

CRUD SEDERHANA OOP PHP & MYSQL

LAPORAN PRAKTIKUM PEMROGRAMAN WEB 6

Dosen Pengampu :

Nurfiah, S.ST, M.Kom



OLEH :

Stanley Leonard Darmali

2311532021

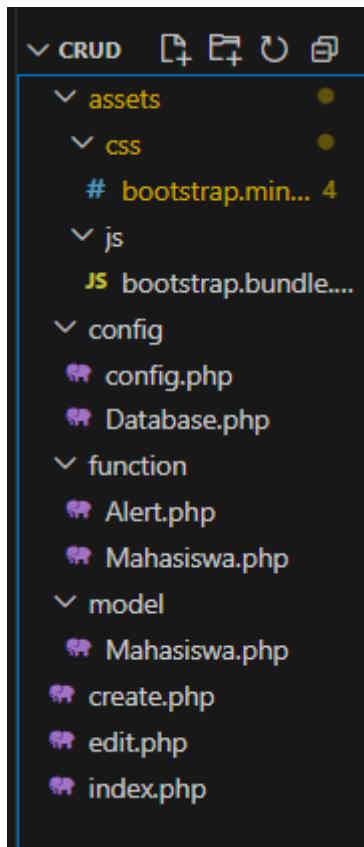
**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

1. Buatlah CRUD sederhana menggunakan konsep OOP PHP dan MySQL sesuai dengan modul

A. Langkah-Langkah:

1. Silahkan buat folder baru didalam folder htdocs dengan nama crud, kemudian buka menggunakan VS Code dan buat struktur file atau direktori seperti gambar berikut ini.



Selanjutnya download file css dan js bootstrap pada link berikut.

CSS <https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css>

JS <https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.js>

Buka link diatas kemudian save as file dan simpan kedalam folder css dan js yang ada didalam folder assets pada project yang akan dibuat.

2. Pertama-tama buat database baru dengan nama pemrograman_web, kemudian buat table mahasiswa sebagai berikut.

```
CREATE TABLE `mahasiswa` (
  `id` int(11) NOT NULL,
  `nim` varchar(16) NOT NULL,
  `nama` varchar(128) NOT NULL,
  `jurusan` varchar(128) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None			Change Drop More
2	nim	varchar(16)	utf8mb4_general_ci		No	None			Change Drop More
3	nama	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More
4	jurusan	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More

3. Selanjutnya membuat koneksi ke database, buka file config/Database.php dan ketikan kode program Berkut ini.

```

EXPLORER
crud
  assets
    css
      bootstrap.min.css
    js
      bootstrap.bundle.js
  config
    config.php
    Database.php
  function
    Alert.php
    Mahasiswa.php
  model
    create.php
    edit.php
    index.php
    Mahasiswa.php

config > Database.php > ...
1: <?php
2:   // references [Implementation]
3:   class Database {
4:     // references
5:     private $host = "localhost";
6:     // references
7:     private $db_name = "pemrograman_web";
8:     // references
9:     private $username = "root";
10:    // references
11:    private $password = "";
12:    // references
13:    public $conn;
14:    // references [0 overrides]
15:    public function getConnection(): mysqli
16:    {
17:      $this->conn = null;
18:      try {
19:        $this->conn = new mysqli(
20:          host: $this->host,
21:          username: $this->username,
22:          password: $this->password,
23:          database: $this->db_name
24:        );
25:      } catch (Exception $e) {
26:        echo "Connection error: " . $e->getMessage();
27:      }
28:      return $this->conn;
29:    }
30:  }
31: 
```

4. File Config.php digunakan untuk membuat variable constant yang mana nantinya dapat diakses dari kelas manapun, pada praktikum ini akan membuat variable constant dengan nama BASE_URL yang berisi strng base url dari project yang akan dibuat. Tambahkan kode program berikut pada Config.php

```

EXPLORER
crud
  assets
    css
      bootstrap.min.css
    js
      bootstrap.bundle.js
  config
    config.php
    Database.php

config > config.php > PHP Intelephense > BASE_URL
1: <?php
2: define(constant_name: 'BASE_URL', value: 'http://localhost/crud/');
3: 
```

5. Alert.php digunakan untuk membuat sebuah function yang berfungsi untuk menampilkan pesan Ketika melakukan operasi terhadap data, fungsi ini memiliki 2 buah argument statement yaitu \$msg yang berisi pesan dan \$sts yang berisi kode jika status 1 maka pesan berhasil dan jika 0 maka pesan gagal. Berikut kode program fungsi alert.

The screenshot shows two code editor panes. The left pane displays the contents of the file `Alert.php`:

