

BAB 1

PENGENALAN PHP

taryanarx@email.unikom.ac.id

1.1. Mengenal PHP

1.1.1. Sejarah Singkat PHP

Pada awalnya, PHP merupakan kependekan dari Personal Home Page karena hanya digunakan pribadi oleh pembuatnya. PHP pertama kali dibuat oleh Rasmus Lerdorf, seorang programmer dari Denmark pada tahun 1995.

Karena PHP bersifat open souce, banyak yang tertarik mengembangkan kode PHP ini. Salah satunya adalah perusahaan yang bernama Zend. Akhirnya Zend merilis PHP versi 2.0 pada tahun 1997. Setahun kemudian, PHP 3.0 juga dirilis.

Tidak berhenti sampai disitu, perkembangan PHP semakin maju dengan dirilisnya PHP 4.0, di mana versi ini banyak digunakan oleh programer karena mempunyai kemampuan untuk membangun aplikasi web dengan kecepatan dan stabilitas yang tinggi. Kemudian Juni 2004, Zend kembali merilis PHP versi 5.0 yang lebih dikenal dengan model OOP (Object Oriented Programming).

Banyak kelebihan yang bisa diperoleh jika anda menggunakan PHP. Berikut ini kelebihan PHP dibandingkan bahasa pemrograman yang lain:

1. Gratis. Karena bersifat open source, PHP dapat kita gunakan dan dapat kita kembangkan secara bebas.
2. Lebih cepat. Tidak diragukan lagi bahwa PHP lebih cepat dibandingkan dengan aplikasi CGI dengan Perl atau Phyton, bahkan dengan ASP maupun Java dalam berbagai aplikasi web.
3. Mudah dipelajari atau digunakan. Banyak sekali forum-forum yang berisi tutorial belajar PHP ataupun diskusi yang membahas tentang PHP. Jika Anda ingin belajar PHP, Anda tidak perlu khawatir karena bisa mempelajarinya lewat forum maupun situs yang ada. (<http://phphostingindonesia.com>)

1.1.2. Sintak PHP

Untuk menuliskan sesuatu dalam format PHP dapat dilakukan dengan cara sebagai berikut:

NamaFile:Latihan1.php

```
<h1>Program Pertama</h1>
<?php
echo "Belajar PHP!";
?>
```

Simpan Code Program tersebut di folder

c:/xampp/htdocs/HI/latihan1.php

Untuk melihat hasilnya dapat dilakukan dengan cara pada address bar browser ketikkan:localhost/HI/latihan1.php

Ini adalah output yang dihasilkan dari program di atas.

Program Pertama

Belajar PHP!

Gambar 1.1.Hasil Latihan1.php

1.1.3. Pemisahan Instruksi

Pemisahan instruksi dalam PHP sama dengan pemisahan intruksi yang digunakan dalam bahasa C atau PERL, yaitu dengan menggunakan titik koma sebagai penutup tiap statement dan diakhiri dengan tag (?>) dan juga berfungsi sebagai akhir statement.

NamaFile:latihan2.php

```
<h1>Instruksi Pemisahan</h1>
<p>
<?php
    echo "Cara Pemisahan Pertama";
?>
```

```
<p>  
<?php echo "Cara Pemisahan Kedua" ?>
```

Instruksi Pemisahan

Cara Pemisahan Pertama

Cara Pemisahan Kedua

Gambar 1.2. Pemisahan Instruksi

1.1.4. Komentar

Dalam PHP, Anda dapat menggunakan dua jenis komentar: komentar satu baris dan komentar multi-baris.

Untuk menambahkan baris komentar dalam PHP dapat digunakan beberapa style yang didukung oleh PHP ini, yaitu komentar 'C', 'C++' dan Unix Shell-style, contoh:

```
// Ini adalah komentar satu baris  
$variable = 5; // Komentar di belakang kode
```

```
<html>  
<body>  
Menulis Komentar  
<?php  
// Satu Baris Komen  
# Satu baris Komen Juga  
  
/*  
Menuliskan Komen yang lebih dari  
satu baris  
*/  
  
// Atau bisa juga komen ditulis diantara code program
```

```
$x = 5 /* + 15 */ + 5;  
echo $x;  
?>  
</body>  
</html>
```

Kode Program: latihan3.php

Cara ini adalah standar dalam banyak bahasa pemrograman, dan komentar tersebut tidak akan dijalankan atau dieksekusi oleh interpreter PHP. Fungsinya adalah untuk memberikan penjelasan atau dokumentasi di dalam kode tanpa mempengaruhi perilaku program secara fungsional.

