**LAPORAN HASIL PRAKTIKUM**

**PEMROGRAMAN WEB DAN MOBILE 1**

****

**NAMA : AXEL BERKATI**

**NIM : 193010503007**

**KELAS : A**

**MODUL : IV (ASYNCHRONOUS JAVASCRIPT AND XML (AJAX))**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2021**

**BAB 1**

**TUJUAN DAN LANDASAN TEORI**

* 1. **Tujuan**
     1. **Mahasiswa mampu membuat program dengan menggabungkan fitur PHP dan JavaScript.**
     2. **Mahasiswa mampu membuat program web yang dinamis.**
  2. **Landasan Teori**

AJAX diperkenalkan oleh Jesse James Garret dari Adaptive Path pada tahun 2005. Ia mendeskripsikan bagaimana mengembangkan web yang berbeda dengan metode tradisional(Sunyoto, 2010). Ia mempublikasikan sebuah artikel yang berjudul *“AJAX: A New* *Approach to Web Applications”*. Pada artikelnya, Garret yakin bahwa aplikasi web dapat menutup jurang pemisah antara web dan aplikasi desktop.

Pengembangan web secara tradisional bekerja secara *synchronously*, antara aplikasi dan server, setiap kali melakukan link atau melakukan operasi “submit” pada form. Caranya, browser mengirim data ke server, server merespons dan seluruh halaman akan di refresh.

Aplikasi web yang bekerja dengan AJAX bekerja secara *asynchronously*, yang berarti mengirim dan menerima data dari user ke server tanpa perlu me-load kembali seluruh halaman, melainkan hanya melakukan penggantian pada bagian web yang hendak diubah. Penggunaan AJAX mulai popular ketika digunakan oleh Google pada tahun 2005.

AJAX bukanlah bahasa pemrograman baru, tetapi merupakan teknik baru penggunaan standar yang telah ada. Dengan AJAX kita dapat menjadi lebih baik, cepat dan menambah unsur user-friendly dan interaktif pada aplikasi web kita. AJAX berbasiskan pada JavaScript dan request HTTP.

AJAX berbasiskan standar terbuka seperti:

1. JavaScript
2. XML
3. HTML/XHTML
4. CSS

Dengan menggunakan JavaScript AJAX dapat mengirim dan menerima data antara web server dan web browser. Teknik yang dimiliki AJAX akan bergantian bertukar data dan mere-load ulang seluruh halaman.

Dengan demikian, untuk mempelajari AJAX, harus dipahami pula konsep standar di atas. AJAX di-support sebagian besar browser popular sehingga aplikasi AJAX adalah aplikasi *cross-platform* dan *cross-browser*.

Melalui AJAX, JavaScript dapat dikomunikasikan secara langsung dengan server menggunakan obyek JavaScript XMLHttpRequest. Obyek JavaScript ini dapat men-*trade* data sebuah web server tanpa harus me-*reload* (*refresh*) halaman web.

AJAX menggunakan *asynchronouse* data transfer (pada HTTP request) antara browser dan web server, yang memperbolehkan halaman web me-request bit yang kecil atau seluruh informasi dari server. Teknik AJAX membuat aplikasi internet menjadi kecil, cepat dan lebih user-friendly.

AJAX adalah aplikasi web yang lebih baik. Aplikasi web menambah

keuntungan dibanding aplikasi desktop:

1. Dapat menjangkau pengguna yang luas
2. Mudah diinstal
3. Mudah dikembangkan
4. Mudah dipelihara

Seperti yang kita ketahui, aplikasi internet tidak selalu susah dan user friendly seperti aplikasi dekstop. Dengan AJAX aplikasi internet semakin kecil, cepat dan mudah digunakan.

* + 1. **Model AJAX**

Perlu diingat, AJAX akan berkomunikasi dengan server tanpa harus me-refresh semua halaman. Konsep ini berbeda dengan aplikasi web sistem tradisional. Dalam model AJAX, aksi dari sisi klien dibagi menjadi dua bagian, yaitu layer user interface dan layer AJAX.



**Gambar 1.1. Arsitektur model AJAX**

Ketika user mengklik sebuah link atau mengirimkan sebuah form maka input tersebut akan ditangani oleh layer AJAX dan diinteraksikan dengan server, kemudian meng-update user interface (UI). Jadi, dalam AJAX, interaksi UI secara logika terpisah dengan interaksi jaringan. Pendekatan yang digunakan AJAX adalah bagaimana mengirimkan jumlah data yang kecil dari dan ke server atas request dari user. Model aplikasi web tradisional di mana browser akan merespons sendiri atas inisialisasi request, memproses request dari web server. Dalam AJAX terdapat layer yang diberi nama “AJAX engine” untuk menangani komunikasi. “AJAX Engine” adalah sebuah obyek JavaScript atau function yang dipanggil ketika informasi diperlukan dari server.

Proses yang dilakukan mesin AJAX (*AJAX Engine*) adalah menerima respons dari server, melakukan parsing data dan melakukan sedikit perubahan untuk keperluan menampilkan informasi. Oleh karena proses mengirimkan sedikit informasi, jika dibanding dengan model aplikasi web tradisional maka user interface di-update lebih cepat dan user mampu melakukan pekerjaannya dengan cepat.

Beberapa poin penting untuk menggambarkan AJAX:

1. Layer AJAX tidak memerlukan komunikasi dengan server (contohnya untuk validasi form karena dapat ditangani sepenuhnya oleh *client-side*).
2. Oleh karena request antara layer AJAX dan server berupa bagian kecil dari informasi (tidak komplit satu halaman) maka sering digunakan untuk interaksi dengan database sehingga waktu render dan waktu pengiriman menjadi pendek.
3. Layer UI secara langsung tergantung pada respons server sehingga user dapat melanjutkan interaksi dengan sebuah halaman selama aktivitas dikerjakan di background (*background process*)*.* Berarti, untuk beberapa interaksi, waktu tunggu user hampir tidak ada.
4. Komunikasi antarhalaman dan server tidak selama memerlukan AJAX untuk mengubah perubahan UI. Contoh, beberapa aplikasi menggunakan AJAX untuk notifikasi dengan halaman, tetapi tidak melakukan apa pun terhadap response dari server.
   * 1. **Teknologi di Balik AJAX**

Teknologi AJAX di dalamnya meliputi beberapa komponen, yaitu:



**Gambar 1.2. Teknologi di balik AJAX**

1. XHTML (*Extensible HyperText Markup Language*) adalah bahasa markup seperti HTML, tetapi dengan gaya bahasa lebih baik. XHTML digunakan membuat halaman web dan dokumen-dokumen lain yang dapat ditampilkan dalam browser. XHTML merupakan standard internasional dengan spesifikasi yang ditetapkan oleh W3C (*World Wide Web* *Consortium*)*.* Versi terakhir saat tulisan ini dibuat adalah HTML 2.0.
2. CSS (*Cascading Style Sheets*) adalah sebuah mekanisme sederhana untuk memberikan style (seperti font, warna, jarak spasi, dll) kepada dokumen web yang ditulis dalam HTML atau XML (termasuk beberapa variasi bahasa XML seperti XHTML dan SVG).
3. JavaScript adalah bahasa scripting kecil, ringan, berorientasiobjek dan lintas platform. JavaScript tidak dapat berjalan dengan baik sebagai bahasa mandiri, melainkan dirancang untuk ditanamkan pada produk.
4. DOM (*Document Object Model*) adalah sebuah API (*Application Program Interface*) untuk dokumen HTML dan XML. DOM menyediakan representasi dokumen secara terstruktur, dimungkinkan untuk mengubah isi dan presentasi visual. Pada dasarnya, DOM menghubungkan halaman web dengan script atau bahasa pemrograman.
5. XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan khusus. Keperluan utama XML adalah untuk pertukaran data antarsistem yang beraneka ragam.
6. XSLT (*Extensible Stylesheet Language Transformations*) adalah sebuah bahasa berbasis-XML untuk transformasi dokumen XML. Walaupun proses merujuk pada transformasi, dokumen asli tidak berubah melainkan dokumen XML baru dibuat dengan basis isi dokumen yang sudah ada. XSLT biasanya digunakan untuk mengubah skema XML ke halaman web atau dokumen PDF.
7. Objek XMLHttpRequest berkemampuan melakukan pertukaran data secara asinkron dengan web server. AJAX menggunakan obyek XMLHttpRequest untuk melakukan pertukaran data dengan web server.
8. JSON *(JavaScript Object Notation*) yaitu format pertukaran data komputer yang ringan dan mudah. Keuntungan JSON dibandingkan dengan XML adalah pada proses penerjemahan data menggunakan Javascript. Javascript dapat menerjemahkan JSON menggunakan built-in procedure eval().

Dalam kenyataannya, semua teknologi dapat digunakan untuk AJAX, tetapi hanya tiga yang dibutuhkan: HTML/XHTML, DOM dan JavaScript. XHTML ternyata diperlukan untuk menampilkan informasi, sedangkan DOM diperlukan untuk halaman XHTML tanpa di-reload ulang.

* + 1. **Pengguna AJAX**

Beberapa web komersial sudah menggunakan teknologi AJAX untuk menaikkan pelayanan terhadap user. Berikut beberapa contoh web yang menggunakan AJAX.

1. Google Suggest: http://www.google.com/webhp?complete=1
2. Gmail: http://www.gmail.com
3. Google Maps: http://maps.google.com
4. Yahoo! News: http://news.yahoo.com
5. Bitflux Blog: http://blog.bitflux.ch
6. Dll
   * 1. **Keistimewaan AJAX**

AJAX sebagai sebagai teknik pemrograman baru memiliki beberapa keistimewaan. Keistimewaan AJAX antara lain:

1. Membuat permintaan kepada server tanpa memuat kembali (*reload*) halaman.
2. Mengurai (*parse*) dan bekerja dengan dokumen XML dan atau JSON.
3. Data yang dikirim sedikit sehingga menghemat bandwidth dan mempercepat koneksi.
4. Proses dilakukan di belakang layar.
5. Banyak didukung oleh browser-browser modern yang popular.
6. Aplikasi yang dibangun semakin interaktif dan dinamis.

Penggunaan AJAX akan mendatangkan beberapa keuntungan.

Keuntungan tersebut antara lain.

