



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-8: PHP - Form Upload, Cookies dan Session
Mata Kuliah Desain dan Pemrograman Web
Tim Ajar Desain dan Pemrograman Web
Oktober 2024

Topik

- Konsep Form Upload dengan PHP dan JQuery.
- Pengenalan konsep cookies.
- Pengenalan konsep session.

Tujuan

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat form upload menggunakan PHP dan jQuery.
2. Memahami konsep dasar cookies dan penggunaannya.
3. Memahami konsep dasar session dan penggunaannya.

Pendahuluan

File Upload

File upload dalam PHP adalah proses mengunggah file dari komputer pengguna ke server web. File upload berguna dalam banyak kasus seperti saat pengguna ingin mengunggah gambar profil, dokumen, atau media lainnya ke situs web. Untuk melakukan ini, dapat menggunakan elemen HTML `<input type="file">` bersama dengan PHP untuk mengelola proses pengunggahan.

Berikut adalah langkah-langkah umum untuk mengimplementasikan file upload dalam PHP:

- Buat formulir HTML untuk mengunggah file:

```
<!DOCTYPE html>
<html>
<head>
  <title>File Upload</title>
</head>
<body>
  <form action="upload.php" method="post" enctype="multipart/form-data">
    <input type="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload File" name="submit">
  </form>
</body>
</html>
```

- Buat skrip PHP (**upload.php**) untuk mengelola pengunggahan file:

```

<?php
if (isset($_POST["submit"])) {
    $targetDirectory = "uploads/"; // Direktori tujuan untuk menyimpan file
    $targetFile = $targetDirectory . basename($_FILES["fileToUpload"]["name"]);

    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $targetFile)) {
        echo "File berhasil diunggah.";
    } else {
        echo "Gagal mengunggah file.";
    }
}

```

Di atas, kita melakukan beberapa langkah penting:

- Membuat formulir HTML dengan elemen `<input type="file">` yang memungkinkan pengguna memilih file yang ingin diunggah.
- Mengatur atribut `enctype` formulir menjadi `"multipart/form-data"` agar file dapat diunggah.
- Ketika pengguna mengklik tombol "Upload File," data formulir akan dikirim ke `upload.php`. - Di `upload.php`, kita memeriksa apakah pengguna telah mengklik tombol submit (`$_POST["submit"]`) dan kemudian menentukan direktori tujuan untuk menyimpan file. - `move_uploaded_file()` digunakan untuk memindahkan file dari direktori sementara (temp) ke direktori tujuan yang telah ditentukan.
- Pesan yang sesuai (berhasil atau gagal) akan ditampilkan ke pengguna.

Pastikan telah memiliki direktori `uploads` pada server Anda, dan berikan izin yang sesuai agar PHP dapat menyimpan file di sana.

Praktikum Bagian 1. Upload File

Langkah	Keterangan
1	Buatlah file baru di dalam direktori <code>dasarWeb</code> , beri nama <code>form_upload.php</code> .
2	Ketikkan ke dalam file <code>form_upload.php</code> kode di bawah ini. <div data-bbox="335 1344 1433 1713" data-label="Text"> <pre> <html> <head> <title>File Upload</title> </head> <body> <form action="upload.php" method="POST" enctype="multipart/form-data"> <input type="file" name="myfile"> <input type="submit" name="submit"> </form> </body> </html> </pre> </div>
3	Buat file baru bernama <code>upload.php</code> yang akan digunakan untuk pemrosesan <code>form_upload.php</code> .

```
<?php
if(isset($_POST["submit"])){
    $targetdir = "uploads/"; //Direktori tujuan untuk menyimpan file
    $targetfile = $targetdir . basename($_FILES["myfile"]["name"]);

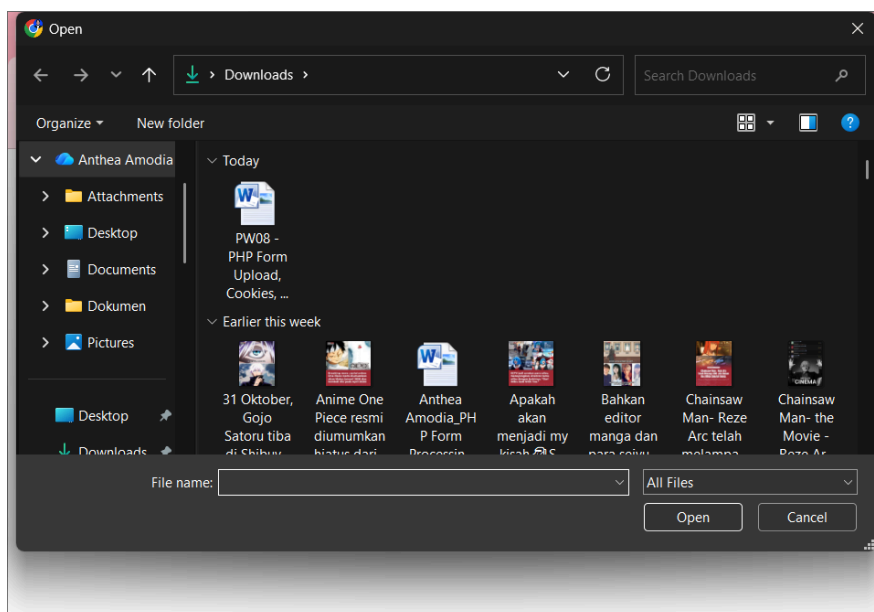
    if(move_uploaded_file($_FILES["myfile"]["tmp_name"], $targetfile)){
        echo "File berhasil diunggah.";
    }
    else{
        echo "Gagal mengunggah file.";
    }
}
?>
```

4

Simpan file, kemudian buka browser dan jalankan localhost/dasarWeb/form_upload.php.

Pilih sebuah file dan klik tombol Submit. Amati yang terjadi dan catat pemahaman Anda. (Pertanyaan No. 1)

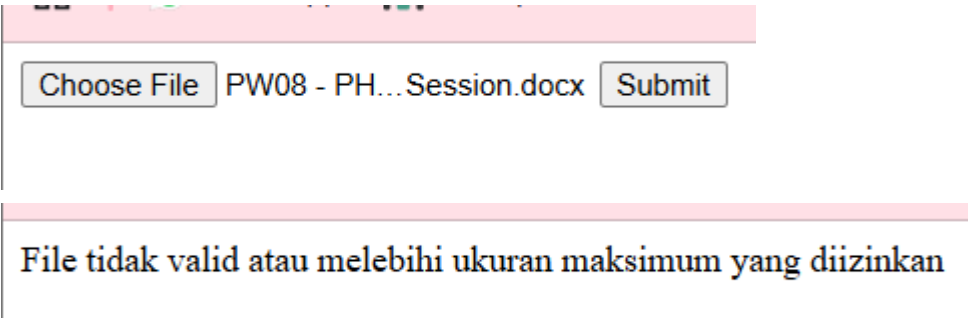
Choose File No file chosen Submit

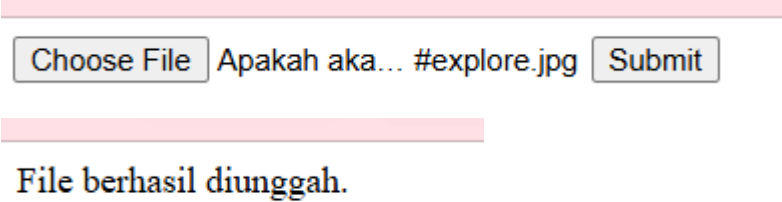


Gagal mengunggah file.