```
function > Alert.php > PHP > alert()
1  <?php
2  3 references
3  function alert($msg, $sts): void {
4      if($sts == 1){
5          $type = 'success';
6      } else {
7          $type = 'danger';
8      }
9      echo '';
10     <div class="alert alert-".$type.'" role="alert">
11         '.$msg.'
12     </div>
13     ';
14 }
15 ?>
```

The right pane displays the contents of the file `Mahasiswa.php`:

```
model > Mahasiswa.php > PHP Intelephit
1  <?php
2  session_start();
3  3 references | 0 implementations
4  <?php
5  class Mahasiswa
6  {
7 }
```

6. Class Mahasiswa digunakan untuk membuat fungsi operasi Create, Read, Update dan Delete data Mahasiswa, berikut ini Langkah-langkah pembuatan class Mahasiswa.

- Tambahkan `session_start()` pada baris paling atas kode program
- Buat class dengan nama Mahasiswa

The screenshot shows the beginning of the `Mahasiswa` class definition in the file `Mahasiswa.php`:

```
model > Mahasiswa.php > PHP Intelephit
1  <?php
2  session_start();
3  3 references | 0 implementations
4  <?php
5  class Mahasiswa
6  {
7 }
```

- Selanjutnya deklarasikan variable-variabel yang dibutuhkan seperti gambar berikut.

The screenshot shows the properties defined within the `Mahasiswa` class:

```
5 references
private $conn;
5 references
private $table_name = "mahasiswa";

4 references
public $id;
4 references
public $nim;
4 references
public $nama;
4 references
public $jurusan;
```

- Buat fungsi constructor yg berisi memanggil koneksi database seperti gambar berikut.

The screenshot shows the implementation of the `__construct` constructor:

```
5 references, 0 overrides
public function __construct($db)
{
    $this->conn = $db;
}
```

- buat fungsi create yang nantinya digunakan untuk menambahkan data mahasiswa seperti kode program berikut. Catatan : `$_SESSION['flash_message']` berfungsi untuk membuat session yang berisi pesan berhasil atau tidak berhasil sebuah data ditambahkan. Sedangkan header digunakan untuk redirect ke file index.php dengan parameter msg

```
1 reference | 0 overrides
public function create(): void
{
    $query = "...INSERT INTO " . $this->table_name . " SET nim=?, nama=?, jurusan=?";
    $stmt = $this->conn->prepare($query);
    $stmt->bind_param("ssi", $this->nim, $this->nama, $this->jurusan);
    if ($stmt->execute()) {
        $_SESSION['flash_message'] = "Data berhasil disimpan!";
        header(header: "Location: " . BASE_URL . "index.php?msg=1");
    } else {
        $_SESSION['flash_message'] = "Data gagal disimpan!";
        header(header: "Location: " . BASE_URL . "index.php?msg=0");
    }
}
```

- buat fungsi read untuk menampilkan data mahasiswa, fungsi read memiliki default parameter `$id` yang berisi string kosong, jika `$id == ""` maka akan mengeksekusi query untuk menampilkan seluruh data mahasiswa, jika `$id != ""` maka akan mengeksekusi query untuk menampilkan data mahasiswa berdasarkan id mahasiswa. Berikut kode program fungsi read.

```
public function read($id = ""): mixed
{
    if ($id == "") {
        $query = "...SELECT * FROM " . $this->table_name;
    } else {
        $query = "...SELECT * FROM " . $this->table_name . " WHERE id =" . $id;
    }
    $result = $this->conn->query($query);
    return $result;
}
```

- buat fungsi update untuk melakukan perubahan data mahasiswa berdasarkan ID mahasiswa, berikut ini kode programnya.

