

LAPORAN WORKSHOP SISTEM INFORMASI BERBASIS WEB

KONSEP LANJUT OOP



Dosen Pengajar :

Dony Bahtera Firmawan, S.Kom., M.Kom

Oleh :

Rafika Dwi Shefira E41220791

Golongan B

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2023

Acara 8

Materi Pembelajaran	: Konsep Dasar OOP
Acara Praktikum/Praktik	: 4/2
Tempat	: Daring/Luring Politeknik Negeri Jember
Alokasi Waktu	: 2 x 100 menit

A. Prosedur Kerja

Kalkulator sederhana menggunakan constructor.

```
1  <?php
   1 reference | 0 implementations
2  class hitung {
   9 references
3  |     public $angka1;
   9 references
4  |     public $angka2;
5  |
6  |     1 reference | 0 overrides
   function __construct($angka1, $angka2) {
7  |         $this->angka1 = $angka1;
8  |         $this->angka2 = $angka2;
9  |     }
10 |
11 |     1 reference | 0 overrides
   function tambah() {
12 |         $tambah = $this->angka1 + $this->angka2;
13 |         echo "Hasil dari ". $this->angka1." + " . $this->angka2." = ". $tambah."\\n";
14 |     }
15 |
16 |     1 reference | 0 overrides
   function kurang() {
17 |         $kurang = $this->angka1 - $this->angka2;
18 |         echo "Hasil dari ". $this->angka1." - " . $this->angka2." = ". $kurang."\\n";
19 |     }
20 |
21 |     1 reference | 0 overrides
   function pembagian() {
22 |         $pembagian = $this->angka1 / $this->angka2;
23 |         echo "Hasil dari ". $this->angka1." / " . $this->angka2." = ". $pembagian."\\n";
24 |     }
}
```

```

26 | 1 reference | 0 overrides
27 | function perkalian() {
28 |     $perkalian = $this->angka1 * $this->angka2;
29 |     echo "Hasil dari ". $this->angka1." * " . $this->angka2." = ". $perkalian."\n";
30 | }
31 |
32 | $hitung = New hitung(25, 5);
33 | $hitung->tambah();
34 | $hitung->kurang();
35 | $hitung->pembagian();
36 | $hitung->perkalian();
37 | ?>

```

1. Terdapat class person yang didalamnya memiliki properti dengan nama \$angka1 dan \$angka2. Properti ini juga bisa disebut dengan attribute.

```

2 | class hitung {
3 |     9 references
4 |     public $angka1;
5 |     9 references
6 |     public $angka2;

```

2. Membuat sebuah constructor, merupakan method khusus yang dijalankan secara otomatis saat sebuah objek di instansiasi. Instansiasi objek dilakukan dengan perintah “**new**”. Terdapat method __construct(\$angka1, \$angka2) yang merupakan constructor dari class hitung, dengan 2 parameter \$angka1 dan \$angka2. Method ini akan dipanggil secara otomatis ketika class hitung di instansiasi.

```

6 | function __construct($angka1, $angka2) {
7 |     $this->angka1 = $angka1;
8 |     $this->angka2 = $angka2;
9 | }

```

3. Membuat method untuk aritmatika yaitu, tambah, kurang, pembagian dan perkalian. Method ini digunakan sebagai perhitungan aritmatikanya. Terdapat pada baris ke 11 sampai baris 30.
4. Membuat objek dari class hitung yang dinamakan instansiasi. Memasukkan 2 parameter yang telah di set pada method __construct. Yaitu parameter \$angka 1 dengan nilai 25 dan parameter \$angka2 dengan nilai 5.

```

31 | $hitung = New hitung(25, 5);

```

5. Selanjutnya mengakses nilai pada method aritmatikanya.

```
32 $hitung->tambah();  
33 $hitung->kurang();  
34 $hitung->pembagian();  
35 $hitung->perkalian();
```

Berikut adalah hasil ketika code program di run.

