

Nama : Fikri Maulana  
Kelas ABP : IF-44-01 / IF-44-10  
NIM : 1301200239

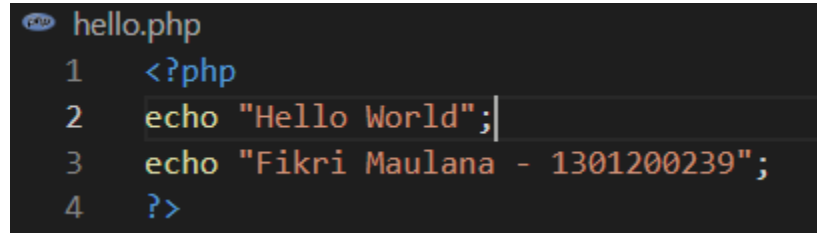
## MODUL 5

### 5.1 Web Server dan Server Side Scripting

Web Server merupakan sebuah perangkat lunak dalam server yang berfungsi menerima permintaan(request) berupa halaman web melalui HTTP atau HTTPS dari client yang dikenal dengan web browser dan mengirimkan kembali (response) hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.

### 5.2 Pengenalan PHP

Setiap satu statement (perintah) biasanya diakhiri dengan titik-koma (;). PHP juga case sensitive untuk nama identifier yang dibuat oleh user sedangkan identifier bawaan dari PHP tidak case sensitive.



```
hello.php
1  <?php
2  echo "Hello World";
3  echo "Fikri Maulana - 1301200239";
4  ?>
```

Hello World Fikri Maulana - 1301200239

### 5.3 Variable

Pada PHP, tipe data dari suatu variabel tidak didefinisikan langsung oleh programmer, akan tetapi secara otomatis akan ditentukan oleh interpreter PHP. Namun demikian, PHP mendukung 8 (delapan) buah tipe data primitif, yaitu :

1. Boolean
2. Integer
3. Float
4. String
5. Array
6. Object
7. Resource
8. NULL

```

1  <?php
2      $nim = "1301200239";
3      $nama = "Fikri Maulana";
4
5      echo "NIM : " . $nim;
6      echo "Nama : " . $nama;
7  ?>

```

NIM : 1301200239 Nama : Fikri Maulana

## 5.4 Konstanta

Konstanta merupakan variabel konstan yang nilainya tidak berubah-ubah. Untuk mendefinisikan konstanta pada PHP, dapat menggunakan fungsi define() yang telah tersedia pada PHP

```

1  <?php
2  define("NAMA" , "Fikri Maulana");
3  define("NIM" , "1301200239");
4  echo "Nama : " . NAMA;
5  echo "NIM : " . NIM;
6  ?>

```

Nama : Fikri Maulana NIM : 1301200239

## 5.5 Operator dalam PHP

PHP memiliki beberapa jenis operator seperti:

- Aritmetika : + , - , \* , / , %
- Penugasan : =
- Bitwise : & , | , ^ , ~ , << , >>
- Perbandingan : == , === , != , <> , != , < , > , <= , >=
- Logika : and , && , or , || , xor
- String : .

```

1  <?php
2      $a = 10;
3      $b = 20;
4      $c = $a + $b;
5      $d = $b + $a;
6      $nama1 = "Fikri";
7      $nama2 = "Maulana";
8      echo $a , "<br>";
9      $nama3 = "IF 44 10";
10     echo $c , "<br>";
11     var_dump($a == $b);
12     echo "<br>";
13     var_dump($a > $b);
14     echo "<br>";
15     if ($a == 10 && $d == 30){
16         echo "logika && bekerja <br>";
17     };
18     $nama = $nama1 . $nama2 . $nama3;
19     echo $nama;
20  ?>

```

10

30

bool(false)

bool(false)

berhasil run

FikriMaulana IF 44 10

## 5.6 Struktur Kondisi

```

1  <?php
2      $a = 10; $b = 20;
3      if ($a < $b){
4          echo " a lebih kecil dari b <br>";
5      } else{
6          echo " a lebih besar dari a <br>";
7      }
8
9      switch($b){
10         case 2:
11             echo "b bernilai 2 <br>";
12             break;
13         case 3:
14             echo "b bernilai 3 <br>";
15         }
16         echo "Fikri Maulana - 1301200239"
17     ?>

