

Sintaksis PHP

Skrip PHP dijalankan di server, dan hasil HTML biasa dikirim kembali ke browser.

Apa itu sintaksis? Sintaksis adalah aturan penulisan kode program. Pada dasarnya setiap bahasa pemrograman itu sama, yang membedakan adalah sintak dan fitur.

Struktur Program PHP yang Paling Dasar

Ini adalah bentuk paling dasar program PHP:

```
<?php  
  
echo "Hello World!";
```

Keterangan:

- **<?php** ini adalah pembuka program PHP. Pembuka ini wajib ada di setiap program PHP.
- **echo "Hello World!";** adalah sebuah statement atau perintah untuk menampilkan teks.

Tutup sebenarnya bersifat opsional. Tutup program dibutuhkan saat kita menggabungkan kode PHP dengan HTML.

Menulis Kode HTML dan PHP

Saat kita menulis kode PHP di dalam HTML, maka wajib hukumnya membuat tutup program.

Contoh:

```
<!DOCTYPE html>  
  
<html>  
  
    <head>  
  
        <title><?php echo "Belajar PHP" ?></title>  
  
    </head>  
  
    <body>  
  
        <?php
```

```
        echo "saya sedang belajar PHP<br>";

        echo "<p>Belajar PHP hingga jadi master</p>";

    ?>

</body>

</html>
```

Perhatikan contoh di atas! Apa yang akan terjadi bila kita menghapus tutup PHP (**?>**)? Tentunya program akan error. Yang ditulis di dalam HTML, filenya harus disimpan dengan ekstensi **.php** bukan **.html** meskipun isinya HTML dan PHP.

Lalu ada juga yang menulis seperti ini:

```
<?php

echo "<html>";

echo "<head>";

echo "<title>Judul Web</title>";

echo "</head>";

echo "<body>";

echo "<h1>Selamat datang</h1>";

echo "</body>";

echo "</html>";
```

Catatan:

PHP Statements diakhiri dengan tanda titik koma/semicolon (;).

Pada kasus program ini, anda boleh tidak memberikan tag penutup, karena kode HTML-nya ditulis di dalam sintak PHP.

Aturan Penulisan Case PHP

PHP adalah bahasa pemrograman yang bersifat *case sensitive*. Artinya, huruf besar (kapital) dan huruf kecil akan dibedakan.

Pada PHP, tidak ada keywords (if, else, while, echo dan lain-lain) dan user-defined function yang *case-sensitive*.

Contoh:

```
<!DOCTYPE html>

<html>

<body>


<?php

ECHO "Hello World!<br>";

echo "Hello World!<br>";

EcHo "Hello World!<br>";

?>

</body>

</html>
```

Dalam contoh di atas ketiga statement benar dan diperbolehkan.

Aturan Penulisan Variabel pada PHP

Semua nama variabel peka huruf besar kecil (**Case-Sensitive**).

Lihatlah contoh di bawah ini; hanya pernyataan pertama yang akan menampilkan nilai variabel \$color! Ini karena \$color, \$COLOR, dan \$coLOR diperlakukan sebagai tiga variabel berbeda:

```
<!DOCTYPE html>
<html>
<body>

<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>

</body>
```

```
</html>
```

Ini akan menyebabkan error. Untuk menghindari ini, kita harus konsisten dalam penamaan variabel dan konstanta. Anda dapat menggunakan nama variabel dengan huruf kecil saja dan konstanta dengan huruf kapital.

```
const INI_KONSTANTA = 123;  
$ini_variabel = 23;  
$iniJugaVariabel = 49;
```

Mengapa ada **\$iniJugaVariabel**?

Jika variabel terdiri dari dua atau lebih suku kata, maka kita bisa memisahkannya dengan huruf kapital atau bisa juga dengan garis bawah (underscore) ini dinamakan dengan gaya penulisan **Case camelCase**.

Kadang ada yang menulis variable/konstanta seperti *seperti_ini*, *SepertiItu*, dan *semacam-ini*. Untuk belajar lebih dalam tentang ini, Mari kita bahas gaya penulisan huruf-huruf pada kode program.

1. Gaya Snake Case



Gaya penulisan case yang ini cirinya menggunakan garis bawah atau *underscore* (`_`).

Suku kata dipisahkan dengan garis bawah sehingga membentuk sebuah ular.

