

LAPORAN PRAKTIKUM
PERANCANGAN DAN PEMROGRAMAN WEB

MODUL 9

PHP



Oleh:

Aulia Jasifa Br Ginting - 2311104060

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
DIREKTORAT KAMPUS PURWOKERTO
UNIVERSITAS TELKOM

2025

BAB I

PENDAHULUAN

A. Dasar Teori

Pada praktikum mempelajari dasar-dasar pemrograman PHP sebagai bahasa *server-side scripting* yang digunakan untuk membangun halaman web dinamis. PHP dijalankan di sisi server sehingga setiap script diproses terlebih dahulu sebelum hasil akhirnya dikirim ke browser dalam bentuk HTML. Konsep ini berbeda dengan *client-side scripting* seperti JavaScript yang berjalan di sisi pengguna. Dasar teori yang dipelajari meliputi variabel, tipe data, operator, struktur kondisi, perulangan, fungsi, dan array. PHP menggunakan tanda \$ untuk variabel, mendukung operator aritmatika, perbandingan, dan logika, serta menyediakan struktur percabangan seperti if, else, dan switch untuk pengambilan keputusan. Selain itu, PHP juga memiliki beberapa bentuk perulangan seperti for, while, do-while, dan foreach, serta mendukung pembuatan fungsi yang dapat digunakan kembali (*reusable*). Array pada PHP dapat berupa array numerik maupun array asosiatif. Dalam praktikum ini digunakan XAMPP sebagai server lokal yang menyediakan Apache dan interpreter PHP untuk menjalankan file .php secara langsung di browser

B. Tujuan

1. Mahasiswa mampu memahami konsep dan implementasi PHP pada web.
2. Mahasiswa mampu memahami sintaks, elemen, dan fungsi pada PHP.

BAB II

HASIL

A. GUIDED (Praktikum Terbimbing)

a. Pengenalan PHP

```
<?php  
echo "Hello World";  
echo "<br>";  
echo "Nama Saya Jasifa";  
echo "<br>";  
echo "Kelas SE-07-02";
```



Penjelasan:

Kode PHP tersebut berfungsi untuk menampilkan beberapa teks ke layar browser secara berurutan. Baris pertama echo "Hello World"; digunakan untuk mencetak tulisan **Hello World** sebagai output. Pada baris berikutnya, echo "
"; digunakan untuk menambahkan jeda baris (line break), sehingga teks selanjutnya akan muncul di baris baru. Kemudian echo "Nama Saya Jasifa"; mencetak teks

Nama Saya Jasifa, diikuti lagi oleh `
` untuk memberi jarak ke baris berikutnya. Terakhir, baris `echo "Kelas SE-07-02";` menampilkan teks **Kelas SE-07-02**. Secara keseluruhan, script ini hanya bertugas menampilkan tiga baris tulisan dengan jeda baris di antaranya.

Output:



b. Variabel

```
<?php
$nama = "Aulia";
$nim = "2311104060";
$hobi = "listening to music";

echo "Nama: " . $nama;
echo "<br>";
echo "NIM: " . $nim;
echo "<br>";
echo "Hobi: " . $hobi;
```



```

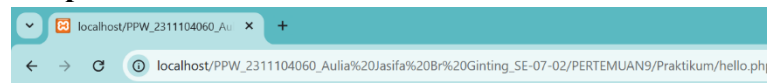
 9  $nama = "Aulia";
10  $nim = "2311104060";
11  $hobi = "listening to music";
12
13  echo "Nama: " . $nama;
14  echo "<br>";
15  echo "NIM: " . $nim;
16  echo "<br>";
17  echo "Hobi: " . $hobi;

```

Penjelasan:

Kode PHP di atas digunakan untuk menampilkan informasi berupa nama, NIM, dan hobi. Pertama, tiga variabel didefinisikan: **\$nama** berisi "Aulia", **\$nim** berisi "2311104060", dan **\$hobi** berisi "listening to music". Setelah itu, fungsi **echo** digunakan untuk menampilkan teks beserta nilai dari setiap variabel tersebut. Setiap baris output dipisahkan dengan tag HTML **
** agar muncul di baris baru pada browser. Hasil akhirnya, halaman akan menampilkan teks: *Nama: Aulia*, *NIM: 2311104060*, dan *Hobi: listening to music* masing-masing di baris terpisah.

Output



Nama: Aulia
 NIM: 2311104060
 Hobi: listening to music

c. Konstanta

```

<?php
define("Nama", "Jasifa");
define("NIM", "2311104060");
define("Asal", "Medan");

echo "Nama : " . Nama . "<br>";
echo "NIM : " . NIM . "<br>";
echo "Asal : " . Asal;

```

```
1  <?php
2  define("Nama", "Jasifa");
3  define("NIM", "2311104060");
4  define("Asal", "Medan");
5
6  echo "Nama : " . Nama . "<br>";
7  echo "NIM : " . NIM . "<br>";
8  echo "Asal : " . Asal;

20  define("Nama", "Jasifa");
21  define("NIM", "2311104060");
22  define("Asal", "Medan");
23
24  echo "Nama : " . Nama . "<br>";
25  echo "NIM : " . NIM . "<br>";
26  echo "Asal : " . Asal;
```

Penjelasan:

Kode di atas berisi beberapa baris perintah PHP yang semuanya sedang **dicomment** menggunakan `//`, sehingga tidak akan dijalankan. Bagian pertama adalah tiga perintah `echo` yang seharusnya menampilkan teks *"Hello World"*, *"Nama Saya Jasifa"*, dan *"Kelas SE-07-02"* masing-masing di baris baru menggunakan tag HTML `
`. Bagian kedua mendefinisikan tiga variabel, yaitu **\$nama**, **\$nim**, dan **\$hobi**, lalu menampilkannya menggunakan `echo` dengan format *"Nama: ..."*, *"NIM: ..."*, dan *"Hobi: ..."*; setiap output juga dipisahkan dengan `
`. Namun karena seluruh baris diberi tanda komentar, kode ini tidak memberikan output apa pun ketika dijalankan. Jika tanda `//` dihapus, barulah teks dan data tersebut akan muncul di browser.

