

# Laporan Praktikum CRUD Dan Login PHP Native



Disusun oleh:

1. Naila kaisya majid (22)
2. Viola Ananda nova arini (34)
3. Zarah awal masruroh (35)

REKAYASA PERANGKAT LUNAK  
SMKTEXMACO SEMARANG  
2024/2025

## A.Database.php

```
<?php
class Database {
    private $host = "localhost";
    private $uname = "root";
    private $pass = "";
    private $db = "oop2";

    public $connection;

    function __getConnection() {
        $this->connection = null;
        try {
            $this->connection = new PDO("mysql:host=" . $this->host . ";dbname="
            . $this->db, $this->uname, $this->pass);
            $this->connection->setAttribute(PDO::ATTR_ERRMODE,
            PDO::ERRMODE_EXCEPTION);
        } catch(PDOException $exception) {
            echo"connection error:" . $exception->getMessage();
        }
        return $this->connection;
    }
}
```

Penjelasan:

- 1.class Database : Kode ini mendefinisikan sebuah class bernama Database, yang nantinya dapat digunakan untuk membuat objek dengan properti dan metode untuk menghubungkan aplikasi ke database
2. Properti Private : Properti host, uname, pass, dan db adalah properti yang hanya bisa diakses dalam class Database.
- 3.Properti Public : Properti connection dideklarasikan sebagai public sehingga bisa diakses dari luar class.
- 4.Metode getConnection : Metode ini bertugas membuat koneksi ke database menggunakan informasi yang diberikan dalam properti.

## 5. Blok try-catch untuk membuat koneksi :

- try-catch adalah blok error-handling. Dalam blok try, script mencoba membuat koneksi ke database menggunakan PDO.
- PDO adalah kelas PHP yang memungkinkan koneksi ke berbagai database secara aman dan efisien.
- Jika koneksi berhasil, connection akan berisi objek PDO untuk koneksi database.
- Jika terjadi kesalahan, blok catch akan menangkap error (PDOException) dan menampilkan pesan error dengan getMessage().

## B.tampil.php

```
<?php
include './config/database.php';
include './classes/minuman2.php';
$db = new database();
$minuman2 = new minuman2($db -> __getConnection());

session_start();
if(!isset($_SESSION['username'])) {
    echo "<script>alert('anda belum login. Silahkan login terlebih dahulu.');"
    window.location = 'index.php'; </script>";
    exit();
}
?>
<h1>CRUD OOP PHP</h1>
<link rel="stylesheet" href="./css/tampil.css">
<a href="input.php">Input Data</a>
<a href="logout.php">logout</a>
<table border="3">
    <tr>
        <th>Id</th>
        <th>nama</th>
        <th>rasa</th>
        <th>harga</th>
        <th>Aksi</th>
    </tr>
    <?php
    $id = 1;
    foreach($minuman2->tampil_data() as $x) {
        ?>
        <tr>
```

```

        <td><?php echo $id++; ?></td>
        <td><?php echo htmlspecialchars($x['nama']); ?></td>
        <td><?php echo htmlspecialchars($x['rasa']); ?></td>
        <td><?php echo htmlspecialchars($x['harga']); ?></td>
        <td>
            <a href="edit.php?id=<?php echo $x['id']; ?>&aksi=edit"
onclick="return confirm('Apakah Anda yakin ingin mengedit data ini?');">Edit</a>
            <a href="delete.php?id=<?php echo $x['id']; ?>&aksi=delete"
onclick="return confirm('Apakah Anda yakin ingin menghapus data
ini?');">Delete</a>
        </td>
    </tr>
<?php
}
?>
</table>

```

Penjelasan :

- 1.Include : Untuk mengimpor file
- 2.new database () : Membuat objek baru dari class Database.
- 3.\$db ->\_\_getConnection(): Memanggil metode koneksi getConnection untuk terhubung ke database dan meneruskannya sebagai parameter ke objek minuman2
4. session\_start(): Memulai session PHP untuk mengakses data session
5. if (!isset(\$\_SESSION['username'])) : Mengecek apakah pengguna telah login. Jika belum, muncul pesan "anda belum login" dan diarahkan ke halaman login (index.php)
6. <link rel="stylesheet" href="/css/tampil.css"> : Menghubungkan halaman dengan file CSS eksternal untuk gaya tampilan.
- 7.Tabel header :  
Membuat tabel untuk menampilkan data dari database dengan kolom **Id**, **nama**, **rasa**, **harga**, dan **Aksi**.
8. \$id = 1: Menginisialisasi nomor urut id untuk ditampilkan.
9. htmlspecialchars(): Mengamankan data yang ditampilkan dari potensi serangan XSS dengan mengubah karakter khusus menjadi entitas HTML.