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
	Hasil dari 25 + 5 = 30			
	Hasil dari 25 - 5 = 20			
	Hasil dari 25 / 5 = 5			
	Hasil dari 25 * 5 = 125			

Kalkulator sederhana menggunakan getter dan setter.

```
2  class Perhitungan {
    4 references
3  |  private $bil1;
    4 references
4  |  private $bil2;
5  |
    2 references | 0 overrides
6  |  public function setBil1($bil) {
7  |  |  $this->bil1 = $bil;
8  |  }
    2 references | 0 overrides
9  |  public function getBil1() {
10 |  |  return $this->bil1;
11 |  }
12 |
    2 references | 0 overrides
13 |  public function setBil2($bil) {
14 |  |  $this->bil2 = $bil;
15 |  }
16 |
    2 references | 0 overrides
17 |  public function getBil2() {
18 |  |  return $this->bil2;
19 |  }
20 |
    1 reference | 0 overrides
21 |  public function kurang() {
22 |  |  return $this->bil1 - $this->bil2;
23 |  }
    1 reference | 0 overrides
24 |  public function tambah() {
25 |  |  return $this->bil1 + $this->bil2;
26 |  }
27 }
```

```
29 $perhitungan = new Perhitungan();
30 $perhitungan->setBil1(10);
31 $perhitungan->setBil2(5);
32
33 $perhitungan1 = new Perhitungan();
34 $perhitungan1->setBil1(10);
35 $perhitungan1->setBil2(20);
36
37 echo "Hasil dari ".$perhitungan->getBil1(). " - ".$perhitungan->getBil2(). " = ".$perhitungan->kurang()."\n";
38 echo "Hasil dari ".$perhitungan1->getBil1(). " + ".$perhitungan1->getBil2(). " = ".$perhitungan1->tambah()."\n";
```

1. Membuat class Perhitungan() yang berisi beberapa property dengan nama \$bil1 dan \$bil2.

```
2  class Perhitungan {
    4 references
3  |  private $bil1;
    4 references
4  |  private $bil2;
```

2. Terdapat method getter dan setter. Pada program dibawah terdapat method setter dengan nama setBil1. Berfungsi untuk mengatur nilai yang terdapat pada attribute atau property \$bil.

```
6      public function setBil1($bil) {
7      |      $this->bil1 = $bil;
8      |      }
```

Begitupun untuk method setter lainnya yaitu setBil2, setKurang dan setTambah yang berfungsi untuk memberi nilai pada atributenya.

3. Membuat method getter yang bernama getBil1 yang berfungsi untuk mendapatkan value atau nilai pada attribute setter yaitu \$bil.

```
9      public function getBil1() {
10     |      return $this->bil1;
11     |      }
```

Sama dengan method getter lainnya seperti getBil2, getKurang dan getTambah yang berfungsi untuk mendapat nilai dari atributenya.

4. Melakukan instansiasi object, dengan membuat sebuah object perhitungan dan perhitungan1.

```
29     $perhitungan = new Perhitungan();
33     $perhitungan1 = new Perhitungan();
```

5. Masukkan sebuah value atau nilai didalam method setter.

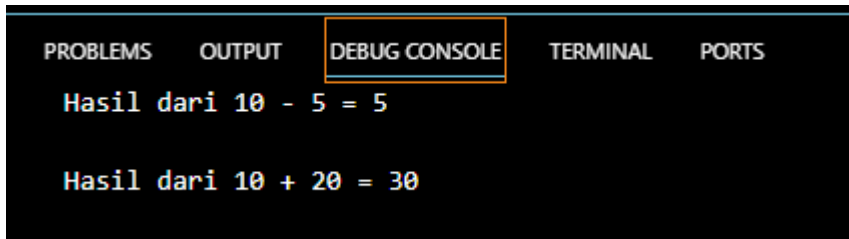
```
30     $perhitungan->setBil1(10);
31     $perhitungan->setBil2(5);

34     $perhitungan1->setBil1(10);
35     $perhitungan1->setBil2(20);
```

6. Menampilkan hasil dari aritmatika dengan nilai yang ada pada attribute.

```
37     echo "Hasil dari ".$perhitungan->getBil1(). " - ".$perhitungan->getBil2(). " = ".$perhitungan->kurang()."\n";
38     echo "Hasil dari ".$perhitungan1->getBil1(). " + ".$perhitungan1->getBil2(). " = ".$perhitungan1->tambah()."\n";
```

Berikut adalah hasil ketika code program di run.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Hasil dari 10 - 5 = 5
Hasil dari 10 + 20 = 30
```