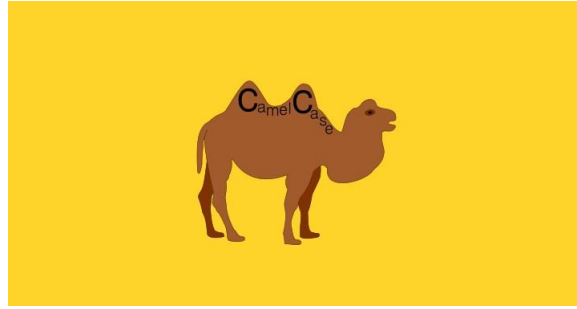
Karena itu, gaya ini disebut **Snake Case**.

Contohnya seperti ini:

```
The_quick_brown_fox_jumps_over_the_lazy_dog  
  
nama_variabel  
  
Nama_class
```

Gaya penulisan Snake Case biasanya kita jumpai pada pemrograman prosedural dan fungsional seperti C, Pascal, PHP, dsb.

2. Gaya Camel Case



Gaya penulisan Camel Case biasanya sering digunakan pada pemrograman berorientasikan objek atau OOP.

Cirinya, semua suku kata menyatu dan terdapat huruf kapital untuk memisahkannya.

Hal tersebut membuatnya terlihat seperti punggung unta. Karena itu, gaya ini disebut **Camel Case**.

Contoh:

```
TheQuickBrownFoxJumpsOverTheLazyDog  
  
iniPenulisanCamelCase();  
  
NamaClass  
  
namaVariabel  
  
iPhsatu  
  
eBay  
  
camelCase
```

3. Kebab Case



Gaya Kebab Case menggunakan tanda minus (-) untuk memisah suku kata. Sehingga membuatnya terlihat seperti kebab. Karena itu, gaya ini dinamakan **Kebab Case**.

Contohnya seperti ini:

```
ini-gaya-kebab-case
```

```
The-quick-brown-fox-jumps-over-the-lazy-dog  
  
btn-primary  
  
-moz-transition
```

Gaya penulisan Kebab Case biasanya kita jumpai pada CSS.

4. All caps

All Caps artinya semua kapital. Gaya penulisan ini sering digunakan untuk menuliskan nama sebuah konstanta.

Contoh:

```
PI  
DATABASE  
HOSTNAME  
URL_STRING
```

Penulisan Komentar di PHP

Komentar adalah bagian yang tidak akan dieksekusi oleh komputer. Biasanya digunakan untuk keterangan, penjelasan, dan dokumentasi kode program.

Komentar di PHP dapat ditulis dengan dua cara:

1. Menggunakan tanda `//` untuk komentar satu baris;
2. Menggunakan tanda `/*` untuk komentar lebih dari satu baris.

Contoh:

```
<?php  
  
    // ini adalah komentar  
  
    echo "Hello world";  
  
    /*  
  
    ini adalah komentar  
  
    yang lebih dari satu  
  
    baris
```

```
* /  
?>
```

Penulisan Blok Program

Blok program merupakan kumpulan dari statement dan ekspresi. Blok program di PHP dibungkus dengan kurung kurawal {.....}

Contoh:

```
<?php  
$umur=20;  
if ($umur > 18){  
    echo "Kamu sudah dewasa";  
    echo "Tapi masih saja malas";  
    echo "Jangan gitu lagi ya, ada orang yang  
        harus kamu banggakan";  
}  
?>
```

Program di atas adalah contoh blok kode *if* yang berisi tiga statement.

Menggabungkan html dan PHP

Anda dapat memisahkan file PHP dan HTML didalam sebuah file atau menggabungkan keduanya dengan kombinasi sebagai berikut:

- HTML didalam PHP
- PHP didalam HTML

Contoh:

Contoh dibawah ini adalah file yang hanya terdiri dari statement PHP.

```
<?php  
    echo "Namaku Jiemi Ardian";  
?>
```

Contoh dibawah ini adalah file yang terdiri dari statement PHP dan HTML pada file yang sama.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>PHP itu Mudah</title>
</head>
<body>
    <?php
    echo "<h1>Hallo </h1>";
    echo "Namaku <br>";
    echo "Jiemi Ardian";
    ?>
    Hallo Semuanya!!!
</body>
</html>
```

Komentar dan Variabel

Penjelasan tentang komentar dan variable telah dijelaskan pada awal praktikum ini. Berikut contoh penerapannya dalam program PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>PHP itu Mudah</title>
</head>
<body>
    <?php
    //ini syntax komentar untuk satu baris

    echo "<h1>Hallo </h1>";
    echo "Namaku <br>";

    /*
    ini syntax
    komentar untuk
    beberapa baris
    */
    echo "Jiemi Ardian";
    ?>
    Hallo Semuanya!!!
</body>
</html>
```


Tipe Data

Variabel dapat menyimpan data dari tipe yang berbeda, dan tipe data yang berbeda dapat melakukan hal yang berbeda.

