

MODUL 4

PEMROGRAMAN WEB

FUNCTION

A. TUJUAN PERKULIAHAN

- Mahasiswa mengenali konsep fungsi pada bahasa PHP
- Mahasiswa mampu mengimplementasikan built in function pada aplikasi web berbasis PHP
- Mahasiswa mampu membuat user defined function pada aplikasi web berbasis PHP

B. ALOKASI WAKTU 3 x 50 menit

C. DASAR TEORI

1. FUNCTION

Fungsi (Function) adalah suatu potongan kode terpisah di dalam blok sendiri yang berfungsi sebagai sub-program (modul program). Function dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Penulisan function memiliki standar yang baku dan tidak boleh untuk dilanggar. Setiap function harus diberi nama, setiap nama function diawali dengan menggunakan huruf atau underscore (_) dan tidak boleh menggunakan spasi.

```
<?php
function namaFungsi($arg1, $arg2){
    //...
}

Atau

Function __namaFungsi($arg1, $arg2){
    //...
}
?>
```

Function pada dasarnya dibangun untuk mempermudah kita dalam membuat program karena dapat digunakan ulang tanpa harus menulis ulang instruksi di dalamnya

(*reusability*). Function yang ada di PHP dibedakan menjadi dua jenis, yaitu built-in function dan user-defined function.

2. BUILT-IN FUNCTION

Built-in function merupakan fungsi bawaan yang dimiliki oleh PHP dan dapat langsung digunakan dengan memanggil nama fungsi ke dalam program kita. Referensi terkait built-in function PHP dapat dibaca pada laman <https://www.php.net/manual/en/funcref.php>. Beberapa diantara built-in function yang kerap digunakan adalah sebagai berikut

a. `date()`

Fungsi `date()` berguna untuk menampilkan waktu (hari, tanggal, bulan, tahun, jam, menit dan detik). Fungsi `date` memiliki syntax `date(format, timestamp)` dimana format merupakan parameter yang wajib ditulis untuk menentukan format dari string waktu yang akan ditampilkan, apakah seluruh tanggal dan jam atau hanya sebagian saja. Sementara timestamp adalah parameter opsional yang berguna untuk menetapkan waktu sesuai dengan hitungan UNIX TIME. Nilai default dari `date()` adalah local time dari sistem. Untuk format lebih detail dapat diakses pada link berikut https://www.w3schools.com/php/func_date_date.asp

```
<?php
echo date("l") . "<br>";
echo date("l jS \of F Y h:i:s A") . "<br>";
echo date("d-M-Y h:i:s") . "<br>";
?>
```

Contoh `date()` di atas akan menghasilkan tampilan berikut. 24 Agustus 2021 merupakan tanggal dimana modul ini dibuat.

```
Tuesday
Tuesday 24th of August 2021 02:08:26 PM
24-Aug-2021 02:08:26
```

b. time()

Fungsi `time()` memiliki peran dalam perhitungan waktu yang didasarkan pada UNIX TIME. UNIX TIME merupakan format waktu yang dihitung berdasarkan jumlah detik yang sudah berjalan sejak tanggal kesepakatan UNIX TIME, yakni 1 Januari 1970. Fungsi `time()` biasa dikombinasikan dengan fungsi `date()` agar dapat dibaca sesuai format waktu pada umumnya.

```
<?php
$t=time();
echo($t . "<br>");
echo(date("Y-m-d", $t));
?>
```

Contoh `time()` di atas akan menghasilkan tampilan berikut. 24 Agustus 2021 merupakan tanggal dimana modul ini dibuat.

```
1629814465
2021-08-24
```

c. explode()

Fungsi `explode()` digunakan untuk memotong kalimat per kata dalam bentuk string dan dikemas dalam bentuk array. Delimiter atau pemisah dari parameter string adalah tanda spasi.

```
<?php
$str = "Hello world. It's a beautiful day.";
print_r (explode(" ", $str));
?>
```