Selain itu, jika Anda ingin menyisipkan komentar HTML dalam blok PHP, Anda dapat menggunakan sintaks HTML untuk komentar:

```
<?php  
  echo "Ini adalah kode PHP";  
  // <!-- Ini adalah komentar HTML di dalam PHP -->  
?>
```

Penting untuk menggunakan komentar dengan bijak untuk menjelaskan logika dan fungsi kode, yang dapat membantu pemeliharaan dan kolaborasi antar pengembang.

1.2 Tipe data dan Variabel

Dalam PHP, variabel digunakan untuk menyimpan data dan tipe data variabel didefinisikan secara otomatis berdasarkan nilai yang disimpan dalam variabel tersebut. Berikut adalah beberapa tipe data dan cara mendeklarasikan variabel di PHP:

Pemrograman PHP mendukung beberapa tipe data, sebagai berikut:

1. **Integer**

```
$number = 123;
```

2. **Float (Floating Point Numbers)**

```
$floatNumber = 3.14;
```

3. String

```
$text = "Halo Gaes Wilujeng Siang";
```

4. Boolean

```
$isTrue = true;  
$isFalse = false;
```

5. Array

```
$fruits = array("Apple", "Banana", "Orange");
```

6. Object

```
class MyClass {  
    // isi kelas  
}  
$objectInstance = new MyClass();
```

7. Null

```
$nullValue = null;
```

Tipe variabel ini tidak perlu ditentukan oleh programmer, melainkan ditentukan pada saat runtime oleh PHP dengan kebergantungan pada konteks pemakaian variabel tersebut.

1.2.1 Menginisialisasi Variabel

Untuk menginisialisasi suatu variabel dalam PHP, secara sederhana berikan saja suatu nilai padanya. Tetapi untuk tipe seperti array dan objek membutuhkan mekanisme yang berbeda.

```
$angka = 42;           // integer  
$pi = 3.14;           // float  
$nama = "Fahra Ragita"; // string  
$benar = true;        // boolean  
$buah = array("Apel", "Jeruk", "Pisang"); // array  
$objek = new MyClass(); // object  
$kosong = null;       // NULL
```

1.2.2. Menginisialisasi Array

Array dapat diinisialisasi dengan dua cara: dengan pemberian nilai langsung, dan menggunakan konstruksi `array()`.

Untuk memberikan nilai secara langsung pada suatu array, secara sederhana adalah memberikan nilai ke variabel array dengan subscript kosong. Nilai tersebut akan ditambahkan sebagai elemen terakhir dari array.

```
$namasiswa[] = "Ilham"; // $namasiswa[0] = "Ilham"  
$namasiswa[] = "Gaga"; // $namasiswa[1] = "Gaga"  
Sesuatu yang perlu diingat adalah elemen array dimulai dari index 0 (bukan 1)
```

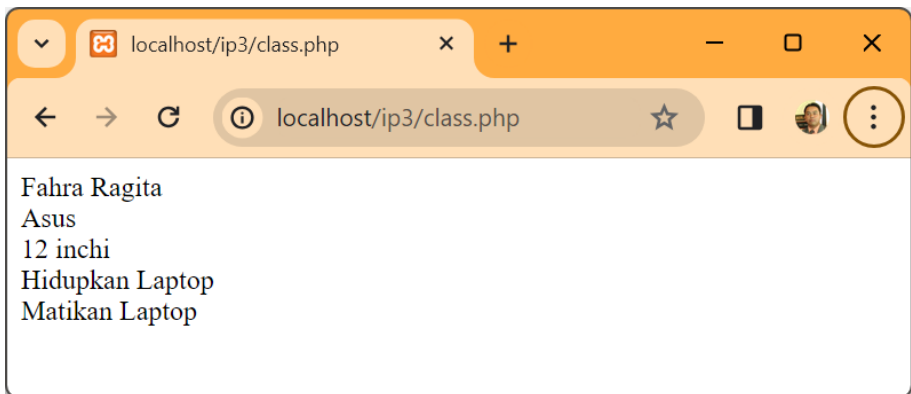
1.2.3. Menginisialisasi Objek

Untuk menginisialisasi suatu object, anda dapat menggunakan perintah `new`. Perintah ini digunakan untuk menginisialisasi suatu object kepada suatu variabel.

