



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. <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>
3	Buat file baru bernama <code>upload.php</code> yang akan digunakan untuk pemrosesan <code>form_upload.php</code> .

	<pre><?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."; } } ?></pre>
4	<p>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)</p>
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>
6	<p>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("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>

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.</p> <p>(Pertanyaan No. 4)</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.</p> <p>Screenshoot tambahan kode. Jelaskan pemahaman Anda setelah penambahan kode program tersebut.</p> <p>(Pertanyaan No. 5)</p>
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.</p> <p>(Pertanyaan No. 6)</p>

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.
	<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	Buatlah file baru bernama proses_upload.php . Ketikkan kode berikut.
	<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)
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
	<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 upload.js . 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)
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	<p>Modifikasi kode pada Praktikum Bagian 3 yaitu file form_upload_ajax.php</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 upload.js 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: 400px;
    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;
}

```

	<pre>.file-label { background: #3498db; color: #fff; padding: 10px 20px; border-radius: 5px; cursor: pointer; } .upload-button { background: #2ecc71; color: #fff; padding: 10px 20px; border: none; border-radius: 5px; cursor: pointer; opacity: 0.5; /* Opacity to make it appear faded */ } .upload-button:disabled { background: #ccc; /* Change color when disabled */ cursor: not-allowed; /* Change cursor style when disabled */ } .upload-status { margin-top: 20px; font-weight: bold; }</pre>
4	<p>Simpan file tersebut. Buka browser dan jalankan localhost/dasarWeb/form_upload_ajax.php.</p> <p>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 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 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 dibuat 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. <pre>1 <?php 2 setcookie("user", "Polinema", time() + 3600); 3 ?></pre>
2	Buat file baru dengan nama cookiesCall.php , kemudian ketikkan kode berikut. <pre>1 <?php 2 echo \$_COOKIE['user']; 3 ?></pre>
3	Buka <i>browser</i> 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 <i>browser</i> 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	<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	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. <pre> 1 <?php 2 setcookie("user", "Polinema", time() -3600); 3 ?></pre>
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	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. <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="prosesBeli.php" method="POST"> 6 <p> Jumlah Novel yang dibeli : 7 <input type="text" name="beliNovel" value= "0" size="2"> </p> 8 <p> Jumlah Buku Teks yang dibeli : 9 <input type="text" name="beliBuku" value= "0" size="2"> </p> 10 <input type="submit"> 11 </form> 12 </body> 13</html></pre>
2	Buat file baru dengan nama prosesBeli.php , kemudian ketikkan kode berikut. <pre> 1 <?php 2 if(isset(\$_POST["beliNovel"]) && isset(\$_POST["beliBuku"])){ 3 setcookie("beliNovel", \$_POST["beliNovel"]); 4 setcookie("beliBuku", \$_POST["beliBuku"]); 5 header("location:keranjangBelanja.php"); 6 } 7 ?></pre>
3	Buat file baru dengan nama keranjangBelanja.php , kemudian ketikkan kode berikut. <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <h2> Keranjang Belanja </h2> 6 7 <?php 8 \$beliNovel=\$_COOKIE['beliNovel']; 9 \$beliBuku = \$_COOKIE['beliBuku']; 10 11 echo "Jumlah Novel:" . \$beliNovel . "
"; 12 echo "Jumlah Buku :" . \$beliBuku ; 13 14 ?> 15 </body> 16</html></pre>
4	Buka <i>browser</i> 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 <i>browser</i> kemudian buka kembali <i>browser</i> 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
<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 penggunaan 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. <pre> 1 <?php 2 session_start(); 3 ?> 4 5 <!DOCTYPE html> 6 <html> 7 <body> 8 <?php 9 \$_SESSION["favcolor"] = "green"; 10 \$_SESSION["favanimal"] = "cat"; 11 echo "Session variables are set."; 12 ?> 13 </body> 14 </html> </pre>
2	Buat file baru dengan nama sessionCall.php , kemudian ketikkan kode berikut. <pre> 1 <?php 2 session_start(); 3 ?> 4 <!DOCTYPE html> 5 <html> 6 <body> 7 <?php 8 echo "Favorite color is " . \$_SESSION["favcolor"] . ".
"; 9 echo "Favorite animal is " . \$_SESSION["favanimal"] . "."; 10 ?> 11 </body> 12 </html> </pre>
3	Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan mengetikkan localhost/dasarWeb/sessionCall.php
4	Buka <i>browser</i> 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.  <pre>1 <?php 2 session_start(); 3 ?> 4 5 <!DOCTYPE html> 6 <html> 7 <body> 8 <?php 9 session_unset(); 10 session_destroy(); 11 12 echo "All session variables are now removed, and the session is destroyed." 13 ?> 14 </body> 15 </html></pre>
2	Buka <i>browser</i> dan jalankan kode program dengan mengetikkan localhost/dasarWeb/sessionDel.php
3	Buka <i>browser</i> 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

```

<html>
    <head>
        <title>File Upload</title>
    </head>
    <body>
        <form action="sessionLoginProcess.php" method="POST">
            <table>
                <tr>
                    <td>Username</td>
                    <td><input type="text" name="username" size="20"></td>
                </tr>
                <tr>
                    <td>Password</td>
                    <td><input type="password" name="password" size="20"></td>
                </tr>
                <tr>
                    <td>&nbsp;</td>
                    <td><input type="submit" name="login" value="Login"></td>
                </tr>
            </table>
        </form>
    </body>
</html>

```

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

```

<?php
$username = $_POST['username'];
$password = $_POST['password'];

if($username=="admin" && $password=="1234"){
    session_start();
    $_SESSION["username"] = $username;
    $_SESSION["status"] = 'login';
    echo "Anda berhasil login. Silahkan menuju <a href='homeSession.php'>Halaman Home</a>";
}
else{
    echo "Gagal login. Silahkan login lagi <a href='sessionLoginForm.html'>Halaman Login</a>";
}
?>

```

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

```

<html>
    <head>
        <title>Halaman Home</title>
    </head>
    <body>
        <?php
            session_start();
            if($_SESSION['status']=='login'){
                echo "Selamat datang " . $_SESSION['username'];
            }
            <br><a href="sessionLogout.php">Logout</a>
        <?php
            }
            else{
                echo "Anda belum login, silahkan";
            }
            <a href="sessionLoginForm.html">Login</a>
        <?php
            }
        ?>
    </body>
</html>

```

4	<p>Buat file dengan nama sessionLogout.php, kemudian ketikkan kode berikut.</p> <pre><?php session_start(); session_destroy(); echo "Anda berhasil logout"; ?></pre>
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>
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)</p>
10	<p>Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (Pertanyaan No. 24)</p>