Output




```
Nama :Jasifa
NIM :2311104060
Asal :Medan
```

d. Struktur Kondisi

```
<?php
$nilai = 87;

if ($nilai > 50) {
    echo "Nilai anda adalah " . $nilai . ". Selamat, Anda lulus";
} else {
    echo "Nilai anda adalah " . $nilai . ". Maaf, Anda tidak lulus";
}
```



The image shows two screenshots of a code editor. The top screenshot shows a full PHP script with line numbers 1 through 8. The bottom screenshot shows a zoomed-in view of the same code, with line numbers 29 through 35. The code is as follows:

```
1  <?php
2  $nilai = 87;
3
4  if ($nilai > 50) {
5      echo "Nilai anda adalah " . $nilai . ". Selamat, Anda lulus";
6  } else {
7      echo "Nilai anda adalah " . $nilai . ". Maaf, Anda tidak lulus";
8  }
29 $nilai = 87;
30
31 if ($nilai > 50) {
32     echo "Nilai anda adalah " . $nilai . ". Selamat, Anda lulus";
33 } else {
34     echo "Nilai anda adalah " . $nilai . ". Maaf, Anda tidak lulus";
35 }
```

Penjelasan:

Kode PHP di atas digunakan untuk menentukan apakah seorang siswa lulus atau tidak berdasarkan nilai yang diperoleh. Variabel **\$nilai** diisi dengan angka 49. Kemudian struktur **if-else** digunakan untuk mengecek apakah nilai tersebut lebih besar dari 50. Jika lebih besar, program menampilkan pesan bahwa siswa lulus. Namun, karena nilai yang dimasukkan adalah 49, kondisi tersebut tidak terpenuhi, sehingga blok **else** dijalankan dan program menampilkan pesan bahwa siswa tidak lulus. Program ini membantu menunjukkan cara kerja logika percabangan dalam PHP.

Output



Nilai anda adalah 87. Selamat, Anda lulus

e. Struktur Kondisi

```
<?php
echo "Hello World";
echo "<br>";
echo "Nama Saya Jasifa";
echo "<br>";
echo "Kelas SE-07-02";
$nilai = 87;

switch ($nilai) {
    case ($nilai > 50 && $nilai <= 60):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda C";
        break;
    case ($nilai > 60 && $nilai <= 70):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda BC";
        break;
    case ($nilai > 70 && $nilai <= 75):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda B";
        break;
    case ($nilai > 75 && $nilai <= 80):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda AB";
```



```

        break;

    case ($nilai > 80 && $nilai <= 100):

        echo "Nilai Anda adalah $nilai. Indeks nilai anda A";

        break;

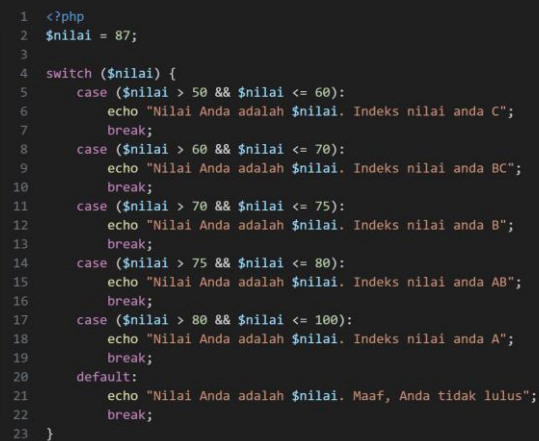
    default:

        echo "Nilai Anda adalah $nilai. Maaf, Anda tidak lulus";

        break;

}

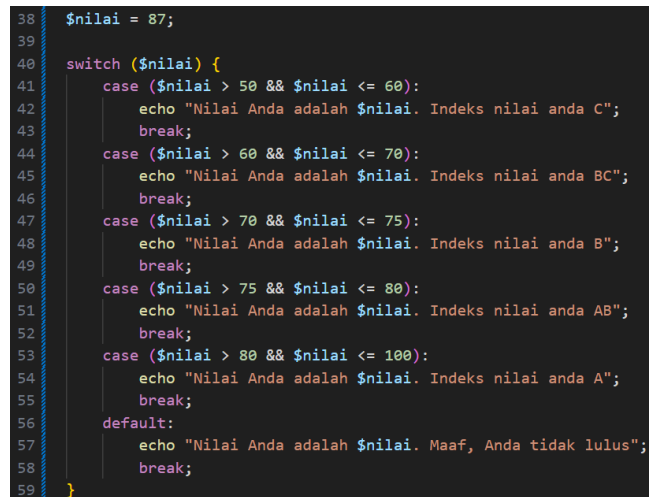
```



```

1  <?php
2  $nilai = 87;
3
4  switch ($nilai) {
5      case ($nilai > 50 && $nilai <= 60):
6          echo "Nilai Anda adalah $nilai. Indeks nilai anda C";
7          break;
8      case ($nilai > 60 && $nilai <= 70):
9          echo "Nilai Anda adalah $nilai. Indeks nilai anda BC";
10         break;
11     case ($nilai > 70 && $nilai <= 75):
12         echo "Nilai Anda adalah $nilai. Indeks nilai anda B";
13         break;
14     case ($nilai > 75 && $nilai <= 80):
15         echo "Nilai Anda adalah $nilai. Indeks nilai anda AB";
16         break;
17     case ($nilai > 80 && $nilai <= 100):
18         echo "Nilai Anda adalah $nilai. Indeks nilai anda A";
19         break;
20     default:
21         echo "Nilai Anda adalah $nilai. Maaf, Anda tidak lulus";
22         break;
23 }

