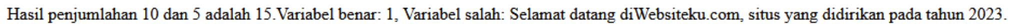
Name: Antonius Kaharap Kautsar
Class: 1-I
NIM: 2341720067

Assignment:

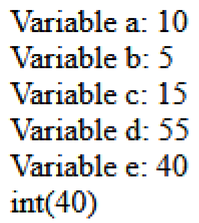
Practical section 1:
Question 1



```
Hasil penjumlahan 10 dan 5 adalah 15.Variabel benar: 1, Variabel salah: Selamat datang diWebsiteku.com, situs yang didirikan pada tahun 2023.
```

Variables in PHP are used to store data dynamically. We can declare a variable with a dollar sign ($) followed by the name, and the value assigned to it can change throughout the script. PHP variables do not require data type declaration, making them flexible for handling different types of data.

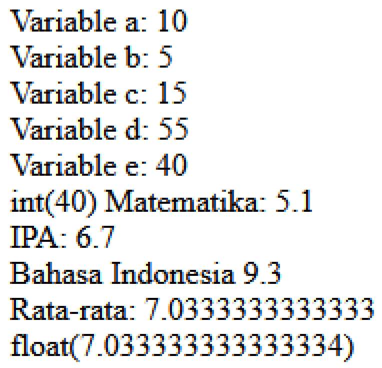
Practical section 2:
Question 2



```
Variable a: 10
Variable b: 5
Variable c: 15
Variable d: 55
Variable e: 40
int(40)
```

The code demonstrates how variables in PHP can store different types of data (e.g., strings, integers, booleans). The output depends on how the variables are used, and PHP automatically handles the conversion of data types.

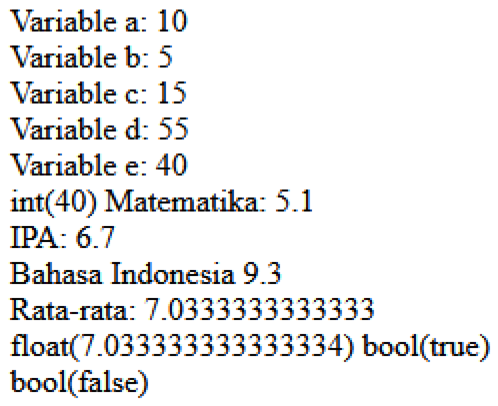
Question 3



```
Variable a: 10
Variable b: 5
Variable c: 15
Variable d: 55
Variable e: 40
int(40) Matematika: 5.1
IPA: 6.7
Bahasa Indonesia 9.3
Rata-rata: 7.0333333333333
float(7.033333333333334)
```

Similar to the previous question we created 4 variables with 1 for the summation of all 3 divided by 3. We used echo to print the output and then used vardump to know the data type which is float. We're able to know that PHP is flexible to its variable.

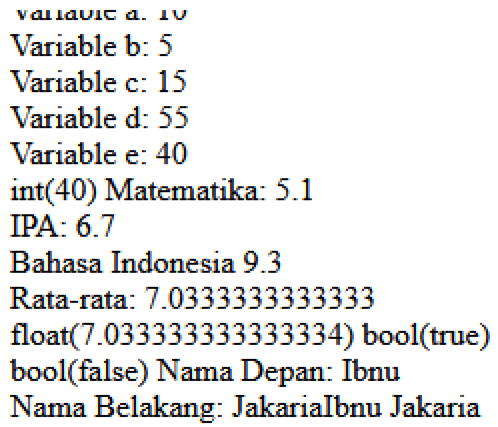
Question 4



```
Variable a: 10
Variable b: 5
Variable c: 15
Variable d: 55
Variable e: 40
int(40) Matematika: 5.1
IPA: 6.7
Bahasa Indonesia 9.3
Rata-rata: 7.0333333333333
float(7.033333333333334) bool(true)
bool(false)
```

Likewise, we created 2 variables this time with a Boolean data type thanks to php flexibility when it comes to defining variable. And then var dump its output.

