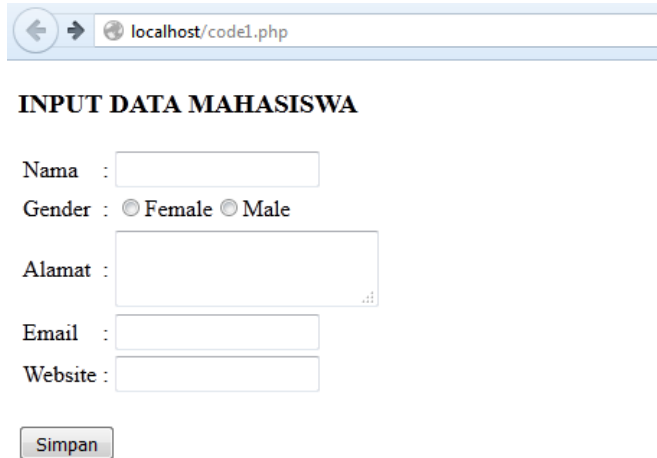


## Tugas 8 Web Lanjut

### PEMBAHASAN!

Hasil running masing-masing kode :

- Code1.php



← → localhost/code1.php

**INPUT DATA MAHASISWA**

Nama :

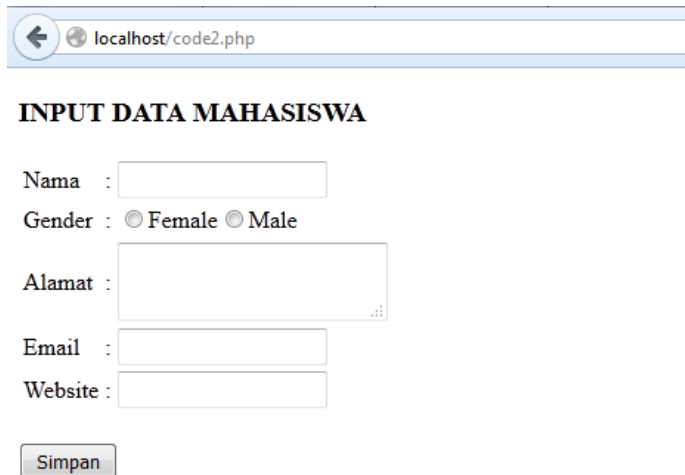
Gender : ☐ Female ☐ Male

Alamat :

Email :

Website :

- Code2.php



← localhost/code2.php

**INPUT DATA MAHASISWA**

Nama :

Gender : ☐ Female ☐ Male

Alamat :

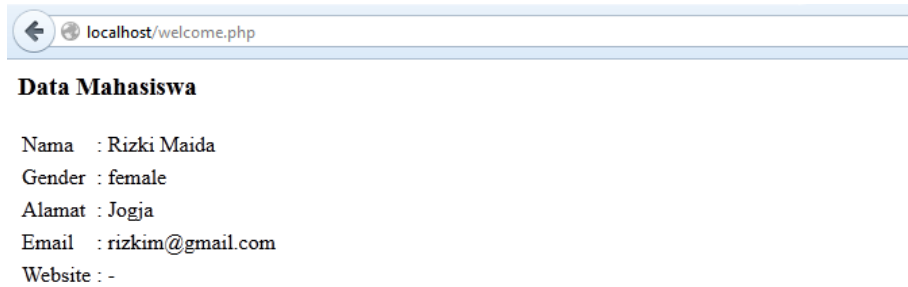
Email :

Website :

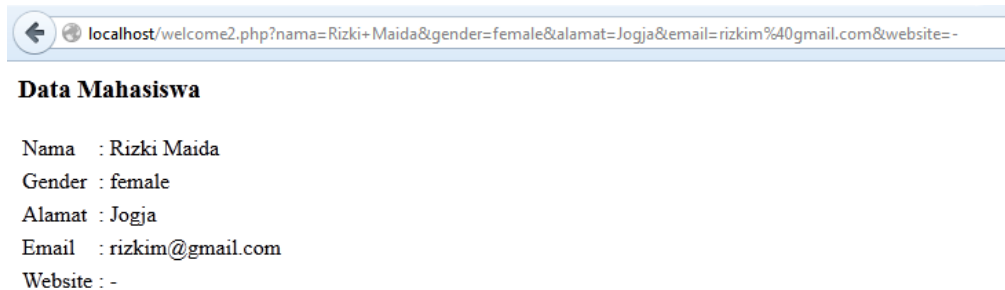
Penjelasan : Code 1 dan code 2 menampilkan hasil running yang sama, code1.php mengacu pada script welcome.php, sedangkan code2.php mengacu pada script welcome2.php