```



```

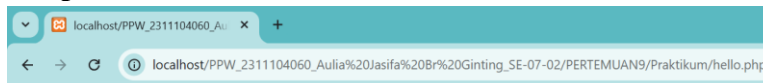
38  $nilai = 87;
39
40  switch ($nilai) {
41      case ($nilai > 50 && $nilai <= 60):
42          echo "Nilai Anda adalah $nilai. Indeks nilai anda C";
43          break;
44      case ($nilai > 60 && $nilai <= 70):
45          echo "Nilai Anda adalah $nilai. Indeks nilai anda BC";
46          break;
47      case ($nilai > 70 && $nilai <= 75):
48          echo "Nilai Anda adalah $nilai. Indeks nilai anda B";
49          break;
50      case ($nilai > 75 && $nilai <= 80):
51          echo "Nilai Anda adalah $nilai. Indeks nilai anda AB";
52          break;
53      case ($nilai > 80 && $nilai <= 100):
54          echo "Nilai Anda adalah $nilai. Indeks nilai anda A";
55          break;
56      default:
57          echo "Nilai Anda adalah $nilai. Maaf, Anda tidak lulus";
58          break;
59  }

```

Penjelasan:

Kode PHP di atas digunakan untuk menentukan indeks nilai berdasarkan angka yang tersimpan dalam variabel **\$nilai**, yang dalam contoh ini adalah 87. Struktur **switch** digunakan bersama beberapa kondisi pada setiap *case* untuk memeriksa rentang nilai tertentu. Setiap *case* berisi kondisi rentang angka yang menentukan indeks, seperti C, BC, B, AB, hingga A. Karena nilai 87 berada dalam rentang **80–100**, program mengeksekusi *case* tersebut dan menampilkan pesan bahwa indeks nilainya adalah **A**. Jika nilai tidak masuk ke salah satu rentang yang tersedia, blok **default** akan dijalankan untuk menampilkan pesan bahwa pengguna tidak lulus. Program ini menunjukkan cara menggunakan switch untuk logika pemilihan berdasarkan kategori nilai.

Output



Nilai Anda adalah 87. Indeks nilai anda A

f. Perulangan (Looping)

```
<?php
echo "Hello World";
echo "<br>";
echo "Nama Saya Jasifa";
echo "<br>";
echo "Kelas SE-07-02";
<?php

echo "Ini adalah contoh perulangan for";

echo "<br>";


for ($i = 1; $i <= 10; $i++) {

    echo $i . " ";

}
```

```
echo "<br>";
echo "<br>";
echo "Ini adalah contoh perulangan while";
echo "<br>";
$i = 1;
while ($i <= 20) {
    echo $i . " ";
    $i += 2;
}

echo "<br>";
echo "<br>";
echo "Ini adalah contoh perulangan do-while";
echo "<br>";
$i = 28;
do {
    echo $i . " ";
    $i -= 3;
} while ($i > 0);
```



```

1  <?php
2  echo "Ini adalah contoh perulangan for";
3  echo "<br>";
4  for ($i = 1; $i <= 10; $i++) {
5      echo $i . " ";
6  }
7
8  echo "<br>";
9  echo "<br>";
10 echo "Ini adalah contoh perulangan while";
11 echo "<br>";
12 $i = 1;
13 while ($i <= 20) {
14     echo $i . " ";
15     $i += 2;
16 }
17
18 echo "<br>";
19 echo "<br>";
20 echo "Ini adalah contoh perulangan do-while";
21 echo "<br>";
22 $i = 28;
23 do {
24     echo $i . " ";
25     $i -= 3;
26 } while ($i > 0);

```

```

62 echo "Ini adalah contoh perulangan for";
63 echo "<br>";
64 for ($i = 1; $i <= 10; $i++) {
65     echo $i . " ";
66 }
67
68 echo "<br>";
69 echo "<br>";
70 echo "Ini adalah contoh perulangan while";
71 echo "<br>";
72 $i = 1;
73 while ($i <= 20) {
74     echo $i . " ";
75     $i += 2;
76 }
77
78 echo "<br>";
79 echo "<br>";
80 echo "Ini adalah contoh perulangan do-while";
81 echo "<br>";
82 $i = 28;
83 do {
84     echo $i . " ";
85     $i -= 3;
86 } while ($i > 0);

```

Penjelasan:

Kode di atas menampilkan tiga jenis perulangan yang umum digunakan dalam PHP: **for**, **while**, dan **do-while**. Perulangan *for* digunakan untuk menampilkan angka dari 1 sampai 10 secara berurutan. Selanjutnya, perulangan *while* dimulai dari angka 1 dan terus berjalan selama nilai masih kurang atau sama dengan 20, sambil menambah angka sebanyak 2 pada setiap iterasi sehingga menghasilkan deret bilangan ganjil. Terakhir, perulangan *do-while* dimulai dari angka 28 dan mengurangi nilai sebesar 3 setiap pengulangan, serta tetap mengeksekusi satu kali meskipun kondisi diperiksa setelah baris kode dijalankan. Program ini memberikan gambaran sederhana mengenai perbedaan cara kerja ketiga struktur perulangan tersebut.

