

LAPORAN PRAKTIKUM PEMROGRAMAN WEB

PRAKTIKUM 6 – PHP & DBMS



Disusun oleh:

Nama : Alfath Roziq Widhayaka

Nim : L0122012

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

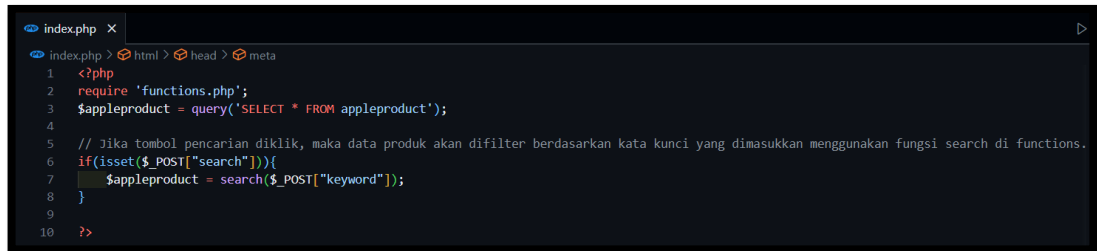
UNIVERSITAS SEBELAS MARET

2024

BAB I

A. Penjelasan Source Code “index.php”

- PHP



```
1 <?php
2 require 'functions.php';
3 $appleproduct = query('SELECT * FROM appleproduct');
4
5 // Jika tombol pencarian diklik, maka data produk akan difilter berdasarkan kata kunci yang dimasukkan menggunakan fungsi search di functions.
6 if(isset($_POST["search"])){
7     $appleproduct = search($_POST["keyword"]);
8 }
9
10 ?>
```

Gambar 1

Kode PHP tersebut ada di dalam file **index.php** yang merupakan halaman utama, memuat file **functions.php** yang berisi definisi fungsi-fungsi yang akan digunakan. Setelah memuatkan functions.php, kode tersebut mengambil data produk apple dari database menggunakan fungsi query yang didefinisikan di dalam file **functions.php**. Data produk apple ini kemudian disimpan dalam variabel **\$appleproduct**. Selanjutnya, terdapat tombol untuk pencarian. Jika tombol pencarian diklik **isset(\$_POST["search"])**, maka akan dilakukan filter data produk apple berdasarkan kata kunci yang dimasukkan oleh pengguna. Proses ini menggunakan fungsi search yang juga didefinisikan di dalam file **functions.php**. Hasilnya akan disimpan kembali dalam variabel **\$appleproduct**. Dengan demikian, fungsi ini mempermudah pengguna untuk mencari produk apple berdasarkan kata kunci yang dimasukkan, dan hasil pencarian akan ditampilkan kembali di halaman tersebut.

- HTML CSS

```

164 <table border="1" cellpadding="10">
165 <tr>
166 <th>No.</th>
167 <th>Code</th>
168 <th>Product</th>
169 <th>Type</th>
170 <th>Price</th>
171 <th>Image</th>
172 <th>Action</th>
173 </tr>
174 <?php $i = 1; ?>
175 <?php foreach($appleproduct as $row) : ?>
176 <tr>
177 <td><? $i++; ?></td>
178
179 <td><? $row['code']; ?></td>
180 <td><? $row['product_name']; ?></td>
181 <td><? $row['product_type']; ?></td>
182 <td>Rp. <? $row['price']; ?></td>
183 <td>
184 
185 </td>
186 <td>
187 <a href="edit.php?id=<? $row['id']; ?>" class="edit-button">Edit</a>
188 <a href="hapus.php?id=<? $row['id']; ?>" onclick="return confirm('Are you sure to delete?');" class="delete-button">Delete</a>
189 </td>
190 </tr>
191 <?php endforeach; ?>
192 </table>
193 <a href="tambah.php" class="button">Add Product</a>
194 </div>
195 </body>
196 </html>

```

Gambar 2

Disini saya menggunakan HTML dan CSS untuk memperindah tampilan pada halaman **index.php**. Di dalamnya saya menyelipkan sedikit kode PHP penyempurnaan pembuatan tabel secara dinamis serta untuk mengisi produk apple dari variabel **\$appleproduct**. Untuk setiap baris produk, terdapat dua tautan untuk aksi “Edit” dan “Delete”. Tautan “Edit” mengarahkan pengguna ke halaman **edit.php** dengan menyertakan parameter ID produk yang akan diedit, sedangkan tautan “Delete” mengarahkan pengguna ke halaman **hapus.php** dengan konfirmasi penghapusan data. Di bagian bawah tabel, terdapat sebuah tautan “Add Product” yang mengarahkan pengguna ke halaman **tambah.php** untuk menambahkan produk apple baru.

