
SERVER SIDE PROGRAMMING (PART 6)

ERROR HANDLING DAN EXCEPTION

MATERI KE-12





ERROR



ERROR DI PHP

- Dalam pembuatan aplikasi, *error handling* sangat penting untuk dilakukan karena dapat mempengaruhi UX (User Experience) serta rawan masalah keamanan
- Secara default, PHP akan melaporkan error, warning (peringatan), dan notice (pemberitahuan) untuk permasalahan yang biasa terjadi pada kode dan saat eksekusi.
- Dengan mengetahui permasalahan tersebut, maka akan lebih mudah untuk melakukan debug dan perbaikan kode.

Nilai	Constant	Penjelasan
1	E_ERROR (integer)	Error pada run-time yang fatal. Mengindikasikan sebuah error yang tidak bisa diperbaiki, misalnya alokasi memori. Eksekusi script dihentikan
2	E_WARNING (integer)	Peringatan pada run-time(tidak fatal). Eksekusi script tidak dihentikan
4	E_PARSE (integer)	Error parsing pada saat compile.
8	E_NOTICE (integer)	Pemberitahuan saat run-time. Dapat mengindikasikan error tetapi juga bisa terjadi saat eksekusi secara normal.
16	E_CORE_ERROR (integer)	Error fatal yang terjadi saat startup PHP . Seperti E_ERROR tetapi permasalahan terjadi pada core PHP
32	E_CORE_WARNING (integer)	Peringatan (tidak fatal) terjadi saat startup PHP . Seperti E_WARNING tetapi permasalahan terjadi pada core PHP
64	E_COMPILE_ERROR (integer)	Error fatal yang terjadi saat compile. Seperti E_ERROR tetapi terjadi karena Zend Scripting Engine
128	E_COMPILE_WARNING (integer)	Peringatan yang terjadi saat compile. Seperti E_WARNING tetapi terjadi karena Zend Scripting Engine
256	E_USER_ERROR (integer)	Pesan error yang dibuat ole user. Seperti E_ERROR tetapi dihasilkan dalam kode PHP menggunakan trigger_error() .
512	E_USER_WARNING (integer)	Peringatan yang dibuat ole user. Seperti E_WARNING tetapi dihasilkan dalam kode PHP menggunakan trigger_error() .
1024	E_USER_NOTICE (integer)	Pemberitahuan yang dibuat oleh user. Seperti E_NOTICE tetapi dihasilkan dalam kode PHP menggunakan trigger_error() .
2048	E_STRICT (integer)	Memungkinkan PHP untuk menyarankan perubahan pada kode untuk memastikan operasi dan kompatibilitas kode
4096	E_RECOVERABLE_ERROR (integer)	Error fatal yang dapat dilakukan <i>recovery</i> . Mengindikasikan error fatal tetapi tidak mengakibatkan <i>Engine</i> dalam kondisi tidak stabil. Jika error ini tidak ditangkap oleh <i>error handling</i> yang dibuat user, maka aplikasi akan membatalkan kode seperti saat E_ERROR terjadi.
8192	E_DEPRECATED (integer)	Peringatan pada saat run-time. Opsi ini bisa diaktifkan untuk memberikan peringatan kode yang tidak dapat berjalan pada versi baru
16384	E_USER_DEPRECATED (integer)	Pesan peringatan dari user. Seperti E_DEPRECATED tetapi dihasilkan oleh kode menggunakan fungsi PHP trigger_error() .
32767	E_ALL (integer)	Semua error dan peringatan yang didukung, kecuali level E_STRICT yang ada pada versi sebelum PHP 5.4.0

Sumber : <https://www.php.net/manual/en/errorfunc.constants.php>

ERROR HANDLING

- Beberapa cara error handling pada PHP:

- Menggunakan sintaks `die()`
- Menggunakan *custom error* dan *trigger*

- Catatan:

Error handling pada PHP 7 berbeda dengan PHP 5. Materi ini hanya mencakup *error handling* pada PHP 5. Untuk mempelajari *error* pada PHP 7 silakan buka tautan berikut

<https://www.php.net/manual/en/language.errors.php7.php>

MENGGUNAKAN SINTAKS `die()`

```
<?php  
$file=fopen("data.txt","r");  
?>
```

- Pada contoh kode di atas apabila terjadi kegagalan dalam membuka file, misalnya file tidak tersedia di server maka akan muncul error seperti berikut:

```
Warning: fopen(data.txt) [function.fopen]: failed to open stream:  
No such file or directory in C:\xampp\htdocs\test.php on line 2
```

- Error yang dihasilkan oleh PHP secara default menunjukkan permasalahan dan bagian kode yang menghasilkan error. Pada contoh di atas errornya adalah “failed to open stream: No such file or directory” yang menunjukkan bahwa file data.txt tidak dapat dibuka karena tidak ditemukan. Permasalahan terjadi pada kode di file test.php pada baris kode ke-2

MENGGUNAKAN SINTAKS `die()` (2)

- Untuk mencegah munculnya error tersebut dapat menggunakan `die()`

(lihat contoh pada slide Materi 10)

```
<?php
if(!file_exists("data.txt")) {
    die("File tidak ditemukan");
} else {
    $file=fopen("data.txt","r");
}
?>
```

- Pesan error yang ditampilkan akan menjadi seperti berikut:

```
File tidak ditemukan
```

MENGGUNAKAN *CUSTOM ERROR*

- Untuk membuat *custom error* cukup mudah dilakukan dengan membuat fungsi yang dipanggil saat error terjadi
- Sintaks yang digunakan adalah

```
error_function(level_error,pesan_error,file_error,baris_error,context_error)
```

