

Modul Praktikum

WEB APPLICATION DEVELOPMENT 2022

Modul 2
Dasar PHP

ASISTEN

| | |
|-------------------------------|------|
| Alaric Rasendriya Aniko | MQSU |
| Alfian Maulana Fardhani | HEHE |
| Alif Yanuar Aditya Subagyo | SAYA |
| Andrian Firmansyah | ANDA |
| Anggraeni Xena Paradita | MAAX |
| Arif Raihan Alaudin | RIPA |
| Barajati Syakurnia | NIAA |
| Fakhri Hassan Maulana | HASH |
| Ferda Ayu Dwi Putri Febrianti | FEYA |
| Habib Al Fajri | HDMI |
| Ilham Maulana Abdurrahman | MAUL |
| Joel Rayapoh Damanik | DJUL |
| Karisma Nabil Santosa | W WK |
| Mohammad Wasik Amin | AMIN |
| Muhammad Iqbal F W | ARDN |
| Nabeel Azhar | BEEL |
| Naufal Aqil Himawan | NAHH |
| Pradita Cahyani | TACA |
| Syariif Abdurrahman Bathik | FXXY |
| Zulfa Irfaninda Ramadhanti | YUHU |

Daftar Isi

| | |
|--|-----------|
| Daftar Isi | 3 |
| Definisi | 5 |
| Sejarah singkat | 5 |
| Fungsi PHP | 5 |
| Tipe data pada PHP | 6 |
| Variabel | 7 |
| Peraturan penamaan variabel | 7 |
| Operator | 8 |
| Operator Aritmatika | 8 |
| Operator Penugasan | 8 |
| Operator Perbandingan | 9 |
| Operator Perbandingan Equal dan Identik | 10 |
| Operator logika | 11 |
| Operator Increment & Decrement | 12 |
| Logika Percabangan | 13 |
| Percabangan if-else | 13 |
| Percabangan if | 13 |
| Percabangan else | 13 |
| Membuat lebih dari 1 kondisi else if | 14 |
| Percabangan switch-case | 14 |
| Kapan harus menggunakan if, switch case? | 15 |
| Kapan menggunakan if | 15 |
| Kapan menggunakan switch-case | 15 |
| Perulangan | 16 |
| Jenis-jenis Perulangan | 16 |
| Perulangan for | 16 |
| Perintah break dan continue | 17 |
| Perulangan while | 17 |
| Perulangan do-while | 18 |
| Perulangan foreach | 18 |
| Array | 20 |
| Array terindeks | 21 |
| Array asosiatif | 21 |
| Array multidimensi | 22 |
| GET dan POST | 23 |
| Metode GET | 23 |
| Metode POST | 24 |

| | |
|---|-----------|
| Contoh penggunaan method GET | 25 |
| Contoh penggunaan method POST | 29 |
| Tutorial GitHub (Push dan Pull), Akses Program Modul 2 | 33 |
| Sumber Referensi | 34 |

Definisi

PHP adalah bahasa pemrograman yang umum dipakai dalam pembuatan dan pengembangan suatu web. 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 pesat membuatnya bertumbuh menjadi bahasa pemrograman web yang kuat.

PHP bukan lagi sekadar dipakai membangun halaman web sederhana, tetapi membuat berbagai web ternama dunia seperti WordPress dan Wikipedia.

Sejarah singkat

Membahas apa itu PHP berarti tidak lepas dari nama Rasmus Lerdorf, sang penemu. Pengembangan bahasa pemrograman script server-side ini pertama kali dilakukan tahun 1994. Saat ini pengelolaannya berada di bawah naungan The PHP Group dengan alamat web resmi di www.php.net.

Dari nama Personal Home Page Tools, nama PHP sempat berganti menjadi FI atau Forms Interpreter. Namun, sejak versi 3.0, FI tidak lagi dipakai dan digantikan dengan singkatan PHP terbaru. Kini bahasa PHP yang banyak dipakai adalah versi 5.0. Versi tersebut pertama kali dirilis bulan Juni 2004.

Terjadi perubahan besar dalam inti interpreter PHP. Model pemrograman berorientasi pada objek juga turut ditanamkan seiring tren bahasa pemrograman terkini. Hal tersebut mendorong pengaplikasian PHP pada berbagai program sistem informasi di sektor industri, kesehatan, keuangan, hingga akademik. Plus, PHP bisa anda pakai secara gratis dan tergolong open source.

Kemudahan dan popularitas menempatkan PHP sebagai standar keterampilan bahasa pemrograman yang harus dikuasai oleh programmer web.