1. *High Interactivity*: Aplikasi AJAX lebih interaktif dibanding dengan aplikasi web konvensional.
2. *High Usability*: Update data tidak me-reload keseluruhan halaman, melainkan hanya yang relevan.
3. *High Speed*: Aplikasi AJAX lebih cepat dibanding dengan aplikasi web konvensional
   * 1. **Membuat Grafik Dari Database MySQL dan PHP dengan Chart.JS**

Chart.js adalah salah satu library Javascript yang digunakan untuk membuat grafik.

1. **Persiapan Library Chart.js**

Sebelumnya kita memulai kita akan persiapkan dulu library Chart.js, silahkan buka website resmi Chart.js :

https://www.chartjs.org/

didalamnya terdapat banyak jenis chart yang bisa anda buat dengan menggunakan Chart.js, anda melihat contoh grafiknya di menu sample : https://www.chartjs.org/samples/latest/

1. **Buat Folder Project**

Berikutnya silahkan buat folder projectnya. Karena dalam contoh ini kita menggunakan xampp, sehingga kita akan simpan file project di drive C:/xampp/htdocs

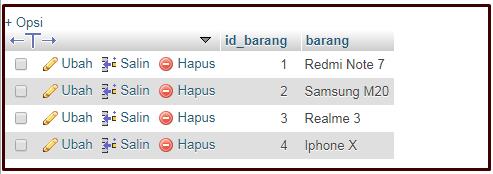
Dalam contoh ini kita akan buat folder dengan nama belajarchart, sehingga folder project kita ada di direktori :

C:/xampp/htdocs/belajarchart

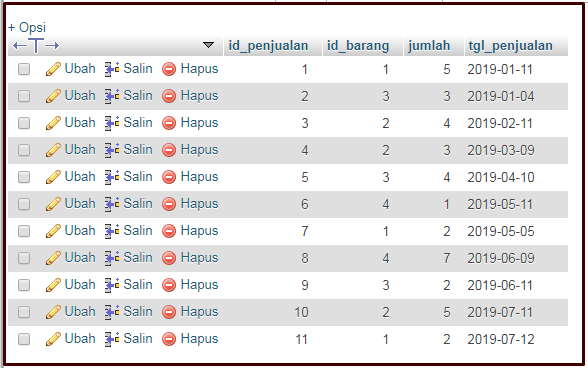
Pada direktori tersebut, silahkan letakkan file Chart.js hasil download sebelumnya, lalu berikutnya silahkan buat 4 file PHP lainnya :

1. grafik\_bulan.php
2. grafik\_pie.php
3. index.php
4. koneksi.php
5. **Membuat Database**

Berikutnya silahkan buat sebuah database dengan nama db\_penjualan, yang didalamnya terdapat 2 tabel, antara lain:

****

**Gambar 1.3. Tabel tb\_barang**

****

**Gambar 1.4. Tabel tb\_penjualan**

1. **Membuat File Koneksi.php**

Buka file koneksi.php yang sebelumnya telah di buat, lalu isikan code dibawah ini :

|  |
| --- |
| <?php  $host = "localhost";  $user = "root";  $password = "";  $database = "db\_penjualan";  $koneksi = mysqli\_connect($host, $user, $password, $database);  ?> |

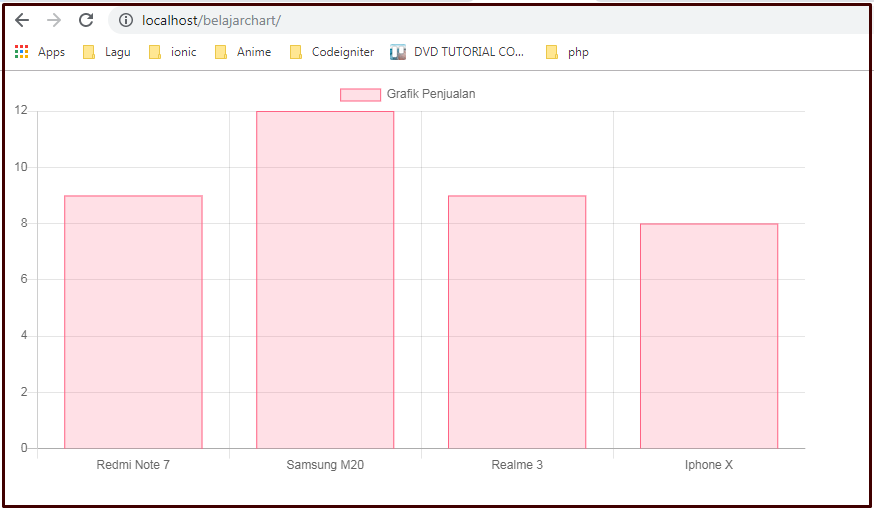
Keterangan :

1. Perintah diatas digunakan untuk koneksi ke database db\_penjualan yang sebelumnya telah kita buat.
2. $host dengan nilai “localhost” yang merupakan alamat server databasenya
3. $user dengan nilai “root” yang merupakan user dari databasenya
4. $password dengan nilai “” yang merupakan password untuk mengakses database, nilainya kosong karena default dari xampp untuk password user root adalah kosong
5. $database dengan nilai “db\_penjualan” yang merupakan nama dari database yang sebelumnya telah kita buat
6. Baris 6 adalah perintah untuk melakukan koneksi ke database, dan hasilnya disimpan di variabel $koneksi.
7. **Membuat Grafik Penjualan Per Barang**

Buka file index.php yang sebelumnya telah dibuat buat, lalu isikan code dibawah ini :

|  |
| --- |
| <?php  include('koneksi.php');  $produk = mysqli\_query($koneksi,"select \* from tb\_barang");  while($row = mysqli\_fetch\_array($produk)){  $nama\_produk[] = $row['barang'];    $query = mysqli\_query($koneksi,"select sum(jumlah) as jumlah from tb\_penjualan where id\_barang='".$row['id\_barang']."'");  $row = $query->fetch\_array();  $jumlah\_produk[] = $row['jumlah'];  }  ?>  <!DOCTYPE html>  <html>  <head>  <title>Membuat Grafik Menggunakan Chart JS</title>  <script type="text/javascript" src="Chart.js"></script>  </head>  <body>  <div style="width: 800px;height: 800px">  <canvas id="myChart"></canvas>  </div>  <script>  var ctx = document.getElementById("myChart").getContext('2d');  var myChart = new Chart(ctx, {  type: 'bar',  data: {  labels: <?php echo json\_encode($nama\_produk); ?>,  datasets: [{  label: 'Grafik Penjualan',  data: <?php echo json\_encode($jumlah\_produk); ?>,  backgroundColor: 'rgba(255, 99, 132, 0.2)',  borderColor: 'rgba(255,99,132,1)',  borderWidth: 1  }]  },  options: {  scales: {  yAxes: [{  ticks: {  beginAtZero:true  }  }]  }  }  });  </script>  </body>  </html> |

Untuk mengakses file index.php anda bisa mengaksesnya melalui alamat: localhost/belajarchart. Maka tampilannya adalah seperti berikut ini :



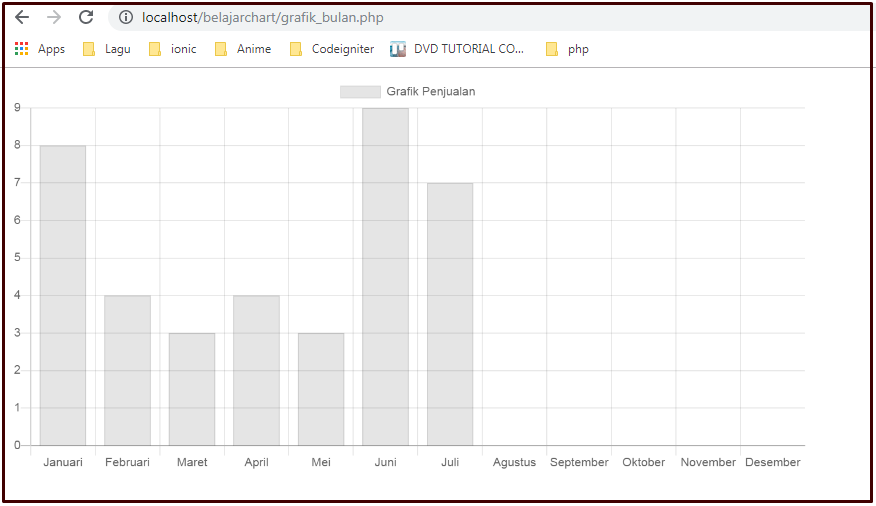
**Gambar 1.5. Tampilan Grafik Penjualan Per Barang**

1. **Membuat Laporan Penjualan Per Bulan**

Berikutnya kita akan membuat laporan penjualan per bulan, silahkan buka file grafik\_bulan.php, lalu isikan code dibawah ini:

|  |
| --- |
| <?php  include('koneksi.php');  $label = ["Januari","Februari","Maret","April","Mei","Juni","Juli","Agustus","September","Oktober","November","Desember"];  for($bulan = 1;$bulan < 13;$bulan++)  {  $query = mysqli\_query($koneksi,"select sum(jumlah) as jumlah from tb\_penjualan where MONTH(tgl\_penjualan)='$bulan'");  $row = $query->fetch\_array();  $jumlah\_produk[] = $row['jumlah'];  }  ?>  <!DOCTYPE html>  <html>  <head>  <title>Membuat Grafik Menggunakan Chart JS</title>  <script type="text/javascript" src="Chart.js"></script>  </head>  <body>  <div style="width: 800px;height: 800px">  <canvas id="myChart"></canvas>  </div>  <script>  var ctx = document.getElementById("myChart").getContext('2d');  var myChart = new Chart(ctx, {  type: 'bar',  data: {  labels: <?php echo json\_encode($label); ?>,  datasets: [{  label: 'Grafik Penjualan',  data: <?php echo json\_encode($jumlah\_produk); ?>,  borderWidth: 1  }]  },  options: {  scales: {  yAxes: [{  ticks: {  beginAtZero:true  }  }]  }  }  });  </script>  </body>  </html> |

Untuk menampilkan grafiknya anda bisa mengakses alamat: localhost/belajarchart/grafik\_bulan.php. Maka tampilannya adalah seperti berikut ini :

