# **BAB V**

# **FUNCTION DAN METHOD**

## **Tujuan**

Dapat membedakan konsep dasar *function* dan *method*.

Dapat memahami dasar penggunaan *function* dan *method*.

Dapat mengimplementasikan *function* dan *method* dalam bahasa pemrograman Java, C#, PHP, dan Python.

Mampu membedakan *source code* program *function* dalam bahasa pemrograman Java, C#, PHP, dan Python.

Mampu memahami metode pemanggilan *function* dan *method* pada bahasa pemrograman yang berbeda.

Mampu menyelesaikan kasus-kasus sederhana dengan mengaplikasikan *function* maupun *method*.

## **Dasar Teori**

* 1. **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 memudahkan dalam pembuatan program, serta dapat menghemat ukuran program.

Keuntungan memakai fungsi :

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

(sumber :[*http://klikgss.com/fungsi-pada-pemrograman-dasar/*](http://klikgss.com/fungsi-pada-pemrograman-dasar/))

* 1. **Method**

*Method* adalah satu kontainer pada class yang memuat baris-baris kode Dalam bahasa Pemrograman yang lain sering disebut dengan *Procedure* atau *Function*, lengkap dengan signature-nya. *Signature* disini adalah nama dan parameter.*Method* merupakan *encapsulation* kode program.

Keuntungan Menggunakan Method :

1. *Method* membuat program lebih mudah dibaca dan mudah untuk dipelihara.
2. *Method* membuat proses pengembangan dan perawatan ( *maintenance* ) menjadi lebih cepat.
3. *Method* merupakan dasa runtuk melakukan membuat software yang re-usable.
4. *Method* memungkinkan obyek-obyek yang berbeda untuk berkomunikasi dan untuk mendistribusikan beban kerja yang dipikul oleh program.

(sumber : [*https://rininuraini.wordpress.com/2009/01/15/menggunakan-method/*](https://rininuraini.wordpress.com/2009/01/15/menggunakan-method/))

* 1. **Penggunaan Function**

Menggunakan fungsi dalam teori pemograman sering juga disebut dengan istilah ‘memanggil fungsi’ (calling a function). **Fungsi** dipanggil dengan menulis nama dari fungsi tersebut, dan diikuti dengan ***argumen***(jika ada). ***Argumen*** ditulis di dalam tanda kurung, dan jika jumlah ***argumen*** lebih dari satu, maka diantaranya dipisahkan oleh karakter koma.

Setelah memproses nilai inputan, hampir semua fungsi akan memberikan nilai hasil pemrosesan tersebut (walaupun ada fungsi yang tidak memberikan nilai). Cara fungsi memberikan nilainya ini sering disebut dengan ‘mengembalikan nilai’ (*return a value*). Nilai yang dikembalikan oleh sebuah fungsi dapat ditampung ke dalam variabel, atau langsung ditampilkan ke *web browser.*

* 1. **Penggunaan Method**

|  |
| --- |
| *nama class(spasi)objek = new nama class();*  *tipe data objek = objek.method()* |

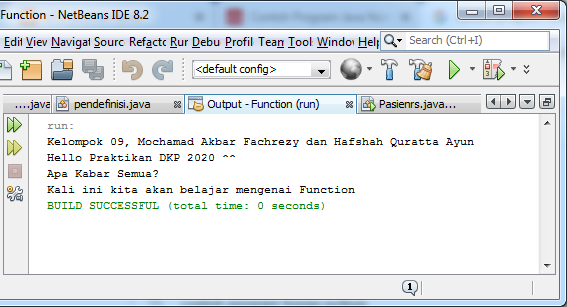
*Main method* merupakan *method* utama yang di jalankan pada saat aplikasi di run. Setiap class dalam sebuah aplikasi boeh memimiliki *main method* tetapi hanya ada satu *main method* yang di run pada saat eksekusi program.

(sumber : [*http://jusitice.blogspot.com/2012/11/penggunaan-method-pada-java.html*](http://jusitice.blogspot.com/2012/11/penggunaan-method-pada-java.html))

## **Hasil Percobaan dan Analisis**

1. **Function pada Java**

|  |
| --- |
| public class Function {    static void cetak(String kelompok){  System.out.println(kelompok);  }  static void cetak1(){  System.out.println("Hello Praktikan DKP 2020 ^^ ");  }  static String cetak2(String text){  return text;  }  static String cetak3(){  return "Kali ini kita akan belajar mengenai Function";  }  public static void main(String[] args) {  cetak("Kelompok 09, Mochamad Akbar Fachrezy dan Hafshah Quratta Ayun");    cetak1();    System.out.println(cetak2("Apa Kabar Semua?"));    System.out.println(cetak3());  }  } |

****

Gambar 5.3.1

Pertama-tama kita akan membuat pernyataan terlebih dahulu sepersi yang ada di *source code*. Seperti, cetak(),cetak1(),cetak2(),cetak3(). Kemudian kita panggil dengan fungsi System.out.println(cetak3()); yang berarti yang dipanggil adalah pernyataan cetak 3, cetak (),cetak1(), cetak2() dapat dipanggil juga seperti itu yang dapat dilihat di source code.

