



# **PEMROGRAMAN WEB #3**

## **METODE PENGIRIMAN VARIABEL DI PHP**

Sufajar Butsianto, M.Kom

Rev.00

# PENGIRIMAN VARIABEL

---

- PHP merupakan pemrograman berbasis web. PHP banyak digunakan untuk membuat aplikasi berbasis web seperti website, dimana setiap aplikasi dapat terdiri dari beberapa halaman yang saling terhubung. Dan memang seperti itulah karakteristik dasar dari web, keterhubungan antara satu halaman dengan halaman yang lain. Kondisi tersebut membuat PHP perlu menyediakan suatu metode pengiriman variabel antar halaman, karena seringkali suatu variabel diperlukan di beberapa halaman sekaligus.

## PENGIRIMAN VARIABEL

---

- ❑ Terdapat 4 Metode pengiriman variabel antar halaman di PHP. Keempat metode tersebut yaitu metode **POST, GET, SESSION dan COOKIES**. Beberapa contoh program sederhana terkait dengan keempat metode tersebut juga akan diberikan agar dapat memberi gambaran yang lebih jelas.

# Metode POST

---

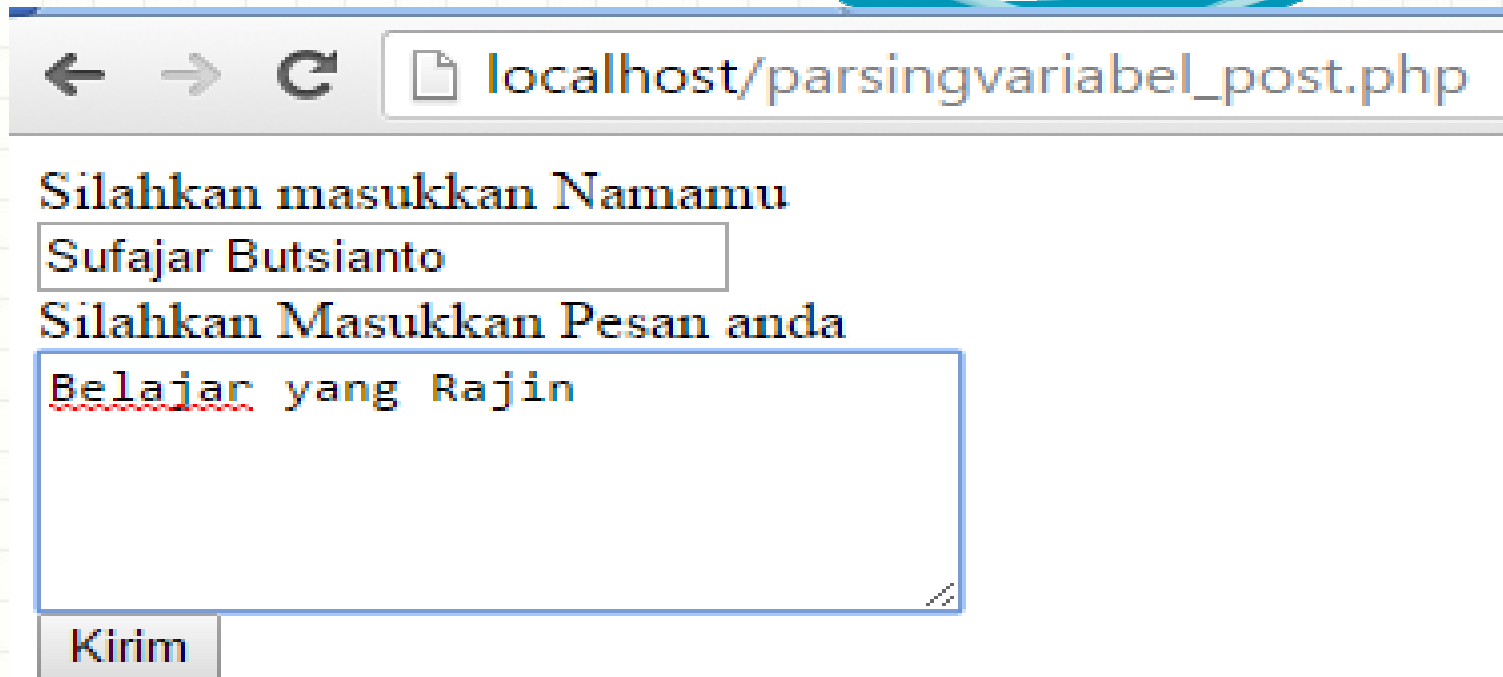
Metode **pengiriman variabel** dengan POST pada dasarnya digunakan dalam pengiriman variabel yang berasal dari sebuah form (inputan). Melalui atribut action yang terdapat dalam tag `<form>` kita dapat mengatur dimana data inputan akan diproses. Misalnya ditentukan bahwa form akan diproses di halaman “proses.php” maka seluruh data inputan dapat diambil dan diproses di halaman “proses.php” tersebut. Cara pengambilan data dapat menggunakan variabel array **\$ POST** dengan index berupa nama inputan yang akan diambil.