B. Penjelasan Source Code “tambah.php”

■ PHP

```

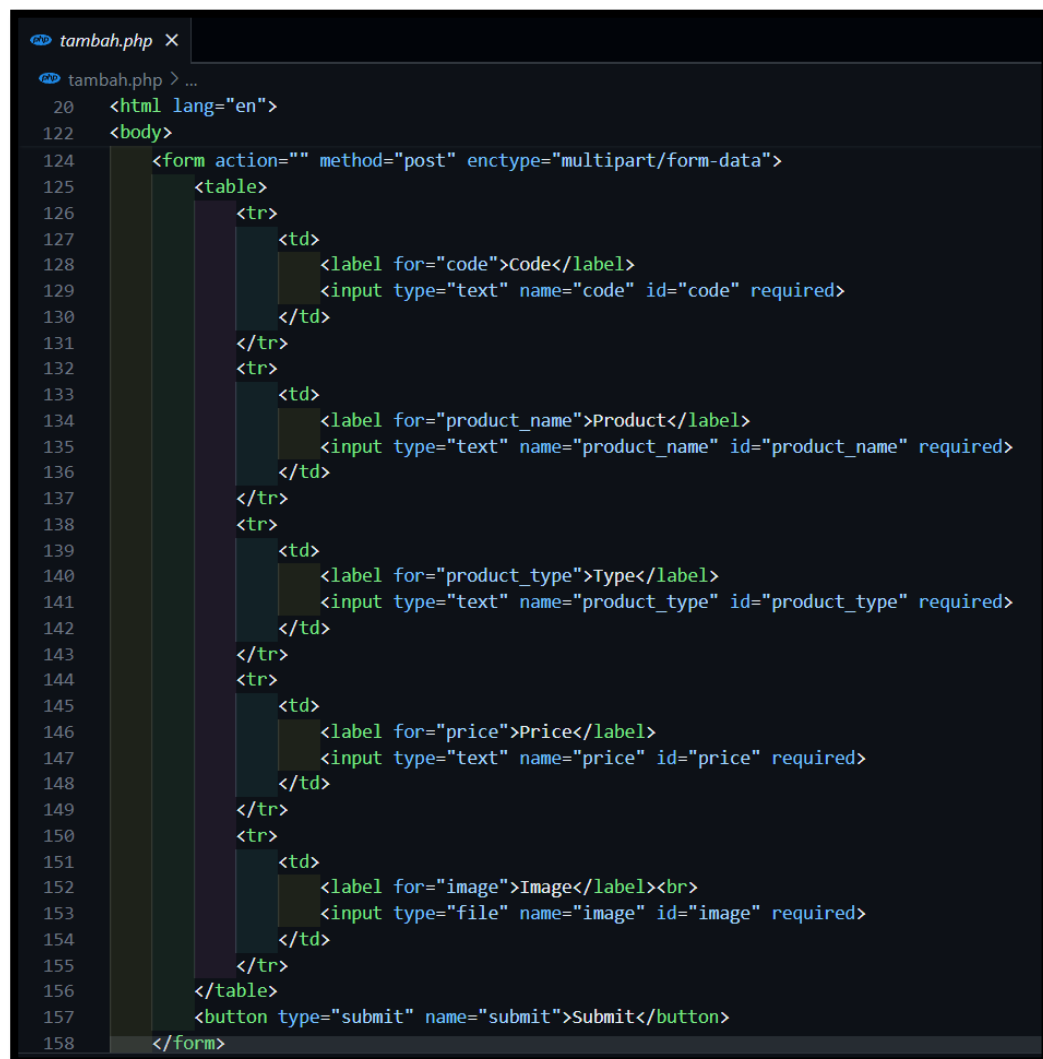
tambah.php x
tambah.php > html > head > style > body
1 <?php
2
3 require 'functions.php';
4
5 // Memanggil fungsi tambah dari file functions.php dengan parameter data dari $_POST, jika nilai kembalian lebih dari 0, artinya data berhasil
6 if(isset($_POST["submit"])){
7     if (tambah($_POST) > 0) {
8         echo "<script>
9         alert('Success to Add Product!');
10        </script>";
11    } else {
12        echo "<script>
13        alert('Failed to Add Product!');
14        </script>";
15    }
16 }
17 ?>

```

Gambar 3

Kode PHP di atas untuk menangani proses penambahan produk apple baru ke dalam database melalui formulir. Saat formulir disubmit, kode akan memeriksa apakah tombol submit telah ditekan dengan menggunakan `isset($_POST["submit"])`. Jika tombol submit telah ditekan, maka kode akan memanggil fungsi `tambah($_POST)` dari file `functions.php` untuk menambahkan data baru berdasarkan informasi yang dikirimkan melalui metode **POST**. Setelah pemanggilan fungsi `tambah()`, kode akan memeriksa apakah hasil operasi penambahan data lebih dari 0. Jika iya, berarti penambahan data berhasil dilakukan, dan pesan sukses akan ditampilkan kepada pengguna menggunakan alert JavaScript melalui echo `<script>alert('Success to Add Product!');</script>`. Namun, jika hasil operasi penambahan data tidak lebih dari 0, maka pesan kegagalan akan ditampilkan dengan cara yang sama.

- **HTML CSS**



```
tambah.php x
tambah.php > ...
20  <html lang="en">
122  <body>
124  <form action="" method="post" enctype="multipart/form-data">
125      <table>
126          <tr>
127              <td>
128                  <label for="code">Code</label>
129                  <input type="text" name="code" id="code" required>
130              </td>
131          </tr>
132          <tr>
133              <td>
134                  <label for="product_name">Product</label>
135                  <input type="text" name="product_name" id="product_name" required>
136              </td>
137          </tr>
138          <tr>
139              <td>
140                  <label for="product_type">Type</label>
141                  <input type="text" name="product_type" id="product_type" required>
142              </td>
143          </tr>
144          <tr>
145              <td>
146                  <label for="price">Price</label>
147                  <input type="text" name="price" id="price" required>
148              </td>
149          </tr>
150          <tr>
151              <td>
152                  <label for="image">Image</label><br>
153                  <input type="file" name="image" id="image" required>
154              </td>
155          </tr>
156      </table>
157      <button type="submit" name="submit">Submit</button>
158  </form>
```

Gambar 4

Disini saya menggunakan HTML dan CSS untuk memperindah tampilan pada halaman **tambah.php**. Terdapat sebuah formulir **<form>** dengan atribut action yang menunjukkan lokasi pengiriman data dan metode post. Form ini digunakan untuk mengirimkan produk apple baru ke server. Formulir ini memiliki beberapa elemen input yang mewakili berbagai atribut produk apple, seperti kode, nama produk, macam produk, harga, dan gambar. Setiap elemen input diikuti dengan label yang menjelaskan atributnya. Dibawahnya terdapat button **"Go Back"** yang digunakan untuk kembali menuju ke halaman utama atau **index.php**.

C. Penjelasan Source Code "hapus.php"

- PHP



```
hapus.php X
hapus.php > ...
1  <?php
2
3  require 'functions.php'; // Memuat file functions.php yang berisi definisi fungsi-fungsi yang diperlukan
4
5  // Mendapatkan ID dari URL
6  $id = $_GET["id"];
7
8  // Memanggil fungsi hapus dari file functions.php dengan parameter ID yang didapatkan dari URL
9  if(hapus($id) > 0){
10     echo "<script>
11         alert('Success to delete!');
12         document.location.href='index.php';
13     </script>";
14 } else{
15     echo "<script>
16         alert('Failed to delete!');
17         document.location.href='index.php';
18     </script>";
19 }
20
21 >
```

