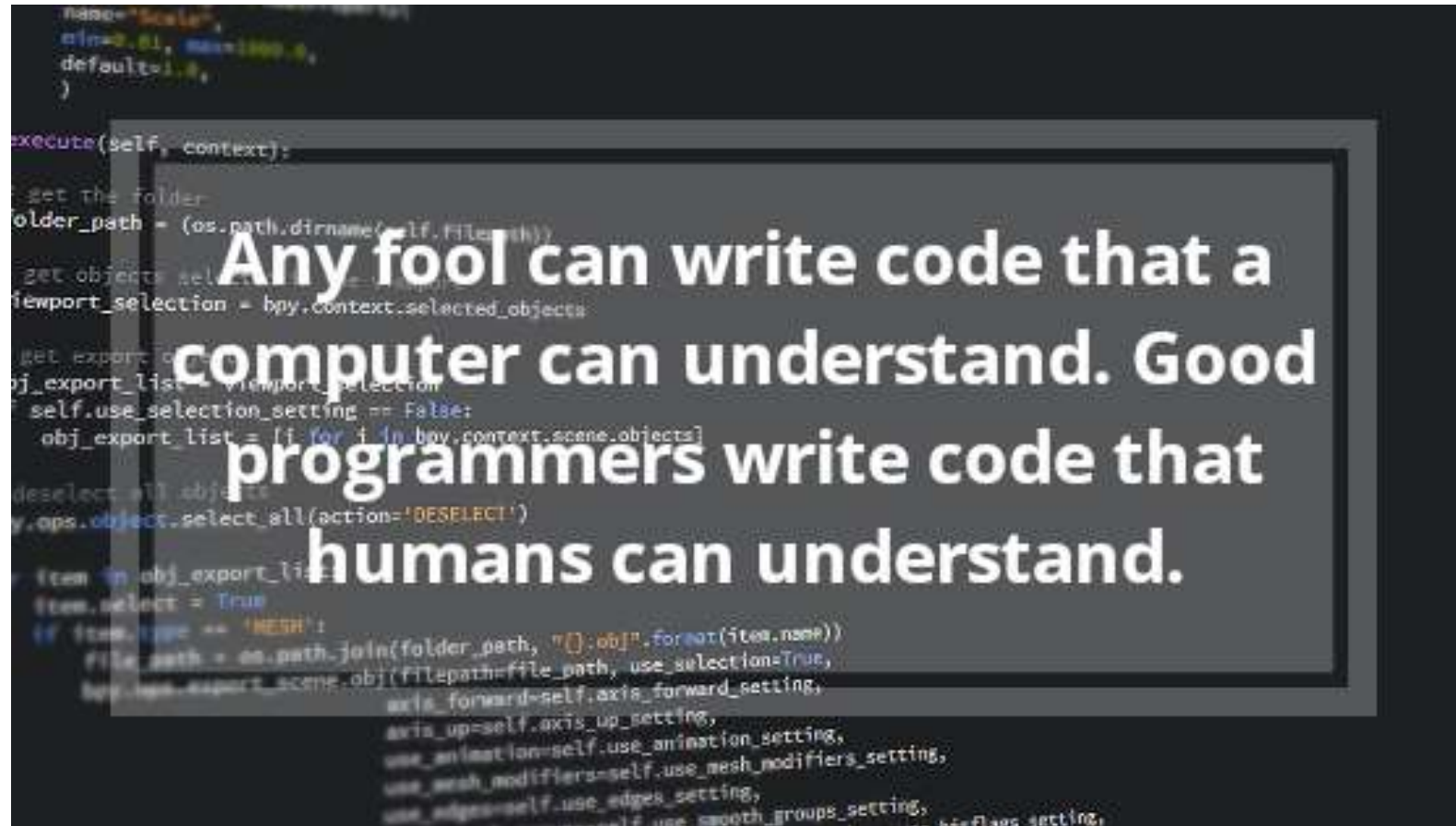
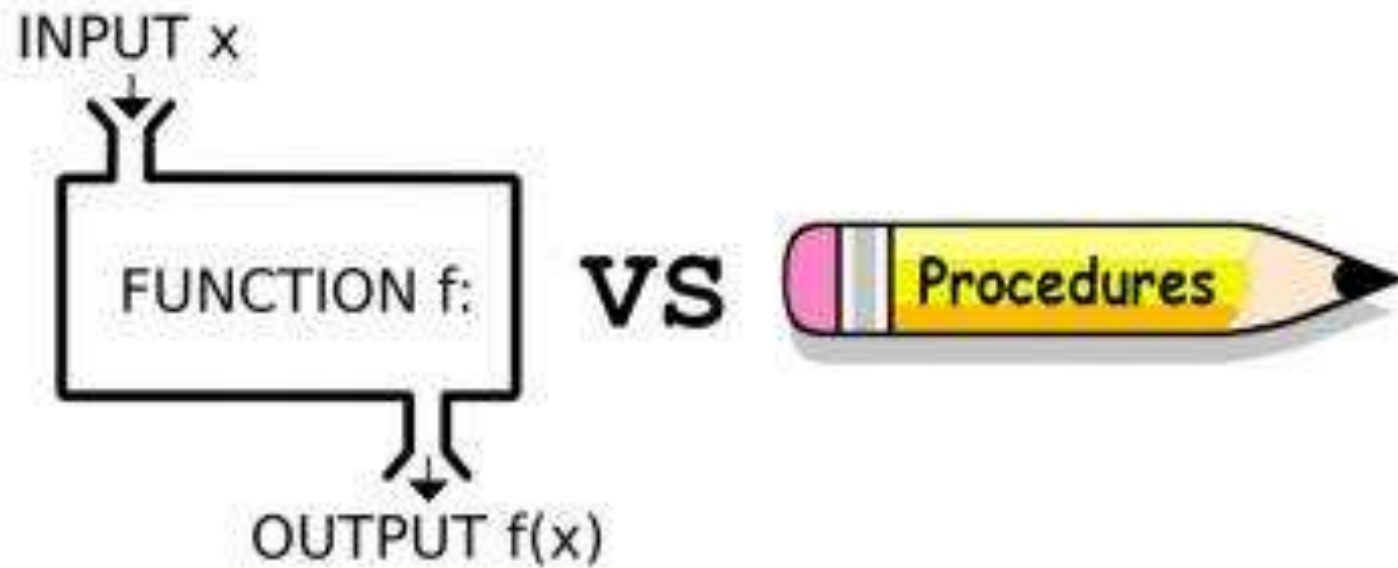


