**PENJELASAN PROGRAM**

17

**BASIS DATA NON RELASIONAL**

**MongoDB dan PHP**



**Disusun oleh:**

**Nama : Bayu Murti Setiawan**

**Nim : 18 311 0 019**

**Jurusan : Manajemen Informatika**

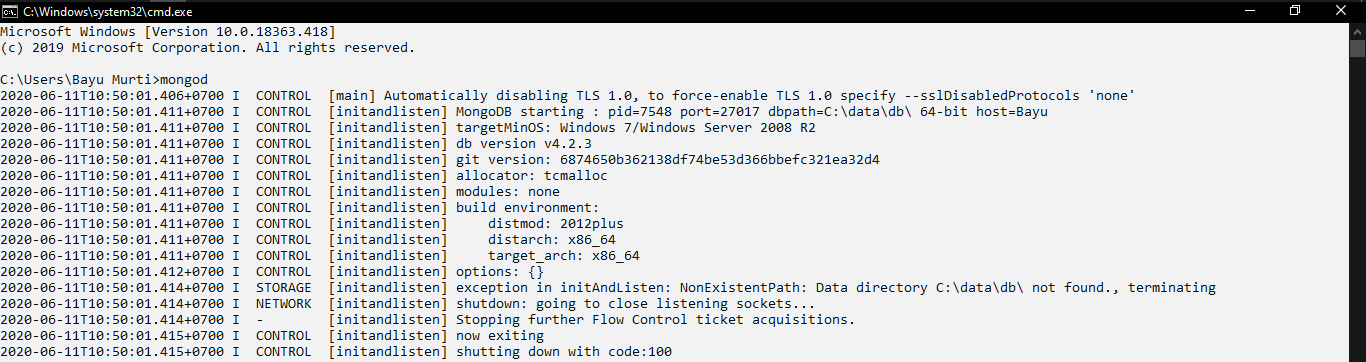
**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**AKAKOM YOGYAKARTA**

**2020**

**Membuat** **Aplikasi Interaktif dengan MongoDB dan PHP**

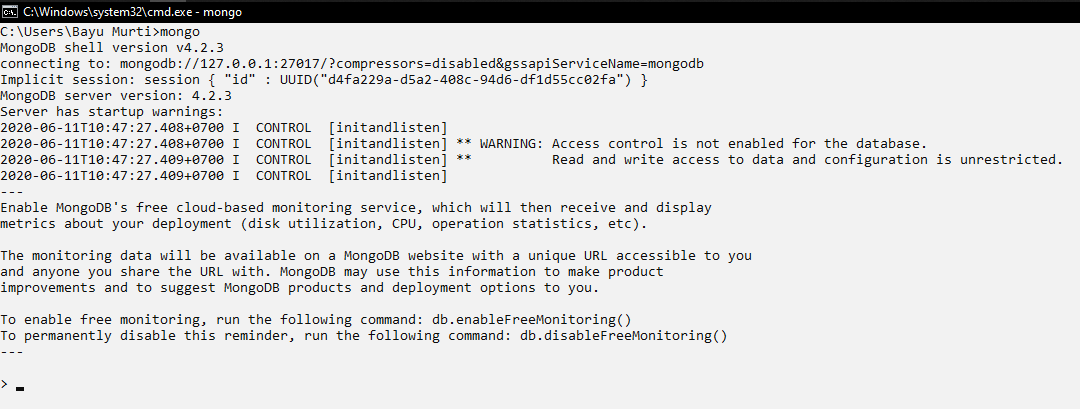
1. **Alat atau Software yang digunakan:**
2. Windows 10 Pro 64-bit
3. Command Prompt atau cmd
4. MongoDB 64-bit versi 4.2.3
5. File Explorer
6. XAMPP Control Panel versi 3.2.4
7. PHP versi 7.4.3
8. Browser (Google Chrome)
9. **Proses**
10. Membuka MongoDB Server melalui cmd dengan perintah mongod



**Penjelasan:**

Pada tampilan diatas merupakan perintah untuk memulai menjalankan MongoDB, pertama kali yang harus dilakukan adalah membuka MongoDB Server dengan perintah mongod, jika perintah mongod dijalankan maka akan muncul tamplilan diatas, dan jika MongoDB server tidak dijalankan maka MongoDB Client tidak dapat melakukan koneksi ke MongoDB Server.

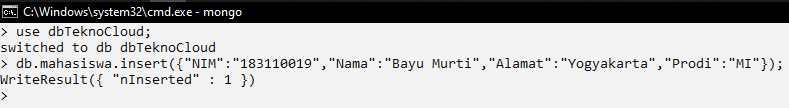
1. Membuka cmd baru dan menjalankan MongoDB Client dengan perintah mongo



**Penjelasan:**

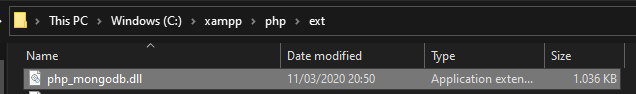
Pada tampilan diatas merupakan tampilan dari MongoDB Client, untuk menjalankan MongoDB Client dapat membuka jendela command prompt baru dan mengetikkan perintah mongo, lalu akan muncul bebrapa keterangan seperti diatas.

