

VARIABEL, PERCABANGAN, ARRAY & STRING, LOOPING & FUNCTION

Gede & Ardian

Materi

1. [Variabel](#)
2. [Percabangan](#)
3. [Array](#)
4. [String](#)
5. [Looping](#)
6. [Function](#)

Variabel

Variabel

variabel merupakan sebuah wadah untuk menampung sebuah data
contoh:

```
1  <?php
2  $variabel1 = 1;
3  $nama_variabel = "nama";
4  $iniVariabel = true;
```

Type Data Variabel

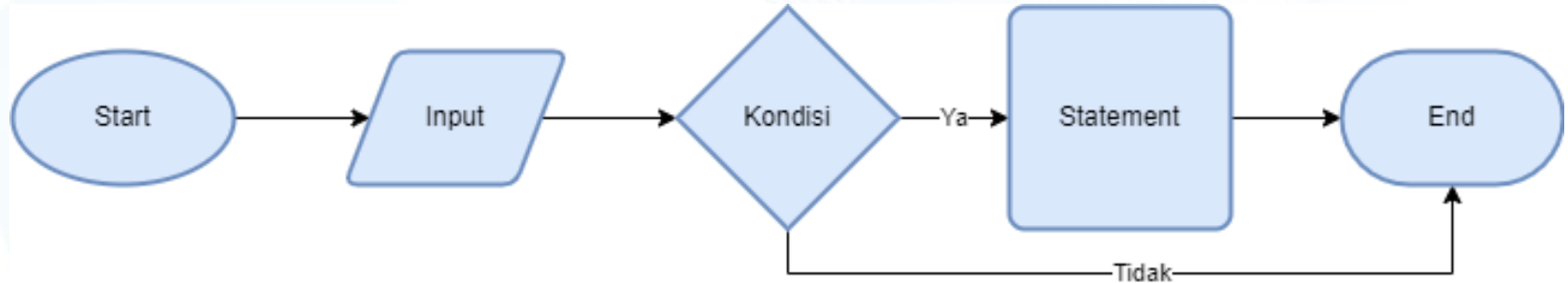
```
2  $variabelInteger = 15;  
3  $variabelString = "string";  
4  $variabelFloat = 1.327;  
5  $variabelBoolean = false;  
6
```

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 X
app > Http > Controllers > PercabanganController.php > PercabanganController
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PercabanganController extends Controller
8  {
9      //
10 }
11
```

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

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PercabanganController extends Controller
8  {
9      public function pertama(){
10         $kegiatan = "Pergi keluar";
11
12         $hujan = true;
13
14         if($hujan){
15             $kegiatan = $kegiatan." membawa payung";
16         }
17
18         return $kegiatan;
19     }
20 }
```

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

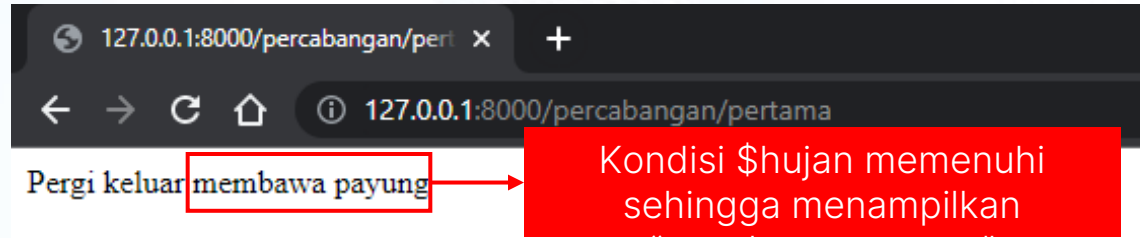
```
20 Route::get('/percabangan/pertama', 'App\Http\Controllers\PercabanganController@pertama');
```

