

## **BAB II**

### **DASAR TEORI**

#### **2.1. PENGERTIAN SISTEM PELELANGAN**

Lelang adalah penjualan barang dimuka umum termasuk melalui media elektronik yang dipimpin dan atau dilaksanakan dihadapan Pejabat lelang, dengan cara penawaran harga semakin menurun, dan atau dengan penawaran harga secara tertulis yang didahului dengan usaha mengumpulkan para peminat. Dimana pembentukan harga yang terjadi dilakukan secara transparan tanpa ada kolusi antar pelaku usaha dan tanpa tekanan dari pihak manapun.

#### **2.2. PENGERTIAN SISTEM PELELANGAN SECARA ONLINE**

Lelang Online adalah transaksi berbasis Web yang memungkinkan pengguna membeli dan menjual produk dan layanan tertentu melalui proses tawar menawar.

#### **2.3. KELEBIHAN DARI LELANG ONLINE**

1. Tidak terbatas waktu

Dengan lelang online, Anda dapat melakukan penawaran kapanpun Anda mau, 24 jam sehari dan 7 hari seminggu.

2. Tidak Terbatas oleh Tempat.

Anda tidak perlu melihat wajah saingan-saingan yang ikut menawar item favorit Anda, karena bisa jadi Anda dan mereka berdiri pada benua yang berbeda, tentunya selama tempat tersebut terjangkau Internet.

3. Jumlah Penawar yang Besar.

Jika seluruh penawar pada sebuah website lelang online yang besar (misalnya eBay) dikumpulkan menjadi satu, rasanya tidak akan ada tempat yang nyaman untuk menampung mereka semua. Sementara semakin banyak penawar, maka itulah yang diharapkan oleh pengelola lelang. Lelang online menyediakan ruang yang cukup sekalipun semua manusia didunia berpartisipasi.

4. Jumlah Penjual yang Besar.

Dengan adanya jumlah penawar yang besar, tentunya juga memicu penjual untuk memasang barangnya, tentunya ini harus didukung oleh kemudahan-kemudahan yang disediakan pada website yang bersangkutan.

5. Jaringan Ekonomi

Dengan adanya jumlah penawar yang besar, akan memicu lebih banyak penjual, demikian juga sebaliknya. Hal ini akan menimbulkan siklus ekonomi permintaan dan persediaan, menjadikan sebuah sistem yang berguna bagi pesertanya.

## 2.4. KOMPONEN PENDUKUNG SISTEM LELANG ONLINE

### 2.4.1 Sistem Operasi

Sistem operasi yang dibutuhkan disini adalah sistem operasi yang mendukung sepenuhnya untuk dijadikan komputer *server*. Disini akan ditempatkan semua data atau dokumen yang diperlukan untuk proses pembuatan tabel, pembuatan *database*, hingga keamanan data sangat diperlukan. Dalam perangkat lunak PHP menggunakan bahasa scripting yang menyatu dengan HTML dan berada di server (*server-side scripting*). Artinya sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di *server*. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi-aplikasi yang dijalankan oleh *web*. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *web server*.

### 2.4.2 Internet

Secara teknis, internet (*interconnection network*) merupakan sebuah bangunan yang terdiri atas kumpulan dari banyak sekali kumpulan komputer yang saling terhubung dan membentuk suatu jaringan yang tidak mempermasalahkan perbedaan geografis.

### 2.4.3 Local Area Network (LAN) dan Topologinya.

Adalah sejumlah komputer yang saling dihubungkan didalam suatu areal tertentu yang tidak begitu luas. Secara garis

besar, terdapat dua tipe jaringan LAN, yaitu: Jaringan *Peer to Peer* dan jaringan *Client-Server*. Pada jaringan *Peer to Peer*, setiap komputer yang terhubung ke jaringan dapat bertindak baik sebagai *workstation* maupun *Server*. Sedangkan pada jaringan *Client-Server*, hanya satu komputer yang bertugas sebagai *server* dan komputer lain berperan sebagai *workstation*. Antara dua tipe jaringan tersebut masing-masing memiliki keunggulan dan kelemahan.

#### 2.1.1.1 Topologi jaringan

Adalah gambaran secara fisik dari pola hubungan antara *komponen-komponen* jaringan, yang meliputi *server*, *workstation*, *hub*, dan pengkabelan. Terdapat tiga macam topologi jaringan umum digunakan, yaitu:

1. Topologi *Bus*: Digunakan sebuah kabel tunggal atau kabel pusat dimana seluruh *workstation* dan *server* dihubungkan.
2. Topologi *Star*: Masing-masing *workstation* dihubungkan secara langsung ke *server* atau *hub*.
3. Topologi *Ring*: Di dalam semua topologi *ring*, semua *workstation* dan *server* dihubungkan sehingga terbentuk suatu pola lingkaran atau cincin. Tiap *workstation* ataupun *server* akan menerima dan melewatkan informasi dari satu komputer ke

komputer lain, bila alamat-alamat yang dimaksud sesuai maka akan diterima dan bila tidak sesuai maka akan dilewatkan.

#### 2.1.1.2 Sistem Operasi Jaringan

*Untuk* mengelola suatu jaringan diperlukan adanya sistem operasi jaringan. Sistem ini dibedakan menjadi dua berdasarkan tipe jaringannya, yaitu: sistem operasi *Client-Server* dan sistem operasi *Peer to Peer*.