1. Membuat database, collection serta contoh isi key dan field dalam MongoDB Client

** Penjelasan:**

Pada tampilan diatas merupakan tampilan syntax MongoDB, syntax yang digunakan antara lain use dbTeknoCloud yang berarti membuat database dengan nama dbTeknoCloud lalu dilanjutkan db.mahasiswa.insert({"NIM":"183110019", "Nama":"Bayu Murti", "Alamat":"Yogyakarta", "Prodi":"MI"}); yang berarti membuat collection / table dengan nama mahasiswa dan diberikan isi dengan format key dan value antara lain key / field NIM dengan value 183110019, key / field Nama dengan value Bayu Murti, key / field Alamat dengan value Yogyakarta, dan key / field Prodi dengan value MI.

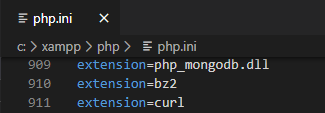
1. Menambahkan driver php\_mongodb.dll kedalam folder ext



**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari folder ext, didalam tersebut berisikan configurasi dari program PHP dengan format file dll, didalam folder tersebut ditambahkan file driver php\_mongodb.dll yang bisa di download sesuai versi php yang dipakai, kali ini menggunakan php versi 7.4.3 jadi bisa mendownload driver MongoDB yang versi 1.7.4 di web <https://pecl.php.net/package/mongodb/1.7.4/windows>

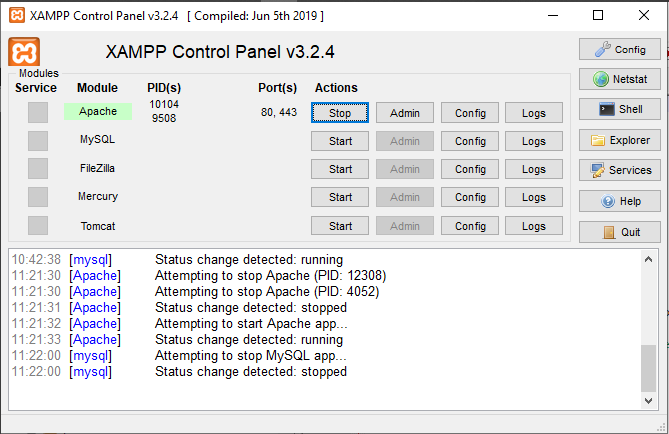
1. Melakukan instalisasi driver php\_mongodb.dll didalam file php.ini



**Penjelasan:**

Pada tampilan diatas merupakan tampilan file php.ini, fie ini digunakan untuk menginstall drive yang telah di download dan di masukkan ke folder ext, cara melakukan instalisasi yaitu dengan menambahkan script extension=php\_mongodb.dll didalam file php.ini.

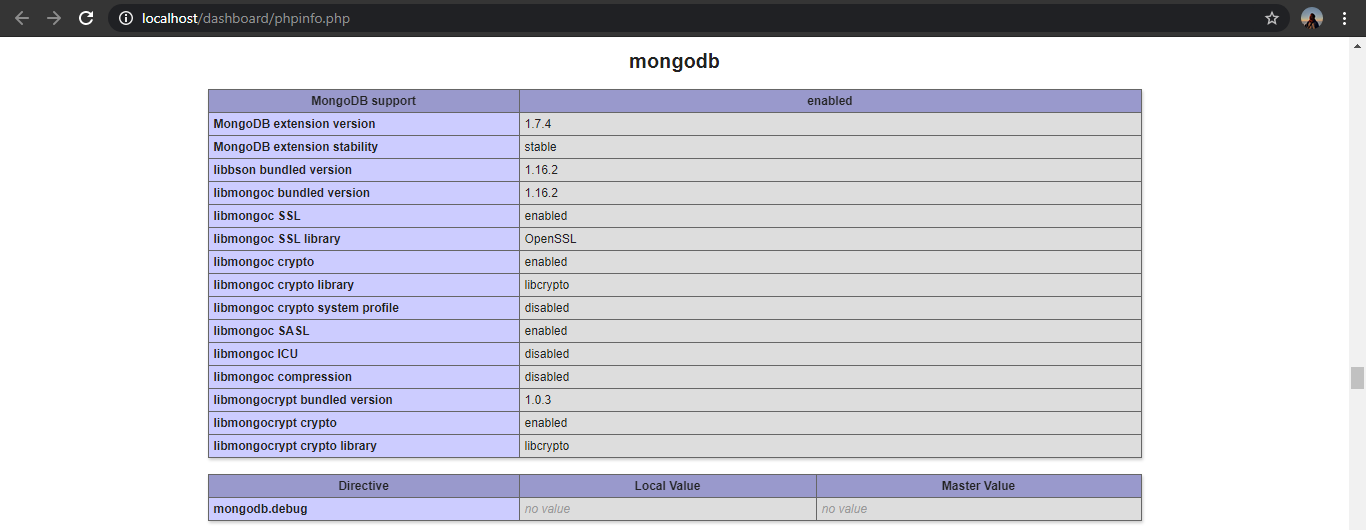
1. Menghidupkan xampp control panel untuk Apache



**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari XAMPP Control Panel, software ini digunakan untuk mengaktifkan module apache yang berfungsi untuk menjalankan localhost dan juga digunakan untuk menjalankan file php di server local.

