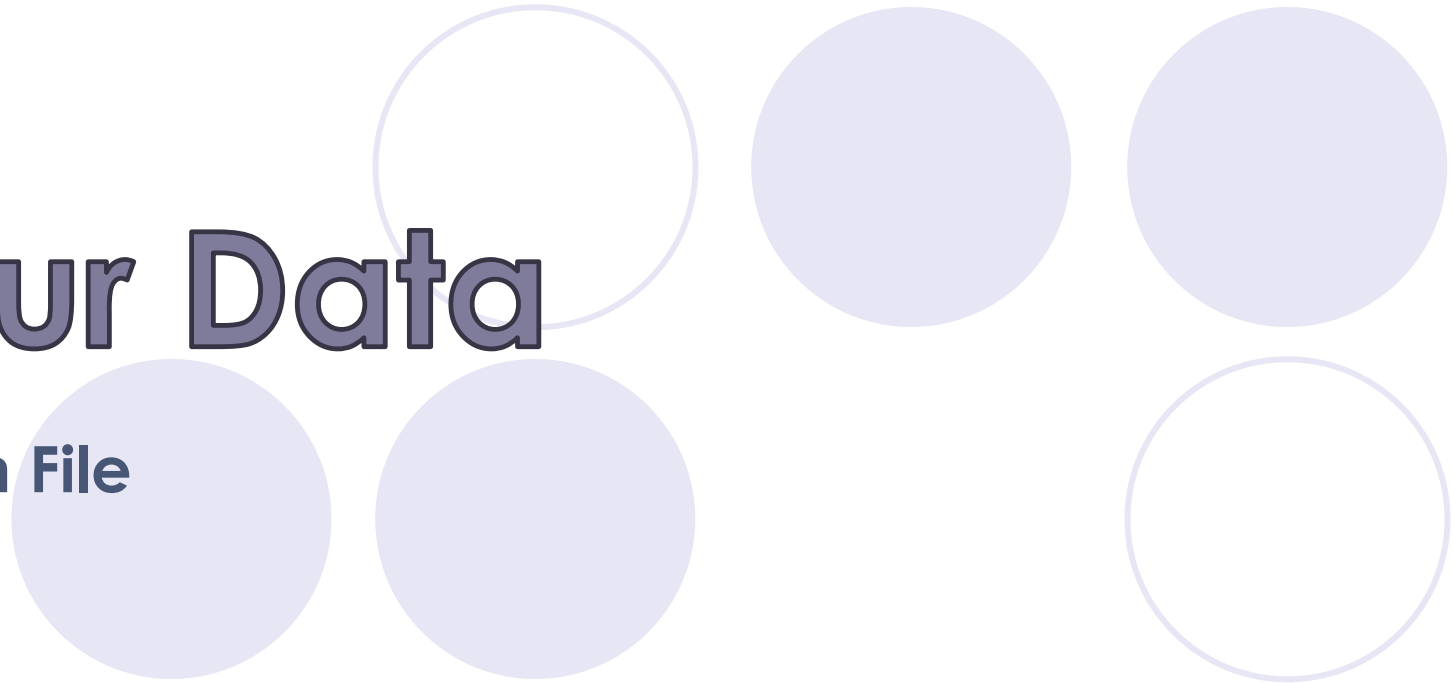


Struktur Data

Pemrograman File



Fungsi-Fungsi File PHP

- ◉ Memeriksa ketersediaan file
 - ◉ **file_exists()**
- ◉ Menulis File
 - ◉ **file_put_contents**
- ◉ Membaca file
 - ◉ **readfile()**
 - ◉ **file_get_contents()**
 - ◉ **fopen(), fread(), fclose**
 - ◉ **fgets()**

Fungsi file_exists()

- ⦿ Untuk memeriksa apakah file benar-benar ada atau tidak
- ⦿ Mengembalikan nilai **true** atau **false**

```
<?php
    if (file_exists('buah.txt')) {
        // file ada
    }
    else {
        echo "File buah.txt tidak ditemukan <br>";
    }
?>
```

Menulis file fungsi `file_put_contents`

◉ Sintaks

◉ **`file_put_contents ($filename,$data,$flags)`**

| Parameter | Keterangan |
|-------------------------|--|
| <code>\$filename</code> | target file yang akan ditulis |
| <code>\$data</code> | Data yang akan ditulis ke file, bisa string atau array |
| <code>\$flags</code> | Optional, values <ul style="list-style-type: none">• <code>FILE_USE_INCLUDE_PATH</code> : Pencarian untuk nama file dalam direktori• <code>FILE_APPEND</code> : menambahkan / menyisipkan konten ke akhir file• <code>LOCK_EX</code> : untuk mencegah orang lain menulis ke file pada saat yang sama |

Contoh Menulis File 1

```
<?php
    $file = "tes.txt";
    $data = "Ini teks yang akan dituliskan pada file";
    file_put_contents($file, $data);
?>
```

- ⦿ Catatan : Contoh diatas akan menimpa semua teks dalam file tes.txt dengan nilai pada variable \$data

Contoh Menulis File 2

```
<?php
    $file = "tes.txt";
    $data = "Ini teks yang akan disisipkan pada file";
    file_put_contents($file, $data, FILE_APPEND);
?>
```