"Gagal mengunggah file." akan ditampilkan. Pemahaman: Ini terjadi karena skrip upload.php mencoba memindahkan file ke direktori uploads/ (\$targetdir = "uploads/";). Namun, direktori tersebut belum dibuat (sesuai langkah 4), sehingga fungsi move_uploaded_file() gagal dieksekusi.

5	<p>Selanjutnya buat folder bernama uploads pada direktori dasarWeb. Jalankan kembali localhost/dasarWeb/form_upload.php. Pilih sebuah file dan klik tombol Submit. Amati yang terjadi dan catat pemahaman Anda. (Pertanyaan No. 2)</p> <hr/> <div data-bbox="341 353 1062 405"> <input type="button" value="Choose File"/> Apakah aka... #explore.jpg <input type="button" value="Submit"/> </div> <hr/> <p>File berhasil diunggah.</p> <p>"File berhasil diunggah." akan ditampilkan, dan file yang Anda pilih akan muncul di dalam folder uploads yang baru dibuat. Pemahaman: Setelah folder uploads dibuat (langkah 5), skrip upload.php sekarang memiliki direktori tujuan yang valid. Fungsi move_uploaded_file() berhasil memindahkan file dari lokasi sementara ke direktori uploads.</p>
6	<p>Ubah isi dari file upload.php dengan kode berikut</p> <pre data-bbox="328 860 1449 1563"><?php if(isset(\$_POST["submit"])){ \$targetdir = "uploads/"; //Direktori tujuan untuk menyimpan file \$targetfile = \$targetdir . basename(\$_FILES["myfile"]["name"]); \$fileType = strtolower(pathinfo(\$targetfile, PATHINFO_EXTENSION)); \$allowedExtensions = array("jpg", "jpeg", "png", "gif"); \$maxsize = 5*1024*1024; if (in_array(\$fileType, \$allowedExtensions) && \$_FILES["myfile"]["size"]<=\$maxsize) { if(move_uploaded_file(\$_FILES["myfile"]["tmp_name"], \$targetfile)){ echo "File berhasil diunggah."; } else{ echo "Gagal mengunggah file."; } } else{ echo "File tidak valid atau melebihi ukuran maksimum yang diizinkan"; } } ?></pre>
7	<p>Simpan file, buka browser dan jalankan localhost/dasarWeb/form_upload.php Pilih sebuah file dengan ekstensi .pdf atau .docx. Klik tombol Submit. Amati yang terjadi dan catat pemahaman Anda. (Pertanyaan No. 3)</p> <hr/> <div data-bbox="341 1809 1062 1861"> <input type="button" value="Choose File"/> No file chosen <input type="button" value="Submit"/> </div>

	 <p>Pesan "File tidak valid atau melebihi ukuran maksimum yang diizinkan" akan ditampilkan. Pemahaman: Kode di Langkah 6 menambahkan validasi ekstensi file. Array \$allowedExtensions hanya berisi "jpg", "jpeg", "png", dan "gif". Karena file yang diunggah adalah .pdf atau .docx, ekstensi ini tidak ditemukan dalam array, sehingga kondisi in_array() gagal dan pesan error ditampilkan.</p>
--	--

8	<p>Selanjutnya jalankan kembali localhost/dasarWeb/form_upload.php. Pilih sebuah file dengan ekstensi .jpg, .jpeg, .png, atau .gif. Klik tombol Submit. Amati yang terjadi dan catat pemahaman Anda. (Pertanyaan No. 4)</p>  <p>"File berhasil diunggah." akan ditampilkan. Pemahaman: Berbeda dengan pertanyaan 3, kali ini ekstensi file (.jpg, .jpeg, .png, atau .gif) sesuai dengan yang ada di \$allowedExtensions. Asalkan ukuran file di bawah 5MB (batas \$maxsize), semua kondisi validasi terpenuhi dan file berhasil diunggah.</p>
9	<p>Tambahkan script langkah 6 untuk menampilkan file gambar thumbnail dengan ukuran lebar 200 dan tinggi mengikuti perubahan secara otomatis setelah file gambar berhasil diunggah. Screenshoot tambahan kode. Jelaskan pemahaman Anda setelah penambahan kode program tersebut. (Pertanyaan No. 5)</p> <p>menambahkan baris echo "
";.</p> <ul style="list-style-type: none"> • Kode ini akan menampilkan gambar yang baru saja diunggah ke browser. • Tag adalah tag HTML standar untuk gambar. • Atribut src diisi dengan variabel \$targetfile, yang berisi path ke file yang diunggah (misal: uploads/namafile.jpg). • Atribut style='width: 200px;' digunakan untuk mengatur lebar gambar

	(thumbnail) menjadi 200 piksel. Tingginya akan menyesuaikan secara proporsional.
10	<p>Selanjutnya, ubah isi dari file upload.php dengan kode berikut.</p> <pre> <?php if(isset(\$_POST["submit"])){ \$targetdir = "uploads/"; //Direktori tujuan untuk menyimpan file \$targetfile = \$targetdir . basename(\$_FILES["myfile"]["name"]); \$fileType = strtolower(pathinfo(\$targetfile, PATHINFO_EXTENSION)); \$allowedExtensions = array("txt", "pdf", "doc", "docx"); \$maxsize = 3*1024*1024; if (in_array(\$fileType, \$allowedExtensions) && \$_FILES["myfile"]["size"]<=\$maxsize) { if(move_uploaded_file(\$_FILES["myfile"]["tmp_name"], \$targetfile)){ echo "File berhasil diunggah"; } else{ echo "Gagal mengunggah file."; } } else{ echo "File tidak valid atau melebihi ukuran maksimum yang diizinkan"; } } ?> </pre>
11	<p>Simpan file, buka browser dan jalankan localhost/dasarWeb/form_upload.php Pilih sebuah file dengan ekstensi .txt, .pdf, .doc, atau .docx yang berukuran lebih dari 5 MB. Klik tombol Submit. Amati yang terjadi dan catat pemahaman Anda. (Pertanyaan No. 6)</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>File tidak valid atau melebihi ukuran maksimum yang diizinkan</p> </div> <p>Pertanyaan No. 6 Hasil: Pesan "File tidak valid atau melebihi ukuran maksimum yang diizinkan" akan ditampilkan. Pemahaman: Kode di Langkah 10 mengubah validasi. Ekstensi .txt, .pdf, .doc, atau .docx sekarang diizinkan. Namun, batas ukuran file (\$maxsize) diatur menjadi 3MB (3*1024*1024). Karena file yang diunggah berukuran lebih dari 5MB, kondisi \$_FILES["myfile"]["size"]<=\$maxsize gagal, dan pesan error ditampilkan.</p>
12	<p>Selanjutnya jalankan kembali localhost/dasarWeb/form_upload.php . Pilih sebuah file dengan ekstensi .txt, .pdf, .doc, atau .docx yang berukuran kurang dari 3 MB. Klik tombol Submit. Amati yang terjadi dan catat pemahaman Anda. (Pertanyaan No. 7)</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>File berhasil diunggah.</p> </div>

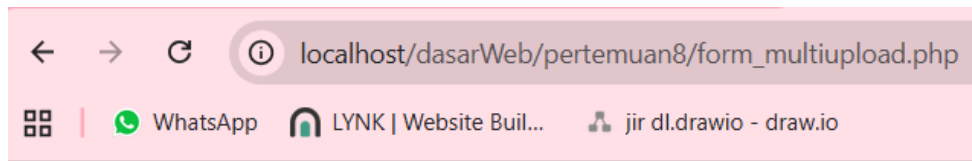
	Pesan "File berhasil diunggah." akan ditampilkan. Pemahaman: Kali ini, file yang diunggah memenuhi <i>kedua</i> kondisi: ekstensinya diizinkan (misal .txt) DAN ukurannya (kurang dari 3MB) berada di bawah batas \$maxsize.
--	--

