



Topic

- Introduction to PHP

Name : Evan Diantha Fafian
Class : SIB 2G
Absent : 09
NIM : 2341760163

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**, **\$alamat**.
- **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 dasarWeb 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	<p>Save the file, then open a browser and run localhost/dasarWeb/variabelkonstanta.php.</p>
9	<p>What do you understand from the use of variables on the file? Record your understanding below. (Question No. 1)</p>  <p>Hasil penjumlahan 10 dan 5 adalah 15. Variabel benar: 1, Variabel salah: Selamat datang di WebsiteKu.com, situs yang didirikan pada tahun 2023.</p> <hr/> <p>- Variables such as “\$angka`, ‘\$angka2` , and ‘\$hasil” are used to store and modify data in those PHP files, with “echo” to display the summation result. The boolean variables “\$benar” and “\$salah” store the values “true” and “false”, printed as “1” and empty respectively. The “define()” function can be used to define constants such as “NAME_SITUS” and “TAHUN_PENDIRIAN” for fixed and global values, and can be used in strings by concatenation. Variables and constants are essential for storing and displaying data in PHP scripts.</p>

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/dasarWeb/tipedata.php
4	<p>Explain what you understand from the code below. (Question No. 2)</p> <pre> < → ⌂ localhost/dasarweb/week%204%20(JS04-PHP)/source%20code/tipedata.php Variabel a: 10 Variabel b: 5 Variabel c: 15 Variabel d: 55 Variabel e: 40 int(40) </pre> <p>- This PHP code declares the variables “\$a” and “\$b”, performs calculations to produce the variables “\$c”, “\$d”, and “\$e”, and displays the value of each with “echo”, separated by the tag “
.” The variable “\$c” is the sum of “\$a” and ”</p>
5	Add the contents of the tipedata.php file with the code below.

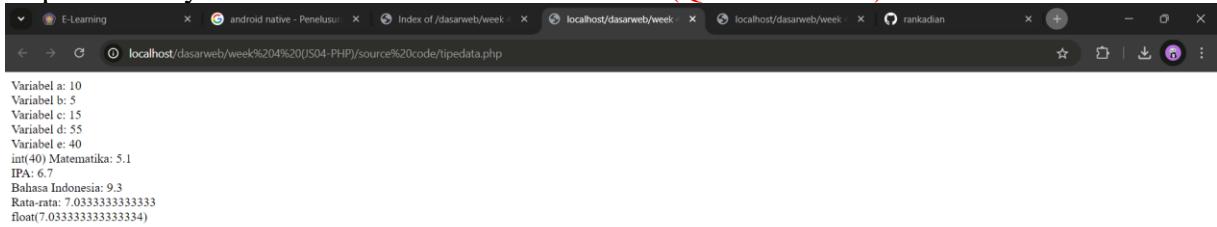
```

16 $nilaiMatematika = 5.1;
17 $nilaiIPA = 6.7;
18 $nilaiBahasaIndonesia = 9.3;
19
20 $rataRata = ($nilaiMatematika + $nilaiIPA + $nilaiBahasaIndonesia) / 3;
21
22 echo "Matematika: {$nilaiMatematika} <br>";
23 echo "IPA: {$nilaiIPA} <br>";
24 echo "Bahasa Indonesia: {$nilaiBahasaIndonesia} <br>";
25 echo "Rata-rata: {$rataRata} <br>";
26
27 var_dump($rataRata);
28 ?>

```

6 Save the file, then open a browser and run localhost/dasarWeb/tipedata.php

Explain what you understand from the code below. (Question No. 3)



```

Variabel a: 10
Variabel b: 5
Variabel c: 15
Variabel d: 55
Variabel e: 40
int(40) Matematika: 5.1
IPA: 6.7
Bahasa Indonesia: 9.3
Rata-rata: 7.033333333333333
float(7.033333333333334)

```

7

- The PHP code will calculate the average scores of Math, Bahasa Indonesia, and Science and display the results along with the scores of each subject if this additional code is added. The variables “\$nilaiMatematika”, “\$nilaiIPA”, and “\$nilaiBahasaIndonesia” store float values, and “\$rataRata” calculates the average value of the three values. The value and average are displayed with “echo” respectively, and the data type and average value are displayed as floats with “var_dump(\$rataRata).“

Add the contents of the tippedata.php file with the code below.

```

29 $apakahSiswaLulus = true;
30 $apakahSiswaSudahUjian = false;
31
32 var_dump($apakahSiswaLulus);
33 echo "<br>";
34 var_dump($apakahSiswaSudahUjian);
35 ?>

```

8

Save the file, then open a browser and run localhost/dasarWeb/tipedata.php

Explain what you understand from the code below. (Question No. 4)

```
Variabel a: 10
Variabel b: 5
Variabel c: 15
Variabel d: 55
Variabel e: 40
int(40) Matematika: 5.1
IPA: 6.7
Bahasa Indonesia: 9.3
Rata-rata: 7.03333333333333
float(7.03333333333334) bool(true)
bool(false)
```

10

- PHP will display the data type and value of two boolean variables, “\$ApakahSiswaLulus” which has a value of “true” and “\$ApakahSiswaSudahUjian” which has a value of “false”. The boolean value “true” will be displayed after a new line, and the boolean value “false” will be displayed by “var_dump”.

Add the contents of the tipedata.php file with the code below.

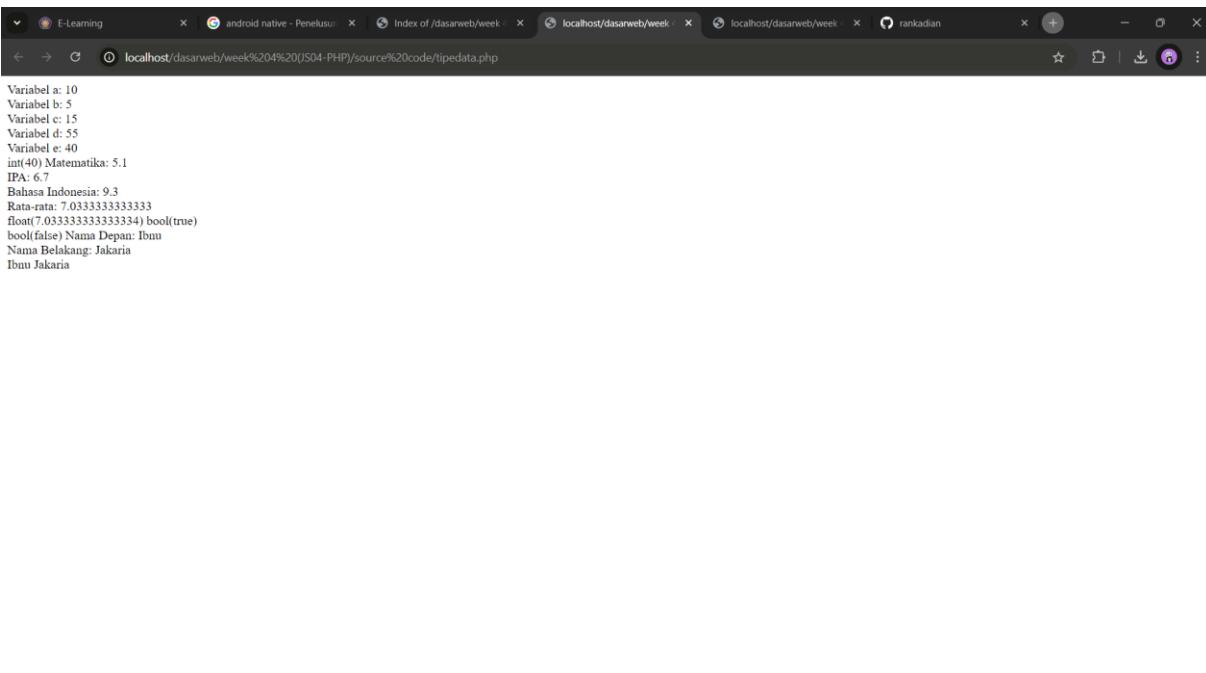
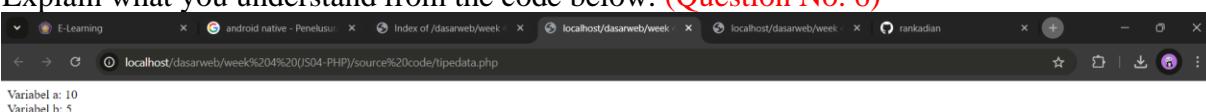
```
36 $namaDepan = "Ibnu";
37 $namaBelakang = 'Jakaria';
38
39 $namaLengkap = "{$namaDepan} {$namaBelakang}";
40 $namaLengkap2 = $namaDepan . ' ' . $namaBelakang;
41
42 echo "Nama Depan: {$namaDepan} <br>";
43 echo 'Nama Belakang: ' . $namaBelakang . '<br>';
44
45 echo $namaLengkap;
46 ?>
```

11

Save the file, then open a browser and run localhost/dasarWeb/tipedata.php

12

Explain what you understand from the code below. (Question No. 5)

	 <p>Variabel a: 10 Variabel b: 5 Variabel c: 15 Variabel d: 55 Variabel e: 40 int(40) Matematika: 5.1 IPA: 6.7 Bahasa Indonesia: 9.3 Rata-rata: 7.033333333333334 float(7.033333333333334) bool(true) bool(false) Nama Depan: Ibnu Nama Belakang: Jakarta Ibnu Jakarta</p>
-	The PHP code will declare and concatenate the variables “\$namaDepan” and “\$namaBelakang” to produce “\$namaLengkap”, which will be displayed. In the first line, there is “Nama Depan: Ibnu”, “Nama Belakang: Jakarta”, and ‘Ibnu Jakarta’ on the next line.
14	Add the contents of the tipedata.php file with the code below. <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 localhost/dasarWeb/tipedata.php
16	<p>Explain what you understand from the code below. (Question No. 6)</p>  <p>Variabel a: 10 Variabel b: 5 Variabel c: 15 Variabel d: 55 Variabel e: 40 int(40) Matematika: 5.1 IPA: 6.7 Bahasa Indonesia: 9.3 Rata-rata: 7.033333333333334 float(7.033333333333334) bool(true) bool(false) Nama Depan: Ibnu Nama Belakang: Jakarta Ibnu JakartaWahid Abdullah</p> <p>- This code is to declare an array “\$listMahasiswa” which contains three student names. Then, “echo \$listStudent[0];” will display the first element of the array, which is “Wahid Abdullah.”</p>

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.

