

BAB 2

Dasar PHP, Variabel, Tipe Data

Taryana Suryana M.Kom
Teknik Informatika
2023/2024

taryanarx@email.unikom.ac.id

taryanarx@gmail.com

[HP.081221480577](tel:081221480577) - [WA.087832908499](tel:087832908499)

2.1.Mengenal PHP

2.1.1. Sejarah Singkat PHP

Pada awalnya, PHP merupakan kependekan dari Personal Home Page karena hanya digunakan pribadi oleh pembuatnya. PHP pertama kali dibuat oleh Rasmus Lerdorf, seorang programmer dari Denmark pada tahun 1995. Karena PHP bersifat open source, banyak yang tertarik mengembangkan kode PHP ini. Salah satunya adalah perusahaan yang bernama Zend. Akhirnya Zend merilis PHP versi 2.0 pada tahun 1997. Setahun kemudian, PHP 3.0 juga dirilis.

Tidak berhenti sampai disitu, perkembangan PHP semakin maju dengan dirilisnya PHP 4.0, di mana versi ini banyak digunakan oleh programmer karena mempunyai kemampuan untuk membangun aplikasi web dengan kecepatan dan stabilitas yang tinggi. Kemudian Juni 2004, Zend kembali merilis PHP versi 5.0 yang lebih dikenal dengan model OOP (Object Oriented Programming).

Banyak kelebihan yang bisa diperoleh jika anda menggunakan PHP. Berikut ini kelebihan PHP dibandingkan bahasa pemrograman yang lain:

1. Gratis. Karena bersifat open source, PHP dapat kita gunakan dan dapat kita kembangkan secara bebas.
2. Lebih cepat. Tidak diragukan lagi bahwa PHP lebih cepat dibandingkan dengan aplikasi CGI dengan Perl atau Python, bahkan dengan ASP maupun Java dalam berbagai aplikasi web.
3. Mudah dipelajari atau digunakan. Banyak sekali forum-forum yang berisi tutorial belajar PHP ataupun diskusi yang membahas tentang PHP. Jika Anda ingin belajar PHP, Anda tidak perlu khawatir karena bisa mempelajarinya lewat forum maupun situs yang ada. (<http://phphostingindonesia.com>)

2.1.2. Sintak PHP

Untuk menuliskan sesuatu dalam format PHP dapat dilakukan dengan cara sebagai berikut:

Latihan1.php

```
<h1>Program Pertama</h1>
<?php
echo "Belajar PHP!";
?>
```

Simpan Code Program tersebut di folder

c:/xampp/htdocs/HI/latihan1.php

Untuk melihat hasilnya dapat dilakukan dengan cara pada address bar browser ketikkan:localhost/HI/latihan1.php

Ini adalah output yang dihasilkan dari program di atas.

Program Pertama

Belajar PHP!

Gambar 2.1.Hasil Latihan1.php

2.1.3. Pemisahan Instruksi

Pemisahan instruksi dalam PHP sama dengan pemisahan intruksi yang digunakan dalam bahasa C atau PERL, yaitu dengan menggunakan titik koma sebagai penutup tiap statement dan diakhiri dengan tag (?>) dan juga berfungsi sebagai akhir statement.

```
<h1>Instruksi Pemisahan</h1>
<p>
<?php
    echo "Cara Pemisahan Pertama";
?>
<p>
<?php echo "Cara Pemisahan Kedua" ?>
```

Kode Program: latihan2.php

Instruksi Pemisahan

Cara Pemisahan Pertama

Cara Pemisahan Kedua

Gambar 2.2. Pemisahan Instruksi

2.1.4. Komentar

Untuk menambahkan baris komentar dalam PHP dapat digunakan beberapa style yang didukung oleh PHP ini, yaitu komentar 'C', 'C++' dan Unix Shell-style, contoh:

```
<html>
<body>
Menulis Komentar
<?php
// Satu Baris Komen
# Satu baris Komen Juga

/*
Menuliskan Komen yang lebih dari
```

```
satu baris
*/

// Atau bisa juga komen ditulis diantara code program
$x = 5 /* + 15 */ + 5;
echo $x;
?>
</body>
</html>
```

Kode Program: latihan3.php

2.2 Tipe data dan Variabel

Pemrograman PHP mendukung beberapa tipe data, sebagai berikut:

- integer
- double
- string
- array
- object

Tipe variabel ini tidak perlu ditentukan oleh programmer, melainkan ditentukan pada saat runtime oleh PHP dengan kebergantungan pada konteks pemakaian variabel tersebut.

