

Modul Panduan Praktikum Pemrograman Web

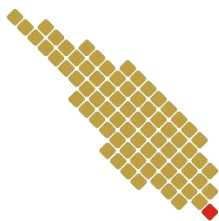
PERTEMUAN

5

JavaScript II

Asisten Praktikum

- Aminudin Fadila
- Defangga Aby Vonega
- Markus Togi Fedrian Rivaldi Sinaga



TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI PRODUKSI INDUSTRI DAN INFORMASI
INSTITUT TEKNOLOGI SUMATERA
2022

Dasar Teori

1. DOM

Document Object Model (DOM) merupakan representasi data dari seluruh objek yang meliputi struktur dan isi dalam halaman web. DOM pada dasarnya dapat dikatakan sebagai Application Programming Interface (API) untuk dokumen web. DOM merepresentasikan dokumen web sebagai *nodes* dan *objects*, sehingga program (JavaScript) dapat mengontrol struktur, *styling*, dan isi dari dokumen web tersebut.

*“DOM bukanlah bagian dari JavaScript, melainkan Web API yang digunakan untuk membangun website. DOM dikembangkan oleh **World Wide Web Consortium (W3C)** untuk memungkinkan bahasa pemrograman untuk mengakses sturuktur dasar dari sebuah dokumen web”*

Jadi, DOM tidak hanya tersedia di JavaScript, berikut adalah contoh penggunaan DOM di bahasa Python :

```
# Python DOM example
import xml.dom.minidom as m
doc = m.parse(r"C:\Projects\Py\chap1.xml")
doc.nodeName # DOM property of document object
p_list = doc.getElementsByTagName("para")
```

Terdapat beragam *object* dan *data types* dalam DOM, namun berikut adalah ini beberapa yang paling banyak digunakan dan penting untuk dikenali serta dipahami, di antaranya:

Type Data (Interface)	Deskripsi
<u>document</u>	Mewakili halaman web yang saat ini dimuat di browser kita.
<u>node</u>	Merupakan kelas abstrak yang menjadi dasar bagi hampir seluruh objek objek DOM lainnya, misalnya <u>element node</u> , <u>text node</u> , <u>attribute node</u> , <u>document node</u> , dsb.
<u>element</u>	Merupakan kelas yang berdasar pada kelas <u>node</u> , yang menjadi kelas bagi seluruh objek “elemen” (<u>node</u> bertipe elemen) di dalam dokumen.
<u>nodeList</u>	Sebuah <u>nodeList</u> merupakan array dari beberapa <u>node</u> (umumnya diterapkan pada <u>node</u> bertipe elemen). Item di dalam sebuah <u>nodeList</u> dapat diakses melalui indeksinya dengan 2 cara antara lain: <ul style="list-style-type: none">• <u>nodeList.item(index)</u>• <u>nodeList [index]</u>

Tipe Data (Interface)	Deskripsi
<u>attributes</u>	Meskipun jarang digunakan sebagai objek, faktanya attributes (yang umumnya dimiliki hampir seluruh element) juga merupakan node di dalam DOM sama seperti element itu sendiri. Jadi, meskipun kita akan lebih sering mengakses data nilai suatu attributes sebagai <i>String</i> , namun sebenarnya kita juga dapat mengaksesnya sebagai objek dengan menggunakan metode tertentu seperti element.getAttributeNode() .

Kemudian, berikut adalah beberapa *interfaces* penting dalam DOM yang banyak digunakan :