****

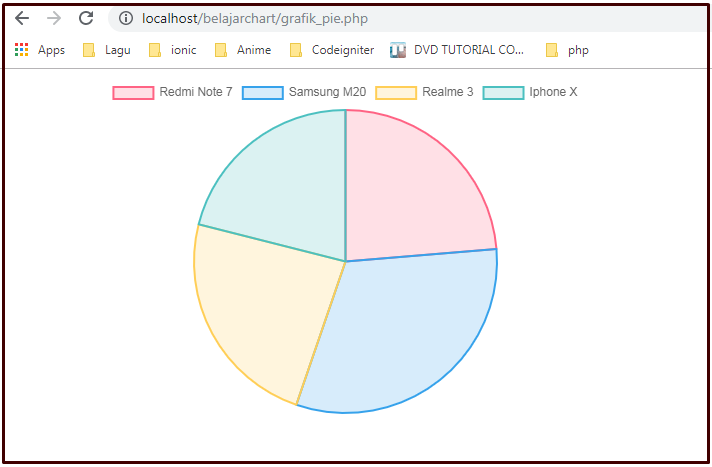
**Gambar 1.6. Tampilan Grafik Penjualan Per Bulan**

1. **Membuat Grafik PIE**

Berikutnya kita akan membuat grafik PIE untuk menampilkan presentase penjualan dengan bentuk PIE, baik silahkan buka file grafik\_pie.php yang sebelumnya telah kita buat, lalu isikan dengan code dibawah ini:

|  |
| --- |
| <?php  include('koneksi.php');  $produk = mysqli\_query($koneksi,"select \* from tb\_barang");  while($row = mysqli\_fetch\_array($produk)){  $nama\_produk[] = $row['barang'];    $query = mysqli\_query($koneksi,"select sum(jumlah) as jumlah from tb\_penjualan where id\_barang='".$row['id\_barang']."'");  $row = $query->fetch\_array();  $jumlah\_produk[] = $row['jumlah'];  }  ?>  <!doctype html>  <html>  <head>  <title>Pie Chart</title>  <script type="text/javascript" src="Chart.js"></script>  </head>  <body>  <div id="canvas-holder" style="width:50%">  <canvas id="chart-area"></canvas>  </div>  <script>  var config = {  type: 'pie',  data: {  datasets: [{  data:<?php echo json\_encode($jumlah\_produk); ?>,  backgroundColor: [  'rgba(255, 99, 132, 0.2)',  'rgba(54, 162, 235, 0.2)',  'rgba(255, 206, 86, 0.2)',  'rgba(75, 192, 192, 0.2)'  ],  borderColor: [  'rgba(255,99,132,1)',  'rgba(54, 162, 235, 1)',  'rgba(255, 206, 86, 1)',  'rgba(75, 192, 192, 1)'  ],  label: 'Presentase Penjualan Barang'  }],  labels: <?php echo json\_encode($nama\_produk); ?>},  options: {  responsive: true  }  };  window.onload = function() {  var ctx = document.getElementById('chart-area').getContext('2d');  window.myPie = new Chart(ctx, config);  };  document.getElementById('randomizeData').addEventListener('click', function() {  config.data.datasets.forEach(function(dataset) {  dataset.data = dataset.data.map(function() {  return randomScalingFactor();  });  });  window.myPie.update();  });  var colorNames = Object.keys(window.chartColors);  document.getElementById('addDataset').addEventListener('click', function() {  var newDataset = {  backgroundColor: [],  data: [],  label: 'New dataset ' + config.data.datasets.length,  };  for (var index = 0; index < config.data.labels.length; ++index) {  newDataset.data.push(randomScalingFactor());  var colorName = colorNames[index % colorNames.length];  var newColor = window.chartColors[colorName];  newDataset.backgroundColor.push(newColor);  }  config.data.datasets.push(newDataset);  window.myPie.update();  });  document.getElementById('removeDataset').addEventListener('click', function() {  config.data.datasets.splice(0, 1);  window.myPie.update();  });  </script>  </body>  </html> |

Untuk menampilkan grafik, silahkan mengakses menggunakan alamat: localhost/belajarchart/grafik\_pie.php. Maka tampilannya akan seperti berikut:

****

**Gambar 1.7. Tampilan Grafik PIE Penjualan Per Barang**

* + 1. **Membuat Polling AJAX Sederhana Dengan jQuery dan PHP**

1. **Membuat Database**

Kita buat dulu database dan tabel yang akan menyimpan pertanyaan dan jawaban hasil pollingnya, isinya silahkan diisikan sendiri.

1. **Membuat file koneksi.php**

Untuk polling AJAX ini, kembali saya menggunakan jQuery sebagai framework javascript-nya. Kita buat dulu file *koneksi.php*-nya.

|  |
| --- |
| <?php  $host="localhost";  $user="root";  $pass="";  $db="db\_poll";  mysql\_connect($host,$user,$pass);  mysql\_select\_db($db);  ?> |

1. **Membuat file index.php**

Pada file *index.php*, kita akan membuat form polling-nya dengan syntax seperti di bawah ini:

|  |
| --- |
| <div id="loading" style="display:none"><img src="images/loading.gif" /><br />Mohon tunggu. Vote anda sedang disimpan.....</div>  <div id="results">  <h4>  <?php  include('koneksi.php');  $q = mysql\_query("select \* from tbl\_poll where status='Y'");  $id = "";  $soal = "";  while($r=mysql\_fetch\_array($q))  {  $id=$r['id\_soal'];  $soal=$r['soal'];  echo $r['soal'];  }  ?>  </h4><br>  <form name="myform" id="myform" action="" method="POST">  <input type="hidden" value="<?php echo $id; ?>" name="id\_soal">  <input type="hidden" value="<?php echo $soal; ?>" name="soal">  <?php  $qj = mysql\_query("select \* from tbl\_jwb\_poll where id\_soal='$id'");  $id = "";  while($rj=mysql\_fetch\_array($qj))  {  ?>  <input type="radio" name="name" checked="checked" value="<?php echo $rj['id\_jwb']; ?>"/> <?php echo $rj['jawaban']; ?><br>  <?php  }  ?>  <br><input type="submit" name="submit" value="Submit">  </form>  <div> |

Sekarang kita menuju ke bagian javascript-nya, ketikkan syntax javascript di bawah ini diantara tag *head*

|  |
| --- |
| <script type="text/javascript" src="js/jquery.js"></script>  <script type="text/javascript" src="js/jquery.validate.min.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $("#myform").validate({  debug: false,  rules: {  name: "required"  },  messages: {  name: "Pilih pilihan anda.",  },  submitHandler: function(form) {  $.post("process.php", $("#myform").serialize(), function(data) {  $('#results').html(data);  });  }  });  });  $(function() {  $('#loading').ajaxStart(function(){  $(this).fadeIn();  }).ajaxStop(function(){  $(this).fadeOut();  });  $('#menu a').click(function() {  var url = $(this).attr('href');  $('#results').load(url);  return false;  });  });  </script> |

1. **Membuat process.php**

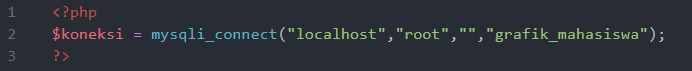
Hasil vote yang diberikan oleh user, akan diproses oleh file process.php dan disimpan ke dalam database dengan syntax php di bawah ini:

|  |
| --- |
| <?php  include('koneksi.php');  $id = $\_POST['id\_soal'];  $idj = $\_POST['name'];  $soal = $\_POST['soal'];  $jum = 0;  $qjum = mysql\_query("select \* from tbl\_jwb\_poll where id\_soal='$id'");  while($rjum=mysql\_fetch\_array($qjum))  {  $jum += $rjum['vote'];  }  setcookie("pengunjung", "sudah berkunjung", time() + 200 \* 24);  if (!isset($\_COOKIE["pengunjung"]))  {  echo '<h4>'.$soal.'</h4><br>';  $q = mysql\_query("update tbl\_jwb\_poll set vote=vote+1 where id\_jwb='$idj'");  $qj = mysql\_query("select \* from tbl\_jwb\_poll where id\_soal='$id'");  while($rj=mysql\_fetch\_array($qj))  {  $pr = sprintf("%2.1f",(($rj['vote']/$jum)\*100));  $gbr = $pr \* 3;  echo ''.$rj['jawaban'].'<br>  <div style="padding:10px; float:none; margin-bottom:5px; background-color:#000; width:'.$gbr.'px; color:#fff;" >'.$pr.'%</div>';  }  }  else  {  echo "<h4>Anda sudah melakukan voting sebelumnya..!!!<br>Terima Kasih telah melakukan voting.</h4>";  }  ?> |

**BAB II**

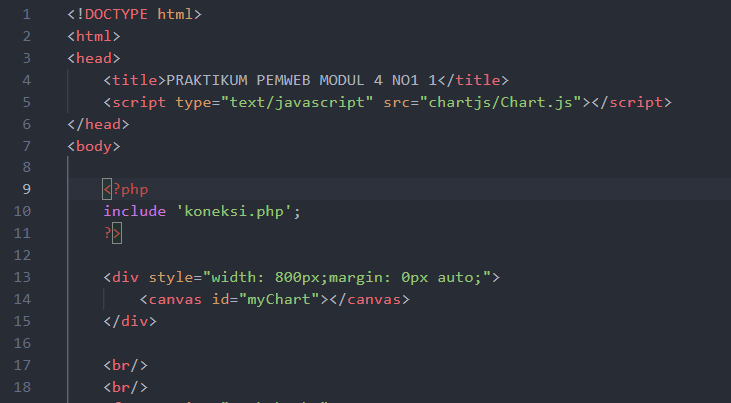
**PEMBAHASAN**

Untuk membuat program pada tugas 1 Modul 4, digunakan 3 buah file php, yaitu konseksi.php, index.php, tambah.php, dan library chartjs untuk menampilkan grafik dari hasil survey. Berikut penjelasan dari masing-masing code pada tiap php.

****

**Gambar 2.1. Koneksi.php**

Pada code diatas, pertama-tama dibuat pembuka php, yang dimana berisi code untuk menghubungkan PHP dengan MySQL, dengan menggunakan fungsi mysqli\_connect() dengan alamat server sebagai parameter pertama yaitu localhost, user database sebagai parameter kedua defaultnya root, password user sebagai parameter ketiga (defaultnya kosong), dan nama database sebagai parameter keempat yaitu grafik\_mahasiswa. Fungsi ini didefinisikan ke variabel $koneksi.

****

**Gambar 2.2. Index.php(1)**

Pada code diatas, pertama-tama dibuat struktur html terlebih dahulu. Kemudian, pada bagian head dibuat title, yaitu PRAKTIKUM PEMWEB MODUL 4 NO 1, kemudian membuat penghubung dengan chart.js. Pada bagian body, dibuat pembuka php untuk memanggil koneksi.php. kemudian, terdapat code untuk membuat element canvas dan diberi nama idnya myChart.

