

DIKTAT

PEMROGRAMAN PHP dan MYSQL



STMIK MUHAMMADIYAH JAKARTA

Setelah mempelajari DIKTAT Pemrograman PHP dan MYSQL, Mahasiswa diharapkan akan dapat:

- Mengetahui Web Server Apache, PHP, dan MYSQL
- Memahami Dasar-dasar PHP
- Memahami dan Menggunakan Pemilihan, Perulangan, serta Array dalam Kode Pemrograman PHP
- Memahami PHP lebih lanjut
- Memahami Dasar-dasar MYSQL dalam mengelola Database
- Melakukan kolaborasi antara PHP dan MYSQL

Modul 1

Pokok Bahasan:

- Mengetahui Web Server Apache
- Mengetahui PHP
- Mengetahui MySQL
- Mengetahui XAMPP
- Mengetahui Apache, PHP, dan MySQL dengan memanfaatkan XAMPP

Mengenal Apache, PHP dan MySQL

Indikator Keberhasilan:

- Dapat menjelaskan perbedaan halaman web yang bersifat statis dan dinamis
- Dapat menjelaskan fungsi dan kegunaan Apache, PHP dan MySQL
- Dapat menginstall dan menjalankan Apache, PHP dan MySQL dengan memanfaatkan XAMPP

1.1 Mengenal Web Server Apache

Halaman web yang dibuat menggunakan kode HTML bersifat *client-side*. Artinya, isi halaman web tersebut diolah di sisi client. Ketika seseorang membuka halaman web, maka server yang menyediakan halaman web tersebut akan langsung mengirimkan halaman web itu kepada si peminta tanpa diolah terlebih dahulu.

Hal tersebut berbeda dengan halaman web yang dibuat dengan bantuan PHP. Halaman web yang dibuat dengan bantuan PHP dapat mengakses dan memanfaatkan database sehingga lebih bersifat dinamis. Isi halaman web-nya dapat berubah tanpa perlu diprogram ulang. Selain itu, halaman web-nya akan diolah terlebih dahulu di sisi server sebelum di kirim ke si pemintanya (*server-side*). Oleh karena sifatnya yang *server-side*, maka dibutuhkan sebuah *web server*. Dalam hal ini, web server yang banyak digunakan untuk PHP adalah Apache. Tugas utama apache adalah menghasilkan halaman web yang benar kepada client peminta, berdasarkan kode PHP yang dituliskan oleh pembuat web. Jika diperlukan, juga berdasarkan kode PHP yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misalkan dalam MySQL) untuk mendukung isi halaman web yang dihasilkan.

1.2 Mengenal PHP

PHP adalah singkatan dari *PHP: hypertext preprocessor*. PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan anda untuk membuat halaman web yang bersifat dinamis.

PHP dapat dijalankan pada berbagai macam system operasi, misalkan: Windows, LINUX, dan Mac OS. Selain Apache, PHP juga mendukung beberapa web server lain, misalkan Microsoft IIS, Caudium, PWS dan lain-lain.

Sistem manajemen database yang sering digunakan bersama PHP adalah MySQL. Namun, PHP juga mendukung system manajemen database Oracle, Microsoft Access, Interbase, dBase, PostgreSQL, dan lain-lain.

Hingga kini, PHP sudah berkembang hingga versi 5. PHP 5 mendukung penuh *Object-Oriented programming (OOP)*, integrasi XML, mendukung semua ekstensi terbaru MySQL, serta ratusan peningkatan lainnya, dibandingkan versi sebelumnya.

PHP juga bersifat open source sehingga setiap orang dapat menggunakannya secara gratis.

1.3 Mengetahui MySQL

MySQL adalah sebuah system manajemen database bersifat open source. MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia.

MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. MySQL dapat dimanfaatkan untuk menambah, mengubah, dan menghapus data yang berada di dalam database.

MySQL merupakan system manajemen database yang bersifat relational. Artinya data-data yang dikelola dalam database akan diletakkan pada beberapa table yang terpisah sehingga manipulasi akan menjadi lebih cepat.

MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan sangat besar. MySQL juga dapat dijalankan perintah-perintah *Structured Query Language (SQL)* untuk mengelola database-database relational yang ada di dalamnya.

Modul 2

Pokok Bahasan:

- Menyisipkan kode program PHP
- Variable
- Tipe data
- Konstanta
- Operator

Dasar-Dasar PHP

Indikator Keberhasilan:

- Dapat menggunakan tag `<?PHP` dan `?>` serta perintah `echo` untuk menampilkan teks dan isi variable dan konstanta
- Dapat membuat dan menggunakan variable atau konstanta
- Dapat menggunakan beberapa operator serta mengerti urutan pengerjaannya

2.1 Menyisipkan Kode Program PHP

Kode-kode PHP dapat dibuat dalam sebuah file tersendiri atau disisipkan ke dalam HTML. Bagian dalam file HTML yang berisi kode program PHP diawali tag "`<?php`" dan diakhiri tag "`?>`", contoh:

```
<?php
?>
```

atau bisa juga dalam bentuk singkatannya, yaitu menggunakan "`<?>`" dan "`?>`", contoh:

```
<?
?>
```

Setiap baris kode program PHP selalu diakhiri dengan karakter titik koma "`;`", contoh:

```
<?php
    echo "ini file PHP pertama ku";
?>
```

dapat juga ditambahkan baris-baris komentar ke dalam kode program PHP yang dibuat. Baris-baris komentar hanya ditujukan untuk menjelaskan isi kode program, dan tidak akan mempengaruhi tampilan web. Sebuah baris komentar diawali dengan karakter "`//`". Juga bisa membuat beberapa baris komentar sekaligus, yaitu dengan diawali tanda "`/*`" dan diakhiri dengan tanda "`*/`", contoh:

```
<?php
    echo "ini file PHP pertama ku";
    // ini satu baris komentar
    /* ini beberapa
        baris komentar */
?>
```

File PHP dapat disisipkan ke dalam HTML.

Perintah **echo** dalam PHP biasa digunakan untuk menampilkan sebuah teks ke dalam halaman web. Perintah **echo** juga dapat digunakan untuk menampilkan tag-tag HTML untuk selanjutnya akan diterjemahkan sebagai tag HTML yang biasa, contoh:

```
<html>
<head>
<title>contoh 1</title>
<body>
<?php
    echo "ini contoh teks";
    echo "<br>";
?>
</body>
</html>
```

Kode program PHP pada contoh di atas akan memberikan hasil yang sama dengan contoh berikut:

```
<html>
<head>
<title>contoh 1</title>
<body>
ini contoh teks
<br>
</body>
</html>
```

Halaman web yang dibuat dengan bantuan PHP, disimpan dalam file berekstensi **.php**.

Tugas 1:

Langkah-langkah

1. Jalankan Notepad
2. ketik baris-baris kode program di bawah ini

```
<html>
<head>
<title>tugas 1</title>
<body>
<?php
    echo "<br>";
    echo "teks ini muncul dari dalam kode PHP";
    // ini conoth baris komentar
    /* ini contoh beberapa
       baris komentar */
    echo "<hr>";
?>
<br>
</body>
</html>
```

3. Simpan file diatas, klik menu **File** → **Save as** pada menu Notepad. **File name = latihan1.php**. **Save as type = all files**
4. Jalankan web server Apache menggunakan control panel XAMPP
5. Jalankan browser Internet Explorer atau Mozilla Firefox, lalu ketikkan alamat pada address :
localhost/nama_folder/latihan1.php

2.2 Variable dan Tipe Data

Variable adalah tempat yang digunakan oleh PHP untuk menyimpan nilai, data, atau informasi yang kita olah dan gunakan selama halaman web kita jalankan. Nama setiap variabel dalam kode php di tulis dengan karakter *dollar* "\$". Nama variabel bersifat *casesensitive*, artinya varibel dengan nama **\$baju** akan dianggap berbeda dengan variabel bernama **\$BAJU**.

Untuk mengisi atau meletakkan sebuah nilai ke dalam variabel, cukup gunakan tanda sama dengan "=" . Misalkan kita ingin meletakkan sebuah informasi judul buku "**Pemrograman web**" ke dalam sebuah

variabel bernama **\$judul**, maka cukup di tuliskan kode program sebagai berikut:

```
$judul = "Pemrograman web";
```

Untuk menampilkan isi sebuah variabel, cukup gunakan perintah **echo** seperti berikut:

```
Echo $judul;
```

Setiap variabel memiliki tipe data yang menentukan nilai seperti apa saja yang dapat ditampung oleh variabel tersebut. PHP memiliki beberapa jenis tipe data, beberapa di antaranya adalah:

Tipe Data	Keterangan
Integer	Untuk menyimpan angka-angka bilangan bulat tanpa ada nilai decimal. Contoh 123
Double	Untuk menyimpan angka-angka dengan nilai decimal, contoh, 12,25
Boolean	Untuk menyimpan nilai logika benar (TRUE) atau salah (FALSE)
String	Untuk menyimpan deretan karakter. Contoh: "Buku ini bagus sekali".