1. Jaringan *Client-Server*: *server* di jaringan ini disebut dengan *dedicated server* karena murni berperan sebagai *server* yang menyediakan fasilitas kepada *workstation* dan *server* tersebut tidak dapat berperan sebagai *workstation*.
2. Jaringan *Peer To Peer*: *server* di jaringan ini diistilahkan *non-dedicated server*, karena *server* tidak berperan sebagai *server* murni melainkan sekaligus dapat berperan sebagai *workstation*.

#### 2.4.4 Pemrograman database berbasis web

Pemrograman *Database Berbasis Web*. Salah satu strategi pengelolaan yang penting untuk diperhatikan adalah pengelolaan pada media penyimpan informasi tersebut, yaitu pengelolaan basis data. Proses back up dan restore data, pemeliharaan validitas data adalah hal terpenting kedua setelah pengelolaan sistem utama.

#### 2.4.5 Pemrograman aplikasi atau web

1. *PHP* (PHP: *Hypertext Preprocessor*). Merupakan sebuah bahasa pemrograman dimana file-filenya diletakkan di server dan seluruh prosesnya dikerjakan di server, kemudian hasilnya dikirimkan kepada klien, tempat pemakai menggunakan browser (lebih dikenal dengan istilah *server-side scripting*). Jadi setiap data yang diminta atau secara *default* ada dalam *site* akan diproses dikomputer server tanpa melibatkan komputer *client*. Komputer client hanya menampilkan hasil dari proses aksi yang diminta.

Versi pertama PHP dibuat oleh Rasmus Lerdorf pada tahun 1995. Versi pertama ini berupa sekumpulan skrip PERL yang digunakan untuk membuat *web* pada home page pribadinya. Rasmus kemudian menulis ulang skrip-skrip PERL tersebut dengan bahasa C. pada tahun 1997 PHP mencapai rilis ketiga dimana pengembangan selanjutnya tidak hanya oleh Rasmus sendiri tetapi melibatkan programmer lain yang antusias untuk mengembangkan PHP diantaranya Zend Suraski dan Andi Gutmans. Kedua programmer ini menyempurnakan bahasa PHP dengan menambahkan integrasi Zend Engine.

PHP bekerja di dalam sebuah dokumen HTML (*Hypertext Markup Language*) untuk dapat menghasilkan isi

dari sebuah halaman *web* sesuai permintaan. Dengan PHP, kita dapat merubah sebuah situs menjadi sebuah aplikasi berbasis *web* yang dinamis.

PHP sebagai bahasa pemrograman *web* memiliki berbagai keunggulan diantaranya:

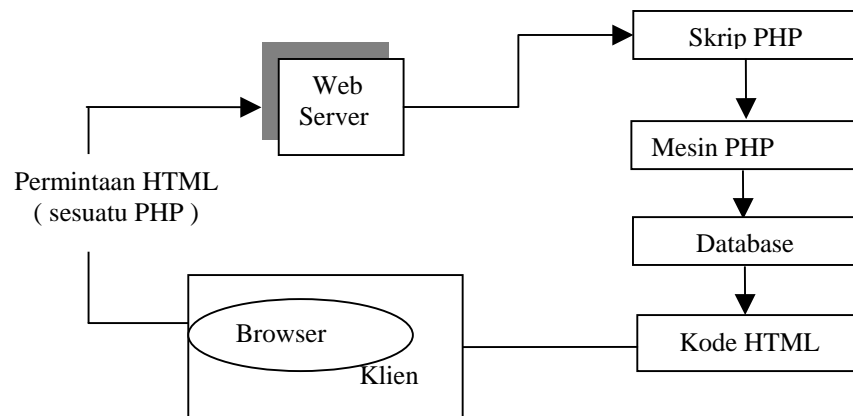
- a. *Open source*, dikarenakan PHP bersifat *open source* maka banyak programmer di seluruh dunia yang terlibat dalam pengembangannya sehingga bug atau kesalahan pada program dapat lebih cepat diatasi, selain itu penambahan fasilitasnya juga lebih cepat.
- b. *Cross Platform*, PHP dapat dipakai di hampir semua server *web* yang ada di pasaran, antara lain Apache, AOLServer, Xitami, Microsoft IIS, dan sebagainya dan dapat dijalankan diberbagai sistem operasi seperti Microsoft, Linux, FreeBSD, Solaris, Machintosh dan sebagainya. PHP mendukung banyak sekali paket *database* baik yang komersial seperti Microsoft SQL Server, Oracle, Informix, Interbase, Sybase, maupun yang non-komersial, seperti PostgreSQL, MySQL, dan FireBird.
- c. *Kecepatan proses* yang tinggi, PHP engine yang merupakan *inti* dari pemroses skrip PHP dibuat dengan

bahasa C yang sudah dikenal akan keampuhan dan kecepatannya.

