



Topic

- Introduction to PHP

Purpose

Students are expected to:

1. Students are able to create static websites using PHP

Introduction

What is PHP?

PHP (PHP: Hypertext Preprocessor) is an open source server-side scripting programming language. As a scripting language, PHP executes programming instructions during runtime. The results of the instructions will certainly differ depending on the data processed. PHP is a server-side programming language, so scripts from PHP will later be processed on the server. Types of servers that are often used in conjunction with PHP include Apache, Nginx, and LiteSpeed. In addition, PHP is also an open source programming language. Users are free to modify and develop according to their needs.

Why Use PHP?

Today, no less than 78% of websites around the world use the programming language created by Rasmus Lerdorf in 1995. Even big platforms like Facebook use it.

So, what makes PHP so popular? Why use PHP that's more than two decades old? Here are some reasons:

- **It tends to be easy to learn** — compared to some other popular programming languages, PHP is easier to learn.
- **Abundant learning materials** — the "quite old" age of PHP has led to a lot of documentation, guidance, and active communities scattered around the virtual space. So, there is no need to be afraid if you are having difficulties.
- **PHP is open-source** — anyone can use PHP without spending a dime.
- **High speed** — PHP has been proven to improve loading speed compared to other languages. For example, it is three times faster than Python in some cases.
- **Lots of database options** — PHP can be used in almost any type of database. Starting from MySQL, to non-relational databases such as Redis.
- **Good compatibility with HTML** — PHP scripts don't interfere with HTML at all. Instead, the two of them complement each other.
- **High flexibility** — PHP can be combined with a lot of other programming languages. So you can use it as needed.
- **Multi-platform** — PHP can be used on a variety of operating systems. Starting from Windows, Linux, to MacOS.

- **It's always up to date** — since it first appeared in 1995, PHP is now at version 8.2.4.
- **Supports cloud services** — who would have thought, even though PHP is almost two decades old, it can support cloud services with good scalability.

PHP Basic Syntax

Each programming language has its own coding rules. The same goes for PHP. All PHP code written should be stored with .php extension files. The basic syntax opens with <?php and closes with ?> as seen in the following example:

```
<?php
    echo "Welcome";
?>
```

Description:

- **<?php** : mandatory code to open PHP programs.
- **echo** : a command to display text.
- **"Welcome"**: the text to be displayed and written between quotation marks and semicolons.
- **?>** is the code to terminate PHP and is mandatory when combined with other programming languages such as HTML.

PHP syntax is **case sensitive**. So, the use of uppercase or lowercase letters will also affect the output given.

Example:

```
<?php
    $alamat = "Yogyakarta";
    echo $alamat;
?>
```

The above code will produce the output: **Yogyakarta** on the website page.

However, if it is written like this:

```
<?php
    $alamat = "Yogyakarta";
    echo $Alamat;
?>
```

Then an error will be generated. The reason is that there is a difference between **the \$alamat** and **\$Alamat** variables.

In PHP, comments can be written as an explanation of the code. Comments in PHP use `or /*/` and will not be displayed as output in the browser. Examples are as follows:

```
<?php
    Here's an example of using comments
    echo "How are you?";
    /*
    Well this is also an example of a comment
    written more than one line
    */
?>
```

Writing PHP Codes

Basically, the coding of this programming language is divided into two:

1. PHP Native

Native is writing PHP code from scratch when designing a website. PHP Native is often used by developers who have good coding skills or those who want to create unique pipeline frameworks with high functionality.

2. PHP Framework

When using a framework, developers can take advantage of ready-made website management frameworks. This means that there is no need to make it from scratch making the work easier. A framework is a framework that can help developers work more efficiently and complete website development faster. Some of the popular PHP frameworks used include: CodeIgniter, Laravel framework, Yii, Symfony and Zend Framework. If you are already proficient in PHP native, it is highly recommended to try switching to the PHP framework. That's because the code in the framework has been optimized according to standards, in terms of speed and security.