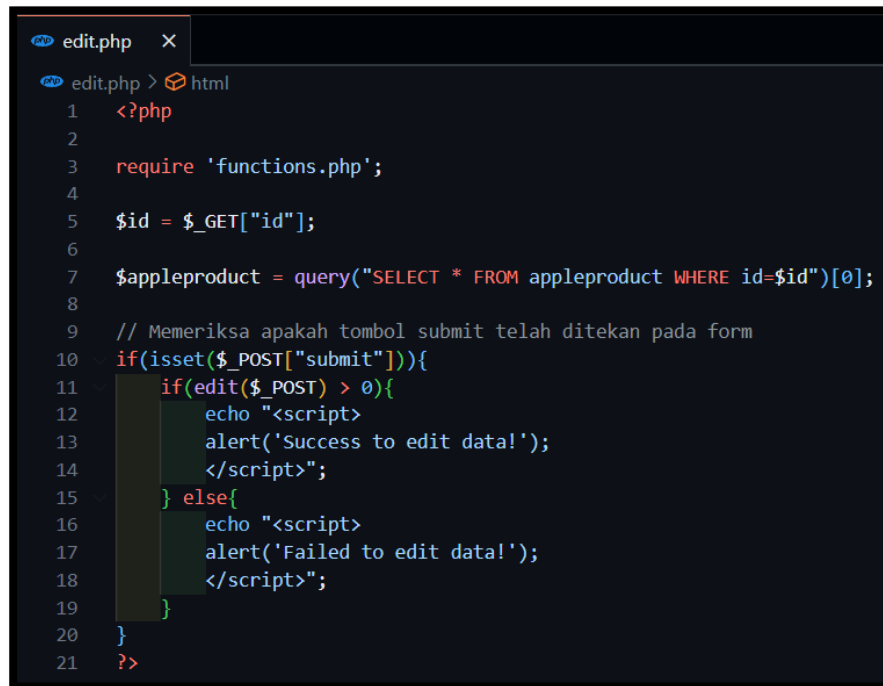
Gambar 5

Kode PHP di atas digunakan untuk menghapus salah satu produk apple dari database berdasarkan ID yang diperoleh dari URL. Pertama, file **functions.php** dimuat menggunakan perintah **require** untuk memuat definisi fungsi-fungsi yang diperlukan. Kemudian, ID produk apple diperoleh dari URL menggunakan **\$_GET["id"]** yang memungkinkan kita untuk mengetahui ID produk apple mana yang akan dihapus. Selanjutnya, fungsi **hapus(\$id)** dari file **functions.php** dipanggil dengan parameter ID appleproduct yang didapatkan dari URL. Fungsi ini bertugas untuk menghapus data produk apple dari database berdasarkan ID. Jika penghapusan berhasil (nilai kembalian dari fungsi **hapus(\$id)**), maka akan muncul pesan sukses menggunakan **alert** JavaScript, dan pengguna akan diarahkan kembali ke halaman **index.php**.

menggunakan **document.location.href**. Jika penghapusan gagal, akan muncul pesan gagal menggunakan alert JavaScript, dan pengguna juga akan diarahkan kembali ke halaman **index.php** menggunakan **document.location.href**.

D. Penjelasan Source Code “edit.php”

- PHP



```
edit.php X
edit.php > html
1  <?php
2
3  require 'functions.php';
4
5  $id = $_GET["id"];
6
7  $appleproduct = query("SELECT * FROM appleproduct WHERE id=$id")[0];
8
9  // Memeriksa apakah tombol submit telah ditekan pada form
10 if(isset($_POST["submit"])){
11     if(edit($_POST) > 0){
12         echo "<script>
13             alert('Success to edit data!');
14         </script>";
15     } else{
16         echo "<script>
17             alert('Failed to edit data!');
18         </script>";
19     }
20 }
21 ?>
```

Gambar 6

Kode PHP di atas untuk menangani pengeditan produk apple berdasarkan ID yang diperoleh dari URL. Skrip dimulai dengan memuat file **functions.php**, yang berisi definisi fungsi-fungsi yang dibutuhkan. Selanjutnya, ID produk apple diambil dari URL menggunakan metode **\$_GET["id"]**. Data produk apple yang sesuai dengan ID tersebut kemudian diambil dari database menggunakan fungsi **query("SELECT * FROM appleproduct WHERE id=\$id")[0]**. Data ini nantinya akan digunakan untuk mengisi formulir edit. Setelah mendapatkan produk apple, memeriksa apakah tombol submit pada formulir edit telah ditekan menggunakan kondisi **isset(\$_POST["submit"])**. Jika proses edit berhasil dilakukan (nilai kembalian fungsi **edit()** lebih dari 0), maka akan ditampilkan pesan sukses menggunakan alert JavaScript. Sebaliknya, jika proses edit gagal, juga akan ditampilkan pesan gagal.

■ HTML CSS

```
edit.php x
edit.php > html
25 <html lang="en">
117 <body>
119 <form action="" method="post" enctype="multipart/form-data">
120 <input type="hidden" name="id" value="<?=$appleproduct["id"]; ?>">
121 <input type="hidden" name="gambarAwal" value="<?=$appleproduct["image"]; ?>">
122 <table>
123 <tr>
124 <td>
125 <label for="code">Code</label>
126 <input type="text" name="code" id="code" value="<?=$appleproduct["code"]; ?>" required>
127 </td>
128 </tr>
129 <tr>
130 <td>
131 <label for="product_name">Product</label>
132 <input type="text" name="product_name" id="product_name" value="<?=$appleproduct["product_name"]; ?>" required>
133 </td>
134 </tr>
135 <tr>
136 <td>
137 <label for="product_type">Type</label>
138 <input type="text" name="product_type" id="product_type" value="<?=$appleproduct["product_type"]; ?>" required>
139 </td>
140 </tr>
141 <tr>
142 <td>
143 <label for="price">Price</label>
144 <input type="text" name="price" id="price" value="<?=$appleproduct["price"]; ?>" required>
145 </td>
146 </tr>
147 <tr>
148 <td>
149 <label for="image">Image</label><br>
150 " alt="none" width="80px" height="60px"><br>
151 <input type="file" name="image" id="image" required>
152 </td>
153 </tr>
```

Gambar 7

Disini penggunaan HTML dan CSS untuk memperindah tampilan milik **edit.php**. Di dalamnya terdapat formulir **<form>** dengan atribut action yang menunjukkan lokasi pengiriman data dan metode post. Formulir ini digunakan untuk mengirimkan produk apple yang telah diubah ke server. Setiap elemen input dalam formulir mewakili atribut produk apple, seperti kode, nama produk, macam produk, harga, dan gambar. Setiap input diikuti oleh label yang menjelaskan atributnya. Input gambar diwajibkan (**required**) dan menggunakan tipe file untuk mengunggah gambar. Terdapat juga dua input tersembunyi (**<input type="hidden">**) yang digunakan untuk menyimpan ID dan nama file gambar produk apple sebelumnya yang digunakan untuk memastikan bahwa data yang diubah sesuai dengan entri yang benar dalam database. Dibawahnya terdapat button **"Submit"** dan **"Go Back"** yang digunakan untuk penyelesaian edit dan kembali menuju ke halaman utama atau **index.php**.

