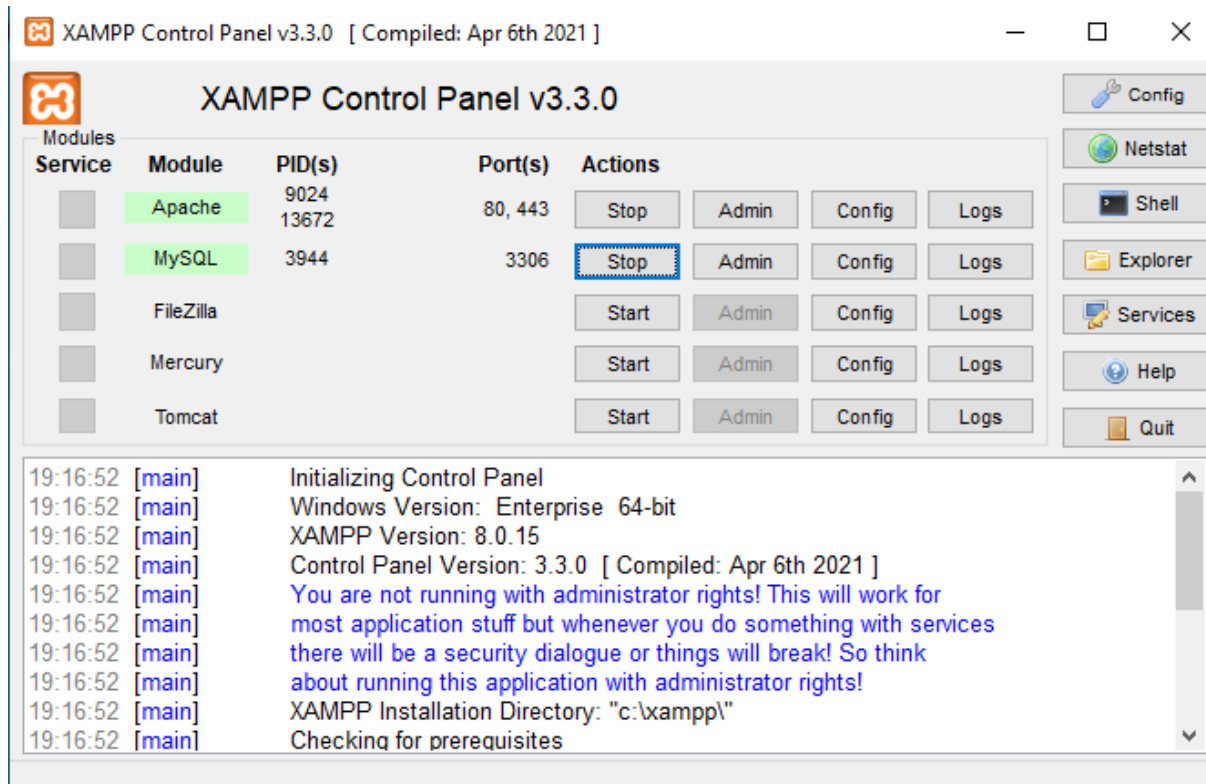
Rayhan Rizal M	20081010045
Farkhan	20081010060
Faris Syaifullah	20081010078
M. Randy Oktavianus	20081010153

