

Nama : La Ode Muhammad Gazali  
NIM : 222212696  
Kelas : 2KS2

## MODUL 8 PEMROGRAMAN BERBASIS WEB

### (Konsep Dasar PHP)

#### Penugasan

1. Sebutkan minimal 5 perbedaan konsep dan atau penulisan script pada Javascript dan PHP, beserta ilustrasi contohnya.
2. Apa yang dimaksud dengan Heredocs di PHP?
3. Jelaskan Type Casting dan Type Juggling di PHP!
4. Jelaskan dengan contoh, penggunaan dari gettype(), istype(), dan var\_dump()!
5. Kerjakan Modul 8

#### Penyelesaian

1. Berikut 5 perbedaan antara javascript dan php dari segi penulisan script:

- 1) Pemanggilan fungsi eksternal

JavaScript: Fungsi JavaScript eksternal dapat dimasukkan ke dalam file terpisah dan dipanggil menggunakan tag `<script>` di dalam file HTML. Contoh:

```
<script src="script.js"></script>
```

PHP: Fungsi PHP eksternal biasanya disimpan dalam file terpisah dan disertakan menggunakan pernyataan include atau require di dalam skrip PHP. Contoh:

```
<?php include 'fungsi.php'; ?>
```

- 2) Penulisan variable

JavaScript: menggunakan kata kunci var, let, atau const untuk mendeklarasikan variabel. Contoh

```
<script>
    var nama = "Gazali";
    let umur = 21;
    const PI = 3.14;
</script>
```

PHP: menggunakan tanda dollar (\$) untuk mendeklarasikan variable dan keyword define untuk mendefinisikan suatu konstanta. Contoh

```
<?php
    $nama = "Gazali";
    $umur = 21;
    define("PI", 3.14);
?>
```

### 3) Penanganan string

JavaScript: Menggunakan tanda kutip tunggal (') atau ganda (") untuk mendefinisikan string, dan bisa melakukan interpolasi variabel dalam string menggunakan tanda kutip backtick (` `). Contoh:

```
<script>
    var nama = "Gazali";
    console.log(`Halo, nama saya ${nama}`);
</script>
```

PHP: Juga menggunakan tanda kutip tunggal (') atau ganda (") untuk mendefinisikan string, namun interpolasi variabel hanya bisa dilakukan dengan tanda kutip ganda. Contoh:

```
<?php
    $nama = "Gazali";
    echo "Halo, nama saya $nama";
?>
```

### 4) Penanganan error dan exception

JavaScript: Menggunakan try, catch, dan finally untuk menangani error atau eksepsi. Contoh:

```
<script>
    try {
        // Kode yang mungkin memunculkan error
    } catch (error) {
        // Tangkap dan tangani error
    } finally {
        // Eksekusi kode akhir (opsional)
    }
</script>
```

PHP: Juga menggunakan try, catch, dan finally untuk menangani error atau eksepsi, namun objek errornya berbeda. Contoh:

```
<?php
    try {
```

```

        // Kode yang mungkin memunculkan error
    } catch (Exception $e) {
        // Tangkap dan tangani error
    } finally {
        // Eksekusi kode akhir (opsional)
    }
?>

```

## 5) Lingkup variable dan penanganannya

JavaScript: Memiliki konsep variabel lokal dan global, di mana variabel lokal dideklarasikan di dalam fungsi dan global di luar fungsi. Contoh:

```

<script>
    var globalVar = "Variabel global";

    function exampleFunction() {
        var localVar = "Variabel lokal";
    }
</script>

```

PHP: Variabel secara default adalah global, tapi dapat didefinisikan sebagai lokal dengan menggunakan kata kunci global atau dideklarasikan dalam fungsi. Contoh:

```

<?php
    $globalVar = "Variabel global";

    function exampleFunction() {
        global $globalVar;
        $localVar = "Variabel lokal";
    }
?>

```

2. Heredoc adalah salah satu fitur dalam PHP yang memungkinkan untuk membuat string multibaris tanpa perlu menggunakan tanda kutip ganda atau tanda kutip satu. Heredoc memungkinkan untuk menulis string yang lebih panjang dan kompleks dengan lebih mudah, terutama jika string tersebut mengandung banyak karakter khusus atau baris baru.