Variables in PHP

Variables are storage places that are used to store data that can change during the course of a program. Variables can be thought of as containers filled with different types of data. Here are a few things you need to know about variables in PHP:

- **Variable Declaration:** the variable is declared with a dollar sign (\$) followed by the variable name. For example: `$namaVariabel`.
- **Variable Naming Rules:** variable names must begin with a letter or underscore (_) and may contain letters, numbers, or underscores. However, there is a difference between case and case sensitive. Examples: `$nama`, `$umur`, `$salamat`.
- **Assignment Value:** a value can be assigned to a variable by using the assignment operator (=). Example: `$nama = "John";`.
- **Data Types:** PHP is a programming language whose data types are dynamic, meaning that there is no need to initialize data types when declaring variables. Variables can store different types of data, such as strings, numbers, booleans, arrays, and others.
- **Accessing Variable Values:** variable values can be accessed by using their variable names. Example: `$nama`.

Example of implementing a variable script:

- In the form of a string

```
<?php
$name = "John";
$job = "Programmer";
echo "Hello, my name is $name and I am a $job.";
?>
```

- In the form of numbers

```
<?php
$age = 25;
$height = 175.5;
echo "I am $age years old and I am $height cm tall.";
?>
```

Constants in PHP

A constant is a fixed value that cannot be changed during the course of the program. They are often used to store values that should not change during the execution of a program. Here are a few things about constants in PHP:

- **Constant Declaration:** Constants are declared using the **define() function** with two parameters: the name of the constant and its value. Example: **define("PI", 3.14);**.
- **Constant Naming Rules:** Constants are generally written in capital letters and use an underscore (_) as a word separator if the name of the constant consists of several words. Example: **MAX_VALUE**.
- **Constant Data Types:** Constants can also store different types of data, such as strings, numbers, booleans, or arrays. But, once a constant is defined, its value cannot be changed during the program.
- **Accessing Constants:** You can access constants by using the name of the constant. Example: **echo PI;**

Example of a constant script implementation:

```
<?php
define("PI", 3.14159265);
$radius = 5;
$area = PI * ($radius * $radius);
echo "The area of a circle with a radius of $radius is $area.";
?>
```

Practical Section 1. Variables and Constants

Step	Description
1	Create a new file in the week4 folder and name it variabelkonstanta.php .
2	Write the following code in the variabelkonstanta.php file
3	<pre>1 <?php 2 \$angka1 = 10; 3 \$angka2 = 5; 4 \$hasil = \$angka1 + \$angka2; 5 echo "Hasil penjumlahan \$angka1 dan \$angka2 adalah \$hasil."; 6 ?></pre>
4	Add the contents of the variabelkonstanta.php file with the code below.
5	<pre>7 \$benar = true; 8 \$salah = false; 9 echo "Variabel benar: \$benar, Variabel salah: \$salah"; 10 ?></pre>
6	Add the contents of the variabelkonstanta.php file with the code below.

7	<pre> 11 // Mendefinisikan konstanta untuk nilai tetap 12 define("NAMA_SITUS", "WebsiteKu.com"); 13 define("TAHUN_PENDIRIAN", 2023); 14 15 echo "Selamat datang di " . NAMA_SITUS . ", situs yang didirikan pada tahun " . TAHUN_PENDIRIAN . "."; 16 ?> </pre>
8	Save the file, then open a browser and run localhost/week4/variabelkonstanta .
9	What do you understand from the use of variables on the file? Record your understanding below. (Question No. 1)

Data Types in PHP

A data type is a classification of a type of data or a form of data. There are a wide variety of data types ranging from native data types to artificial data types. PHP supports at least eight scalar data types. The eight types of data are:

Data Type	Description
Integer	Contains whole numbers.
Float	Contains decimal numbers.
Boolean	Contains only 2 values: true and false.
String	Contains text data enclosed by a " or " ' ' sign.
Array	Contains a dataset.
Object	Used to create objects from predefined classes.
NULL	It is used to state that a variable has no value.
Callable	Used to store a reference to a callable function or method.