Praktikum Bagian 2. Multi Upload File

Langkah	Keterangan
1	Buatlah file baru bernama form_multiupload.php .
2	<p>Ketikkan ke dalam form_multiupload.php kode berikut.</p> <pre><!DOCTYPE html> <html> <head> <title>Multiupload Dokumen</title> </head> <body> <h2>Unggah Dokumen</h2> <form action="proses_upload.php" method="post" enctype="multipart/form-data"> <input type="file" name="files[]" multiple="multiple" accept=".pdf, .doc, .docx" <input type="submit" value="Unggah" /> </form> </body> </html></pre>
3	<p>Buatlah file baru bernama proses_upload.php. Ketikkan kode berikut.</p> <pre><?php // Lokasi penyimpanan file yang diunggah \$targetDirectory = "documents/"; // Periksa apakah direktori penyimpanan ada, jika tidak maka buat if (!file_exists(\$targetDirectory)) { mkdir(\$targetDirectory, 0777, true); } if (\$_FILES['files']['name'][0]) { \$totalFiles = count(\$_FILES['files']['name']); // Loop melalui semua file yang diunggah for (\$i = 0; \$i < \$totalFiles; \$i++) { \$fileName = \$_FILES['files']['name'][\$i]; \$targetFile = \$targetDirectory . \$fileName; // Pindahkan file yang diunggah ke direktori penyimpanan if (move_uploaded_file(\$_FILES['files']['tmp_name'][\$i], \$targetFile)) { echo "File \$fileName berhasil diunggah.
"; } else { echo "Gagal mengunggah file \$fileName.
"; } } } else { echo "Tidak ada file yang diunggah."; }</pre>

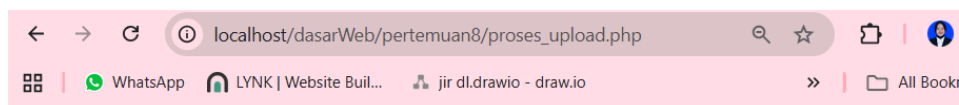
4

Simpan file, buka browser dan jalankan **localhost/dasarWeb/form_multiupload.php**. Pilih beberapa file sekaligus untuk diunggah. Apa yang anda pahami dari script pada file tersebut? Catat pemahaman anda. (Pertanyaan No. 8)



Unggah Dokumen

Choose Files 2 files Unggah



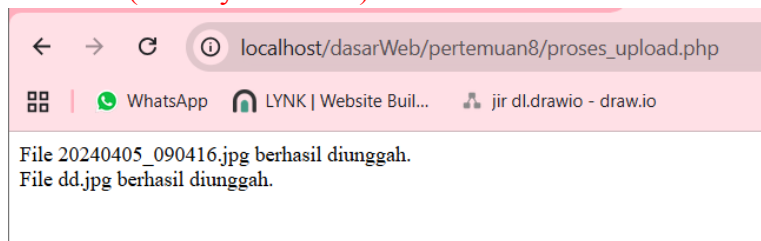
File 2A_Anthea Amodia_Desprog2.docx berhasil diunggah.
File Jobsheet 05 - Index dan Optimasi Query - Petunjuk Praktikum.pdf berhasil diunggah.

HTML (form_multiupload.php) : Menggunakan atribut `multiple="multiple"` pada `<input>` agar pengguna bisa memilih banyak file. Nama input `name="files[]"` (dengan `[]`) memberi tahu PHP untuk memperlakukan file yang diunggah sebagai array.

PHP (proses_upload.php) : Skrip ini pertama-tama memeriksa apakah direktori `documents/` ada, jika tidak, ia akan membuatnya. Kemudian, skrip menghitung jumlah file yang diunggah (`$totalFiles`). Menggunakan loop `for`, skrip memproses *setiap* file satu per satu, mengambil nama dan lokasi sementara, lalu memindahkannya ke direktori `documents/` menggunakan `move_uploaded_file()`.

5

Buat seperti langkah 3 dengan multi upload khusus gambar. Screen shoot perubahan kode dan berikan penjelasan kode tersebut. (Pertanyaan No. 9)



Array `$allowedExtensions` yang berisi ekstensi gambar ditambahkan di awal.

Di dalam loop `for`, kita sekarang mendapatkan ekstensi (`$fileType`) dari *setiap* file.

Ditambahkan `if(in_array($fileType, $allowedExtensions))` untuk memeriksa apakah file tersebut adalah gambar.

File hanya akan dipindahkan (`move_uploaded_file()`) jika ekstensinya valid. Jika tidak, pesan error khusus untuk file tersebut akan ditampilkan.

Praktikum Bagian 3. Upload File dengan PHP dan JQuery

Langkah	Keterangan
1	Buatlah file baru bernama <code>form_upload_ajax.php</code>
2	<p>Ketikkan kode berikut pada <code>form_upload_ajax.php</code></p> <pre><!DOCTYPE html> <html> <head> <title>Unggah File Dokumen</title> </head> <body> <form id="upload-form" action="upload_ajax.php" method="post" enctype= "multipart/form-data"> <input type="file" name="file" id="file"> <input type="submit" name="submit" value="Unggah"> </form> <div id="status"></div> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> <script src="upload.js"></script> </body> </html></pre>
3	Buatlah file baru bernama <code>upload.js</code> . Tuliskan kode berikut.

```

$(document).ready(function(){
    $('#upload-form').submit(function(e){
        e.preventDefault();

        var formData = new FormData(this);

        $.ajax({
            type: 'POST',
            url: 'upload_ajax.php',
            data: formData,
            cache: false,
            contentType: false,
            processData: false,
            success: function(response){
                $('#status').html(response);
            },
            error: function(){
                $('#status').html('Terjadi kesalahan saat mengunggah file.');

```

4

Buatlah file baru bernama **upload_ajax.php**. Tuliskan kode berikut.

```

<?php
if (isset($_FILES['file'])) {
    $errors = array();
    $file_name = $_FILES['file']['name'];
    $file_size = $_FILES['file']['size'];
    $file_tmp = $_FILES['file']['tmp_name'];
    $file_type = $_FILES['file']['type'];
    @ $file_ext = strtolower(" . end(explode('.', $_FILES['file']['name'])) . "");
    $extensions = array("pdf", "doc", "docx", "txt");

    if (in_array($file_ext, $extensions) === false) {
        $errors[] = "Ekstensi file yang diizinkan adalah PDF, DOC, DOCX, atau TXT.";
    }

    if ($file_size > 2097152) {
        $errors[] = 'Ukuran file tidak boleh lebih dari 2 MB';
    }

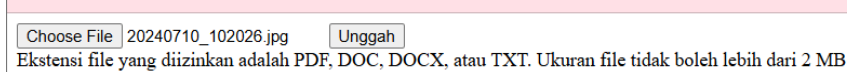
    if (empty($errors) == true) {
        move_uploaded_file($file_tmp, "documents/" . $file_name);
        echo "File berhasil diunggah.";
    } else {
        echo implode(" ", $errors);
    }
}

```

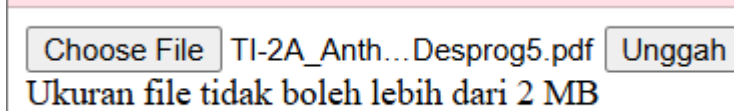
5

Simpan file tersebut, kemudian buka browser dan jalankan `localhost/dasarWeb/ form_upload_ajax.php`.