Setiap variabel dalam PHP memiliki tipe data tertentu, namun variabel tersebut tidak perlu didefinisikan terlebih dahulu tipe data apa yang digunakan. Sebab, ketika menuliskan nama variabel beserta isinya, variabel tersebut langsung memilih dan menggunakan tipe data yang cocok bagi nilai yang diisikan. Contoh:

```
$judul="Pemrograman PHP"  
$harga=1000  
$harga="Seribu"
```

Pada contoh diatas, variabel **\$judul** secara otomatis akan memiliki tipe data string karena isinya adalah deretan karakter, sedangkan pada baris kedua, variabel **\$harga** secara otomatis akan bertipe integer Karena isinya adalah bilangan bulat.

Sebuah variabel dapat saja langsung berganti tipe data secara otomatis, ketika isinya berubah. Pada baris ketiga dalam contoh di atas, variabel tipe data **\$harga** berubah dari yang tadinya bertipe integer menjadi string.

2.3 Konstanta

Konstanta mirip seperti variabel yang juga digunakan untuk menyimpan nilai, data, atau informasi yang kita olah dan digunakan selama halaman web kita jalankan. Namun, isi sebuah konstanta selalu tetap dan tidak berubah.

Nama sebuah konstanta biasanya dituliskan dalam huruf-huruf capital dan tidak perlu diawali dengan karakter *dollar* "\$". Konstanta dibuat menggunakan sebuah fungsi bernama **define()** dengan aturan penulisan **define(NAMAKONSTANTA,NILAINYA)**. Misalkan ingin membuat sebuah konstanta bernama **HARGA**, yang berisi nilai 1000, maka gunakan fungsi **define()** sebagai berikut:

```
Define("SERIBU",1000);
```

Untuk menampilkan isi sebuah konstanta, gunakan perintah `echo` seperti dibawah ini:

```
Echo HARGA;
```

2.4 Operator

PHP memiliki berbagai macam operator untuk mengoperasikan setiap variabel yang dimilikinya. Beberapa operator sering digunakan adalah operator aritmatika, operator perbandingan, dan operator logika.

Operator aritmatika digunakan untuk melakukan operasi aritmatika terhadap variabel-variabel, misalkan penambahan, pengurangan, dan lain-lain. Berikut penjelasannya:

Operator	Keterangan	Contoh
+	Penambahan	$\$c = \$a + \$b$

		Jika \$a=5 dan \$b=7 maka \$c=12
-	Pengurangan	\$c=\$a - \$b Jika \$a=5 dan \$b=7 maka \$c=-2
*	Perkalian	\$c=\$a * \$b Jika \$a=5 dan \$b=7 maka \$c=35
/	Pembagian	\$c=\$a + \$b Jika \$a=12 dan \$b=4 maka \$c=3
%	Modulus (sisa pembagian)	\$c=\$a % \$b Jika \$a=7 dan \$b=2 maka \$c=1

Operator Perbandingan digunakan untuk melakukan operator perbandingan terhadap variabel-variabel, hasilnya berupa nilai Boolean benar(TRUE) atau salah (FALSE). Berikut penjelasannya:

Operator	Keterangan	Contoh
==	Sama dengan	\$a == \$b Hasilnya adalah TRUE jika isi \$a sama dengan \$b
!=	Tidak sama dengan	\$a != \$b Hasilnya adalah TRUE jika isi \$a tidak sama dengan \$b
<>	Tidak sama dengan	\$a <> \$b Hasilnya adalah TRUE jika isi \$a tidak sama dengan \$b
<	Kurang dari	\$a < \$b Hasilnya adalah TRUE jika

		isi \$a kurang dari \$b
>	Lebih dari	\$a > \$b Hasilnya adalah TRUE jika isi \$a lebih dari \$b
<=	Kurang dari atau sama dengan	\$a <= \$b Hasilnya adalah TRUE jika isi \$a kurang dari atau sama dengan \$b
>=	Lebih dari atau sama dengan	\$a >= \$b Hasilnya adalah TRUE jika isi \$a lebih dari atau sama dengan \$b

Operator Logika digunakan untuk melakukan operasi logika terhadap variabel-variabel yang memiliki tipe data Boolean, hasilnya juga berupa nilai Boolean benar (TRUE) atau salah (FALSE). Berikut ini penjelasannya:

Operator	Keterangan	Contoh
and	Dan	\$a and \$b Hasilnya adalah TRUE jika isi \$a=TRUE dan \$b=TRUE
Or	Atau	\$a or \$b Hasilnya adalah TRUE jika salah satu atau kedua isi \$a=TRUE dan \$b=TRUE
xor	Atau yang eksklusif	\$a xor \$b Hasilnya adalah TRUE jika salah satu isi \$a=TRUE dan \$b=TRUE, namun hasilnya akan FALSE jika keduanya sama-sama TRUE
!	Negasi	!\$a Hasilnya adalah TRUE jika

		isi \$a=FALSE, dan hasilnya FALSE jika\$a bernilai TRUE
&&	Dan	\$a && \$b Hasilnya adalah TRUE jika isi \$a=TRUE dan \$b=TRUE
 	Atau	\$a \$b Hasilnya adalah TRUE jika isi \$a=TRUE atau \$b=TRUE

Jika sebuah operasi diletakkan dalam tanda kurung, operasi tersebut akan dikerjakan terlebih dahulu, misalkan $(1+2)*3$ hasilnya adalah 9.

Namun, jika beberapa operasi diletakkan dalam satu penulisan tanpa tanda kurung, maka urutan pengerjaannya dapat berbeda, misalkan $1+2*3$ hasilnya 7. Hal itu terjadi karena operasi perkalian dikerjakan lebih dahulu sebelum operator penambahan " + ". Berikut ini pengerjaan operator mulai dari yang paling dahulu di kerjakan.

Operator	Keterangan
!	
*, /, %	Ketiganya memiliki derajat yang sama
+, -	Keduanya memiliki derajat yang sama
<, <=, >, >=	Keempatnya mempunyai derajat yang sama
==, !=	Keduanya mempunyai derajat yang sama
&&	
 	
And	
Xor	
Or	

Exercise II

1. Tugas 1

Simpan dengan nama tugas1.php

```
<html>
<head>
<title>tugas 1</title>
<body>
<?php
    $judul="Pemrograman PHP";
    echo "teks ini adalah isi variabel judul= ";
    echo $judul;
    echo "<br>",

?>
<br>
<?php
    $harga="Seribu";
    echo "teks ini adalah isi variabel harga= ";
    echo $harga;
?>
</body>
</html>
```

2. Tugas 2

simpan dengan nama tugas2.php

```
<html>
<head>
<title>tugas 2</title>
<body>
<?php
    define("JUDUL", "Pemrograman PHP");
    echo "teks ini adalah isi konstanta JUDUL= ";
    echo JUDUL;
    echo "<br>";
```

```
define("HARGA",1000);  
echo "teks ini adalah isi konstanta HARGA + 100= ";  
echo HARGA+100;  
?>  
</body>  
</html>
```

3. Tugas 3

simpan dengan nama tugas3.php

```
<html>  
<head>  
<title>tugas 3</title>  
<body>  
<?php  
    $a=5;  
    $b=$a+3;  
    echo $a;  
    echo "<br>"; echo $b;  
  
    $c="buku ini bagus" + $a;  
    echo "<br>.$c;  
  
?>  
</body>  
</html>
```

4. Tugas 4

simpan dengan nama tugas4.php

```
<html>  
<head>  
<title>tugas 4</title>  
<body>  
<?php  
    $a=5;  
    $b=$a+3;
```

```
define("NILAI1",100);  
define("NILAI2",100);  
echo $a;  
echo "<br>".$b;  
echo "<br>".$b+NILAI1;  
echo "<br>".(NILAI1+NILAI2);  
?>  
</body>  
</html>
```

Modul 3

Pokok Bahasan:

- Pemilihan
- Perulangan
- Array
- Fungsi Array()
- Fungsi List()
- Array Multidimensi

Pemilihan, Perulangan, dan Array

Indikator Keberhasilan:

- Dapat menggunakan struktur pemilihan dalam kode program PHP
- Dapat menggunakan struktur perulangan dalam kode program PHP
- Dapat membuat dan memanfaatkan array dalam kode program PHP

3.1 Pemilihan

Pemilihan adalah sebuah kondisi di mana satu atau lebih baris kode program (*statement*) akan di eksekusi jika kondisi tertentu terpenuhi. PHP memiliki 3 jenis struktur pemilihan, yaitu **if...**, **if... else...**, dan **switch**. Perhatikan ilustrasi di bawah ini:

1. jika Adit punya uang 5000 ia membeli pesawat
2. jika Adit punya uang 2000 ia membeli mobil
3. jika Adit punya uang 1000 ia membeli motor
4. jika Adit punya uang 500 ia membeli sepeda
5. selainnya, Adit tidak akan membeli apa-apa

Struktur pemilihan **if...** akan memeriksa kondisi **benar** atau **salah** dari suatu ekspresi, kemudian menentukan langkah yang perlu diambil **jika uji kondisi terpenuhi**.

