VARIABLE, DATA TYPES, PHP OPERATORS, PERCABANGAN, ARRAY & STRING, LOOPING & FUNCTION

Ardian Yusuf Wicaksono
Arrasyid Fadel Fatonsyah

Materi

1. Variable
2. Data Types
3. PHP Operators
4. Percabangan
5. Array
6. String
7. Looping
8. Function

Variable

Variable

- Variable merupakan sebuah wadah untuk menampung sebuah data
- Untuk membuat variable, kita menggunakan dollar ($) diikuti dengan nama variable
- Penamaan variable tidak boleh mengandung spasi

Contoh Variable

```php
$nama = "Bambang";
$no_punggung = 10;
$apakahTimnas = true;
```

Data Types

Tipe Data Variabel

- Di PHP variabel bisa menampung berbagai jenis tipe data, dan bisa berubah-ubah tipe datanya

```php
$integer = 10;
$float = 10.5;
$string = "string";
$boolean = true; // atau false
```

PHP Operators

PHP Operator

Operator digunakan untuk melakukan operasi pada variable dan nilai. PHP membagi operator dalam kelompok berikut :

1. Operator aritmatika
2. Operator penugasan
3. operator perbandingan
4. Operator Increment/Decrement
5. Operator logika

Operator Aritmatika (1)

Operator	Keterangan
+$variable	Positif
-$variable	Negatif
$variable + $variable	Penambahan
$variable - $variable	Pengurangan
$variable / $variable	Pembagian

Operator Aritmatika (2)

Operator	Keterangan
$variable % $variable	Sisa Bagi
$variable ** $variable	Pangkat

Kode: Operator Aritmatika

```php
$pertambahan = 10 + 10;
var_dump($pertambahan); // 20

$pengurangan = 10 - 5;
var_dump($pengurangan); // 5
```

Operator Penugasan

- Operator Penugasan di PHP sama seperti bahasa pemrograman lain, yaitu dengan menggunakan karakter = (sama dengan)
- Operator penugasan sudah sering kita gunakan, terutama ketika mengubah value sebuah variable
- Namun selain hal itu, operasi penugasan juga bisa digunakan untuk operasi aritmatika

Operator Penugasan Aritmatika

Penugasan	Keterangan
$a += $b	$a = $a + $b
$a -= $b	$a = $a - $b
$a *= $b	$a = $a * $b
$a /= $b	$a = $a / $b
$a %= $b	$a = $a % $b

Kode: Operator Penugasan Aritmatika

```php
$total = 0;

$apel = 10000;
$ayam = 35000;
$airMineral = 5000;

$total += $apel; // 10000
$total += $ayam; // 45000
$total += $airMineral; // 50000

var_dump($total); // 50000
```

Operator Perbandingan

- Operator Perbandingan, seperti namanya, digunakan untuk membandingkan dua buah value/variable
- hasil dari operator perbandingan adalah boolean, **true** jika benar, dan **false** jika salah

Operator Perbandingan (1)

Operator	Nama	Keterangan
$a == $b	Sama dengan	true jika $a sama dengan $b, dan sebaliknya
$a === $b	Identik	true jika $a sama dengan $b jika memiliki tipe data yang sama
$a != $b	Tidak sama dengan	true jika $a tidak sama dengan $b, dan sebaliknya

Operator Perbandingan (2)

Operator	Nama	Keterangan
$a !== $b	Tidak identik	true jika $a tidak sama dengan $b atau tidak sama tipe data
$a < $b	Kurang dari	true jika $a kurang dari $b, dan sebaliknya
$a <= $b	Kurang dari sama dengan	true jika $a kurang atau sama dengan $b, dan sebaliknya

Operator Perbandingan (3)

Operator	Nama	Keterangan
$a > $b	Lebih dari	true jika $a lebih dari $b, dan sebaliknya
$a >= $b	Lebih dari sama dengan	true jika $a lebih atau sama dengan $b, dan sebaliknya

Kode: Operator Perbandingan

```php
var_dump("10" == 10); // true
var_dump("10" === 10); // false

var_dump(10 > 9); // true
var_dump(10 >= 10); // true
```

Operator Logika

- Operator logika adalah operator untuk membandingkan dua nilai boolean
- Hasil operator logika adalah boolean lagi

Operator Logika

Operator	Nama	Keterangan
$a && $b	And	true jika $a dan $b keduanya bernilai true
$a || $b	Or	true jika $a dan $b salah satu atau keduanya bernilai true
!$a	Not	true jika $a bernilai false

Kode: Operator Logika

```php
var_dump(true && true); // true
var_dump(true && false); // false

var_dump(true || false); // true
var_dump(false || false); // false

var_dump(!false); // true
var_dump(!true); // false
```

Operator Increment dan Decrement

- PHP mendukung gaya bahasa pemrograman C untuk menaikkan dan menurunkan data number sejumlah 1 angka
- Ini bisa mempersingkat kita ketika ingin menaikkan data

Operator Increment dan Decrement

Contoh	Nama	Keterangan
$a++	Post Increment	Kembalikan $a lalu naikkan 1 angka
++$a	Pre Increment	Naikkan $a satu angka, lalu kembalikan $a
$a–	Post Decrement	Kembalikan $a lalu turunkan 1 angka
–$a	Pre Increment	Turunkan $a satu angka, lalu kembalikan $a

Kode: Operator Increment & Decrement

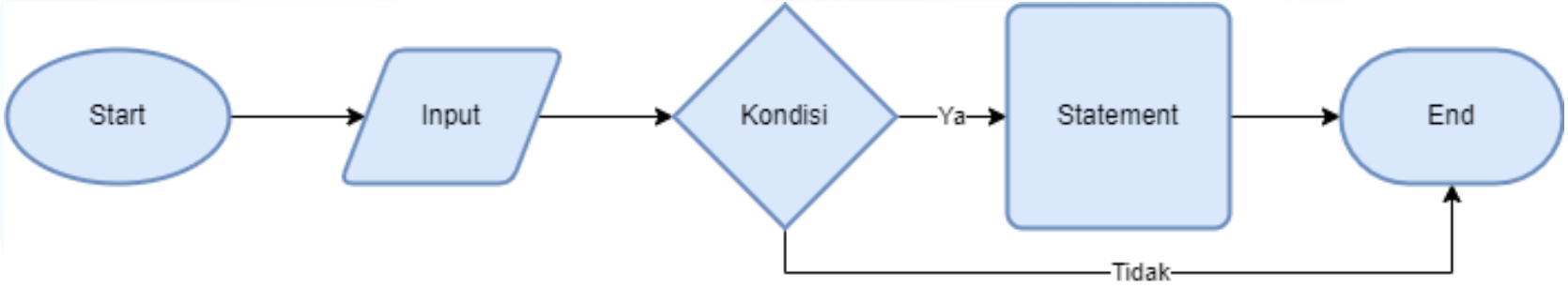
```php
$a = 10;
$b = ++$a;
// $a = $a + 1; -> 11
// $b = $a; -> 11

var_dump($b); // 11
var_dump($a); // 11

$c = 10;
$d = $c++;
// $d = $c; -> 10
// $c = $c + 1; -> 11

var_dump($d); // 10
var_dump($c); // 11
```

Percabangan

Percabangan

- Percabangan merupakan suatu perintah yang berfungsi untuk mengeksekusi suatu tindakan berdasarkan apakah kondisi tersebut benar atau salah sesuai yang sudah ditetapkan oleh pembuat program.