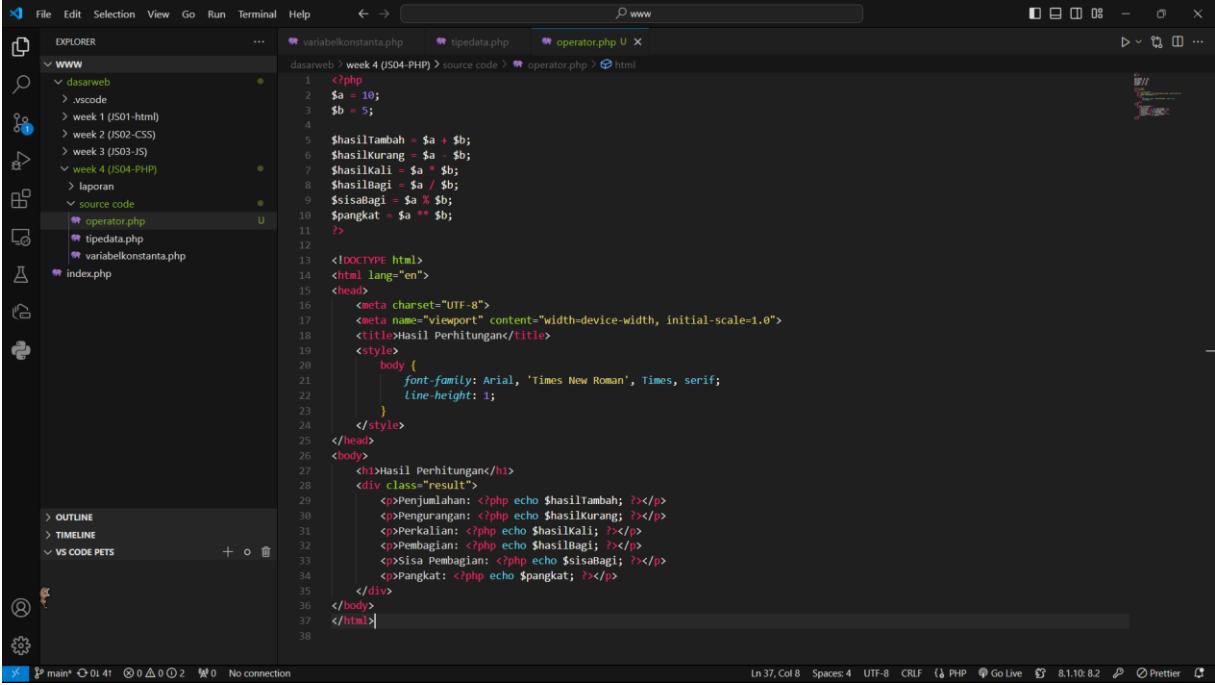
Types of Operators in PHP

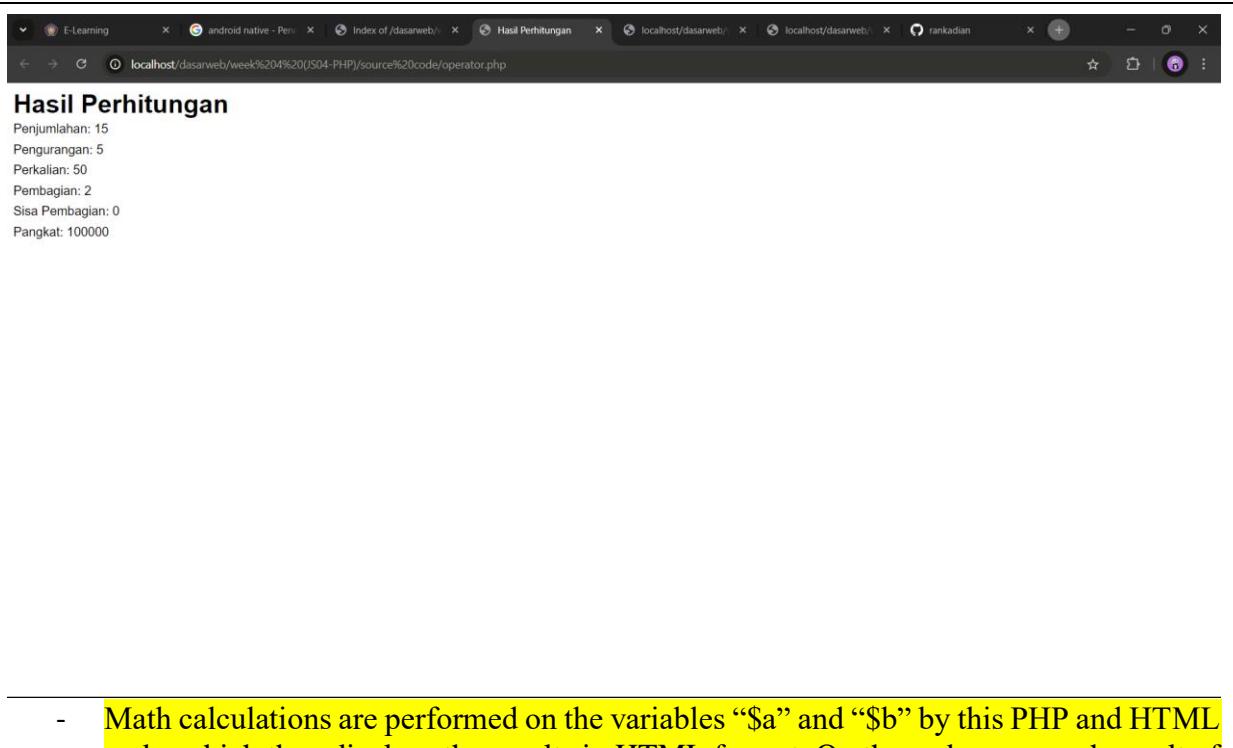
Here is a table that contains a list of some of the commonly used operators in PHP

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

Operator Type	Symbol	Description and Examples of Use
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.</p> <p>(Question No. 7)</p>  <pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Hasil Perhitungan</title> <style> body { font-family: Arial, 'Times New Roman', Times, serif; line-height: 1; } </style> </head> <body> <h1>Hasil Perhitungan</h1> <div class="result"> <p>Penjumlahan: <?php echo \$hasilTambah; ?></p> <p>Pengurangan: <?php echo \$hasilKurang; ?></p> <p>Perkalian: <?php echo \$hasilKali; ?></p> <p>Pembagian: <?php echo \$hasilBagi; ?></p> <p>Sisa Pembagian: <?php echo \$sisaBagi; ?></p> <p>Pangkat: <?php echo \$pangkat; ?></p> </div> </body> </html> </pre>
4	Save the file, then open a browser and run localhost/dasarWeb/operator.php . Observe what is happening and explain what you understand. (Question No. 8)



- Math calculations are performed on the variables “\$a” and “\$b” by this PHP and HTML code, which then displays the results in HTML format. On the web page, each result of addition, subtraction, multiplication, division, remainder of division, and power is displayed in a separate paragraph. Using the CSS style used, the page font is set to Arial with a line spacing of 10px.