Struktur pemilihan **if...** Memiliki aturan penulisan sebagai berikut:

```
If (<ujikondisi>
    {
        <baris-baris program terpenuhi>;
    }
```

Jika poin no 1 dalam ilustrasi di atas akan dituliskan dalam struktur pemilihan **if...**, maka dapat digunakan pernyataan **if...** sebagai berikut.

```
If(uangAdit == 5000)
    {
        Adit_beli_pesawat;
    }
```

Struktur pemilihan **if... else...** akan memeriksa kondisi **benar** atau **salah** dari suatu ekspresi, kemudian menentukan langkah yang perlu diambil **jika uji kondisi terpenuhi** dan **jika tidak terpenuhi**. Jika pemeriksaan kondisi terpenuhi, maka akan dikerjakan baris-baris kode program di dalam bagian **if...** Jika pemeriksaan kondisi tidak terpenuhi, akan dikerjakan baris-baris kode program di dalam bagian **else...**

Struktur pemilihan **if... else...** memiliki aturan penulisan sebagai berikut:

```
if (<ujikondisi>
    {
        <baris-baris program terpenuhi>;
    }
else
    {
        <baris-baris program terpenuhi>;
    }
```

Jika poin no 4 dan no 5 dalam ilustrasi di atas digabung, maka dapat digunakan pernyataan **if... else...** sebagai berikut:

```
if uangAdit == 5000)
    {
        Adit_beli_sepeda;
    }
else
    {
        Adit_tidak_beli_apa-apa;
    }
```

PHP mendukung penggunaan struktur pemilihan bersarang. Yaitu, suatu kondisi dimana dalam struktur pemilihan ada struktur pemilihan yang lain. Struktur pemilihan ada struktur pemilihan lain. Struktur pemilihan bersarang dapat dituliskan dalam bentuk struktur pemilihan **if... else...** dalam sebuah struktur pemilihan **if... else...** yang lain.

Misalkan:

1. Jika jumlah uang Andri sama dengan Gofu, maka andri dan Gofu beli mobil.
2. Jika jumlah uang Andri tidak sama dengan Gofu, maka lakukan salah satu dari dua kondisi berikut:
 - a. Jika jumlah uang andri lebih dari Gofu, maka andri saja yang membeli mobil.
 - b. Selain itu, maka Gofu saja yang membeli mobil.

Penjelasan ilustrasi Andri dan Gogo di atas adalah sebagai berikut:

Pertama-tama akan di periksa dahulu apakah kondisi jumlah uang Andri sama dengan Gogo. Jika ya, andri dan Gogo membeli mobil. Namun, jika jumlah uangnya tidak sama, periksa lagi kondisi uang keduanya. Jika jumlah uang Andri lebih dari uang Gogo, Andri saja yang membeli mobil. Namun, jika jumlah uang andri kurang dari Gogo maka Gogo yang membeli mobil.

Ilustrasi tersebut dapat dituliskan sebagai berikut:

```
If(uangAndri == uangGogo)
    {
        Andri_dan_Gogo_beli_mobil:
    }
else
    {
        if(uangAndri > uangGogo)
            {
                Andri_saja_yang_beli_mobil;
            }
        else
            {
                Gogo_saja_yang_beli_mobil;
            }
    }
}
```

Perhatikan kembali ilustrasi uang Adit di atas. Pada ilustrasi tersebut terdapat 5 buah kondisi pemilihan. Jika ingin menuliskannya satu per satu menggunakan

akan struktur pemilihan **if...** atau **if... else...**, maka akan sangat tidak efektif dan efisien. Oleh karena itu, PHP menyediakan sebuah struktur pemilihan lain, yaitu **switch**.

Struktur pemilihan dengan **switch** digunakan untuk melakukan pemilihan terhadap beberapa kondisi sekaligus. **Switch** memiliki aturan penulisan sebagai berikut:

```
switch(<ujikondisi>
{
    case <kondisi_1>;
        <baris-baris kode jika kondisi 1 terpenuhi>;
        break;
    case <kondisi_2>;
        <baris-baris kode jika kondisi 2 terpenuhi>;
        break;
    default;
        <baris-baris kode jika kondisi 1 terpenuhi>;
}
```

Oleh karena itu, kelima kondisi yang kita temui pada ilustrasi uang Adit di atas, dapat dituliskan dalam struktur pemilihan switch seperti berikut:

```
switch(uangAdit)
{
    case 5000;
        Adit_beli_pesawat;
        Break;
    case 2000;
        Adit_beli_mobil;
        break;
    case 1000;
        Adit_beli_motor;
        break;

    case 500;
        Adit_beli_sepeda;
    default:
        adit_tidak_beli_apa_apa;
}
```

3.2 Perulangan

Perulangan adalah sebuah kondisi di mana satu atau beberapa baris kode program (*statement*) dieksekusi secara berulang-ulang. Struktur perulangan yang paling sering digunakan adalah **while** dan **for**.

Perulangan dengan **while** membuat PHP akan menguji kondisi perulangan yang digunakan terlebih dahulu. Jika uji kondisi yang dimaksud terpenuhi, baris-baris kode program di dalam perulangan **while** akan diulang terus-menerus hingga hasil uji kondisi tidak terpenuhi lagi. Aturan penulisan perulangan dengan **while** adalah sebagai berikut:

```
While (<ujikondisi>
    {
        <baris-baris kode di dalam while>;
    }
```

Contoh: ingin mencetak kalimat "**pemrograman PHP**" sebanyak 10 kali menggunakan perulangan while, maka dapat kita tuliskan sebagai berikut:

/ variable \$iter dibawah ini digunakan untuk mengontrol banyaknya perulangan. Awalnya \$iter=0 */*

```
$iter = 0
while($iter < 10)
    {
        echo "pemrograman PHP";
        $iter =$iter + 1;
    }
```

Struktur perulangan for digunakan untuk perulangan yang lebih kompleks. Struktur perulangan ini akan mengulang-ulang eksekusi beberapa baris kode program di dalamnya berdasarkan perubahan isi sebuah variabel. Jika isi variabel yang di acu sudah dampai titik akhir, maka perulangan akan berhenti. Struktur perulangan **for** memiliki aturan penulisan sebagai berikut:

```
For(<init-exp>;<terminate-check>;<loop-end-exp>)  
    {  
        <baris-baris kode di dalam for>;  
    }
```

di mana,

- **init-exp** adalah awal dalam variabel yang diacu
- **terminate-check** adalah kondisi yang akan diperiksa untuk menentukan apakah perulangan diteruskan atau tidak. Jika kondisi terpenuhi, maka perulangan dilanjutkan. Jika kondisi sudah tidak terpenuhi, maka perulangan di hentikan.
- **Loop-end-exp** adalah bagian yang menunjukkan bagaimana isi variabel yang diacu berubah-ubah selama perulangan.

Misalkan kita mau mencetak "pemrograman PHP" sebanyak 10 kali menggunakan struktur perulangan **for**, maka dapat kita tuliskan sebagai berikut:

```
for($iter=0; $iter<10; $iter=$iter+1)  
    {  
        echo "pemrograman PHP";  
    }
```

3.3 Array

Array adalah jenis variabel yang mampu menyimpan beberapa nilai data bertipe sama dalam urutan tertentu. Isi atau nilai data yang berada dalam array tersebut dengan elemen array. Letak urutan masing-masing elemen array ditandai oleh indeks elemen array. Dalam PHP, indeks suatu array pada dasarnya secara otomatis selalu dimulai dari 0.

```
$array[0]= "pertama";  
$array[1]= "kedua";
```

pada contoh di atas, terdapat sebuah variabel array bernama **\$array**. Variabel **\$array** tersebut memiliki dua buah elemen. Elemen pada

indeks **0** adalah kata "pertama", dan elemen pada indeks **1** adalah kata "kedua".