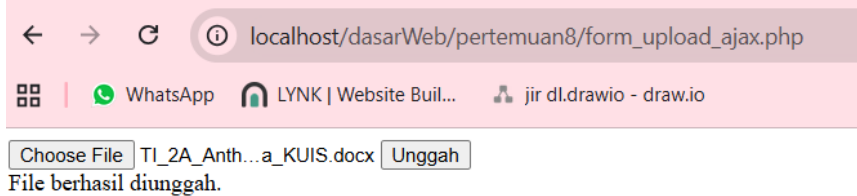
- Upload file berupa gambar.



- Upload file PDF yang berukuran > 4 MB.



- Upload file .docx dengan ukuran < 2 MB.



Amati apa yang terjadi dan jelaskan pemahaman anda.

(Pertanyaan No. 10)

Upload file gambar: Halaman tidak me-reload. Pesan error "Ekstensi file yang diizinkan adalah PDF, DOC, DOCX, atau TXT." akan muncul di `div#status`.

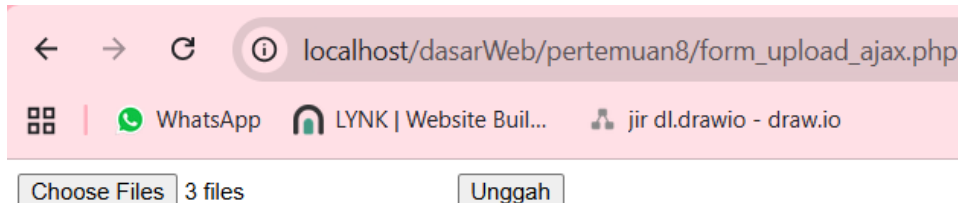
Upload file PDF > 4 MB: Halaman tidak me-reload. Pesan error "Ukuran file tidak boleh lebih dari 2 MB" akan muncul di `div#status`. (Batas di kode adalah 2MB/2097152 bytes).

Upload file .docx < 2 MB: Halaman tidak me-reload. Pesan sukses "File berhasil diunggah." akan muncul di `div#status`.

6

Ubah kode menjadi agar dapat dilakukan multi upload file khusus gambar. Screenshoot perubahan kode dan jelaskan kode tersebut.

(Pertanyaan No. 11)



upload.js mencegah form melakukan submit normal (yang me-reload halaman) menggunakan `e.preventDefault()`.

FormData mengemas file untuk dikirim.

`$.ajax` mengirim file ke `upload_ajax.php` di latar belakang.

`upload_ajax.php` melakukan validasi di sisi server (ekstensi dan ukuran).

Respon dari server (sukses atau error) diterima oleh fungsi `success AJAX` dan ditampilkan di `div#status` tanpa perlu me-reload halaman.

--	--

Praktikum Bagian 4. Menghias Upload File

Langkah	Keterangan
1	<p>Modifikasi kode pada Praktikum Bagian 3 yaitu file <code>form_upload_ajax.php</code></p> <pre> <!DOCTYPE html> <html> <head> <link rel="stylesheet" type="text/css" href="upload.css"> <title>Unggah File Dokumen</title> </head> <body> <div class="upload-form-container"> <h2>Unggah File Dokumen</h2> <form id="upload-form" action="upload.php" method="post" enctype= "multipart/form-data"> <div class="file-input-container"> <input type="file" name="file" id="file" class="file-input"> <label for="file" class="file-label">Pilih File</label> </div> <button type="submit" name="submit" class="upload-button" id="upload-button" disabled>Unggah</button> </form> <div id="status" class="upload-status"></div> </div> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> <script src="upload.js"></script> </body> </html> </pre>
2	Modifikasi juga file <code>upload.js</code> seperti kode berikut.

```
$(document).ready(function(){
    $('#file').change(function(){
        if (this.files.length > 0) {
            $('#upload-button').prop('disabled', false).css('opacity', 1);
        } else {
            $('#upload-button').prop('disabled', true).css('opacity', 0.5);
        }
    });

    $('#upload-form').submit(function(e){
        e.preventDefault();

        var formData = new FormData(this);

        $.ajax({
            type: 'POST',
            url: 'upload_ajax.php',
            data: formData,
            cache: false,
            contentType: false,
            processData: false,
            success: function(response){
                $('#status').html(response);
            },
            error: function(){
                $('#status').html('Terjadi kesalahan saat mengunggah file.');
```

3

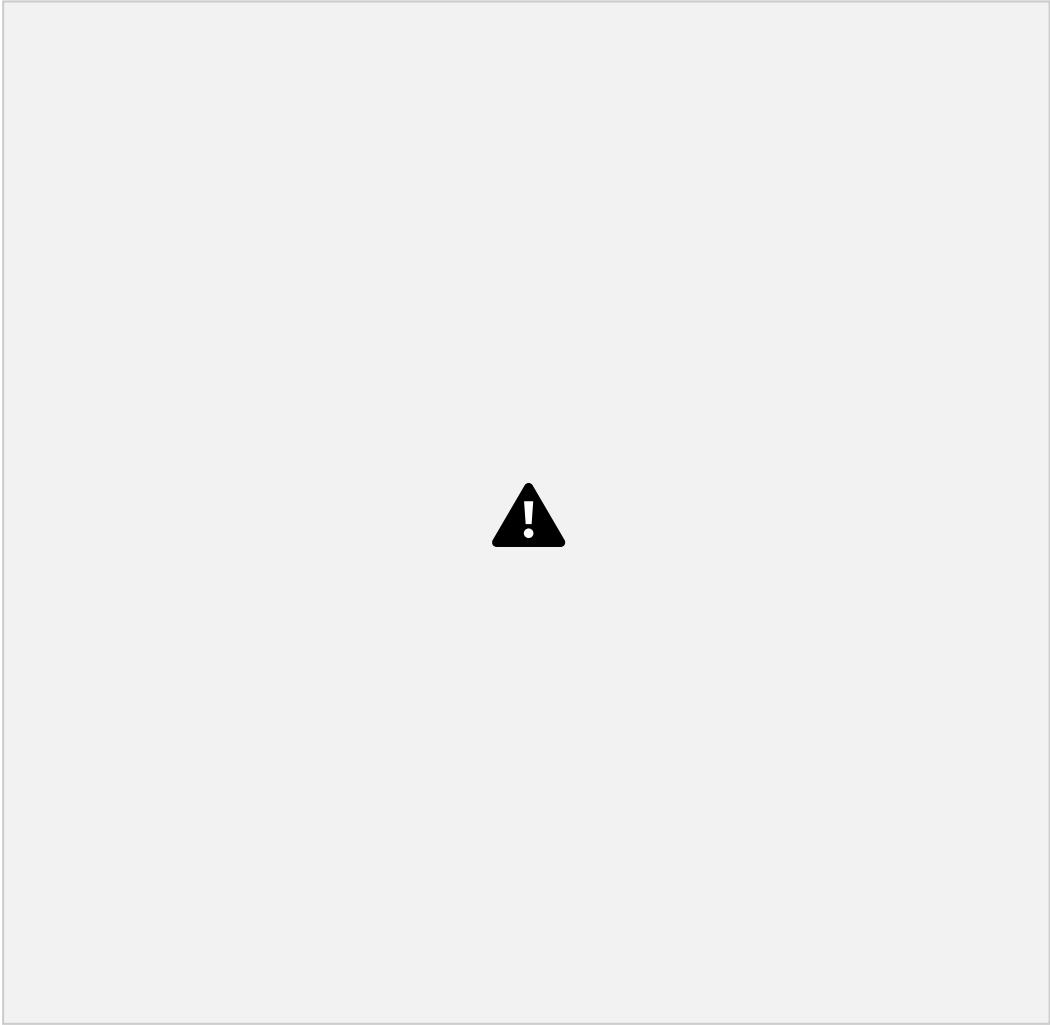
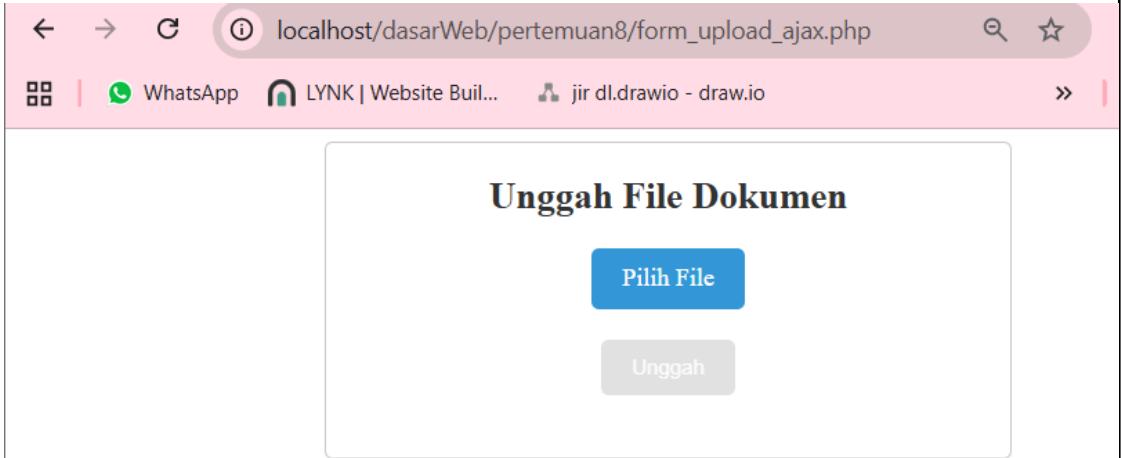
Buat file baru bernama upload.css. Ketikkan kode berikut..

