BELAJAR PHP

Filenya ada di PHPDasar, di htdocs

Pertemuan 9 > menghubungkan html ke database

Pertemuan 10 > menambahkan fitur tambah (create) ke database dan delete data didatabase

Pertemuan 11> menambahkan fitur ubah (edit) database

Pertemuan 12 > menambah fitur pencarian data

Pertemuan 13 > menambah fitur upload

Pertemuan 14> menambahkan fitur registrasi

Pertemuan 15> menambahkan fitur login

Pertemuan 16> menambahkan fitur session, agar tidak ke index tanpa pernah login

Pertemuan 17> menambahkan cookie sehingga bisa ada fitur rememberme yaitu kita tak harus login dulu

Pertemuan 18> menambahkan fitur pagination, agar terdapat fitur sorting

Pertemuan 19> menambahkan fitur session, agar tidak ke index tanpa pernah login

Pertemuan20> mengganti script.js menjadi jquery dan menambahkan gif loader

* **SINTAKS MYSQLI**

1. mysqli\_connect

ini untuk mengkoneksikan atau menghubungkan ke database

terdapat 4 parameter,

contoh penggunaan:

// koneksi ke database, paraneternya "namahost", "root", "passwprdnya(isikosoong)", "namadatabase"

$conn = mysqli\_connect("localhost", "root", "", "phpsepatu");

1. Mysqli\_query> ini untuk mengambil data dari tabel yang diinginkan atau mengquery data tabel

Penggunaan :

//ambil data dari tabel sepatu / query data sepatu, sintak sql ditulis besar

$result = mysqli\_query($conn, "SELECT \* FROM sepatu");

1. Mysqli\_fetch\_row / mysqli\_fetch\_assoc

Ini akan mengembalikan array, dimana terdapat index angka untuk row

Sedangkan assoc, mengembalikan array asosiasi yang menggunakan nama untuk panggilnya

* **ARRAY**

Nilai yang ada di dalam array disebut elemen .

Untuk menampilkan array bisa dengan var\_dump() atau print\_r(), ini untuk proses debug.

**Var\_dump(); die;** >>>ini untuk menampilkan isi, lalu kode dibawahnya tidak akan dijalankan.

Contoh **array numerik** yaitu array yang keynya berupa index, contoh

$mahasiswi = [

['rizky ananda f', '231321321', 'nanda@gmail,com', '90'],

['rizky f', '23132132142321', 'nanda@gmai23,com', '12'],

];

Contoh **array associative,** yaitu keynya adalah string yang kita buat sendiri

<?php

$mahasiswi =[

   [

      'Nama' => 'Sepatu Satu',

      'Warna' => 'Putih',

      'Ukuran' => 39,

      'Merek' => 'Nike',

      'Gambar' => 'sepatu1.jpg',

   ],   [

      'Nama' => 'Sepatu Dua',

      'Warna' => 'Hitam',

      'Ukuran' => 41,

      'Merek' => 'Nike',

      'Gambar' => 'sepatu2.jpg',

   ],   [

      'Nama' => 'Sepatu Tiga',

      'Warna' => 'Abu',

      'Ukuran' => 40,

      'Merek' => 'Nike',

      'Gambar' => 'sepatu3.jpg',

   ],   [

      'Nama' => 'Sepatu Empat',

      'Warna' => 'Biru',

      'Ukuran' => 39,

      'Merek' => 'Nike',

      'Gambar' => 'sepatu4.jpg',

   ],   [

      'Nama' => 'Sepatu Lima',

      'Warna' => 'Putih',

      'Ukuran' => 43,

      'Merek' => 'Nike',

      'Gambar' => 'sepatu5.jpg',

   ],

];

?>

<?php foreach ($mahasiswi as $mhs )  { ?>

<ul>

   <li> <img src="gambar/<?= $mhs['Gambar']?>" alt=""></li>

   <li><?php  echo $mhs['Nama'] ?></li>

   <li><?php  echo $mhs['Warna'] ?></li>

   <li><?php  echo $mhs['Ukuran'] ?></li>

   <li><?php  echo $mhs['Merek'] ?></li>

</ul>

* **SUPERGLOBALS**

Var yang ada di dalam function, berbeda dengan variable yang diluar function

<?php

$x= 10;

*function* Tampilx()

{

   $x = 20;

   echo $x;

}

echo $x;

echo "<br>";

echo Tampilx();

?>

Output

10

20

Untuk akalinnya, sehingga x di dalam tampilX menjad 10 dengan cara pakai **global**

<?php

$x= 10;

*function* Tampilx()

{

   global $x;

   echo $x;

}

Tampilx();

?>

Terdapat juga **superglobals,** ini variable yang bisa kita akses dimanapun,

**Superglobals** adalah tipenya **array associative**

Ini sudah disediakan oleh php, yaitu

1. $\_GET > ini akan menambahkan data di urlnya,
2. $\_POST >ini akan mengambil data yang diisimpan dalam array
3. $\_REQUEST
4. $\_SESSION
5. $\_COOKIE
6. $\_SERVER
7. $\_ENV

**Isset() >** ini mengecek apakah variable sudah pernah dibuat atau belum.

<?php

if (!isset($\_GET['nama'])) {

   header('Location: baru1.php');

   exit;

}

?>

Fungsi ini untuk, ketika variable nama belum pernah dicetak, maka kembali ke halaman baru1.php, dan exit untuk tidak mengeksekusi baris dibawahnya.