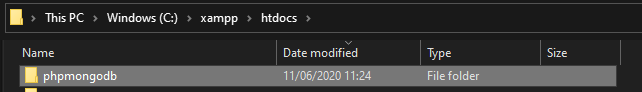
1. Memastikan driver MongoDB



**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file phpinfo.php yang berfungsi untuk memastikan driver MongoDB sudah terinstal dan bisa mengecek di <http://localhost/dashboard/phpinfo.php> dan cari mongodb pada halaman ini, jika sudah ada maka mongodb, berarti driver mongodb sudah terintsall pada php.

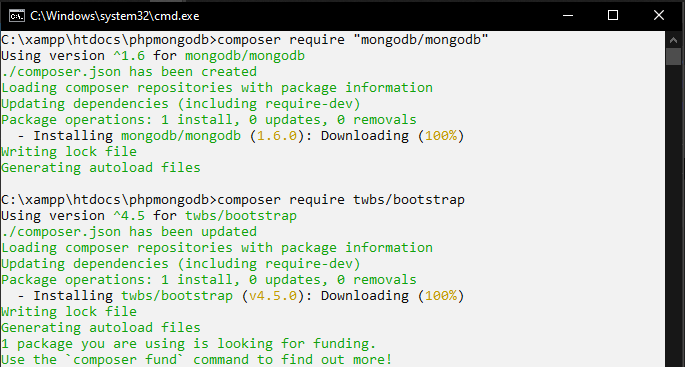
1. Membuat folder phpmongodb didalam htdocs



**Penjelasan:**

Pada tampilan diatas merupakan tampilan didalam folder htdocs, didalam htdocs ini dibuatkan foder phpmongodb yang digunakan untuk menjalankan file – file yang nantinya akan digunakan.

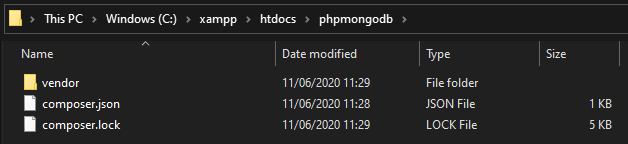
1. Install library mongodb/mongodb dan twbs/boostrap didalam folder phpmongodb



**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari command prompt yang menjalankan syntax composer require “mongodb/mongodb” yang berfungsi menginstal library mongodb yang nantinya akan digunakan dalam membuat koneksi dengen database dan melakukan operasi crud pada database mongodb didalam file php, selanjutnya composer require twbs/bootstrap yang berfungsi untuk mengintsal library bootstrap yang nantinya akan mempermudah dalam mengatur layout yang akan ditampilkan dihalaman web.

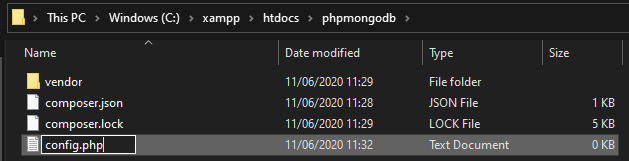
Isi dari folder phpmongodb setelah selesai install library mongodb/mongodb dan twbs/boostrap



**Penjelasan:**

Pada tampilan diatas merupakan tampilan jika kedua library telah berhasil diinstall.

1. Membuat file config.php



config.php

<?php

    require\_once \_\_DIR\_\_ . "/vendor/autoload.php";

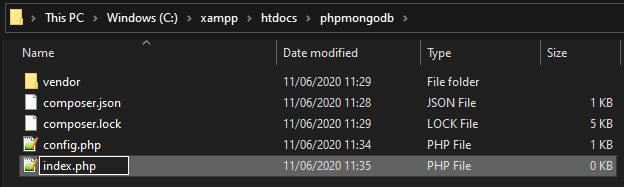
    $collection = (new MongoDB\Client)->dbTeknoCloud->mahasiswa;

?>

**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file config.php, file config.php berfungsi untuk melakukan koneksi ke server mongodb melalui file php, file config ini akan terus dipakai dalam program berikut – berikutnya.

1. Membuat file index.php



index.php

<?php session\_start(); ?>

<!DOCTYPE html>

<html>

    <head>

        <title>Mongo DB</title>

        <link rel="stylesheet" href="./vendor/twbs/bootstrap/dist/css/bootstrap.min.css">

    </head>

    <body>

        <div class="container">

            <br>

            <CENTER><h1>Data Mahasiswa</h1></CENTER>

            <a href="create.php" class="btn btn-success">Tambah Mahasiswa</a>

            <?php

                if (isset($\_SESSION['success'])) {

                    echo "<div class='alert alert-success'>".$\_SESSION['success']."</div>";

                }

            ?>

            <table class="table">

                <thead class="thead-dark">

                    <tr>

                        <th scope="col">NIM</th>

                        <th scope="col">Nama</th>

                        <th scope="col">Alamat</th>

                        <th scope="col">Prodi</th>

                        <th scope="col">Aksi</th>

                    </tr>

                </thead>

                <?php

                  require 'config.php';

