

Laporan Praktikum 15 mata pelajaran Basis Data 2 CRUD PHP + MongoDB

NIM : 452024611053
Nama : Farrel Ghozy Affifudin
Kelas : TI 4 A1
Link GitHub : https://github.com/FarrelGhozy/CRUD_PHP-MONGODB_Praktikum15.git

A) Tujuan Praktikum

- B) Tujuan dari praktikum ini adalah untuk memahami dan mengimplementasikan integrasi antara bahasa pemrograman PHP native dengan basis data NoSQL MongoDB. Melalui praktikum ini, mahasiswa diharapkan mampu melakukan operasi CRUD (Create, Read, Update, Delete) pada MongoDB menggunakan PHP, memahami perbedaan konsep penyimpanan data antara database relasional dan non-relasional, serta membangun aplikasi web sederhana yang dapat mengelola data mahasiswa secara dinamis. Selain itu, praktikum ini bertujuan untuk melatih mahasiswa dalam menggunakan Composer dan library resmi MongoDB sebagai bagian dari pengembangan aplikasi web modern.

C) Alat dan bahan

Alat dan bahan yang digunakan dalam praktikum ini adalah sebagai berikut:

1. **Laptop / PC** dengan sistem operasi **Linux Ubuntu**
2. **Web Server Apache** sebagai server lokal
3. **PHP versi 8.x** sebagai bahasa pemrograman backend
4. **MongoDB Server** sebagai basis data NoSQL
5. **MongoDB PHP Extension** untuk menghubungkan PHP dengan MongoDB
6. **Composer** sebagai dependency manager PHP
7. **Text Editor / Code Editor** (seperti Visual Studio Code atau editor bawaan Linux)
8. **Web Browser** (Mozilla Firefox / Google Chrome)

D) Persiapan Sebelum Praktikum

Sebelum melakukan praktikum, dilakukan beberapa tahap persiapan dan konfigurasi pada sistem operasi Linux Ubuntu agar aplikasi dapat berjalan dengan baik. Adapun langkah-langkah yang dilakukan adalah sebagai berikut:

1. Menginstal dan menjalankan **Apache Web Server** serta memastikan layanan Apache berjalan dengan baik.
2. Menginstal **PHP** beserta modul pendukung yang diperlukan untuk pengembangan aplikasi web.
3. Menginstal **MongoDB Server** dan memastikan service MongoDB aktif serta dapat diakses melalui terminal.
4. Menginstal dan mengaktifkan **MongoDB PHP Extension** agar PHP dapat terhubung dengan database MongoDB.
5. Menginstal **Composer** untuk mengelola library PHP.
6. Membuat folder project **CRUD MongoDB** pada direktori /var/www/html di Ubuntu.
7. Menggunakan Composer untuk menginstal **library resmi MongoDB untuk PHP**, sehingga terbentuk folder vendor dan file autoload.php.
8. Membuat file koneksi MongoDB (koneksi.php) untuk menghubungkan aplikasi PHP dengan database MongoDB.
9. Membuat file-file CRUD yang terdiri dari index.php, tambah.php, simpan.php, edit.php, update.php, dan hapus.php.
10. Melakukan pengujian aplikasi melalui web browser untuk memastikan proses **Create, Read, Update, dan Delete** berjalan dengan baik.
11. Melakukan **backup database MongoDB** menggunakan perintah mongodump setelah data berhasil disimpan.

E) Langkah praktikum

Langkah-langkah praktikum yang dilakukan pada modul CRUD PHP + MongoDB (Native) adalah sebagai berikut:

1. Persiapan Lingkungan Praktikum

Menyiapkan lingkungan pengembangan pada sistem operasi Linux Ubuntu dengan memastikan web server Apache dan PHP telah terinstal dan dapat berjalan dengan baik. Selain itu, memastikan MongoDB Server telah terinstal dan service MongoDB dalam keadaan aktif.

2. Instalasi MongoDB PHP Extension

Menginstal MongoDB PHP Extension menggunakan PECL agar PHP dapat terhubung dengan database MongoDB. Setelah instalasi selesai, extension diaktifkan pada file konfigurasi php.ini, kemudian web server Apache direstart untuk menerapkan perubahan.

3. Instalasi Composer

Menginstal Composer sebagai dependency manager PHP yang digunakan untuk mengelola library tambahan. Composer kemudian diverifikasi untuk memastikan dapat dijalankan melalui terminal.

4. Instalasi Library MongoDB PHP

Membuat direktori project CRUD MongoDB dan menggunakan Composer untuk menginstal library resmi MongoDB PHP. Proses ini menghasilkan folder vendor dan file autoload.php yang digunakan untuk memuat library MongoDB ke dalam aplikasi.