//Ketika tombol sumbit blm pernah ditekan, maka jangan munculkan h1, dan sebaliknya, ketika sudah  ditekan munculkan nama dari input

if (isset($\_POST['submit'])) : ?>

   <h1>Selamat Datang, <?= $\_POST['nama'] ?></h1>

<?php endif;  ?>

Di atas, ketika kita blm klik submit, maka tidak akan muncul selamat datang,

Tapi setelah kita mengisi nama di from dan mengklik submit, maka akan muncul tulisan selamat datang nama

<?php

//cek apakah tombol submit sudah tekan apa blm

if (isset($\_POST['submit'])) {

   //cek username dan password

   if ($\_POST['username']=='admin' &&$\_POST['password'] =='admin') {

      header('Location: admin.php');

      exit;

   } else {

   //jika salah tampilkan kesalahan

   $error = true;

   }

}

?>

* **DATABASE**

Kolom = field

Baris = record

Setiap table harus mempunyai primary key (sebuah data, yang dapat merepresentasikan satu baris record secara unik).

Relationship yaitu hubungan antara table

Pada id, bisa menbamhakn attribute auto\_increment, agar bertambah +1 setiap kali ada data yang masuk, dan juga bisa ditambah attribute primary key

Cara membuat database di cmd:

Masuk ke E:\xampp\mysql\bin>mysql -u root -p > default paaswordnya kosong

Contoh, create database {namadatabase};

Lalu, kita menggunakan database tersebut, dengan use {namdatabase}

Lalu membuat table, dengan create table {namatable} (

Id int primary key auto\_increment,

nama varchar(100),

nrp varchar(100), //untuk angka yang pasti gunakan char(angkapasti),

email varchar(100),

jurusan varchar(100),

gambar varchar(100) );

untuk melihat database dengan cara show databases;

untuk melihat table dengan cara, show tables

untuk melihat struktur tabel dengan cara, describe {namadatabase}

untuk memasukkan nilai ke database dengan cara:

untuk string&int dengan kutip 1, kita harus berurut untuk massukkan datanya

insert into {namadatabase} values (‘ ‘, blablabla );

untuk melihat data tersebut dengan cara, select \* from {namatables}

atau untuk melihat data tertentu:

contoh select nama, nrp from {namatables}

atau select \* from {namatables} where nrp = ‘412421’;

untuk mengganti seuatu di tables, dengan cara:

update mahasiswa set jurusa = ‘Teknik Industri’ where id = 2;

untuk menghapus salah satu data dengan cara:

delete from {namatalbes| where id = 3;

untuk menghapus semua tables dengan cara :

drop table {namatables};

/ untuk hapus databases dengan cara {namatables} diganti dengan {namadatabases}

Contoh drop database phpsepatu;

Pada mysql

Terdapat null> ini untuk ngasih tau apakah field tersebut boleh kosong atau tidak

Terdapat index> untuk menunjukkan ia primary key atau tidak

Terdapat AI> untuk auto\_increment

* **MENGHUBUNGKAN DATABASE KE PHP**

1. Ekstensi MySQL < ini versi lama
2. Ekstensi MySQLI (ini sangat teroptimasi)
3. PDO atau PHP Data Objek (Kalo ini bisa terhubung kebanyak database)
4. <?php
5. // koneksi ke database, paraneternya "namahost", "root", "passwprdnya(isikosoong)", "namadatabase"
6. $conn = mysqli\_connect("localhost", "root", "", "phpsepatu");
7. //ambil data dari tabel sepatu / query data sepatu, sintak sql ditulis besar
8. $result = mysqli\_query($conn, "SELECT \* FROM sepatu");
9. //untuk melihat apakah sudah terhubung apa tidak
10. if(!$result){
11. echo mysqli\_error($conn);
12. }
13. //mengambil data sepatu dari object $result, bisa dengan cara:
14. // 1. mysqli\_fetch\_row() ini mengembalikan array numerik
15. // 2. mysqli\_fetch\_assoc() mengembalikan array assosiative
16. // 3. mysqli\_fetch\_array() mengembalikan array asosiatif atau numerik tapi jadi double memorinya
17. // 4. mysqli\_fetch\_object() ini mengembalikan object, dengan cara $sepatu->warna
18. ?>
19. <!DOCTYPE html>
20. <html lang="en">
21. <head>
22. <meta charset="UTF-8">
23. <meta http-equiv="X-UA-Compatible" content="IE=edge">
24. <meta name="viewport" content="width=device-width, initial-scale=1.0">
25. <title>Halaman Admin Sepatu</title>
26. </head>
27. <body>
28. <h1>Daftar Sepatu</h1>
29. <!-- border untuk ketebalan, cellpadding untuk ruangan tablenya, cellspacing agar tidak ada putih didalam garisnya  -->
30. <table border="1"  cellpadding="10"  cellpadding="10" cellspacing="0"  >
31. <!-- tr untuk rownya -->
32. <tr>
33. <!-- td untuk collume -->
34. <th>No.</th>
35. <th>Aksi</th>
36. <th>Gambar</th>
37. <th>Merek</th>
38. <th>Warna</th>
39. <th>Ukuran</th>
40. </tr>
41. <?php
42. while ($sepatu = mysqli\_fetch\_assoc($result)) { ?>
43. <tr>
44. <td>
45. <?php echo($sepatu["id"])  ?>
46. </td>
47. <td>
48. <a href="">ubah</a> |
49. <a href="">hapus</a>
50. </td>
52. <td>
53. <img src="gambar/<?= ($sepatu["gambar"])?>" width="50"  alt="">
54. </td>
56. <td>
57. <?php echo($sepatu["merek"])  ?>
58. </td>
59. <td>
60. <?php echo($sepatu["warna"])  ?>
61. </td>
63. <td>
64. <?php echo($sepatu["ukuran"])  ?>
65. </td>
66. </tr>
67. <?php  } ?>
68. </table>
69. </body>
70. </html>

