

**10111026 – KARISMA AINUN NISSA**

**SISTEM INFORMASI 2B**

JOB SHEET   
PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN 5

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 : Inheritance, Overloading, dan Overriding |
| Mata Kuliah : PBO | Judul : Inheritance, Overloading, dan Overriding |
| Tanggal : 17 September 2025 | Pertemuan ke : 5 |

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

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

# 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** |
|  | **Latihan Soal 5** |  | <?php  class **Warung** {      private $barang;      public function **\_\_construct**($barang) {          $this->barang = $barang;      }      public function **menampilkanBarang** (){          foreach ($this->barang as $namaBarang => $harga) {  **echo** "-$namaBarang dengan harga Rp $harga\n";          }      }  }  $barang = [      "Kecap" => 3000,      "Tepung Terigu" => 4000  ];  $barang1 = new **Warung**($barang);  $barang1 -> **menampilkanBarang**();  ?> | Mengulas Kembali materi pertemuan 4 |
| 1. | **Latihan Soal 5.1** |  | <?php  *// class induk*  class **manusia** {  *// property class manusia*      public $nama\_saya;  *// method pada class manusia*      function **berinama**($saya) {          $this->nama\_saya = $saya;      }  }  *// class turunan atau sub class dari class manusia*  *// kita menghubungkan class dengan syntax extends*  class **teman** extends **manusia** {  *// property class teman*      public $nama\_teman;  *// method pada class teman*      function **berinamateman**($teman) {          $this->nama\_teman = $teman;      }  }  *// instansiasi class teman*  $objectteman = new **teman**;  *// method berinama ada di class manusia*  *// tetap bisa dipanggil karena class teman mewarisi class manusia*  $objectteman->**berinama**("Karisma");  $objectteman->**berinamateman**("Khaila");  *// menampilkan isi property*  **echo** "Nama Saya : " . $objectteman->nama\_saya . "<br/>";  **echo** "Nama Teman Saya : " . $objectteman->nama\_teman;  ?> | Contoh program keturunan class manusia dan teman. |
| 2. | **Latihan Soal 5.2** |  | <?php  class **manusia** {      public $nama;      public $umur;      public $gender;      function **biodata**() {  **echo** "Selamat Datang<br/>";      }      function **getInfo**() {  **echo** "Nama  : " . $this->nama . "<br/>";  **echo** "Umur  : " . $this->umur . "<br/>";  **echo** "JK    : " . $this->gender . "<br/>";      }  }  class **ayah** extends **manusia** {      function **pekerjaan**() {  **echo** "Pegawai Negeri Sipil<br/>";      }  }  class **ibu** extends **manusia** {      function **pekerjaan**() {  **echo** "Ibu Rumah Tangga<br/>";      }  }  class **anak** extends **manusia** {      function **pekerjaan**() {  **echo** "Pelajar<br/>";      }  }  *// --- Buat object ayah ---*  $objectAyah = new **ayah**();  $objectAyah->nama = "Joko";  $objectAyah->umur = 45;  $objectAyah->gender = "Laki-laki";  $objectAyah->**biodata**();  $objectAyah->**getInfo**();  $objectAyah->**pekerjaan**();  **echo** "<br/>";  *// --- Buat object ibu ---*  $objectIbu = new **ibu**();  $objectIbu->nama = "Susi";  $objectIbu->umur = 38;  $objectIbu->gender = "Perempuan";  $objectIbu->**biodata**();  $objectIbu->**getInfo**();  $objectIbu->**pekerjaan**();  **echo** "<br/>";  *// --- Buat object anak ---*  $objectAnak = new **anak**();  $objectAnak->nama = "Andi";  $objectAnak->umur = 15;  $objectAnak->gender = "Laki-laki";  $objectAnak->**biodata**();  $objectAnak->**getInfo**();  $objectAnak->**pekerjaan**();  ?> | Contoh program keturunan dari class keluarga kemudian ada ayah, ibu, dan anak. |
