

LAPORAN PRAKTIKUM PEMROGRAMAN WEB
PERTEMUAN 10
STATEMENT KONTROL



2411102441052

Angga Maulana Saputra

FAKULTAS SAINS DAN TEKNOLOGI
PROGRAM STUDI S1 TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

Latihan 10.1

if1.php

```
1 <?php
2 $bil = 10;
3 if ($bil > 5) {
4     echo "Bilangan lebih besar dari 5";
5 }
6
```

if2.php

```
1 <?php
2 $bil = 10;
3 if ($bil == 5) {
4     echo "Bilangan sama dengan 5";
5 }
6
```

Keduanya sama-sama menginisialisasi variable \$bil1 dengan nilai 10. Pada if1.php, kondisi yang diperiksa adalah (\$bil > 5). Karena 10 memang lebih besar dari 5, kondisi ini bernilai True (Benar) dan skrip berhasil mencetak output "Bilangan lebih besar dari 5". Sebaliknya, if2.php memeriksa kondisi (\$bil == 5). Karena 10 tidak sama dengan 5, kondisi ini bernilai False (Salah).

Latihan 10.2

logika1.php

```
1 <?php
2 $hobi = "makan";
3 if (($hobi == "makan") || ($hobi == "ngmeil")) {
4     echo "Hati-hati, jangan terlalu banyak. <br />";
5     echo "Jangan lupa olahraga ..";
6 }
7
```

Variabel \$hobi diinisialisasi dengan nilai "makan". Skrip kemudian mengevaluasi pernyataan if yang menggunakan operator logika OR (||), yang memeriksa apakah nilai \$hobi adalah "makan" atau "ngmeil". Karena perbandingan pertama (\$hobi == "makan") bernilai True (Benar), keseluruhan kondisi if dianggap terpenuhi. Kode di dalam blok if dieksekusi, dan skrip tersebut akan menghasilkan output berupa dua baris teks: "Hati-hati, jangan terlalu banyak.
Jangan lupa olahraga .."

logika2.php

```
● ● ●  
1 <?php  
2 $bil = 10;  
3 if ($bil == 10) {  
4     echo "Bilangan sama dengan 10";  
5 } else {  
6     echo "Bilangan tidak sama dengan 10";  
7 }  
8
```

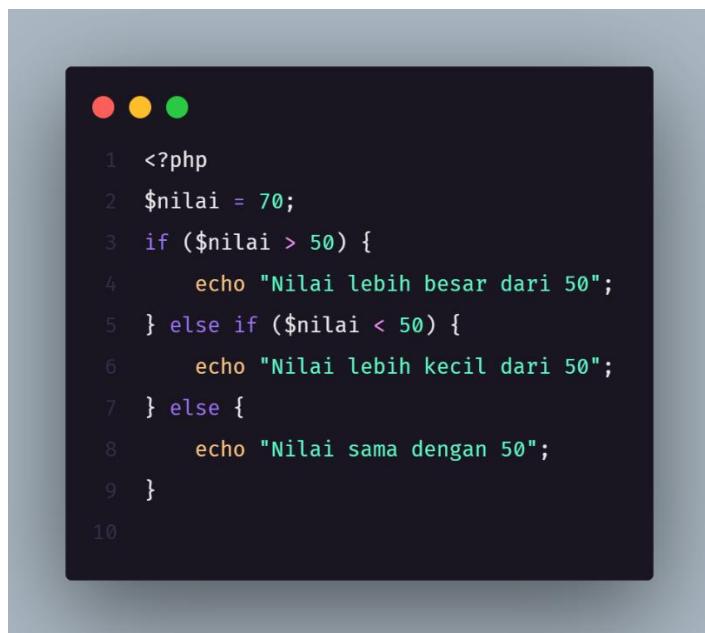
Variabel bernama \$bil diinisialisasi dengan nilai 10. Program kemudian mengecek kondisi if ($\$bil == 10$). Karena nilai 10 memang sama dengan 10, kondisi ini bernilai True (Benar). Blok kode if dieksekusi, yang menghasilkan output "Bilangan sama dengan 10". Blok else, yang hanya akan berjalan jika kondisi if bernilai False,

logika3.php

```
● ● ●  
1 <?php  
2 $bil = 12;  
3 if ($bil == 10) {  
4     echo "Bilangan sama dengan 10";  
5 } else {  
6     echo "Bilangan tidak sama dengan 10";  
7 }  
8
```