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

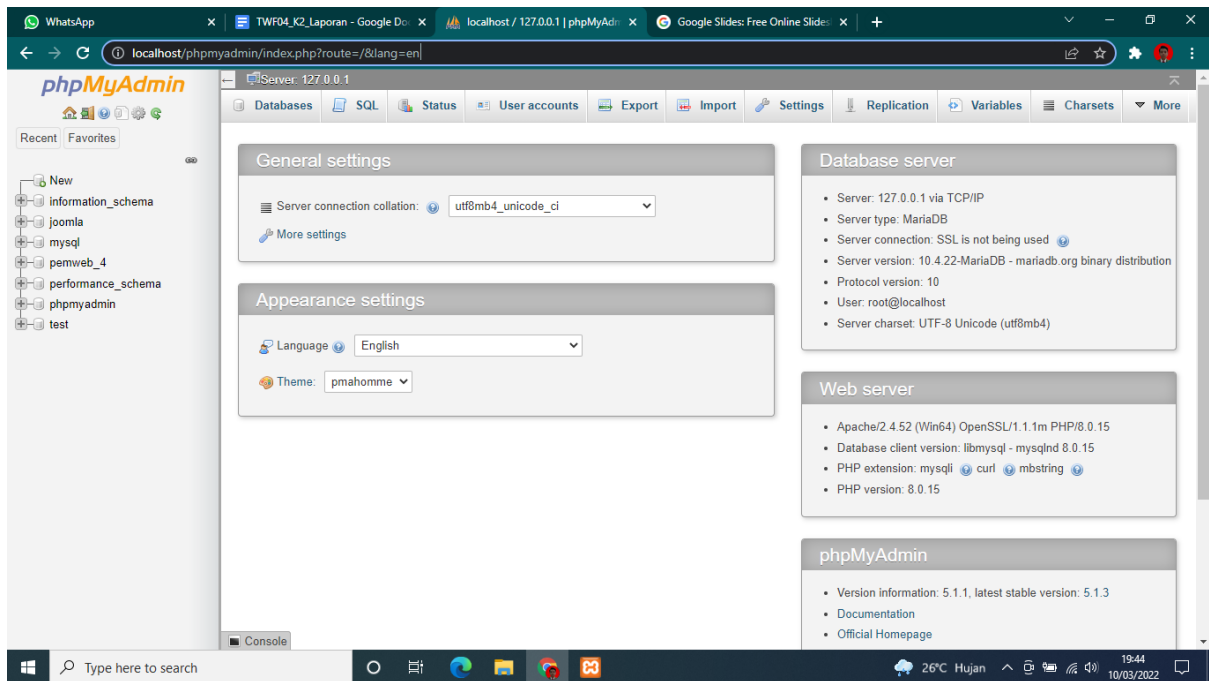
2022

Membuat Data Base Menggunakan MySQL

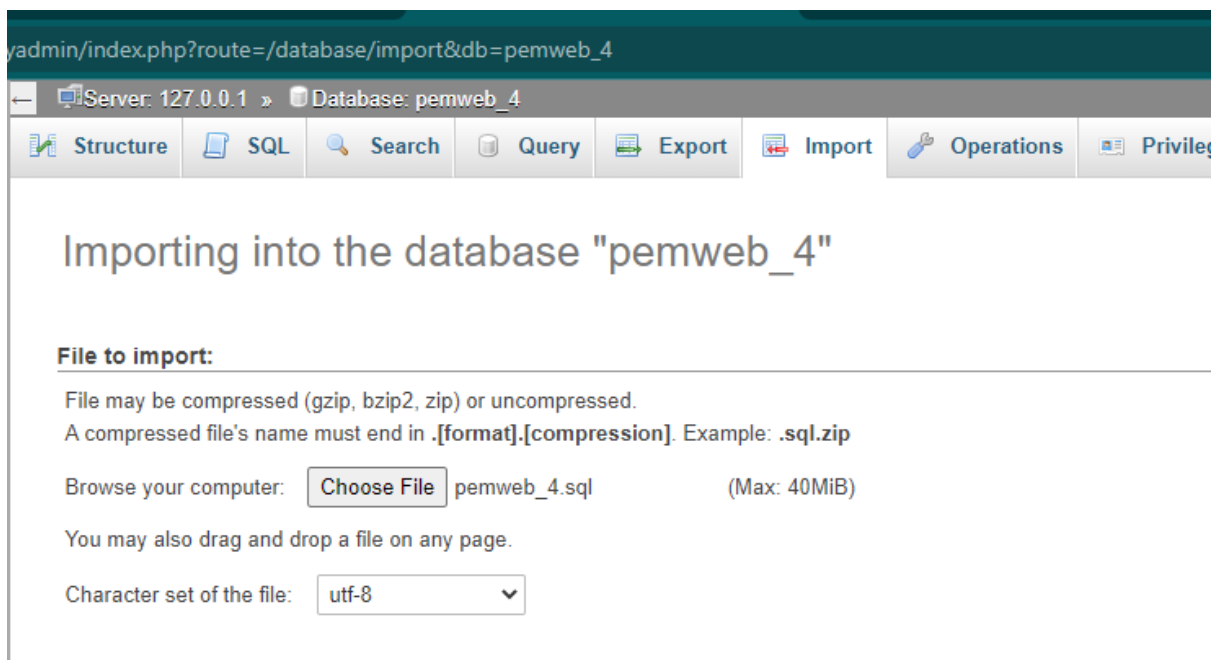
Untuk membuat database di MySQL pertama, yang dilakukan adalah membuka terlebih dahulu XAMPP control panel. setelah membuka XAMPP control panel, pada menu module apache dan MySQL, menu tersebut harus distart/diaktifkan terlebih dahulu agar bisa mengakses localhost/phpmyadmin. Untuk gambar bisa dilihat dibawah ini:



Setelah diaktifkan, kemudian langkah selanjutnya buka browser kemudian ketikkan <http://localhost/phpmyadmin>. setelah itu akan muncul tampilan seperti ini :



Setelah berhasil masuk ke localhost nya kita membuat data baru dengan cara klik new kemudian buat baru dengan nama pemweb_4. lalu import data file yang telah dibuat dengan cara klik import pada menu diatas lalu akan tampil gambar seperti berikut :



setelah masuk ke menu import kemudian pilih file yang telah dibuat source code nya bernama pemweb_4.sql. lalu scroll kebawah dan pilih go agar file yang diimport berhasil ditambahkan. Jika selesai ditambahkan maka akan muncul sebuah database baru yang bernama mahasiswa yang nantinya akan dibuat chart kedalam web browser. berikut ini adalah tampilan jika telah berhasil meng-import database mahasiswa.

SELECT * FROM `mahasiswa`

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	npm	nama	email	prodi	sks
<input type="checkbox"/> Edit Copy Delete	20081010045	Rayhan Rizal Mahendra	rayhan@gmail.com	Informatika	24
<input type="checkbox"/> Edit Copy Delete	20081010060	Farkhan	farkhanjayadi@gmail.com	Informatika	48

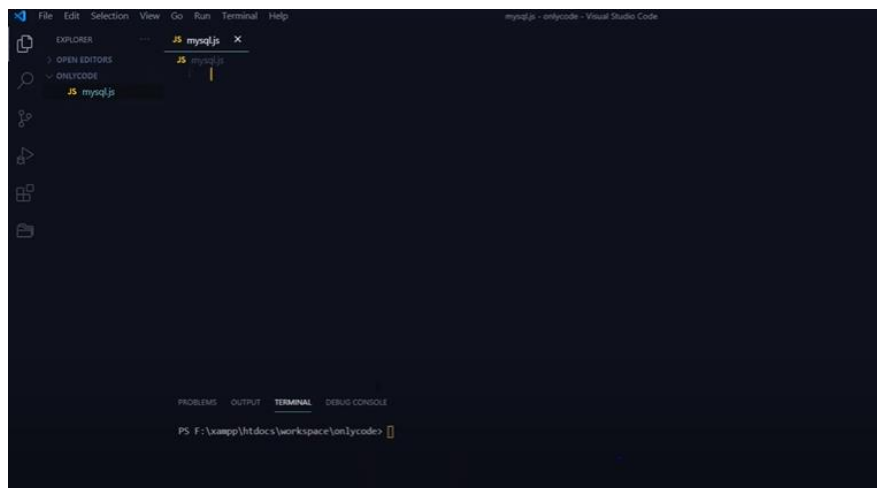
☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

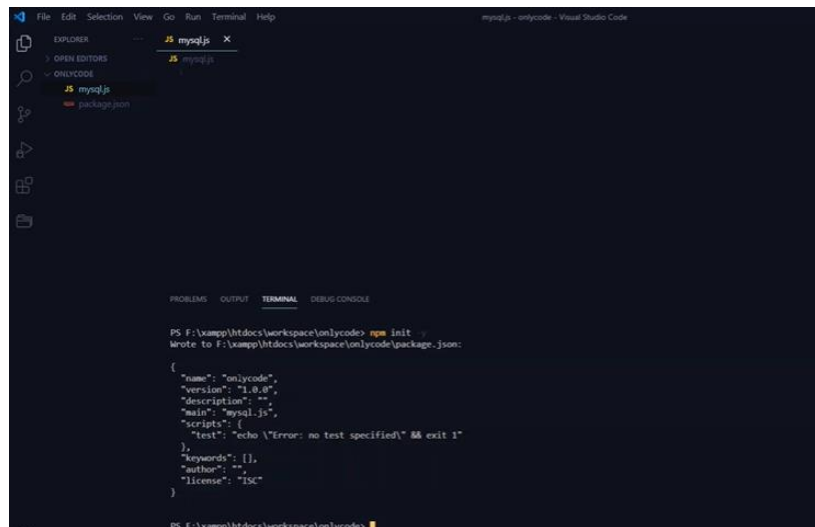
contoh diatas adalah database mahasiswa yang terdiri dari nama mahasiswa, NPM mahasiswa, email, prodi dan sks mahasiswa. database tersebut akan ditampilkan kedalam web browser berupa diagram batang dan lingkaran.

