**BAB IV**

**FUNCTION DAN METHOD**

* 1. Tujuan

1. Memahami pengertian serta konsep *function* dan *method.*
2. Mampu membedakan penggunaan *function* dan *method.*
3. Memahami penggunaan *return type function* dan *non-return type function.*
4. Mengetahui macam-macam *modifier* dalam Pemrograman Berbasis Objek.
5. Mengetahui konsep *instance* untuk membuat objek dari suatu *class*.
6. Mengimplementasikan *function* dan *method* dalam bahasa pemrograman C#, Java, dan PHP.
   1. Dasar Teori
      1. **Konsep Pemrograman Menggunakan *Function***

Fungsi / *Function* adalah satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang di eksekusi ketika dipanggil dari bagian lain dalam suatu program.

Tujuan pembuatan fungsi adalah:

1. Memudahkan dalam pembuatan program.
2. Menghemat ukuran program.

Keuntungan memakai fungsi:

1. Menguraikan tugas pemrograman rumit menjadi langkah-langkah yang lebih sederhana atau kecil.
2. Mengurangi duplikasi kode (kode yang sama ditulis berulang-ulang) dalam program.
3. Dapat menggunakan kode yang ditulis dalam berbagai program yang berbeda.
4. Memecah program besar menjadi kecil sehingga dapat dikerjakan oleh programmer-programmer atau dipecah menjadi beberapa tahap sehingga mempermudah pengerjaan dalam sebuah projek.
5. Menyembunyikan informasi dari *user* sehingga mencegah adanya perbuatan iseng seperti memodifikasi atau mengubah program yang kita buat.
6. Meningkatkan kemampuan pelacakan kesalahan, jika terjadi suatu kesalahan kita tinggal mencari fungsi yang bersangkutan saja dan tak perlu mencari kesalahan tersebut di seluruh program.

Bentuk umum sebuah fungsi adalah sebagai berikut:

1. Tipe data namafungsi (daftarparameter).

Mendefinisikan *Function.*

1. *Return Type* − Fungsi dapat mengembalikan nilai. *Return\_type* adalah tipe data dari nilai fungsi yang dikembalikan. Beberapa fungsi melakukan operasi yang diinginkan tanpa mengembalikan nilai.
2. *Function Name* − Ini adalah nama sebenarnya dari fungsi. Nama fungsi dan daftar parameter bersama merupakan *function signature.*
3. Parameter *List* − Parameter seperti *placeholder.* Saat sebuah fungsi dipanggil, Anda melewatkan sebuah nilai sebagai parameter. Nilai ini disebut sebagai parameter atau argumen Daftar parameter mengacu pada tipe, urutan, dan jumlah parameter fungsi. Parameter bersifat opsional; artinya, fungsi tidak mengandung parameter.
4. *Function Body* − berisi kumpulan pernyataan yang mendefinisikan fungsi yang dilakukannya.

*(Sumber:* [*http://www.sistem-informasi.xyz/2017/09/pengertian-function-dalam-bahasa.html*](http://www.sistem-informasi.xyz/2017/09/pengertian-function-dalam-bahasa.html)*)*

*(Sumber:* [*http://setyo2511.blogspot.com/2017/01/bab-6-fungsi-by-setiyo-budi.html*](http://setyo2511.blogspot.com/2017/01/bab-6-fungsi-by-setiyo-budi.html)*)*

* + 1. **Jenis-jenis *Function***

Ada dua jenis fungsi yaitu:

1. *Void* (Fungi tanpa nilai balik)

Fungsi yang *void* sering disebut juga prosedur. Disebut *void* karena fungsi tersebut tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses fungsi tersebut.

Ciri-ciri dari jenis fungsi *Void* adalah sebagai berikut:

* Tidak adanya *keyword return.*
* Tidak adanya tipe data di dalam deklarasi fungsi.
* Menggunakan *keyword void.*
* Tidak dapat langsung ditampilkan hasilnya.
* Tidak memiliki nilai kembalian fungsi.
* *Keyword void* juga digunakan jika suatu *function* tidak mengandung suatu parameter apapun.