### Ketika dimasukkan nilai :

- Code1.php :



- Code2.php



### Perbandingan:

1. Terdapat perbedaan pada code1.php dan code2.php yaitu pada code1.php nilai variable yang dikirim tidak ditampilkan pada URL, sedangkan pada code2.php nilai variabel yang dikirim ditampilkan pada URL.
2. Code1.php menggunakan <form action="welcome.php" method="post">, sedangkan code2.php menggunakan <form action="welcome2.php" method="get">
3. Pada welcome.php menggunakan <?php echo \$\_POST["variabel"]; ?>, sedangkan pada welcome2.php menggunakan <?php echo \$\_GET["variabel"]; ?>  
Variabel dapat berisi nama, gender, alamat, email dan website

### Analisis source code PHP:

1. Action  
Atribut action berfungsi untuk mendefinisikan tindakan apa yang akan dilakukan ketika formulir dikirimkan. Pada kasus diatas, atribut ini akan mendefinisikan akan dieksekusi kemana ketika user sudah memasukkan data menekan tombol klik “simpan”.
  - Code1  
Action ="welcome.php" -> artinya action akan diproses pada halaman lain, yaitu halaman script yang bernama welcome.php.

- Code2  
Action="welcome2.php" -> sama dengan code1, pada action code2 action juga diakses di halaman lain, yaitu halaman script bernama welcome2.php

## 2. Method

Terdapat dua method dalam php yaitu POST dan GET

- Code 1  
Menggunakan method POST
- Code 2  
Menggunakan method GET

## Code 3

- Ketika kode 3 di run, maka tampilan sebagai berikut

The screenshot shows a web browser window with the address bar displaying 'localhost/code3.php'. The browser's tab bar shows several tabs, including 'Apps', 'New Tab', 'Blog Pribadi Mahasiswa', and others. The main content area displays a form titled 'INPUT DATA MAHASISWA'. The form contains the following fields and labels:

- Nama :  \*
- Jenis Kelamin : ☐ Perempuan ☐ Laki-laki \*
- Alamat :
- Email :  \*
- Website :

Below the form is a button labeled 'Simpan'. Below the button, a red message states: '\* harus diisi.'.

Below the form, the output of the script is displayed under the heading 'Data Mahasiswa':

- Nama :
- Jenis Kelamin : Notice: Undefined index: gender in C:\xampp\htdocs\code3.php on line 122
- Alamat : Notice: Undefined index: alamat in C:\xampp\htdocs\code3.php on line 127
- Email : Notice: Undefined index: email in C:\xampp\htdocs\code3.php on line 132
- Website : Notice: Undefined index: website in C:\xampp\htdocs\code3.php on line 137

- Setelah di klik “simpan”

Tanpa dimasukkan nilai terlebih dahulu

### INPUT DATA MAHASISWA

Dewi Anisa Istiqomah dewianisaist.ugm@gmail.com

Nama :  \* Nama harus diisi  
Jenis Kelamin : ☐ Perempuan ☐ Laki-laki \* Jenis Kelamin harus diisi  
Alamat :   
Email :  \* Email harus diisi  
Website :

\* harus diisi.

### Data Mahasiswa

Nama :  
Jenis Kelamin : Notice: Undefined index: gender in C:\xampp\htdocs\code3.php on line 122  
Alamat :  
Email :  
Website :

Jika dimasukkan nilai terlebih dahulu

### INPUT DATA MAHASISWA

Dewi Anisa Istiqomah dewianisaist.ugm@gmail.com

Nama : rizki \*  
Jenis Kelamin : ☒ Perempuan ☐ Laki-laki \*  
Alamat : aa  
Email : aa \* Format email salah  
Website : aa URL salah

\* harus diisi.