| 3. | **Latihan Soal 5.3** |  | <?php  *// Parent class*  abstract class **BangunDatar** {      abstract public function **luas**();      abstract public function **keliling**();  }  *// Class Persegi*  class **Persegi** extends **BangunDatar** {      private $sisi;      public function **\_\_construct**($sisi) {          $this->sisi = $sisi;      }      public function **luas**() {          return $this->sisi \* $this->sisi;      }      public function **keliling**() {          return 4 \* $this->sisi;      }  }  *// Class Lingkaran*  class **Lingkaran** extends **BangunDatar** {      private $r;      public function **\_\_construct**($r) {          $this->r = $r;      }      public function **luas**() {          return M\_PI \* $this->r \* $this->r;      }      public function **keliling**() {          return 2 \* M\_PI \* $this->r;      }  }  *// Class Persegi Panjang*  class **PersegiPanjang** extends **BangunDatar** {      private $panjang, $lebar;      public function **\_\_construct**($panjang, $lebar) {          $this->panjang = $panjang;          $this->lebar = $lebar;      }      public function **luas**() {          return $this->panjang \* $this->lebar;      }      public function **keliling**() {          return 2 \* ($this->panjang + $this->lebar);      }  }  *// Class Segitiga*  class **Segitiga** extends **BangunDatar** {      private $alas, $tinggi;      public function **\_\_construct**($alas, $tinggi) {          $this->alas = $alas;          $this->tinggi = $tinggi;      }      public function **luas**() {          return 0.5 \* $this->alas \* $this->tinggi;      }      public function **keliling**() {  *// asumsinya segitiga sama sisi (biar simple)*          return 3 \* $this->alas;      }  }  *// Main*  **echo** "=== Output Luas dan Keliling ===<br>";  $persegi = new **Persegi**(5);  **echo** "Persegi - Luas: ".$persegi->**luas**().", Keliling: ".$persegi->**keliling**()."<br>";  $lingkaran = new **Lingkaran**(7);  **echo** "Lingkaran - Luas: ".$lingkaran->**luas**().", Keliling: ".$lingkaran->**keliling**()."<br>";  $pp = new **PersegiPanjang**(10, 4);  **echo** "Persegi Panjang - Luas: ".$pp->**luas**().", Keliling: ".$pp->**keliling**()."<br>";  $segitiga = new **Segitiga**(6, 8);  **echo** "Segitiga - Luas: ".$segitiga->**luas**().", Keliling: ".$segitiga->**keliling**()."<br>";  ?> | Membuat Class Diagram menjadi Code Inheritance PHP dengan ouput luas dan keliling pada masing-masing bangun datar. |
| 4. | **Latihan Soal 5.4 (Latihan Soal Inheritance)** |  | <?php  *// Class induk*  class **Kendaraan** {  *// Bisa tambahkan properti umum kendaraan kalau perlu*  }  *// Class turunan*  class **Pesawat** extends **Kendaraan** {      private $tinggiMaks;      private $kecepatanMaks;      public $harga;      public function **\_\_construct**($harga) {          $this->harga = $harga;      }      public function **setTinggiMaks**($tinggi) {          $this->tinggiMaks = $tinggi;      }      public function **setKecepatanMaks**($kecepatan) {          $this->kecepatanMaks = $kecepatan;      }      public function **bacaTinggiMaks**() {          return $this->tinggiMaks;      }      public function **getKecepatanMaks**() {          return $this->kecepatanMaks;      }      public function **biayaOperasional**() {          $hargaDalamRupiah = $this->harga \* 1000000; *// harga dalam juta*          if ($this->tinggiMaks > 5000 && $this->kecepatanMaks > 800) {              return 0.30 \* $hargaDalamRupiah;          } elseif ($this->tinggiMaks > 3000 && $this->tinggiMaks <= 5000 && $this->kecepatanMaks >= 500 && $this->kecepatanMaks <= 800) {              return 0.25 \* $hargaDalamRupiah;          } elseif ($this->tinggiMaks > 3000 && $this->kecepatanMaks < 500) {              return 0.10 \* $hargaDalamRupiah;          } else {              return 0.05 \* $hargaDalamRupiah;          }      }  }  *// Buat objek pesawat*  $boeing737 = new **Pesawat**(2000); *// harga juta*  $boeing737->**setTinggiMaks**(7500);  $boeing737->**setKecepatanMaks**(650);  $boeing747 = new **Pesawat**(1500);  $boeing747->**setTinggiMaks**(3500);  $boeing747->**setKecepatanMaks**(700);  $cassa = new **Pesawat**(750);  $cassa->**setTinggiMaks**(2500);  $cassa->**setKecepatanMaks**(500);  *// Fungsi untuk tampilkan hasil*  function **tampilkanBiaya**($nama, $pesawat) {      $harga = **number\_format**($pesawat->harga \* 1000000, 0, ',', '.');      $tinggi = $pesawat->**bacaTinggiMaks**();      $kecepatan = $pesawat->**getKecepatanMaks**();      $biaya = **number\_format**($pesawat->**biayaOperasional**(), 0, ',', '.');  **echo** "Biaya operasional pesawat '$nama' dengan harga Rp {$harga} yang memiliki tinggi maksimum {$tinggi} feet dan kecepatan maksimum {$kecepatan} km/jam adalah Rp {$biaya}<br><br>";  }  *// Tampilkan semua*  **tampilkanBiaya**("Boeing 737", $boeing737);  **tampilkanBiaya**("Boeing 747", $boeing747);  **tampilkanBiaya**("Cassa", $cassa);  ?