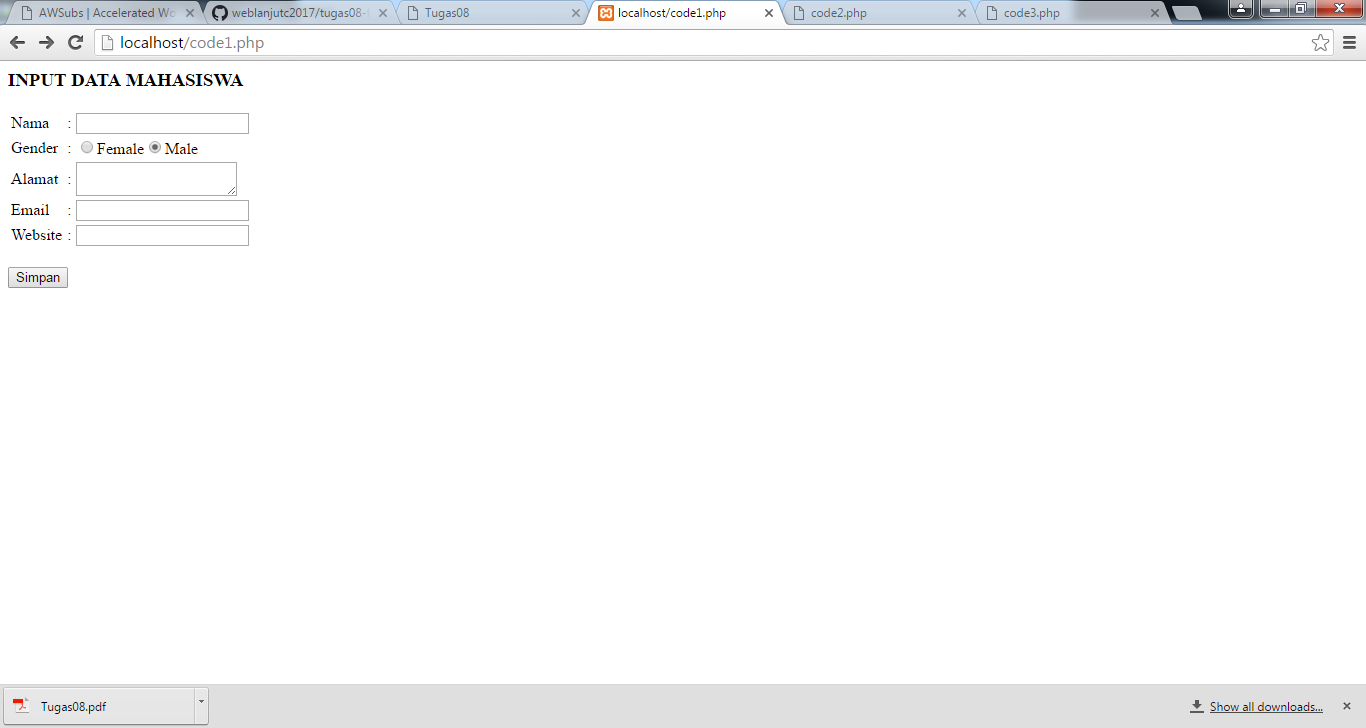
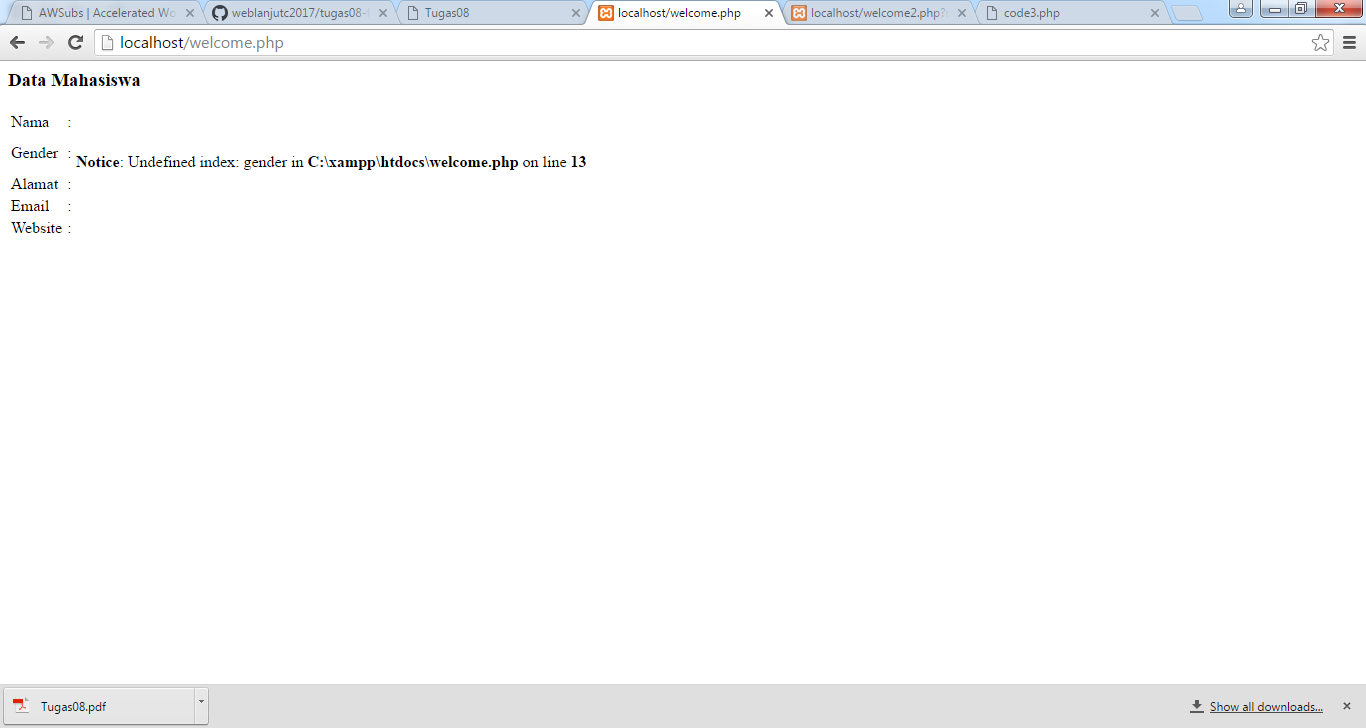