Instalasi Package

untuk proses instalasi package dimulai dari membuat file baru, seperti gambar dibawah ini bisa ditambahkan dengan nama mysql.js



Selanjutnya, masuk ke konsol dan pada terminal menambahkan npm init menambahkan -y, maka akan muncul general seperti gambar berikut.



```
File Edit Selection View Go Run Terminal Help
mysql.js - onlycode - Visual Studio Code

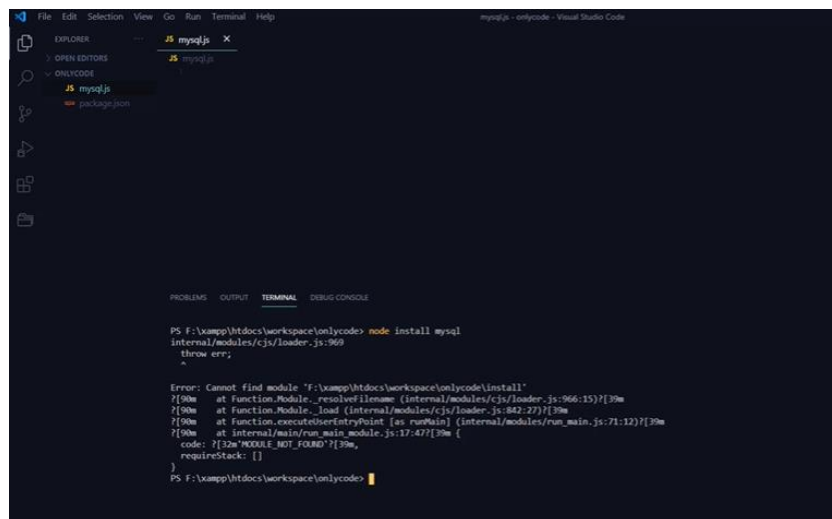
EXPLORER
  OPEN EDITORS
  ONLYCODE
    mysql.js
    package.json

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS F:\xampp\htdocs\workspace\onlycode> npm init
wrote to F:\xampp\htdocs\workspace\onlycode\package.json:

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

Pada gambar diatas, deskripsi belum dicantumkan dan masih hanya sebuah skrip. selanjutnya menginstal mysql dan perpustakaan, dan menambahkan node install mysql seperti gambar berikut.



```
File Edit Selection View Go Run Terminal Help
mysql.js - onlycode - Visual Studio Code

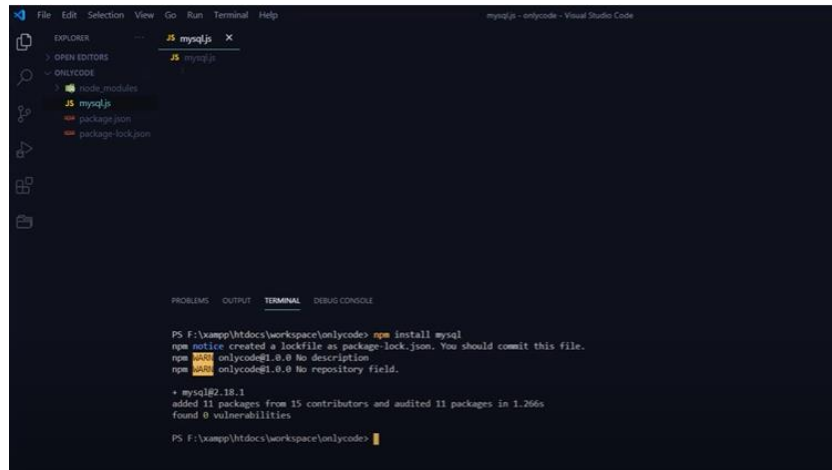
EXPLORER
  OPEN EDITORS
  ONLYCODE
    mysql.js
    package.json