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
```

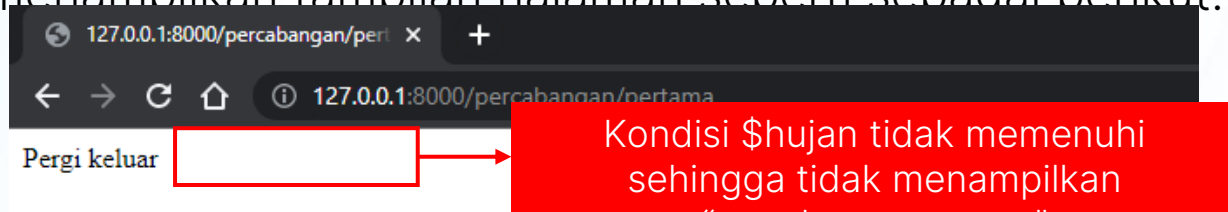
Buka browser kemudian masukan URL **http://127.0.0.1:8000/percabangan/pertama** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



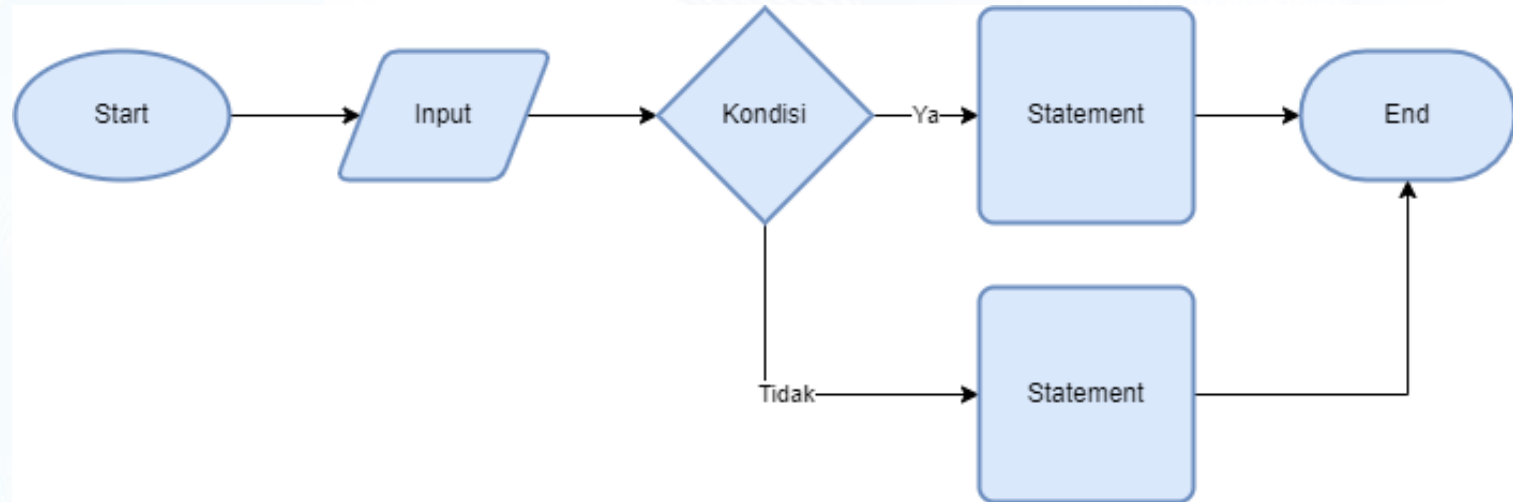
Coba mengubah nilai variable **\$hujan** dengan false pada file **PercabanganController.php**

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PercabanganController extends Controller
8  {
9      public function pertama(){
10         $kegiatan = "Pergi keluar";
11
12         $hujan = false;
13
14         if($hujan){
15             $kegiatan = $kegiatan." membawa payung";
16         }
17
18         return $kegiatan;
19     }
20 }
```