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 pada PHP

Ada berbagai macam tipe data: mulai dari tipe data asli dan tipe data buatan. Untuk PHP sendiri mendukung setidaknya 8 tipe data skalar. Akan tetapi dalam modul ini, sementara kita akan fokuskan dahulu pada pembahasan 6 tipe data saja, karena 6 tipe data ini adalah tipe data yang paling dasar.

6 tipe data tersebut adalah:

| Tipe data | Keterangan | Contoh |
|-----------|---|--|
| Integer | Berisi bilangan bulat. Tipe integer dapat ditulis dalam bentuk desimal (basis 10), oktal (basis 8) serta heksadesimal (basis 16). | 2010, +920, 0770, 0xFF |
| Float | Berisi bilangan numerik, pecahan, atau mengandung angka desimal di belakang koma | 3.14, 0.999, -9.1, 0.314E1 |
| Boolean | Berisi 2 nilai yaitu True atau false | Ketika suatu logika if else menghasilkan nilai balik maka hasilnya true atau false |
| String | Berisi data teks yang diapit oleh tanda " atau " | 'String dalam bahasa pemrograman PHP' |
| Char | Berisi nilai karakter tunggal | 'P' 'h' 'o' |
| Array | Berisi suatu himpunan data | \$data[0] = 'Aku cinta WAD'; |

Variabel

Variabel adalah suatu “wadah” untuk menyimpan suatu data atau nilai. Kita bisa menyimpan berbagai macam data dari berbagai macam tipe pada variabel

1. Peraturan penamaan variabel

Dalam pembuatan variabel terdapat beberapa aturan. Berikut aturan sebagaimana terdapat di tabel berikut

| Peraturan | Keterangan |
|--|---|
| diawali tanda \$ | Setiap nama variabel dalam bahasa pemrograman PHP didefinisikan dengan tanda \$ lalu diikuti oleh nama variabel itu sendiri |
| nama diawali huruf atau underscore (_) | Nama variabel PHP harus diawali huruf, atau tanda underscore (_). Kita bisa membuat variabel dengan nama \$_nilai atau \$nilai123 akan tetapi tidak bisa membuat variabel dengan nama \$1nilai |
| case sensitive | PHP membedakan huruf besar dan kecil dalam penamaan variabel. Maka variabel \$nilai, \$niLai dan juga \$nllAi dianggap 3 variabel yang berbeda |
| hanya boleh huruf dan angka | nama variabel 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. Kita juga tidak bisa menggunakan spasi dalam penamaan variabel di PHP |
| inisiasi dengan tanda = | Kita bisa memberi nilai terhadap suatu variabel dengan menggunakan operator sama dengan (=). Contoh: \$praktikum = "WAD"; |
| tanda titik koma (;) | Tanda titik koma (;) memberitahu mesin PHP bahwa akhir dari pernyataan saat ini telah tercapai. |

Operator

Operator berfungsi untuk berinteraksi dengan variabel, seperti misalnya menambahkan 2 variabel integer, atau mengalikan satu variabel dengan variabel lainnya. Semua itu membutuhkan penggunaan operator.

1. Operator Aritmatika

Operator aritmatika adalah operator untuk menghitung operasi matematika dasar. Mulai dari:

- A. penjumlahan (+)
- B. pengurangan (-)
- C. perkalian (*)
- D. dan pembagian (/).

Bentuk paling sederhananya, ekspresi aritmatika terdiri dari dua buah operand dan satu buah operator. Misalnya kita akan menjumlahkan angka 10 dan angka 5, kita bisa menulisnya seperti berikut:

```
<?php
$a = 10 + 5; // Addition
$b = 10 - 5; // Subtraction
$c = 10 * 5; // Multiplication
$d = 10 / 5; // Division
```

Tidak hanya 2 operan seperti contoh di atas. Di dalam PHP, kita juga bisa mendefinisikan ekspresi aritmatika yang lebih kompleks. Misalnya:

```
<?php
$a = (10 + (-3 * 5)) / 3; // -1.6666666666666667
```

2. Operator Penugasan

Operator penugasan atau assignment adalah operator yang berfungsi untuk memberi nilai kepada suatu variabel. Nilai tersebut bisa berupa nilai tunggal yang sudah jelas, atau juga bisa nilai yang berupa hasil dari suatu ekspresi tertentu; misal ekspresi aritmatika atau juga ekspresi logika.

Operator penugasan dalam PHP adalah operator `=`. Kita telah menggunakan operator tersebut sejak awal ketika mulai membuat variabel di dalam PHP.