```

a lebih kecil dari b

Fikri Maulana - 1301200239

## 5.7 Perulangan

```
1 <?php
2     echo "Fikri Maulana - 1301200239"
3     for($i = 1; $i <= 10; $i+=2){
4         echo "Ini perulangan for ke - " . $i . "<br>";
5     }
6     $i = 1;
7     while($i <= 5){
8         echo "Ini perulangan while ke - " . $i . "<br>";
9         $i += 1;
10    }
11    $i = 1;
12    do{
13        echo "Ini perulangan do - while ke - " . $i . "<br>";
14        $i += 3;
15    } while ($i < 10);
16 ?>
```

Fikri Maulana - 1301200239

Ini perulangan for ke - 1

Ini perulangan for ke - 3

Ini perulangan for ke - 5

Ini perulangan for ke - 7

Ini perulangan for ke - 9

Ini perulangan while ke - 1

Ini perulangan while ke - 2

Ini perulangan while ke - 3

Ini perulangan while ke - 4

Ini perulangan while ke - 5

Ini perulangan do - while ke - 1

Ini perulangan do - while ke - 4

Ini perulangan do - while ke - 7

## 5.8 Function

```

1  <?php
2      echo "Fikri Maulana - 1301200239";
3      function cetak_ganjil(){
4          for($i = 1; $i <= 10; $i++){
5              if($i % 2 == 0){
6                  echo $i . " adalah bilangan genap <br>";
7              }
8          }
9      }
10     function cek_ganjil_atau_genap($a){
11         if($a % 2 == 1){
12             echo "<br><br>" . $a . " adalah bilangan ganjil";
13         }else{
14             echo "<br><br>" . $a . " adalah bilangan genap";
15         }
16     }
17     cetak_ganjil();
18     cek_ganjil_atau_genap(459)
19 ?>

```

Fikri Maulana - 13012002392 adalah bilangan genap  
 4 adalah bilangan genap  
 6 adalah bilangan genap  
 8 adalah bilangan genap  
 10 adalah bilangan genap

459 adalah bilangan ganjil

## 5.9 Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun array disebut elemen array, yang masing-masing elemen dapat diakses tersendiri melalui index array. Index array dapat berupa bilangan integer atau string. Untuk mendeklarasikan atau mendefinisikan sebuah array di PHP bisa menggunakan keyword `array()`.

```

1  <?php
2      $arrNama = array("Fikri", "Maulana", "1301200239", "IF4410");
3      echo $arrNama[3] . "<br>";
4      echo $arrNama[0] . "<br>";
5
6      $arrKota = array();
7      $arrKota[] = 'Bandung';
8      $arrKota[] = 'Garut';
9      $arrKota[] = 'Jakarta';
10     $arrKota[] = 'Yogyakarta';
11     echo $arrKota[0] . "<br>";
12     echo $arrKota[1] . "<br>";
13     echo $arrKota[2] . "<br>";
14     echo $arrKota[3] . "<br>";
15  ?>

```

```

IF4410
Fikri
Bandung
Garut
Jakarta
Yogyakarta

```

## 5.10 GET dan POST

Pada dasarnya GET dan POST adalah metode pengiriman yang digunakan dalam proses pengiriman data pada Form.

### 5.10.1 Metode GET

```

1  <?php
2      echo "Fikri Maulana - 1301200239";
3  ?>
4  <form method = "GET">
5      <input type = "text" name = "nama">
6      <input type = "submit" name = "kirim">
7  <pre>
8      <?php print_r($_GET) ?>
9  </pre>
10 </form>
11

```

### 5.10.2 Metode POST

Berbanding terbalik dengan GET, pada metode POST data yang dikirim tidak akan ditampilkan pada url browser, sehingga metode pengiriman ini lebih aman untuk data yang sifatnya sensitif seperti password atau nomor rekening. Metode POST lebih sering digunakan dalam hal registrasi user atau login.

```
1 <?php
2     echo "Fikri Maulana - 1301200239";
3     ?>
4 <form method = "POST">
5     <p>username : <input type="text" name="user" placeholder="username"></p>
6     <p>password : <input type="password" name="pass" placeholder="password"></p>
7     <p><input type="submit" name="kirim"></p>
8     <pre>
9         <?php print_r($_POST) ?>
10    </pre>
11 </form>
12
```

## 5.11 XML

### 5.11.1 Pengertian XML

XML (Extensible Markup Language) adalah sebuah bahasa markup yang digunakan untuk menyimpan dan mengirim data dalam bentuk teks. XML digunakan untuk menggambarkan struktur data, bukan untuk menentukan tampilan atau tindakan seperti pada HTML.

XML digunakan untuk pertukaran data antar aplikasi dan sistem yang berbeda, karena XML memiliki format standar yang dapat diinterpretasikan oleh berbagai jenis program dan platform. XML juga digunakan untuk membuat dokumen yang mudah diakses dan dapat dimanipulasi dengan mudah.

