|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran PHP Video 23 - 24  ( Tugas Minggu ke – 11 ) | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Objek oriented programming ( OOP ) yaitu membuat kumpulan variabel dan function kemudian diletakkan di dalam satu tempat dan bisa digunakan berkali - kali sesuai kebutuhan.
2. Class adalah wadah dari variabel dan function.
3. Setiap variabel di dalam class harus diberi awalan, yaitu bisa menggunakan awalan privat, awalan publik, awalan protected, dan awalan default.
4. Variabel yang terdapat di dalam class disebut dengan properti.
5. Function yang terdapat di dalam class disebut dengan method.
6. Awalan private berarti properti ( variabel ) dan method ( properti ) hanya dapat diakses di dalam lingkup class dimana properti dan method didefinisikan.
7. Penulisan private yaitu dengan menambahkan tulisan private di depan properti ( variabel ) dan method ( properti ).

* Cth : private $host;

Artinya, pada class DB terdapat $host yang mempunyai visibilty private.

1. Awalan public berarti properti ( variabel ) dan method ( properti ) dapat diakses melalui class itu sendiri atau menggunakan object.
2. Penulisan awalan public yaitu : public function nama\_function( Type $var = null )

* Cth : public function selectData()

{

echo 'select data';

}

Artinya, pada class DB mempunyai function selectData yang visibility nya adalah public.

1. Untuk menggunakan properti dan method di dalam class harus membuat objek terlebih dulu.
2. Cara pembuatan objek adalah $nama\_objek = new nama\_class;

* Cth : $db = new DB;

Artinya, sudah dibuat nama objek dengan menggunakan class DB

1. Untuk melihat nilai dari objek dapat menggunakan var\_dump.

* Cth : var\_dump($db);

1. Nilai yang akan keluar dari objek yang dipanggil adalah: object(nama\_objek) nilai objek { [ "properti":"nama\_objek":tipe visibility objek]=>tipedata(banyaknya data) "isi properti" }
2. Untuk menampilkan function di dalam class yaitu dengan memanggil properti yang disambungkan dengan nama class, lalu dipanggil nama functionnya.

* Cth : $db->selectData();

Artinya, menampilkan isi dari selectData pada $db.

1. Jika properti mempunyai visibility private, maka akan terjadi error apabila dioutput pada program.
2. Penulisan dan output pemanggilan properti ( variabel ) dengan menggunakan echo dan memanggil nama dari variabel yang sudah disambungkan dengan class.

* Cth : echo $db->host;

Artinya, $db sudah disambungkan pada class DB, maka jika ingin menampilkan $host yang ada di dalam class hanya menggunakan nama variabel nya saja, tidak perlu menggunakan tanda $.

1. Apabila ingin memanggil properti yang mempunyai visibilty privat, maka harus membuat function di dalam class yang isinya properti yang akan dipanggil.

* Cth : public function getDatabase()

{

echo $this->database;

}

Artinya, terdapat function yang bernama getDatabase, yang isinya memanggil $database.

1. $this digunakan untuk menunjukkan bahwa properti / method yang dipanggil berada di luar dari fungsi tersebut.
2. Untuk memanggil fuction yang berada di dalam function bisa menggunakan $this->nama\_function(); , dan untuk penampilannya menggunakan $nama\_objek->nama\_function();

* Cth : public function tampil()

{

$this->selectData();

}

Artinya, terdapat function bernama tampil yang isinya adalah function selectData, dipanggil menggunakan $this karena function selectData terletak di luar function tampil.

1. function\_\_construct adalah function yang dapat berjalan otomatis ketika sebuah objek dibuat.
2. static function adalah function yang disambungkan pada class, dan untuk pemanggilan static function tidak perlu menggunakan objek melainkan dapat langsung dipanggil classnya.
3. Pemanggilan dari static function adalah dengan menuliskan nama\_class :: nama\_function()

* Cth : DB::insertData();

1. Untuk menyambungkan properti dan method maka dapat digabungkan menggunakan

$this->nama\_properti = $this->nama\_function;

* Cth : public function \_\_construct()

{

$this->koneksi = $this->koneksiDB();

}

Artinya, isi di dalam function tersebut akan langsung berjalan otomatis ketika objek dibuat karena menggunakan function \_\_construct, dan di dalam function tersebut terdapat penyambungan dari $this->koneksi yang isi nya diambil dari $this->koneksiDB, jadi isi dari $koneksiDB akan masuk ke dalam $koneksi, dan objek akan langsung berjalan.

1. mysqli\_query adalah fungsi yang digunakan untuk mengirim perintah sql ke server mysql untuk melakukan aktivitas :
   * menambahkan data dengan perintah SQL INSERT.
   * membaca data dengan perintah SQL SELECT.
   * mengubah data dengan perintah SQL UPDATE
   * menghapus data dengan perintah SQL DELETE.
2. mysqli\_fetch\_assoc adalah fungsi php yang menghasilkan array assosiatif dimana indexnya sesuai dengan nama kolom yang diseleksi.
3. Return adalah perintah khusus PHP pada fungsi untuk menginstruksikan PHP bahwa pemrosesan fungsi selesai.
4. While berfungsi untuk membuat perulangan yang tidak perlu diketahui berapa kali perulangannya.
5. Untuk menampilkan semua data, maka pertama harus membuat function untuk memanggil data.