Variabel \$bil diinisialisasi dengan nilai 12. Program kemudian mengevaluasi kondisi if ($\$bil == 10$). Karena nilai 12 tidak sama dengan 10, kondisi ini dievaluasi sebagai False (Salah). Akibatnya, blok kode if dilewati, dan program melanjutkan untuk mengeksekusi blok else. Oleh karena itu, output yang akan dihasilkan oleh skrip ini adalah "Bilangan tidak sama dengan 10".

logika4.php



```
1 <?php
2 $nilai = 70;
3 if ($nilai > 50) {
4     echo "Nilai lebih besar dari 50";
5 } else if ($nilai < 50) {
6     echo "Nilai lebih kecil dari 50";
7 } else {
8     echo "Nilai sama dengan 50";
9 }
10
```

Variabel \$nilai diinisialisasi dengan nilai 70. Program kemudian mengevaluasi kondisi pertama, if (\$nilai > 50). Karena 70 memang lebih besar dari 50, kondisi ini bernilai True (Benar). Karena bernilai true, skrip langsung mengeksekusi blok kode pertama dan menghasilkan output "Nilai lebih besar dari 50".

logika5.php



```
1 <?php
2 $nilai = 50;
3 if ($nilai > 50) {
4     echo "Nilai lebih besar dari 50";
5 } else {
6     if ($nilai < 50) {
7         echo "Nilai lebih kecil dari 50";
8     }
9 }
10
```

Variabel \$nilai diinisialisasi ke 50. Kondisi if pertama (\$nilai > 50) dievaluasi, dan karena 50 tidak lebih besar dari 50, kondisi ini bernilai False (Salah). Program kemudian masuk ke blok else. Di dalam blok else, kondisi if kedua (\$nilai < 50) dievaluasi. Karena 50 juga tidak lebih kecil dari 50, kondisi kedua ini juga bernilai False. Akibatnya, echo di dalam if bersarang tidak dieksekusi, dan skrip tersebut tidak menghasilkan output apapun.

logika6.php



```
1 <?php
2 $nilai = 50;
3 if ($nilai > 50) {
4     echo "Nilai lebih besar dari 50";
5 } else {
6     if ($nilai < 50) {
7         echo "Nilai lebih kecil dari 50";
8     } else {
9         echo "Nilai sama dengan 50";
10    }
11 }
12
```

Kode PHP tersebut melakukan pengecekan terhadap nilai yang disimpan dalam variabel \$nilai, yaitu 50. Program menggunakan struktur if–else bersarang untuk menentukan apakah nilai tersebut lebih besar, lebih kecil, atau sama dengan 50. Pertama, kode memeriksa apakah nilai lebih besar dari 50; jika benar, akan muncul pesan “Nilai lebih besar dari 50”. Jika tidak, program memeriksa apakah nilai lebih kecil dari 50; apabila kondisi ini terpenuhi, pesan yang ditampilkan adalah “Nilai lebih kecil dari 50”. Jika kedua kondisi tersebut tidak terpenuhi, berarti nilai sama dengan 50, sehingga program menampilkan “Nilai sama dengan 50”. Dalam kasus ini, karena nilai benar-benar 50, output yang dihasilkan adalah “Nilai sama dengan 50”.