Ouput



Ini adalah contoh perulangan for
1 2 3 4 5 6 7 8 9 10

Ini adalah contoh perulangan while
1 3 5 7 9 11 13 15 17 19

Ini adalah contoh perulangan do-while
28 25 22 19 16 13 10 7 4 1

g. Function

```
<?php
function cetakGenap()
{
    for ($i = 1; $i <= 60; $i++) {
        if ($i % 2 == 0) {
            echo "$i ";
        }
    }
}
//pemanggilan fungsi
cetakGenap();
```

```

1 <?php
2 function cetakGenap()
3 {
4     for ($i = 1; $i <= 60; $i++) {
5         if ($i % 2 == 0) {
6             echo "$i ";
7         }
8     }
9 }
10 //pemanggilan fungsi
11 cetakGenap();

```

```

89 function cetakGenap()
90 {
91     for ($i = 1; $i <= 60; $i++) {
92         if ($i % 2 == 0) {
93             echo "$i ";
94         }
95     }
96 }
97 //pemanggilan fungsi
98 cetakGenap();

```

Penjelasan:

Kode PHP di atas mendefinisikan sebuah fungsi bernama **cetakGenap()** yang bertugas menampilkan semua bilangan genap dari 1 hingga 60. Di dalam fungsi, digunakan perulangan *for* yang berjalan dari angka 1 sampai 60. Setiap angka kemudian dicek menggunakan kondisi $\$i \% 2 == 0$, yang berarti angka tersebut genap jika sisa hasil baginya terhadap 2 adalah nol. Jika kondisinya benar, angka tersebut dicetak. Setelah fungsi didefinisikan, baris terakhir memanggil fungsi **cetakGenap()** sehingga seluruh bilangan genap dari 2 sampai 60 ditampilkan.

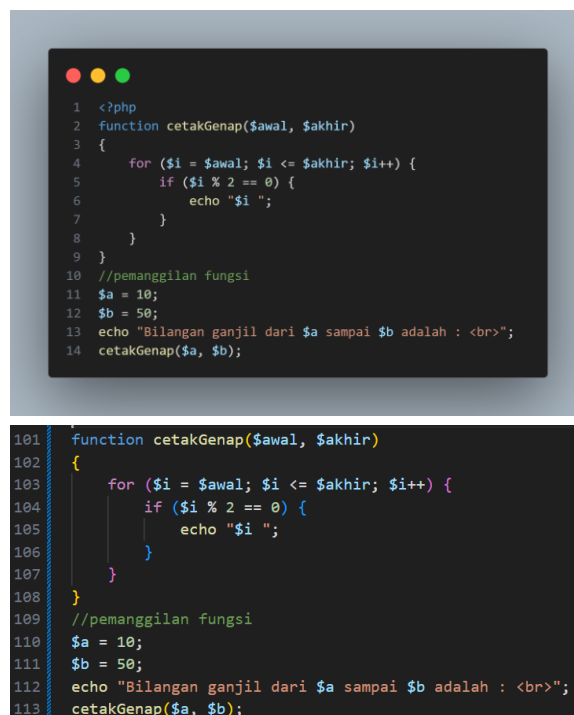
Output



2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60

h. Function

```
<?php
function cetakGenap($awal, $akhir)
{
    for ($i = $awal; $i <= $akhir; $i++) {
        if ($i % 2 == 0) {
            echo "$i ";
        }
    }
}
//pemanggilan fungsi
$a = 10;
$b = 50;
echo "Bilangan ganjil dari $a sampai $b adalah : <br>";
cetakGenap($a, $b);
```



```
1  <?php
2  function cetakGenap($awal, $akhir)
3  {
4      for ($i = $awal; $i <= $akhir; $i++) {
5          if ($i % 2 == 0) {
6              echo "$i ";
7          }
8      }
9  }
10 //pemanggilan fungsi
11 $a = 10;
12 $b = 50;
13 echo "Bilangan ganjil dari $a sampai $b adalah : <br>";
14 cetakGenap($a, $b);

101 function cetakGenap($awal, $akhir)
102 {
103     for ($i = $awal; $i <= $akhir; $i++) {
104         if ($i % 2 == 0) {
105             echo "$i ";
106         }
107     }
108 }
109 //pemanggilan fungsi
110 $a = 10;
111 $b = 50;
112 echo "Bilangan ganjil dari $a sampai $b adalah : <br>";
113 cetakGenap($a, $b);
```

Penjelasan:

Kode PHP di atas mendefinisikan fungsi **cetakGenap(\$awal, \$akhir)** yang digunakan untuk menampilkan bilangan genap dalam suatu rentang angka tertentu. Fungsi ini menerima dua parameter, yaitu angka awal dan angka akhir,

lalu melakukan perulangan dari nilai awal hingga nilai akhir. Pada setiap iterasi, angka diperiksa menggunakan kondisi $\$i \% 2 == 0$ untuk menentukan apakah angka tersebut genap. Jika benar, angka tersebut akan ditampilkan. Setelah itu, variabel **\$a** diisi dengan 10 dan **\$b** diisi dengan 50, lalu ditampilkan teks keterangan sebelum memanggil fungsi. Hasil akhirnya adalah daftar bilangan genap dari 10 sampai 50.

Output



i. Function

```
<?php
function luasSegitiga($alas, $tinggi) {
    return 0.5 * $alas * $tinggi;
}
//pemanggilan fungsi
$a = 5;
$t = 35;
echo "Luas Segitiga dengan alas $a dan tinggi $t adalah : " . luasSegitiga($a, $t);
```



Penjelasan:

Kode PHP di atas membuat sebuah fungsi bernama **luasSegitiga()** yang bertugas menghitung luas segitiga berdasarkan nilai alas dan tinggi. Fungsi tersebut menerima dua parameter, yaitu **\$alas** dan **\$tinggi**, kemudian mengembalikan hasil perhitungan menggunakan rumus luas segitiga yaitu $\frac{1}{2} \times \text{alas} \times \text{tinggi}$. Setelah fungsi dibuat, variabel **\$a** diisi dengan nilai 5 dan **\$t** dengan nilai 35. Kemudian fungsi dipanggil dan hasil perhitungannya ditampilkan dalam kalimat lengkap. Program ini menunjukkan cara sederhana untuk membuat dan menggunakan fungsi dengan nilai kembali (return) di PHP.