2.2.1 Menginisialisasi Variabel

Untuk menginisialisasi suatu variabel dalam PHP, secara sederhana berikan saja suatu nilai padanya. Tetapi untuk tipe seperti array dan objek membutuhkan mekanisme yang berbeda.

```
$nama = "Fahra Ragita";
$umur = 6;
$berat = 20.5;
```

2.2.2. Menginisialisasi Array

Array dapat diinisialisasi dengan dua cara: dengan pemberian nilai langsung, dan menggunakan konstruksi `array()`.

Untuk memberikan nilai secara langsung pada suatu array, secara sederhana adalah memberikan nilai ke variabel array dengan subscript kosong. Nilai tersebut akan ditambahkan sebagai elemen terakhir dari array.

```
$namasiswa[] = "Ilham"; // $namasiswa[0] = "Ilham"  
$namasiswa[] = "Melati"; // $namasiswa[1] = "Melati"  
Sesuatu yang perlu diingat adalah elemen array dimulai dari index 0 (bukan 1)
```

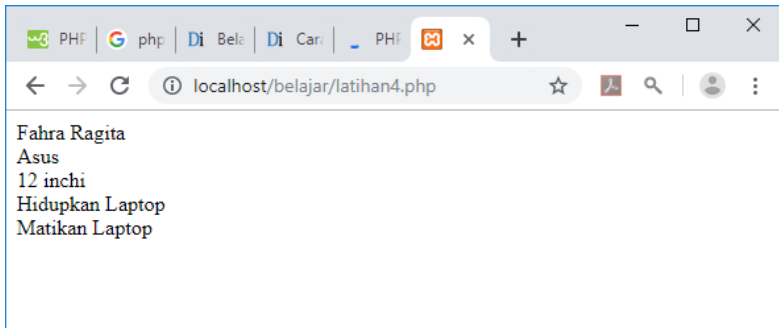
2.2.3. Menginisialisasi Objek

Untuk menginisialisasi suatu object, anda dapat menggunakan perintah `new`. Perintah ini digunakan untuk menginisialisasi suatu object kepada suatu variabel.

```
<?php  
// buat class laptop  
class laptop {  
  
    // buat property untuk class laptop  
    var $pemilik;  
    var $merk;  
    var $ukuran_layar;  
  
    // buat method untuk class laptop  
    function hidupkan_laptop() {  
        return "Hidupkan Laptop";  
    }  
    function matikan_laptop() {  
        return "Matikan Laptop";  
    }  
}  
  
// buat objek dari class laptop (instansiasi)  
$laptop_baru = new laptop();  
  
// set property  
$laptop_baru->pemilik="Fahra Ragita";
```

```
$laptop_baru->merk="Asus";  
$laptop_baru->ukuran_layar="12 inchi";  
  
// tampilkan property  
echo $laptop_baru->pemilik;  
echo "<br />";  
echo $laptop_baru->merk;  
echo "<br />";  
echo $laptop_baru->ukuran_layar;  
echo "<br />";  
  
// tampilkan method  
echo $laptop_baru->hidupkan_laptop();  
echo "<br />";  
echo $laptop_baru->matikan_laptop();  
?>
```

Kode Program: latihan4.php



Gambar 2.3. Output Latihan4

2.2.4. Jangkauan dari variabel

Jangkauan dari suatu variabel adalah tergantung pada dimana variabel tersebut didefinisikan. Pada umumnya semua variabel PHP hanya memiliki suatu jangkauan.