****

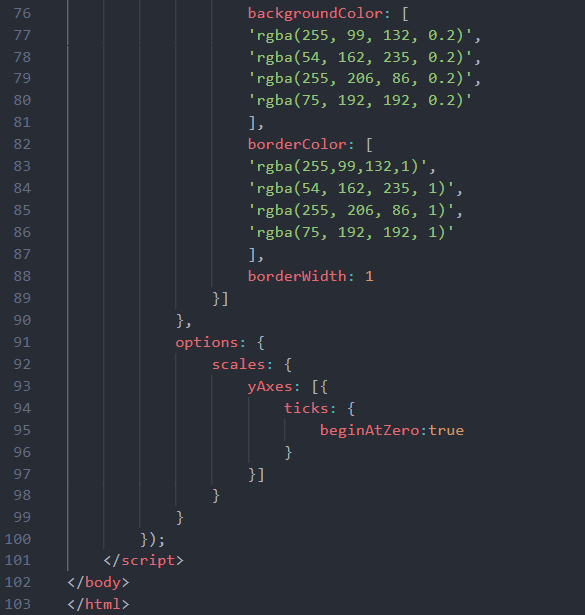
**Gambar 2.3. Index.php(2)**

Pada code diatas terdapat form, yang dimana aksi yang akan dilakukan pada form ini akan menggunakan tambah.php. Kemudian terdapat button dengan nama INSERT. Jadi, ketika button ini diklik, maka aksi pada tambaht.php akan dilakukan. Kemudian, dibuat tabel, yang dimana didalam tabel terdapat kolom No, Nama Mahasiswa, NIM, Fakultas. Kemudian, dibuat pembuka php, dimana didalamnya terdapat sebuah code untuk menampilkan semua isi record pada tabel mahasiswa dan didefinisikan ke dalam variabel $data. Kemudian, terdapat perulangan while untuk menampilkan array pada $data. Yang akan ditampilkan yaitu record data nama, nim, dan fakultas dari tabel mahasiswa.

****

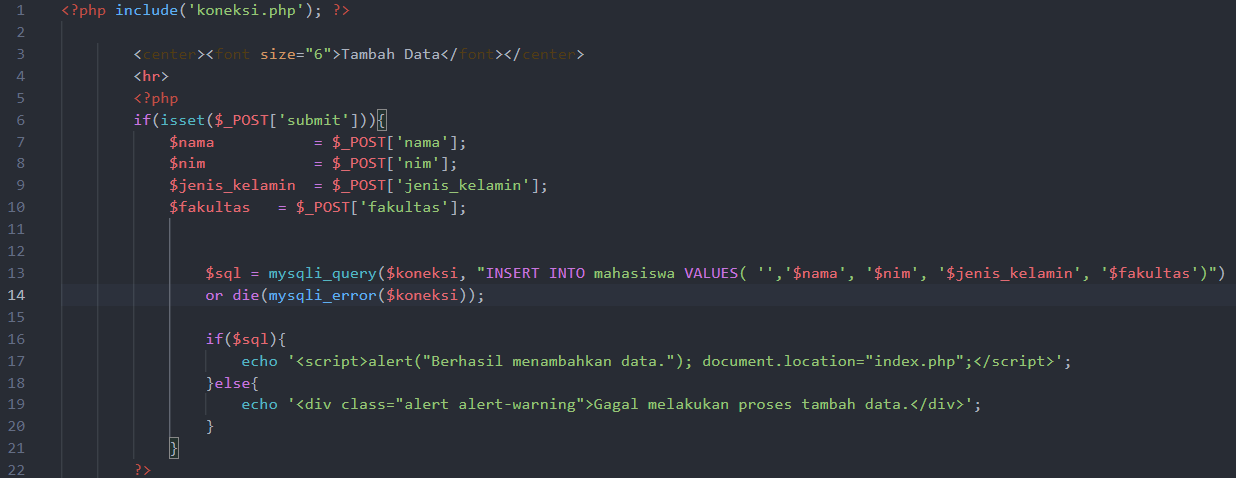
**Gambar 2.4. Index.php(3)**

Pada code diatas, merupakan inti dari pembuatan grafik, dimana file charts.js akan dipanggil untuk membuat grafik berdasarkan jumlah mahasiswa teknik, ekonomi, fisip, dan pertanian. Kemudian, terdapat array data, dimana akan ditampilkan jumlah dari data mahasiswa berdasarkan fakultas nya, yaitu teknik, ekonomi, fisip, dan pertanian.

****

**Gambar 2.5. Index.php(4)**

Pada code diatas, merupakan code untuk membuat background color dan border color pada label Teknik, Fisip, Ekonomi, dan pertanian.

****

**Gambar 2.6. Tambah.php(1)**

Pada code diatas, pertama-tama dipanggil koneksi.php. kemudian, terdapat exception, dimana jika submit ditekan, maka data elemen pada nama akan diambil dan disimpan ke dalam variabel $nama, data elemen pada data nim diambil dan disimpan ke dalam variabel $nim, data elemen pada data jenis\_kelamin diambil dan disimpan ke dalam variabel $jenis\_kelamin, data elemen pada data fakultas diambil dan disimpan ke dalam variabel $fakultas. Kemudian, terdapat fungsi mysqli\_query yang memiliki parameter $koneksi yang merupakan variabel untuk menyimpan hasil eksekusi mysqli\_connect sebelumnya, kemudian dibuat query untuk menginputkan data kevariabel $nama, $nim, $jeniskelamin, $fakultas. Kemudian terdapat kondisi if untuk cek $sql, apakah data berhasil diinput atau tidak. Jika berhasil diinput, maka akan tampil alert Berhasil Menambahkan data, dan page akan langsung kembali ke index.php. jika inputan gagal, maka terdapat alert bahwa Gagal melakukan proses tambah data.

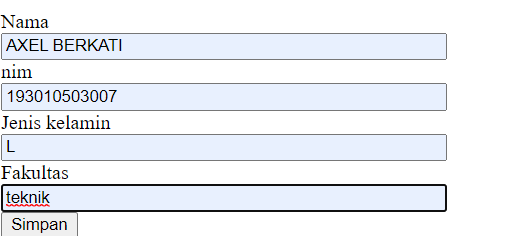
****

**Gambar 2.7. Tambah.php(2)**

Pada code diatas dibuat form, dimana hasil data dalam form ini akan dikirimkan ke kedalam file itu sendiri dengan menggunakan method post. Kemudian didalam form dibuat label untuk menginput pada nama, nim, jeniskelamin, dan fakultas. Kemudian terdapat tombol submit yang diberi nama simpan.

****

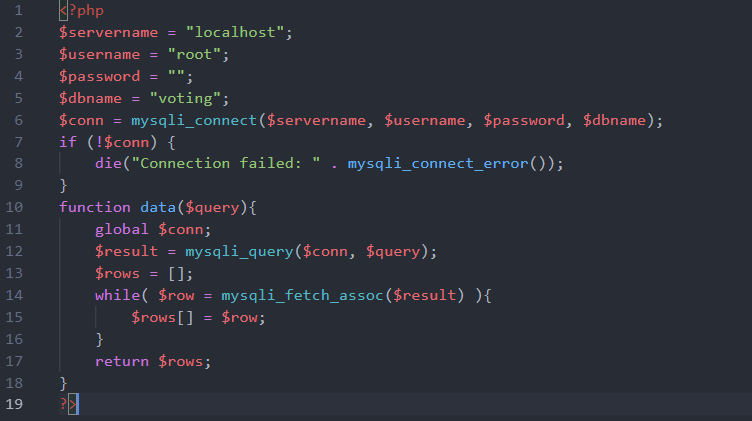
**Gambar 2.8. Tampilan Awal Web**

****

**Gambar 2.9. Tambah Data Mahasiswa**

****

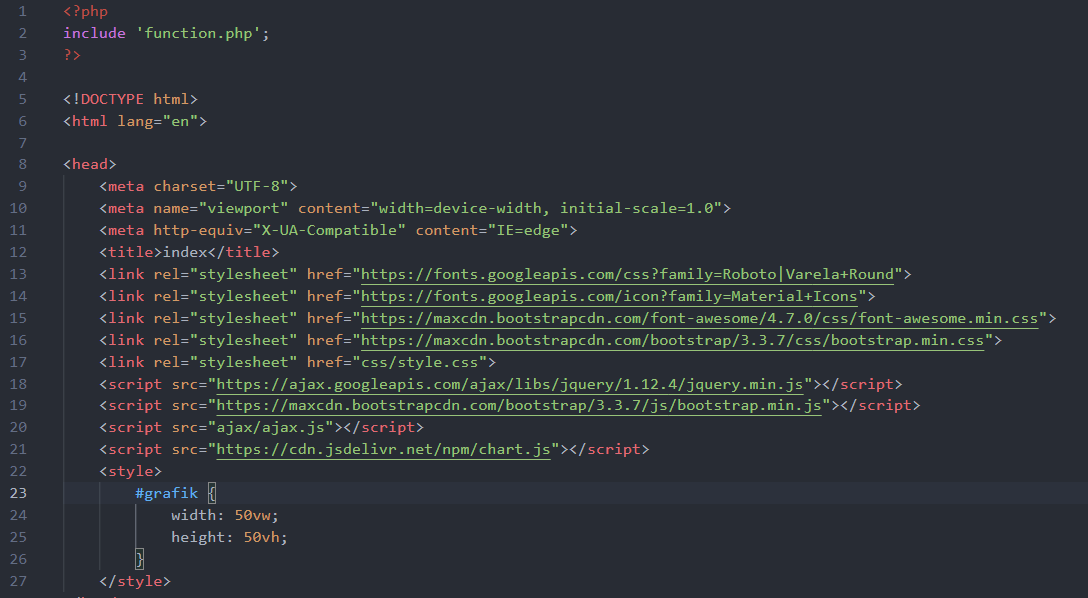
**Gambar 2.10. Tampilan Setelah Data Mahasiswa di Tambah**

****

**Gambar 2.11. Function.php**

Pada code di atas, di bagian awalnya terdapat deklarasi variabel serta value yang ditentukan untuk variabel ini. Biasanya ada empat variabel untuk pembuatan koneksi database: $servername, $database, $username, dan $password. Informasi database ditetapkan sebagai value untuk keempat variabel tersebut sehingga dapat dikirimkan ke fungsi. Pada $servername didefinisikan hostnya (biasanya localhost). Pada $database didefinisikan nama database yang akan digunakan yaitu voting. Pada $username didefinisikan username nya(biasanya root). Pada $password mendefinisikan password database (biasanya di xampp tidak ada passwordnya, jadi tidak usah di isi). Untuk menghubungkan PHP dengan MySQL, digunakan fungsi mysqli\_connect() dengan alamat server sebagai parameter pertama, user database sebagai parameter kedua, password user sebagai parameter ketiga, dan nama database sebagai parameter keempat. Fungsi ini didefinisikan ke variabel $koneksi.

