

# MODUL 5

## PEMROGRAMAN WEB

### GET DAN POST

#### A. TUJUAN PERKULIAHAN

- Mahasiswa memahami konsep variabel super global pada PHP
- Mahasiswa dapat mengimplementasikan teknik pengiriman data dengan GET
- Mahasiswa dapat mengimplementasikan teknik pengiriman data dengan POST

#### B. ALOKASI WAKTU 3 x 50 menit

#### C. DASAR TEORI

##### 1. VARIABLE SCOPE

Jika dilihat dari ruang lingkupnya, **PHP** memiliki **tiga tingkatan variabel**, yakni **variabel lokal**, **variabel global** dan **variabel super global**.

##### a. Variabel Lokal

Variabel lokal adalah variabel yang hanya dikenal di dalam suatu fungsi tertentu, dengan kata lain, variabel lokal tidak akan dikenal oleh fungsi lain maupun kode di luar dari fungsi yang telah dibuat.

```
<?php
$a=1;
$b=3;
function hasil(){
    $a=2;
    $b=4;
    $hasil=$a+$b;
    return $hasil;
}

#memanggil variabel lokal dari luar fungsi
echo "Hasil penjumlahan = " . hasil();
?>
```

Hasil dari program di atas adalah

Hasil penjumlahan = 6
-----------------------

Meskipun pada awal program terdapat variabel \$a dan \$b yang bernilai 1 dan 3, namun perintah eksekusi di dalam fungsi tersebut memiliki variabel \$a dan \$b – nya sendiri yang bernilai 2 dan 4 dan melakukan pemrosesan berdasarkan variabel yang dimilikinya saja. Sehingga dapat disimpulkan bahwa kedua jenis \$a maupun \$b yang berada di dalam dan di luar fungsi pada dasarnya tidak saling terhubung.

## b. Variabel Global

Variabel global adalah variabel yang dapat digunakan atau dipanggil atau dikenali oleh semua fungsi / prosedur / dikenali diseluruh program. Variabel global ini dapat digunakan pada beberapa fungsi / prosedur, Hal ini bertujuan untuk menghemat penulisan, karena tidak perlu lagi berkali - kali menuliskan variabel yang sama pada beberapa fungsi / prosedur.. Jika suatu fungsi akan menerapkan konsep variable global, maka tambahkan kata kunci *global* di depan nama variable tersebut. Sebagai contoh adalah sebagai berikut

```
<?php
$a=1;
$b=3;
function hasil(){
    global $a;
    global $b;
    $hasil=$a+$b;
    return $hasil;
}

#memanggil variabel lokal dari luar fungsi
echo "Hasil penjumlahan = " . hasil();

?>
```

Hasil dari contoh di atas adalah

Hasil penjumlahan = 4
-----------------------

Meskipun deklarasi variabel dan inisialisasinya berada di luar fungsi, namun fungsi hasil() memanggil variabel \$a dan \$b dengan menggunakan keyword global. Hal ini akan membuat variabel \$a dan \$b memiliki nilai yang sama dengan \$a dan \$b

yang ada di luar fungsi. Cara lain penulisan global juga dapat menggunakan array, dengan contoh seperti berikut

```
$a=$GLOBALS['a'];  
$b=$GLOBALS['b'];
```

### c. Variabel Super Global

Variabel super global yakni variabel bawaan yang bersifat global. Variabel bawaan yang dimaksud adalah: variabel yang sudah otomatis ada tanpa perlu kita definisikan sendiri. Dan ia bersifat global dalam artian bisa kita akses dari mana pun dan kapan pun.

Variabel super global menyimpan banyak sekali data penting dan juga bermanfaat yang dapat Anda gunakan dalam menyelesaikan proyek yang sedang Anda kerjakan. Setidaknya terdapat 9 variabel super global di PHP.

