

LAPORAN PRAKTIKUM PEMROGRAMAN WEB

PRAKTIKUM 10 – SESSION & COOKIE



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 10 ini saya menggunakan data pada praktikum sebelumnya yaitu Apple Store. Pada praktikum ini saya menambahkan fitur Login dan Register pada halaman awal website yang mengimplementasikan materi **Session & Cookie**. Fitur Login memungkinkan pengguna untuk memasukkan kredensial mereka dan mengakses akun pribadi mereka. Saat pengguna berhasil login, sesi akan dibuat dan cookie akan disimpan untuk mengelola status login mereka selama sesi berlangsung. Fitur Register memberikan fasilitas bagi pengguna baru untuk membuat akun dengan mengisi formulir pendaftaran. Data pendaftaran ini kemudian disimpan di basis data dan dapat digunakan untuk autentikasi pada saat login.

B. Cookies

▪ login.php

```
login.php X
login.php > ...
1  <?php
2  require 'koneksi.php';
3
4  if (session_status() == PHP_SESSION_NONE) {
5      session_start();
6  }
7
8  // Redirect jika pengguna sudah login
9  if (!empty($_SESSION["id"])) {
10     header("Location: index.php");
11     exit;
12 }
13
14 // Cek cookie untuk login otomatis
15 if (isset($_COOKIE["user_id"])) {
16     $_SESSION["id"] = $_COOKIE["user_id"];
17     header("Location: index.php");
18     exit;
19 }
20
21 $error_messages = [];
22
23 if (isset($_POST["submit"])) {
24     $usernameemail = trim($_POST["usernameemail"]);
25     $password = $_POST["password"];
26
27     // Validasi input
28     if (empty($usernameemail)) {
29         $error_messages[] = "Username atau email tidak boleh kosong.";
30     }
31
32     if (empty($password)) {
33         $error_messages[] = "Kata sandi tidak boleh kosong.";
34     }
35
36     if (empty($error_messages)) {
37         $result = mysqli_query($conn, "SELECT * FROM user WHERE username = '$usernameemail' OR email = '$usernameemail'");
```



```

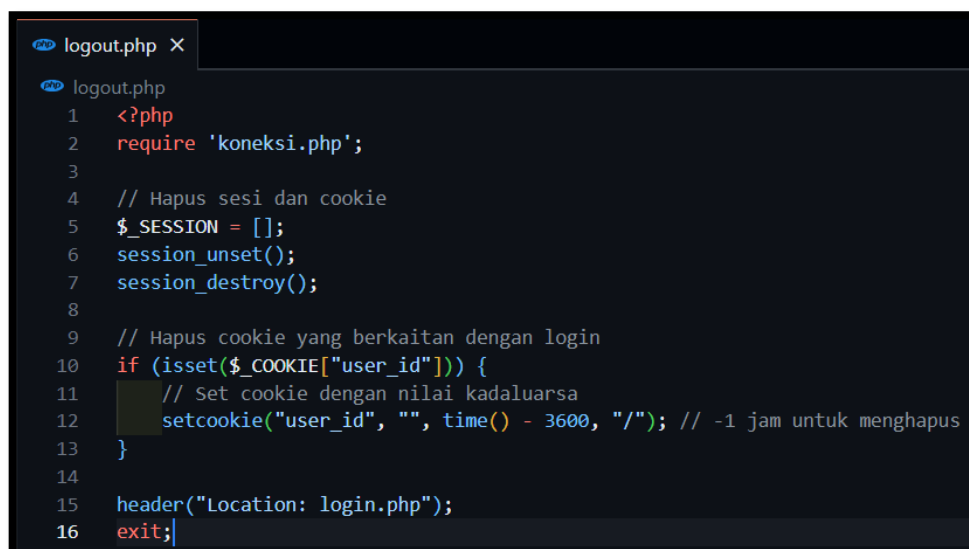
35
36 if (empty($error_messages)) {
37     $result = mysqli_query($conn, "SELECT * FROM user WHERE username = '$usernameemail' OR email = '$usernameemail'");
38     if (mysqli_num_rows($result) > 0) {
39         $row = mysqli_fetch_assoc($result);
40         if (password_verify($password, $row['password'])) {
41             $_SESSION["login"] = true;
42             $_SESSION["id"] = $row["id"];
43             // Set cookie untuk login otomatis
44             setcookie("user_id", $row["id"], time() + (100), "/"); //100 detik
45             header("Location: index.php");
46             exit;
47         } else {
48             $error_messages[] = "Kata sandi salah.";
49         }
50     } else {
51         $error_messages[] = "Pengguna tidak terdaftar.";
52     }
53 }
54 }
55 ?>

