**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:



* Buat skrip PHP (**upload.php**) untuk mengelola pengunggahan 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 **dasarWeb**, beri nama **form\_upload.php**. |
| 2 | Ketikkan ke dalam file **form\_upload.php** kode di bawah ini. |
| 3 | Buat file baru bernama **upload.php** yang akan digunakan untuk pemrosesan  **form\_upload.php**. |
|  |  |
| 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) |
| 5 | 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) |
| 6 | Ubah isi dari file **upload.php** dengan kode berikut |
| 7 | 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) |

|  |  |
| --- | --- |
|  |  |
| 8 | 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) |
| 9 | 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) |
| 10 | Selanjutnya, ubah isi dari file **upload.php** dengan kode berikut. |
| 11 | 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) |

|  |  |
| --- | --- |
| 12 | 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) |

**Praktikum Bagian 2. Multi Upload File**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah file baru bernama **form\_multiupload.php**. |
| 2 | Ketikkan ke dalam **form\_multiupload.php** kode berikut. |
| 3 | Buatlah file baru bernama **proses\_upload.php**. Ketikkan kode berikut. |

|  |  |
| --- | --- |
| 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) |
| 5 | Buat seperti langkah 3 dengan multi upload khusus gambar.  Screen shoot perubahan kode dan berikan penjelasan kode tersebut. (Pertanyaan No. 9) |

**Praktikum Bagian 3. Upload File dengan PHP dan Jquery**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah file baru bernama **form\_upload\_ajax.php** |
| 2 | Ketikkan kode berikut pada **form\_upload\_ajax.php** |
| 3 | Buatlah file baru bernama **upload.js**. Tuliskan kode berikut. |

|  |  |
| --- | --- |
|  |  |
| 4 | Buatlah file baru bernama **upload\_ajax.php**. Tuliskan kode berikut. |
| 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) |
| 6 | Ubah kode menjadi agar dapat dilakukan multi upload file khusus gambar. Screenshoot perubahan kode dan jelaskan kode tersebut.  (Pertanyaan No. 11) |

**Praktikum Bagian 4. Menghias Upload File**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Modifikasi kode pada Praktikum Bagian 3 yaitu file **form\_upload\_ajax.php** |
| 2 | Modifikasi juga file **upload.js** seperti kode berikut. |

|  |  |
| --- | --- |
|  |  |
| 3 | Buat file baru bernama upload.css. Ketikkan kode berikut.. |

|  |  |
| --- | --- |
|  |  |
| 4 | 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) |

**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 infomasi 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 mengujungi 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 **cookiesCreate.php**, kemudian ketikkan kode berikut.  A close up of a text  Description automatically generated |
| 2 | Buat file baru dengan nama **cookiesCall.php**, kemudian ketikkan kode berikut.  A close-up of a sign  Description automatically generated |
| 3 | Buka *browser* dan jalankan kode program pada langkah ke-2 dengan mengetikkan  **localhost/dasarWeb/cookiesCall.php** |
| 4 | Amati dan jelaskan hasil pengamatanmu (Pertanyaan No. 13) |

|  |  |
| --- | --- |
| 5 | Buka *browser* dan jalankan kode program langkah ke-1 dengan mengetikkan  **localhost/dasarWeb/cookiesCreate.php** |
| 6 | Ulangi langkah ke-3. |
| 7 | Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 14) |
| 8 | *Restart* komputer Anda. |
| 9 | Setelah komputer menyala, nyalakan kembali Apache pada laragon. |
| 10 | Buka *browser* yang sama dengan sebelumnya kemudian ulangi langkah ke-3. |
| 11 | Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 15) |

# 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 | Buat file baru dengan nama **cookiesDel.php**, kemudian ketikkan kode berikut.  A close up of a text  Description automatically generated |
| 2 | Buka *browser* dan jalankan kode program dengan mengetikkan  **localhost/dasarWeb/cookiesDel.php** |
| 3 | Buka *browser* dan jalankan kode program dari praktikum bagian 5 dengan mengetikkan  **localhost/dasarWeb/cookiesCall.php** |
| 4 | Amati dan jelaskan hasil yang dihasilkan dari langkah 2 dan 3, kemudian beri kesimpulan. (Pertanyaan No. 16) |

