



# Hypertext Preprocessor (PHP)

## Pemrograman Web

Danny Sebastian, S.Kom., M.M., M.T.

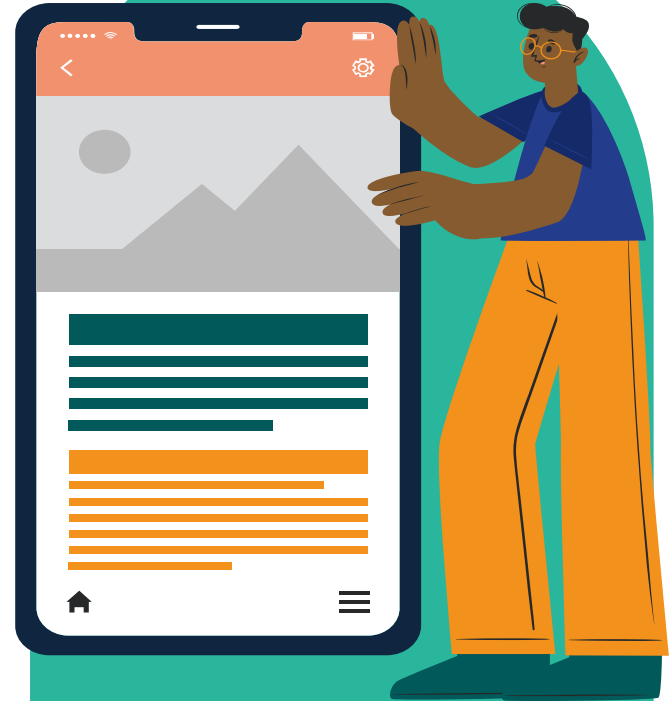
Maria Nila Anggia Rini, S.T., M.Tl.

Agata Filiana, S.Kom., M.Sc.

**INFORMATIKA UKDW GENAP 2022/2023**

# TOPIK HARI INI

- ★ **Include**
- ★ **Upload File**
- ★ **Aktivitas Kelas**



# PHP INCLUDE

Include merupakan fitur yang dapat anda gunakan untuk memakai sebuah kode lebih dari satu kali.

## Contoh kasus:

CRUD yang sudah kita lakukan, setiap halaman yang terkoneksi pada database harus dibuka koneksinya pada kode kita menggunakan **mysqli\_connect**.

Kode ini diulang-ulang di beberapa halaman. Hal ini bisa dipersingkat dengan meletakkan kode untuk koneksi database pada satu halaman PHP dan dipanggil pada halaman-halaman yang membutuhkan.

Pemakaian include ini sangat disarankan karena kode dapat dipakai ulang (**code reusability**), dan jika ada kesalahan maka cukup mengganti pada satu halaman kode saja.

# PHP INCLUDE

Untuk melakukan hal tersebut:

1. Letakkan kode koneksi database pada sebuah halaman PHP baru, misalnya: **dbconnect.php**
2. Untuk memanggil halaman tersebut, ada dua cara yaitu dengan **include** atau **require**. Perbedaan dari keduanya ada pada perlakuan ketika terjadi error.
  - a. Ketika ada error pada include, maka akan terjadi peringatan namun script akan tetap jalan.
  - b. Ketika ada error pada require maka script akan berhenti.

# PHP INCLUDE: CONTOH

Isi dari halaman **dbconnect.php**:

```
dbconnect.php > ...
1  <?php
2      //KONEKSI
3      $servername="localhost";
4      $username="root";
5      $password="";
6      $dbname="progweb";
7      $conn= mysqli_connect($servername,$username,$password,$dbname) or die("Koneksi gagal");
8  ?>
```

# PHP INCLUDE: CONTOH

Cara memanggil dbconnect.php pada halaman lain, contoh pada halaman delete-mahasiswa.php:

```
delete-mahasiswa.php > ...
1  <?php
2  include "dbconnect.php";
3
4  if($_GET){
5      $id=$_GET['id'];
6      $sql="SELECT * from mahasiswa where id=".$id;
7      $result=mysqli_query($conn,$sql);
8      if(mysqli_num_rows($result)>0){
9          $sqldel="DELETE FROM mahasiswa where id=".$id;
```

# FILE HANDLING

Ada saatnya ketika dibutuhkan akses pada file-file tertentu pada server, misalnya mengambil file gambar. Disarankan untuk penamaan file menggunakan lowercase karena beberapa sistem (seperti contohnya Linux) menganggap nama file case-sensitive. Untuk lebih aman, selalu gunakan lower case dan jangan beri spasi atau karakter yang tidak umum.