	Variable	Definition
1	GLOBALS	References all variables available in global scope
2	_SERVER	Server and execution environment information
3	_GET	HTTP GET variables
4	_POST	HTTP POST variables
5	_FILES	HTTP File upload variables
6	_COOKIE	HTTP Cookies
7	_SESSION	Session variables
8	_REQUEST	HTTP Request variables
9	_ENV	Environment variables

Variabel super global dapat dipanggil di manapun tanpa harus didefinisikan terlebih dahulu. Nilai dari variabel super global sangat bervariasi tergantung lokasi dan personalisasi aplikasi dari user. Fokus utama penggunaan variabel super global ini dapat dibagi menjadi tiga, yakni:

- Pemanggilan sesi aplikasi dan user
- Lingkungan pengembangan aplikasi dari user
- Lingkungan pengembangan aplikasi dari server

## 2. METODE GET

Browser web berkomunikasi dengan server biasanya menggunakan salah satu dari dua metode HTTP (Hypertext Transfer Protocol) – GET dan POST. Dalam bahasa Inggris kita akrab dengan istilah GETting, dari istilah tersebut dapat diartikan bahwa metode *GET* pada HTTP ditujukan untuk mengambil (*get*) data dari server.

Pada metode ini umumnya data berbentuk query string yang dikirim via **URL**, data tersebut berupa pasangan **key=value** yang dipisahkan dengan tanda **&**. Data tersebut digabung dengan url utama yang dipisahkan dengan tanda **?**.

Sebelum dikirim, terlebih dahulu data diproses sehingga memenuhi standar format URL. URL hanya boleh memuat huruf (besar dan kecil), angka, dan beberapa karakter lain dalam ASCII Character Set seperti (“.\_~”), karakter di luar itu akan diubah ke format tertentu yang diawali tanda % kemudian diikuti dengan 2 digit hexadesimal, beberapa contohnya adalah seperti berikut

Karakter	URL Encoded
?	%3F
@	%40
=	%3D
<spasi>	%20

Angka pada kolom (**URL Encoded**) merupakan nilai hexadecimal dari character ASCII, disamping itu URL juga tidak boleh memuat spasi, sehingga spasi akan diubah menjadi tanda + atau %20. Semua proses tersebut disebut **url encoding**. Maka salah satu contoh URL yang menerapkan metode GET adalah seperti berikut

<a href="http://ptik.fkip.uns.ac.id/dosen.php?nama=nurcahya%20pradana&amp;nip=199109242019031015">http://ptik.fkip.uns.ac.id/dosen.php?nama=nurcahya%20pradana&amp;nip=199109242019031015</a>
---

**Analisa :** data dikirim ke **domain** **http://ptik.fkip.uns.ac.id/dosen.php** dengan dua parameter GET, yakni **nama** yang berisi “**nurcahya pradana**” dan **nip** yang berisi “**199103242019031015**”

Sekarang perhatikan contoh kode berikut

```
<html>

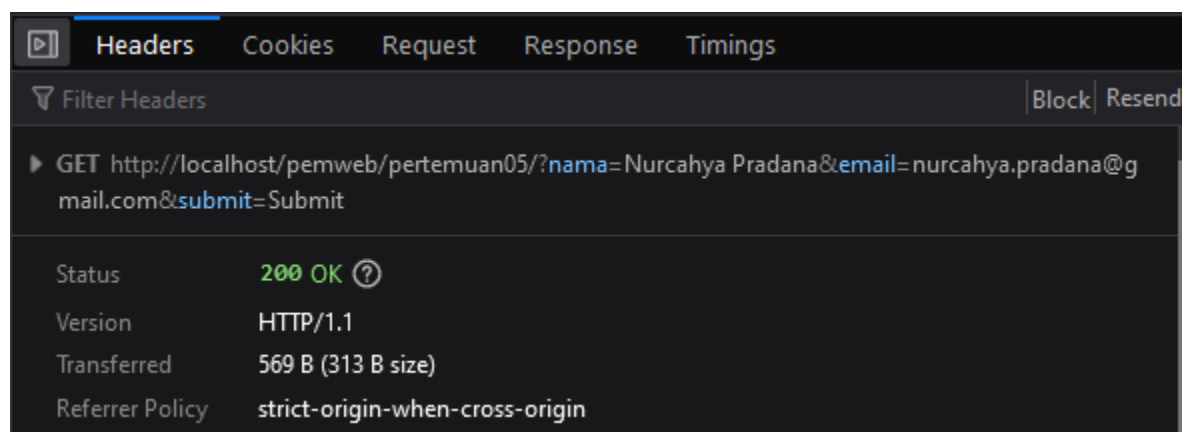
<body>
  <form method="GET" action="">
    Nama: <input type="text" name="nama"><br>
    Email: <input type="text" name="email"><br>
    <input type="submit" name="submit" value="Submit">
  </form>

  <?php
  if ($_GET) {
    echo 'Nama: ' . $_GET['nama'];
    echo '<br>';
    echo 'Email: ' . $_GET['email'];
  }
  ?>
</body>

</html>
```

Kode di atas akan menghasilkan sebuah form yang berisi input berupa nama dan email sederhana. Coba masukkan nama = “Nurchaya Pradana” dan email = “nurchaya.pradana@gmail.com” lalu tekan submit.