Output



Luas Segitiga dengan alas 5 dan tinggi 35 adalah : 87.5

j. Array

```
<?php
$arrKendaraan = ["Mobil", "Pesawat", "Kereta Api", "Kapal Laut"];
echo $arrKendaraan[0] . "<br>"; //Mobil
echo $arrKendaraan[2] . "<br>"; //Kereta Api

$arrKota = [];
$arrKota[] = "Jakarta";
$arrKota[] = "Medan";
$arrKota[] = "Bandung";
$arrKota[] = "Malang";
$arrKota[] = "Sulawesi";

array_push($arrKota, "Padang", "Riau");

echo $arrKota[1] . "<br>"; //Medan
echo $arrKota[2] . "<br>"; //Bandung
echo $arrKota[4] . "<br>"; //Sulawesi
echo $arrKota[5] . "<br>";
```

```
echo $arrKota[6] . "<br>";
```

```
1  <?php
2  $arrKendaraan = ["Mobil", "Pesawat", "Kereta Api", "Kapal Laut"];
3  echo $arrKendaraan[0] . "<br>"; //Mobil
4  echo $arrKendaraan[2] . "<br>"; //Kereta Api
5
6  $arrKota = [];
7  $arrKota[] = "Jakarta";
8  $arrKota[] = "Medan";
9  $arrKota[] = "Bandung";
10 $arrKota[] = "Malang";
11 $arrKota[] = "Sulawesi";
12
13 array_push($arrKota, "Padang", "Riau");
14
15 echo $arrKota[1] . "<br>"; //Medan
16 echo $arrKota[2] . "<br>"; //Bandung
17 echo $arrKota[4] . "<br>"; //Sulawesi
18 echo $arrKota[5] . "<br>";
19 echo $arrKota[6] . "<br>";
```

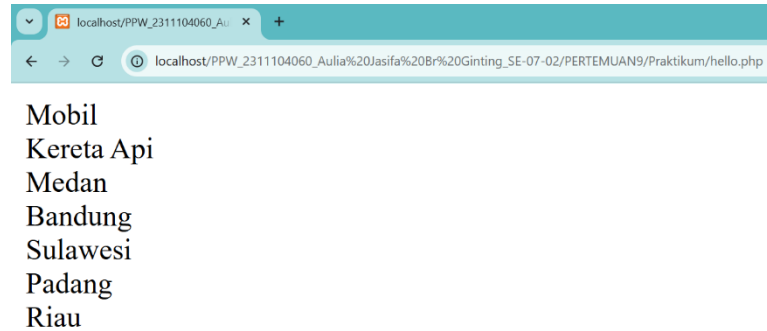
```
134 $arrKendaraan = ["Mobil", "Pesawat", "Kereta Api", "Kapal Laut"];
135 echo $arrKendaraan[0] . "<br>"; //Mobil
136 echo $arrKendaraan[2] . "<br>"; //Kereta Api
137
138 $arrKota = [];
139 $arrKota[] = "Jakarta";
140 $arrKota[] = "Medan";
141 $arrKota[] = "Bandung";
142 $arrKota[] = "Malang";
143 $arrKota[] = "Sulawesi";
144
145 array_push($arrKota, "Padang", "Riau");
146
147 echo $arrKota[1] . "<br>"; //Medan
148 echo $arrKota[2] . "<br>"; //Bandung
149 echo $arrKota[4] . "<br>"; //Sulawesi
150 echo $arrKota[5] . "<br>";
151 echo $arrKota[6] . "<br>";
```

Penjelasan:

Kode PHP di atas menunjukkan cara bekerja dengan array dalam PHP, mulai dari membuat array, menambah elemen, hingga menampilkan data berdasarkan indeksnya. Pertama, sebuah array **\$arrKendaraan** dibuat dengan empat nilai awal, lalu dua data ditampilkan menggunakan indeks 0 dan 2. Selanjutnya, array kosong **\$arrKota** dibuat dan beberapa kota ditambahkan satu per satu menggunakan sintaks **\$arrKota[] =** Setelah itu, fungsi **array_push()** digunakan untuk menambahkan dua elemen tambahan sekaligus, yaitu “Padang” dan “Riau”. Beberapa elemen array kemudian ditampilkan menggunakan indeks tertentu, seperti Medan (indeks 1), Bandung (indeks 2), hingga elemen yang baru

ditambahkan—Padang berada di indeks 5 dan Riau di indeks 6. Program ini memperlihatkan cara menambah dan mengakses data dalam array secara fleksibel.

Output



k. Array

```
<?php
$arrAlamat = [
    "Rona" => "Banjarmasin",
    "Dhiva" => "Bandung",
    "Ilham" => "Medan",
    "Oku" => "Hongkong",
];

echo $arrAlamat["Dhiva"] . "<br>"; //Bandung
echo $arrAlamat['Oku'] . "<br>"; //Hongkong

$arrNim = [];
$arrNim["Rona"] = "11011112";
$arrNim["Dhiva"] = "11011101";
$arrNim["Ilham"] = "11011309";
$arrNim["Oku"] = "11014765";
$arrNim["Fadhlan"] = "11011113";

echo $arrNim["Ilham"] . "<br>"; //11011309
echo $arrNim['Fadhlan'] . "<br>"; //11011113
```

```

1  <?php
2  $arrAlamat = [
3      "Rona" => "Banjarmasin",
4      "Dhiva" => "Bandung",
5      "Ilham" => "Medan",
6      "Oku" => "Hongkong",
7  ];
8
9  echo $arrAlamat["Dhiva"] . "<br>"; //Bandung
10 echo $arrAlamat['Oku'] . "<br>"; //Hongkong
11
12 $arrNim = [];
13 $arrNim["Rona"] = "11011112";
14 $arrNim["Dhiva"] = "11011101";
15 $arrNim["Ilham"] = "11011309";
16 $arrNim["Oku"] = "11014765";
17 $arrNim["Fadhlan"] = "11011113";
18
19 echo $arrNim["Ilham"] . "<br>"; //11011309
20 echo $arrNim['Fadhlan'] . "<br>"; //11011113