1. *NonVoid* (Fungsi dengan nilai balik)

Fungsi *non-void* disebut juga function. Disebut *non-void* karena mengembalikan nilai kembalian yang berasal dari keluaran hasil proses *function* tersebut.

Ciri-ciri dari jenis fungsi *non void* adalah sebagai berikut:

* + Ada *keyword return.*
  + Ada tipe data yang mengawali fungsi.
  + Tidak ada *keyword void.*
  + Memiliki nilai kembalian.
  + Dapat dianalogikan sebagai suatu variabel yang memiliki tipe data tertentu.
  + Sehingga dapat langsung ditampilkan hasilnya.

*(Sumber: https://belajarohbelajar.blogspot.com/2012/04/jenis-jenis-fungsi-pada-c.htm)*

* + 1. **Konsep Pemrograman Menggunakan *Method***

*Method* pada dasarnya adalah *function* yang berada di dalam class. Seluruh fungsi dan sifat *function* bisa diterapkan kedalam *method*, seperti argumen/parameter, mengembalikan nilai (dengan *keyword return*), dan lain-lain.

*(Sumber:* [*https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-class-object-property-dan-method/*](https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-class-object-property-dan-method/)*)*

Adapun cara menggunakan *method* yakni tidak langsung memanggil dengan nama fungsi tersebut melainkan harus melalui sebuah objek yang merupakan *instance* (perwujudan) dari class yang mendefinisikan *function/method* tersebut.

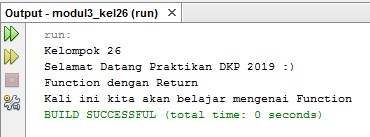
Proses ‘mencetak’ objek dari *class* ini disebut dengan ‘instansiasi’ (atau *instantiation* dalam bahasa inggris). Proses instansiasi dilakukan dengan menggunakan *keyword ‘new’*. Hasil cetakan *class* akan disimpan dalam variabel untuk selanjutnya digunakan dalam proses program.

Perbedaan paling mudah diamati dari function dan method yakni pada cara pemanggilannya. Suatu fungsi dipanggil dengan langsung menyebutkan nama fungsi dan menyertakan argumen (jika diperlukan). Sedangkan suatu method dipanggil melalui instance object yang merupakan perwujudan dari suatu class dengan menggunakan (umumnya) tanda titik (.).

*(Sumber: modul 3,, https://drive.google.com/drive/folders/17L8hQSuVUJ\_c4Lsinio4iFssw7vKxmiO)*

* 1. Hasil Percobaan dan Analisis
     1. ***Function* pada Java**

|  |
| --- |
| package modul3\_kel26;  public class Modul3\_kel26 {  static void cetak(String kelompok){  System.out.println(kelompok);  }  static void cetak1(){  System.out.println("Selamat Datang Praktikan DKP 2019 :)");  }  static String cetak2(String text){  return text;  }  static String cetak3(){  return "Kali ini kita akan belajar mengenai Function";  }  public static void main(String[] args) {  // //penggunaan function non-return type dengan megisikan argument untuk parameter kelompok  // cetak("Kelompok 26");  // //penggunaan function non-return type tanpa parameter  // cetak1();  // //penggunaan function return type dengan mengisikan nilai pada parameter text  // System.out.println(cetak2("Function dengan Return"));  // //penggunaan function return type tanpa parameter  // System.out.println(cetak3());  }  } |



Gambar 4.1 Output Function pada Java

Fungsi *main* akan memanggil fungsi cetak() dengan argumen “Kelompok 26” kemudian kemudian argumen ini akan dikirimkan sebagai parameter *string* ke fungsi cetak() lalu fungsi cetak akan mengeksekusi perintah System.out.println untuk mencetak *output* variabel kelompok yang isinya “Kelompok 26”. Selanjutnya fungsi cetak1() akan dipanggil tanpa parameter dan langsung mengeksekusi perintah didalam fungsi tersebut dan menampilkan “Selamat Datang Praktikan DKP 2019 :)”.