E. Penjelasan Source Code “functions.php”

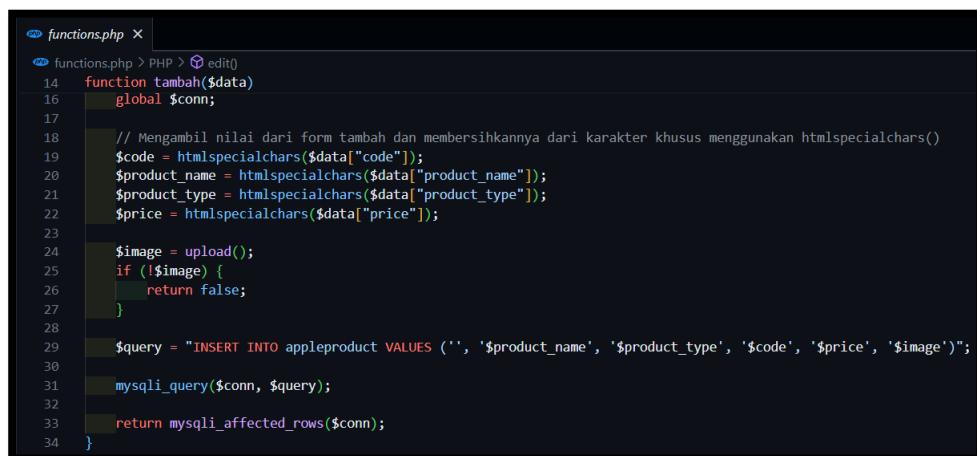


```
functions.php X
functions.php > PHP > edit()
1  <?php
2  $conn = mysqli_connect("localhost", "root", "", "dbapplestore"); // Menghubungkan ke database menggunakan mysqli_connect()
3
4  3 references
5  function query($query){
6      global $conn;
7      $result = mysqli_query($conn, $query); // Menjalankan query pada database
8      $rows = [];
9      while($row = mysqli_fetch_assoc($result)){
10         $rows[] = $row;
11     }
12     return $rows;
13 }
```

Gambar 8

Kode PHP di atas merupakan bagian dari file **functions.php** yang bertanggung jawab untuk mengelola koneksi ke database dan mengeksekusi query SQL. Fungsi **mysqli_connect()** digunakan untuk menghubungkan ke database MySQL. Dalam contoh ini, digunakan host localhost, username root, password kosong (“”), dan nama database **dbapplestore**. Koneksi ini disimpan dalam variabel **\$conn** agar dapat diakses di seluruh skrip. Fungsi **query()** digunakan untuk menjalankan query SQL pada database yang telah terhubung. Fungsi ini menerima parameter **\$query**, yang merupakan query SQL yang akan dieksekusi. Di dalam fungsi ini, terlebih dahulu dilakukan pengambilan hasil query dengan **mysqli_query()**, kemudian hasilnya disimpan dalam variabel **\$result**. Selanjutnya, hasil query tersebut diambil baris per baris menggunakan **mysqli_fetch_assoc()**, dan setiap baris disimpan dalam array **\$rows**. Akhirnya, array **\$rows** yang berisi hasil query tersebut dikembalikan sebagai hasil fungsi. Selain itu terdapat function-function yang diterapkan, berikut penjelasan lebih detail:

- **Function tambah(\$data)**

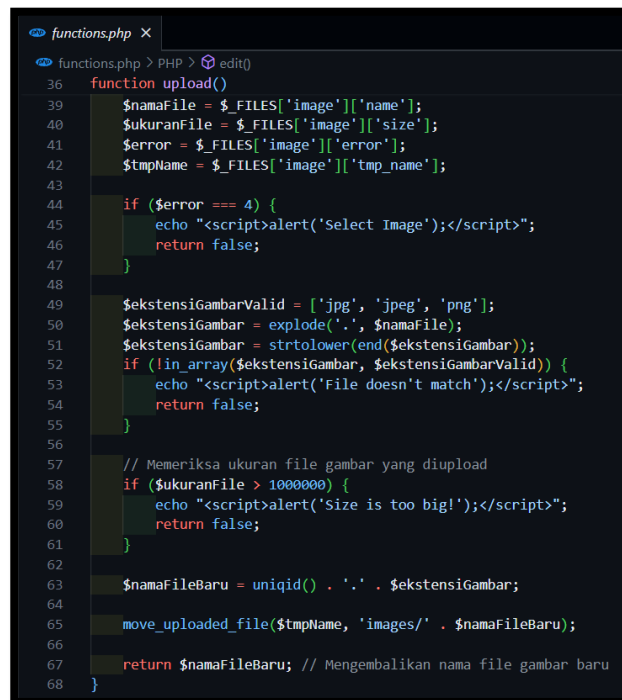


```
functions.php X
functions.php > PHP > edit()
14 function tambah($data)
15 {
16     global $conn;
17
18     // Mengambil nilai dari form tambah dan membersihkannya dari karakter khusus menggunakan htmlspecialchars()
19     $code = htmlspecialchars($data["code"]);
20     $product_name = htmlspecialchars($data["product_name"]);
21     $product_type = htmlspecialchars($data["product_type"]);
22     $price = htmlspecialchars($data["price"]);
23
24     $image = upload();
25     if (!$image) {
26         return false;
27     }
28
29     $query = "INSERT INTO appleproduct VALUES ('', '$product_name', '$product_type', '$code', '$price', '$image')";
30
31     mysqli_query($conn, $query);
32
33     return mysqli_affected_rows($conn);
34 }
```

Gambar 9

Pertama, fungsi **tambah(\$data)** berfungsi untuk menambahkan produk apple baru ke dalam database. Fungsi ini menerima produk apple yang dikirim dari form tambah sebagai parameter **\$data**. Data tersebut kemudian dibersihkan dari karakter khusus menggunakan **htmlspecialchars()**. Fungsi **upload()** dipanggil untuk mengunggah gambar produk apple. Jika proses upload gambar berhasil, sebuah query SQL disusun dengan menggunakan nama file gambar yang baru diunggah. Query tersebut kemudian dieksekusi menggunakan fungsi **mysqli_query()**, yang akan menjalankan perintah SQL untuk menyimpan data baru ke dalam tabel **appleproduct** di dalam database. Setelah query dieksekusi, fungsi **tambah()** mengembalikan jumlah baris yang terpengaruh oleh operasi query tersebut.

- **Function upload()**



```
functions.php X
functions.php > PHP > edit()
36 function upload()
39     $namaFile = $_FILES['image']['name'];
40     $ukuranFile = $_FILES['image']['size'];
41     $error = $_FILES['image']['error'];
42     $tmpName = $_FILES['image']['tmp_name'];
43
44     if ($error === 4) {
45         echo "<script>alert('Select Image');</script>";
46         return false;
47     }
48
49     $ekstensiGambarValid = ['jpg', 'jpeg', 'png'];
50     $ekstensiGambar = explode('.', $namaFile);
51     $ekstensiGambar = strtolower(end($ekstensiGambar));
52     if (in_array($ekstensiGambar, $ekstensiGambarValid)) {
53         echo "<script>alert('File doesn't match');</script>";
54         return false;
55     }
56
57     // Memeriksa ukuran file gambar yang diupload
58     if ($ukuranFile > 1000000) {
59         echo "<script>alert('Size is too big!');</script>";
60         return false;
61     }
62
63     $namaFileBaru = uniqid() . '.' . $ekstensiGambar;
64
65     move_uploaded_file($tmpName, 'images/' . $namaFileBaru);
66
67     return $namaFileBaru; // Mengembalikan nama file gambar baru
68 }
```