```

Gambar 1 login.php fungsi cookies

Kode PHP di atas menggunakan cookies untuk mengimplementasikan fitur login otomatis. Saat pengguna berhasil login, cookie dengan nama **user_id** diset menggunakan **setcookie()** yang berisi ID pengguna dan memiliki waktu kedaluwarsa **100 detik**. Ketika halaman diakses kembali, script mengecek keberadaan cookie tersebut dengan **isset(\$_COOKIE["user_id"])**. Jika cookie ditemukan, maka nilai ID dari cookie tersebut disimpan dalam session (**\$_SESSION["id"]**), dan pengguna akan langsung diarahkan ke halaman index.php tanpa perlu login ulang. Cookie ini memungkinkan pengguna tetap login meskipun browser ditutup, selama cookie belum kedaluwarsa.

- **logout.php**



```

logout.php
1 <?php
2 require 'koneksi.php';
3
4 // Hapus sesi dan cookie
5 $_SESSION = [];
6 session_unset();
7 session_destroy();
8
9 // Hapus cookie yang berkaitan dengan login
10 if (isset($_COOKIE["user_id"])) {
11     // Set cookie dengan nilai kadaluwarsa
12     setcookie("user_id", "", time() - 3600, "/"); // -1 jam untuk menghapus
13 }
14
15 header("Location: login.php");
16 exit;


```

Gambar 2 logout.php fungsi cookies

Kode PHP di atas menangani proses logout dengan cara menghapus cookie yang terkait dengan login otomatis. Setelah mengosongkan dan menghancurkan sesi menggunakan `session_unset()` dan `session_destroy()`, script memeriksa keberadaan cookie "user_id" dengan `isset($_COOKIE["user_id"])`. Jika cookie tersebut ada, maka cookie tersebut dihapus dengan `setcookie("user_id", "", time() - 3600, "/")`, di mana waktu kedaluwarsa cookie diatur ke waktu yang telah lewat (**satu jam yang lalu**) untuk memastikan cookie dihapus dari browser. Setelah itu, pengguna diarahkan ke halaman login.php menggunakan header **Location: login.php**.

C. Session

- koneksi.php



```
koneksi.php X
koneksi.php > ...
1 <?php
2 session_start();
3 $conn = mysqli_connect("localhost", "root", "", "ppw10_sescok");
```

Gambar 3 koneksi.php fungsi session

Pada kode di atas, bagian `session_start();` berfungsi untuk memulai sesi di PHP. Ketika `session_start();` dipanggil, PHP akan memeriksa apakah ada sesi yang sudah ada untuk pengguna tersebut dengan melihat cookie sesi yang dikirim oleh browser. Jika sesi sudah ada, PHP akan melanjutkan sesi tersebut dan membuat data sesi tersedia di skrip. Jika tidak ada sesi yang ada, PHP akan membuat sesi baru dan mengirim cookie sesi baru ke browser pengguna. Sesi ini memungkinkan penyimpanan data pengguna yang dapat diakses di berbagai halaman selama pengguna berinteraksi dengan aplikasi web.