2. *Cara Kerja PHP*. Model kerja HTML, diawali dengan permintaan suatu hal web oleh *browser*. Berdasarkan URL ( *Uniform Resource Locator* ) atau sebuah alamat internet, *browser* mendapatkan alamat dari web server, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *web server* akan mencari berkas yang diminta dan memberikan isinya ke *browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan menampilkan ke layer pemakai. Apabila yang diminta adalah sebuah halaman PHP maka prinsipnya serupa dengan HTML, hanya saja saat berkas PHP yang diminta didapatkan oleh web server, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang akan memproses dan memberikan hasilnya ( berupa kode HTML ) ke *web server*. Selanjutnya, *web server* menyampaikan ke klien. Apabila yang diminta adalah sebuah halaman PHP maka prinsipnya serupa dengan HTML, hanya saja saat berkas PHP yang diminta didapatkan oleh *web server*, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang akan memproses dan memberikan hasilnya ( berupa kode HTML ) ke *web server*.



Selanjutnya, *web server* menyampaikan ke klien. Gambar 2.1 memperlihatkan skema HTML sedangkan gambar 2.2 memperlihatkan skema PHP.



Gambar 2.2 Skema PHP

### 3. Elemen Dasar PHP

Elemen dasar dari PHP meliputi :

- a. Karakter. Adalah bahasa pemrograman komputer yang dapat berupa huruf, sebuah angka tunggal, sebuah spasi, tanda control seperti *carriager return*, atau sebuah simbol seperti + dan ?.
- b. Pengenal. Pengenal ( *identifier* ) bannyak digunakan dalam program untuk memberi nama variable, fungsi, atau kelas. Oleh karena itu mengetahui aturan penamaan pengenal adalah hal yang penting. Aturan yang berlaku untuk pengenal :

1. Karakter yang dapat digunakan adalah huruf, angka, atau garis bawah ( \_ ).
  2. Karakter pertama harus berupa huruf atau garis bawah.
  3. Panjang pengenalan bisa berapa saja.
  4. Huruf kecil dan huruf kapital harus dibedakan.
4. Tipe Data

Tipe data pada PHP ada tiga macam, yaitu :

1. Integer
2. Double
3. String

Integer menyatakan tipe data bilangan bulat dengan jangkauan kira-kira - 2 milyar hingga + 2 milyar. Double menyatakan tipe bilangan real atau titik mengambang, yaitu bilangan yang mempunyai bagian pecahan. String menyatakan tipe data teks ( sederetan karakter yang tidak menyatakan bilangan ), seperti nama orang.

5. Konstanta. Konstanta menyatakan nilai yang tetap di dalam program. Sebagai contoh pada pernyataan :

```
Print ( "hallo" );
```

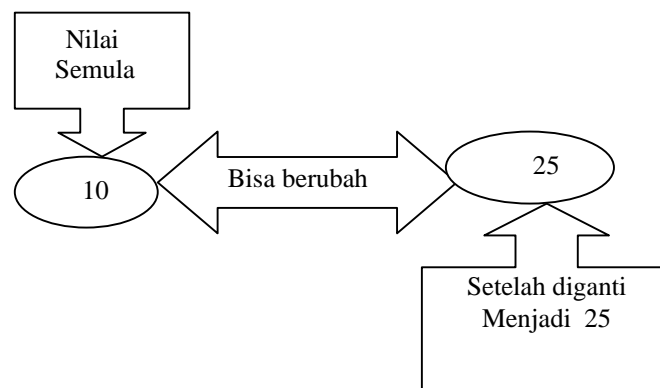
“ Hallo “ adalah jkonstanta string. Perlu dimengerti, dalam PHP dikenal sejumlah karakter yang menggunakan penulisan secara khusus, yaitu didahului dengan simbol

*backslash* ( \ ). Misalnya \n berarti *newline* ( karakter pindah baris ). Contohnya untuk menulis “ selamat belajar “ , pada *browser* adalah sebagai berikut :

`print (“\” selamt belajar\””):`. Untuk mengujinya bias dengan menulis skrip di bawah ini :

```
< HTML >
< HEAD >
< TITLE > Contoh penulisan karakter dengan \ “ </TITLE>
</ HEAD >
< BODY >
  <?php
    $bahasa = “ PHP ”;
    print (“\” Selamat Belajar \” < BR > \n ”);
    printf(‘\ $bahasa = $bahasa ’);
  ?>
</ BODY >
</ HTML >
```

6. Variabel. Variabel digunakan dalam program untuk menyimpan nilai yang berubah-ubah. Variabel dapat dibayangkan seperti suatu kotak. Mula-mula menyimpan nilai 10 ke dalam kotak tersebut. Di saat yang lain dimungkinkan mengganti isinya dengan 25. Gambar 2.3 memperlihatkan ilustrasi suatu variabel.



### Gambar 2.3 Ilustrasi variable

Sebagaimana diketahui, Variabel pada PHP selalu ditulis dengan awalan berupa \$. Sebagai contoh, program berikut memberikan gambaran tentang pengubahan variabel.