Percabangan if dengan satu kondisi



Membuat atau membuka project Laravel dengan VS Code atau text editor lain. Kemudian buat controller baru untuk percabangan dengan memasukkan perintah pada terminal

php artisan make:controller *PercabanganController*

```
PS C:\xampp\htdocs\php-dasar> php artisan make:controller PercabanganController
Controller created successfully.
```

PercabanganController.php

app > Http > Controllers > PercabanganController.php > PercabanganController

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PercabanganController extends Controller
{
    //
}

```

Menambahkan fungsi baru pada controller untuk **Percabangan if dengan Satu Kondisi** seperti pada gambar.

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PercabanganController extends Controller
{
    public function pertama(){
        $kegiatan = "Pergi keluar";

        $hujan = true;

        if($hujan){
            $kegiatan = $kegiatan." membawa payung";
        }

        return $kegiatan;
    }
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya.

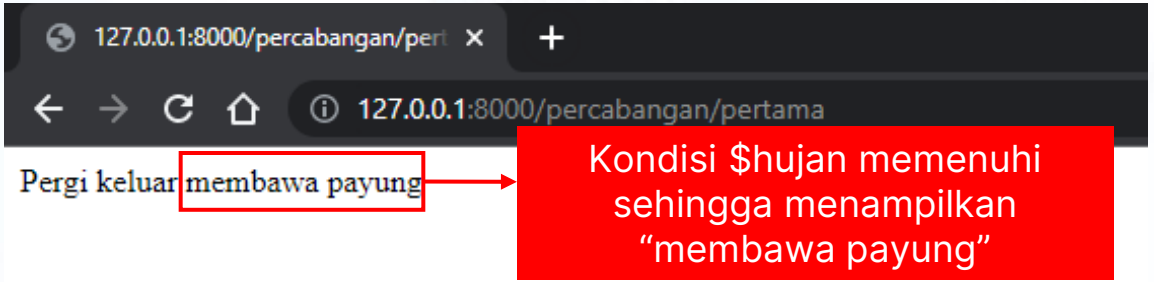
```
20    Route::get('/percabangan/pertama', 'App\Http\Controllers\PercabanganController@pertama');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

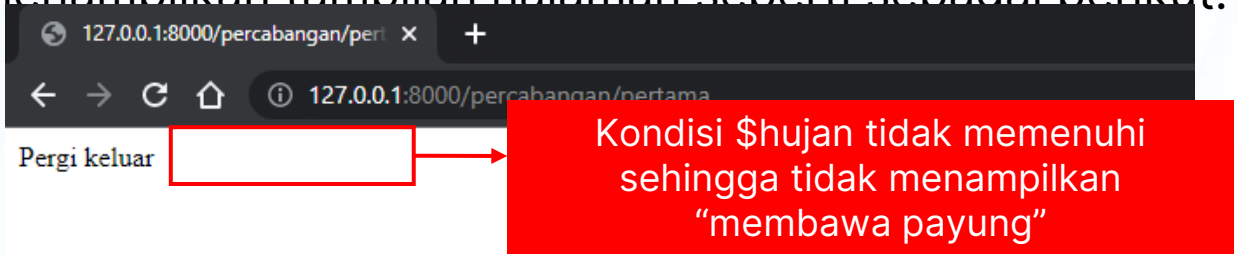
Buka browser kemudian masukan URL **http://127.0.0.1:8000/percabangan/pertama** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



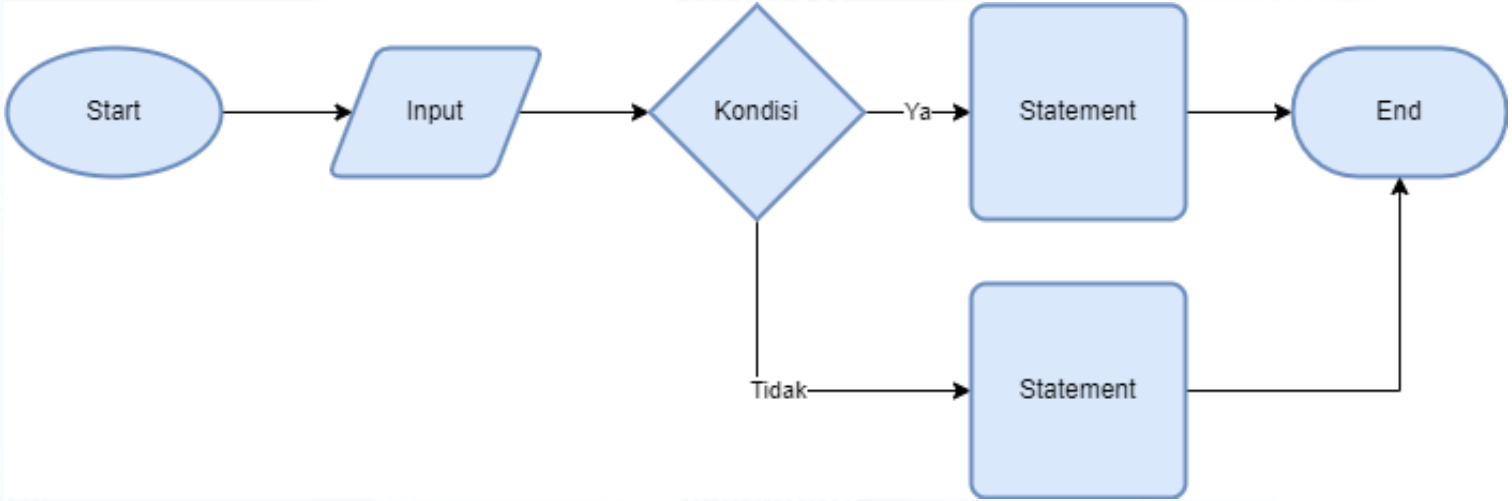
Coba mengubah nilai variable **$hujan** dengan false pada file **PercabanganController.php**

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PercabanganController extends Controller
{
    public function pertama(){
        $kegiatan = "Pergi keluar";

        $hujan = false;

        if($hujan){
            $kegiatan = $kegiatan." membawa payung";
        }

        return $kegiatan;
    }
}
```

Refresh browser dengan URL **http://127.0.0.1:8000/percabangan/pertama** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Percabangan if dan else dengan satu kondisi



Menambahkan fungsi baru pada controller untuk **Percabangan if dan else dengan Satu Kondisi** seperti pada gambar.