Perhatikan dan praktikkan kode program di bawah:

```
<?php
$valueMath = 8;
$average = (10 + 7 + $valueMath) / 3;

$a = 5;
$a += 10; // equals $a = $a + 10;
echo var_dump($a) . "<br>"; // Output: int(15)

$a -= 20; // equals $a = $a - 20;
echo var_dump($a) . "<br>"; // Output: int(-5)

$a *= 100; // equals $a = $a * 100;
echo var_dump($a) . "<br>"; // Output: int(-500)

$a /= 10; // equals $a = $a / 10;
echo var_dump($a) . "<br>"; // Output: int(-5)
```

3. Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan antara dua nilai. Hasil yang didapatkan dari operator perbandingan adalah suatu nilai dengan tipe data boolean, yaitu true atau false.

Jika pernyataan perbandingannya benar, nilai yang dikembalikan adalah true. Dan jika pernyataan perbandingannya salah, nilai yang akan didapatkan pun menjadi false.

Berikut adalah operator perbandingan yang bisa kita gunakan dalam PHP:

| Simbol | Keterangan |
|--------------------|------------------------------|
| <code>></code> | Lebih dari |
| <code>>=</code> | Lebih dari atau sama dengan |
| <code><</code> | Kurang dari |
| <code><=</code> | Kurang dari atau sama dengan |

Berikut ini contoh penggunaan operator perbandingan pada PHP. Silakan dipraktikkan agar bisa lebih paham.

```
<?php

$score = 90;
# comparison variabel
$pass = $score > 80;

echo "{$score} > 80 = ";
var_dump($pass); // Output: 90 > 80 = bool(true)
echo "<br>";

# directly compare numbers
echo "3 >= 3 = ";
var_dump(3 >= 3); // Output: 3 >= 3 = bool(true)
echo "<br>";

echo "3 < 6 = ";
var_dump(3 < 6); // Output: 3 < 6 = bool(true)
echo "<br>";

echo "5 <= 3 = ";
var_dump(5 <= 3); // Output: 5 <= 3 = bool(false)
echo "<br>";

# you can also compare between 2 strings
echo "'a' < 'b' = ";
var_dump('a' < 'b'); // Output: 'a' < 'b' = bool(true)
echo '<br>';

echo "'abc' < 'b' = ';
var_dump('abc' < 'b'); // Output: 'abc' < 'b' = bool(true)
echo '<br>';
```

4. Operator Perbandingan Equal dan Identik

Dalam PHP, selain perbandingan seperti di atas. Kita juga bisa menggunakan perbandingan equal dan juga identik. Perbandingan equal tidak memperdulikan tipe data sehingga ia menganggap sama antara 10 dan "10". Sedangkan perbandingan yang bersifat identik, ia menganggap penting tipe data sehingga dua nilai akan dikatakan sama jika dia sama dari sisi nilai dan sama dari sisi tipe datanya.

| Simbol | Keterangan |
|--------|---------------------------|
| == | sama secara equal |
| ==== | sama secara identik |
| != | tidak sama secara equal |
| !== | tidak sama secara identik |

| | |
|----|-------------------------|
| <> | sama dengan operator != |
|----|-------------------------|

5. Operator logika

Operator logika membutuhkan dua operan dan satu operator. Operator yang paling sering digunakan adalah operator and (&& / and), operator or (|| / or) dan operator negasi atau not (!).

| Operator | Simbol | Keterangan |
|----------|-------------|--|
| and | && atau and | Akan bernilai true jika kedua operand bernilai true |
| or | or | |
| negasi | ! | akan membalik yang true menjadi false dan yang false menjadi true. |

Contoh:

```
?php

$a = true && true; # true
$b = true and false; # false
$c = false && false; # false
$d = true || false; # false

# the following expressions will be solved from left to right
$e = true && false && false;

# the expression inside the brackets is completed first
$f = "a" === "a" && ((3 > 5) or 3 + 5 >= 7);

# negation
$g = !true;
$h = !(true && !false);

# var_dump all the above variables to see the result
var_dump($a); echo "<br>"; // Output: bool(true)
var_dump($b); echo "<br>"; // Output: bool(false)
var_dump($c); echo "<br>"; // Output: bool(false)
var_dump($d); echo "<br>"; // Output: bool(true)
var_dump($e); echo "<br>"; // Output: bool(false)
var_dump($f); echo "<br>"; // Output: bool(true)
var_dump($g); echo "<br>"; // Output: bool(false)
var_dump($h); echo "<br>"; // Output: bool(false)
```

6. Operator Increment & Decrement

Operator increment (++) atau decrement (--) adalah jalan pintas dari operator += atau -= yang telah berlalu penjelasannya pada bagian operator penugasan. Dan kedua operator tersebut hanya berfungsi pada variabel saja.