	Add the following code to the operator.php
5	<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	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/basicWeb/operator.php Observe what is happening and explain what you understand. (Question No. 9)

```

1  <?php
2 // 1
3 $a = 10;
4 $b = 5;
5
6 $hasilTambah = $a + $b;
7 $hasilKurang = $a - $b;
8 $hasilKali = $a * $b;
9 $hasilBagi = $a / $b;
10 $sisaBagi = $a % $b;
11 $pangkat = $a ** $b;
12
13 // 2
14 $hasilSama = $a == $b;
15 $hasilTidakSama = $a != $b;
16 $hasilLebihKecil = $a < $b;
17 $hasilLebihBesar = $a > $b;
18 $hasilLebihKecilSama = $a <= $b;
19 $hasilLebihBesarSama = $a >= $b;
20 ?>
21
22 <!DOCTYPE html>
23 <html lang="en">
24
25 <head>
26     <meta charset="UTF-8">
27     <meta name="viewport" content="width=device-width, initial-scale=1.0">
28     <title>Hasil Perhitungan</title>
29     <style>
30         body {
31             font-family: Arial, 'Times New Roman', Times, serif;
32             line-height: 10px;
33         }
34     </style>
35 </head>
36
37 <body>
38     <h1>Hasil Perhitungan</h1>
39     <div class="result">
40         <h3>Operasi Matematika</h3>
41         <p>Penjumlahan: <?php echo $hasilTambah; ?></p>
42         <p>Pengurangan: <?php echo $hasilKurang; ?></p>
43         <p>Perkalian: <?php echo $hasilKali; ?></p>
44         <p>Pembagian: <?php echo $hasilBagi; ?></p>
45         <p>Sisa Pembagian: <?php echo $sisaBagi; ?></p>
46         <p>Pangkat: <?php echo $pangkat; ?></p>
47
48         <h3>Perbandingan</h3>
49         <p>Sama: <?php echo $hasilSama ? 'True' : 'False'; ?></p>
50         <p>Tidak Sama: <?php echo $hasilTidakSama ? 'True' : 'False'; ?></p>
51         <p>Lebih Kecil: <?php echo $hasilLebihKecil ? 'True' : 'False'; ?></p>
52         <p>Lebih Besar: <?php echo $hasilLebihBesar ? 'True' : 'False'; ?></p>
53         <p>Lebih Kecil Sama: <?php echo $hasilLebihKecilSama ? 'True' : 'False'; ?></p>
54         <p>Lebih Besar Sama: <?php echo $hasilLebihBesarSama ? 'True' : 'False'; ?></p>
55     </div>
56 </body>
57
58 </html>

```

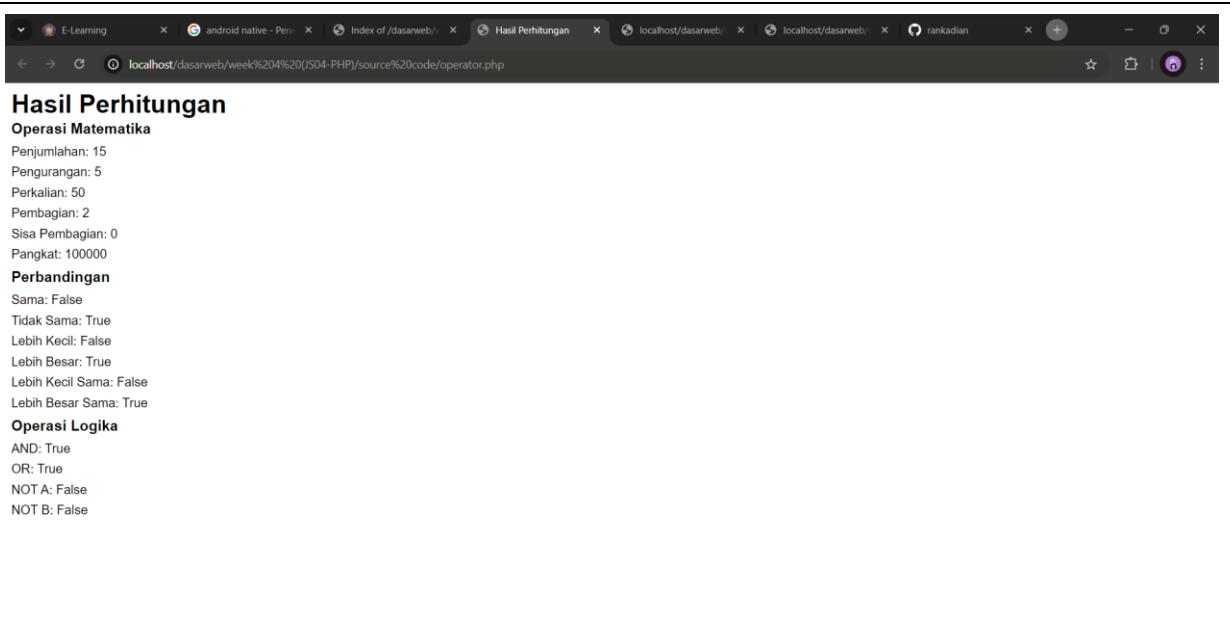
	<pre> Hasil Perhitungan Operasi Matematika Penjumlahan: 15 Pengurangan: 5 Perkalian: 50 Pembagian: 2 Sisa Pembagian: 0 Pangkat: 100000 Perbandingan Sama: False Tidak Sama: True Lebih Kecil: False Lebih Besar: True Lebih Kecil Sama: False Lebih Besar Sama: True </pre>
	<ul style="list-style-type: none"> - An HTML section is added to display the results of math calculations and comparisons in a neat format. The result of the comparison operation is displayed as "True" or "False". The webpage will display the results of math calculation and comparison separately with this addition.

7	Add the code in step 8 on operator.php
8	<pre> 19 \$hasilAnd = \$a && \$b; 20 \$hasilOr = \$a \$b; 21 \$hasilNotA = !\$a; 22 \$hasilNotB = !\$b; </pre>
9	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/basicWeb/operator.php . Observe what is happening and explain what you understand. (Question No. 10)

```

1  <?php
2 // 1
3 $a = 10;
4 $b = 5;
5
6 $hasilTambah = $a + $b;
7 $hasilKurang = $a - $b;
8 $hasilKali = $a * $b;
9 $hasilBagi = $a / $b;
10 $sisaBagi = $a % $b;
11 $pangkat = $a ** $b;
12
13 // 2
14 $hasilSama = $a == $b;
15 $hasilTidakSama = $a != $b;
16 $hasilLebihKecil = $a < $b;
17 $hasilLebihBesar = $a > $b;
18 $hasilLebihKecilSama = $a <= $b;
19 $hasilLebihBesarSama = $a >= $b;
20 ?>
21
22 <!DOCTYPE html>
23 <html lang="en">
24
25 <head>
26     <meta charset="UTF-8">
27     <meta name="viewport" content="width=device-width, initial-scale=1.0">
28     <title>Hasil Perhitungan</title>
29     <style>
30         body {
31             font-family: Arial, 'Times New Roman', Times, serif;
32             line-height: 10px;
33         }
34     </style>
35 </head>
36
37 <body>
38     <h1>Hasil Perhitungan</h1>
39     <div class="result">
40         <h3>Operasi Matematika</h3>
41         <p>Penjumlahan: <?php echo $hasilTambah; ?></p>
42         <p>Pengurangan: <?php echo $hasilKurang; ?></p>
43         <p>Perkalian: <?php echo $hasilKali; ?></p>
44         <p>Pembagian: <?php echo $hasilBagi; ?></p>
45         <p>Sisa Pembagian: <?php echo $sisaBagi; ?></p>
46         <p>Pangkat: <?php echo $pangkat; ?></p>
47
48         <h3>Perbandingan</h3>
49         <p>Sama: <?php echo $hasilSama ? 'True' : 'False'; ?></p>
50         <p>Tidak Sama: <?php echo $hasilTidakSama ? 'True' : 'False'; ?></p>
51         <p>Lebih Kecil: <?php echo $hasilLebihKecil ? 'True' : 'False'; ?></p>
52         <p>Lebih Besar: <?php echo $hasilLebihBesar ? 'True' : 'False'; ?></p>
53         <p>Lebih Kecil Sama: <?php echo $hasilLebihKecilSama ? 'True' : 'False'; ?></p>
54         <p>Lebih Besar Sama: <?php echo $hasilLebihBesarSama ? 'True' : 'False'; ?></p>
55     </div>
56 </body>
57
58 </html>

```



```

Hasil Perhitungan
Operasi Matematika
Penjumlahan: 15
Pengurangan: 5
Perkalian: 50
Pembagian: 2
Sisa Pembagian: 0
Pangkat: 100000
Perbandingan
Sama: False
Tidak Sama: True
Lebih Kecil: False
Lebih Besar: True
Lebih Kecil Sama: False
Lebih Besar Sama: True
Operasi Logika
AND: True
OR: True
NOT A: False
NOT B: False

```

- This code complements web pages by adding the results of logical operations (AND, OR, and NOT) below the results of math calculations and comparisons. The results of logical operations such as `&&` (AND), `||` (OR), `!$a` (NOT A), and `!$b` (NOT B) are displayed as “True” or “False” based on the boolean value of the operation result.