- login.php

```
login.php x
login.php > ...
1  <?php
2  require 'koneksi.php';
3
4  if (session_status() == PHP_SESSION_NONE) {
5      session_start();
6  }
7
8  // Redirect jika pengguna sudah login
9  if (!empty($_SESSION["id"])) {
10     header("Location: index.php");
11     exit;
12 }
13
14 // Cek cookie untuk login otomatis
15 if (isset($_COOKIE["user_id"])) {
16     $_SESSION["id"] = $_COOKIE["user_id"];
17     header("Location: index.php");
18     exit;
19 }
20
21 $error_messages = [];
22
23 if (isset($_POST["submit"])) {
24     $usernameemail = trim($_POST["usernameemail"]);
25     $password = $_POST["password"];
26
27     // Validasi input
28     if (empty($usernameemail)) {
29         $error_messages[] = "Username atau email tidak boleh kosong.";
30     }
31
32     if (empty($password)) {
33         $error_messages[] = "Kata sandi tidak boleh kosong.";
34     }
35
36     if (empty($error_messages)) {
37         $result = mysqli_query($conn, "SELECT * FROM user WHERE username = '$usernameemail' OR email = '$usernameemail'");
```

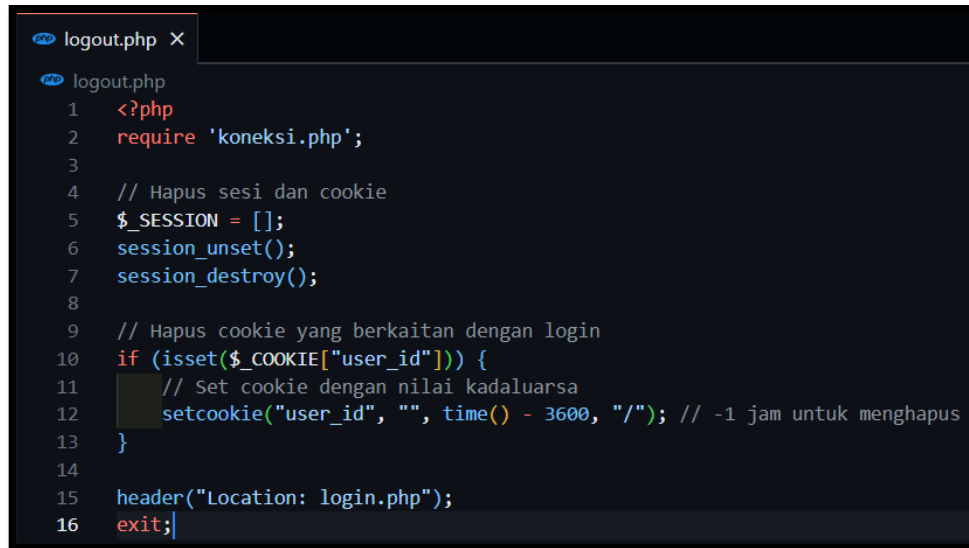
```
login.php x
login.php > ...
35
36     if (empty($error_messages)) {
37         $result = mysqli_query($conn, "SELECT * FROM user WHERE username = '$usernameemail' OR email = '$usernameemail'");
38         if (mysqli_num_rows($result) > 0) {
39             $row = mysqli_fetch_assoc($result);
40             if (password_verify($password, $row['password'])) {
41                 $_SESSION["login"] = true;
42                 $_SESSION["id"] = $row["id"];
43                 // Set cookie untuk login otomatis
44                 setcookie("user_id", $row["id"], time() + (100), "/"); //100 detik
45                 header("Location: index.php");
46                 exit;
47             } else {
48                 $error_messages[] = "Kata sandi salah.";
49             }
50         } else {
51             $error_messages[] = "Pengguna tidak terdaftar.";
52         }
53     }
54 }
55 ?>
```

Gambar 4 login.php fungsi session

Pada kode di atas, sesi digunakan untuk mengelola status login pengguna. `session_start()`; digunakan untuk memulai sesi atau melanjutkan sesi yang ada jika belum dimulai. Kode tersebut memeriksa apakah sesi sudah ada dengan `session_status() == PHP_SESSION_NONE` sebelum memulai sesi baru. Jika pengguna sudah login, disimpan dalam `$_SESSION["id"]`, mereka akan diarahkan ke `index.php`. Selain itu, jika terdapat cookie `user_id`, nilai tersebut akan disimpan dalam `$_SESSION["id"]` untuk login otomatis, dan pengguna juga akan diarahkan

ke index.php. Saat pengguna berhasil login, `$_SESSION["id"]` dan `$_SESSION["login"]` diatur untuk menandai bahwa pengguna tersebut sudah terotentikasi.