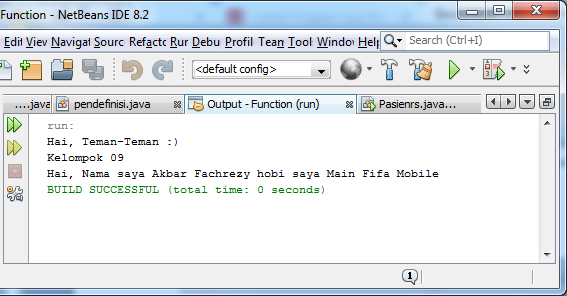
1. **Method pada Java**

Class Pemanggil

|  |
| --- |
| package function;  public class Function {    static void cetak(String kelompok){  System.out.println(kelompok);  }  static void cetak1(){  System.out.println("Hello Praktikan DKP 2020 ^^ ");  }  static String cetak2(String text){  return text;  }  static String cetak3(){  return "Kali ini kita akan belajar mengenai Function";  }  public static void main(String[] args) {  // cetak("Kelompok 09, Mochamad Akbar Fachrezy dan Hafshah Quratta Ayun");    // cetak1();    // System.out.println(cetak2("Apa Kabar Semua?"));    // System.out.println(cetak3());  //  pendefinisi objek = new pendefinisi();    objek.greeting();  objek.kelompok("Kelompok 09");  String print = objek.kenalan("Akbar Fachrezy", "Main Fifa Mobile");  System.out.println(print);  }  } |

Class Pendefinisi

|  |
| --- |
| package function;  public class pendefinisi { public void greeting(){  System.out.println("Hai, Teman-Teman :)");  }  public void kelompok(String kelompok){  System.out.println(kelompok);  }  public String kenalan (String nama, String hobi){  return "Hai, Nama saya " + nama + " hobi saya " + hobi;  }  } |