Gambar 10

Fungsi **upload()** digunakan untuk mengelola proses upload gambar produk apple. Saat fungsi ini dipanggil, ia melakukan berbagai langkah untuk memeriksa, memvalidasi, dan mengelola file gambar yang diunggah. Pertama, fungsi ini mengambil informasi yang terkait dengan file yang diunggah, seperti nama file, ukuran, dan status error menggunakan variabel **\$_FILES**. Selanjutnya, fungsi memeriksa apakah file gambar berhasil diunggah atau tidak dengan memeriksa nilai error dari array **\$_FILES**. Jika nilai error adalah 4, itu menunjukkan bahwa tidak ada file yang diunggah, dan sebuah

pesan peringatan akan ditampilkan kepada pengguna menggunakan JavaScript melalui echo. Fungsi kemudian mengembalikan false bahwa proses upload gagal. Setelah itu, fungsi memeriksa ekstensi file gambar yang diunggah untuk memastikan bahwa itu sesuai dengan ekstensi yang diizinkan. Jika ekstensi tidak sesuai, fungsi kembali menampilkan pesan peringatan dan mengembalikan false. Langkah berikutnya adalah memeriksa ukuran file gambar yang diunggah. Jika ukuran melebihi batas yang ditentukan (1000000 byte atau sekitar 1MB), fungsi kembali menampilkan pesan peringatan dan mengembalikan false. Jika semua validasi berhasil, fungsi mengganti nama file gambar agar tidak ada duplikasi dengan menggunakan fungsi **uniqid()** dan ekstensi file yang telah divalidasi sebelumnya. Kemudian, file gambar yang diunggah dipindahkan ke folder tujuan yang ditentukan dengan menggunakan fungsi **move_uploaded_file()**. Terakhir, fungsi mengembalikan nama file gambar baru yang telah diunggah yang akan digunakan oleh fungsi **tambah()** atau **edit()** untuk menyimpan nama file gambar dalam database, atau jika proses upload gagal, fungsi false akan mengembalikan ke fungsi pemanggil bahwa proses upload gagal.

- **Function edit(\$data)**



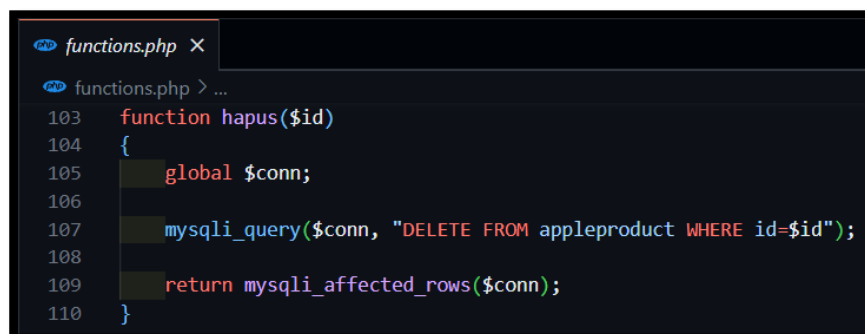
```
functions.php X
functions.php > PHP > edit()
70 function edit($data)
71 {
72     global $conn;
73     $id = $data["id"];
74
75     // Mengambil nilai dari form edit dan membersihkannya dari karakter khusus menggunakan htmlspecialchars()
76     $code = htmlspecialchars($data["code"]);
77     $product_name = htmlspecialchars($data["product_name"]);
78     $product_type = htmlspecialchars($data["product_type"]);
79     $price = htmlspecialchars($data["price"]);
80
81     $gambarAwal = $data["gambarAwal"];
82
83     if ($_FILES['image']['error'] === 4) {
84         $image = $gambarAwal;
85     } else {
86         $image = upload();
87     }
88
89     $query = "UPDATE appleproduct SET
90         code='$code',
91         product_name='$product_name',
92         product_type='$product_type',
93         price = '$price',
94         image='$image'
95         WHERE id=$id
96     ";
97
98     mysqli_query($conn, $query);
99
100     return mysqli_affected_rows($conn);
101 }
```

Gambar 11

Fungsi **edit(\$data)** digunakan untuk memperbarui produk apple yang sudah ada dalam database. Ketika fungsi ini dipanggil, langkah-langkahnya dimulai dengan ekstraksi

ID produk apple yang akan diedit dari parameter **\$data**. Selanjutnya, nilai-nilai dari form edit juga diekstraksi dan dibersihkan dari karakter khusus dengan menggunakan fungsi **htmlspecialchars()**, sehingga memastikan bahwa data yang dimasukkan aman dari serangan XSS. Setelah nilai-nilai yang dibersihkan, seperti kode, nama produk, macam produk, dan harga telah disiapkan, langkah berikutnya adalah memeriksa apakah ada gambar yang diunggah untuk produk apple yang akan diedit. Jika tidak ada gambar yang diunggah (dinyatakan dengan nilai **\$_FILES['gambar']['error'] === 4**), maka nama gambar tetap menggunakan gambar yang sudah ada sebelumnya. Namun, jika ada gambar yang diunggah, fungsi **upload()** dipanggil untuk mengelola proses upload gambar baru. Setelah gambar telah diubah, query SQL disusun untuk memperbarui data produk apple di database. Query ini mencakup perubahan nilai-nilai seperti kode, nama produk, macam produk, harga, dan nama gambar (jika ada perubahan).

- **Function hapus(\$id)**



```
functions.php x
functions.php > ...
103 function hapus($id)
104 {
105     global $conn;
106
107     mysqli_query($conn, "DELETE FROM appleproduct WHERE id=$id");
108
109     return mysqli_affected_rows($conn);
110 }
```

Gambar 12

Fungsi **hapus(\$id)** digunakan untuk menghapus produk apple dari database berdasarkan ID yang diberikan. Ketika fungsi ini dipanggil, ia menggunakan parameter **\$id** untuk menentukan produk apple mana yang akan dihapus. Kemudian, dengan menggunakan query **SQL DELETE**, produk apple yang memiliki ID yang sesuai dihapus dari tabel **appleproduct** dalam database. Setelah query dijalankan menggunakan **mysqli_query()**, fungsi mengembalikan jumlah baris yang dipengaruhi oleh operasi query tersebut.