Latihan 10.3

logikahuruf.php



```
1 <?php
2 $huruf = "A";
3 if (($huruf == "A") || ($huruf == "U") || ($huruf == "O") || ($huruf == "E") || ($huruf == "I")) {
4     echo "Termasuk huruf vokal";
5 } else {
6     echo "Tidak termasuk huruf vokal";
7 }
```

Variabel \$huruf berisi nilai "A", kemudian program melakukan pengecekan menggunakan kondisi if dengan operator logika OR (||). Kondisi tersebut membandingkan nilai variabel dengan huruf vokal seperti A, U, E, dan O. Namun, terdapat kesalahan pada salah satu pengecekan, yaitu penggunaan operator = (assignment) pada bagian (\$huruf = "U"), sehingga variabel \$huruf akan berubah menjadi "U" saat kondisi dievaluasi. Meski demikian, karena hasil akhir kondisi tetap dianggap benar, program akan menampilkan "Termasuk huruf vokal".

logikamax.php

```
1 <?php
2 $bil1 = 6;
3 $bil2 = 9;
4 $bil3 = 10;
5 if ($bil1 > $bil2) {
6     $max = $bil1;
7 } else {
8     $max = $bil2;
9 }
10 if ($max > $bil3) {
11     $maxSemua = $max;
12 } else {
13     $maxSemua = $bil3;
14 }
15 echo "Nilai terbesar dari ketiga bilangan adalah " . $maxSemua;
16
```

Kode PHP diatas digunakan untuk mencari nilai terbesar dari tiga bilangan, yaitu \$bil1 = 6, \$bil2 = 9, dan \$bil3 = 10. Program pertama-tama membandingkan bilangan pertama dan kedua; jika \$bil1 lebih besar dari \$bil2, maka nilai maksimum sementara disimpan dalam variabel \$max, tetapi jika tidak, \$max berisi \$bil2. Setelah itu, nilai maksimum sementara tersebut dibandingkan kembali dengan \$bil3 untuk menentukan nilai terbesar dari ketiganya. Hasil akhirnya disimpan dalam variabel \$maxSemua, dan

program menampilkan pesan bahwa nilai terbesar dari ketiga bilangan tersebut adalah 10.

input.html

```
1 <form action="login.php" method="post">
2   Username <input type="text" name="username" /> <br />
3   Password <input type="password" name="pass" /> <br />
4   <input type="submit" name="submit" value="Submit" />
5 </form>
6
```

login.php

```
1 <?php
2 $username = $_POST['username'];
3 $password = $_POST['pass'];
4 if ((($username == "joko") && ($password == "passjoko")) || ((username == "amir") && ($password == "passamir"))) {
5     echo "Login sukses";
6 } else {
7     echo "Login gagal";
8 }
9
```

Kode tersebut terdiri dari dua bagian, yaitu input.html dan login.php, yang bekerja bersama untuk membuat proses login sederhana. Pada input.html, pengguna diminta memasukkan username dan password melalui formulir yang menggunakan metode POST. Data yang

dimasukkan kemudian dikirim ke login.php untuk diproses. Di dalam login.php, program mengambil nilai yang dikirim melalui form menggunakan `$_POST['username']` dan `$_POST['pass']`. Selanjutnya, program memeriksa apakah kombinasi username dan password yang diberikan sesuai dengan pasangan yang telah ditentukan, yaitu ("joko", "passjoko") atau ("amir", "passamir"). Jika salah satu kombinasi cocok, program menampilkan "Login sukses", tetapi jika tidak cocok, program menampilkan "Login gagal".