Practicum 2: Use of Data Types

Step	Description
1	Create a new file named tipedata.php . Type the code in step 2 inside the tipedata.php

2	<pre> 1 <?php 2 \$a = 10; 3 \$b = 5; 4 \$c = \$a + 5; 5 \$d = \$b + (10 * 5); 6 \$e = \$d - \$c; 7 8 echo "Variabel a: {\$a}
"; 9 echo "Variabel b: {\$b}
"; 10 echo "Variabel c: {\$c}
"; 11 echo "Variabel d: {\$d}
"; 12 echo "Variabel e: {\$e}
"; 13 14 var_dump(\$e); 15 ?> </pre>
3	Save the file, then open a browser and run localhost/week4/tipedata.php
4	Explain what you understand from the code below. (Question No. 2)
5	<p>Add the contents of the <code>tipedata.php</code> file with the code below.</p> <pre> 16 \$nilaiMatematika = 5.1; 17 \$nilaiIPA = 6.7; 18 \$nilaiBahasaIndonesia = 9.3; 19 20 \$rataRata = (\$nilaiMatematika + \$nilaiIPA + \$nilaiBahasaIndonesia) / 3; 21 22 echo "Matematika: {\$nilaiMatematika}
"; 23 echo "IPA: {\$nilaiIPA}
"; 24 echo "Bahasa Indonesia: {\$nilaiBahasaIndonesia}
"; 25 echo "Rata-rata: {\$rataRata}
"; 26 27 var_dump(\$rataRata); 28 ?> </pre>
6	Save the file, then open a browser and run localhost/week4/tipedata.php
7	Explain what you understand from the code below. (Question No. 3)
8	<p>Add the contents of the <code>tipedata.php</code> file with the code below.</p> <pre> 29 \$apakahSiswaLulus = true; 30 \$apakahSiswaSudahUjian = false; 31 32 var_dump(\$apakahSiswaLulus); 33 echo "
"; 34 var_dump(\$apakahSiswaSudahUjian); 35 ?> </pre>

9	Save the file, then open a browser and run <code>localhost/week4/tipedata.php</code>
10	Explain what you understand from the code below. (Question No. 4)
11	<p>Add the contents of the <code>tipedata.php</code> file with the code below.</p> <pre> 36 \$namaDepan = "Ibnu"; 37 \$namaBelakang = 'Jakaria'; 38 39 \$namaLengkap = "{\$namaDepan} {\$namaBelakang}"; 40 \$namaLengkap2 = \$namaDepan . ' ' . \$namaBelakang; 41 42 echo "Nama Depan: {\$namaDepan}
"; 43 echo 'Nama Belakang: ' . \$namaBelakang . '
'; 44 45 echo \$namaLengkap; 46 ?> </pre>
12	Save the file, then open a browser and run <code>localhost/week4/tipedata.php</code>
13	Explain what you understand from the code below. (Question No. 5)
14	<p>Add the contents of the <code>tipedata.php</code> file with the code below.</p> <pre> 47 \$listMahasiswa = ["Wahid Abdullah", "Elmo Bachtiar", "Lendis Fabri"]; 48 echo \$listMahasiswa[0]; 49 ?> </pre>
15	Save the file, then open a browser and run <code>localhost/week4/tipedata.php</code>
16	Explain what you understand from the code below. (Question No. 6)

Operators in PHP

An operator in programming is a symbol or sign used to perform operations on data. Operators are used to perform various actions such as mathematical operations, comparisons, logic, and more on the values present in the program. We use operators in the PHP programming language when it is necessary to perform basic operations such as data manipulation, value comparison, condition evaluation, and various other actions on the values present in the program. Operators make it possible to change, manipulate, or make decisions based on those values.

Types of Operators in PHP

Here is a table that contains a list of some of the commonly used operators in PHP

