

# Struktur Dasar PHP

# PHP dasar (sintaks / aturan penulisan kode program

## Sebelum Belajar PHP

1. Diasumsikan sudah menginstall dan telah mengerti bagaimana menggunakan salah satu editor seperti Notepad++, Visual studio code.
2. Diasumsikan bahwa Apache, PHP, dan MySQL (Xampp) di komputer yang digunakan sudah berjalan sebagaimana mestinya.
3. diasumsikan telah mengerti berbagai macam jenis operator dan cara penggunaannya.
4. Diasumsikan telah mengenal dan mengerti dengan cukup baik tentang tag-tag HTML dan juga bagaimana menggunakannya.
5. Diasumsikan bahwa Anda bisa membedakan antara :
  - Tulisan nama, Nama dan NAMA
  - Koma (,), titik-koma (;) dan titik-dua (:)
  - Kutip-tunggal (') dan kutip-ganda ("")
  - Tulisan namamhs, nama\_mhs dan nama mhs

PHP itu.....???

1. Merupakan singkatan recursive dari PHP : ***Hypertext Preprocessor***
2. Pertama kali dibuat oleh **Rasmus Lerdorf** pada tahun **1994**.
3. Memulai PHP harus ditulis di antara tag :
  - ☐ `<? dan ?>`
  - ☐ `<?php dan ?>`
  - ☐ `<script language="php"> dan </script>`
  - ☐ `<% dan %>`
4. Setiap satu statement (perintah) biasanya diakhiri dengan titik-koma (;).
5. CASE SENSITIVE untuk nama *identifier* yang dibuat oleh user (berupa variable, konstanta, fungsi dll), namun TIDAK CASE SENSITIVE untuk identifier built-in dari PHP. Jadi :
  - `$nama ≠ $Nama ≠ $NAMA`
  - `echo = ECHO`
  - `while = WHILE`

**Sampai disini, sudahkah bisa dipahami??**

***Kalau sudah silahkan dilanjutkan ke slide berikutnya.***

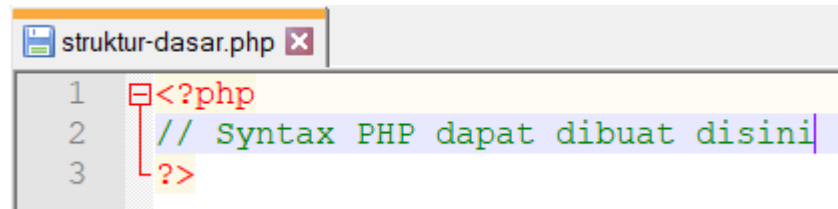
***Jika belum silahkan kembali lagi ke slide awal***

# Struktur Dasar PHP

`<?php`

`// Syntax PHP dapat dibuat disini`

`?>`

A screenshot of a code editor window titled 'struktur-dasar.php'. The editor shows three lines of code: line 1 contains '<?php', line 2 contains '// Syntax PHP dapat dibuat disini', and line 3 contains '?>'. The code is color-coded: the opening tag is red, the comment is green, and the closing tag is red. A red bracket on the left side of the editor groups the three lines together.

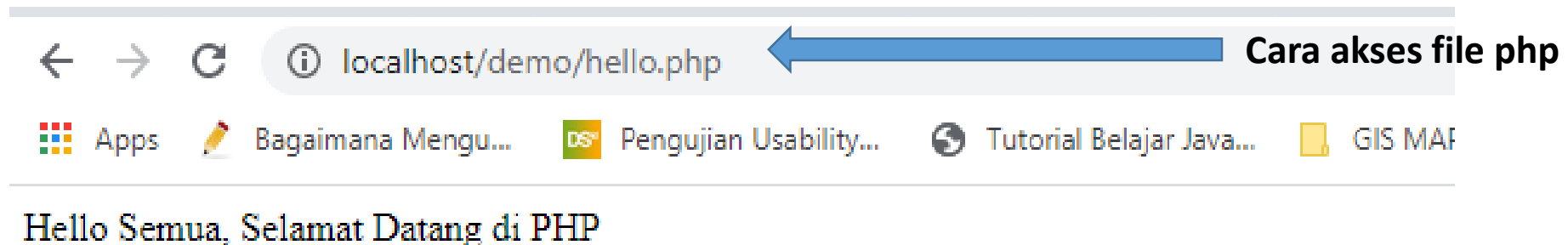
```
1 <?php
2 // Syntax PHP dapat dibuat disini
3 ?>
```