* **CRUD**
* *function* query($query){
* //untuk mengacu pada $conn yang ada di atas
* global $conn;
* //$result akan mengambil data dari koneksi database ($connn), dan isi dari database tersebut ($query)
* $result = mysqli\_query($conn, $query);
* //$rows akan menampung nilai dari $result
* $rows = [];
* while ($row = mysqli\_fetch\_assoc($result)) {
* $rows[] = $row;
* }
* return $rows;
* }

*function* tambah($data)

{

global $conn;

//htmlspecialchars untuk menghalangi input yang tidak sesuai atau memasukkan char yang aneh

$merek = htmlspecialchars($data['merek']);

$warna = htmlspecialchars($data['warna']);

$gambar = htmlspecialchars($data['gambar']);

$ukuran = htmlspecialchars($data['ukuran']);

$query = "INSERT INTO sepatu

            VALUES

            ('', '$merek', '$warna', '$ukuran', '$gambar')

               ";

            mysqli\_query($conn, $query);

return mysqli\_affected\_rows($conn);

}

*function* hapus($id){

   global $conn;

   mysqli\_query($conn, "DELETE FROM sepatu WHERE id = $id");

}

*function* hapus($id){

   global $conn;

   //untuk menghapus rows, dimana mengacu pada idnya

   mysqli\_query($conn, "DELETE FROM sepatu WHERE id = $id");

}

* **MEMBUAT PENCARIAN**
* *function* cari($keyword)
* {
* // nilai $keyword yang diambil dari cari($\_POST["keyword"]) dimasukkan ke nama
* // lalu nilai tersebut dimasukkan ke $query
* //LIKE digunakan untuk pencariannya tidak terlalu spesifik, dimana %menenjukkan huruf depan, dan kalo %belakang mencari huruf belakang
* $query = "SELECT \* FROM sepatu WHERE
* merek LIKE '%$keyword%' OR
* warna LIKE '%$keyword%' OR
* ukuran LIKE '%$keyword%'
* ";
* // function query akan menampung nilai dari $query, yang akan dikembalikan
* return query($query);
* }

**FILE INDEX**

// ketika tombol cari dipencet, masukkan nilai dari keyword yang ada di $\_post ke function cari dan disimpan di $sepatu

if ( isset($\_POST['cari']) ) {

   $sepatu = cari($\_POST['keyword']);

}

?>

<form action="" method="POST">

   <!-- autofocus untuk mengarahkan pertama kali langsung ke field tersebut -->

   <!-- placeholder untuk menampilkan teks pada field pencarian -->

   <!-- autocomplete = off untuk menghilangkan suggest pada field pencarian -->

   <input type="text" name="keyword" size="40" autofocus placeholder="Masukkan Pencarian.." autocomplete="off">

   <button type="submit" name="cari">Cari</button>

</form>

* **FILE HANDLING**

1. <input type=’file’..
2. Enctype
3. $\_FILES
4. Move\_uploaded\_file

Gambar bisa kita masukkan ke database langsung, karena di dalam mysql terdapat tipe data yang namanya blob (binary large object), dimana kita dapat menyimpan gambar dalam bentuk biner>>> kekurangnnya database jadi besar,berat.

Isi dari **$\_files** ketika kita vardump, dan gambar telah diupload menggunakan **<input type=”file”** dengan **<form encytype=”multipart/form-data”**

Contoh:

if (isset($\_POST['submit'])) {

var\_dump($\_FILES);

die;

 }

?>

   <form action=""  method='POST' enctype="multipart/form-data">

   <ul>

      <li>

         <label for="gambar">Gambar : </label>

         <br>

         <input type="file" name="gambar" id="gambar">

      </li>

      <li>

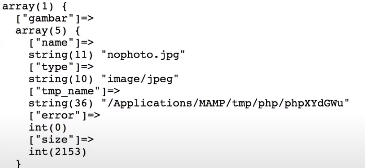
         <button type="submit" name="submit" >Tambah Data</button>

      </li>

   </ul>

   </form>

Hasil vardump:



