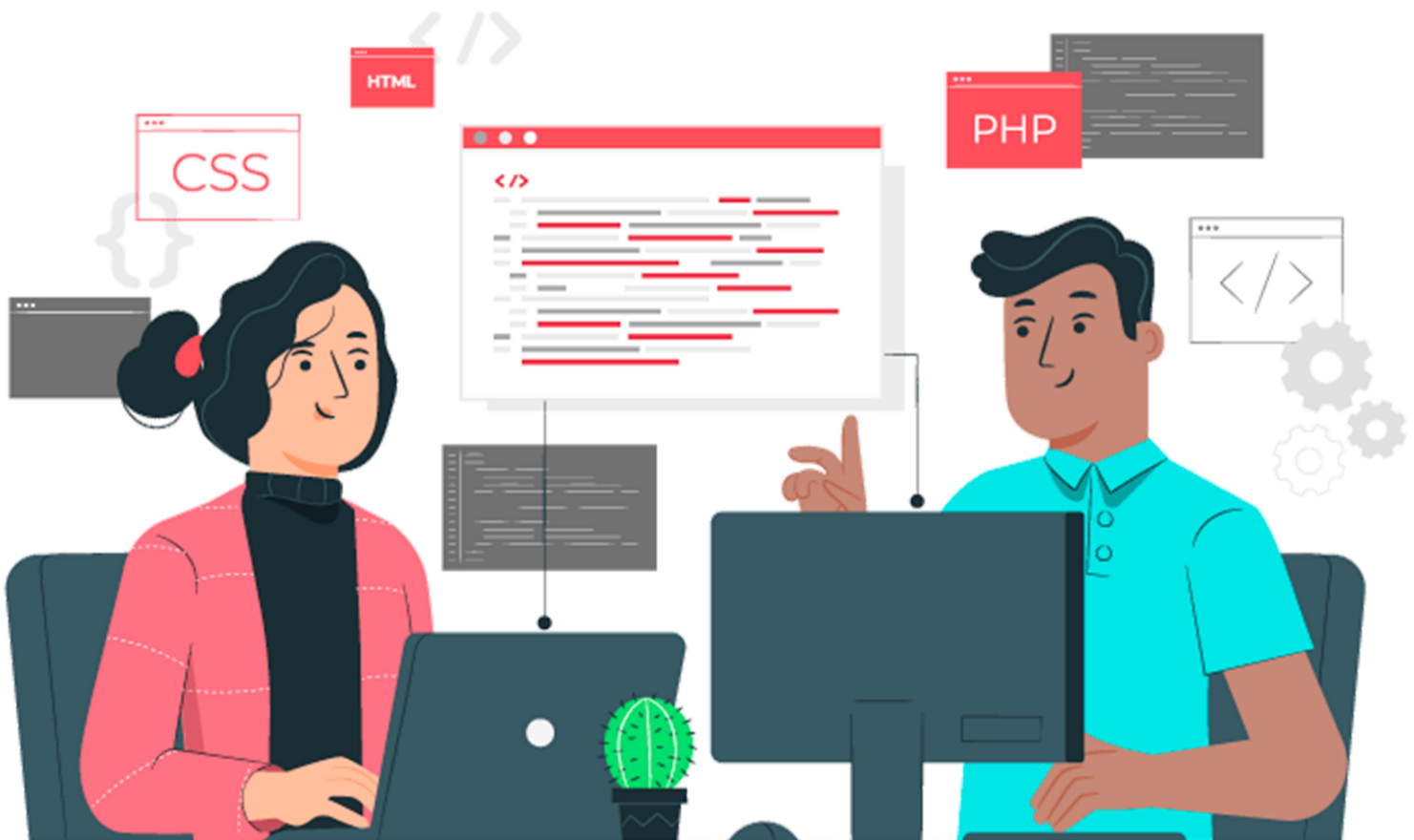


WEB APPLICATION DEVELOPMENT



MODUL 2 DASAR PHP



ENTERPRISE
APPLICATION
DEVELOPMENT

TIM PENYUSUN

Nama	Kode Asisten
Afina Ramadhani	NOY
Alwan Alyafi Mulyawan	ALW
Annisa Syafarani Callista	CAL
Arrizal Fauzan Barliana	AFB
Arsike Cipta Pelangi	ARS
Bima Susila Mukti	BIM
Desita Nur R.	DST
I Komang Gede Andhi Kurniawan	AND
Idelia Muthia Nurbaisa	IDL
Ivan Priyambudi	IVN
M. Ikhsan Pratama P.	ISP
Muhammad Ricky Chandra Dinata	RCD
Mukhamad Furqon	WGS
Nafidzah Kiasati Shadrina	NFZ
Rizky Azis Jayasutisna	SKY
Sabila Chanifah	BYL
Sherly Fratista	SHE
Syfa Nur Lathifah	SYF
Tomi Mulhartono	TOM