# Contoh Program PHP

Ketikkan perintah di bawah ini di text editor :

```
<?php  
echo "Hello Semua, Selamat Datang di PHP ";  
?>
```

1. Simpan dengan nama “**hello.php**” di local server anda.
2. Buka browser dan ketikkan :  
localhost/nama\_folder\_anda/nama\_file.
3. hasilnya di browser akan terlihat seperti berikut :



# Variabel

1. Digunakan untuk menyimpan sebuah value (nilai), data atau informasi
2. Nama variabel selalu diawali dengan tanda/symbol \$
3. Panjang tidak terbatas
4. Setelah tanda/symbol \$ diawali oleh huruf atau under-score (\_). Karakter berikutnya bisa terdiri dari huruf, angka, dan karakter tertentu yang diperbolehkan (karakter ASCII dari 127 – 255).
5. Bersifat case-sensitive.
6. Tidak perlu dideklarasikan.
7. Tidak boleh mengandung spasi.

# Contoh :

Penulisan variable yang **Benar** :

- `$_name`
- `$first_name`
- `$name3`
- `$name_3`

Penulisan variable yang **Salah** :

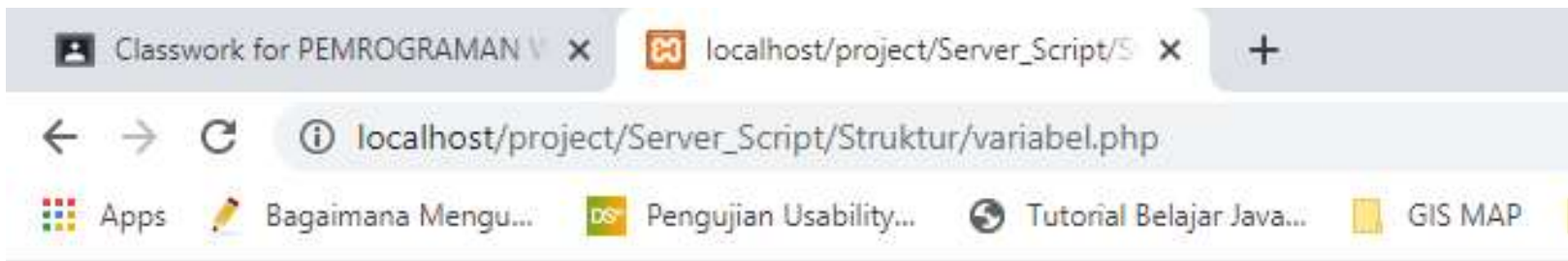
- *`$3name`*
- *`$name?`*
- *`$first+name`*
- *`$first.name`*
- *`$first name`*



# Contoh Program dengan Variabel

```
<?php
$nim = "1234567890";
$matakuliah = 'Pemrograman Web Server Side';
echo "NIM : " . $nim . "<br>";
echo "Mata Kuliah Anda : $matakuliah";
?>
```

Hasil dari program diatas adalah,



NIM : 1234567890  
Mata Kuliah Anda : Pemrograman Web Server Side

# Penanganan request (POST dan GET) dari inputan Form HTML

## **Apa yang dimaksud dengan Method?**

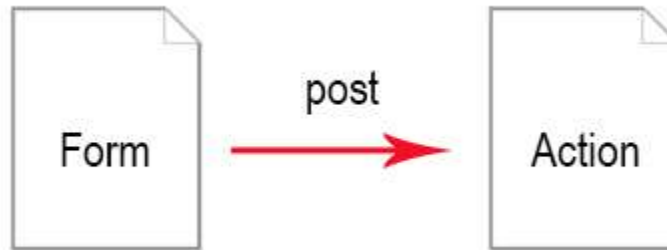
### **Perbedaan method POST dan GET.**

Method merupakan metode atau cara yang digunakan untuk mengirimkan data atau nilai yang diinputkan dari form ke halaman lain untuk diproses.

