

LAPORAN PRAKTIKUM PEMROGRAMAN WEB
PRAKTIKUM 7 & 9 – OOP & FORM VALIDATION



Disusun oleh:

Nama : Alfath Roziq Widhayaka

Nim : L0122012

Kelas : A

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET

2024

BAB I

A. Penjelasan Source Code

Tugas praktikum pemrograman web week ke 7 dan 9 ini adalah melanjutkan tugas pada praktikum pemrograman web week ke 6. Di praktikum week ke 6 kemarin saya sudah membuat PHP dan DBMS yang menerapkan CRUD. Disini penugasan week ke 7 dan 9 adalah menambahkan Object Oriented Programming (OOP) menggunakan `mysql_query` dan menambahkan Form Validation pada CRUD tersebut. Berikut penjelasan source code pada OOP dan Form Validation.

1. Penerapan Object Oriented Programming `mysql_query` (WEEK 7)

A screenshot of a code editor with a dark background and light-colored text. The code is written in PHP and defines a class named 'Database'. The class has four private properties: '\$host' with value 'localhost', '\$username' with value 'root', '\$password' with value '', and '\$database' with value 'dbapplestore'. It also has a public property '\$conn'. The class has two public methods: '__construct()' which initializes the connection using 'mysqli' and checks for connection errors, and 'query()' which returns the result of a query. At the bottom of the code, an instance of the Database class is created and assigned to the variable '\$db'.

Gambar 1

Untuk penerapan OOP pada program ini berada di file **functions.php** dikarenakan file ini merupakan file yang berfungsi untuk menjalankan seluruh fungsi CRUD yang diminta. Kode PHP di atas merupakan implementasi dari OOP (Object-Oriented Programming) dalam manajemen koneksi dan pengiriman query ke database MySQL. Dalam kode ini, koneksi ke database diwakili oleh sebuah class yang disebut **Database**, yang memiliki properti untuk menyimpan detail koneksi seperti **host**, **username**, **password**, dan **nama database**. Konstruktors class Database digunakan untuk membuat objek koneksi ke database MySQL menggunakan **class mysqli** bawaan PHP. Selain itu, class ini juga menyediakan sebuah metode **query()** untuk menjalankan query SQL ke database yang telah terkoneksi. Fungsi **global query()** yang digunakan di luar class Database memanfaatkan objek koneksi yang dibuat di

dalam class tersebut, sehingga memungkinkan untuk melakukan query tanpa perlu membuat koneksi baru setiap kali. Contoh penggunaannya adalah setelah objek Database dibuat, dapat menggunakan fungsi **query()** untuk menjalankan query SQL ke database dan mendapatkan hasilnya. Dalam contoh ini, dilakukan **SELECT * FROM products** dan menampilkan hasilnya.

2. Penerapan Form Validation di functions.php (WEEK 9)

```
1 function upload()
2 {
3     // Mendapatkan informasi file yang diupload
4     $namaFile = $_FILES['image']['name'];
5     $ukuranFile = $_FILES['image']['size'];
6     $error = $_FILES['image']['error'];
7     $tmpName = $_FILES['image']['tmp_name'];
8
9     if ($error === 4) {
10         echo "<script>alert('Select Image');</script>";
11         return false;
12     }
13
14     $ekstensiGambarValid = ['jpg', 'jpeg', 'png'];
15     $ekstensiGambar = explode('.', $namaFile);
16     $ekstensiGambar = strtolower(end($ekstensiGambar));
17     if (!in_array($ekstensiGambar, $ekstensiGambarValid)) {
18         echo "<script>alert('File doesn't match');</script>";
19         return false;
20     }
21
22     // Periksa ukuran file gambar yang diupload
23     if ($ukuranFile > 1000000) {
24         echo "<script>alert('Size is too big!');</script>";
25         return false;
26     }
27
28     $namaFileBaru = uniqid() . '.' . $ekstensiGambar;
29
30     move_uploaded_file($tmpName, 'images/' . $namaFileBaru);
31
32     return $namaFileBaru;
33 }
```