Selanjutnya perhatikan address bar pada browser atau inspect element, lalu masuk ke tab Network.



Dari gambar diatas terlihat bahwa metode yang kita gunakan adalah GET disertai query string nya, disamping itu url pada browser juga akan berubah menjadi:

```
http://localhost/pemweb/pertemuan05/?nama=Nurcahya+Pradana&email=nurcahya.pradana%40gmail.com&submit=Submit
```

URL di atas menunjukkan bahwa <spasi> baik pada query string maupun pada url diencode menjadi tanda + dan @ menjadi %40. Ketika kita ambil data tersebut dengan PHP maka otomatis data pada url akan didecode sehingga kita dapatkan bentuk asli dari data yang kita kirim. Hasil yang diperoleh dari kode di atas adalah

```
Nama: Nurcahya Pradana
Email: nurcahya.pradana@gmail.com
```

Variabel \$\_GET pada PHP berbentuk associative array. Variabel ini bentuknya sama seperti variabel pada umumnya, bedanya \$\_GET ini merupakan variabel super global sehingga bisa diakses dimana saja.

Karena bentuknya sama dengan yang lain, variabel ini dapat kita manipulasi sebagaimana kita memanipulasi variabel array lainnya, misal dengan menambahkan nilainya: \$\_GET[nama] = "Randy Orton" atau menghapusnya unset(\$\_GET['nama']).

### **Kelebihan dan Kekurangan**

Terdapat beberapa kelebihan penggunaan metode GET, diantaranya adalah:

1. Sempel, dan data mudah diedit, misal untuk menuju halaman 5 dari suatu website, kita tinggal mengganti urlnya.
2. Halaman dapat dibookmark dan disimpan pada history browser sehingga mudah untuk diakses kembali.
3. Dapat kembali ke halaman sebelumnya dengan mudah (dengan mengklik tombol Back pada browser).
4. Dapat direfresh dengan mudah.
5. Dapat didistribusikan hanya dengan copy-paste.

Meskipun banyak kelebihanannya, penggunaan metode ini memiliki beberapa kelemahan yaitu:

1. Panjang data terbatas hanya 2kb – 8kb (tergantung browsernya), jika melebihi batas tersebut akan muncul pesan error 414 Request-URI Too Long, sehingga tidak dapat digunakan untuk mengirim data dalam jumlah besar.
2. Hanya dapat mengirim data jenis teks, jenis lainnya seperti: gambar, file zip, dll tidak dapat dikirim.
3. Karena data dikirim via URL, data tersebut mudah terekspose dan direkayasa.

### **3. METODE POST**

Metode POST digunakan untuk mengirim data yang biasanya digunakan untuk menambah/merubah data pada server. Pada protokol HTTP, metode POST dikirim baik melalui query string maupun body. seperti pada GET, data yang dikirim melalui query string akan ditampilkan pada URL dan sedangkan yang dikirim melalui body tidak terlihat oleh user.

Pada PHP, data POST bisa juga dikirim melalui query string disimpan pada variabel `$_GET` (seperti metode GET), namun utamanya data yang dikirim melalui body disimpan pada variabel `$_POST`.

Variabel `$_POST` juga berbentuk associative array dan bersifat global yang artinya dapat diakses dimana saja, selain itu juga dapat dilakukan manipulasi sebagaimana variabel array lainnya, sama seperti pada `$_GET` yang kita pelajari di atas.