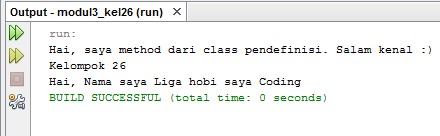
Kemudian fungsi *main* akan menjalankan perintah System.out.println dengan argumen dimana akan memanggil fungsi cetak2() dengan parameter *string* yang nilainya akan dikembalikan ke fungsi cetak2(“Function dengan Return”) dan akan ditampilkan menjadi *output*.

Yang terakhir fungsi *main* akan menjalankan perintah System.out.println dengan argument yang akan memanggil fungsi cetak3() yang isinya akan dikembalikan sebagai *string* dan akan dicetak menjadi *output.*

* + 1. ***Method* pada Java**

|  |
| --- |
| package modul3\_kel26;  public class pendefinisi {  public void greeting(){  System.out.println("Hai, saya method dari class pendefinisi. Salam kenal :)");  }  public void kelompok(String kelompok){  System.out.println(kelompok);  }  public String kenalan (String nama, String hobi){  return "Hai, Nama saya " + nama + " hobi saya " + hobi;  }  } |

|  |
| --- |
| public static void main(String[] args) {  pendefinisi objek = new pendefinisi();    objek.greeting();  objek.kelompok("Kelompok 26");  //String print = objek.kenalan("Liga", "Coding");  System.out.println(objek.kenalan("Liga", "Coding"));  }    } |

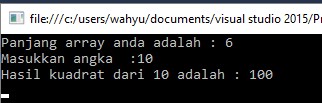


Gambar 4.2 Output Method pada Java

Pada fungsi *main*, pertama akan dibuat objek baru dari *class* pendefinisi menggunakan perintah new, sehingga *method* yang ada di *class* pendefinisi dapat digunakan pada fungsi *main*. Objek.greeting() akan memanggil *method* greeting() pada *class* pendefinisi dan akan menampilkan *output* seperti pada gambar 4.2. Lalu objek.kelompok(“Kelompok 26”) akan dipanggil dengan parameter *string* kemudian variabelnya akan dicetak pada *method* kelompok() yang sebelumnya telah dipanggil. Kemudian fungsi main akan mengeksekusi System.out.println(objek.kenalan("Liga", "Coding")) dengan argument “Liga” dan “Coding” yang akan digunakan sebagai parameter *string* nama dan hobi pada *method* kenalan dan akan mengembalikan nilai yang akan dicetak sebagai *output.*

* + 1. ***Function* pada C#**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Modul3\_Kel26  {  class Program  {  static void cetakPanjangArray(Array list)  {  Console.WriteLine("Panjang array anda adalah : " + list.Length);  }  static int kuadrat(int angka)  {  return angka \* angka;  }  static void Main(string[] args)  {  int[] array = { 0, 1, 2, 3, 4, 5 };  cetakPanjangArray(array);  int angka;  Console.Write("Masukkan angka\t:");  int.TryParse(Console.ReadLine(), out angka);  Console.WriteLine("Hasil kuadrat dari {0} adalah : {1}", angka, kuadrat(angka));  Console.ReadLine();  //pendefinisi objek = new pendefinisi();  //double hasilPangkat = objek.pangkat(4, 2);  //Console.WriteLine("Hasil dari 4 pangkat 2 adalah {0}", hasilPangkat);  //objek.bandingAngka();  //Console.ReadLine();  }  }  } |



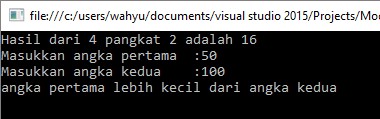
Gambar 4.3 Output function pada C#

Pertama pendeklarasian *array* tipe *integer* dengan nama array = { 0, 1, 2, 3, 4, 5 }. Lalu memanggil fungsi cetakPanjangArray(array) dengan parameter yang akan diproses oleh fungsi untuk mencetak panjang *array* dengan perintah .Length. Lalu deklarasi *integer* dengan nama angka digunakan sebagai variabel untuk mengambil nilai inputan dari *console* yang akan dikonversi menjadi *integer* dengan int.TryParse. Kemudian mencetak *output* dengan memberikan fungsi kuadrat(angka) sebagai argumen, fungsi ini akan mengembalikan nilai angka \* angka yang akan dicetak sebagai hasil dari fungsi kuadrat.

