

## Modul 2

# WEB APPLICATION DEVELOPMENT 2023



## Dasar PHP

# Practicum Assistant



**Abdi Fatih**

PATH



**Febrilia Putri Inzani**

YAYA



**Adhwaa Hilmiyazhar**

HILM



**Ikram Zaidan Wicaksono**

ICAM



**Aebil Taskari Rusydi**

ASKA



**Muhammad Adam Nugraha**

ANUG



**Afif Zaky Muhana**

APIP



**M.Hafidz Alkhair Pasaribu**

MHAP



**Ario Kusuma Purboyo**

VITE



**Muhammad Kautsar Firdaus**

KATZ



**Ataya Najla**

TAYA



**Muhammad Nurul Afif Maliki**

MNAM



**Bayu Satrio Trilaksono**

BYST



**Najma Syarifa Rahmah**

NAFR



**Donna Lolita Harahap**

DONE



**Putu Wisnu Wirayuda Putra**

PUTU



**Fadli Izurohman**

BANA



**Rafid Fadhil**

DHIL



**Fanny Irawan**

AONE



**Raif Fawwazdzaky**

RAEP

## Daftar Isi

|   |           |
|---|-----------|
| <b>Daftar Isi</b>   | <b>3</b>  |
| <b>Definisi</b>   | <b>6</b>  |
| <b>Sejarah Singkat</b>  | <b>6</b>  |
| <b>Fungsi PHP</b>   | <b>7</b>  |
| <b>Tipe Data atau Variabel</b>                                  | <b>7</b>  |
| Cara penulisan variabel sesuai kaidah PHP                       | 7         |
| a. String   | 8         |
| b. Integer  | 9         |
| c. Float(Double)  | 9         |
| d. Boolean  | 9         |
| e. Array  | 10        |
| f. Null   | 10        |
| <b>Operator PHP</b>   | <b>11</b> |
| a. Operator Aritmatika  | 11        |
| Contoh Penggunaan dengan Case Kalkulator Sederhana              | 12        |
| b. Operator Penugasan   | 13        |
| c. Operator Perbandingan  | 13        |
| d. Operator Perbandingan Equal dan Identik                      | 14        |
| e. Operator logika  | 14        |
| f. Operator Increment & Decrement                               | 15        |
| <b>Logika Percabangan PHP</b>                                   | <b>15</b> |
| • Percabangan If - Else   | 15        |
| a. Percabangan If   | 15        |
| b. Percabangan If - Else  | 16        |
| c. Percabangan If - Else if -Else                               | 16        |
| • Percabangan Switch - Case                                     | 17        |
| <b>Perulangan (Looping)</b>                                     | <b>18</b> |
| • Perulangan for  | 18        |
| • Perulangan while  | 20        |
| • Perulangan do-while   | 20        |
| • Perulangan foreach  | 21        |
| <b>Array</b>  | <b>22</b> |
| Apa itu Array?  | 22        |
| Solusinya adalah dengan membuat array!                          | 22        |
| Di PHP, ada tiga jenis array:                                   | 23        |
| 1. Array terindeks - Array dengan indeks numerik                | 23        |
| 2. Array asosiatif - Array dengan kunci bernama                 | 23        |
| 3. Array multidimensi - Array yang berisi satu atau lebih array | 24        |

|  |           |
|--|-----------|
| <b>Method GET dan POST</b>   | <b>25</b> |
| 1. Method GET  | 25        |
| 2. Method POST   | 27        |
| Jadi Kapan Harus kita harus menggunakan Method GET dan Kapan kita harus menggunakan Method POST? | 29        |
| Gunakan Method GET ketika:   | 29        |
| Gunakan Method POST ketika:  | 30        |
| <b>Sumber Referensi</b>  | <b>31</b> |



## Definisi

PHP ialah suatu bahasa pemrograman berbasiskan kode-kode (script) yang umum dipakai untuk mengolah suatu data server dan pembuatan website dan juga PHP ini bersifat open source. Dilansir dari PHP.net, PHP merupakan singkatan dari PHP: Hypertext Preprocessor.

Menurut situs tersebut, PHP adalah bahasa pemrograman yang banyak digunakan secara luas dan secara khusus sesuai untuk pengembangan web. Sebelum dinamakan demikian, arti PHP adalah Personal Home Page Tools karena dipakai membangun web pribadi. Namun, perkembangan yang cukup pesat membuatnya bertumbuh menjadi bahasa pemrograman web yang kuat.

PHP adalah bahasa pemrograman yang berjalan disisi server, sehingga script PHP akan dijalankan di server. Beberapa server yang sering digunakan bersama dengan PHP mencakup Apache, Nginx, dan LiteSpeed.

Selain itu, PHP juga merupakan bahasa pemrograman yang bersifat open source. Pengguna memiliki kebebasan untuk mengubah dan mengembangkannya sesuai dengan kebutuhannya.

## Sejarah Singkat

Awalnya diciptakan oleh Rasmus Lerdorf pada tahun 1994, PHP dimulai sebagai kumpulan script sederhana untuk mengelola data server dan membangun situs web pribadi. Dari nama Personal Home Page Tools, PHP mengalami perubahan nama singkat menjadi FI atau Forms Interpreter.