10	Add the code in step 11 on operator.php
11	<pre> 24 \$a += \$b; 25 \$a -= \$b; 26 \$a *= \$b; 27 \$a /= \$b; 28 \$a %= \$b; </pre>
12	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/basicWeb/operator.php Observe what is happening and explain what you understand. (Question No. 11)

```
1  <?php
2 // 1
3 $a = 10;
4 $b = 5;
5
6 $hasilTambah = $a + $b;
7 $hasilKurang = $a - $b;
8 $hasilKali = $a * $b;
9 $hasilBagi = $a / $b;
10 $sisaBagi = $a % $b;
11 $pangkat = $a ** $b;
12
13 // 2
14 $hasilSama = $a == $b;
15 $hasilTidakSama = $a != $b;
16 $hasilLebihKecil = $a < $b;
17 $hasilLebihBesar = $a > $b;
18 $hasilLebihKecilSama = $a <= $b;
19 $hasilLebihBesarSama = $a >= $b;
20
21 // 3
22 $hasilAnd = $a && $b;
23 $hasilOr = $a || $b;
24 $hasilNotA = !$a;
25 $hasilNotB = !$b;
26
27 // 4
28 $a = 10;
29 $a += $b;
30 $hasilTambah = $a;
31 $a -= $b;
32 $hasilKurang = $a;
33 $a *= $b;
34 $hasilKali = $a;
35 $a /= $b;
36 $hasilBagi = $a;
37 $a %= $b;
38 $hasilModulus = $a;
39 ?>
40
41 <!DOCTYPE html>
42 <html lang="en">
43
44 <head>
45     <meta charset="UTF-8">
46     <meta name="viewport" content="width=device-width, initial-scale=1.0">
47     <title>Hasil Perhitungan</title>
48     <style>
49         body {
50             font-family: Arial, 'Times New Roman', Times, serif;
51             line-height: 10px;
52         }
53     </style>
54 </head>
55
56 <body>
57     <h1>Hasil Perhitungan</h1>
58     <div class="result">
59         <!-- 1 -->
60         <h3>Operasi Matematika</h3>
61         <p>Penjumlahan: <?php echo $hasilTambah; ?></p>
62         <p>Pengurangan: <?php echo $hasilKurang; ?></p>
63         <p>Perkalian: <?php echo $hasilKali; ?></p>
64         <p>Pembagian: <?php echo $hasilBagi; ?></p>
65         <p>Sisa Pembagian: <?php echo $sisaBagi; ?></p>
66         <p>Pangkat: <?php echo $pangkat; ?></p>
67
68         <!-- 2 -->
69         <h3>Perbandingan</h3>
70         <p>Sama: <?php echo $hasilSama ? 'True' : 'False'; ?></p>
71         <p>Tidak Sama: <?php echo $hasilTidakSama ? 'True' : 'False'; ?></p>
72         <p>Lebih Kecil: <?php echo $hasilLebihKecil ? 'True' : 'False'; ?></p>
73         <p>Lebih Besar: <?php echo $hasilLebihBesar ? 'True' : 'False'; ?></p>
74         <p>Lebih Kecil Sama: <?php echo $hasilLebihKecilSama ? 'True' : 'False'; ?></p>
75         <p>Lebih Besar Sama: <?php echo $hasilLebihBesarSama ? 'True' : 'False'; ?></p>
76
77         <!-- 3 -->
78         <h3>Operasi Logika</h3>
79         <p>AND: <?php echo $hasilAnd ? 'True' : 'False'; ?></p>
80         <p>OR: <?php echo $hasilOr ? 'True' : 'False'; ?></p>
81         <p>NOT A: <?php echo $hasilNotA ? 'True' : 'False'; ?></p>
82         <p>NOT B: <?php echo $hasilNotB ? 'True' : 'False'; ?></p>
83
84         <!-- 4 -->
85         <h3>Operasi Penugasan</h3>
86         <p>Setelah A += B: <?php echo $hasilTambah; ?></p>
87         <p>Setelah A -= B: <?php echo $hasilKurang; ?></p>
88         <p>Setelah A *= B: <?php echo $hasilKali; ?></p>
89         <p>Setelah A /= B: <?php echo $hasilBagi; ?></p>
90         <p>Setelah A %= B: <?php echo $hasilModulus; ?></p>
91     </div>
92 </body>
93
94 </html>
```

-	The results of the code are various types of math, comparison, logic, and assignment operations. After the reset, \$a is used to perform an assignment operation that stores the result in a new variable and is displayed in the HTML.

13	Add the code in step 14 on operator.php
14	<pre>30 \$hasilIdentik = \$a === \$b; 31 \$hasilTidakIdentik = \$a !== \$b;</pre>
15	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/basicWeb/operator.php . Observe what is happening and explain what you understand. (Question No. 12)

```

1 <?php
2 // 1
3 $a = 10;
4 $b = 5;
5
6 $hasilTambah = $a + $b;
7 $hasilKurang = $a - $b;
8 $hasilKali = $a * $b;
9 $hasilBagi = $a / $b;
10 $sisaBagi = $a % $b;
11 $pangkat = $a ** $b;
12
13 // 2
14 $hasilSama = $a == $b;
15 $hasilTidakSama = $a != $b;
16 $hasilLebihKecil = $a < $b;
17 $hasilLebihBesar = $a > $b;
18 $hasilLebihKecilSama = $a <= $b;
19 $hasilLebihBesarSama = $a >= $b;
20
21 // 3
22 $hasilAnd = $a && $b;
23 $hasilOr = $a || $b;
24 $hasilNotA = !$a;
25 $hasilNotB = !$b;
26
27 // 4
28 $a = 10;
29 $a += $b;
30 $hasilTambah = $a;
31 $a -= $b;
32 $hasilKurang = $a;
33 $a *= $b;
34 $hasilKali = $a;
35 $a /= $b;
36 $hasilBagi = $a;
37 $a %= $b;
38 $hasilModulus = $a;
39
40 // 5
41 $hasilIdentik = $a === $b;
42 $hasilTidakIdentik = $a !== $b;
43 ?>
44
45 <!DOCTYPE html>
46 <html lang="en">
47
48 <head>
49     <meta charset="UTF-8">
50     <meta name="viewport" content="width=device-width, initial-scale=1.0">
51     <title>Hasil Perhitungan</title>
52     <style>
53         body {
54             font-family: Arial, 'Times New Roman', Times, serif;
55             line-height: 10px;
56         }
57     </style>
58 </head>
59
60 <body>
61     <h1>Hasil Perhitungan</h1>
62     <div class="result">
63         <!-- 1 -->
64         <h3>Operasi Matematika</h3>
65         <p>Penjumlahan: <?php echo $hasilTambah; ?></p>
66         <p>Pengurangan: <?php echo $hasilKurang; ?></p>
67         <p>Perkalian: <?php echo $hasilKali; ?></p>
68         <p>Pembagian: <?php echo $hasilBagi; ?></p>
69         <p>Sisa Pembagian: <?php echo $sisaBagi; ?></p>
70         <p>Pangkat: <?php echo $pangkat; ?></p>
71
72         <!-- 2 -->
73         <h3>Perbandingan</h3>
74         <p>Sama: <?php echo $hasilSama ? 'True' : 'False'; ?></p>
75         <p>Tidak Sama: <?php echo $hasilTidakSama ? 'True' : 'False'; ?></p>
76         <p>Lebih Kecil: <?php echo $hasilLebihKecil ? 'True' : 'False'; ?></p>
77         <p>Lebih Besar: <?php echo $hasilLebihBesar ? 'True' : 'False'; ?></p>
78         <p>Lebih Kecil Sama: <?php echo $hasilLebihKecilSama ? 'True' : 'False'; ?></p>
79         <p>Lebih Besar Sama: <?php echo $hasilLebihBesarSama ? 'True' : 'False'; ?></p>
80
81         <!-- 3 -->
82         <h3>Operasi Logika</h3>
83         <p>AND: <?php echo $hasilAnd ? 'True' : 'False'; ?></p>
84         <p>OR: <?php echo $hasilOr ? 'True' : 'False'; ?></p>
85         <p>NOT A: <?php echo $hasilNotA ? 'True' : 'False'; ?></p>
86         <p>NOT B: <?php echo $hasilNotB ? 'True' : 'False'; ?></p>
87
88         <!-- 4 -->
89         <h3>Operasi Penugasan</h3>
90         <p>Setelah A += B: <?php echo $hasilTambah; ?></p>
91         <p>Setelah A -= B: <?php echo $hasilKurang; ?></p>
92         <p>Setelah A *= B: <?php echo $hasilKali; ?></p>
93         <p>Setelah A /= B: <?php echo $hasilBagi; ?></p>
94         <p>Setelah A %= B: <?php echo $hasilModulus; ?></p>
95
96         <!-- 5 -->
97         <h3>Perbandingan Identik</h3>
98         <p>Identik: <?php echo $hasilIdentik ? 'True' : 'False'; ?></p>
99         <p>Tidak Identik: <?php echo $hasilTidakIdentik ? 'True' : 'False'; ?></p>
100    </div>
101 </body>
102
103 </html>

```

Hasil Perhitungan

Operasi Matematika

- Penjumlahan: 15
- Pengurangan: 10
- Perkalian: 50
- Pembagian: 10
- Sisa Pembagian: 0
- Pangkat: 100000

Perbandingan

- Sama: False
- Tidak Sama: True
- Lebih Kecil: False
- Lebih Besar: True
- Lebih Kecil Sama: False
- Lebih Besar Sama: True

Operasi Logika

- AND: True
- OR: True
- NOT A: False
- NOT B: False

Operasi Penugasan

- Setelah A += B: 15
- Setelah A -= B: 10
- Setelah A *= B: 50
- Setelah A /= B: 10
- Setelah A %= B: 0

Perbandingan Identik

- Identik: False
- Tidak Identik: True

- The results of the code are various types of math, comparison, logic, and assignment operations. After the reset, \$a is used to perform an assignment operation that stores the result in a new variable and is displayed in the HTML.