- **logout.php**



```
logout.php X
logout.php
1  <?php
2  require 'koneksi.php';
3
4  // Hapus sesi dan cookie
5  $_SESSION = [];
6  session_unset();
7  session_destroy();
8
9  // Hapus cookie yang berkaitan dengan login
10 if (isset($_COOKIE["user_id"])) {
11     // Set cookie dengan nilai kadaluarsa
12     setcookie("user_id", "", time() - 3600, "/"); // -1 jam untuk menghapus
13 }
14
15 header("Location: login.php");
16 exit;
```

Gambar 5 logut.php fungsi session

Pada kode di atas, sesi digunakan untuk mengelola status logout pengguna. Pertama, `$_SESSION = [];` digunakan untuk mengosongkan semua data sesi. Kemudian, `session_unset();` menghapus semua variabel sesi, dan `session_destroy();` menghentikan sesi dan menghapus data sesi di server. Jika ada cookie `user_id` yang menyimpan informasi login pengguna, cookie tersebut dihapus dengan `setcookie("user_id", "", time() - 3600, "/");`, yang mengatur cookie tersebut dengan nilai kadaluarsa di masa lalu (1 jam yang lalu) sehingga browser menghapusnya. Akhirnya, pengguna diarahkan kembali ke halaman login menggunakan `header("Location: login.php");` dan `exit;` memastikan skrip berhenti setelah pengalihan tersebut.

- register.php

```

register.php x
register.php > ...
1  <?php
2  require 'koneksi.php';
3
4  if (session_status() == PHP_SESSION_NONE) {
5      session_start();
6  }
7
8  // Redirect jika pengguna sudah login
9  if (!empty($_SESSION["id"])) {
10     header("Location: index.php");
11     exit;
12 }
13
14 $error_messages = [];
15
16 if (isset($_POST["submit"])) {
17     $name = trim($_POST["name"]);
18     $username = trim($_POST["username"]);
19     $email = trim($_POST["email"]);
20     $password = $_POST["password"];
21     $confirmpassword = $_POST["confirmpassword"];
22
23     if (empty($name)) {
24         $error_messages[] = "Nama tidak boleh kosong.";
25     }
26
27     if (empty($username)) {
28         $error_messages[] = "Username tidak boleh kosong.";
29     }
30
31     if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
32         $error_messages[] = "Format email tidak valid.";
33     }
34
35     if (strlen($password) < 6) {
36         $error_messages[] = "Kata sandi harus minimal 6 karakter.";
37     }