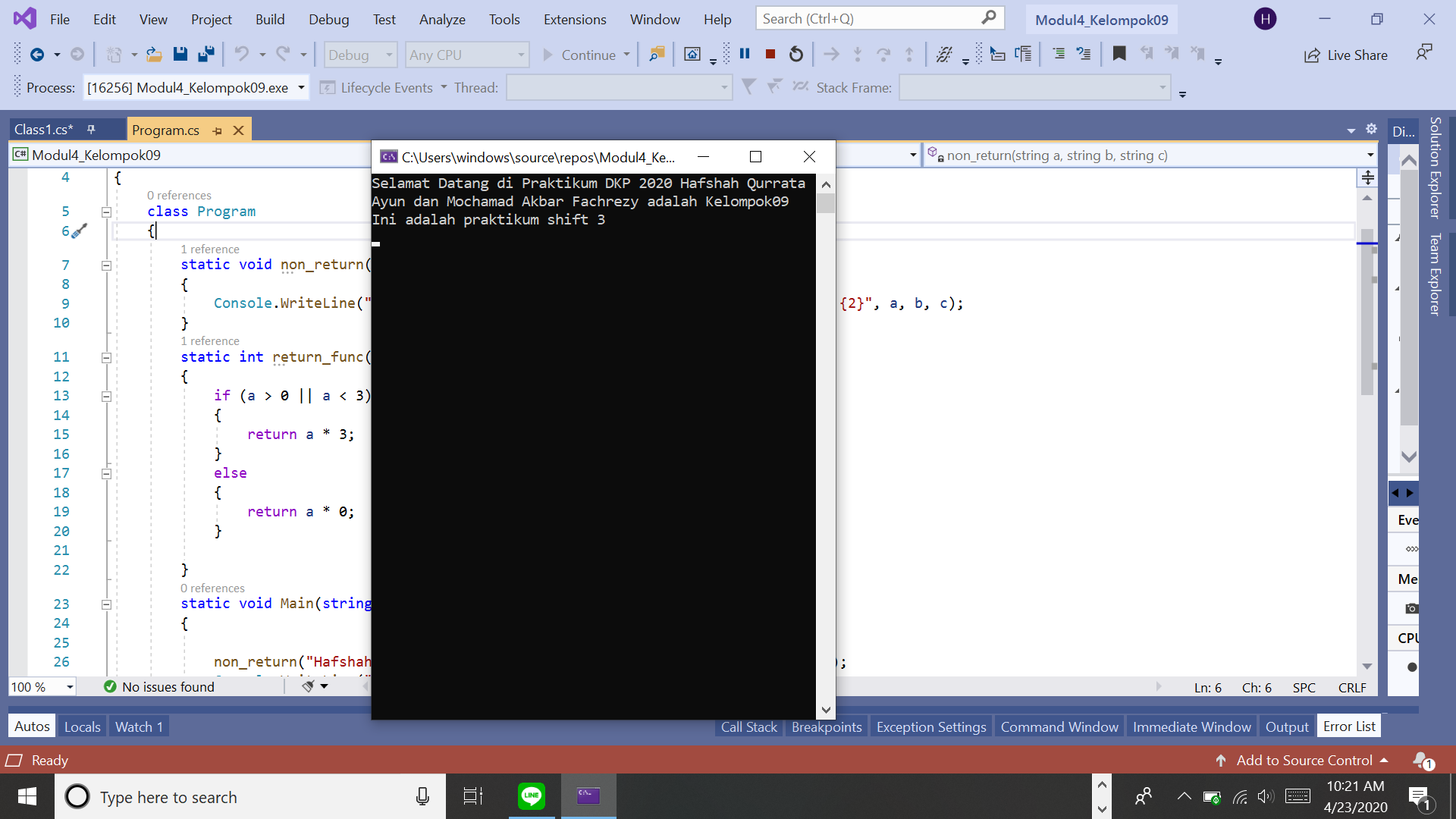


Gambar 5.3.2

Pertama tama kita akan membuat pernyataan terlebih dahulu di kelas tambahan yang dibuat. Dengan contohpublic void greeting() System.out.println("Hai, Teman-Teman :)");Yang lain dapat dilihat dalam *source code* kemudian kita akan panggil pernyataan tersebut pada kelas utamanya dengan terlebih dahulu membuat definisi terhadap yang dipanggil seperti,pendefinisi objek = new pendefinisi(); kemudian kita buat *syntax* untuk memanggil pernyataan yang ada di kelas tambahan dengan objek.greeting();yang berarti kita akan menarik pernyataan yang ada di *greeting*.

1. **Function pada C#**

|  |
| --- |
| using System;  namespace Modul4\_Kelompok09  {  class Program  {  static void non\_return(String a, String b, String c)  {  Console.WriteLine("Selamat Datang di Praktikum DKP 2020 {0} dan {1} adalah {2}", a, b, c);  }  static int return\_func(int a)  {  if (a > 0 || a < 3)  {  return a \* 3;  }  else  {  return a \* 0;  }  }  static void Main(string[] args)  {  non\_return("Hafshah Qurrata Ayun", "Mochamad Akbar Fachrezy", "Kelompok09");  Console.WriteLine("Ini adalah praktikum shift {0}", return\_func(1));  //percobaan2 objek = new percobaan2();  //objek.pembagian(39, 35);  //Console.WriteLine("Hasil pengurangan {0} dengan {1} adalah {2}", 39, 35, perkalian(39, 35));  Console.ReadKey();  }  }  } |

****

Gambar 5.3.3

Pada program, *syntax* Console.WriteLine digunakan untuk mencetak serta memanggil *nonreturn* pada program sehingga dapat menampilkan *output* yang diinginkan yaitu ("Hafshah Qurrata Ayun", "Mochamad Akbar Fachrezy", "Kelompok09"). Sedangkan Console.WriteLine("Ini adalah praktikum shift {0}", return\_func(1)); dignakan untuk mencetak *function* return\_func.

Console.WriteLine("Hasil pembagian {0} dengan {1} adalah {2}", a, b, c); digunakan untuk menampilkan objek.pembagian(39, 35). public int pengurangan(int a, int b) dan int c = a – b. digunakan untuk menampilkan Console.WriteLine("Hasil pengurangan {0} dengan {1} adalah {2}", 39, 35, pengurangan (39, 35). Setelah itu program selesai.

