

MODUL PRAKTIKUM PEMROGRAMAN WEB I

**Jurusan Teknik Informatika
Fakultas Teknik
Universitas Palangka Raya**

Kebutuhan Praktikum

Modul ini akan menjelaskan teknik pemrograman web menggunakan bahasa pemrograman PHP. Diharapkan setiap mahasiswa yang akan menggunakan modul ini sudah memiliki pemahaman dasar tentang:

- HTML
- CSS
- Javascript

Untuk memulai pemrograman PHP, anda harus mempersiapkan server host yang mendukung PHP dan database MySQL.

Dalam menggunakan modul ini, mahasiswa diharapkan mengaplikasikan setiap listing program yang dicontohkan dan memperhatikan hasilnya. Setiap bagian pada modul ini disertakan dengan soal latihan yang harus dikerjakan oleh mahasiswa. Mahasiswa diharapkan mampu mengembangkan kreatifitasnya secara mandiri, merangkai potongan-potongan program yang dicontohkan pada modul ini untuk membuat program web yang baik.

Modul I : PHP (Hypertext Preprocessor)

Tujuan

- Mahasiswa memahami dasar-dasar pemrograman PHP yang biasa digunakan pada umumnya.
- Mahasiswa mampu membuat program sederhana menggunakan bahasa pemrograman PHP.

Script PHP bisa diletakkan di manapun dalam dokumen, dimulai dengan tanda “<?php” dan diakhiri dengan tanda “?>”. Dokumen PHP disimpan dengan ekstensi “.php”, dimana file ini bisa menyimpan tag HTML dan beberapa script PHP. Setiap statement dalam PHP harus diakhiri dengan semicolon “;”. Berikut adalah contoh script PHP untuk menampilkan text “Hello World!”

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Halaman PHP Pertama ku</h1>
    <?php
      echo "Hello World!";
    ?>
  </body>
</html>
```

Mendeklarasikan variabel PHP

Pada pemrograman php, variabel dimulai dengan tanda “\$” dan diikuti dengan nama variabel. Tidak seperti bahasa pemrograman lainnya, php tidak memiliki perintah khusus untuk mendeklarasikan variabel. Tipe data ditentukan ketika nilai dimasukkan kedalam variabel tersebut. Sebuah variabel bisa memiliki nama yang singkat (seperti x dan y) atau nama deskriptif (contoh : umur, namamobil, total_volume). Aturan-aturan untuk variabel php adalah sebagai berikut:

- Sebuah variabel dimulai dengan tanda “\$”, diikuti oleh nama variabel.
- Sebuah nama variabel harus mulai dengan huruf atau karakter garis bawah “_”.
- Sebuah nama variabel tidak boleh dimulai dengan angka.
- Sebuah nama variabel hanya bisa mengandung karakter alpha-numeric dan garis bawah (A-z, 0-9, dan _)

- Nama-nama variabel adalah case-sensitive, yaitu huruf besar dan huruf kecil mempengaruhi perbedaan variabel (\$umur dan \$UMUR adalah dua variabel yang berbeda)

Perintah yang biasa digunakan untuk mencetak isi dari sebuah variabel adalah echo. Berikut adalah contoh program menentukan nilai variabel dan mencetaknya di browser:

```
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
echo "I love " . $txt . "!"; // menghasilkan keluaran yang sama dengan
perintah cetak diatas
?>
```

Lingkup dari sebuah variabel adalah bagian dari skrip dimana variabel bisa dirujuk atau digunakan. Ada tiga lingkup variabel: Lokal, Global dan Static. Variabel yang dideklarasikan diluar fungsi memiliki lingkup global, sedangkan variabel yang dideklarasikan didalam fungsi memiliki lingkup lokal yang hanya bisa diakses didalam fungsi saja. Kelebihan dari penggunaan variabel lokal adalah kita bisa memiliki variabel lokal dengan nama yang sama pada fungsi yang berbeda, karena variabel lokal hanya dikenali oleh fungsi yang mendeklarasikannya saja. Keyword global digunakan untuk mengakses variabel global dari dalam fungsi. Php juga menyimpan semua variabel global dalam sebuah array yang disebut \$GLOBALS[index]. Index adalah nama dari variabel, array ini juga bisa diakses dari dalam fungsi dan bisa digunakan untuk mengupdate variabel global secara langsung. Normalnya, ketika fungsi dieksekusi, semua variabel-variabelnya dihapus. Namun terkadang kita perlu untuk mempertahankan nilai variabel lokal supaya tidak dihapus. Untuk itu, gunakan keyword static dalam pendeklarasian variabel yang nilainya ingin dipertahankan.

```
<?php
$x = 5; // lingkup global
function myTest() {
    //menggunakan x didalam fungsi ini akan menghasilkan error.
    echo "<p>Variabel x di dalam fungsi adalah : $x</p>";
}
myTest();

echo "<p>Variabel x di luar fungsi adalah : $x</p>";
?>
```

```
<?php
    function myTest() {
        $x = 5; // Lingkup lokal
        echo "<p>Variabel x di dalam fungsi adalah : $x</p>";
    }
    myTest();
    // Menggunakan x diluar fungsi akan menghasilkan error
    echo "<p>Variabel x di luar fungsi adalah: $x</p>";
?>
```

```
<?php
    $x = 5;
    $y = 10;

    function myTest() {
        global $x, $y;
        $y = $x + $y;
    }

    myTest();
    echo $y; // output 15
?>
```

```
<?php
    $x = 5;
    $y = 10;

    function myTest() {
        $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
    }

    myTest();
    echo $y; // output 15
?>
```

```
<?php
    function myTest() {
        static $x = 0;
```

```
        echo $x;
        $x++;
    }

    myTest();
    myTest();
    myTest();
?>
```

Echo dan Print

Ada dua cara untuk menampilkan output di PHP yaitu dengan menggunakan perintah echo dan print. Terdapat sedikit perbedaan dalam penggunaan echo dan print, echo tidak mengembalikan nilai sedangkan print mengembalikan nilai 1 sehingga bisa digunakan dalam ekspresi. Echo bisa menggunakan banyak parameter sedangkan print bisa menggunakan satu argument. Secara marjinal echo lebih cepat dari print.

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";

print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

Tipe Data PHP

Variabel bisa menyimpan data dengan tipe yang berbeda-beda, berikut adalah tipe-tipe data yang didukung oleh PHP

- String
- Integer
- Float
- Boolean
- Array
- Object
- NULL
- Resource

String

String adalah susunan dari karakter-karakter seperti "Hello world!". String dituliskan didalam kutip satu maupun kutip dua. Berikut adalah contoh variabel yang menyimpan data string.

```
<?php
    $x = "Hello world!";
    $y = 'Hello world!';
    echo $x;
    echo "<br>";
    echo $y;
?>
```

Integer

Tipe data integer atau bilangan bulat adalah bilangan bukan pecahan antara -2.147.483.648 hingga 2.147.483.647. Aturan-aturan pada integer adalah sebagai berikut:

- integer harus memiliki sedikitnya satu digit.
- Integer tidak boleh memiliki koma.
- Integer bisa positif maupun negatif.
- Integer bisa dispesifikasikan dalam tiga format: desimal berbasis 10, hexadesimal berbasis 16 atau oktal berbasis 8.

Berikut adalah contoh variabel integer. Fungsi `var_dump()` mengembalikan tipe data dan nilainya:

```
<?php
    $x = 5985;
    var_dump($x);
?>
```

Float

Float (bilangan floating-point) adalah bilangan dengan tanda desimal atau bilangan dalam bentuk eksponensial. Berikut contoh penggunaan variabel float.

```
<?php
    $x = 10.365;

    var_dump($x);
?>
```

Boolean

Boolean merepresentasikan dua kemungkinan keadaan: TRUE atau FALSE. Boolean sering digunakan pada kondisi percobaan. Berikut contoh penggunaannya:

```
$x = true;  
$y = false;
```

Object / objek

Object adalah sebuah tipe data yang menyimpan data dan informasi bagaimana memproses data tersebut. Object harus di deklarasikan secara eksplisit. Pertama kita harus mendeklarasikan class dari object. Class adalah struktur yang bisa memiliki properties dan method. Berikut contohnya:

```
<?php  
class Car {  
    function Car() {  
        $this->model = "VW";  
    }  
}  
// membuat sebuah objek  
$herbie = new Car();  
// Menunjukkan isi dari  
objek echo $herbie->model;  
?>
```

Nilai NULL

NULL adalah tipe data khusus yang bisa memiliki hanya satu nilai: NULL. Sebuah variabel dengan tipe data NULL adalah variabel yang tidak memiliki nilai. Jika sebuah variabel dibuat tanpa nilai, maka otomatis akan ditentukan dengan nilai NULL. Variabel bisa juga dikosongkan dengan mengatur nilainya menjadi NULL:

```
<?php  
$x = "Hello world!";  
$x = null;  
var_dump($x);  
?>
```


Resource

Tipe khusus resource bukanlah merupakan tipe data yang sebenarnya. Ia menyimpan referensi fungsi dan sumberdaya eksternal untuk PHP. Contoh umum penggunaan tipe data resource adalah pada pemanggilan database.

Fungsi-fungsi Manipulasi String

String merupakan susunan dari karakter yang memiliki fungsi khusus untuk mengolahnya. Seperti fungsi `strlen()`, berfungsi menghitung panjang dari string. Contohnya sebagai berikut:

```
<?php
    echo strlen("Hello world!");
?>
```

Hasil dari kode tersebut adalah panjang dari string "Hello world!" yaitu 12 karakter. Panjang string juga bisa diukur perkata menggunakan fungsi `str_word_count()` seperti contoh berikut:

```
<?php
    echo str_word_count("Hello world!");
?>
```

Keluaran dari kode tersebut berupa jumlah kata pada string "Hello world!" yaitu 2. PHP juga memiliki fungsi yang bisa digunakan untuk membalikkan susunan string, yaitu menggunakan fungsi `strrev()` seperti contoh kode program berikut yang menghasilkan string "dlrow olleH" dari inputan string "Hello world!".