```

```

154 $arrAlamat = [
155     "Rona" => "Banjarmasin",
156     "Dhiva" => "Bandung",
157     "Ilham" => "Medan",
158     "Oku" => "Hongkong",
159 ];
160
161 echo $arrAlamat["Dhiva"] . "<br>"; //Bandung
162 echo $arrAlamat['Oku'] . "<br>"; //Hongkong
163
164 $arrNim = [];
165 $arrNim["Rona"] = "11011112";
166 $arrNim["Dhiva"] = "11011101";
167 $arrNim["Ilham"] = "11011309";
168 $arrNim["Oku"] = "11014765";
169 $arrNim["Fadhlan"] = "11011113";
170
171 echo $arrNim["Ilham"] . "<br>"; //11011309
172 echo $arrNim['Fadhlan'] . "<br>"; //11011113

```

Penjelasan:

Kode PHP di atas memperlihatkan cara membuat dan menggunakan **array asosiatif** di PHP, yaitu array yang menggunakan *key* berupa teks (string) untuk

mengakses datanya. Array **\$arrAlamat** dibuat dengan pasangan *key-value* yang berisi nama sebagai *key* dan kota sebagai nilainya. Dua data kemudian ditampilkan, misalnya nilai dengan *key* "Dhiva" menghasilkan "Bandung" dan *key* "Oku" menghasilkan "Hongkong".

Selanjutnya, array asosiatif kedua bernama **\$arrNim** dibuat untuk menyimpan data NIM berdasarkan nama. Setiap nama dijadikan *key* dan diberikan nilai NIM masing-masing. Dua data juga ditampilkan, yaitu NIM milik Ilham dan Fadhlan. Contoh ini menunjukkan bahwa array asosiatif sangat berguna untuk menyimpan data yang membutuhkan identifikasi jelas menggunakan *key* daripada angka indeks biasa.

Output



B. UNGUIDED (Tugas Mandiri)

(Bagian ini berisi latihan atau tugas tambahan yang diberikan di akhir modul. Tuliskan nomor soal, isi soalnya, source code jawaban, screenshot output, dan penjelasan setiap nomor.)

1. Soal 1:

Buat program konversi suhu yang dapat mengkonversi:

- Celcius ke Fahrenheit
- Fahrenheit ke Celcius
- Celcius ke Kelvin

Tampilkan menggunakan 2 desimal di belakang koma

Jawaban:

Source Code:

```
<?php
// 1. Program Konversi Suhu
```

```
function celsiusToFahrenheit($celsius) {
    return ($celsius * 9/5) + 32;
}

function fahrenheitToCelsius($fahrenheit) {
    return ($fahrenheit - 32) * 5/9;
}

function celsiusToKelvin($celsius) {
    return $celsius + 273.15;
}

$celsius = 25;
$fahrenheit = 77;
echo "Konversi Suhu:<br>";
echo "Celcius ke Fahrenheit: " . number_format(celsiusToFahrenheit($celsius),
2) . " °F<br>";
echo "Fahrenheit ke Celcius: " .
number_format(fahrenheitToCelsius($fahrenheit), 2) . " °C<br>";
echo "Celcius ke Kelvin: " . number_format(celsiusToKelvin($celsius), 2) . "
K<br>";
echo "<br>";
```



```
1 <?php
2 // 1. Program Konversi Suhu
3 function celsiusToFahrenheit($celsius) {
4     return ($celsius * 9/5) + 32;
5 }
6
7 function fahrenheitToCelsius($fahrenheit) {
8     return ($fahrenheit - 32) * 5/9;
9 }
10
11 function celsiusToKelvin($celsius) {
12     return $celsius + 273.15;
13 }
14
15 $celsius = 25;
16 $fahrenheit = 77;
17 echo "Konversi Suhu:<br>";
18 echo "Celcius ke Fahrenheit: " . number_format(celsiusToFahrenheit($celsius), 2) . " °F<br>";
19 echo "Fahrenheit ke Celcius: " . number_format(fahrenheitToCelsius($fahrenheit), 2) . " °C<br>";
20 echo "Celcius ke Kelvin: " . number_format(celsiusToKelvin($celsius), 2) . " K<br>";
21 echo "<br>";
```

Output:



Konversi Suhu:

Celcius ke Fahrenheit: 77.00 °F

Fahrenheit ke Celcius: 25.00 °C

Celcius ke Kelvin: 298.15 K

Penjelasan:

Kode PHP di atas merupakan program sederhana untuk melakukan konversi suhu antara tiga satuan, yaitu Celsius, Fahrenheit, dan Kelvin. Pertama, terdapat beberapa fungsi yang masing-masing memiliki tugas khusus. Fungsi `celsiusToFahrenheit()` menerima nilai suhu dalam Celsius dan mengubahnya ke Fahrenheit dengan rumus $F = (C \times 9/5) + 32$. Fungsi kedua, `fahrenheitToCelsius()`, bekerja kebalikan dari sebelumnya, yaitu mengubah Fahrenheit menjadi Celsius menggunakan rumus $(F - 32) \times 5/9$. Selanjutnya, fungsi `celsiusToKelvin()` mengonversi suhu dari Celsius ke Kelvin dengan cara menambahkan angka 273.15 sesuai standar konversi.