## C.input.php

```
<?php
// Konfigurasi database
$host = 'localhost';
$username = 'root';
$password = '';
$database = 'oop2';

$conn = new mysqli($host, $username, $password, $database);

if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST['nama'];
    $rasa = $_POST['rasa'];
    $harga = $_POST['harga'];

    $sql="INSERT INTO minuman2 (nama, rasa, harga) VALUES ('$nama', '$rasa', '$harga')"; // echo $sql;
    $stmt = $conn->prepare($sql);
    //$stmt->bind_param("ssis", $nama, $rasa, $harga);

    if ($stmt->execute()) {
        echo "Data berhasil ditambahkan!";
    } else {
        echo "Error: " . $stmt->error;
    }

    $stmt->close();
}

$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Input Data minuman</title>
    <link rel="stylesheet" href="../css/input.css">
</head>
```

```

<body>
  <h1>Input Data minuman</h1>
  <form method="POST" action="">
    <label for="nama">nama:</label><br />
    <input type="text" id="nama" name="nama" required><br /><br />

    <label for="rasa">rasa:</label><br />
    <input type="text" id="rasa" name="rasa" required><br /><br />

    <label for="harga">harga:</label><br />
    <input type="text" id="harga" name="harga" required><br /><br />

    <input type="submit" value="Tambahkan minuman">
  </form>
</body>
</html>

```

Penjelasan :

1. \$\_SERVER["REQUEST\_METHOD"] == "POST" : Memeriksa apakah permintaan yang diterima adalah POST, yang berarti data form telah dikirimkan.
2. INSERT INTO minuman2 (nama, rasa, harga) VALUES ('\$nama', '\$rasa', '\$harga');  
Membuat perintah SQL untuk memasukkan data nama, rasa, dan harga ke tabel minuman2.
3. prepare(\$sql): Membuat pernyataan SQL yang siap dijalankan dengan variabel.
4. \$stmt->execute(): Menjalankan perintah SQL. Jika berhasil, menampilkan pesan **"Data berhasil ditambahkan!"**
5. <!DOCTYPE html>, <html lang="en">, <head>, <meta>, dan <title> :  
Menyusun struktur HTML standar.
6. method="POST" : Mengirimkan data menggunakan metode POST.
7. action="" : Mengirimkan data ke halaman yang sama setelah form disubmit.
8. required : Menjadikan input wajib diisi sebelum form dapat dikirimkan.

D.edit.php

```
<?php
```

```

include 'database.php';
$db = new database();

// Check if ID is set in the URL
if (isset($_GET['id'])) {
    $id = $_GET['id'];

    // Fetch existing minuman2 data
    $minuman2 = $db->get_minuman2_by_id($id); // Ensure this method is defined in
your database class

    if (!$minuman2) {
        die("minuman2 tidak ditemukan.");
    }
} else {
    die("ID minuman2 tidak ditentukan.");
}

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST['nama'];
    $rasa = $_POST['rasa'];
    $harga = $_POST['harga'];

    // Update minuman2 data
    if ($db->update_minuman2($id, $nama, $rasa, $harga)) {
        header("Location: tampil.php"); // Redirect back to the list
        exit();
    } else {
        echo "Error: Gagal mengupdate data.";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Data minuman2</title>
    <link rel="stylesheet" href="edit.css">
</head>
<body>
    <h1>Edit Data minuman2</h1>
    <form method="POST" action="">

```

```

        <label for="nama">nama:</label><br />
        <input type="text" id="nama" name="nama" value="<?php echo
htmlspecialchars($minuman2['nama']); ?>" required><br /><br />

        <label for="rasa">rasa:</label><br />
        <input type="text" id="rasa" name="rasa" value="<?php echo
htmlspecialchars($minuman2['rasa']); ?>" required><br /><br />

        <label for="harga">harga:</label><br />
        <input type="text" id="harga" name="harga" value="<?php echo
htmlspecialchars($minuman2['harga']); ?>" required><br /><br />

        <input type="submit" value="Update minuman2">
    </form>

    <div class="ta">
        <a href="tampil.php">Kembali ke Daftar minuman2</a>
    </div>
</body>
</html>

```

Penjelasan:

1. `if (isset($_GET['id']))`: Mengecek apakah id telah diberikan melalui URL (untuk data yang akan diedit)
2. `if ($_SERVER["REQUEST_METHOD"] == "POST")`: Mengecek apakah request adalah POST, yang berarti form telah disubmit.
3. `header("Location: tampil.php")`:: Jika pembaruan berhasil, pengguna diarahkan kembali ke halaman `tampil.php` untuk melihat daftar data.
4. Terdapat tiga input yaitu nama, rasa, dan harga, masing-masing menggunakan `value="<?php echo htmlspecialchars($minuman2['nama']); ?>"`, sehingga nilai saat ini dari database ditampilkan sebagai default.
5. `htmlspecialchars()`: Digunakan untuk mengamankan data dari karakter khusus agar tidak dieksekusi sebagai kode HTML.



## E.delete.php

```
<?php
include 'database.php';
$db = new database();

if (isset($_GET['id'])) {
    $id = $_GET['id'];

    // Call the method to delete the data
    if ($db->delete_data($id)) {
        // Redirect back to the main page with a success message
        header("Location: tampil.php?message=Data berhasil dihapus");
        exit();
    } else {
        // Redirect back with an error message
        header("Location: tampil.php?message=Gagal menghapus data");
        exit();
    }
} else {
    // Redirect back with an error message if no ID is set
    header("Location: tampil.php?message=ID tidak valid");
    exit();
}
?>
```

### Penjelasan:

1. `if (isset($_GET['id']))`: Mengecek apakah ada parameter id yang dikirimkan melalui URL, yang menunjukkan item yang ingin dihapus.
2. `$id = $_GET['id'];`: Menyimpan nilai id yang diterima dalam variabel \$id.
3. `if ($db->delete_data($id))`: Memanggil metode `delete_data` dari class `database` dengan parameter \$id untuk menghapus data. Pastikan bahwa metode `delete_data()` sudah didefinisikan di class `database` agar data dapat dihapus.
4. `header("Location: tampil.php?message=Data berhasil dihapus");`: Jika penghapusan berhasil, pengguna akan diarahkan kembali ke halaman `tampil.php` dengan pesan sukses **"Data berhasil dihapus"**.
5. `header("Location: tampil.php?message=Gagal menghapus data");`: Jika penghapusan gagal, pengguna akan diarahkan kembali ke halaman `tampil.php` dengan pesan **"Gagal menghapus data"**.



```

        <input type="text" name="username"
        class="form-control"required>
    </div>
    <div class="form-group">
        <label>password</label>
        <input type="password" name="password"
        class="form-control"required>
    </div>
    <button type="submit" class="btn btn-primary btn-block">Login</button>
    <a href="register.php" class="btn btn-warning btn-block">Register</a>

</form>
</div>
</div>
</div>
</body>
</html>

```

Penjelasan:

1. `if($_SERVER["REQUEST_METHOD"] == "POST")`: Mengecek apakah request yang diterima adalah POST, yang berarti form telah disubmit.
2. `$username = $_POST['username'];` dan `$password = $_POST['password'];`: Menyimpan username dan password yang dimasukkan pengguna.
3. `if($user->login($username, $password))`: Memanggil metode login dari class user, yang memeriksa apakah username dan password valid. Jika valid, pengguna diarahkan ke halaman tampil.php dengan perintah `header("Location: tampil.php")`;
4. Jika login gagal, variabel `$error_message` diisi dengan pesan "username atau password salah!" untuk ditampilkan di halaman.
5. `<?php if(isset($error_message)): ?>`: Mengecek apakah variabel `$error_message` telah diatur (jika login gagal). Jika ya, menampilkan pesan error.
6. `<div class="alert alert-danger">`: Menggunakan komponen Bootstrap untuk menampilkan pesan error dalam kotak merah.
7. `<form action="" method="POST">`: Form menggunakan metode POST dan dikirim ke halaman yang sama.
8. `<input type="text" name="username" class="form-control" required>`: Input untuk username dengan kelas Bootstrap form-control.
9. `<input type="password" name="password" class="form-control" required>`: Input untuk password, ditetapkan sebagai password untuk menyembunyikan karakter yang dimasukkan.
10. `<button type="submit" class="btn btn-primary btn-block">Login</button>`: Tombol untuk mengirim form dengan gaya Bootstrap.
11. `<a href="register.php" class="btn btn-warning btn-block">Register</a>`: Tombol untuk mengarahkan pengguna ke halaman register.php jika belum memiliki akun.