Latihan 10.4

switch01.php



```
1 <?php
2 $angkaBln = date("n");
3 switch ($angkaBln) {
4     case 1:
5         $namaBln = "Januari";
6         break;
7     case 2:
8         $namaBln = "Februari";
9         break;
10    case 3:
11        $namaBln = "Maret";
12        break;
13    case 4:
14        $namaBln = "April";
15        break;
16    case 5:
17        $namaBln = "Mei";
18        break;
19    case 6:
20        $namaBln = "Juni";
21        break;
22
23    case 7:
24        $namaBln = "Juli";
25        break;
26    case 8:
27        $namaBln = "Agustus";
28        break;
29    case 9:
30        $namaBln = "September";
31        break;
32    case 10:
33        $namaBln = "Oktober";
34        break;
35    case 11:
36        $namaBln = "Nopember";
37        break;
38    case 12:
39        $namaBln = "Desember";
40    }
41 echo "Nama bulan sekarang adalah : " . $namaBln;
```

Program mengambil angka bulan dari fungsi `date("n")`, yang menghasilkan nilai 1 sampai 12 sesuai bulan berjalan. Nilai ini kemudian diperiksa menggunakan struktur `switch`, di mana setiap kasus menentukan nama bulan yang sesuai, seperti Januari untuk 1, Februari untuk 2, dan seterusnya hingga Desember untuk 12. Setelah mencocokkan angka bulan, program menyimpan nama bulan ke variabel `$namaBln` dan menampilkannya melalui perintah `echo`. Dengan demikian, program menghasilkan output berupa nama bulan saat ini sesuai tanggal sistem.

Contoh Studi Kasus

Latihan 10.5

script7-1.php

```
● ● ●
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mencari Bilangan Terbesar dari 3 Bilangan</title>
8  </head>
9
10 <body>
11     <h1>Mencari Bilangan Terbesar dari 3 Bilangan</h1>
12
13     <form action="script7-1proses.php" method="post">
14         <table>
15             <tr>
16                 <td>Masukkan bilangan ke-1</td>
17                 <td>:</td>
18                 <td>
19                     <input type="text" name="bil1">
20                 </td>
21             </tr>
22             <tr>
23                 <td>Masukkan bilangan ke-2</td>
24                 <td>:</td>
25                 <td>
26                     <input type="text" name="bil2">
27                 </td>
28             </tr>
29             <tr>
30                 <td>Masukkan bilangan ke-3</td>
31                 <td>:</td>
32                 <td>
33                     <input type="text" name="bil3">
34                 </td>
35             </tr>
36         </table>
37         <input type="submit" name="submit" value="Submit">
38         <input type="reset" name="reset" value="Hapus">
39     </form>
40 </body>
41
42 </html>
43
```