Gambar 2

Form Validation yang pertama ada di file **functions.php** yaitu pada fungsi **upload()**. Fungsi **upload()** berperan dalam memeriksa berbagai aspek terkait file yang diunggah, seperti apakah sebuah file gambar dipilih, apakah ekstensinya valid, dan apakah ukurannya tidak terlalu besar. Pertama, dilakukan pemeriksaan apakah pengguna telah memilih file gambar. Jika tidak, akan muncul pesan yang mengingatkan untuk memilih gambar. Selanjutnya, ekstensi file gambar yang

diunggah dibandingkan dengan daftar ekstensi yang diizinkan. Jika ekstensi tidak sesuai, pengguna akan diberitahu bahwa file yang diunggah tidak cocok. Selain itu, dilakukan pengecekan terhadap ukuran file gambar; jika melebihi batas yang ditetapkan (dalam hal ini, 1 MB), akan diberikan pesan bahwa ukuran gambar terlalu besar. Jika semua pengecekan berhasil, gambar akan dipindahkan ke direktori yang ditentukan dengan nama unik menggunakan fungsi **uniqid()**, untuk menghindari kemungkinan nama file yang sama. Semua validasi ini terjadi sebelum data dikirimkan ke database, memastikan bahwa formulir hanya dianggap valid jika gambar yang diunggah memenuhi semua kriteria yang ditetapkan sebelumnya.

3. Penerapan Form Validation di tambah.php (WEEK 9)

```
1 <?php
2 require 'functions.php';
3
4 if(isset($_POST["submit"])){
5     // Validasi form sebelum ditambahkan
6     $errors = [];
7     if (empty($_POST["code"])) {
8         $errors[] = "Harap isi bidang ini.";
9     }
10    if (empty($_POST["product_name"])) {
11        $errors[] = "Harap isi bidang ini.";
12    }
13    if (empty($_POST["product_type"])) {
14        $errors[] = "Harap isi bidang ini.";
15    }
16    if (empty($_POST["price"])) {
17        $errors[] = "Harap isi bidang ini.";
18    }
19    if (empty($_FILES["image"]["name"])) {
20        $errors[] = "Harap isi bidang ini.";
21    }
22
23    if (empty($errors)) {
24        if (tambah($_POST) > 0) {
25            echo "<script>
26                alert('Success to Add Product!');
27            </script>";
28        } else {
29            echo "<script>
30                alert('Failed to Add Product!');
31            </script>";
32        }
33    } else {
34        echo "<script>alert('" . implode("\n", $errors) . "');</script>";
35    }
36 }
37 ?>
```

Gambar 3

Form Validation yang kedua ada di file **tambah.php**. Fungsi **tambah.php** ini digunakan untuk menambahkan produk apple pada laman website dengan proses validasi formulir sebelum data produk ditambahkan ke dalam database. Saat formulir disubmit, kode PHP pertama-tama memanggil **functions.php** yang berisi definisi

fungsi untuk menambahkan produk. Selanjutnya, setiap input dari formulir diverifikasi untuk memastikan bahwa mereka telah diisi dengan benar. Kode produk, nama produk, tipe produk, harga dan upload gambar diperiksa untuk memastikan tidak kosong. Jika tidak, pesan kesalahan "(code/name/tyoe/price/image) is required" dimasukkan ke dalam array **\$errors**. Jika terdapat kesalahan pada satu atau lebih input, pesan-pesan kesalahan akan ditampilkan kepada pengguna menggunakan JavaScript. Namun, jika tidak ada kesalahan yang terdeteksi setelah semua validasi dilakukan, fungsi **tambah()** akan dipanggil untuk memasukkan data produk ke dalam database.

4. Penerapan Form Validation di edit.php (WEEK 9)

```
1  <?php
2  require 'functions.php';
3
4  $id = $_GET["id"];
5
6  $appleproduct = query("SELECT * FROM appleproduct WHERE id=$id")[0];
7
8  $errors = array();
9
10 if(isset($_POST["submit"])){
11     // Validasi input sebelum melakukan pengeditan
12     if(empty($_POST['code'])) {
13         $errors[] = 'Harap isi bidang ini.';
14     }
15     if(empty($_POST['product_name'])) {
16         $errors[] = 'Harap isi bidang ini.';
17     }
18     if(empty($_POST['product_type'])) {
19         $errors[] = 'Harap isi bidang ini.';
20     }
21     if(empty($_POST['price'])) {
22         $errors[] = 'Harap isi bidang ini.';
23     }
24     if(empty($errors)) {
25         if(edit($_POST) > 0){
26             echo "<script>
27                 alert('Success to edit data!');
28             </script>";
29         } else{
30             echo "<script>
31                 alert('Failed to edit data!');
32             </script>";
33         }
34     }
35 }
36 >>
```