G.user.php

```
<?php
//class/user.php
require_once './config/database.php';

class user{
    private $connection;
    private $table_name="users";
    public function __construct($db) {
        $this->connection = $db;
    }
    /**
     * @param string $username
     * @param string $password
     * @return boolean
     */
    public function login($username, $password) { //typo awalnya usetname
        $query ="SELECT * FROM " . $this -> table_name . " WHERE username =
:username LIMIT 1"; //typo awalnya username
        $stmt = $this -> connection -> prepare($query);
        $stmt -> bindParam(":username", $username);

        $stmt -> execute();

        if($stmt -> rowCount() . 0){
            $row = $stmt -> fetch (PDO :: FETCH_ASSOC);
            if (password_verify($password,$row['password'])) {
                session_start();
                $_SESSION['username'] = $username;
                return true;
            }
        }
        return false;
    }
    //fungsi untuk menambahkan user baru
    public function register($username, $password){
        $query = "INSERT INTO " . $this -> table_name . " (username,password)
VALUES (:username, :password)"; //typo awalnya username
        $stmt = $this-> connection -> prepare($query);
        //enkripsi password sebelum disimpan ke database
        $hashed_password = password_hash($password, PASSWORD_DEFAULT);

        $stmt-> bindparam(":username" , $username);
```

```

        $stmt -> bindparam (":password", $hashed_password);

        if($stmt->execute()){
            return true;
        }
        return false;
    }
}
?>

```

Penjelasan:

1. \$stmt->execute();: Menjalankan query di database.
2. if(\$stmt->rowCount() > 0): Memeriksa apakah ada hasil yang cocok dengan username.
3. \$row = \$stmt->fetch(PDO::FETCH\_ASSOC);: Mengambil data hasil query dalam bentuk array asosiatif.
4. if (password\_verify(\$password, \$row['password'])): Memverifikasi password input pengguna dengan password hash yang tersimpan dalam database.
5. session\_start();: Memulai sesi untuk menyimpan data pengguna yang telah login.
6. \$\_SESSION['username'] = \$username;: Menyimpan username ke dalam sesi.
7. return true;: Mengembalikan nilai true jika login berhasil.
8. return false;: Mengembalikan false jika login gagal.

H.minuman.php

```

<?php
require_once './config/database.php';

class minuman2 {
    private $connection;
    private $table_name = "minuman2";
    public function __construct($db) {
        $this->connection = $db;
    }

    private function get_minuman2_by_id($id) {
        $stmt = $this->connection->prepare("SELECT * FROM minuman2 WHERE id = ?");
        $stmt->bind_param("i", $id);
        $stmt->execute();
        return $stmt->get_result()->fetch_assoc();
    }
}

```

```

private function delete_data($id) {
    $stmt = $this->connection->prepare("DELETE FROM minuman2 WHERE id = ?");
    $stmt->bind_param("i", $id);
    return $stmt->execute();
}

private function update_minuman2($id, $nama, $rasa, $harga) {
    //$stmt = $this->connection->prepare("UPDATE minuman2 SET nama=?, rasa=?,
harga=? WHERE Id=?");
    //$stmt->bind_param("i", $id, $nama, $rasa, $harga);
    $stmt = $this->connection->prepare("UPDATE minuman2 SET nama=?, rasa=?,
harga=? WHERE id=?");
    $stmt->bind_param("ssdi", $nama, $rasa, $harga, $id);

    return $stmt->execute();
}

function tampil_data() {
    $hasil = [];
    $data = $this->connection->query("SELECT * FROM minuman2");

    while ($d = $data->fetch(PDO::FETCH_ASSOC)) {
        $hasil[] = $d;
    }

    return $hasil;
}
}
?>

```

Penjelasan:

Kelas minuman2 ini menyediakan fungsi untuk:

1. Mengambil data minuman berdasarkan ID dengan get\_minuman2\_by\_id.
2. Menghapus data minuman berdasarkan ID dengan delete\_data.
3. Memperbarui data minuman berdasarkan ID dengan update\_minuman2.
4. Menampilkan seluruh data minuman dengan tampil\_data

I.register.php

```

<?php
// register.php

```

```

require_once 'config/database.php';
require_once 'classes/user.php';

$database = new Database();
$db = $database->__getConnection();
$user = new User($db);

if($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = $_POST['password'];

    if($user->register($username, $password)) {
        $success_message = "Register berhasil silahkan login.";
    } else{
        $error_message = "Register gagal. Username mungkin sudah digunakan";
    }
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>login</title>
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-md-4">
                <h3 class="text-center">login</h3>
                <?php if(isset($error_message)): ?>
                <div class="alert alert-danger">
                    <?php echo $error_message?>
                </div>
                <?php endif; ?>
            <form action="" method="POST">
                <div class="form-group">
                    <label>username</label>
                    <input type="text" name="username"
class="form-control"required>
                </div>
                <div class="form-group">