Anda bisa mencoba kode program di bawah agar lebih jelas bagaimana cara kerja kedua operator tersebut:

```
<?php
// Increment
$a = 1;
echo "a = {$a} <br>"; // Output: a = 1
$a++;
echo "a = {$a} <br>"; // Output: a = 2
$a++;
echo "a = {$a} <br>"; // Output: a = 3

# the contents of the variable $a are first entered into the variable $b
# then the variable $a adds itself to the number 1
$b = $a++;
echo "b = {$b} <br>"; // Output: b = 3
echo "a = {$a} <br>"; // Output: a = 4

// Decrement
$a--;
echo "a = {$a} <br>"; // Output: a = 3
$a--;
echo "a = {$a} <br>"; // Output: a = 2

// the contents of the variable $a are entered first into the variable $b
// then the variable $a subtracts itself by the number 1
$b = $a--;
echo "b = {$b} <br>"; // Output: b = 2
echo "a = {$a} <br>"; // Output: a = 1
```

Logika Percabangan

Untuk struktur kontrol percabangan pada PHP, terdapat beberapa macam. Ada if else, ada switch case, dan ada ternary. Dalam tutorial ini kita akan membahas ketiga jenis percabangan tersebut.

1. 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.

Contoh:

```
<?php
$pass = true;

echo 'Checking variable $pass <br>';

if ($ pass) {
    echo "Pass variable is true <br>";
}
```

b. Percabangan 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

$value = 60;

echo 'Checking variable $value <br>';
echo "Value: {$value} <br>";

if ($value >= 70) {
    echo "Congratulations, students!";
} else {
    echo "Sorry, students did not pass";
}

echo "<br>";

// Output
// Checking variable $value
// Value: 60
// Sorry, students did not pass
```

c. Membuat lebih dari 1 kondisi else if

```
● ● ●

<?php

$value = 60;

echo 'Checking variable $value <br>';
echo "Value: {$value} <br>";

if ($value >= 85) {
    echo "Very impressive!";
} elseif ($value >= 70) {
    echo "Congratulations you passed!";
} else {
    echo "Don't give up, you can do it!";
}

echo "<br>";

// Output
// Checking variable $value
// Value: 60
// Don't give up, you can do it!
```

2. Percabangan switch-case

Percabangan switch-case merupakan percabangan yang membandingkan isi sebuah variabel dengan beberapa nilai. Jika proses perbandingan tersebut menghasilkan nilai **true**, maka baris kode di dalamnya akan diproses.

Kondisi SWITCH CASE terdiri dari 2 bagian, yakni perintah **SWITCH** dimana terdapat nama variabel yang akan diperiksa, serta 1 atau lebih perintah **CASE** untuk setiap nilai yang akan diperiksa.

```
<?php

$url = '/about';

switch ($url) {
    case '/':
        echo 'Welcome to the dashboard.';
        break;
    case '/about':
        echo 'Welcome to the about page.';
        break;
    case '/contact':
        echo 'Welcome to the contact page.';
        break;
    default:
        echo 'Sorry, the page you were looking for was not found.';
}

echo '<br>';
```

3. Kapan harus menggunakan if, switch case?

a. Kapan menggunakan if

Ketika anda memiliki logika yang cukup rumit. Melibatkan lebih dari satu variabel atau juga membutuhkan ekspresi logika yang panjang, maka gunakanlah if.

b. Kapan menggunakan switch-case

Anda bisa menggunakan switch case jika cabang pengkondisian anda hanya melibatkan satu variabel saja dan tidak mengandung operator logika semisal lebih dari (>) atau kurang dari (<) dan sebagainya.

Perulangan

Dengan blok kode perulangan, kita bisa membuat suatu statement atau perintah dalam PHP dengan satu kali tulis. Untuk kemudian kita biarkan sistem yang akan menjalankannya berulang kali.

Misal kita ingin menampilkan tulisan: “**Saya berjanji tidak akan mengulangi lagi**” sebanyak 100 kali. Kita cukup menggunakan satu perintah echo saja, lalu kita masukkan kode tersebut ke dalam blok kode perulangan, dan mengaturnya agar sistem mengeksekusinya sebanyak 100 kali.

1. Jenis-jenis Perulangan

- perulangan for
- perulangan while
- perulangan do-while
- perulangan foreach
- perulangan rekursif

2. Perulangan for

Perulangan for adalah perulangan yang paling simpel. Kita bisa menggunakan metode ini dengan menentukan jumlah perulangan yang kita inginkan. Misalkan saya ingin menampilkan tulisan “Saya berjanji tidak akan mengulangi” sebanyak 100x, saya bisa melakukannya seperti ini:

```
<?php
for ($i = 0; $i < 100; $i++) {
    echo "I promise not to repeat <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-contoh yang lain dari perulangan for:

```
<?php

// variable initialization does not have to be 0
for ($i = 1; $i <= 10; $i++) {
    echo "{$i} <br>";
}
```

3. Perintah break dan continue

Kita bisa men-skip satu step perulangan dengan perintah continue. Kita juga bisa memberhentikan proses perulangan dengan perintah break sebelum ia sampai pada iterasi yang terakhir.

Perhatikan contoh berikut:

```
<?php

// variable initialization does not have to be 0
for ($i = 1; $i <= 50; $i++) {
    if ($i % 10 === 0) {
        continue; // skip the loop if the value of $i is divisible by 10
    }

    echo "{$i} <br>";

    if ($i > 40) {
        break; // stop loop if $i is more than 20
    }
}
```

4. Perulangan while

Jenis perulangan yang kedua adalah perulangan while. Ia 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.

Nah, dalam kasus-kasus yang tidak jelas seperti itu, perulangan yang cocok untuk kita gunakan adalah perulangan while.

Berikut ini adalah contoh penggunaan while:

```
<?php  
  
$i = 0;  
  
// this loop will be executed as long as the value of $i is less than 20.  
while ($i < 20) {  
    echo "{$i} <br>";  
    $i++;  
}
```

5. 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.

Bedanya hanya satu:

- 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.

Apa konsekuensinya?

- Pada while, jika kondisi pertama kali bernilai false, perulangan tidak akan dieksekusi sama sekali.
- Sedangkan pada do...while, perulangan akan tetap dilakukan satu kali jika kondisi pertama sudah bernilai false.

Agar lebih jelas, perhatikan dan coba contoh berikut:

6. 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.

```
<?php  
  
$listStudent = ['Joel', 'Adit', 'Alif'];  
  
foreach ($listStudent as $student) {  
    echo "Name : {$student} <br>";  
}  
  
// Output  
// Name : Joel  
// Name : Adit  
// Name : Alif
```

Penjelasan:

- 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

Kelebihan menggunakan foreach:

- Sintaks lebih singkat daripada perulangan for untuk menampilkan array
- Item array per-index sudah disimpan dalam satu variabel sehingga kita tidak perlu memanggil

Array

Array adalah suatu tipe data yang bersifat spesial dalam PHP, ia bisa menyimpan himpunan beberapa nilai dalam satu variabel saja.

Misalkan kita memiliki 3 buah variabel yang menyimpan data nama mahasiswa.

Dari pada kita membuat satu variabel untuk tiap nama mahasiswa seperti ini:

```
<?php  
  
$student1 = "Joel";  
$student2 = "Alif";  
$student3 = "Adit";
```

Lebih baik kita simpan ketiga variabel di atas dalam satu array saja seperti berikut:

```
<?php  
  
$listStudent = [  
    "Joel",  
    "Alif",  
    "Adit"  
];
```

Untuk menampilkan array, kita bisa menggunakan indeks. Indeks pertama dimulai dari 0 dan seterusnya.

```
<?php  
  
echo $listStudents[2] . "<br>"; // Joel  
echo $listStudents[0] . "<br>"; // Alif  
echo $listStudents[1] . "<br>"; // Adit
```

Atau kita juga bisa menggunakan perulangan semisal foreach atau for untuk menampilkan array tersebut.

```
<?php  
  
foreach ($listStudent as $student) {  
    echo $student . "<br>";  
}
```

Jenis-jenis Array

1. Array terindeks

Array terindeks merupakan array yang di dalamnya setiap itemnya terbedakan dengan indeks tertentu. Indeks tersebut bertipe data integer yang selalu dimulai dari angka 0.



```
<?php

$listStudent = [
    "Joel",
    "Adit",
    "Alif"
];
echo $listStudent[0] . "<br>"; // Joel
echo $listStudent[1] . "<br>"; // Adit
echo $listStudent[2] . "<br>"; // Alif
```

2. Array asosiatif

Jenis kedua dari array dalam bahasa pemrograman PHP adalah: array asosiatif.

Ia 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.

Berikut ini adalah contoh dari array asosiatif:



```
<?php

$student = [
    'name' => 'Asep Dayat',
    'domicile' => 'Bandung',
    'gender' => 'male'
];
```

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.