```

```

register.php x
register.php > ...
38
39 if ($password != $confirmpassword) {
40     $error_messages[] = "Kata sandi dan konfirmasi kata sandi tidak cocok.";
41 }
42
43 $duplicate = mysqli_query($conn, "SELECT * FROM user WHERE username = '$username' OR email = '$email'");
44 if (mysqli_num_rows($duplicate) > 0) {
45     $error_messages[] = "Username atau Email sudah digunakan.";
46 }
47
48 if (empty($error_messages)) {
49     $hashed_password = password_hash($password, PASSWORD_BCRYPT);
50     $query = "INSERT INTO user (name, username, email, password) VALUES ('$name', '$username', '$email', '$hashed_password')";
51     if (mysqli_query($conn, $query)) {
52         echo "<script>alert('Registrasi berhasil!');</script>";
53     } else {
54         $error_messages[] = "Terjadi kesalahan pada server.";
55     }
56 }
57 }
58 ?>

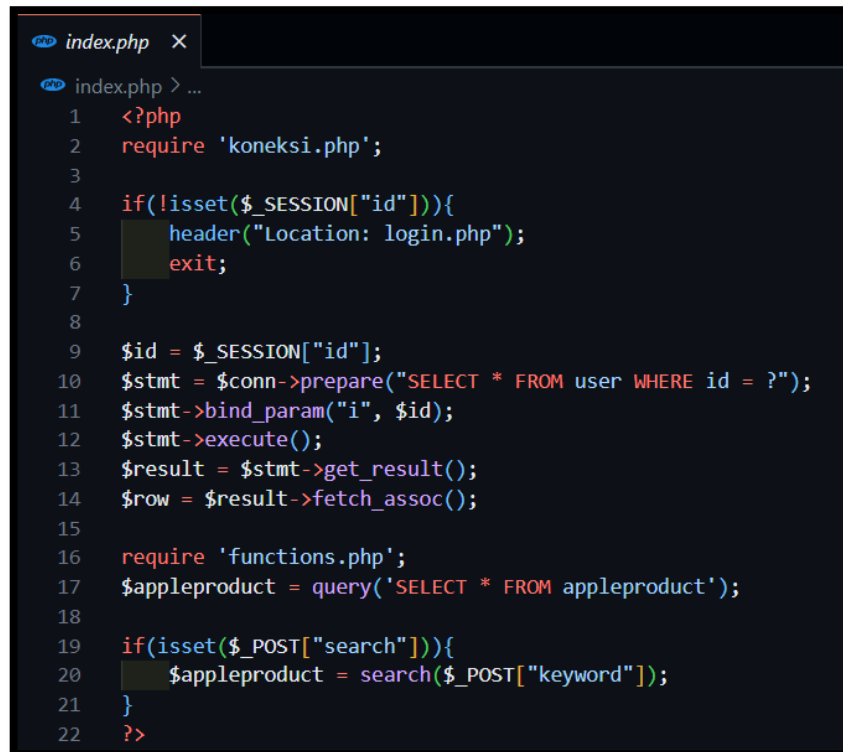
```

Gambar 6 register.php fungsi session

Pada kode register.php di atas, sesi digunakan untuk mengelola status login pengguna dan mengarahkan pengguna yang sudah login. Pertama, **session_start()**; digunakan untuk memulai atau melanjutkan sesi yang ada jika belum dimulai. Kode tersebut

memeriksa apakah pengguna sudah login dengan memeriksa apakah `$_SESSION["id"]` tidak kosong. Jika `$_SESSION["id"]` berisi data, pengguna akan diarahkan ke `index.php`, mencegah pengguna yang sudah login untuk mengakses halaman registrasi lagi.

- **index.php**



```
index.php X
index.php > ...
1  <?php
2  require 'koneksi.php';
3
4  if(!isset($_SESSION["id"])){
5      header("Location: login.php");
6      exit;
7  }
8
9  $id = $_SESSION["id"];
10 $stmt = $conn->prepare("SELECT * FROM user WHERE id = ?");
11 $stmt->bind_param("i", $id);
12 $stmt->execute();
13 $result = $stmt->get_result();
14 $row = $result->fetch_assoc();
15
16 require 'functions.php';
17 $appleproduct = query('SELECT * FROM appleproduct');
18
19 if(isset($_POST["search"])){
20     $appleproduct = search($_POST["keyword"]);
21 }
22 ?>
```