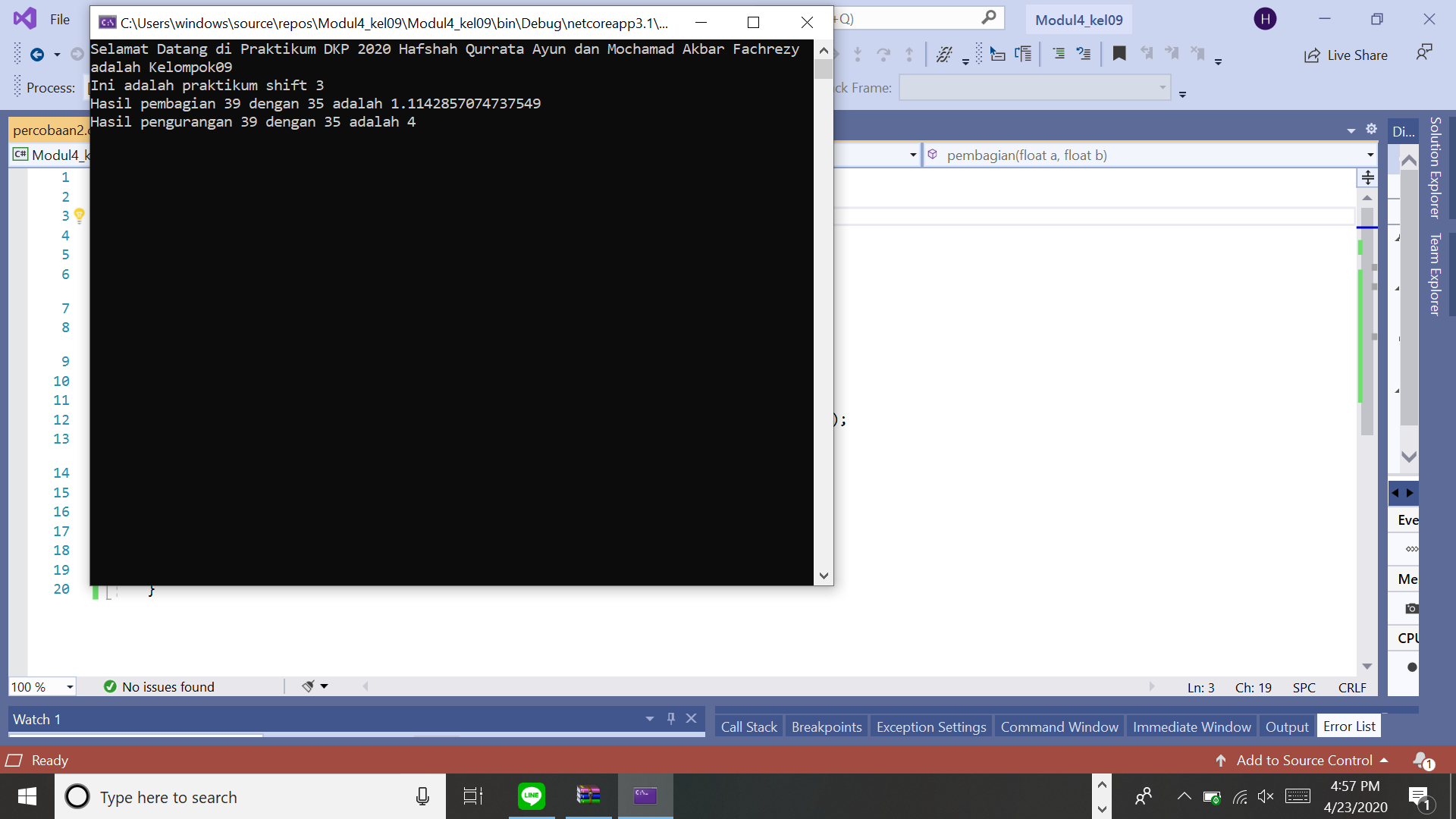
1. **Method pada C#**

Pendefinisi Method

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Text;  namespace Modul4\_kel09  {  class percobaan2  {  public void pembagian(float a, float b)  {  double c = a / b;  Console.WriteLine("Hasil pembagian {0} dengan {1} adalah {2}", a, b, c);  }  public int pengurangan(int a, int b)  {  int c = a - b;  return c;  }  }  } |

Pemanggil Method

|  |
| --- |
| using System;  namespace Modul4\_Kelompok09  {  class Program  {  static void non\_return(String a, String b, String c)  {  Console.WriteLine("Selamat Datang di Praktikum DKP 2020 {0} dan {1} adalah {2}", a, b, c);  }  static int return\_func(int a)  {  if (a > 0 || a < 3)  {  return a \* 3;  }  else  {  return a \* 0;  }  }  static void Main(string[] args)  {  non\_return("Hafshah Qurrata Ayun", "Mochamad Akbar Fachrezy", "Kelompok09");  Console.WriteLine("Ini adalah praktikum shift {0}", return\_func(1));  percobaan2 objek = new percobaan2();  objek.pembagian(39, 35);  Console.WriteLine("Hasil pengurangan {0} dengan {1} adalah {2}", 39, 35, objek.pengurangan(39, 35));  Console.ReadKey();  }  }  } |