- **document.querySelector("selector")**
Mengembalikan elemen pertama dalam dokumen yang sesuai dengan selector yang diberikan.
- **document.querySelectorAll("selector")**
Mengembalikan nodelist berisi elemen-elemen dalam dokumen yang sesuai dengan selector yang diberikan.
- **document.getElementById("idName")**
Mengembalikan elemen yang memiliki atribut **id** yang sesuai dengan argumen diberikan.
- **document.getElementsByClassName("className")**
Mengembalikan nodelist berisi elemen-elemen yang memiliki atribut **class** yang sesuai dengan argumen diberikan.
- **document.createElement("anyValidTagName")**
Membuat elemen dengan **tag** yang sesuai dengan argumen diberikan.
- **parentNode.appendChild(node)**
Memasukkan node ke dalam elemen yang padanya digunakan metode ini.
- **element.innerHTML**
Mengakses elemen-keturunan dari elemen yang padanya digunakan metode ini.
- **element.innerText**
Mengakses teks di dalam elemen yang padanya digunakan metode ini.
- **element.style.anyValidStyleRule**
Memanipulasi *styling rule* elemen yang padanya digunakan metode ini.
- **element.setAttribute("attrName", "value")**
Menetapkan atribut dan nilainya yang diinginkan pada elemen yang padanya digunakan metode ini.
- **element.getAttribute("attrName")**
Mengakses atribut yang diinginkan dari elemen yang padanya digunakan metode ini.

- `element.addEventListener("anyValidEventType")`
Memberikan *event listener* (pendeteksi terjadinya aksi tertentu) pada elemen yang padanya digunakan metode ini.
- `Window.onload`
Berupa *event listener* yang mendeteksi apakah halaman sudah selesai dimuat sepenuhnya.

Selengkapnya terkait DOM, dapat dibaca di [Document Object Model](#)

2. Pengelolaan Event

Dengan DOM, JavaScript mampu mengakses dan memanipulasi elemen HTML dalam bentuk objek. Selanjutnya kita akan membahas terkait kemampuan JavaScript untuk mendeteksi kejadian (*event*) atau aksi yang terjadi pada dokumen HTML.

Tipe-tipe *event* dalam JavaScript :

- **Mouse event**, *event* yang berkaitan dengan aksi mouse seperti `click` dan `hover`.
- **Keyboard event**, *event* yang berkaitan dengan keyboard antara lain, `keypress`, `keydown`, dan `keyup`.
- **Inputs event**, *event* yang berkaitan dengan elemen *input* seperti `change` pada `<input>`, `<select>`, dan `<textarea>`.
- **Focus event**, *event* yang berkaitan dengan fokus pada suatu elemen seperti `focusin` dan `focusout`.
- **Form event**, *event* yang berkaitan dengan elemen `<form>` seperti `submit` dan `reset`.
- **Loading event**, *event* yang berkaitan dengan *loading* dan *unloading* pada dokumen atau kontennya, seperti `load` dan `loadstart`.
- **dsb.**

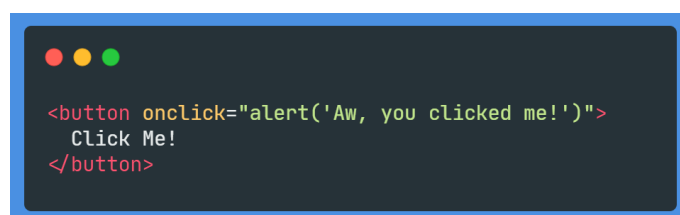
Selengkapnya terkait tipe-tipe *event* dalam JavaScript dapat dilihat di [Event Reference](#).

Mendeteksi Event

Untuk mendeteksi *event*, dapat kita lakukan dengan 2 cara, antara lain :

- Melalui **inline attribute**

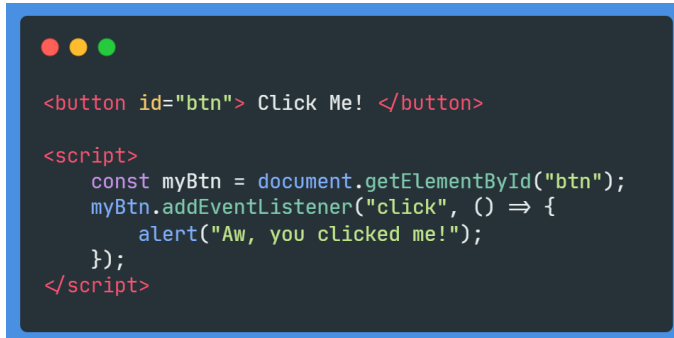
Biasanya ditambahkan "`on`" di awalan nama *event* untuk membentuk atribut pada elemen.



```
<button onclick="alert('Aw, you clicked me!')">
  Click Me!
</button>
```