```php
public function kedua(){
    $umur = 30;

    if($umur<18){
        $hasil = "Belum cukup umur.";
    }
    else{
        $hasil = "Cukup umur.";
    }
    return $hasil;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

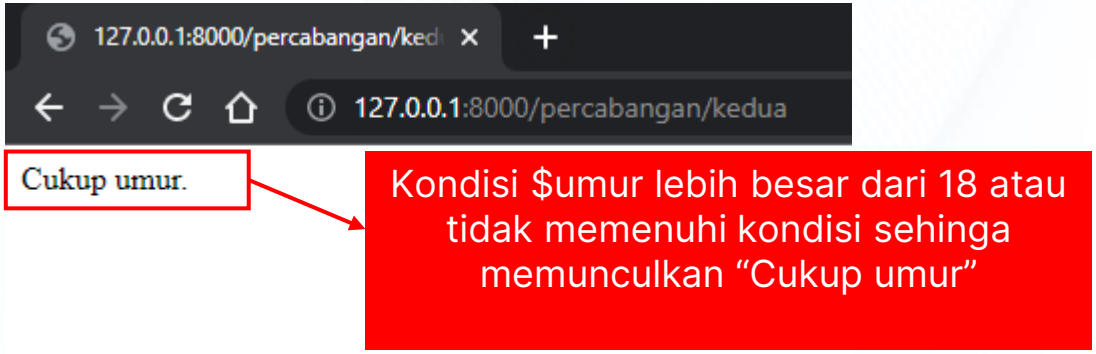
```
Route::get('/percabangan/kedua', 'App\Http\Controllers\PercabanganController@kedua');
```

Jika project belum dijalankan, maka jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

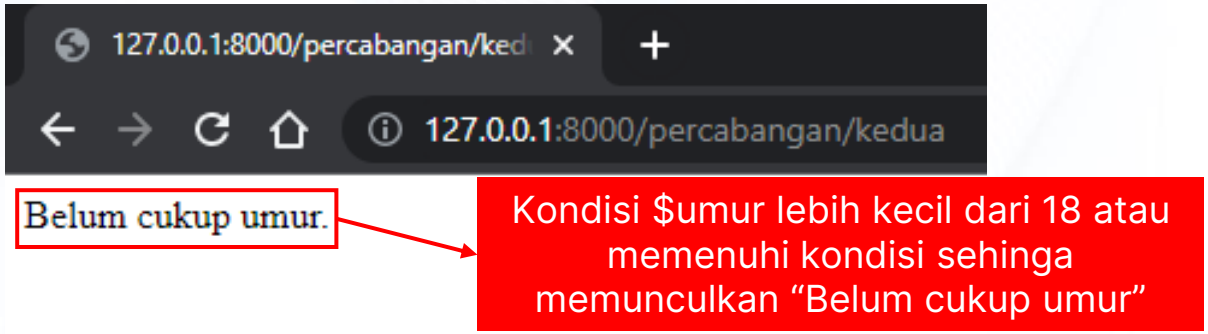
Jika sudah refresh browser dengan URL **http://127.0.0.1:8000/percabangan/kedua** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



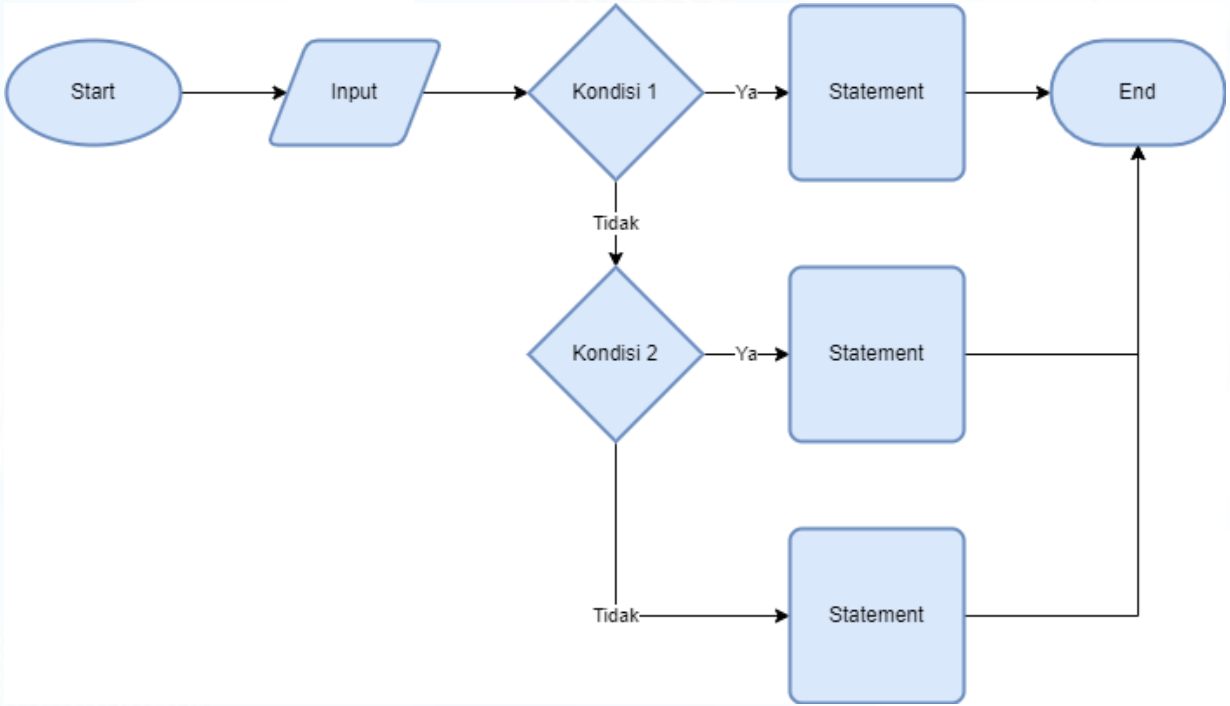
Coba mengubah nilai variable **$umur** dengan angka lebih kecil dari 18 pada file **PercabanganController.php ,** misalnya diubah menjadi angka 10.

```php
public function kedua(){
    $umur = 10;

    if($umur<18){
        $hasil = "Belum cukup umur.";
    }
    else{
        $hasil = "Cukup umur.";
    }
    return $hasil;
}
```