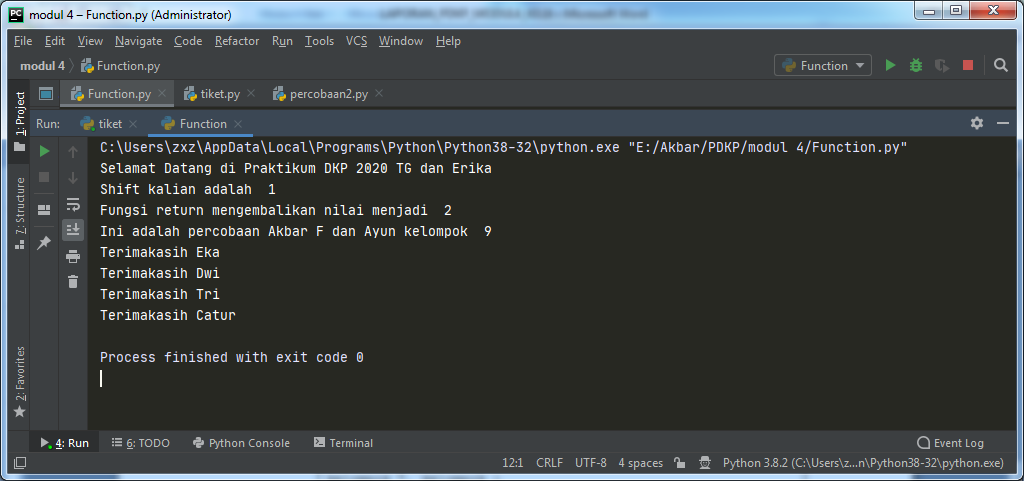
****

Gambar 5.3.4

Program diawali dengan pendeklarasian *variable* angka, nilai menggunakan tipe data *double*.*Method* pada *class* pendefinisi,public void pembagian(float a, float b) dan double c = a / b . Console.WriteLine("Hasil pembagian {0} dengan {1} adalah {2}", a, b, c); digunakan untuk menampilkanobjek.pembagian(39, 35). public int pengurangan(int a, int b) dan int c = a – b. digunakan untuk menampilkanConsole.WriteLine("Hasil pengurangan {0} dengan {1} adalah {2}", 39, 35, objek.pengurangan (39, 35).Kemudian pada *class* pemanggil diberikan masukan Console.WriteLine("Hasil pembagian {0} dengan {1} adalah {2}", a, b, c);. Dan jika program dijalankan maka akan menampilkan *output* sebagaimana gambar diatas.

1. **Function pada Python**

|  |
| --- |
| #Function dengan non return type def non\_return\_func(praktikan1, praktikan2):  print(f"Selamat Datang di Praktikum DKP 2020 {praktikan1} dan {praktikan2}")  #Function dengan return type def return\_func(shift):  print(f"Shift kalian adalah ", shift)  if (shift == 1) or (shift == 2) :  return print(f"Fungsi return mengembalikan nilai menjadi ", shift \* 2)  else:  return print("Tidak ada shift tersebut")  #Function dengan Arbitrary Type def arbitrary\_func(\*penutup):  for nama in penutup:  print("Terimakasih", nama)  #Anonymous Function anonim\_func = lambda praktikan1, praktikan2, kelompok: print(f"Ini adalah percobaan {praktikan1} dan {praktikan2} kelompok ", kelompok )  #Pemanggilan Fungsi non\_return\_func("TG", "Erika") return\_func(1) anonim\_func("Akbar F", "Ayun", 9) arbitrary\_func("Eka", "Dwi", "Tri", "Catur") |

****

Gambar 5.3.5

def non\_return\_func(praktikan1, praktikan2) kita membuat pernyataan penggunaan function dimana pernyataan tersebut dengan *function* non\_return\_func("TG", "Erika") kata TG dan Erika merupakan string pengganti dari praktikan 1 dan praktikan 2. def return\_func(shift)merupakan *function* tipe *return* dimana setelah dijalankan akan kembali lagi. def arbitrary\_func(\*penutup):,Merupakan jenis *function* yang dapat digunakan jika kita tidak mengetahui parameternya.  *anonymous function,* merupakan fungsi yang dapat dideklarasikan tanpa memberikan nama fungsi. Fungsi ini memanfaatkan *lambda* yang ada pada python.