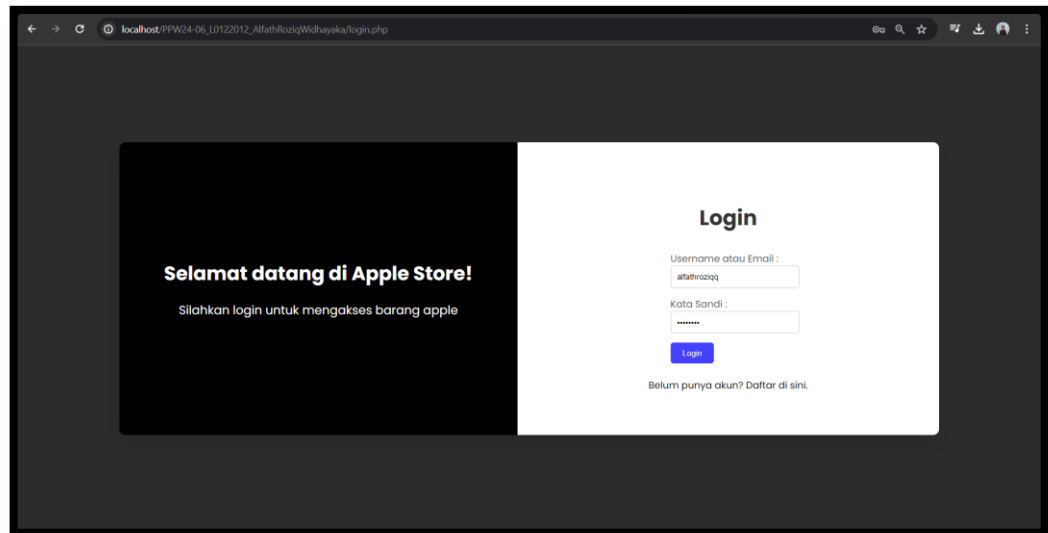
Gambar 7 index.php fungsi session

Kode `index.php` menggunakan sesi untuk memastikan bahwa hanya pengguna yang sudah login yang dapat mengakses halaman tersebut. Dengan memeriksa keberadaan `$_SESSION["id"]`, pengguna yang belum login diarahkan kembali ke halaman login. Sesi ini memastikan keamanan dan kontrol akses ke halaman utama aplikasi.

BAB II

A. Hasil Output

■ Halaman Login

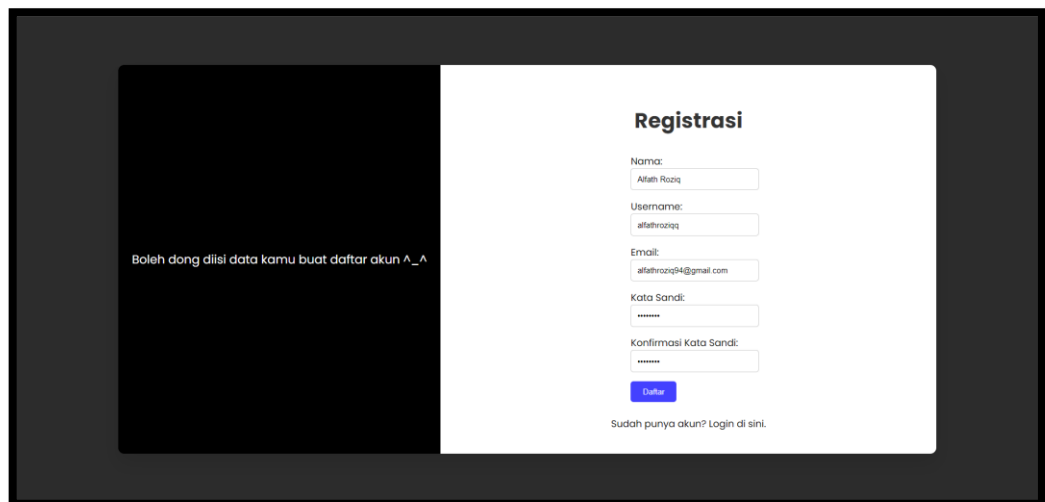


The screenshot shows a web browser window with the URL `localhost:PPW24-06_10122012_AlfathRoziqWidhayaka/login.php`. The page has a dark gray background. On the left, a black box contains the text "Selamat datang di Apple Store!" and "Silahkan login untuk mengakses barang apple". On the right, a white box titled "Login" contains a form with two input fields: "Username atau Email :" (containing "alfathroziq") and "Kata Sandi :" (containing "*****"). Below the fields is a blue "Login" button. At the bottom of the white box, it says "Belum punya akun? Daftar di sini."

Gambar 8 halaman login

Gambar di atas menunjukkan halaman login. Pada halaman ini, pengguna diminta untuk memasukkan Username atau Email serta Kata Sandi. Jika pengguna sudah terdaftar, mereka cukup mengisi informasi tersebut dan klik tombol login untuk mengakses halaman utama. Namun, jika belum memiliki akun, pengguna harus mendaftar terlebih dahulu dengan mengklik "**Belum punya akun? Daftar di sini.**".