PHP mendukung tipe data berikut:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

String PHP

String adalah urutan karakter, seperti "Halo dunia!".

Sebuah string dapat berupa teks apa pun di dalam tanda kutip. Anda dapat menggunakan tanda kutip tunggal atau ganda:

```
<?php
$x = "Hello world!";
$y = 'Hello world!';

echo $x;
echo "<br>";
echo $y;
?>
```

Berikut cara pengaksesan nilai dari variable dengan tipe data string.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>PHP itu Mudah</title>
</head>
<body>
    <?php
    //variabel pada php
    $nama = "belajar PABI";
    $nama2 = "sambil mendengar lagu";

    //akses nilai pada variabel
```

```
echo $nama . "<br>";

//cara menggabungkan nilai dari kedua variabel
echo $nama . " " . $nama2 . "<br>";
echo $nama ." ". " untuk " . $nama2;

?>
</body>
</html>
```

Integer PHP

Tipe data integer adalah angka non-desimal antara -2,147.483.648 dan 2.147.483.647.

Aturan untuk bilangan bulat:

- Integer harus memiliki setidaknya satu digit
- Bilangan bulat tidak boleh memiliki titik desimal
- Integer dapat berupa positif atau negatif
- Bilangan bulat dapat ditentukan dalam: desimal (basis 10), heksadesimal (basis 16), oktal (basis 8), atau notasi biner (basis 2)
- Dalam contoh berikut \$x adalah bilangan bulat. Fungsi PHP **var_dump()** mengembalikan tipe data dan nilai:

Contoh:

```
<?php
$x = 5985;
var_dump($x);
?>
```

PHP Float

Float (angka titik mengambang) adalah angka dengan titik desimal atau angka dalam bentuk ekspsatunsial.

Dalam contoh berikut \$x adalah float. Fungsi PHP **var_dump()** mengembalikan tipe data dan nilai.

Contoh:

```
<?php
$x = 10.365;
var_dump($x);
?>
```

PHP Boolean

Tipe boolean adalah tipe data pada php yang paling sederhana dalam bahasa pemrograman apapun. karena tipe data ini hanya memiliki dua nilai yaitu true dan false. tipe data boolean sering kali digunakan pada operasi logika seperti kondisi if dan looping. dan berikut adalah tipe data boolean jika dikonversi dari tipe data lainnya:

```
<?php
$x = false;
$y = true;
//variabel di atas bertipe data boolean karena berisi nilai
benar atau salah.
?>
```

Array PHP

Array menyimpan banyak nilai dalam satu variabel tunggal.

Dalam contoh berikut, **\$showRoom** adalah sebuah array. Fungsi PHP `var_dump()` mengembalikan tipe data dan nilai:

```
<?php
$showRoom = array("Volvo", "BMW", "Toyota");
var_dump($showRoom);
?>
```

Objek PHP

Objek adalah tipe data yang menyimpan data dan informasi tentang cara memproses data itu.

Dalam PHP, suatu objek harus dinyatakan secara eksplisit.

Pertama kita harus mendeklarasikan kelas objek. Untuk ini, kita menggunakan kata kunci/keywords **Class**. Class adalah struktur yang dapat berisi properti dan metode:

```
//class manusia
class manusia{
    //property
    var $nama;
    var $warna;

    //method manusia
    function tampilkan_nama(){
        return "Nama saya Rivaldo Tampan <br/>";
    }

    function warna_kulit(){
        return "Warna kulit saya hitam <br/>";
    }
}
```

```
//instansiasi class manusia
$manusia = new manusia();

//memanggil method tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();