5. Pembuatan Struktur Folder Project

Menyusun struktur folder dan file project sesuai dengan modul praktikum, yang terdiri dari file koneksi database dan file CRUD seperti index.php, tambah.php, simpan.php, edit.php, update.php, dan hapus.php.

6. Pembuatan Koneksi MongoDB

Membuat file koneksi.php untuk menghubungkan aplikasi PHP dengan database MongoDB. Koneksi dilakukan ke database kampus dan koleksi mahasiswa menggunakan MongoDB Client.

7. Implementasi Operasi Read (READ)

Membuat halaman utama (index.php) untuk menampilkan data mahasiswa yang tersimpan dalam database MongoDB ke dalam bentuk tabel menggunakan PHP dan Bootstrap.

8. Implementasi Operasi Create (CREATE)

Membuat form input data mahasiswa pada file tambah.php dan memproses penyimpanan data ke database MongoDB melalui file simpan.php.

9. Implementasi Operasi Update (UPDATE)

Membuat form edit data mahasiswa pada file edit.php dan memproses perubahan data menggunakan file update.php sehingga data pada MongoDB dapat diperbarui.

10. Implementasi Operasi Delete (DELETE)

Membuat proses penghapusan data mahasiswa menggunakan file hapus.php berdasarkan NIM yang dipilih.

11. Validasi Data

Menambahkan validasi agar NIM mahasiswa bersifat unik sehingga tidak terjadi duplikasi data pada database MongoDB.

12. Modifikasi Tampilan

Memodifikasi tampilan halaman form tambah dan edit menggunakan Bootstrap dan CSS sederhana agar tampilan aplikasi lebih rapi dan mudah digunakan.

13. Pengujian Aplikasi

Melakukan pengujian terhadap seluruh fitur CRUD (Create, Read, Update, Delete) untuk memastikan aplikasi berjalan dengan baik tanpa error.

14. Backup Database

Melakukan backup database kampus menggunakan perintah mongodump sebagai bentuk pengamanan data setelah proses praktikum selesai.

F) Implementasi tugas praktikum

Implementasi Tugas Praktikum

Pada praktikum ini, mahasiswa diminta untuk mengimplementasikan beberapa tugas tambahan sesuai dengan modul praktikum. Adapun implementasi tugas yang telah dilakukan adalah sebagai berikut:

1. Implementasi Validasi NIM Unik

Validasi NIM dilakukan untuk memastikan bahwa data mahasiswa yang disimpan ke dalam database MongoDB tidak memiliki NIM yang sama. Proses validasi ini dilakukan pada saat penyimpanan data (Create). Sistem akan melakukan pengecekan terlebih dahulu ke koleksi mahasiswa berdasarkan NIM yang diinput. Apabila NIM tersebut sudah terdaftar di database, maka proses penyimpanan data akan dibatalkan dan sistem akan menampilkan pesan bahwa NIM sudah digunakan. Dengan adanya validasi ini, duplikasi data mahasiswa dapat dihindari dan integritas data tetap terjaga. Dengan menambahkan kode PHP :

```
<?php
include 'koneksi.php';

// cek NIM dulu
$cek = $collection->findOne(['nim' => $_POST['nim']]);

if ($cek) {
    echo "
    <script>
    alert('NIM sudah terdaftar!');
    window.location.href = 'tambah.php';
    </script>
    ";
```

```

exit;
}

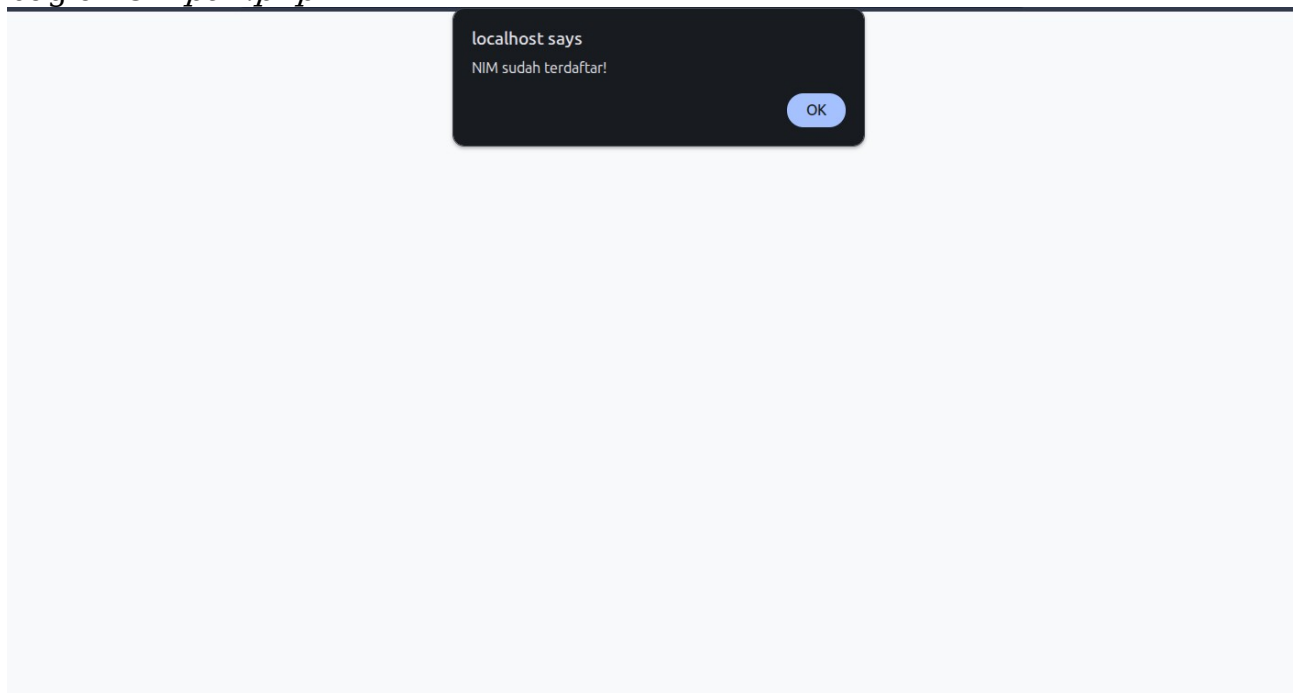
// jika belum ada, baru simpan
$data = [
    'nim' => $_POST['nim'],
    'nama' => $_POST['nama'],
    'prodi' => $_POST['prodi'],
    'umur' => (int) $_POST['umur']
];

$collection->insertOne($data);

header('Location: index.php');
exit;