```

```

    <label>password</label>
    <input type="password" name="password"
    class="form-control"required>
  </div>
  <button type="submit" class="btn btn-primary btn-block">Daftar</button>
</form>
  <a class="text-center mt-3">Sudah punya akun?<a href="index.php">Login di
sini</a></p>
</div>
</div>
</div>
</body>
</html>

```

Penjelasan:

Kode ini memungkinkan pengguna mendaftar akun baru dengan menyimpan data username dan password (terenkripsi) di database. Jika berhasil, pengguna mendapatkan pesan konfirmasi, dan jika gagal (misalnya, karena nama pengguna sudah digunakan), mereka akan diberi pesan kesalahan

## J.generate.php

```

<?php
echo password_hash("galih12345678", PASSWORD_DEFAULT);

```

Penjelasan:

Penggunaan password\_hash memastikan kata sandi "galih12345678" aman sebelum disimpan ke database, sehingga mengurangi risiko kebocoran data jika ada pelanggaran keamanan.

## K.logout.php

```

<?php
session_start();
session_unset();
session_destroy();

// Redirect ke halaman login setelah logout
header("Location: index.php");
exit();

```

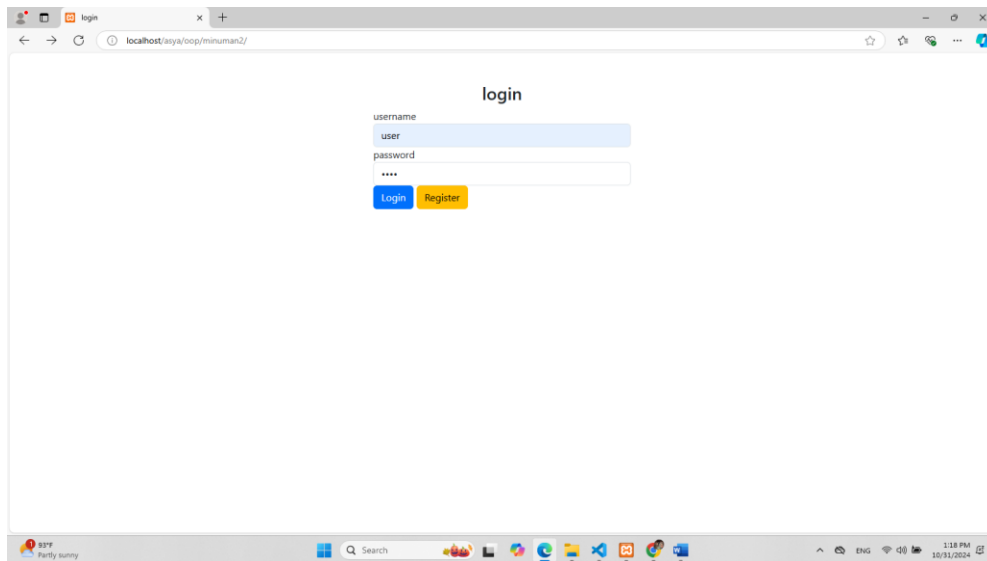


Penjelasan:

Kode ini efektif untuk mengakhiri sesi pengguna dan mengembalikannya ke halaman login, sehingga memastikan bahwa pengguna harus login kembali untuk mengakses aplikasi, yang sering kali dilakukan saat logout.