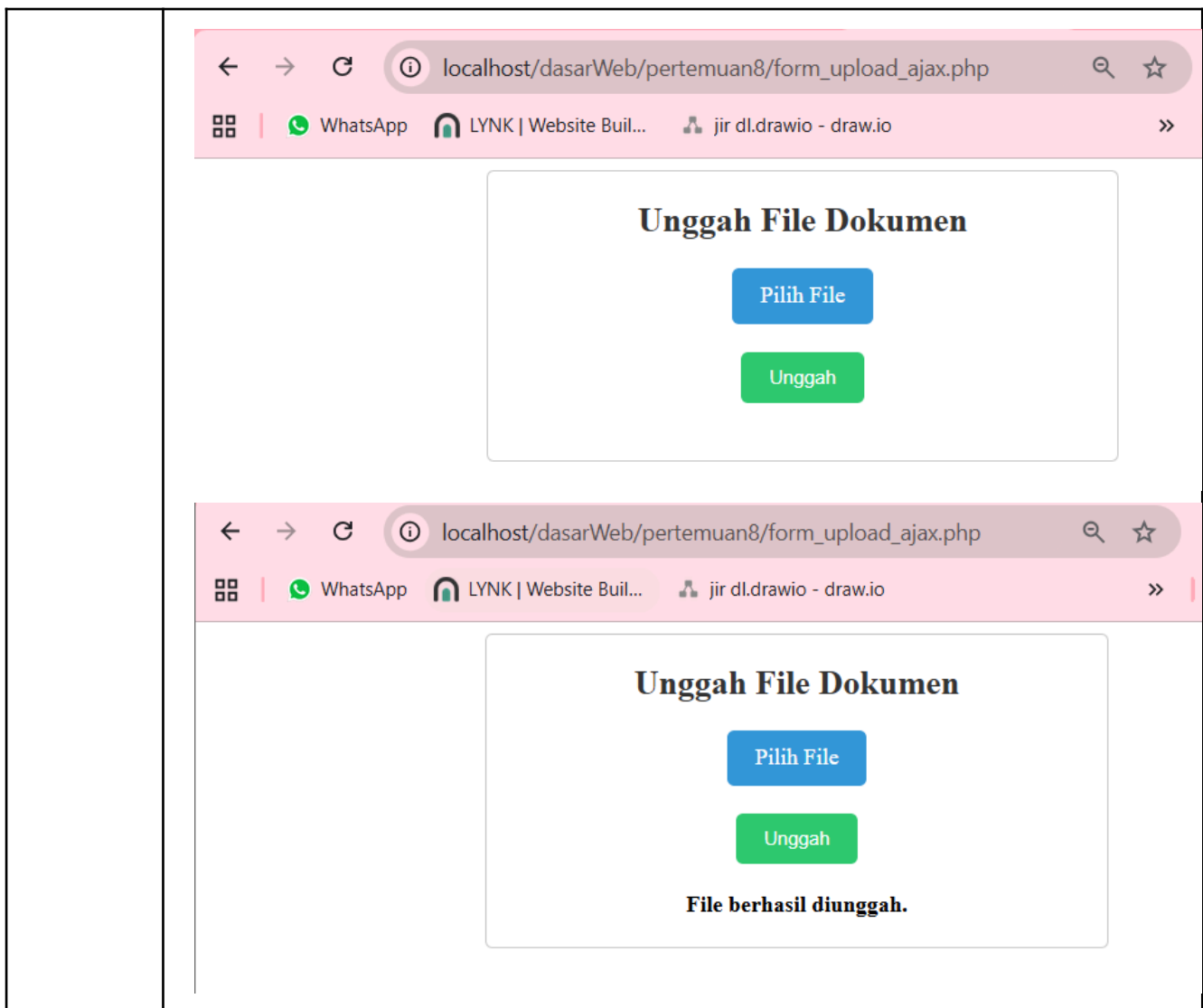
```
.upload-form-container {
  max-width: 480px;
  margin: 0 auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  text-align: center;
}

h2 {
  margin: 0;
  font-size: 24px;
  color: #333;
}

.file-input-container {
  display: flex;
  justify-content: center;
  align-items: center;
  margin: 20px 0;
}

.file-input {
  display: none;
}
```

	
4	<p>Simpan file tersebut. Buka browser dan jalankan localhost/dasarWeb/form_upload_ajax.php. Apa yang anda pahami dari kode program di atas? Catat pemahaman anda. (Pertanyaan No. 12)</p> 



Pengenalan Cookies

Cookies adalah sebuah nilai yang dikirimkan dan ditanamkan *server* pada komputer *client*. *Cookies* berupa berkas berukuran kecil umumnya kurang dari 100 bytes, sehingga tidak akan mempengaruhi kecepatan *browsing*. *Cookies* menampung informasi berkaitan dengan pengguna. *Cookies* berguna untuk memudahkan pengguna, seperti "mengingat" pengguna setiap kali mengunjungi website yang sama.

Informasi-informasi yang disimpan dalam *cookies* adalah data-data sederhana tentang kebiasaan pengguna. Contoh sederhananya saat pengguna mengunjungi sebuah website yang menawarkan tampilan dengan versi bahasa Inggris dan bahasa Indonesia. Apabila pengguna memilih bahasa Indonesia, *cookies* akan secara otomatis menyimpan konfigurasi tersebut. Konfigurasi tersebut akan diingat, sehingga apabila pengguna mengunjungi kembali website tersebut, pengguna tidak perlu memilih jenis bahasa lagi.