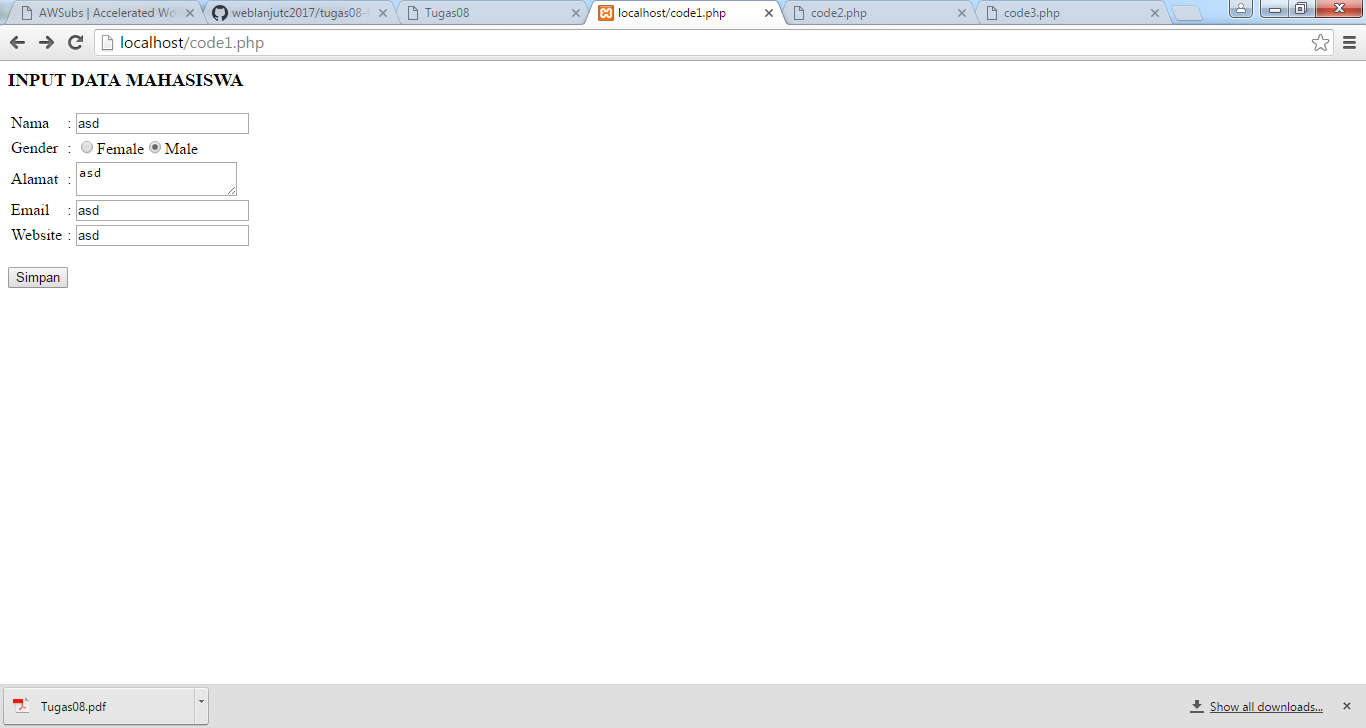
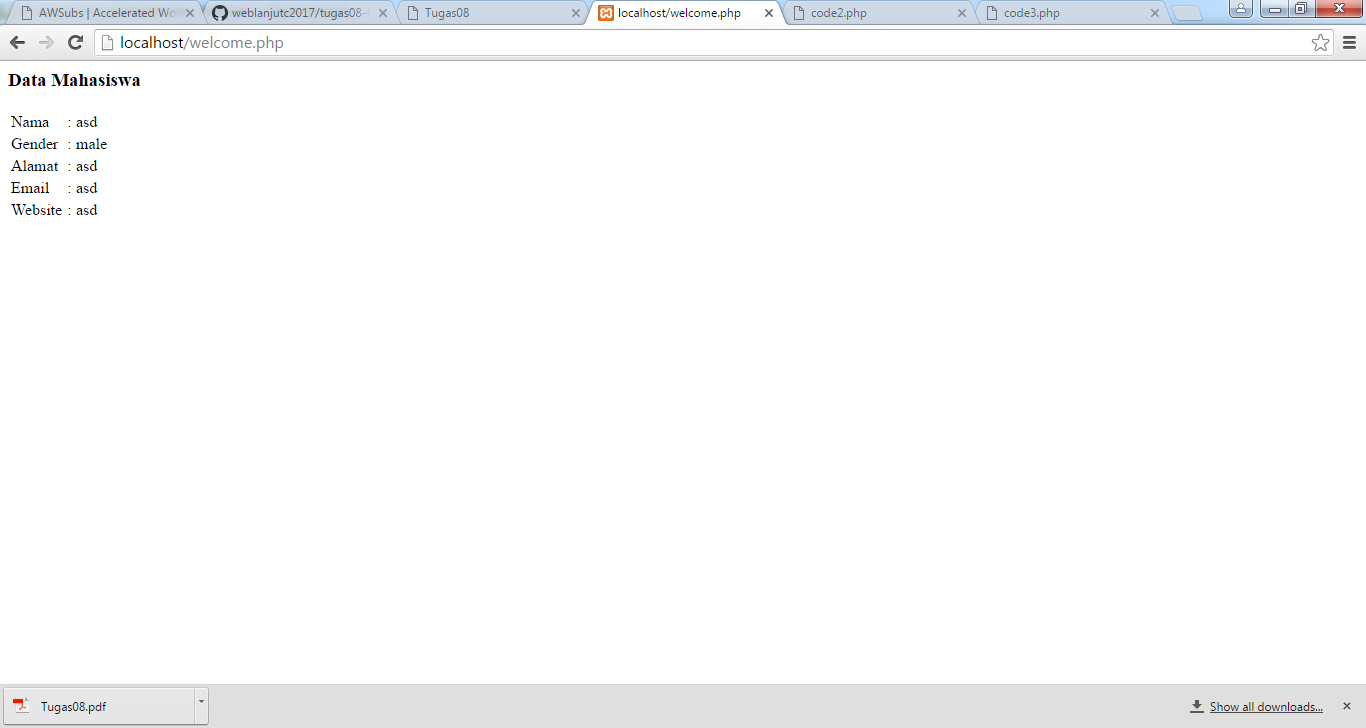
1. Code 1