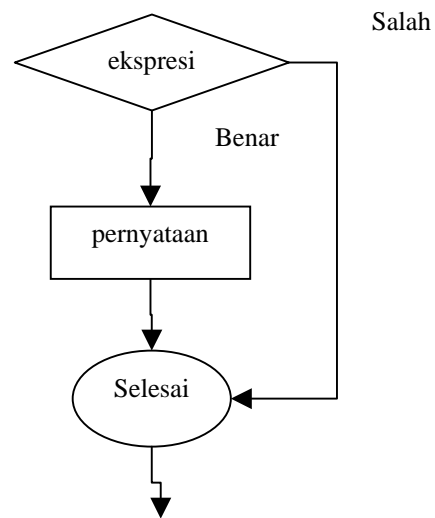
```
< HTML >
< HEAD >
< TITLE >Contoh kenaikan isi variable</Title>
</ HEAD >
< BODY >
  <?php
    $gaji = 20000;
    printf(“Gaji semula = %d < BR >\n”, $gaji );
    $ = 1.5 * $gaji;
    printf( “Gaji sekarang = % < BR >\n“, $gaji );
  </ BODY >
</ HTML >
```

7. Pernyataan Kontrol. Pernyataan kontrol ini dengan pengambilan keputusan ( **if** dan **switch** ), pengulangan ( **while,do-while, for** ) serta pernyataan ( **break, continue, dan exit** ). Adapun pernyataan-pernyataan tersebut adalah sebagai berikut.

a. Pernyataan **if**. Biasa dipakai untuk mengambil keputusan berdasarkan suatu kondisi. PHP mempunyai tiga macam bentuk **if**, yaitu :

Bentuk pertama **if** berupa :

**If** ( ekspresi )



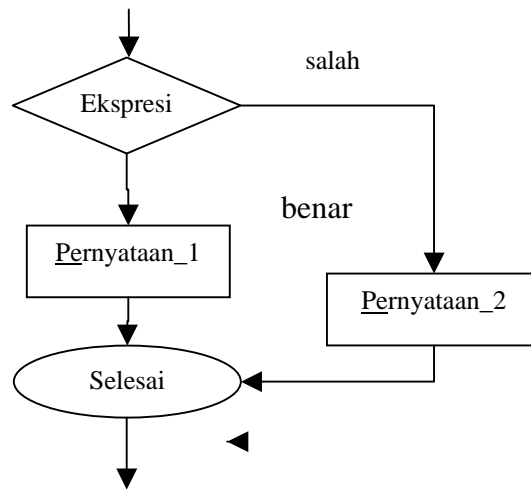
Gambar 2.4 Diagram pernyataan if

- b. Pernyataan **if-else**. Bila menginginkan untuk menjalankan suatu tindakan tertentu bila kondisi bernilai benar dan menjalankan tindakan yang lain kalau kondisi bernilai salah.

**If** ( ekspresi )  
     *Pernyataan \_1*  
**Else**  
     *Pernyataan \_2*

Pada bentuk ini bagian pernyataan\_1 dijalankan kalau ekspresi bernilai benar, dan bagian pernyataan\_2 dijalankan kalau ekspresi bernilai salah.

Gambar diagramalir if-else.



Gambar 2.5 Diagram alir if-else

Bentuk if-else juga bias berbentuk :

```

If ( ekspresi )
{
    pernyataan_1
}
else
{
    Pernnyatan_2
}
  
```

- c. Pernyataan if-elseif. Pernyataan if-elseif sangat bermanfaat untuk melakukan pengambilan keputusan yang melibatkan banyak alternative. Sebagai contoh pada skrip berikut, if-elseif digunakan untuk menentukan nama hari sekarang ( diambil dari sistem ).

```

< HTML >
< HEAD >
< TITLE > latihan menentukan nama hari </TITLE>
</HEAD >
Hari ini :
<?php
    $nama_hari = date("l");
    if ($nama_hari == "Sunday")
  
```

```

        print("Minggu");
elseif ($nama_hari == "Monday")
    print("Senin");
elseif ($nama_hari == "Tuesday")
    print("Selasa");
elseif ($nama_hari == "Wednesday")
    print("Rabu");
elseif ($nama_hari == "Thursday")
    print("Kamis");
elseif ($nama_hari == "Friday")
    print("Jumat");
elseif ($nama_hari == "Saturday")
    print("Sabtu");
?>
</BODY >
</HTML >

```

Pada akhir skrip diatas, pernyataan if-elseif digunakan untuk menentukan nama hari sekarang (berdasar sistem). Mula-mula, pernyataan berikut:

```
$nama_hari = date("l");
```

dipakai untuk memperoleh hari dalam bahasa Inggris. Kemudian nilainya adalah Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, dan Saturday. Selanjutnya nilai-nilai ini ditangani oleh if-elseif, bila menggunakan if\_else, perintahnya akan menjadi lebih panjang.