```
<?php
    echo strrev("Hello world!");
?>
```

PHP juga memiliki fungsi untuk menemukan teks yang spesifik didalam sebuah string, yaitu dengan menggunakan fungsi `strpos()`. Jika teks yang dicari ditemukan didalam string, maka fungsi akan mengembalikan index posisi dari karakter awal teks pertama yang cocok. Jika tidak ditemukan, maka fungsi `strpos()` akan mengembalikan nilai `FALSE`. Kode program berikut adalah contoh dari penggunaan fungsi `strpos()` pada string "Hello world!" untuk menemukan teks "world", sehingga keluaran yang dihasilkan adalah 6.

```
<?php
    echo strpos("Hello world!", "world");
?>
```

PHP juga memiliki fungsi untuk mengganti teks didalam string, yaitu dengan menggunakan fungsi `str_replace()`. Contoh berikut adalah kode program yang digunakan untuk mengganti teks "world" pada string "Hello world!" dengan teks "Dolly".

```
<?php
    echo str_replace("world", "Dolly", "Hello world!");
?>
```

Konstanta / Constant

Konstanta seperti variabel, tetapi setelah dideklarasikan nilai konstanta tidak bisa dirubah. Konstanta adalah nama atau indintifikasi untuk nilai sederhana. Nilainya tidak bisa diubah selama program berjalan. Nama konstanta yang valid dimulai dengan huruf atau garis bawah, tidak ada tanda \$ sebelum nama konstanta. Konstanta memiliki lingkup global. Gunakan fungsi `define(nama, nilai, case-sensitif)` untuk mendeklarasikan konstanta. Parameter nama adalah nama dari konstanta yang akan digunakan, parameter nilai adalah nilai dari konstanta yang akan digunakan pada program, dan parameter case-sensitive digunakan untuk menspesifikasikan apakah nama konstanta harus case-sensitive. Secara default, nilai case-sensitive adalah false.

```
<?php
    // nama constant adalah case-sensitive
    define("GREETING", "Welcome to W3Schools.com!");
    echo GREETING;
?>
```

Operator PHP

Operator digunakan untuk melakukan operasi pada variabel-variabel dan nilai-nilai. Operator dibagi menjadi beberapa kelompok sebagai berikut:

Operator-operator aritmatika

Operator aritmatika pada PHP digunakan dengan nilai numeris untuk melakukan operasi aritmatika umum, seperti penjumlahan, pengurangan, perkalian, dan lain-lain.

Operator	Nama	Contoh	Keterangan
+	Penjumlahan	$\$x + \y	Nilai x ditambah dengan nilai y
-	Pengurangan	$\$x - \y	Nilai x dikurangi dengan nilai y
*	Perkalian	$\$x * \y	Kelipatan nilai x sebanyak y
/	Pembagian	$\$x / \y	Nilai x dibagi dengan nilai y
%	Modulus	$\$x \% \y	Menghasilkan sisa pembagian dari nilai x dengan nilai y
**	Perpangkatan	$\$x ** \y	Nilai x pangkat y (baru diperkenalkan pada PHP 5.6)

Operator-operator assignment (penugasan)

Operator assignment pada PHP digunakan dengan nilai numeris untuk menuliskan nilai pada variabel. Operator assignment dasar adalah "=" yang memiliki maksud menentukan nilai variabel sebelah kiri dengan hasil operasi atau nilai variabel sebelah kanan.

Assignment	Persamaan	Keterangan
$x = y$	$x = y$	Memasukkan nilai variabel y kedalam variabel x
$x += y$	$x = x + y$	Menambahkan nilai variabel x dengan nilai variabel y
$x -= y$	$x = x - y$	Mengurangi nilai variabel x dengan nilai variabel y
$x *= y$	$x = x * y$	Mengalikan nilai variabel x dengan nilai variabel y
$x /= y$	$x = x / y$	Membagi nilai variabel x dengan nilai variabel y
$x \% = y$	$x = x \% y$	Modulus

Operator-operator perbandingan

Operator perbandingan digunakan untuk membandingkan nilai dua variabel.

Operator	Nama	Contoh	Keterangan
'=='	persamaan	$\$x == \y	Mengembalikan nilai true jika \$x sama dengan \$y
'==='	Identik	$\$x === \y	Mengembalikan nilai true jika \$x sama dengan \$y dan tipe mereka sama

'!=='	Tidak sama	$\$x \neq \y	Mengembalikan nilai true jika \$x tidak sama dengan \$y
'<>'	Tidak sama	$\$x \neq \y	Mengembalikan nilai true jika \$x tidak sama dengan \$y
'!=='	Tidak identik	$\$x \neq \y	Mengembalikan nilai true jika \$x tidak sama dengan \$y dan tipe mereka tidak sama
'>'	Lebih besar dari	$\$x > \y	Mengembalikan nilai true jika nilai \$x lebih besar dari nilai \$y
'<'	Lebih kecil dari	$\$x < \y	Mengembalikan nilai true jika nilai \$x lebih kecil dari nilai \$y
'>='	Lebih besar atau sama dengan	$\$x \geq \y	Mengembalikan nilai true jika nilai \$x lebih besar atau sama dengan nilai \$y
'<='	Lebih kecil atau sama dengan	$\$x \leq \y	Mengembalikan nilai true jika nilai \$x lebih kecil atau sama dengan nilai \$y

Operator-operator iterasi

Iterasi memiliki dua jenis, iterasi penambahan dan iterasi pengurangan. operator iterasi penambahan digunakan untuk menambah nilai variabel, sedangkan operator iterasi pengurangan digunakan untuk mengurangi nilai variabel.

Operator	Nama	Keterangan
++\$x	Pre-increment	Menambah nilai x dengan satu, kemudian kembalikan nilai x
\$x++	Post-increment	Kembalikan nilai x, kemudian tambah nilai x dengan satu
--\$x	Pre-decrement	Mengurangi nilai x dengan satu, kemudian kembalikan nilai x
\$x--	Post-decrement	Kembalikan nilai x, kemudian kurangi nilai x dengan satu

Operator-operator logika

Operator logika digunakan untuk mengkombinasikan pernyataan kondisional.

Operator	Nama	Contoh	Hasil
and	dan	$\$x \text{ and } \y	True jika x dan y adalah true
or	atau	$\$x \text{ or } \y	True jika x atau y adalah true

xor	Atau eksklusif	\$x xor \$y	True jika x atau y adalah true, tetapi tidak dua-duanya true.
&&	Dan	\$x && \$y	True jika x dan y adalah true
	atau	\$x \$y	True jika x atau y adalah true
!	tidak	!\$x	True jika x tidak true

Operator-operator string

Dua operator didesain khusus untuk string.

Operator	Nama	Contoh	Keterangan
.	Concatenation	\$text1 . \$text2	Menggabungkan isi dari string text1 dengan isi dari string text2
.=	Concatenation assignment	\$text1 .= \$text2	Membubuhi string pada text2 ke text1

Operator-operator array

Operator array php digunakan untuk membandingkan dua array.

Operator	Nama	Contoh	Keterangan
'+'	Union	\$x + \$y	Penggabungan dari x dan y
'=='	Equality	\$x == \$y	Mengembalikan nilai true jika x dan y memiliki pasangan nilai yang sama
'==='	Identity	\$x === \$y	Mengembalikan nilai true jika x dan y memiliki pasangan nilai yang sama dengan susunan dan tipe yang sama
'!='	Inequality	\$x != \$y	Mengembalikan nilai true jika x tidak sama dengan y
'<>'	Inequality	\$x <> \$y	Mengembalikan nilai true jika x tidak sama dengan y
'!=='	Non-identity	\$x !== \$y	Mengembalikan nilai true jika x tidak identik dengan y

Percabangan dengan if ... else ... elseif

Percabangan atau Pernyataan kondisional digunakan untuk melakukan aksi yang berbeda berdasarkan pada kondisi yang berbeda. dalam PHP terdapat beberapa pernyataan kondisional yaitu sebagai berikut:

Pernyataan if

Mengeksekusi beberapa kode jika satu kondisi benar. Syntaks dari kondisi ini adalah sebagai berikut:

```
if (kondisi) {  
    kode yang akan dieksekusi jika kondisi true;  
}
```

Pernyataan If ... else

Mengeksekusi beberapa kode jika kondisi benar dan mengeksekusi kode yang lain jika kondisinya tidak benar. Syntax dari kondisi ini adalah sebagai berikut:

```
if (kondisi) {  
    kode yang akan dieksekusi jika kondisi true.  
} else {  
    kode yang akan dieksekusi jika kondisi false.  
}
```

Pernyataan if ... elseif ... else

Mengeksekusi kode yang berbeda untuk lebih dari dua kondisi. elseif bisa dibuat lebih dari satu. Syntax dari kondisi ini adalah sebagai berikut:

```
if (kondisi pertama) {  
    kode yang akan dieksekusi jika kondisi pertama true.  
} elseif (kondisi kedua) {  
    kode yang akan dieksekusi jika kondisi kedua true.  
} else {  
    kode yang akan dieksekusi jika semua kondisi false.  
}
```

Pernyataan switch

Pernyataan switch digunakan untuk melakukan aksi yang berbeda berdasarkan kondisi yang berbeda. Pernyataan ini digunakan untuk memilih satu dari banyak blok kode yang akan dieksekusi. Sintax dari pernyataan ini adalah sebagai berikut:

```
switch (n) {  
    case label1:  
        kode yang akan dieksekusi jika n = label1;  
        break;  
    case label2:  
        kode yang akan dieksekusi jika n = label2;  
        break;  
    case label3:  
        kode yang akan dieksekusi jika n = label3;
```

```
        break;
    ...
    default:
        kode yang akan dieksekusi jika n berbeda dari semua label yang ada.
}
```

Perulangan

Perulangan digunakan untuk mengulang eksekusi kode program satu kali atau lebih dari satu kali. Terdapat beberapa macam perulangan yaitu sebagai berikut:

Perulangan dengan while

While loop pada PHP akan mengeksekusi blok kode selama kondisi yang dispesifikasikan benar. Syntax dari while loop adalah sebagai berikut:

```
while (kondisi true) {
    kode yang akan dieksekusi.
}
```

Perulangan dengan do ... while

Perulangan do ... while akan langsung mengeksekusi blok kode kemudian memeriksa kondisi. Blok kode akan terus dieksekusi selama kondisi true. Syntax untuk perulangan do ... while adalah sebagai berikut:

```
do {
    kode yang akan dieksekusi.
} while ( kondisi true );
```

Perulangan dengan for loop

Perulangan for loop digunakan ketika sudah diketahui lebih lanjut berapa banyak kode harus diulang. Syntax perulangan loop adalah sebagai berikut:

```
for (Inisialisasi; kondisi; peningkatan) {  
    kode yang akan dieksekusi;  
}
```

Parameter:

Inisialisasi : menentukan nilai awal iterasi.

Kondisi : evaluasi terhadap kondisi nilai iterasi, jika kondisi nilai iterasi true, maka perulangan akan dilanjutkan, jika kondisi nilai iterasi false, maka perulangan berhenti.

Penjumlahan : peningkatan nilai iterasi.

Perulangan dengan foreach

Perulangan foreach hanya bekerja pada array, dan digunakan untuk perulangan melalui setiap nilai pada array. Syntax dari perulangan foreach adalah sebagai berikut:

```
foreach ($array as $variabel) {  
    kode yang akan dieksekusi;  
}
```

Setiap nilai dalam array akan dimasukkan kedalam variabel dalam setiap perulangan.

Fungsi

Kekuatan dari PHP berasal dari fungsi-fungsinya. PHP memiliki lebih dari 1000 fungsi built-in. Namun dalam PHP bisa juga dibuat fungsi sendiri. Fungsi adalah blok kode yang bisa digunakan secara berulang-ulang kali dalam program. Fungsi tidak akan dieksekusi secara langsung jika halaman web sedang dimuat. Sebuah fungsi akan dieksekusi oleh fungsi yang memanggilnya. Syntax dari fungsi yang dibuat sendiri adalah sebagai berikut:

```
function nama_fungsi() {  
    kode yang akan dieksekusi.  
}
```

Nama fungsi bisa dimulai dengan huruf atau garis bawah (bukan bilangan). Tentukan nama fungsi berdasarkan apa kegunaannya. Informasi dari luar fungsi bisa dimasukkan kedalam fungsi melalui argument. Argumen adalah seperti variabel. Argumen dispesifikasikan setelah nama fungsi didalam kurung. Argument bisa dibuat banyak, antara deklarasi argument harus dipisah dengan tanda koma. Jika argumen memiliki nilai default, maka argument tersebut bisa langsung ditentukan nilainya dengan tanda sama dengan. Berikut adalah contoh penetapan argument dengan nilai default.


```
<?php
    function setHeight($minheight = 50) {
        echo "The height is : $minheight <br>";
    }

    setHeight(350);
    setHeight(); // will use the default value of 50
    setHeight(135);
    setHeight(80);
?>
```

Fungsi juga bisa memberikan nilai kembalian. Untuk memberikan nilai kembalian, fungsi tersebut harus menggunakan perintah return. Berikut adalah contoh dari fungsi yang mengembalikan sebuah nilai:

```
<?php
    function sum($x, $y) {
        $z = $x + $y;
        return $z;
    }

    echo "5 + 10 = " . sum(5, 10) . "<br>";
    echo "7 + 13 = " . sum(7, 13) . "<br>";
    echo "2 + 4 = " . sum(2, 4);
?>
```

Array

Array adalah variabel khusus, yang bisa menyimpan banyak nilai pada satu waktu. Nilai dalam array bisa diakses dengan cara merujuk pada indexnya. Pada PHP fungsi array() digunakan untuk membuat sebuah array. Pada PHP, terdapat tiga macam tipe array:

Array Terindex

Array terindex adalah array dengan index numerik. Ada dua cara untuk membuat array terindex, pertama, array bisa ditentukan secara otomatis dimana indexnya selalu mulai dari 0, contoh : `$mobil = array("volvo","BMW","Toyota");` dan yang kedua, array bisa ditentukan secara manual, contohnya sebagai berikut:

```
$mobil[0] = "Volvo";  
$mobil[1] = "BMW";  
$mobil[2] = "Toyota";
```

Panjang array atau jumlah element bisa juga dihitung menggunakan fungsi `count()`, untuk menghitung jumlah element pada variabel array `$mobil[]` dapat ditulis `count($mobil)`, fungsi ini akan mengembalikan nilai integer sebagai jumlah element dalam variabel `$mobil[]`.

Array Assosiatif

Array asosiatif adalah array dengan nama kunci yang ditentukan. Ada dua cara membuat array assosiatif, pertama dengan mendeklarasikan ketika memanggil fungsi `array()`, contoh : `$umur = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");` yang kedua dengan mendeklarasikan nilainya satu-persatu, contoh:

```
$umur['Peter'] = "35";  
$umur['Ben'] = "37";  
$umur['Joe'] = "43";
```

Untuk melakukan perulangan pada array assosiatif, dapat menggunakan `foreach($variabel as $kunci => $nilai)` dimana `$variabel` adalah variabel array, `$kunci` adalah variabel yang akan berisi kunci array dan `$nilai` adalah variabel yang berisi nilai dari array.

Element dalam sebuah array bisa diurutkan dalam susunan alfabet atau numeris, descending atau ascending. Berikut adalah fungsi-fungsi yang bisa digunakan untuk mengurutkan element-element dalam array:

sort() - Mengurutkan array dalam susunan ascending. Misalkan fungsi ini digunakan untuk mengurutkan array `$mobil`, maka isi array `$mobil` akan terurut berdasarkan alfabet, yaitu sebagai berikut:

```
BMW  
Toyota  
Volvo
```

rsort() - Mengurutkan array dalam susunan descending. Misalkan fungsi ini digunakan untuk mengurutkan array \$mobil, maka urutan nilai pada array mobil akan menjadi sebagai berikut:

Volvo
Toyota
BMW

asort() - Mengurutkan array asosiatif dalam susunan ascending, berdasarkan pada nilainya. Misalkan fungsi ini digunakan untuk mengurutkan nilai pada variabel array \$umur, maka urutannya akan berdasarkan nilai pada variabel tersebut. Jika dicetak dengan kuncinya tampilan isi dari array \$umur adalah sebagai berikut:

Peter 35
Ben 37
Joe 43

ksort() - Mengurutkan array asosiatif dalam susunan ascending, berdasarkan kuncinya. Misalkan fungsi ini digunakan untuk mengurutkan kunci pada array \$umur, maka kunci pada variabel tersebut akan terurut ascending seperti berikut:

Joe 43
Ben 37
Peter 35

arsort() - Mengurutkan array asosiatif dalam susunan descending, berdasarkan pada nilainya. Misalkan fungsi ini digunakan untuk mengurutkan data pada variabel \$umur, maka urutan data pada variabel tersebut akan mengikuti susunan nilai dari besar ke kecil.

Joe 43
Ben 37
Peter 35

krsort() - Mengurutkan array asosiatif dalam susunan descending, berdasarkan kuncinya. Misalkan fungsi ini digunakan untuk mengurutkan data pada variabel array \$umur, maka susunannya akan mengurutkan kunci dari besar akhir pada alphabet ke awal pada alphabet.

Peter 35
Joe 43
Ben 37

Array Multidimensi

Array multidimesi adalah array yang mengandung satu atau banyak array. Dimensi dari sebuah array mengindikasikan jumlah index yang diperlukan untuk memilih element. Array dua dimensi memerlukan dua index untuk memilih sebuah element, array tiga dimensi memerlukan tiga index untuk memilih sebuah element.

Name	Stock Sold	
Volvo	22	18
BMW	15	13
Saab	5	2
Land Rover	17	15

Misalkan array dua dimensi digunakan untuk menyimpan data-data pada tabel diatas, maka contoh kode program untuk menyimpan data-data tersebut adalah sebagai berikut:

```
$cars = array(  
    array("Volvo",22,18),  
    array("BMW",15,13),  
    array("Saab",5,2),  
    array("Land Rover",17,15)  
);
```

Variabel Global dan Superglobal

Superglobal diperkenalkan pada PHP 4.1.0 dan merupakan variabel built-in yang selalu ada pada semua scope atau lingkup. Beberapa variabel yang didefinisi sebelumnya dalam PHP adalah “superglobal”, yang berarti bahwa mereka selalu bisa diakses tanpa terpengaruh oleh lingkup atau scope sehingga bisa diakses oleh semua fungsi atau class tanpa harus melakukan suatu teknik khusus.

Variabel PHP superglobal adalah sebagai berikut:

PHP \$GLOBALS

\$GLOBALS adalah sebuah variabel PHP yang super global yang digunakan untuk mengakses variabel dari mana saja didalam script PHP, termasuk dari dalam fungsi atau method. PHP menyimpan semua variabel-variabel global dalam sebuah array \$GLOBALS[index]. Index adalah nama dari variabel. Contoh berikut menunjukkan cara menggunakan variabel \$GLOBALS:

```
<?php
    $x = 75;
    $y = 25;
    function addition() {
        $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
    }
    addition();
    echo $z;
?>
```

\$_SERVER

\$_SERVER adalah variabel super global yang menyimpan informasi tentang header-header, path-path (jalur), dan skrip lokasi. Berikut adalah contoh penggunaan variabel global \$_SERVER:

```
<?php
    echo $_SERVER['PHP_SELF'];
    echo "<br>";
    echo $_SERVER['SERVER_NAME'];
    echo "<br>";
    echo $_SERVER['HTTP_HOST'];
    echo "<br>";
    echo $_SERVER['HTTP_REFERER'];
    echo "<br>";
    echo $_SERVER['HTTP_USER_AGENT'];
    echo "<br>";
    echo $_SERVER['SCRIPT_NAME'];
?>
```

Tabel berikut adalah daftar dari elemen-elemen penting yang bisa digunakan pada variabel \$_SERVER:

Elemen / Kode	Deskripsi
<code>\$_SERVER['PHP_SELF']</code>	Memberikan nama file dari skrip yang sedang dieksekusi.
<code>\$_SERVER['GATEWAY_INTERFACE']</code>	Memberikan versi dari CGI (Common Gateway Interface) yang digunakan oleh server.
<code>\$_SERVER['SERVER_ADDR']</code>	Memberikan alamat IP dari server host.
<code>\$_SERVER['SERVER_NAME']</code>	Memberikan nama dari server host. Misal : <code>www.umrah.ac.id</code>
<code>\$_SERVER['SERVER_SOFTWARE']</code>	Memberikan string identifikasi server seperti Apache/2.2.24)
<code>\$_SERVER['SERVER_PROTOCOL']</code>	Memberikan nama dan revisi dari protokol informasi (seperti HTTP/1.1)
<code>\$_SERVER['REQUEST_METHOD']</code>	Memberikan method request yang digunakan untuk mengakses halaman (seperti POST)
<code>\$_SERVER['REQUEST_TIME']</code>	Memberikan waktu atau timestamp dari permulaan request (seperti 1377687496)
<code>\$_SERVER['QUERY_STRING']</code>	Memberikan string query jika halaman diakses via string query.
<code>\$_SERVER['HTTP_ACCEPT']</code>	Memberikan header penerima dari request saat ini.
<code>\$_SERVER['HTTP_ACCEPT_CHARSET']</code>	Memberikan header Accept_Charset dari request saat ini (seperti utf-8,ISO-8859-1)
<code>\$_SERVER['HTTP_HOST']</code>	Memberikan header host dari permintaan atau request saat ini.
<code>\$_SERVER['HTTP_REFERER']</code>	Memberikan URL lengkap dari halaman saat ini (kurang handal karena tidak semua user-agents mendukung)
<code>\$_SERVER['HTTPS']</code>	Skrip yang di-query melalui protokol keamanan HTTP.
<code>\$_SERVER['REMOTE_ADDR']</code>	Memberikan alamat IP user yang sedang mengakses halaman saat ini.
<code>\$_SERVER['REMOTE_HOST']</code>	Memberikan nama host dari user yang mengakses halaman saat ini.
<code>\$_SERVER['REMOTE_PORT']</code>	Memberikan port yang sedang digunakan pada komputer user untuk berkomunikasi pada server web.
<code>\$_SERVER['SCRIPT_FILENAME']</code>	Memberikan nama path / jalur yang sedang mengeksekusi skrip.
<code>\$_SERVER['SERVER_ADMIN']</code>	Memberikan nilai yang diberikan untuk

	SERVER_ADMIN yang menunjuk pada file konfigurasi server web (jika skrip berjalan pada host virtual, ia akan menjadi nilai yang ditentukan untuk virtual host) (seperti seseorang@w3schools.com)
<code>\$_SERVER['SERVER_PORT']</code>	Memberikan port pada komputer server yang sedang digunakan oleh server web untuk komunikasi (seperti port 80)
<code>\$_SERVER['SERVER_SIGNATURE']</code>	Memberikan versi server dan nama virtual host yang ditambahkan pada halaman yang dibangkitkan oleh server.
<code>\$_SERVER['PATH_TRANSLATED']</code>	Memberikan sistem file yang berdasarkan pada jalur/path ke skrip saat ini.
<code>\$_SERVER['SCRIPT_NAME']</code>	Memberikan jalur / path dari skrip saat ini.
<code>\$_SERVER['SCRIPT_URI']</code>	Memberikan URI dari halaman saat ini.

`$_REQUEST`

`$_REQUEST` PHP digunakan untuk mengumpulkan data setelah submit sebuah form HTML. Contoh berikut menunjukkan sebuah form dengan field input dan tombol submit. Ketika user melakukan submit data dengan menekan "Submit", data form dikirimkan untuk file yang dispesifikasikan dalam atribut aksi dari tag `<form>`. Dalam contoh ini, kita memperhatikan file ini memproses data form. Jika pengguna ingin menggunakan file PHP lain untuk memproses data form, menggantinya dengan nama file yang dipilih, maka gunakan variabel super global `$_REQUEST` untuk mengumpulkan field nilai input.

```
<html>
  <body>
    <form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
      Name: <input type="text" name="fname">
      <input type="submit">
    </form>
    <?php
      if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // mengumpulkan nilai input field
        $name = $_REQUEST['fname'];
        if (empty($name)) {
          echo "Name is empty";
        }
      }
    }
```

```

        } else {
            echo $name;
        }
    }
    ?>
</body>
</html>

```

\$_POST

\$_POST PHP sangat banyak digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method="post". \$_POST juga banyak digunakan untuk melewati variabel. Contoh berikut menunjukkan sebuah form dengan sebuah input field dan sebuah tombol submit. Ketika user melakukan submit data dengan menekan tombol submit, data form dikirim ke file yang dispesifikasikan dalam atribut aksi dari tag <form>. Pada contoh ini, perhatikan file untuk pemrosesan data. Jika ingin menggunakan file PHP lain untuk memproses data form, ganti dengan nama file yang diinginkan. Kemudian, gunakan variabel super global \$_POST untuk mengumpulkan nilai field input.

```

<html>
  <body>
    <form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
      Name: <input type="text" name="fname">
      <input type="submit">
    </form>
    <?php
      if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // collect value of input field
        $name = $_POST['fname'];
        if (empty($name)) {
          echo "Name is
          empty"; } else {
          echo $name;
        }
      }
    ?>
  </body>
</html>

```


\$_GET

\$_GET PHP bisa digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method="get". Juga bisa digunakan untuk mengumpulkan data yang dikirimkan dalam URL. Asumsikan halaman HTML yang mengandung hyperlink dengan parameter berikut:

```
<html>
  <body>
    <a href="test_get.php?subject=PHP&web=umrah.ac.id">Test $GET
  </a> </body>
</html>
```

kemudian user meng-klik link "Test \$GET", parameter "subject" dan "web" dikirimkan ke skrip "test_get.php", nilai dari kedua parameter tersebut bisa diakses dengan \$_GET. Contoh berikut adalah skrip yang bisa mengakses nilai dari dua parameter tersebut.

```
<html>
  <body>
    <?php
      echo "Study " . $_GET['subject'] . " at " . $_GET['web'];
    ?>
  </body>
</html>
```

TUGAS

Buatlah program web untuk mengolah nama-nama anggota keluarga anda dengan kriteria sebagai berikut:

1. Nama-nama disimpan didalam variabel array.
2. Program menampilkan jumlah kata dan jumlah huruf pada nama yang anda tuliskan.
3. Program menampilkan kebalikan dari nama yang diinputkan.
4. Program menampilkan jumlah konsonan dan jumlah vokal pada nama yang telah diinputkan.

Modul II : Form Handling

Tujuan

- Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML.
- Mahasiswa mampu membuat batasan-batasan untuk menangani inputan dari form HTML.

Variabel superglobal PHP \$_GET dan \$_POST digunakan untuk mengumpulkan data-form. Contoh berikut menunjukkan form HTML sederhana dengan dua field input dan tombol submit:

```
<html>
  <body>
    <form action="welcome.php" method="post">
      Name: <input type="text" name="name"><br>
      E-mail: <input type="text" name="email"><br>
      <input type="submit">
    </form>
  </body>
</html>
```

Ketika user mengisi form, dan menekan tombol click, data form dikirim untuk memproses file PHP dengan nama “welcome.php”. Data form dikirimkan dengan method HTTP POST. Untuk menampilkan data yang sudah disubmit bisa dilakukan dengan mencetak data tersebut menggunakan perintah echo. File “welcome.php” adalah sebagai berikut:

```
<html>
  <body>
    Welcome <?php echo $_POST["name"]; ?><br>
    Your email address is: <?php echo $_POST["email"];
    ?> </body>
</html>
```

Jika field nama diinputkan dengan Tono dan email diinputkan dengan tono@mail.com maka output yang akan tampil adalah sebagai berikut:

Welcome Budi

Your email address is tono@mail.com

Hasil yang sama juga akan tampil dengan menggunakan method get sebagai berikut:

```
<html>
  <body>

    <form action="welcome_get.php" method="get">
      Name: <input type="text" name="name"><br>
      E-mail: <input type="text" name="email"><br>
      <input type="submit">
    </form>
  </body>
</html>
```

dengan file “welcome_get.php” sebagai berikut:

```
<html>
  <body>
    Welcome <?php echo $_GET["name"]; ?><br>
    Your email address is: <?php echo $_GET["email"];
    ?> </body>
</html>
```

GET vs. POST

GET dan POST membuat sebuah array (contoh array(kunci => nilai, kunci2 => nilai2, kunci3 => nilai3, ...)). Array ini menyimpan pasangan kunci/nilai, dimana kunci-kunci adalah nama-nama dari form control dan nilai-nilai adalah data input dari user. Method GET diakses menggunakan \$_GET dan method POST diakses menggunakan \$_POST. Kedua variabel ini adalah variabel superglobal, yang selalu bisa diakses, tanpa memperhatikan lingkup dan bisa diakses dari fungsi, class atau file yang berbeda tanpa harus melakukan teknik khusus. \$_GET adalah sebuah array dari variabel yang dikirimkan ke skrip melalui parameter URL. \$_POST adalah sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST.

Kapan sebaiknya menggunakan GET?

Informasi dikirim dari sebuah form dengan method GET bisa dilihat oleh semua orang (semua nama dan nilai variabel ditampilkan di URL). GET juga memiliki batas pada jumlah informasi yang dikirim. Batasannya adalah sekitar 2000 karakter. Namun, karena variabel ditunjukkan di URL, ia memungkinkan untuk dilakukan bookmark halaman. Dalam beberapa kasus, hal ini sangat bermanfaat. GET bisa digunakan untuk mengirimkan data yang tidak sensitif.

Ingat! GET tidak boleh digunakan untuk mengirimkan password atau informasi sensitif lainnya!

Kapan menggunakan POST?

Informasi yang dikirim dari sebuah form dengan method POST tidak bisa dilihat oleh siapapun (semua nama-nama atau nilai-nilai tertanam didalam body request HTTP) dan tidak memiliki batasan jumlah informasi yang akan dikirim. POST juga mendukung fungsionalitas lanjutan seperti dukungan untuk input biner multi-part ketika sedang melakukan upload file ke server. Namun, karena variabel tidak ditampilkan di URL, tidak mungkin untuk dilakukan bookmark halaman (data tidak ter-bookmark). Developer lebih baik menggunakan POST untuk mengirimkan data form.

Validasi Form PHP

Pertimbangkan keamanan ketika memproses form PHP!

PHP Form Validation Example

** required field.*

Name: *

E-mail: *

Website:

Comment:

Gender: ☐ Female ☐ Male *

Form HTML yang akan kita gunakan pada modul ini, mengandung bermacam-macam field input, misalnya text field yang harus diisi dan text field yang opsional, tombol pilihan (radio button), dan tombol submit. Rule atau aturan validasi untuk form diatas adalah sebagai berikut:

Field	Rule Validasi
Name	Dibutuhkan. + Harus hanya mengandung huruf dan spasi
E-mail	Dibutuhkan. + Harus mengandung sebuah alamat email yang valid dengan @ dan .
Website	Opsional. Jika ada, harus mengandung URL yang valid.
Comment	Opsional. Field input multi-line (text area).
Gender	Dibutuhkan. Harus memilih salah satu

Kode HTML untuk membentuk Form tersebut adalah sebagai berikut:

Text Field

Field nama, email dan website adalah elemen-elemen text input, dan field komentar adalah textarea yaitu sebagai berikut:

Name: <input type="text" name="name">

E-mail: <input type="text" name="email">

Website: <input type="text" name="website">

Comment: <textarea name="comment" rows="5" cols="40"></textarea>

Radio Button

Field jenis kelamin adalah radio button yaitu sebagai berikut:

Gender:

<input type="radio" name="gender" value="female">Female

<input type="radio" name="gender" value="male">Male

Form Element

Kode HTML untuk membentuk form pada gambar diatas adalah sebagai berikut:

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

Ketika form disubmit, data pada form dikirim dengan method “post”. **\$_SERVER[“PHP_SELF”]** adalah variabel super global yang mengembalikan nama file dari skrip yang sedang dieksekusi. Sehingga kode form diatas mengirim data pada form ke halaman itu sendiri. Sedangkan fungsi **htmlspecialchars()** adalah fungsi yang mengkonversikan karakter-karakter spesial ke entitas HTML. Sebagai contoh, fungsi tersebut akan mengkonversikan karakter < dan > menjadi < dan >. Fungsi ini mencegah injeksi yang bisa dilakukan dengan HTML atau javascript (Cross-site Scripting Attack) pada form tersebut.

Catatan Penting pada Keamanan Form PHP

Variabel **\$_SERVER[“PHP_SELF”]** bisa digunakan oleh hacker! Jika **PHP_SELF** digunakan pada halaman web, user bisa memasukkan skrip dengan terlebih dahulu memasukkan garis miring (/) kemudian beberapa perintah Cross Site Scripting (XSS) untuk dieksekusi. XSS adalah tipe kelemahan keamanan komputer yang secara tipikal ditemukan dalam aplikasi web.

Asumsikan kita memiliki halaman web dengan nama “test_form.php”, dan form hanya kita deklarasikan sebagai berikut:

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

Kemudian user memasukkan URL pada address bar dengan alamat sebagai berikut:

[http://localhost/<nama_folder>/test_form.php/%22%3E%3Cscript%3Ealert\('hacked'\)%3C/script%3E](http://localhost/<nama_folder>/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E)

yang jika ditranslasikan akan menjadi:

```
<form method="post" action="test_form.php/"><script>alert('hacked')</script>
```

Kode ini menambah tag script dan perintah alert atau peringatan, ketika halaman dibuka, kode javascript tersebut akan dieksekusi, maka user akan melihat kotak peringatan dengan tulisan “hacked”.

Berhati-hatilah dengan kemungkinan penambahan kode javascript pada tag <script>!

Hacker bisa mengarahkan user ke file pada server yang lain, dan file itu bisa mengandung kode yang bisa merubah variabel global atau melakukan submit form pada alamat web yang berbeda untuk mencuri data user.

Bagaimana menghindari penyalahgunaan \$_SERVER["PHP_SELF"]?

Caranya adalah dengan menggunakan fungsi htmlspecialchars(). Fungsi tersebut akan mengkonversikan karakter khusus ke entitas HTML. Ketika user memasukkan URL dengan tag script seperti contoh sebelumnya, maka akan ditranslasikan sebagai berikut:

```
<form method="post"
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

dengan cara ini, percobaan penyalahgunaan akan gagal.

Memvalidasi data Form dengan PHP

Hal pertama yang akan kita lakukan adalah memasukkan semua variabel melalui fungsi htmlspecialchars(). Kemudian ada juga dua hal ketika user melakukan submit form:

1. Membuang karakter-karakter yang tidak dibutuhkan (seperti spasi extra, tab extra, dan baris baru yang ekstra) dari data input user (dengan fungsi trim()).
2. Membuang backslash (\) satu garis miring dari data input user (dengan fungsi stripslashes()).

Langkah berikutnya adalah membuat fungsi yang akan melakukan pemeriksaan kebenaran data yang diinputkan oleh user. Contohnya adalah sebagai berikut:

```
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

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

?>
```

Ingat bahwa pada permulaan skrip, adalah pemeriksaan apakah form sudah disubmit menggunakan `$_SERVER["REQUEST_METHOD"]`. Jika `REQUEST_METHOD` adalah `POST`, maka form telah disubmit dan seharusnya tervalidasi. Jika belum tersubmit, lewati langkah validasi dan tampilkan form kosong. Namun pada contoh diatas semua field input adalah opsional. Skrip bekerja baik bahkan jika user tidak melakukan entri data.

Field yang Dibutuhkan

Kode program berikut terdapat tambahan variabel baru yaitu: `$nameErr`, `$emailErr`, `$genderErr`. Variabel-variabel error ini akan menangani pesan error untuk field yang dibutuhkan. Percabangan dengan `if else` juga akan ditambahkan untuk setiap variabel `$_POST`. Fungsinya untuk memeriksa apakah variabel `$_POST` kosong, hal ini dilakukan dengan menggunakan fungsi `empty()`. Jika kosong, maka pesan error disimpan dalam variabel error yang berbeda, dan jika tidak kosong, ia akan mengirim data input user melalui fungsi `test_input()`:

```
<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
    }

    if (empty($_POST["website"])) {
        $website = "";
    }
}
```



```

    } else {
        $website = test_input($_POST["website"]);
    }

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

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}
?>

```

Setelah kode diatas ditambahkan, beberapa skrip ditambahkan pada setiap field yang dibutuhkan pada form, fungsinya untuk menampilkan pesan error jika field yang dibutuhkan tidak diisi. Form HTMLnya adalah sebagai berikut:

```

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

    Name: <input type="text" name="name">
    <span class="error">* <?php echo
    $nameErr;?></span> <br><br>
    E-mail:
    <input type="text" name="email">
    <span class="error">* <?php echo $emailErr;?></span>
    <br><br>
    Website:
    <input type="text" name="website">
    <span class="error"><?php echo $websiteErr;?></span>
    <br><br>
    Comment: <textarea name="comment" rows="5" cols="40"></textarea>
    <br><br>
    Gender:
    <input type="radio" name="gender" value="female">Female

```

```
<input type="radio" name="gender" value="male">Male  
<span class="error">* <?php echo $genderErr;?></span>  
<br><br>  
<input type="submit" name="submit" value="Submit">  
  
</form>
```

Validasi Nama

Kode berikut menunjukkan cara sederhana untuk memeriksa apakah field nama hanya mengandung huruf dan spasi. Jika nilai dari nama tidak valid, maka pesan error akan disimpan didalam variabel \$nameErr:

```
$name = test_input($_POST["name"]);  
if (!preg_match("/^[a-zA-Z ]*$/", $name)) {  
    $nameErr = "Only letters and white space allowed";  
}
```

Fungsi preg_match() mencari string berdasarkan pola, mengembalikan nilai true jika polanya ada, false jika polanya tidak ada.