Seiring berjalannya waktu, PHP mengalami berbagai perbaikan dan perkembangan. Versi pertama resmi yaitu PHP/FI 2.0 dirilis pada tahun 1997, memungkinkan pengembang untuk menambahkan kode mereka sendiri. Pada tahun 1998, PHP v3 menjadi tonggak penting dengan pengenalan kemampuan pemrograman berorientasi objek. PHP v4, yang dirilis pada tahun 2000, membawa peningkatan signifikan dalam hal kinerja dan fitur, sementara PHP v5 pada tahun 2004 membawa dukungan objek yang lebih kuat.

Terakhir, PHP v7, yang dirilis pada tahun 2015, menandai evolusi terbaru dengan peningkatan kinerja yang luar biasa. Hari ini, PHP digunakan secara luas oleh komunitas pengembang web di seluruh dunia dan bersifat open source, memberikan kebebasan kepada pengguna untuk menyesuaikan

dan mengembangkan sesuai dengan kebutuhan mereka. PHP menjalankan skrip di sisi server, berinteraksi dengan server seperti Apache, Nginx, atau LiteSpeed, dan memungkinkan pengembangan berbagai jenis aplikasi web yang kuat.

## Fungsi PHP

Seperti disebut sebelumnya, PHP adalah bahasa pemrograman yang ada di sisi server. Bahasa ini digunakan untuk mengelola konten dinamis, database, session tracking, dan bahkan keseluruhan situs e-commerce. Sebetulnya, anda bisa membuat web dengan memakai HTML saja. Namun, hasilnya terasa kurang menarik lantaran HTML membuat tampilan web statis dengan konten dan halaman yang bersifat permanen. Ketika suatu web dibangun dengan PHP, tampilan web justru lebih dinamis. Anda bisa mengatur tampilan konten sesuai situasi atau momen tertentu tanpa mengubah dasar pemrograman web tersebut. Web yang dinamis ini juga berfungsi sebagai penyimpanan data dalam database, membuat tampilan halaman web berubah sesuai input dari pengguna, menampilkan gambar, hingga pemakaian form.

Dalam pembuatan web, kode PHP umumnya disisipkan dalam dokumen HTML yang sudah tersedia. Itulah mengapa PHP dikenal sebagai scripting language atau bahasa pemrograman script. Penyisipan sebuah variabel PHP bisa menghasilkan tampilan berbeda pada web, tanpa perlu menambah banyak kode baris.

## Tipe Data atau Variabel

Variabel atau Tipe Data berperan sebagai wadah untuk menampung data atau nilai. Dengan variabel, kita dapat menyimpan beragam jenis data, seperti teks untuk nama mahasiswa, angka desimal untuk nilai IPK mahasiswa, dan banyak jenis data lainnya sesuai kebutuhan.

### Cara penulisan variabel sesuai kaidah PHP

| Peraturan                 | Penjelasan  |
|---------------------------|---|
| Diawali tanda dollar "\$" | Setiap variabel dalam bahasa pemrograman PHP diidentifikasi dengan tanda \$ diikuti oleh nama variabel yang sesuai. |

|  |   |
|--|---|
| Case Sensitive   | Variabel PHP memperhatikan perbedaan antara huruf besar dan kecil dalam penamaan variabel, sehingga variabel \$vateh, \$Vateh, dan \$vatEH dianggap sebagai tiga variabel yang berbeda.   |
| Diawali huruf atau underscore “_”.<br>Tidak boleh angka. | Variabel PHP harus dimulai dengan huruf atau tanda underscore “_”. Variabel dengan nama \$_vateh atau \$vatehpinjam100 dapat dibuat, tetapi tidak diperbolehkan menggunakan \$100dongvateh sebagai nama variabel.   |
| Inisiasi dengan tanda “=”                                | Kita bisa memberi nilai terhadap suatu variabel dengan menggunakan operator sama dengan (=).<br><br>Contoh: \$nama = "Nurul Huda";  |
| Hanya boleh huruf dan angka                              | Nama variabel PHP hanya boleh tersusun dari huruf [a-z] atau [A-Z], dan juga angka [0-9]. Kita tidak bisa memberi nama variabel misalkan dengan tanda ^ atau & dan lain sebagainya. Anda juga tidak bisa menggunakan spasi dalam penamaan variabel di PHP |

#### a. String

String adalah urutan karakter. Yang terletak di antara tanda kutip dua yaitu “ ”, seperti "Hello world!", "Vateh", "Pinjam", "100".

```
<?php
$variabel1 = "Vateh", "Pinjam", "100";
$variabel2 = "Hello World!";

echo $variabel1;
echo $variabel2;
}
```

#### b. Integer

Jenis data integer adalah bilangan non-desimal antara - 2, 147, 483, 648 sampai 2, 147.483, 647.

```
<?php
$variabelString = "Vateh", "Pinjam"; ##String
$variabelInteger = 100; ##Integer

echo $variabel1 + $variabelInteger; #Output Vateh Pinjam 100
}
```

#### c. Float(Double)

Float atau Double adalah bilangan dengan titik desimal atau bilangan dalam bentuk eksponensial.

```
<?php
$variabelString = "Vateh", "Pinjam"; ##String
$variabelFloat = 3.14; ##Float atau Double

echo $variabel1 + $variabelFloat; #Output Vateh Pinjam 3.14
}
```