Refresh browser dengan URL **http://127.0.0.1:8000/percabangan/kedua** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Percabangan if dan else dengan banyak kondisi



Menambahkan fungsi baru pada controller untuk **Percabangan if dan else dengan Banyak Kondisi** seperti pada gambar.

```php
public function ketiga(){
    $angka = 33;

    if($angka == 0){
        $hasil = "Bilangan 0";
    }
    else if($angka % 2 == 0){
        $hasil = "Bilangan Genap";
    }
    else{
        $hasil = "Bilangan Ganjil";
    }

    return $hasil;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

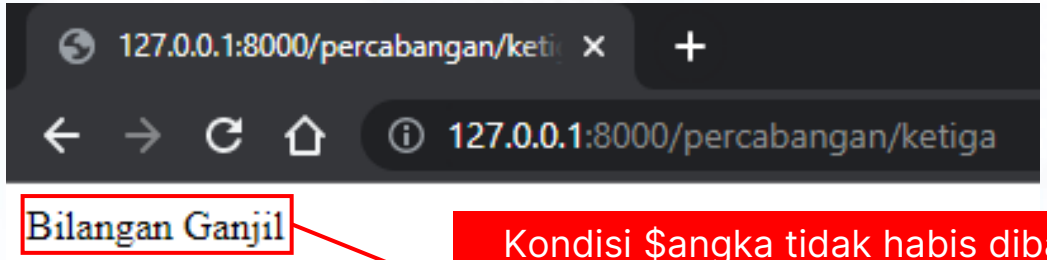
```
Route::get('/percabangan/ketiga', 'App\Http\Controllers\PercabanganController@ketiga');
```

Jika project belum dijalankan, maka jalankan project dengan memasukkan perintah pada terminal

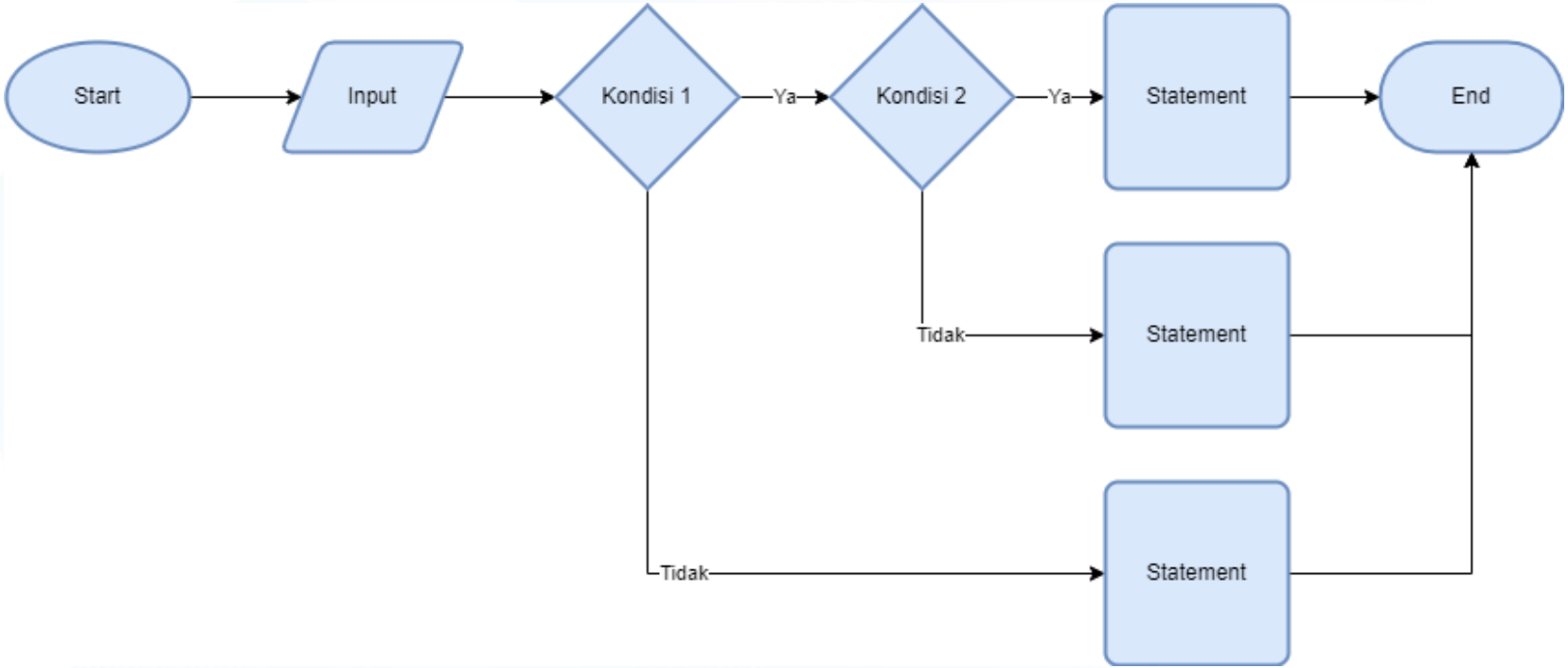
php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Jika sudah refresh browser dengan URL **http://127.0.0.1:8000/percabangan/ketiga** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Percabangan Nested if



Menambahkan fungsi baru pada controller untuk **Percabangan Nested if** seperti pada gambar.

```php
public function keempat(){
    $cintaku = 13;
    $cintadia = 9;

    if ($cintaku > $cintadia){
        $usahadia = 4;
        $cintadia = $cintadia + $usahadia;

        if($cintaku > $cintadia){
            $hasil = "Kamu mencintaiku";
        }
        else{
            $hasil = "Kamu mencintai dia";
        }
    }
    else{
        $hasil = "Kamu mencintai dia";
    }
    return $hasil;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat

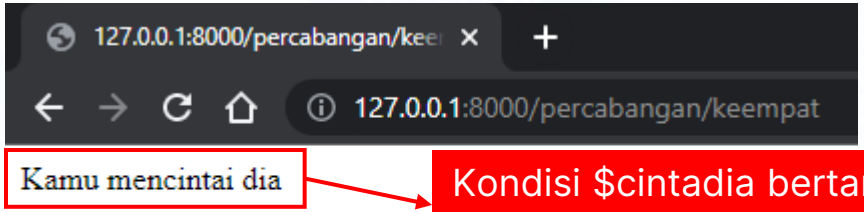
```
Route::get('/percabangan/keempat', 'App\Http\Controllers\PercabanganController@keempat');
```

Jika project belum dijalankan, maka jalankan project dengan memasukkan perintah pada terminal