■ Halaman Register

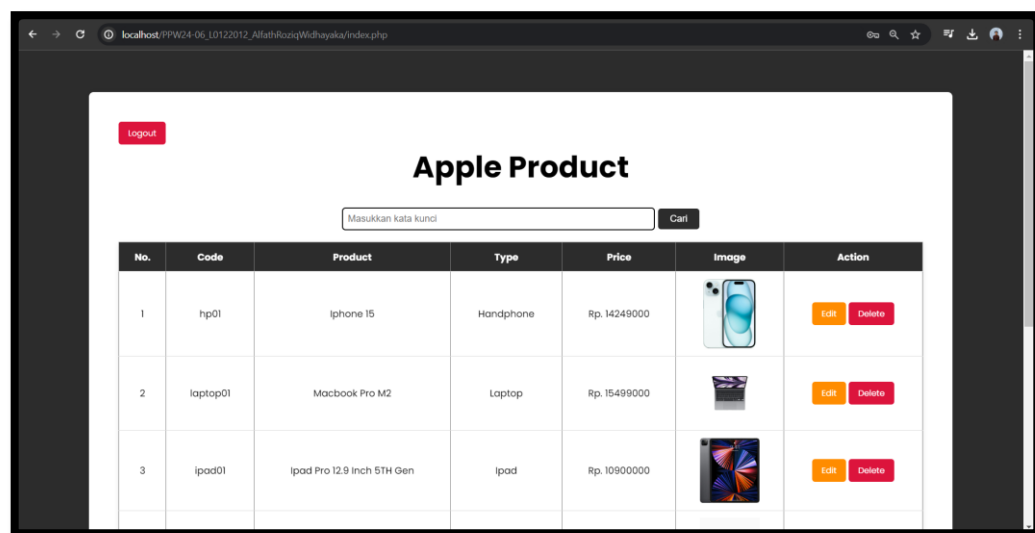


The screenshot shows a web browser window with the same URL as the login page. The page has a dark gray background. On the left, a black box contains the text "Boleh dong diisi data kamu buat daftar akun ^_^". On the right, a white box titled "Registrasi" contains a form with five input fields: "Nama:" (containing "Alfath Roziq"), "Username:" (containing "alfathroziq"), "Email:" (containing "alfathroziq4@gmail.com"), "Kata Sandi:" (containing "*****"), and "Konfirmasi Kata Sandi:" (containing "*****"). Below the fields is a blue "Daftar" button. At the bottom of the white box, it says "Sudah punya akun? Login di sini."

Gambar 9 halaman register

Gambar di atas menunjukkan halaman registrasi. Pengguna akan diarahkan ke halaman ini setelah mengklik **"Belum punya akun? Daftar di sini."** pada halaman login. Halaman ini ditujukan untuk pengguna yang belum memiliki akun. Formulir ini memerlukan pengguna untuk mengisi Nama, Username, Email, Kata Sandi, dan Konfirmasi Kata Sandi. Pada bagian Kata Sandi dan Konfirmasi Kata Sandi, pengguna bisa memilih untuk menampilkan kata sandi sebagai teks biasa atau menyembunyikannya kembali. Setelah mengisi formulir, pengguna dapat mengklik tombol Daftar. Data yang dimasukkan akan disimpan di database dan dapat digunakan untuk login.