- d. Pernyataan switch. Bila menjumpai persoalan yang melibatkan banyak alternative semacam penentuan hari pada skrip di depan, juga bisa digunakan pernyataan switch. Secara umum, banyak pernyataan sebagai berikut :

```

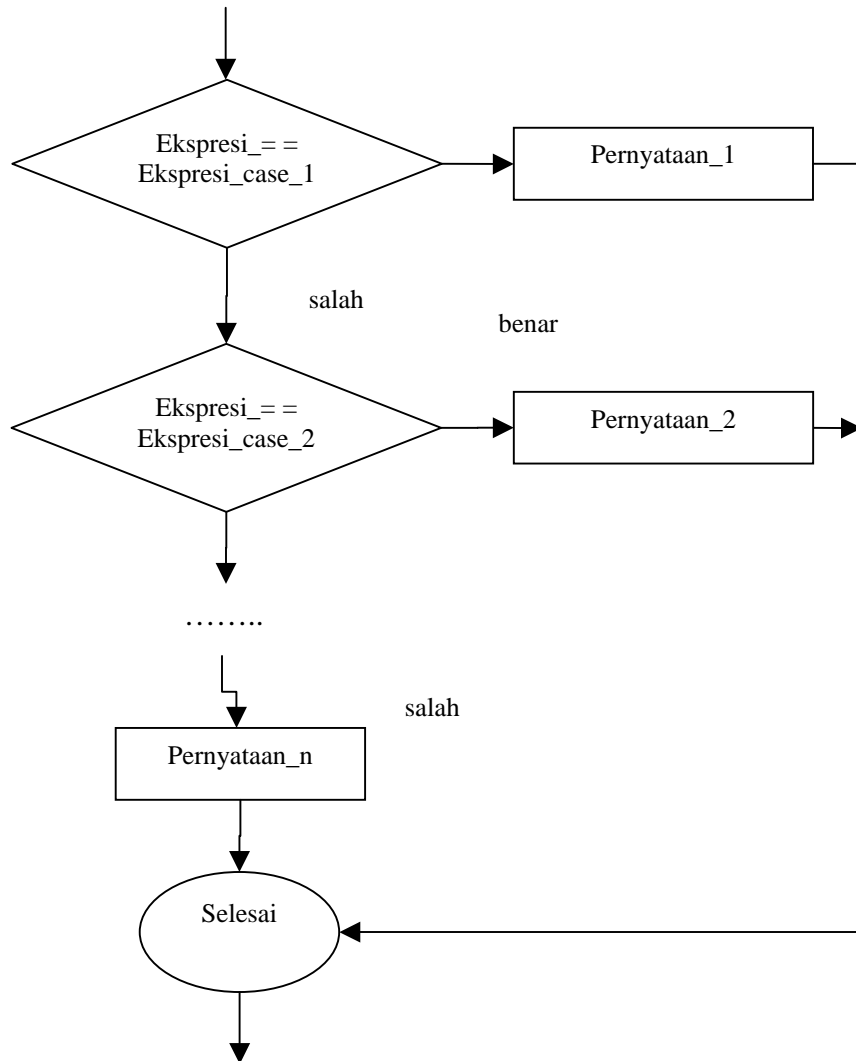
Switch ( ekspresi )
{
    case ekspresi_case_1 ;
        pernyataan_1;
    break;
    case ekspresi_case_2;
        pernyataan_2;
    break;
    case ekspresi_case-1;
        pernyataan_1;
    break;
    case ekspresi_case_3;
        .....
    default:
        Pernyataan_n
}

```

Pada pernyataan switch, *ekspresi\_case\_1* akan diperiksa terlebih dahulu. Bila nilainya cocok dengan nilai *ekspresi* maka pernyataan\_1 akan dijalankan dan kemudian eksekusi dilanjutkan ke pernyataan yang terletak sesudah tanda penutup switch ( } ). Kalau tidak cocok, perbandingan nilai *ekspresi* dan *ekspresi\_case\_2* akan dilakukan. Jika hasil pada perbandingan yang kedua ini benar maka pernyataan-2 akan dijalankan, dan kemudian eksekusi akan dilanjutkan ke pernyataan yang terletak sesudah tanda penutup switch ( } ). Sekiranya tidak ada bagian *ekspresi\_case* yang cocok, maka pernyataan\_n yang mengikuti kata default akan dijalankan.



Gambarkan dibawah ini memperlihatkan diagram alir *switch*.



Gambar 2.6 Diagram alir switch

Pada pernyataan switch, pernyataan break memegang peranan sangat penting. Pernyataan break-lah yang membuat eksekusi dilanjutkan ke pernyataan yang terletak sesudah tanda penutup switch ( } ). Gambar efek break.

```

Switch ( ... )
{
  Case ....;
  ....
  break;
  ....
  brerak;
}

```

Gambar Efek break

- e. Pernyataan while. Adalah pernyataan yang berguna untuk melakukan suatu pengulangan. Sebagai contoh untuk menampilkan bilangan 1 sampai dengan 20 cukup dengan menggunakan kode yang pendek. Bentuk pernyataan tersebut adalah :

```

While ( ekspresi )
{
    Pernyataan_pernyataan
}

```

Pernyataan while akan memeriksa nilai ekspresi terlebih dahulu. Jika bernilai benar maka pernyataan\_pernyataan yang terdapat dalam { } akan dijalankan dan ekspresi dievaluasi lagi. Proses ini diulang terus-menerus sampai ekspresi bernilai salah. Contoh berikut menunjukkan penggunaan while untuk menampilkan bilangan 1 sampai 20.

```

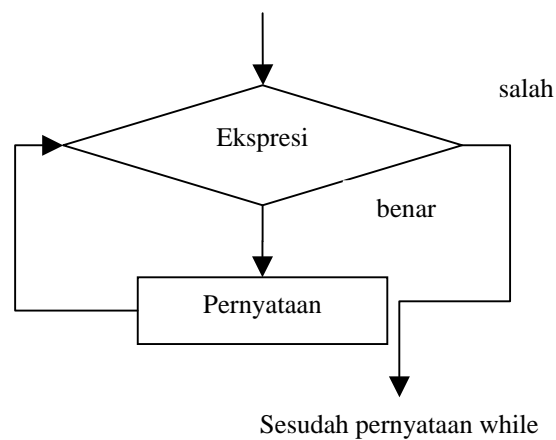
< HTML >
< HEAD >
< TITLE > menampilkan bilangan 1-20</TITLE >
</ HEAD >