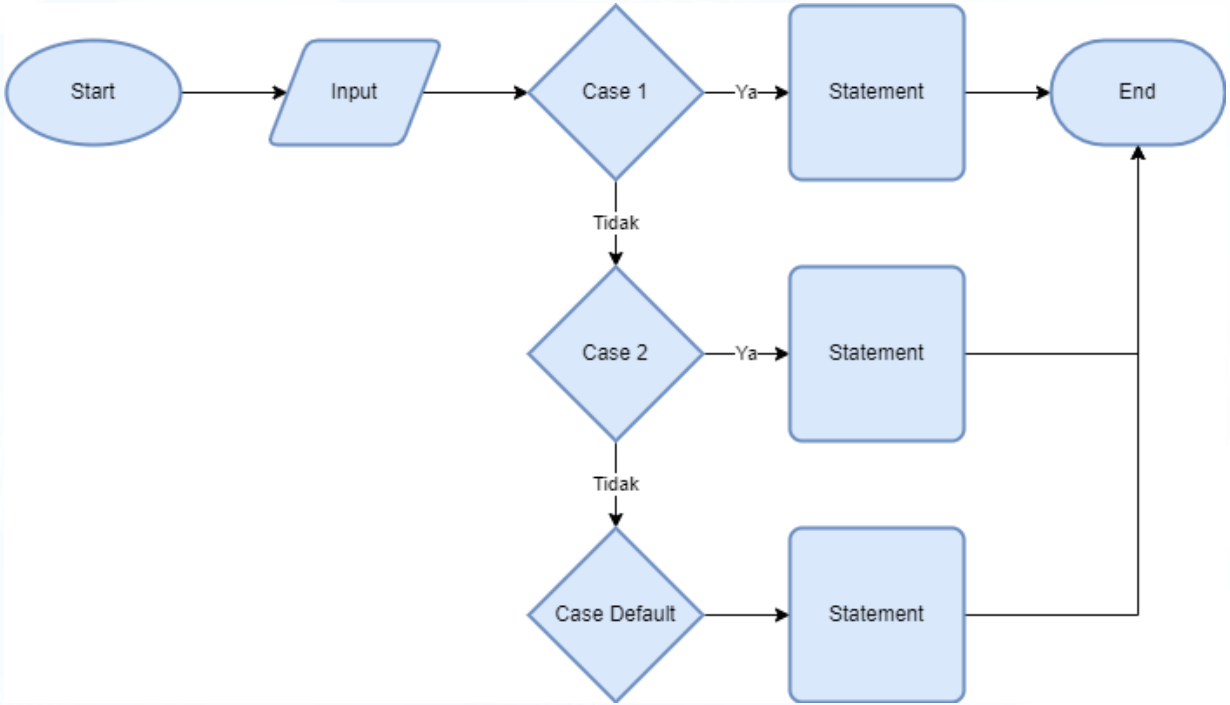
php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Jika sudah refresh browser dengan URL **http://127.0.0.1:8000/percabangan/keempat** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Percabangan switch case



Menambahkan fungsi baru pada controller untuk **Percabangan Switch Case** seperti pada gambar.

```php
public function kelima(){
    $nilai = "C";

    switch($nilai){
        case "A":
            $hasil = "Sangat Baik";
            break;
        case "B":
            $hasil = "Baik";
            break;
        case "C":
            $hasil = "Cukup";
            break;
        case "D":
            $hasil = "Kurang";
            break;
        case "E":
            $hasil = "Sangat Kurang";
            break;
        default:
            $hasil = "Nilai Tidak Valid";
    }
    return $hasil;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

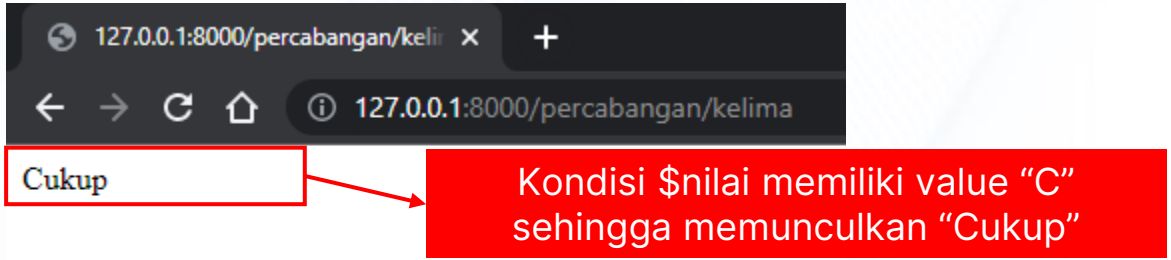
```
Route::get('/percabangan/kelima', 'App\Http\Controllers\PercabanganController@kelima');
```

Jika project belum dijalankan, maka jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Jika sudah refresh browser dengan URL **http://127.0.0.1:8000/percabangan/kelima** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Array

Array

- Array adalah struktur data yang memungkinkan untuk menyimpan beberapa value dalam satu variabel.

Contoh:

```php
$makananFavorit = ["ayam", "pecel"];
$makananFavorit = array("ayam", "pecel");
```

Membuat atau membuka project Laravel dengan VS Code atau text editor lain. Kemudian buat controller baru untuk percabangan dengan memasukkan perintah pada terminal

php artisan make:controller *ArrayController*

```
PS C:\xampp\htdocs\php-dasar> php artisan make:controller ArrayController
Controller created successfully.
```

Menambahkan fungsi baru pada controller untuk **Array** seperti pada gambar.

```php
public function pertama(){
    $makananFavorit = ["ayam","pecel"];
    $minumanFavorit = array("teh","kopi");

    foreach($makananFavorit as $makananku){
        echo "$makananku, ";
    }

    echo "<br>"; //ini enter

    foreach($minumanFavorit as $minumanku){
        echo "$minumanku, ";
    }

}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat

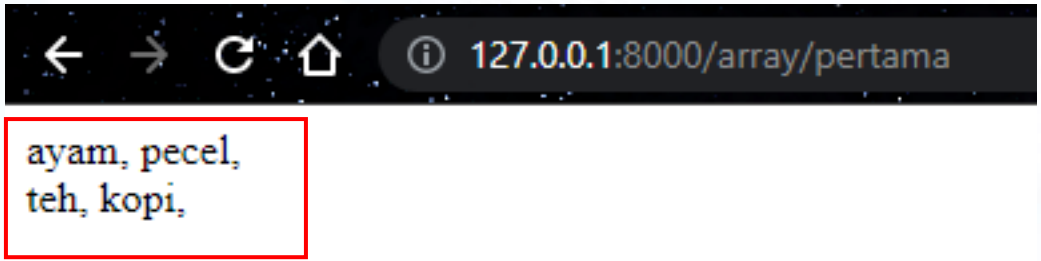
```
Route::get('/array/pertama', 'App\Http\Controllers\ArrayController@pertama');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/array/pertama** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Indexes Arrays

- Indexed Arrays merupakan array yang menggunakan angka sebagai index/key
- secara default arrays pada php menggunakan Indexed Arrays.

Contoh:

```php
$makanan = ["ayam", "pecel", "sate"];
echo "Saya suka makan " . $makanan[0] . ", " . $makanan[1] . " dan " . $makanan[2] . ".";
```

Menambahkan fungsi baru pada controller untuk **Array** seperti pada gambar.

