

8.1. Pendahuluan

1. Apa yang ada dalam modul ini sebagian memerlukan tersedianya web server. Oleh karena itu, file-file HTML yang anda tulis perlu disimpan di direktori/folder sesuai konfigurasi web servernya masing-masing. Untuk melihat hasilnya, bisa langsung dibuka dengan browser, seperti Chrome, Edge, Safari, atau browser lainnya dengan memasukkan alamat URL dalam browser.
2. Gunakan hanya text editor seperti Notepad++ atau Sublime Text. Anda dapat gunakan text editor yang biasa anda gunakan. Jangan gunakan aplikasi pengolah kata seperti MS Word, MS Wordpad, dst.
3. Sebelum memulai latihan, anda harus menyiapkan lingkungan untuk dapat menyediakan layanan web, yaitu sebuah web server yaitu Apache di komputernya masing-masing. Setelah itu atau bersamaan dengan itu, tugas anda juga untuk menyiapkan PHP sebagai *extension* atau *plug-in* pada web server, serta MariaDB (MySQL) sebagai sistem manajemen basis datanya.
4. Ada beberapa cara untuk melakukan instalasi Apache, PHP dan MariaDB. Pilih salah satu dari metode berikut:
 - a. Apache, PHP dan MariaDB merupakan software yang masing-masing fungsinya sebagai web server, bahasa pemrograman, dan sistem manajemen basis data. Masing-masing merupakan software yang terpisah yang dibuat oleh developer yang berbeda. Cara yang paling *straightforward* dan pasti adalah melakukan instalasi masing-masing.

Untuk mendapatkan informasi cara melakukan instalasi untuk masing-masing software, dapat melihat website berikut. Referensi berikut adalah untuk sistem operasi Windows. Untuk sistem operasi yang lain agar menyesuaikan.

- Apache Lounge: Apache 2.4 VS16 Windows Binaries and Modules. Anda bisa pilih paket 64-bit atau 32-bit, sesuai dengan platform sistem operasi yang anda gunakan. URL: <https://www.apachelounge.com/download/>
 - PHP for Windows. Anda dapat pilih versi *Thread Safe* (TS). URL: <https://windows.php.net/download/>
 - Anda dapat juga merujuk ke tulisan ini, "How to Install PHP on Windows", walaupun yang dituliskan bukan versi terbaru dari PHP. <https://www.sitepoint.com/how-to-install-php-on-windows/>
 - MariaDB: Download. Pilih versi "*stable*" terbaru yang tersedia yang sesuai dengan platform sistem operasi anda. Anda dapat memilih paket MSI untuk instalasi yang lebih mudah. <https://downloads.mariadb.org>
- b. Cara yang kedua adalah dengan memanfaatkan bantuan suatu skrip/program yang melakukan instalasi ketiga software secara otomatis. Skrip/program ini akan mengunduh, menginstall, dan melakukan konfigurasi ketiga software tersebut menyesuaikan dengan lingkungan sistem operasi anda. Ada beberapa skrip/program

yang sudah banyak digunakan dan direkomendasikan seperti XAMPP, WAMP atau Web.Developer. Anda dapat juga menggunakan Portable Apache + PHP + MariaDB for Windows dari cubiclesoft <https://github.com/cubicsoft/portable-apache-maria-db-php-for-windows>. Berbeda dengan kompetitornya, paket installer ini tidak menyediakan GUI dan program manajemen; hanya program yang dijalankan dalam console/terminal. Namun prosesnya lebih *straightforward* dan informatif.

5. Setelah anda selesai melakukan instalasi dengan salah satu dari metode di atas, pastikan web server anda berfungsi dengan membukanya di web browser anda. Jika belum bisa berfungsi, coba baca dokumentasi yang disediakan oleh masing-masing software atau paket installer.
6. Mari kita lakukan uji coba dengan membuat "Hello World!" dengan PHP.
 - a. Buka editor teks dan tuliskan kode program PHP berikut. Tuliskan nama anda di tempat `<your name>`.

```
<!DOCTYPE html> <html lang='en-GB'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Our first PHP script</h1>
  <?php
$user = "<your name>";
print ("<p><b>Hello $user<br>\nHello World!</b></p>\n"); ?>
  </body>
</html>
```

- b. Simpan kode program tersebut dalam file **php08A.php**. Jangan lupa, simpan dengan ekstensi php, agar kode program PHP di dalamnya bisa berfungsi sebagai program PHP.
 - c. Coba buka dengan browser dan akses dengan URL.