Cookies mempunyai umur, maksudnya setiap data yang disimpan dalam komputer pengguna suatu saat bisa hilang atau musnah. Berdasarkan umurnya *cookies* dibagi menjadi dua jenis, yaitu *session cookies* dan *persistent cookies*. *Session cookies* adalah jenis *cookies* yang tersimpan sementara dan akan hilang saat pengguna menutup *browser*. *Session cookies* ini biasanya digunakan pada fitur "keranjang belanja" di situs belanja online. *Persistent cookies* adalah jenis *cookies* yang tersimpan

secara permanen di dalam komputer sampai pengguna memutuskan untuk menghapusnya. *Persistent cookies* umumnya digunakan pada fitur “*remember me*” saat login.

PHP memiliki fungsi yang dapat digunakan untuk membuat dan mengambil nilai *cookies*. *Cookies* harus dideklarasikan sebelum halaman ditampilkan, yang artinya dituliskan sebelum tag `<html>`. *Cookies* dibuat menggunakan fungsi `setcookie()`. Berikut adalah sintaks dasar fungsi `setcookie()`:

```
<?php
    setcookie(name, value, expire);
?>
```


Keterangan:

- **name**, adalah nama *cookies* atau variabel yang menjadi pengenal *cookies*
- **value**, berisi nilai yang tersimpan dalam *cookies*
- **expire**, adalah jangka waktu *cookies* akan disimpan di dalam komputer

Setelah mengetahui cara membuat *cookies*, selanjutnya nilai *cookies* yang telah di buat dapat diambil dengan menggunakan variabel `$_COOKIE`. Berikut ini adalah sintaks untuk mengambil nilai *cookies*:

```
<?php
    $_COOKIE[ 'user' ];
?>
```


Praktikum Bagian 5. Membuat *Cookies*

Langkah	Keterangan
1	Buat file baru dengan nama <code>cookiesCreate.php</code> , kemudian ketikkan kode  berikut.
2	Buat file baru dengan nama <code>cookiesCall.php</code> , kemudian ketikkan kode  berikut.
3	Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan mengetikkan <code>localhost/dasarWeb/cookiesCall.php</code>
4	Amati dan jelaskan hasil pengamatanmu (Pertanyaan No. 13) Halaman akan menampilkan pesan seperti "User cookie is not set!". Pemahaman: Skrip <code>cookiesCall.php</code> mencoba membaca <code>\$_COOKIE['user']</code> . Karena skrip <code>cookiesCreate.php</code> belum pernah dijalankan, cookie tersebut belum ada di browser client. Fungsi <code>isset(\$_COOKIE['user'])</code> akan mengembalikan false.

5	Buka <i>browser</i> dan jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/cookiesCreate.php
6	Ulangi langkah ke-3.
7	<p>Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 14)</p> <p>Halaman akan menampilkan "Cookie 'user' is set!" dan "Value is: John Doe". Pemahaman: Menjalankan cookiesCreate.php (langkah 5) akan memanggil fungsi setcookie() dan menyimpan cookie bernama "user" dengan nilai "John Doe" di browser. Saat cookiesCall.php dijalankan lagi (langkah 6), isset(\$_COOKIE['user']) akan bernilai true, dan skrip akan menampilkan nilai cookie tersebut.</p>
8	<i>Restart</i> komputer Anda.
9	Setelah komputer menyala, nyalakan kembali Apache pada laragon.
10	Buka <i>browser</i> yang sama dengan sebelumnya kemudian ulangi langkah ke-3.
11	<p>Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 15)</p> <p>Halaman akan tetap menampilkan "Cookie 'user' is set!" dan "Value is: John Doe". Pemahaman: Cookie yang dibuat di Praktikum 5 adalah <i>persistent cookie</i> karena memiliki waktu kedaluwarsa (expire) yang diatur di masa depan (time() + (86400 * 30)) . Cookie ini disimpan secara permanen (hingga kedaluwarsa) di komputer client dan tidak akan hilang hanya dengan me-restart komputer atau menutup browser .</p>

Praktikum Bagian 6. Menghapus Nilai Cookies

Pada praktikum ini akan dibahas mengenai cara menghapus nilai *cookies*. Jika pada praktikum bagian sebelumnya *cookies* di set dengan waktu *expire* **time()+3600**, maka untuk menghapus nilai *cookies* adalah dengan sebagai berikut:

Langkah	Keterangan
1	<p>Buat file baru dengan nama cookiesDel.php, kemudian ketikkan kode</p> <div style="border: 1px solid black; padding: 10px; text-align: center;">  </div> <p>berikut.</p>
2	Buka <i>browser</i> dan jalankan kode program dengan mengetikkan localhost/dasarWeb/cookiesDel.php
3	Buka <i>browser</i> dan jalankan kode program dari praktikum bagian 5 dengan mengetikkan localhost/dasarWeb/cookiesCall.php

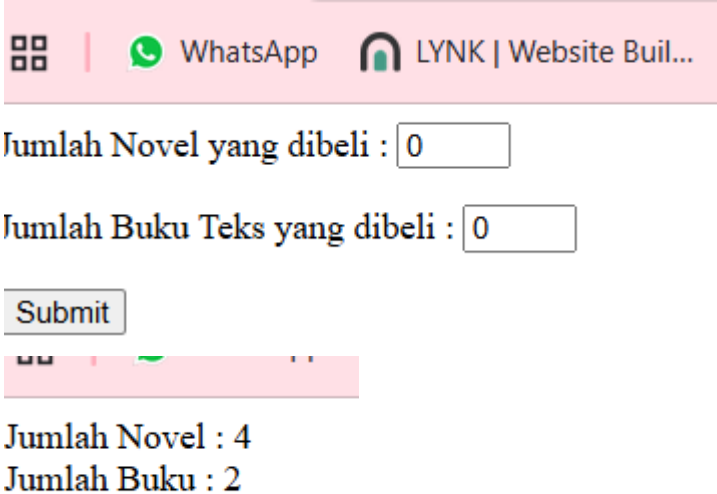
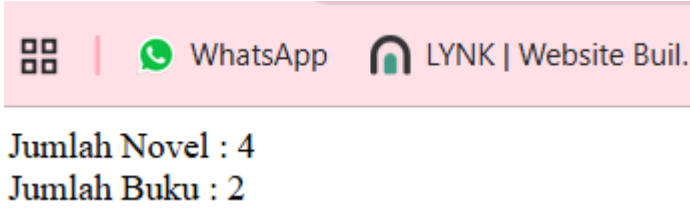
4	<p>Amati dan jelaskan hasil yang dihasilkan dari langkah 2 dan 3, kemudian beri kesimpulan. (Pertanyaan No. 16)</p> <p>Polinema</p> <p>Langkah 2 (menjalankan cookiesDel.php): Halaman akan menampilkan pesan seperti "User cookie is deleted."</p> <p>Langkah 3 (menjalankan cookiesCall.php): Halaman akan menampilkan "User cookie is not set!". Kesimpulan: Untuk menghapus cookie, kita menggunakan fungsi setcookie() dengan nama cookie yang sama, tetapi mengatur waktu kedaluwarsa (expire) ke waktu di masa lalu (misalnya time() - 3600). Ini adalah instruksi bagi browser untuk segera menghapus cookie tersebut.</p>
---	---

Praktikum Bagian 7. Penerapan *Cookies* pada Fitur Keranjang Belanja

Salah satu contoh pemanfaatan *cookies* adalah pada fitur “keranjang belanja” pada aplikasi web toko online. Keranjang belanja berisi barang-barang yang akan dibeli oleh user. *Cookies* digunakan untuk mengingat jumlah barang yang dipilih user. Berikut adalah contoh penggunaan *cookies* pada fitur keranjang belanja:

Langkah	Keterangan
1	<p>Buat file baru dengan nama formBeli.html, kemudian ketikkan kode berikut.</p> <div style="text-align: center;">  </div>
2	<p>Buat file baru dengan nama prosesBeli.php, kemudian ketikkan kode berikut.</p> <div style="text-align: center;">  </div>

3	<p>Buat file baru dengan nama keranjangBelanja.php, kemudian ketikkan kode</p> <div data-bbox="414 208 1393 716" data-label="Image"> </div> <p>berikut.</p>
4	<p>Buka <i>browser</i> dan jalankan kode program langkah ke-3 dengan mengetikkan localhost/dasarWeb/keranjangBelanja.php</p>
5	<p>Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 17)</p> <div data-bbox="316 976 1401 1126" data-label="Image"> </div> <p>Jumlah Novel : 0 Jumlah Buku : 0</p> <p>Halaman akan menampilkan "Jumlah Novel: 0" dan "Jumlah Buku Teks: 0". Pemahaman: Skrip keranjangBelanja.php mencoba membaca cookie beliNovel dan beliBuku. Karena kita belum pernah mengunjungi formBeli.html atau prosesBeli.php, cookie tersebut belum dibuat. isset() akan mengembalikan false, dan skrip akan menampilkan nilai default (0).</p>
6	<p>Jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/formBeli.html</p>
7	<p>Isikan jumlah novel dan buku teks yang akan dibeli kemudian klik tombol “submit”.</p>
8	<p>Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 18)</p>

	 <p>Jumlah Novel yang dibeli : 0</p> <p>Jumlah Buku Teks yang dibeli : 0</p> <p>Submit</p> <p>Jumlah Novel : 4 Jumlah Buku : 2</p> <p>Halaman akan dialihkan ke keranjangBelanja.php dan menampilkan jumlah barang sesuai input Anda (misal: "Jumlah Novel: 3" dan "Jumlah Buku Teks: 1"). Pemahaman: Form di formBeli.html mengirimkan data jumlah barang ke prosesBeli.php (via POST). Skrip prosesBeli.php kemudian menggunakan setcookie() untuk menyimpan data tersebut di browser. Setelah itu, prosesBeli.php mengarahkan (redirect) pengguna ke keranjangBelanja.php, yang kini dapat membaca dan menampilkan nilai dari cookie yang baru saja dibuat.</p>
9	<p>Tutup <i>browser</i> kemudian buka kembali <i>browser</i> kemudian jalankan kembali kode program langkah ke-3 dengan mengetikkan localhost/dasarWeb/keranjangBelanja.php</p>
10	<p>Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 19)</p>  <p>Jumlah Novel : 4 Jumlah Buku : 2</p> <p>Halaman akan kembali menampilkan "Jumlah Novel: 0" dan "Jumlah Buku Teks: 0". Pemahaman: Fitur keranjang belanja, seperti yang dijelaskan di pendahuluan jobsheet, biasanya menggunakan <i>session cookies</i>. <i>Session cookie</i> adalah cookie yang tidak memiliki waktu kedaluwarsa spesifik dan akan otomatis dihapus saat pengguna menutup browser. Saat browser ditutup, cookie <code>beliNovel</code> dan <code>beliBuku</code> terhapus.</p>

Pengenalan Session

Session adalah salah satu fasilitas yang ada pada PHP yang digunakan untuk menyimpan data sememntara ke dalam variabel, sehingga data tadi dapat diakses oleh pengguna selama variabel *session* tersebut tidak dikosongkan atau dihilangkan. *Session* pada prinsipnya hampir sama dengan *cookies*, yang membedakan adalah posisi penyimpanan data. Jika *cookies* menyimpan data pada *client*, data *session* akan disimpan pada sisi *server*. Jadi *session* relatif lebih aman digunakan untuk menyimpan

variabel nilai yang bersifat rahasia seperti *username* dan *password* pada saat login. Contoh penggunaan *session* diantaranya untuk menyimpan informasi login yang berlaku hanya dalam satu sesi dan menyimpan catatan pemesanan barang dalam sistem *e-commerce*/transaksi online. Tabel berikut menjelaskan perbandingan *cookies* dan *session*:

Perbandingan *Cookies* dan *Session*

Jenis	Menyimpan Informasi	Lokasi Penyimpanan	Dapat dipanggil	Bersifat Global	Dapat dipanggil Lain Waktu	Dibatasi Browser	Dapat diedit User
<i>Cookies</i>	Ya	<i>Client</i>	Ya	Ya	Ya	Ya	Ya
<i>Session</i>	Ya	<i>Server</i>	Ya	Ya	Tidak	Tidak	Tidak

Untuk membuat *session* dalam PHP digunakan fungsi **`session_start()`**. Berikut adalah sintaks dasar untuk memulai *session*:


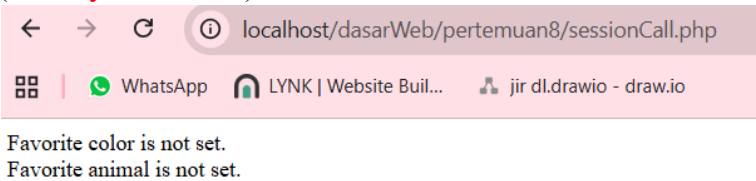
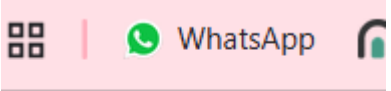
```
<?php
session_start();
?>
```

Setelah *session* dimulai, maka variabel *session* sudah dapat mulai digunakan. Variabel yang dimaksud adalah PHP global variable **`$_SESSION`**. Berikut adalah sintaks menggunakan variable global **`$_SESSION`**:

```
<?php
$_SESSION['namaVariabel'];
?>
```

Praktikum Bagian 8. Membuat *Session*

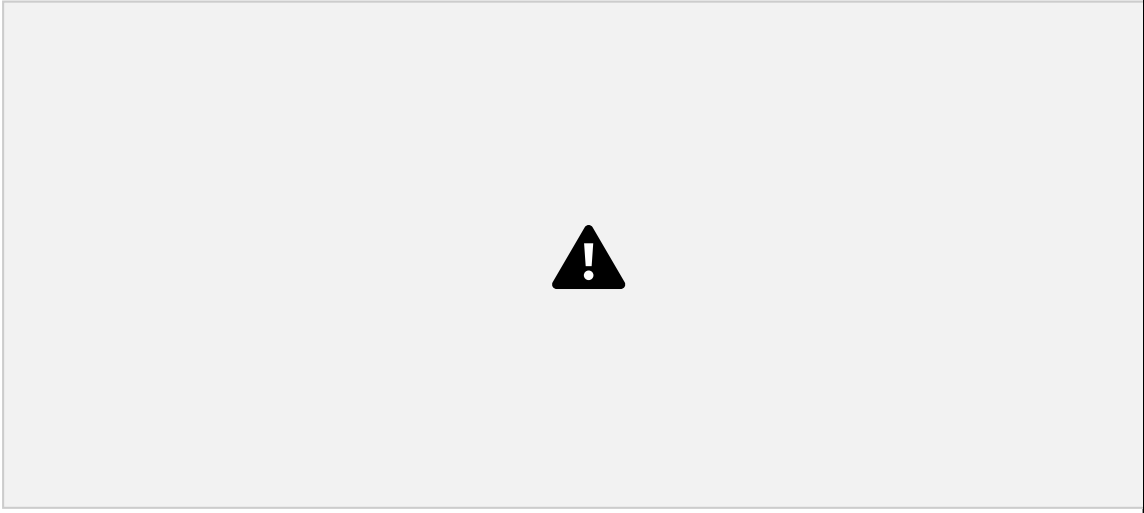
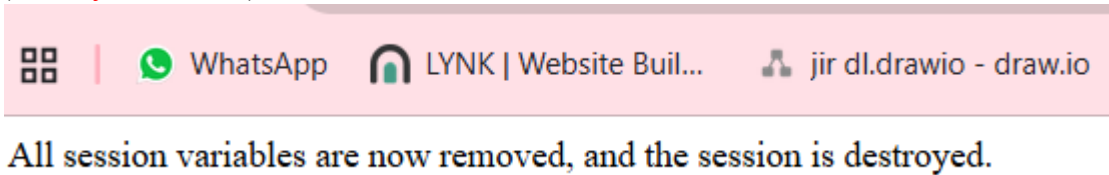
Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>sessionCreate.php</code>, kemudian ketikkan kode</p> <div style="text-align: center; height: 150px;">  </div> <p>berikut.</p>