                  $Mahasiswa = $collection->find();

                  foreach ($Mahasiswa as $mhs) {

                   echo "<tr>";

                   echo "<th scope='row'>".$mhs->NIM."</th>";

                   echo "<td>".$mhs->Nama."</td>";

                   echo "<td>".$mhs->Alamat."</td>";

                   echo "<td>".$mhs->Prodi."</td>";

                   echo "<td>";

                   echo "<a href = 'edit.php?id=".$mhs->\_id."'class='btn btn-primary'>EDIT</a>";

                   echo "<a href = 'hapus.php?id=".$mhs->\_id."'class='btn btn-danger'>HAPUS</a>";

                   echo "</td>";

                   echo "</tr>";

                  }

                ?>

            </table>

        </div>

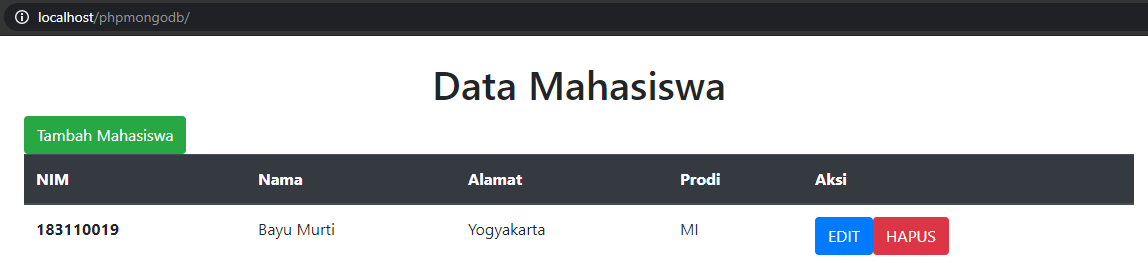
    </body>

</html>

**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file index.php, file index php merupakan file yang pertama kali dijalankan dalam server local, didalam file ini terdapat beberapa perintah dasar php yang gunakan operasi dasar database dan beberapa tag html dasar yang digunakan untuk mengatur layout tampilan.

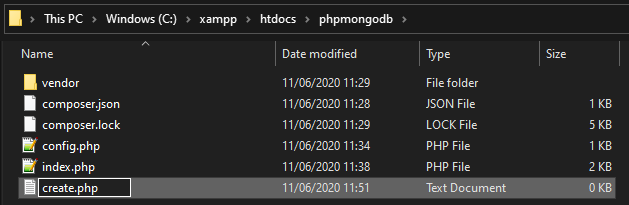
1. Tampilan File index.php



**Penjelasan:**

Pada tampilan diatas merupakan tampilan awal dari server local ketika dijalankan, jika server local tersebut dijalankan akan secara otomatis menampilkan isi dari file index.php, isi dari tampilan diatas antara lain judul, tombol tambah mahasiswa, tombol edit data dan tombol hapus data.

1. Membuat file create.php



create.php

<?php session\_start();

   if(isset($\_POST['submit'])){

      require 'config.php';

      $insertOneResult = $collection->insertOne([

          'NIM' => $\_POST['NIM'],

          'Nama' => $\_POST['Nama'],

          'Alamat' => $\_POST['Alamat'],

          'Prodi' => $\_POST['Prodi'],

      ]);

      $\_SESSION['success'] = "Data Mahasiswa Berhasil di tambahkan";

      header("Location: index.php");

   }

?>

<!DOCTYPE html>

<html>

   <head>

      <title>MongoDB</title>

      <link rel="stylesheet" href="./vendor/twbs/bootstrap/dist/css/bootstrap.min.css">

   </head>

   <body>

      <div class="container">

         <br>

         <CENTER><h1>Tambah Data Mahasiswa</h1></CENTER>

         <form method="POST">

            <div class="form-group">

                <label for="NIM"><strong>NIM:</strong></label>

                <input type="text" class="form-control" name="NIM" required="" placeholder="NIM">