Dalam PHP, indeks masing-masing elemen array tidak harus selalu ditunjukkan dalam bentuk angka. Indeks sebuah elemen array bisa juga berupa string yang diapit oleh dua buah karakter kutip satu, Perhatikan conoth berikut:

```
$array[1]= "pertama";  
$array[2]= "kedua";  
$array['indekstiga']= "ketiga";
```

Selain dengan mengisikan langsung, sebuah array dapat juga dibuat dengan memanfaatkan sebuah fungsi bernama **array()**. Perhatikan contoh berikut:

```
$array= array("satu","dua","tiga");
```

Exercise III

1. Tugas 5

Simpan dengan nama tugas_pemilihan.php

```
<html><head>
<title>latihan menggunakan pemilihan</title>
<body>
<?php
    $a=5;
    $b=15;
    if ($a > $b)
    {
        echo "a lebih besar dari b";
    }
    else
    {
        echo "a sama atau lebih kecil dari b";
    }
?>
<br>
<?php
    $harga=1500;
    switch($harga)
    {
    case 2000;
        echo "harganya mahal";
        break;
    case 1500;
        echo "harganya murah";
        break;
    default:
        echo "saya tidak tahu";
    }
?>
</body>
</html>
```

2. Tugas 6

Simpan dengan nama tugas_perulangan.php

```
<html>
<head>
<title>latihan menggunakan perulangan</title>
<body>
<?php
    $judul="pemrograman PHP";
    echo "judul di cetak 5 kali";
    echo "<br>";
    $a=1;
    while($a <= 5)
    {
        echo 4judul; echo "<br>";
        $a=$a+1;
    }
?>
</body>
</html>
```

3. Tugas 7

Simpan dengan nama tugas_array.php

```
<html>
<head>
<title>latihan menggunakan array</title>
<body>
<?php
    $array=array("A","B","C","D");
    $array2[1]= 15;
    $array2[2]= 16;
    $array2['tiga']= 17;
    $array2[4]= 18;

    $matriks[1][1]= 15;
    $matriks[1][2]= 17;
    $matriks[2][1]= 18;
    $matriks[2][2]= 19;

    $nilaimutu=$array[3];
```

```
list($adit, $sandri, $gofo)=$array2;

echo "isi elemen array pada indeks 3= ";
echo $array[3];
echo "<br>";

echo "isi elemen array2 pada indeks 2= ";
echo $array2[2];
echo "<br>";

?>
</body>
</html>
```

4. Tugas 8

Simpan dengan nama tugas_array_2.php

```
<html>
<head>
<title>latihan menggunakan array lanjutan</title>
<body>
<?php
    $fruit = array("Aplle", "Banana", "Orange", "Mango");
    list($red_fruit, $orange_fruit) = $fruit;
    echo $red_fruit. "<br>";
    echo $orange_fruit;
    echo "<br>";

?>
</body>
</html>
```

Modul 4

Pokok Bahasan:

- Pengolahan String
- Pengiriman Antarhalaman Web
- Membuat dan Menggunakan Fungsi
- Menggunakan Fungsi Include
- Menggunakan Fungsi Require

PHP Lanjutan

Indikator Keberhasilan:

- Dapat menggunakan dan mengolah String
- Dapat melakukan pengiriman antar halaman web
- Dapat memanfaatkan fungsi include atau require untuk mengefisiensikan program.

4.1 Pengolahan String

String adalah salah satu tipe data dalam PHP. String dapat diartikan sebagai rangkaian/deretan karakter. Ada dua cara untuk menuliskan

string dalam kode program PHP, yaitu dengan diapit dua buah tanda kutip satu atau dengan diapit dua buah tanda kutip ganda.

```
$string1 = "ini string dengan kutip satu";  
$string2 = "ini string dengan kutip ganda";
```

Untuk menggabungkan dua buah string menjadi satu, gunakan karakter titik ".". Hasil gabungannya juga akan menjadi sebuah string.

Contoh:

```
$string1 = "string kesatu";  
$string2 = "string kedua";  
  
$stringgabung = $string1.string2;
```

Ada beberapa fungsi yang dapat digunakan untuk mengolah string. Fungsi **strlen()** dapat dimanfaatkan untuk mendapatkan panjang sebuah string.

Fungsi **rtrim()** dapat digunakan untuk menghilangkan seluruh karakter spasi yang ada pada bagian akhir (kanan) suatu string.

Fungsi **substr()** digunakan untuk mendapatkan sebagian kalimat dari sebuah string. Fungsi ini memiliki aturan penulisan **substr()**.

Substr(<stringnya>,<mulai dari>,<berapa banyak>)

Fungsi **strtoupper()** dapat digunakan untuk membuat semua huruf dalam string menjadi huruf besar (kapital).. dan fungsi **strtolower()** digunakan untuk membuat semua huruf dalam string menjadi huruf kecil(sebaliknya)

Contoh:

```
$string1 = "string kesatu";  
$string2 = "string kedua";  
  
$stringgabung = strlen(string1);
```

```
$stringawal = "string kesatu";  
$stringbaru = rtrim($stringawal);
```

```
$stringawal = "buku in bagus sekali";
```

```
$stringbaru = substr($stringawal, 9, 12);
```

```
$stringawal = "string kesatu";  
$stringkecil = strtolower($stringawal);  
$stringbesar = strtoupper($stringawal);
```

4.2 Pengiriman Antarhalaman Web

Dalam PHP, jika pengiriman dilakukan dengan method **GET**, maka pada halaman web si penerima (pemroses hasil kiriman) akan digunakan variabel array bernama **\$_GET** untuk menangkapnya. Nama setiap masukan yang tertulis pada atribut **NAME**, dalam Tag **<INPUT>** pada Form yang dikirimkan akan menjadi indeks bagi array **\$_GET**. Untuk mengakses nilainya, gunakan aturan penulisan **\$_GET[<indeks>]**.

Jika pengiriman dilakukan dengan method **POST**, maka pada halaman web si penerima akan digunakan variabel array bernama **\$_POST** untuk menangkapnya. Nama setiap masukan yang tertulis pada atribut **NAME**, nama Tag **<INPUT>** pada Form yang dikirimkan, akan menjadi indeks bagi array **\$_POST**. Untuk mengakses nilainya gunakan aturan penulisan **\$_POST[<indeks>]**.

4.3 Fungsi

Adakalanya ketika kita membuat program, ada beberapa baris kode program yang ditulis berulang-ulang. Hal tersebut tentu saja menyita waktu dan tidak efisien. Baris kode program menjadi sangat banyak dan terlalu kompleks.

Untuk lebih menyederhanakan kode program, sebaliknya program dibagi menjadi beberapa sub program. Fungsi adalah jenis sub program yang digunakan dalam PHP.

Fungsi adalah subprogram yang terdiri atas beberapa baris kode program yang dapat dipanggil kapanpun dan dari manapun. Fungsi akan menerima input berupa argument-argumen untuk kemudian digunakan dalam fungsi tersebut. Fungsi akan menghasilkan sebuah nilai tertentu untuk dikirmkan kembali ke si pemanggilnya.

Untuk membuat Fungsi, gunakan aturan penulisan sebagai berikut:

```
function nama_fungsi($arg1, $arg2)
{
    <baris-baris kode program dalam fungsi>;
    return ($nilaiyangdikembalikan);
}
```

perhatikan contoh berikut:

```
function nama_fungsi($arg1, $arg2)
{
    $hasil=$a * $a;
    return ($hasil);
}
```

Sebuah fungsi bernama **isset()** dapat digunakan untuk memeriksa apakah sebuah variabel sudah mempunyai isi atau belum. Jika variabel tersebut sudah ada isinya, fungsi **isset()** akan menghasilkan nilai **TRUE** dan nilai **FALSE** jika sebaliknya.

4.4 Include dan Require

Untuk membuat kode program lebih efisien, dapat meletakkan semua fungsi dan berbagai macam kode program PHP lainnya dalam sebuah file terpisah. Kemudian, file tersebut dapat dipanggil sewaktu-waktu jika diperlukan.

PHP menyediakan fungsi **include()** dan **require()** untuk memanggil file lain yang kita butuhkan ke dalam kode program.