DAFTAR ISI

TIM PENYUSUN.....	1
DAFTAR ISI.....	2
TUJUAN PRAKTIKUM.....	3
KEBUTUHAN PRAKTIKUM	3
PEMBAHASAN MATERI	4
1 PHP	4
2 Web Server.....	4
3 XAMPP.....	5
4 Variabel dan Tipe Data.....	5
5 Tipe Operator	6
5.1 Operator Aritmatika.....	7
5.2 Operator Penugasan	8
5.3 Operator <i>Increment & Decrement</i>	10
5.4 Operator Relasi atau Pembandingan	11
5.5 Operator Logika	12
6 Pengkondisian dan Perulangan.....	13
6.1 Fungsi Pengkondisian.....	13
6.1.1 <i>If else</i>	13
6.1.2 <i>Switch case</i>	14
6.1.3 <i>Ternary operator</i>	15
6.2 Fungsi Perulangan.....	15
6.2.1 <i>While</i>	15
6.2.2 <i>Do while</i>	16
6.2.3 <i>For</i>	16
6.2.4 <i>Foreach</i>	17
7 GET dan POST.....	18
DAFTAR PUSTAKA	26

TUJUAN PRAKTIKUM

1. Praktikan dapat memahami dan menggunakan sintaks dasar PHP.
2. Praktikan dapat memahami dan menggunakan pengkondisian, perulangan, operator, variabel, dan array dengan PHP.
3. Praktikan dapat memahami dan menggunakan fungsi *request method* dengan PHP.

KEBUTUHAN PRAKTIKUM

1. IDE (Visual Studio Code, Sublime, PHPstorm, dll)
2. XAMPP
3. Web browser (Microsoft Edge, Google Chrome, dll)

PEMBAHASAN MATERI

1 PHP

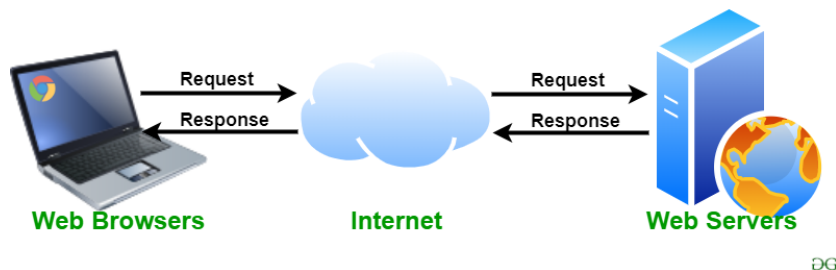
PHP atau *Hypertext Preprocessor* merupakan bahasa *scripting* sisi *server* yang berfungsi agar halaman web menjadi lebih dinamis dan dapat berinteraksi dengan database. PHP diciptakan oleh Rasmus Lerdorf pada tahun 1994. *File* yang berisikan kode PHP harus berekstensi “**.php**” (contoh: modul.php). Sintaks PHP dapat disematkan ke dalam HTML walaupun ekstensi file tersebut tetap “**.php**”. Menulis kode PHP diawali dengan “**<?php**” dan diakhiri dengan “**?>**”.

Contoh penulisan:

```
<?php
// tulis kode disini
?>
```

2 Web Server

PHP dapat dieksekusi jika terdapat web *server* pada *local machine* atau komputer / laptop. Web *server* adalah sebuah layanan untuk menerima atau *request* berupa halaman web melalui HTTP atau HTTPS dari *client* atau *user*, lalu mengirimkan kembali *request* tersebut (merespon) dalam bentuk dokumen HTML.



Contoh web *server*:

1. Apache
2. Nginx (Engine X)
3. Apache Tomcat
4. Microsoft's Internet Information Server (IIS)

Pada praktikum ini, web *server* yang digunakan adalah Apache.

3 XAMPP

XAMPP adalah web *server open source* yang berjalan pada sistem operasi *cross-platform* (Windows, Linux, MacOS). XAMPP menyediakan berbagai keperluan dalam mengelola *website* seperti Apache, MySQL / MariaDB, PHP dan PERL. XAMPP digunakan untuk membuat web *server* pada *local machine*. Dengan menggunakan XAMPP, *developer* tidak perlu mengkonfigurasi kebutuhan pengembangan web secara manual.

4 Variabel dan Tipe Data

Penulisan variabel PHP diawali dengan tanda “\$” kemudian diikuti dengan huruf atau *underscore* “_”. PHP merupakan *case sensitive language* dimana huruf kapital dan huruf kecil dibedakan. Contoh penulisan variabel adalah sebagai berikut:

```
<?php
// penulisan variabel yang benar
$variabel1;
$_variabel2;
$Variabel3;
?>
```

Aturan penulisan variabel pada PHP:

- Tidak boleh diawali dengan angka
- Tidak boleh mengandung simbol
- Tidak boleh mengandung spasi

PHP disebut sebagai *weakly typed or dynamic typed language*, maksudnya adalah tipe data suatu variabel ditentukan oleh nilai yang ada pada variabel tersebut. Jika pada Java, penulisan variabel dituliskan sebelum nama variabel yang dibuat.

Berikut adalah contoh penulisan variabel di PHP:

Contoh tipe data	Output
<pre> <?php // pendeklarasian variabel \$string = "halo"; // string \$int = 20; // integer \$double = 15.5; // double \$array = ["kopi", "teh", "susu"]; // array \$boolean = TRUE; // boolean \$var_null = null; // null // mem-print output dengan echo echo \$string."
"; echo \$int."
"; echo \$double."
"; echo "Saya tidak suka \$array[0]
"; echo \$boolean."
"; echo \$var_null."
"; ?> </pre>	<pre> halo 20 15.5 Saya tidak suka kopi 1 </pre>

Jenis-jenis tipe data pada PHP:

1. string
2. integer
3. double
4. array
5. boolean
6. object
7. null

5 Tipe Operator

Sama seperti bahasa pemrograman lainnya, PHP juga memiliki operator. Fungsi dari operator yang ada pada PHP sama seperti bahasa pemrograman lainnya, hanya berbeda cara penulisannya. Tipe - tipe operator pada PHP antara lain:

- Operator Aritmatika
- Operator Penugasan
- Operator *Increment & Decrement*
- Operator Relasi atau Pembandingan
- Operator Logika
- Operator *Bitwise*
- Operator *Ternary*

5.1 Operator Aritmatika

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika umum seperti penjumlahan, pengurangan, perkalian, pembagian, dan lain-lain. Berikut ini merupakan macam - macam operator aritmatika:

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Pembagian	/
Perkalian	*
Pemangkatan	**
Sisa bagi	%

Contoh penggunaan operator aritmatika:

Kode	Hasil
<pre> <?php // pendeklarasian variabel \$a = 7; \$b = 3; // Penjumlahan \$c = \$a + \$b; echo "\$a + \$b = \$c <hr>"; // Pengurangan \$c = \$a - \$b; echo "\$a - \$b = \$c <hr>"; // Perkalian \$c = \$a * \$b; echo "\$a * \$b = \$c <hr>"; // Pembagian \$c = \$a / \$b; echo "\$a / \$b = \$c <hr>"; // Sisa bagi (modulus) \$c = \$a % \$b; echo "\$a % \$b = \$c <hr>"; // Pemangkatan \$c = \$a ** \$b; echo "\$a ** \$b = \$c <hr>"; ?> </pre>	$7 + 3 = 10$ <hr/> $7 - 3 = 4$ <hr/> $7 * 3 = 21$ <hr/> $7 / 3 = 2.33333333333333$ <hr/> $7 \% 3 = 1$ <hr/> $7 ** 3 = 343$ <hr/>

Penulisan kode juga bisa divariasikan seperti memasukkan fungsi ke dalam echo seperti contoh berikut:

Kode	Hasil
<pre> 1 <?php 2 3 \$a = 5; 4 \$b = 20; 5 6 // persentase 7 echo "\$a adalah ".\$c = \$a * 100 / \$b."% dari \$b"; 8 9 ?> </pre>	<p>5 adalah 25% dari 20</p>

5.2 Operator Penugasan

Operator penugasan (*assignment*) merupakan operator yang digunakan untuk memberikan suatu nilai kepada sebuah variabel. Operator penugasan

dasar dalam PHP adalah "=". Artinya, variabel di sebelah kiri dinyatakan nilainya di variabel atau operasi di sebelah kanan.

Simbol	Operasi	Sama seperti...	Deskripsi
=	$\$x = \y	$\$x = \y	Variabel x memberikan nilainya kepada variabel y (memiliki nilai yang sama)
+=	$\$x += \y	$\$x = \$x + \$y$	Menambahkan nilai variabel x dengan variabel y
-=	$\$x -= \y	$\$x = \$x - \$y$	Mengurangi nilai variabel x dengan variabel y
*=	$\$x *= \y	$\$x = \$x * \$y$	Mengalikan nilai variabel x dengan variabel y
**=	$\$x **= \y	$\$x = \$x ** \$y$	Mengeksponenkan (memangkatkan) nilai variabel x dengan variabel y
/=	$\$x /= \y	$\$x = \$x / \$y$	Membagi nilai variabel x dengan variabel y
%=	$\$x \% = \y	$\$x = \$x \% \$y$	Modulus nilai variabel x dengan variabel y
.=	$\$x .= \y	$\$x = \$x . \$y$	Menambahkan variabel string x dengan variabel string y

Contoh penggunaan operator - operator di atas adalah sebagai berikut:

Kode	Hasil
<pre> <?php // deklarasi variabel \$a = 3; \$b = 4; \$string1 = "EAD "; \$string2 = "Laboratory"; // operasi penjumlahan echo "\$a += \$b => ".\$a+\$b; echo"<hr>"; // operasi pangkat echo "\$a ** \$b => ".\$a**\$b; echo"<hr>"; // operasi penggabungan string echo "\$string1 .= \$string2 => ".\$string1.\$string2; echo"<hr>"; ?> </pre>	<pre> 3 += 4 => 7 7 ** 4 => 2401 EAD .= Laboratory => EAD Laboratory </pre>

5.3 Operator *Increment & Decrement*

Operator *increment* dan *decrement* merupakan operator yang digunakan untuk menambah atau mengurangi nilai numerik sebuah variabel. *Increment* adalah operasi peningkatan, menggunakan simbol '++', sedangkan *decrement* adalah operasi penurunan, menggunakan simbol '--'. Berikut ini merupakan macam - macam operator *increment* dan *decrement*:

Simbol	Nama	Deskripsi
++\$x	<i>Pre-increment</i>	Nilai \$x ditambah 1 dahulu, kemudian nilai \$x dikembalikan
\$x++	<i>Post-increment</i>	Mengembalikan nilai \$x dahulu, kemudian nilai \$x ditambah 1
--\$x	<i>Pre-decrement</i>	Nilai \$x dikurangi 1 dahulu, kemudian nilai \$x dikembalikan
\$x--	<i>Post-decrement</i>	Mengembalikan nilai \$x dahulu, kemudian nilai \$x dikurangi 1

Contoh penggunaan operator - operator di atas adalah sebagai berikut:

Kode	Hasil
<pre><?php // pendeklarasian variabel \$a = 1; \$b = 2; // pre-increment echo "++\$a = ".\$++\$a."<hr>"; // pre-decrement echo "--\$a = ".\$--\$a."<hr>"; // post-increment echo "\$b++ = ".\$b++."<hr>"; // post-decrement echo "\$b-- = ".\$b--."