```

```

< BODY >
< ? PHP
$ bilangan = 1 ;
while ( $bilangan < = 20 )
print { “ bilangan < BR >/n”};
$bilangan++;
}
?>
< / BODY >
< HTML >

```



**Gambar 2.7 Diagram Alir While**

- f. Pernyataan do-while. Pernyataan do-while berfungsi serupa dengan while.

Bentuk pernyataannya sebagai berikut :

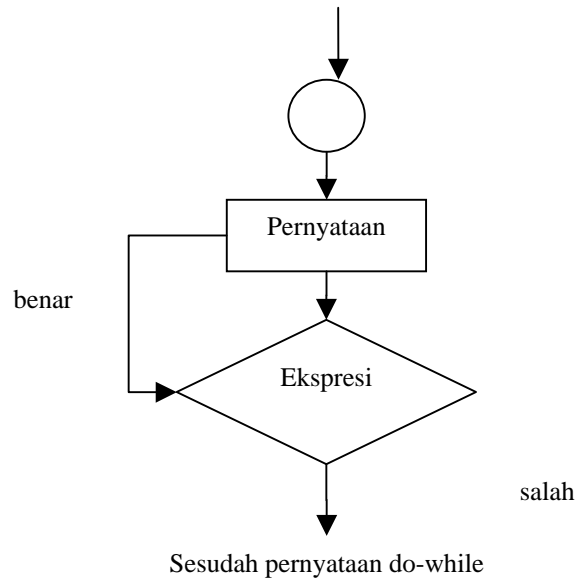
```

Do
{
pernyataan_pernyataan
} while ( ekspresi );

```

Pengulangan akan berakhir jika ekspresi (yang diuji sesudah pernyataan\_pernyataan yang dijalankan)

bernilai salah. Gambar dibawah memperlihatkan diagram alir pernyataan do-while.



Gambar 2.8 diagram alir pernyataan do-while

Berdasarkan diagram alir di atas terlihat bahwa paling tidak pernyataan yang berada dalam { } akan dieksekusi sekali. Skrip di bawah ini menunjukkan penggunaan do-while untuk menampilkan bilangan 1 sampai 20.

```

< HTML >
< HEAD >
< TITLE > menampilkan bilangan 1_20</TITLE>
</HEAD >
< BODY >
  <? Php
    $bilangan = 1;
  do
  {
    print ( "$bilangan <BR >\n" );
    $bilangan++;
  } while ( $bilangan <21);
  ?>
  
```

```
</BODY >
</HTML >
```

- g. Pernyataan for. Pernyataan for juga merupakan pernyataan yang biasa digunakan untuk menangani pengulangan proses. Pernyataan ini mempunyai bentuk sebagai berikut :

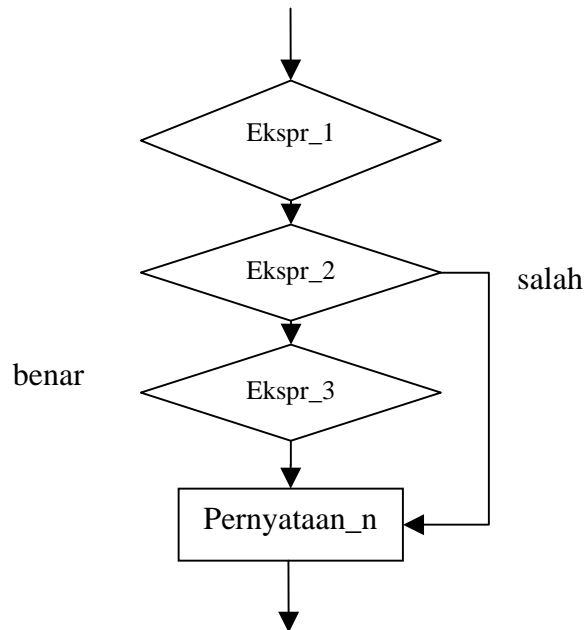
```
For ( ekspresi;ekspr2; eksp3 )
{
    pernyataan_pernyataan
}
```

Apabila yang terletak antara tanda { dan } hanya sebuah pernyataan, tanda { dan } bias dituliskan. Pernyataan diatas identik dengan :

```
Ekspr1;
While ( eksp2 )
{
    pernyataan_pernyataan
}
ekspr3;
```

Keterangan :

1. eksp1 adalah ekspresi untuk memberikan nilai awal terhadap variable yang akan digunakan pencacahan pengulangan.
2. eksp2 berlaku sebagai kondisi untuk menentukan pengulangan terhadap pernyataan yang ada di dalam { } akan dilakukan atau tidak.
3. eksp3 digunakan untuk mengatur nilai variable yang digunakan dalam eksp1