Setiap variabel yang digunakan dalam suatu fungsi, maka secara default jangkauannya adalah lokal. Sebagai contoh:

```
$awal = 1; /* jangkauan global */  
function Coba () {  
    echo $awal; /*mengacu pada variabel jangkauan lokal */  
}  
Coba ();
```

Script diatas tidak mengeluarkan apapun, karena \$awal dalam fungsi Coba mengacu pada \$awal lokal yang tidak memiliki nilai apapun. Hal ini berbeda dengan bahasa C dimana setiap variabel global akan berlaku bagi semua function, kecuali didefinisikan sebagai local. Dalam PHP, variabel global harus dideklarasikan sebagai global didalam suatu fungsi jika mereka akan digunakan dalam fungsi tersebut, contoh :

```
$panjang = 10;  
$lebar = 5;  
function Hitungluas () {  
    global $panjang, $lebar,$luas;  
    $luas = $panjang * $lebar;  
}  
Hitungluas ();  
echo $luas;
```

Script diatas akan menghasilkan keluaran "50". Dengan mendeklarasikan \$panjang dan \$lebar sebagai global didalam fungsi.

Cara kedua untuk mengakses variabel sebagai acuan global adalah dengan menggunakan definisi \$GLOBAL array pada PHP. Contoh sebelumnya dapat juga ditulis menjadi:

```
echo "<hr>Cara Kedua<br>";  
$panjang= 5;  
$lebar= 3;  
function Hitungluas2() {  
    $GLOBALS["luas"]= $GLOBALS["panjang"] * $GLOBALS["lebar"];  
}  
Hitungluas2();  
echo $luas;
```

\$GLOBALS array adalah suatu asosiasi array dengan key adalah nama dari variabel global.

Hal lain yang penting dari jangkauan adalah variabel static. Suatu variabel hanya ada pada jangkauan lokal fungsi, dan nilainya akan tetap dipertahankan:

Contoh berikut menggunakan variabel static.

```
<?php
function Coba () {
    static $a = 0;
    echo $a;
    $a++;
}
Coba();
?>
```

Sekarang, setiap kali fungsi Coba(), dipanggil maka nilai \$a adalah nilai \$a pada pemanggilan sebelumnya.

2.2.5. Variabel variabel

Kadang-kadang adalah lebih nyaman menggunakan variable-variabel; Yang mana adalah nama variabel yang dapat digunakan secara dinamis. Secara normalnya variabel dibuat dengan :

```
$a = "hello";
```

Suatu variabel variabel akan menggunakan nilai dari suatu variabel menjadi nama variabel, Contoh:

```
$a = "hello";
$$a = "world";
```

Pada contoh diatas akan terbentuk dua variabel yaitu \$a dan \$hello. Dimana \$a berisi "hello" dan \$hello berisi "world". Selanjutnya perintah berikut ini:

```
echo "$a ${$a}";
atau
echo "$a $hello"
```

Akan menghasilkan keluaran : hello world

2.2.6. Penentuan tipe variabel

PHP tidak membutuhkan deklarasi variabel secara eksplisit; tipe variabel ditentukan berdasarkan konteks pemakaiannya pada saat runtime. Dengan

kata lain; jika anda memberi nilai string kesuatu variabel `var`, `var` menjadi suatu variabel tipe string. Jika anda memberi nilai integer ke `var`, maka otomatis berubah menjadi tipe integer.

Suatu contoh dari otomatisasi konversi tipe pada PHP adalah operator penjumlahan '+'. Jika salah satu operandnya adalah tipe double, maka semua operand lainnya dievaluasi sebagai double dan hasilnya adalah double.

```
$coba = "0";           // $coba adalah string (ASCII 48)
$coba++;              // $coba adalah string "1" (ASCII 49)
$coba += 1;          // $coba sekarang adalah integer (2)
$coba = $coba + 1.3; // $coba sekarang adalah double (3.3)
$coba = 5 + "10 kotak"; // $coba adalah integer (15)
$coba = 5 + "10 kotak kecil"; // $coba adalah integer (15)
```

2.2.7. Mengetahui tipe suatu variabel

Karena PHP menentukan tipe variabel sesuai dengan konteks pemakaiannya, maka anda dapat menggunakan fungsi berikut untuk memeriksa tipe pada suatu variabel `gettype()`, `is long()`, `is double()`, `is string()`, `is array()`, dan `is object()`.