<http://localhost/php08A.php>

Bagian yang berwarna biru (berikut pathnya) sesuaikan dengan lingkungan atau konfigurasi web server anda. Pastikan hasil yang ditampilkan sesuai dengan yang diharapkan.

Selalu cantumkan informasi tentang siapa membuat/menulis skrip/halaman web, siapa yang memiliki hak cipta, serta kapan skrip/halaman web tersebut dibuat/ditulis merupakan praktis yang baik. Tambahkan tiga deklarasi meta ke dalam skrip untuk menunjukkan informasi tersebut.

8.2. Kegiatan Praktikum

1. Latihan berikut tentang tipe, *type casting* (konversi tipe) dan identifikasi tipe dalam PHP.
 - a. Buka teks editor dan tuliskan markup HTML dan kode program PHP berikut. Simpan dalam file **php08B.php**.

```
<!DOCTYPE html>
```

```

<html lang='en-GB'>
<head>
  <title>PHP 08B</title>
</head>
<body>
  <h1>Hello World</h1>
  <?php
    error_reporting( E_ALL );
    ini_set('display_errors', 1);
    ini_set('display_startup_errors', 1);
    echo "<h2>Types and Type Casting</h2>\n";
    $mode = rand(1,4);
    if ($mode == 1)
      $randvar = rand();
    elseif ($mode == 2)
      $randvar = (string) rand();
    elseif ($mode == 3)
      $randvar = rand()/(rand()+1);
    else
      $randvar = (bool) $mode;
    echo "Random scalar: $randvar<br>\n"; ?>
  </body>
</html>

```

- b. Muat ke dalam browser, dan coba muat ulang (*reload/refresh*) beberapa kali. Amati bagaimana nilai dari variabel `$randvar` berubah setiap kali dimuat ulang.
- c. Kembangkan kode program sehingga menampilkan:
 - i. This is an integer ... Jika `$randvar` berisi bilangan bulat
 - ii. This is a floating-point number ... Jika `$randvar` berisi bilangan pecahan
 - iii. This is a string ... Jika `$randvar` berisi *string*
 - iv. I don't know what this is ... Untuk kasus lainnya

setelah baris `Random scalar`:

Hints: Anda dapat gunakan `gettype()`, `is_int()`, `is_string()`, dst.

2. Latihan berikut tentang *array* dan *operator* dalam *array* dalam PHP.
 - a. PHP memiliki operator yang dapat memanfaatkan array sebagai *stack* dan *queue*. Mari kita lihat cara kerjanya. Buka file baru dalam teks editor dan masukkan markup HTML dan kode PHP ke dalamnya.

```

<!DOCTYPE html>
<html>
<head>
  <title>PHP 08C</title>
</head>
<body>
  <h1>Array Operators and Regular Expressions</h1>
  <?php
    echo "<h2>Exercise 2A</h2>\n";
    $planets = array("earth");

```

```

array_unshift($planets,"mercury","venus");
array_push($planets,"mars","jupiter","saturn");
echo "(1) \$planets = [",join(" ", $planets),"]<br>\n";
$last = array_pop($planets);
echo "(2) \$planets = [",join(" ", $planets),"]<br>\n";
$first = array_shift($planets);
echo "(3) \$planets = [",join(" ", $planets),"]<br>\n";
echo "(4) \$first = $first, \$last = $last<br>\n";
?>
</body>
</html>

```

Simpan dalam file **php08C.php** di dalam folder public web server. Pastikan bisa diakses menggunakan web browser.

- b. Buka file php08C.php menggunakan web browser. Pastikan hasilnya berupa pesan yang menunjukkan bagaimana berbagai operator melakukan manipulasi terhadap array. Pastikan anda memahami cara kerja masing-masing operator.
- c. Cara kerja pemberian nilai yang melibatkan array dalam PHP berbeda dengan JavaScript. Untuk memastikan hal tersebut, tambahkan kode PHP berikut di akhir file php08C.php.

```

echo "<h2>Exercise 2B</h2>\n";
$spheres = $planets;
echo "(5) \$spheres = [",join(" ", $spheres),"]<br>\n";
$planets[1] = "midgard";
echo "(6) \$planets = [",join(" ", $planets),"]<br>\n";
echo "(6) \$spheres = [",join(" ", $spheres),"]<br>\n";
$spheres = &$planets;
echo "(7) \$spheres = [",join(" ", $spheres),"]<br>\n";
$planets[0] = "friga";
echo "(8) \$planets = [",join(" ", $planets),"]<br>\n";
echo "(8) \$spheres = [",join(" ", $spheres),"]<br>\n";

```

Simpan file dan muat ulang ke dalam browser. Pastikan anda memahami hasil keluarannya berkaitan dengan pemberian nilai pada array dalam PHP.