#### d. Boolean

Boolean mewakili dua keadaan yang mungkin: TRUE atau FALSE.



```
<?php

$apakahVatehBolehPinjam = true;
$apakahVatehSudahMeminjam = false;

var_dump($apakahVatehBolehPinjam);
echo "<br>";
var_dump($apakahVatehSudahMeminjam);

##output
bool(true);
bool(false);
```

#### e. Array

Sebuah array menyimpan beberapa nilai dalam satu variabel.

```
<?php

$listPinjam100 = ["Vateh Simonangkir", "Vateh El Classico", "Rafathar Vateh"];
```

#### f. Null

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

```
<?php
// x pertama kali diberi nilai 2
$x = 2;
if($x > 0){
    echo "Bisa dibandingkan karena tidak null";
}

// x diberi NULL
$x = null;
if($x > 0){
    echo "Tidak bisa dibandingkan karena null";
}

##Outputnya Bisa dibandingkan karena tidak null
```

## Operator PHP

Operator digunakan untuk melakukan operasi pada variabel dan nilai.

### a. Operator Aritmatika

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika.

Operator aritmatika terdiri dari:

| Operator | Nama                          | Contoh       | Keterangan                             |
|----------|-------------------------------|--------------|--|
| +        | Addition /<br>Tambahan        | $\$x + \$y$  | Jumlah $\$x$ dan $\$y$                 |
| -        | Subtraction /<br>Pengurangan  | $\$x - \$y$  | Selisih $\$x$ dan $\$y$                |
| *        | Multiplication /<br>Perkalian | $\$x * \$y$  | Produk dari $\$x$ dan $\$y$            |
| /        | Division /<br>Pembagian       | $\$x / \$y$  | Hasil bagi $\$x$ dan $\$y$             |
| %        | Modulus                       | $\$x \% \$y$ | Sisa $\$x$ dibagi $\$y$                |
| **       | Exponential                   | $\$x ** \$y$ | Hasil menaikkan $\$x$ ke pangkat $\$y$ |

## Contoh Penggunaan dengan Case Kalkulator Sederhana

```
<!DOCTYPE html>
<html>
<head>
  <title>Kalkulator Sederhana</title>
</head>
<body>
  <h2>Kalkulator Sederhana</h2>
  <form method="POST" action="">
    Angka pertama: <input type="text" name="angka1"><br>
    Angka kedua: <input type="text" name="angka2"><br>
    <input type="submit" name="tambah" value="Tambah">
  </form>

  <?php
  if (isset($_POST['tambah'])) {
    // Mengambil nilai dari inputan pengguna
    $angka1 = $_POST['angka1'];
    $angka2 = $_POST['angka2'];

    // Melakukan perhitungan penambahan
    $hasil = $angka1 + $angka2;

    // Menampilkan hasilnya
    echo "Hasil penambahan: $hasil";
  }
  ?>
</body>
</html>
```

Output dari code diatas:



← → ↻ 🌐 localhost/kalkulator.php

### Kalkulator Sederhana

Angka pertama:

Angka kedua:

Hasil penambahan: 3

### b. Operator Penugasan

Operator ini adalah operator untuk memberikan tugas kepada variabel. Biasanya digunakan untuk mengisi nilai.

| Operator | Nama            | Contoh   | Keterangan   |
|----------|-----------------|--|--|
| =        | Pengisian Nilai | <pre><code>\$speed = 83;</code></pre><br><pre><code>\$nilai = \$speed + 10;</code></pre> | Menggunakan variabel baru untuk menentukan value tertentu. maka nilai <code>\$nilai</code> akan sama dengan $83 + 10 = 93$ |

### c. Operator Perbandingan

Operator Perbandingan adalah operator untuk membandingkan dua buah nilai. Hasil operasi dari operator relasi akan menghasilkan nilai dengan tipe data boolean, yaitu **true** (benar) dan **false** (salah).

| Operator | Nama                    | Contoh                      | Keterangan  |
|----------|-------------------------|-----------------------------|---|
| >        | Lebih besar             | <code>\$a &gt; \$b;</code>  | <code>\$a</code> lebih besar dari <code>\$b</code>        |
| <        | Lebih kecil             | <code>\$a &lt; \$b;</code>  | <code>\$a</code> lebih kecil dari <code>\$b</code>        |
| >=       | Lebih besar sama dengan | <code>\$a &gt;= \$b;</code> | <code>\$a</code> lebih besar sama dengan <code>\$b</code> |
| <=       | Lebih kecil sama dengan | <code>\$a &lt;= \$b;</code> | <code>\$a</code> lebih kecil sama dengan <code>\$b</code> |