- Menggunakan metode `addEventListener()`

Metode ini menerima 2 buah argumen, yang pertama adalah jenis *event* yang ingin dideteksi, dan yang kedua adalah fungsi berisi aksi apa yang ingin dieksekusi bila *event* yang dimaksud terdeteksi.



```
<button id="btn"> Click Me! </button>

<script>
  const myBtn = document.getElementById("btn");
  myBtn.addEventListener("click", () => {
    alert("Aw, you clicked me!");
  });
</script>
```

Selengkapnya terkait *event* dalam JavaScript dapat dilihat di [Events](#), dan daftar referensi lengkap terkait seluruh *event* di JavaScript dapat dilihat juga di [Event Listing](#).

3. AJAX

Asynchronous JavaScript and XML (AJAX) merupakan perpaduan dari beragam teknologi yang sudah ada, yang memanfaatkan XMLHttpRequest sebagai sarana komunikasi dengan *server side script*. Dengan AJAX memungkinkan kita untuk dapat menerima dan mengirim data/informasi dalam format JSON, XML, HTML dan TEXT tanpa perlu melakukan *reload* halaman web. Meskipun X dalam singkatan AJAX mewakili XML, namun format JSON lebih dipilih karena lebih ringan dan lebih mudah dipahami karena ditulis dalam bahasa JavaScript. Pada dasarnya AJAX menggunakan bahasa JavaScript namun dalam penggunaannya, kode JavaScript cukup sulit bagi pemula sehingga lebih mudah jika kita menggunakan *library* JavaScript yaitu jQuery.

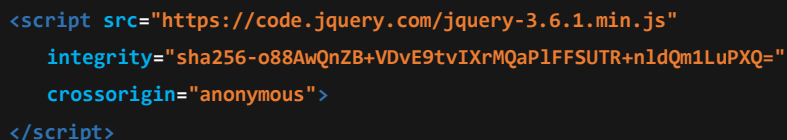
Beberapa fungsi AJAX :

- Mengambil data dari server secara background
- Mengirim data ke server secara background
- Mengubah tampilan web tanpa *reload page*

Instalasi jQuery

Untuk menggunakan jQuery bisa dengan beberapa cara, 2 di antaranya adalah :

- Mengunduh melalui : <https://code.jquery.com/jquery-3.6.1.min.js>
- Menggunakan CDN dengan menyalin kode berikut ke file HTML kita :



```
<script src="https://code.jquery.com/jquery-3.6.1.min.js"
  integrity="sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPXQ="
  crossorigin="anonymous">
</script>
```

Selengkapnya tentang AJAX dapat dibaca di [w3schools](#), di dokumentasi API [jQuery](#), atau di [mdn web docs](#).

Latihan Percobaan

Percobaan menggunakan jQuery untuk menjalankan metode-metode AJAX

❖ `load()` method

Metode `load()` adalah metode yang tersedia di dalam library jQuery yang digunakan untuk memuat data dari server yang kemudian dapat ditempatkan pada elemen tertentu di halaman website.