NamaFile:class.php

```
<?php  
// buat class laptop  
class laptop {  
    // buat property untuk class laptop  
    var $pemilik;  
    var $merk;  
    var $ukuran_layar;  
  
    // buat method untuk class laptop  
    function hidupkan_laptop() {  
        return "Hidupkan Laptop";  
    }  
    function matikan_laptop() {  
        return "Matikan Laptop";  
    }  
}  
  
// buat objek dari class laptop (instansiasi)  
$laptop_baru = new laptop();  
  
// set property  
$laptop_baru->pemilik="Fahra Ragita";  
$laptop_baru->merk="Asus";  
$laptop_baru->ukuran_layar="12 inchi";  
  
// tampilkan property  
echo $laptop_baru->pemilik;  
echo "<br />";  
echo $laptop_baru->merk;
```

```
echo "<br />";  
echo $laptop_baru->ukuran_layar;  
echo "<br />";  
  
// tampilkan method  
echo $laptop_baru->hidupkan_laptop();  
echo "<br />";  
echo $laptop_baru->matikan_laptop();  
?>
```



Gambar 1.3. Output Latihan4

1.2.4. Jangkauan dari variabel

Jangkauan dari suatu variabel adalah tergantung pada dimana variabel tersebut didefinisikan. Pada umumnya semua variabel PHP hanya memiliki suatu jangkauan.

Setiap variabel yang digunakan dalam suatu fungsi, maka secara default jangkauannya adalah lokal. Sebagai contoh:

```
$awal = 1; /* jangkauan global */
function Coba () {
    echo $awal; /*mengacu pada variabel jangkauan lokal */
}
Coba ();
```

Script diatas tidak mengeluarkan apapun, karena \$awal dalam fungsi Coba mengacu pada \$awal lokal yang tidak memiliki nilai apapun. Hal ini berbeda dengan bahasa C dimana setiap variabel global akan berlaku bagi semua function, kecuali didefinisikan sebagai local. Dalam PHP, variabel global harus dideklarasikan sebagai global didalam suatu fungsi jika mereka akan digunakan dalam fungsi tersebut, contoh :

```
$panjang = 10;
$lebar = 5;
function Hitungluas () {
    global $panjang, $lebar,$luas;
    $luas = $panjang * $lebar;
}
Hitungluas ();
echo $luas;
```

Script diatas akan menghasilkan keluaran **"50"**. Dengan mendeklarasikan \$panjang dan \$lebar sebagai global didalam fungsi.

Cara kedua untuk mengakses variabel sebagai acuan global adalah dengan menggunakan definisi \$GLOBAL array pada PHP. Contoh sebelumnya dapat juga ditulis menjadi:

```
echo "<hr>Cara Kedua<br>";
$panjang= 5;
$lebar= 3;
function Hitungluas2() {
    $GLOBALS["luas"]= $GLOBALS["panjang"] * $GLOBALS["lebar"];
}
Hitungluas2();
echo $luas;
```

Script diatas akan menghasilkan nilai luas = **15**

\$GLOBALS array adalah suatu asosiasi array dengan key adalah nama dari variabel global.

Hal lain yang penting dari jangkauan adalah variabel static. Suatu variabel hanya ada pada jangkauan lokal fungsi, dan nilainya akan tetap dipertahankan:

Contoh berikut menggunakan variabel static.

```
<?php
function Coba () {
    static $a = 0;
    echo $a;
    $a++;
}
Coba();
?>
```

Sekarang, setiap kali fungsi Coba(), dipanggil maka nilai \$a adalah nilai \$a pada pemanggilan sebelumnya, karena awalnya \$a=0, maka setiap dipanggil variabel \$a akan selalu **0 (nol)**.

1.2.5. Variabel variabel

Kadang-kadang adalah lebih nyaman menggunakan variable-variabel; Yang mana adalah nama variabel yang dapat digunakan secara dinamis. Secara normalnya variabel dibuat dengan :

```
$a = "hello";
```

Suatu variabel variabel akan menggunakan nilai dari suatu variabel menjadi nama variabel, Contoh:

```
$a = "hello";
$$a = "world";
```

Pada contoh diatas akan terbentuk dua variabel yaitu \$a dan \$hello. Dimana \$a berisi "hello" dan \$hello berisi "world". Selanjutnya perintah berikut ini:

```
echo "$a ${$a}";
atau
echo "$a $hello"
```

Akan menghasilkan keluaran : hello world

1.2.6. Penentuan tipe variabel

PHP tidak membutuhkan deklarasi variabel secara eksplisit; tipe variabel ditentukan berdasarkan konteks pemakaiannya pada saat runtime. Dengan kata lain; jika anda memberi nilai string kesuatu variabel var, var menjadi suatu variabel tipe string. Jika anda memberi nilai integer ke var, maka otomatis berubah menjadi tipe integer.