"; echo "Hasilnya 3 karena nilai \$b awal (2)
 sudah ditambah 1 dari operasi
 post-increment sebelumnya"; ?></pre>	<pre>++1 = 2 --2 = 1 2++ = 2 3-- = 3 Hasilnya 3 karena nilai 2 awal (2) sudah ditambah 1 dari operasi post-increment sebelumnya</pre>

5.4 Operator Relasi atau Pembandingan

Operator relasi atau pembandingan merupakan operator yang digunakan untuk membandingkan nilai dari dua buah variabel. Output yang dihasilkan berupa *boolean true* (benar) atau *false* (salah) sesuai hasil perbandingannya. Berikut ini merupakan macam - macam operator perbandingan:

Simbol	Deskripsi	Operasi	Hasil
<code>==</code>	Sama dengan	<code>\$x == \$y</code>	<i>True</i> jika nilai <code>\$x</code> sama dengan nilai <code>\$y</code>
<code>===</code>	Identik	<code>\$x === \$y</code>	<i>True</i> jika nilai <code>\$x</code> sama dengan nilai <code>\$y</code> , dan mereka bertipe sama
<code>!=</code>	Tidak sama dengan	<code>\$x != \$y</code>	<i>True</i> jika nilai <code>\$x</code> tidak sama dengan nilai <code>\$y</code>
<code><></code>	Tidak sama dengan	<code>\$x <> \$y</code>	<i>True</i> jika nilai <code>\$x</code> tidak sama dengan nilai <code>\$y</code>
<code>!==</code>	Tidak identik	<code>\$x !== \$y</code>	<i>True</i> jika nilai <code>\$x</code> tidak sama dengan nilai <code>\$y</code> , atau tipe mereka berbeda
<code>></code>	Lebih dari	<code>\$x > \$y</code>	<i>True</i> jika nilai <code>\$x</code> lebih besar daripada nilai <code>\$y</code>
<code><</code>	Kurang dari	<code>\$x < \$y</code>	<i>True</i> jika nilai <code>\$x</code> lebih kecil daripada nilai <code>\$y</code>
<code>>=</code>	Lebih dari sama dengan	<code>\$x >= \$y</code>	<i>True</i> jika nilai <code>\$x</code> lebih besar atau sama dengan nilai <code>\$y</code>
<code><=</code>	Kurang dari sama dengan	<code>\$x <= \$y</code>	<i>True</i> jika nilai <code>\$x</code> lebih kecil atau sama dengan nilai <code>\$y</code>
<code><=></code>	<i>Spaceship</i>	<code>\$x <=> \$y</code>	<ul style="list-style-type: none"> 1 jika <code>\$x > \$y</code> 0 jika <code>\$x = \$y</code>
<code>??</code>	<i>Null coalescing</i>	<code>\$x <=> \$y</code>	<ul style="list-style-type: none"> <code>\$x</code> jika <code>\$x</code> tidak <i>null</i> <code>\$y</code> jika <code>\$x</code> <i>null</i>

Contoh penggunaan operator - operator di atas adalah sebagai berikut:

Kode	Hasil
<pre> <?php \$a = 1; \$b = 2; \$c = "2"; \$d = null; // Operasi kurang dari (<) echo \$a, ' < '.\$b.' => '; var_dump(\$a < \$b); echo'<hr>'; // Operasi sama dengan (=) echo \$a, ' == '.\$b.' => '; var_dump(\$a == \$b); echo'<hr>'; // Operasi identik (===) echo \$b, ' === '.\$c.' => '; var_dump(\$b === \$c); echo'<hr>'; // Operasi null coalescing (??) echo \$c, ' ?? '.\$d.' => '; var_dump(\$c ?? \$d); echo'<hr>'; ?> </pre>	<pre> 1 < 2 => bool(true) 1 == 2 => bool(false) 2 === 2 => bool(false) 2 ?? => string(1) "2" </pre>

5.5 Operator Logika

Operator logika merupakan operator yang digunakan untuk menggabungkan pernyataan kondisional. Berikut ini merupakan macam - macam operator logika:

Operator	Nama	Operasi	Hasil
&&	AND	$\$x \ \&\& \ \y	<i>True</i> jika $\$x$ dan $\$y$ bernilai <i>true</i>
	OR	$\$x \ \ \y	<i>True</i> jika salah satu di antara $\$x$ atau $\$y$ bernilai <i>true</i>
!	NOT	$!\$x$	<i>True</i> jika $\$x$ bernilai <i>false</i>

Contoh penggunaan operator - operator di atas adalah sebagai berikut:

Kode	Hasil
<pre> <?php \$a = true; \$b = false; // Operasi AND (&&) echo '\$a && \$b = '; if (\$a && \$b) { echo 'true <hr>'; } else { echo 'false <hr>'; } // Operasi OR () echo '\$a \$b = '; if (\$a \$b) { echo 'true <hr>'; } else { echo 'false <hr>'; } // Operasi NOT (!) echo '(\$a != \$b) ==> '; if (\$a != \$b) { echo 'true <hr>'; } else { echo 'false <hr>'; } ?> </pre>	<pre> \$a && \$b = false ----- \$a \$b = true ----- (\$a != \$b) ==> true ----- </pre>