Contoh fungsi `explode()` di atas akan menghasilkan tampilan berikut

```
Array ([0] => Hello [1] => world. [2] => It's [3] => a
[4] => beautiful [5] => day.)
```

d. `strtoupper()` / `strtolower()`

Fungsi `strtoupper()` digunakan untuk membuat seluruh argument string yang dimasukkan menjadi huruf kapital seluruhnya. Sedangkan `strtolower()` berfungsi sebaliknya, yakni membuat argument string di dalam fungsi tersebut menjadi huruf kecil semua. Berikut contoh penggunaannya

```
<?php
echo strtoupper("Hello WORLD!");
echo "<br>";
echo strtolower("Hello WORLD!");
?>
```

Contoh fungsi `strtoupper()` dan `strtolower()` di atas akan menghasilkan tampilan berikut

```
HELLO WORLD!
hello world!
```

e. `round()` / `floor()` / `ceil()`

Fungsi `round()`, `floor()` dan `ceil()` sama-sama digunakan untuk mendapatkan hasil pembulatan dari suatu nilai desimal. Perbedaan di antara ketiga fungsi tersebut adalah :

- Fungsi `round()` digunakan untuk melakukan pembulatan ke nilai integer terdekat,
- Fungsi `floor()` digunakan untuk mencari nilai pembulatan ke bawah.
- Fungsi `ceil()` digunakan untuk mencari nilai pembulatan ke atas.

Berikut contoh penggunaannya

```
<?php
echo(round(0.60) . "<br>");
echo(round(4.50) . "<br>");
echo(floor(7.59) . "<br>");
echo(ceil(-4.40));
?>
```

Contoh fungsi *round()*, *floor()* dan *ceil()* di atas akan menghasilkan tampilan berikut

```
1
5
7
-4
```

3. USER DEFINED FUNCTION

User defined function adalah fungsi yang dibentuk sendiri oleh programmer, dimana tujuan dari pembuatan user defined function ini disesuaikan dengan kebutuhan dan tujuan dari programmer tersebut. User-defined function biasanya dibuat apabila terdapat suatu blok kode yang dibuat khusus untuk menangani masalah spesifik dan digunakan pada lebih dari satu bagian di dalam program sehingga perlu pemanggilan berulang.

Kita mendefinisikan function dengan *function(\$parameter)*. Sebuah function biasanya mengembalikan nilai , caranya dengan menggunakan *return*. Berikut ini adalah contohnya:

```
<?php
function hello($name){
    return "Selamat datang $name";
}

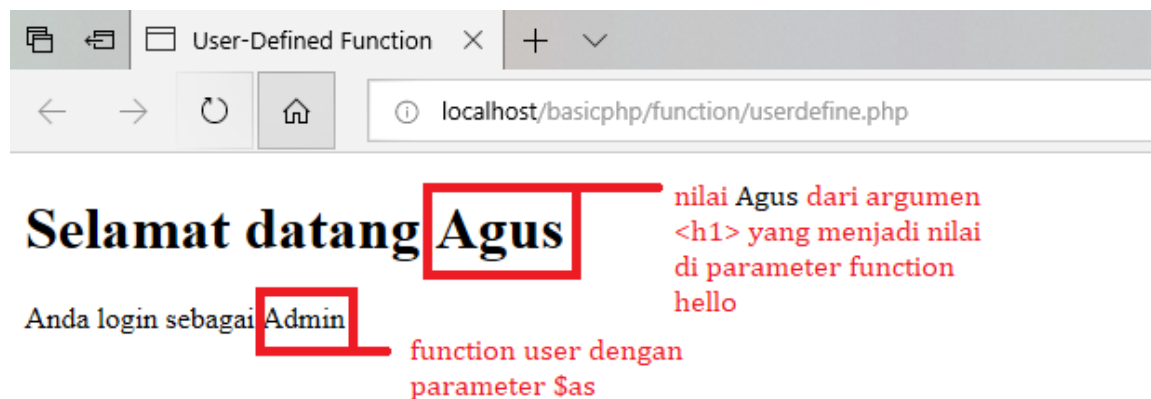
function user($as){
    return "Anda login sebagai $as";
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>User-Defined Function</title>
</head>
<body>
    <h1><?php echo hello("Agus"); ?> </h1>
    <p><?php echo user("Admin");?> </p>
</body>
</html>
```

Pada baris kode diatas, kita memiliki dua fungsi, pertama *function hello()* dengan parameter name dan function user dengan parameter *\$as*. Dengan masing-masing fungsi memiliki nilai *return*. User-defined function selalu didefinisikan terlebih dahulu sebelum dipanggil.