### d. Operator Perbandingan Equal dan Identik

| Operator | Nama                       | Contoh                    | Keterangan  |
|----------|----------------------------|---------------------------|---|
| ==       | Sama dengan secara equal   | <code>\$a == \$b;</code>  | <code>\$a</code> Sama dengan secara equal dengan <code>\$b</code> .   |
| ===      | Sama dengan secara identik | <code>\$a === \$b;</code> | <code>\$a</code> Sama dengan secara identik dengan <code>\$b</code> . |
| !=       | Tidak sama                 | <code>\$a != \$b;</code>  | <code>\$a</code> Tidak sama   |

|                 |                                  |                          |  |
|-----------------|----------------------------------|--------------------------|--|
|                 | dengan secara equal              |                          | dengan secara equal dengan \$b.                  |
| <code>!=</code> | Tidak sama dengan secara identik | <code>\$a != \$b;</code> | \$a Tidak sama dengan secara identik dengan \$b. |

**e. Operator logika**

Operator logika adalah operator untuk melakukan operasi logika seperti **AND**, **OR**, dan **NOT**.

| Operator                | Nama                 | Contoh  | Keterangan   |
|-------------------------|----------------------|---|--|
| <code>&amp;&amp;</code> | AND                  | <pre>\$x = true; \$y = false;  \$result = \$x &amp;&amp; \$y;</pre> | \$result akan menjadi false karena satu kondisi adalah false     |
| <code>  </code>         | OR                   | <pre>\$x = true; \$y = false;  \$result = \$x    \$y;</pre>         | \$result akan menjadi true karena salah satu kondisi adalah true |
| <code>!</code>          | Negasi/kebalikan/NOT | <pre>\$x = true;  \$result = !\$x;</pre>                            | \$result akan menjadi false karena nilai awalnya adalah true     |

**f. Operator Increment & Decrement**

Operator increment dan decrement merupakan operator yang digunakan untuk menambah **+1 (tambah satu)** dan mengurangi **-1 (kurangi dengan satu)**.

| Operator        | Nama      | Contoh                     | Keterangan                   |
|-----------------|-----------|----------------------------|------------------------------|
| <code>++</code> | Increment | <pre>\$a = 5; \$a++;</pre> | Nilai \$a sekarang menjadi 6 |
| <code>--</code> | Decrement | <pre>\$b = 5; \$b--;</pre> | Nilai \$b sekarang menjadi 4 |



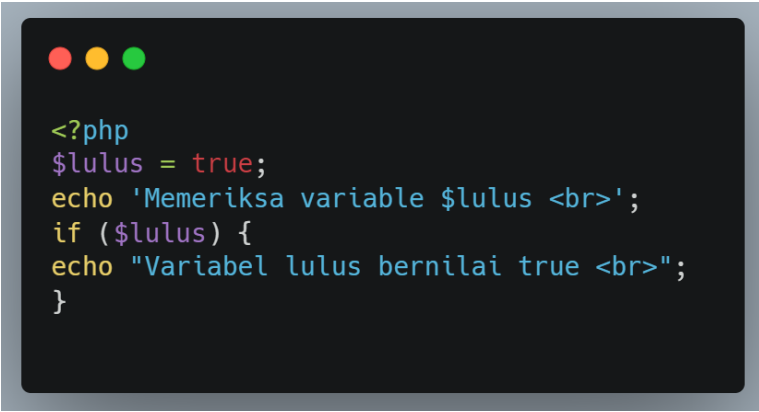
## Logika Percabangan PHP

- **Percabangan If - Else**

- a. **Percabangan If**

Percabangan if adalah percabangan yang paling dasar. Tugasnya adalah memeriksa nilai boolean atau sebuah ekspresi logika.


Jika suatu variabel atau suatu ekspresi logika bernilai true, maka proses yang ada di dalam blok kode if akan dijalankan. Jika tidak, maka perintah/kode yang ada di dalam blok if tidak akan dijalankan.



```
<?php
$lulus = true;
echo 'Memeriksa variable $lulus <br>';
if ($lulus) {
    echo "Variabel lulus bernilai true <br>";
}
```

- b. **Percabangan If - Else**

Bagaimana jika ternyata kondisi yang didefinisikan di dalam if ternyata tidak terpenuhi alias bernilai false? Kita bisa menangani hal tersebut dengan membuat blok kode else. Kita ubah Kode 1 di atas menjadi seperti berikut:



```
<?php
$nilai = 60;
echo 'Memeriksa variable $nilai <br>';
echo "Nilai: {$nilai} <br>";
if ($nilai >= 70) {
    echo "Selamat, siswa !";
} else {
    echo "Mohon maaf, siswa tidak lulus";
}

echo "<br>";
```

### c. Pecabangan If - Else if -Else

Pecabangan if-else if-else pada bahasa pemrograman PHP adalah salah satu struktur kontrol yang digunakan untuk mengambil keputusan berdasarkan kondisi yang berbeda. Struktur ini memungkinkan untuk mengevaluasi beberapa kondisi secara berurutan dan menjalankan blok kode yang sesuai dengan kondisi yang pertama kali benar (true). Jika tidak ada kondisi yang benar, blok kode dalam bagian else akan dijalankan.



```
<?php
$nilai = 60;
echo 'Memeriksa variable $nilai <br>';
echo "Nilai: {$nilai} <br>";
if ($nilai >= 85) {
    echo "Sangat mengesankan!";
} elseif ($nilai >= 70) {
    echo "Selamat anda lulus!";
} else {
    echo "Jangan menyerah, anda pasti bisa!";
}

echo "<br>";
```

- **Percabangan Switch - Case**

Percabangan yang kedua adalah switch - case. Ini adalah alternatif yang bisa kita gunakan untuk memecahkan permasalahan logika dalam PHP. Akan tetapi, penggunaan switch - case ditujukan untuk kasus-kasus yang lebih sederhana daripada if .. else.