* + 1. ***Method* pada C#**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Modul3\_Kel26  {  class pendefinisi  {  public double pangkat(double angka, double derajat)  {  return Math.Pow(angka, derajat);  }  public void bandingAngka()  {  int angka1 = 0;  int angka2 = 0;  String input;  Console.Write("Masukkan angka pertama\t:");  input = Console.ReadLine();  int.TryParse(input, out angka1);  Console.Write("Masukkan angka kedua\t:");  input = Console.ReadLine();  int.TryParse(input, out angka2);  if (angka1 > angka2)  {  Console.WriteLine("angka pertama lebih besar dari angka kedua");  }  else if (angka1 == angka2)  {  Console.WriteLine("angka pertama sama dengan angka kedua");  }  else  {  Console.WriteLine("angka pertama lebih kecil dari angka kedua");  }  Console.ReadLine();  }  }  } |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Modul3\_Kel26  {  class Program  {  static void cetakPanjangArray(Array list)  {  Console.WriteLine("Panjang array anda adalah : " + list.Length);  }  static int kuadrat(int angka)  {  return angka \* angka;  }  static void Main(string[] args)  {  pendefinisi objek = new pendefinisi();  double hasilPangkat = objek.pangkat(4, 2);  Console.WriteLine("Hasil dari 4 pangkat 2 adalah {0}", hasilPangkat);  objek.bandingAngka();  Console.ReadLine();  }  }  } |



Gambar 4.4 Output method pada C#

Pada fungsi *main*, pertama akan dibuat objek baru dari *class* pendefinisi menggunakan perintah new, sehingga *method* yang ada di *class* pendefinisi dapat digunakan pada fungsi *main*. Lalu deklarasi variabel tipe *double* dengan nama hasilPangkat, *method* objek.pangkat(4, 2) dipanggil dengan menggunakan parameter dan disimpan dalam variabel hasilPangkat. *Method* ini akan menghitung pangkat berdasarkan parameter dan me-*return* nilai sebagai fungsi matematika Math.Pow(angka, derajat) parameter angka sebagai basis dan derajat sebagai eksponennya.

Lalu program akan melanjutkan dengan memanggil *method* objek.bandingAngka, yang akan mengeksekusi perintah untuk menginputkan dua buah bilangan angka1 dan angka2 kemudian membandingkan kedua bilangan tersebut dengan 3 perkondisian, yang pertama jika angka1 > angka2 maka akan mencetak “angka pertama lebih besar dari angka kedua”, yang kedua jika angka1 == angka2 maka akan mencetak “angka pertama sama dengan angka kedua”, dan yang ketiga jika tidak keduanya maka akan mencetak “angka pertama lebih kecil dari angka kedua”.

* + 1. ***Function* pada PHP**

<?php

function hitung ($bil1, $bil2){

return $bil1 + $bil2 \* $bil2;

}

$bil1 = 5;

$bil2 = 8;

$hasil = hitung($bil1, $bil2);

echo "ini dikerjakan oleh Function hitung <br>";

echo "angka yang akan dihitung adalah $bil1 dan $bil2 <br>";

echo "hasil hitung menggunakan Function hitung yaitu : $hasil";

echo "<hr>";

Function hitung2 ($bil1, $bil2){

$hasil = hitung($bil1, $bil2);

echo "angka yang akan dihitung adalah $bil1 dan $bil2 <br>";

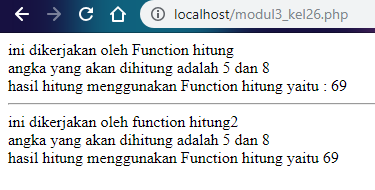
echo "hasil hitung menggunakan Function hitung yaitu $hasil";

}

echo "ini dikerjakan oleh function hitung2 <br>";

hitung2($bil1, $bil2);

?>



Gambar 4.5 Output function pada PHP

Pada program ini pertama dimulai dengan pendeklarasian *function* yang bernama hitung dengan parameter variabel bilangan 1 dan bilangan 2. Dengan menggunakan *return* yang berisikan formula yang nantinya apabila terjadi pemanggilan fungsi hitung maka akan terjadi perhitungan bilangan 1 + bilangan 2 \* bilangan 1.