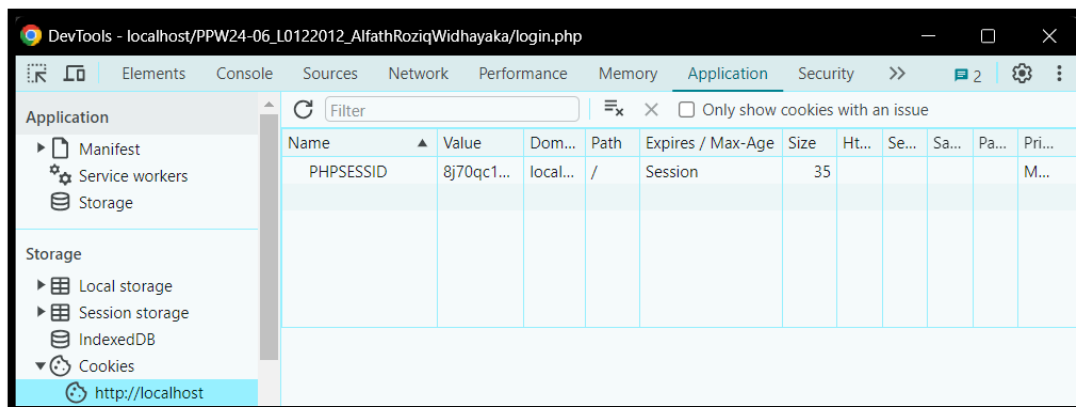
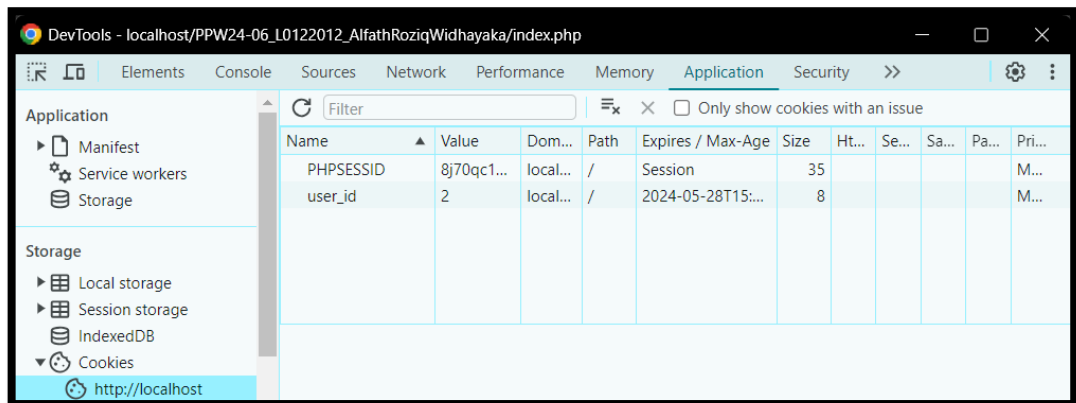
- **Halaman Index dengan tombol Logout**



Gambar 10 halaman register

Gambar di atas menunjukkan halaman utama pada website ini, yaitu index.html. Halaman ini hanya dapat diakses setelah pengguna melakukan login. Pada halaman ini, ditampilkan data yang diambil dari database dbapplestore. Halaman ini mencakup judul, tampilan data, dan beberapa barang apple yang isinya diambil dari database. Setiap menu memiliki fungsi untuk mengedit dan menghapus data. Selain itu, terdapat opsi untuk menambah barang baru jika ingin menambahkan barang. Halaman index juga memiliki tombol logout untuk keluar dari akun.

- **Cek Cookies**



Gambar 11 hasil cek cookies sesudah login dan sesudah logout

Untuk memeriksa cookie, kita perlu melakukan login terlebih dahulu dan masuk ke halaman index. Di sana, kita dapat melihat keberadaan cookie. Namun, saat kita melakukan logout, cookie tersebut akan dihapus.

BAB III

D. Kesimpulan

Praktikum Pemrograman Web ini memberikan pemahaman yang mendalam tentang implementasi Session & Cookie dalam pengembangan aplikasi web. Dengan adanya fitur Login dan Register, pengguna dapat mengakses akun pribadi mereka dengan aman dan nyaman. Penggunaan Session memungkinkan penyimpanan data pengguna yang dapat diakses di berbagai halaman, sementara Cookie memberikan kemampuan untuk tetap login meskipun browser ditutup. Dalam praktikum ini, saya berhasil mengimplementasikan mekanisme otentikasi pengguna dengan memanfaatkan Session & Cookie, sehingga meningkatkan keamanan dan kenyamanan pengguna dalam menggunakan aplikasi web.

Selain itu, pemahaman tentang pengelolaan Session & Cookie juga memperluas pengetahuan saya dalam mengelola status login dan logout pengguna. Dengan adanya fitur Logout, pengguna dapat secara aman keluar dari akun mereka, sementara cookie yang terkait dengan login otomatis dapat dihapus untuk menjaga keamanan informasi pengguna.