- **Function search(\$keyword)**

```

functions.php > ...
112 function search($keyword){
113     // Membuat query SQL untuk mencari data berdasarkan kata kunci yang dimasukkan
114     $query = "SELECT * FROM appleproduct WHERE
115     product_name LIKE '%$keyword%' OR
116     product_type LIKE '%$keyword%' OR
117     code LIKE '%$keyword%'
118     ";
119
120     return query($query); // Menjalankan fungsi query() dengan parameter query yang sudah dibuat
121 }
122
123

```

Gambar 13

Fungsi **search(\$keyword)** digunakan untuk mencari data produk apple berdasarkan kata kunci yang dimasukkan oleh pengguna. Ketika fungsi ini dipanggil, ia membentuk sebuah query SQL yang mencari data dalam tabel appleproduct yang memiliki kesamaan dengan kata kunci yang diberikan dalam nama produk, macam produk, atau kode. Kata kunci ini diperoleh dari parameter **\$keyword**. Setelah query dibuat, fungsi memanggil fungsi **query()** dengan parameter query yang sudah dibuat untuk mengeksekusi query tersebut. Hasil pencarian, berupa array yang berisi data produk apple yang sesuai dengan kata kunci, kemudian dikembalikan untuk digunakan.

F. Penjelasan Database “dbapplestore”

	id	product_name	product_type	code	price	image
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	15	Iphone 15	Handphone	hp01	14249000	662e54079c924.jpeg
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	16	Iphone 14 Pro Max	Handphone	hp02	21900000	662e54135ef23.jpg
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	17	Macbook Pro M2	Laptop	laptop01	15499000	662e5432f647.jpg
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	18	Ipad Pro 12.9 Inch 5TH Gen	Ipad	ipad01	10900000	662e5457c5ebb.jpg
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	19	Apple Watch Series 9	Smart Watch	watch01	6999000	662e547aeb56e.jpeg
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	20	Airpods Pro 2nd Gen	Airpods	airpods01	3082000	662e548d10e07.png
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	21	Apple Vision Pro	Vision	vision01	57999000	662e54a721988.jpg

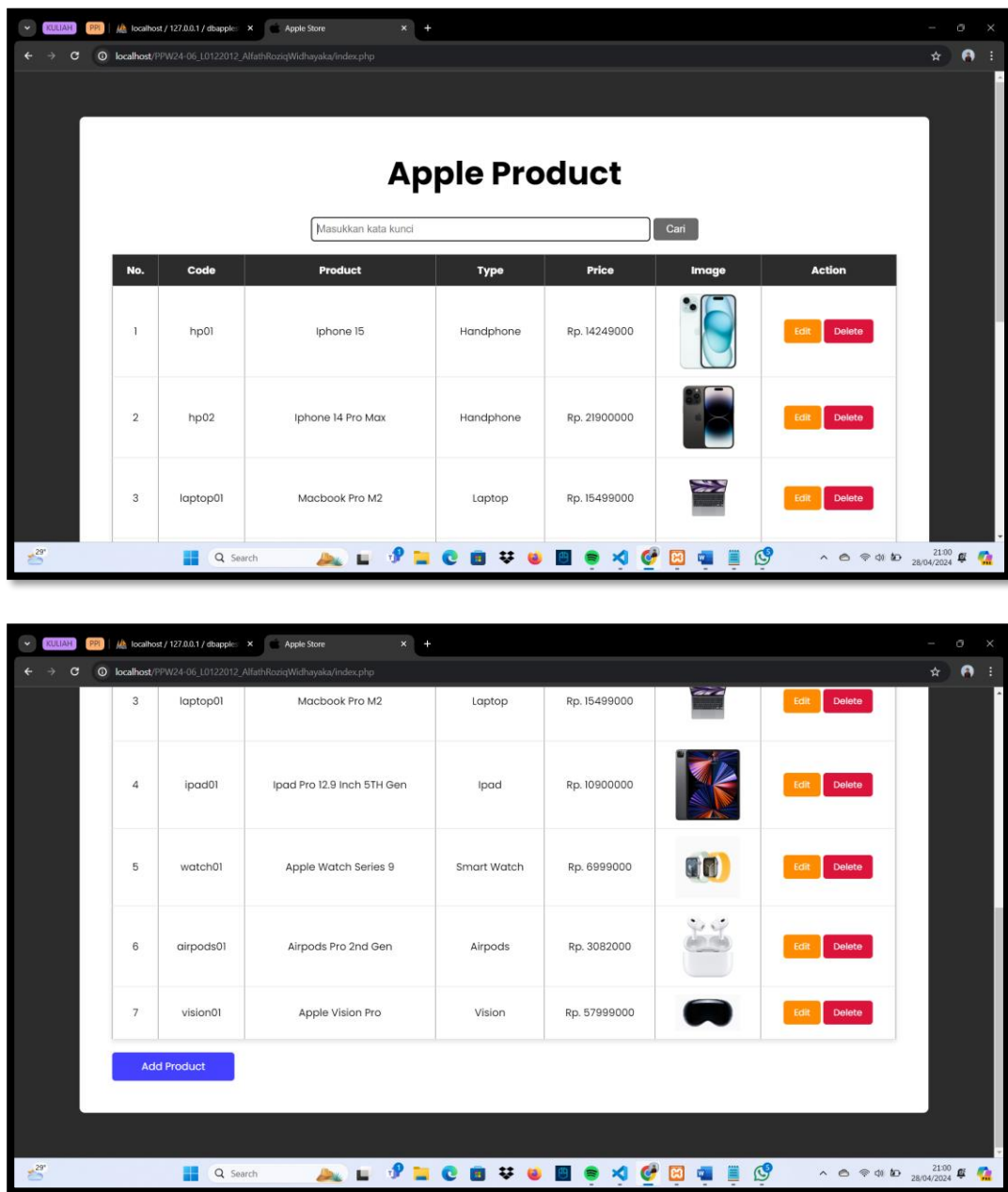
Gambar 14

Database di atas merupakan struktur tabel dan data menggunakan phpMyAdmin. SQL menciptakan sebuah database bernama dbapplestore dan tabel appleproduct di dalamnya. Tabel ini terdiri dari beberapa kolom seperti id, product_name, product_type, code, price,

dan image. Setiap kolom memiliki tipe data yang sesuai, seperti varchar untuk teks dan int untuk bilangan bulat, dan beberapa di antaranya dinyatakan sebagai NOT NULL, yang berarti kolom tersebut harus diisi dengan nilai yang valid. Data kemudian dimasukkan ke dalam tabel appleproduct menggunakan perintah INSERT INTO, yang mencakup beberapa entri produk apple beserta atributnya seperti nama produk, macam produk, kode, harga, dan nama gambar.