Kemudian, dibuat fungsi dengan nama data yang memiliki parameter $query. Kemudian, karena kita ingin menggunakan variabel $conn yang telah didefinisikan sebelumnya kita tambahkan kata kunci global. Kemudian, terdapat fungsi mysqli\_query yang digunakan unutk mengirimkan perintah SQL ke server MySQL untuk melakukan aktifitas CRUD. Kemudian pada parameternya terdapat $conn yang merupakan variabel yang digunakan untuk menyimpan hasil dari eksekusi, dan $query merupakan perintah SQL untuk melakukan manipulasi data seperti UPDATE, DELETE, INSERT, dan SELECT. Fungsi ini didefinisikan ke $result. Kemudian, Perulangan while akan melakukan perulangan selama kondisi (syarat) terpenuhi, disini menggunakan mysqli\_fetch\_assoc, dimana tipe hasil nya adalah array asosiatif dimana setiap nama kolom dan nilai dari satu baris dikaitkan bersama sebagai nama, pasangan nilai. Selama masih ada baris di tabel tersebut maka akan ditampilkan datanya (kondisi while masih terpenuhi), jika tidak ada baris lagi maka while akan berhenti.

****

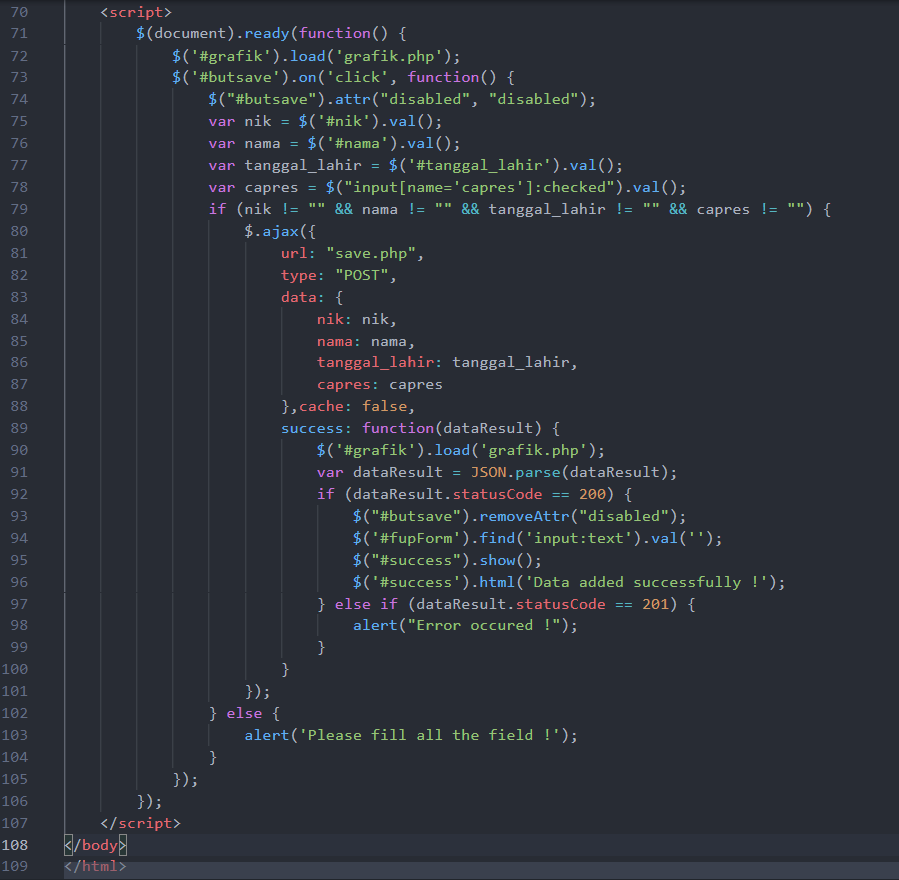
**Gambar 2.12. Index.php(1)**

Pada code diatas, pertama-tama dibuat pembuka php untuk memanggil function.php. kemudian, terdapat struktutr html dimana didalam bagian head nya dibuat penghubung ke bootstrap, css, ajax, dan lain-lain melalui web secara online.

****

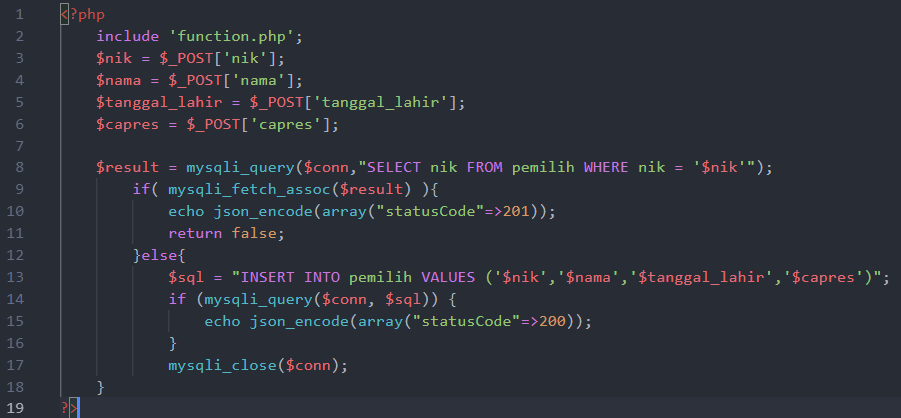
**Gambar 2.13. Index.php(2)**

Pada code diatas, form dengan method post, dimana pada form ini terdapat label untuk menginputkan nik denga type text, nama dengan type text, tanggal\_lahir dengan type date, dan capres dengan type radio, dengan value a,b dan c. Kemudian, terdapat button dengan nama save dengan value vote dan id butsave.

****

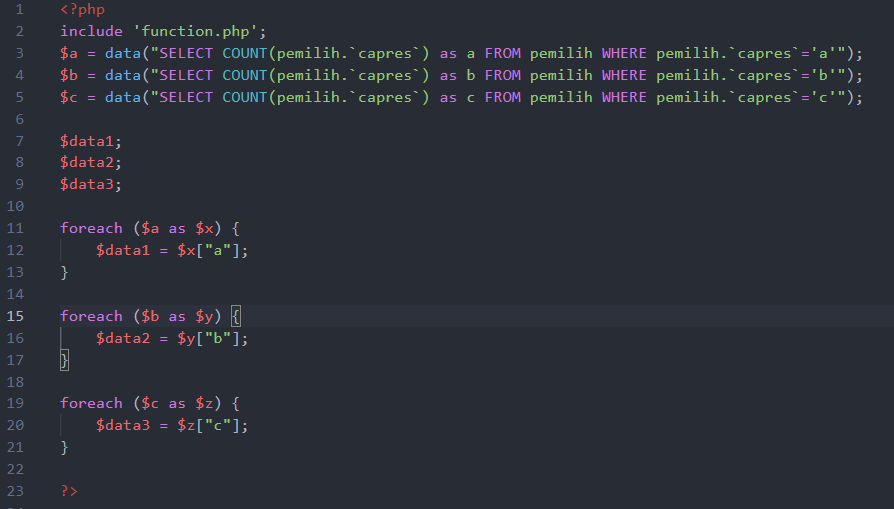
**Gambar 2.14. Index.php(3)**

Pada code diatas, merupakan script ajax, dimana script ini akan memanggil grafik.php. Kemudian, jika id butsave atau submit pada form sebelumnya diklik, maka hasil inputan pada form sebelumnya akan disimpan kedalam variabel nik, nama, tanggal\_lahir, dan capres. Kemudian, terdapat kondisi jika nik, nama, tanggal lahir, dan capres tidak kosong, maka save.php akan dituju dan dipost ke save.php. dimana data nya nik=nik, nama = nama, tanggal\_lahir=tanggal\_lahir, capres= capres. Jika berhasil, maka grafik.php akan diload kembali. Kemudian, terdapat kondisi jika statuscode nya sama dengan 200, maka akan ditampilkan Data added succesfully, tetapi jika status code nya sama dengan 201, maka akan muncul alert error occured. Jika, inputan salah satu pada nik, nama, tanggal\_lahir, dan capres kosong maka akan muncul alert Please fill all the field.