Suatu contoh dari otomatisasi konversi tipe pada PHP adalah operator penjumlahan '+'. Jika salah satu operandnya adalah tipe double, maka semua operand lainnya dievaluasi sebagai double dan hasilnya adalah double.

```
$coba = "0";           // $coba adalah string (ASCII 48)
$coba++;              // $coba adalah string "1" (ASCII 49)
$coba += 1;           // $coba sekarang adalah integer (2)
$coba = $coba + 1.3; // $coba sekarang adalah double (3.3)
$coba = 5 + "10 kotak"; // $coba adalah integer (15)
$coba = 5 + "10 kotak kecil"; // $coba adalah integer (15)
```

1.2.7. Mengetahui tipe suatu variabel

Karena PHP menentukan tipe variabel sesuai dengan konteks pemakaiannya, maka anda dapat menggunakan fungsi berikut untuk memeriksa tipe pada suatu variabel gettype(), is long(), is double(), is string(), is array(), dan is object().

1. Menggunakan fungsi `gettype()`:

Fungsi `gettype()` dapat digunakan untuk mengembalikan string yang berisi nama tipe variabel dari suatu nilai.

```
$angka = 42;
echo gettype($angka); // Output: integer

$teks = "Hello";
echo gettype($teks); // Output: string
```

2. Menggunakan Operator `instanceof`:

Operator instanceof digunakan untuk memeriksa apakah suatu objek adalah instance dari suatu kelas.

```
<?php
class MyClass {
    // isi kelas
}
$objek = new MyClass();
if ($objek instanceof MyClass) {
    echo "Variabel ini adalah instance dari
MyClass.";
} else {
    echo "Variabel ini bukan instance dari MyClass.";
}
?>
```

3. Fungsi is_*:

Terdapat sejumlah fungsi bawaan di PHP yang dimulai dengan "is_" dan dapat digunakan untuk memeriksa tipe variabel. Beberapa contoh:

- 1) is_int(): Memeriksa apakah variabel adalah tipe integer.
- 2) is_float(): Memeriksa apakah variabel adalah tipe float.
- 3) is_string(): Memeriksa apakah variabel adalah tipe string.
- 4) is_bool(): Memeriksa apakah variabel adalah tipe boolean.

```
<?php
$angka = 62;
if (is_int($angka)) {
    echo "Variabel ini adalah integer.";
}

$teks = "Gang Katineung";
if (is_string($teks)) {
    echo "Variabel ini adalah string.";
}
?>
```

Dengan menggunakan salah satu dari metode di atas, Anda dapat menentukan tipe variabel dalam kode PHP Anda. Namun, dalam kebanyakan kasus, tipe variabel yang bersifat dinamis membuat Anda lebih fleksibel dalam bekerja dengan data.

1.2.8 Tipe casting

Tipe casting adalah konversi nilai dari satu tipe data ke tipe data lain. Dalam PHP, terdapat beberapa cara untuk melakukan tipe casting, memungkinkan Anda untuk mengubah tipe variabel sesuai kebutuhan.

Tipe casting dalam PHP bekerja seperti dalam C: nama dari tipe yang diinginkan ditulis didalam kurung sebelum variabel yang akan di cast.

```
$coba = 10; // $coba adalah suatu integer
$bar = (double) $coba; // $bar adalah suatu double
```

Berikut adalah beberapa cara umum untuk melakukan tipe casting di PHP:

1. Tipe Casting Otomatis (Automatic Type Casting):

PHP dapat melakukan tipe casting otomatis dalam beberapa situasi, terutama ketika Anda menggunakan operator atau fungsi tertentu.

Contoh:

```
$angka = "123";
$hasil = $angka + 5; // $angka akan secara otomatis diubah
menjadi integer
echo $hasil; // Output: 128
```

2. Fungsi settype():

Fungsi settype() digunakan untuk mengubah tipe variabel secara eksplisit.

```
$angka = "123";
settype($angka, "integer");
echo $angka; // Output: 123 (sekarang sebagai
integer)
```

3. Fungsi (tipe_data), alias Type Casting Operator:

Anda dapat menggunakan operator tipe casting untuk mengubah tipe variabel.

Contoh:

```
$angka = "123";  
$angka = (int) $angka;  
echo $angka; // Output: 123 (sekarang sebagai integer)
```

4. Fungsi intval(), floatval(), strval(), dll.:

Fungsi-fungsi ini dapat digunakan untuk secara eksplisit mengubah nilai menjadi integer, float, atau string.