```
public function update(): void
{
    $query = "...UPDATE " . $this->table_name . " SET nim=?, nama=?, jurusan=? WHERE id=?";
    $stmt = $this->conn->prepare($query);
    $stmt->bind_param("sssi", $this->nim, $this->nama, $this->jurusan, $this->id);
    if ($stmt->execute()) {
        $_SESSION['flash_message'] = "Data berhasil diupdate!";
        header(header: "Location: " . BASE_URL . "index.php?msg=1");
    } else {
        $_SESSION['flash_message'] = "Data gagal diupdate!";
        header(header: "Location: " . BASE_URL . "index.php?msg=0");
    }
}
```

- buat fungsi delete untuk menghapus data mahasiswa berdasarkan ID, berikut ini kode program untuk fungsi delete

```

    Preference|Overrides
public function delete(): void
{
    $query = "DELETE FROM " . $this->table_name . " WHERE id=?";
    $stmt = $this->conn->prepare($query);
    $stmt->bind_param("i", $this->id);
    if ($stmt->execute()) {
        $_SESSION['flash_message'] = "Data berhasil dihapus!";
        header(header: "Location: " . BASE_URL . "index.php?msg=1");
    } else {
        $_SESSION['flash_message'] = "Data gagal dihapus!";
        header(header: "Location: " . BASE_URL . "index.php?msg=0");
    }
}
}

```

7. Fungsi mahasiswa ini berfungsi sebagai controller yang menghubungkan antara class/model mahasiswa dengan view, buka file Mahasiswa.php yang berada dalam folder function kemudian tambahkan kode program seperti berikut.

- Tambahkan file Database.php, Config.php dan model/Mahasiswa.php dengan menggunakan perintah include seperti gambar dibawah ini.

```

function > 📁 Mahasiswa.php > ...
1   <?php
2   include_once '../config/Database.php';
3   include '../config/Config.php';
4   include_once '../model/Mahasiswa.php';
5

```

- Buat object baru Database dan mahasiswa kemudian koneksi dengan database seperti kode program berikut.

```

$database = new Database();
$db = $database->getConnection();
$mahasiswa = new Mahasiswa(db: $db);

```

- Selanjutnya tambahkan kode program berikut. Pertama akan dilakukan pengecekan sebuah parameter action yang dikirimkan melalui method GET, jika ada terdapat parameter action yang dikirimkan maka nilai parameter akan diambil dan digunakan sesuai dengan fungsinya, ada 3 buah isi parameter yang dikirimkan yaitu create, delete dan update.

```

if (isset($_GET['action'])) {
    if ($_GET['action'] == 'create') {
        $mahasiswa->nim = $_POST['nim'];
        $mahasiswa->nama = $_POST['nama'];
        $mahasiswa->jurusan = $_POST['jurusan'];
        $mahasiswa->create();
    } elseif ($_GET['action'] == 'delete') {
        $mahasiswa->id = $_GET['id'];
        $mahasiswa->delete();
    } elseif ($_GET['action'] == 'update') {
        $mahasiswa->id = $_POST['id'];
        $mahasiswa->nim = $_POST['nim'];
        $mahasiswa->nama = $_POST['nama'];
        $mahasiswa->jurusan = $_POST['jurusan'];
        $mahasiswa->update();
    }
}

```

- Berikut ini kode program Fungsi Mahasiswa secara lengkap

```
function > w Mahasiswa.php >...  
1 <?php  
2 include_once '../config/Database.php';  
3 include '../config/Config.php';  
4 include_once '../model/Mahasiswa.php';  
5  
6 $database = new Database();  
7 $db = $database->getConnection();  
8 $mahasiswa = new Mahasiswa(db: $db);  
9  
10 if (isset($_GET['action'])) {  
11     if ($_GET['action'] == 'create') {  
12         $mahasiswa->nim = $_POST['nim'];  
13         $mahasiswa->nama = $_POST['nama'];  
14         $mahasiswa->jurusan = $_POST['jurusan'];  
15         $mahasiswa->create();  
16     } elseif ($_GET['action'] == 'delete') {  
17         $mahasiswa->id = $_GET['id'];  
18         $mahasiswa->delete();  
19     } elseif ($_GET['action'] == 'update') {  
20         $mahasiswa->id = $_POST['id'];  
21         $mahasiswa->nim = $_POST['nim'];  
22         $mahasiswa->nama = $_POST['nama'];  
23         $mahasiswa->jurusan = $_POST['jurusan'];  
24         $mahasiswa->update();  
25     }  
26 }
```

8. View berfungsi untuk menampilkan data, menampilkan form input dan update data mahasiswa.

- Index.php

File Index.php ini berisi tampilan data-data mahasiswa yang telah berhasil diinputkan kedalam database, berikut ini kode program Index.php