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.

```
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

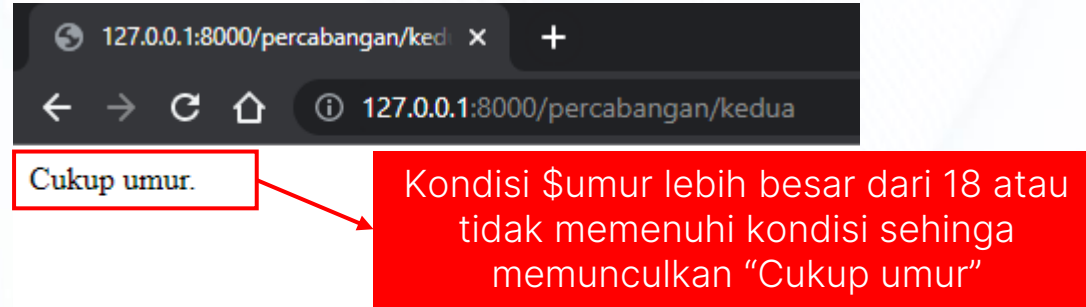
```
Route::get('/percabangan/kedua', 'App\Http\Controllers\PercabanganController@kedua');
```

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
```

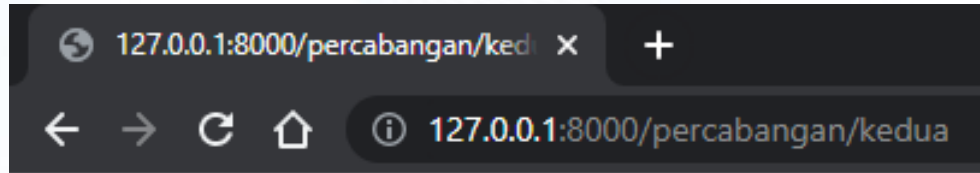
Jika sudah refresh browser dengan URL **<http://127.0.0.1:8000/percabangan/kedua>** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



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

```
public function kedua(){  
    $umur = 10;  
  
    if($umur<18){  
        $hasil = "Belum cukup umur.";  
    }  
    else{  
        $hasil = "Cukup umur.";  
    }  
    return $hasil;  
}
```

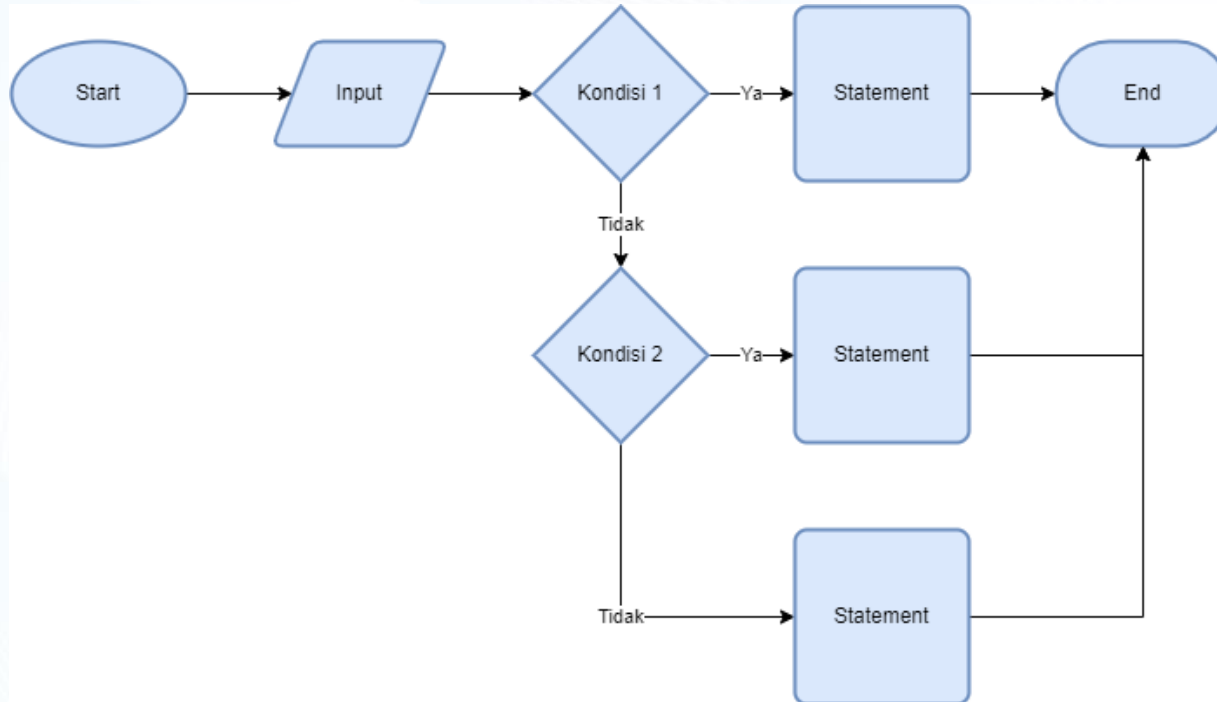
Refresh browser dengan URL **<http://127.0.0.1:8000/percabangan/kedua>** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Belum cukup umur.