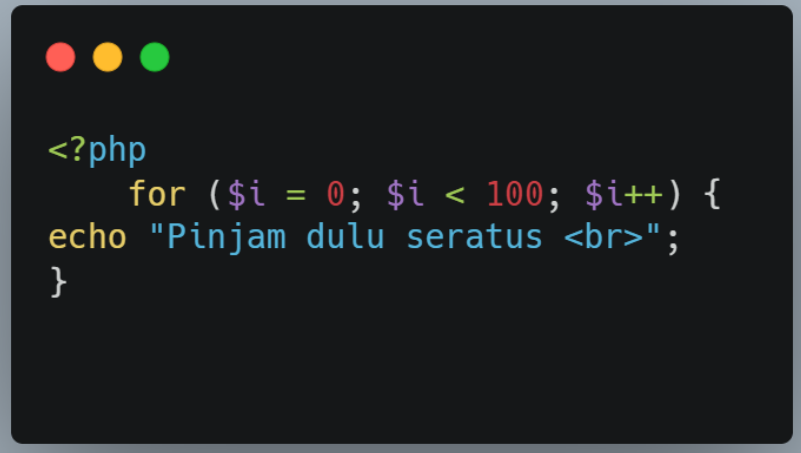
```
<?php
$url = '/about';
switch ($url) {
case '/':
    echo 'Selamat datang di dashboard.';
break;
case '/about':
    echo 'Selamat datang di halaman about.';
break;
case '/contact':
    echo 'Selamat datang di halaman kontak.';
break;
default:
    echo 'Maaf halaman yang anda cari tidak ditemukan.';
}

echo '<br>';
```

## Perulangan (Looping)

- **Perulangan for**

Perulangan for adalah perulangan yang paling simpel. Kita bisa menggunakan metode ini dengan menentukan jumlah perulangan yang kita inginkan. Misalkan kita ingin menampilkan tulisan “Pinjam dulu seratus” sebanyak 100x, kita bisa melakukannya seperti ini:




```
<?php
    for ($i = 0; $i < 100; $i++) {
    echo "Pinjam dulu seratus <br>";
    }
```

Perulangan for membutuhkan 3 buah ekspresi yang mana ekspresi tersebut masing-masing dipisahkan oleh tanda titik koma (;).

Penjelasan:

- **Ekspresi pertama** digunakan untuk menginisialisasi variabel
- **Ekspresi kedua** digunakan untuk boolean statement, yang mana perulangan akan terus dilakukan selama statemen ini bernilai true
- **Ekspresi ketiga** adalah suatu aksi yang akan selalu dijalankan setiap kali satu perulangan telah selesai dilakukan.

Contoh lain untuk menggunakan variable:



```
<?php
#inisialisasi variabel tidak harus dari angka 0
for ($i = 1; $i <= 10; $i++) {
    echo "Perulangan ke-{$i} <br>";
}
```

Contoh lain untuk hitung mundur:

```
<?php
for ($i = 10; $i > 0; $i--) {
    echo "Nilai i = {$i} <br>";
}
```

Contoh lain untuk menggunakan array:

```
<?php
$listMahasiswa = ['Vateh', 'Raveed', 'Afeef'];
for ($i = 0; $i < count($listMahasiswa); $i++) {
    echo "Nama: {$listMahasiswa[$i]} <br>";
}
```

- **Perulangan while**

Perulangan while hampir sama dengan for. Akan tetapi ia lebih ditujukan untuk kasus-kasus di mana kita tidak mengetahui secara pasti ada berapa jumlah perulangan yang harus dilakukan

Contohnya adalah ketika kita membuat game seperti *Flappy Bird*, maka kita akan selalu menggambar pipa baru selama user belum mati. Kondisi “selama user belum mati”, membuat kita tidak benar-benar tahu berapa kali pipa baru akan di-generate dan ditampilkan pada layar. Bisa jadi 100 kali, 500 kali atau bahkan hanya 3 kali saja.



Berikut adalah contoh penggunaan perulangan while:

```
<?php

# kita mulai dari angka 1
$i = 1;

# perulangan akan dilakukan selama variabel $i bukan kelipatan 3
while ($i % 3 !== 0) {
    echo "Nilai i = {$i} <br>";

    # ubah nilai $i secara random dari angka 1-100
    $i = rand(1, 100);
}
```

Contoh lain penggunaan while dengan continue

```
<?php

# kita mulai dari angka 1
$i = 1;

# perulangan akan dilakukan selama variabel $i bukan kelipatan 3
while ($i % 3 !== 0) {
    if ($i % 5 === 0) {
        # skip ke perulangan selanjutnya jika $i adalah kelipatan 5
        # jangan lupa di-random dulu agar tidak terjadi perulangan tanpa batas
        $i = rand(1, 500);
        continue;
    }

    echo "Nilai i = {$i} <br>";

    if ($i > 50 && $i < 60) {
        break; # jika $i bernilai lebih dari 50 dan kurang dari 60 maka stop
    }

    # ubah nilai $i secara random dari angka 1-500
    $i = rand(1, 500);
}
```

- **Perulangan do-while**

Perulangan do-while sama persis dengan perulangan while. Ia sama-sama mengevaluasi ekspresi boolean. Selama ekspresi tersebut bernilai true, perulangan akan terus dilakukan. Dan jika nilai boolean tersebut false, perulangan akan dihentikan.