2.2.8. Tipe casting

Tipe casting dalam PHP bekerja seperti dalam C: nama dari tipe yang diinginkan ditulis didalam kurung sebelum variabel yang akan di cast.

```
$coba = 10;           // $coba adalah suatu integer
$bar = (double) $coba; // $bar adalah suatu double
```

2.2.9. Cast yang diperbolehkan:

- (int), (integer) - cast ke integer
- (real), (double), (float) - cast ke double
- (string) - cast ke string
- (array) - cast ke array
- (object) - cast ke object

2.2.10. Konversi String

Ketika suatu string dievaluasi sebagai suatu nilai numerik, nilai hasil dan tipe ditentukan sebagai berikut.

String akan dievaluasi sebagai suatu double jika mengandung salah satu karakter '.', 'e' atau 'E'. Jika tidak akan dievaluasi sebagai suatu string.

Jika string dimulai dengan data numerik yang sah, maka nilai tersebut akan digunakan, jika tidak akan memiliki nilai 0 (nol).

```
$coba = 1 + "10.5";           // $coba adalah double (11.5)
$coba = 1 + "-1.3e3";        // $coba adalah double (-1299)
$coba = 1 + "bob-1.3e3";     // $coba adalah integer (1)
$coba = 1 + "bob3";         // $coba adalah integer (1)
$coba = 1 + "10 Small Pigs"; // $coba adalah integer (11)
$coba = 1 + "10 Little Piggies"; // $coba adalah integer (11);
$coba = "10.0 pigs " + 1;    // $coba adalah int (11)
$coba = "10.0 pigs " + 1.0;  // $coba adalah double (11)
```

Tipe dari variabel tergantung pada ekspresi kedua, jika ekspresi pertamanya adalah string

2.2.11. Menangani variabel Form, Cookies dan Environment

Pada pemrograman CGI, program kita akan berinteraksi dengan variabel-variabel dari luar yang dikirim melalui form baik dengan metode GET maupun metode POST.

Ketika suatu form dikirim ke suatu PHP script, semua variabel dari form secara otomatis dapat diproses oleh script PHP sebagaimana variabel biasanya. Sebagai contoh, perhatikan form berikut ini:

Nama File:form1.html

```
<form action="proses.php" method="post">
  Nama: <input type="text" name="nama"><br>
  <input type="submit" value="Kirim">
</form>
```

Nama File:proses.php

```
<?php
  $nama = $_POST["nama"];
  echo $nama;
?>
```

Ketika form disubmit, maka PHP akan membuat variabel \$nama, yang mana mengandung apa yang diketikkan pada field Nama: di form tersebut.

PHP juga mendukung variabel array dalam kontek form, tetapi dibatasi hanya 1 dimensi, Contoh:

Nama File:form2.html

```
<form action="array.php" method="post">
  Nama: <input name="personal[nama]"><br>
  Email: <input name="personal[email]"><br>
  Jurusan: <br>
  <select name="jurusan[]">
    <option value="Informatika">Informatika
    <option value="Akuntansi">Akuntansi
    <option value="Elektro">Elektro
    <option value="Komputer">Komputer
    <option value="Ekonomi">Ekonomi
  </select>
  <input type="submit">
</form>
```

Nama File:array.php

```
<?php
  $nama = $_POST["personal"]["nama"];
  $email = $_POST["personal"]["email"];
  $jurusan = $_POST["jurusan"];
  echo "Nama: " . $nama . "<br>";
  echo "Email: " . $email . "<br>";
  echo "Jurusan yang dipilih:<br>";
  foreach ($jurusan as $selectedJurusan) {
    echo $selectedJurusan . "<br>";
  }
?>
```

2.2.11.1. Variabel pada IMAGE SUBMIT

Ketika mengirim suatu form, juga dimungkinkan untuk menggunakan suatu gambar sebagai pengganti tombol submit dengan tag HTML berikut ini:

```
<input type=image src="image.gif" name="sub">
```

Ketika pemakai melakukan klik pada gambar tersebut, makan form akan dikirim ke CGI dengan dua variabel tambahan, yaitu sub_x dan sub_y. Kedua variabel ini adalah koordinat dimana klik dilakukan pada gambar.

Form3.html

```
<html>  
<form action="aksi3.php" method="post">  
<input type=image src="button3.jpg" name="sub">
```

aksi3.php

```
<pre>  
<?php print_r($_REQUEST);  
$x=$_POST['sub_x'];  
$y=$_POST['sub_y'];  
echo $x;  
echo $y;  
?>  
</pre>
```

2.2.11.2. HTTP Cookies

PHP secara transparan mendukung HTTP cookies. Cookies adalah suatu mekanisme penyimpanan data secara remote pada browser klien. Hal ini dapat digunakan untuk mengidentifikasi pemakai pada kunjungan berikutnya.