# Motivasi



# Aspersepsi

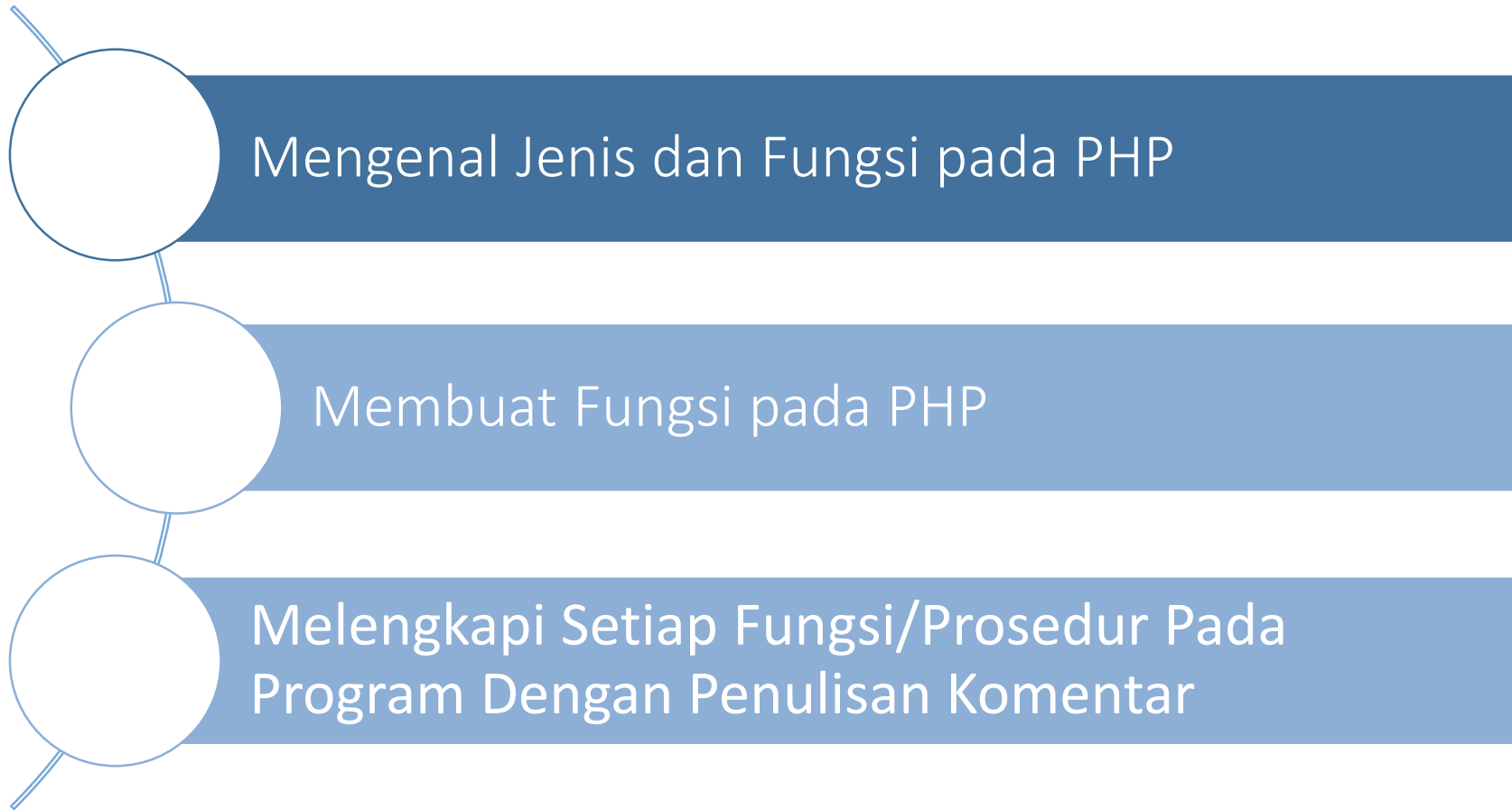


# Fungsi dan Prosedur

# Ringkasan Mata Pelatihan

- Unit Kompetensi Acuan: Mengimplementasikan Pemrograman Terstruktur
- Kode Unit Kompetensi Acuan: J.620100.017.02
- Deskripsi singkat: Mata pelatihan ini menentukan kompetensi, pengetahuan dan sikap kerja yang diperlukan dalam membuat program terstruktur atau prosedural.
- Tujuan Pembelajaran: Peserta dapat membuat program terstruktur atau procedural sesuai dengan paradigma yang berlaku.

# Agenda



# Mengenal Jenis dan Fungsi pada PHP

# Memahami Fungsi Pada PHP

- Fungsi adalah sekumpulan intruksi yang dibungkus dalam sebuah blok. Fungsi dapat digunakan ulang tanpa harus menulis ulang instruksi di dalamnya.