script7-1proses.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mencari Bilangan Terbesar dari 3 Bilangan</title>
8  </head>
9
10 <body>
11     <h1>Mencari Bilangan Terbesar dari 3 Bilangan</h1>
12
13     <?php
14         $bilangan1 = $_POST['bil1']; // membaca input bilangan pertama
15         $bilangan2 = $_POST['bil2']; // membaca input bilangan kedua
16         $bilangan3 = $_POST['bil3']; // membaca input bilangan ketiga
17
18         // cara ke -1
19         // idenya adalah mencari bilangan x yang terbesar dari bil 1 dan bil2, lalu x dibandingkan dengan
20         // bilangan ke-3. Jika x > dari bil 3, maka x adalah max dari semua bilangan. Jika tidak, maka bil 3 sbg max nya
21
22         if ($bilangan1 > $bilangan2)
23             $maxSementara = $bilangan1;
24         else
25             $maxSementara = $bilangan2;
26
27         if ($bilangan3 > $maxSementara)
28             $maxAkhir = $bilangan3;
29         else
30             $maxAkhir = $maxSementara;
31
32         echo "<p>Nilai maksimum dari " . $bilangan1 . ", " . $bilangan2 . ", dan " . $bilangan3 . " adalah : " . $maxAkhir . "</p>";
33
34         // cara ke - 2
35         // idenya adalah langsung membandingka masing-masing bilangan dengan bilangan yang lain.
36         // misal: jika bil 1 > bil2 dan bil 1 > bil 3, maka max adalah bil1.
37         if (($bilangan1 > $bilangan2) && ($bilangan1 > $bilangan3))
38             $maxAkhir = $bilangan1;
39         else
40             if (($bilangan2 > $bilangan1) && ($bilangan2 > $bilangan3))
41                 $maxAkhir = $bilangan2;
42             else
43                 if (($bilangan3 > $bilangan1) && ($bilangan3 > $bilangan2))
44                     $maxAkhir = $bilangan3;
45
46         echo "<p>Nilai maksimum dari " . $bilangan1 . ", " . $bilangan2 . ", dan " . $bilangan3 . " adalah : " . $maxAkhir . "</p>";
47
48         // cara ke - 3
49         // idenya hampir sama dengan cara ke - 1, tapi tidak dengan mencari max sementara.
50
51         if ($bilangan1 > $bilangan3) {
52             if ($bilangan1 > $bilangan2)
53                 $maxAkhir = $bilangan1;
54             else
55                 $maxAkhir = $bilangan3;
56         } else {
57             if ($bilangan2 > $bilangan3)
58                 $maxAkhir = $bilangan2;
59             else
60                 $maxAkhir = $bilangan3;
61         }
62
63         echo "<p>Nilai maksimum dari " . $bilangan1 . ", " . $bilangan2 . ", dan " . $bilangan3 . " adalah : " . $maxAkhir . "</p>";
64     ?>
65 </body>
66
67 </html>
```

Program ini terdiri dari dua file, yaitu script7-1.php dan script7-1proses.php. File pertama berfungsi sebagai halaman input yang meminta pengguna memasukkan tiga bilangan melalui

sebuah formulir HTML. Setelah data dikirim menggunakan metode POST, file script7-1proses.php akan memproses ketiga bilangan tersebut untuk menentukan nilai yang terbesar.

script7-2.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mencari Status Bilangan</title>
8  </head>
9
10 <body>
11     <h1>Mencari Status Bilangan</h1>
12
13     <form action="script7-2proses.php" method="post">
14         Masukkan sebuah bilangan <input type="text" name="bil">
15         <input type="submit" name="submit" value="Submit">
16     </form>
17 </body>
18
19 </html>
20
```

script7-2proses.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mencari Status Bilangan</title>
8  </head>
9
10 <body>
11     <h1>Mencari Status Bilangan</h1>
12
13     <?php
14     $bilangan = $_POST['bil'];
15
16     // cara ke - 1
17
18     if ($bilangan > 0)
19         echo "<p>Bilangan " . $bilangan . " adalah positif</p>";
20     else
21         if ($bilangan < 0)
22             echo "<p>Bilangan " . $bilangan . " adalah negatif</p>";
23     else
24         echo "<p>Bilangan " . $bilangan . " adalah nol</p>";
25
26     // cara ke - 2
27     if ($bilangan > 0)
28         $status = "positif";
29     else
30         if ($bilangan < 0)
31             $status = "negatif";
32     else
33         $status = "nol";
34
35     echo "<p>Bilangan " . $bilangan . " adalah " . $status . "</p>";
36     ?>
37 </body>
38
39 </html>
40
```

Program ini terdiri dari dua file, yaitu script7-2.php sebagai halaman input dan script7-2proses.php sebagai halaman pemrosesan. Pada halaman input, pengguna diminta memasukkan sebuah bilangan melalui formulir yang kemudian dikirimkan menggunakan metode POST. File script7-2proses.php kemudian membaca bilangan tersebut dan menentukan statusnya, apakah termasuk bilangan positif, negatif, atau nol. Terdapat dua metode pengecekan yang digunakan dalam program.

Soal

Soal 1

soal1.php



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Cek Tahun Kabisat</title>
8 </head>
9
10 <body>
11   <h1>Cek Tahun Kabisat</h1>
12
13   <form action="soal1proses.php" method="post">
14     Masukkan Tahun <input type="number" name="tahun">
15     <input type="submit" name="submit" value="Submit">
16   </form>
17 </body>
18
19 </html>
20
```