Operator Type	Symbol	Description and Examples of Use
Aritmatika	+	Sum: \$hasil = \$angka1 + \$angka2;
	-	Subtraction: \$hasil = \$angka1 - \$angka2;
	*	Multiply: \$hasil = \$angka1 * \$angka2;
	/	Division: \$hasil = \$angka1 / \$angka2;
	%	Remainder Divided: \$sisia = \$angka1 % \$angka2;
Checklists	==	Equal to: \$hasil = \$nilai1 == \$nilai2;
	!=	Not equal to: \$hasil = \$nilai1 != \$nilai2;
	<	Smaller than: \$hasil = \$nilai1 < \$nilai2;
	>	Greater than: \$hasil = \$nilai1 > \$nilai2;
	<=	Less than or equal to: \$hasil = \$nilai1 <= \$nilai2;
	>=	Greater than or equal to: \$hasil = \$nilai1 >= \$nilai2;
Logic	&& / and	AND (dan): \$hasil = \$a && \$b;
	/ or	OR (atau): \$hasil = \$a \$b;
	! / not	NOT (negasi): \$hasil = !\$a;
Assignment	=	Typical assignment: \$a = \$b;
	+=	Assignment with summation: \$a += \$b;
	-=	Assignment with subtraction: \$a -= \$b;
	*=	Assignment by multiplication: \$a *= \$b;
	/=	Assignment by division: \$a /= \$b;
	%=	Assignment with remainder divided by: \$a %= \$b;
Increment/ Decrement	++	Increment (menambah 1): \$a++;
	--	Decrement (mengurangi 1): \$a--;
Bitwise	&	AND bitwise: \$hasil = \$a & \$b;
		OR bitwise: \$hasil = \$a \$b;
	^	XOR bitwise: \$hasil = \$a ^ \$b;
	~	NOT bitwise: \$hasil = ~\$a;
Appointment	**	Rank: \$hasil = \$angka ** \$pangkat;
Identik	===	Identical (values and data types are the same): \$hasil = \$a === \$b;
	!==	Not Identical (values or data types are not the same): \$hasil = \$a !== \$b;
Ternary	? :	Conditional operator (ternary): \$hasil = (\$nilai > 10) ? "Bigger" : "Smaller";
Array	[]	Access array elements: \$nilai = \$array[0];
	+	Array merging: \$array 3 = \$array 1 + \$array 2;

Practical Section 3: The Use of PHP Operators

Step	Description
1	Create a new file named operator.php . Type the code in step 2 inside the operator.php

2	<pre> 1 <?php 2 \$a = 10; 3 \$b = 5; 4 5 \$hasilTambah = \$a + \$b; 6 \$hasilKurang = \$a - \$b; 7 \$hasilKali = \$a * \$b; 8 \$hasilBagi = \$a / \$b; 9 \$sisabagi = \$a % \$b; 10 \$pangkat = \$a ** \$b; </pre>
3	<p>Complete the program code above so that it can display the results and be neat. (Question No. 7)</p>
4	<p>Save the file, then open a browser and run <code>localhost/week4/operator.php</code> . Observe what is happening and explain what you understand. (Question No. 8)</p>
5	<p>Add the following code to <code>the operator.php</code></p> <pre> 12 \$hasilSama = \$a == \$b; 13 \$hasilTidakSama = \$a != \$b; 14 \$hasilLebihKecil = \$a < \$b; 15 \$hasilLebihBesar = \$a > \$b; 16 \$hasilLebihKecilSama = \$a <= \$b; 17 \$hasilLebihBesarSama = \$a >= \$b; 18 </pre>
6	<p>Complete the program code above so that it can display the results and be neat. Save the file, then open a browser and run /refresh <code>localhost/week4/operator.php</code> Observe what is happening and explain what you understand. (Question No. 9)</p>
7	<p>Add the code in step 8 on <code>operator.php</code></p>
8	<pre> 19 \$hasilAnd = \$a && \$b; 20 \$hasilOr = \$a \$b; 21 \$hasilNotA = !\$a; 22 \$hasilNotB = !\$b; </pre>
9	<p>Complete the program code above so that it can display the results and be neat. Save the file, then open a browser and run /refresh <code>localhost/week4/operator.php</code> Observe what is happening and explain what you understand. (Question No. 10)</p>
10	<p>Add the code in step 11 on <code>operator.php</code></p>