Contoh:

```
$teks = "123";  
$angka = intval($teks);  
echo $angka; // Output: 123 (sebagai integer)  
  
$angka = 3.14;  
$teks = strval($angka);  
echo $teks; // Output: "3.14" (sebagai string)
```

5. Fungsi type_casting() (hanya di PHP 8):

Dalam PHP 8, fungsi type_casting() diperkenalkan sebagai fungsi bawaan yang dapat digunakan untuk melakukan tipe casting.

Contoh:

```
$angka = "123";  
$angka = type_casting($angka, "int");  
echo $angka; // Output: 123 (sekarang sebagai integer)
```

Penting untuk diingat bahwa tipe casting dapat menyebabkan kehilangan presisi atau informasi, terutama jika Anda mengonversi tipe data dengan rentang nilai yang berbeda. Oleh karena itu, gunakan tipe casting dengan bijak dan pertimbangkan implikasinya terhadap data Anda.

1.2.9. Cast yang diperbolehkan:

Di PHP, Anda dapat melakukan berbagai jenis tipe casting untuk mengubah nilai dari satu tipe data ke tipe data lain. Berikut adalah beberapa tipe casting yang diperbolehkan di PHP:

(int), (integer) - cast ke integer

(real), (double), (float) - cast ke double

(string) - cast ke string

(array) - cast ke array

(object) - cast ke object

1.2.10. Konversi String

Ketika suatu string dievaluasi sebagai suatu nilai numerik, nilai hasil dan tipe ditentukan sebagai berikut.

String akan dievaluasi sebagai suatu double jika mengandung salah satu karakter '.', 'e' atau 'E'. Jika tidak akan dievaluasi sebagai suatu string.

Jika string dimulai dengan data numerik yang sah, maka nilai tersebut akan digunakan, jika tidak akan memiliki nilai 0 (nol).

```
$coba = 1 + "10.5";           // $coba adalah double (11.5)
$coba = 1 + "-1.3e3";        // $coba adalah double (-1299)
$coba = 1 + "bob-1.3e3";     // $coba adalah integer (1)
$coba = 1 + "bob3";         // $coba adalah integer (1)
$coba = 1 + "10 Small Pigs"; // $coba adalah integer (11)
$coba = 1 + "10 Little Piggies"; // $coba adalah integer (11);
$coba = "10.0 pigs " + 1;    // $coba adalah int (11)
$coba = "10.0 pigs " + 1.0;  // $coba adalah double (11)
```

Tipe dari variabel tergantung pada ekspresi kedua, jika ekspresi pertamanya adalah string

1.3. Menangani variabel Form, Cookies dan Environment

Dalam PHP, Anda dapat menangani variabel dari formulir (form variables), cookies, dan environment. Ini dapat dilakukan menggunakan variabel super global yang disediakan oleh PHP. Berikut adalah cara menangani ketiga jenis variabel ini:

1. Variabel Form (Form Variables):

Variabel form dapat diakses melalui superglobal `$_POST` atau `$_GET`, tergantung pada metode yang digunakan dalam formulir HTML (method="post" atau method="get").

Pada pemrograman CGI, program kita akan berinteraksi dengan variabel-variabel dari luar yang dikirim melalui form baik dengan metode GET maupun metode POST.

Ketika suatu form dikirim ke suatu PHP script, semua variabel dari form secara otomatis dapat diproses oleh script PHP sebagaimana variabel biasanya. Sebagai contoh, perhatikan form berikut ini:

NamaFile:form1.html

```
<form action="proses_form.php" method="post">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username">
  <input type="submit" value="Submit">
</form>
```

Mengakses Variabel Form di proses_form.php:

Nama File:proses_form.php

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Mengakses variabel form menggunakan $_POST
    $username = $_POST["username"];
    // Lakukan sesuatu dengan $username
}
?>
```

Ketika form disubmit, maka PHP akan membuat variabel \$username, yang mana mengandung apa yang diketikkan pada field username: di form tersebut.

PHP juga mendukung variabel array dalam konteks form, tetapi dibatasi hanya 1 dimensi, Contoh:

Nama File:form2_array.html

```
<form action="array.php" method="post">
  Nama: <input name="personal[nama]"><br>
  Email: <input name="personal[email]"><br>
  Jurusan: <br>
  <select name="jurusan[]">
    <option value="Informatika">Informatika
    <option value="Akuntansi">Akuntansi
    <option value="Elektro">Elektro
    <option value="Komputer">Komputer
    <option value="Ekonomi">Ekonomi
  </select>
  <input type="submit">
</form>
```

Nama File:array.php

