

**10111026 – KARISMA AINUN NISSA**

**SI 2B**

JOB SHEET   
PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN 6

Dosen Pengampu

Sari Azhariyah, M.Pd.T.  
Usep Abdul Rosid, S.T., M.Kom

|  |  |
| --- | --- |
| POLITEKNIK NEGERI SUBANG |  |
| Jurusan : TIK | Waktu : 240 Menit |
| Program Studi : Sistem Informasi | Topik : Construct, Destruct dan PHP CLI |
| Mata Kuliah : PBO | Judul : Construct, Destruct dan PHP CLI |
| Tanggal : 24 September 2025 | Pertemuan ke : 6 |

# Tujuan : Mahasiswa mampu memahami dan menjelaskan konsep dasar dan mengimplementasi serta berlatih penggunaan elemen pada HTML.

# Alat dan Bahan : XAMPP, Web Browser, Text Editor, CLI

# Link Github : https://github.com/karisss26/karis\_2B\_PBO.git

**Langkah Kerja :**

1. **Siapkan dan jalankan xampp, web browser, dan text editor**
2. **Buat folder praktikum pemrograman web pada folder htdocs**
3. **Lakukan penulisan sintak program pada text editor**
4. **Simpan file hasil penulisan sintak tersebut kedalam folder praktikum**
5. **Jalankan web browser**
6. **Jalankan web pada mode local dengan mengetikan localhost/folder praktikum/file sintak pada web browser**
7. **Output program akan tampil pada web browser**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Praktik** | **Output Program** | **Sintak Kode Program** | **Keterangan** |
| 1. | **Latihan 6** |  | <?php  class **Belanja** {      public $nama\_barang;      public $jumlah;      public $harga;      public $total;      public function **\_\_construct**($nama\_barang, $jumlah, $harga) {          $this->nama\_barang = $nama\_barang;          $this->jumlah = $jumlah;          $this->harga = $harga;          $this->total = $harga \* $jumlah;  **echo** "Constructor : Data Belanja '$this->nama\_barang' dibuat. \n<br>";      }      public function **\_\_destruct**() {  **echo** "Destruct : Data Belanja '$this->nama\_barang' dihapus. \n<br>";      }      function **getInfo**() {          return $this->nama\_barang . " (" . $this->harga . " x " . $this->jumlah . ") = " . $this->total;      }  }  **echo** "Masukkan jumlah barang belanja yang dibeli: ";  $jml = (int)**trim**(**fgets**(STDIN));  $barang = [];  $totalBelanja = 0;  for ($i = 1; $i <= $jml; $i++) {  **echo** "\nBarang ke-$i\n";  **echo** "Nama barang: ";      $nama\_barang = **trim**(**fgets**(STDIN));  **echo** "Harga satuan: ";      $harga = (int)**trim**(**fgets**(STDIN));  **echo** "Jumlah: ";      $jumlah = (int)**trim**(**fgets**(STDIN));      $item = new **Belanja**($nama\_barang, $jumlah, $harga);      $barang[] = $item;      $totalBelanja += $item->total;  }  **echo** "\n---------- Daftar Belanja ----------\n";  foreach ($barang as $item) {  **echo** $item->**getInfo**() . "\n";  }  **echo** "------------------------------------\n";  **echo** "Total Belanja : " . $totalBelanja . "\n";  ?> | Latihan construct belanja |
| 2. | **Latihan 6.1** |  | <?php  class **Fruit**{      public $name;      public $color;      function **\_\_construct**($name, $color) {          $this->name = $name;          $this->color = $color;      }      function **get\_name**() {          return $this->name;      }      function **get\_color**() {          return $this->color;      }  }  $apple = new **Fruit**("Apple", "red");  **echo** $apple->**get\_name**();  **echo**"<br>";  **echo** $apple->**get\_color**();  ?> | Latihan construct dengan class fruit |
| 3. | **Latihan 6.2** |  | <?php  class **segitiga** {      public $tinggi;      public $lebar;      function **\_\_construct**($tinggi, $lebar) {          $this->tinggi = $tinggi;          $this->lebar = $lebar;      }      function **Luas**(){          $luas = $this->tinggi \* $this->lebar/2;  **echo** "Tinggi Segitiga = " . $this->tinggi . " <br/>";  **echo** "Lebar Segitiga = " . $this->lebar . " <br/>";  **echo** "<br/><b>Luas Segitiga = $luas</b>";      }  }  $segitiga = new **segitiga** (200, 500);  $segitiga -> **Luas**();  ?> | Membuat kode program dengan constructor dengan menampilkan output luas, tinggi dan lebar |
| 4. | **Latihan 6.3** |  | <?php  class **KonversiSuhu** {  private $celsius;  *// Constructor*  public function **\_\_construct**($celsius) {  $this->celsius = $celsius;  }  *// Fungsi konversi (array)*  public function **konversiSemua**() {  return [  "celsius" => $this->celsius,  "kelvin" => $this->celsius + 273.15,  "fahrenheit" => (9/5) \* $this->celsius + 32,  "reamur" => (4/5) \* $this->celsius  ];  }  }  *// === INPUT GET ===*  $c = **isset**($\_GET['c']) ? $\_GET['c'] : 36; *// default 36 derajat*  *// === BUAT OBJEK ===*  $suhu = new **KonversiSuhu**($c);  *// === HASIL ===*  $hasil = $suhu->**konversiSemua**();  **echo** "Konversi Suhu dari Celcius <br/>";  *// === PERULANGAN + PERCABANGAN ===*  foreach ($hasil as $satuan => $nilai) {  if ($satuan == "celsius") {  **echo** "Suhu dalam Celsius = $nilai derajat <br>";  } elseif ($satuan == "kelvin") {  **echo** "Suhu dalam Kelvin = $nilai derajat <br>";  } elseif ($satuan == "fahrenheit") {  **echo** "Suhu dalam Fahrenheit = $nilai derajat <br>";  } elseif ($satuan == "reamur") {  **echo** "Suhu dalam Reamur = $nilai derajat <br>";  }  }  **echo** "Sekian konversi suhu yang bisa ditampilkan  ";  ?> | Membuat kode program dengan constructor dengan output konversi suhu |