Penggunaan metode POST sering kita jumpai terutama pada saat pengiriman data menggunakan form HTML. Misal: meneruskan contoh sebelumnya, pada file kode kita ganti method pada bagian form dari GET menjadi POST.

```

<html>

<body>
  <form method="POST" action="">
    Nama: <input type="text" name="nama"><br>
    Email: <input type="text" name="email"><br>
    <input type="submit" name="submit" value="Submit">
  </form>

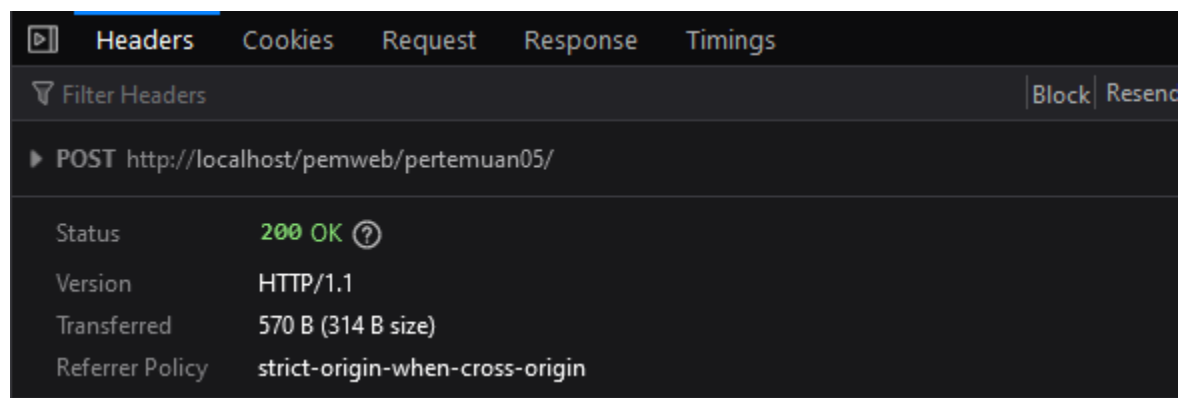
  <?php
  if ($_POST) {
    echo 'Nama: ' . $_POST['nama'];
    echo '<br>';
    echo 'Email: ' . $_POST['email'];
  }
  ?>
</body>

</html>

```

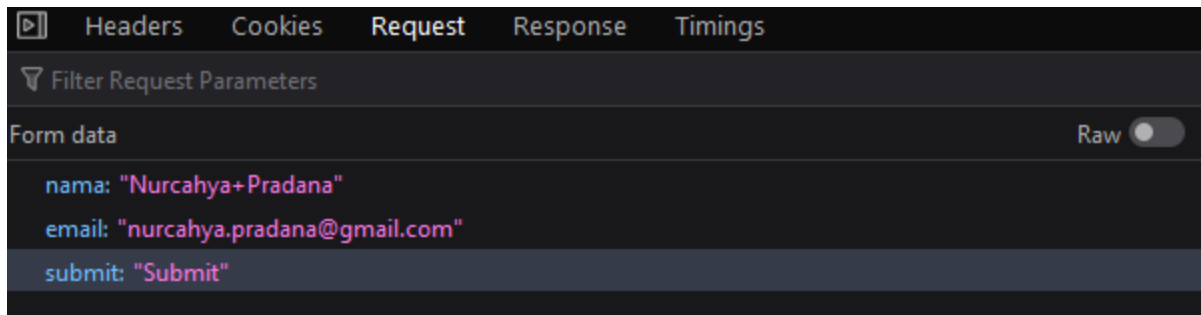
Kode di atas akan menghasilkan sebuah form yang berisi input berupa nama dan email sederhana yang mirip dengan contoh sebelumnya. Coba masukkan nama = “Nurcahya Pradana” dan email = “nurcahya.pradana@gmail.com” lalu tekan submit.

Selanjutnya buka inspect element, lalu masuk ke tab Network. Perhatikan pada bagian **Headers**.



Pada gambar di atas, menunjukkan bahwa query string pada URL yang kita tuju tidak mengalami penambahan data parameter. Selanjutnya pindah ke bagian **Request**.





Disini request data dikirim secara terpisah dalam body. Dengan metode POST, data yang dikirim akan lebih aman karena terhindar dari manipulasi. Hasil yang diperoleh dari kode di atas adalah :

```
Nama: Nurcahya Pradana
Email: nurcahya.pradana@gmail.com
```

Data yang dikirim diencode dengan sistem sama dengan GET (url encode) yang memang secara default cara pengiriman data pada POST adalah menggunakan **application/x-www-form-urlencoded**, disamping itu juga ada **multipart/form-data** yang digunakan untuk pengiriman data berupa file/binary.