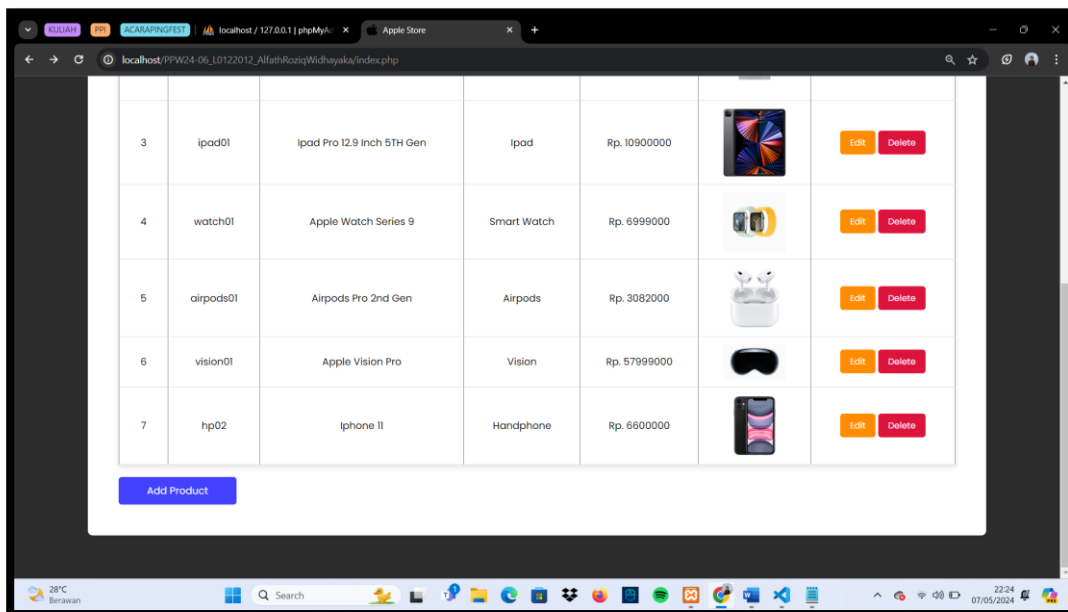
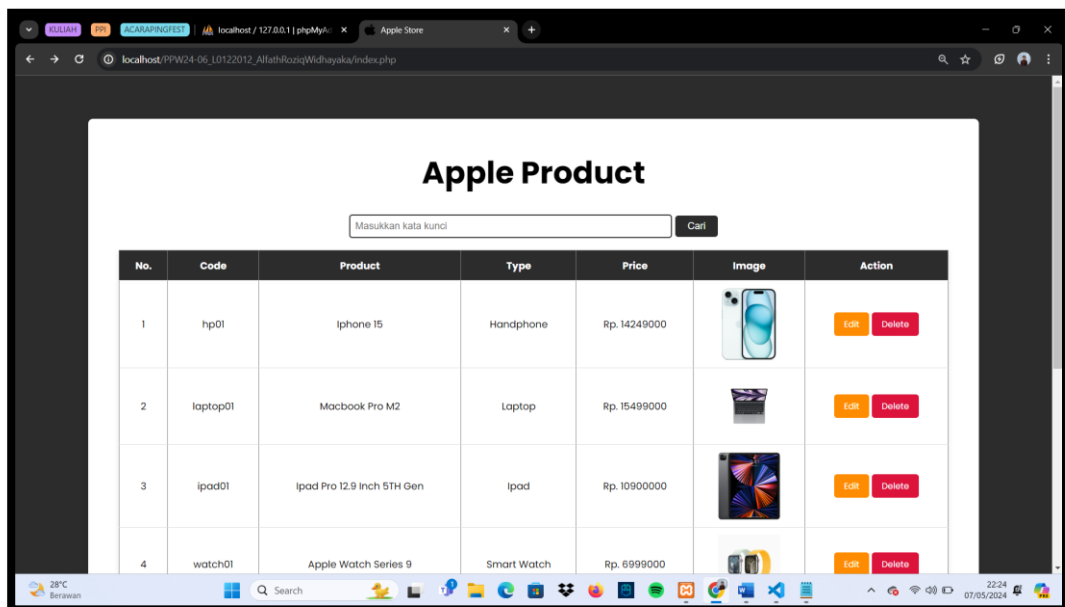
Gambar 4

Form Validation yang ketiga ada di file **edit.php**. Fungsi **edit.php** ini digunakan untuk memodifikasi/edit produk apple pada laman website dengan proses validasi formulir sebelum data produk dimodifikasi di dalam database. Proses validasi formulir

disini sama seperti pada **tambah.php** yaitu melakukan validasi di bagian kode produk, nama produk, tipe produk, dan harga produk. Untuk gambar produk tidak dilakukan validasi pada program ini dalam pengeditan. Untuk penjalanan fungsi validasi formulir juga sama seperti di **tambah.php**.