Selanjutnya yaitu pendeklarasian variabel 1 dan variabel 2 dengan variabel 1 yakni 5 dan variabel 2 yakni 8. Dan diikuti dengan pendeklarasian variabel hasil yang dimana variabel hasil ini berupa fungsi hitung tadi.

Selanjutnya yaitu menampilkan *output* kalimat seperti yang tertera pada gambar 4.5 dengan menggunakan echo dan menyertakan variabel hasil diakhir untuk menampilkan hasil dari bilangan bilangan tadi yang telah di masukkan kedalam formula.

Untuk *function* kedua, tidak jauh berbeda denga *function*  yang pertama, namun pada *function* kedua, langsung disertakan *output* yang nantinya akan muncul.

**4.3.6*. Method* pada PHP**

<?php

class pendefinisi {

public function akar(float $angka) : float {

return sqrt($angka);

}

public function jalaninaja (string $nama, string $merk) {

echo "Hidupkan segera handphone $merk milik $nama"; }

}

$angka = 64;

$objek = new pendefinisi();

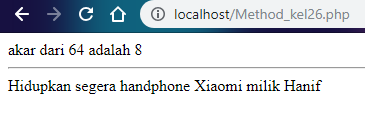
// OOP pada PHP menggunakan -> bukan .

echo "akar dari $angka adalah ".$objek -> akar($angka);

echo "<hr>";

$objek -> jalaninaja('Hanif', 'Xiaomi');

?>

****

Gambar 4.6 Output method pada PHP

Pada program ini menggunakan *method*. Dimana diawal dideklarasikan dulu *function* yang akan digunakan. Di program ini terdapat 2 *function* yaitu akar yang memiliki parameter variabel angka berbentuk *float* dan menggunakan *return* yang apabila dipanggil, nanti akan muncul hasil dari perhitungan akar dari angka yang di deklarasikan oleh variabel angka.

Lalu pada *function* kedua yang diberi nama jalaninaja menggunakan parameter yang berisikan variabel nama dan variabel merk yang berbentuk *string*. Pada *function* ini tidak menggunakan return, namun langsung menggunakan echo untuk menampilkan *output* nya.

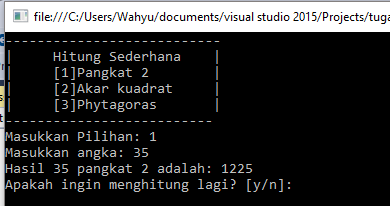
Lalu selanjutnya mendeklarasikan 2 variabel. Yakni variabel angka, dan juga variabel objek. Variabel objek sendiri nantinya akan menjadi pendefinisi baru. Lalu untuk mengubah pendefenisi tersebut menggunakan symbol “->”. dengan membuat pendefenisi baru. Maka nantinya tidak perlu lagi menuliskan nama *functionnya* secara berulang2 karena sudah digantikan oleh pendefinisi yang di program, diberi nama objek.