                <label for="Nama"><strong>Nama:</strong></label>

                <input type="text" class="form-control" name="Nama" placeholder="Nama Lengkap">

                <label for="Alamat"><strong>Alamat:</strong></label>

                <input type="text" class="form-control" name="Alamat" placeholder="Alamat">

                <label for="Prodi"><strong>Prodi:</strong></label>

                <select name="Prodi" id="Prodi" class="form-control">

                    <option value="MI">Manajemen Informatika</option>

                    <option value="KA">Komputerisasi Akuntasi</option>

                    <option value="TK">Teknik Komputer</option>

                    <option value="TI">Teknik Informatika</option>

                    <option value="SI">Sistem Informasi</option>

                </select>

                <br>

                <button type="submit" name="submit" class="btn btn-success">Tambah</button>

                <a href="index.php" class="btn btn-primary">Kembali</a>

            </div>

         </form>

      </div>

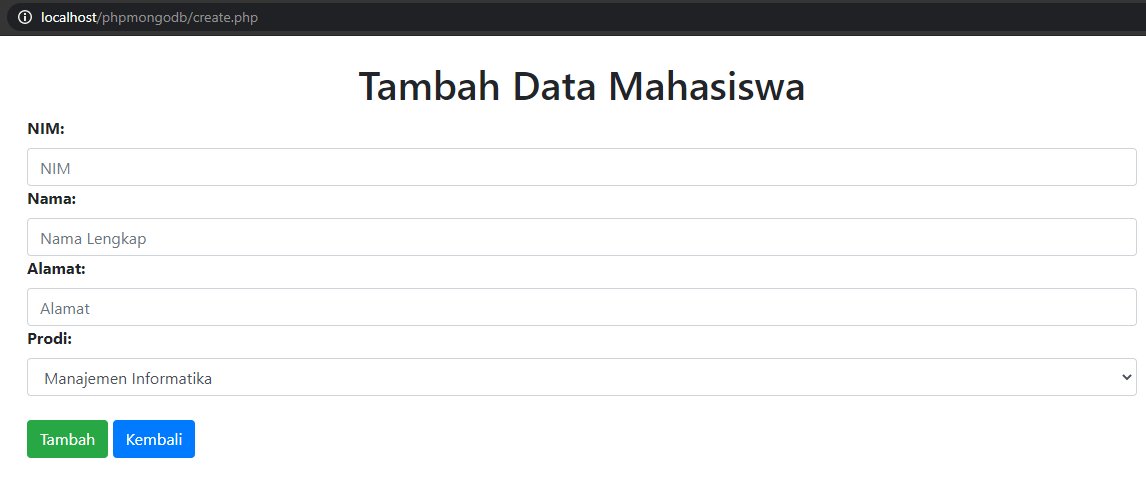
   </body>

</html>

**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file create.php, file create php merupakan file yang digunakan untuk menambah data mahasiswa, file ini akan berjalan jika tombol tambah mahasiswa didalam file index.php di klik, file ini berisi operasi dasar database yaitu menambahkan record melalui form yang telah dibuat dengan method post, dalam menambahkan record, hanya bisa satu persatu data yang ditambahkan karena menggunakan **InsertOne**.

1. Tampilan file create.php

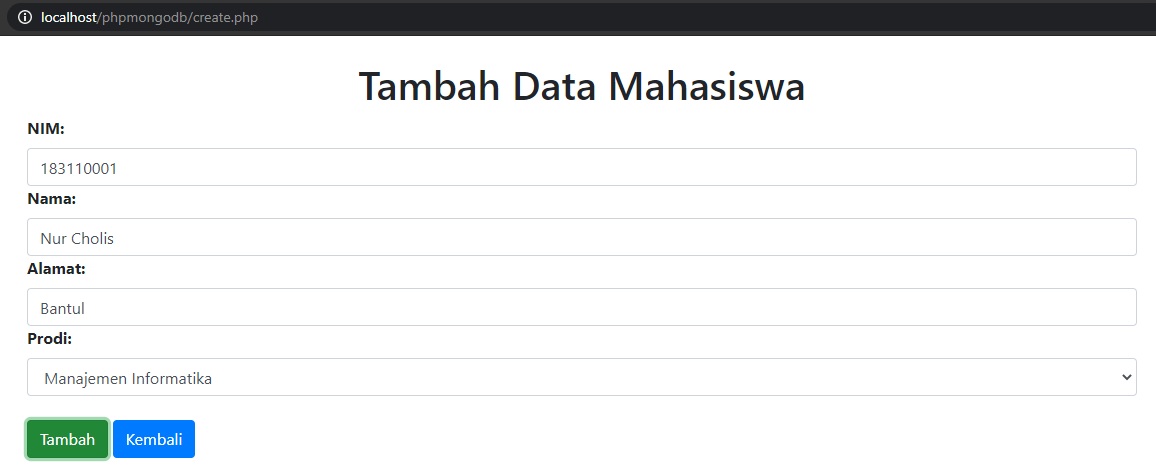


**Penjelasan:**

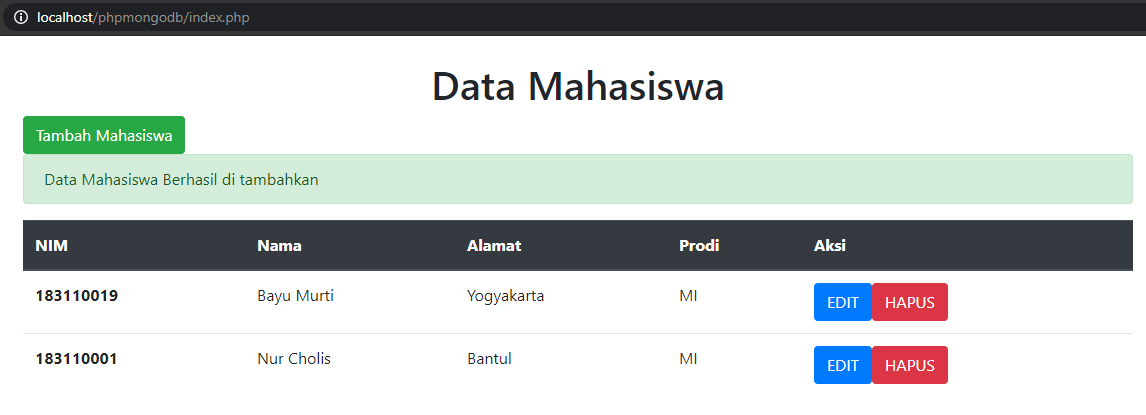
Pada tampilan diatas merupakan tampilan dari file create.php dijalankan, isi dari tampilan diatas antara lain judul, form data yang akan dimasukkan berupa NIM, Nama, Alamat, dan Prodi dengan metode memilih prodi yang disediakan, tombol tambah yang berguna untuk menambahkan data ke collection, dan tombol kembali untuk kembali ke file index.php

.