Contoh:

```
Include("konfigurasi.php");
```

Exercise IV

1. Tugas 9

Simpan dengan nama tugas_string.php

```
<html>
<head>
<title>latihan menggunakan string</title>
<body>
<?php
    $string1 = "string kesatu";
    echo $string1. "<br>";

    $string2 = "string kedua";
    echo $string1. "<br>";

    $stringgabung = $string1.$string2;
    echo $stringgabung. "<br>";
    $pjpgstring = strlen(string1);
    echo $pjpgstring1. "<br>";

    $stringawal = "string kesatu";
    $stringbaru = ltrim($stringawal);
    echo $stringbaru. "<br>";

?>
</body>
</html>
```

2. Tugas 10

Simpan dengan nama tugas_pengiriman_get.php

```
<html>
<head>
<title>latihan mengirim dengan get</title>
<body>
Dibawah ini adalah form yang akan dikirmkan ke halaman web
bernama lat9.php
<form name="dataku" action="lat9.php" method="get">

Masukkan Nama Anda:
```

```
<input type="text" name="namaku">
<br>

ini tombol input dengan type="submit" :
<input type="submit" value="Kirimkan">
</body>
</html>
```

3. Tugas 11

Simpan dengan nama lat9.php

```
<html>
<head>
<title>latihan penerima</title>
<body>
<?php
    echo "SELAMAT DATANG";
    echo $_GET["namaku"];

?>
</body>
</html>
```

Modul 5

Pokok Bahasan:

- Mengetahui PHPMyAdmin
- Membuat Database
- Membuat Tabel
- Menambah record
- Mengubah Record
- Menghapus Record
- Menghapus Tabel

Dasar-dasar MySQL

Indikator Keberhasilan:

- Dapat membuat atau menghapus database
- Dapat membuat atau menghapus tabel
- Dapat melihat isi table
- Dapat membuat, mengubah, dan menghapus record

5.1 Mengetahui PHPMyAdmin

Pada dasarnya, mengelola database dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai untuk

setiap maksud tertentu. Jika ingin membuat database, ketikkan baris perintah yang sesuai untuk membuat database.

Dengan PHPMyAdmin, kita dapat membuat database, membuat tabel, mengisi data, dan lain sebagainya.

Ketika menginstall XAMPP, secara otomatis akan terinstal pula PHPMyAdmin. Untuk menjalankan PHPMyAdmin, lakukanlah hal seperti berikut:

1. Aktifkan web server Apache dan MySQL
2. Jalankan browser, kemudian ketikkan pada address bar <http://localhost/phpmyadmin/> lalu tekan Enter

5.2 Membuat Database

Misalkan kita akan membuat sebuah database dengan nama **datasekolah** untuk menyimpan data sekolah, maka lakukan perintah berikut:

1. Jalankan browser, kemudian ketikkan pada address bar <http://localhost/phpmyadmin/> lalu tekan Enter
2. Ketik kata "**datasekolah**" pada kotak isian **Create new database**. Kemudian klik tombol **Create**.
3. Maka database baru sudah terbuat
4. Jika ingin menghapus database tersebut, pilih nama databasenya pada kotak database dalam panel sebelah kiri, lalu klik tombol **Drop** pada panel sebelah kanan.

5.3 Membuat Tabel

Misalkan kita akan membuat sebuah database dengan nama **datasekolah** untuk menyimpan data sekolah, maka lakukan perintah berikut:

1. Jalankan browser, kemudian ketikkan pada address bar <http://localhost/phpmyadmin/> lalu tekan Enter
2. Ketik kata "**datasekolah**" pada kotak isian **Create new database**. Kemudian klik tombol **Create**.
3. Maka database baru sudah terbuat
4. Jika ingin membuat tabel pada database datasekolah, aktifkan atau klik database datasekolah lalu isi kolom pada **Create new table on database datasekolah**. **Name: dataseiswa** **Field: 3**
5. Klik tombol **Go**

Exercise V

Kerjakan soal dibawah ini untuk lebih menguasai materi yang sudah diberikan:

1. Buatlah sebuah database baru bernama **databasecoba**.

Kemudian buatlah tabel baru didalamnya dengan nama **datamatapelajaran**, yang terdiri dari field **kode_mapel**, **nama_mapel**, dan **guru**.

Isi tabel tersebut seperti berikut:

Kode_mapel	Nama_mapel	guru
1	Matematika	Aditya Nugroho
2	Fisika	Andri Arivian Asari
3	Kimia	Abdulgofo

2. Ubahlah guru mata pelajaran Kimia menjadi Rizki Trianto Rakhim

3. Buatlah sebuah database baru bernama **databasecobalagi**.

Kemudian buatlah tabel baru didalamnya dengan nama **datakaryawan**, yang terdiri dari field **kode_kar**, **nama_kar**, **gaji**, dan **kode_cab**.

Isi tabel tersebut seperti berikut:

Kode_kar	Nama_kar	gaji	Kode_cab
11	Andi	1500	A
12	Budi	2500	B
13	Cici	1500	A
14	Didi	3000	C
15	Fifi	5000	B

4. Ubahlah gaji Andi menjadi 2900
5. Hapuslah karyawan bernama Fifi

Modul 6

Pokok Bahasan:

- Apa itu SQL
- Perintah SELECT
- Perintah INSERT
- Perintah UPDATE
- Perintah DELETE
- Fungsi Agregasi
- Klausa WHERE
- Klausa ORDER BY

Mengenal SQL

Indikator Keberhasilan:

- Dapat menggunakan perintah SELECT, INSERT, UPDATE, dan DELETE
- Dapat menggunakan beberapa fungsi agregasi dalam perintah SELECT
- Dapat menggunakan klausa WHERE dan ORDER BY dalam perintah SELECT

6.1 Apa itu SQL?

SQL adalah singkatan dari *Structures Query Language (SQL)*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. SQL pertama kali didefinisikan oleh *American National Standart Institute (ANSI)* pada tahun 1986.

Sebagai sebuah bahasa untuk mengolah database, SQL memiliki dua komponen utama, yaitu *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*.

DDL digunakan untuk mendefinisikan struktur database serta mengatur bagaimana data dapat diakses. Bentuk perintah DDL yang umum digunakan adalah **CREATE**, **ALTER** dan **DROP**. Perintah **CREATE** digunakan untuk membuat database atau table. Perintah **ALTER** digunakan untuk mengubah definisi/nama database atau table. Perintah **DROP** digunakan untuk menghapus database atau table.

DML digunakan untuk menampilkan data, menambah data, mengubah data, atau menghapus data. Perintah DML terdiri atas perintah **SELECT**, **INSERT**, **UPDATE**, dan **DELETE**.

6.2 Perintah SELECT

Perintah **SELECT** digunakan untuk mengambil dan menampilkan data dari satu atau lebih table dalam database. Perintah **SELECT** merupakan bukan perintah SQL yang paling sering digunakan dan memiliki pola penulisan sebagai berikut:

```
SELECT <nama-namafield> FROM <namatabel>
[WHERE <kondisi>]
[GROUP BY <daftarfield>]
[HAVING <kondisi>]
[ORDER BY <daftarfield>]
```

perintah yang dibatasi oleh kurung siku "[" dan "]" adalah perintah yang boleh dituliskan atau tidak.

Pada bagian <namatabel>, setelah kata SELECT, tuliskan nama-nama field.

Contoh:

```
SELECT no_induk , nama_siswa FROM datasiswa
```

Dan berikut contoh SELECT untuk menampilkan seluruh field dari sebuah table bernama datasiswa.

```
SELECT * FROM datasiswa
```

1. Jalankan web server Apache dan Mysql
2. Kemudian ketik **PHPMysqladmin** pada web browser
3. Pilih database bernama **datasekolah**
4. Kemudian buatlah sebuah tabel baru dengan nama **datakelas**, yang terdiri atas field **kode_kelas**, **nama_kelas**, dan **kapasitas**.
Kode_kelas sebagai **Primary Key**
5. Isilah tabel tersebut
6. Klik pada menu **SQL**, kemudian pada kotak isian **Run SQL query/queries on database datasekolah** ketiklah perintah berikut, lalu tekan **Go**:

```
SELECT * FROM datakelas
```

7. Ulangi langkah diatas, namun perintah yang diketikan berbeda. Berikut perintah SQL-nya:

```
SELECT nama_kelas,kapasitas FROM datakelas
```

6.3 Perintah INSERT

Perintah INSERT digunakan untuk menambahkan record baru ke dalam table. Aturan penulisan perintah INSERT adalah sebagai berikut.

```
INSERT [INTO] <namatabel> [(<daftarfield>)]  
VALUES (<daftar_nilai_data>)
```

Urutan data dimasukan pada bagian <daftar_nilai_data> harus sesuai dengan urutan field yang di tuliskan pada bagian <daftarfield>.

Langkah-langkahnya sebagai berikut:

1. Jalankan web server Apache dan Mysql
2. Kemudian ketik **PHPMysqladmin** pada web browser
3. Pilih database bernama **datasekolah**
4. Klik pada menu **SQL**, kemudian pada kotak isian **Run SQL query/queries on database datasekolah** ketiklah perintah berikut, lalu tekan **Go**:

```
INSERT INTO datakelas  
(kode_kelas,nama_kelas,kapasitas)  
VALUES ('K93', 'Kelas 9.3', '35')
```

6.4 Perintah UPDATE

Perintah **UPDATE** untuk mengubah record dalam table. Aturan penulisan perintah **UPDATE** adalah sebagai berikut:

```
UPDATE <namatabel> SET <namafield=nilainya>  
[WHERE kondisi]
```

Perintah **UPDATE** disertai klausa **WHERE** untuk menandai record yang akan diubah. Jika klausa **WHERE** tidak digunakan, maka seluruh record akan ikut mengalami perubahan.