### Data Mahasiswa

Nama : rizki  
Jenis Kelamin : Perempuan  
Alamat : aa  
Email : aa  
Website : aa

Berikut adalah analisis coding code3.php

```
// define variables and set to empty values  
$namaErr = $genderErr = $alamatErr = $emailErr = $websiteErr = "";  
$nama = $gender = $alamat = $email = $website = "";
```

Berfungsi sebagai pemberi nilai kosong pada variabel diatas

```
]if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

Mendeklarasikan bahwa menggunakan metode “POST”

```
if (empty($_POST["nama"])) {  
    $namaErr = "Nama harus diisi";  
}  
else{  
    $nama = test_input($_POST["nama"]);  
    // check if name only contains letters and whitespace  
    if (!preg_match("/^[a-zA-Z ]*$/", $nama)) {  
        $namaErr = "Inputan salah. Hanya dapat huruf dan spasi";  
    }  
}
```

Berfungsi sebagai pengecekan:

- Jika form “nama” kosong maka akan ditampilkan tulisan “nama harus diisi”
- Jika tidak kosong maka “\$nama” akan berisi inputan user pada form “nama” serta dilakukan pengecekan kembali yaitu jika input berisi karakter selain huruf dan spasi maka akan ditampilkan “inputan salah. Hanya dapat huruf dan spasi”.

```
if (empty($_POST["gender"])) {  
    $genderErr = "Jenis Kelamin harus diisi";  
}  
else {  
    $gender = test_input($_POST["gender"]);  
}
```

Berfungsi sebagai pengecekan:

- Jika form “gender” kosong maka akan ditampilkan tulisan “Jenis kelamin harus diisi”
- Jika tidak kosong maka “\$gender” akan berisi inputan user pada form “gender”.

```
if (empty($_POST["alamat"])) {  
    $alamat = "";  
}  
else {  
    $alamat = test_input($_POST["alamat"]);  
}
```

Berfungsi sebagai pengecekan:

- Jika alamat kosong maka form “alamat” tidak akan menampilkan apapun pada fo
- Jika tidak kosong maka “alamat” akan berisi data alamat dari inputan user.

```
if (empty($_POST["email"])) {  
    $emailErr = "Email harus diisi";  
}  
else {  
    $email = test_input($_POST["email"]);  
    // check if e-mail address is well-formed  
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
        $emailErr = "Format email salah";  
    }  
}
```

Berfungsi sebagai pengecekan:

- Jika form “email” kosong maka akan muncul tulisan “Email harus diisi”
- Jika tidak maka email berisi data email inputan user pada variabel \$email, serta akan dilakukan pengecekan kembali sebagai validasi email, jika format salah maka akan muncul tulisan “Format email salah”.

```
if (empty($_POST["website"])) {  
    $website = "";  
}  
else {  
    $website = test_input($_POST["website"]);  
    // check if URL address syntax is valid (this regular expression also allows dashes in the URL)  
    if (!preg_match("/\b(?:?:https?|ftp):\/\/[a-z0-9+&@#\/%?~_!|:,.;]*[-a-z0-9+&@#\/%?~_!|]/i",$website)){  
        $websiteErr = "URL salah";  
    }  
}
```

Berfungsi sebagai pengecekan website:

- Jika form “website” tidak diisi maka form website tidak berisi apa-apa
- Jika tidak maka isi \$website adalah data website dari inputan user, serta dilakukan pengecekan format url, jika format salah maka akan ditampilkan “URL salah”.

```
function test_input($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```

Merupakan suatu function test\_input -> untuk membatasi data-data yang akan masuk

- Trim(\$data) : menghapus spasi di kanan dan kiri teks, jika spasi lebih dari satu maka akan dihitung satu saja
- Stripslashes(\$data) : menghilangkan backslash
- htmlspecialchars(\$data) : mengkonversi karakter yang telah ditetapkan "<" (less than) dan ">" (greater than) ke entitas HTML.

```
?>  
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);  
?>
```

Metode menggunakan post, action menggunakan htmlspecialchars seperti dijelaskan diatas.

### Perbedaan GET dan POST

No.	POST	GET
1.	tidak menampilkan nilai variable pada URL	menampilkan nilai variable pada URL
2.	Lebih aman	Kurang aman, contohnya password ditampilkan pada url
3.	Dibatasi panjang string hingga 2047 karakter	Tidak dibatasi
4.	Pengambilan data menggunakan \$_POST	Pengambilan data menggunakan \$_GET
5.	Biasanya digunakan untuk input dari form	Biasanya menggunakan input dari link atau akses menggunakan link
6.	Digunakan untuk mengirimkan data rahasia, seperti password	Digunakan untuk mengirim/mengambil data pablik seperti id_user atau id_halaman