# Metode POST

**Program 1** (Nama File : parsingvariabel\_post.php, Program menampilkan form inputan dengan method POST)

```
<HTML>
<HEAD>
  <TITLE> Parsing variabel </TITLE>
</HEAD>
<BODY>
  <form action="hasilvariabel_post.php" method="POST">
    Silahkan masukkan Namamu <BR>
    <INPUT TYPE="text" NAME="nama"><BR>
    Silahkan Masukkan Pesan anda <BR>
    <TEXTAREA NAME="pesan" ROWS="5" COLS="30"></TEXTAREA><BR>
    <INPUT TYPE="submit" value="Kirim">
  </FORM>
</BODY>
</HTML>
```

# Metode POST



← → ↻ 📄 localhost/parsingvariabel\_post.php

Silahkan masukkan Namamu

Silahkan Masukkan Pesan anda

Kirim

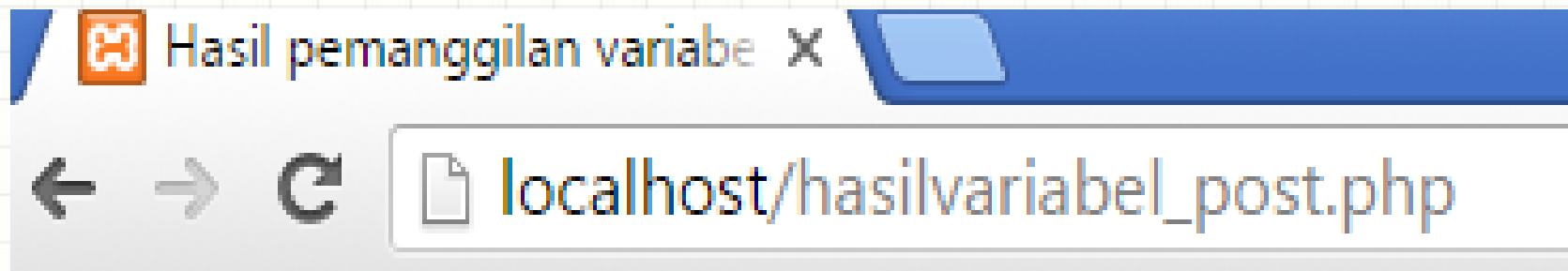
# Metode POST

**Program 2** (Nama File : hasilvariabel\_post.php, Program penanganan form inputan untuk Program 1)

```
<HTML>
<HEAD>
<TITLE> Hasil pemanggilan variabel </TITLE>
</HEAD>
<BODY>
<?php
echo "Data yang anda masukkan : <BR>";
echo "Nama Anda : $_POST[nama] <BR>";
echo "Pesan Anda : $_POST[pesan] <BR>";
?>
</BODY>
</HTML>
```

# Metode POST

**Program 2** (Nama File : hasilvariabel\_post.php, Program penanganan form inputan untuk Program 1)



Data yang anda masukkan :