BAB II

A. Tampilan halaman utama (index.php)

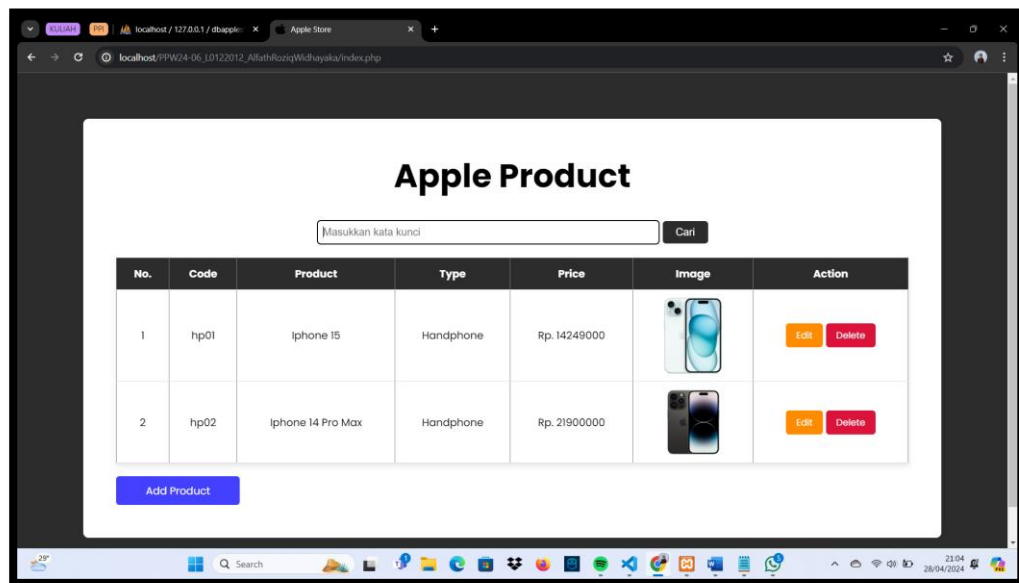


Gambar 15

Gambar di atas merupakan halaman utama yaitu **index.php**. Di mana pada halaman ini menampilkan data yang dimuat dari database **dbappleproduct**. Pada halaman ini terdapat judul, display pencarian menurut kata kunci yang dimasukkan pengguna. Kemudian terdapat tabel yang isinya dimuat dari database, di mana setiap barisnya terdapat fungsi edit serta hapus data. Selain itu juga ada aksi untuk menambah data, ini digunakan apabila ingin menambah produk apple baru.

B. Aksi Halaman Index

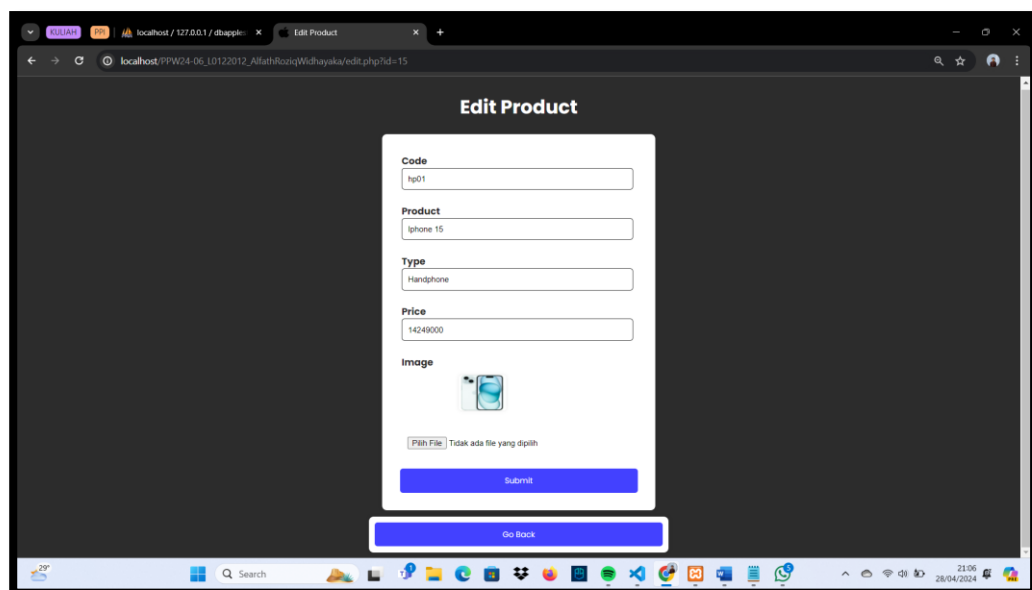
- Search Bar



Gambar 16

Gambar di atas merupakan contoh halaman yang akan ditampilkan apabila pengguna mencari produk apple dengan kata kunci. Kata kunci yang dapat dimasukkan yaitu kode, nama produk, dan macam produk. Pada contoh di atas, apabila pengguna memasukkan kata kunci “hp” maka akan muncul produk apple dengan macam ”hp”.