11	<pre> 24 \$a += \$b; 25 \$a -= \$b; 26 \$a *= \$b; 27 \$a /= \$b; 28 \$a %= \$b; </pre>
12	<p>Complete the program code above so that it can display the results and be neat. Save the file, then open a browser and run /refresh localhost/week4/operator.php Observe what is happening and explain what you understand. (Question No. 11)</p>
13	<p>Add the code in step 14 on operator.php</p>
14	<pre> 30 \$hasilIdentik = \$a === \$b; 31 \$hasilTidakIdentik = \$a !== \$b; </pre>
15	<p>Complete the program code above so that it can display the results and be neat. Save the file, then open a browser and run /refresh localhost/week4/operator.php Observe what is happening and explain what you understand. (Question No. 12)</p>
16	<p>Question: A restaurant has 45 seats in it. On one night, 28 seats were occupied by customers. What percentage of seats are still empty in the restaurant?</p>
17	<p>Create a file with named latihanoperator.php. Write down the program code for step 16 and display the result below along with the program code (Question No. 13)</p>

Control Structure In PHP

Basically, programming is about controlling the execution flow of a program. A control structure is a tool used to decide how code will be executed based on certain conditions.

Conditional Statement

Conditional statements are used to make decisions based on certain conditions. There are several types of conditional statements in PHP:

- **if:** Used to execute a block of code if certain conditions are true.

```

if ($nilai > 90) {
    echo "Nilai Anda sangat baik!";
}

```

- **else:** Used in conjunction with **if** to execute a block of code if the condition is incorrect.

```
if ($nilai > 90) {
    echo "Nilai Anda sangat baik!";
} else {
    echo "Anda perlu belajar lebih keras.";
}
```

- **elseif:** Used to add additional conditions after **if**.

```
if ($nilai > 90) {
    echo "A+";
} elseif ($nilai > 80) {
    echo "A";
} else {
    echo "Belajar lebih keras.";
}
```

- **switch:** Used to evaluate expressions and execute code based on matching values.

```
switch ($hari) {
    case "Senin":
        echo "Hari kerja.";
        break;
    case "Sabtu":
    case "Minggu":
        echo "Akhir pekan.";
        break;
    default:
        echo "Hari tidak valid.";
}
```

Looping

Looping allows you to iterate through the code multiple times based on certain conditions. There are three types of loops that are most commonly used in PHP:

- **for:** Used to repeat the code a certain number of times.

```
for ($i = 1; $i <= 5; $i++) {
    echo "Iterasi ke-$i <br>";
}
```

- **while:** Used to repeat code as long as a certain condition is true.

```
$counter = 1;
while ($counter <= 5) {
    echo "Iterasi ke-$counter <br>";
    $counter++;
}
```

- **do-while:** Similar to **while**, but will execute a block of code at least once before checking for conditions.

```
$counter = 1;
do {
    echo "Iterasi ke-$counter <br>";
    $counter++;
} while ($counter <= 5);
```

Flow Control Statement

This statement allows you to control the flow of program execution in a certain way:

- **break:** Used to stop the current loop or switch.

```
for ($i = 1; $i <= 10; $i++) {
    if ($i == 5) {
        break; // Menghentikan perulangan saat $i = 5
    }
    echo "Iterasi ke-$i <br>";
}
```

```
Iterasi ke-1
Iterasi ke-2
Iterasi ke-3
Iterasi ke-4
```

- **continue:** Used to resume the next iteration in the loop.

```
for ($i = 1; $i <= 5; $i++) {
    if ($i == 3) {
        continue; // Melanjutkan ke iterasi berikutnya saat $i = 3
    }
    echo "Iterasi ke-$i <br>";
}
```

```
Iterasi ke-1
Iterasi ke-2
Iterasi ke-4
Iterasi ke-5
```

Practical Section 4: The Use of Control Structures in PHP

Step	Description
1	Create a new file named <code>strukturkontrol.php</code> . Type the code in step 2 inside the <code>strukturkontrol.php</code>
2	<pre>1 <?php 2 \$nilaiNumerik = 92; 3 4 if (\$nilaiNumerik >= 90 && \$nilaiNumerik <= 100) { 5 echo "Nilai huruf: A"; 6 } elseif (\$nilaiNumerik >= 80 && \$nilaiNumerik < 90) { 7 echo "Nilai huruf: B"; 8 } elseif (\$nilaiNumerik >= 70 && \$nilaiNumerik < 80) { 9 echo "Nilai huruf: C"; 10 } elseif (\$nilaiNumerik < 70) { 11 echo "Nilai huruf: D"; 12 } 13 ?></pre>
3	Save the file, then open a browser and run /refresh <code>localhost/week4/strukturkontrol.php</code>