```php
public function kedua(){
    $makanan = ["ayam", "pecel", "sate"];
    echo "Saya suka makan ". $makanan[0].", ".$makanan[1].", dan ".$makanan[2].".";

}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat

```
Route::get('/array/kedua', 'App\Http\Controllers\ArrayController@kedua');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/array/kedua** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.

Saya suka makan ayam, pecel, dan sate.

Associative Arrays

- Associative Arrays merupakan array yang dapat menggunakan keys yang ditentukan biasanya berupa string.

Contoh:

```php
$makananFavorit = ["Budi" =>"ayam", "Ayah" => "pecel"];
echo "Budi suka makan ". $makananFavorit['Budi']. " dan ayah suka makan ". $makananFavorit['Ayah'];
```

Menambahkan fungsi baru pada controller untuk **Array** seperti pada gambar.

```
public function ketiga(){
    $makananFavorit = ["Budi"=>"ayam", "ayah"=>"pecel"];
    echo "Budi suka makan ".$makananFavorit["Budi"]." dan ayah suka makan ".$makananFavorit["ayah"];
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

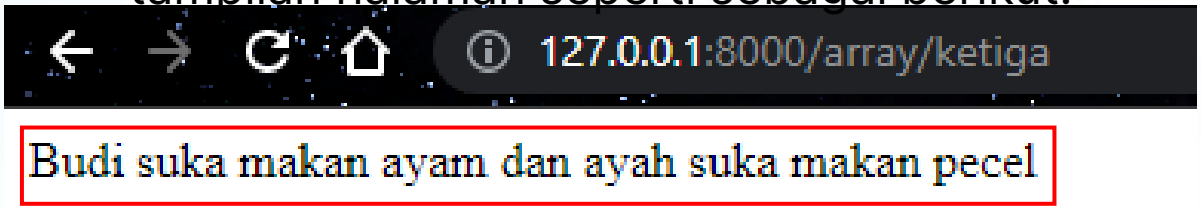
```
Route::get('/array/ketiga', 'App\Http\Controllers\ArrayController@ketiga');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/array/ketiga** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Multidimensional Arrays

- Multidimensional Arrays merupakan array yang memiliki array sebagai valuenya.

Contoh:

```php
$dataStatus = [
    ["Budi", 12, "jomblo"],
    ["Joko", 24, "menikah"]
];
echo $dataStatus[0][0]. " berumur ". $dataStatus[0][1]. " berstatus ". $dataStatus[0][2];
// return Budi berumur 12 berstatus jomblo
echo $dataStatus[1][0]. " berumur ". $dataStatus[1][1]. " berstatus ". $dataStatus[1][2];
// return Joko berumur 24 berstatus menikah
```

Menambahkan fungsi baru pada controller untuk **Array** seperti pada gambar.

```php
public function keempat(){
    $dataStatus = [
        ["Budi", 12, "jomblo"],
        ["Joko", 24, "menikah"]
    ];

    echo $dataStatus[0][0]. " berumur ".$dataStatus[0][1]." berstatus ".$dataStatus[0][2]. "<br>";
    echo $dataStatus[1][0]. " berumur ".$dataStatus[1][1]." berstatus ".$dataStatus[1][2];

}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

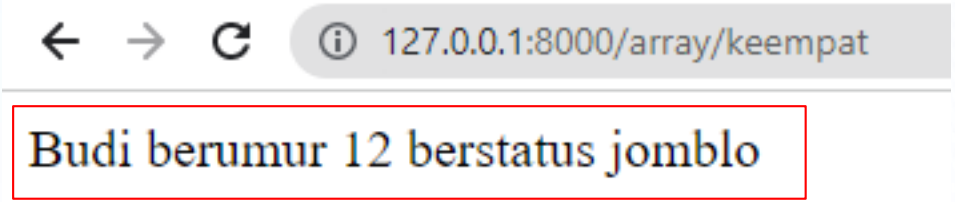
```
Route::get('/array/keempat', 'App\Http\Controllers\ArrayController@keempat');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/array/keempat** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



String

String

Pada bahasa pemrograman string adalah sebuah tipe data yang berisi text, kalimat, atau kumpulan karakter.

Contoh “Hello World!” , “Selamat Siang”, “ini string”, dll.

PHP String Functions

Pada php terdapat fungsi - fungsi untuk memanipulasi string

- **strlen()**
  Fungsi untuk mengambil panjang dari string($string)
  Example :

```php
echo strlen("Hello"); // return 5
```

- **str_replace()**
  Fungsi untuk me-replace string pada sebuah string
  Example :

```php
echo str_replace("siang", "malam", "Selamat siang!"); // return "Selamat malam"
```

PHP String Functions

Pada php terdapat fungsi - fungsi untuk memanipulasi string

- **str_word_count()**
  Fungsi untuk mengambil jumlah kata dari string
  Example :

```php
echo str_word_count("kucing tidak bisa terbang"); // outputs 4
```

- **strpos()**
  Fungsi untuk mengambil posisi string pada sebuah string
  Example :

```php
echo strpos("Selamat siang bapak!", "siang"); // outputs 8
```

Looping

Looping

Looping atau perulangan merupakan suatu perintah yang berfungsi untuk melakukan sesuatu secara berulang-ulang dengan menggunakan kode program dalam batas yang telah ditentukan. Dengan menggunakan proses perulangan, penulisan kode program dapat menjadi lebih efisien.

Perulangan for

merupakan suatu jenis perulangan yang berfungsi untuk mengulang suatu proses yang telah diketahui jumlahnya

Membuat atau membuka project Laravel dengan VS Code atau text editor lain. Kemudian buat controller baru untuk percabangan dengan memasukkan perintah pada terminal

php artisan make:controller *LoopingController*

```
PS C:\xampp\htdocs\php-dasar> php artisan make:controller LoopingController
Controller created successfully.
```

Menambahkan fungsi baru pada controller untuk **Looping** seperti pada gambar.