****

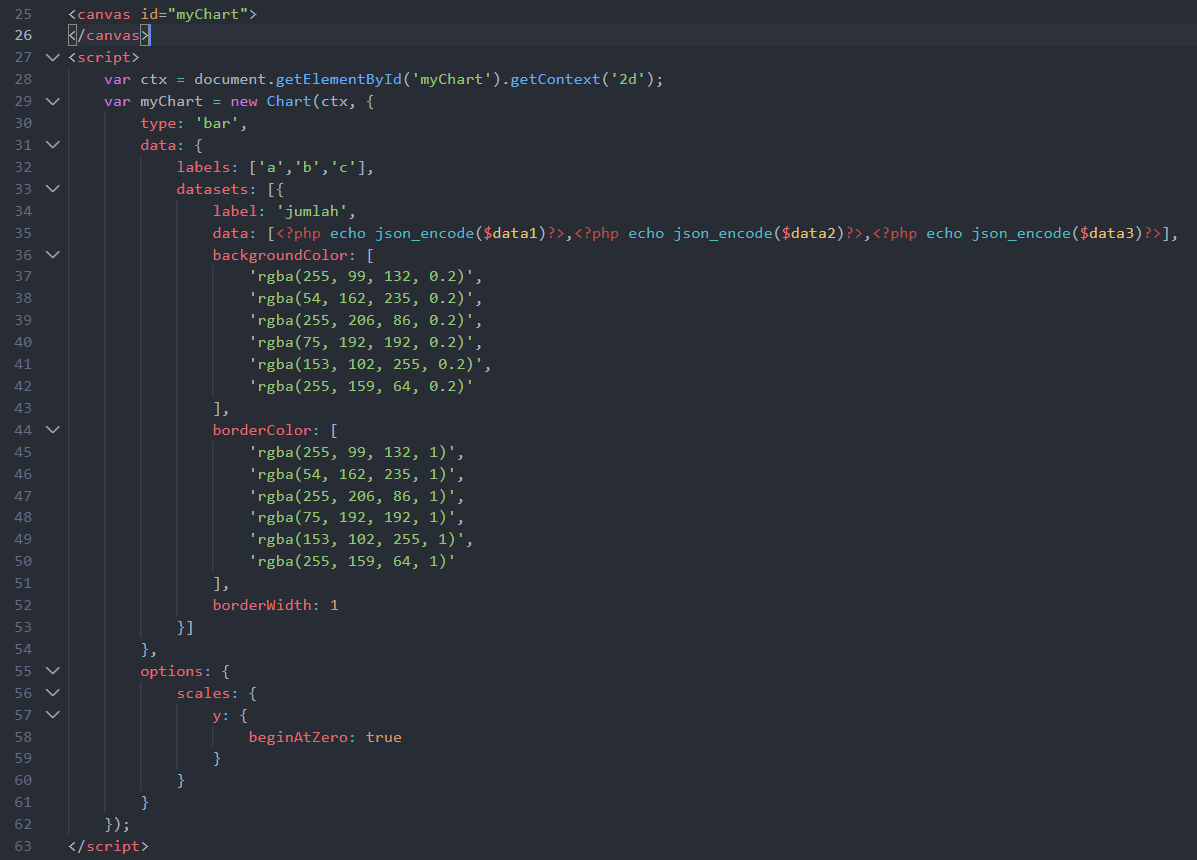
**Gambar 2.15. Save.php**

Pada code diatas, pertama-tama dibuat pembuka php dan function.php dipanggil. Kemudian, terdapat method post untuk menampung data pada nik, didefinisikan kedalam variabel $nik, method post untuk menampung data pada nama, didefinisikan kedalam variabel $nama, method post untuk menampung data pada tanggal\_lahir, didefinisikan kedalam variabel $tanggal\_lahir, method post untuk menampung data pada capres, didefinisikan kedalam variabel $capres. Kemudian, terdapat fungsi mysqli\_query yang memiliki parameter $conn yang merupakan variabel untuk menyimpan hasil eksekusi mysqli\_connect sebelumnya, kemudian dibuat query untuk menampilkan nik dari tabel pemilh dimana nik = $nik dan didefinisikan ke variabel $result. Kemudian, terdapat kondisi jika hasil array sesuai dengan queru pada $result, maka akan ditampilkan json\_encode dengan status code 201, dan akan direturn false. Tetapi jika tidak, akan diinputkan data ke variabel $nik, $nama, $tanggal\_lahir, $capres didefinisikan ke variabel $sql. Kemudian, terdapat kondisi baru, dimana jika query pada $sql berhasil dilakukan, maka akan ditampilkan json\_encode dengan status code 200.

****

**Gambar 2.16. Grafik.php(1)**

Pada gambar diatas, pertama-tama dibuat pembuka php dan dipanggil function.php. Kemudian dipanggil fungsi data yang dimana parameter dari fungsi ini akan menampilkan menghitung capres dengan alias a dari tabel pemilih dimana capres = a, didefinisikan kedalam variabel $a. Kemudian dipanggil fungsi data yang dimana parameter dari fungsi ini akan menampilkan menghitung capres dengan alias b dari tabel pemilih dimana capres = b, didefinisikan kedalam variabel $b. Kemudian dipanggil fungsi data yang dimana parameter dari fungsi ini akan menampilkan menghitung capres dengan alias c dari tabel pemilih dimana capres = c, didefinisikan kedalam variabel $c. Kemudian dideklarasikan variabel $data1, $data2, dan $data3. Kemudian dibuat perulangan foreach, yang akan menampilkan array pada $a dan variabel perantaranya $x. Yang akan ditampilkan yaitu data pada array $x yaitu a, didefinisikan kedalam variabel $data1. Kemudian dibuat perulangan foreach lagi, yang akan menampilkan array pada $b dan variabel perantaranya $y. Yang akan ditampilkan yaitu data pada array $y yaitu b, didefinisikan kedalam variabel $data2. Kemudian dibuat perulangan foreach lagi, yang akan menampilkan array pada $c dan variabel perantaranya $z. Yang akan ditampilkan yaitu data pada array $z yaitu c, didefinisikan kedalam variabel $data3.

****

**Gambar 2.17. Grafik.php(2)**

Pada code diatas, terdapat code untuk membuat element canvas dan diberi nama idnya myChart. Kemduian, file charts.js akan dipanggil untuk membuat grafik berdasarkan jumlah capres yang memilih a, b, dan c. Kemudian, didalam label data, akan diubah format data array pada variabel $data1, $data2, $data3 menjadi JSON menggunakan json\_encode.

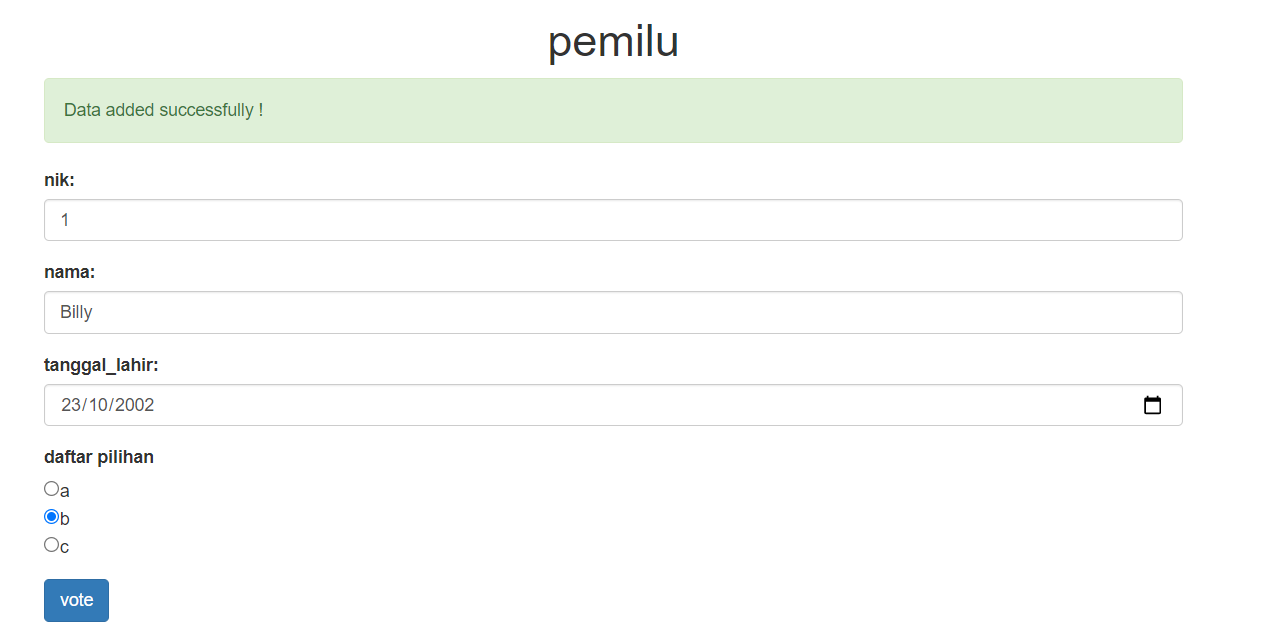
Kemudian terdapat code untuk membuat background color dan border color pada label a, b, dan c.

****

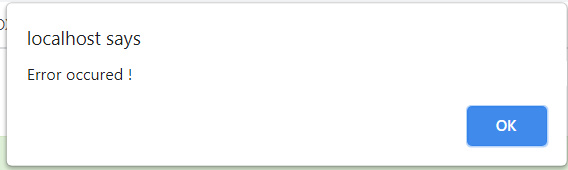
**Gambar 2.18. Polling Pemilu**

****

**Gambar 2.19. Hasil Polling**

****

**Gambar 2.19. Test Polling, Jika Nik Sama Dengan Nik Polling Sebelumnya**

****

**Gambar 2.19. Tampil Alert, Jika Nik Sama**

**BAB III**

**KESIMPULAN**

Dari praktikum yang telah dilaksanakan, dapat disimpulkan bahwa jika dalam konten HTML yang dibuat pada halaman web dilakukan oleh browser menggunakan javascript yang merupakan bahasa pemrograman client-side, maka halaman yang dibuat menggunakan ajax dapat dimuat relatif lebih cepat.

Keuntungan lain menggunakan Ajax adalah pendekatan Ajax membuat programmer untuk memisahkan antara metode dan format yang digunakan untuk penyampaian informasi melalui web. Penyusun halaman web yang dipisahkan antara lain :

1. Data mentah yang akan ditampilkan yang biasanya dibuat dalam format khusus seperti XML atau lainnya.
2. Format atau struktur halaman web biasanya dibuat menggunakan HTML atau XHTML yang kemudian dioleh menggunakan Dynamic Manipulation di DOM.
3. Pada bagian Element style halaman web mendeskripsikan bagaimana tampilan halaman web dari mulai font hingga metode penampilan gambar. Style ini biasanya diembed ke halaman web atau di referensikan melalui file .css.
4. Fungsionalitas halaman web, biasanya terdiri dari Javascript (DHTML), HTTP standar, komunikasi XMLHttp, dan bahasa pemograman server-side.

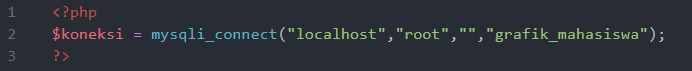
**DAFTAR PUSTAKA**

Sunyoto, A. (2010). AJAX (Asynchronus JavaScript and XML) Andi Sunyoto. *Jurnal Dasi*, *11*(3), 1–11.