//memanggil method warna_kulit dari class manusia
echo $manusia->warna_kulit();
```

Nilai PHP NULL

Null adalah tipe data khusus yang hanya dapat memiliki satu nilai: NULL.

Variabel tipe data NULL adalah variabel yang tidak memiliki nilai yang ditetapkan padanya.

Tip: Jika suatu variabel dibuat tanpa nilai, variabel itu secara otomatis akan diberi nilai NULL. Variabel juga dapat dikosongkan dengan menetapkan nilai ke NULL:

```
<?php
$x = "Hello world!";
$x = null;
var_dump($x);
?>
```

Resource PHP

Jenis Sumber daya/Resource bukan tipe data aktual. Ini adalah penyimpanan referensi ke fungsi dan sumber daya eksternal ke PHP.

Contoh umum menggunakan tipe data sumber daya adalah panggilan basis data (database).

Operator Matematika

Adalah operator/fungsi operasi yang terdiri dari:

- +
- -
- *
- /
- %
- ++
- --

Referensi: <https://www.php.net/manual/en/language.operators.php>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>PHP itu Mudah</title>
</head>
    <body>

        <?php
            //variabel pada php
            $nama = "belajar PABI";
            $nama2 = "sambil mendengar lagu";

            $num1 = "1000";
            $num2 = "900";

            //akses nilai pada variabel
            echo "Selamat datang di $nama <br>";

            echo $num1 . $num2 . "<br>";

            //variabel dengan tipe data integer
            $angka = 100;

            //variabel dengan tipe float
            $angka2 = 20.50;

            $angka3 = 5;
            $angka4 = 8;

            /*
            pemberian nilai dari sebuah variabel dapat dioperasikan dengan
            nilai sebelumnya dioperasikan dengan nilai variabel lain sesuai
            hasil/output dengan urutan operasi matematika ketika SD
            */

            $hasil = $angka + $angka2 * $angka3 / $angka4 - $angka3;

            // Penulisan $angka = $angka + $angka3; dapat disingkat

            $angka *= $angka3;

            echo $hasil . "<br>";
            echo $angka . "<br>";

            //operator increment
            $angka++;
            echo $angka . "<br>";

            //operator decrement
            $angka--;
            echo $angka . "<br>";

        ?>
    </body>
</html>

```

Built in Method Matematika

Referensi: <https://www.php.net/manual/en/ref.math.php>

Contoh kompleks Method matematika pada PHP.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>PHP itu Mudah</title>
</head>
<body>

    <?php
        //variabel pada php
        $nama = "Belajar PABI";

        //akses nilai pada variabel
        echo "Selamat datang di $nama <br>";

        $angka = 2.44;
        $angka2 = 2.62;

        $num = rand(0,100);

        //round (minimum_value, max_value);
        $num2 = round($angka);
        $num3 = round($angka2);

        echo $num . "<br>";
        echo $num2 . "<br>";
        echo $num3 . "<br>";

        $maksimum = max ($num, $num2, $num3);
        echo "Angka terbesar adalah dari sampel diatas
adalah angka: " . $maksimum;
    ?>

</body>
</html>
```

Method String

Contoh kompleks method string pada PHP.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar PHP</title>
</head>
<body>

    <?php

    //variabel pada php
    $nama = "Belajar PABI";

    //akses nilai pada variabel
    echo "Selamat datang di $nama <br>";

    $teks = "hai semuanya yang ada disini";
    echo strlen($teks) . "<br>";
    echo str_word_count($teks) . "<br>";
    echo str_replace ("hai", "hallo", $teks) . "<br>";
    echo str_repeat("Hai! <br> ", 10);
    echo str_repeat (" $teks <br>", 10);
    echo str_repeat(str_replace("hai", "hallo", $teks .
"<br>"), 4);
    ?>
</body>
</html>

```

Array

Contoh kompleks program array pada PHP.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar PHP</title>
</head>
<body>
    <?php
    //variabel pada php
    $nama = "Belajar PABI";

    //akses nilai pada variabel
    echo "Selamat datang di $nama <br>";

    // ----- Tipe data Array -----
    $sifat = array('baik semangat ya', 'perhatian', 'cuek',
'lemah lembut');
    $nama = ['Jiemi Ardian Tampan', 'Bien', 'anak baik'];

```

```

    print_r ($nama);
    echo "<br>";

    print_r ($nama[2]);
    echo "<br>";

    echo "Hai" . " " . $nama[1] . " " . $sifat[0];