6 Pengkondisian dan Perulangan

6.1 Fungsi Pengkondisian

Fungsi pengkondisian adalah salah satu fungsi pada bahasa pemrograman yang berisi pernyataan kondisional yang digunakan untuk melakukan suatu aksi berdasarkan kondisi tertentu. Pada PHP, terdapat beberapa pernyataan kondisional, yaitu sebagai berikut:

6.1.1 *If else*

Pernyataan *if else* akan melakukan suatu aksi apabila kondisi terpenuhi (*true*). Contoh penggunaan pernyataan *if else* adalah sebagai berikut:

Kode	Hasil
<pre> <?php \$score = 75; echo "Score : \$score
"; if (\$score > 80) { // jika kondisi terpenuhi \$grade = "A"; } else if (60 < \$score && \$score < 80) { // jika kondisi sebelumnya tidak terpenuhi \$grade = "B"; } else { // jika semua kondisi sebelumnya tidak terpenuhi \$grade = "C"; } echo "Grade : \$grade"; ?> </pre>	<p>Score : 75 Grade : B</p>

6.1.2 Switch case

Pernyataan *switch case* digunakan untuk melakukan aksi tertentu berdasarkan kondisi - kondisi yang berbeda. Pernyataan *switch case* mirip dengan pernyataan *if else*, perbedaannya terletak pada sintaksnya. Contoh penggunaan pernyataan *switch case* adalah sebagai berikut:

Kode	Hasil
<pre> <?php \$prak = "WAD"; echo "Praktikum : \$prak
"; switch (\$prak) { case "AC": \$lab = "ERP"; break; case "WAD": \$lab = "EAD"; break; case "SISOP": \$lab = "Sisjar"; break; } echo "Laboratorium : \$lab"; ?> </pre>	<p>Praktikum : WAD Laboratorium : EAD</p>

6.1.3 Ternary operator

Operator *ternary* merupakan bentuk lain yang lebih pendek dari struktur operasi pengkondisian (`if {} else {}`). Berikut adalah perbandingannya:

Sintaks operasi pengkondisian	Sintaks operator <i>ternary</i>
<pre>// sintaks pengkondisian biasa \$nilai = 75; if (\$nilai > 70) { \$hasil = 'Lulus'; } else { \$hasil = 'Tidak Lulus'; }</pre>	<pre>// ternary operator \$hasil = (\$nilai > 80) ? 'Lulus' : 'Tidak Lulus';</pre>

Contoh penggunaan operator *ternary* adalah sebagai berikut:

Kode	Hasil
<pre><?php // ternary operator \$nilai = 75; \$hasil = (\$nilai > 80) ? 'Lulus' : 'Tidak Lulus'; echo \$hasil; ?></pre>	Tidak Lulus

6.2 Fungsi Perulangan

Fungsi perulangan adalah salah satu fungsi pada bahasa pemrograman yang digunakan mengeksekusi sebuah kode secara berulang-ulang sampai kondisi yang ditentukan terpenuhi. Pada PHP, terdapat beberapa fungsi perulangan, yaitu sebagai berikut:

6.2.1 While

While adalah salah satu bentuk perulangan yang memiliki jumlah perulangan sesuai dengan suatu kondisi tertentu. Pernyataan di dalam *while* akan dieksekusi secara berulang-ulang selama kondisi memenuhi (bernilai *true*).

Contoh penggunaan *while* adalah sebagai berikut:

Kode
<pre><?php \$a = 1; while (\$a <= 9) { // selama kondisi ini terpenuhi // aksi di bawah ini akan dilakukan berulang-ulang echo " ".\$a; \$a++; } ?></pre>
Hasil
1 2 3 4 5 6 7 8 9

6.2.2 Do while

Do while adalah bentuk perulangan yang mirip dengan *while*. *Do while* akan mengeksekusi pernyataan secara berulang-ulang selama kondisi memenuhi (bernilai *true*). Perbedaannya adalah pada *do while* pengecekan kondisi atau syarat perulangan dilakukan setelah eksekusi pernyataan yang diulang.

Contoh penggunaan *do while* adalah sebagai berikut:

Kode	Hasil
<pre><?php \$b = 1; do { // aksi yang akan dilakukan berulang-ulang echo "do while " . \$b++. "
"; } // kondisi yang diinginkan while (\$b <= 10); ?></pre>	do while 1 do while 2 do while 3 do while 4 do while 5 do while 6 do while 7 do while 8 do while 9 do while 10

6.2.3 For

For adalah salah satu bentuk perulangan yang digunakan untuk melakukan perulangan yang sebelumnya telah dipastikan jumlahnya. Contoh penggunaan *for* adalah sebagai berikut:

Kode	Hasil
<pre><?php // (inisialisasi; kondisi; perulangan) for (\$x=1; \$x<=5; \$x++) { // aksi yang akan dilakukan berulang-ulang echo "Nomor \$x
"; } ?></pre>	<p>Nomor 1 Nomor 2 Nomor 3 Nomor 4 Nomor 5</p>

6.2.4 Foreach

Foreach adalah bentuk perulangan yang hanya dapat digunakan pada variabel bertipe data *array*, dan bisa juga digunakan untuk perulangan melalui setiap pasangan kunci-nilai (*key-value*) dalam *array*. Berikut adalah sintaks dari *foreach*:

```
foreach ($array as $nilai) {
    //operasi yang ingin dijalankan
}
```