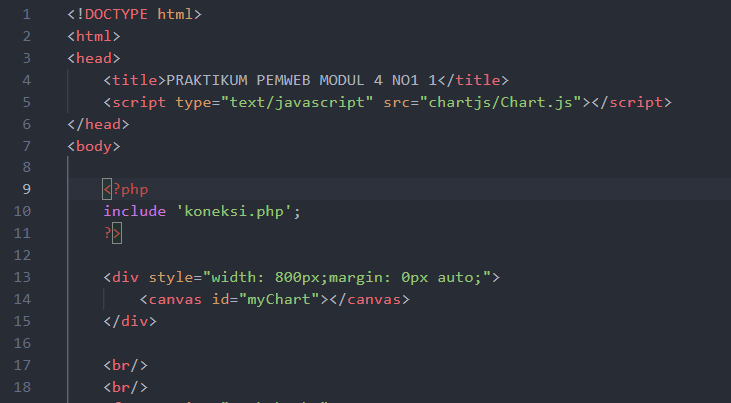
Samsudin, Aris. (2019). *Tutorial PHP Part 55 – Membuat Grafik Dari Database MySQL dan PHP dengan Chart.JS.* Diakses pada tanggal 15 April 2021 di: https://www.warungbelajar.com/tutorial-php-part-55-membuat-grafik-dari-database-mysql-dan-php-dengan-chart-js.html

Lumbung, Gede. (2011). *Tutorial AJAX : Membuat Polling AJAX Sederhana Dengan jQuery dan PHP.* Diakses pada tanggal 15 April 2021 di: https://gedelumbung.com/tutorial-ajax-membuat-polling-ajax-sederhana-dengan-jquery-dan-php/

