**BAB V**

**FUNCTION DAN METHOD**

# **2.1. Tujuan**

Praktikum kali ini membahas tentang *Function* dan *Method* . Tujuan diadakannya praktikum ini yatu:

1. Praktikan dapat membedakan konsep dasar function dan method
2. Praktikan dapat memahami dasar penggunaan function dan method
3. Praktikan dapat mengimplementasikan function dan method dalam bahasa pemrograman C#, Python, Java dan PHP

# **2.2. Dasar Teori**

**2.2.1. Function**

Fungsinya seperti resep. Mereka dapat menjalankan serangkaian instruksi pada data atau variabel dan mengembalikan hasilnya. Keindahan fungsi adalah bahwa mereka dapat didaur ulang. Artinya, fungsinya dapat digunakan berulang kali tanpa harus menulis kode yang sama lagi.

**2.2.2. Method**

Metode, seperti fungsi, adalah sekumpulan instruksi yang melakukan tugas. Perbedaannya adalah bahwa suatu metode dikaitkan dengan suatu objek, sedangkan suatu fungsi tidak. Mari kita menjelajahi beberapa metode bawaan JavaScript.

**2.2.3. Pengunaan Function**

Pengkondisian *if - else if - else* adalah pengkondisian yang memiliki 3 atau lebih kondisi. Apabila kondisi satu terpenuhi, program dalam tubuh *if* kondisi satu akan berlangsung. Jika tidak maka akan dilanjutkan ke kondisi kedua. Jika terpenuhi maka program di dalam tubuh kondisi kedua akan berlangsung. Jika tidak maka else digunakan untuk menangani semua kondisi yang telah dituliskan.

**2.2.4. Penggunaan Method**

Switch adalah operator kondisi yang memiliki fungsi yang sama dengan *if elseif*. Operator ini juga dipakai untuk kondisi percabangan lebih dari satu. Bedanya if elseif mengerjakan tugasya dengan cara memeriksa statement kondisi yang ada satu persatu, maka switch tidak. *Switch* akan memeriksa nilai statement kondisi bersamaan lalu menjalankan yag nilai kondisinya sesuai.

<https://www.codecademy.com/articles/fwd-js-methods-functions>

# **2.3. Hasil dan Analisa Percobaan**

Sub bab ini memaparkan sebuah program sederhana sebagai percobaan yang mana dibuat dengan *source code* kemudian di *run* untuk mengetahui hasilnya. Kemudian program tersebut dianalisis untuk mengetahui *detail* dari setiap *code*  yang digunakan,