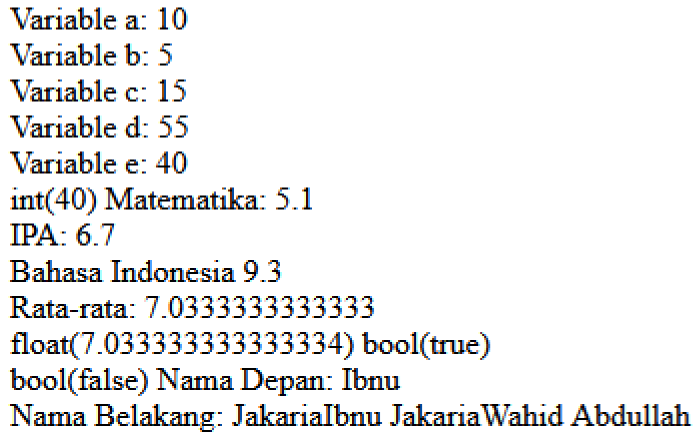
Question 5



```
Variable a: 10
Variable b: 5
Variable c: 15
Variable d: 55
Variable e: 40
int(40) Matematika: 5.1
IPA: 6.7
Bahasa Indonesia 9.3
Rata-rata: 7.0333333333333
float(7.0333333333333334) bool(true)
bool(false) Nama Depan: Ibnu
Nama Belakang: JakariaIbnu Jakaria
```

The code creates two variables, $namaDepan and $namaBelakang, to store a first name and a last name. It then combines these to create a full name ($namaLengkap) in two different ways. Finally, it prints the first name, last name, and the full name to the browser.

Question 6



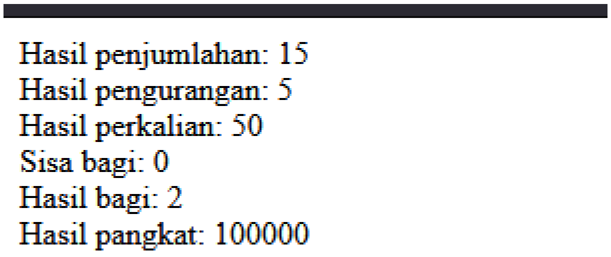
```
Variable a: 10
Variable b: 5
Variable c: 15
Variable d: 55
Variable e: 40
int(40) Matematika: 5.1
IPA: 6.7
Bahasa Indonesia 9.3
Rata-rata: 7.0333333333333
float(7.0333333333333334) bool(true)
bool(false) Nama Depan: Ibnu
Nama Belakang: JakariaIbnu JakariaWahid Abdullah
```

The provided PHP code snippet demonstrates a fundamental concept in programming: working with lists or collections of data. It does this by utilizing an array, a data structure that can hold multiple values.

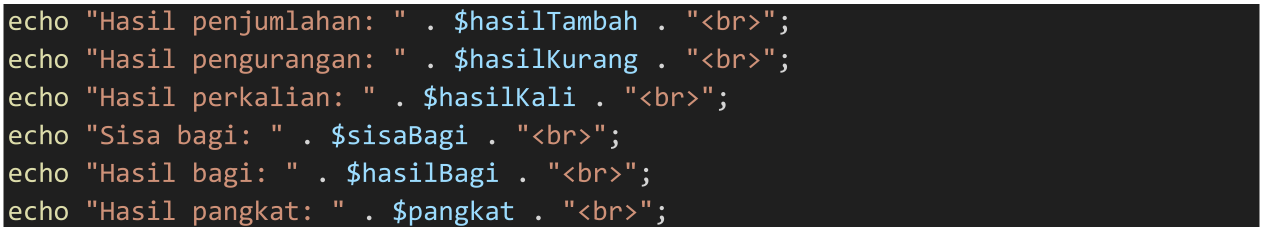
First, an array named $listMahasiswa is created. This array is specifically designed to store a list of student names. Next, the code uses echo $listMahasiswa[0]; to display the very first name in the list. This is possible because arrays in PHP are zero-indexed, which means the first item is located at position 0.

Practices section 3:

Question 7



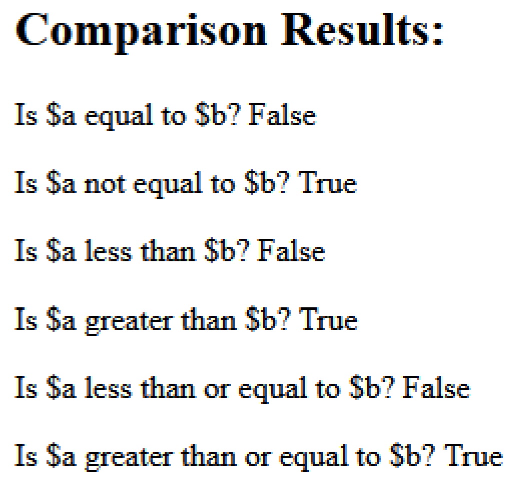
```
Hasil penjumlahan: 15
Hasil pengurangan: 5
Hasil perkalian: 50
Sisa bagi: 0
Hasil bagi: 2
Hasil pangkat: 100000
```



```php
echo "Hasil penjumlahan: " . $hasilTambah . "<br>";
echo "Hasil pengurangan: " . $hasilKurang . "<br>";
echo "Hasil perkalian: " . $hasilKali . "<br>";
echo "Sisa bagi: " . $sisaBagi . "<br>";
echo "Hasil bagi: " . $hasilBagi . "<br>";
echo "Hasil pangkat: " . $pangkat . "<br>";
```