Nama Anda : Sufajar Butsianto

Pesan Anda : Belajar yang Rajin



# Metode GET

---

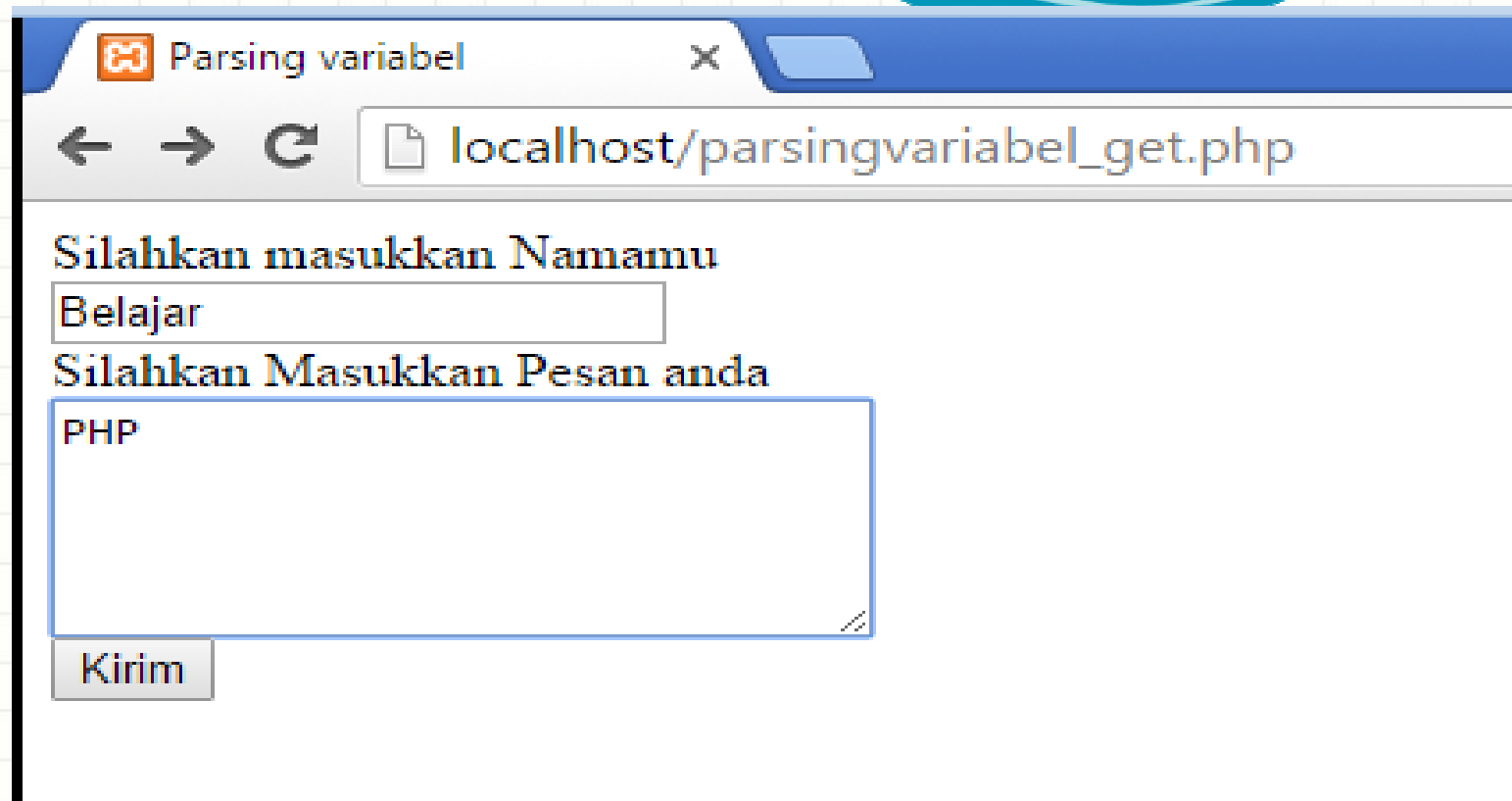
Pada **metode pengiriman variabel** antar halaman menggunakan **GET**, data dapat dikirimkan melalui 2 (dua) cara yaitu melalui form inputan dan melalui parameter URL. Jika dikirimkan melalui form inputan, maka variabel yang akan dikirimkan diwujudkan dalam bentuk inputan (text, checkbox, radio, textarea dsb). Namun jika dikirimkan melalui parameter URL, maka variabel yang dikirimkan beserta value (nilai)nya diletakkan di alamat URL. Alamat URL sendiri dapat diwujudkan dalam bentuk link dan jika diklik maka URL akan ditampilkan di address bar browser. Parameter URL yang dikirimkan juga akan terlihat di address bar. Oleh karena itu, metode ini kurang cocok untuk pengiriman data yang sifatnya rahasia, seperti dalam proses login.