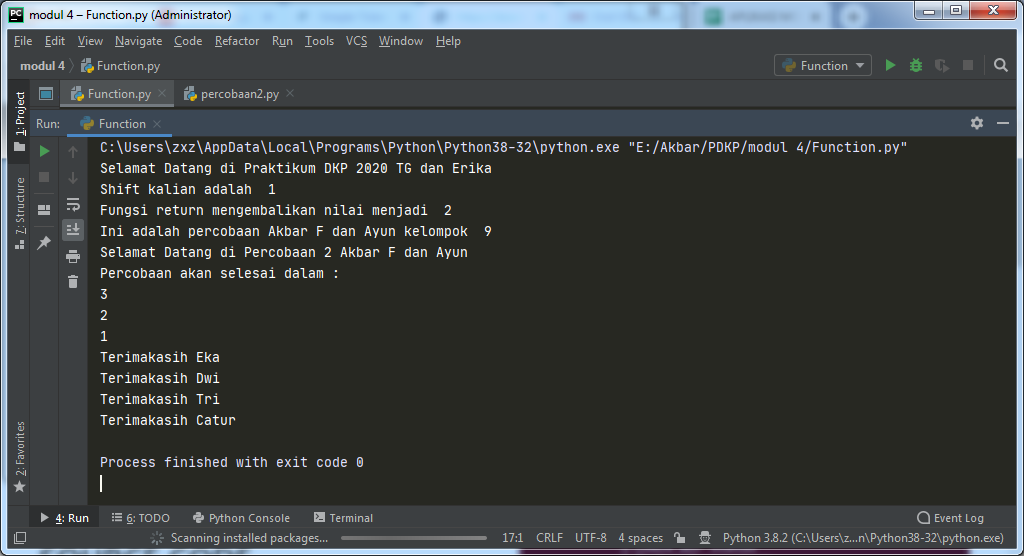
1. **Method pada Python**

Class Pendefinisi

|  |
| --- |
| lass contoh\_method:    def \_\_init\_\_ (self, praktikan1, praktikan2):  self.praktikan1 = praktikan1  self.praktikan2 = praktikan2    def mulai(self):  print(f"Selamat Datang di Percobaan 2 {self.praktikan1} dan {self.praktikan2}")     def selesai(self, waktu):  print("Percobaan akan selesai dalam :")  while waktu > 0:  print(waktu)  waktu -= 1 |

Class Pemanggil

|  |
| --- |
| import percobaan2  #Function dengan non return type def non\_return\_func(praktikan1, praktikan2):  print(f"Selamat Datang di Praktikum DKP 2020 {praktikan1} dan {praktikan2}")  #Function dengan return type def return\_func(shift):  print(f"Shift kalian adalah ", shift)  if (shift == 1) or (shift == 2) :  return print(f"Fungsi return mengembalikan nilai menjadi ", shift \* 2)  else:  return print("Tidak ada shift tersebut")  #Function dengan Arbitrary Type def arbitrary\_func(\*penutup):  for nama in penutup:  print("Terimakasih", nama)  #Anonymous Function anonim\_func = lambda praktikan1, praktikan2, kelompok: print(f"Ini adalah percobaan {praktikan1} dan {praktikan2} kelompok ", kelompok )  #Pemanggilan Fungsi non\_return\_func("TG", "Erika") return\_func(1) anonim\_func("Akbar F", "Ayun", 9)  p1 = percobaan2.contoh\_method("Akbar F","Ayun")  p1.mulai()  p1.selesai(3)   arbitrary\_func("Eka", "Dwi", "Tri", "Catur") |

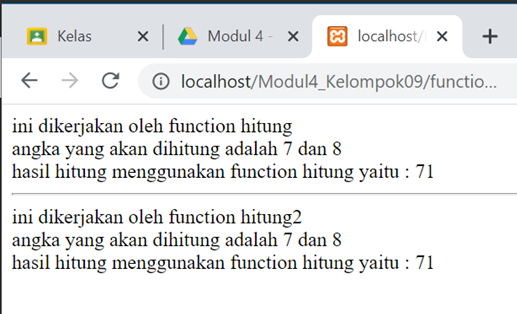


Gambar 5.3.6

*import* merupakan *syntax* yang digunakan untuk memdatangkan *coding* dari file lain yang dalam satu *packet*.def non\_return\_func(praktikan1, praktikan2) kita membuat pernyataan penggunaan *function* dimana pernyataan tersebut dengan function non\_return\_func ("TG", "Erika") kata TG dan Erika merupakan *string* pengganti dari praktikan 1 dan praktikan 2. def return\_func(shift)merupakan *function* tipe return dimana setelah dijalankan akan kembali lagi. def arbitrary\_func(\*penutup):,Merupakan jenis function yang dapat digunakan jika kita tidak mengetahui parameternya.  *anonymous function,* merupakan fungsi yang dapat dideklarasikan tanpa memberikan nama fungsi. Fungsi ini memanfaatkan *lambda* yang ada pada python. p1 = percobaan2.contoh\_method("Akbar F","Ayun") disini kita membuat pernyataan dimana kita akan menjalankan dari si *import*. p1.mulai() syntax yang digunakan untuk menjalankan program.