Contoh penggunaan *foreach* adalah sebagai berikut:

Kode
<pre><?php echo "<p> Mata kuliah semester ini terdiri dari: </p>"; echo ""; \$matkul = array("WAD", "Sisop", "Manprosi", "EA"); foreach (\$matkul as \$value) { echo " \$value "; } echo ""; ?></pre>
Hasil
<p>Mata kuliah semester ini terdiri dari:</p> <ol style="list-style-type: none"> 1. WAD 2. Sisop 3. Manprosi 4. EA

Contoh penggunaan *foreach* untuk pasangan kunci-nilai (*key-value*) dalam *array* adalah sebagai berikut:

Kode	Hasil
<pre><?php \$info = array("Nama" => "Ikhsan", "Kelas" => "SI-41-01", "Prodi" => "Sistem Informasi"); echo "<p>Informasi Mahasiswa</p>"; foreach (\$info as \$key => \$value) { echo "\$key : \$value
"; } ?></pre>	<p>Informasi Mahasiswa</p> <p>Nama : Ikhsan Kelas : SI-41-01 Prodi : Sistem Informasi</p>

7 GET dan POST

GET dan POST adalah salah satu dari beberapa *request method* pada protokol HTTP. PHP menyediakan sarana untuk berkomunikasi dengan kedua *request method* tersebut.

GET dan POST memiliki fungsi yang sama, yaitu untuk mengirim dan menerima data dari sebuah form HTML. Adapun perbedaan dari kedua variabel tersebut yaitu:

- *Method* POST tidak menampilkan *query string* pada *address bar*, sehingga lebih aman dibandingkan *method* GET yang akan terlihat pada *address bar*.
- *Method* GET dapat di-cache dalam *browser*.
- *Method* GET dapat di-bookmark.
- *Method* GET mempunyai batasan jumlah panjang data, yaitu maksimal 2048 karakter.

Perlu diingat bahwa *request method* pengiriman dan penerimaan data harus **sama**. Misalnya, ketika atribut *method* pada *form* diisi dengan GET, maka saat melakukan pengambilan datanya harus menggunakan *method* GET juga. Begitu juga dengan *method* POST.

Berikut adalah contoh penulisan dari variabel POST dan GET:

```
<?php
// mengambil data beratribut name 'email' dan ber-method GET di
form, kemudian disimpan di variabel bernama 'emailGET'
$emailGET = $_GET['email'];

// mengambil data beratribut name 'gender' dan ber-method POST
di form, kemudian disimpan di variabel bernama 'genderPOST'
$genderPOST = $_POST['gender'];
?>
```

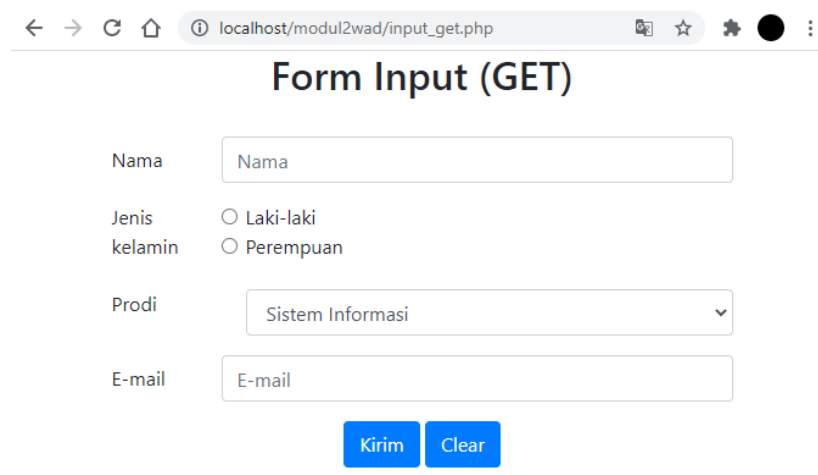
7.1 Contoh penggunaan *method* GET

Source code input_get.php:

```
input_get.php X output_get.php
modul2wad > input_get.php > html > head
9 <body>
10 <div class="container-sm">
11 <h2 style="text-align: center;">Form Input (GET)</h2>
12 <br>
13 <form action="output_get.php" method="get">
14 <fieldset class="form-group">
15 <div class="form-group row">
16 <label class="col-sm-2 col-form-label">Nama</label>
17 <div class="col-sm-10">
18 <input type="text" class="form-control" placeholder="Nama" name="nama">
19 </div>
20 </div>
21 <div class="form-group row">
22 <legend class="col-form-label col-sm-2 pt-0">Jenis kelamin</legend>
23 <div class="col-sm-10">
24 <div class="form-check">
25 <input class="form-check-input" type="radio" name="gender" value="Laki-laki">
26 <label class="form-check-label">
27 Laki-laki
28 </label>
29 </div>
30 <div class="form-check">
31 <input class="form-check-input" type="radio" name="gender" value="Perempuan">
32 <label class="form-check-label">
33 Perempuan
34 </label>
35 </div>
36 </div>
37 </div>
38 <div class="form-group row">
39 <legend class="col-form-label col-sm-2 pt-0">Prodi</legend>
40 <div class="col-sm-10">
41 <div class="form-check">
42 <select class="form-control" name="prodi">
43 <option value="Sistem Informasi">Sistem Informasi</option>
44 <option value="Teknik Industri">Teknik Industri</option>
45 <option value="Teknik Logistik">Teknik Logistik</option>
46 </select>
47 </div>
48 </div>
49 </div>
```

```
49     </div>
50     <div class="form-group row">
51         <label class="col-sm-2 col-form-label">E-mail</label>
52         <div class="col-sm-10">
53             <input type="email" class="form-control"
54                 placeholder="E-mail" name="email">
55         </div>
56     </div>
57     <div style="text-align: center;">
58         <input type="submit" value="Kirim" class="btn btn-primary
59             mb-2">
60         <input type="reset" value="Clear" class="btn btn-primary
61             mb-2">
62     </div>
63 </form>
</div>
</body>
```