### Kelebihan dan Kekurangan

Pengiriman data menggunakan metode POST memiliki beberapa kelebihan diantaranya:

1. Lebih aman dari pada metode GET karena data yang dikirim tidak terlihat, serta parameter yang dikirim tidak disimpan pada history browser/log browser.
2. Dapat mengirim data dalam jumlah besar.
3. Dapat mengirim berbagai jenis data seperti gambar, file, dll, tidak harus teks.

Meskipun terdapat kelebihan, penggunaan metode ini juga memiliki beberapa kelemahan, walaupun sebenarnya bukan kelemahan melainkan memang menjadi karakteristik dari metode ini:

1. Data tidak disimpan pada history browser.
2. Data tidak dapat dibookmark.

3. Karena dianggap sebagai data sensitif, maka ketika kita mererefresh browser, akan muncul konfirmasi pengiriman ulang data, demikian juga ketika kita tekan tombol back

#### 4. KAPAN MENGGUNAKAN GET / POST

GET dan POST merupakan metode yang digunakan protokol HTTP untuk pertukaran data. Pada PHP, data yang dikirim menggunakan metode GET akan disimpan dalam variabel `$_GET`, sedangkan POST akan disimpan pada variabel `$_POST` (untuk data yang dikirim via url disimpan pada variabel `$_GET`).

Masing masing metode memiliki kelebihan dan kekurangan dan memang sebenarnya kedua metode tersebut ditujukan untuk keperluan berbeda, sehingga kita harus tahu kapan menggunakan GET dan kapan menggunakan POST.

Untuk memutuskan apakah kita akan menggunakan metode GET atau POST, kita harus selalu mengajukan pertanyaan: **apakah akan terjadi perubahan pada server?** jika ya, maka kita gunakan POST, jika tidak maka kita gunakan GET.

Contohnya adalah ketika membuat pagination, kita cukup menggunakan GET tidak perlu POST, bisa dibayangkan ketika kita sudah sampai halaman 5 suatu artikel, dan kita mau membookmarknya untuk dibaca di lain waktu, dan ternyata tidak bisa karena halaman tersebut menggunakan POST, dengan terpaksa kita harus membuka halaman 1 lagi! Begitu juga dengan sistem pencarian seperti pada google, bisa dibayangkan jika metode yang digunakan adalah POST.

Contoh lainnya adalah penggunaan ID untuk menghapus data, jangan pernah gunakan method GET! Perhatikan url berikut: **`http://www.contoh.com/delete.php?id=1`**, url ini akan berbahaya jika hal ini benar-benar terjadi. Karena data user akan terhapus dengan begitu mudahnya, lain halnya jika ID tersebut digunakan hanya untuk menampilkan data seperti **`http://www.contoh.com/view.php?id=1`**.

Pertanyaan kedua adalah **apakah ada data sensitif yang dikirim?** jika ya maka kita gunakan POST.

Contohnya adalah pengiriman username dan password ketika login, atau data keuangan seperti kartu kredit. Jika data tersebut dikirim via url maka jelas data tersebut akan terekspose kemana-mana terutama jika ter index oleh google. (Kecuali pada penggunaan AJAX).

singkatnya GET untuk READ data, POST untuk CREATE, UPDATE, DELETE data.

## D. PRAKTIKUM : GET dan POST

### *Kegiatan 1 – Kalkulator Sederhana*

Brock Lesnar ingin membuat sebuah kalkulator sederhana dengan menerapkan **method POST**. Adapun potongan kode halaman HTML yang telah disediakan sebagai berikut:

kalkulator.html

```
<!DOCTYPE html>
<html>
<head>
    <title>.....</title>
</head>
<body>
    <h1>Kalkulator Sederhana</h1>

    <form ..... action="hitung.php">
        Bilangan pertama <input type="text" name=".....">
        <select name="operasi">
            <option>Pilih operator</option>
            <option>+</option>
            <option>-</option>
            <option>x</option>
            <option>/</option>
            <option>^</option>
        </select>
        Bilangan ke dua <input type="text" name=".....">
        <input type="submit" name="submit"
value="Hitung">
    </form>
</body>
</html>
```

Tapi Brock Lesnar lupa untuk membuat file **hitung.php**. Bantulah Brock Lesnar untuk membuat kalkulator sederhana yang dapat berfungsi dengan baik dan benar!