Perbedaannya adalah sebagai berikut:

- Kalau dalam metode **while**, ekspresi **boolean** akan diperiksa terlebih dahulu sebelum memulai perulangan.
- Ada pun dengan **do-while**, ia akan mengeksekusi perulangan terlebih dahulu, baru setelah itu ia akan mengevaluasi kondisi **boolean**.

```
<?php
while (false) {
    echo "Perulangan ke-1 dengan while! <br>"; }

do {
    echo "Perulangan ke-1 dengan do while! <br>";
} while (false);
```

Pada kode program di atas, perintah di dalam perulangan **while** sama sekali tidak dieksekusi, karena ekspresi **boolean**-nya bernilai **false**. Sedangkan dalam perulangan **do-while**, perintah tersebut akan dieksekusi terlebih dahulu, baru setelah itu ia akan mengevaluasi kondisi **boolean**, jika **true** dia akan melakukan perulangan selanjutnya, dan jika tidak, dia akan **berhenti**.

- **Perulangan foreach**

Jenis perulangan selanjutnya adalah menggunakan **foreach**. Jenis perulangan ini khusus untuk tipe data array baik array asosiatif maupun array terindeks.

Kita bisa melakukan perulangan pada setiap elemen array tanpa harus tahu berapa panjang dari **array** itu sendiri. Sintaksnya ada dua versi, kita bisa menggunakan yang mana saja sesuai dengan kebutuhan.

#### a. Sintaks Singkat

```
<?php
$listMahasiswa = ['Vateh', 'Rafeed', 'Dono']; foreach ($listMahasiswa as $mahasiswa) {
echo "Nama : {$mahasiswa} <br>";
}
```

Perulangan **foreach** (versi singkat) hanya menerima **2 buah variabel** yang dipisahkan kata kunci **as** Variabel pertama adalah variabel bertipe data array Dan variabel kedua adalah variabel item dari array tersebut.

#### b. Sintaks Lengkap Menggunakan Key

```
<?php
$listMahasiswa = ['Vateh', 'Rafeed', 'Dono']; foreach ($listMahasiswa as $key => $mahasiswa) {
echo "Nama : {$mahasiswa} <br>";
}
```

Sintaks foreach versi lengkap ketambahan tanda **=>** dan satu variabel Variabel **\$key** di dalam kode program di atas berisi nilai indeks dari iterasi yang sedang dilakukan Variabel **\$key** bisa menggunakan nama bebas.

## Array

### Apa itu Array?

Array adalah variabel khusus, yang dapat memegang lebih dari satu nilai pada satu waktu. Jika Anda memiliki daftar item (seperti daftar nama mobil), menyimpan mobil dalam variabel tunggal bisa terlihat seperti ini:

```
$cars1 = "Volvo";
$cars2 = "BMW";
$cars3 = "Toyota";
```

Namun, bagaimana jika Anda ingin menelusuri mobil-mobil tersebut dan menemukan mobil tertentu? Dan bagaimana jika Anda tidak punya 3 mobil, tapi 300?

**Solusinya adalah dengan membuat array!**

Karena Sebuah array dapat menampung banyak nilai dalam satu nama, dan Anda dapat mengakses nilai tersebut dengan mengacu pada nomor indeks.

**Di PHP, ada tiga jenis array:**

**1. Array terindeks - Array dengan indeks numerik**

Array terindeks adalah array yang setiap itemnya terbedakan dengan indeks tertentu. Indeks tersebut bertipe data integer dan selalu dimulai dari angka 0. Array \$listMahasiswa yang kita buat dalam contoh diatas adalah termasuk jenis array terindeks.

```
<?php  
  
$listMahasiswa = ['Vateh', 'Raveed', 'Afeef'];  
for ($i = 0; $i < count($listMahasiswa); $i++) {  
    echo "Nama: {$listMahasiswa[$i]} <br>";  
}
```

**2. Array asosiatif - Array dengan kunci bernama**

Array asosiatif adalah suatu array di mana key atau kuncinya bukan berupa indeks integer yang dimulai dari 0, akan tetapi yang menjadi **key**-nya adalah suatu teks bertipe data string. Oleh karena itu ia dinamakan array asosiatif.

```
<?php

$mahasiswa = [
    'nama' => 'Nurul Huda',
    'domisili' => 'Surabaya',
    'jenis_kelamin' => 'Laki-
laki'
```

Untuk menampilkan item yang berada di dalam **array asosiatif**, kita bisa melakukannya dengan memanggil **key**-nya. Hal ini sama persis dengan array **terindeks**, hanya saja jika pada array terindeks **key**-nya bertipe data **integer**, maka pada array asosiatif, **key**-nya bertipe data string.


```
<?php

echo "Nama : {$mahasiswa['nama']}<br>";
echo "Domisili : {$mahasiswa['domisili']}<br>";
echo "Jenis Kelamin : {$mahasiswa['jenis_kelamin']}<br>";
```