Ekspresi umum metode `load()` :

```
$(selector).load("url", data, callback);
```

Buat file **demo.php** yang akan menjadi file yang akan dimuat dengan AJAX menggunakan metode `load()` :

```
demo.php

<?php
    echo "Echo and Narcissus.";
?>
```

Buat file **index.php** yang akan menjadi file utama yang akan memuat informasi yang di-request menggunakan metode `load()` pada file **demo.php** :

```
index.php

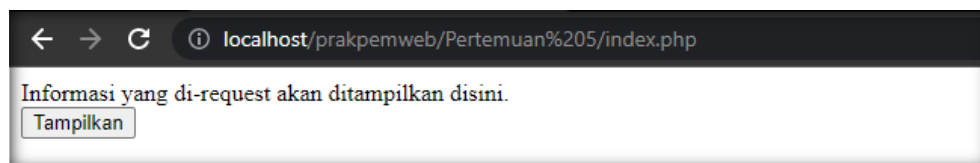
<div id="tampil">
    Informasi yang di-request akan ditampilkan disini.
</div>

<button id="btn_tampil"> Tampilkan </button>

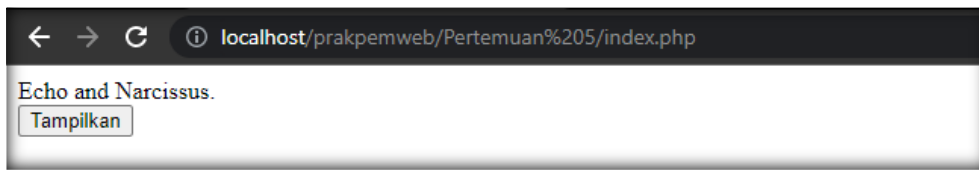
<script src = "https://code.jquery.com/jquery-3.6.1.min.js"
    integrity = "sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPXQ="
    crossorigin = "anonymous">
</script>

<script>
    $(document).ready(function () {
        $("#btn_tampil").click(function () {
            $("#tampil").load("demo.php");
        });
    });
</script>
```

Jalankan file **index.php** :



Setelah tombol “Tampilkan” diklik :



Selengkapnya tentang metode `load()` dapat dilihat di [sini](#).

❖ `get()` method

Metode `get()` adalah metode yang tersedia di dalam library jQuery yang digunakan untuk memuat data dari server melalui HTTP GET request. Ekspresi umum :

```
$.get("url", callback);
```

Buat file **info.php** yang akan menjadi file yang akan dimuat dengan AJAX menggunakan metode `get()` :

```
info.php

<?php
    echo "Echo and Narcissus.";
?>
```

Buat file **get.php** yang akan menjadi file utama yang akan memuat informasi yang di-request menggunakan metode `get()` pada file **info.php** :

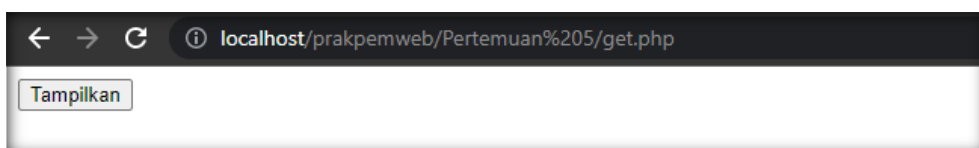
```
get.php

<button id="btn_tampil"> Tampilkan </button>

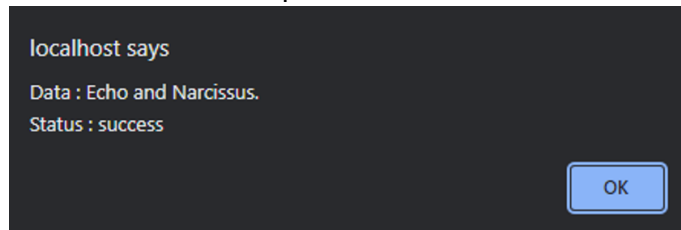
<script src = "https://code.jquery.com/jquery-3.6.1.min.js"
    integrity = "sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPXQ="
    crossorigin = "anonymous">
</script>

<script>
    $(document).ready(function () {
        $("#btn_tampil").click(function () {
            $.get("info.php", function (data, status) {
                alert("Data : "+ data +"\nStatus : "+ status);
            })
        });
    });
</script>
```

Jalankan file **get.php** :



Setelah tombol “Tampilkan” diklik :



Selengkapnya tentang metode `get()` dapat dilihat di [sini](#).