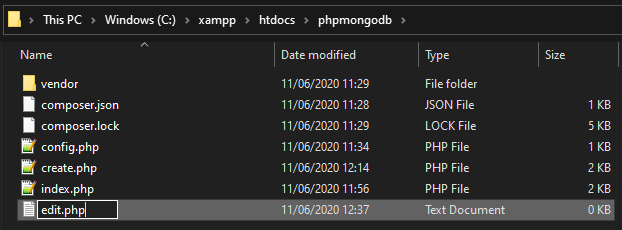
1. Menambahkan data melalui file create.php



Setelah data berhasil ditambahkan



1. Membuat file edit.php



edit.php

<?php session\_start();

   require 'config.php';

   if (isset($\_GET['id'])) {

      $mhs = $collection->findOne(['\_id' => new MongoDB\BSON\ObjectID($\_GET['id'])]);

   }

   if(isset($\_POST['submit'])){

      $collection->updateOne(

          ['\_id' => new MongoDB\BSON\ObjectID($\_GET['id'])],

          ['$set' => ['NIM' => $\_POST['NIM'], 'Nama' => $\_POST['Nama'],

                      'Alamat' => $\_POST['Alamat'],'Prodi' => $\_POST['Prodi'],

          ]]

      );

      $\_SESSION['success'] = "Data Mahasiswa berhasil diubah";

      header("Location: index.php");

   }

?>

<!DOCTYPE html>

<html>

   <head>

      <title>MongoDB</title>

      <link rel="stylesheet" href="./vendor/twbs/bootstrap/dist/css/bootstrap.min.css">

   </head>

   <body>

      <div class="container">

         <br>

         <CENTER><h1>Tambah Data Mahasiswa</h1></CENTER>

         <form method="POST">

            <div class="form-group">

                <label for="NIM"><strong>NIM:</strong></label>

                <input type="text" class="form-control"

name="NIM" required="" value="<?php echo "$mhs->NIM"; ?>">

                <label for="Nama"><strong>Nama:</strong></label>

                <input type="text" class="form-control" name="Nama" placeholder="Nama Lengkap">

                <label for="Alamat"><strong>Alamat:</strong></label>

                <input type="text" class="form-control" name="Alamat" placeholder="Alamat">

                <label for="Prodi"><strong>Prodi:</strong></label>

                <select name="Prodi" id="Prodi" class="form-control">

                    <option value="MI">Manajemen Informatika</option>

                    <option value="KA">Komputerisasi Akuntasi</option>

                    <option value="TK">Teknik Komputer</option>

                    <option value="TI">Teknik Informatika</option>

                    <option value="SI">Sistem Informasi</option>

                </select>

                <br>

                <button type="submit" name="submit" class="btn btn-success">Ubah</button>

                <a href="index.php" class="btn btn-primary">Kembali</a>

            </div>

         </form>

      </div>

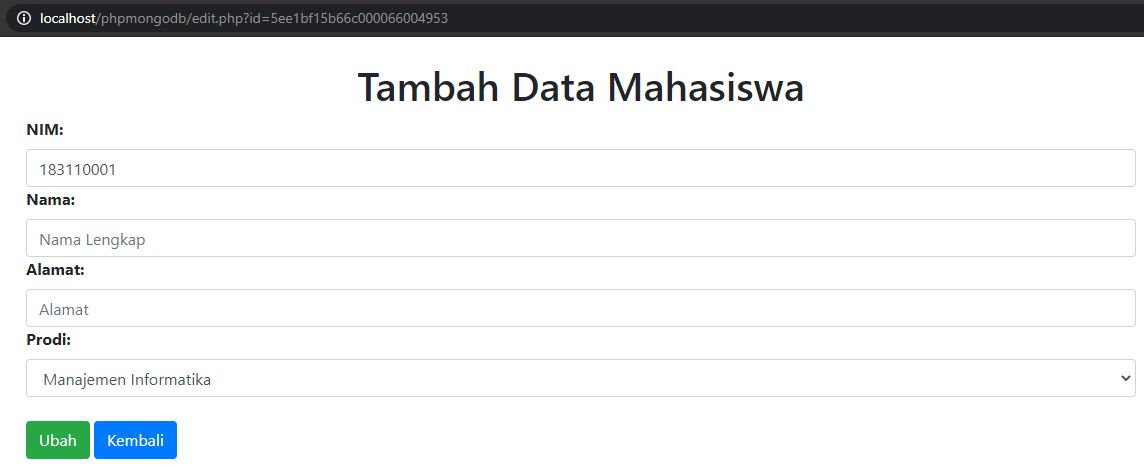
   </body>

</html>

**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file edit.php, file edit php merupakan file yang digunakan untuk mengubah data mahasiswa, file ini akan berjalan jika tombol edit didalam file index.php di klik, file ini berisi operasi dasar database yaitu mengubah record melalui form yang telah dibuat dengan method post, dalam mengubah record perlu dilakukan pencarian data terlebih dahulu, dan mengubah data hanya bisa satu persatu data yang diubah karena menggunakan **findOne** untuk mencari data dan **updateOne** untuk mengubah data.