Question 8
I simply used echo + the variable and then end it with a <br> to make a line break

Question 9



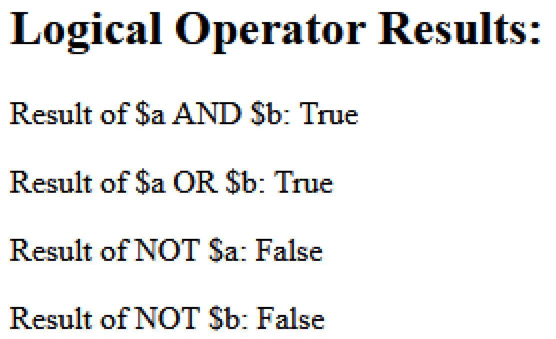
```
Comparison Results:

Is $a equal to $b? False

Is $a not equal to $b? True

Is $a less than $b? False

Is $a greater than $b? True

Is $a less than or equal to $b? False

Is $a greater than or equal to $b? True
```

The provided code snippet defines several variables ($hasilSama, $hasilTidakSama, etc.) and assigns them the results of comparison operations between two variables, $a and $b (which we assume are defined elsewhere in the operator.php file). These

comparison operations include checking for equality (==), inequality (!=), less than (<), greater than (>), less than or equal to (<=), and greater than or equal to (>=).

Using echo we're able to make the result nice and neat.

Question 10

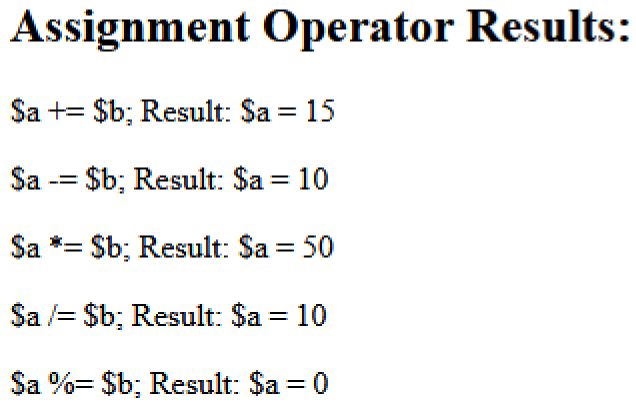


```
Logical Operator Results:

Result of $a AND $b: True

Result of $a OR $b: True

Result of NOT $a: False

Result of NOT $b: False
```

The code snippets uses logical operator, the following are the explanation:
$hasilAnd = $a && $b; performs a logical AND operation. It will be true only if both $a and $b are true.
$hasilOr = $a || $b; performs a logical OR operation. It will be true if either $a or $b (or both) are true.
$hasilNotA = !$a; and $hasilNotB = !$b; perform logical NOT operations. They invert the truth value of $a and b, respectively. If $a is true, $hasilNotA will be false, and vice versa.

Using echo we're able to make the output nice to look at.

Question 11



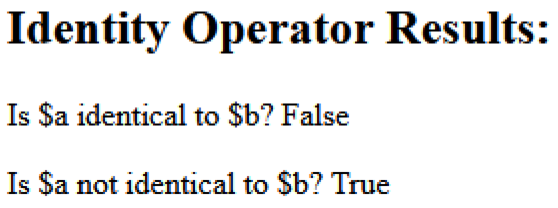
```
Assignment Operator Results:

$a += $b; Result: $a = 15

$a -= $b; Result: $a = 10

$a *= $b; Result: $a = 50

$a /= $b; Result: $a = 10

$a %= $b; Result: $a = 0
```

In the code snippet we're able to know that it's a assignment operator.
The following are its explanation

+= (Addition assignment): Adds the right operand to the left operand and assigns the result to the left operand.
-= (Subtraction assignment): Subtracts the right operand from the left operand and assigns the result to the left operand.
*= (Multiplication assignment): Multiplies the left operand by the right operand and assigns the result to the left operand.
/= (Division assignment): Divides the left operand by the right operand and assigns the result to the left operand.
%= (Modulo assignment): Divides the left operand by the right operand and assigns the remainder to the left operand.