- d. Kembangkan skrip php08C.php dengan menambahkan kode program yang fungsinya menghapus elemen array \$planets dari depan sampai kosong, dan pada setiap langkah, tampilkan isi elemen yang dihapus. Keluarannya seperti ini.

```

Removed: friga Remaining: [midgard, mars, jupiter]
Removed: midgard Remaining: [mars, jupiter]
Removed: mars Remaining: [jupiter]
Removed: jupiter Remaining: []

```

3. Latihan berikut tentang manipulasi array, operasi string, dan pencarian dan penggantian string dengan menggunakan pola.
 - a. Tambahkan kode PHP berikut ke akhir skrip php08C.php.

```

echo "<h2>Exercise 2C</h2>\n";
$names = ["Dave Shield", "Mr Andy Roxburgh",

```

```

        "Dr George Christodoulou", "Dr Ullrich Hustadt",
        "Dr David Jackson"];
foreach ($names as $name)
    echo "(1) Name: $name<br>\n";
// Your own code here
foreach ($names as $name)
    echo "(2) Name: $name<br>\n";

```

Simpan file dan muat ulang ke dalam browser. Keluaran yang dihasilkan seperti ini.

```

(1) Name: Dave Shield
(1) Name: Mr Andy Roxburgh
(1) Name: Dr George Christodoulou
(1) Name: Dr Ullrich Hustadt
(1) Name: Dr David Jackson

```

dan seterusnya, dengan (1) digantikan dengan (2).

- b. Tambahkan kode PHP di tempat (komentar) yang ditunjukkan dalam latihan 3a yang mengubah isi string yang tersimpan di variabel `$names` sehingga keluarannya menjadi seperti ini.

```

(2) Name: SHIELD, Dave
(2) Name: ROXBURGH, Andy
(2) Name: CHRISTODOULOU, George
(2) Name: HUSTADT, Ullrich
(2) Name: JACKSON, David

```

Petunjuk: Gunakan *foreach-loop* untuk mengubah setiap elemen dalam suatu array.

4. Dalam PHP, tidak ada perbedaan antara array (*number-indexed array*) dengan *hash-indexed array* atau *dictionary*. Dalam PHP hanya ada *associative array* atau sama dengan *hash-indexed array*.

- a. Buat file baru, **php08D.php** dengan kode PHP berikut.

```

<!DOCTYPE html>
<html>
    <head>
        <title>PHP 08D</title>
    </head>
    <body>
        <h1>Associative Arrays</h1>
    <?php
        $dict1 = array('a' => 1, 'b' => 2);
        $dict2 = $dict1;
        $dict1['b'] = 4;
        echo "\$dict1['b'] = ", $dict1['b'], "<br>\n";
        echo "\$dict2['b'] = ", $dict2['b'], "<br>\n";
    ?>
    </body>
</html>

```

Buka dengan browser dan amati bagaimana hasilnya. Pastikan anda paham dengan cara kerja setiap operasi.

- b. Tambahkan kode PHP yang menampilkan keluaran pasangan key-value dalam variabel array `$dict1`.
- c. Salah satu pemanfaatan *dictionary* atau *hashed* array adalah menghitung jumlah kemunculan suatu string dalam array. Misalnya, kita bisa menghitung berapa kemunculan string yang memuat huruf 'a' atau 'b' dalam array `$dict1`. Kita akan membangun kode PHP untuk melakukan penghitungan itu.