soal1proses.php



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Cek Tahun Kabisat</title>
8 </head>
9
10 <body>
11   <h1>Cek Tahun Kabisat</h1>
12   <?php
13     $tahun = $_POST['tahun'];
14
15     if (($tahun % 400 == 0) || ($tahun % 4 == 0 && $tahun % 100 != 0)) {
16       echo "<p>Tahun $tahun adalah tahun kabisat.</p>";
17     } else {
18       echo "<p>Tahun $tahun bukan tahun kabisat.</p>";
19     }
20   ?>
21 </body>
22
23 </html>
24
```

Program ini terdiri dari dua file, yaitu soal1.php sebagai halaman input dan soal1proses.php sebagai halaman pemrosesan. Pada halaman soal1.php, pengguna diminta memasukkan sebuah tahun melalui formulir yang kemudian dikirimkan menggunakan metode POST. File

soal1proses.php menerima input tersebut dan memprosesnya menggunakan aturan penentuan tahun kabisat. Program menggunakan kondisi yang mengecek apakah tahun habis dibagi 400, atau habis dibagi 4 tetapi tidak habis dibagi 100. Jika salah satu kondisi terpenuhi, program menyatakan bahwa tahun tersebut adalah tahun kabisat. Jika tidak, tahun tersebut dinyatakan bukan tahun kabisat. Hasilnya kemudian ditampilkan dalam bentuk teks di halaman web.

soal2.php



```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hitung Gaji Karyawan Honorer</title>
8  </head>
9
10 <body>
11     <h1>Hitung Gaji Karyawan Honorer</h1>
12
13     <form action="soal2proses.php" method="post">
14         Jumlah jam kerja per minggu:
15         <input type="number" name="jam">
16         <input type="submit" value="Hitung">
17     </form>
18 </body>
19
20 </html>
21
```

soal2proses.php



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Hitung Gaji Karyawan</title>
8 </head>
9
10 <body>
11     <h1>Hitung gaji Karyawan</h1>
12
13     <?php
14         $jamKerja = $_POST['jam'];
15         $upahPerJam = 2000;
16         $upahLembur = 300;
17         $batasJam = 48;
18
19         if ($jamKerja > $batasJam) {
20             $jamBiasa = $batasJam;
21             $jamLembur = $jamKerja - $batasJam;
22
23             $gaji = ($jamBiasa * $upahPerJam) + ($jamLembur * $upahLembur);
24         } else {
25             $gaji = $jamKerja * $upahPerJam;
26         }
27
28         echo "<p>Jumlah jam kerja: $jamKerja jam</p>";
29         echo "<p>Gaji yang diterima: Rp. $gaji</p>";
30     ?>
31 </body>
32
33 </html>
34
```

Program ini terdiri dari dua file, yaitu soal2.php sebagai halaman input dan soal2proses.php sebagai halaman pemrosesan. Pada halaman soal2.php, pengguna diminta memasukkan jumlah jam kerja per minggu seorang karyawan honorer melalui formulir. Setelah data dikirimkan menggunakan metode POST, file soal2proses.php akan memproses jumlah jam tersebut untuk menghitung total gaji. Program menetapkan upah kerja normal sebesar Rp 2.000 per jam dan upah lembur sebesar Rp 300 per jam, dengan batas jam kerja normal sebanyak 48 jam per minggu. Jika jumlah jam kerja melebihi 48, program membagi jam kerja menjadi jam biasa dan jam lembur, kemudian menghitung gaji berdasarkan masing-masing tarif. Jika jam kerja tidak melebihi batas, perhitungan dilakukan hanya dengan tarif jam biasa.