**SIMPAN DALAM RAR / ZIP DENGAN NAMA p0501.rar/zip**

## Kegiatan 2 – Perhitungan Gaji

Stone Cold adalah manajer di bagian HRD salah satu software house terkemuka di kota Solo. Dia ingin membuat aplikasi berbasis web yang dapat menetapkan gaji karyawannya berdasarkan jabatan dan kinerja. Adapun tabel standar yang dimilikinya adalah sebagai berikut

**Tabel jabatan**

Jabatan	Gaji
Junior Programmer	Rp 4.000.000,-
Senior Programmer	Rp 6.000.000,-
Chief Technology Officer	Rp 8.000.000,-
Manager	Rp 10.000.000,-

**Tabel kinerja**

Skor kinerja	Perolehan gaji
< 50	50%
51 – 60	60%
61 – 70	70%
71 – 80	80%
81 – 90	90%
90 – 100	100%

Dan berikut adalah potongan kode yang telah dia buat.

Gaji.html

```
<!DOCTYPE html>
<html>
<head>
    <title>.....</title>
</head>
<body>
    <h1>Kalkulator Sederhana</h1>

    <form ..... action="hitung.php">
        Nama Pegawai : <input type="text" name=".....">
        Jabatan : <select name="jabatan">
            <option>Pilih jabatan</option>
            <option>Junior Programmer</option>
            <option>Senior Programmer</option>
```

```
        <option>Chief Technology Officer</option>
        <option>Manager</option>
    </select>
    Kinerja : <input type="text" name=".....">
    <input type="submit" name="submit" value="Hitung
Gaji">
    </form>
</body>
</html>
```

Bantu Stone Cold untuk membuat file **hitung.php** sehingga dapat melakukan perhitungan gaji bulanan dengan baik dan benar!

**SIMPAN DALAM RAR / ZIP DENGAN NAMA p0502.rar/zip**

# TUGAS TEORI

Diberikan sebuah data dalam format array yang memiliki data ukuran (diameter dan tinggi) dari beberapa buah tabung yang isinya sbb:

- A,10,30
- B,15,34
- C,8.5,27.8
- D,12.8,17.56
- E,13.125,15.7

Keterangan:

A,10,30 -> A adalah nama tabung, 10 adalah ukuran diameternya, 30 adalah tingginya.

Buatlah script PHP **viewtabung.php** untuk mengakomodasi data di atas kemudian munculkan ke browser dalam bentuk tabel sbb:

**DATA UKURAN TABUNG**

NAMA TABUNG	DIAMETER	TINGGI	LUAS
<b>A</b>	10	30	view
<b>B</b>	15	34	view
<b>C</b>	8.5	27.8	view
<b>D</b>	12.8	17.56	view
<b>E</b>	13.125	15.7	view

Selanjutnya buat script **hitungluas.php** untuk menampilkan luasnya dengan ketentuan sbb:

Kolom LUAS pada tabel di atas nantinya berisi text 'view' yang merupakan hyperlink untuk melihat luas setiap tabung, yang diarahkan ke <http://.../hitungluas.php?n=...&d=...&t=...> di mana n adalah parameter untuk nama tabung, d untuk diameternya, dan t untuk tingginya.

Sebagai contoh, misal untuk tabung A, maka link view dari baris tersebut diarahkan ke <http://.../hitungluas.php?n=A&d=10&t=30>. Sehingga ketika link view itu diklik akan memunculkan tampilan sbb:

Luas tabung A dengan diameter 10 dan tinggi 30 adalah ... satuan luas
---

## KETENTUAN :

- Tulis urutan langkah pengerjaan dilengkapi dengan potongan kode dalam **docx/pdf**.
- Tambahkan **screenshot** hasil pekerjaan di dalam dokumen
- Simpan dalam **rar / zip** dengan nama **t0503.rar/zip**





## E. REFERENSI

- <https://www.php.net/>
- <https://jagongoding.com/web/php/menengah/variabel-super-global/>
- <https://jagowebdev.com/memahami-get-dan-post-pada-php-dan-http/>