BAB II

A. Penggunaan Object Oriented Programming mysql_query (WEEK 7)

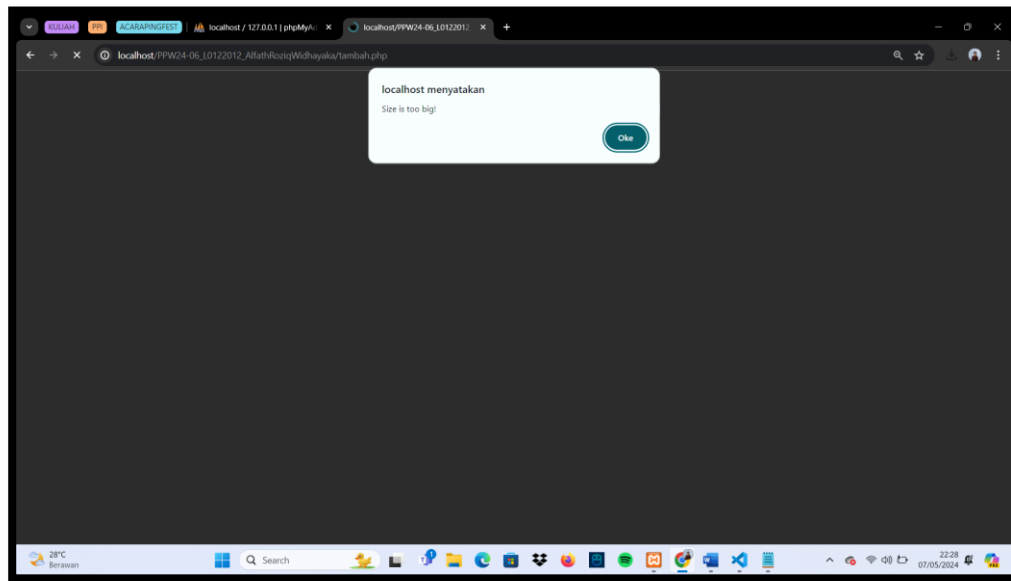


Gambar 5 & 6

Gambar di atas merupakan tampilan yang dihasilkan ketika user menggunakan OOP dengan mysql_query. Dengan menggunakan OOP, koneksi ke database diabstraksikan dalam sebuah kelas Database. Ini memungkinkan pengguna untuk dengan mudah membuat, mengelola, dan menggunakan koneksi database di seluruh aplikasi web tanpa perlu mengulangi kode koneksi yang sama di setiap halaman.

B. Penggunaan Form Validation (WEEK 9)

- Upload Image

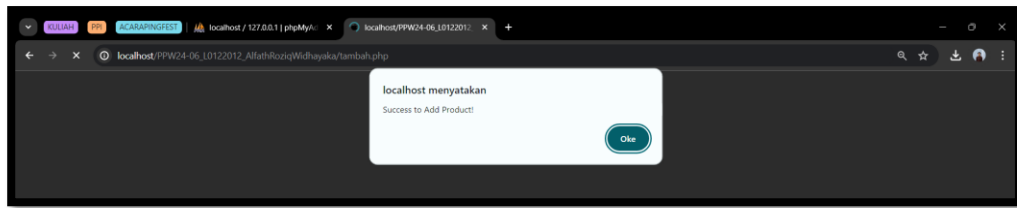


Gambar 7

Gambar 7 di atas merupakan contoh halaman jika user melakukan upload foto terlalu besar lebih dari 1 MB maka akan menampilkan pesan yang menyatakan **”Size is too big!”**. Maka, disini user diharapkan untuk melakukan upload foto yang memiliki ukuran dibawah 1 MB.

