

BAB IV

COOKIE DAN SESSION

A. TUJUAN

- Memahami konsep dasar cookie dan session di dalam pemrograman web.
- Memahami penggunaan cookie dan session.
- Mampu memanfaatkan cookie atau session untuk memproteksi halaman web.

B. ALOKASI WAKTU

4 JS (4 x 50 menit)

C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

D. DASAR TEORI

1. Cookie

Pada dasarnya cookie merupakan mekanisme untuk meletakkan data pada remote browser sehingga memudahkan penelusuran atau identifikasi user. Dapat juga dikatakan bahwa cookie merupakan informasi dalam bentuk teks yang dipertukarkan oleh client dan server, di mana pembuat cookie adalah pihak server.

2. Session

Pada kenyataannya, penggunaan cookie memiliki beberapa masalah atau kendala. Bisa jadi ada browser yang tidak menerima cookie, serta kemungkinan adanya user yang men-disable cookie. Permasalahan ini merupakan salah satu alasan session PHP menggunakan method cookie/URL ganda.

Tidak seperti cookie, session tersimpan di server. Dengan demikian, client tidak memiliki akses untuk mendapatkan informasi session.

E. LATIHAN

1. Menggunakan Cookie

Untuk dapat memanfaatkan cookie di dalam aplikasi, kita perlu memahami operasi-operasi dasar penggunaan cookie.

- Menciptakan dan Mengakses Cookie

PHP menyediakan fungsi `setcookie()` yang digunakan untuk menetapkan nilai cookie. Adapun untuk mendapatkan nilai cookie, kita memanfaatkan variabel superglobal `$_COOKIE`.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Set Cookie</title>
</head>

<body>

<?php

// Men-set nilai cookie
setcookie('nama_cookie', 'nilai_cookie');

// Mendapatkan nilai cookie
echo $_COOKIE['nama_cookie'];

?>

<p>
Tekan Refresh (F5);

</body>

</html>
```

Fungsi `setcookie()` sebenarnya menyediakan enam parameter opsional, namun seringnya hanya tiga yang digunakan.

```
setcookie ( string name [, string value [, int expire ]] )
```

- Memeriksa Dukungan Cookie

Implementasi cookie sangat bergantung pada dukungan browser. Terkait hal ini, kita bisa melakukan pemeriksaan guna memastikan bahwa cookie diaktifkan atau tidak.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
```

```
<title>Cek Dukungan Cookie</title>
</head>

<body>

<?php

if (isset($_GET['q']) && $_GET['q'] == 1) {

    if (isset($_COOKIE['test'])) {
        echo 'support';
    } else {
        echo 'tidak support';
    }

} else {
    setcookie('test', 'value');
    $self = $_SERVER['PHP_SELF'];

    // Redireksi ke current script
    header('Location: ' . $self . '?q=1');
    exit;
}

?>

<p>
Tekan Refresh (F5);

</body>

</html>
```

Untuk menguji kode program, coba disable dukungan cookie pada browser, kemudian tekan Refresh.

▪ Menghapus Cookie

Penghapusan cookie dilakukan dengan cara mengirimkan nama cookie sama tetapi nilainya kosong. Untuk langkah yang lebih baik, tambahkan argumen opsional dengan nilai waktu yang telah lewat.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Hapus Cookie</title>
</head>

<body>

<?php

setcookie('nama_cookie', 'nilai_cookie');

if (isset($_COOKIE['nama_cookie'])) {
    echo 'cookie di-set <br />';
}
```

```
// Menghapus cookie, dengan masa berlaku 3 jam yang lalu
setcookie('nama_cookie', '', time() - 3 * 3600);
echo 'cookie dihapus';

} else {
    echo 'unset';
}

?>

<p>
Tekan Refresh (F5);

</body>

</html>
```

2. Menggunakan Session

Untuk memudahkan penggunaan session lebih lanjut, bagian ini akan menjelaskan beberapa operasi dasar pada session.

▪ Menciptakan dan Mengakses Session

PHP menyediakan fungsi `session_start()` yang harus selalu dipanggil sebelum kita dapat menetapkan maupun mengakses variabel session. Adapun untuk menetapkan nilai variabel session, kita menggunakan superglobal `$_SESSION`.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Set Session</title>
</head>

<body>

<?php

// Inisialisasi data session
session_start();

// Set session jika belum ada
if (!isset($_SESSION['test'])) {
    $_SESSION['test'] = 'value';
} else {
    echo 'Session di-set <br />';
    // Mencetak nilai session test
    echo 'Nilai: ' . $_SESSION['test'] . '<br />';

    // Mencetak semua elemen session
    print_r($_SESSION);
}

?>
```

```
<p>
Tekan Refresh (F5);

</body>

</html>
```

▪ Menghapus Session

Untuk menghapus data session, Anda bisa menggunakan konstruksi bahasa `unset()` atau fungsi `session_destroy()`.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Hapus Session</title>
</head>

<body>

<?php

// Inisialisasi data session
session_start();

// Set session jika belum ada
if (isset($_SESSION['test'])) {

    // Hapus session test
    unset($_SESSION['test']);

    echo 'session dihapus';

} else {
    echo 'unset';

    // Mencetak semua elemen session
    print_r($_SESSION);
}

?>

<p>
Tekan Refresh (F5);