sebelum diisikan data pada code 1

Setelah diisikan informasi pada code 1 dan meng-klik tombol simpan

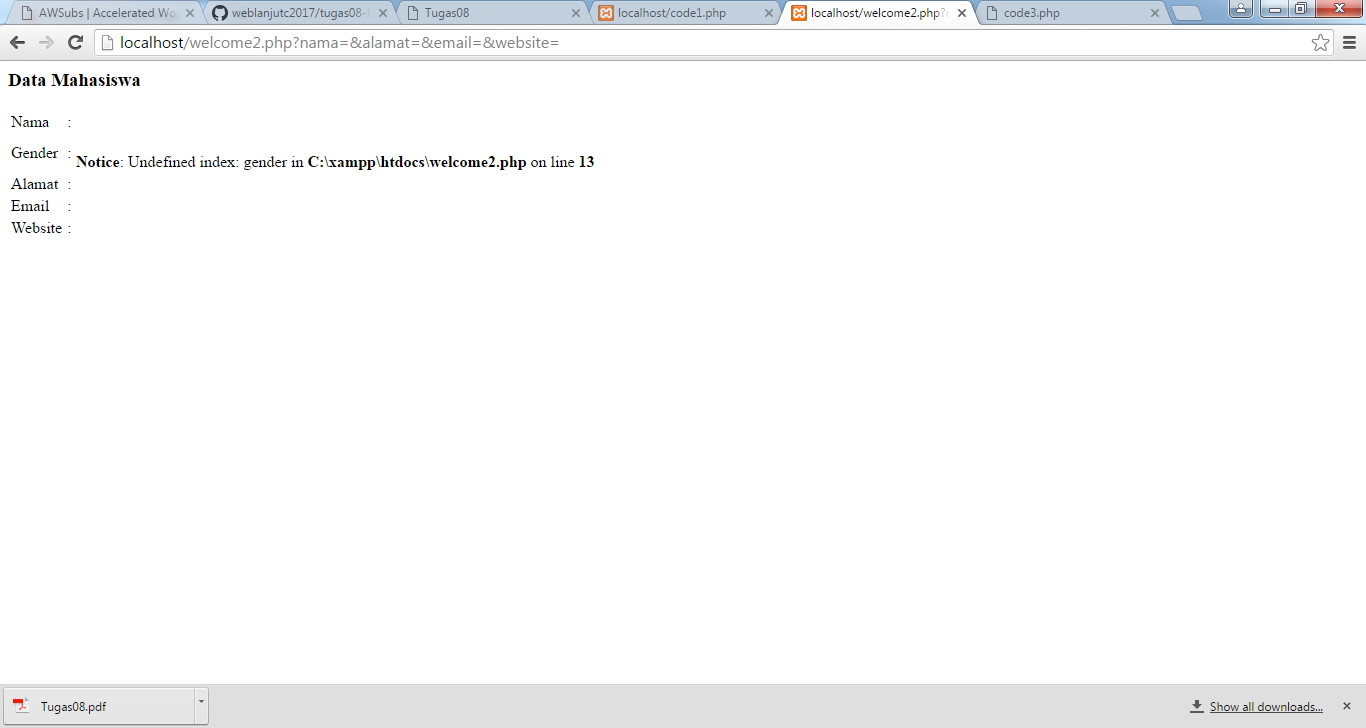
Pada code 1, dia terhubung ke welcome.php. Code 1 merupakan tempat untuk menginputkan informasi dari mahasiswa. Yang kemudian akan dikirim ke welcome.php ketika tombol simpan di klik. Pada halaman welcome.php menampilkan apa yang telah diisikan pada code 1. Ketika diisikan informasi dan klik simpan, hanya terjadi perubahan pada halamannya saja.