4	Note here what you observe from the addition of the program code above. (Question No. 14)
5	Add the code in step 6 in <code>strukturkontrol.php</code>
6	<pre> \$jarakSaatIni = 0; \$jarakTarget = 500; \$peningkatanHarian = 30; \$hari = 0; while (\$jarakSaatIni < \$jarakTarget) { \$jarakSaatIni += \$peningkatanHarian; \$hari++; } echo "Atlet tersebut memerlukan \$hari hari untuk mencapai jarak 500 kilometer.";</pre>
7	Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh <code>localhost/week4/strukturkontrol.php</code>
8	Note here what you observe from the addition of the program code above. (Question No. 15)
9	Add the code in step 9 inside the <code>strukturkontrol.php</code>
10	<pre> \$jumlahLahan = 10; \$tanamanPerLahan = 5; \$buahPerTanaman = 10; \$jumlahBuah = 0; for (\$i = 1; \$i <= \$jumlahLahan; \$i++) { \$jumlahBuah += (\$tanamanPerLahan * \$buahPerTanaman); } echo "Jumlah buah yang akan dipanen adalah: \$jumlahBuah";</pre>
11	Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh <code>localhost/week4/strukturkontrol.php</code>
12	Note here what you observe from the addition of the program code above. (Question No. 16)
13	Add the code in step 14 inside the <code>strukturkontrol.php</code>

14	<pre> \$skorUjian = [85, 92, 78, 96, 88]; \$totalSkor = 0; foreach (\$skorUjian as \$skor) { \$totalSkor += \$skor; } echo "Total skor ujian adalah: \$totalSkor"; </pre>
15	Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh localhost/week4/strukturkontrol.php
16	Note here what you observe from the addition of the program code above. (Question No. 17)
17	Add the code in step 18 inside the strukturkontrol.php
18	<pre> \$nilaiSiswa = [85, 92, 58, 64, 90, 55, 88, 79, 70, 96]; foreach (\$nilaiSiswa as \$nilai) { if (\$nilai < 60) { echo "Nilai: \$nilai (Tidak lulus)
"; continue; } echo "Nilai: \$nilai (Lulus)
"; } </pre>
19	Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh localhost/week4/strukturkontrol.php
20	Note here what you observe from the addition of the program code above. (Question No. 18)
21	Question: A teacher wants to calculate the total score of 10 students in a math exam. This teacher wanted to ignore the two highest grades and the two lowest grades. Help this teacher calculate the total grades that will be used to determine the average grade after ignoring the highest and lowest grades. The following is a list of grades from 10 students (85, 92, 78, 64, 90, 75, 88, 79, 70, 96)
22	Create a file with the name kontroll1.php . Create the program code for step 21 and display the result below along with the program code (Question No. 19)
23	There is a story: A customer wants to buy a product at a price of Rp 120,000. The store offers a 20% discount for purchases above IDR 100,000. Help these customers to calculate the price to be paid after getting the discount.

24	Create a file with the name <code>kontrol2.php</code> . Create the program code for step 23 and display the result below along with the program code (Question No. 20)
25	Question: A gamer wants to calculate their total score in the game. They get a score based on the points they collect. If they have more than 500 points, then they will get additional rewards. Create the first line of "Player's total score is: (points)". And the second line "Do players get additional rewards? (YES/NO)"
26	Create a file with the name <code>kontrol3.php</code> . Create the program code for step 25 show the result below along with the program code (Question No. 21)

Practical Section 5: Using Arrays in PHP

One-Dimensional Array

A one-dimensional array is a data structure that is used to store a number of values in a single variable. You can declare a one-dimensional array in PHP using several methods:

Method 1: Using `array()`

```
$buah = array("apel", "jeruk", "pisang", "mangga");
```