- ⦿ Catatan : ini akan menambahkan / menyisipkan konten ke akhir file, tanpa harus menimpa data yang sudah ada

Contoh Menulis File 3

```
<?php
```

```
    $myarray[] = "baris 1";
```

```
    $myarray[] = "baris 2";
```

```
    $myarray[] = "baris 3";
```

```
    file_put_contents("tes.txt", $myarray);
```

```
?>
```

- ⦿ Catatan : ini akan menambahkan / menyisipkan konten ke akhir file, tanpa harus menimpa data yang sudah ada

Membaca file → readfile()

- ◉ Membaca seluruh data yang ada didalam file
- ◉ Memiliki keterbatasan fitur

```
<?php
```

```
echo readfile( 'buah.txt' );
```

```
?>
```


Membaca file → file_get_contents()

- ◉ Fungsi tersebut bisa untuk membaca berbagai file, baik secara lokal mau pun global (maksudnya url suatu situs tertentu), dan juga bisa membaca berbagai format file baik itu gambar, teks, json, xml, dan lain-lain.
- ◉ Mengembalikan data dalam bentuk string

```
<?php
```

```
    echo file_get_contents('buah.txt');
```

```
?>
```

Membaca file → file()

- ◉ Fungsi tersebut bisa untuk membaca berbagai file, baik secara lokal mau pun global (maksudnya url suatu situs tertentu), dan juga bisa membaca berbagai format file baik itu gambar, teks, json, xml, dan lain-lain
- ◉ Mengembalikan data dalam bentuk array

```
<?php
```

```
    $isiFileBuah = file('buah.txt');
```

```
    echo '<pre>';
```

```
    print_r($isiFileBuah);
```

```
    echo '</pre>';
```

```
?>
```

Membaca file → `fopen()`, `fread()`, `fclose()`

- ◉ Lebih rumit dan memerlukan beberapa step untuk menghasilkan sebuah data
- ◉ Lebih direkomendasikan karena memiliki banyak fungsi
- ◉ Skenario umum memiliki 3 langkah
 - ◉ Membuka file yang diinginkan dengan fungsi **`fopen()`**
 - ◉ Membaca file dengan fungsi **`fread()`**
 - ◉ Menutup file dengan fungsi **`fclose()`**

Contoh fopen(), fread(), fclose()

```
<?php
```

```
    $file = fopen('buah.txt', 'r');
```

```
    if (!$file) {
```

```
        die('File tidak ada');
```

```
    }
```

```
    # mulai baca file
```

```
    echo fread($file, filesize('buah.txt'));
```

```
    # tutup file agar ia dihapus dari memory  
    fclose($file);
```

```
?>
```

Penjelasan `fopen()`, `fread()`, `fclose()`

- ◉ **`fopen('buah.txt', 'r')`** menerima dua parameter
 - ◉ Nama file
 - ◉ Mode pembukaan file (**r**) artinya read
- ◉ **`fread($file, filesize('buah.txt'))`** menerima dua parameter
 - ◉ Hasil kembalian / return
 - ◉ Parameter kedua berisi berapa *byte* yang akan dibaca dari file tersebut dan sekaligus membaca semua isi data.
- ◉ **`fclose($file)`** menutup data yang sudah terbuka.

Mode fungsi fopen()

| Mode | Deskripsi |
|------|---|
| r | Membuka file dalam mode read only. |
| w | Membuka file dalam mode write only. Ia akan menghapus keseluruhan isi file dan menyimpannya dengan yang baru, atau jika file tersebut belum pernah ada, ia akan membuatnya terlebih dahulu. |
| a | Membuka file dalam mode write only. Isi dari file yang sebelumnya tetap dipertahankan. |
| x | Membuat file baru dalam mode write only. Mengembalikan nilai false jika file telah ada sebelumnya. |
| r+ | Membuka file dalam mode read dan write. |
| w+ | Membuka file dalam mode read dan write. Menghapus konten file sebelumnya atau membuat file baru jika belum ada. |
| a+ | Membuka file dalam mode read dan write. Sama seperti w+ akan tetapi konten file yang sebelumnya tidak dihapus. |
| x+ | Membuka file dalam mode read dan write. Mengembalikan nilai false jika file telah ada sebelumnya. |

Membaca file → fgets()

- ◉ Membaca file atau isi secara perbaris

<?php

```
$file = fopen('buah.txt', 'r');  
if (!$file){  
    die('File tidak ada');  
}  
  
# mulai baca file  
echo fgets($file). '<br>'; # baris 1  
echo fgets($file). '<br>'; # baris 2  
echo fgets($file). '<br>'; # baris 3  
echo fgets($file). '<br>'; # baris 4  
echo fgets($file). '<br>'; # baris 5  
  
# tutup file agar ia dihapus dari memory  
fclose($file);
```

?>

Mengetahui Akhir Baris Suatu File → feof

- ◉ Untuk mengetahui bahwa pointer pembacaan file sudah berada pada baris paling bawah
- ◉ Mengembalikan nilai **true** atau **false**

```
<?php
$file = fopen('buah.txt', 'r');
if (!$file){
    die('File tidak ada');
}
# lakukan perulangan selama feof bernilai false while (!feof($file)){
    echo fgets($file). '<br>';
}# tutup file agar ia dihapus dari memory
fclose($file);
?>
```