```
<?php
$name = $_POST["personal"]["nama"];
$email = $_POST["personal"]["email"];
$jurusan = $_POST["jurusan"];
echo "Nama: " . $name . "<br>";
```

```
echo "Email: " . $email . "<br>";  
echo "Jurusan yang dipilih:<br>";  
foreach ($jurusan as $selectedJurusan) {  
    echo $selectedJurusan . "<br>";  
}  
?>
```

2. Cookies:

Cookies adalah informasi yang disimpan di sisi klien. Anda dapat mengakses nilai cookies menggunakan superglobal `$_COOKIE`.

Contoh Pengaksesan Cookie:

```
<?php  
// Set cookie  
setcookie("user", "Fahra Ragita", time() + 3600, "/");  
  
// Mengakses cookie  
if (isset($_COOKIE["user"])) {  
    $username = $_COOKIE["user"];  
  
    // Lakukan sesuatu dengan $username  
}  
?>
```

3. Variabel Environment:

Variabel environment menyediakan informasi tentang server dan lingkungan eksekusi PHP. Informasi ini dapat diakses melalui superglobal `$_SERVER`.

Contoh Pengaksesan Variabel Environment:

```
<?php  
// Mengakses informasi tentang server  
$serverName = $_SERVER["SERVER_NAME"];  
$port = $_SERVER["SERVER_PORT"];  
  
// Mengakses informasi tentang file yang sedang  
dieksekusi  
$scriptName = $_SERVER["SCRIPT_NAME"];  
  
// Lakukan sesuatu dengan informasi yang diakses
```

|?>

Pastikan untuk melakukan validasi dan sanitasi data yang diterima dari formulir atau cookies untuk mencegah risiko keamanan seperti serangan injeksi atau Cross-Site Scripting (XSS).gunakan teknik keamanan yang sesuai dengan kebutuhan proyek Anda.

1.4. Variabel pada IMAGE SUBMIT

Ketika mengirim suatu form, juga dimungkinkan untuk menggunakan suatu gambar sebagai pengganti tombol submit dengan tag HTML berikut ini:

```
<input type=image src="image.gif" name="sub">
```

Ketika pemakai melakukan klik pada gambar tersebut, makan form akan dikirim ke CGI dengan dua variabel tambahan, yaitu sub_x dan sub_y. Kedua variabel ini adalah koordinat dimana klik dilakukan pada gambar.

Form3.html

```
<html>
<form action="aksi3.php" method="post">
<input type=image src="button3.jpg" name="sub">
```

aksi3.php

```
<pre>
<?php print_r($_REQUEST);
$x=$_POST['sub_x'];
$y=$_POST['sub_y'];
echo $x;
echo $y;
?>
</pre>
```

1.5. HTTP Cookies

PHP secara transparan mendukung HTTP cookies. Cookies adalah suatu mekanisme penyimpanan data secara remote pada browser klien. Hal ini dapat digunakan untuk mengidentifikasi pemakai pada kunjungan berikutnya.

Anda dapat menggunakan fungsi SetCookie(). Cookie adalah bagian dari HTTP header, jadi fungsi SetCookie harus dipanggil sebelum output lainnya dikirim ke browser. Hal ini adalah sama batasannya untuk fungsi Header(). Semua cookies yang dikirim pada anda dari klien akan secara otomatis di ubah kedalam suatu variabel PHP seperti method GET dan POST data.

Jika anda ingin memberikan banyak nilai pada suatu cookie tunggal, tambahkan saja [] pada nama cookie.

Sebagai contoh:

```
SetCookie ("MyCookie[]", "Testing", time()+3600);
```

Catatan bahwa suatu cookie akan menimpa cookie sebelumnya yang memiliki nama yang sama dalam browser anda, kecuali path atau domainnya berbeda. Jadi untuk suatu aplikasi shopping cart anda perlu menyimpan suatu counter dan mengirimnya bersamaan.

Contoh. SetCookie.php

```
<?php
setcookie('username', 'administrator', time() + (60 * 60 * 24 * 5), '/');
setcookie('nama', 'Fahra Ragita', time() + (60 * 60 * 24 * 15), '/');
?>
```

GetCookie.php