1. **Function pada PHP**

|  |
| --- |
| <?php  //ini return type  function hitung ($bil1, $bil2) {  return $bil1 + $bil2 \* $bil2;  }  $bil1 = 7;  $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 : $hasil";  echo "<hr>";  //ini non return type  function hitung2 ($bil1, $bil2) {  $hasil = hitung($bil1, $bil2);  echo "angka yang akan dihitung adalah $bil1 dan $bil2 <br>";  echo "hasil hitung menggunakan function hitung yaitu : $hasil";  } |

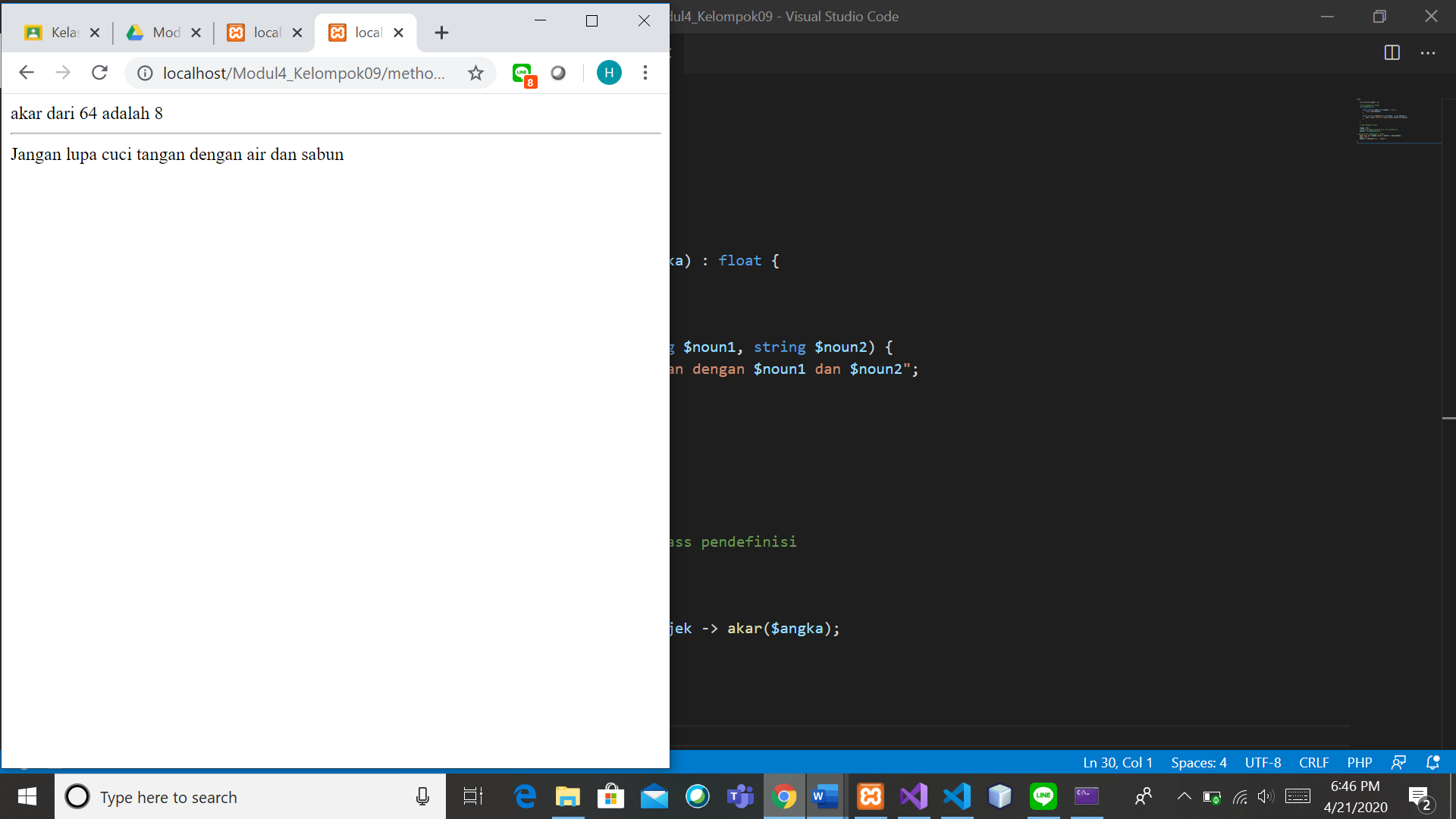
****

Gambar 5.3.8

Pada program diatas menggunakan bahasa PHP. Program diawali dengan pendeklarasian *function* dengan tipe data *integer.* Kemudian masuk pada rumus function hitung ($bil1, $bil2) { menggunakan *return type* return $bil1 + $bil2 \* $bil2;. Yang kedua menggunakan *non return type* function hitung2 ($bil1, $bil2) { $hasil = hitung($bil1, $bil2);. Kemudian akan muncul keluaraan seperti gambar *output* diatas.