Using echo we're able to make the output nice and neat.

Question 12



Identity Operator Results:

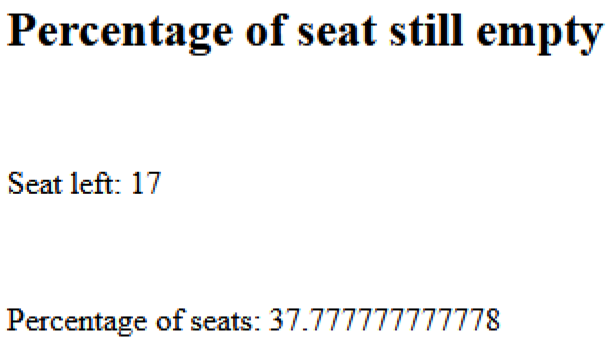
Is $a identical to $b? False

Is $a not identical to $b? True

Based on the code snippet we're able to know that it's a identity operator.
Here are its explanations:
=== (Identical): Returns true only if both operands have the same value and the same data type.
!== (Not identical): Returns true if the operands have different values or different data types.

As usual we used echo with a bit of HTML syntax to make it neat and nice.

Question 13



Percentage of seat still empty

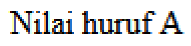
Seat left: 17

Percentage of seats: 37.777777777778

Using arithmetic operator we're able to know the amount of seats left and the percentage of seats left.

Practical Section 4:

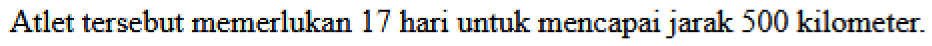
Question 14



```
Nilai huruf A
```

The code defines a grading system that converts numerical scores to letter grades (A, B, C, or D) using conditional logic (if, elseif, else). The provided score of 92 results in an output of "Nilai huruf: A".

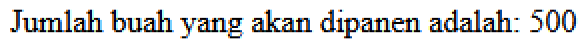
Question 15



```
Atlet tersebut memerlukan 17 hari untuk mencapai jarak 500 kilometer.
```

The code calculates how many days an athlete needs to reach a target distance of 500 kilometers. It uses a while loop to simulate the athlete's daily progress. The while loop will keep adding 30 to jarakSaatini until it has reached 500

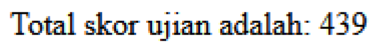
Question 16



```
Jumlah buah yang akan dipanen adalah: 500
```

The code calculates the total number of fruits that will be harvested from a given number of plots of land. It initializes variables for the number of plots, plants per plot, fruits per plant, and a counter for the total fruit. The for loop iterates through each plot, multiplying the number of plants, fruits per plant, and adding this to the total. Finally, it displays the total number of fruits.

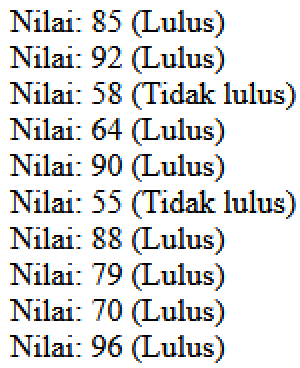
Question 17



```
Total skor ujian adalah: 439
```

The code calculates and displays the total score from an array of exam scores. It initializes a variable $totalSkor to 0. Then it uses a foreach loop to iterate through each score in the $skorUjian array, adding each score to the $totalSkor. Finally, it prints the total score.

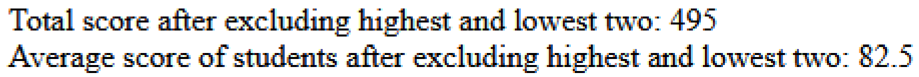
Question 18



```
Nilai: 85 (Lulus)
Nilai: 92 (Lulus)
Nilai: 58 (Tidak lulus)
Nilai: 64 (Lulus)
Nilai: 90 (Lulus)
Nilai: 55 (Tidak lulus)
Nilai: 88 (Lulus)
Nilai: 79 (Lulus)
Nilai: 70 (Lulus)
Nilai: 96 (Lulus)
```

The code iterates through an array of student scores ($nilaiSiswa) and categorizes each score as "Lulus" (Pass) or "Tidak Lulus" (Fail) based on whether it's above or below 60. It uses a foreach loop to process each score and an if statement to check if it's below 60. If so, it prints "Tidak Lulus" and uses continue to skip to the next iteration, preventing the "Lulus" message from being printed for that score.