Validasi Email

Cara paling mudah dan paling aman untuk memeriksa apakah sebuah alamat email memiliki pola yang sesuai adalah dengan menggunakan fungsi filter_var(). Kode dibawah memeriksa apakah alamat email yang dimasukkan menggunakan pola yang sesuai atau tidak, jika tidak, maka pesan error akan disimpan kedalam variabel \$emailErr:

```
$email = test_input($_POST["email"]);  
if (!filter_var($email, FILTER_VALIDATE_EMAIL))  
    { $emailErr = "Invalid email format";  
}
```

Validasi URL

Kode program berikut menunjukkan cara untuk memeriksa apakah sintaks alamat URL valid atau tidak. Ekspresi reguler ini mengizinkan keberadaan tanda pisah pada URL. Jika sintaks alamat URL tidak valid, maka pesan error akan disimpan kedalam variabel \$websiteErr:

```

$website = test_input($_POST["website"]);
if (!preg_match("/^b(?:(:https?|ftp):\\V|www\\.)(-a-z0-9+&@#V%?=-_!|,.;)*[-a-z0-9+&@#V %=-_]/i",$website)) {
    $websiteErr = "Invalid URL";
}

```

Biasanya, jika user salah menginputkan nilai, maka halaman yang tampil adalah halaman yang sama dengan field yang sudah terisi dengan nilai field yang sudah diinput sebelumnya. Untuk menunjukkan nilai dalam field input setelah user menekan tombol submit, ada beberapa skrip PHP yang perlu ditambahkan didalam atribut value pada field input name, email, dan website. Khusus untuk field textarea, akan skrip tersebut akan ditambahkan antara tag <textarea> dan tag </textarea>. Skrip yang singkat akan mengeluarkan nilai dari variabel \$name, \$email, \$website dan \$comment. Untuk radio button atau tombol radio, akan ditambahkan kode yang membuat salah satu pilihan terpilih.

```

Name: <input type="text" name="name" value="<?php echo $name;?>">

E-mail: <input type="text" name="email" value="<?php echo $email;?>">

Website: <input type="text" name="website" value="<?php echo $website;?>">

Comment: <textarea name="comment" rows="5" cols="40"><?php echo
$comment;? ></textarea>

Gender:
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="female") echo
"checked";?> value="female">Female
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="male") echo
"checked";?> value="male">Male

```

TUGAS

Buatlah program web untuk menginputkan username dan password menggunakan form dan penanganan input data dengan kriteria sebagai berikut:

1. username yang diinputkan tidak boleh lebih dari tujuh karakter.
2. password yang diinputkan harus terdiri dari huruf kapital, huruf kecil, angka dan karakter khusus.
3. Jumlah karakter password tidak boleh kurang dari sepuluh karakter.

MODUL III : PHP MySQL Database

TUJUAN

- Mahasiswa mampu membuat program yang bisa menyimpan data dalam jumlah yang banyak.
- Mahasiswa mampu membuat program yang bisa mengolah data yang tersimpan dalam database.

Data dalam database MySQL disimpan dalam tabel-tabel. Sebuah tabel adalah koleksi dari data yang berelasi dan mengandung kolom dan baris. Database sangat bermanfaat untuk menyimpan informasi secara kategori. Contoh yang akan diberikan pada modul praktikum ini adalah tabel yang mengandung data Employees (Pekerja), Products (Produk), Customers (Pelanggan) dan Orders (Pesanan).

Membuka Koneksi

Sebelum mengakses data dalam database MySQL, kita harus terhubung ke server database MySQL. Berikut adalah contoh kode program agar terhubung dengan server mySQL:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);

    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

Jika kode diatas tidak berhasil, kemungkinan variabel \$connect_error sudah tidak tersedia pada versi PHP yang kita gunakan, maka gunakan kode program berikut:

```

<?php

$servername = "localhost";
$username = "username";
$password = "password";

// Membuat Hubungan
$conn = mysqli_connect($servername, $username, $password);

// Memeriksa
Hubungan if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>

```

Ketika selesai menggunakan data dari database, sebaiknya koneksi atau hubungan ke server ditutup, caranya dengan menggunakan kode program berikut:

```
mysqli_close($conn);
```

Membuat Database

Database pada MySQL bisa juga dibuat menggunakan kode program PHP. Program tersebut akan berisi statement SQL "CREATE DATABASE". Berikut adalah contoh program yang digunakan untuk membuat database "myDB":

```

<?php

$servername = "localhost";
$username = "username";
$password = "password";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password);
// Memeriksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

// Membuat Database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {

```

```

        echo "Database created successfully";
    } else {
        echo "Error creating database: " . mysqli_error($conn);
    }

    mysqli_close($conn);

```

?>

Membuat Tabel

Pembuatan tabel pada bahasa pemrograman PHP juga menggunakan statement SQL, yaitu statement “CREATE TABLE”. Contoh yang akan diberikan adalah pembuatan tabel MyGuests. Statement pembuatan tabel MyGuests adalah sebagai berikut:

```

CREATE TABLE MyGuests (
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
  KEY, firstname VARCHAR(30) NOT NULL, lastname
  VARCHAR(30) NOT NULL,
  email VARCHAR(50),
  reg_date TIMESTAMP
)

```

Berikut adalah contoh program untuk membuat tabel diatas:

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    // SQL untuk membuat tabel
    $sql = "CREATE TABLE MyGuests (

```

```

        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
        KEY, firstname VARCHAR(30) NOT NULL, lastname
        VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP
    );

    if (mysqli_query($conn, $sql)) {
        echo "Table MyGuests created successfully";
    } else {
        echo "Error creating table: " . mysqli_error($conn);
    }

    mysqli_close($conn);

```

?>

Memasukkan Data ke Database

Untuk memasukkan data kedalam tabel di database, terdapat beberapa aturan sintaks yang harus diikuti:

- Query SQL harus diberikan kutip dalam PHP.
- Nilai string didalam query SQL harus diberikan kutip.
- Nilai numeris tidak harus diberikan kutip.
- Nilai NULL tidak harus diberikan kutip.

Statement INSERT INTO digunakan untuk menambahkan baris pada tabel MySQL, query untuk menambahkan data tersebut adalah sebagai berikut:

```

INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)

```

adapun contoh kode program PHP untuk menambahkan data tersebut adalah sebagai berikut:

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

```

```

$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);

```

?>

Jika kita melakukan perintah INSERT pada tabel yang menggunakan field dengan opsi AUTO_INCREMENT, kita bisa mendapatkan ID dari baris yang terakhir diinputkan. Caranya menggunakan fungsi `mysqli_insert_id($conn)`; nilai kembalian dari fungsi ini akan memberikan id dari record atau baris terakhir yang diinputkan.

Untuk menambahkan record dengan jumlah yang banyak, kita bisa menggunakan fungsi `mysqli_multi_query($conn, $sql)`, berikut adalah contoh program untuk menambahkan record dengan jumlah yang banyak:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){

```



```

        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('John', 'Doe', 'john@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Mary', 'Moe', 'mary@example.com');";
    $sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Julie', 'Dooley', 'julie@example.com');";

    if (mysqli_multi_query($conn, $sql)) {
        echo "New records created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);

```

?>

Mengambil Data dari Database

Untuk mengambil data, statement SQL yang digunakan adalah SELECT nama_kolom() FROM nama_tabel, atau kita bisa menggunakan karakter * untuk memilih semua kolom yang ada pada tabel. Pengambilan data dengan kriteria tertentu bisa dilakukan dengan menggunakan statement WHERE setelah nama_tabel. Adapun contoh kode program untuk mengambil data dari database adalah sebagai berikut:

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

```

```

$sql = "SELECT id, firstname, lastname FROM
MyGuests"; $result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Menampilkan data pada setiap baris
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>

```

Menghapus Data dari Database

Statement DELETE digunakan untuk menghapus baris data atau records dari tabel. Clause WHERE digunakan untuk menspesifikasikan baris yang akan dihapus. Jika statement DELETE digunakan tanpa menggunakan klausa WHERE, maka semua record yang ada pada tabel akan dihapus.

	id	namadepan	namabelakang	email	reg_date
1	John	Doe	john@example.com	2014-10-22 14:26:15	
2	Mary	Moe	mary@example.com	2014-10-23 10:22:30	
3	Julie	Dooley	julie@example.com	2014-10-26 10:48:23	

Jika kita memiliki data seperti tabel “MyGuest” seperti terlihat diatas, kemudian kita ingin menghapus data dengan nama depan Julie, maka kode program untuk menghapus data tersebut adalah sebagai berikut:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);

```

```

// Memeriksa koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

// sql untuk menghapus record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);

```

?>

Update Data dalam Database

Untuk melakukan perubahan data di dalam database, statement UPDATE digunakan, yaitu sebagai berikut:

```

UPDATE nama_tabel
SET kolom1=nilai1, kolom2=nilai2, ...
WHERE kolom_penentu=nilai_penentu ...

```

Jika statement ini digunakan untuk update data ke dua pada tabel “MyGuest” diatas untuk mengganti nama belakang dari “Moe” menjadi “Doe”, maka kode program untuk update tersebut adalah sebagai berikut:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){

```

```
        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

    if (mysqli_query($conn, $sql)) {
        echo "Record updated successfully";
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }

    mysqli_close($conn);
```

?>

TUGAS

1. Buatlah database pada MySQL untuk menyimpan data-data pegawai beserta relasi tabelnya.
2. Buatlah program untuk menyimpan, menghapus dan mengubah data-data pegawai tersebut.