> | Latihan soal inheritance soal peswat |
| 5. | **Latihan Soal 5.5** |  | <?php     class **Shape** {   const PI = 3.142 ;   function **\_\_call**($name,$arg){   if($name == 'area')   switch(**count**($arg)){   case 0 : return 0 ;   case 1 : return self::PI \* $arg[0] ;   case 2 : return $arg[0] \* $arg[1];   }   }   }   $circle = new **Shape**();  **echo** $circle->**area**(3);   $rect = new **Shape**();  **echo** $rect->**area**(8,6);   ?> | Latihan Overloading |
| 6. | **Latihan Soal 5.6** |  | <?php     class **Base** {   function **display**() {  **echo** "\nBase class function declared final!";   }   function **demo**() {  **echo** "\nBase class function!";   }   }   class **Derived** extends **Base** {   function **demo**() {  **echo** "\nDerived class function!";   }   }   $ob = new **Base**;   $ob->**demo**();   $ob->**display**();   $ob2 = new **Derived**;   $ob2->**demo**();   $ob2->**display**();   ?> | Latihan Overriding |
| 7. | **Latihan Soal 5.7 (Overloading)** |  | <?php    class **Test**   {   public function **\_\_construct**()   {  *//Your logic for constructor*   }   public function **\_\_call**($method\_name , $parameter)   {   if($method\_name == "Function")  {   $count = **count**($parameter);   switch($count)   {   case "1":  **echo** "You are passing 1 argument";   break;   case "2": *//Incase of 2 parameter*  **echo** "You are passing 2 parameter";   break;   default:   throw new **exception**("Bad argument");   }   }   else   { throw new **exception**("Function $method\_name does not exists ");   }   }   }   $a = new **Test**();   $a->**Function**("ankur");   $a->**Function**("techflirt" , "ankur");   ?> | Latihan analisis overloading |
| 8. | **Latihan Soal 5.7 (Overriding)** |  | <?php    class **testParent**   {   public function **f1**()   {  **echo** 1;   }   public function **f2**()   {  **echo** 2;   }   }   class **testChild**   {   function **f2**($a)   {  **echo** $a;   }   }   $a = new **testChild**();   $a->**f2**('ankur');*//it will print ankur*   ?> | Latihan analisis overriding |
| 9. | **Latihan Soal 5.8** |  | <?php  *//Materi Inheritance, Overloading, Oveririding*  class **Warung** {      public $namaBarang;      public $harga;      public function **\_\_construct**($namaBarang, $harga) {          $this->namaBarang = $namaBarang;          $this->harga = $harga;      }      public function **informasi**() {  **echo** "Barang : $this->namaBarang, Harga : Rp $this->harga<br>";      }  }  class **Warung2** extends **Warung** {      public $exp;      public function **\_\_construct**($namaBarang, $harga, $exp) {          parent::**\_\_construct**($namaBarang, $harga);          $this->exp = $exp;      }  *//overriding*      public function **informasi**() {  **echo** "Barang2 : $this->namaBarang, Harga : Rp $this->harga, Kadaluarsa : $this->exp<br>";      }  }  *//overloading*  class **Warung3** {      public function **\_\_call**($namaBarang, $x){          if ($namaBarang == "total") {              if (**count**($x) == 1) {                  return $x[0];              }              else if (**count**($x) == 2) {                  return $x[0]\* $x[1];              }              else {                  return 0;              }          }      }  }  $barang1 = new **Warung**("Susu Kotak", 6000);  $barang1->**informasi**();  $barang2 = new **Warung2**("Yogurt", 12000, "15-11-2025");  $barang2->**informasi**();  $barang3 = new **Warung3**();  **echo** "Harga Indomie : Rp " . $barang3->**total**(4000) . "<br>";  **echo** "Harga Telur : Rp " . $barang3->**total**(2000, 5) . "<br>";  ?> | Mengenal inheritance, overriding dan overloading dengan praktik langsung dalam file ini dengan class Warung, warung2 dan Warung3. |