Perhatikan kode program berikut:

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

3. Array multidimensi

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

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

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.

1. Metode GET

Metode GET adalah metode yang datanya dikirim melalui URL, data yang dikirim di URL berupa rangkaian pasangan nama dan nilai yang dipisahkan oleh ampersand (&).

```
localhost:3000/test.php?nama=WAD&umur=2022&submit=Submit
```

```
1 <head>
2     <title>Method GET</title>
3 </head>
4
5 <body>
6     <form action="" method="GET">
7         <input type="text" name="nama"><br />
8         <input type="number" name="umur"><br />
9         <input type="submit" name="submit" value="Submit">
10    </form>
11
12    <?php
13    if ($_GET) {
14        echo "Nama: " . $_GET["nama"];
15        echo "<br/>";
16        echo "Umur: " . $_GET["umur"];
17    }
18    ?>
19 </body>
20
21 </html>
```

2. Metode POST

Metode POST adalah metode yang pengiriman data yang datanya tidak disimpan pada URL. Dengan metode POST data tetap dikirimkan tetapi data tidak tampil pada URL seperti GET.

localhost:3000/test.php?

```
1 <html lang="en">
2
3 <head>
4     <title>Method POST</title>
5 </head>
6
7 <body>
8     <form action="" method="POST">
9         <input type="text" name="nama"><br />
10        <input type="number" name="umur"><br />
11        <input type="submit" name="submit" value="Submit">
12    </form>
13
14    <?php
15    if ($_POST) {
16        echo "Nama: " . $_POST["nama"];
17        echo "<br/>";
18        echo "Umur: " . $_POST["umur"];
19    }
20    ?>
21 </body>
22
23 </html>
```

Contoh penggunaan method GET

Source Code getInput.php

```
1 <body>
2   <div class="container">
3     <h1 class="text-center">GET INPUT</h1>
4     <form action="getOutput.php" method="GET">
5       <div class="mb-3">
6         <label for="name" class="form-label">Nama Lengkap</label>
7         <input type="text" class="form-control" id="name" name="name" placeholder="Masukkan Nama Lengkapmu">
8       </div>
9       <div class="mb-3">
10        <label for="name" class="form-label">Email</label>
11        <input type="email" class="form-control" id="email" name="email" placeholder="Masukkan Emailmu">
12      </div>
13      <div class="mb-3">
14        <label for="number" class="form-label">No Handphone</label>
15        <input type="text" class="form-control" id="number" name="number" placeholder="Masukkan No Handphonemu">
16      </div>
17      <div class="mb-3">
18        <label for="address" class="form-label">Alamat</label>
19        <textarea class="form-control" name="address" id="address" rows="4"></textarea>
20      </div>
21      <div class="mb-3">
22        <label for="mobil" class="form-label">Pilih Mobil</label>
23        <select name="mobil" id="" class="form-select">
24          <option selected disabled> Pilih Jenis Mobil</option>
25          <option value="toyota">Toyota CH-R</option>
26          <option value="lexus">Lexus LS 2020</option>
27          <option value="honda">Honda Odyssey 2020</option>
28          <option value="suzuki">New Suzuki Swift 2020</option>
29        </select>
30      </div>
31      <div class="mb-3">
32        <label for="supir">Membutuhkan Supir?</label>
33        <div class="form-check">
34          <label class="form-check-label">Ya</label>
35          <input type="radio" class="form-check-input" value="Ya" name="butuhSupir">
36        </div>
37        <div class="form-check">
38          <label class="form-check-label">Tidak</label>
39          <input type="radio" class="form-check-input" value="Tidak" name="butuhSupir">
40        </div>
41      </div>
42      <div class="mb-3">
43        <div class="d-flex justify-content-center">
44          <button class="btn btn-primary" type="submit">Submit</button>
45        </div>
46      </div>
47    </form>
48  </div>
49 </body>
```

Tampilan Code getInput.php

GET INPUT

Nama Lengkap

Email

No Handphone

Alamat

Pilih Mobil

Membutuhkan Supir?