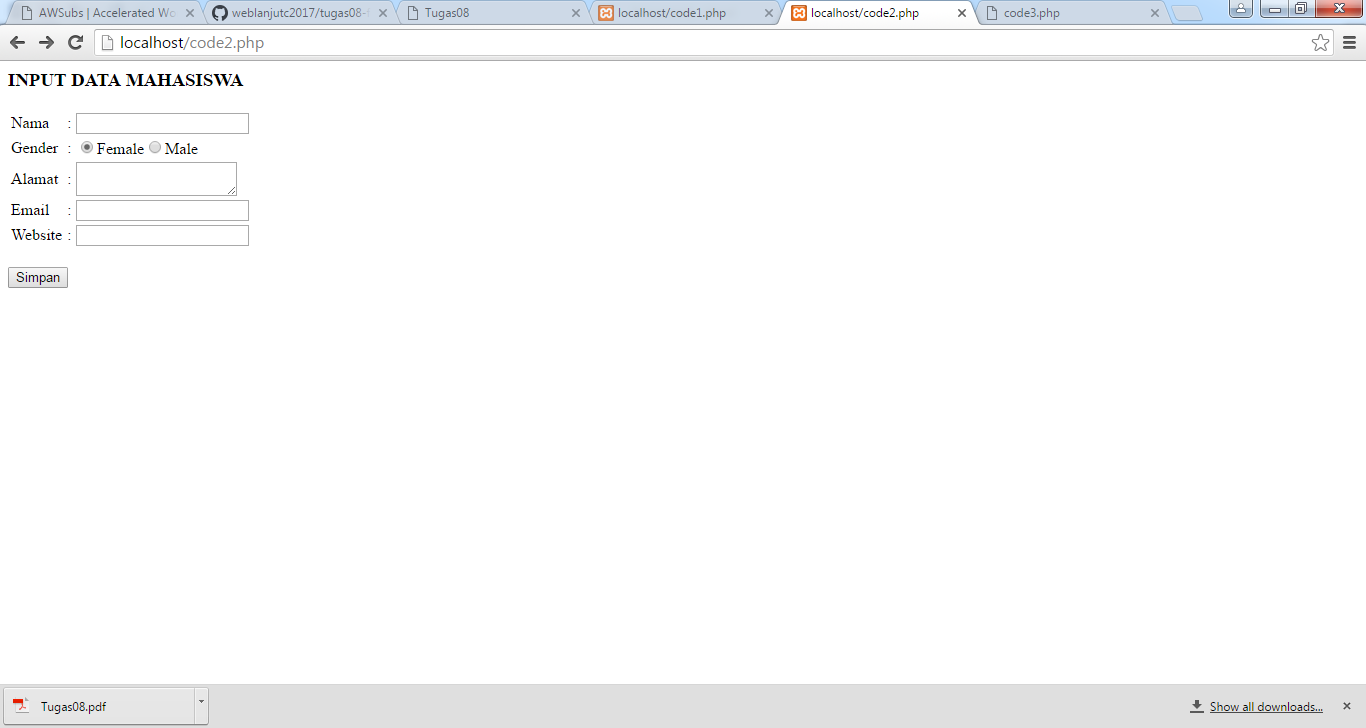
<form action="welcome.php" method="post">

Ini merupakan kode untuk menjalankan function POST pada welcome.php ketika tombol simpan dijalankan.