|  | **Latihan Soal Praktikum** |  | <?php  class **Employee** {      protected $gaji;      protected $masaKerja;      public function **\_\_construct**($gaji, $masaKerja) {          $this->gaji = $gaji;          $this->masaKerja = $masaKerja;      }      public function **getgaji**() {          return $this->gaji;      }      public function **getmasaKerja**() {          return $this->masaKerja;      }      public function **hitungBonus**() {          return 0;      }      public function **getTotalGaji**() {          return $this->gaji + $this->**hitungBonus**();      }  }  class **Programmer** extends **Employee** {      public function **hitungBonus**() {          $tahun = $this->masaKerja;          if ($tahun < 1) {              return 0;          } elseif ($tahun >= 1 && $tahun <= 10) {              return $this->gaji \* 0.01 \* $tahun;          } else { *// > 10 tahun*              return $this->gaji \* 0.02 \* $tahun;          }      }  }  class **Direktur** extends **Employee** {      public function **hitungBonus**() {  *// Bonus 0.5 \* masaKerja \* gaji*          return $this->gaji \* 0.5 \* $this->masaKerja;      }      public function **hitungTunjangan**() {  *// tunjangan 0.1 \* masaKerja \* gaji*          return $this->gaji \* 0.1 \* $this->masaKerja;      }      public function **getTotalGaji**() {          return $this->gaji + $this->**hitungBonus**() + $this->**hitungTunjangan**();      }  }  class **PegawaiMingguan** extends **Employee** {      private $hargaBarang;      private $stok;      private $penjualan;      public function **\_\_construct**($gaji, $masaKerja, $hargaBarang, $stok, $penjualan) {          parent::**\_\_construct**($gaji, $masaKerja);          $this->hargaBarang = $hargaBarang;          $this->stok = $stok;          $this->penjualan = $penjualan;      }      public function **hitungBonus**() {          $totalPenjualan = $this->penjualan / $this->stok;          if ($totalPenjualan > 0.7) {  *// 10% dari harga per item per penjualan*              return $this->hargaBarang \* 0.10 \* $this->penjualan;          } else {  *// 3% dari harga per item per penjualan*              return $this->hargaBarang \* 0.03 \* $this->penjualan;          }      }      public function **getTotalGaji**() {          return $this->gaji + $this->**hitungBonus**();      }  }  $programmer = new **Programmer**(10000000, 5);  **echo** "Programmer <br>";  **echo** "Gaji Pokok: Rp " . **number\_format**($programmer->**getgaji**(), 0, ',', '.') . "<br>";  **echo** "Bonus: Rp " . **number\_format**($programmer->**hitungBonus**(), 0, ',', '.') . "<br>";  **echo** "Total Gaji: Rp " . **number\_format**($programmer->**getTotalGaji**(), 0, ',', '.') . "<br><br>";  $direktur = new **Direktur**(15000000, 3);  **echo** "Direktur <br>";  **echo** "Gaji Pokok: Rp " . **number\_format**($direktur->**getgaji**(), 0, ',', '.') . "<br>";  **echo** "Bonus: Rp " . **number\_format**($direktur->**hitungBonus**(), 0, ',', '.') . "<br>";  **echo** "Tunjangan: Rp " . **number\_format**($direktur->**hitungTunjangan**(), 0, ',', '.') . "<br>";  **echo** "Total Gaji: Rp " . **number\_format**($direktur->**getTotalGaji**(), 0, ',', '.') . "<br><br>";  $pegawaiMingguan = new **PegawaiMingguan**(2000000, 2, 100000, 100, 80);  **echo** "Pegawai Mingguan <br>";  **echo** "Gaji Pokok: Rp " . **number\_format**($pegawaiMingguan->**getgaji**(), 0, ',', '.') . "<br>";  **echo** "Bonus: Rp " . **number\_format**($pegawaiMingguan->**hitungBonus**(), 0, ',', '.') . "<br>";  **echo** "Total Gaji: Rp " . **number\_format**($pegawaiMingguan->**getTotalGaji**(), 0, ',', '.') . "<br>";  ?> | Studi kasus menghitung gaji karyawan dengan menerapkan inhiritance dan overriding. |