Anda dapat menggunakan fungsi SetCookie(). Cookie adalah bagian dari HTTP header, jadi fungsi SetCookie harus dipanggil sebelum output lainnya dikirim ke browser. Hal ini adalah sama batasannya untuk fungsi Header(). Semua cookies yang dikirim pada anda dari klien akan secara otomatis di ubah kedalam suatu variabel PHP seperti method GET dan POST data.

Jika anda ingin memberikan banyak nilai pada suatu cookie tunggal, tambahkan saja [] pada nama cookie.

Sebagai contoh:

```
SetCookie ("MyCookie[]", "Testing", time()+3600);
```

Catatan bahwa suatu cookie akan menimpa cookie sebelumnya yang memiliki nama yang sama dalam browser anda, kecuali path atau domainnya berbeda. Jadi untuk suatu aplikasi shopping cart anda perlu menyimpan suatu counter dan mengirimnya bersamaan.

Contoh. SetCookie.php

```
<?php  
setcookie('username', 'administrator', time() + (60 * 60 * 24 * 5), '/');  
setcookie('nama', 'Fahra Ragita', time() + (60 * 60 * 24 * 15), '/');  
?>
```

GetCookie.php

```
<?php  
echo "Username " . $_COOKIE['username'];  
echo "<br/>";  
echo "Nama " . $_COOKIE['nama'];  
?>
```

2.2.11.3. Variabel Environment

Variabel lingkungan (environment variables) digunakan dalam PHP untuk mengakses informasi konfigurasi dan variabel lingkungan yang disetel pada server web atau sistem operasi.

Contoh umum penggunaan variabel lingkungan termasuk mengakses informasi seperti alamat IP pengguna, nama server, atau direktori tempat skrip PHP dijalankan.

Berikut adalah contoh penggunaan beberapa variabel lingkungan umum di PHP:

1. `$_SERVER['REMOTE_ADDR']`: Digunakan untuk mendapatkan alamat IP pengguna yang mengakses situs web Anda. Contoh:

```
$ip_address = $_SERVER['REMOTE_ADDR'];  
echo "IP Address pengguna: " . $ip_address;
```

2. `$_SERVER['HTTP_USER_AGENT']`: Digunakan untuk mendapatkan informasi tentang agen pengguna (browser) yang digunakan oleh pengguna. Contoh:

```
$user_agent = $_SERVER['HTTP_USER_AGENT'];  
echo "Agen Pengguna: " . $user_agent;
```

3. `$_SERVER['SERVER_NAME']`: Digunakan untuk mendapatkan nama server. Contoh:

```
$server_name = $_SERVER['SERVER_NAME'];  
echo "Nama Server: " . $server_name;
```

4. `$_SERVER['REQUEST_METHOD']`: Digunakan untuk mendapatkan metode permintaan HTTP yang digunakan oleh pengguna (GET, POST, PUT, dll.). Contoh:

```
$request_method = $_SERVER['REQUEST_METHOD'];  
echo "Metode Permintaan: " . $request_method;
```

5. `$_ENV['VARIABLE_NAME']`: Anda juga dapat mengakses variabel lingkungan kustom yang telah Anda setel sendiri. Misalnya, jika Anda memiliki variabel lingkungan bernama "MY_CUSTOM_VAR," Anda dapat mengaksesnya seperti ini:

```
$custom_var = $_ENV['MY_CUSTOM_VAR'];  
echo "Nilai variabel lingkungan kustom: " . $custom_var;
```

Pastikan bahwa Anda telah mengatur variabel lingkungan yang Anda akses pada server atau dalam lingkungan Anda sebelum mencoba mengaksesnya dalam skrip PHP.