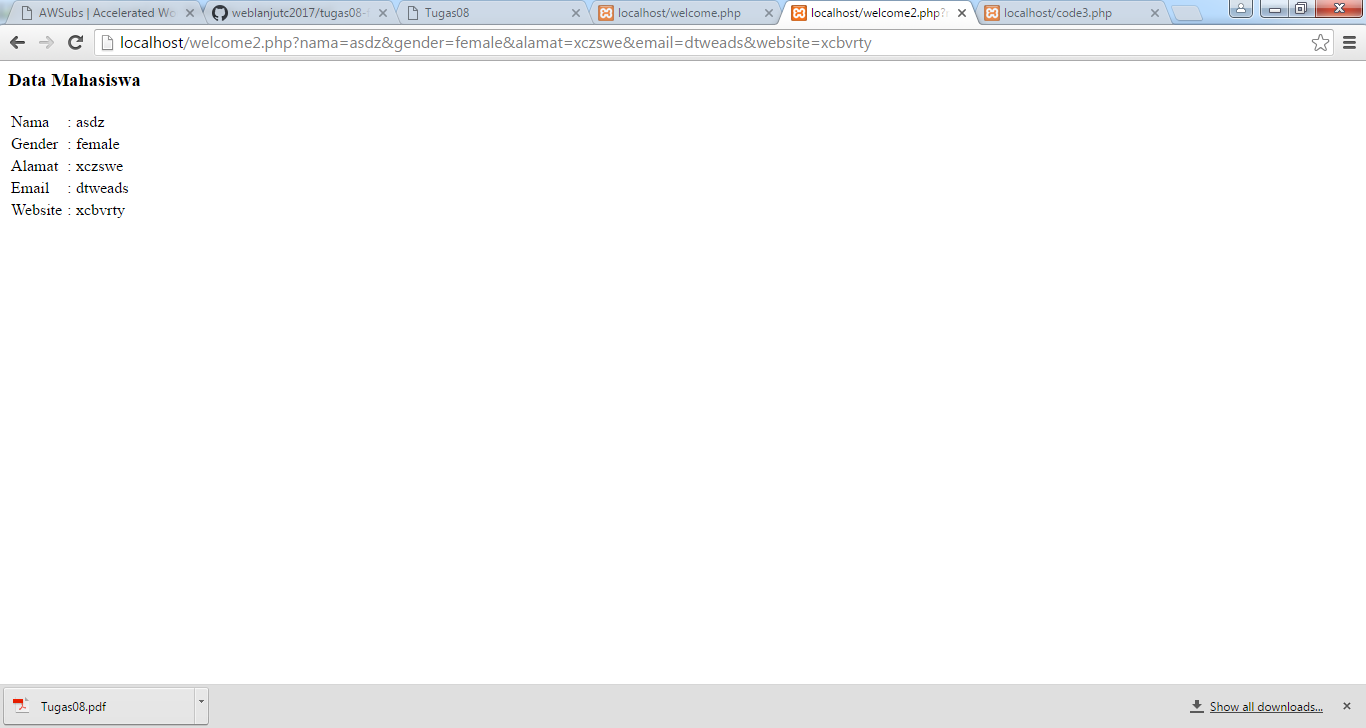
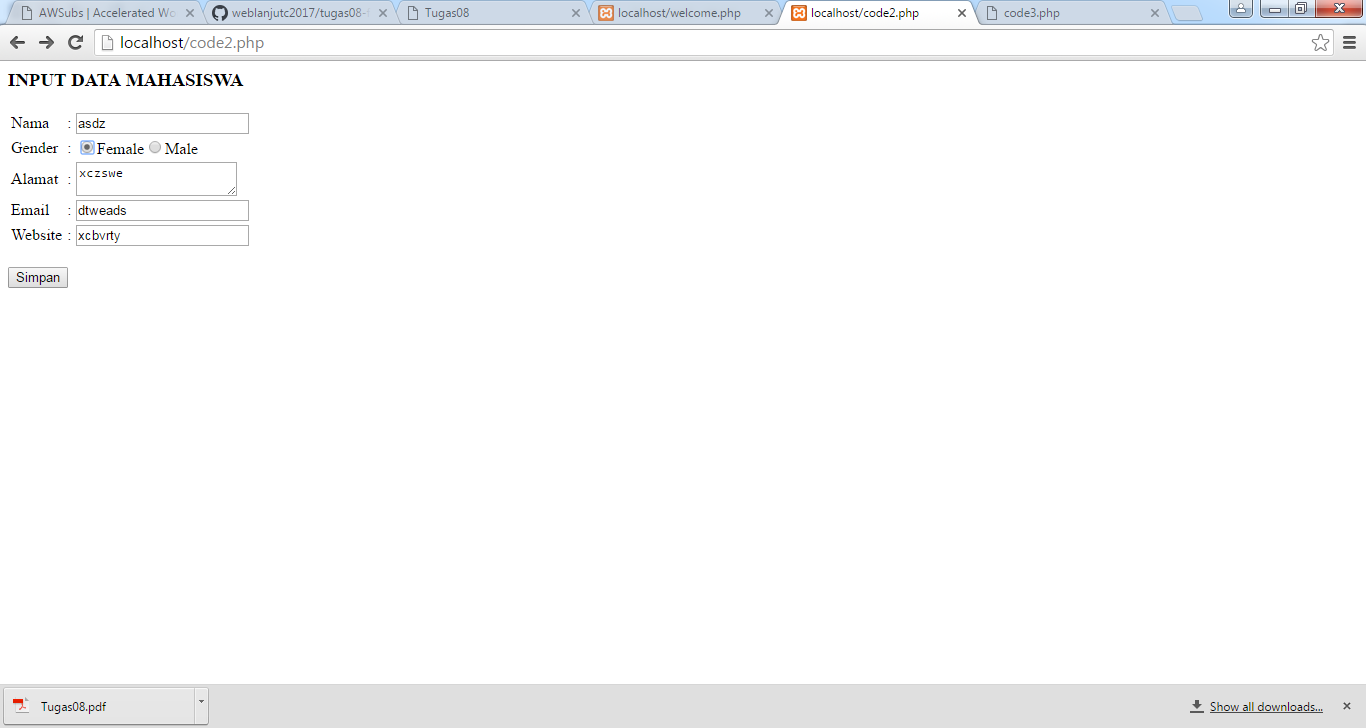
Variable yang terdapat pada welcome.php antara lain:

* echo $\_POST["nama"] : yang berfungsi menampilkan langsung nilai dari form bervalue nama.
* echo $\_POST["gender"] : yang berfungsi menampilkan langsung nilai dari form bervalue gender.
* echo $\_POST["alamat"] : yang berfungsi menampilkan langsung nilai dari form bervalue alamat.
* echo $\_POST["email"] : yang berfungsi menampilkan langsung nilai dari form bervalue email.
* echo $\_POST["website"] : yang berfungsi menampilkan langsung nilai dari form bervalue website.

1. Code 2



Setelah diisikan informasi dan klik simpan

pada code 2 sama saja dengan code 1 namun ketika diisikan informasi dan klik simpan, yang terjadi pada welcome2.php adanya perubahan pada URL dan halamannya. Informasi yang ditampilkan juga tertampil pada URL.

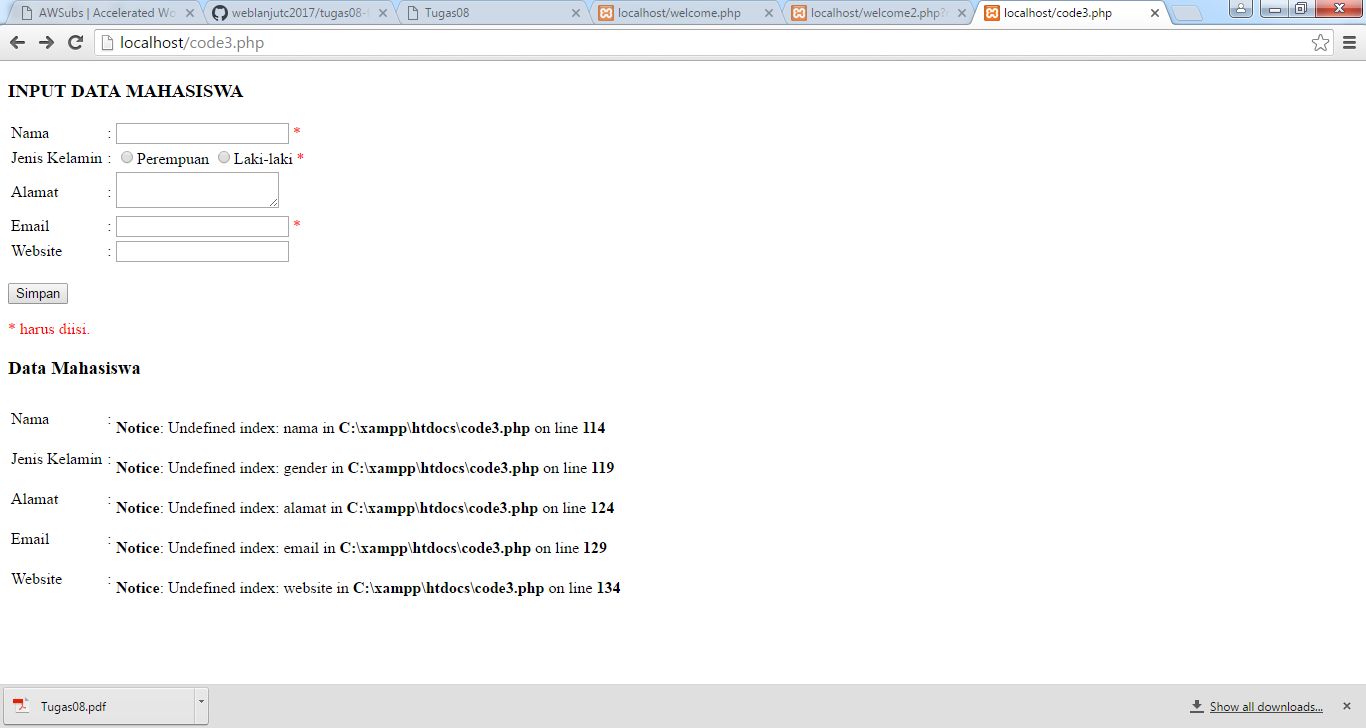
<form action="welcome2.php" method="get">