Isinya adalah array assosiatif tapi 2 dimensi,

Dimana di dimensi 1, terdapat 1 array yaitu name = “gambar”,

Dan dimensi ke2, terdapat 5 buah elemen array

1. Name, yaitu nama file
2. Tipe, tipe file
3. Tmp name, tempat penyimpanan sementara, tapi begitu di upload akan dipindahkan ke tempat kita mau
4. Error, menghasilkan angka, 0 artinya tidak ada error, contoh 4 yaitu ketika tidak ada yang diupload
5. Size, ini ukuran imgnya

* **LOGIN**
* $result = mysqli\_query($conn, "SELECT \* FROM users WHERE
* username = '$username'");
* mysqli\_num\_rows($result)

1. **$result**, untuk mengambil data username yang sama dari database dengan yang dikirimkan lewat method post .
2. **Mysqli\_num\_rows($result)**, untuk mengetahui jumlah baris, yang nilai usernamenya sama dengan nilai $username yang diinput, dalam kasus ini jika ada, maka bernilai 1, karena hanya ada 1 username yang sama dengan username inputan.

* **SESSION**

Ini adalah mekanisme penyimpanan infromasi ke dalam variable agar bisa digunakan di lebih dari satu halaman.

Informasi di dalam session akan diisimpan dalam server.

Untuk $\_POST dan $\_GET, itu hanya bisa dipakai 2 halaman, sedangkan session lebih dari 1 halaman.

**Session** mirip dengan **cookie**, bedanya

**Session** disimpan di server sedangkan **cookie** di client (didalam browsernya).

Untuk menjalankan **$\_SESSION** kita perlu menjalankan **session\_start()** di awal scriptnya.

**Session\_destory()** digunakan untuk menghilangkan session

Ketika kita menggunakan session, harus memuat halaman yang terdapat nilai sessionnya terlebih dahulu, untuk inisialisasi session tersebut, jika tidak, tidak akan muncul halamannya.

* **COOKIE**

Client bisa memanipulasi cookie, yaitu bisa mengubah, mengedit, menambah cookie baru.

Ini bisa digunakan untuk mengenali user, shopping cart berfungsi untuk keranjang ketika ingin berganti halaman, personalisasi> untuk mencari behavior iklan.

Untuk membuat cookie, harus menambah **setcookie()**.

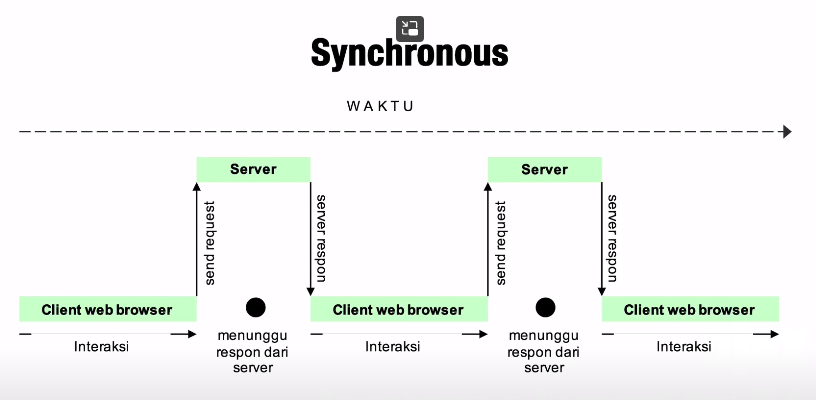
**Time() digunakan untuk mengset berapa lama cookie akan bertahan.**