Contoh:

```
UPDATE datakelas SET kapasitas=45  
WHERE kode_kelas= 'K93'
```

Langkah-langkahnya sebagai berikut:

1. Jalankan web server Apache dan Mysql
2. Kemudian ketik **PHPMysqladmin** pada web browser
3. Pilih database bernama **datasekolah**
4. Klik pada menu **SQL**, kemudian pada kotak isian **Run SQL query/queries on database datasekolah** ketiklah perintah berikut, lalu tekan **Go**:

```
UPDATE datakelas SET kapasitas=45
```

```
WHERE kode_kelas= 'K93'
```

6.5 Perintah DELETE

Perintah **DELETE** digunakan untuk menghapus beberapa record dari sebuah table. Aturan penulisan perintah **DELETE** adalah:

```
DELETE FROM <namatabel>  
[WHERE <kondisi>]
```

Perintah **DELETE** disertai klausa **WHERE** untuk menandai record yang akan dihapus. Jika klausa **WHERE** tidak digunakan, seluruh record dalam table tersebut akan dihapus.

Contoh

```
DELETE FROM datakelas WHERE kode_kelas= '93'
```

Langkah-langkahnya sebagai berikut:

1. Jalankan web server Apache dan Mysql
2. Kemudian ketik **PHPMysqladmin** pada web browser
3. Pilih database bernama **datasekolah**
4. Klik pada menu **SQL**, kemudian pada kotak isian **Run SQL query/queries on database datasekolah** ketiklah perintah berikut, lalu tekan **Go**:

```
DELETE FROM datakelas WHERE kode_kelas= '93'
```

6.6 Fungsi Agregasi

Fungsi agregasi merupakan fungsi perhitungan yang bekerja pada sebuah field dalam table dan akan mengembalikan sebuah nilai tertentu. Fungsi-fungsi agregasi yang biasa digunakan adalah **COUNT**, **SUM**, **AVG**, **MIN** dan **MAX**. Fungsi agregasi biasanya dijalankan bersama dengan perintah **SELECT**.

Berikut penjelasan dari beberapa fungsi agregasi:

Fungsi	Keterangan
COUNT	Untuk menghitung banyak baris data
SUM	Untuk menjumlahkan semua nilai yang berada pada field tertentu
AVG	Untuk mendapatkan nilai rata-rata dari semua nilai yang berada pada field tertentu
MIN	Untuk mendapatkan nilai minimum dari seluruh nilai dari sebuah field
MAX	Untuk mendapatkan nilai maksimum dari seluruh nilai dalam sebuah field

Misalkan ingin mendapatkan total kapasitas kelas dari field kapasitas dalam tabel datakelas, maka perintah SQLnya sebagai berikut:

```
SELECT SUM(kapasitas) FROM datakelas
```

6.7 Klausula WHERE dan ORDER BY

Dalam perintah SELECT terdapat klausula **WHERE**, **GROUP BY**, **HAVING** dan **ORDER BY**.

Klausula **WHERE** digunakan untuk menyaring record mana saja yang akan diambil sesuai dengan kondisi yang diberikan.

Misalkan kita ingin menampilkan data sebuah kelas yang memiliki kode_kelas=K71 saja, maka gunakan perintah **SELECT** berikut:

```
SELECT * FROM datakelas WHERE kode_kelas= 'K71'
```

Klausula **WHERE** juga dapat digunakan pada perintah **UPDATE** atau **DELETE** seperti dibahas sebelumnya.

Klausula **ORDER BY** digunakan untuk mengurutkan data berdasarkan satu atau lebih field. Misalkan kita ingin menampilkan seluruh record dari tabel datakelas yang diurutkan secara menaik berdasarkan field kapasitas, maka gunakan perintah berikut:

```
SELECT * FROM datakelas ORDER BY kapasitas
```

Untuk pengurutan secara menurun, tambahkan klausa **DESC** seperti berikut:

```
SELECT * FROM datakelas ORDER BY kapasitas DESC
```

Untuk pengurutan berdasarkan lebih dari satu field, pisahkan masing-masing nama field dengan karakter koma “,” seperti contoh:

```
SELECT * FROM datakelas  
ORDER BY kapasitas, nama_kelas
```

Exercise VI

Kerjakan soal dibawah ini untuk lebih menguasai materi yang sudah diberikan:

6. Buatlah sebuah database baru bernama **databasesql**.
Kemudian buatlah tabel baru didalamnya dengan nama **datanilai**, yang terdiri dari field **nama_siswa**, **nilai_MTK**, **nilai_BI**, dan **nilai_Bing**.
Isi tabel tersebut seperti berikut:

Nama_siswa	Nilai_MTK	Nilai_BI	Nilai_Bing
Aditya Nugroho	70	65	85
Andri Arivian Asari	60	65	95
Abdulgofo	80	70	95
Hariadi Sentosa Hanis	60	70	80
Faris Sofi	55	75	75

7. Buat dan jalankan sebuah perintah SQL untuk menampilkan perintah field **nama_siswa** dan **nilai_Bing**.
8. Buat dan jalankan sebuah perintah SQL untuk menampilkan perintah field **nilai_MTK**, **nilai_BI**, dan **nilai_Bing**.
9. Buat dan jalankan sebuah perintah SQL untuk menampilkan data baru bernama **budi** dengan **nilai_MTK=55**, **nilai_BI**, dan **nilai_Bing=70**

Modul 7

Pokok Bahasan:

- Koneksi dari PHP ke MySQL
- Menjalankan perintah SQL atau QUERY dari PHP
- Mengambil dan menampilkan hasil Query

Kolaborasi PHP dan MySQL

Indikator Keberhasilan:

- Dapat membangun koneksi antara PHP dengan MySQL
- Dapat menjalankan perintah SQL atau Query dari kode program PHP
- Dapat mengambil dan menampilkan hasil query ke halaman web

7.1 Koneksi dari PHP ke MySQL

PHP memiliki fungsi **mysql_connect()** yang dapat digunakan untuk membangun koneksi dengan database MySQL. Fungsi **mysql_connect()** memiliki aturan penulisan sebagai berikut:
mysql_connect(namahost, namauser, password)

Bagian **namahost** diisi dengan sebuah string yang merupakan nama host penyedia koneksi MySQL. Biasanya digunakan adalah **localhost**. Bagian **namauser** diisi dengan string yang merupakan nama pengguna user MySQL yang terdaftar. Biasanya digunakan adalah **root**.

Bagian **password** diisi dengan sebuah string yang merupakan password bagi user yang telah dituliskan pada bagian **namauser**. Jika user yang dimaksud tidak memiliki *password*, isikan bagian ini dengan dua buah tanda kutip saja.

Jika koneksi gagal dilakukan, fungsi **mysql_connect()** akan mengembalikan nilai **FALSE**. Contoh penulisan fungsi **mysql_connect()** adalah:

```
$hasil=mysql_connect('localhost', 'root', '');
```

untuk melihat dan menambahkan sebuah user baru dengan cara mengklik **Privilage** pada saat tampilan PHPMyAdmin.

Setelah koneksi dibangun, selanjutnya harus memilih database yang akan digunakan dalam koneksi tersebut. Fungsi yang digunakan adalah fungsi **mysql_select_db()**. Fungsi ini akan mengembalikan nilai **TRUE** jika berhasil dan **FALSE** jika salah. Aturan penulisan sebagai berikut:

```
Mysql_select_db(namadatabase);
```

7.2 Menjalankan SQL dari PHP

Perintah SQL atau biasa disebut query, juga dapat dijalankan/dieksekusi dari kode program PHP. PHP memiliki fungsi bernama **mysql_query()** untuk menjalankan/mengeksekusi sebuah query. Fungsi **mysql_query()** akan mengembalikan nilai **FALSE** jika query tidak berhasil dieksekusi.

Fungsi **mysql_query()** memiliki aturan penulisan sebagai berikut:

```
$hasil=mysql_query("SELECT * FROM datasiswa");
```

Jika query yang dieksekusi adalah perintah **INSERT**, **UPDATE**, atau **DELETE** serta query tersebut berhasil dieksekusi, selanjutnya dapat menggunakan fungsi bernama **mysql_affected_rows()** untuk mengetahui berapa banyak record yang terpengaruh oleh query tersebut.

Contoh:

```
$banyaknya=mysql_affected_rows();
```

Jika query yang dieksekusi adalah perintah **SELECT** serta query tersebut berhasil dieksekusi, selanjutnya dapat menggunakan sebuah fungsi bernama **mysql_num_rows()** untuk mengetahui berapa banyak record yang berhasil diambil oleh perintah **SELECT** tersebut.