```

ini akan mengecek terlebih dahulu sebelum menyimpan data nim karena nim bersifat UNIK dan tidak boleh ada duplikat data. Code ini di tambahkan di bagian *simpan.php*




2. Modifikasi Tampilan Form Tambah dan Edit

Modifikasi tampilan dilakukan pada halaman form tambah dan edit data mahasiswa agar tampilan aplikasi menjadi lebih rapi dan mudah digunakan. Perbaikan tampilan dilakukan dengan menambahkan CSS sederhana serta memanfaatkan komponen Bootstrap, seperti form control, tombol, dan layout container.

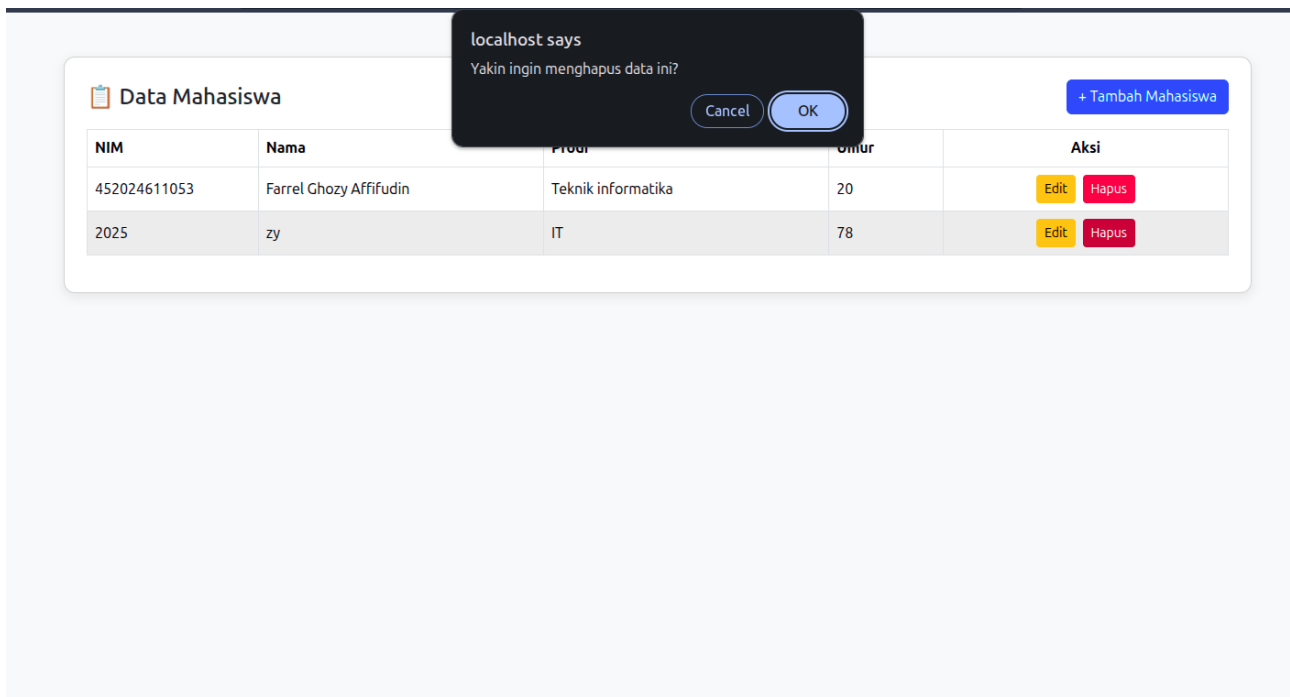
Dengan modifikasi ini, form input menjadi lebih terstruktur, nyaman

digunakan oleh pengguna, serta meningkatkan aspek user interface dari aplikasi CRUD yang dibuat.