* Cth : public function getALL($sql)

{

$result = mysqli\_query($this->koneksi,$sql);

while($row = mysqli\_fetch\_assoc($result)){

$data[]=$row;

}

return $data;

}

Artinya, terdapat function bernama getALL yang disambungkan oleh $result melalui mysqli\_query menggunakan $this dengan koneksi dan $sql. Lalu untuk menampilkan isi data menggunakan while karena perulangan yang tidak perlu diketahui jumlahnya dan disimpan ke dalam $row, di dalam $row terdapat mysqli\_fetch\_assoc untuk menghasilkan array assosiatif, array assosiatif disimpan di dalam $data[] yang isinya sama dengan $row. Untuk menginstruksi bahwa function telah selesai maka menggunakan return.

1. Untuk pemanggilan dari function getALL maka $row diisi dengan perintah SELECT lalu ditampilkan dengan menggunakan foreach.

* Cth : $row = $db->getALL("SELECT \* FROM tblkategori");

foreach ($row as $key) {

echo $key['kategori'];

echo "<br>";

}

Artinya, $row di dalam objek $db terdapat perintah SELECT untuk memilih data yang ada di dalam tblkategori, lalu ditampilkan menggunakan foreach, yang $row berguna sebagai $key, bagian kolom yang ditampilkan adalah kolom kategori.

1. Untuk menampilkan satu item, maka dibuat function getITEM yang isinya mengatur tampilan pemanggilan item.

* Cth : public function getITEM($sql)

{

$result = mysqli\_query($this->koneksi,$sql);

$row = mysqli\_fetch\_assoc($result);

return $row;

}

Artinya, terdapat function getITEM yang disambungkan oleh $result melalui mysqli\_query menggunakan $this dengan koneksi dan $sql. Lalu untuk menampilkan isi data menggunakan $row, di dalam $row terdapat mysqli\_fetch\_assoc untuk menghasilkan array assosiatif, array assosiatif. Untuk menginstruksi bahwa function telah selesai maka menggunakan return.

1. 28. Untuk pemanggilan data dari function getITEM, $row diisi dengan SELECT untuk memilih data dan ditampilkan menggunakan echo.

* Cth : $row = $db->getITEM("SELECT \* FROM tblkategori WHERE idkategori = 12");

echo $row['kategori'];

Artinya, $row di dalam objek $db terdapat perintah SELECT untuk memilih data dari tblkategori yang ada pada idkategori urutan 12. Lalu ditampilkan menggunakan echo $row['kategori'];

1. Pembuatan function rowCOUNT yang digunakan untuk menghitung berapa baris yang ada di dalam data.

* Cth : public function rowCOUNT($sql)

{

$result = mysqli\_query($this->koneksi,$sql);

$count = mysqli\_num\_rows($result);

return $count;

}

Artinya, terdapat function rowCOUNT yang disambungkan oleh $result melalui mysqli\_query menggunakan $this dengan koneksi dan $sql. Variabel $count berisi mysqli\_num\_rows berguna untuk mengetahui jumlah baris pada data. Lalu proses pada function selesai dan diakhiri dengan return

1. mysqli\_num\_rows berfungsi untuk mengetahui total jumlah baris pada tabel database dan memampilkannya menggunakan angka.
2. Untuk menampilkan rowCOUNT maka dapat menggunakan SELECT di dalam $countt untuk memilih data.

* Cth : $count = $db->rowCOUNT("SELECT \* FROM tblkategori");

Artinya, $count di dalam objek $db terdapat perintah SELECT untuk memilih data di dalam tblkategori.

1. Membuat function runSQL untuk menambahkan data.

* Cth : public function runSQL($sql)

{

$result = mysqli\_query($this->koneksi,$sql);

}

Artinya, terdapat function runSQL yang disambungkan oleh $result melalui mysqli\_query menggunakan $this dengan koneksi dan $sql.

1. Untuk menambahkan data, menggunakan INSERT INTO.

* Cth : $db->runSQL("INSERT INTO tblkategori VALUES ('','Es Cincau Bandung')");

Artinya, di dalam objek $db, terdapat runSQL yang berisi INSERT INTO untuk menambahkan data di dalam tblkategori dengan VALUE Es Cincau Bandung, karena tipe data adalah varchar, maka diberi petik atas sebanyak 2 kali, yang satu berisi kosong dan yang satu berisi value yang akan ditambahkan.

1. Menggunakan fungsi delete data untuk menghapus data.

* Cth : $db->runSQL("DELETE FROM tblkategori WHERE idkategori = 20");

Artinya, di dalam objek $db terdapat runSQL yang berisi perintah untuk DELETE FROM pada tblkategori yang terdapat di idkategori urutan 20.

1. Membuat function pesan untuk menambahkan tulisan.

* Cth : public function pesan($text="")

{

echo $text;

}

Artinya, di dalam function pesan terdapat $text="" lalu dipanggil dengan echo $text, $text dikosongi karena apabila kita lupa untuk mengisi agar tidak terjadi error.

1. Untuk menampilkan function pesan maka menggunakan $db->pesan("Anda berhasil :)")

Artinya, di dalam objek $db terdapat pesan yang berisi tulisan Anda berhasil :).

**Saya Belum Mengerti**