Contoh:

```
$hasil=mysql_query("SELECT * FROM datasiswa");  
$banyaknya=mysql_num_rows($hasil);
```

7.3 Mengambil dan Menampilkan Hasil Query

Setelah query berisi perintah **SELECT** berhasil dijalankan menggunakan **mysql_query()**, selanjutnya dapat mengambil record-record yang dihasilkan dari perintah **SELECT** tersebut untuk kemudian ditampilkan ke dalam halaman web.

PHP memiliki beberapa fungsi untuk mengambil hasil query dan yang paling sering digunakan adalah:

- `mysql_fetch_row()`
- `mysql_fetch_array()`
- `mysql_result()`
- `mysql_fetch_row()`

Fungsi `mysql_fetch_row()` akan mengambil sebuah record dari hasil query, serta mengubahnya menjadi berbentuk array. Banyaknya elemen array disesuaikan dengan banyaknya field. Masing-masing elemen array mewakili masing-masing field. Indeks elemen array-nya adalah berupa angka dimulai dari **0**. Contoh penggunaan fungsi ini adalah :

```
//query nya:
$query="SELECT * FROM datasiswa";

//eksekusi query :
$hasil=mysql_query($query);

//selama masih bias diambil record-nya
While ($rec = mysql_fetch_row($hasil)) {
    //tampilkan hasil

    //isi field no_induk
    echo $rec[0]. "          ";

    //isi field nama_siswa
    echo $rec[1]. "          ";

    //isi field alamat_siswa
    echo $rec[2];
}
```

Fungsi `mysql_fetch_array()` akan mengambil sebuah record dari hasil query, serta mengubahnya menjadi berbentuk array. Banyaknya

elemen array disesuaikan dengan banyaknya field. Namun, indeks dari elemen array mewakili masing-masing field-nya. Contoh:

```
//query nya:
$query="SELECT * FROM datasiswa";

//eksekusi query :
$hasil=mysql_query($query);

//selama masih bias diambil record-nya
While ($rec = mysql_fetch_array($hasil)) {
    //tampilkan hasil

    //isi field no_induk
    echo $rec['no_induk']. "          ";

    //isi field nama_siswa
    echo $rec['nama_siswa']. "          ";

    //isi field alamat_siswa
    echo $rec['alamat_siswa'];
}
```

Fungsi **mysql_result()** digunakan untuk mengambil tepat satu nilai data saja. Fungsi ini biasanya digunakan untuk mengambil hasil eksekusi query yang berisi fungsi agregasi. Contoh:

```
//query nya:
$query="SELECT SUM(kapasitas) FROM datakelas";

//eksekusi query:
$hasil=mysql_query($query);

$total = mysql_result($hasil, 0, 0);
echo "total kapasitas kelasnya adalah = ";
echo $total;
```

Untuk berlatih mengambil dan menampilkan hasil query, lakukanlah langkah-langkah berikut:

1. Jalankan Notepad
2. Ketik baris berikut:

```
<HTML>
<HEAD>
<TITLE>Latihan mengambil dan menampilkan hasil query</TITLE>
</HEAD>
<BODY>
<?php
//konfigurasi nama host, nama user dan password
$host = 'localhost';
$user = 'root';
$pass = '';

//Membangun koneksi
$hasil = mysql_connect($host, $user, $pass)
    or die ("Koneksi GAGAL ");

$namadb = 'datasekolah';

//Memilih database
$db=mysql_select_db($namadb);

//Perintah SELECT
$query="SELECT * FROM datasiswa";
$hasil=mysql_query($query);
$banyaknya=mysql_num_rows($hasil);

echo "Diambil dengan mysql_fetch_rows() :";
echo "<br>";

//selama masih bias diambil record-nya
While ($rec = mysql_fetch_row($hasil) {
    //tampilkan hasil
    //isi field no_induk
    echo $rec[0]. "    ";
}
```

```
//isi field nama_siswa
    echo $rec[1]. "      ";

    //isi field alamat_siswa
    echo $rec[2]; echo "<br> ";
}
echo "<hr>";

$hasil=mysql_query($query);
echo "Diambil dengan mysql_fetch_array() : ";
echo "<br>";

//selama masih bisa diambil record-nya
While ($rec = mysql_fetch_array($hasil)) {
    //tampilkan hasil

    //isi field no_induk
    echo $rec['no_induk']. "      ";

    //isi field nama_siswa
    echo $rec['nama_siswa']. "      ";

    //isi field alamat_siswa
    echo $rec['alamat_siswa'];
    echo "<br>";
}
echo "<hr>";
echo "Contoh dengan mysql_result() :";
echo "<br>";

//query nya:
$query="SELECT SUM(kapasitas) FROM datakelas";

//eksekusi query :
$hasil=mysql_query($quey);
$total = mysql_result($hasil, 0, 0);
```

```
echo "Total kapasitas kelasnya adalah =";  
echo $total;  
  
?>  
</BODY>  
</HTML>
```

3. Kemudian disimpanlah dengan nama "**latihan14.php**" pada folder yang sudah dibuat

7.4 Study Kasus 1

Membuat Halaman Input Data Siswa Baru

Kali ini kita akan membuat database datasekolah dan table datasiswa untuk membuat halaman web yang dapat digunakan untuk menginput data siswa baru. Ikuti langkah – langkah berikut:

1. Buat sebuah folder nama **Studikusus1** pada folder C:\Program Files\xampp\htdocs.
2. ketik baris berikut:

```
<HTML><HEAD>  
<TITLE> Halaman Input Data Siswa </TITLE>  
</HEAD>  
<BODY>  
<FORM ACTION="inputdata.php" METHOD="post">  
Nomor Induk : <br>  
<INPUT TYPE="text" NAME="noinduknya"><BR>  
Nama : <BR>  
<INPUT TYPE="text" NAME="namanya"><BR>  
Alamat : <BR>  
<INPUT TYPE="submit" VALUE="inputkan"><BR>  
<INPUT TYPE="reset" VALUE="ulang"><BR>  
</FORM>  
</BODY></HTML>
```

3. Simpan dengan nama "index.php"
4. Kemudian ketik baris – baris teks berikut:

```
<HTML>
<HEAD>
<TITLE>Halaman hasil Data Siswa Baru </TITLE>
</HEAD>
<BODY>
<?php
//konfigurasi nama host, nama user dan password
$host = 'localhost';
$user = 'root';
$pass = '';

//Membangun koneksi
$hasil = mysql_connect($host, $user, $pass)
    or die ("Koneksi GAGAL ");

$namadb = 'datasekolah';
//Memilih database
$db=mysql_select_db($namadb);

If (isset($_POST["noinduknya"])==TRUE &&
    isset($_POST["namanya"])==TRUE &&
    isset($_POST["alamatnya"])==TRUE &&
    {
    $query="INSERT INTO datasiswa
    (no_induk,nama_siswa,alamat_siswa)
    VALUES ('".$_POST["no_induknya"]."',
        '".$_POST["namanya"]."',
        '".$_POST["alamatnya"]."',
        '";

    $hasil=mysql_query($query);
    }
else
    {
    echo "<h1> Penambahan data GAGAL</h1>";
    }

```

```
//Perintah SELECT
$query="SELECT * FROM datasiswa";
$hasil=mysql_query($query);
echo "<hr>";
//selama masih bias diambil record-nya
while ($rec = mysql_fetch_row($hasil) {
    //tampilkan hasil
    //isi field no_induk
    echo $rec[0]. "      ";

    //isi field nama_siswa
    echo $rec[1]. "      ";

    //isi field alamat_siswa
    echo $rec[2]; echo "<br> ";
    echo "<hr>";
}
?>
<FORM ACTION="index.php">
<INPUT TYPE="submit" VALUE="Ke halaman Input">
</FORM>
</BODY>
</HTML>
```

5. Simpan dengan nama "inputdata.php"

7.5 Study Kasus 2

Membuat Buku Tamu

1. Buat database baru bernama **bukutamu**
2. Buat sebuah tabel baru dengan nama **datatamu**, yang terdiri dari **nama**, **alamat**, **jeniskelamin**, dan **kesan**. Sesuaikan type data setiap field dengan data yang akan disimpannya.
3. Simpan pada folder baru bernama **Studikasus2**
4. Ketik baris-baris teks berikut ini:

```
<?php
    $host = 'localhost';
    $user = 'root';
    $pass = '';
    $konek = mysql_connect($host, $user, $pass)
        Or die ("Koneksi GAGAL ");

    $namadb = 'bukutamu';
    //Memilih database
    $db=mysql_select_db($namadb);
?>
```