Tambahkan kode berikut di akhir skrip `php08D.php`.

```
$text = 'lorem ipsum elit congue auctor inceptos taciti suscipit
tortor auctor integer congue hac nullam hac auctor tellus nullam
inceptos nullam integer tellus nullam auctor elit lorem ipsum elit';
```

Tambahkan kode PHP menggunakan *associative* array `$dict3` untuk menghitung jumlah kemunculan suatu kata dalam string. Jika kode program benar, keluaran akan menjadi seperti ini.

```
lorem -> 2
ipsum -> 2
elit -> 3
congue -> 2
auctor -> 4
inceptos -> 2
taciti -> 1
suscipit -> 1
tortor -> 1
integer -> 2
hac -> 2
nullam -> 4
tellus -> 2
```

- d. Tambahkan kode PHP agar dapat menampilkan keluaran pasangan key-value `$dict3` secara berurutan mulai dari yang jumlah kemunculan terbesar. Keluaran dalam bentuk tabel seperti berikut.

Kata	Jumlah kemunculan
nullam	4
auctor	4
elit	3
tellus	2
ipsum	2
suscipit	1
tortor	1

Petunjuk:

Lihat

manual

PHP

berikut:

<http://php.net/manual/en/array.sorting.php>.

5. Pada latihan ini, kita akan melihat argumen/parameter dinamis dalam fungsi.
- a. Buat file baru dengan nama **php08E.php** dan masukkan markup HTML dan kode PHP berikut:

```
<!DOCTYPE html>
<html lang='en-GB'>
<head>
  <title>PHP 08E</title>
  <META name="description" content="php08E.php">
</head>
<body>
  <h1>Variable-length Argument Lists</h1>
<?php
try {
  echo "1: ", reduceOp(2,3), "<br>\n"; # throws an exception
}
catch (Exception $e) {
  echo "1: Exception ", $e->getMessage(), "<br>\n"; # 'TypeError'
}
try {
  echo "2: ", reduceOp(2,3,array('op' => '/')), # throws an
exception
  "<br>\n";
}
catch (Exception $e) {
  echo "2: Exception ", $e->getMessage(), "<br>\n"; # 'ValueError'
}
echo "3: ", reduceOp(array('op' => '+')), # should return NULL
  "<br>\n";
echo "4: ", reduceOp(2,array('op' => '*')), # should return 2
  "<br>\n";
echo "5: ", reduceOp(2,3,5,array('op' => '+')), # should return 10
  "<br>\n";
echo "6: ", reduceOp(2,3,5,7,array('op' => '*')), # should return
210
  "<br>\n";
echo "7: ", reduceOp(2,3,5,7,11,array('op' => '-')), # should return -
24
  "<br>\n";
?>
</body>
</html>
```

- b. Sebelum dibuka dengan browser, tambahkan deklarasi dan definisi fungsi `reduceOp` yang memiliki fungsi berikut:
- Fungsi `reduceOp` menerima *associative* array sebagai argumen terakhir yang memetakan *key* 'op' ke salah satu dari operator '+', '-', atau '*'. Jika argumen ini tidak ada, *exception* 'TypeError' akan muncul. Jika argumen ini ada, tapi tidak memiliki nilai/elemen yang valid, maka *exception* 'ValueError' akan muncul.
 - Sebagai tambahan di samping argumen di atas, fungsi `reduceOp` juga menerima sejumlah bilangan sebagai argumennya. Jika argumen *associative* array memiliki

nilai valid tapi tidak ada bilangan yang dikirimkan, maka fungsi akan mengembalikan NULL. Jika argumen *associative* array memiliki nilai valid dan juga ada bilangan yang dikirimkan, maka fungsi akan mengembalikan hasil operasi '*op*' yang ditentukan. Misalnya, *associative* array adalah ('*op*' => '-') dan bilangan yang dikirimkan adalah 2, 3, dan 5, maka fungsi akan mengembalikan ((2 - 3) - 5) yaitu -6.

- Pastikan fungsi `reduceOp` yang anda buat memberikan hasil atau memiliki perilaku sesuai harapan.

8.3. Responsi

1. Screenshot *source code* hasil pekerjaan praktikum nomor 1, salin pada file docx.
2. Screenshot output pada browser untuk pekerjaan praktikum nomor 1, salin pada file docx.
3. Ulangi langkah 1 dan 2 untuk kelima kegiatan praktikum.
4. Simpan ulang file docx ke dalam format pdf. Beri nama dengan format <<*nim*>>_modul8.pdf, contoh: 192191234_modul8.pdf, lalu kumpulkan sebagai responsi melalui Google Classroom.