* 1. Tugas

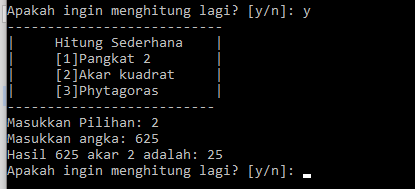
Bahasa C#

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace tugasmodul3  {  class Program  {  static double pangkat(double bil, double pang)  {  return Math.Pow(bil, pang);  }  static void Main(string[] args)  {  Class1 hitung = new Class1();  String pil;  Char lagi;  menu:  Console.WriteLine("---------------------------");  Console.WriteLine("| Hitung Sederhana |");  Console.WriteLine("| [1]Pangkat 2 |");  Console.WriteLine("| [2]Akar kuadrat |");  Console.WriteLine("| [3]Phytagoras |");  Console.WriteLine("--------------------------");  pilihan:  Console.Write("Masukkan Pilihan: ");  pil = Console.ReadLine();  if(pil == "1")  {  int a;  Console.Write("Masukkan angka: ");  int.TryParse(Console.ReadLine(), out a);  Console.WriteLine("Hasil "+a+" pangkat 2 adalah: "+pangkat(a, 2));  } else if (pil == "2")  {  int a;  Console.Write("Masukkan angka: ");  int.TryParse(Console.ReadLine(), out a);  Console.WriteLine("Hasil " + a + " akar 2 adalah: " + hitung.akar(a));  } else if (pil == "3")  {  int a, b;  Console.Write("Masukkan sisi depan: ");  int.TryParse(Console.ReadLine(), out a);  Console.Write("Masukkan sisi samping: ");  int.TryParse(Console.ReadLine(), out b);  Console.WriteLine("Sisi miring = " + hitung.phytaghoras(a, b));  }  else  {  Console.Write("Salah pilih bosque\n");  goto pilihan;  }  Console.Write("Apakah ingin menghitung lagi? [y/n]: ");  char.TryParse(Console.ReadLine(), out lagi);  while (lagi == 'y')  {  goto menu;  }  Console.Write("Bye...");  Console.ReadLine();  }  }  } |

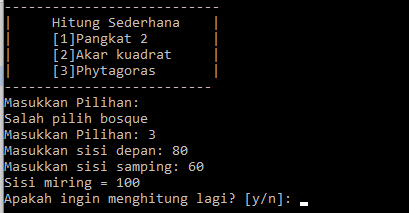
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace tugasmodul3  {  class Class1  {  public double akar(double angka)  {  return Math.Sqrt(angka);  }  public double phytaghoras(double a, double b)  {  double c;  a = Math.Pow(a, 2);  b = Math.Pow(b, 2);  c = Math.Sqrt(a + b);  return c;  }  }  } |



Gambar 4.7 Hitung pangkat



Gambar 4.8 Hitung akar



Gambar 4.9 Hitung sisi miring

Dalam program utama terdapat dua fungsi yaitu fungsi pangkat() yang mengembalikan nilai bertipe *double* dengan menggunakan dua parameter yaitu bil dan pangkat. Kemudian dalam fungsi ini akan me-*return* Math.Pow(bil, pang).

Lalu fungsi main merupakan fungsi *void* yang akan membuat objek bernama hitung dari *class* yang dipanggil. Kemudian menampilkan tampilan menu yang berisi tiga pilihan, ketika salah satu pilihan dimasukkan akan diperiksa dengan perkondisian. Yang pertama if(pil == "1") akan mengeksekusi perintah untuk memasukkan nilai, *input* yang berupa *string* akan dikonversi menjadi *integer* lalu akan dimasukkan kedalam variabel a. Setelah itu akan ditampilkan hasil dengan mengirim nilai variabel a sebagai argument ke fungsi pangkat(). Yang kedua if(pil == "2") akan mengeksekusi perintah untuk memasukkan nilai, *input* yang berupa *string* yang akan dikonversi menjadi *integer* lalu akan dimasukkan kedalam variabel a. Setelah itu akan ditampilkan hasil dengan mengirim nilai variabel a sebagai argument ke *method* hitung.akar(a). Yang ketiga if(pil == "3") akan mengeksekusi perintah untuk memasukkan nilai, *input* yang berupa *string* akan dikonversi menjadi *integer* lalu akan dimasukkan kedalam variabel a, lalu memasukkan nilai, *input* lagi yang berupa *string* akan dikonversi menjadi *integer* lalu akan dimasukkan kedalam variabel b. Setelah itu akan ditampilkan hasil dengan mengirim nilai variabel a dan b sebagai argument ke fungsi hitung.phytaghoras(a, b).