Kondisi \$umur lebih kecil dari 18 atau memenuhi kondisi sehingga memunculkan "Belum cukup umur"

Percabangan if dan else dengan banyak kondisi



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

```
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

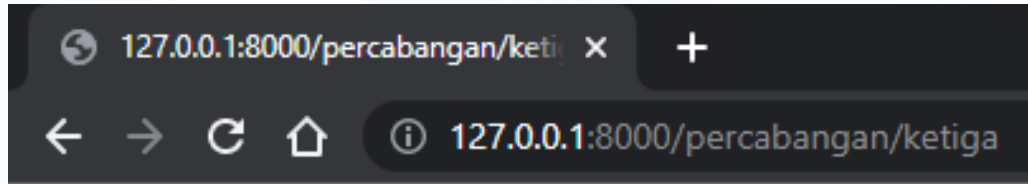
```
Route::get('/percabangan/ketiga', 'App\Http\Controllers\PercabanganController@ketiga');
```

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
```

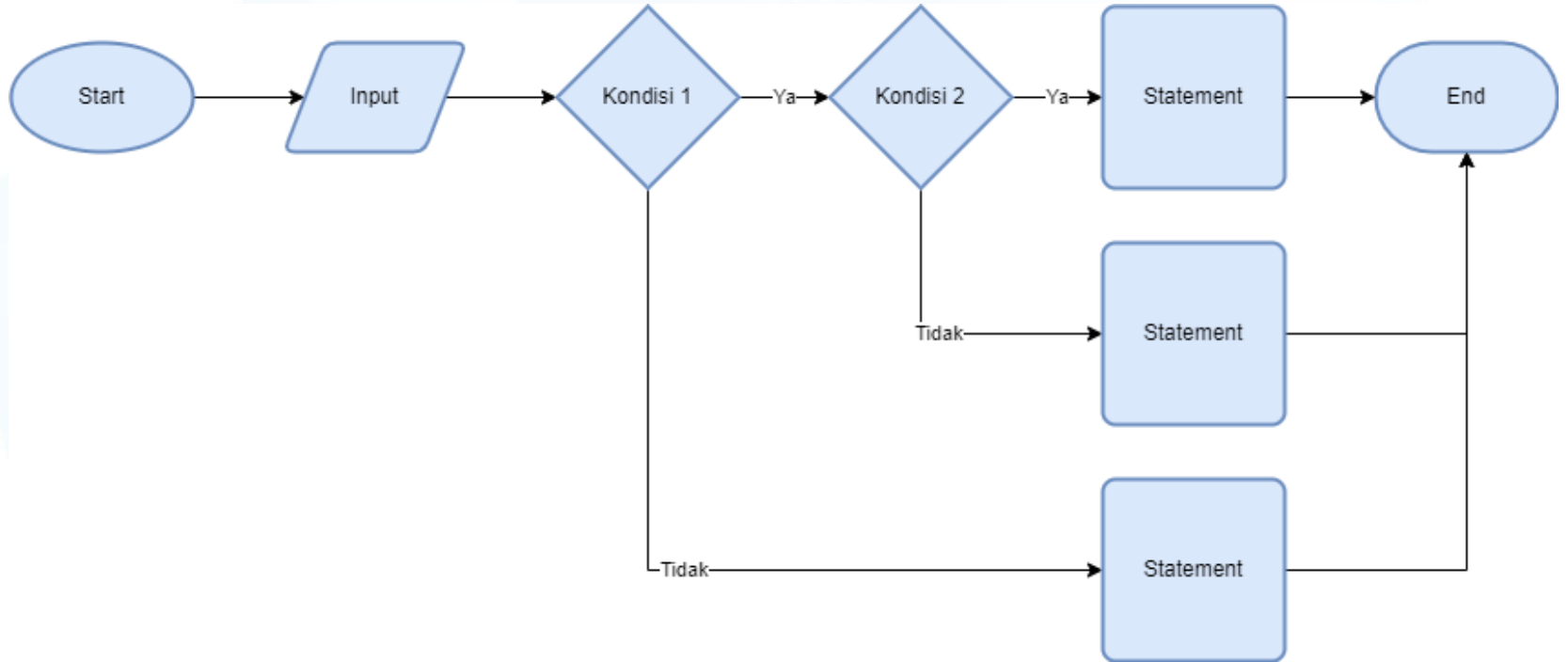
Jika sudah refresh browser dengan URL **<http://127.0.0.1:8000/percabangan/ketiga>** , maka browser akan menampilkan tampilan halaman seperti sebagai berikut.