MODUL IV : Asynchronous JavaScript and XML (AJAX)

TUJUAN :

- Mahasiswa mampu membuat program dengan menggabungkan fitur PHP dan JavaScript.
- Mahasiswa mampu membuat program web yang dinamis.

AJAX digunakan untuk membuat halaman web yang cepat dan dinamis. AJAX mengizinkan halaman web diperbaharui secara asinkron oleh pertukaran data dalam jumlah kecil dengan aktifitas server dibelakang layar. Jadi, pembaharuan halaman web dilakukan tanpa merubah seluruh tampilan halaman. Halaman web yang klasik (tidak menggunakan AJAX) harus memuat seluruh halaman jika kontennya harus berubah. Contoh aplikasi penggunaan AJAX adalah seperti Google Maps, Gmail, Youtube, dan Tab Facebook. AJAX didasari oleh Standar internet dan menggunakan kombinasi dari:

- Objek XMLHttpRequest (untuk merubah data secara asinkron dengan sebuah server)
- JavaScript/DOM (untuk menampilkan atau berinteraksi dengan informasi)
- CSS (untuk memodelkan data)
- XML (sering digunakan sebagai format pengiriman data)

DOM adalah singkatan dari Document Object Model.

Kode program berikut adalah contoh penggunaan AJAX untuk membuat saran otomatis pada text input:

```
<html>
<head>
<script>
function showHint(str) {
    if (str.length == 0) {
        document.getElementById("txtHint").innerHTML =
            ""; return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("txtHint").innerHTML =
```

```

this.responseText;
        }
    };
    xmlhttp.open("GET", "gethint.php?q=" + str,
    true); xmlhttp.send();
    }
}
</script>
</head>
<body>

<p><b>Mulai ketikkan sebuah nama pada field input
dibawah:</b></p> <form>
Nama depan: <input type="text"
onkeyup="showHint(this.value)"> </form>
<p>Saran: <span id="txtHint"></span></p>
</body>
</html>

```

Penjelasan:

Pertama, program memeriksa apakah field input kosong (`str.length == 0`). Jika kosong, bersihkan konten dari placeholder `txtHint` dan keluar dari fungsi. Namun, jika field input tidak kosong, maka akan melakukan hal-hal sebagai berikut:

- Membuat sebuah objek XMLHttpRequest.
- Membuat fungsi untuk dieksekusi ketika jawaban server adalah siap (ready).
- Mengirim permintaan off ke file PHP (`gethint.php`) pada server.
- Memberitahukan bahwa parameter `q` ditambah ke URL (`gethint.php?q="+str`)
- Variabel `str` menyimpan konten field input.

Kode program diatas menggunakan file PHP dengan nama "gethint.php", kode program dari file tersebut adalah sebagai berikut:

```

<?php
    // Array with names
    $a[ ] = "Anna"; $a[
    ] = "Brittany"; $a[ ]
    = "Cinderella"; $a[ ]
    = "Diana"; $a[ ] =
    "Eva";
    $a[ ] = "Fiona";

```

```
$a[ ] = "Gunda";  
$a[ ] = "Hege";  
$a[ ] = "Inga";  
$a[ ] = "Johanna";  
$a[ ] = "Kitty";  
$a[ ] = "Linda";  
$a[ ] = "Nina";  
$a[ ] = "Ophelia";  
$a[ ] = "Petunia";  
$a[ ] = "Amanda";  
$a[ ] = "Raquel";  
$a[ ] = "Cindy";  
$a[ ] = "Doris";  
$a[ ] = "Eve";  
$a[ ] = "Evita";  
$a[ ] = "Sunniva";  
$a[ ] = "Tove";  
$a[ ] = "Unni";  
$a[ ] = "Violet";  
$a[ ] = "Liza";  
$a[ ] = "Elizabeth";  
$a[ ] = "Ellen";  
$a[ ] = "Wenche";  
$a[ ] = "Vicky";
```

```
// get the q parameter from  
URL $q = $_REQUEST["q"];
```

```
$hint = "";
```

```
// lookup all hints from array if $q is different from  
"" if ($q != "") {  
    $q = strtolower($q);  
    $len=strlen($q);  
    foreach($a as $name) {  
        if (stristr($q, substr($name, 0, $len)))  
        { if ($hint == "") {  
            $hint = $name;  
        } else {  
            $hint .= ", $name";  
        }  
    }  
}
```

```

    }
  }
}

// Output "no suggestion" if no hint was found or output correct
values echo $hint === "" ? "no suggestion" : $hint;
?>

```

AJAX dan MySQL

AJAX bisa digunakan untuk komunikasi interaktif dengan sebuah database. Dibawah ini akan diberikan contoh kode program untuk mengakses database menggunakan AJAX, namun sebelum itu, perlu dibuat tabel “user” pada database “my_db” dan diisi dengan data sebagai berikut:

id	FirstName	LastName	Age	Hometown	Job
1	Peter	Griffin	41	Quahog	Brewery
2	Lois	Griffin	40	Newport	Piano Teacher
3	Joseph	Swanson	39	Quahog	Police Officer
4	Glenn	Quagmire	41	Quahog	Pilot

Berikut adalah contoh program tersebut:

```

<html>
<head>
<script>
function showUser(str) {
  if (str == "") {
    document.getElementById("txtHint").innerHTML =
    ""; return;
  } else {
    if (window.XMLHttpRequest) {
      // code for IE7+, Firefox, Chrome, Opera,
      // Safari xmlhttp = new XMLHttpRequest();
    } else {
      // code for IE6, IE5
      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
  }
}

```



```

        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("txtHint").innerHTML =
                    this.responseText; }
        };
        xmlhttp.open("GET","getuser.php?q="+str,true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

<form>
<select name="users" onchange="showUser(this.value)">
    <option value="">Select a person:</option>
    <option value="1">Peter Griffin</option> <option
    value="2">Lois Griffin</option>
    <option value="3">Joseph Swanson</option>
    <option value="4">Glenn Quagmire</option>
</select>
</form>
<br>
<div id="txtHint"><b>Person info will be listed here...</b></div>

</body>
</html>

```

Penjelasan kode:

Pertama, memeriksa apakah ada data orang yang terpilih. Jika tidak ada maka (str == ""), bersihkan konten dari txtHint dan keluar dari fungsi. Jika ada data orang terpilih, maka akan melakukan hal berikut:

- Membuat objek XMLHttpRequest.
- Membuat fungsi untuk dieksekusi ketika jawaban server adalah siap (ready).
- Kirim permintaan off untuk sebuah file pada server.
- Memberitahukan bahwa sebuah parameter (q) ditambahkan ke URL (dengan konten dropdown list).

Contoh program diatas menggunakan file "getuser.php". File tersebut berisi kode program untuk melakukan query pada database MySQL, Kode program dari file tersebut adalah sebagai berikut:

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
    width: 100%;
    border-collapse: collapse;
}

table, td, th {
    border: 1px solid black;
    padding: 5px;
}

th {text-align: left;}
</style>
</head>
<body>

<?php
    $q = intval($_GET['q']);

    $con = mysqli_connect('localhost','peter','abc123','my_db'); if
    (!$con){
        die('Could not connect: ' . mysqli_error($con));
    }

    mysqli_select_db($con,"ajax_demo");
    $sql="SELECT * FROM user WHERE id =
    ".$q.""; $result = mysqli_query($con,$sql);

    echo "<table>
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Age</th>
            <th>Hometown</th>
            <th>Job</th>
        </tr>";

```

```

while($row = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>" . $row['FirstName'] . "</td>";
    echo "<td>" . $row['LastName'] . "</td>";
    echo "<td>" . $row['Age'] . "</td>";
    echo "<td>" . $row['Hometown'] . "</td>";
    echo "<td>" . $row['Job'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close($con);
?>
</body>
</html>

```

Penjelasan dari kode program:

Ketika query dikirimkan dari JavaScript ke file PHP, hal-hal berikut terjadi:

- PHP membuka koneksi ke server MySQL.
- Menemukan data orang.
- Tabel HTML dibuat, menambahkan data ke tabel tersebut, dan mengirimkannya kembali ke placeholder "txtHint".

AJAX dan XML

AJAX bisa digunakan untuk komunikasi interaktif dengan file XML. Berikut adalah contoh program AJAX yang digunakan untuk menampilkan data XML:

```

<html>
<head>
<script>
function showCD(str) {
    if (str=="") {
        document.getElementById("txtHint").innerHTML="";
        return;
    }
    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera,
        // Safari
        xmlhttp=new XMLHttpRequest();
    } else { // code for IE6, IE5

```

```

        xmlhttp=new XMLHttpRequest("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function() {
        if (this.readyState==4 && this.status==200) {
            document.getElementById("txtHint").innerHTML=this.responseText;
        }
    }
    xmlhttp.open("GET","getcd.php?q="+str,true);
    xmlhttp.send();
}
</script>
</head>
<body>

<form>
Select a CD:
<select name="cds" onchange="showCD(this.value)">
<option value="">Select a CD:</option>
<option value="Bob Dylan">Bob Dylan</option>
<option value="Bee Gees">Bee Gees</option>
<option value="Cat Stevens">Cat Stevens</option>
</select>
</form>
<div id="txtHint"><b>CD info will be listed here...</b></div>