Dalam XML, data disajikan dalam bentuk elemen-elemen (tags) yang diatur dalam struktur hierarki. Setiap elemen terdiri dari tag pembuka dan tag penutup yang diapit oleh data atau elemen lain yang lebih dalam. Elemen-elemen tersebut dapat memiliki atribut yang memberikan informasi tambahan tentang elemen tersebut.

### 5.11.2 Sintaks HTML

```
1 <?xml version="1.0" encoding="UTF-8" standalone='no' ?>
2 <Warga-Telyu>
3     <Pekerjaan = "Mahasiswa">
4         <nama>Fikri Maulana</nama>
5         <ID>1301200239</ID>
6     </Pekerjaan>
7     <Pekerjaan = "Dosen">
8         <nama>Agus Setia </nama>
9         <ID>1312312312</ID>
10    </Pekerjaan>
11 </Warga-Telyu>
```

## 5.12 JSON

### 5.12.1 Pengertian JSON

JSON (JavaScript Object Notation) adalah sebuah format data yang digunakan untuk pertukaran data antar aplikasi. JSON sering digunakan dalam aplikasi web untuk mengirim data antar browser dan server.

#### 5.12.2 Perbedaan JSON dan XML

Suatu objek pada JSON dinotasikan dengan simbol kurung kurawal ({}), sedangkan kumpulan dari beberapa objek dapat dihimpun dalam notasi kurung siku ([]).

JSON :

```
1  {
2    "Mahasiswa": [
3      {
4        "Nama" : "Fikri Maulana",
5        "ID" : "1301200239"
6      },
7      {
8        "Nama" : "Agus Setia",
9        "ID" : "1231231233"
10     }
11   ]
12 }
```

XML :

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <root>
3    <Mahasiswa>
4      <Nama>Fikri Maulana</Nama>
5      <ID>1301200239</ID>
6    </Mahasiswa>
7    <Mahasiswa>
8      <Nama>Agus Setia</Nama>
9      <ID>1231231233</ID>
10   </Mahasiswa>
11 </root>
12
```

### 5.13 AJAX

#### 5.13.1 Ajax

AJAX (Asynchronous JavaScript and XML) adalah teknologi pengembangan web yang memungkinkan pengiriman permintaan (request) data ke server dan menerima respons (response) dari server secara asinkronus (tanpa perlu memuat ulang halaman web secara keseluruhan).



### 5.13.2 Cara Kerja

Berikut adalah langkah-langkah kerja AJAX:

Event handler: AJAX diaktifkan saat terjadi suatu peristiwa (event) pada halaman web, seperti klik tombol atau mengisi formulir

Object instantiation: JavaScript membuat objek XMLHttpRequest yang akan digunakan untuk berkomunikasi dengan server.

Open request: Setelah objek XMLHttpRequest dibuat, JavaScript mengirimkan permintaan (request) ke server dengan memanggil metode open() pada objek XMLHttpRequest. Pada tahap ini, kita menentukan jenis permintaan yang akan dikirimkan (GET atau POST), URL tujuan, dan apakah permintaan tersebut bersifat asynchronus atau tidak.

Send request: Setelah permintaan dibuka, permintaan dikirimkan ke server dengan memanggil metode send() pada objek XMLHttpRequest. Jika permintaan bersifat asynchronus, maka JavaScript akan melanjutkan eksekusi kode selanjutnya tanpa harus menunggu respons dari server.

Receive response: Setelah server menerima permintaan, server akan memproses permintaan tersebut dan mengirimkan respons (response) kembali ke klien. Respons dari server bisa berupa data dalam format XML, JSON, atau teks biasa. Setelah respons diterima, JavaScript akan memproses respons tersebut dan mengubah isi halaman web sesuai dengan respons dari server.

Update UI: Setelah respons dari server diterima dan diproses oleh JavaScript, halaman web akan di-update secara dinamis tanpa perlu memuat ulang halaman secara keseluruhan.

### 5.13.3 Event Handling

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h2>The XML Http Request Object</h2>
5 <p id="demo">Tekan tombol</p>
6 <button type="button" onclick="loadDoc()">Change Content</button>
7 <script> function loadDoc() { var xhttp = new XMLHttpRequest();
8 xhttp.onreadystatechange = function() { if (this.readyState == 4 &&
9 this.status == 200) {
10 document.getElementById("demo").innerHTML = this.responseText;
11 } }; xhttp.open("GET", "5-13-4-Event-Handling.txt", true); xhttp.send();
12 }
13 </script>
14 </body>
15 </html>
```