1. **Method pada PHP**

|  |  |
| --- | --- |
| <?php  declare(strict\_types = 1);  //class pendefinisi method  class pendefinisi {  public function akar(float $angka) : float {  return sqrt($angka);  }  public function staysafe (string $noun1, string $noun2) {  echo "Jangan lupa cuci tangan dengan $noun1 dan $noun2";  }  }  //code pemanggil method  $angka = 64;  //$objek merupakan instance dari class pendefinisi  $objek = new pendefinisi();    // OOP pada PHP menggunakan -> bukan .  echo "akar dari $angka adalah ".$objek -> akar($angka);  echo "<hr>";  $objek -> staysafe('air', 'sabun');    ?> |  |

****

Gambar 5.3.8

## **Tugas**

|  |
| --- |
| import time;  waktu = time.asctime(time.localtime(time.time()))  print ("Waktu sekarang :", waktu)  print("Nama : Mochamad Akbar Fachrezy") print("NIM : 21120119120029") print("Nama : Hafshah Quratta Ayun") print("NIM : 21120119120029") print("Kelopok 9") print("Shift : 1") print("=== Program Pengecekan Pengiriman Barang ===") def fungsi():  print("Selamat Datang di Express Wonder") fungsi()  print("Masukkan Nama Anda : ") nama = input ("Nama : ")  print("Masukkan Password Anda : ") Kunci = input ("Password : ")  if Kunci == "12345678" :    while(True):  print(" ===============SELAMAT DATANG DI APLIKASI PENGIRIMAN BAWANG===============")  print("1. MENGIRIM BARANG")  print("2. CEK PENGIRIMAN")  print("0. Keluar")  print("===========================================================")  pil = int(input("Masukkan pilihan anda : "))  if pil == 1:  nama1 = input ("Nama Pengirim : ")  alamat1 = input ("Alamat Pengirim : ")  nama2 = input ("Nama Penerima : ")  alamat2 = input ("Alamat Penerima : ")  id\_barang = input ("Id Barang : ")  print (" ========================== Cetak ============================")  print ("Nama Pengirim : ",nama1,"\n","Alamat Pengirim : ",alamat1,"\n","Nama Penerima : ",nama2,"\n","Alamat Penerima : ",alamat2,"\n"''"ID Barang : ",id\_barang)  if pil == 2:  print("masukkan id yang diterima")  id = input ("ID Barang : ")  if id == "12345678":  while (True):  print("Nama Pengirim : Akbar")  print("Alamat : Jalan CUmi Raya")  print("Barang Anda : HP")  print("Berat Barang : 500 gram")  print("Nama Penerima : Taufik")  print("Alamat Penerima :Jalan Madu Manis")  break   print("Terima Kasih Telah Menggunakan Aplikasi Kami")  pilihan = input("Apakah Anda ingin mencoba lagi ? (Y/N) : ")  if pilihan == "Y": continue  if pilihan == "N": break if Kunci != "12345678":  print("Password Salah") |

## 

## Gambar 5.4.1

## 

## Gambar 5.4.2

Pada tugas kali ini kami membuat program prngiriman barang**,** import time merupakan fungsi untuk mengambil waktu. Kemudian waktu = time.asctime(time.localtime(time.time())) fungsi tersebut digunakan untuk mengambil waktu local dan kemudian diprint menggunakan print ("Waktu sekarang :", waktu) sehingga dapat dilihat pad ass keterangan waktu tersebut. def fungsi():kita membuat fungsi yang isinya ("Selamat Datang di Express Wonder") kemudian kita panggil penyataan tersebut menggunakan *syntax* fungsi().

Kemudian fungsi input digunakan untuk melakukan penginputan apa yang diketik di keyboeard. Kemudian ada pengkondisian dima kunci yang harus dimasukkan adalah 12345678 apabila tidak tepat maka program akan berhenti. Apabila benar akan berlanjut ke pengkondisian dimana pilihan yang akan dipilih yaitu mengirim barang dan cek barang. Kemudian pilih salh satu, jalankan program sesuai source codenya. Kemudian setelah selesai kita ditanya ingin mengulang kembali atau tidak bila ketik “Y” akan mengulangi dan “N” program akan berhenti.

## Link Github (<https://github.com/akbarfach/Modul4_Kel9/>)

## **Kesimpulan**

1. *Function* merupakan suatu bagian dari program yang digunakan untuk mengerjakan suatu tugas tertentu yang menghasilkan suatu nilai untuk dikembalikan ke program pemanggil dan letaknya dipisahkan dari bagian program yang menggunakannya.
2. Perbedaan *function* dan *method*terdapat 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 (.).
3. *Method* menggunakan tipe data void sedangkan *function* dapat menggunakan banyak macam tipe data.
4. Cara pemanggilan *method* berbeda-beda dalam berbagai bahasa pemrograman. Cara pemanggilan *method* pada Java dan C# dapat menggunakan(.), sedangkan pada PHP menggunakan (->).
5. Instansiasi adalah proses pembuatan objek dari kelas yang sudah ada.