Tampilan dari input_get.php:



← → ↻ 🏠 ⓘ localhost/modul2wad/input_get.php 📄 ☆ ⚙️ ● ⋮

Form Input (GET)

Nama

Jenis kelamin ☐ Laki-laki ☐ Perempuan

Prodi

E-mail

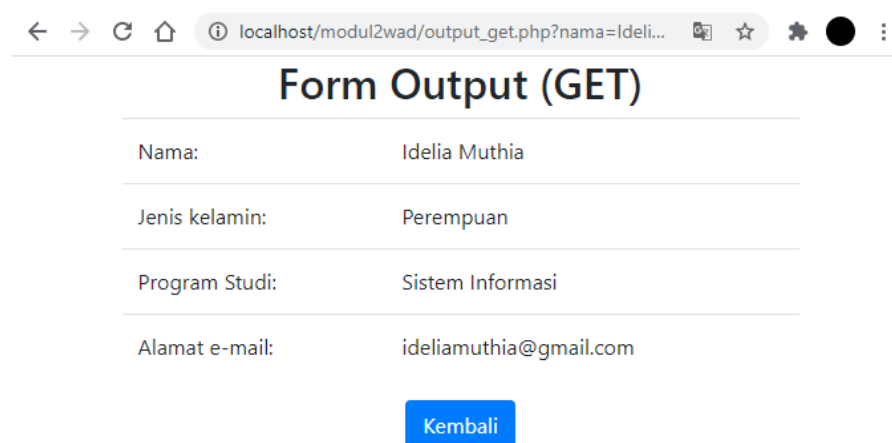
Source code output_get.php:

```

input_get.php  output_get.php X
modul2wad > output_get.php > html > head
9  <body>
10  <?php
11  $nama = $_GET['nama'];
12  $gender = $_GET['gender'];
13  $prodi = $_GET['prodi'];
14  $email = $_GET['email'];
15  ?>
16  <div class="container-sm">
17  <h2 style="text-align: center;">Form Output (GET)</h2>
18  <fieldset>
19  <table class="table">
20  <tr>
21  <td>Nama: </td>
22  <td><?= $nama ?> </td>
23  </tr>
24  <tr>
25  <td>Jenis kelamin: </td>
26  <td><?= $gender ?> </td>
27  </tr>
28  <tr>
29  <td>Program Studi: </td>
30  <td><?= $prodi ?> </td>
31  </tr>
32  <tr>
33  <td>Alamat e-mail: </td>
34  <td><?= $email ?> </td>
35  </tr>
36  </table>
37  <div style="text-align: center;">
38  <a href="input_get.php" class="btn btn-primary mb-2"> Kembali</a>
39  </div>
40  </fieldset>
41  </div>
42  </body>

```

Tampilan hasil dari output_get.php yang telah menerima data dengan *method* GET dari input_get.php:



← → ↻ 🏠 ⓘ localhost/modul2wad/output_get.php?nama=Ideli... 📄 ☆ ⚙️ ● ⋮

Form Output (GET)

Nama:	Idelia Muthia
Jenis kelamin:	Perempuan
Program Studi:	Sistem Informasi
Alamat e-mail:	ideliamuthia@gmail.com

[Kembali](#)

Jika menggunakan *method* GET, maka terlihat data-data yang diinput dalam *query string* di URL:

localhost/modul2wad/output_get.php?nama=Idelia+Muthia&gender=Perempuan
=Perempuan&prodi=Sistem+Informasi&email=ideliamuthia%40gmail.com