16

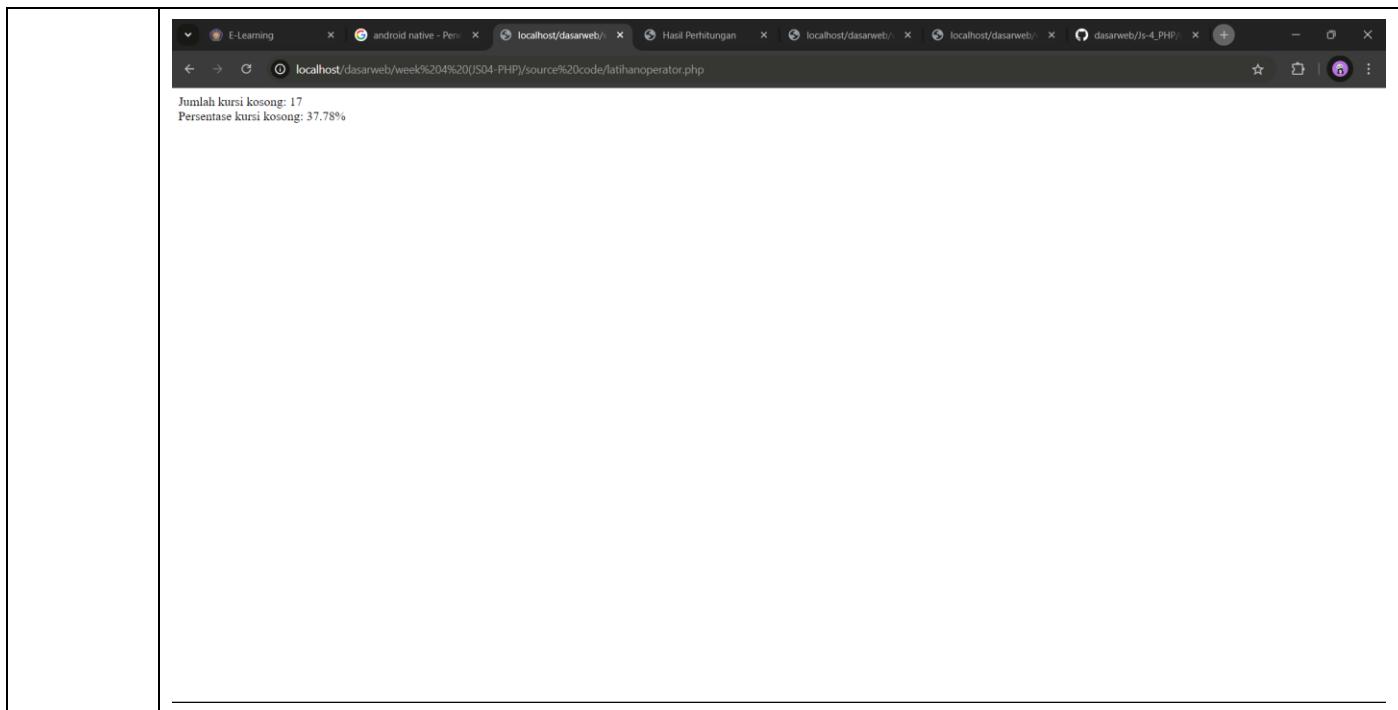
Question: A restaurant has 45 seats in it. On one night, 28 seats were occupied by customers. What percentage of seats are still empty in the restaurant?

Answer: On a given night, a 45-seat restaurant has only 17 empty seats, 28 of which are full, so the percentage of empty seats is 37.78%. The formula for the percentage of empty seats can be calculated by dividing the number of empty seats from the total seats and then multiplying by 100%. The results of this calculation show that the percentage of empty seats in the restaurant is around 37.78%.

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)

17

```
<?php
$totalKursi = 45;
$kursiTerisi = 28;
$kursiKosong = $totalKursi - $kursiTerisi;
$persentaseKosong = ($kursiKosong / $totalKursi) * 100;
echo "Jumlah kursi kosong: $kursiKosong<br>";
echo "Percentase kursi kosong: " . number_format(num: $persentaseKosong, decimals: 2) . "%";
?>
```



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.

```

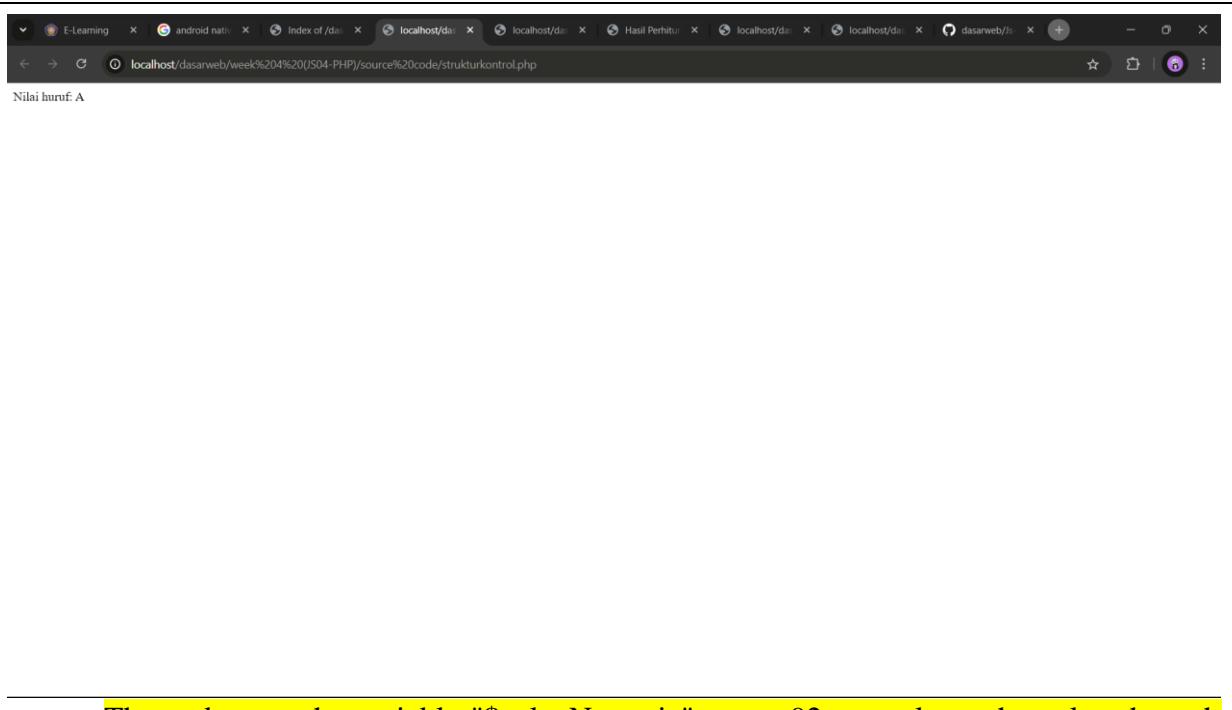
for ($i = 1; $i <= 5; $i++) {
    if ($i == 3) {
        continue; // Melanjutkan ke iterasi berikutnya saat $i = 3
    }
    echo "Iterasi ke-$i <br>";
}

```

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

Practical Section 4: The Use of Control Structures in PHP

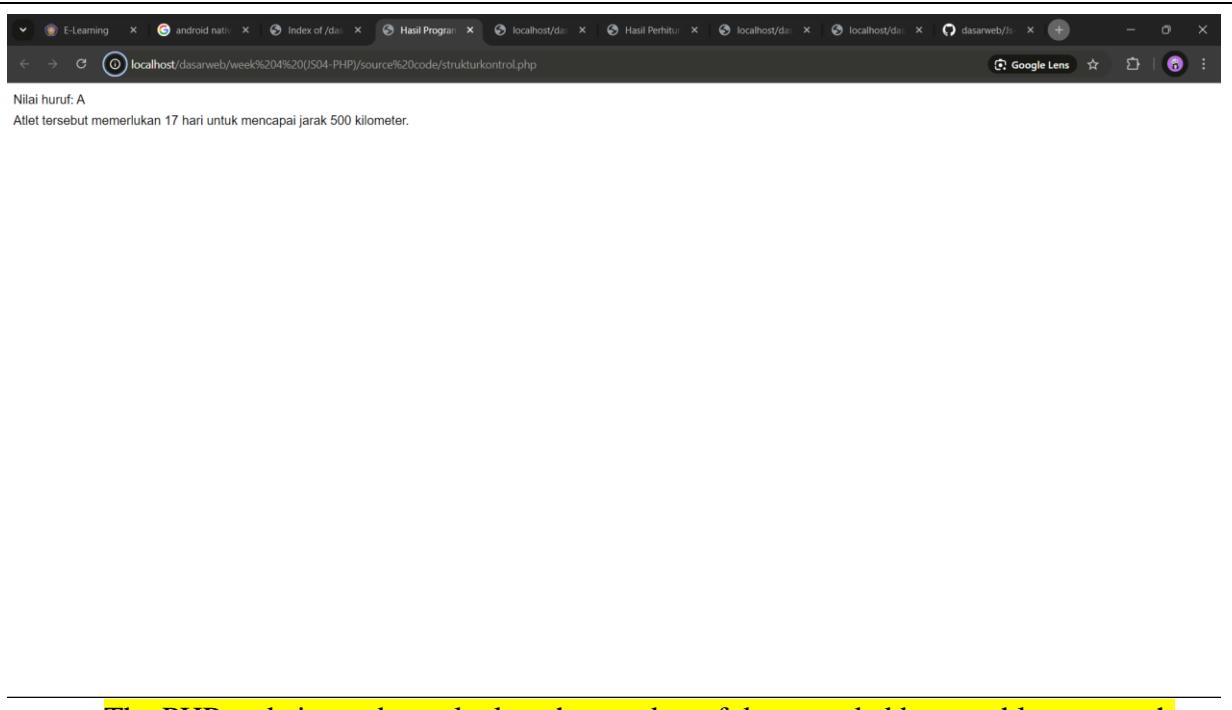
Step	Description
1	Create a new file named strukturkontrol.php . Type the code in step 2 inside the strukturkontrol.php
2	<pre> 1 <?php 2 \$nilaiNumerik = 92; 3 4 if (\$nilaiNumerik >= 90 && \$nilaiNumerik <= 100) { 5 echo "Nilai huruf: A"; 6 } elseif (\$nilaiNumerik >= 80 && \$nilaiNumerik < 90) { 7 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 localhost/dasarWeb/strukturkontrol.php
4	Note here what you observe from the addition of the program code above. (Question No. 14)



- The code uses the variable "\$valueNumeric" set to 92 to evaluate the value through various "if" and "elseif" conditions. This code checks the conditions sequentially to ensure only one "echo" statement is executed according to the appropriate range, and the result is "Letter grade: A" if the numeric value is in the range 90 to 100. The value 92 qualifies for the range "A", so the result is "Letter grade: A".