### 3. Array multidimensi - Array yang berisi satu atau lebih array

Array multidimensi adalah suatu istilah untuk sebuah array, yang mana ia memiliki item berupa array yang lain.





```
<?php
```

```
$histogram = [  
    [1, 2, 3, 4, 5],  
    [6, 7, 3, 9, 2],  
    [3, 5, 1, 0, 5],  
    [5, 8, 1, 3, 1]  
];
```

Pada contoh di atas, indeks ke-0 dari array \$histogram adalah suatu array. Begitu juga indeks ke-1, ke-2, dan ke-3. Misal kita ingin menampilkan angka 0 pada array tersebut, maka kita harus memanggil array terluarnya dahulu, baru indeks dari array yang di dalam.

## Method GET dan POST

GET dan POST adalah Method yang bisa anda gunakan untuk menyimpan dan mengirimkan sebuah nilai ke halaman yang anda inginkan. Kedua method tersebut memiliki masing-masing fungsi mengirimkan sebuah data atau nilai. Namun pada dasarnya memiliki sebuah perbedaan yang signifikan. Berikut perbedaan nya:

### 1. Method GET

Method GET merupakan metode yang datanya dikirim melalui URL. Data yang dikirim di URL berupa rangkaian pasangan nama dan nilai yang dipisahkan oleh ampersand (&). Berikut contoh Method GET :

<https://vateh.id/data.php?name=alfian&age=21>

Apabila anda lihat dari URL tersebut, disana terdapat nama untuk file PHP yaitu data.php, Nah File ini lah yang kita tuju pada Formulir GET. Berikutnya ada pembatas antara File PHP dan juga Variabel yang dikirimkan yaitu Tanda Tanya (?), dan yang terakhir yaitu Variabel yang dikirimkan beserta isi datanya yang dipisah dengan Tanda Ampersand (&). Variabel yang dikirimkan adalah name dan age.

Sekarang anda coba praktikan bagaimana penggunaan Method GET ini pada Codingan anda, yang anda butuhkan adalah syntax PHP yang berisi Form dan memiliki Method GET. Berikut codenya:

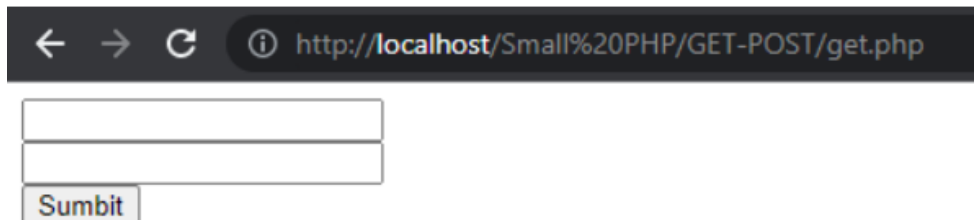
```

<html lang="en">
<head>
<title>Method GET</title>
</head>
<body>
<form action="" method="GET">
<input type="text" name="nama"><br />
<input type="number" name="umur"><br />
<input type="submit" name="submit" value="Sumbit">
</form>

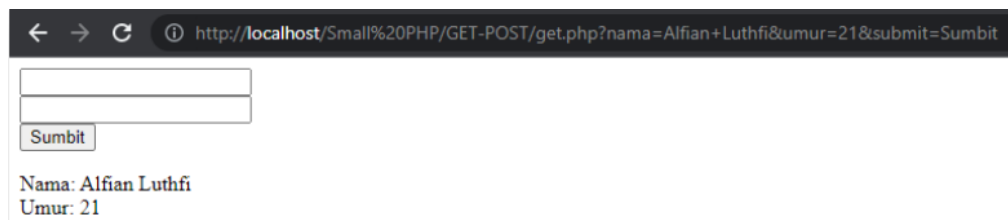
<?php
if ($_GET) {
echo "Nama: " . $_GET["nama"];
echo "<br/>";
echo "Umur: " . $_GET["umur"];
}
?>
</body>
</html>

```

Maka Code diatas akan menghasilkan output berikut:



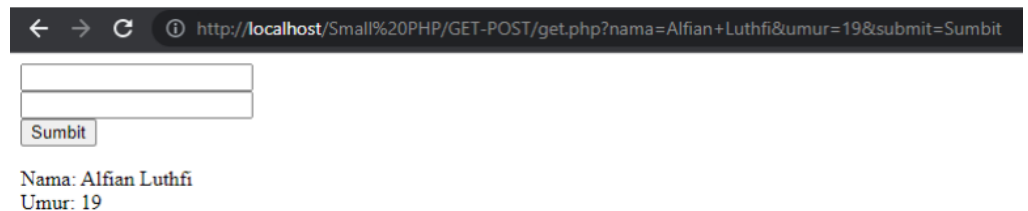
Misalkan kita mengisi Form diatas dengan Nama dan Umur kita. Maka Codingan kita akan mengambil Data yang kita isi menggunakan \$\_GET lalu ditampilkan dengan echo.