# Metode GET

**Program 1** (Nama File : parsingvariabel\_get.php, Program menampilkan form inputan dengan method POST)

```
<HTML>
<HEAD>
<TITLE> Parsing variabel </TITLE>
</HEAD>
<BODY>
<form action="hasil_variabelGet.php" method="Get">
Silahkan masukkan Namamu <BR>
<INPUT TYPE="text" NAME="nama"><BR>
Silahkan Masukkan Pesan anda <BR>
<TEXTAREA NAME="pesan" ROWS="5" COLS="30"></TEXTAREA><BR>
<INPUT TYPE="submit" value="Kirim">
</FORM>
</BODY>
</HTML>
```

# Method GET



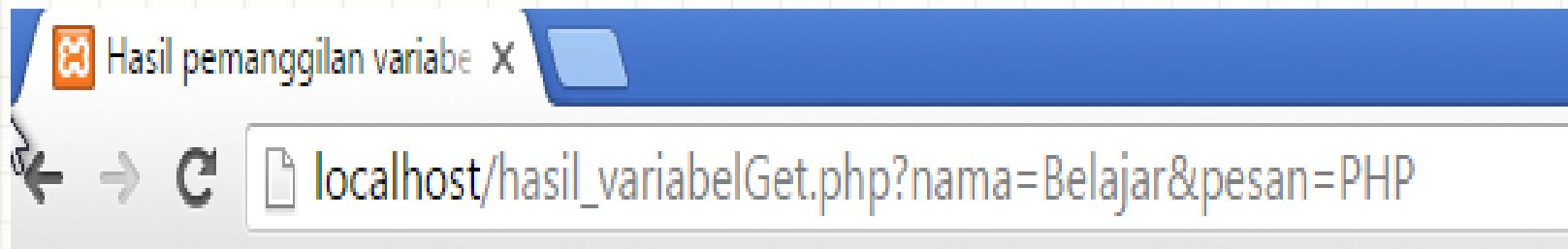
# Method *GET*

**Program 2** (Nama File : hasil\_variabelGet.php, Program penanganan form inputan untuk Program 1)

```
<HTML>
<HEAD>
  <TITLE> Hasil pemanggilan variabel </TITLE>
</HEAD>
<BODY>
  <?php
    echo "Data yang anda masukkan : <BR>";
    echo "Nama Anda : $_GET[nama] <BR>";
    echo "Pesan Anda : $_GET[pesan] <BR>";
  ?>
</BODY>
</HTML>
```

# Method *GET*

**Program 2** (Nama File : hasil\_variabelGet.php, Program penanganan form inputan untuk Program 1)



Data yang anda masukkan :

Nama Anda : Belajar

Pesan Anda : PHP

# Metode SESSION

---

**Session** merupakan hal yang cukup penting dalam aplikasi berbasis web. Dengan session memungkinkan programmer menyimpan informasi user secara semi-permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di server, jadi relatif lebih aman karena tidak dapat dimanipulasi secara langsung oleh client.

Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login.

Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman. Misalnya saja, untuk melihat halaman kotak surat pada email, kita harus login terlebih dahulu. Dalam proses login antara lain akan terjadi pembuatan suatu session yang akan dibawa oleh user di setiap halaman

# Metode SESSION

---

Dihalaman kotak surat, session tersebut akan diperiksa. Jika session benar maka user dipersilahkan membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta untuk login terlebih dahulu. Itulah sebabnya, user tidak bisa mengakses halaman kotak surat secara langsung tanpa melakukan login.

Selain untuk autentifikasi login, session juga dapat digunakan untuk mengirimkan suatu nilai (data) antar halaman. Caranya cukup mudah. Pada halaman pertama, ciptakan sebuah variabel session dan isikan dengan nilai yang akan dikirimkan. Selanjutnya pada halaman kedua, isi variabel session dapat diambil dengan predefined variable **\$\_SESSION**.