**Setcookie(“namaparameter”, “isiparameter”, time()+60);**

**Hash(‘methodehasnya’, ‘yang mau dihashnya’)>>**ini untuk memberikan hash

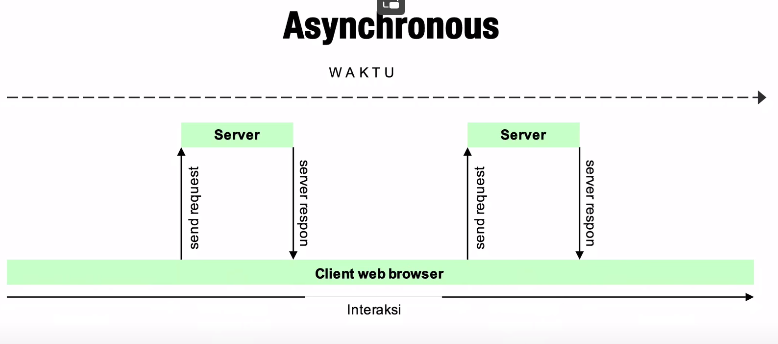
* **PAGINATION**
* $jumlahDataPerhalaman = 2;
* //ini akan menampilkan jumlah baris yang ada di dalam tabel sepatu
* $jumlahData = count(query("SELECT \* FROM sepatu"));
* //ini akan menentukan jumlahHalaman yang mengacu pada jumlahdata/jumlahdataPerhalaman
* //ceil akan membulatkan kebawah hasil pembagian tersebut
* $jumlahHalaman = ceil(($jumlahData/$jumlahDataPerhalaman));
* //[ //cek, apakah diurl pada halaman, jika ada, maka
* // if (isset($\_GET["halaman"])) {
* //    //ambil halaman diurl, dimasukkan ke $halamanaktif
* //    $halamanAktif = $\_GET["halaman"];
* // } else {
* //    //jika tidak ada, maka halaman aktif di 1
* //    $halamanAktif = 1;
* // } ]
* //diatas menjadi dibawah, kita pakai operator ternary
* //jika terdapat value halaman di url, maka masukkan ke var halamanaktif
* //jika tidak ada, masukkan value 1, di halaman aktif
* $halamanAktif = (isset($\_GET["halaman"]) ? $\_GET["halaman"] : 1);
* //ini rumus, untuk mengetahui index ketika jumlah data perhalaaman ada dua
* $awalData = (($halamanAktif-1)\*$jumlahDataPerhalaman);
* <!-- jika halamanaktifnya lebih dari satu atau tidak 0, maka munculkan tanda < -->
* <?php if ($halamanAktif>1) { ?>
* <!-- ini akan mengurangi jumlah halamannya, setiap diklik -->
* <a href="?halaman=
* <?=$halamanAktif-1?>">&lt;
* <?php } ?>
* </a>
* <!-- ini akan looping sesuai dengan jumlahhalaman yang mengacu pada jumlah data yang ditampilkan -->
* <?php  for ($i=1; $i <=$jumlahHalaman ; $i++) {
* //jika nilai $i sama dnegna halaman aktif maka akan ada style pada link, sisanya akan tidak ada style
* if ($i==$halamanAktif) { ?>
* <a  style = "font-weight-bold; color:red"
* href="?halaman= <?= $i; ?>">  <?= $i?>  </a>
* <?php }
* else{ ?>
* <a href="?halaman= <?= $i; ?>">  <?= $i?></a>
* <?php }
* }?>
* <!-- jika halamanaktifnya kurang jumlahhalaman maka hilangkan tanda > -->
* <?php if ($halamanAktif<$jumlahHalaman) { ?>
* <!-- ini akan menambahkan jumlah halamannya, setiap diklik -->
* <a href="?halaman=
* <?=$halamanAktif+1?>">&gt;
* <?php } ?>
* </a>
* **LIVE SEARCH (AJAX)**

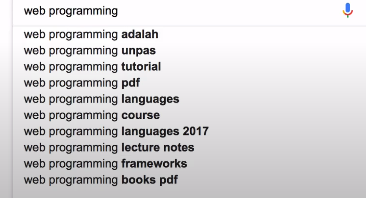
Proses ini memakan wak tu lama, karena proses synchronous akan mengembalikan seluruh konten



Untuk asynchronous lebih efisien, karena hanya mengembalikan nilai yang kita butuhkan saja

Untuk contohnya adalah chattingan di fb





Diatas adalah contoh live search

* **COMPOSER**

Mengeola dependency atau ketergantungan antara file2 yang berkaitan .

Terdapat dependency manager lain, yaitu npm, bower, yarn, bundler dll

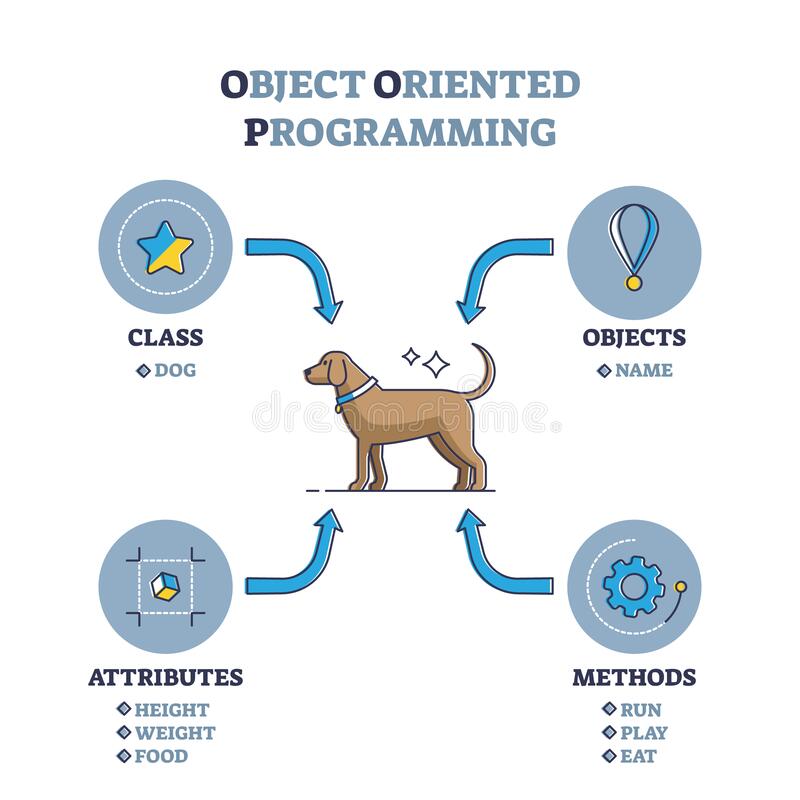
**OOP DASAR PHP**

**Procedural Programming**

1. Instruksi dilakukan langkah demi langkah
2. Memecah program menjadi bagian-bagian kecil > bisa disebut prosedut, subroutine atau function
3. Linear / top to bottom