```
<?php
echo "Username " . $_COOKIE['username'];
echo "<br/>";
echo "Nama " . $_COOKIE['nama'];
?>
```

1.6. Variabel Environment

Variabel lingkungan (environment variables) digunakan dalam PHP untuk mengakses informasi konfigurasi dan variabel lingkungan yang disetel pada server web atau sistem operasi.

Contoh umum penggunaan variabel lingkungan termasuk mengakses informasi seperti alamat IP pengguna, nama server, atau direktori tempat skrip PHP dijalankan.

Berikut adalah contoh penggunaan beberapa variabel lingkungan umum di PHP:

1. `$_SERVER['REMOTE_ADDR']`: Digunakan untuk mendapatkan alamat IP pengguna yang mengakses situs web Anda. Contoh:

```
$ip_address = $_SERVER['REMOTE_ADDR'];  
echo "IP Address pengguna: " . $ip_address;
```

2. `$_SERVER['HTTP_USER_AGENT']`: Digunakan untuk mendapatkan informasi tentang agen pengguna (browser) yang digunakan oleh pengguna. Contoh:

```
$user_agent = $_SERVER['HTTP_USER_AGENT'];  
echo "Agen Pengguna: " . $user_agent;
```

3. `$_SERVER['SERVER_NAME']`: Digunakan untuk mendapatkan nama server. Contoh:

```
$server_name = $_SERVER['SERVER_NAME'];  
echo "Nama Server: " . $server_name;
```

4. `$_SERVER['REQUEST_METHOD']`: Digunakan untuk mendapatkan metode permintaan HTTP yang digunakan oleh pengguna (GET, POST, PUT, dll.). Contoh:

```
$request_method = $_SERVER['REQUEST_METHOD'];  
echo "Metode Permintaan: " . $request_method;
```

5. `$_ENV['VARIABLE_NAME']`: Anda juga dapat mengakses variabel lingkungan kustom yang telah Anda setel sendiri. Misalnya, jika Anda

memiliki variabel lingkungan bernama "MY_CUSTOM_VAR," Anda dapat mengaksesnya seperti ini:

```
$custom_var = $_ENV['MY_CUSTOM_VAR'];  
echo "Nilai variabel lingkungan kustom: " . $custom_var;
```

Pastikan bahwa Anda telah mengatur variabel lingkungan yang Anda akses pada server atau dalam lingkungan Anda sebelum mencoba mengaksesnya dalam skrip PHP.

Untuk mengatur atau mengkonfigurasi environment dapat menggunakan fungsi putenv(), dan menggunakan Fungsi getenv() untuk mengaksesnya.

```
<pre>  
<?php  
$ip_address = $_SERVER['REMOTE_ADDR'];  
echo "Alamat IP pengguna: " . $ip_address;  
echo "<br>";  
$user_agent = $_SERVER['HTTP_USER_AGENT'];  
echo "Agen Pengguna: " . $user_agent;  
echo "<br>";  
$server_name = $_SERVER['SERVER_NAME'];  
echo "Nama Server: " . $server_name;  
echo "<br>";  
$request_method = $_SERVER['REQUEST_METHOD'];  
echo "Metode Permintaan: " . $request_method;  
echo "<br>";  
  
// Menetapkan nilai variabel lingkungan  
putenv("MY_VARIABLE=Belajar_PHP");  
$myVar = getenv("MY_VARIABLE");  
echo "Nilai variabel lingkungan MY_VARIABLE adalah: " . $myVar;  
echo "<br>";  
?>
```



1.7. Konstruksi Bahasa PHP

Suatu PHP script terdiri dari sejumlah perintah yang berurutan. Suatu perintah dapat berupa pemberi nilai, pemanggilan terhadap fungsi, perulangan, perintah kondisi dan baris kosong.

Setiap perintah diakhiri dengan sebuah titik koma (;). Dan beberapa perintah dapat dikelompokkan menjadi satu dalam kurung kurawal {dan }.

Catatan : Setiap perintah diakhir dengan sebuah titik koma (;).

1.8. Konstanta

PHP memiliki sejumlah konstanta yang telah didefinisikan, dan anda diberi kesempatan untuk membuat konstanta sesuai dengan kebutuhan anda.

Konstanta yang telah didefinisikan adalah `__FILE__` (nama file yang sedang diproses) `__LINE__` (nomor baris dari file yang sedang diproses)

Contoh :

```
<?php
function report_error($file, $line, $message) {
    echo "Suatu kesalahan terjadi pada file $file baris $line: $message.";
}
report_error(__FILE__, __LINE__, "Telah terjadi suatu kesalahan!");
?>
```

Anda dapat mendefinsikan konstanta ciptaan dengan menggunakan fungsi `define()`.

Contoh :

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // akan mencetak "Hello world."
?>
```

1.9. Ekspresi

Dalam konteks PHP, ekspresi merujuk pada bagian dari kode yang menghasilkan nilai. Ekspresi dapat terdiri dari nilai-nilai konstan, variabel,

operator, dan fungsi. PHP mendukung berbagai macam ekspresi yang dapat digunakan untuk melakukan operasi aritmatika, perbandingan, logika, dan banyak lagi.