5	Add the code in step 6 in strukturkontrol.php
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 localhost/dasarWeb/strukturkontrol.php
8	Note here what you observe from the addition of the program code above. (Question No. 15)

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hasil Program</title>
8      <style>
9          body {
10              font-family: Arial, 'Times New Roman', Times, serif;
11              line-height: 10px;
12          }
13      </style>
14  </head>
15
16 <body>
17     <div class="result">
18         <?php
19             $nilaiNumerik = 92;
20
21             if ($nilaiNumerik >= 90 && $nilaiNumerik <= 100) {
22                 echo "<p>Nilai huruf: A</p>";
23             } elseif ($nilaiNumerik >= 80 && $nilaiNumerik < 90) {
24                 echo "<p>Nilai huruf: B</p>";
25             } elseif ($nilaiNumerik >= 70 && $nilaiNumerik < 80) {
26                 echo "<p>Nilai huruf: C</p>";
27             } elseif ($nilaiNumerik < 70) {
28                 echo "<p>Nilai huruf: D</p>";
29             }
30
31             $jarakSaatIni = 0;
32             $jarakTarget = 500;
33             $peningkatanHarian = 30;
34             $hari = 0;
35
36             while ($jarakSaatIni < $jarakTarget) {
37                 $jarakSaatIni += $peningkatanHarian;
38                 $hari++;
39             }
40
41             echo "<p>Atlet tersebut memerlukan $hari hari untuk mencapai jarak 500 kilometer.</p>";
42         ?>
43     </div>
44  </body>
45
46 </html>
```



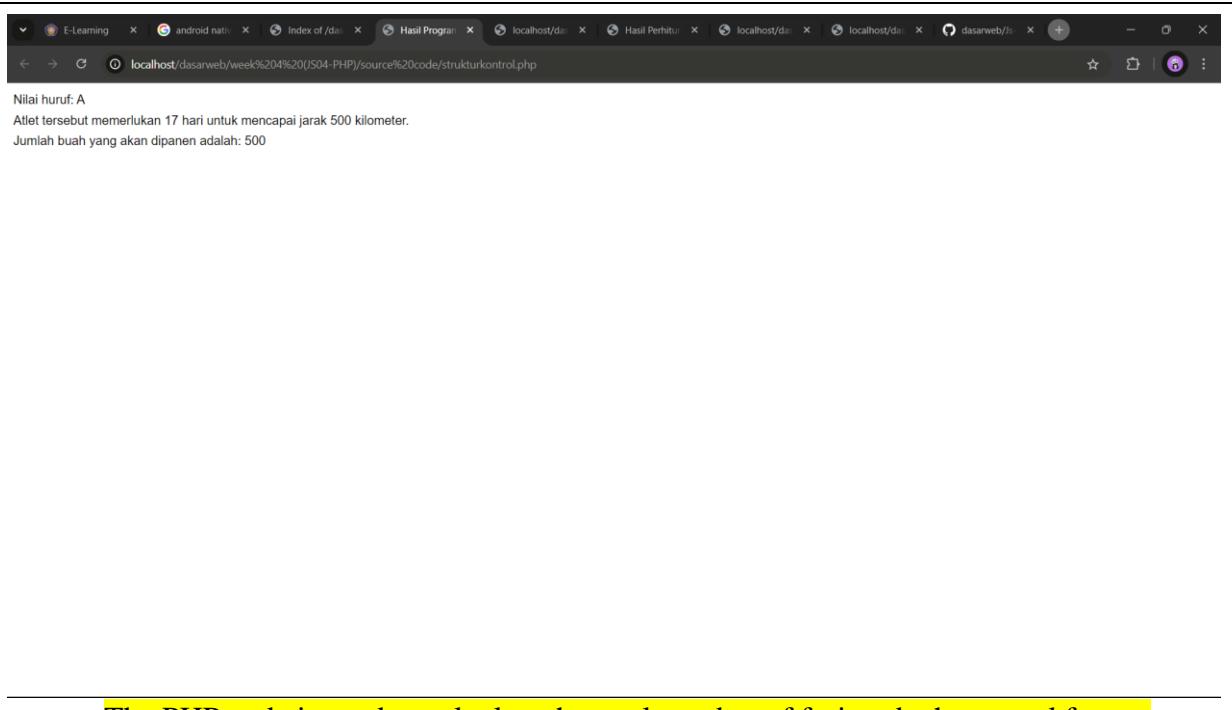
- The PHP code is used to calculate the number of days needed by an athlete to reach the target distance of 500 kilometers by adding 30 kilometers every day in the "while" loop. Starting with a distance of 0 kilometers, the code iteratively counts the days and adds 30 kilometers to the distance traveled each day until the distance reaches or exceeds 500 kilometers. Once the loop is complete, the code displays the number of days required by the athlete to reach the target distance of 500 kilometers.

9	Add the code in step 9 inside the strukturekontrol.php
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 localhost/dasarWeb/strukturekontrol.php
12	Note here what you observe from the addition of the program code above. (Question No. 16)

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hasil Program</title>
8      <style>
9          body {
10              font-family: Arial, 'Times New Roman', Times, serif;
11              line-height: 10px;
12          }
13      </style>
14  </head>
15
16 <body>
17     <div class="result">
18         <?php
19             $nilaiNumerik = 92;
20
21             if ($nilaiNumerik >= 90 && $nilaiNumerik <= 100) {
22                 echo "<p>Nilai huruf: A</p>";
23             } elseif ($nilaiNumerik >= 80 && $nilaiNumerik < 90) {
24                 echo "<p>Nilai huruf: B</p>";
25             } elseif ($nilaiNumerik >= 70 && $nilaiNumerik < 80) {
26                 echo "<p>Nilai huruf: C</p>";
27             } elseif ($nilaiNumerik < 70) {
28                 echo "<p>Nilai huruf: D</p>";
29             }
30
31             $jarakSaatIni = 0;
32             $jarakTarget = 500;
33             $peningkatanHarian = 30;
34             $hari = 0;
35
36             while ($jarakSaatIni < $jarakTarget) {
37                 $jarakSaatIni += $peningkatanHarian;
38                 $hari++;
39             }
40
41             echo "<p>Atlet tersebut memerlukan $hari hari untuk mencapai jarak 500 kilometer.</p>";
42
43             $jumlahLahan = 10;
44             $tanamanPerLahan = 5;
45             $buahPerTanaman = 10;
46             $jumlahBuah = 0;
47
48             for ($i = 1; $i <= $jumlahLahan; $i++) {
49                 $jumlahBuah += ($tanamanPerLahan * $buahPerTanaman);
50             }
51
52             echo "Jumlah buah yang akan dipanen adalah: $jumlahBuah";
53             ?>
54         </div>
55     </body>
56
57 </html>