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS F:\xampp\htdocs\workspace\onlycode> node install mysql
internal/modules/cjs/loader.js:969
  throw err;
  ^

Error: Cannot find module 'F:\xampp\htdocs\workspace\onlycode\install'
?{90m at Function.Module._resolveFilename (internal/modules/cjs/loader.js:966:15)?{39m
?{90m at Function.Module._load (internal/modules/cjs/loader.js:842:27)?{39m
?{90m at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:71:12)?{39m
?{90m at internal/main/run_main_module.js:17:47}?{39m {
  code: ?{32m"MODULE_NOT_FOUND"?{39m,
  requireStack: []
}
PS F:\xampp\htdocs\workspace\onlycode>
```

selanjutnya kembali ke konsol awal dengan menambahkan cls. Selanjutnya menginstall mysql dengan menambahkan npm install mysql seperti gambar berikut.



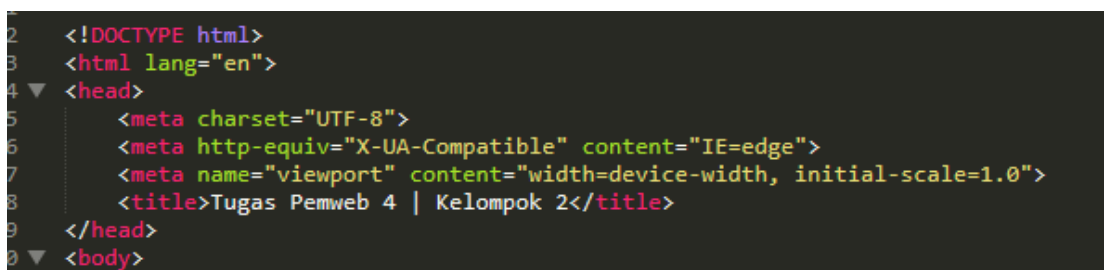
pada gambar diatas, proses instalasi mysql terjadi lambat karena hanya memiliki 1 bookstore saja yaitu node_modulus, selesai.

Penjelasan HTML

untuk langkah pertama dari Source Code HTML yaitu deklarasi variabel dengan tag `<!DOCTYPE html>`,



selanjutnya, menambahkan tag `<head>` untuk memasukkan sebuah data, didalam tag head menambahkan tag `<meta charset>` untuk memberi tahu karakter set apa yang akan ditampilkan, selanjutnya menambahkan tag `<meta http-equiv>` untuk header HTTP, dan menambahkan tag `<tittle>` untuk memberi judul pada web.



selanjutnya, memberi styling pada diagram batang dan diagram lingkaran

```

9     </head>
10    <body>
11
12        <div style="height: 500px; width: 500px;">
13            <canvas id="myChart"></canvas>
14        </div>
15
16        <div style="height: 500px; width: 500px;">
17            <canvas id="barChart"></canvas>
18        </div>
19
20    </body>
21    </html>

```

selanjutnya adalah menambahkan tag `<script>` untuk memanggil file chart js, jquery dan my script dengan menambahkan src=

```

20    <!-- Chart JS -->
21    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.7.1/chart.min.js" integrity="
sha512-QSKVND0CYL73J4hbmVo0V6KVZuM1uZl1oC+trLpewV8qHjsWqLIQvkn1KGX2StWvPMdWGBqim1xlC8kr11EKQ==" crossorigin="anonymous" referrerpolicy="
no-referrer"></script>
22    <!-- jQuery -->
23    <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-xUj+30JUSyExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="
anonymous"></script>
24    <!-- My Script -->
25    <script src="script.js"></script>
26
27    </body>
28    </html>

```

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.

Penjelasan Javascript

langkah yang pertama yaitu menunggu dokumen siap, lalu memanggil fungsi updateChart

```

C: > Users > user > Downloads > Telegram Desktop > JS script.js > updateChart > fetchData
1  $(document).ready(function(){
2    |    updateChart();
3  });

```

Untuk langkah selanjutnya sebagai berikut