- Ya
 Tidak

Submit

Source Code getOutput.php

```
1 <body>
2 <?php
3     // jika ada data yang diinput maka tampilakan namanya, jika tidak maka kosong
4     $name = isset($_GET['name']) ? $_GET['name'] : '';
5     $email = isset($_GET['email']) ? $_GET['email'] : '';
6     $number = isset($_GET['number']) ? $_GET['number'] : '';
7     $address = isset($_GET['address']) ? $_GET['address'] : '';
8     $mobil = isset($_GET['mobil']) ? $_GET['mobil'] : '';
9     $butuhSupir = isset($_GET['butuhSupir']) ? $_GET['butuhSupir'] : '';
10    ?>
11    <div class="container">
12        <h1 class="text-center">GET OUTPUT</h1>
13        <table class="table">
14            <thead>
15                <tr>
16                    <th scope="col">#</th>
17                    <th scope="col">Nama</th>
18                    <th scope="col">Email</th>
19                    <th scope="col">No Handphone</th>
20                    <th scope="col">Alamat</th>
21                    <th scope="col">Jenis Mobil</th>
22                    <th scope="col">Butuh Supir</th>
23                </tr>
24            </thead>
25            <tbody>
26                <tr>
27                    <th scope="row">1</th>
28                    <td><?php echo $name ?></td>
29                    <td><?php echo $email ?></td>
30                    <td><?php echo $number ?></td>
31                    <td><?php echo $address ?></td>
32                    <td><?php echo $mobil ?></td>
33                    <td><?php echo $butuhSupir ?></td>
34                </tr>
35            </tbody>
36        </table>
37        <div class="mb-3">
38            <div class="d-flex justify-content-center">
39                <a href="getInput.php" class="btn btn-outline-primary w-50" >Kembali</a>
40            </div>
41        </div>
42    </div>
43 </body>
```

Tampilan Code getOutput.php

| GET OUTPUT | | | | | | |
|------------|-------------|--------------------|--------------|---------------------------------|-------------|-------------|
| # | Nama | Email | No Handphone | Alamat | Jenis Mobil | Butuh SUpir |
| 1 | Anya Forger | WAKUWAKU@gmail.com | +555351421 | Jl. Gang Pelit, No. 1, Westalis | honda | Tidak |
| | | | | | | |

Kembali

Contoh penggunaan method POST

Source Code postInput.php

```
1  <body>
2      <div class="container">
3          <h1 class="text-center">POST INPUT</h1>
4          <form action="postOutput.php" method="POST">
5              <div class="mb-3">
6                  <label for="name" class="form-label">Nama Lengkap</label>
7                  <input type="text" class="form-control" id="name" name="name" placeholder="Masukkan Nama Lengkapmu">
8              </div>
9              <div class="mb-3">
10                 <label for="name" class="form-label">Email</label>
11                 <input type="email" class="form-control" id="email" name="email" placeholder="Masukkan Emailmu">
12             </div>
13             <div class="mb-3">
14                 <label for="number" class="form-label">No Handphone</label>
15                 <input type="text" class="form-control" id="number" name="number" placeholder="Masukkan No Handphonemu">
16             </div>
17             <div class="mb-3">
18                 <label for="address" class="form-label">Alamat</label>
19                 <textarea class="form-control" name="address" id="address" rows="4"></textarea>
20             </div>
21             <div class="mb-3">
22                 <label for="mobil" class="form-label">Pilih Mobil</label>
23                 <select name="mobil" id="" class="form-select">
24                     <option selected disabled> Pilih Jenis Mobil</option>
25                     <option value="toyota">Toyota CH-R</option>
26                     <option value="lexus">Lexus LS 2020</option>
27                     <option value="honda">Honda Odyssey 2020</option>
28                     <option value="suzuki">New Suzuki Swift 2020</option>
29                 </select>
30             </div>
31             <div class="mb-3">
32                 <label for="supir">Membutuhkan Supir?</label>
33                 <div class="form-check">
34                     <label class="form-check-label">Ya</label>
35                     <input type="radio" class="form-check-input" value="Ya" name="butuhSupir">
36                 </div>
37                 <div class="form-check">
38                     <label class="form-check-label">Tidak</label>
39                     <input type="radio" class="form-check-input" value="Tidak" name="butuhSupir">
40                 </div>
41             </div>
42             <div class="mb-3">
43                 <div class="d-flex justify-content-center">
44                     <button class="btn btn-primary" type="submit">Submit</button>
45                 </div>
46             </div>
47         </form>
48     </div>
49 </body>
```

Tampilan postInput.php

POST INPUT

Nama Lengkap

Email

No Handphone

Alamat

Pilih Mobil

Membutuhkan Supir?