Format umum Heredoc adalah dengan menggunakan kata kunci/penanda <<< **EOT (End of Text)** atau penanda lain misalnya <<<**HTML** atau <<<**JSON** selama penanda tersebut tidak muncul dalam string itu sendiri. Contoh:

```

<?php
    $paragraph = <<<EOT

```

```

        Ini adalah contoh Heredoc.
        Ini adalah string multibaris.
        Ini sangat berguna ketika Anda ingin membuat string yang panjang dan
        kompleks.
        EOT;

        echo $paragraph;
?>

```

### 3. Type casting

Type casting adalah proses secara eksplisit mengubah tipe data variabel dari satu tipe ke tipe lain. Dalam type casting, pengguna secara langsung menentukan konversi tipe data yang diinginkan. PHP menyediakan beberapa fungsi dan cara untuk melakukan type casting, seperti (int), (float), (string), (array), dll. Contoh

```

<?php
    $angka = "10"; // string
    $angka_integer = (int) $angka; // casting string ke integer
    echo $angka_integer; // output: 10
?>

```

### Type juggling

Type juggling adalah proses di mana PHP secara otomatis mengonversi tipe data variabel secara implisit saat operasi dilakukan. Dalam type juggling, PHP akan mencoba untuk menyesuaikan tipe data variabel sesuai dengan operasi yang sedang dilakukan. PHP akan mengubah tipe data secara dinamis sesuai kebutuhan untuk menyelesaikan operasi. Contoh:

```

<?php
    $angka = "10number"; // string
    $hasil = $angka + 3; // otomatis mengubah string "10" menjadi
    integer
    echo $hasil; // output: 13
?>

```

4. Fungsi **gettype()**, **is\_type()**, dan **var\_dump()** adalah fungsi yang berguna dalam PHP untuk memeriksa dan menampilkan informasi tentang tipe data dari variable.

### gettype()

Digunakan untuk mengembalikan tipe data dari sebuah variabel. Contoh:

```

<?php
    $angka = 10;

```

```

    $tipe = gettype($angka);
    echo "Tipe variabel \ $angka adalah: $tipe"; // Output: Tipe variabel
$angka adalah: integer
?>

```

### is\_type()

Digunakan untuk memeriksa apakah sebuah variabel adalah dari tipe tertentu. **is\_type()** mengembalikan true jika variabel tersebut adalah tipe yang diinginkan, dan false jika tidak. Contoh:

```

<?php
    $angka = 10;
    if (is_int($angka)) {
        echo "Variabel \ $angka adalah integer";
    } else {
        echo "Variabel \ $angka bukan integer";
    }
?>

```

### var\_dump()

Digunakan untuk menampilkan informasi lengkap tentang satu atau lebih variabel, termasuk tipe data, nilai, dan panjang string (jika berlaku). **var\_dump()** sangat berguna untuk debugging dan pemahaman lebih lanjut tentang struktur dan isi dari variabel. Contoh:

```

<?php
    $nama = "John";
    $umur = 25;
    var_dump($nama, $umur);

    /*
    output:
    string(4) "John"
    int(25)
    */
?>

```

## 5. Penugasan Modul 8

### Php08A.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Hello World</title>

```

```

</head>
<body>
  <h1>Our first PHP script</h1>
  <?php
    $user = "La Ode Muhammad Gazali";
    print("<p><b>Hello $user<br>\nHello World!</b></p>\n");
  ?>
</body>
</html>

```

**Output:**

# Our first PHP script

**Hello La Ode Muhammad Gazali**  
**Hello World!**

- 1) Latihan berikut tentang tipe, type casting (konversi tipe) dan identifikasi tipe dalam PHP.

**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";
  ?>

```

```

        // Menambahkan pengecekan tipe data
        if (is_int($randvar))
            echo "This is an integer.";
        elseif (is_float($randvar))
            echo "This is a floating-point number.";
        elseif (is_string($randvar))
            echo "This is a string.";
        else
            echo "I don't know what this is.";

    ?>
</body>
</html>

```

**Output:**

# Hello World

## Types and Type Casting

The mode is: 2

Random scalar: 192160630

This is a string.

- 2) Latihan berikut tentang array dan operator dalam array dalam PHP.

### Php08C.php

```

<!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";

echo "<h2>Exercise 2c</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";

while (!empty($planets)) {
    $removed = array_shift($planets);
    echo "Removed: $removed Remaining: [", join(", ", $planets),
"]<br>\n";
}

?>
</body>
</html>

```

**Output:**

## Array Operators and Regular Expressions

### Exercise 2a

- (1) \$planets = [mercury, venus, earth, mars, jupiter, saturn]
- (2) \$planets = [mercury, venus, earth, mars, jupiter]
- (3) \$planets = [venus, earth, mars, jupiter]
- (4) \$first = mercury, \$last = saturn

### Exercise 2c

- (5) \$spheres = [venus, earth, mars, jupiter]
- (6) \$planets = [venus, midgard, mars, jupiter]
- (6) \$spheres = [venus, earth, mars, jupiter]
- (7) \$spheres = [venus, midgard, mars, jupiter]
- (8) \$planets = [friga, midgard, mars, jupiter]
- (8) \$spheres = [friga, midgard, mars, jupiter]
- 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.

