Property dan **method** tidak lain adalah sebutan untuk *variabel* dan *function* yang berada di dalam class. Cara penulisannya pun sama seperti variabel dan function, tapi dengan tambahan *access modifier* di awal penulisan seperti contoh berikut:

05.property_and_method.php

```
1   <?php
2   class Produk {
3     public $sku = "001";
4     public $merek = "Samsung";
5     public $harga = 4000000;
6
7     public function pesanProduk(){
8       return "Produk dipesan...";
9     }
10  }
11
12  $televisi = new Produk();
```

Di baris 3 – 5 saya membuat 3 buah property dengan nama $sku, $merek dan $harga. Pada awal penulisan ketiganya, terdapat *keyword* public yang disebut sebagai *access modifier* atau *visibility*.

Access modifier berfungsi untuk mengatur batasan hak akses dari sebuah property atau method. *Access modifier* public artinya semua property dan method bisa diakses dari mana saja, termasuk dari luar class. Kita akan mempelajari *access modifier* dengan lebih detail pada bahasan tersendiri, untuk saat ini anggap saja semua property dan method harus memiliki awalan public, jika tidak PHP akan mengeluarkan pesan error.

Untuk ketiga property: $sku, $merek dan $harga masing-masing saya input dengan string "001", "Samsung" dan integer 4000000. Cara pemberian nilai ini sama seperti variabel biasa, yakni menggunakan operator *assignment* tanda sama dengan " = ". Di baris 7 terdapat sebuah function bernama pesanProduk(). Karena function ini berada di dalam class, namanya berubah menjadi **method**. Sama seperti property, method juga diawali dengan *access modifier* yang saya set sebagai public. Isi dari method pesanProduk() berupa perintah return "Produk dipesan...". Selanjutnya di baris 12, class Produk saya inisialisasi ke dalam variabel $televisi.

Cara Mengakses Property dan Method

Bagaimana cara mengakses property dan method? Ingat bahwa dalam pemrograman berorientasi objek, yang akan kita akses adalah object, bukan class. Artinya untuk mengakses property dan method class Produk, di lakukan dari object yang dalam contoh di atas adalah variabel $televisi:

06.property_and_method_access.php

```php
<?php
class Produk {
  public $sku = "001";
  public $merek = "Samsung";
  public $harga = 4000000;

  public function pesanProduk(){
    return "Produk dipesan...";
  }
}

$televisi = new Produk();
echo $televisi->sku;              // 001
echo "<br>";
echo $televisi->merek;            // Samsung
echo "<br>";
echo $televisi->harga;            // 4000000
echo "<br>";
echo $televisi->pesanProduk();    // Produk dipesan...
```

Untuk mengakses property dan method yang ada di dalam sebuah object, kita menggunakan operator tanda panah " -> ", yakni gabungan dari tanda minus " - " dan tanda lebih besar " > ". Setelah membuat object $televisi di baris 12, saya mengakses isi property $sku dengan perintah $televisi->sku. Perhatikan bahwa kita tidak menulis tanda dollar untuk $sku. Penulisan yang benar adalah $televisi->sku, bukan $televisi->$sku. Begitu juga dengan cara mengakses property lain, yakni $televisi->merek dan $televisi>harga. Perintah Echo sendiri Di pakai untuk Menampilkan nilai Dari setiap property. Di baris 19, saya mengakses method pesanProduk(). Caranya sangat mirip seperti pengaksesan property, namun kali ini dengan tambahan tanda kurung di akhir, yakni $televisi>pesanProduk(). Kemudian, bagaimana cara mengisi nilai ke dalam property? Berikut contohnya:

07.property_and_method_access_outside.php

```php
<?php
class Produk {
  public $sku = "001";
  public $merek = "Samsung";
  public $harga = 4000000;

  public function pesanProduk(){
    return "Produk dipesan...";
  }
}

$mesinCuci = new Produk();
$mesinCuci->sku = "002";
$mesinCuci->merek = "LG";
$mesinCuci->harga = 1500000;

echo $mesinCuci->sku;              // 002
echo "<br>";
echo $mesinCuci->merek;            // LG
echo "<br>";
echo $mesinCuci->harga;            // 1500000
echo "<br>";
echo $mesinCuci->pesanProduk();    // Produk dipesan...
```

Kali ini saya membuat object $mesinCuci yang berasal dari class Product di baris 12. Di baris 13, 14 dan 15 saya mengakses ketiga property dari object $mesinCuci dan mengubah nilainya. Cara pemberian nilai ini kurang lebih sama seperti pemberian nilai ke dalam variable PHP biasa, hanya saja sekarang menggunakan tanda panah "->". Di baris 17 – 23 saya menampilkan kembali isi property yang sudah diubah serta mengakses method pesanProduk(). Terlihat bahwa isi ketiga property sudah ter-update dengan nilai baru. Sebuah class bisa diinisialisasi untuk banyak object, seperti contoh berikut:

```php
<?php
class Produk {
  public $sku = "000";
  public $merek = "";
  public $harga = 0;

  public function pesanProduk(){
    return "Produk dipesan...";
  }
}

$televisi = new Produk();
$televisi->sku = "001";
$televisi->merek = "4000000";
$televisi->harga = 1500000;

$mesinCuci = new Produk();
$mesinCuci->sku = "002";
$mesinCuci->merek = "LG";
$mesinCuci->harga = 1500000;

$speaker = new Produk();
$speaker->sku = "003";
$speaker->merek = "Edifier ";
$speaker->harga = 950000;


print_r ($televisi);
echo "<br>";
print_r ($mesinCuci);
echo "<br>";
print_r ($speaker);
```

Terdapat sedikit perubahan untuk class Produk. Kali ini saya memberi nilai awal "000", string kosong "" dan angka 0 ke dalam property $sku, $merek dan $harga. Nilai "dummy" ini nantinya akan ditimpa pada saat pembuatan object. Di baris 12 – 15, saya membuat object $televisi dari class Produk(), kemudian mengisi ulang ketiga property, hal yang sama juga dilakukan untuk object $televisi di baris 17 – 20, serta object $speaker di baris 22 – 25. Terakhir, saya menggunakan function print_r() untuk menampilkan isi ketiga object. Function print_r() ini bisa dianggap sebagai versi sederhana dari var_dump(), yakni untuk menampilkan isi detail sebuah variabel (termasuk object). Function print_r() dan var_dump() sangat pas dipakai untuk proses pencarian kesalahan (*debugging*). Hasilnya, ketiga object sudah berisi nilai yang berbeda meskipun semuanya berasal dari class yang sama, yakni Produk.