Untuk mengatur atau mengkonfigurasi environment dapat menggunakan fungsi `putenv()`, dan menggunakan Fungsi `getenv()` untuk mengaksesnya.

```
<pre>  
<?php  
$ip_address = $_SERVER['REMOTE_ADDR'];  
echo "Alamat IP pengguna: " . $ip_address;  
echo "<br>";  
$user_agent = $_SERVER['HTTP_USER_AGENT'];  
echo "Agen Pengguna: " . $user_agent;  
echo "<br>";
```

```
$server_name = $_SERVER['SERVER_NAME'];  
echo "Nama Server: " . $server_name;  
echo "<br>";  
$request_method = $_SERVER['REQUEST_METHOD'];  
echo "Metode Permintaan: " . $request_method;  
echo "<br>";  
  
// Menetapkan nilai variabel lingkungan  
putenv("MY_VARIABLE=Belajar_PHP");  
$myVar = getenv("MY_VARIABLE");  
echo "Nilai variabel lingkungan MY_VARIABLE adalah: " . $myVar;  
echo "<br>";  
?>
```

2.3. Konstruksi Bahasa PHP

Suatu PHP script terdiri dari sejumlah perintah yang berurutan. Suatu perintah dapat berupa pemberi nilai, pemanggilan terhadap fungsi, perulangan, perintah kondisi dan baris kosong.

Setiap perintah diakhiri dengan sebuah titik koma (;). Dan beberapa perintah dapat dikelompokkan menjadi satu dalam kurung kurawal {dan}. Catatan : Setiap perintah diakhir dengan sebuah titik koma (;).

2.4. Konstanta

PHP memiliki sejumlah konstanta yang telah didefinisikan, dan anda diberi kesempatan untuk membuat konstanta sesuai dengan kebutuhan anda. Konstanta yang telah didefinisikan adalah `__FILE__` (nama file yang sedang diproses) `__LINE__` (nomor baris dari file yang sedang diproses)

Contoh :

```
<?php  
function report_error($file, $line, $message) {  
    echo "Suatu kesalahan terjadi pada file $file baris $line: $message."  
}  
report_error(__FILE__, __LINE__, "Telah terjadi suatu kesalahan!");  
?>
```

Anda dapat mendefinisikan konstanta ciptaan dengan menggunakan fungsi `define()`.

Contoh :

```
<?php  
define("CONSTANT", "Hello world.");  
echo CONSTANT; // akan mencetak "Hello world."  
?>
```

2.5. Ekspresi

Hampir semua yang anda tulis dalam PHP script adalah ekspresi. Definisi yang paling mudah dari ekspresi adalah "segala sesuatu yang memiliki nilai".

Contohnya yang paling sederhana dari ekspresi adalah konstanta dan variabel. Ketika anda mengetik "\$a = 5", anda telah memberikan nilai '5' ke \$a.

Contoh yang lebih kompleks untuk suatu ekspresi adalah fungsi. Untuk jelasnya perhatikan contoh berikut:

```
function hello () {  
    return 5;  
}
```

maka kalau anda menulis \$c = hello() adalah sama dengan memberikan nilai 5 ke variabel \$c, karena fungsi hello mengembalikan 5. Diatas adalah contoh fungsi yang sederhana.

PHP mendukung tiga tipe nilai skalar yaitu : nilai integer, floating point dan string. (nilai skalar adalah nilai yang tidak dapat dibagi menjadi bagian yang lebih kecil, seperti array). PHP juga mendukung dua tipe nilai bukan skalar yaitu array dan objek.

DAFTAR PUSTAKA

1. Suryana, Taryana (2023) [Materi 1. Pengenalan E-commerce Dan Persiapan Software Pendukungnya](#). [Teaching Resource]
2. Suryana, Taryana (2023) [Materi 11 Input Dan Label](#). [Teaching Resource]
3. Suryana, Taryana (2023) [Materi 10. Display Layout Dan Button](#). [Teaching Resource]