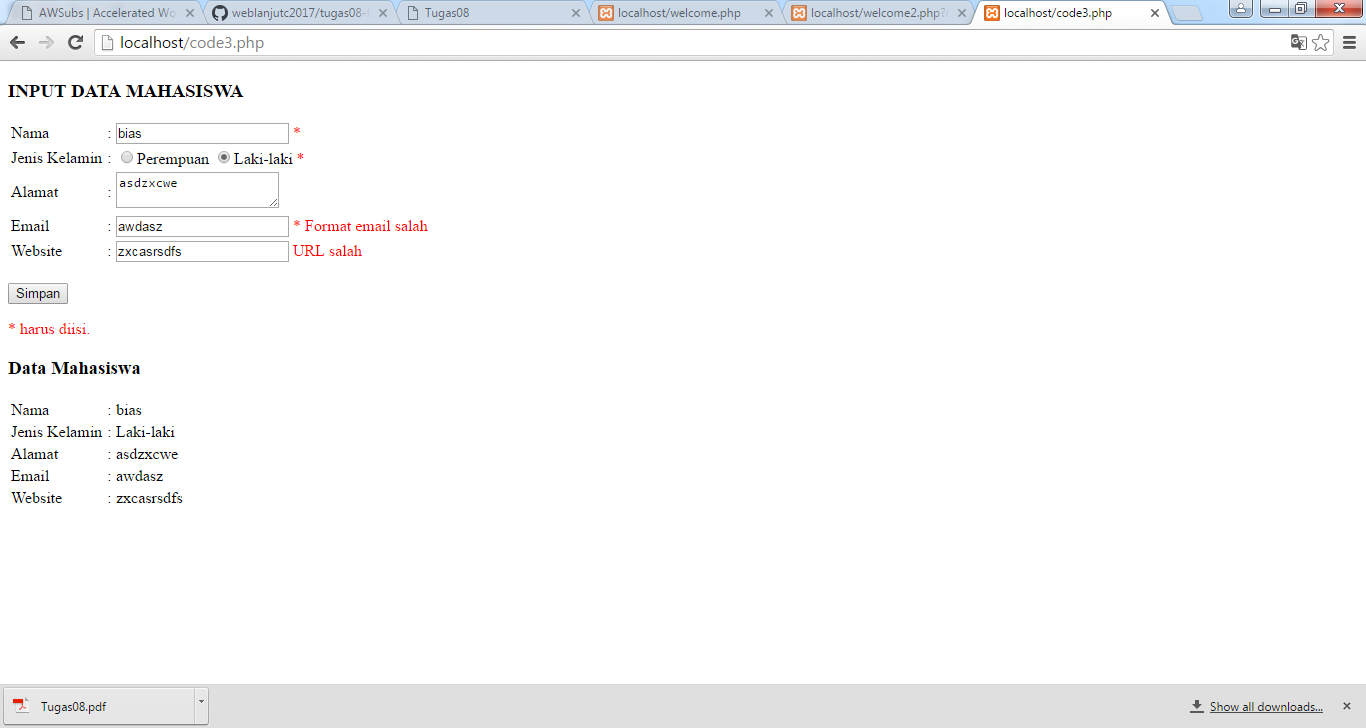
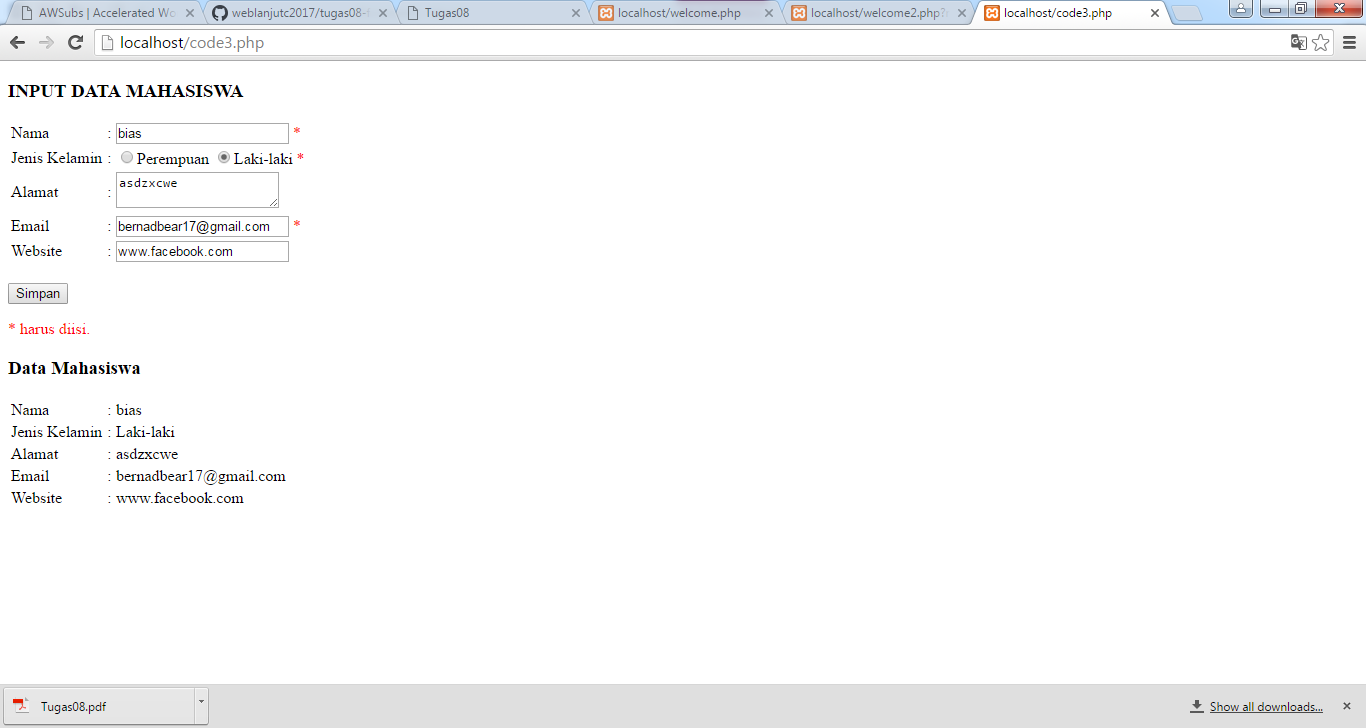
Ini juga merupakan kode untuk menjalankan fungsi GET pada welcome2.php ketika tombol simpan dijalankan