**Php08C.php menambahkan skrip Exercise 3 (Sebelum modifikasi)**

```
<!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";

        echo "<h2>Exercise 2c</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";

        while (!empty($planets)) {
            $removed = array_shift($planets);
            echo "Removed: $removed Remaining: [" . join(" ", $planets),
"]<br>\n";
        }

        echo "<h2>Exercise 3</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";

    ?>
</body>
</html>

```

**Output:**

## Array Operators and Regular Expressions

### Exercise 2a

(1) \$planets = [mercury, venus, earth, mars, jupiter, saturn]  
 (2) \$planets = [mercury, venus, earth, mars, jupiter]  
 (3) \$planets = [venus, earth, mars, jupiter]  
 (4) \$first = mercury, \$last = saturn

### Exercise 2c

(5) \$spheres = [venus, earth, mars, jupiter]  
 (6) \$planets = [venus, midgard, mars, jupiter]  
 (6) \$spheres = [venus, earth, mars, jupiter]  
 (7) \$spheres = [venus, midgard, mars, jupiter]  
 (8) \$planets = [friga, midgard, mars, jupiter]  
 (8) \$spheres = [friga, midgard, mars, jupiter]  
 Removed: friga Remaining: [midgard, mars, jupiter]  
 Removed: midgard Remaining: [mars, jupiter]  
 Removed: mars Remaining: [jupiter]  
 Removed: jupiter Remaining: []

### Exercise 3

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

**Php08C.php menambahkan skrip Exercise 3 (Setelah modifikasi)**

```

<!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";

echo "<h2>Exercise 2</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";

while (!empty($planets)) {
    $removed = array_shift($planets);
    echo "Removed: $removed Remaining: [", join(" ", $planets),
"]<br>\n";
}

echo "<h2>Exercise 3</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) {
    // memisahkan nama menjadi beberapa bagian
    $parts = explode(" ", $name);

    // menyusun kembali bagian nama sesuai format yang diinginkan
    $rearranged_name = strtoupper($parts[count($parts) - 1]) . ", "
. $parts[0];
    $name = $rearranged_name;
}

// Output the modified names
foreach ($names as $name) {
    echo "(2) Name: $name<br>\n";
}

```

```
?>
</body>
</html>
```

**Output:**

## Array Operators and Regular Expressions

### Exercise 2a

- (1) \$planets = [mercury, venus, earth, mars, jupiter, saturn]
- (2) \$planets = [mercury, venus, earth, mars, jupiter]
- (3) \$planets = [venus, earth, mars, jupiter]
- (4) \$first = mercury, \$last = saturn

### Exercise 2c

- (5) \$spheres = [venus, earth, mars, jupiter]
- (6) \$planets = [venus, midgard, mars, jupiter]
- (6) \$spheres = [venus, earth, mars, jupiter]
- (7) \$spheres = [venus, midgard, mars, jupiter]
- (8) \$planets = [friga, midgard, mars, jupiter]
- (8) \$spheres = [friga, midgard, mars, jupiter]
- Removed: friga Remaining: [midgard, mars, jupiter]
- Removed: midgard Remaining: [mars, jupiter]
- Removed: mars Remaining: [jupiter]
- Removed: jupiter Remaining: []

### Exercise 3

- (1) Name: Dave Shield
- (1) Name: Mr Andy Roxburgh
- (1) Name: Dr George Christodoulou
- (1) Name: Dr Ullrich Hustadt
- (1) Name: Dr David Jackson
- (2) Name: SHIELD, Dave
- (2) Name: ROXBURGH, Mr
- (2) Name: CHRISTODOULOU, Dr
- (2) Name: HUSTADT, Dr
- (2) Name: HUSTADT, Dr

- 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.