Ya

Tidak

Source Code postOutput.php

```
1 <body>
2 <?php
3 $name = $_POST['name'];
4 $email = $_POST['email'];
5 $number = $_POST['number'];
6 $address = $_POST['address'];
7 $mobil = $_POST['mobil'];
8 $butuhSupir = $_POST['butuhSupir'];
9 ?>
10    <div class="container">
11        <h1 class="text-center">POST OUTPUT</h1>
12        <table class="table">
13            <thead>
14                <tr>
15                    <th scope="col">#</th>
16                    <th scope="col">Nama</th>
17                    <th scope="col">Email</th>
18                    <th scope="col">No Handphone</th>
19                    <th scope="col">Alamat</th>
20                    <th scope="col">Jenis Mobil</th>
21                    <th scope="col">Butuh SUpir</th>
22                </tr>
23            </thead>
24            <tbody>
25                <tr>
26                    <th scope="row">1</th>
27                    <td><?php echo $name ?></td>
28                    <td><?php echo $email ?></td>
29                    <td><?php echo $number ?></td>
30                    <td><?php echo $address ?></td>
31                    <td><?php echo $mobil ?></td>
32                    <td><?php echo $butuhSupir ?></td>
33                </tr>
34            </tbody>
35        </table>
36        <div class="mb-3">
37            <div class="d-flex justify-content-center">
38                <a href="getInput.php" class="btn btn-outline-primary w-50">Kembali</a>
39            </div>
40        </div>
41    </div>
42 </body>
```

Tampilan postOutput.php

POST OUTPUT

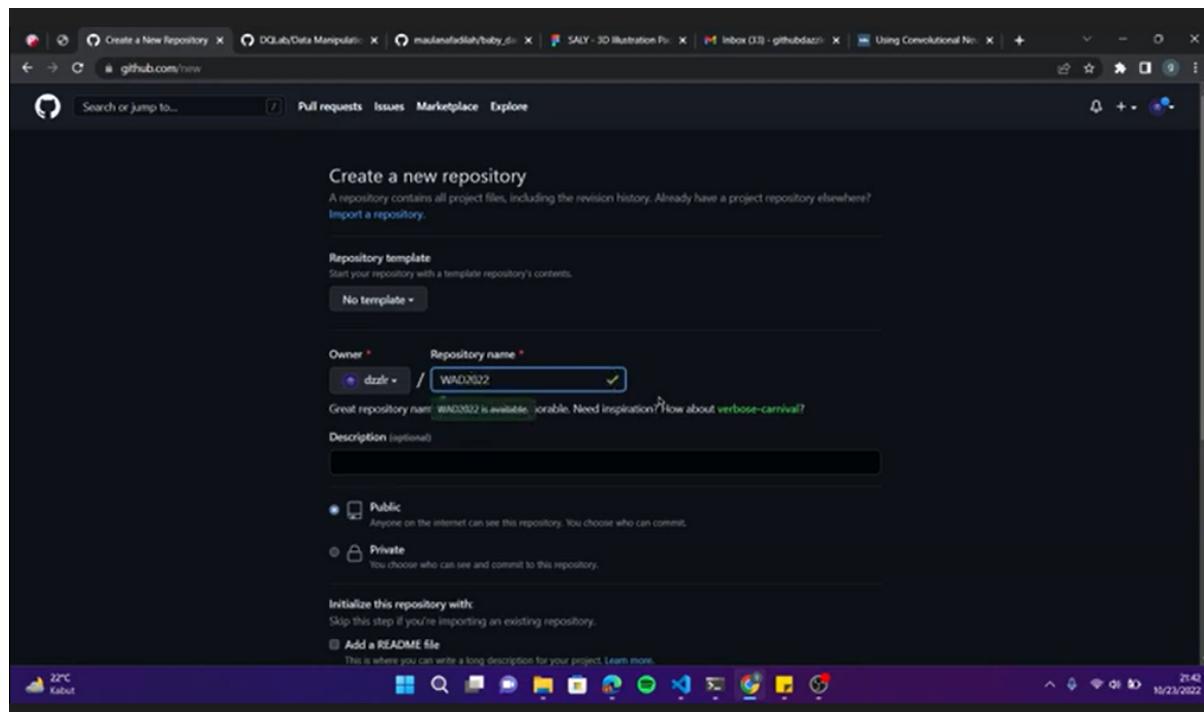
| # | Nama | Email | No Handphone | Alamat | Jenis Mobil | Butuh SUPir |
|---|-------------|--------------------|--------------|---------------------------|-------------|-------------|
| 1 | Anya Forger | WAKUWAKU@gmail.com | +555351421 | Jl. Waku Waku Sabisi No 1 | suzuki | Ya |

Kembali

Tutorial GitHub (Push dan Pull), Akses Program Modul 2

Berikut lampiran link video tutorialnya:

<https://drive.google.com/file/d/1BTUUrzAGd9f1uxUSzjnTCuWviwStOyNG/view>



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/>