Variable yang terdapat pada welcome2.php antara lain:

* echo $\_GET["nama"] : yang berfungsi menampilkan langsung nilai dari form bervalue nama.
* echo $\_GET["gender"] : yang berfungsi menampilkan langsung nilai dari form bervalue gender.
* echo $\_GET["alamat"] : yang berfungsi menampilkan langsung nilai dari form bervalue alamat.
* echo $\_GET["email"] : yang berfungsi menampilkan langsung nilai dari form bervalue email.
* echo $\_GET["website"] : yang berfungsi menampilkan langsung nilai dari form bervalue website.

1. Code 3



setelah diiikan data dan klik simpan.

Pada code 3, antara form dan penampilnya dijadikan satu atau digabungkan antara form dan phpnya. Code 3 menggunakan function Post yang sama dengan code 1. Hanya saja pada code 3 ditambahkan method pengecekan setiap formnya.

Kode kode yang terdapat pada code 3:

* if ($\_SERVER["REQUEST\_METHOD"] == "POST") {}   
  digunakan untuk melakukan pengecekan ketika fungsi Post dijalankan atau ketika tombol simpan di klik.
* if (empty($\_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";

}

}  
kode untuk melakukan pengecekan pada nama, ketika nama tidak diisi maka akan keluar kalimat “nama harus diisi”. Kemudian, ketika nama yang dimasukkan bukan merupakan huruf dan spasi, maka akan keluar kalimat “inputan salah. Hanya dapat berupa huruf dan spasi”

* if (empty($\_POST["gender"])) {

$genderErr = "Jenis Kelamin harus diisi";

} else {

$gender = test\_input($\_POST["gender"]);

}

Merupakan pengecekan pada form gender, jika radio pada gender tidak dipilih maka akan keluar kalimat “Jenis kelamin harus diisi”. Jika telah diisi dengan benar maka tidak keluar kalimat tersebut.

* if (empty($\_POST["alamat"])) {

$alamat = "";

} else {

$alamat = test\_input($\_POST["alamat"]);

}

Pada kode ini, alamat hanya akan dicheck apakah diisi atau tidak, jika tidak maka keluarkan kosong jika diisi maka keluarkan alamatnya.

* 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";

}

}

Pengecekan dilakukan pada email. Jika kosong maka akan keluar kalimat”email harus diisi”. Kemudian jika diisi, akan dilakukan pengecekan apakah isian tersebut merupakan email atau bukan.

* 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):\/\/|www\.)[-a-z0-9+&@#\/%?=~\_|!:,.;]\*[-az0-9+&@#\/%=~\_|]/i",$website))

{

$websiteErr = "URL salah";

}

}

Pada pengecekan website, akan dicek apakah website kosong atau diisi. Ketika kosong maka kolom website akan kosong. Ketika telah diisi maka akan dicek apakah format email telah sesuai atau belum, ketika salah maka akan keluar kalimat “URL salah”

* function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

Digunakan untuk memeriksa input data.

1. Perbedaan get dan post

|  |  |
| --- | --- |
| POST | GET |
| Tidak ditampilkan di URL | Ditampilkan di URL |
| Kodenya $\_POST | Kodenya $\_GET |
| Tidak terbatas karakternya | Terbatas 2047 karakter |