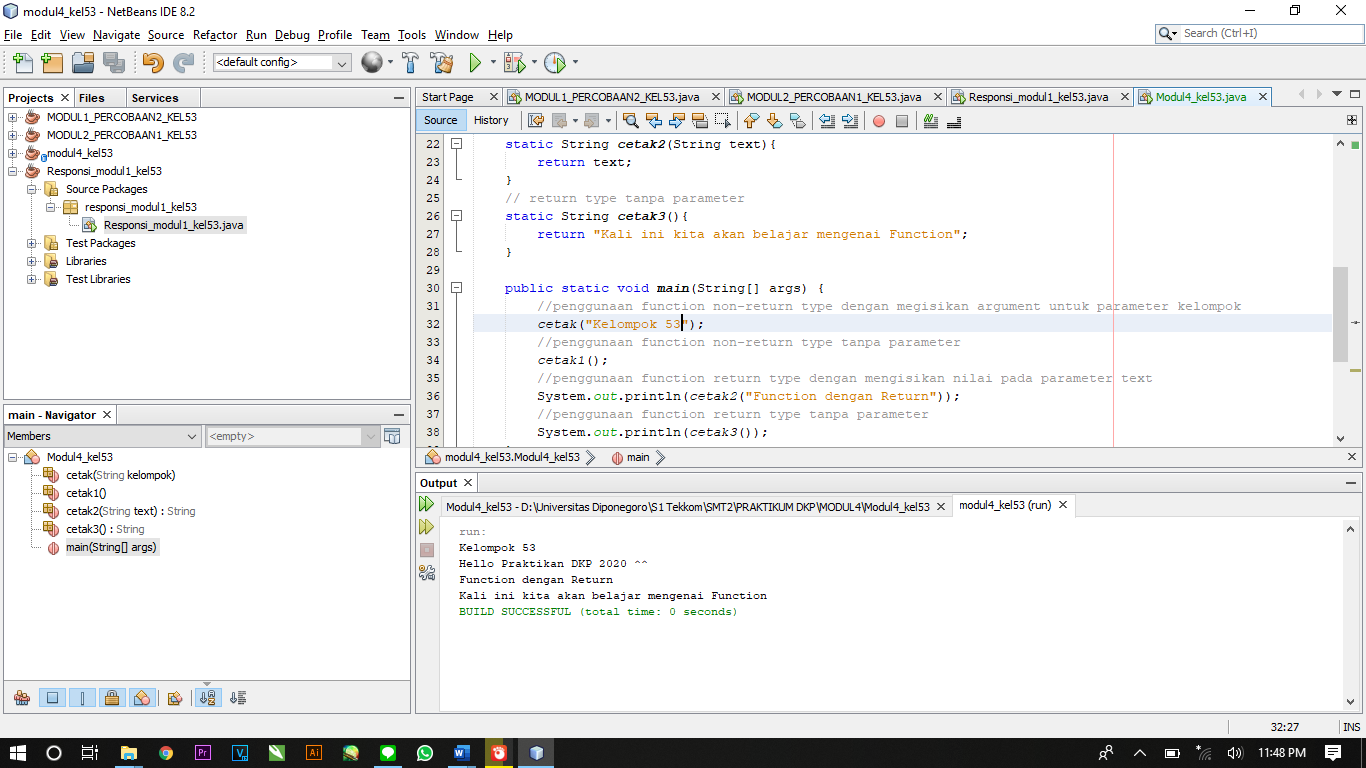
**2.3.1. Function dalam Java**

Dibawah ini

*Source code:*

|  |
| --- |
| public class Modul4\_kelxx {  // non-return type dengan parameter  static void cetak(String kelompok){  System.out.println(kelompok);  }  // non-return type tanpa parameter  static void cetak1(){  System.out.println("Hello Praktikan DKP 2020 ^^ ");  }  // return type dengan parameter  static String cetak2(String text){  return text;  }  // return type tanpa parameter  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 53");  //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());  }  } |

Hasil (*Output*):



Gambar 2.3.1. Function pada Java

Analisis:

Sebelum main program kita deklarasikan beberapa *function*. Yang pertama ada *function* non-return type dengan parameter. *Function non-return type* ini bernama cetak karena sudah dideklarasikan dengan tipe data void dan parameternya adalah variabel bernama kelompok dengan tipe data String pada *function* cetak ini kita isikan *source code* dengan System.out.println(kelompok); artinya *source code* itu akan mencetak *variabel* kelompok itu sendiri. Yang kedua ada *function* non-return type tanpa parameter yang nama *function*nya adalah cetak1() jadi yang di dalam kurung itu kosong dan mengisi *function* itu dengan *source code* System.out.println("Hello Praktikan DKP 2020 ^^ ");dengan tipe data *string.* Yang ketiga ada *function* return type dengan parameteryang nama *function*nya cetak2 dan ada parameternya String text dengan tipe data *String*. Lalu ada return text artinya ketika dieksekusi akan mengembalikan nilai dari variabel *text*. Dan yang terakhir ada *function* return type tanpa parameter yang nama *function*nya adalah cetak3() karena tidak ada parameternya akan mengembalikan nilai yaitu "Kali ini kita akan belajar mengenai Function" dengan tipe data *String*. Untuk bagian *main* programnya langsung memanggil dari *function* itu sendiri. Yang pertama memanggil *function* cetak("Kelompok 53")dengan parameter Kelompok 53. Yang kedua memanggil *function* cetak1()yang hasilnya akan langsung jadi Hello Praktikan DKP 2020 ^^ karena sudah langsung ada perintah untuk mencetak pada *function* awal tadi dan karena tanpa parameter. Yang ketiga *function* dengan *return* karena cetak2(String text) disini variabel *text* diisi dengan parameter Function dengan Return jadi ketika dipanggil *function* tadi dengan perintah System.out.println(cetak2 maka akan menghasilkan Function dengan Return. Yang terakhir (cetak3()) dengan perintah System.out.println akan mencetak parameter yang sudah diisikan pada *function* sebelumnya yaitu Kali ini kita akan belajar mengenai Function