Bilangan Ganjil

Kondisi \$angka tidak habis dibagi 2 dan bukan angka 0 sehingga memunculkan "Bilangan Ganjil"

Percabangan Nested if



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

```
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

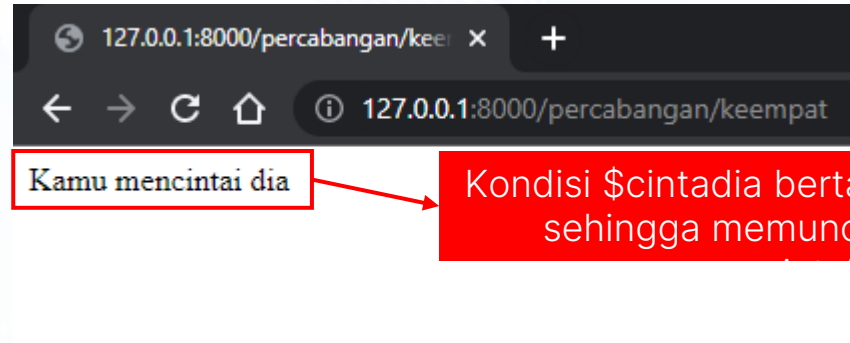
```
Route::get('/percabangan/keempat', 'App\Http\Controllers\PercabanganController@keempat');
```

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

php artisan serve

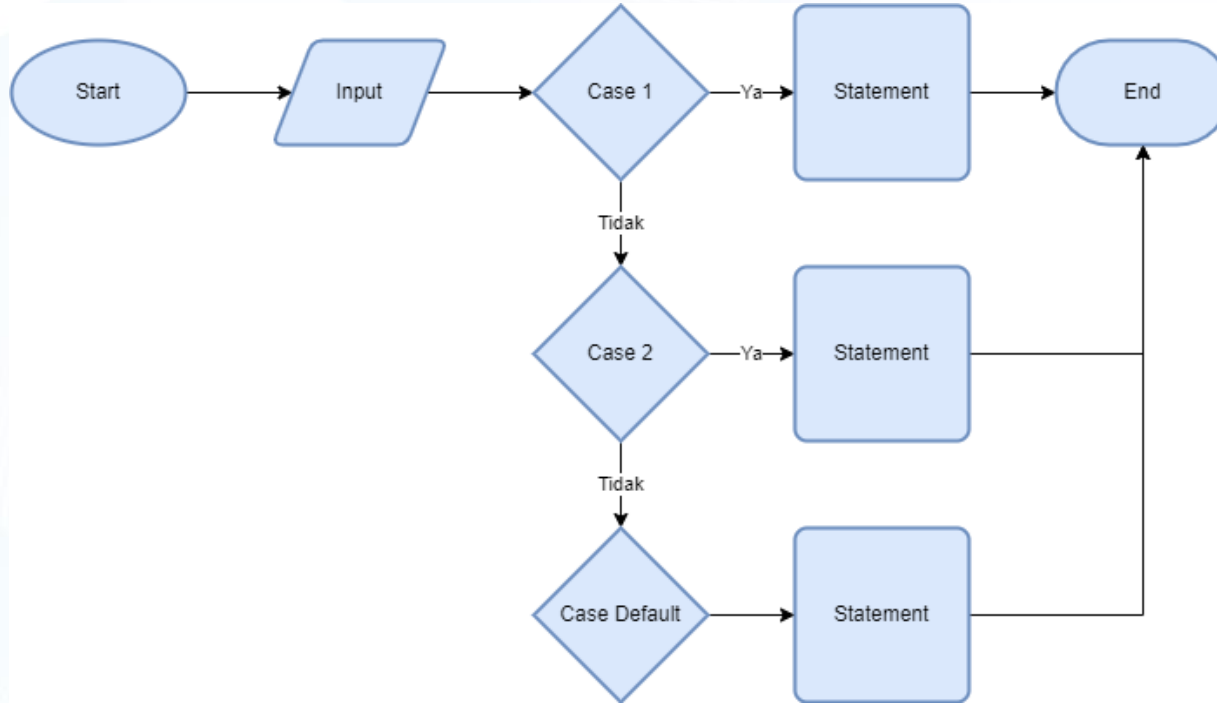
```
PS C:\xampp\htdocs\php-dasar> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