# Jenis Fungsi Pada PHP

- Fungsi pada PHP dibagi menjadi dua yaitu *built-in function* yang merupakan fungsi bawaan PHP dan *user-defined function*, dimana kita membuat fungsi sendiri.
- Banyak sekali fungsi yang disediakan php, seperti `substr()`, dll yang dapat kita gunakan langsung, list lengkapnya dapat dilihat pada halaman: [PHP: Function and Method Listing – Manual](#)



# Perbedaan Fungsi dan Prosedur pada PHP

- Prosedur: tidak mengembalikan nilai
- Fungsi: mengembalikan nilai
- Pada PHP, prosedur dan fungsi menggunakan sintaks function

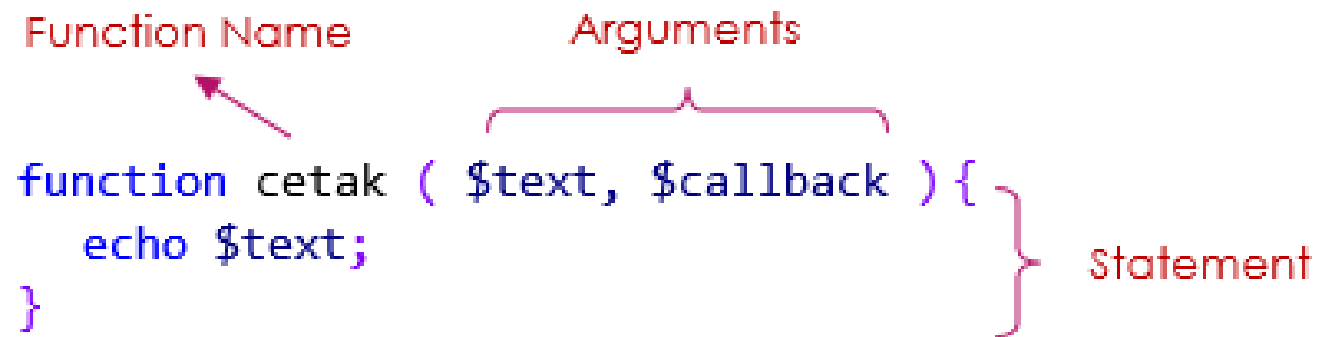
# Membuat Fungsi pada PHP

# Struktur Penulisan Fungsi

Function Name                      Arguments

```
function cetak ( $text, $callback ) {  
    echo $text;  
}
```