soal3.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hitung Gaji Karyawan</title>
8  </head>
9
10 <body>
11     <h1>Hitung gaji Karyawans</h1>
12
13     <form action="soal3proses.php" method="post">
14         Jumlah jam kerja per minggu:
15         <input type="number" name="jam"> <br>
16
17         Golongan:
18         <select name="golongan">
19             <option value="">--Pilih Golongan--</option>
20             <option value="A">A (Rp 4.000/jam)</option>
21             <option value="B">B (Rp 5.000/jam)</option>
22             <option value="C">C (Rp 6.000/jam)</option>
23             <option value="D">D (Rp 7.500/jam)</option>
24         </select> <br>
25
26         <input type="submit" name="submit" value="Submit">
27     </form>
28 </body>
29
30 </html>
31
```

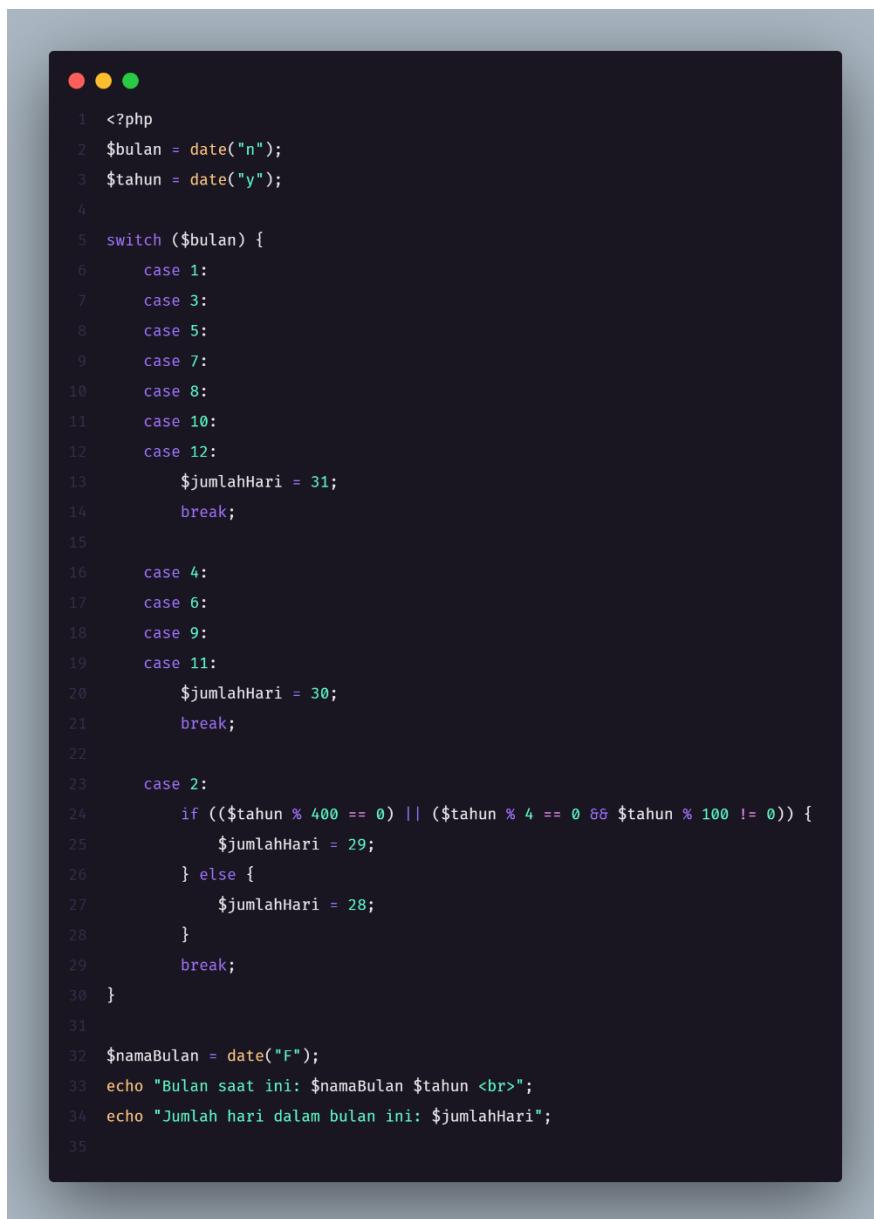
soal3proses.php



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Hitung Gaji Karyawan</title>
8 </head>
9
10 <body>
11     <h1>Hitung Gaji Karyawan</h1>
12
13
14     <?php
15         $jamKerja = $_POST['jam'];
16         $golongan = $_POST['golongan'];
17         $upahLembur = 3000;
18         $batasJam = 48;
19
20         if ($golongan == "A") {
21             $upahPerJam = 4000;
22         } elseif ($golongan == "B") {
23             $upahPerJam = 5000;
24         } elseif ($golongan == "C") {
25             $upahPerJam = 6000;
26         } else {
27             $upahPerJam = 7500;
28         }
29
30         if ($jamKerja > $batasJam) {
31             $jamBiasa = $batasJam;
32             $jamLembur = $jamKerja - $batasJam;
33
34             $gaji = ($jamBiasa * $upahPerJam) + ($jamLembur * $upahLembur);
35         } else {
36             $gaji = $jamKerja * $upahPerJam;
37         }
38
39         echo "<p>Golongan: $golongan</p>";
40         echo "<p>Jumlah jam kerja: $jamKerja jam</p>";
41         echo "<p>Gaji yang diterima: Rp. $gaji</p>";
42     ?>
43 </body>
44
45 </html>
```