```php
public function pertama(){
    $total = 5;
    $hasilFaktorial = 1;

    for($i=1; $i<=$total; $i++){
        $hasilFaktorial = $hasilFaktorial * $i;
    }

    return $hasilFaktorial;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

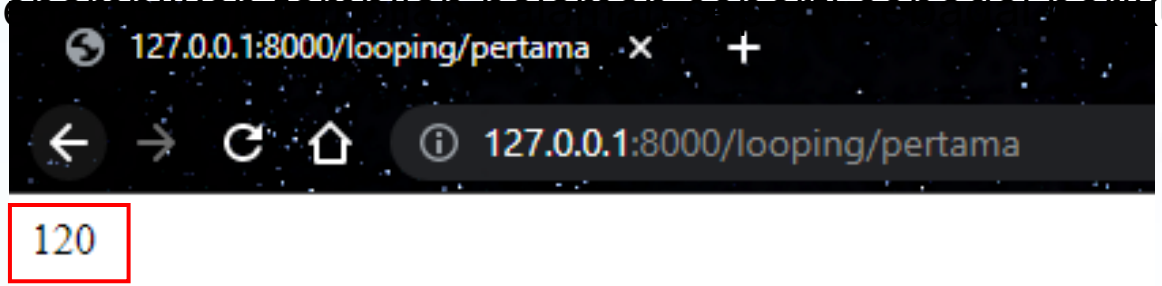
```
Route::get('/looping/pertama', 'App\Http\Controllers\LoopingController@pertama');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/looping/pertama** , maka browser akan m[illegible]ut.



Perulangan while

merupakan jenis perulangan yang akan selesai bila tidak memenuhi kondisi yang diberikan

Menambahkan fungsi baru pada controller untuk **Looping** seperti pada gambar.

```php
public function kedua(){
    $counter = 0;

    while($counter < 10){
        echo "Saya berjanji tidak akan melakukannya lagi <br>";

        $counter++;
    }
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

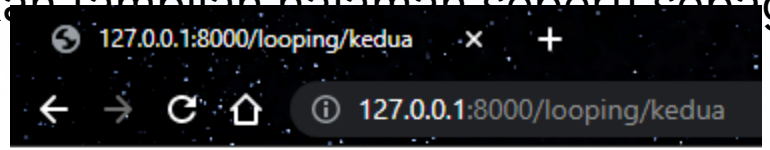
```
Route::get('/looping/kedua', 'App\Http\Controllers\LoopingController@kedua');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/looping/kedua** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi
Saya berjanji tidak akan melakukannya lagi

Percabangan do while

merupakan jenis percabangan yang sama seperti while, namun suatu instruksi dijalankan terlebih dahulu baru setelah itu dicek kondisinya

Menambahkan fungsi baru pada controller untuk **Looping** seperti pada gambar.

```php
public function ketiga(){
    $angka = -2;

    do {
        $angka--;
    }while($angka > 0);

    return "angka = ".$angka;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

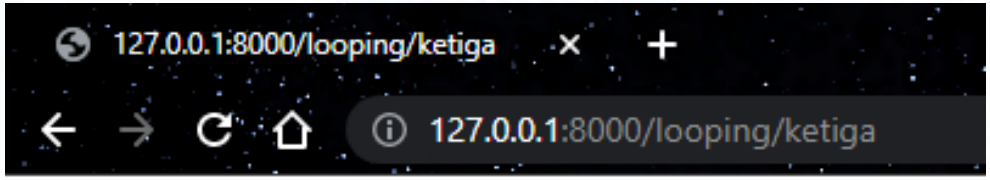
```
Route::get('/looping/ketiga', 'App\Http\Controllers\LoopingController@ketiga');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/looping/ketiga** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Percabangan foreach

merupakan perulangan khusus untuk array

Menambahkan fungsi baru pada controller untuk **Looping** seperti pada gambar.

```php
public function keempat(){
    $makananFavorit = [
        "Budi"=>"ayam",
        "Ayah"=>"pecel",
        "Ibu"=>"nasi padang",
        "Joko"=>"gado-gado",
        "Sari"=>"bakso"
    ];

    foreach($makananFavorit as $key => $makanan){
        echo $key." suka makan ".$makanan.". <br>";
    }

}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

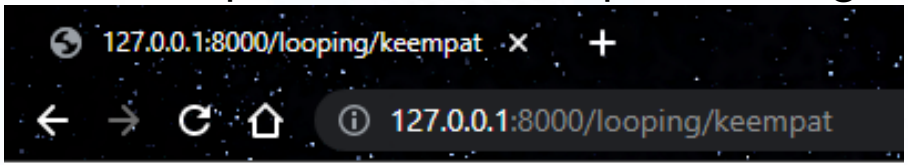
```
Route::get('/looping/keempat', 'App\Http\Controllers\LoopingController@keempat');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/looping/keempat** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Budi suka makan ayam.
Ayah suka makan pecel.
Ibu suka makan nasi padang.
Joko suka makan gado-gado.
Sari suka makan bakso.

Function

Function

function merupakan sekumpulan intruksi yang dibungkus dalam sebuah blok, sehingga kita bisa menggunakannya tanpa menulis ulang intruksi didalamnya. function bisa menerima input berupa parameter dan bisa mengembalikan nilai

Fungsi dengan parameter

sebuah function bisa menerima input berupa parameter yang akan dipakai didalam fungsi tersebut

contoh:

```
public function printHello(string $nama){
    echo "Halo ".$nama;
}
```

Membuat atau membuka project Laravel dengan VS Code atau text editor lain. Kemudian buat controller baru untuk percabangan dengan memasukkan perintah pada terminal

php artisan make:controller *FunctionController*

```
PS C:\xampp\htdocs\php-dasar> php artisan make:controller FunctionController
Controller created successfully.
```

Menambahkan fungsi baru pada controller untuk **Function** seperti pada gambar.

```php
public function pertama(){
    $this->printHello("Dunia", 2);
}

public function printHello(string $nama, int $angka){
    echo "Halo ".$nama.". Ini angka ".$angka;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

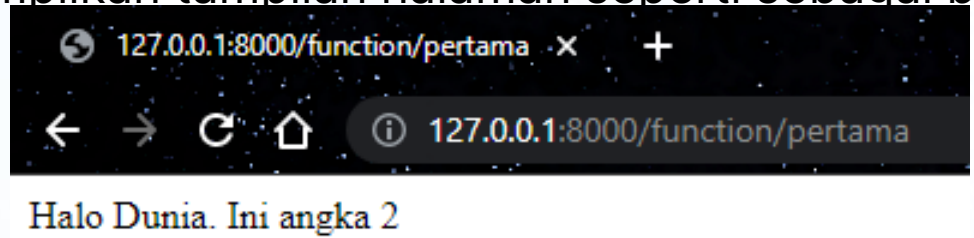
```
Route::get('/function/pertama', 'App\Http\Controllers\FunctionController@pertama');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/function/pertama** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Parameter default

parameter di dalam function bisa diset default bila kita memerlukan nilai default dari parameter

contoh:

```php
public function makan(bool $lapar=false){
    if($lapar){
        echo "Ayo makan";
    }else{
        echo "Sudah kenyang";
    }
}
```

Menambahkan fungsi baru pada controller untuk **Function** seperti pada gambar.