**Keuntungan dari procedural**

1. To The point, langsung melakukan apa yang kita mau, tanpa harus mendefiniskan sesuatu
2. Simplicity dan kemudahan implementasi untuk compiler dan interpreter
3. Mudah ditelusuri
4. Membutuhkan sedikit memory



**OOP**

1. Menyusun semua kode program dan struktur data sebagai objek
2. Objek adalah unit dasar dari program
3. Objek menyimpan data dan perilaku

**Keuntungan OOP**

1. Representasi dunia nyata
2. Enkapsulasi (aman, karena tidak bisa diakses sembarangan)
3. Abstrak (menyembunyikan kerumitan pemrograman)
4. Reusability (mudah di gunakan ulang)
5. Memudahkan ketika program menjadi berkembang
6. Kemudahan pengelolaan
7. Kolaborasi
8. Digunakan oleh framework

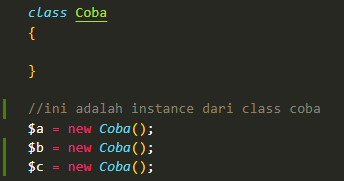
* **CLASS & OBJECT**

**Object**

* Instance yang didefinisakn oleh Class
* Bisa membuat banyak object dengan 1 class
* Object dibuat pakai keyword **new**

**Class**

* Merupakan blueprint/template untuk membuat instance dari object
* Class mendefiniskan Object
* Menympan data dan perilaku yang disebut property dan method



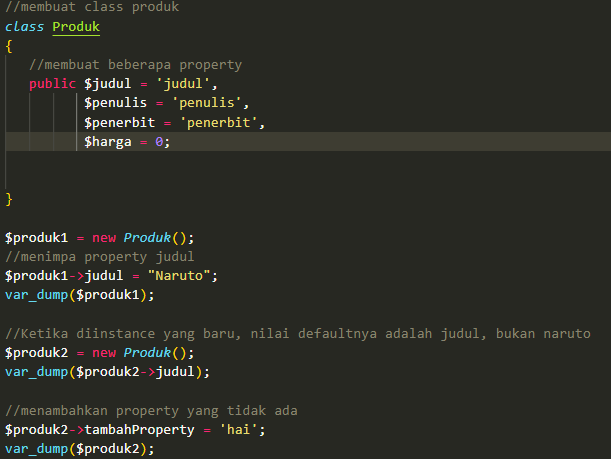
* **PROPERTY & METHOD**

**METHOD**

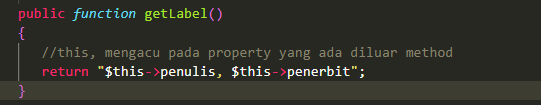
* Merepresentasikan perilaku dari sebuah object
* Function yang ada di dalam object

**PROPERTY**

* Merepresentasikan data/keadaan dari sebuah object
* Variabel yang ada di dalam object (member variable)



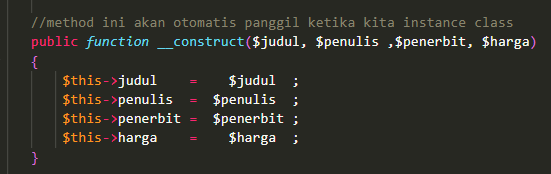
Function



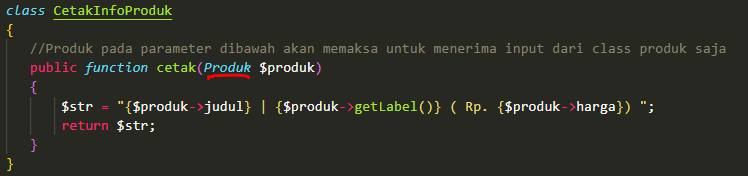
* **CONSTRUCTER METHOD**

**CONSTRUCTOR**

* Method khusus yang ada didalam class, karena otomatis ada



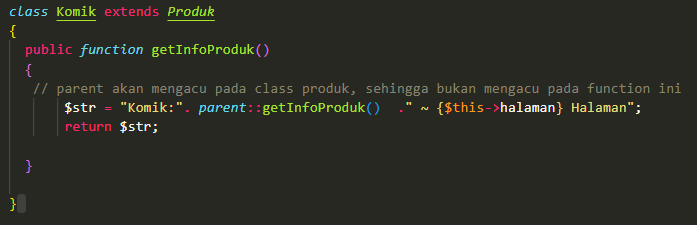
* **OBJECT TYPE**



* **INHERITANCE**

**INHERITANCE**

* Menciptakan hierarki antar kelas (**parent & child**)
* **Child Class,** ,mewarisi semua properti dan method dari **parent**nya (yang visilble)
* **Child Class,** memperluas (extends) fungsionalitas dari **parent**nya