</body>
</html>

```

Penjelasan kode program:

Fungsi showCD() melakukan hal-hal berikut:

- Memeriksa apakah ada CD yang terpilih.
- Membuat sebuah objek XMLHttpRequest.
- Membuat fungsi yang akan dieksekusi ketika jawaban server adalah siap (ready).
- Mengirimkan permintaan off untuk sebuah file pada server.
- Memberitahukan bahwa sebuah parameter (q) ditambahkan ke URL (dengan konten dari list dropdown).

Pada kode program tersebut, terdapat file "getcd.php". Kode program dari file tersebut memuat dokumen XML "cd_catalog.xml", menjalankan query terhadap file XML dan mengembalikan hasilnya. File "cd_catalog.xml" bisa didownload melalui link

https://www.w3schools.com/php/cd_catalog.xml. Berikut adalah kode program dari file "getcd.php":

```
<?php
    $q=$_GET["q"];

    $xmlDoc = new DOMDocument();
    $xmlDoc->load("cd_catalog.xml");

    $x=$xmlDoc->getElementsByTagName('ARTIST');

    for ($i=0; $i<=$x->length-1; $i++) {
        //Hanya memproses element node
        if ($x->item($i)->nodeType==1){
            if ($x->item($i)->childNodes->item(0)->nodeValue == $q)
                { $y=($x->item($i)->parentNode);
                }
        }
    }

    $cd=($y->childNodes);

    for ($i=0;$i<$cd->length;$i++) {
        //Hanya memproses element node
        if ($cd->item($i)->nodeType==1){
            echo("<b>" . $cd->item($i)->nodeName . "</b> ");
            echo($cd->item($i)->childNodes->item(0)-
            >nodeValue); echo("<br>");
        }
    }
?>
```

Ketika query CD dikirimkan dari JavaScript ke halaman PHP, hal-hal berikut terjadi:

- PHP membuat sebuah objek XML DOM.
- Menemukan semua element yang sesuai dengan nama yang dikirimkan dari JavaScript.
- Mengeluarkan output berupa informasi album (mengirimkan ke placeholder "txtHint")

AJAX – Pencarian Langsung (Live Search)

AJAX bisa digunakan untuk membuat pencarian keyword yang lebih interaktif dan user-friendly. Berikut adalah contoh program untuk melakukan pencarian melalui query pada sebuah file XML "links.xml". File ini bisa didownload pada link: <https://www.w3schools.com/php/links.xml> . Kode program dari contoh program pencarian adalah sebagai berikut:

```
<html>
<head>
<script>
function showResult(str) {
    if (str.length==0) {
        document.getElementById("livesearch").innerHTML="";
        document.getElementById("livesearch").style.border="0px";
        return;
    }
    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera,
        // Safari xmlhttp=new XMLHttpRequest();
    } else { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function() {
        if (this.readyState==4 && this.status==200) {
            document.getElementById("livesearch").innerHTML=this.responseText;
            document.getElementById("livesearch").style.border="1px solid
            #A5ACB2"; }
        }
    xmlhttp.open("GET","livesearch.php?q="+str,true);
    xmlhttp.send();
}
</script>
</head>
<body>

<form>
<input type="text" size="30" onkeyup="showResult(this.value)">
<div id="livesearch"></div>
</form>

</body>
</html>
```

Penjelasan kode program:

Jika field input kosong (str.length==0), fungsi akan membersihkan konten dari placeholder livesearch dan keluar dari fungsi. Jika field input tidak kosong, fungsi showResult() melakukan hal-hal berikut:

- Membuat objek XMLHttpRequest.
- Membuat fungsi yang akan dieksekusi ketika jawaban server adalah siap (ready).
- Mengirimkan permintaan off untuk sebuah file pada server.
- Memberitahukan bahwa sebuah parameter (q) ditambahkan ke URL (dengan konten field input)

Contoh kode program diatas menggunakan file "livesearch.php". Kode program dari file tersebut adalah sebagai berikut:

```
<?php
$xmlDoc=new DOMDocument();
$xmlDoc->load("links.xml");

$x=$xmlDoc->getElementsByTagName('link');

//Mengambil parameter q dari URL
$q=$_GET["q"];

//lookup semua link dari file xml jika panjang dari
q>0 if (strlen($q)>0) {
    $hint="";
    for($i=0; $i<($x->length); $i++){
        $y=$x->item($i)->getElementsByTagName('title');
        $z=$x->item($i)->getElementsByTagName('url');
        if ($y->item(0)->nodeType==1){
            //menemukan sebuah link yang sesuai dengan text yang
dicari

            if (stristr($y->item(0)->childNodes->item(0)-
                >nodeValue,$q)){ if ($hint==""){
                $hint="<a href=" .
                $z->item(0)->childNodes->item(0)->nodeValue .
                "' target='_blank'>" .
                $y->item(0)->childNodes->item(0)->nodeValue . "</a>";
```

```

    }else{
    $hint=$hint . "<br /><a href=" .
    $z->item(0)->childNodes->item(0)->nodeValue .
    "' target='_blank'>" .
    $y->item(0)->childNodes->item(0)->nodeValue .
    "</a>"; }
    }
  }
}

// Mengatur output menjadi "no suggestion" jika tidak ada petunjuk
yang ditemukan atau nilai yang sesuai
if ($hint==""){
    $response="no suggestion";
}else{
    $response=$hint;
}

//output the response
echo $response;
?>

```

Penjelasan dari kode program:

Jika ada text yang dikirimkan dari JavaScript (`strlen($q) > 0`), hal-hal berikut akan dilakukan:

- Memuat file XML menjadi objek XML DOM baru.
- Perulangan melalui semua element untuk menemukan kata yang sesuai dengan text yang dikirimkan dari JavaScript.
- Menugaskan URL yang benar dan judul dalam variabel `$response`. Jika lebih dari satu yang sesuai, semua yang sesuai akan ditambahkan ke variabel tersebut.
- Jika tidak ada yang sesuai, variabel `$response` akan dimasukkan nilai text "no suggestion".

AJAX Polling

AJAX juga baik digunakan untuk menampilkan secara langsung hasil polling. Berikut adalah contoh kode program untuk melakukan polling:


```

<html>
<head>
<script>
    function getVote(int) {
        if (window.XMLHttpRequest) {
            // code for IE7+, Firefox, Chrome, Opera,
            // Safari xmlhttp=new XMLHttpRequest();
        } else { // code for IE6, IE5
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange=function() {
            if (this.readyState==4 && this.status==200) {
                document.getElementById("poll").innerHTML=this.responseText;
            }
        }
        xmlhttp.open("GET","poll_vote.php?vote="+int,true);
        xmlhttp.send();
    }
</script>
</head>
<body>

<div id="poll">
<h3>Apakah kamu menyukai AJAX atau PHP?
</h3> <form>
Yes:
<input type="radio" name="vote" value="0"
onclick="getVote(this.value)"> <br>No:
<input type="radio" name="vote" value="1" onclick="getVote(this.value)">
</form>
</div>

</body>
</html>

```

Fungsi getVote() melakukan hal-hal sebagai berikut:

- Membuat sebuah objek XMLHttpRequest
- Membuat fungsi untuk dieksekusi ketika jawaban server adalah siap (ready).
- Mengirimkan permintaan off untuk sebuah file pada server.

- Memberitahukan bahwa parameter (vote) ditambahkan ke URL (dengan nilai opsi yes atau no)

Kode program diatas menggunakan file “poll_vote.php”. Kode program dari file tersebut adalah sebagai berikut:

```
<?php
    $vote = $_REQUEST['vote'];

    //mendapatkan konten dari textfile
    $filename = "poll_result.txt";
    $content = file($filename);

    //meletakkan konten dalam array
    $array = explode("||", $content[0]);
    $yes = $array[0];
    $no = $array[1];

    if ($vote == 0){
        $yes = $yes + 1;
    }
    if ($vote == 1){
        $no = $no + 1;
    }

    //menyisipkan vote ke file txt
    $insertvote = $yes."||".$no;
    $fp = fopen($filename,"w");
    fputs($fp,$insertvote);
    fclose($fp);
?>

<h2>Result:</h2>
<table>
<tr>
<td>Yes:</td>
<td>

' height='20'>
<?php echo(100*round($yes/($no+$yes),2)); ?>%

```

```

</td>
</tr>
<tr>
<td>No:</td>
<td>
' height='20'>
<?php echo (100*round($no/($no+$yes),2)); ?>%
</td>
</tr>
</table>

```

Nilai dikirim dari JavaScript dan hal-hal berikut akan terjadi:

- Mengambil konten dari file “poll_result.txt”.
- Meletakkan konten ke dalam variabel dan menambahkan satu ke variabel yang terpilih.
- Menuliskan hasil ke file “poll_result.txt”
- Mengeluarkan representasi grafis dari hasil polling.

File text (poll_result.txt) adalah tempatn menyimpan data polling. Data tersebut disimpan dalam bentuk berikut:

0||0

Bilangan pertama merepresentasikan voting “Yes” sedangkan bilangan kedua merepresentasikan voting “No”. Jangan lupa untuk mengizinkan web browser merubah text file.

TUGAS

1. Buatlah program web untuk menyimpan data survei yang disimpan didalam database. Carilah library javascript diinternet untuk menampilkan grafik, kemudian rangkum data-data hasil survei kedalam bentuk grafik.
2. Misalkan anda seorang programer yang diminta untuk membuat program web untuk melakukan pemungutan suara pada PEMILU. Jumlah calon yang akan dipilih adalah lima orang. Buatlah program web untuk pemilihan tersebut, dimana setiap pemilih hanya bisa memilih satu kali.

Referensi:

PHP Tutorial, <https://www.w3schools.com/php/>