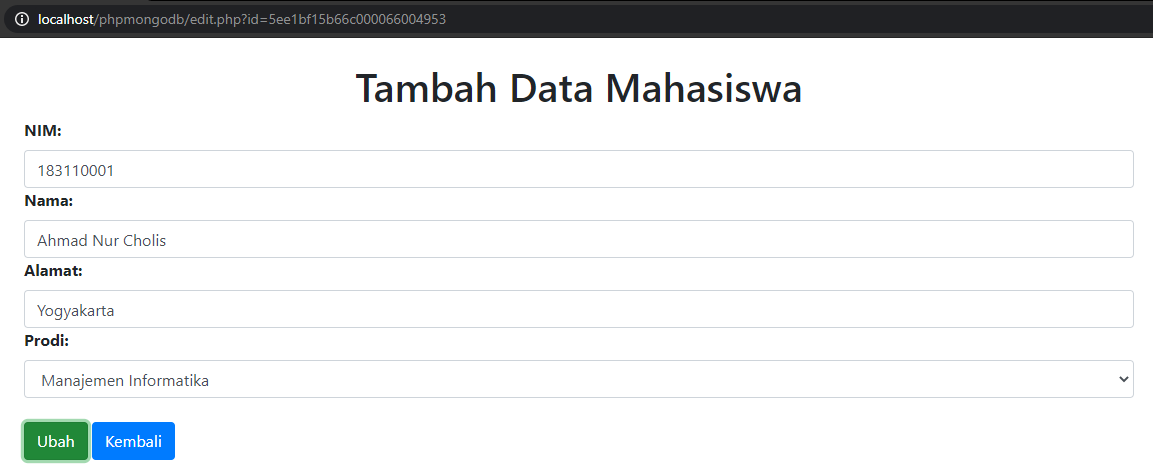
1. Tampilan file edit.php



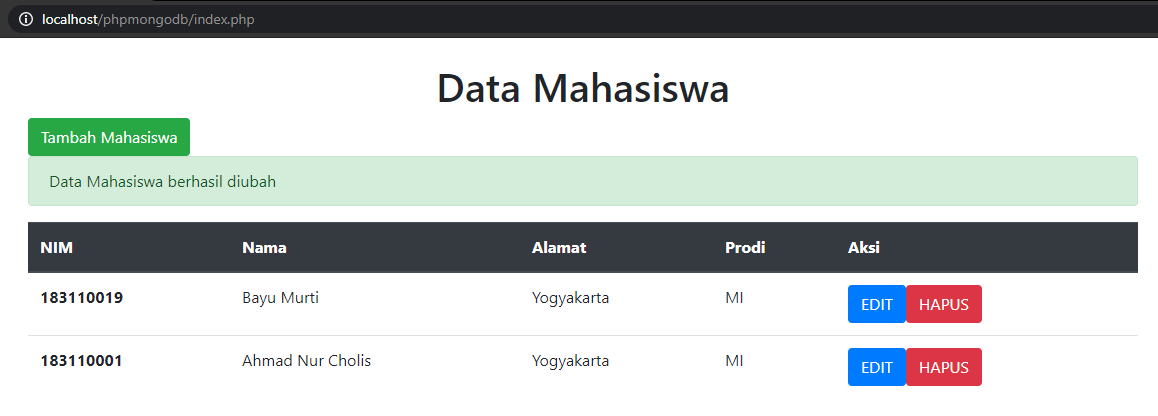
**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file edit.php dijalankan, isi dari tampilan diatas antara lain judul, form data yang akan menampilkan NIM dari data yang lama lalu NIM tersebut bisa diubah, dan dimasukkan data baru berupa NIM, Nama, Alamat, dan Prodi dengan metode memilih prodi yang disediakan, tombol tambah yang berguna untuk menambahkan data ke collection, dan tombol kembali untuk kembali ke file index.php

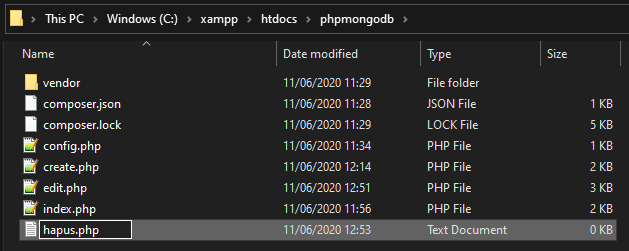
1. Mengedit data mahasiswa melalui file edit.php



Setelah data berhasil diubah



1. Membuat file hapus.php



hapus.php

<?php session\_start();

   require 'config.php';

   if (isset($\_GET['id'])) {

      $mhs = $collection->findOne(['\_id' => new MongoDB\BSON\ObjectID($\_GET['id'])]);

   }

   if(isset($\_POST['submit'])){

         require 'config.php';

   $collection->deleteOne(['\_id' => new MongoDB\BSON\ObjectID($\_GET['id'])]);

   $\_SESSION['success'] = "Data Mahasiswa Berhasil dihapus";

   header("Location: index.php");

   }

?>

<!DOCTYPE html>

<html>

   <head>

      <title>APLIKASI INTERAKTIF</title>

      <link rel="stylesheet" href="./vendor/twbs/bootstrap/dist/css/bootstrap.min.css">