```



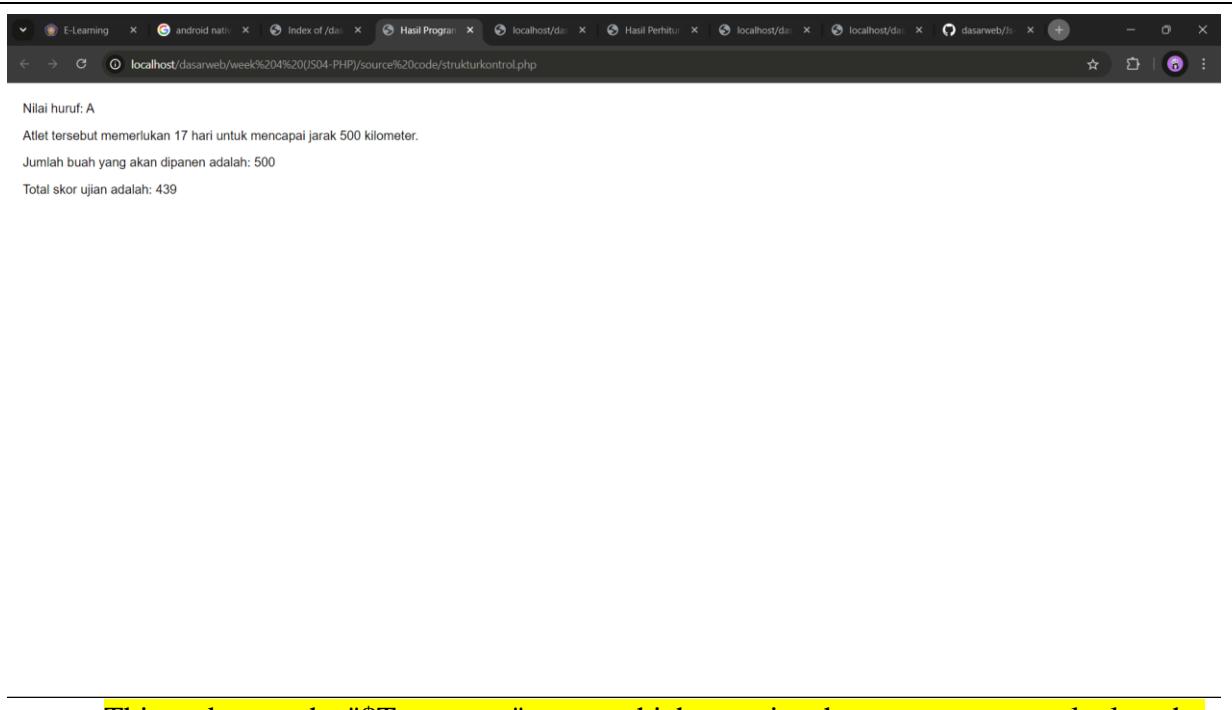
-
- The PHP code is used to calculate the total number of fruit to be harvested from a number of fields. For example, if there are 10 fields planted with 5 plants and each plant produces 10 fruits, this code uses a for loop to calculate the total number of fruits. For each iteration, the number of fruit from one field (the product of plants per field and fruit per plant) is added to the total number of fruit. Once the loop is complete, the code displays

13	Add the code in step 14 inside the strukturkontrol.php
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/dasarWeb/strukturkontrol.php
16	Note here what you observe from the addition of the program code above. (Question No. 17)

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hasil Program</title>
8      <style>
9          body {
10              font-family: Arial, 'Times New Roman', Times, serif;
11              line-height: 1.5;
12              margin: 20px;
13          }
14
15          p {
16              margin: 10px 0;
17          }
18      </style>
19  </head>
20
21  <body>
22      <div class="result">
23          <?php
24              // 1
25              $nilaiNumerik = 92;
26              if ($nilaiNumerik >= 90 && $nilaiNumerik <= 100) {
27                  echo "<p>Nilai huruf: A</p>";
28              } elseif ($nilaiNumerik >= 80 && $nilaiNumerik < 90) {
29                  echo "<p>Nilai huruf: B</p>";
30              } elseif ($nilaiNumerik >= 70 && $nilaiNumerik < 80) {
31                  echo "<p>Nilai huruf: C</p>";
32              } else {
33                  echo "<p>Nilai huruf: D</p>";
34              }
35
36              // 2
37              $jarakSaatIni = 0;
38              $jarakTarget = 500;
39              $peningkatanHarian = 30;
40              $hari = 0;
41              while ($jarakSaatIni < $jarakTarget) {
42                  $jarakSaatIni += $peningkatanHarian;
43                  $hari++;
44              }
45              echo "<p>Atlet tersebut memerlukan $hari hari untuk mencapai jarak 500 kilometer.</p>";
46
47              // 3
48              $jumlahLahan = 10;
49              $tanamanPerLahan = 5;
50              $buahPerTanaman = 10;
51              $jumlahBuah = 0;
52              for ($i = 1; $i <= $jumlahLahan; $i++) {
53                  $jumlahBuah += ($tanamanPerLahan * $buahPerTanaman);
54              }
55              echo "<p>Jumlah buah yang akan dipanen adalah: $jumlahBuah</p>";
56
57              // 4
58              $skorUjian = [85, 92, 78, 96, 88];
59              $totalSkor = 0;
60              foreach ($skorUjian as $skor) {
61                  $totalSkor += $skor;
62              }
63              echo "<p>Total skor ujian adalah: $totalSkor</p>";
64          ?>
65      </div>
66  </body>
67
68  </html>

```



- This code uses the "\$Testscores" array, which contains the test scores, to calculate the total test score. It starts by initializing the "\$totalScores" variable as 0, and then uses a "foreach" loop to iterate over each value in the "\$Testscores" array. Each looped value (or score) will be added to "\$totalScore". Once all the scores are added up, an "echo" is used to display the total result. The final result is the sum of all test scores in the array.

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/dasarWeb/strukturkontrol.php
20	Note here what you observe from the addition of the program code above. (Question No. 18)

```
● ○ ■
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Hasil Program</title>
8   <style>
9     body {
10       font-family: Arial, 'Times New Roman', Times, serif;
11       line-height: 1.5;
12       margin: 20px;
13     }
14
15     p {
16       margin: 10px 0;
17     }
18   </style>
19 </head>
20
21 <body>
22   <div class="result">
23     <?php
24     // 1
25     $nilaiNumerik = 92;
26     if ($nilaiNumerik >= 90 && $nilaiNumerik <= 100) {
27       echo "<p>Nilai huruf: A</p>";
28     } elseif ($nilaiNumerik >= 80 && $nilaiNumerik < 90) {
29       echo "<p>Nilai huruf: B</p>";
30     } elseif ($nilaiNumerik >= 70 && $nilaiNumerik < 80) {
31       echo "<p>Nilai huruf: C</p>";
32     } else {
33       echo "<p>Nilai huruf: D</p>";
34     }
35
36     // 2
37     $jarakSaatIni = 0;
38     $jarakTarget = 500;
39     $peningkatanHarian = 30;
40     $hari = 0;
41     while ($jarakSaatIni < $jarakTarget) {
42       $jarakSaatIni += $peningkatanHarian;
43       $hari++;
44     }
45     echo "<p>Atlet tersebut memerlukan $hari hari untuk mencapai jarak 500 kilometer.</p>";
46
47     // 3
48     $jumlahLahan = 10;
49     $tanamanPerLahan = 5;
50     $buahPerTanaman = 10;
51     $jumlahBuah = 0;
52     for ($i = 1; $i <= $jumlahLahan; $i++) {
53       $jumlahBuah += ($tanamanPerLahan * $buahPerTanaman);
54     }
55     echo "<p>Jumlah buah yang akan dipanen adalah: $jumlahBuah</p>";
56
57     // 4
58     $skorUjian = [85, 92, 78, 96, 88];
59     $totalSkor = 0;
60     foreach ($skorUjian as $skor) {
61       $totalSkor += $skor;
62     }
63     echo "<p>Total skor ujian adalah: $totalSkor</p>";
64
65     // 5
66     $nilaiSiswa = [85, 92, 58, 64, 90, 55, 88, 79, 70, 96];
67
68     foreach ($nilaiSiswa as $nilai) {
69       if ($nilai < 60) {
70         echo "Nilai: $nilai (Tidak lulus) <br>";
71         continue;
72
73         echo "Nilai: $nilai (Lulus) <br>";
74       }
75     ?>
76   </div>
77 </body>
78 </html>
```

```

Nilai huruf: A
Atlet tersebut memerlukan 17 hari untuk mencapai jarak 500 kilometer.
Jumlah buah yang akan dipanen adalah: 500
Total skor ujian adalah: 439
Nilai: 58 (Tidak lulus)
Nilai: 55 (Tidak lulus)

```

- This code evaluates the student's grades stored in the \$ nilaiSiswa array and then displays the passing status. The code checks all student grades with a foreach loop. The code displays "Value: \$nilai (Tidak lulus)" if the value is less than 60. Then, the next iteration is performed with continue. However, because it comes after continue in the if block, the echo statement, "Value: \$nilai (Lulus)
", is never executed if the value is greater than 60. As a result, this code does not print "Lulus" for qualified values.

	<p>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)</p> <p>Answer: After ignoring the two highest and lowest marks of the ten students, the marks are sorted from lowest to top to calculate the total marks to be used. First, the values are sorted from lowest to highest: 64, 70, 75, 78, 79, 85, 88, 90, 92. After removing the two lowest values (64, 70) and the two highest values (92, 96), the remaining values are 75, 78, 79, 85, 88, and 90. The total of these values is 495, which will be used to determine the average.</p>
21	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)

The screenshot shows a Microsoft Edge browser window. The address bar at the top displays the URL `localhost/dasarweb/week%204%20(JS04-PHP)/source%20code/kontrol1.php`. Below the address bar, the browser's toolbar includes icons for back, forward, search, and refresh, along with tabs for E-Learning, android nativ, localhost/dar, Hasil Program, localhost/das, Hasil Perhit..., localhost/das, localhost/das, and dasarweb/. The main content area of the browser shows the output of the PHP script, which is a single line of text: "Total nilai setelah mengabaikan dua nilai tertinggi dan dua nilai terendah adalah: 495". Above the browser window, the Visual Studio Code interface is visible. The Explorer sidebar on the left shows a project structure under the 'www' folder, including 'dasarweb', 'source code', and files like 'kontrol1.php', 'operator.php', 'strukturnkontrol.php', and 'index.php'. The 'kontrol1.php' file is open in the center editor, displaying the following PHP code:

```

1 <?php
2 $nilaiSiswa = [85, 92, 78, 64, 98, 75, 88, 79, 70, 96];
3 sort($nilaiSiswa);
4
5 array_splice($nilaiSiswa, 0, 2);
6 array_splice($nilaiSiswa, -2);
7
8 $totalNilai = array_sum($nilaiSiswa);
9
10 echo "Total nilai setelah mengabaikan dua nilai tertinggi dan dua nilai terendah adalah: $totalNilai";
11

```

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 kontrol2.php . Create the program code for step 23 and display the result below along with the program code (Question No. 20)

```

<?php
$hargaProduk = 120000;
$diskon = 0.20;

if ($hargaProduk > 100000) {
    $diskon = 0.20 * $hargaProduk;
}

$hargaFinal = $hargaProduk - $diskon;

echo "Harga yang harus dibayar setelah diskon: Rp $hargaFinal";

```

The price to be paid after discount: Rp 96000

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 kontrol3.php . Create the program code for step 25 show the result below along with the program code (Question No. 21)

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar has an 'EXPLORER' view with a tree structure showing a 'www' folder containing 'dasarweb', 'vscode', 'week 1 (JS01-HTML)', 'week 2 (JS02-CSS)', 'week 3 (JS03-JS)', 'week 4 (JS04-PHP)', 'laporan', 'source code', 'kontrol1.php', 'kontrol2.php', 'kontrol3.php' (which is currently selected), 'lathianoperator.php', 'operator.php', 'strukturkontrol.php', 'tipedata.php', 'varabelkonstanta.php', and 'index.php'. Below the tree are sections for 'OUTLINE' and 'TIMELINE'. The main editor area contains the following PHP code:

```
<?php  
$points = 520;  
echo "Player's total score is: $points\n";  
if ($points > 500) {  
    echo "Do players get additional rewards? YES";  
} else {  
    echo "Do players get additional rewards? NO";  
}  
11 |
```

The status bar at the bottom indicates 'Ln 11, Col 1' and other settings like 'Spaces: 4', 'UTF-8', 'CRLF', 'PHP', 'Go Live', and '8.1.10: 8.2'. The bottom right corner shows the Prettier icon. The bottom of the screen shows a browser-like interface with multiple tabs, one of which is active and displays the output: 'localhost/dasarweb/week%204%20(JS04-PHP)/source%20code/kontrol3.php'. The output text is: 'Player's total score is: 520 Do players get additional rewards? YES'

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
```

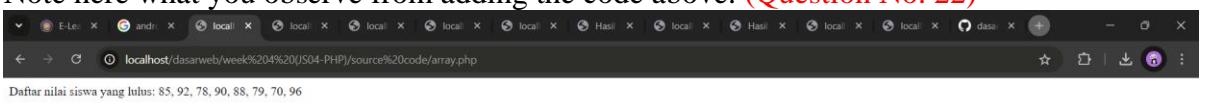
4. Iteration Through Associative Arrays

```
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/dasarWeb/array.php
4	<p>Note here what you observe from adding the code above. (Question No. 22)</p> 
5	<p>Type the additional code in step 6 inside the array.php</p> <ul style="list-style-type: none"> - This PHP code processes an array of student grades, `'\$nilaiSiswa'`, to filter out those that meet the passing criteria. It initializes an empty array, `'\$nilaiLulus'`, to store grades that are 70 or higher. The `foreach` loop iterates through each grade, checking if it is at least 70. If so, the grade is added to the `'\$nilaiLulus'` array. Finally, the `implode` function creates a comma-separated string of the passing grades, which is displayed as the list of grades that meet the passing threshold.

```

$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);

```

6

7 Save the file, then open a browser and run /refresh localhost/dasarWeb/array.php

Note here what you observe from the addition of the code above. (Question No. 23)



8

- This PHP code processes an array of employee data where each entry contains a name and years of experience. It initializes an empty array `\$karyawanPengalamanLimaTahun` to store names of employees with more than 5 years of experience. The `foreach` loop iterates through each employee, checking if their years of experience exceed 5. If so, their name is added to the `\$karyawanPengalamanLimaTahun` array. Finally, the `implode` function is used to create a comma-separated string of these names, which is then displayed as a list of employees with over 5 years of experience.

9

Type the additional code in step 10 inside the [array.php](#)

10

```
$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 $mataKuliah: <br>";
foreach ($daftarNilai[$mataKuliah] as $nilai) {
    echo "Nama: {$nilai[0]}, Nilai: {$nilai[1]} <br>";
}
```

11

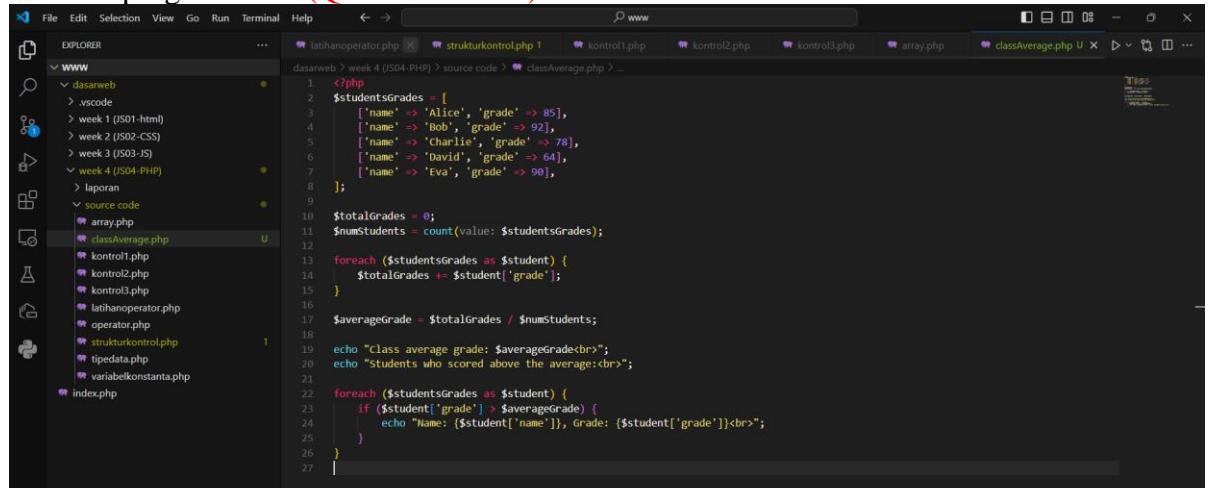
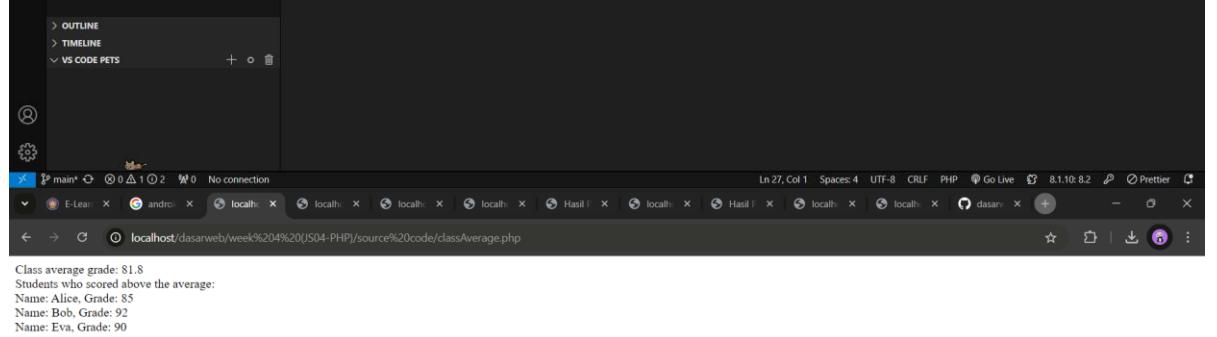
Save the file, then open a browser and run /refresh localhost/dasarWeb/array.php

Note here what you observe from the addition of the program code above. (Question No. 24)



12

- This PHP code retrieves and displays student grades for a specific subject from a nested associative array called `\$daftarNilai`. The array contains subjects as keys, each mapping to a list of students and their scores. The variable `\$mataKuliah` is set to 'Fisika', indicating that the code should focus on this subject. The `foreach` loop iterates over the list of grades for 'Fisika', and for each student, it prints their name and score in a formatted manner. The output will list the names and scores of students in the 'Fisika' course.

13	<p>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</p>
14	<p>Create the code for step 13 with a two-dimensional array and display the result below along with the program code (Question No. 25)</p>  <pre> <?php \$studentsGrades = [['name' => 'Alice', 'grade' => 85], ['name' => 'Bob', 'grade' => 92], ['name' => 'Charlie', 'grade' => 78], ['name' => 'David', 'grade' => 64], ['name' => 'Eva', 'grade' => 90],]; \$totalGrades = 0; \$numStudents = count(\$studentsGrades); foreach (\$studentsGrades as \$student) { \$totalGrades += \$student['grade']; } \$averageGrade = \$totalGrades / \$numStudents; echo "Class average grade: \$averageGrade
"; echo "Students who scored above the average:
"; foreach (\$studentsGrades as \$student) { if (\$student['grade'] > \$averageGrade) { echo "Name: {\$student['name']}, Grade: {\$student['grade']}
"; } } </pre>  <p>Class average grade: 81.8 Students who scored above the average: Name: Alice, Grade: 85 Name: Bob, Grade: 92 Name: Eva, Grade: 90</p>