## Hasil:

### -Halaman login



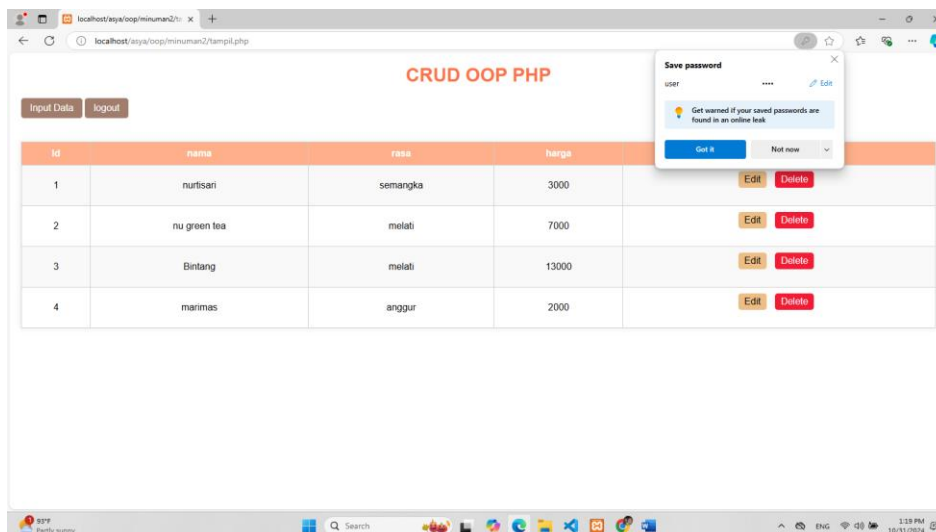
login

username  
user

password  
\*\*\*\*

Login Register

### -Tampilan



CRUD OOP PHP

Input Data logout

id	nama	rasa	harga	
1	nurtisari	semangka	3000	Edit Delete
2	nu green tea	melati	7000	Edit Delete
3	Bintang	melati	13000	Edit Delete
4	marimas	anggur	2000	Edit Delete

## -Edit

localhost/asya/oop/minuman2/tampil.php

localhost says  
Apakah Anda yakin ingin mengedit data ini?

Input Data logout

Id	nama	rasa	harga	Aksi
1	nurtisari	semangka	3000	<a href="#">Edit</a> <a href="#">Delete</a>
2	nu green tea	melati	7000	<a href="#">Edit</a> <a href="#">Delete</a>
3	Bintang	melati	13000	<a href="#">Edit</a> <a href="#">Delete</a>
4	marimas	anggur	2000	<a href="#">Edit</a> <a href="#">Delete</a>