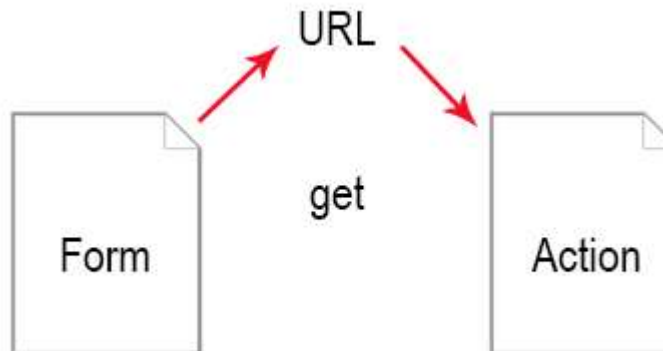
Ada 2 Method dalam pemrograman PHP yang dapat digunakan pada form, yakni method POST dan GET.

## Perbedaan method POST dan GET.

1. Method POST akan mengirimkan data atau nilai langsung ke action untuk ditampung, tanpa menampilkan pada URL.



Sedangkan method GET akan menampilkan data/nilai pada URL, kemudian akan ditampung oleh action.



*lanjutan*


2. Method POST data yang dikirim tidak terbatas. Sedangkan method GET tidak boleh lebih dari 2047 karakter.

## Contoh Program Method Get

Buatlah form dengan script berikut,

```
<html>
<body>
<form method="get" action="tujuan.php">
Nama: <input type="text" name="nama"><br>
Mata Kuliah: <input type="text" name="mk"><br>
<input type="submit" value="Kirim">
</form>
</body>
</html>
```

Menjadi file yang dituju dari data yang  
Diinput dari form



**Kemudian simpan script diatas menjadi file PHP (form.php)**

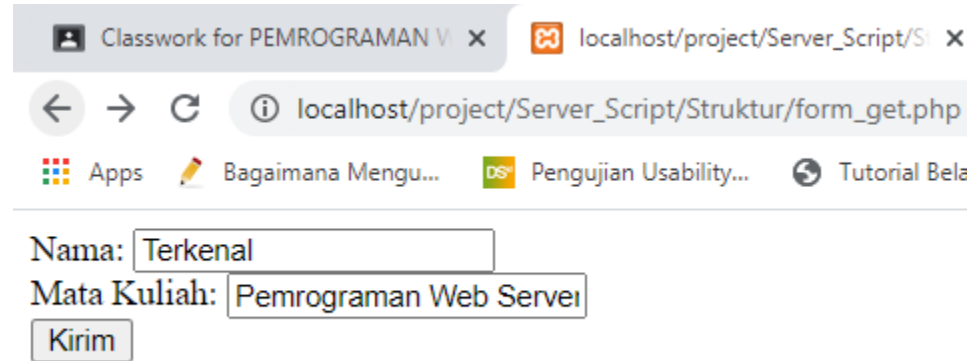
# File “tujuan.php”

```
<?php  
echo "Selamat Datang, $_GET[nama]<br>  
Mata Kuliah Anda adalah : $_GET[mk]";  
?>
```

Simpan script diatas dengan nama file : **tujuan.php**

# Hasil program dengan method Get

## Program Form Input



Classwork for PEMROGRAMAN V x localhost/project/Server\_Script/S x

localhost/project/Server\_Script/Struktur/form\_get.php

Apps Bagaimana Mengu... Pengujian Usability... Tutorial Bela

Nama:

Mata Kuliah:

## Program tujuan.php



Classwork for PEMROGRAMAN V x localhost/project/Server\_Script/S x

localhost/project/Server\_Script/Struktur/tujuan.php?nama

Apps Bagaimana Mengu... Pengujian Usability... Tutorial Belajar J

Selamat Datang, Terkenal

Mata Kuliah Anda adalah : Pemrograman Web Server Side

Lakukan hal yang sama untuk method POST, Ganti  
\$\_Get Menjadi \$\_POST

Perhatikan perbedaan method tersebut.....!



# Tugas

Buatlah Program Menginputkan data dari form, adapun data yang diinputkan adalah :

- Nama
- Alamat
- Email
- Jenis Kelamin
- Hobby

Gunakan komponen html sesuaikan dengan kebutuhan data inputan. Kemudian gunakan method post atau get. Hasil inputan data dengan akan ditampilkan dengan meng-klik button “tampil”

***Ingat format pengumpulan file***

# Jadwal Pengumpulan Tugas

1. Dikumpulkan sebelum jadwal pertemuan kuliah minggu berikutnya.
2. Capture script program dan hasil output program.



selesai