 Data Mahasiswa

+ Tambah Mahasiswa

NIM	Nama	Prodi	Umur	Aksi
452024611053	Farrel Ghozy Affifudin	Teknik informatika	20	<div>EditHapus</div>



3. Backup Database MongoDB

Setelah data mahasiswa berhasil disimpan dan diuji, dilakukan proses backup database kampus sebagai langkah pengamanan data. Backup database dilakukan menggunakan perintah mongodump melalui terminal Linux Ubuntu. Proses backup ini menghasilkan salinan database yang dapat digunakan untuk pemulihan data apabila terjadi kerusakan atau kehilangan data di kemudian hari. Dengan demikian, data hasil praktikum tetap aman dan terarsipkan dengan baik.

```
Unpacking mongodb-database-tools (100.9.4) over (100.14.0) ...
Setting up mongodb-database-tools (100.9.4) ...
fazy@fazy:~$ mongodump --version
mongodump version: 100.9.4
git version: ce6af0fefca324ad5d9cb689d335130f48c99699
Go version: go1.20.12
os: linux
arch: amd64
compiler: gc
fazy@fazy:~$ mongodump --db kampus --out ~/Documents/backup_mongodb
2026-01-14T21:52:10.174+0700   writing kampus.mahasiswa to /home/fazy/Documents/backup_mongodb/kampus/mahasiswa.bson
2026-01-14T21:52:10.175+0700   done dumping kampus.mahasiswa (1 document)
fazy@fazy:~$
```

```
kampus> show collections
mahasiswa
kampus> db.mahasiswa.find()
[
  {
    _id: ObjectId('6967aa16340a2c00c1088652'),
    nim: '452024611053',
    nama: 'Farrel Chozy Affifudin ',
    prodi: 'Teknik informatika',
    umur: 20
  }
]
kampus>
```

G) Hasil

Berdasarkan praktikum yang telah dilakukan, berhasil dibuat sebuah aplikasi web sederhana berbasis PHP Native yang terintegrasi dengan database MongoDB. Aplikasi ini mampu melakukan pengelolaan data mahasiswa dengan menerapkan operasi CRUD (Create, Read, Update, Delete) secara fungsional.

Data mahasiswa yang disimpan ke dalam database MongoDB dapat ditampilkan dalam bentuk tabel, ditambahkan melalui form input, diperbarui melalui form edit, serta dihapus sesuai kebutuhan. Selain itu, sistem telah dilengkapi dengan validasi NIM unik sehingga tidak terjadi duplikasi data mahasiswa pada database.

Dari sisi tampilan, form tambah dan edit telah dimodifikasi menggunakan CSS sederhana dan Bootstrap sehingga tampilan aplikasi menjadi lebih rapi, terstruktur, dan mudah digunakan. Setelah seluruh data berhasil dimasukkan dan diuji, proses backup database kampus juga berhasil dilakukan menggunakan perintah mongodump pada sistem operasi Linux Ubuntu.

Dengan demikian, seluruh fitur dan tugas yang diminta pada modul praktikum dapat diimplementasikan dan berjalan dengan baik.

H) Kesimpulan

Berdasarkan hasil praktikum CRUD PHP + MongoDB (Native) yang telah dilakukan, dapat disimpulkan bahwa PHP Native dapat diintegrasikan dengan database MongoDB untuk membangun aplikasi web berbasis NoSQL. MongoDB memberikan fleksibilitas dalam penyimpanan data karena menggunakan struktur dokumen, sehingga tidak memerlukan skema tabel yang kaku seperti pada database relasional.

Praktikum ini juga memberikan pemahaman kepada mahasiswa mengenai penggunaan MongoDB PHP Extension dan Composer dalam pengembangan aplikasi web modern. Selain itu, penerapan validasi data, perbaikan tampilan antarmuka, serta proses backup database menunjukkan pentingnya aspek keamanan data, kenyamanan pengguna, dan manajemen data dalam pengembangan sistem informasi.

Secara keseluruhan, praktikum ini berhasil mencapai tujuan yang telah ditetapkan dan memberikan pengalaman praktis dalam membangun aplikasi CRUD menggunakan PHP Native dan MongoDB pada sistem operasi Linux Ubuntu.