Statement



# Contoh

```
function perkenalan(){  
    echo "Assalamulaikmu, ";  
    echo "Perkenalkan, nama saya Ardianta<br/>";  
    echo "Senang berkenalan dengan anda<br/>";  
}
```

Fungsi yang sudah dibuat tidak akan menghasilkan apapun kalau tidak dipanggil. Kita dapat memanggil fungsi dengan menuliskan namanya.

# Code Lengkapnya

```
<?php
// mmbuat fungsi
function perkenalan(){
    echo "Assalamualaikmu, ";
    echo "Perkenalkan, nama saya Ardianta<br/>";
    echo "Senang berkenalan dengan anda<br/>";
}

// memanggil fungsi yang sudah dibuat
perkenalan();

echo "<hr>";

// memanggilnya lagi
perkenalan();
?>
```

Assalamualaikmu,  
Perkenalkan, nama saya Ardianta  
Senang berkenalan dengan anda

---

Assalamualaikmu,  
Perkenalkan, nama saya Ardianta  
Senang berkenalan dengan anda

# Fungsi dengan Parameter atau Arguments

- Supaya intruksi yang di dalam fungsi lebih dinamis, kita dapat menggunakan parameter untuk memasukkan sebuah nilai ke dalam fungsi. Nilai tersebut akan diolah di dalam fungsi.

```
<?php
// mmbuat fungsi
function perkenalan($nama, $salam){
    echo $salam.", ";
    echo "Perkenalkan, nama saya ".$nama."<br/>";
    echo "Senang berkenalan dengan anda<br/>";
}

// memanggil fungsi yang sudah dibuat
perkenalan("Muhardian", "Hi");

echo "<hr>";

$saya = "Indry";
$ucapanSalam = "Selamat pagi";
// memanggilnya lagi
perkenalan($saya, $ucapanSalam);
?>
```

Hi,  
Perkenalkan, nama saya Muhardian  
Senang berkenalan dengan anda

---

Selamat pagi,  
Perkenalkan, nama saya Indry  
Senang berkenalan dengan anda

# Paramter dengan Nilai Default

- Nilai *default* dapat kita berikan di parameter. Nilai *default* berfungsi untuk mengisi nilai sebuah parameter, kalau parameter tersebut tidak diisi nilainya.

```
<?php
// mmbuat fungsi
function perkenalan($nama, $salam="Assalamualaikum"){
    echo $salam.", ";
    echo "Perkenalkan, nama saya ".$nama."<br/>";
    echo "Senang berkenalan dengan anda<br/>";
}

// memanggil fungsi yang sudah dibuat
perkenalan("Muhardian", "Hi");

echo "<hr>";

$saya = "Indry";
$ucapanSalam = "Selamat pagi";
// memanggilnya lagi tanpa mengisi parameter salam
perkenalan($saya);
?>
```

Hi,  
Perkenalkan, nama saya Muhardian  
Senang berkenalan dengan anda

---

Assalamualaikum,  
Perkenalkan, nama saya Indry  
Senang berkenalan dengan anda

# Fungsi yang Mengembalikan Nilai

- Hasil pengolahan nilai dari fungsi mungkin saja kita butuhkan untuk pemrosesan berikutnya. Oleh karena itu, kita harus membuat fungsi yang dapat mengembalikan nilai.

```
<?php
// membuat fungsi
function hitungUmur($thn_lahir, $thn_sekarang){
    $umur = $thn_sekarang - $thn_lahir;
    return $umur;
}

echo "Umur saya adalah ". hitungUmur(1994, 2015) ." tahun";
?>
```