**LAMPIRAN**

****

**Gambar 2.1. Koneksi.php**

****

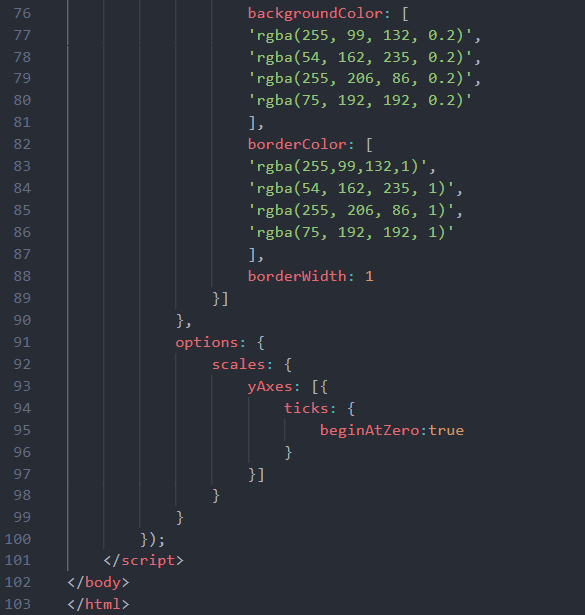
**Gambar 2.2. Index.php(1)**

****

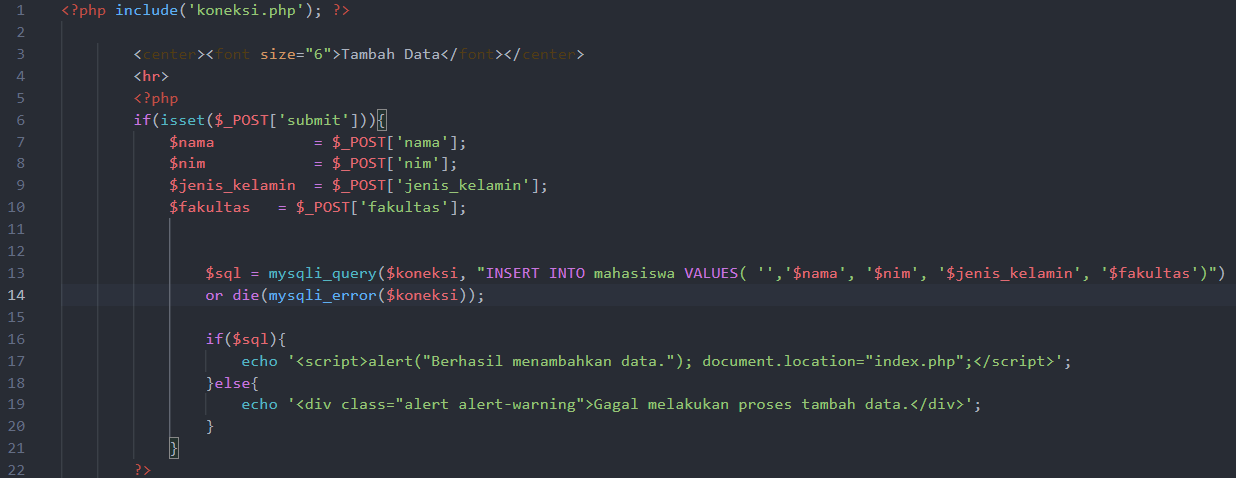
**Gambar 2.3. Index.php(2)**

****

**Gambar 2.4. Index.php(3)**

****

**Gambar 2.5. Index.php(4)**

****

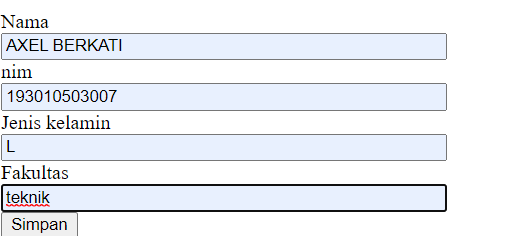
**Gambar 2.6. Tambah.php(1)**

****

**Gambar 2.7. Tambah.php(2)**

****

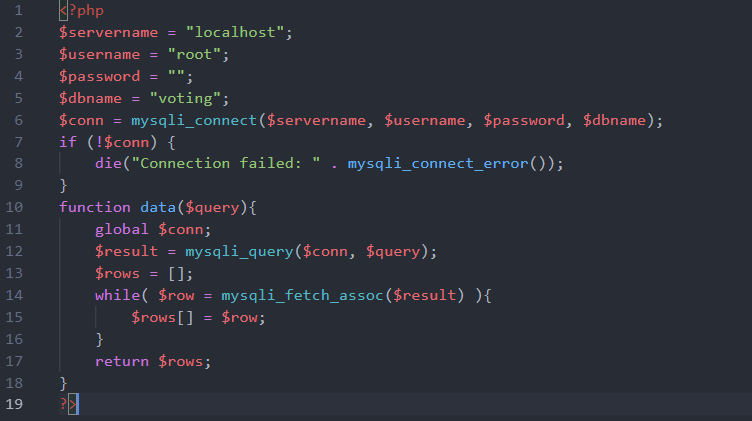
**Gambar 2.8. Tampilan Awal Web**

****

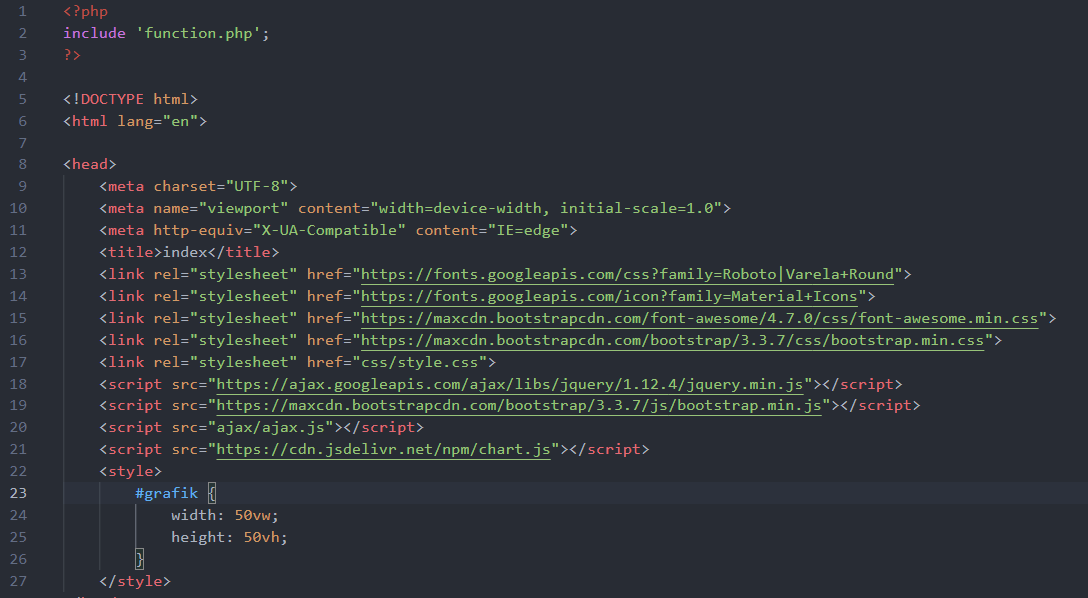
**Gambar 2.9. Tambah Data Mahasiswa**

****

**Gambar 2.10. Tampilan Setelah Data Mahasiswa di Tambah**

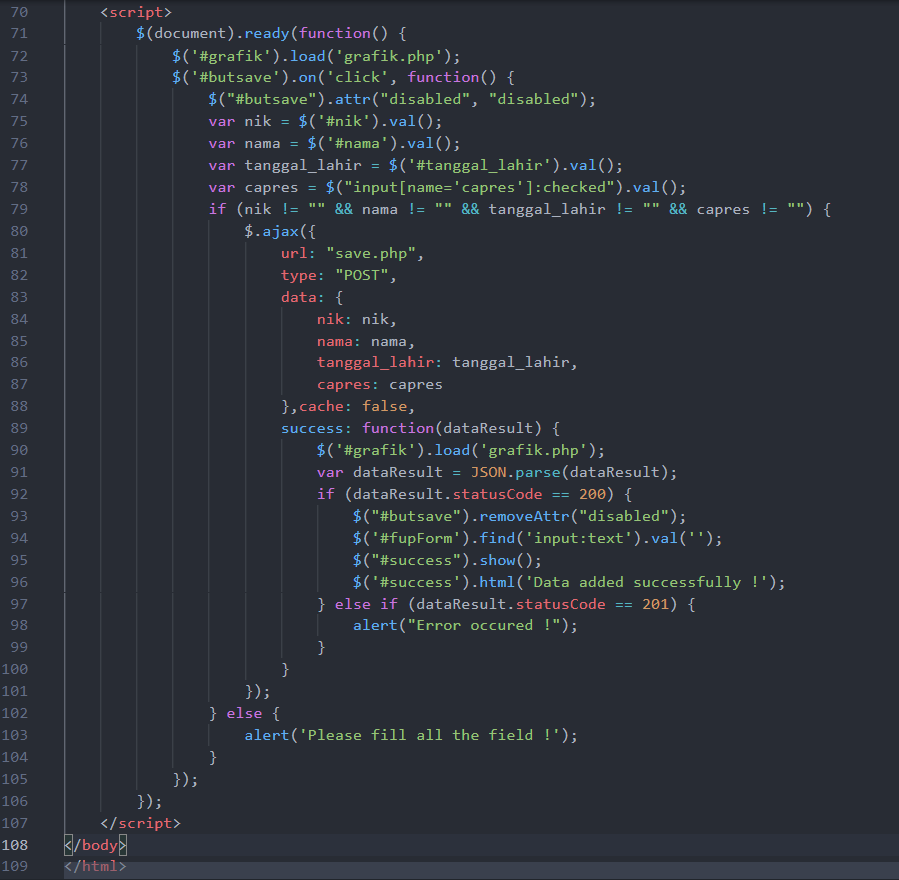
****

**Gambar 2.11. Function.php**

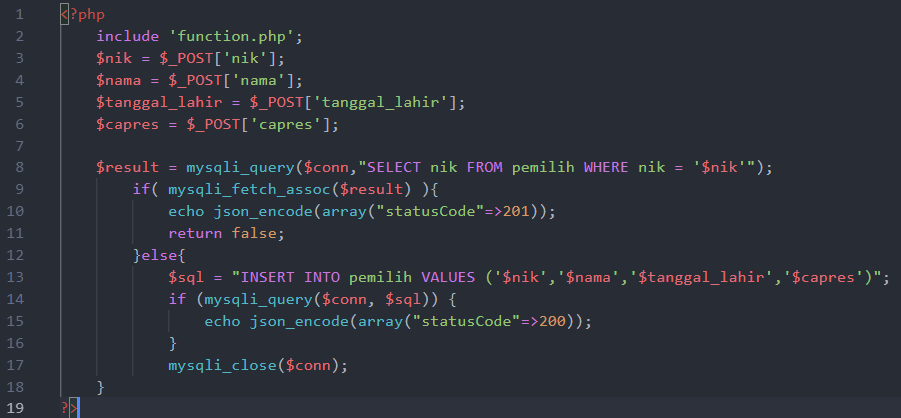
****

**Gambar 2.12. Index.php(1)**

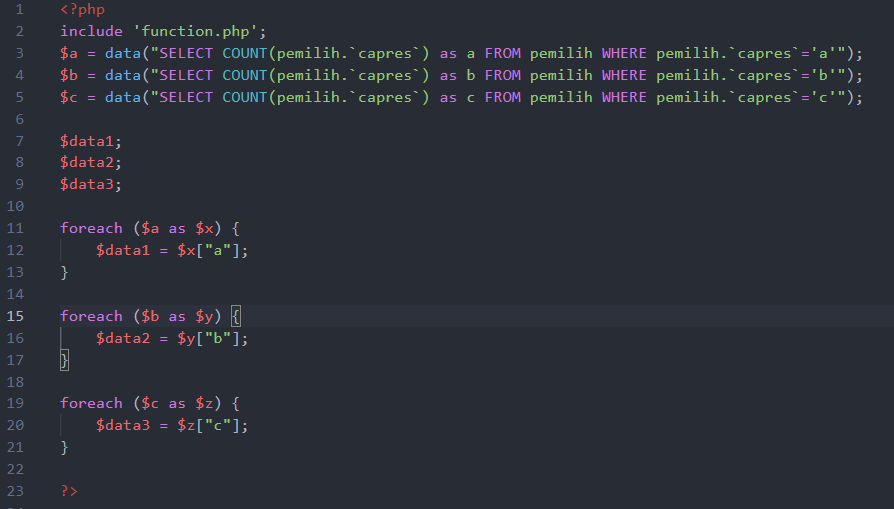
**Gambar 2.13. Index.php(2)**

****

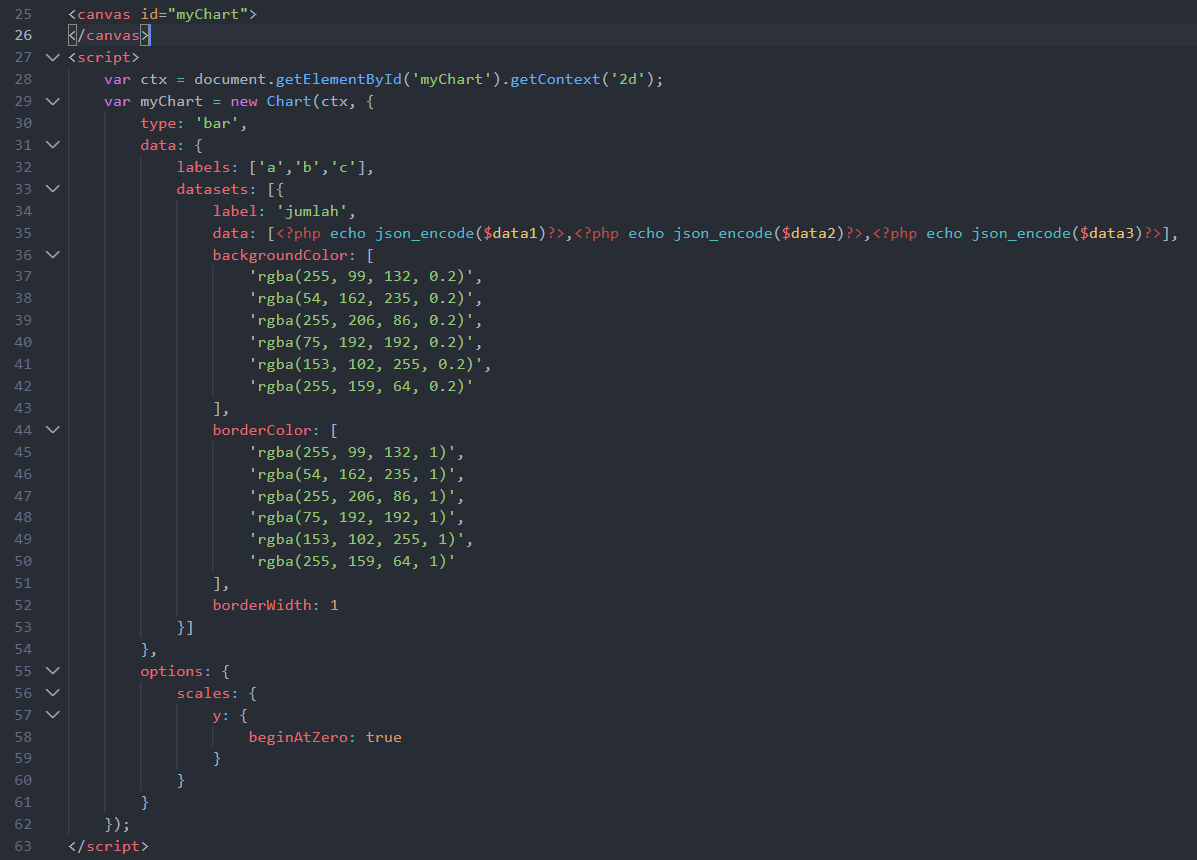
**Gambar 2.14. Index.php(3)**

****

**Gambar 2.15. Save.php**

****

**Gambar 2.16. Grafik.php(1)**

****

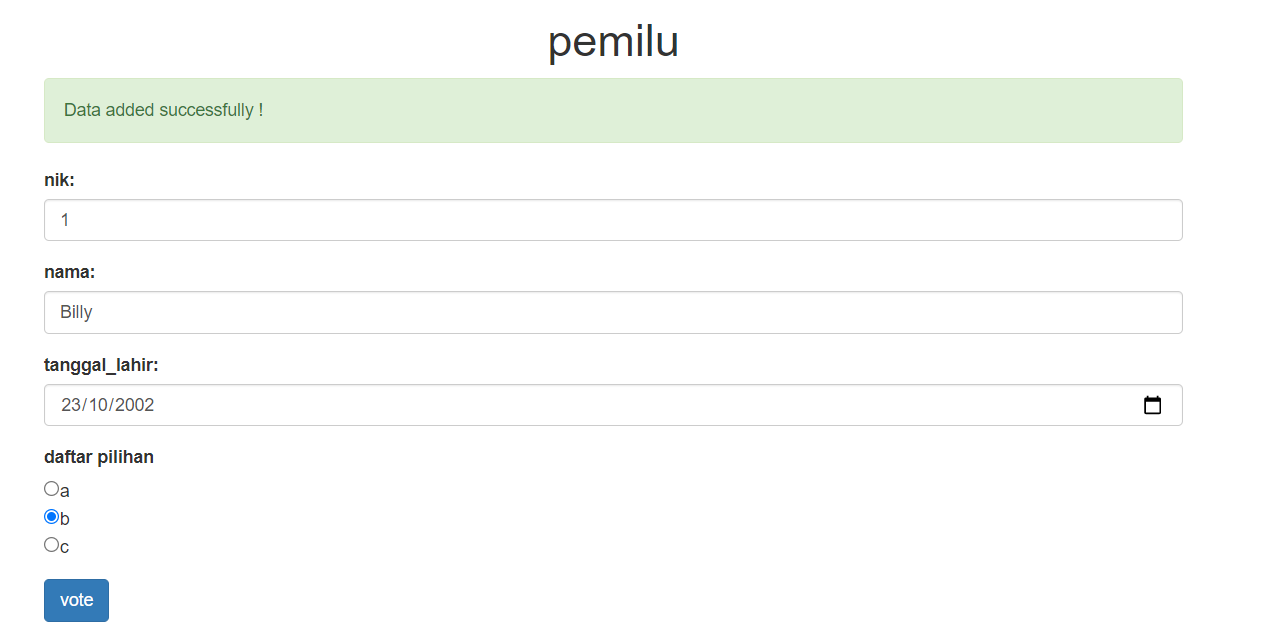
**Gambar 2.17. Grafik.php(2)**

****

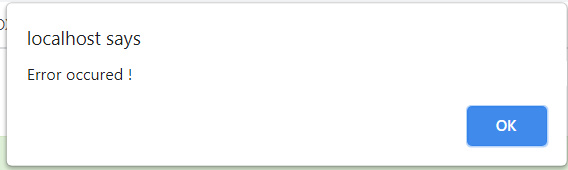
**Gambar 2.18. Polling Pemilu**

****

**Gambar 2.19. Hasil Polling**

****

**Gambar 2.19. Test Polling, Jika Nik Sama Dengan Nik Polling Sebelumnya**

****

**Gambar 2.19. Tampil Alert, Jika Nik Sama**