```

1: <?php
2: include 'function/Alert.php';
3: include_once './config/Database.php';
4: include_once './model/Mahasiswa.php';
5:
6: $database = new Database();
7: $db = $database->getConnection();
8: $mahasiswa = new Mahasiswa($db);
9: $result = $mahasiswa->read();
10:
11: <!doctype html>
12: <html lang="en">
13:
14:   <head>
15:     <meta charset="utf-8">
16:     <meta name="viewport" content="width=device-width, initial-scale=1">
17:     <title>OOP - CRUD</title>
18:     <link href="assets/css/bootstrap.min.css" rel="stylesheet">
19:   </head>
20:
21: <body>
22:   <div class="container">
23:     <div class="row">
24:       <div class="col-lg-12">
25:         <h4 class="text-center">Data Mahasiswa</h4>
26:         <a class="btn btn-primary btn-sm mb-2" href="create.php">Tambah
27:           Mahasiswa</a>
28:         </p>
29:         <?php
30:         if (isset($_SESSION['flash_message']) && !isset($_GET['msg'])) {
31:           if ($_GET['msg'] == '1') {
32:             alert(msg: $_SESSION['flash_message'], stsi: 1);
33:           } else {
34:             alert(msg: $_SESSION['flash_message'], stsi: 0);
35:           }
36:         }
37:       </?>
38:       <table class="table table-striped table-bordered">
39:         <thead>
40:           <tr>
41:             <th>#</th>
42:             <th>NIM</th>
43:             <th>NAMA</th>
44:             <th>JURUSAN</th>
45:             <th>ACTION</th>
46:           </tr>
47:         </thead>
48:         <tbody>
49:           <?php $no = 1;
50:           while ($row = $result->fetch_assoc()) { ?>
51:             <tr>
52:               <td><?= $no++ ?></td>
53:               <td><?= $row['nim'] ?></td>
54:               <td><?= $row['nama'] ?></td>
55:               <td><?= $row['jurusan'] ?></td>
56:               <td>
57:                 <a class="btn btn-success btn-sm"
58:                   href="edit.php?id=<?= $row['id'] ?>">Edit</a>
59:                 <a class="btn btn-danger btn-sm"
60:                   href="function/Mahasiswa.php?action=delete&id=<?= $row['id'] ?>">
61:                     onclick="return confirm('Anda yakin ingin menghapus Data <?php echo $row['nama']; ?>');">Hapus</a>
62:                 </td>
63:               </tr>
64:             <?php } ?>
65:           </tbody>
66:         </table>
67:       </div>
68:     </div>
69:     <script src="assets/js/bootstrap.bundle.min.js"></script>
70:   </body>
71: </html>

```



- Create.php

File ini berisi form inputan untuk memambahkan data mahasiswa



- Edit.php

File ini berisi form untuk melakukan perubahan data mahasiswa.



2. Sempurnakan / costum CRUD sederhana ini menjadi lebih efesien, dinamis dan catat apa saja bagian yang ditambahkan atau diubah.

-Penambahan unset(\$_SESSION['flash_message']) di index.php

```
<?php
if (isset($_SESSION['flash_message']) && isset($_GET['msg'])) {
    if ($_GET['msg'] == '1') {
        alert(msg: $_SESSION['flash_message'], sts: 1);
    } else {
        alert(msg: $_SESSION['flash_message'], sts: 0);
    }
unset($_SESSION['flash_message']);
```



Agar pesan/flash message seperti ‘Data berhasil disimpan!’, ‘Data gagal disimpan!’, etc... menghilang setelah halamannya direload.



3. Upload project pada akun Github masing-masing

4. Buatlah Laporan Praktikum dan sertakan link repository Github

https://github.com/AbyssWatcher07/praktikum-pemrograman-web_231153021

5. Publish Laporan praktikum yang telah dibuat pada website pribadi

https://stanley.infokand23.my.id/Laporan_Praktikum_6.pdf