Method 2: Using square `brackets []` (Starting from PHP 5.4+)

```
$buah = ["apel", "jeruk", "pisang", "mangga"];
```

To access the values in the array used indexes (starting from 0 for the first index):

```
echo $buah[0]; // Output: apel
echo $buah[2]; // Output: pisang
```

You can add a value to an array using a specific index or by the `[]` method to add a new value to the end of the array:

```
$buah[] = "anggur"; // Menambahkan "anggur" ke akhir array
$buah[4] = "durian"; // Menambahkan "durian" pada indeks 4
```

You can change the values in an array by accessing its index and replacing it:

```
$buah[1] = "semangka"; // Mengubah nilai "jeruk" menjadi "semangka"
```

You can iterate through all the elements in the array using `for`, `foreach`, or `while` loops. Here's an example with `foreach`:

```
foreach ($buah as $item) {
    echo $item . "<br>";
}
```

Two-Dimensional Array

A two-dimensional array is a data structure used to store data in the form of a two-dimensional table, which is similar to rows and columns in a spreadsheet. Here is a brief explanation of two-dimensional arrays and examples of their implementation in PHP:

1. Two-Dimensional Array Declaration

You can declare a two-dimensional array in PHP by composing a one-dimensional array in another array:

```
$matrix = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
```

You can also use the square bracket syntax `[][]`:

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

2. Accessing Values in a Two-Dimensional Array

You can access values in a two-dimensional array by using two indexes, one for rows and one for columns:

```
echo $matrix[0][0]; // Output: 1 (baris 1, kolom 1)
echo $matrix[1][2]; // Output: 6 (baris 2, kolom 3)
```

3. Adding and Changing Values in a Two-Dimensional Array

You can add or change values in a two-dimensional array by accessing elements using an index:

```
$matrix[1][1] = 10; // Mengubah nilai pada baris 2, kolom 2 menjadi 10
$matrix[2][] = 11;  // Menambahkan nilai 11 ke akhir array pada baris 3
```

4. Iteration Through Two-Dimensional Arrays

You can iterate through all elements in a two-dimensional array using nested loops, such as loop **for**:


```

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

```

This will print the entire element in a two-dimensional array, resulting in a table-like appearance.

```

1 2 3
4 5 6
7 8 9

```

Multidimensional Array

A multidimensional array is a data structure that allows you to store data in more than two dimensions. This is an expansion of the two-dimensional array described earlier.

1. Three-Dimensional Multidimensional Array Declaration

You can declare a three-dimensional multidimensional array by adding an array within an array:

```

$threeDArray = array(
    array(
        array(1, 2, 3),
        array(4, 5, 6)
    ),
    array(
        array(7, 8, 9),
        array(10, 11, 12)
    )
);

```

This is an example of a three-dimensional array. You can also use the square bracket syntax `[][][]`:

```

$threeDArray = [
    [
        [1, 2, 3],
        [4, 5, 6]
    ],
    [
        [7, 8, 9],
        [10, 11, 12]
    ]
];

```

2. Accessing Values in a Three-Dimensional Multidimensional Array

You can access values in a three-dimensional multidimensional array by using three indexes, each for the corresponding dimension level:

```
echo $threeDArray[0][0][0]; // Output: 1 (dimensi 1, dimensi 2, elemen pertama)
echo $threeDArray[1][1][2]; // Output: 12 (dimensi 2, dimensi 2, elemen ketiga)
```

3. Adding and Changing Values in a Three-Dimensional Multidimensional Array

You can add or change values in a three-dimensional multidimensional array by accessing elements using an index:

```
$threeDArray[0][1][1] = 15; // Mengubah nilai pada dimensi 1, dimensi 3, elemen kedua
$threeDArray[1][0][] = 13;  // Menambahkan nilai 13 ke akhir array pada dimensi 1
```

4. Iteration Through Three-Dimensional Multidimensional Arrays

To iterate through a three-dimensional multidimensional array, you need to use three levels of nested loops:

```
foreach ($threeDArray as $dimensi1) {
    foreach ($dimensi1 as $dimensi2) {
        foreach ($dimensi2 as $elemen) {
            echo $elemen . " ";
        }
        echo "<br>";
    }
    echo "<br>";
}
```