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



Kondisi \$cintadia bertambah \$usahadia sehingga memunculkan "Kamu

Percabangan switch case



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

```
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

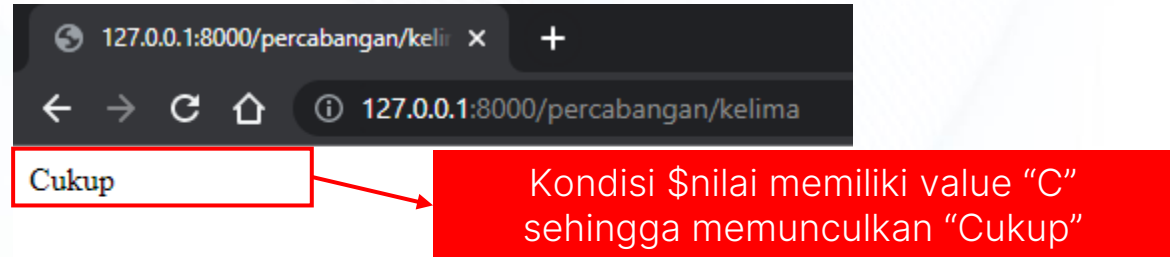
```
Route::get('/percabangan/kelima', 'App\Http\Controllers\PercabanganController@kelima');
```

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:

```
1  $makananFavorit = ["ayam", "pecel"];  
2  $makananFavorit = array("ayam", "pecel");  
3
```

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.

```
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

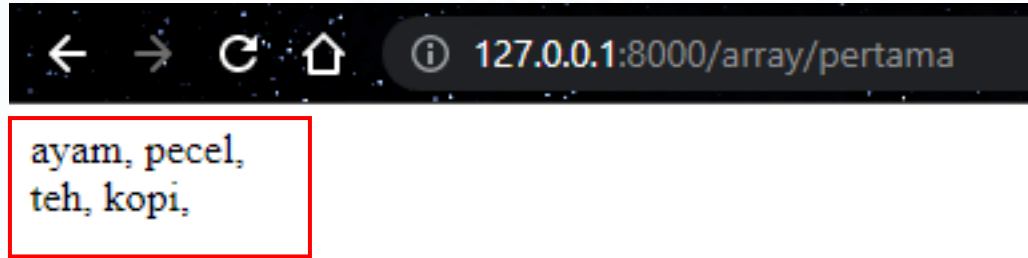
```
Route::get('/array/pertama', 'App\Http\Controllers\ArrayController@pertama');
```