# Method Session

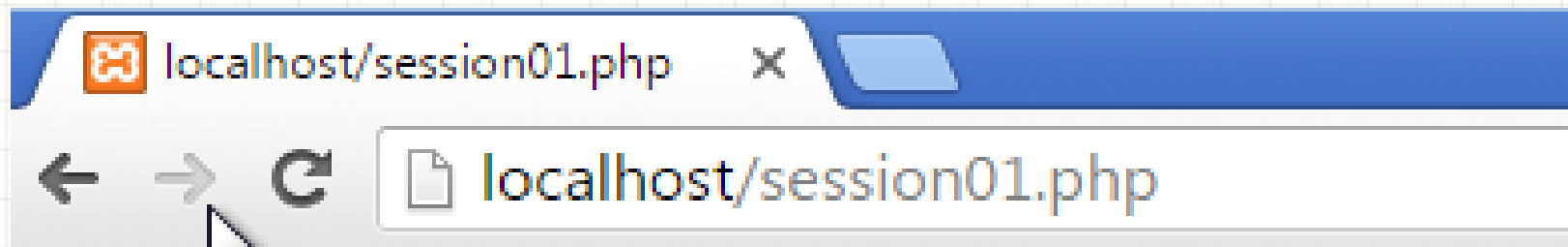
(Nama File : session01.php, Program mengirimkan nilai variabel melalui session)

```
<?php
    /*****
    Nama file : session01.php
    Halaman ini merupakan halaman contoh penciptaan session.
    Perintah session_start() harus ditaruh di perintah pertama
    tanpa spasi di depannya. Perintah session_start() harus ada
    pada setiap halaman yang berhubungan dengan session
    *****/
    session_start();

    $_SESSION['nama'] = "Sufajar Butsianto";
    echo "<a href='session02.php'>Menuju ke halaman kedua</a>";
?>
```



## Method Session



Menuju ke halaman kedua

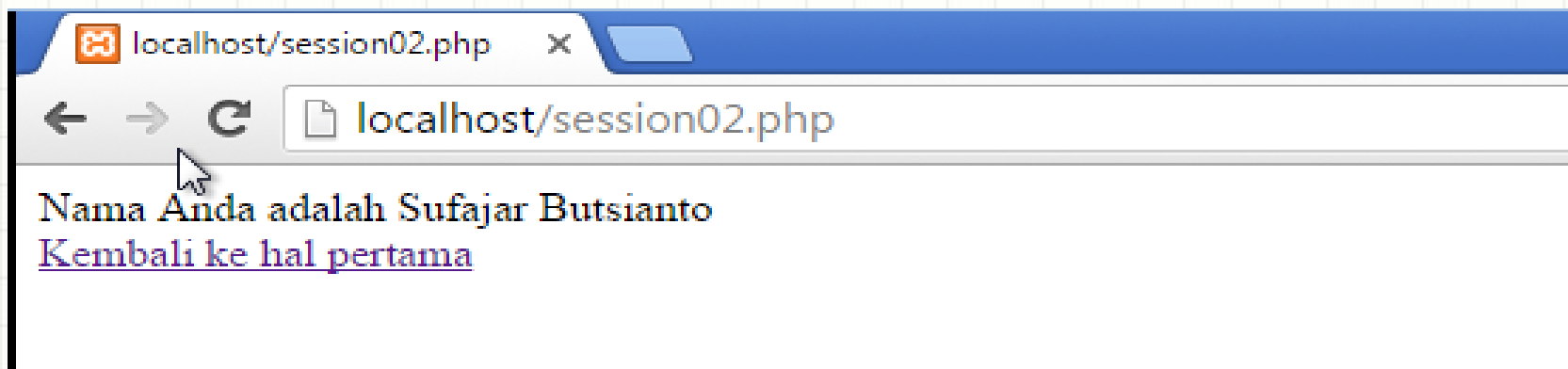
# Method Session

(Nama File : session0.php, Program mengirimkan nilai variabel melalui session)

```
<?php
    session_start();
    echo "Nama Anda adalah " . $_SESSION['nama'];
    echo "<br/><a href='session01.php'>Kembali ke hal pertama</a>";
?>
```

# Method Session

(Nama File : session02.php, Program mengambil nilai yang disimpan di session)



## Method Cookies

Seperti halnya session, cookies juga merupakan sebuah konsep penyimpanan informasi user. Hanya saja, jika session tempat penyimpanan berada di server, cookies berada di client. Oleh karena itu, konsep cookies sebaiknya jangan digunakan untuk menyimpan informasi login user seperti username, password dsb. Selain user bisa melihat informasi yang disimpan, user juga bisa men-disable cookies itu sendiri. Jika cookies di-disable, maka program yang memanfaatkan cookies tentunya tidak akan berjalan dengan baik. Cookies, salah satunya biasa digunakan untuk menyimpan sementara produk-produk yang dipilih oleh pengunjung pada saat berbelanja di sebuah situs toko online.