Setelah fungsi-fungsi tersebut didefinisikan, program membuat dua variabel, yaitu `$celsius = 25` dan `$fahrenheit = 77`, yang menjadi contoh data untuk dikonversi. Kemudian program menampilkan judul "Konversi Suhu:" dan memanggil masing-masing fungsi untuk menghitung hasil konversi. Fungsi `number_format(..., 2)` digunakan untuk membatasi output menjadi dua angka di belakang koma agar hasilnya lebih rapi. Hasil konversi kemudian dicetak ke layar beserta satuan ukurannya seperti °F, °C, dan K. Pada bagian akhir, `echo "
";` digunakan untuk memberikan jarak baris. Secara keseluruhan, kode ini menunjukkan cara membuat fungsi, memproses data, dan menampilkan hasil konversi suhu secara terstruktur.

2. Soal 2:

Kalkulator Diskon:

Input: Total belanja

- Diskon 10% jika belanja \geq Rp 100.000
- Diskon 20% jika belanja \geq Rp 500.000
- Diskon 30% jika belanja \geq Rp 1.000.000

Tampilkan: total belanja, diskon, dan total bayar

Jawaban:

Source Code:

```
<?php
// 2. Kalkulator Diskon
$totalBelanja = 750000;
$diskon = 0;
if ($totalBelanja >= 1000000) {
    $diskon = 0.30;
} elseif ($totalBelanja >= 500000) {
    $diskon = 0.20;
} elseif ($totalBelanja >= 100000) {
    $diskon = 0.10;
}
$jumlahDiskon = $totalBelanja * $diskon;
$totalBayar = $totalBelanja - $jumlahDiskon;
echo "Kalkulator Diskon:<br>";
echo "Total Belanja: Rp " . number_format($totalBelanja, 0, ',', '.') . "<br>";
echo "Diskon: Rp " . number_format($jumlahDiskon, 0, ',', '.') . " (" . ($diskon * 100) . "%)<br>";
echo "Total Bayar: Rp " . number_format($totalBayar, 0, ',', '.') . "<br>";
echo "<br>";
```



```
1 <?php
2 // 2. Kalkulator Diskon
3 $totalBelanja = 750000;
4 $diskon = 0;
5 if ($totalBelanja >= 1000000) {
6     $diskon = 0.30;
7 } elseif ($totalBelanja >= 500000) {
8     $diskon = 0.20;
9 } elseif ($totalBelanja >= 100000) {
10     $diskon = 0.10;
11 }
12 $jumlahDiskon = $totalBelanja * $diskon;
13 $totalBayar = $totalBelanja - $jumlahDiskon;
14 echo "Kalkulator Diskon:<br>";
15 echo "Total Belanja: Rp " . number_format($totalBelanja, 0, ',', '.') . "<br>";
16 echo "Diskon: Rp " . number_format($jumlahDiskon, 0, ',', '.') . " (" . ($diskon * 100) . "%)<br>";
17 echo "Total Bayar: Rp " . number_format($totalBayar, 0, ',', '.') . "<br>";
18 echo "<br>";
```


Output:



Kalkulator Diskon:

Total Belanja: Rp 750.000

Diskon: Rp 150.000 (20%)

Total Bayar: Rp 600.000

Penjelasan:

Kode PHP di atas berfungsi sebagai kalkulator diskon untuk menghitung total pembayaran setelah diskon diberikan berdasarkan jumlah total belanja. Program dimulai dengan mendefinisikan variabel `$totalBelanja` sebesar **750.000 rupiah**, lalu variabel `$diskon` di-set ke 0 sebagai nilai awal. Selanjutnya, kode menggunakan struktur kondisi `if`, `elseif`, dan `else` untuk menentukan besarnya diskon. Jika total belanja mencapai **1.000.000 rupiah atau lebih**, diskon yang diberikan adalah **30%**. Jika total belanja berada di antara **500.000 – 999.999 rupiah**, diskonnya **20%**. Jika total belanja minimal **100.000 rupiah**, diskonnya **10%**. Karena pada contoh nilai `$totalBelanja` adalah 750.000, maka program memilih diskon 20%.

Setelah itu, program menghitung jumlah diskon melalui `$jumlahDiskon = $totalBelanja * $diskon`, dan menghitung total pembayaran setelah diskon melalui `$totalBayar = $totalBelanja - $jumlahDiskon`. Untuk menampilkan hasilnya, program mencetak informasi berupa total belanja, jumlah diskon yang diterima, persentasenya, serta total yang harus dibayar. Fungsi `number_format()` digunakan agar angka tampil dengan format rupiah yang rapi (misalnya menggunakan tanda titik sebagai pemisah ribuan). Pada akhir kode, terdapat `echo "
";` untuk memberikan jarak baris. Dengan demikian, program ini memperlihatkan cara menghitung dan menampilkan diskon belanja secara otomatis berdasarkan nilai pembelian.