```
4
5 function updateChart() {
6     async function fetchData() {
7         const url = './mahasiswa.json';
8         const response = await fetch(url);
9         // wait until the request is complete
10        const data = await response.json();
11        console.log(data);
12        return data;
13    };
14
```

setelah return data; kita masuk ke langkah selanjutnya

```
14
15    fetchData().then(data => {
16        const nama = data.map(
17            function(index) {return index.nama}
18        );
19
20        const sks = data.map(
21            function(index) {return index.sks}
22        );
23
24        myChart.config.data.labels = nama;
25        myChart.config.data.datasets[0].data = sks;
26        myChart.update()
27
28        barChart.config.data.labels = nama;
29        barChart.config.data.datasets[0].data = sks;
30        barChart.update()
31    });
32 }
33
```

Lalu pada bagian ini dan selanjutnya memberikan style pada laman website

```

33
34 const data = {
35   datasets: [{
36     label: 'Jumlah SKS yang telah ditempuh',
37     data: [12, 19, 3, 5, 2, 3],
38     backgroundColor: [
39       'rgba(255, 99, 132, 0.2)',
40       'rgba(54, 162, 235, 0.2)',
41       'rgba(255, 206, 86, 0.2)',
42       'rgba(75, 192, 192, 0.2)',
43       'rgba(153, 102, 255, 0.2)',
44       'rgba(255, 159, 64, 0.2)'
45     ],
46     borderColor: [
47       'rgba(255, 99, 132, 1)',
48       'rgba(54, 162, 235, 1)',
49       'rgba(255, 206, 86, 1)',
50       'rgba(75, 192, 192, 1)',
51       'rgba(153, 102, 255, 1)',
52       'rgba(255, 159, 64, 1)'
53     ],
54     borderWidth: 1
55   }]
56 }
57

```

```

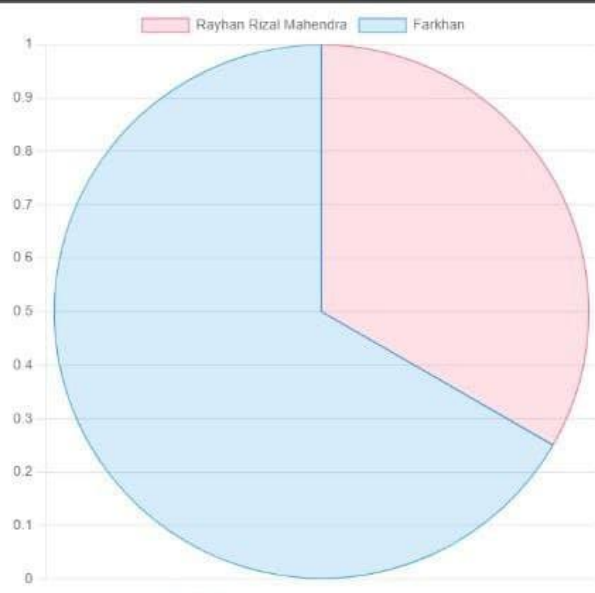
57
58 const config = {
59   type: 'pie',
60   data,
61   options: {
62     scales: {
63       y: {
64         beginAtZero: true
65       }
66     }
67   }
68 }
69
70 const bar = {
71   type: 'bar',
72   data,
73   options: {
74     scales: {
75       y: {
76         beginAtZero: true
77       }
78     }
79   }
80 }
81
82 const myChart = new Chart(
83   document.getElementById('myChart'),
84   config
85 )
86
87 const barChart = new Chart(
88   document.getElementById('barChart'),
89   bar
90 )

```

seperti yang kita lihat diatas, untuk diagram bulat kita menggunakan config type pie, sedangkan untuk bar kita menggunakan const bar type bar.

Penjelasan laman web

Berikut ini adalah tampilan web yang memuat chart lingkaran dan chart batang.



Untuk Diagram lingkaran ini sendiri diberi style dibagian HTML dan data yang dimuat dipanggil menggunakan javascript.



Sama seperti diagram lingkaran diatas, perbedaannya adalah disini kita menggunakan diagram batang untuk memuat informasi yang tersedia di database.