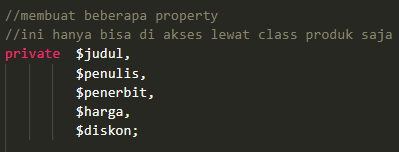
* **VISIBILITY**

**VISIBIILITY**

* Konsep untuk mengatur akses dari **property** dan **method** pada sebuah objek
* 3 keyword visibility yaitu **public, protected,** dan, **private**
* **Public,** dapat digunakan dimana saja, bahkan diluar kelas
* **Protected,** digunakan hanya dalam sebuah kelas serta turunannya
* **Private,** hanya dapat disebuah kelas tertentu saja

**MANFAAT**

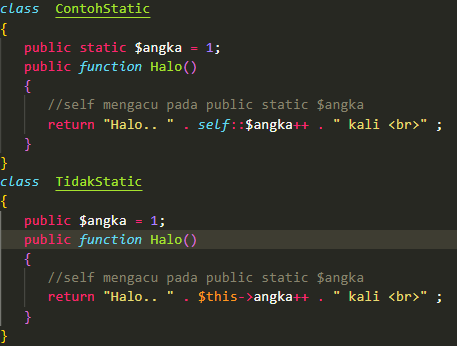
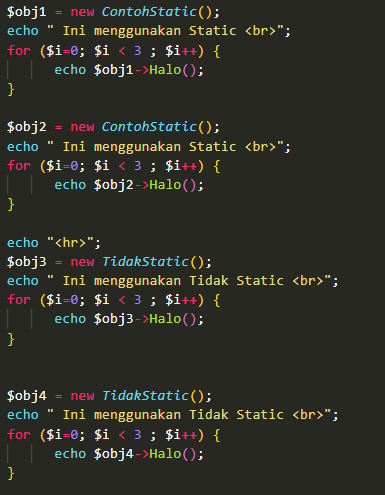
* Hanya memperlihatkan aspek dari class yang dibutuhkan oleh client.
* Menentukan kebutuhan yang jelas untuk object
* Menghindari bug



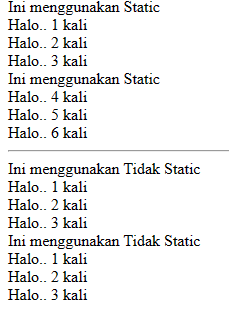
* **STATIC KEYWORD**

**MANFAAT**

* Kita bisa mengakses **property** dan **method** dalam konteks **class**
* Kita tidak perlu membuat **instance** lagi untuk memanggil **property / method**
* Nillainya akan tidak berubah walaupun di instance lagi
* Membuat kode menjadi procedural
* Digunakan untuk membuat fungsi bantuan / helper / diclas2 utility framework

Hasil

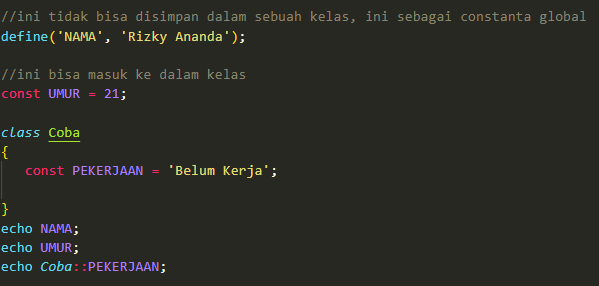


* **CONSTANT**

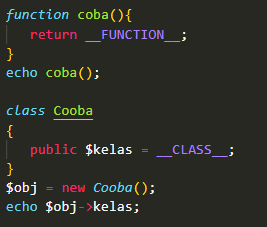
**MANFAAT**

* Sebuah **identifier** untuk menyimpan **nilai,** dimana nilainya tidak dapat berubah
* Bisa menggunakan **define() / const**
* Menggunakan huruf besar semua untuk namanya
* Terdapat magic constant (constant yang sudah ada di dalam php), yaitu

1. **\_\_FILE\_\_ >** menampilkan baris yang ditulis di code
2. **\_\_LINE\_\_** > alamat dari file tersebut
3. **\_\_DIR \_\_** > direktori dari file tersebut
4. **\_\_FUNCTION \_\_** > menentukan ada di function mana
5. **\_\_CLASS \_\_**> menentukan ada di class mana
6. **\_\_TRAIT \_\_** > alamat dari file tersebut
7. **\_\_METHOD \_\_** > alamat dari file tersebut
8. **\_\_NAMESPACE \_\_** > alamat dari file tersebut



PENGGUNAAN MAGIC CONSTANT



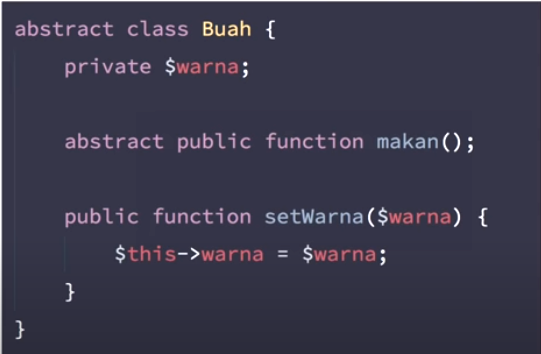
* **ABSTRACT CLASS**

**Apa Itu?**

* Sebuah class yang **tidak dapat di instansiasi** , kita bisa instansiasi kelas turunannya
* Mendefinisikan interface untuk kelas lain yang menjadi turunannya
* Berperan sebagai kerangka dasar untuk kelas turunannya
* Memiliki minimal 1 **method abstrak**
* Digunakan dalam inheritance untuk **memaksakan** implementasi abstrak yang sama untuk semua kelas turunannya.
* Semua kelas turunan, harus mengimplementasikan methiod abstrak yang ada dikelas abstraknya
* Kelas abstrak boleh punya property/method regular atau property/static method