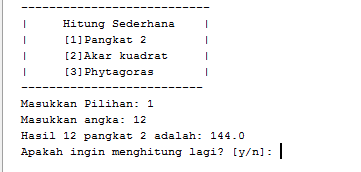
Jika tidak terdapat pilihan yang pas akan menampilkan pesan dan kembali ke pemilihan menu. Lalu bagian akhir terdapat perulangan yang akan dilakukan jika memasukkan huruf yang tepat pada pesan yang meminta untuk menghitung ulang.

Didalam *class* yang dipanggil yaitu *Class1* berisi dua buah *method* yang pertama akar(double angka) dengan parameter angka kemudian mengembalikan nilai dengan fungsi matematika Math.Sqrt(angka) parameter angka adalah nilai yang dikirimkan dari fungsi *main* pada program utama. Yang kedua phytaghoras(double a, double b) memiliki dua parameter a dan b. Didalam *method* ini dideklarasikan variabel c terlebih dahulu, lalu variabel a diset nilainya dengan Math.Pow(a, 2) yang artinya *input* yang dimasukkan kedalam parameter akan dikuadratkan, hal yang sama juga dilakukan pada variabel b yang diberi nilai Math.Pow(b, 2). Lalu variabel c membuat perhitungan Math.Sqrt(a + b) dari variabel a dan b. Terakhir variabel c akan dikembalikan nilainya yang merupakan hasil dari penghitungan sisi miring dalam teorema *phytagoras.*

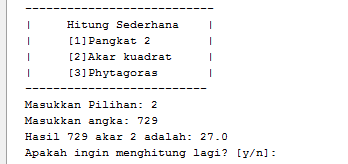
Bahasa Java

|  |
| --- |
| package tugasmodul3;  import java.util.Scanner;  public class Tugasmodul3 {  static double pangkat(double bil, double pang)  {  return Math.pow(bil, pang);  }  public static void main(String[] args) {    NewClass hitung = new NewClass();  Scanner scan = new Scanner(System.in);  int pil;  String lagi;  do  {  System.out.println("\n---------------------------");  System.out.println("| Hitung Sederhana |");  System.out.println("| [1]Pangkat 2 |");  System.out.println("| [2]Akar kuadrat |");  System.out.println("| [3]Phytagoras |");  System.out.println("--------------------------");  System.out.print("Masukkan Pilihan: ");  pil = scan.nextInt();  switch (pil) {  case 1:  {  int a;  System.out.print("Masukkan angka: ");  a = scan.nextInt();  System.out.println("Hasil "+a+" pangkat 2 adalah: "+pangkat(a, 2));  break;  }  case 2:  {  int a;  System.out.print("Masukkan angka: ");  a = scan.nextInt();  System.out.println("Hasil " + a + " akar 2 adalah: " + hitung.akar(a));  break;  }  case 3:  {  int a, b;  System.out.print("Masukkan sisi depan: ");  a = scan.nextInt();  System.out.print("Masukkan sisi samping: ");  b = scan.nextInt();  System.out.println("Sisi miring = " + hitung.phytaghoras(a, b));  break;  }  default:  System.out.print("Salah pilih bosque\n");  break;  }  Scanner sc = new Scanner(System.in);  System.out.print("Apakah ingin menghitung lagi? [y/n]: ");  lagi = sc.nextLine();  } while ("y".equals(lagi) || "Y".equals(lagi));  System.out.print("Bye...");  }    } |

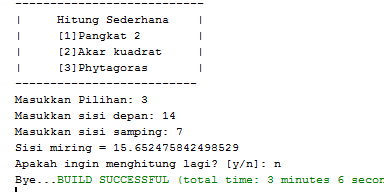
|  |
| --- |
| package tugasmodul3;  public class NewClass {  public double akar(double angka)  {  return Math.sqrt(angka);  }  public double phytaghoras(double a, double b)  {  double c;  a = Math.pow(a, 2);  b = Math.pow(b, 2);  c = Math.sqrt(a + b);  return c;  }  } |



Gambar 4. 10 Hitung pangkat pada java



Gambar 4. 11 Hitung akar pada java



Gambar 4. 12 Hitung sisi miring pada java

Hampir sama seperti program menggunakan bahasa C#, dalam program utama terdapat dua fungsi yaitu fungsi pangkat() yang mengembalikan nilai bertipe *double* dengan menggunakan dua parameter yaitu bil dan pangkat. Kemudian dalam fungsi ini akan me-*return* Math.Pow(bil, pang).