❖ `post()` method

Metode `post()` adalah metode yang tersedia di dalam library jQuery yang digunakan untuk mengirim data ke server melalui HTTP POST request. Ekspresi umum :

```
$.post("url", data, callback);
```

Buat file **info-post.php** yang akan menjadi file yang akan menerima data yang dikirim dengan AJAX menggunakan metode `post()` :

```
info-post.php

<?php
    $nama = $_POST["nama"];
    $nim = $_POST["nim"];
    echo "\nNama : ".$nama;
    echo "\nNim : ".$nim;
?>
```

Buat file **post.php** yang akan menjadi file utama yang akan mengirim informasi menggunakan metode `post()` kepada file **info-post.php** :

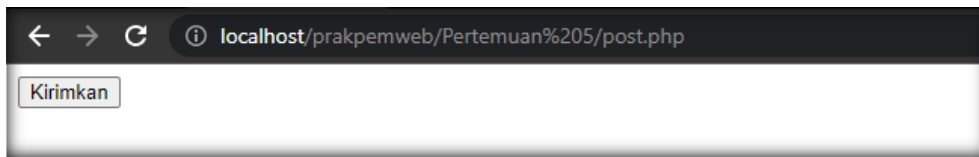
```
post.php

<button id="btn_kirim"> Kirimkan </button>

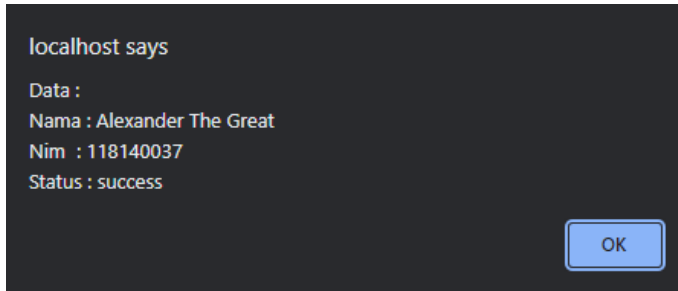
<script src = "https://code.jquery.com/jquery-3.6.1.min.js"
    integrity = "sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPXQ="
    crossorigin = "anonymous">
</script>

<script>
    $(document).ready(function () {
        $("#btn_kirim").click(function () {
            $.post(
                "info-post.php",
                { nama : "Alexander The Great", nim : "118140037" },
                function (data, status) {
                    alert("Data : "+ data +"\nStatus : "+ status);
                }
            );
        });
    });
</script>
```


Jalankan file **post.php** :



Setelah tombol “Kirimkan” diklik :



Selengkapnya tentang metode **post ()** dapat dilihat di [sini](#).

❖ **ajax ()** method

Metode **ajax ()** adalah metode yang tersedia di dalam library jQuery yang digunakan untuk melakukan HTTP request secara asinkronus. Ekspresi umum dari metode ini adalah :

```
$.ajax({ settings }).callbacks();
```

Pada parameter **settings** kita dapat mengirimkan argumen berupa objek dengan *key* dan *value* yang sesuai. Berikut ini adalah beberapa *key* dan tipe *value*-nya yang umum digunakan pada metode **ajax ()** :

Nama key	Tipe value	Deskripsi
url	<i>String</i>	String berupa tautan tujuan dari <i>HTTP request</i> yang ingin dilakukan
type / method	<i>String</i>	Mewakili metode <i>HTTP request</i> yang digunakan (' GET ', ' POST ' atau ' PUT ')
dataType	<i>String</i>	Tipe data yang diharapkan dari respon yang dikirimkan server (pilihannya : <i>xml</i> , <i>json</i> , <i>script</i> , or <i>html</i>)
data	<i>Object</i> atau <i>String</i> atau <i>Array</i>	Data yang dikirimkan ke server bersama <i>HTTP request</i> yang dikirimkan
error	<i>Function</i>	Fungsi yang dipanggil ketika <i>HTTP request</i> yang dilakukan mengalami kegagalan
success	<i>Function</i>	Fungsi yang dipanggil ketika <i>HTTP request</i> yang dilakukan berhasil
complete	<i>Function</i>	Fungsi yang dipanggil ketika <i>HTTP request</i> yang dilakukan sudah selesai