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

Kode program PHP di atas menggunakan konsep pewarisan (inheritance) dalam pemrograman berorientasi objek (OOP) untuk merepresentasikan berbagai jenis karyawan dengan perhitungan bonus dan total gaji yang berbeda. Kelas induk Employee memiliki atribut dasar seperti gaji dan masaKerja, serta menyediakan metode umum getgaji(), getmasaKerja(), hitungBonus() (default 0), dan getTotalGaji() yang menjumlahkan gaji pokok dan bonus. Tiga kelas turunan (Programmer, Direktur, dan PegawaiMingguan) mewarisi dari Employee dan masing-masing mengoverride metode hitungBonus() untuk menghitung bonus sesuai ketentuan jenis karyawan tersebut. Selain itu, kelas Direktur juga memiliki metode tambahan hitungTunjangan() dan override untuk getTotalGaji() yang memperhitungkan tunjangan.

Dalam blok eksekusi program, objek dari masing-masing kelas dibuat dan dipanggil untuk menampilkan informasi gaji pokok, bonus, dan total gaji dengan format mata uang Indonesia. Misalnya, seorang Programmer dengan gaji Rp10.000.000 dan masa kerja 5 tahun mendapatkan bonus sebesar 5% dari gaji (0.01 \* 5), sedangkan Direktur dengan gaji Rp15.000.000 dan masa kerja 3 tahun mendapat bonus dan tunjangan masing-masing sebesar 150% dan 30% dari gaji. PegawaiMingguan dihitung berdasarkan penjualan terhadap stok: bila penjualan lebih dari 70% stok, bonus adalah 10% dari total nilai penjualan; jika kurang, hanya 3%. Hal ini menunjukkan fleksibilitas OOP dalam menangani berbagai tipe entitas dengan perilaku yang berbeda namun berbagi struktur dasar yang sama.