Gambar 2.9 diagram alir pernyataan for

- h. Pernyataan exit. Pernyataan exit berguna untuk mengakiri pengekseskuan. Dengan dijalankan perintah ini, segala kode baik PHP maupun HTML tidak dikirim ke *browser*. Untuk melihat exit, perhatikan skrip dibawah ini.

```

< HTML >
< HEAD >
< TITLE >Contoh Menperlihatkan efek exit</TITLE
>
</ HEAD >
< BODY >
  <?php
    for ($ = 1; $1 = > 25 ; $1++ )
    {
      print ( "$bilangan <BR>\n");
      if ( $1 == 10 )
        exit;
    }
    print ( " selesai < BR >\n" );
  ?>
  
```

</BODY >  
</HTML >

#### 2.4.5.1 Basis Data (Database)

Adalah untuk menyimpan semua data atau hasil dari proses penilaian, yang bisa langsung diakses oleh komputer server. Pemilihan dan konfigurasi basis data harus mempertimbangkan kestabilan dan fleksibilitas, sehingga data yang tersimpan bisa lebih terstruktur yang juga akan membantu *web* server dalam hal waktu akses data. Basis data tersimpan di perangkat keras dalam komputer server, yang dimanipulasi dengan menggunakan perangkat lunak. MySQL merupakan salah satu *software* yang bisa menangani semua proses pengaturan data dalam *database*. Selain menyediakan tempat penyimpanan yang luas, MySQL juga sangat kompatibel bila digabungkan dengan bahasa pemrograman *web* dinamis seperti PHP. Kehandalan MySQL tidak diragukan lagi, hal ini telah dibuktikan dengan banyaknya *web* server yang menggunakannya.

Tabel 2.1. Karakteristik MySQL.

No	Karakteristik	Deskripsi
1.	Standard	MySQL mendukung <i>entry-level</i> ANSI SQL-92 dan ODBC level 0-2
2.	<i>Character set</i>	MySQL secara <i>default</i> menggunakan ISO-8859-I (Latin I) untuk data dan pengurutan.
3.	Bahasa pemrograman	MySQL mendukung aplikasi dalam bahasa Java, C, Perl, PHP.

4.	Kecepatan dan kemudahan pemakaian	MySQL tiga sampai empat kali lebih cepat dibandingkan dengan basisdata komersil, juga lebih mudah dikelola.
5.	MySQL	MySQL adalah <i>open source relational</i> DBMS.

MySQL memiliki tipe-tipe kolom, yaitu :  
 INTEGER (*signed/unsigned*) dengan ukuran 1 – 8 byte,  
 FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB,  
 DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET,  
 ENUM dan tipe-tipe khusus lainnya.

#### 2.4.5.2 Structured Query Language ( SQL )

SQL adalah bahasa yang mulanya berorientasi pada basis data relasional. Bahasa ini menghilangkan banyak pekerjaan yang perlu dilakukan pemrogram berkaitan dengan operasi-operasi terhadap data. Meski SQL diacu sebagai bahasa *query* (yaitu bahasa untuk meminta data) namun sesungguhnya SQL bukan hanya sekedar bahasa *query*. SQL juga berisi fasilitas-fasilitas untuk mendefinisikan struktur data, modifikasi struktur data dan data pada basis data, serta juga digunakan untuk menspesifikasikan konstrain-konstrain integritas dan keamanan. SQL terdiri dari perintah-perintah yang dapat dikelompokkan menjadi beberapa bagian :



1. DDL (*Data Definition Language*).

Berisi perintah-perintah menciptakan dan menghancurkan objek-objek basis data, seperti tabel, indek dan *view*. Perintah-perintah dalam DDL:

a. CREATE nama\_objek.

Digunakan untuk membuat objek/tabel.

b. ALTER nama\_objek.

Digunakan untuk melakukan penambahan, penghapusan atau pengubahan definisi tabel yang sudah ada.

c. DROP nama\_objek.

Digunakan untuk menghancurkan objek/tabel.

2. DML (*Data Manipulation Language*).

Berisi perintah-perintah pengambilan, penyisipan, perbaruan dan penghapusan data dari tabel yang telah didefinisikan.

Perintah-perintah yang digunakan dalam DML :

a. **SELECT.**

Untuk menghasilkan suatu informasi yang tepat dari suatu tabel.

Sintak :

```
SELECT [*]nama_kolom] FROM [nama_tabel]
SELECT [*]nama_kolom] FROM [nama_field] WHERE
[kondisi]
SELECT [*]nama_kolom] FROM [nama_field] WHERE
[kondisi1] AND [kondisi2] ...AND [kondisi n]
SELECT [*]nama_kolom] FROM [nama_field] WHERE
[kondisi1] OR [kondisi2] ...OR [kondisi n]
```

b. **INSERT.**

Digunakan untuk menambah data baru ke dalam sebuah tabel.

Sintak :

```
INSERT INTO nama_tabel VALUES (val1,
val2, val3, ... valN)
```

c. **DELETE.**

Digunakan untuk menghapus data dalam suatu tabel.

Sintak :

```
DELETE FROM nama_tabel [WHERE kondisi]
```

d. **UPDATE.**

Digunakan untuk mengubah data yang telah ada pada tabel.