Untuk melakukan percobaan menggunakan metode **ajax()**, pertama kita akan membuat database bernama **belajar_ajax**, di dalam database tersebut kemudian kita buat tabel **mahasiswa** dengan 3 kolom / field seperti pada gambar berikut ini :

```
CREATE TABLE `belajar_ajax`.`mahasiswa` (  
  `nim` BIGINT NOT NULL,  
  `nama` VARCHAR(50) NOT NULL,  
  `prodi` VARCHAR(50) NOT NULL  
);
```

Kemudian buat file **koneksi.php** yang berfungsi untuk membuat koneksi ke database dengan beberapa konfigurasi pada **host**, **user**, **password** dan **database** yang digunakan. Pastikan konfigurasi ini kalian sesuaikan dengan yang kalian pakai :

koneksi.php

```
<?php  
$host      = "localhost";  
$user      = "root";  
$password  = "";  
$dbName    = "belajar_ajax";  
  
$connection = mysqli_connect($host, $user, $password, $dbName);  
  
if(!$connection){  
    die("Koneksi gagal : ".mysqli_connect_error());  
}  
?>
```

Buat file **tambah.php** untuk menerima data yang akan dikirim dari form menggunakan metode **ajax()** yang selanjutnya akan kita *insert* ke dalam tabel mahasiswa :

tambah.php

```
<?php  
include "koneksi.php";  
  
$nim      = $_POST["nim"];  
$nama     = $_POST["nama"];  
$prodi    = $_POST["prodi"];  
  
$query    = "INSERT INTO mahasiswa (nim, nama, prodi)  
            VALUES ('$nim', '$nama', '$prodi')";  
  
$result   = mysqli_query($connection, $query);  
?>
```

Berikutnya, buat file **tampil.php** yang berfungsi untuk menampilkan setiap data mahasiswa dari database ke halaman website kita. Data ini nantinya akan kita tampilkan menggunakan metode `load()` yang sudah kita bahas pada materi sebelumnya :

tampil.php

```
<table border="1">
  <tr>
    <th> No. </th>
    <th> NIM </th>
    <th> Nama </th>
    <th> Program Studi </th>
  </tr>

  <?php
    include "koneksi.php";

    $query = "SELECT * FROM mahasiswa ORDER BY nim ASC";

    $result = mysqli_query($connection, $query);
    $i = 0;

    while ($data = mysqli_fetch_array($result)) :
      $i++;
    ?>
    <tr>
      <td> <?= $i ?> </td>
      <td> <?= $data["nim"] ?> </td>
      <td> <?= $data["nama"] ?> </td>
      <td> <?= $data["prodi"] ?> </td>
    </tr>
  <?php
    endwhile
  ?>
</table>
```

Terakhir, buat file **index.php** yang berfungsi sebagai halaman utama saat program dijalankan dari **localhost**. Dengan menampilkan form sederhana untuk menerima data mahasiswa, dan tabel sederhana untuk menampilkan data yang telah dimasukkan :