Catatan:

- parameter yang berwarna merah merupakan parameter yang wajib, parameter berwarna hijau adalah parameter opsional (bisa dikosongkan)
- Level error dapat dilihat pada tabel di slide halaman 4

MENGGUNAKAN *CUSTOM ERROR* (2)

1. Membuat fungsi untuk handle error:

```
function customError($errno, $errstr) {  
    echo "<b>Error:</b> [$errno] $errstr";  
}
```

2. Memasang *error handler*:

```
set_error_handler("customError");
```

-
- Contoh kode lengkapnya adalah:

```
<?php
//fungsi error
function customError($errno, $errstr) {
    echo "<b>Pesan error:</b> [$errno] $errstr";
}
set_error_handler("customError"); //memasang error handler
echo($test);                      //baris kode yang menghasilkan error
?>
```

- Pada contoh di atas, variable `$test` belum pernah dideklarasikan atau diisi sebuah nilai, maka error yang muncul adalah:

```
Pesan error: [8] Undefined variable: test
```

Yang menunjukkan bahwa variabel `$test` tidak terdefinisi dengan kode error 8 (lihat table halaman 4)

MENGGUNAKAN *CUSTOM ERROR* (3)

- Dalam kode PHP dapat kita definisikan sebuah *trigger* apabila terdapat sebuah error
- Trigger dapat dibuat dengan menggunakan fungsi `trigger_error()`.

```
<?php
//fungsi untuk error handling
function customError($errno, $errstr) {
    echo "<b>Pesan Error:</b> [$errno] $errstr<br>";
    echo "Script diakhiri";
    die();
}

//memasang error handler
set_error_handler("customError",E_USER_WARNING);

//kode yang mengakibatkan trigger error terjadi
$test = 2;
if ($test >= 1) {
    trigger_error("Angka harus bernilai 1 atau di bawah 1",E_USER_WARNING);
}
?>
```

Kode di atas akan menghasilkan error sebagai berikut:

```
Pesan Error: [512] Angka harus bernilai 1 atau di bawah 1
Script diakhiri
```



EXCEPTION



EXCEPTIONS DI PHP

- Exception digunakan untuk merubah jalannya eksekusi kode saat terjadi kesalahan/error
- Saat exception terjadi biasanya ada beberapa kemungkinan yang terjadi:
 - State dari kode disimpan
 - Eksekusi kode akan dialihkan ke fungsi untuk meng-*handle custom exception* yang dibuat oleh user
 - Tergantung dari situasi, *handler* mungkin akan meneruskan eksekusi dari state yang telah tersimpan atau mengakhiri eksekusi, atau melanjutkan script dari lokasi yang berbedapada kode

MENGGUNAKAN `throw` *EXCEPTION*

```
<?php
//membuat fungsi yang berisi exception menggunakan throw
function cekAngka($angka) {
    if($ angka > 1) {
        throw new Exception("Angka harus bernilai 1 atau di bawah 1");
    }
    return true;
}
//kode yang menghasilkan exception
cekAngka(2);
?>
```

Pada kode di atas angka bernilai 2, maka saat memanggil fungsi `cekAngka()` akan masuk ke dalam Exception dan menghasilkan error seperti berikut:

```
Fatal error: Uncaught exception 'Exception'
with message 'Angka harus bernilai 1 atau di bawah 1' in C:\xampp\htdocs\test.php:6
Stack trace: #0 C:\xampp\htdocs\test.php(12):
cekAngka(28) #1 {main} thrown in C:\xampp\htdocs\test.php on line 6
```

MENGGUNAKAN `try`, `throw`, `catch`

- Untuk menghindari pesan error seperti contoh sebelumnya, dapat digunakan `try`, `throw`, `catch`.
 - **`try` :**
fungsi yang menggunakan exception seharusnya berada dalam blok `try`. Jika tidak masuk trigger exception, maka kode akan berjalan normal. Sedangkan apabila masuk trigger exception, maka exception tersebut akan dilempar (`throw`)
 - **`throw`:**
Setiap harus `throw` mempunyai minimal satu `catch`
 - **`catch`:**
Blok "`catch`" menerima exception dan membuat sebuah obyek yang berisi informasi tentang exception tersebut.


```
<?php
// membuat fungsi yang berisi exception
function cekAngka($angka) {
    if($ angka > 1) {
        throw new Exception("Angka harus bernilai 1 atau di bawah 1");
    }
    return true;
}
//trigger exception dalam blok "try"
try {
    cekAngka(2);
    //Jika exception terjadi maka tulisan ini tidak akan muncul
    echo 'Jika melihat tulisan ini, maka angka bernilai 1 atau kurang dari 1';
}
//catch exception
catch(Exception $e) {
    echo Pesan: ' .$e->getMessage();
}
?>
```

Pada kode di atas angka bernilai 2, maka saat memanggil fungsi `cekAngka()` akan masuk ke blok `catch` karena nilainya salah. Pada blok `catch` terdapat kode untuk menampilkan pesan berikut:

```
Pesan: Angka harus bernilai 1 atau di bawah 1
```

REFERENSI LANJUT

- **Error:**

- <https://www.php.net/manual/en/language.errors.php>
- https://www.w3schools.com/php/php_error.asp

- **Exception:**

- <https://www.php.net/manual/en/language.exceptions.php>
- https://www.w3schools.com/php/php_exception.asp