7.2 Contoh penggunaan *method* POST

Source code input_post.php:

```

input_post.php X output_post.php
modul2wad > input_post.php > html > body > div.container-sm > h2
9 <body>
10 <div class="container-sm">
11 <h2 style="text-align: center;">Form Input (POST)</h2>
12 <form action="output_post.php" method="post">
13 <fieldset class="form-group">
14 <div class="form-group row">
15 <label class="col-sm-2 col-form-label">Nama</label>
16 <div class="col-sm-10">
17 <input type="text" class="form-control" placeholder="Nama"
   name="nama">
18 </div>
19 </div>
20 <div class="form-group row">
21 <legend class="col-form-label col-sm-2 pt-0">Jenis kelamin</legend>
22 <div class="col-sm-10">
23 <div class="form-check">
24 <input class="form-check-input" type="radio" name="gender"
   value="Laki-laki">
25 <label class="form-check-label">
26 Laki-laki
27 </label>
28 </div>
29 <div class="form-check">
30 <input class="form-check-input" type="radio" name="gender"
   value="Perempuan">
31 <label class="form-check-label">
32 Perempuan
33 </label>
34 </div>
35 </div>
36 </div>
37 <div class="form-group row">
38 <legend class="col-form-label col-sm-2 pt-0">Prodi</legend>
39 <div class="col-sm-10">
40 <div class="form-check">
41 <select class="form-control" name="prodi">
42 <option value="Sistem Informasi">Sistem Informasi</option>
43 <option value="Teknik Industri">Teknik Industri</option>
44 <option value="Teknik Logistik">Teknik Logistik</option>
45 </select>
46 </div>

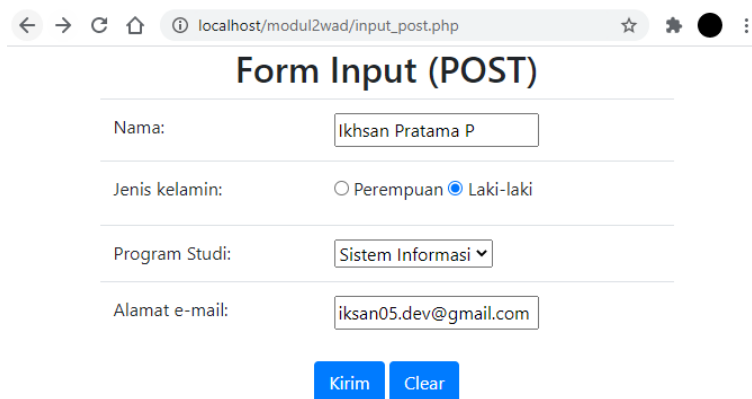
```

```

46         </div>
47     </div>
48 </div>
49 <div class="form-group row">
50 <label class="col-sm-2 col-form-label">E-mail</label>
51 <div class="col-sm-10">
52     <input type="email" class="form-control" placeholder="E-mail"
53         name="email">
54 </div>
55 </div>
56 <div style="text-align: center;">
57     <input type="submit" value="Kirim" class="btn btn-primary mb-2">
58     <input type="reset" value="Clear" class="btn btn-primary mb-2">
59 </div>
60 </form>
61 </div>
62 </body>

```

Tampilan dari input_post.php yang telah mengirim data dengan *method* GET ke output_post.php:



localhost/modul2wad/input_post.php

Form Input (POST)

Nama:

Jenis kelamin: ☐ Perempuan ☒ Laki-laki

Program Studi:

Alamat e-mail:

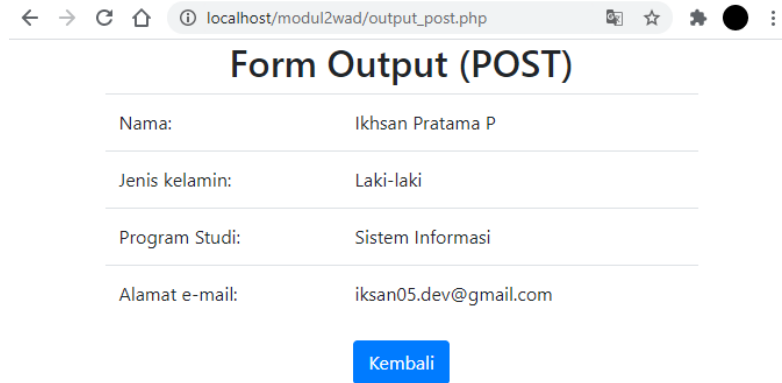
Source code output_post.php:

```

input_post.php  output_post.php X
output_post.php > html > body
4  <head>
5      <title>Form Output (POST)</title>
6      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/
    4.5.2/css/bootstrap.min.css"
    integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP
    +VmmDGMNSt9UJ0Z" crossorigin="anonymous">
7  </head>
8
9  <body>
10     <?php
11         $nama = $_POST['nama'];
12         $gender = $_POST['gender'];
13         $prodi = $_POST['prodi'];
14         $email = $_POST['email'];
15     >?>
16     <div class="container-sm">
17         <h2 style="text-align: center;">Form Output (POST)</h2>
18         <fieldset>
19             <table class="table">
20                 <tr>
21                     <td>Nama: </td>
22                     <td><?= $nama ?> </td>
23                 </tr>
24                 <tr>
25                     <td>Jenis kelamin: </td>
26                     <td><?= $gender ?> </td>
27                 </tr>
28                 <tr>
29                     <td>Program Studi: </td>
30                     <td><?= $prodi ?> </td>
31                 </tr>
32                 <tr>
33                     <td>Alamat e-mail: </td>
34                     <td><?= $email ?> </td>
35                 </tr>
36             </table>
37             <div style="text-align: center;">
38                 <a href="input_post.php" class="btn btn-primary mb-2">
                    Kembali</a>
39             </div>
40         </fieldset>
41     </div>

```

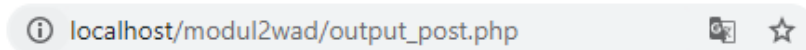
Tampilan hasil dari `output_post.php` yang telah menerima data dengan *method* POST dari `input_post.php`:



Nama:	Ikhsan Pratama P
Jenis kelamin:	Laki-laki
Program Studi:	Sistem Informasi
Alamat e-mail:	iksan05.dev@gmail.com

Kembali

Jika menggunakan *method* POST, pada URL tidak terlihat data-data yang diinput.



localhost/modul2wad/output_post.php

DAFTAR PUSTAKA

<https://php.net>

<https://phptopoint.com>

<https://geeksforgeeks.org>

<https://w3resource.com>

<https://w3schools.com>

Laboratory, EAD. (2019). Modul 3 Praktikum WAD: PHP Dasar (POST & GET).

Modul Praktikum.