Lalu fungsi main merupakan fungsi *void* yang akan membuat objek bernama hitung dari *class* yang dipanggil. Perulangan *do* digunakan diawal untuk mengeksekusi perintah, kemudian menampilkan tampilan menu yang berisi tiga pilihan, ketika salah satu pilihan dimasukkan akan diperiksa dengan perkondisian *switch case,* case 1 akan mengeksekusi perintah untuk memasukkan nilai, *input* yang berupa *integer* lalu akan dimasukkan kedalam variabel a. Setelah itu akan ditampilkan hasil dengan mengirim nilai variabel a sebagai argument ke fungsi pangkat(). Yang kedua case 2 akan mengeksekusi perintah untuk memasukkan nilai, *input* yang berupa *integer* lalu akan dimasukkan kedalam variabel a. Setelah itu akan ditampilkan hasil dengan mengirim nilai variabel a sebagai argument ke *method* hitung.akar(a). Yang ketiga case 3 akan mengeksekusi perintah untuk memasukkan nilai, *input* yang berupa *integer* lalu akan dimasukkan kedalam variabel a, lalu memasukkan nilai, *input* lagi yang berupa *integer* lalu akan dimasukkan kedalam variabel b. Setelah itu akan ditampilkan hasil dengan mengirim nilai variabel a dan b sebagai argument ke fungsi hitung.phytaghoras(a, b). *Case default* untuk menampilkan pesan jika masukkan tidak ada yang sesuai.

Lalu bagian akhir terdapat pengecekan kondisi dari perulangan *do while* yang akan mengulang perintah jika masukan dari *user* bernilai benar dengan menggunakan *sytax* while ("y".equals(lagi) || "Y".equals(lagi)).

Didalam *class* yang dipanggil yaitu *NewClass* berisi dua buah *method* yang pertama akar(double angka) dengan parameter angka kemudian mengembalikan nilai dengan fungsi matematika Math.sqrt(angka) parameter angka adalah nilai yang dikirimkan dari fungsi *main* pada program utama. Yang kedua phytaghoras(double a, double b) memiliki dua parameter a dan b. Didalam *method* ini dideklarasikan variabel c terlebih dahulu, lalu variabel a diset nilainya dengan Math.pow(a, 2) yang artinya *input* yang dimasukkan kedalam parameter akan dikuadratkan, hal yang sama juga dilakukan pada variabel b yang diberi nilai Math.pow(b, 2). Lalu variabel c membuat perhitungan Math.sqrt(a + b) dari variabel a dan b. Terakhir variabel c akan dikembalikan nilainya yang merupakan hasil dari penghitungan sisi miring dalam teorema *phytagoras.*

4.5 Kesimpulan

1. *Function* berguna untukmenghindari penulisan sintaks yang berulang ulang karena dapat digunakan lebih dari satu kali.
2. *Method* juga merupakan fungsi, namun terletak pada *class* yang berbeda.
3. *Function* dan *method* juga berfungsi untuk mempersingkat *source code* dari program yang kita buat.
4. Pada bahasa Java dan C#, pemanggilan *method* menggunakan simbol “.” Sedangkan pada PHP menggunakan simbol “->”.
5. Penggunaan *method* dan *function* dapat digunakan secara sendiri-sendiri, dan keduanya juga dapat digunakan secara bersamaan dalam sebuah kode program.
6. *Function* atau *method* yang memiliki *modifier public* dapat digunakan di kelas mana saja.
7. Penggunaan *method* adalah ciri dari Pemrograman Berbasis Objek.
8. *Function* dan *method* dapat memiliki parameter lebih dari satu atau tidak memiliki parameter sama sekali.