| 5. | **Latihan 6.4** |  | <?php  class **manusia** {      var $nama;      var $warna;      function **\_\_construct**() {  **echo** "ini adalah isi method construct <br/>";      }      function **\_\_destruct**() {  **echo** "ini adalah isi method destruct <br/>";      }      function **tampilkan\_nama**(){          return "Nama saya mahasiwa SI <br/>";      }  }  $manusia = new **manusia**();  **echo** $manusia->**tampilkan\_nama**();  ?> | Latihan destruct |
| 6. | **Latihan 6.5** |  | <?php  **echo** "Siapa nama kamu: ";  $input\_nama = **fopen**("php://stdin","r");  $nama = **trim**(**fgets**($input\_nama));  **echo** "Hello $nama, apa kabar?\n";  ?> | Latihan menggunakan fungsi fopen |
| 7. | **Latihan 6.6** |  | <?php  **echo** "Dari mana anda berasal: ";  $asal = **trim**(**fgets**(STDIN));  **echo** "Oh, dari $asal ya\n";  ?> | Latihan menggunakan fungsi STDIN |
| 8. | **Latihan 6.7 (Praktikum)** |  | <?php  class **Karyawan** {      private $nama;      private $golongan;      private $jamLembur;      private $gajiPokok;      private $gajiLembur;      public function **\_\_construct**($nama, $golongan, $jamLembur) {  **echo** "Constructor dipanggil untuk karyawan: " . $nama . "\n";          $this->nama = $nama;          $this->golongan = $golongan;          $this->jamLembur = $jamLembur;          $this->gajiPokok = $this->**getGajiPokok**($golongan);          $this->gajiLembur = $this->**hitungGajiLembur**($jamLembur);      }      public function **\_\_destruct**() {  **echo** "Destructor dipanggil untuk karyawan: " . $this->nama . "\n";  **unset**($this->nama);  **unset**($this->golongan);  **unset**($this->jamLembur);  **unset**($this->gajiPokok);  **unset**($this->gajiLembur);      }      public function **setNama**($nama) {          $this->nama = $nama;      }      public function **getNama**() {          return $this->nama;      }      public function **setGolongan**($golongan) {          $this->golongan = $golongan;          $this->gajiPokok = $this->**getGajiPokok**($golongan);      }      public function **getGolongan**() {          return $this->golongan;      }      public function **setJamLembur**($jamLembur) {          $this->jamLembur = $jamLembur;          $this->gajiLembur = $this->**hitungGajiLembur**($jamLembur);      }      public function **getJamLembur**() {          return $this->jamLembur;      }      private function **getGajiPokok**($golongan) {          $gajiMap = [              "Ib" => 1250000,              "IIa" => 2000000,              "Ic" => 1300000,              "IIb" => 2100000,              "Id" => 1350000,              "IIc" => 2200000,              "IIIa" => 2400000,              "IId" => 2300000,              "IIIb" => 2500000,              "Va" => 2800000,              "IIIc" => 2600000,              "IVb" => 2900000,              "IIId" => 2700000,              "IVc" => 3000000,              "IVd" => 3100000          ];          return **isset**($gajiMap[$golongan]) ? $gajiMap[$golongan] : 0;      }      private function **hitungGajiLembur**($jamLembur) {          $gajiPerJam = 15000;          return $jamLembur \* $gajiPerJam;      }      public function **getTotalGaji**() {          return $this->gajiPokok + $this->gajiLembur;      }      public function **displayData**() {          return [              'nama' => $this->nama,              'golongan' => $this->golongan,              'jam\_lembur' => $this->jamLembur,              'total\_gaji' => $this->**getTotalGaji**()          ];      }  }  $dataKaryawan = [      ["Winny", "IIb", 30],      ["Stendy", "IIIc", 32],      ["Alfred", "IVb", 30],      ["Ferdinand", "IIIc", 40],  ];  $karyawanList = [];  foreach ($dataKaryawan as $data) {      $karyawan = new **Karyawan**($data[0], $data[1], $data[2]);      $karyawanList[] = $karyawan;  }  **echo** "\n" . **str\_repeat**("=", 80) . "\n";  **echo** "DATA KARYAWAN DAN GAJI\n";  **echo** **str\_repeat**("=", 80) . "\n";  **echo** **sprintf**("%-15s %-10s %-15s %-15s\n", "Nama Karyawan", "Golongan", "Total Jam Lembur", "Total Gaji");  **echo** **str\_repeat**("-", 80) . "\n";  foreach ($karyawanList as $karyawan) {      $data = $karyawan->**displayData**();  **echo** **sprintf**("%-15s %-10s %-15s Rp %-12.0f\n",          $data['nama'],          $data['golongan'],          $data['jam\_lembur'],          $data['total\_gaji']      );  }  **echo** **str\_repeat**("-", 80) . "\n";  foreach ($karyawanList as $karyawan) {  **unset**($karyawan);  }  **echo** "\nSemua objek karyawan telah dihapus.\n";  **echo** "Program selesai dieksekusi.\n";  ?> | Prktikum menghitung gaji karyawan berdasarkan golongan dan jam lembur. |