### Php08D.php

```
<!DOCTYPE html>
<html>
<head>
  <title>PHP 08D</title>
  <style>
    table {
      border-collapse : collapse;
    }
    th, td {
      padding : 5px;
      border : 1px solid black;
      text-align : center;
    }
  </style>

```

```

    </style>
</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";

        // Bagian b: Menampilkan pasangan key-value dalam $dict1
        echo "<h2>Menampilkan pasangan key-value dalam dict1 </h2>\n";
        foreach ($dict1 as $key => $value) {
            echo "key = $key -> value = $value<br>\n";
        }

        // Bagian c: Menghitung jumlah kemunculan suatu kata dalam string
        echo "<h2>Menghitung jumlah kemunculan suatu kata dalam
string</h2>\n";
        $text = 'lorem ipsum elit congue auctor inceptos taciti suscipit
tortor auctor integer congue hac nullam hac auctor tellus nullam
inceptos nullam integer tellus nullam auctor elit lorem ipsum elit';
        echo "$text<br><br>";

        // Memisahkan kata-kata dalam teks menjadi array
        $words = array_unique(explode(" ", $text));

        // Membuat associative array $dict3 untuk menghitung kemunculan kata
        $dict3 = array();
        foreach ($words as $word) {
            if (!empty($word)) { // Pastikan kata tidak kosong
                $dict3[$word] = substr_count($text, $word);
                echo "$word -> ", $dict3[$word], "<br>\n";
            }
        }

        echo "<br> Tabel urutan jumlah kemunculan kata dari yang terbanyak
<br>";
        arsort($dict3);
        echo "<table>\n";
        echo "<tr><th> Kata </th><th> Jumlah Kemunculan </th></tr>\n";

        foreach($dict3 as $kata => $jumlah)
            echo "<tr><td> $kata </td><td> ", $jumlah, "</td></tr>\n";
    </?php>

```

```

        echo "</table>\n";

    ?>
</body>
</html>

```

## Output:

### Associative Arrays

```

$dict1['b'] = 4
$dict2['b'] = 2

```

#### Menampilkan pasangan key-value dalam dict1

```

key = a -> value = 1
key = b -> value = 4

```

#### Menghitung jumlah kemunculan suatu kata dalam string

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

```

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
nullam -> 1

```

Tabel urutan jumlah kemunculan kata dari yang terbanyak

Kata	Jumlah Kemunculan
auctor	4
nullam	4
elit	3
lorem	2
ipsum	2
congue	2
inceptos	2
integer	2
hac	2
tellus	2
taciti	1
suscipit	1
tortor	1
nullam	1

5) Pada latihan ini, kita akan melihat argumen/parameter dinamis dalam fungsi.

```

<!DOCTYPE html>
<html>
<head>
    <title>PHP 08D</title>
    <style>
        table {
            border-collapse : collapse;
        }
        th, td {
            padding : 5px;

```

```

        border : 1px solid black;
        text-align : center;
    }
</style>
</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";

        // Bagian b: Menampilkan pasangan key-value dalam $dict1
        echo "<h2>Menampilkan pasangan key-value dalam dict1 </h2>\n";
        foreach ($dict1 as $key => $value) {
            echo "key = $key -> value = $value<br>\n";
        }

        // Bagian c: Menghitung jumlah kemunculan suatu kata dalam string
        echo "<h2>Menghitung jumlah kemunculan suatu kata dalam
string</h2>\n";
        $text = 'lorem ipsum elit congue auctor inceptos taciti suscipit
tortor auctor integer congue hac nullam hac auctor tellus nullam
inceptos nullam integer tellus nullam auctor elit lorem ipsum elit';
        echo "$text<br><br>";

        // Memisahkan kata-kata dalam teks menjadi array
        $words = array_unique(explode(" ", $text));

        // Membuat associative array $dict3 untuk menghitung kemunculan kata
        $dict3 = array();
        foreach ($words as $word) {
            if (!empty($word)) { // Pastikan kata tidak kosong
                $dict3[$word] = substr_count($text, $word);
                echo "$word -> ", $dict3[$word], "<br>\n";
            }
        }

        echo "<br> Tabel urutan jumlah kemunculan kata dari yang terbanyak
<br>";
        arsort($dict3);
        echo "<table>\n";
        echo "<tr><th> Kata </th><th> Jumlah Kemunculan </th></tr>\n";

```

```
foreach($dict3 as $kata => $jumlah)
echo "<tr><td> $kata </td><td>", $jumlah,"</td></tr>\n";

echo "</table>\n";

?>
</body>
</html>
```

**Output:**

## Variable-length Argument Lists

```
1: 1: Exception TypeError
2: 2: Exception ValueError
3:
4: 2
5: 10
6: 210
7: 11
```