2	<p>Buat file baru dengan nama sessionCall.php, kemudian ketikkan kode berikut.</p> <div style="text-align: center; height: 150px;">  </div>
3	<p>Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan mengetikkan localhost/dasarWeb/sessionCall.php</p>
4	<p>Buka <i>browser</i> dan jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/sessionCreate.php</p>
5	<p>Ulangi langkah ke-3</p>
6	<p>Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 20)</p> <div style="background-color: #f0f0f0; padding: 10px; margin-bottom: 10px;">  </div> <div style="background-color: #f0f0f0; padding: 10px; margin-bottom: 10px;">  </div> <p>Session variables are set.</p> <ul style="list-style-type: none"> • Langkah 3 (menjalankan sessionCall.php pertama): Halaman akan menampilkan error "Undefined index" atau pesan default (jika isset() digunakan). • Langkah 5 (menjalankan sessionCall.php kedua): Halaman akan menampilkan nilai yang di-set di sessionCreate.php (misal, jika kodenya \$_SESSION['user'] = "Hakim", maka akan tampil "User: Hakim"). Pemahaman: Berbeda dengan cookie, session disimpan di server. <ol style="list-style-type: none"> 1. Pada Langkah 3, session_start() dipanggil, tetapi tidak ada data untuk \$_SESSION['user'] di server, sehingga gagal dibaca. 2. Pada Langkah 4 (sessionCreate.php), session_start() dipanggil dan variabel (misal \$_SESSION['user']) dibuat dan disimpan di server.

	3. Pada Langkah 5, sessionCall.php dapat menemukan data session yang sudah ada di server dan menampilkannya.
--	--

Praktikum Bagian 9. Menghapus Nilai Session

PHP menyediakan fungsi `session_destroy()` yang dapat digunakan untuk menghapus *session*.


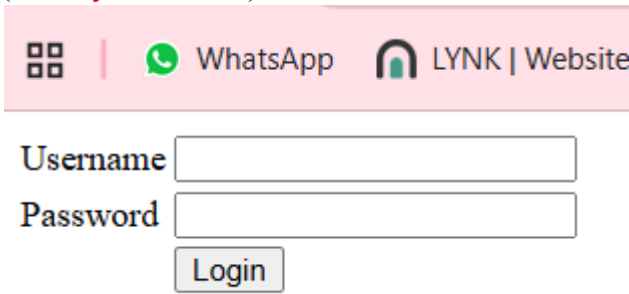
Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>sessionDel.php</code>, kemudian ketikkan kode berikut.</p> 
2	Buka <i>browser</i> dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/sessionDel.php</code>
3	Buka <i>browser</i> dan jalankan kode program dari praktikum bagian 8 dengan mengetikkan <code>localhost/dasarWeb/sessionCall.php</code>
4	<p>Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 21)</p> 

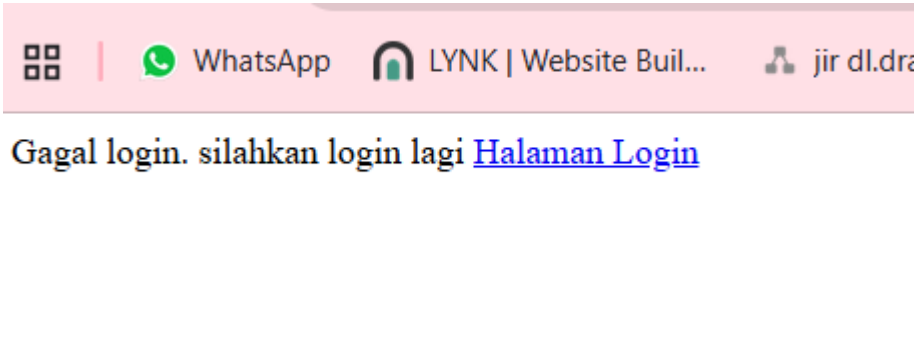
Praktikum Bagian 10. Penerapan Session pada Fitur Login

Langkah	Keterangan
1	Buat file baru dengan nama <code>sessionLoginForm.html</code> , kemudian ketikkan kode berikut

	
2	<p>Buat file dengan nama <code>sessionLoginProcess.php</code>, kemudian ketikkan kode berikut.</p> 

3	<p>Buat file dengan nama homeSession.php, kemudian ketikkan kode</p> <div data-bbox="414 208 1246 853">  </div> <p>berikut.</p>
---	---

4	<p>Buat file dengan nama sessionLogout.php, kemudian ketikkan kode</p> <div data-bbox="414 1016 1066 1256">  </div> <p>berikut.</p>
5	<p>Buka <i>browser</i> dan jalankan kode program dengan mengetikkan localhost/dasarWeb/sessionLoginForm.html</p>
6	<p>Login menggunakan username email Anda dan password “0000”.</p>
7	<p>Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 22)</p> <div data-bbox="316 1547 946 1839">  </div>

	 <p>Gagal login. silahkan login lagi Halaman Login</p>
8	<p>Jalankan kembali kode program dengan mengetikkan localhost/dasarWeb/sessionLoginForm.html Login menggunakan username “admin” dan password “1234”.</p>
9	<p>Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 23) Login akan berhasil. Halaman akan diarahkan ke homeSession.php. Halaman ini akan menampilkan pesan selamat datang (misal "Selamat Datang, admin") dan sebuah link "Logout". Pemahaman: Kredensial "admin" dan "1234" valid. Skrip sessionLoginProcess.php akan membuat variabel session (misal \$_SESSION['username'] = "admin") yang menandakan pengguna telah login, lalu mengarahkan ke homeSession.php.</p>
10	<p>Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (Pertanyaan No. 24) Login (sessionLoginForm.html) : Pengguna mengisi form, data dikirim ke sessionLoginProcess.php. Validasi (sessionLoginProcess.php) : Skrip ini memulai session (session_start()) . Ia memeriksa \$_POST data. Jika username == "admin" dan password == "1234", skrip akan mendaftarkan variabel session (misal \$_SESSION['username'] = "admin") lalu mengarahkan ke homeSession.php. Jika salah, skrip mengarahkan kembali ke sessionLoginForm.html. Halaman Terproteksi (homeSession.php) : Skrip ini memulai session_start(). Ia memeriksa apakah \$_SESSION['username'] ada (isset()). Jika ada, halaman ditampilkan (termasuk link ke sessionLogout.php). Jika tidak ada (pengguna belum login), skrip akan mengarahkan paksa pengguna kembali ke sessionLoginForm.html. Logout (sessionLogout.php) : Skrip ini memulai session_start(), lalu memanggil session_destroy() untuk menghapus semua data session di server. Terakhir, skrip mengarahkan pengguna kembali ke sessionLoginForm.html.</p>