localhost/asya/oop/minuman2/edit.php?id=4&aksi=edit

localhost/asya/oop/minuman2/edit.php?id=4&aksi=edit

### Edit Data minuman

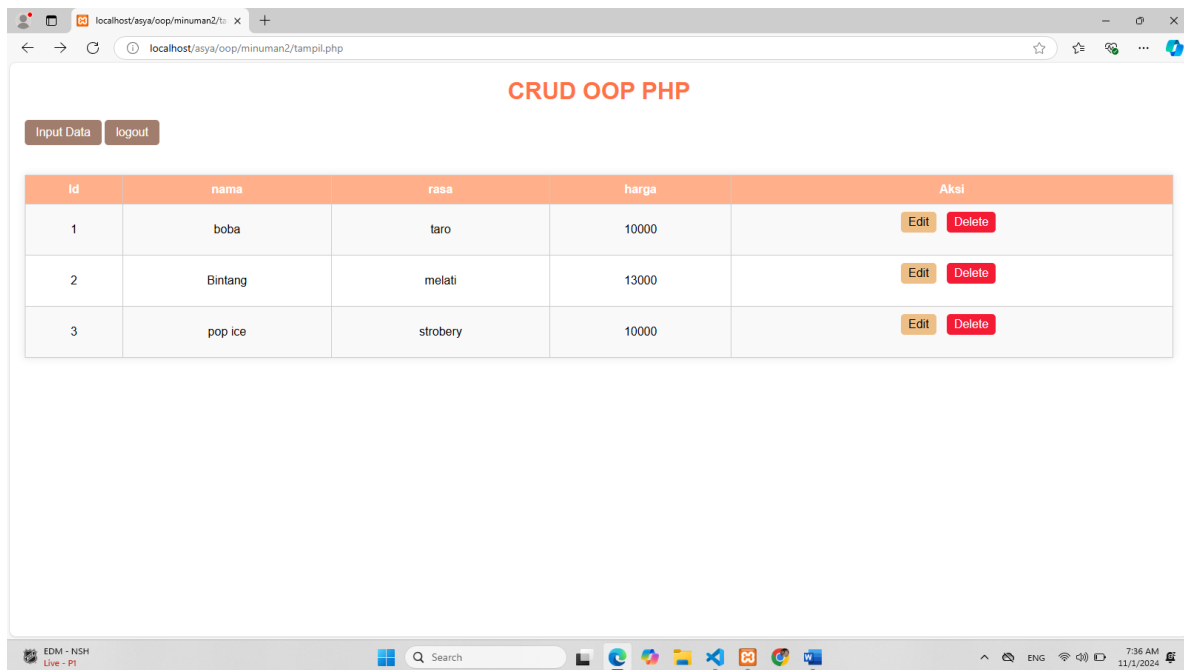
nama:  
boba

rasa:  
taro

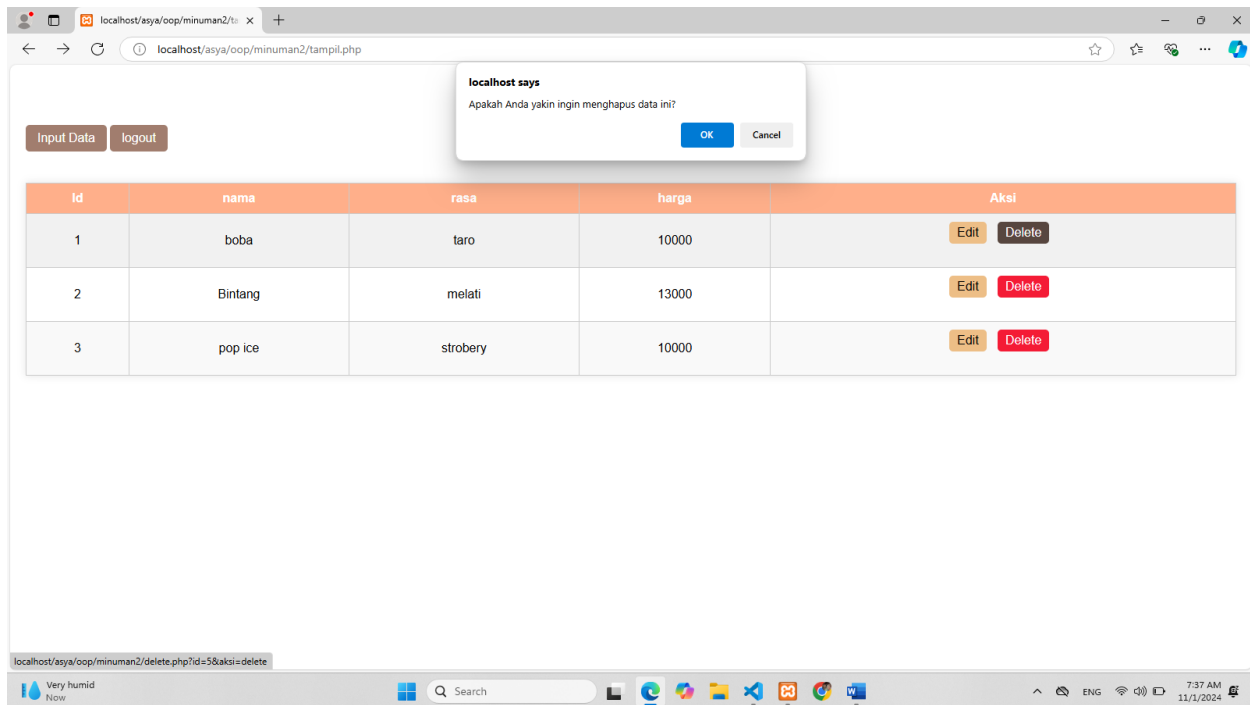
harga:  
10000

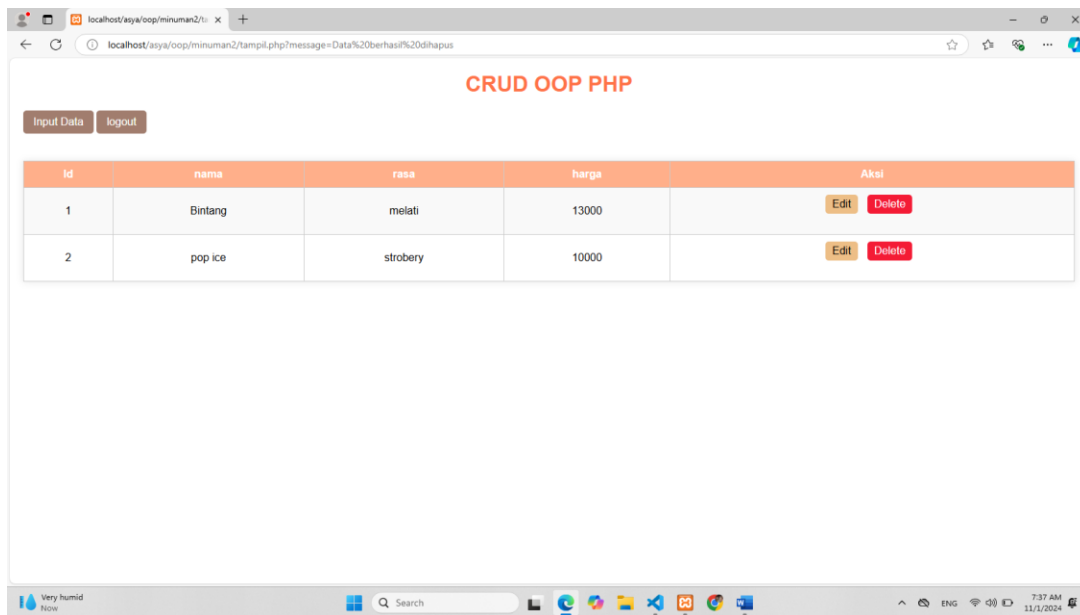
Update minuman

[Kembali ke Daftar minuman2](#)

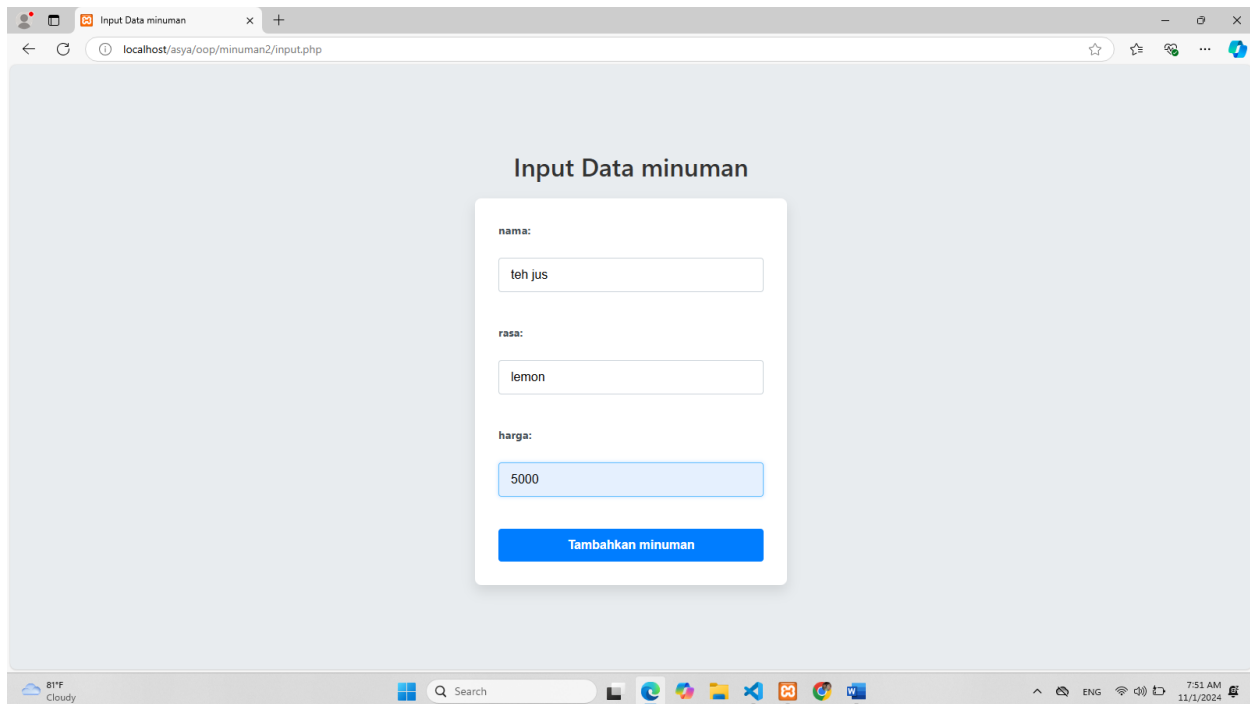


## -Delete





## -Input data



localhost/asya/oop/minuman2/t...

localhost/asya/oop/minuman2/tampil.php?message=Data%20berhasil%20dihapus

CRUD OOP PHP

Input Data

logout

Id	nama	rasa	harga	Aksi
1	Bintang	melati	13000	<div>EditDelete</div>
2	pop ice	strobery	10000	<div>EditDelete</div>
3	teh jus	lemon	5000	<div>EditDelete</div>

81°F Cloudy

Search

7:52 AM

11/1/2024

81°F Cloudy

Search

7:52 AM  
11/1/2024