Sintak:

```
UPDATE nama_tabel SET nama_kolom = nilai
[WHERE kondisi]
```

2.4.5.3 *MySQL Database Server*: adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris: *database management system*) atau *DBMS* yang *multithread*, *multi-user*. Untuk dapat menggunakan semua perintah *SQL* pada PHP, kita memerlukan suatu fungsi *MySQL* .

Tabel.2.1. Fungsi-Fungsi *MySQL*

Fungsi <i>MySQL</i>	Penggunaan
<i>MySQL_change_user</i> ( )	Merubah kunci pengguna dalam pengaktifan koneksi
<i>MySQL_close</i> ( )	Digunakan untuk menutup koneksi dengan <i>MySQL</i>
<i>MySQL_connect</i> ( )	Untuk membuka koneksi dengan database <i>MySQL</i> server
<i>MySQL_create_db</i>	Untuk membuat database
<i>MySQL_db_name</i> ( )	Untuk menghasilkan hasil data
<i>MySQL_db_Query</i> ( )	Untuk mengantarkan <i>Query MySQL</i>
<i>MySQL_drop_db</i> ( )	Untuk menghapus database
<i>MySQL_errno</i> ( )	Untuk menampilkan pesan kesalahan dalam bentuk no dsri server <i>MySQL</i>
<i>MySQL_error</i> ( )	Untuk menampilkan pesan kesalahan dalam bentuk teks dari server <i>MySQL</i>
<i>MySQL_fetch_array</i> ( )	Menghasilkan data berupa array dalam bentuk angka dari isi tabel <i>MySQL</i>
<i>MySQL_fetch_assoc</i> ( )	Menghasilkan data berupa array dalam bentuk field dari tabel <i>MySQL</i>
<i>MySQL_fetch_field</i> ( )	Menghasilkan informasi kolom dari hasil yang kemudian dikembalikan sebagai suatu objek
<i>MySQL_fetch_lengths</i> ( )	Menghasilkan basarfile dari hasil tabel <i>MySQL</i>
<i>MySQL_fetch_object</i> ( )	Menghasilkan sebuah baris sebagai suatu objek
<i>MySQL_fetch_row</i> ( )	Menghasilkan array/ baris dengan keluaran nama field pada tabel <i>MySQL</i>
<i>MySQL_field_name</i> ( )	Menghasilkan nama field khusus pada database <i>MySQL</i>
<i>MySQL_field_len</i> ( )	Mnembalikan besar file
<i>MySQL_num_row</i> ( )	Memeproleh jumlah baris dari suatu hasil <i>Query</i> yang menggunakan perintah <i>SELECT</i>
<i>MySQL_pconnect</i> ( )	Menbuka koneksi langsung dengan <i>MySQL</i>
<i>MySQL_Query</i>	Digunakan untuk mengirimkan perintah <i>SQL</i>
<i>MySQL_result</i> ( )	Untuk menghasilkan data
<i>MySQL_select_db</i> ( )	Digunakan untuk masuk pada database <i>MySQL</i>

#### 2.4.5.4 Apache Web Server

Apache merupakan server *web* yang paling sering digunakan sebagai server Internet dibandingkan dengan server *web* lainnya. *Web server (World Wide Web Server)* merupakan server Internet yang mampu melayani koneksi transfer data dalam protokol HTTP (*Hypertext Transfer Protocol*). Saat ini, *Web server* merupakan bentuk dari server-server internet selain *e-mail server*, *ftp server*, dan *news server*. *Web server* sendiri dirancang untuk dapat melayani beragam jenis data, di antaranya *text*, *hypertext*, gambar (*image*), suara, gambar tiga dimensi, *plug-in*, dan lainnya.

Dengan model sistem modular yang dimiliki, *Web Server Apache* mampu menyediakan dukungan terhadap berbagai bahasa pemrograman *web*, seperti PHP, Perl dan JSP. *Web server apache* dapat dijalankan pada sistem operasi Unix, Linux maupun Windows sehingga *developer* tidak akan kesulitan dalam lingkungan multiplatform.

*Apache Web Server* dibuat berdasarkan NCSA HTTP *daemon*. NCSA (*National center for Supercomputing Applications*) merupakan suatu lembaga penelitian di Universitas Illinois Amerika Serikat. Sejak didirikan grup Apache sekitar tahun 1995 banyak para

programmer yang ikut serta memberikan kontribusi dalam pengembangan *software* Apache.

Apache digunakan sebagai *web* server yang mengolah script php, php digunakan untuk melakukan proses-proses aritmatika, database dan lain-lainya, sedangkan MySQL digunakan untuk pembuatan *database* yang terhubung dengan halaman *web* yang dilakukan oleh script php.

*Web* Server Apache terdiri dari beberapa bagian, kode source apache asli dengan beberapa patch dan beberapa *executable* tambahan.

Sebaiknya kita gunakan *Web* Server Apache yang tergolong 'stable version', yang mungkin akan lebih mudah untuk dikompilasi dan akan menghasilkan daemon yang cukup stabil.

Beberapa direktori penting untuk Apache:

1. cgi-bin/ - direktori CGI scripts
  2. conf/ - konfigurasi file httpd server
  3. htdocs/ - main direktori untuk dokumen
  4. logs/ - direktori dengan log files
- other stuff (bin/, icons/, include/, proxy/, man/.....)