Program ini terdiri dari dua file, yaitu soal3.php sebagai halaman input dan soal3proses.php sebagai halaman pemrosesan. Pada halaman input, pengguna diminta memasukkan jumlah jam kerja per minggu serta memilih golongan karyawan yang menentukan besarnya upah per jam. Setiap golongan memiliki tarif berbeda, yaitu golongan A sebesar Rp 4.000 per jam, golongan B Rp 5.000 per jam, golongan C Rp 6.000 per jam, dan golongan D Rp 7.500 per jam. Setelah data dikirim melalui metode POST, file soal3proses.php membaca input tersebut dan menentukan upah berdasarkan golongan yang dipilih. Program juga menerapkan aturan lembur, di mana jam kerja di atas 48 jam dihitung sebagai lembur dengan tarif Rp 3.000 per jam. Jika jam kerja tidak melebihi batas tersebut, maka gaji dihitung hanya berdasarkan upah per jam sesuai golongan.

soal4.php



```
1 <?php
2 $bulan = date("n");
3 $tahun = date("y");
4
5 switch ($bulan) {
6     case 1:
7     case 3:
8     case 5:
9     case 7:
10    case 8:
11    case 10:
12    case 12:
13        $jumlahHari = 31;
14        break;
15
16    case 4:
17    case 6:
18    case 9:
19    case 11:
20        $jumlahHari = 30;
21        break;
22
23    case 2:
24        if (($tahun % 400 == 0) || ($tahun % 4 == 0 && $tahun % 100 != 0)) {
25            $jumlahHari = 29;
26        } else {
27            $jumlahHari = 28;
28        }
29        break;
30    }
31
32 $namaBulan = date("F");
33 echo "Bulan saat ini: $namaBulan $tahun <br>";
34 echo "Jumlah hari dalam bulan ini: $jumlahHari";
35
```

Program ini berfungsi untuk menampilkan jumlah hari pada bulan berjalan berdasarkan tanggal sistem. Program mengambil nilai bulan dan tahun saat ini menggunakan fungsi date(), kemudian menentukan jumlah hari dalam bulan tersebut melalui struktur switch. Untuk bulan yang memiliki 31 hari maupun 30 hari, program langsung menetapkan jumlah harinya sesuai kelompok bulan masing-masing. Khusus bulan Februari, program melakukan pengecekan tambahan untuk menentukan apakah tahun tersebut merupakan tahun kabisat dengan melihat kondisi pembagian 400, 100, dan 4. Jika tahun kabisat, Februari memiliki 29 hari; jika tidak, jumlah harinya 28. Setelah jumlah hari ditentukan, program menampilkan nama bulan saat ini beserta jumlah hari yang dimilikinya.