# FILE HANDLING

Untuk cek apakah sebuah file ada atau tidak, dapat digunakan fungsi `file_exists()`:

```
<?php
    if(file_exists("agata.txt")) echo "File ada!";
?>
```





# FILE HANDLING

Pada PHP, kita juga dapat membuat sebuah file baru:

```
<?php
    $fh = fopen("testfile.txt","w") or die("Failed to create a file");

    $text = <<<_END
    Halo ini file percobaan
    untuk kelas teori progweb!
    _END;

    fwrite($fh, $text) or die("Unable to write to file");
    fclose($fh);
    echo "File successfully created!";
?>
```

# FILE HANDLING

Pada PHP, kita juga dapat membuat sebuah file baru:

- `fopen()` digunakan untuk membuka file.
- `fwrite()` digunakan untuk menulis pada file. Anda juga dapat menggunakan `fread()` atau `fgets()` untuk membaca file.
- `fclose()` digunakan untuk menutup file.



# FILE HANDLING

Terdapat beberapa mode fopen():

Mode	Action	Description
'r'	Read from file's beginning	Open for reading only; place the file pointer at the beginning of the file. Return FALSE if the file doesn't already exist.
'r+'	Read from file's beginning and allow writing	Open for reading and writing; place the file pointer at the beginning of the file. Return FALSE if the file doesn't already exist.
'w'	Write from file's beginning and truncate file	Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file doesn't exist, attempt to create it.
'w+'	Write from file's beginning, truncate file, and allow reading	Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file doesn't exist, attempt to create it.
'a'	Append to file's end	Open for writing only; place the file pointer at the end of the file. If the file doesn't exist, attempt to create it.
'a+'	Append to file's end and allow reading	Open for reading and writing; place the file pointer at the end of the file. If the file doesn't exist, attempt to create it.

# FILE HANDLING

fgets() dapat digunakan untuk membaca dari file. Kode berikut ini akan mengambil baris pertama dari file:

```
$fh=fopen("testfile.txt",'r') or die("Failed to open a file");
```

```
$line = fgets($fh);
```

```
fclose($fh);
```

```
echo $line;
```



# FILE HANDLING

fgets() dapat digunakan untuk membaca dari file. Kode berikut ini akan mengambil baris pertama dari file:

```
$fh=fopen("testfile.txt",'r') or die("Failed to open a file");
```

```
$line = fgets($fh);
```

```
fclose($fh);
```

```
echo $line;
```



# FILE HANDLING

fgets() dapat digunakan untuk membaca dari file. Kode berikut ini akan mengambil tiga karakter pertama dari file:

```
$fh=fopen("testfile.txt",'r') or die("Failed to open a file");  
  
$text = fread($fh,3);  
fclose($fh);  
echo $text;
```




# FILE HANDLING

Untuk mendapatkan semua isi file maka dapat dilakukan looping dengan memanfaatkan `feof()` yang akan mengecek apakah sudah di akhir sebuah file atau belum.

```
$fh=fopen("testfile.txt",'r') or die("Failed to open a file");
```

```
while(! feof($fh)) {  
    $result = fgets($fh);  
    echo $result;  
}
```

```
fclose($fh);
```



# FILE HANDLING

Untuk copy sebuah file:

```
copy("testfile.txt","testfile2.txt") or die("Couldn't copy file");  
if(file_exists("testfile2.txt")) echo "File sudah dicopy!";
```





# FILE HANDLING

Untuk move sebuah file:

```
rename("testfile.txt","images/testfile.txt");  
if(file_exists("images/testfile.txt")) echo "File sudah dipindah!";
```



# FILE HANDLING

Untuk hapus sebuah file:

```
unlink("images/testfile.txt");  
if(!file_exists("images/testfile.txt")) echo "File berhasil dihapus!";
```



# PHP UPLOAD

Upload file merupakan sebuah fitur yang sering dibutuhkan pada aplikasi website, contoh paling sederhana adalah mengunggah foto profil.

Untuk melakukan upload file pada PHP, ada beberapa hal yang harus diperhatikan:

- Siapkan sebuah form dengan `input type="file"`. Atribut `type="file"` pada tag `<input>` menspesifikasikan input akan diproses sebagai file. Ketika ditampilkan pada browser, maka akan terdapat tombol "browse" di sebelah field inputan
- Pastikan method yang dipakai adalah POST.
- Pastikan `enctype = "multipart/form-data"`. Ini berhubungan dengan tipe konten yang dikirimkan dan dipakai ketika form memerlukan binary data, seperti isi file yang diupload.

# PHP UPLOAD

- Anda perlu menyiapkan sebuah folder pada server untuk menyimpan file yang akan diunggah.
- Apabila ingin menyimpan pada database, maka gunakan tipe data BLOB.