Berikut adalah beberapa contoh ekspresi PHP:

1. Aritmatika

```
$a = 5;
$b = 10;

$sum = $a + $b;           // Penjumlahan
$difference = $a - $b;    // Pengurangan
$product = $a * $b;       // Perkalian
$quotient = $a / $b;      // Pembagian
$remainder = $a % $b;     // Modulus
```

2. Perbandingan

```
$a = 5;
$b = 10;

$equalTo = ($a == $b);    // Sama dengan
$notEqualTo = ($a != $b); // Tidak sama dengan
$greaterThan = ($a > $b); // Lebih besar dari
$lessThan = ($a < $b);    // Lebih kecil dari
$greaterOrEqual = ($a >= $b); // Lebih besar atau sama dengan
$lessOrEqual = ($a <= $b); // Lebih kecil atau sama dengan
```

3. Logika

```
$condition1 = true;
$condition2 = false;

$sandResult = ($condition1 && $condition2); // Operator logika AND
$sorResult = ($condition1 || $condition2); // Operator logika OR
$notResult = !$condition1;
```

4. Concatenation (Penggabungan String):

```
$firstName = "John";  
$lastName = "Doe";  
  
$fullName = $firstName . " " . $lastName; //  
Penggabungan string
```

5. Pemilihan Nilai

```
$age = 25;  
$category = ($age < 18) ? "Child" : "Adult";  
// Operator ternary untuk pemilihan nilai
```

6. Pemanggilan Fungsi

```
function addNumbers($a, $b) {  
    return $a + $b;  
}  
$result = addNumbers(5, 10); // Pemanggilan fungsi
```

Ekspresi PHP dapat digunakan di berbagai konteks, seperti dalam pernyataan kondisional, iterasi, definisi variabel, dan banyak lagi. Mereka membentuk dasar untuk berbagai operasi dan manipulasi data dalam skrip PHP.

DAFTAR PUSTAKA

1. Suryana, Taryana (2024) [Pengantar Pemrograman WEB \(HTML, CSS, PHP, MYSQL\)](#) xviii, 181 hlm Deepublish, ISBN:[978-623-02-8181-5](#)
2. Suryana, Taryana; Koesheryatin (2014), (2023), [Aplikasi Internet Menggunakan HTML, CSS & Java Script](#), Elexmedia Komputindo, Jakarta
3. Suryana, Taryana (2010) [Membuat Web Pribadi dan Bisnis dengan HTML](#)., Gava Media, Yogyakarta
4. Suryana, Taryana; Sarwono, Jonathan (2007) [E-COMMERCE MENGGUNAKAN PHP DAN MYSQL](#), Graha Ilmu, Yogyakarta
5. *Wong Jony. 2010. Internet Marketing for Beginners. Jakarta: PT Elex Media Komputindo.*
6. *Kumar, Anil and Manoj Kumar Dash. 2013. Constructing a Measurement in Service Quality for Indian Banks: Structural Equation Modeling Approach. Journal of Internet Banking and Commerce, April 2013, vol. 18, no.1*
7. *Siregar, Riki R. 2010. Strategi Meningkatkan Persaingan Bisnis Perusahaan dengan Penerapan E-commerce*
8. <https://www.php.net/manual/en/book.mysql.php>
9. <https://www.w3schools.com>
10. <https://tekno.kompas.com/read/2022/09/26/10150007/pengertian-e-commerce-beserta-jenis-contoh-dan-manfaatnya>
11. <https://meenta.net/pengertian-e-commerce/>
12. <https://www.rumahweb.com/journal/session-adalah/>
13. <https://rizkypratama.id/blog/2019/11/01/membuat-session-login-pada-php/>
14. <https://blog.skillacademy.com/supplier-adalah>
15. <https://chat.openai.com/>
16. <http://www.grahailmu.co.id/previewpdf/978-979-756-182-6-231.pdf>
17. <https://www.w3schools.com/html>, September, 2025
18. https://repository.dinus.ac.id/docs/ajar/Pert_5_Percabangan_Bersarang.pptx