- Tambah Data

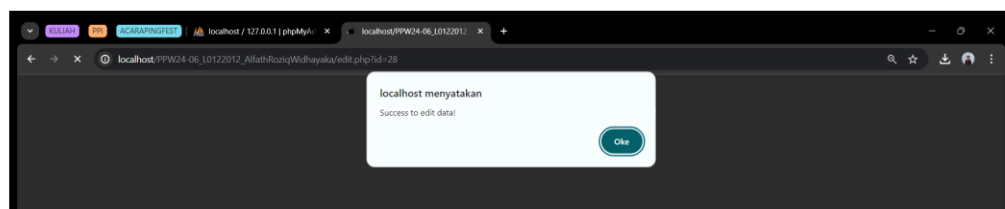
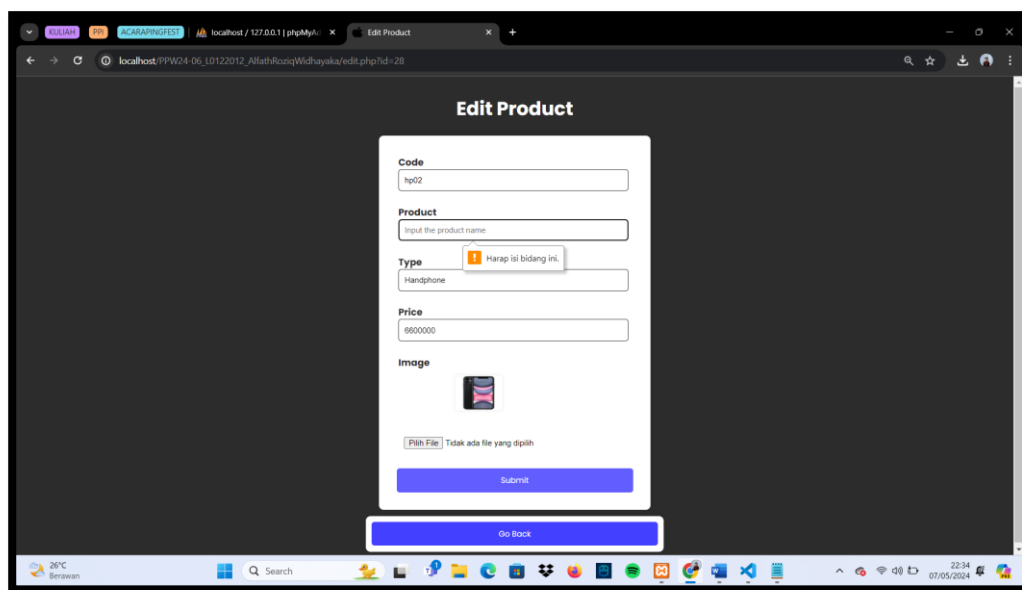
A screenshot of a web browser window showing a form titled 'Add Apple Product'. The form has several input fields: 'Code' (containing 'watch02'), 'Product' (containing 'Apple Watch SE'), 'Type' (containing 'Smart Watch'), and 'Price' (with a placeholder 'Input the product price'). Below these is an 'Image' section with a placeholder 'Harap isi bidang ini.' and a file input field containing 'applevatchse.jpeg'. At the bottom of the form are two buttons: 'Submit' and 'Go Back'. The browser's taskbar at the bottom shows the time as 22:31 on 07/05/2024.



Gambar 8 & 9

Gambar di atas merupakan tampilan ketika user melakukan penambahan produk. Untuk gambar 8 merupakan gambar dimana penggunaan form validation, pada gambar tersebut user belum memasukkan harga produk tetapi user sudah menekan tombol submit. Maka, yang terjadi adalah terdapat pesan error pada bagian form yang kosong dan memunculkan pesan error **"Harap isi bidang ini."**. Selanjutnya untuk gambar ke 9 ini merupakan gambar ketika user telah berhasil mengisi semua inputan pada form dan memunculkan pesan **"Success to Add Product!"**.

- **Edit Data**



Gambar 10 & 11

Gambar di atas merupakan tampilan ketika user melakukan modifikasi atau edit produk. Untuk gambar 10 merupakan gambar dimana penggunaan form validation, pada gambar tersebut user belum memasukkan nama produk tetapi user sudah menekan tombol submit. Maka, yang terjadi adalah terdapat pesan error pada bagian form yang kosong dan memunculkan pesan error **"Harap isi bidang ini."**. Selanjutnya untuk gambar ke 11 ini merupakan gambar ketika user telah berhasil mengisi semua inputan pada form dan memunculkan pesan **"Success to edit data!"**.

BAB III

Dari praktikum pemrograman web week 7 dan week 9 yang telah dilakukan dapat disimpulkan bahwa implementasi Object Oriented Programming (OOP) dan Form Validation secara efektif meningkatkan keamanan, kehandalan, dan kualitas aplikasi web dalam mengelola data produk apple yang saya buat. Dengan penerapan OOP, manajemen koneksi database menjadi lebih terstruktur dan efisien, sementara Form Validation memastikan bahwa data yang dimasukkan pengguna valid sebelum disimpan ke dalam database. Hal ini membantu dalam mengurangi kesalahan input dan meningkatkan keandalan data yang disimpan. Selain itu, pengguna diberikan pesan yang jelas dan informatif ketika terjadi kesalahan, meningkatkan pengalaman pengguna secara keseluruhan. Dengan demikian, praktikum ini tidak hanya memberikan pemahaman mendalam tentang konsep OOP dan Form Validation dalam pengembangan aplikasi web, tetapi juga menghasilkan aplikasi yang lebih robust dan user-friendly.