# PHP UPLOAD

- Dengan menggunakan array global PHP `$_FILES`, kita bisa mengunggah file dari komputer client ke server. Parameter pertama adalah nama input dari form yang memiliki atribut `type="file"`. Pada contoh ini namanya adalah "upload" dan parameter kedua bisa diisi dengan:
  - `$_FILES["upload"]["name"]` □ nama file yang diupload
  - `$_FILES["upload"]["type"]` □ jenis file yang diupload
  - `$_FILES["upload"]["size"]` □ ukuran file yang diupload dalam satuan byte
  - `$_FILES["upload"]["tmp_name"]` □ alamat penyimpanan sementara sebelum dipindahkan ke tempat tujuan

# PHP UPLOAD

Menyiapkan form untuk upload file:

```
<form action="uploadform.php" method="POST" enctype="multipart/form-data">  
  <label>Upload file:</label>  
  <input type="file" name="upload" id="file"> <br>  
  <input type="submit" value="submit" name="submit">  
</form>
```

# PHP UPLOAD

Untuk melihat keterangan dari file yang diunggah, maka dapat menggunakan kode PHP berikut ini:

```
<?php
    if(isset($_FILES["upload"]["name"]))
    {
        echo "Keterangan file: <br>";
        echo "Nama file: " . $_FILES["upload"]["name"] . " <br>";
        echo "Tipe file: " . $_FILES["upload"]["type"] . " <br>";
        echo "Ukuran file: " . ($_FILES["upload"]["size"]/1024) . " KB <br>";
        echo "Disimpan pada: " . $_FILES["upload"]["tmp_name"] . " <br>";
    }
?>
```

# PHP UPLOAD

Hasilnya:

Upload file:  No file chosen

Keterangan file:

Nama file: 20190925\_153718.jpg

Tipe file: image/jpeg

Ukuran file: 660.62109375 KB

Disimpan pada: C:\xampp\tmp\php12D8.tmp



# PHP UPLOAD

Saat ini file sudah tersimpan secara temporary pada folder yang ada pada xampp, sekarang kita akan pindahkan ke tempat penyimpanan yang kita inginkan. Untuk itu, perlu ditambahkan kode berikut ini:

```
$uploadfile = "images/".$_FILES["upload"]["name"];  
if(move_uploaded_file($_FILES["upload"]["tmp_name"],$uploadfile)){  
    echo "Sukses upload";  
}else{  
    echo "Tidak berhasil upload";  
}
```

# PHP UPLOAD: CEK FILE SUDAH ADA

Gunakan `file_exists()` untuk cek apakah file yang akan diunggah sudah ada belum.

```
if(file_exists($uploadfile)){  
    echo "Sudah ada file ini!";  
}else{
```



# PHP UPLOAD: CEK SIZE DAN TIPE FILE

Dengan memanfaatkan `$_FILES["upload"]["type"]` dan `$_FILES["upload"]["size"]` maka anda dapat menentukan tipe-tipe file tertentu untuk diupload dan juga membatasi size yang diunggah.

Berikut ini salah satu contoh untuk membatasi mengunggah file dengan tipe tertentu:

```
$tipefile = strtolower(pathinfo($uploadfile,PATHINFO_EXTENSION));  
if($tipefile != "jpg" && $tipefile != "png" && $tipefile != "jpeg") {  
echo "Hanya bisa JPG, PNG, dan JPEG!";  
}else{
```

# PHP UPLOAD KE DATABASE

Selain menyimpan file pada server, anda juga dapat menyimpannya pada database. Untuk kasus ini, pastikan pada kolom yang diinginkan, anda memakai tipe file BLOB yang khusus untuk binary data.

```
<?php
include "dbconnect.php";
if(isset($_FILES["upload"]["name"])){
    echo "Keterangan file: <br>";
    echo "Nama file: " . $_FILES["upload"]["name"] . " <br>";
    echo "Tipe file: " . $_FILES["upload"]["type"] . " <br>";
    echo "Ukuran file: " . ($_FILES["upload"]["size"]/1024) . " KB <br>";
    echo "Disimpan pada: " . $_FILES["upload"]["tmp_name"] . " <br>";

    $dokumen = addslashes(file_get_contents($_FILES['upload']['tmp_name']));

    $sql = "INSERT INTO fileblob (dokumen, tipe_file) VALUES('" . $dokumen . "', '" . ($_FILES["upload"]["size"]/1024) . "')";
    if(mysqli_query($conn,$sql)){
        echo "Berhasil";
    }else{
        echo "Gagal";
    }
}else{
    echo "Masih kosong";
}
mysqli_close($conn);
?>
```



# **AKTIVITAS KELAS**

# THANKS!

Ada pertanyaan?