</body>

</html>
```

Jika Anda hanya ingin menghapus data session tertentu, gunakan `unset()`. Sementara itu, untuk menghapus semua data session, gunakan `session_destroy()`.

3. Autentikasi Halaman

Salah satu bentuk aplikasi yang sangat memerlukan cookie/session adalah halaman autentikasi. Pada prinsipnya, autentikasi digunakan untuk memproteksi halaman-halaman sensitif.

Untuk mengetahui lebih detail mengenai halaman autentikasi, ikuti langkah-langkah berikut:

1. Buat folder khusus, misalnya autentikasi_cookie.
2. Buat kode untuk halaman index.php.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Halaman Administrator</title>

    <style type="text/css">
        .inner {
            margin: 200px auto;
            padding: 20px;
            width: 240px;
            border: 1px solid #333;
        }
    </style>
</head>

<body>

<?php
ini_set('display_errors', 1);
define('_VALID', 1);

// include file eksternal
require_once('./auth.php');

init_login();
validate();

?>

<h3>Simulasi Halaman Admin</h3>

<p>
<a href="?m=logout">Logout</a>

<p>
Menu-menu admin ada di sini

</body>

</html>
```

3. Buat kode untuk autentikasi dan simpan dengan nama auth.php misalnya.

```
<?php
defined('_VALID') or die('not allowed');

function init_login() {
    // Simulasi data account nama dan password
    $nama = 'admin';
    $pass = 'admin';

    if (isset($_POST['nama']) && isset($_POST['pass'])) {
        $n = trim($_POST['nama']);
        $p = trim($_POST['pass']);

        if ( ($n === $nama) && ($p === $pass) ) {
            // Jika sama, set cookie
            setcookie('nlogin', $n);
            setcookie('time', time());

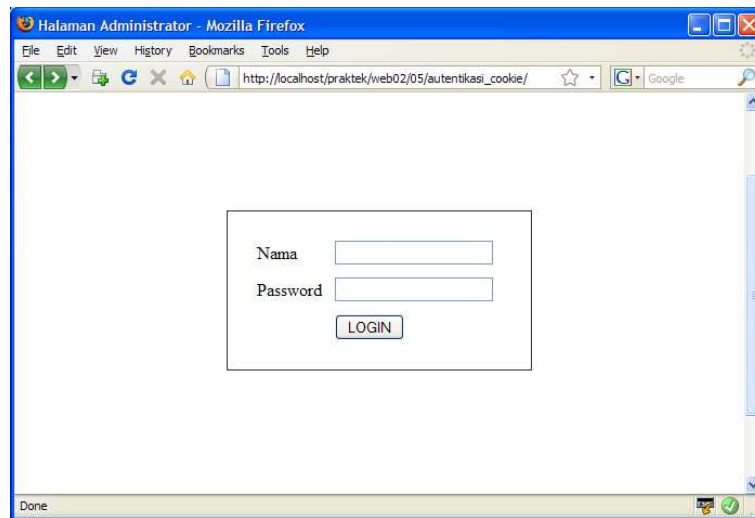
            // redireksi
            ?>
            <script type="text/javascript">
            document.location.href="./";
            </script>
            <?php
        } else {
            echo 'Nama/Password Tidak Sesuai';
            return false;
        }
    }
}

function validate() {
    if (!isset($_COOKIE['nlogin']) || !isset($_COOKIE['time']) ) { ?>
        <div class="inner">
        <form action="" method="post">
        <table border=0 cellpadding=5>
            <tr>
                <td>Nama</td>
                <td><input type="text" name="nama" /></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="password" name="pass" /></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" value="LOGIN" /></td>
            </tr>
        </table>
        </form>
        </div>
        <?php
        exit;
    }

    if (isset($_GET['m']) && $_GET['m'] == 'logout') {
        // Hapus cookie
        if (isset($_COOKIE['nlogin'])) {
            setcookie('nlogin', '', time() - 3 * 3600);
        }
        if (isset($_COOKIE['time'])) {
            setcookie('time', '', time() - 3 * 3600);
        }
    }
}
```

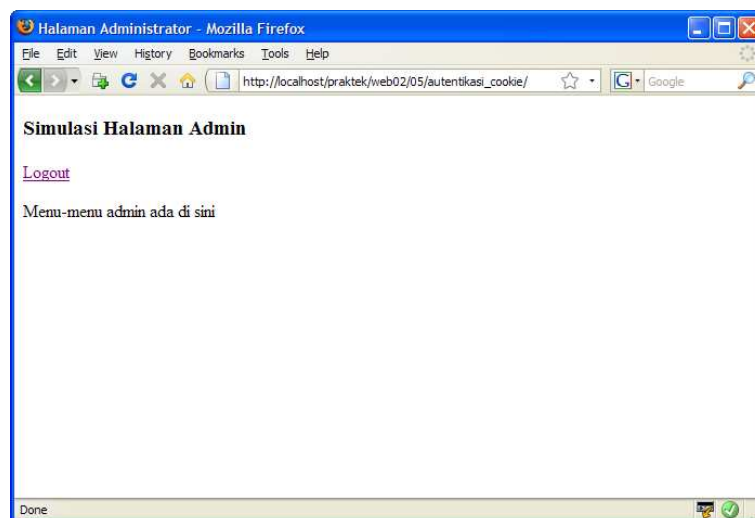
```
// redireksi halaman
?>
<script type="text/javascript">
document.location.href="."/;
</script>
<?php
}
}
?>
```

4. Jalankan halaman index atau browse direktorinya.



Gambar 1. Halaman login

Begitu login berhasil, akan muncul halaman utama administrator seperti terlihat pada Gambar 2.



Gambar 2. Halaman administrator

F. TUGAS PRAKTIKUM

1. Buat halaman autentikasi dengan menggunakan session.