    ?>
</body>
</html>

```

Metode Array

Contoh kompleks program metode array pada PHP.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar PHP</title>
</head>
<body>
    <?php
        //variabel pada php
        $nama = "Belajar PABI";

        //akses nilai pada variabel
        echo "Selamat datang di $nama <br>";

        // ----- Metode Array -----
        $kandang = array('anjing', 'kura-kura', 'koala',
'Anjing');
        $kandang2 = array('anjing', 'koala', 'cacing',
'banteng','semut');
        $angka = [5, 10, 3, 12, 7,21];

        print_r(array_unique($kandang));
        echo "<br>";

        print_r(array_reverse($kandang2));
        echo "<br>";

        shuffle($kandang2);
        print_r ($kandang2);
        echo "<br>";

        echo count($kandang);
        echo "<br>";
    >

```



```
sort($kandang2);  
print_r ($kandang2);  
echo "<br>";  
  
sort($angka);  
print_r ($angka);  
echo "<br>"  
?>  
</body>  
</html>
```

Associative Array

Di PHP, ada tiga jenis array:

- Indexed arrays/Array yang diindeks - Array dengan indeks numerik
- Associative arrays/Array asosiatif - Array dengan kunci bernama
- Multidimensional arrays/Array multidimensi - Array yang mengandung satu atau lebih array

Array asosiatif digunakan untuk menyimpan pasangan nilai kunci. Misalnya, untuk menyimpan tanda subjek yang berbeda dari siswa dalam array, array yang diindeks secara numerik tidak akan menjadi pilihan terbaik. Sebagai gantinya, kami dapat menggunakan nama subjek masing-masing sebagai kunci dalam array asosiatif kami, dan nilainya akan menjadi nilai masing-masing yang diperoleh.

Fungsi `array()` digunakan untuk membuat array asosiatif.

Sintaks

```
array(  
    key  => value,  
    key2 => value2,  
    key3 => value3,  
    ....  
)
```

Contoh:

```
<?php  
/* cara 1 membuat associate array. */  
$mahasiswa_satu = array("Matematika"=>95,  
                        "Fisika"=>90,  
                        "PABI"=>96,  
                        "English"=>93,  
                        "Sisop"=>98  
);
```

```

/* cara kedua membuat associative array. */
$mahasiswa_dua["Matematika"] = 95;
$mahasiswa_dua["Fisika"] = 90;
$mahasiswa_dua["PABI"] = 96;
$mahasiswa_dua["English"] = 93;
$mahasiswa_dua["Sisop"] = 98;

/* akses elemen array langsung */
echo "Marks for mahasiswa satu is:\n";
echo "Matematika:" . $mahasiswa_dua["Matematika"], "\n";
echo "Fisika:" . $mahasiswa_dua["Fisika"], "\n";
echo "PABI:" . $mahasiswa_dua["PABI"], "\n";
echo "English:" . $mahasiswa_satu["English"], "\n";
echo "Sisop:" . $mahasiswa_satu["Sisop"], "\n";
?>

```

Metode Associative Array:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar PHP</title>
</head>
<body>
    <?php
    //variabel pada php
    $nama = "Belajar PABI";

    //akses nilai pada variabel
    echo "Selamat datang di $nama <br>";

    // ----- Metode Associative Array -----

    $data = array ( "nama" => "Jiemi Ardian",
                    "umur" => 21,
                    "Pekerjaan" => "Pedagang ginjal"
                );

    $data2 = array( "isteri" => "belum ada",
                    "laptop" => "Acer Nitro 5"
                );

    print_r(array_values($data));
    echo "<br>";

    print_r(array_keys($data));

```

```
        echo "<br>";

        print_r(array_merge($data, $data2));
        echo "<br>";

        ?>
    </body>
</html>
```

Multi Dimensional Array

Multidimensional arrays/Array multidimensi - Array yang mengandung satu atau lebih array

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar PHP</title>
</head>
<body>
    <?php
        //variabel pada php
        $nama = "Multi Dimensi Array";

        //akses nilai pada variabel
        echo "Selamat datang di $nama <br>";

        // ----- Multi Dimensi Array -----

        $data = array(
            array("Programmer", 21, "ngopi"),
            array("kuli bangunan", 48, "berenang"),
            array("Mahasiswa", 20, "tidur siang")
        );

        //menampilkan nilai dari seluruh array
        print_r($data);
        echo "<br>";

        //menampilkan nilai dari index array
        echo $data[1][1];
        echo "<br>";

        //update nilai dari index array
        $data[0][0] = "Pemain bola";
        echo $data[0][0];
```

```
        echo "<br>";
    ?>
</body>
</html>
```

Loop PHP

Seringkali ketika Anda menulis kode, Anda ingin blok kode yang sama terus berulang beberapa kali. Jadi, jika anda ingin menambahkan beberapa baris kode yang hampir sama dalam sebuah skrip, kita dapat menggunakan loop.