Karena cookies dapat digunakan untuk menyimpan suatu nilai, maka cookies dapat digunakan untuk mengirimkan variabel antara halaman satu dengan yang lainnya. Nilai yang akan dikirimkan cukup disimpan di cookies, maka setiap halaman dapat mengaksesnya dengan menggunakan predefined variable \$ COOKIE.

# Metode Session

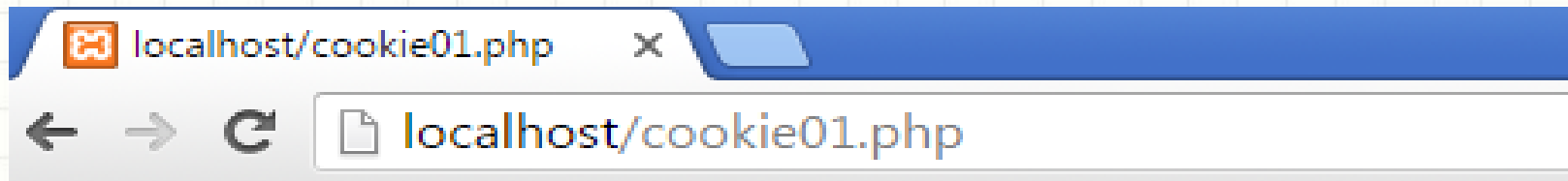
(Nama File : cookie01.php, Program membuat cookie)

```
<?php
$value = 'butsiant';
$value2 = 'Sufajar Butsianto';

setcookie("username", $value);
setcookie("namalengkap", $value2, time()+3600);/* expire in 1 hour */

echo "<h1>Ini halaman pengesetan cookie</h1>";
echo "<h2>Klik <a href='cookie02.php'>di sini</a> untuk pemeriksaan cookies</h2>";
?>
```

## Method Session



**Ini halaman pengesetan cookie**

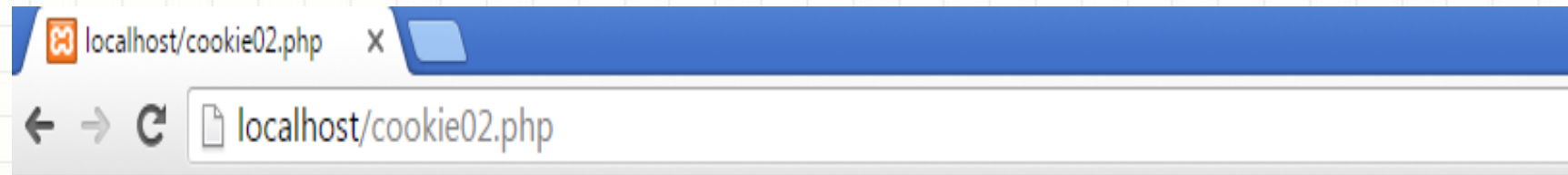
**Klik [di sini](#) untuk pemeriksaan cookies**

# Method Session

(Nama File : cookie02.php, Program membuat cookie)

```
<?php
    if(isset($_COOKIE['username'])) {
        echo "<h1>Cookie 'username' ada. Isinya : " . $_COOKIE['username'];
    } else {
        echo "<h1>Cookie 'username' TIDAK ada.</h1>";
    }
    if(isset($_COOKIE['namalengkap'])) {
        echo "<h1>Cookie 'namalengkap' ada. Isinya : " . $_COOKIE['namalengkap'];
    } else {
        echo "<h1>Cookie 'namalengkap' TIDAK ada.</h1>";
    }
    echo "<h2>Klik <a href='cookie01.php'>di sini</a> untuk penciptaan cookies</h2>";
?>
```

## Method Session



**Cookie 'username' ada. Isinya : butsiant**

**Cookie 'namalengkap' ada. Isinya : Sufajar Butsianto**

**Klik [di sini](#) untuk penciptaan cookies**



## *Kesimpulan Metode Session, Cookie, Get, Post*

---

Dari keempat metode tersebut, mana yang sebaiknya kita gunakan?

Jawabnya yang sesuai kebutuhan. Misalnya dilihat dari data yang akan dikirimkan, untuk data yang tidak terlalu “penting” maka cukup dengan menggunakan metode GET yang dikirimkan melalui parameter URL. Tapi pada kasus lain, jika datanya cukup “penting” dan cukup banyak, maka sebaiknya menggunakan metode POST.



**TERIMA KASIH**