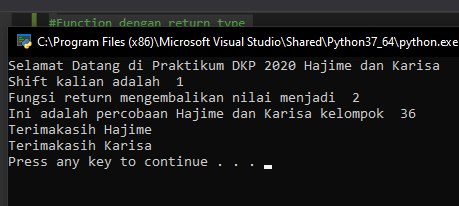
**BAB V**

**FUNCTION DAN METHOD**

**5.3.5. 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("Hajime", "Karisa")  return\_func(1)  anonim\_func("Hajime", "Karisa", 36)  arbitrary\_func("Hajime", "Karisa") |

****

Analisis:

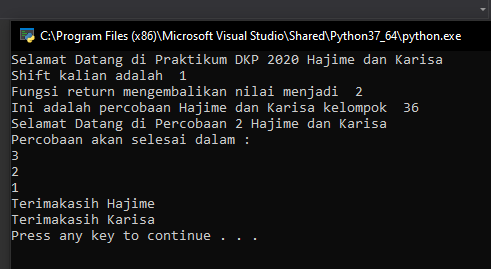
**5.3.6. Method Pada Python**

**Class Method**

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

**File Pertama dengan beberapa tambahan**

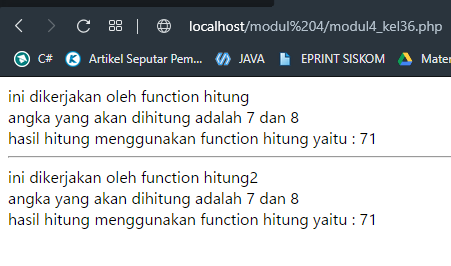
|  |
| --- |
| import percobaan2  #Function dengan non return type  def non\_return\_func(praktikan1, praktikan2):  print(f"Selamat Datang di Praktikum DKP 2020 {praktikan1} dan {praktikan2}")  #Function dengan return type  def return\_func(shift):  print(f"Shift kalian adalah ", shift)  if (shift == 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("Hajime", "Karisa")  return\_func(1)  anonim\_func("Hajime", "Karisa", 36)  #pembuatan object  p1 = percobaan2.contoh\_method("Hajime", "Karisa")  #pemanggilan method dengan self parameter  p1.mulai()  #pemanggilan method dengan parameter  p1.selesai(3)  arbitrary\_func("Hajime", "Karisa") |



Analisis:

**5.3.7. Function Pada PHP**

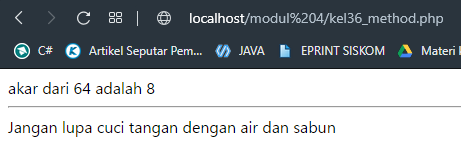
|  |
| --- |
| <?php  //ini return type  function hitung ($bil1, $bil2) {  return $bil1 + $bil2 \* $bil2;  }  $bil1 = 7;  $bil2 = 8;  $hasil = hitung($bil1, $bil2);  echo "ini dikerjakan oleh function hitung <br>";  echo "angka yang akan dihitung adalah $bil1 dan $bil2 <br>";  echo "hasil hitung menggunakan 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 : $hasil";  }  echo "ini dikerjakan oleh function hitung2 <br>";  hitung2($bil1, $bil2);  ?> |



Analisis:

**5.3.8. 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');    ?> |



Analisis:

**5.4. Tugas**

**5.5. Kesimpulan**