Loop digunakan untuk mengeksekusi blok kode yang sama berulang kali, selama kondisi tertentu **benar**.

PHP memiliki jenis loop berikut:

- while - loop melalui blok kode selama kondisi yang ditentukan benar
- do ... while - loop melalui blok kode sekali, dan kemudian mengulangi loop selama kondisi yang ditentukan benar
- for - loop melalui blok kode beberapa kali
- foreach - loop melalui blok kode untuk setiap elemen dalam array

For Loop

for loop digunakan ketika Anda tahu sebelumnya berapa kali skrip harus dijalankan.

Sintaks

```
for(init counter; test counter; increment counter) {
    kode yang akan dieksekusi untuk setiap iterasi;
}
```

Parameter:

- init counter: Menginisialisasi nilai penghitung loop
- test counter: Dievaluasi untuk setiap iterasi loop. Jika bernilai TRUE, loop berlanjut. Jika mengevaluasi ke FALSE, loop berakhir.
- Increment/decrement counter: Meningkatkan/menurunkan nilai loop counter

Penggunaan Loop Untuk pengaksesan elemen Array

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

```

        <title>Belajar PHP</title>
</head>
<body>
    <?php
    //variabel pada php
    $nama = "Belajar PABI";

    //akses nilai pada variabel
    echo "Selamat datang di $nama <br>";

    // ----- Akses Elemen Array dengan Loop ----
    $sayang = ['sayang ibu', 'sayang ayah', 'sayang
    kaka', 'sayang adik'];

    for($i=0;$i<5;$i++)
    {
        echo "Aku anak baik <br>";
    }

    for($i=0;$i<count($sayang); $i++)
    {
        echo $sayang[$i] . "<br>";
    }

    ?>
</body>

```

foreach Loop

Foreach loop hanya bekerja pada array, dan digunakan untuk loop melalui setiap pasangan kunci / nilai dalam array.

Sintaks

```

foreach ($ array sebagai $ value) {
    kode yang akan dieksekusi;
}

```

Untuk setiap iterasi loop, nilai elemen array saat ini ditetapkan ke \$value dan pointer array digerakkan oleh satu, sampai mencapai elemen array terakhir.

Contoh berikut akan menampilkan nilai array yang diberikan (**\$colors**):

```

<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>

```

Contoh berikut akan menampilkan keys dan values array yang diberikan (**\$age**):

```
<?php
$age = array("Jakob"=>"25", "Bien"=>"27", "Putri"=>"13");

foreach($age as $x => $val) {
    echo "$x = $val<br>";
}
?>
```

Contoh kompleks penggunaan Foreach Untuk Pengaksesan Elemen Array

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar PHP</title>
</head>
<body>
    <?php
    //variabel pada php
    $nama = "Multi Dimensi Array";

    //akses nilai pada variabel
    echo "Selamat datang di $nama <br>";

    // ----- Single Dimensi Array -----
    $mahasiswa = ['Thalia','Daniel', 'Rico', 'Cindy',
    'Barney'];

    //print keseluruhan nilai dari array
    foreach ($mahasiswa as $x)
    {
        echo $x . "<br>";
    }

    ?>
</body>
```

Contoh kompleks penggunaan Foreach Untuk Pengaksesan Elemen Associative Array

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar PHP</title>
```

```
</head>
<body>

    <?php
    //variabel pada php
    $nama = "Multi Dimensi Array";

    //akses nilai pada variabel
    echo "Selamat datang di $nama <br>";

    // ----- Multi Dimensi Array -----
    $data = ["nama" => "Aku anak
            baik", "umur" => 18,
            "sifat" => "Semangat belajar, supaya boleh
            membanggakan orang yang ku sayangi"
    ];

    //print keseluruhan nilai dari array
    foreach ($data as $key => $value)
    {
        echo $key . " : ". $value . "<br>";
    }

    ?>
</body>
```

"Programming isn't about what you know; it's about what you can figure out."

Chris Pine