In the example above, we use a three-level nested loop to access all the elements in a three-dimensional array and print them.

```
1 2 3
4 5 6

7 8 9
10 11 12
```

Associative Array

An associative array is a type of array in PHP that uses a pair of "keys" and "values" to organize data. Here is an explanation and example of implementing associative arrays in PHP:

1. Associative Array Declaration

You can declare associative arrays by using **the array()** syntax or square **brackets []**. Each element of an associative array has a key that pairs with its value. Here's an example of using associative arrays in PHP:

```
// Menggunakan sintaks array()
$student = array(
    "nama" => "John",
    "usia" => 20,
    "kelas" => "12A"
);

// Menggunakan sintaks tanda kurung siku []
$student = [
    "nama" => "John",
    "usia" => 20,
    "kelas" => "12A"
];
```

2. Accessing Values in Associative Arrays

You can access the values in an associative array by using the appropriate key:

```
echo $student["nama"]; // Output: John
echo $student["usia"]; // Output: 20
echo $student["kelas"]; // Output: 12A
```

3. Adding and Changing Values in Associative Arrays

You can add or change values in an associative array by accessing elements using keys:

```
$student["nilai"] = 95; // Menambahkan nilai baru ke dalam array
$student["usia"] = 21; // Mengubah nilai usia
```

4. Iteration Through Associative Arrays

```
foreach ($student as $key => $value) {
    echo "$key: $value <br>";
}
```

The output of the loop above will print all key-value pairs in the associative array.

```
nama: John
usia: 21
kelas: 12A
nilai: 95
```

Step	Description
1	Create a new file named array.php . Type the code in step 2 inside the array.php

2	<pre> \$nilaiSiswa = [85, 92, 78, 64, 90, 55, 88, 79, 70, 96]; \$nilaiLulus = []; foreach (\$nilaiSiswa as \$nilai) { if (\$nilai >= 70) { \$nilaiLulus[] = \$nilai; } } echo "Daftar nilai siswa yang lulus: " . implode(', ', \$nilaiLulus); </pre>
3	Save the file, then open a browser and run /refresh localhost/week4/array.php
4	Note here what you observe from adding the code above. (Question No. 22)
5	Type the additional code in step 6 inside the array.php
6	<pre> \$daftarKaryawan = [['Alice', 7], ['Bob', 3], ['Charlie', 9], ['David', 5], ['Eva', 6],]; \$karyawanPengalamanLimaTahun = []; foreach (\$daftarKaryawan as \$karyawan) { if (\$karyawan[1] > 5) { \$karyawanPengalamanLimaTahun[] = \$karyawan[0]; } } echo "Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: " . implode(', ', \$karyawanPengalamanLimaTahun); </pre>
7	Save the file, then open a browser and run /refresh localhost/week4/array.php
8	Note here what you observe from the addition of the code above. (Question No. 23)
9	Type the additional code in step 10 inside the array.php

10	<pre> \$daftarNilai = ['Matematika' => [['Alice', 85], ['Bob', 92], ['Charlie', 78],], 'Fisika' => [['Alice', 90], ['Bob', 88], ['Charlie', 75],], 'Kimia' => [['Alice', 92], ['Bob', 80], ['Charlie', 85],],]; \$mataKuliah = 'Fisika'; echo "Daftar nilai mahasiswa dalam mata kuliah \$mataKuliah:
"; foreach (\$daftarNilai[\$mataKuliah] as \$nilai) { echo "Nama: {\$nilai[0]}, Nilai: {\$nilai[1]}
"; } </pre>
11	Save the file, then open a browser and run /refresh localhost/week4/array.php
12	Note here what you observe from the addition of the program code above. (Question No. 24)
13	Question: A teacher wants to print a list of students' grades in a math exam. The teacher has data on each student consisting of names and grades. Help this teacher print a list of students who achieved grades above the class average. With the provision that the student's name and grade Alice can get 85, Bob can get 92, Charlie can get 78, David can get 64, Eva can get 90
14	Create the code for step 13 with a two-dimensional array and display the result below along with the program code (Question No. 25)