Question 19

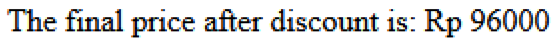


```
Total score after excluding highest and lowest two: 495
Average score of students after excluding highest and lowest two: 82.5
```

The code will first sort the array based on the number of highest to lowest (ascending) and then it will remove the first and last 2 value inside of the index using array_shift and array_pop. Then it will calculate the summation of the array using array_sum and calculate the how many student are left inside the array using count

Then it will calculate the the average score of all the students excluding the highest and lowest 2. Then we output the result using echo

Question 20

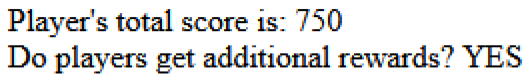


```
The final price after discount is: Rp 96000
```

In the code we first created 3 variable and initialize the variable according to the story question. Then we make an if else statement to check if the original amount are greater than the discount threshold. If It does it will create another variable and that variable will have the discount which is calculated by the discount percentage divided by 100 times the original price. And then we create the final price variable which will have the value of the original price minus the discount amount. If it doesn't meet the threshold it will simply make the final price as the original price.

We then output the result using echo

Question 21

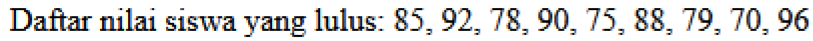


```
Player's total score is: 750
Do players get additional rewards? YES
```

The code first defines a variable points and assigns it the value 750. Then, it checks if the value of points is greater than 500. If it is, the variable additionalRewards is assigned the value "YES". Otherwise, it is assigned the value "NO". Finally, the code prints the player's total score and whether or not they get additional rewards.

Practical Section 5:

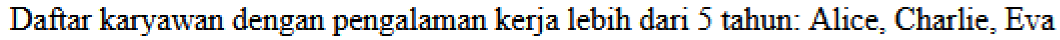
Question 22



```
Daftar nilai siswa yang lulus: 85, 92, 78, 90, 75, 88, 79, 70, 96
```

The code processes an array of student scores ($nilaiSiswa). It iterates through each score using a foreach loop. If a score is greater than or equal to 70 (passing grade), it's added to the $nilailulus array. Finally, the code displays a comma-separated list of passing scores using implode.

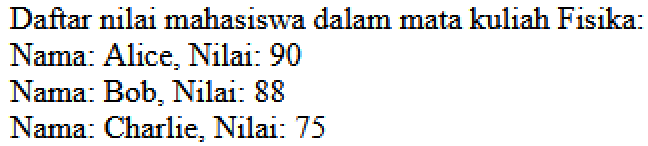
Question 23



```
Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: Alice, Charlie, Eva
```

The code processes an array named $daftarKaryawan which contains information about employees (name and years of experience). It iterates through each employee record using a foreach loop. If an employee's experience ($karyawan[1]) is greater than 5 years, their name ($karyawan[0]) is added to the $karyawanPengalamanLimaTahun array. Finally, the code displays a comma-separated list of employees with more than 5 years of experience using implode.

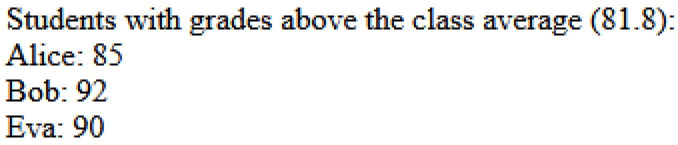
Question 24



```
Daftar nilai mahasiswa dalam mata kuliah Fisika:
Nama: Alice, Nilai: 90
Nama: Bob, Nilai: 88
Nama: Charlie, Nilai: 75
```

The code defines a three dimensional array $daftarNilai that stores student scores for different subjects ('Matematika', 'Fisika', 'Kimia'). It then focuses on a specific subject ('Fisika') stored in the $mataKuliah variable. It iterates through the scores for that subject and prints the student's name and their corresponding score.

Question 25



```
Students with grades above the class average (81.8):
Alice: 85
Bob: 92
Eva: 90
```

The code first creates a two dimensional array named $students which has the student names with their corresponding grades. Then it will first create a variable of $total and initialize it to 0 then it will use a foreach loop and summate all of the grade in the index 1 to the newly created variable of total. Then it will creates a variable of $average which has the value of $total divided by the length of the student's array using count

Then we output the value using echo and loop through each array using foreach

Github link : https://github.com/Valtern/week-4