**MANFAAT**

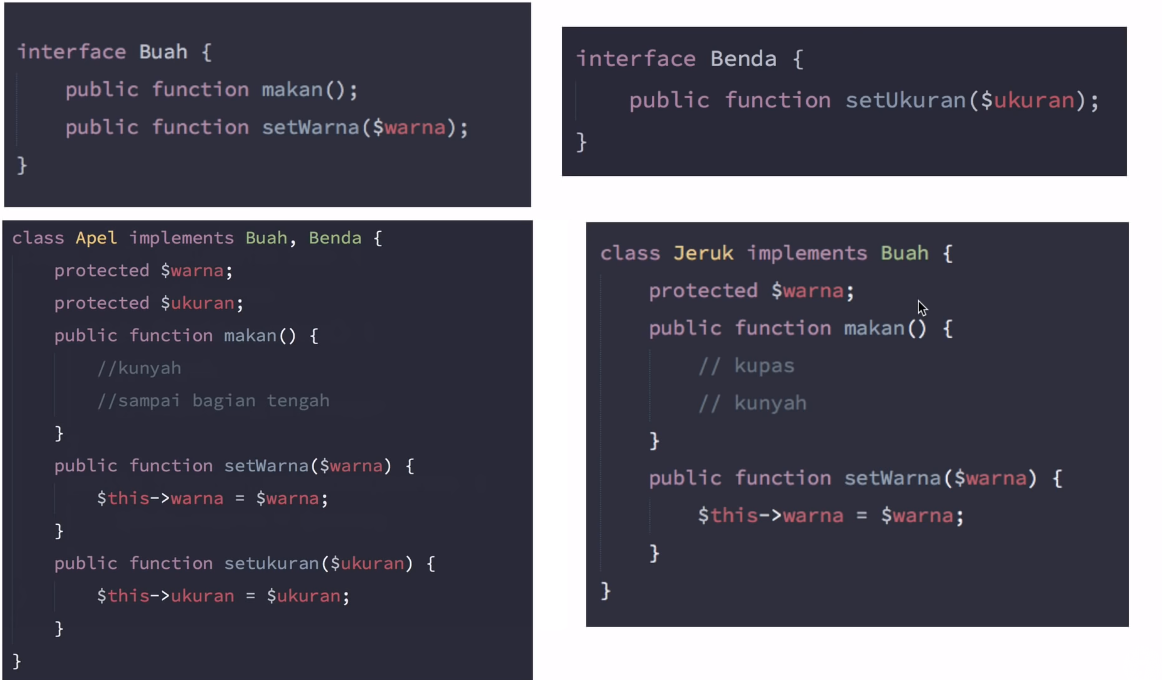
* Merepresentasikan ide atau konsep dasar
* Composition over inheritance
* Salah satu cara menerapkan konsep polymorphism
* Sentralisasi logic
* Mempermudah pengerjaan tim



* **INTERFACE**

**Apa Itu?**

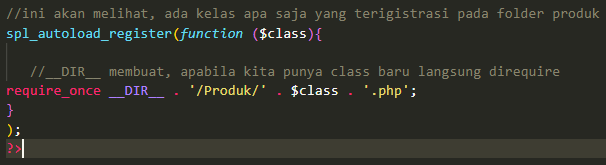
* Kelas abstrak yang sama sekali tidak punya implementasi
* Murni merupakan template untuk kelas turunannya
* Tidak boleh punya property, **hanya deklarasi method saja**
* Semua method harus dideklarasikan dengan visibility **public**
* Boleh mendeklrasaikan \_\_construct()
* Turunannya boleh mengimplementasikan banyak interface
* Dengan menggunakan type-hinting (sebuah object bisa di jadikan parameter) dapat melakukan **dependency injection**
* Mencapai polymorphism



* **AUTOLOADING**

**Apa Itu?**

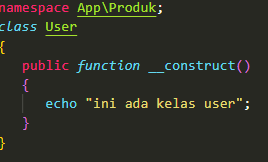
* Memangil class (file) tanpa harus menggunakan **require**
* Dengan menggunakan **spl\_autoload\_register()**



* **NAMESPACE / ENCAPSULATION**

**ALASAN MENGGUNAKAN INI**

* PHP tidak mengizinkan kita untuk punya 2 class dengan nama yang sama
* Untuk membedakan class kita dengan class milik orang lain atau framework
* **Namepace** diiisi dengan direktori folder kita
* Untuk Aturannya seperiti : **namespace** Vendor(Pembuat/nama apk)/Namespace/SubNamespace

 <**cara panggil**