5. Simpan dengan nama "**koneksi.php**"
6. Kemudian ketik baris – baris teks berikut:

```
<HTML>
<HEAD>
<TITLE>Halaman Buku Tamu</TITLE>
</HEAD>
<BODY>
<TABLE CELLPADDING=3 BORDER=1>
<TR>
<TH> Nama </TH>
<TH> Alamat </TH>
<TH> Jenis Kelamin </TH>
<TH> Kesan </TH>
```

```
<?php
//konfigurasi nama host, nama user dan password
include "koneksi.php";
$query="SELECT * FROM datatamu";
$hasil=mysql_query($query);

While ($rec = mysql_fetch_row($hasil)) {
?>
<TR>
<TD> <?php
//selama masih bisa diambil record-nya
    //isi field nama
    echo $rec[0]; ?>
<TD> <?php
    //isi field Alamat
    echo $rec[1]; ?>
<TD> <?php
    //isi field Jenis Kelamin
    echo $rec[2]; ?>
<TD> <?php
    //isi field Kesan
    echo $rec[3]; ?>
<TR>
<?php
}
?>
</TABLE>
<HR>
<FORM ACTION="tambah.php" METHOD="post">
<TABLE CELLPADDING=3 BORDER=1>
<TR>
<TD Colspan=2 ALIGN=center> MASukkan Pendapat anda
Tentang halaman Web ini </TD>
</TR>

<TR>
<TD> Nama : </TD>
```

```
<TD><INPUT TYPE="text" NAME="nama"> </TD>
</TR>
<TR>
<TD> Alamat : </TD>
<TD><INPUT TYPE="text" NAME="alamat"> </TD>
</TR>
<TR>
<TD> Jenis Kelamin : </TD>
<TD><SELECT NAME="jk"> </TD>
<OPTION VALUES="L"> Laki - laki </OPTION>
<OPTION VALUES="P"> Perempuan </OPTION>
</SELECT> </TD>
</TR>
<TR>
<TD> Masukkan Kesan Anda : </TD>
<TD><TEXTAREA NAME="kesan" ROWS=5 COLS=10>
Masukkan kesan disini
</TEXTAREA>
</TD>
</TR>
<TR>
<TD ALIGN=center>
<INPUT TYPE="submit" VALUES="Kirim"> </TD>
</TD>
<TD ALIGN=center>
<INPUT TYPE="reset" VALUES="Ulangi"> </TD>
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

7. Simpan dengan nama "**index.php**"
8. Ketik baris-baris text dibawah ini:

```
<?php
include ("koneksi.php");

if (isset($_POST["nama"])==TRUE &&
    isset($_POST["alamat"])==TRUE &&
    isset($_POST["jk"])==TRUE &&
    isset($_POST["kesan"])==TRUE &&

    {
    $query="INSERT INTO datatamu
    (nama,alamat,jeniskelamin,kesan)
    VALUES (\'".$_POST["nama"].\' , \'
        .$_POST["alamat"].\' , \'
        .$_POST["jk"].\' , \'
        .$_POST["kesan"].\' )";

    $hasil=mysql_query($query);
    //perintah Header untuk pindah ke halaman web
yang dituju
    header("Location: index.php");
    }
?>
```

9. Kemudian simpan dengan nama "inputdata.php"

7.6 Study Kasus 3

Membuat Diary

1. Buat database baru bernama **diary**
2. Buat sebuah tabel baru dengan nama **datadiary**, yang terdiri dari **judul**, **waktu**, dan **isi**. Sesuaikan type data setiap field dengan data yang akan disimpannya.
3. Simpan pada folder baru bernama **Studikusus3**
4. Ketik bari-baris seperti berikut:

```
<?php
    $host = 'localhost';
    $user = 'root';
    $pass = '';
    $konek = mysql_connect($host, $user, $pass)
        Or die ("Koneksi GAGAL ");

    $namadb = 'diary';
    //Memilih database
    $db=mysql_select_db($namadb);
?>
```

5. Simpan dengan nama "**koneksi.php**"
6. Ketik baris berikut:

```
.judul {
    height: 24px;
    font-weight: bold;
    font-size: 14px;
    Color: gray;
    font-family: Verdana, Tahoma
}
.waktu {
    height: 15px;
    font-style: italic;
    font-size: 10px;
    Color: gray;
```

```
font-family: Verdana, Tahoma
}
.text {
    height: 20px;
    font-size: 11px;
    Color: 111111;
    font-family: Verdana, Tahoma
}
```

7. Simpan dengan nama "**cssku.css**" pada folder **Studykasus3**

8. Ketik baris berikut:

```
<HTML>
<HEAD>
<TITLE> Halaman Diary</TITLE>
<LINK REL=Stylesheet HREF="cssku.css"
TYPE="text/css">
</HEAD>
<BODY>
<?php
include "koneksi.php";

$query="SELECT * FROM datadiary";
$hasil=mysql_query($query);
echo "<FORM>

echo      "<INPUT      TYPE=\"button\"      NAME=\"hapusd\"
VALUE=\"Tambah      Data\"
ONCLICK=\"JAVASCRIPT:LOCATION.HREF='TAMBAH.PHP'\">;

echo "<HR>";
if(mysql_num_rows($hasil)>0) {
while ($rec = mysql_fetch_row($hasil)) {
echo "FONT CLASS=judul>".$rec[0]."</FONT>;
echo "&nbsp; &nbsp;";
echo "FONT CLASS=waktu>".$rec[1]."</FONT>;
echo "<br>;
```

```
echo "FONT CLASS=teks>".$rec[2]."</FONT>";
echo "<br>";
echo "<INPUT TYPE=\"button\" NAME=\"hapusd\"
VALUE=\"Hapus\"
ONCLICK=\"javascript:location.href='hapus.php?wkt=".$r
ec[1]."'\>";
echo "<br>";
echo "<hr>";
}
}
else {
echo "Belum ada data Entry";
}
echo "</FORM>";
?>
</BODY></HTML>
```

9. Simpan dengan nama "**index.php**" pada folder **Studykasus3**

10. Ketik baris-baris berikut:

```
<HTML>
<HEAD>
<TITLE>Halaman Tambah DIARY</TITLE>
</HEAD>
<BODY>
<FORM ACTION="tambahdata.php" METHOD="post">
<TABLE CELLPADDING=3 BORDER=1>
<TR>
<TD Colspan=2 ALIGN=center>MASukkan Isi Diary</TD>
</TR>
<TR>
<TD> Judul : </TD>
<TD><INPUT TYPE="text" NAME="judul"> </TD>
</TR>
<TR>
<TD> Isi : </TD>
<TD><TEXTAREA NAME="isi" ROWS=5 COLS=50>
```

```
</TEXTAREA>
</TD>
</TR>
<TR>
<TD ALIGN=center>
<INPUT TYPE="submit" VALUE="Kirim"> </TD>
</TD>
<TD ALIGN=center>
<INPUT TYPE="reset" VALUE="Ulangi"> </TD>
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

11. Simpan dengan nama "**tambah.php**"

12. Ketik baris-baris berikut:

```
<?php
include ("koneksi.php");

if (isset($_GET["wkt"])==TRUE)
    {
        $query="DELETE FROM datadiary
                WHERE waktu='".$_GET["wkt"]."'";
        $hasil=mysql_query($query);

//perintah Header untuk pindah ke halaman web yang
dituju
        header("Location: index.php");
    }
?>
```

13. Simpan dengan nama "**hapus.php**"

14. Ketik baris – baris berikut

```
<?php
```

```
include ("koneksi.php");
if (isset($_POST["judul"])==TRUE &&
    isset($_POST["isi"])==TRUE &&
    {
    $waktu=date("H:m:s d M Y");

    $query="INSERT INTO datadiary
    (judul,waktu,isi)
    VALUES (\'".$_POST["judul"]."\', \'
        .$_waktu."', \'
        .$_POST["isi"]."', \";
    $hasil=mysql_query($query);
    //perintah Header untuk pindah ke halaman web
yang dituju
    header("Location: index.php");
    }
?>
```

15. Simpan dengan nama "**tambahdata.php**"

Exercise VII

Kerjakan soal dibawah ini untuk lebih menguasai materi yang sudah diberikan:

1. Buatlah sebuah halaman web berupa kalkulator sederhana. Kalkulator tersebut setidaknya harus dapat melakukan penjumlahan, pengurangan, perkalian, serta pembagian, dan menampilkan hasilnya dalam sebuah kotak teks.
2. Buatlah sebuah halaman web yang berisi Agenda kegiatan anda. Halaman web ini harus memiliki fasilitas menambah, mengubah, dan menghapus data.
3. Buatlah halaman web untuk mengelola data penjualan dan pembelian barang di sebuah perusahaan. Halaman web harus memiliki fasilitas menambah, mengubah, dan menghapus data penjualan dan pembelian barang.