**Berikut Hasilnya :**

Bisa dilihat di URL diatas. Ada tulisan [nama=Alfian+Luthfi&umur=21](http://localhost/Small%20PHP/GET-POST/get.php?nama=Alfian+Luthfi&umur=21&submit=Sumbit). Seperti yang dijelaskan tadi, Method

GET ini menggunakan URL, jadi kita bisa mengganti Outputnya dengan mengganti isi URL nya. Misal sekarang saya ganti nilai umur pada URL jadi 19 jadi `nama=Alfian+Luthfi&umur=19`. Maka Outputnya juga akan berubah seperti ini



Seperti itulah penggunaan Method GET pada PHP. Ada yang sering mengatakan method ini tidak aman karena Data yang anda inputkan terlihat di URL dan bisa diganti-ganti secara asal

## 2. Method POST

Method POST adalah metode pengiriman data yang Datanya tidak disimpan pada URL. Data pada method POST ini tetap dikirimkan akan tetapi tidak ditampilkan pada URL seperti GET. Method POST ini biasanya digunakan saat registrasi yang membutuhkan input email dan password yang seharusnya tidak muncul di URL.

Method POST ini dirasa lebih lama daripada method GET, bahkan Method ini juga bisa mengirimkan File seperti gambar dan dokumen, tidak hanya Teks saja. Berikut code untuk method POST:

```

<html lang="en">

<head>
  <title>Method POST</title>
</head>

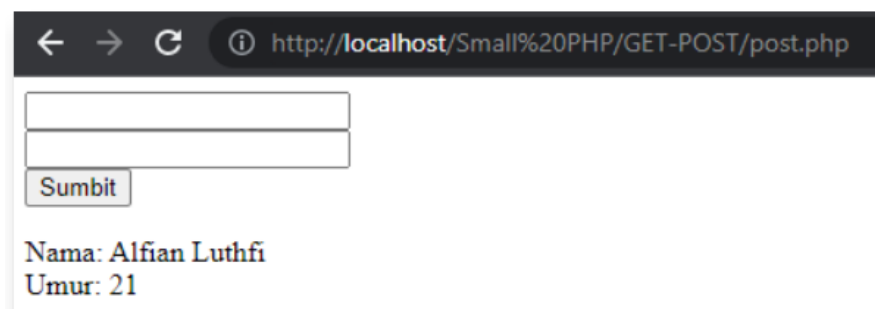
<body>
  <form action="" method="POST">
    <input type="text" name="nama"><br />
    <input type="number" name="umur"><br />
    <input type="submit" name="submit" value="Submit">
  </form>

  <?php
  if ($_POST) {
    echo "Nama: " . $_POST["nama"];
    echo "<br/>";
    echo "Umur: " . $_POST["umur"];
  }
  ?>
</body>

</html>

```

Dari code diatas akan menghasilkan Hasil yang sama seperti Method GET. Tapi perbedaannya, ketika anda isi tidak bisa melihat yang kita inputkan di URL. Beginilah hasilnya :



The screenshot shows a web browser window with the address bar displaying `http://localhost/Small%20PHP/GET-POST/post.php`. The page contains a form with two input fields: a text field and a number field. Below the fields is a button labeled "Submit". After clicking the button, the output is displayed below the form: "Nama: Alfian Luthfi" and "Umur: 21".

Bisa dilihat pada hasil gambar diatas. Output yang keluar sama, akan tetapi di URL tidak muncul data seperti pada method GET. Pada akhirnya kita tidak bisa mengganti Data yang kita inputkan dan tidak bisa melihat data tersebut. Membuat penginputan data lebih Secure.



## Jadi Kapan Harus kita harus menggunakan Method GET dan Kapan kita harus menggunakan Method POST?

Anda harus menggunakan metode GET dan POST dalam pengembangan web sesuai dengan kebutuhan dan tujuan Anda. Berikut panduan umumnya:

### Gunakan Method GET ketika:

1. Anda ingin mengambil data dari server. Metode GET digunakan untuk permintaan baca (read).
2. Data yang Anda kirimkan adalah data publik dan tidak sensitif, misalnya pencarian atau tautan ke halaman.
3. Anda ingin berbagi URL dengan parameter query string yang dapat dibaca oleh pengguna.

Berikut contoh code penggunaan method GET:

```
<form method="GET" action="hasil-pencarian.php">
  <input type="text" name="kata-kunci" placeholder="Cari...">
  <input type="submit" value="Cari">
</form>
```

### Gunakan Method POST ketika:

1. Anda ingin mengirimkan data ke server yang bersifat sensitif, seperti kata sandi atau informasi pribadi.
2. Anda ingin mengirim data yang lebih besar, seperti mengunggah file.
3. Anda ingin menghindari bahwa data yang dikirimkan muncul dalam URL browser karena data tersebut akan muncul di URL dalam metode GET.

Berikut contoh code penggunaan method POST:

```
<form method="POST" action="proses-login.php">
  <input type="text" name="username" placeholder="Username">
  <input type="password" name="password"
placeholder="Password" value="Login">
</form>
```

## **Sumber Referensi**

<https://duniacoding.info/belajar-php-method-get-dan-post-pada-php/>

<https://www.w3schools.com/php/>

<https://jagongoding.com/web/php/dasar/overview/>

<https://www.petanikode.com/php-operator/>