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:

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

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

```
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:

```
1 $makananFavorit = ["Budi" =>"ayam", "Ayah" => "pecel"];  
2 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

```
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:

```
1 ▾ $dataStatus = [  
2     ["Budi", 12, "jomblo"],  
3     ["Joko", 24, "menikah"]  
4 ];  
5 echo $dataStatus[0][0]. " berumur ". $dataStatus[0][1]. " berstatus ". $dataStatus[0][2];  
6 // return Budi berumur 12 berstatus jomblo  
7 echo $dataStatus[1][0]. " berumur ". $dataStatus[1][1]. " berstatus ". $dataStatus[1][2];  
8 // return Joko berumur 24 berstatus menikah
```

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

```
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

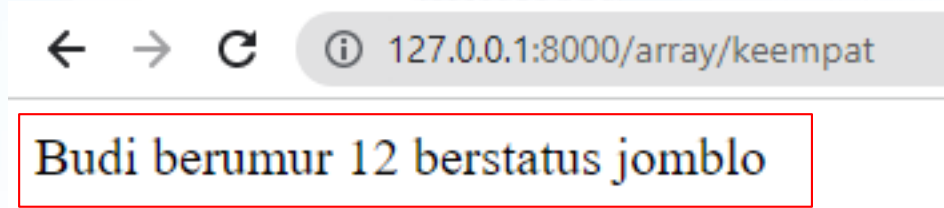
```
Route::get('/array/keempat', 'App\Http\Controllers\ArrayController@keempat');
```

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 :

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

- **str_replace()**

Fungsi untuk me-replace string pada sebuah string

Example :

```
1 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 :

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

- **strpos()**

Fungsi untuk mengambil posisi string pada sebuah string

Example :

```
1 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.

```
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

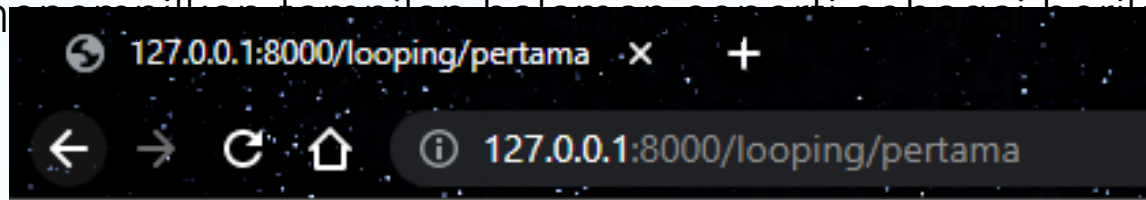
```
Route::get('/looping/pertama', 'App\Http\Controllers\LoopingController@pertama');
```

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



120

Perulangan while

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

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

```
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

```
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.

```
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

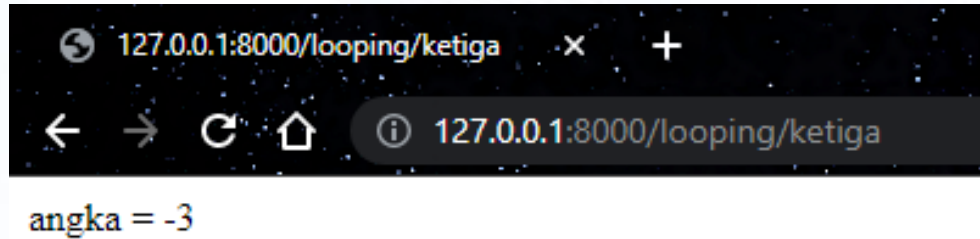
```
Route::get('/looping/ketiga', 'App\Http\Controllers\LoopingController@ketiga');
```