```php
public function kedua(){
    $this->makan();
}

public function makan(bool $lapar = false){
    if($lapar){
        echo "Ayo makan";
    }
    else{
        echo "Sudah kenyang";
    }
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat

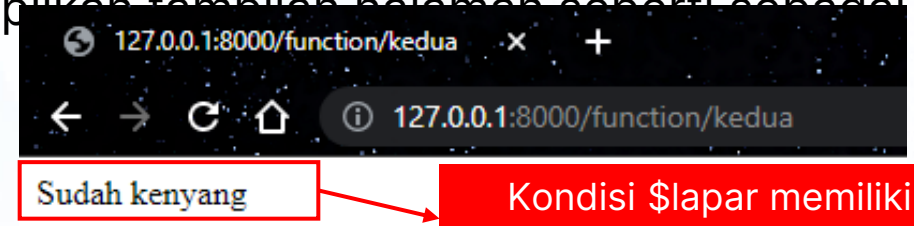
```
Route::get('/function/kedua', 'App\Http\Controllers\FunctionController@kedua');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/function/kedua** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Kondisi $lapar memiliki value false sehingga memunculkan “Sudah kenyang”

Fungsi mengembalikan nilai

sebuah fungsi juga bisa mengembalikan nilai yang kita butuhkan

contoh:

```php
public function menjumlahDuaAngka(int $angka1, int $angka2){
    $hasil = $angka1 + $angka2;

    return $hasil;
}
```

Menambahkan fungsi baru pada controller untuk **Function** seperti pada gambar.

```php
public function ketiga(){
    $hasilPenjumlahan = $this->menjumlahDuaAngka(45, 7);
    return $hasilPenjumlahan;
}

public function menjumlahDuaAngka(int $angka1, int $angka2){
    $hasil = $angka1 + $angka2;
    return $hasil;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat

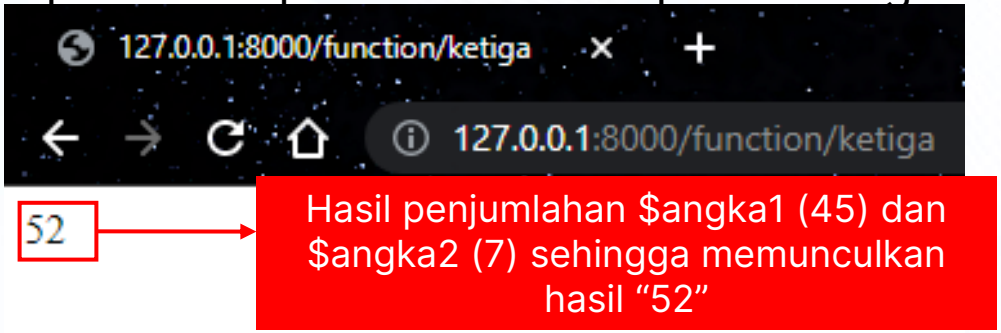
```
Route::get('/function/ketiga', 'App\Http\Controllers\FunctionController@ketiga');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/function/ketiga** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Memanggil fungsi didalam fungsi

sebuah fungsi juga bisa memanggil fungsi yang lain

contoh:

```php
public function pembagianDuaAngka(int $angka1, int $angka2){
    $isZeroValue = $this->cekNilaiNol($angka2);
    if($isZeroValue){
        return "Angka kedua tidak boleh 0";
    }
    $hasil = $angka1 / $angka2;

    return $hasil;
}
```

```php
public function cekNilaiNol(int $angka){
    if($angka == 0){
        return true;
    }

    return false;
}
```

Menambahkan fungsi baru pada controller untuk **Function** seperti pada gambar.

```php
public function keempat(){
    $hasilPembagian = $this->pembagianDuaAngka(44,9);
    return $hasilPembagian;
}

public function pembagianDuaAngka(int $angka1, int $angka2){
    $isZeroValue = $this->cekNilaiNol($angka2);
    if($isZeroValue){
        return "Angka kedua tidak boleh 0";
    }
    $hasil = $angka1 / $angka2;

    return $hasil;
}

public function cekNilaiNol(int $angka){
    if($angka == 0){
        return true;
    }
    else return false;
}
```

Menambahkan router pada file **web.php** untuk menentukan halaman yang akan mengakses fungsi yang telah dibuat sebelumnya

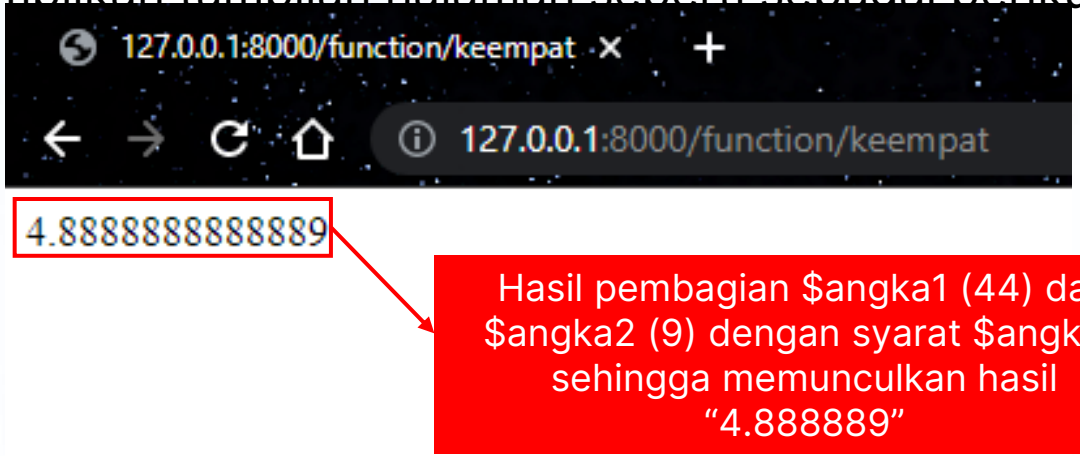
```
Route::get('/function/keempat', 'App\Http\Controllers\FunctionController@keempat');
```

Jalankan project dengan memasukkan perintah pada terminal

php artisan serve

```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Buka browser kemudian masukan URL **http://127.0.0.1:8000/function/keempat** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Terima Kasih