index.php

```
<form id="form_mahasiswa">
  <label for="nim">
    NIM
    <input type="number" name="nim" id="nim" autocomplete="off">
  </label>
  <label for="nama">
    Nama
    <input type="text" name="nama" id="nama" autocomplete="off">
  </label>
  <label for="prodi">
    Prodi
  </label>
  <select name="prodi" id="prodi">
    <option value="IF"> Teknik Informatika </option>
    <option value="EL"> Teknik Elektro </option>
    <option value="ME"> Teknik Mesin </option>
    <option value="TG"> Teknik Geofisika </option>
    <option value="GL"> Teknik Geologi </option>
  </select>
</form>

<button id="btn_tampil"> Tampilkan </button>

<br>
<div id="tampil_data"></div>

<script src = "https://code.jquery.com/jquery-3.6.1.min.js"
  integrity = "sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPXQ="
  crossorigin = "anonymous">
</script>

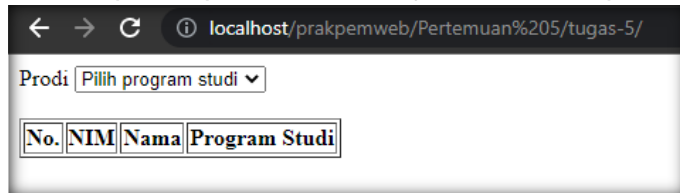
<script>
$(document).ready(function () {
  $("#tampil_data").load("tampil.php");
  $("#btn_tampil").click(function () {
    let data = $("#form_mahasiswa").serialize();
    $.ajax({
      type      : "POST",
      url       : "tambah.php",
      data      : data,
      success   : function (response) {
        $("#tampil_data").load("tampil.php");
      }
    });
  });
});
</script>
```

Tugas

Buatlah aplikasi web yang dapat menampilkan data mahasiswa ketika pengguna melakukan perubahan pada pilihan program studi yang tersedia menggunakan bantuan AJAX.

Spesifikasi :

1. Aplikasi terhubung ke database yang memiliki tabel mahasiswa dengan minimal 3 data yaitu NIM, Nama, dan Program Studi.
2. Terdapat minimal 10 baris data mahasiswa.
3. Ketika aplikasi pertama kali berjalan akan tampak seperti ini :

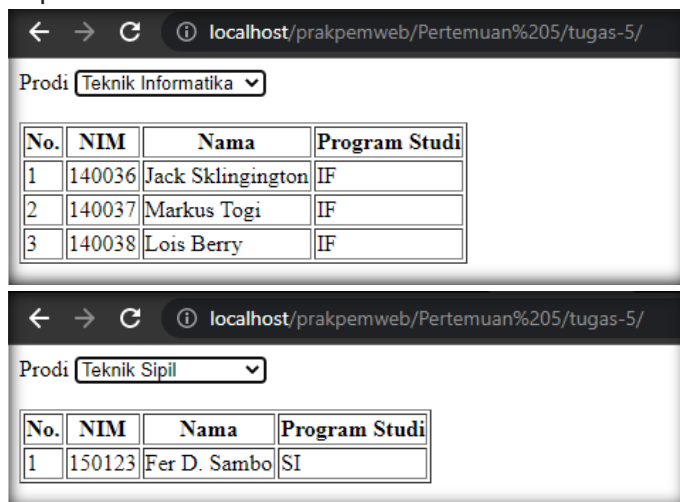


localhost/prakpemweb/Pertemuan%205/tugas-5/

Prodi Pilih program studi ▼

No.	NIM	Nama	Program Studi
-----	-----	------	---------------

4. Ketika pengguna mengubah program studi maka akan ditampilkan data yang sesuai seperti ini :



localhost/prakpemweb/Pertemuan%205/tugas-5/

Prodi Teknik Informatika ▼

No.	NIM	Nama	Program Studi
1	140036	Jack Sklingington	IF
2	140037	Markus Togi	IF
3	140038	Lois Berry	IF

localhost/prakpemweb/Pertemuan%205/tugas-5/

Prodi Teknik Sipil ▼

No.	NIM	Nama	Program Studi
1	150123	Fer D. Sambo	SI

Parameter Penilaian	Bobot Penilaian
Kelengkapan	30%
Kesesuaian penggunaan event pada JavaScript, dan metode dalam jQuery	40%
Program berjalan sesuai spesifikasi yang diberikan	30%

Daftar Pilihan Referensi

- [mdn web docs](#)
- [W3Schools](#)
- [JavaScript Dasar - Web Programming Unpas \(YouTube\)](#)
- [JavaScript Lanjutan - Web Programming Unpas \(YouTube\)](#)