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.

```
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

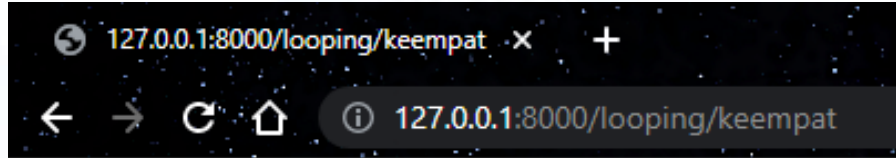
```
Route::get('/looping/keempat', 'App\Http\Controllers\LoopingController@keempat');
```

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.

```
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

```
Route::get('/function/pertama', 'App\Http\Controllers\Controller@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.



Halo Dunia. Ini angka 2

Parameter default

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

contoh:

```
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.

```
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

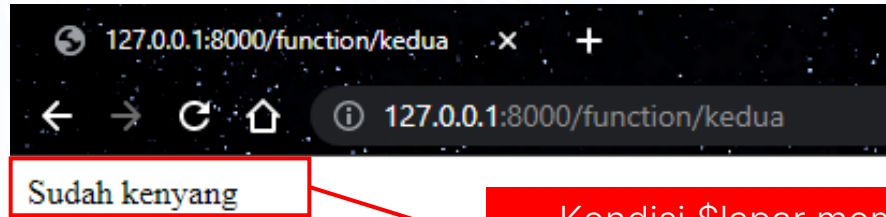
```
Route::get('/function/kedua', 'App\Http\Controllers\FunctionController@kedua');
```

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

Fungsi mengembalikan nilai

sebuah fungsi juga bisa mengembalikan nilai yang kita butuhkan

contoh:

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

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

```
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

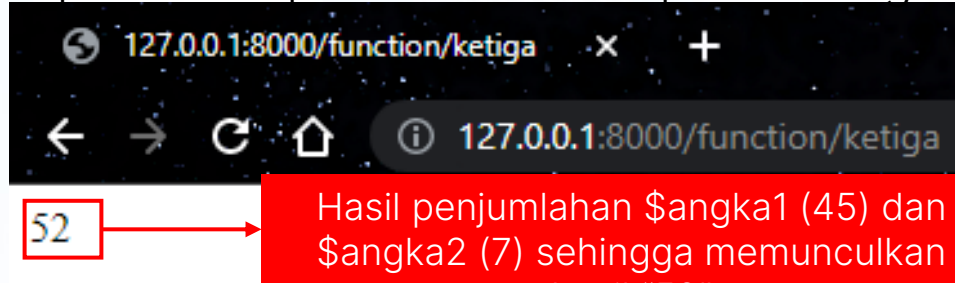
```
Route::get('/function/ketiga', 'App\Http\Controllers\FunctionController@ketiga');
```

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:

```
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;  
    }  
  
    return false;  
}
```

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

```
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.

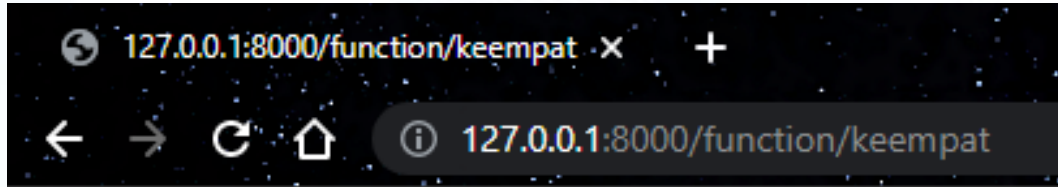
```
Route::get('/function/keempat', 'App\Http\Controllers\FunctionController@keempat');
```

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.



4.88888888888889

Hasil pembagian \$angka1 (44) dan \$angka2 (9) dengan syarat \$angka2 sehingga memunculkan hasil

Terima **Kasih**