Untuk memanggil fungsi dapat kita letakkan langsung pada tag html **<h1>** dan **<p>**. Jadi kita panggil fungsinya dengan tag php : `<?php echo hello("argumen"); ?>` . Jadi argumen yang ada akan menjadi nilai di parameter fungsi *hello()*.

Hasilnya pada browser adalah seperti berikut :



Fungsi tersebut dapat dipanggil berulang-ulang pada halaman yang berbeda, terutama jika Anda memanfaatkan built-in function *include()* untuk memanggil file PHP yang ditulis khusus berisi ragam user-defined function untuk keperluan reusability.

D. PRAKTIKUM : Function PHP

Kegiatan 1 - basic

1. Akan dibuat sebuah tampilan untuk membuat formasi bintang sebagai berikut:

```
*  
* *  
* * *  
* * * *
```

dan juga

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Untuk memudahkan dalam pemrograman, perlu dibuat sebuah function untuk membuat formasi bintang tersebut, yaitu `buatBintang(n)` dengan `n` adalah banyaknya baris bintang yang diinginkan. Simpan script di atas sebagai **p040101.php**

HINT

```
<?php  
function buatBintang($n) {  
    //.....  
}  
buatBintang(4);  
buatBintang(5);  
?>
```

2. Buatlah script PHP yang baru dengan nama file **p040102.php**!
Di dalam script ini, buatlah function buatBintangReverse(n) untuk menampilkan formasi bintang sbb
3.

```
****
***
**
*
```

contoh tampilan untuk n = 4

Kegiatan 2 - advance

1. Function berikut ini digunakan untuk menghitung gaji karyawan berdasarkan golongan dan masa kerjanya dengan ketentuan sebagai berikut. Simpan script di atas dengan nama file **p040201.php**

Golongan	Masa Kerja	Gaji
A	< 10 thn	Rp 5.000.000
A	>= 10 thn	Rp 7.000.000
B	< 10 thn	Rp 6.000.000
B	>= 10 thn	Rp 8.000.000

HINT : Lengkapi function berikut ini lalu panggil fungsi tersebut sesuai dengan ketentuan di atas! Anda dapat memodifikasi sendiri percabangannya dengan menggunakan switch-case atau yang lain.

```
function hitungGaji($gol, $masaKerja){
    if ($gol == ...){
        if ($masaKerja < ...) {
            $gaji = ...
        } else {
            $gaji = ...
        }
    }
    ...
    ...
    return ...
}
```


2. Smitty Werben Man Jensen adalah seorang programmer yang telah membuat aplikasi peminjaman dan pengembalian buku di sebuah perpustakaan. Namun, aplikasi yang telah dibuatnya tersebut belum sempurna karena belum adanya fitur untuk menghitung denda keterlambatan pengembalian buku. Simpan script dengan nama **p040202.php!**
Bantulah Smitty Werben Man Jensen dengan membuatkan fungsi berikut

```
function hitungDenda(tglHarusKembali, tglKembali)
```

Ketentuan:

- Parameter **tglHarusKembali** diisi dengan sebuah string 'YYYY-MM-DD' yang menunjukkan tanggal berapa seharusnya buku itu dikembalikan.
- Parameter **tglKembali** yang juga bertipe string 'YYYY-MM-DD' menunjukkan tanggal pengembalian buku.
- Besar denda keterlambatan perhari adalah Rp 5.000,-
- Function melakukan **return** nilai denda total yang dihitung dengan rumus:
jumlah_hari_terlambat * besar_denda

Contoh:

```
echo "Besarnya denda adalah: Rp ".hitungDenda("2021-01-03",  
"2021-01-05");
```

Akan menghasilkan output

```
Besarnya denda adalah: Rp 10.000,-
```

E. REFERENSI

- <https://www.php.net/manual/en/funcref.php>
- <https://www.w3schools.com/php/>
- <https://www.ketutrare.com/2019/01/Pemrograman-PHP-Built-In-Function-dan-User-Defined-Function-di-PHP.html>