3. Soal 3:

Manipulasi Array:

Buat array nilai mahasiswa: [75, 89, 65, 90, 85, 70, 98, 65, 69, 70, 12]

Tampilkan:

- Nilai tertinggi
- Nilai terendah
- Rata-rata nilai Jumlah mahasiswa yang lulus (≥ 70)

Urutkan nilai dari tertinggi ke terendah

Jawaban:

Source Code:

```
<?php
$nilaiMahasiswa = [75, 89, 65, 90, 85, 70, 98, 65, 69, 70, 12];

$nilaiTertinggi = max($nilaiMahasiswa);
$nilaiTerendah = min($nilaiMahasiswa);
$rataRata = array_sum($nilaiMahasiswa) / count($nilaiMahasiswa);

$jumlahLulus = 0;
foreach ($nilaiMahasiswa as $nilai) {
    if ($nilai >= 70) {
        $jumlahLulus++;
    }
}

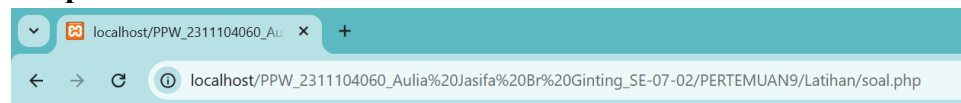
$nilaiUrut = $nilaiMahasiswa;
rsort($nilaiUrut);
echo "Manipulasi Array Nilai Mahasiswa:<br>";
echo "Nilai Tertinggi: " . $nilaiTertinggi . "<br>";
echo "Nilai Terendah: " . $nilaiTerendah . "<br>";
echo "Rata-rata Nilai: " . number_format($rataRata, 2) . "<br>";
echo "Jumlah Mahasiswa Lulus (>=70): " . $jumlahLulus . "<br>";
echo "Nilai Urut dari Tertinggi ke Terendah: " . implode(" ", $nilaiUrut) .
"<br>";
```

```

1 <?php
2 // 3. Manipulasi Array
3 $nilaiMahasiswa = [75, 89, 65, 90, 85, 70, 98, 65, 69, 70, 12];
4
5 $nilaiTertinggi = max($nilaiMahasiswa);
6 $nilaiTerendah = min($nilaiMahasiswa);
7 $rataRata = array_sum($nilaiMahasiswa) / count($nilaiMahasiswa);
8
9 $jumlahLulus = 0;
10 foreach ($nilaiMahasiswa as $nilai) {
11     if ($nilai >= 70) {
12         $jumlahLulus++;
13     }
14 }
15
16 $nilaiUrut = $nilaiMahasiswa;
17 rsort($nilaiUrut);
18 echo "Manipulasi Array Nilai Mahasiswa:<br>";
19 echo "Nilai Tertinggi: " . $nilaiTertinggi . "<br>";
20 echo "Nilai Terendah: " . $nilaiTerendah . "<br>";
21 echo "Rata-rata Nilai: " . number_format($rataRata, 2) . "<br>";
22 echo "Jumlah Mahasiswa Lulus (>=70): " . $jumlahLulus . "<br>";
23 echo "Nilai Urut dari Tertinggi ke Terendah: " . implode(" ", $nilaiUrut) . "<br>";
24 >

```

Output:



Manipulasi Array Nilai Mahasiswa:
 Nilai Tertinggi: 98
 Nilai Terendah: 12
 Rata-rata Nilai: 71.64
 Jumlah Mahasiswa Lulus (>=70): 7
 Nilai Urut dari Tertinggi ke Terendah: 98, 90, 89, 85, 75, 70, 70, 69, 65, 65, 12

Penjelasan:

Kode PHP di atas digunakan untuk melakukan berbagai manipulasi pada sebuah array yang berisi daftar nilai mahasiswa. Array \$nilaiMahasiswa menyimpan sejumlah nilai berupa angka. Program kemudian menggunakan fungsi max() untuk mencari nilai tertinggi dan menyimpannya dalam variabel \$nilaiTertinggi, serta fungsi min() untuk menentukan nilai terendah dan menyimpannya dalam \$nilaiTerendah. Untuk menghitung rata-rata nilai, program menjumlahkan seluruh isi array menggunakan array_sum(), lalu membaginya dengan jumlah elemen array yang diperoleh dari count(), dan hasilnya disimpan dalam variabel \$rataRata.

Selanjutnya, program menghitung berapa banyak mahasiswa yang dinyatakan lulus dengan nilai minimal 70. Hal ini dilakukan melalui perulangan foreach yang memeriksa setiap nilai dalam array, lalu menambahkan satu ke variabel \$jumlahLulus setiap kali menemukan nilai ≥ 70 . Untuk menampilkan nilai-nilai yang telah diurutkan dari yang tertinggi ke terendah, program menyalin array asli ke dalam \$nilaiUrut, kemudian menggunakan fungsi rsort() untuk mengurutkannya secara descending. Setelah semua proses selesai, program menampilkan hasilnya ke layar, termasuk nilai tertinggi, nilai terendah, rata-rata nilai, jumlah mahasiswa lulus, serta daftar nilai yang sudah terurut. Fungsi implode(" ", \$nilaiUrut) digunakan untuk mengubah array menjadi teks yang dipisahkan koma agar lebih mudah dibaca. Dengan ini, program mampu mengolah data nilai mahasiswa secara lengkap dan terstruktur.

BAB III

PENUTUP

A. Kesimpulan

Pada praktikum ini, saya mempelajari berbagai konsep dasar pemrograman PHP yang berkaitan dengan penggunaan fungsi, percabangan, perulangan, serta manipulasi array. Melalui beberapa program yang dibuat, seperti konversi suhu, kalkulator diskon, dan analisis nilai mahasiswa, saya mendapatkan pemahaman tentang bagaimana PHP memproses data, melakukan perhitungan, serta menampilkan hasil secara dinamis. Hal penting yang dipelajari adalah penggunaan fungsi echo, fungsi matematika seperti max(), min(), dan array_sum(), pemanfaatan struktur if-elseif-else untuk pengambilan keputusan, serta teknik pengolahan array dengan foreach, rsort(), dan implode(). Selain itu, praktikum ini membantu saya memahami bagaimana memformat angka menggunakan number_format(), serta bagaimana membangun alur logika yang terstruktur untuk menyelesaikan suatu permasalahan. Setiap program memberikan gambaran nyata bagaimana PHP dapat digunakan untuk kebutuhan komputasi sederhana seperti konversi nilai, perhitungan diskon, maupun analisis data.