Umur saya adalah 21 tahun



# Memanggil Fungsi di dalam Fungsi

- Fungsi yang sudah kita buat, dapat juga dipanggil di dalam fungsi lain.

```
<?php
// membuat fungsi
function hitungUmur($thn_lahir, $thn_sekarang){
    $umur = $thn_sekarang - $thn_lahir;
    return $umur;
}

function perkenalan($nama, $salam="Assalamualaikum"){
    echo $salam.", ";
    echo "Perkenalkan, nama saya ".$nama."<br/>";
    // memanggil fungsi lain
    echo "Saya berusia ". hitungUmur(1994, 2015) ." tahun<br/>";
    echo "Senang berkenalan dengan anda<br/>";
}

// memanggil fungsi perkenalan
perkenalan("Ardianta");

?>
```

Assalamualaikum,  
Perkenalkan, nama saya Ardianta  
Saya berusia 21 tahun  
Senang berkenalan dengan anda

# Fungsi rekursif

- Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Fungsi ini biasanya digunakan untuk menyelesaikan masalah seperti faktorial, bilangan fibonacci, pemrograman dinamis, dll.

```
<?php

function faktorial($angka) {
    if ($angka < 2) {
        return 1;
    } else {
        // memanggil dirinya sendiri
        return ($angka * faktorial($angka-1));
    }
}

// memanggil fungsi
echo "faktorial 5 adalah " . faktorial(5);

?>
```

faktorial 5 adalah 120

# Fungsi vs Prosedur

```
<?php

function
tambahSatu($value){
    $value = $value + 1;
}

$a = 5;
tambahSatu($a);
echo $a;

?>
```

Output:

5

```
<?php

function tambahSatu(&$value){
    $value = $value + 1;
}

$a = 5;
tambahSatu($a);
echo $a;

?>
```

Output:

6

Melengkapi Setiap  
Fungsi/Prosedur  
Pada Program  
Dengan Penulisan  
Komentar

# Komentar pada PHP

- Saat kita membuat baris-baris program didalam PHP kita dapat menggunakan fungsi komentar untuk menjelaskan maksud dari setiap baris atau function yang kita buat. Sehingga terdapat dokumentasi yang jelas dari setiap syntax PHP yang kita buat.
- Komentar di PHP diawali dengan tanda `//`. PHP tidak akan mengeksekusi setiap karakter yang terdapat dibelakang tanda `//`. Karena PHP akan membaca hal tersebut merupakan komentar. Contoh seperti dibawah ini:

# Contoh

```
<?php  
echo "Selamat datang di Kelasprogrammer.com "; // Menampilkan output ke layar monitor  
?>
```



# Pemberian Komentar pada fungsi

- Biasanya fungsi/prosedur/program dilengkapi dengan penulisan komentar di awal mengenai deskripsi fungsi/ prosedur/program tersebut; initial state dan final state; author (pembuat); versi dan/atau tanggal.

```
1  <?php
2  /*
3   * @desc class ini digunakan untuk menyimpan fungsi untuk interaksi user
4   * @author Jake Rocheleau (jakerocheleau@gmail.com)
5   * @required config.php
6   * @version 1.2
7   * @date : Maret 2019
8  */
9
10 class userInteraction{
11
12
13     /*
14     * @desc method digunakan untuk memproses login
15     * @param $userID berisi id user dalam bentuk string
16     * @param $pass berisi pass dari user dalam bentuk string minimal 8 character
17     * @return $loginID yang berisi id login dalam bentuk string
18     */
19     public function login($userID, $pass){ ... }
20
21 }
22
23 ?>
```

- Beberapa dokumentasi Parameter lain yang sering digunakan

@see

@since

@deprecated

@serial

@exception

@link



# Kesimpulan

# Kesimpulan

- Fungsi dan Prosedure dalam PHP hampir sama yang membedakan hanyalah jika fungsi mengembalikan nilai, sebaliknya prosedur tidak mengembalikan nilai

# Referensi / Bacaan Lebih Lanjut

# Referensi / Bacaan Lebih Lanjut

- <https://www.petanikode.com/php-prosedur-fungsi/>
- <https://www.w3schools.com/php/>



Kantor:

Balai Pelatihan dan Pengembangan  
Teknologi Informasi dan Komunikasi  
Kementerian Kominfo

Website: <https://bpptik.kominfo.go.id>

Email: [bpptik@kominfo.go.id](mailto:bpptik@kominfo.go.id)

Twitter: @bpptik

Facebook: @bpptik

Instagram: @bpptik

Google Plus: +bpptikkemkominfo

# Terima Kasih

## BPPTIK