**Analisis dan Penjelasan Kode Program dalam Soal Tes Praktikum :**

Kode PHP ini mendefinisikan sebuah kelas Karyawan yang digunakan untuk menyimpan dan mengelola data karyawan, seperti nama, golongan, jam lembur, serta perhitungan gaji pokok dan gaji lembur berdasarkan golongan dan jumlah jam lembur. Pada saat objek Karyawan dibuat, constructor secara otomatis menetapkan nilai properti dan menghitung gaji pokok serta gaji lembur. Ada juga fungsi setter dan getter untuk mengubah dan mengambil nilai properti, serta fungsi private untuk menentukan gaji pokok berdasarkan golongan dan menghitung gaji lembur. Total gaji dihitung dengan menjumlahkan gaji pokok dan lembur, dan data karyawan bisa ditampilkan dalam format array.

Setelah data karyawan disimpan dalam array, program membuat objek Karyawan untuk setiap data dan menyimpannya dalam list. Selanjutnya, program menampilkan data karyawan dalam format tabel yang rapi, memperlihatkan nama, golongan, jam lembur, dan total gaji dalam bentuk rupiah. Setelah itu, objek-objek karyawan dihapus dengan memanggil destructor secara otomatis, memastikan memori dibersihkan. Program ini memperlihatkan penggunaan prinsip dasar OOP di PHP seperti enkapsulasi, constructor, destructor, dan pemrosesan data secara terstruktur dan efisien.