4. Suryana, Taryana (2023) [Materi 11. Input Dan Label Dalam W3 Framework](#). [Teaching Resource]
5. Suryana, Taryana (2022) [Materi 9. Framework Css](#). [Teaching Resource]
6. Suryana, Taryana (2022) [W3 Css Framework](#). [Teaching Resource]
7. Suryana, Taryana (2022) [Materi 8 Pengenalan Css](#). [Teaching Resource]
8. Suryana, Taryana (2022) [Frame Dalam Html](#). [Teaching Resource]
9. Suryana, Taryana (2022) [Materi 6 Membuat Tautan Atau Link Dalam Html](#). [Teaching Resource]
10. Suryana, Taryana (2022) [Materi 5 Form Dengan Html Dan Html5](#). [Teaching Resource]
11. Suryana, Taryana (2022) [Materi 4. Membuat Tabel Dalam Html](#). [Teaching Resource]
12. Suryana, Taryana (2022) [Materi 3 Membuat List / Daftar](#). [Teaching Resource]
13. Suryana, Taryana (2022) [Materi 2 Memformat Dokumen Html](#). [Teaching Resource]
14. Suryana, Taryana (2022) [Materi 1 Pengenalan Html](#). [Teaching Resource]
15. Suryana, Taryana (2021) [Bab 1. Pengenalan Html](#). [Teaching Resource]
16. Suryana, Taryana (2021) [Pelatihan Pembuatan Website Dengan Menggunakan Html Dan Dreamweaver](#). [Teaching Resource]
17. <https://www.w3schools.com/html>, September, 2021
18. Suryana, Taryana (2021) [Materi 1. Pengenalan Html Web Browser Dan Text Editor](#). [Teaching Resource]
19. Suryana, Taryana (2021) [Pengenalan Komputer Aplikasi It 1 Dan Pemrograman Web](#). [Teaching Resource]
20. Suryana, Taryana; Koesheryatin (2014), [Aplikasi Internet Menggunakan HTML, CSS & Java Script](#), Elexmedia Komputindo, Jakarta
21. Suryana, Taryana (2010) [Membuat Web Pribadi dan Bisnis dengan HTML](#), Gava Media, Yogyakarta
22. Suryana, Taryana; Koesheryatin (2014), [Aplikasi Internet Menggunakan HTML, CSS & Java Script](#), Elexmedia Komputindo, Jakarta
23. Suryana, Taryana (2021) [Bab 1. Pengenalan Html](#). [Teaching Resource]
24. Suryana, Taryana (2021) [Pelatihan Pembuatan Website Dengan Menggunakan Html Dan Dreamweaver](#). [Teaching Resource]
25. Suryana, Taryana (2010) [Membuat Web Pribadi dan Bisnis dengan HTML](#), Gava Media, Yogyakarta
26. <https://www.w3schools.com/html>, September, 2021
27. Suryana, Taryana (2021) [Materi 1. Pengenalan Html Web Browser Dan Text Editor](#). [Teaching Resource]
28. Suryana, Taryana (2021) [Pengenalan Komputer Aplikasi It 1 Dan Pemrograman Web](#). [Teaching Resource]

29. Suryana, Taryana (2021) [Memformat Dokumen Html](#). [Teaching Resource]
 30. Suryana, Taryana (2021) [Membuat List Atau Daftar Dengan Html Materi Jurusan Perancangan Wilayah Dan Kota](#). [Teaching Resource]
 31. Suryana, Taryana (2021) [Membuat List Atau Daftar Dengan Html Untuk Jurusan Teknik Sipil](#). [Teaching Resource]
 32. Suryana, Taryana (2021) [Membuat List Atau Daftar Dengan Html](#). [Teaching Resource]
 33. Suryana, Taryana (2021) [Membuat Tabel Dengan Html Kelas If-wa1-ul2-2021](#). [Teaching Resource]
 34. Suryana, Taryana (2021) [Membuat Tabel Dengan Html Kelas Ts-2021-1](#). [Teaching Resource]
 35. Suryana, Taryana (2021) [Membuat Tabel Dengan Html Kelas Pwk-2021-1](#). [Teaching Resource]
 36. Suryana, Taryana (2021) [Input Dan Output Data Barang Dan Data Member](#). [Teaching Resource]
 37. Suryana, Taryana (2021) [Membuat Aplikasi Pencatatan Perkawinan Disdukcapil](#). [Teaching Resource]
 38. Suryana, Taryana (2021) [Membuat Tabel Dengan Html Kelas If3-2021-1](#). [Teaching Resource]
 39. Suryana, Taryana (2021) [Membuat Tabel Dengan Html Kelas If-2-2021-1](#). [Teaching Resource]
 40. Suryana, Taryana (2021) [Membuat Tabel Dengan Html Kelas If-12021-1](#). [Teaching Resource]
 41. <https://www.w3schools.com/html>
-