   </head>

   <body>

      <div class="container">

         <br>

         <CENTER><h1>Hapus Data Mahasiswa</h1></CENTER>

         <h3> Yang bernama <?php echo "$mhs->Nama"; ?> dengan NIM <?php echo "$mhs->NIM"; ?> ? </h3>

         <form method="POST">

            <div class="form-group">

               <input type="hidden" value="<?php echo "$mhs->NIM"; ?>" class="form-control" name="NIM">

               <a href="index.php" class="btn btn-primary">Kembali</a>

               <button type="submit" name="submit" class="btn btn-danger">Hapus</button>

            </div>

         </form>

      </div>

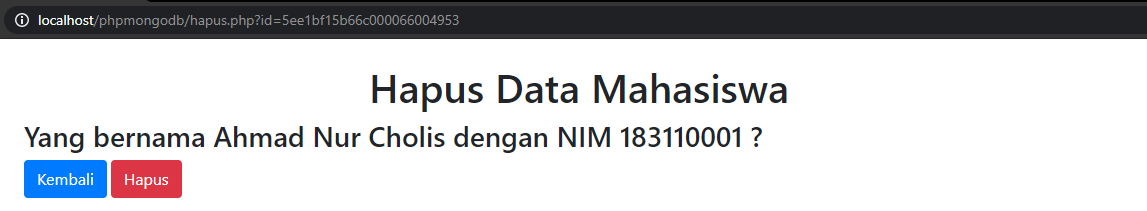
   </body>

</html>

**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file hapus.php, file hapus php merupakan file yang digunakan untuk menghapus data mahasiswa, file ini akan berjalan jika tombol hapsu didalam file index.php di klik, file ini berisi operasi dasar database yaitu menghapus record melalui form yang telah dibuat dengan method post, dalam menghapus record perlu dilakukan pencarian data terlebih dahulu, dan menghapus data hanya bisa satu persatu data yang dihapus karena menggunakan **findOne** untuk mencari data dan **deleteOne** untuk menghapus data.

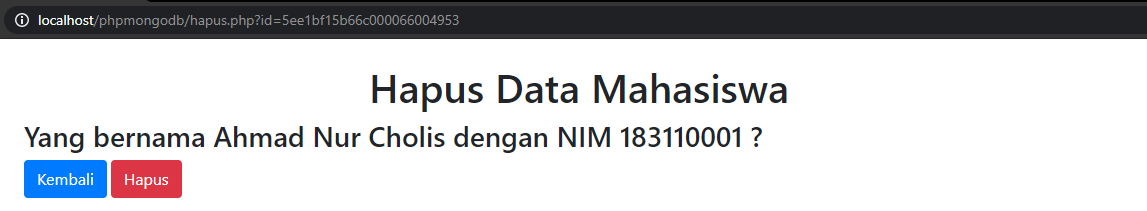
1. Tampilan file hapus.php



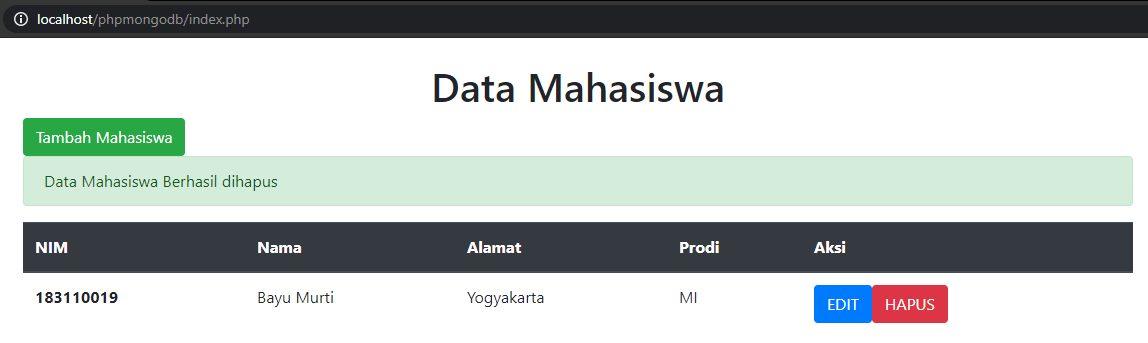
**Penjelasan:**

Pada tampilan diatas merupakan tampilan dari file hapus.php saat dijalankan, isi dari tampilan diatas antara lain judul, tombol kembali untuk kembali ke file index.php, form data yang digunakan hanya berupa tombol hapus yang berguna untuk menghapus data yang ada di collection.

1. Menghapus data mahasiswa menggunakan file hapus.php



Setelah data berhasil dihapus



1. **Kesimpulan**

Dalam membuat Operasi CRUD –Create, Read, Update, Delete– menggunakan database MongoDB yang dihubungkan dengan PHP yang pertama kali dilakukan adalah menginstal library terlebih dahulu, dilanjutkan untuk membuat koneksi terlebih dahulu baru bisa membuat file PHP yang digunakan untuk operasi CRUD database MongoDB, semua source code yang digunakan dapat dilihat di GitHub saya dengan URL <https://github.com/bayumurti1/phpmongodb> nantinya terdapat file phpmongodb.zip, bisa didownload dan diekstrak kedalam folder htdocs.