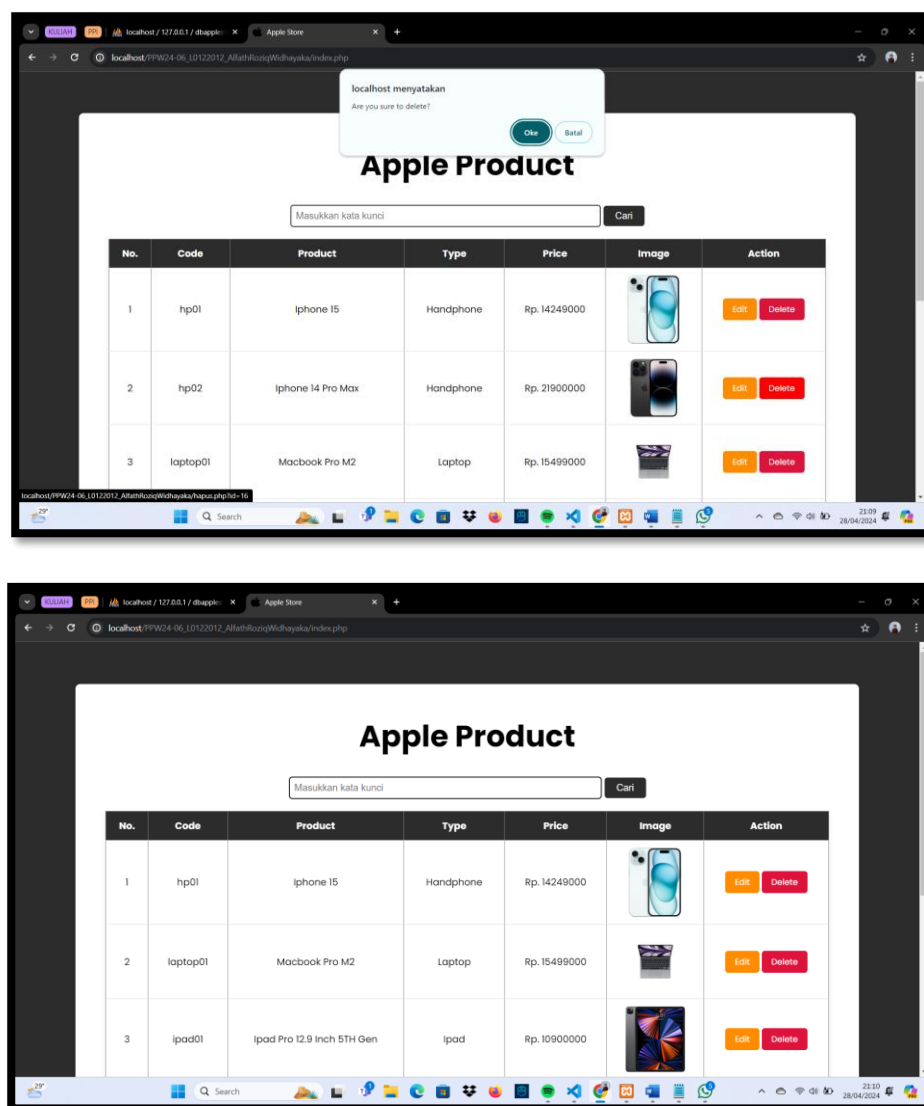
- Edit Data



Gambar 17

Gambar di atas merupakan halaman yang akan ditampilkan apabila pengguna mengklik aksi **Edit** di halaman utama. Di mana pada halaman ini menampilkan data yang ada sebelumnya dan bisa diubah. Pada halaman atau produk apple ini dapat mengubah kode, nama produk, macam produk, harga, bahkan gambar dari produk apple sebelumnya. Jika selesai melakukan edit data, pengguna harus melakukan submit agar data bisa diperbarui di halaman utama dan database. Jika edit data berhasil maka akan menampilkan **alert “Success to edit data!”**, namun jika gagal maka akan menampilkan **alert “Failed to edit data”**. Disini pengguna juga dapat kembali ke **index.php** jika tidak jadi untuk edit produk.

▪ Delete Data

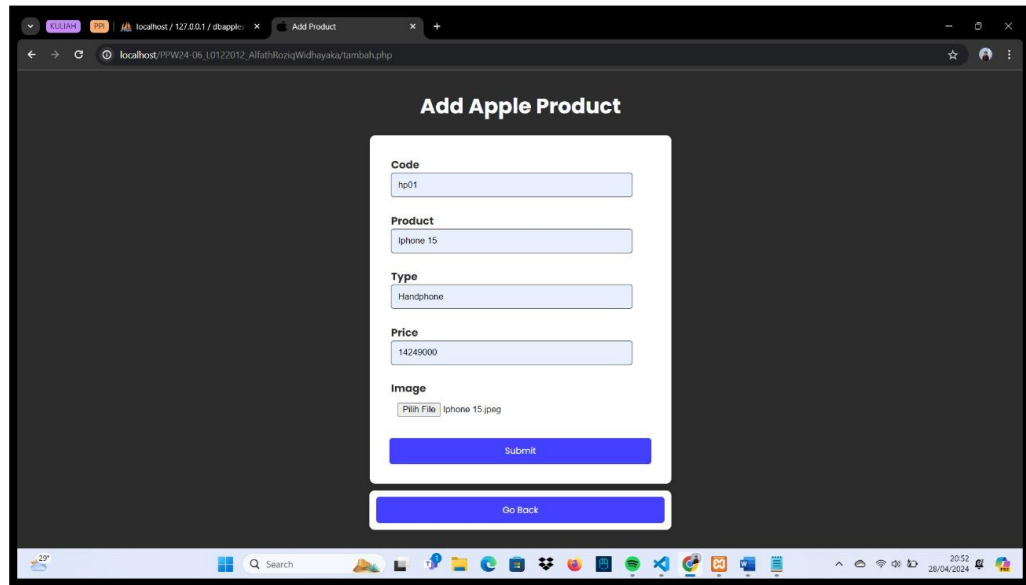


Gambar 18

Gambar di atas merupakan halaman apabila pengguna mengklik aksi **Delete** halaman utama. Di mana akan memunculkan alert yang berisi **“Are you sure to delete?”** jika

pengguna klik **Oke** maka produk apple akan dihapus dari halaman maupun database. Jika hapus data berhasil maka akan menampilkan alert **“Success to delete!”**, namun jika gagal maka akan menampilkan alert **“Failed to delete!”**. Pada contoh diatas saya mencoba untuk menghapus produk apple yang memiliki kode **”hp02”** dan untuk gambar kedua penghapusan produk apple tersebut berhasil dihapus.

- **Add Product**

A screenshot of a web browser displaying a form titled "Add Apple Product". The form is centered on a dark background. It contains several input fields: "Code" with the value "hp01", "Product" with the value "Iphone 15", "Type" with the value "Handphone", and "Price" with the value "14249000". Below these is an "Image" section with a "Pilih File" button and the text "Iphone 15.jpeg". At the bottom of the form are two blue buttons: "Submit" and "Go Back". The browser's address bar shows the URL "localhost/PPW24_06_10122012_AlfathRozqWidhayaka/tambah.php". The Windows taskbar is visible at the bottom of the screen.

Gambar 18

Gambar di atas merupakan halaman apabila pengguna mengklik aksi **Add Product** pada halaman utama. Di mana pada halaman ini menampilkan formulir yang perlu diinputkan isinya apabila ingin menambah data (tidak boleh kosong). Pada halaman ini juga dapat memasukkan gambar. Jika selesai mengisi data, pengguna harus melakukan submit agar data bisa ditambahkan di halaman utama maupun di database. Jika tambah data berhasil maka akan menampilkan alert **“Success to Add Product!”**, namun jika gagal maka akan menampilkan alert **“Failed to Add Product!”**.

BAB III

Dari praktikum yang telah dilakukan, terlihat bahwa halaman utama telah dikembangkan dengan baik dan memenuhi beberapa kriteria penting:

- Tidak menggunakan framework PHP apapun dan halaman tersebut menggunakan PHP Native tanpa framework, sesuai permintaan.
- Halaman yang saya buat memiliki fitur CRUD (Create, Read, Update, Delete) yang merupakan operasi dasar untuk manajemen data. Pengguna dapat menambah, mengedit, dan menghapus produk apple dari database.
- Terdapat fitur untuk mengunggah gambar pada form tambah data. Ini memenuhi syarat tambahan yang mengharuskan mahasiswa untuk menerapkan file upload pada salah satu inputan.

Secara keseluruhan, halaman utama ini memberikan fungsionalitas lengkap untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada produk apple. Pengguna dapat membuat, membaca, memperbarui, dan menghapus data dengan mudah dan intuitif. Dengan demikian, halaman utama telah memenuhi semua kriteria yang diminta dan memberikan pengguna fitur yang lengkap dalam **Apple Store**.