# 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 | Buat file baru dengan nama **formBeli.html**, kemudian ketikkan kode berikut.  A computer code with text  Description automatically generated |
| 2 | Buat file baru dengan nama **prosesBeli.php**, kemudian ketikkan kode berikut.  A computer code with black text  Description automatically generated |
| 3 | Buat file baru dengan nama **keranjangBelanja.php**, kemudian ketikkan kode berikut.  A screen shot of a computer code  Description automatically generated |
| 4 | Buka *browser* dan jalankan kode program langkah ke-3 dengan mengetikkan  **localhost/dasarWeb/keranjangBelanja.php** |
| 5 | Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 17) |
| 6 | Jalankan kode program langkah ke-1 dengan mengetikkan  **localhost/dasarWeb/formBeli.html** |

|  |  |
| --- | --- |
| 7 | Isikan jumlah novel dan buku teks yang akan dibeli kemudian klik tombol “submit”. |
| 8 | Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 18) |
| 9 | Tutup *browser* kemudian buka kembali *browser* kemudian jalankan kembali kode program langkah ke-3 dengan mengetikkan **localhost/dasarWeb/keranjangBelanja.php** |
| 10 | Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 19) |

**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*** |
| *Cookies* | Ya | *Client* | Ya | Ya | Ya | Ya | Ya |
| *Session* | Ya | *Server* | 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 penggunakan variable global

**$\_SESSION**:

<?php

$\_SESSION[‘namaVariabel’];

?>

**Praktikum Bagian 8. Membuat *Session***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama **sessionCreate.php**, kemudian ketikkan kode berikut.  A screen shot of a computer code  Description automatically generated |
| 2 | Buat file baru dengan nama **sessionCall.php**, kemudian ketikkan kode berikut.  A screenshot of a computer code  Description automatically generated |
| 3 | Buka *browser* dan jalankan kode program pada langkah ke-2 dengan mengetikkan  **localhost/dasarWeb/sessionCall.php** |
| 4 | Buka *browser* dan jalankan kode program langkah ke-1 dengan mengetikkan  **localhost/dasarWeb/sessionCreate.php** |
| 5 | Ulangi langkah ke-3 |
| 6 | Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 20) |

# Praktikum Bagian 9. Menghapus Nilai *Session*

PHP menyediakan fungsi **session\_destroy()** yang dapat digunakan untuk menghapus *session*.

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama **sessionDel.php**, kemudian ketikkan kode berikut.  A white screen with black text  Description automatically generated |
| 2 | Buka *browser* dan jalankan kode program dengan mengetikkan  **localhost/dasarWeb/sessionDel.php** |
| 3 | Buka *browser* dan jalankan kode program dari praktikum bagian 8 dengan mengetikkan  **localhost/dasarWeb/sessionCall.php** |
| 4 | Amati dan jelaskan hasil yang ditampilkan. (Pertanyaan No. 21) |

# Praktikum Bagian 10. Penerapan *Session* pada Fitur Login

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama **sessionLoginForm.html**, kemudian ketikkan kode berikut |

|  |  |
| --- | --- |
|  |  |
| 2 | Buat file dengan nama **sessionLoginProcess.php**, kemudian ketikkan kode berikut. |
| 3 | Buat file dengan nama **homeSession.php**, kemudian ketikkan kode berikut. |

|  |  |
| --- | --- |
| 4 | Buat file dengan nama **sessionLogout.php**, kemudian ketikkan kode berikut. |
| 5 | Buka *browser* dan jalankan kode program dengan mengetikkan  **localhost/dasarWeb/sessionLoginForm.html** |
| 6 | Login menggunakan username email Anda dan password “0000”. |
| 7 | Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 22) |
| 8 | Jalankan kembali kode program dengan mengetikkan  **localhost/dasarWeb/sessionLoginForm.html**  Login menggunakan username “admin” dan password “1234”. |
| 9 | Amati dan jelaskan hasil yang ditampilkan (Pertanyaan No. 23) |
| 10 | Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (Pertanyaan No. 24) |