OOP

|  |
| --- |
| Encapsulation Inheritance |
| indexOOPEncap.php  <?php  //oop  //class= blueprint, object=hasil cetak dari class, proses cetak object | 'buat object dari class' disbeut instansiasi  //menghubungkan class yang terdapat di file lain  require 'inheritRobot.php';  //instansiasi dalam php cirinya ada kata new  $robot1 =new *robot*('satu sssss',20);  $robot2 =new *robot*('duaaa aaaa',30);  //set property, tanda -> adalah operator khusus objek disebut Object Operator, diikuti nama properti tanpa tanda $ jika properti berupa variabel, dan diakhiri untuk method()  $robot1->set\_berat(100);  echo 'bunyinya.. ' . $robot1->get\_suara() .'. beratnya.. ' . $robot1->get\_berat() . '<br>';  echo 'bunyinya.. ' . $robot2->get\_suara() . '. beratnya.. ' . $robot2->get\_berat() .'<br>';  //inherit  $robothewan = new *robot\_hewan*('inheritance aaaa',30);  echo $robothewan->get\_kekuatan();  //override  $goodSoldier = New *Soldier*('Thomas', 'Officer');  echo $goodSoldier->getName();  echo $goodSoldier->getRank();  echo $goodSoldier->getFull();  echo "<br /n>";  $goodSoldier->setRank('Colonel');  $goodSoldier->setName('Mustard');  echo $goodSoldier->getName();  echo $goodSoldier->getRank();  echo $goodSoldier->getFull();  //overriding  $tumbuhan = new *tumbuhan*;  $tumbuhan\_hias = new *tumbuhan\_hias*;  echo '<br>contoh overriding<br>';  echo($tumbuhan->daun().' ');  echo($tumbuhan\_hias->daun());  ?> |
| inheritRobot.php  <?php  //kelas  *class* robot{  //properti  public $suara;  public $berat;  //konstruktor  public *function* \_\_construct($suara, $berat){  $this->suara = $suara;  $this->berat = $berat;  }  //encapsulasion = penyembunyian variabel atau method, kasus ini method tidak disembunyikan, setget gak harus selalu ada kata set/get nya kok  public *function* set\_suara($suara){  $this->suara=$suara;  }  public *function* get\_suara(){  return $this->suara;  }  public *function* set\_berat($berat){  $this->berat=$berat;  }  public *function* get\_berat(){  return $this->berat;  }  }  //inherit / pewarisan, parent robot memiliki child robot\_hewan  *class* robot\_hewan extends *robot*{  public *function* get\_kekuatan(){  echo 'inherit robot hewan bisa berenang <br><br>';  }  }  //interface  *interface* Human {  public *function* getName();  public *function* setName($name);  }  // class abstraction  abstract *class* Military {  //encapsuation $rank  private $rank;  public *function* \_\_construct($rank) {  $this->rank = $rank;  }  public *function* setRank($rank) {  $this->rank = $rank;  }  public *function* getRank() {  return $this->rank;  }  }  //inherit class abstraction parent Military child Soldier  *class* Soldier extends *Military* implements *Human* {  private $name;  public *function* \_\_construct($name, $rank) {  $this->name = $name;  *parent*::\_\_construct($rank); # parent::setName($rank);  }  public *function* setName($name) {  $this->name = $name;  }  public *function* getName() {  return "My name is: " . $this->name . "<br /n>";  }  public *function* getRank() {  return "My rank is: " . *parent*::getRank() . "<br /n>";;  }  //polymorphism tambahin overlord  public *function* getFull() {  return "I am " . *parent*::getRank() . " {$this->name}<br /n>";  }  }  //overriding exsample  *class* tumbuhan {  *function* daun() {  return "hijau";  }  }  *class* tumbuhan\_hias extends *tumbuhan* {  *function* daun() {  return "merah";  }  }  }  *class* Soldier extends *Military* implements *Human* {  private $name;  //polymorphism  public *function* \_\_construct($name, $rank) {  $this->name = $name;  *parent*::\_\_construct($rank); # parent::setName($rank);  }  public *function* setName($name) {  $this->name = $name;  }  public *function* getName() {  return "My name is: " . $this->name . "<br /n>";  }  public *function* getRank() {  return "My rank is: " . *parent*::getRank() . "<br /n>";;  }  public *function* getFull() {  return "I am " . *parent*::getRank() . " {$this->name}<br /n>";  }  } |

Result