**2.3.2. Method pada Java**

Dibawah ini *class* pendefinisi *method*

*Source code:*

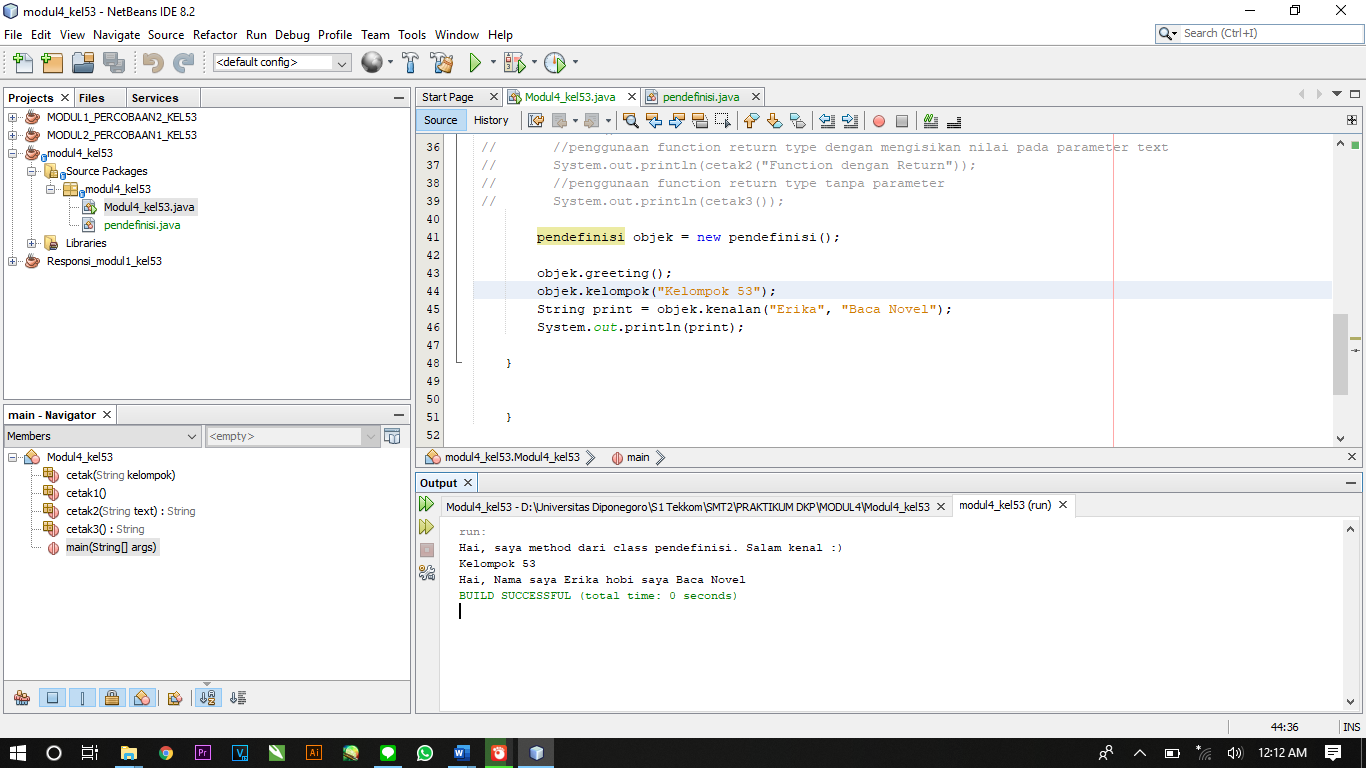
|  |
| --- |
| public class pendefinisi {  public void greeting(){  System.out.println("Hai, saya method dari class pendefinisi. Salam kenal :)");  }  public void kelompok(String kelompok){  System.out.println(kelompok);  }  public String kenalan (String nama, String hobi){  return "Hai, Nama saya "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

Dibawah ini *class* pemanggil *method*

*Source code:*

|  |
| --- |
| public static void main(String[] args) {  pendefinisi objek = new pendefinisi();    objek.greeting();  objek.kelompok("Kelompok 53");  String print = objek.kenalan("Erika", "Baca Novel");  System.out.println(print);  } |

Hasil (*Output*):



Gambar 2.3.2. Method pada Java

Analisis:

Pada *class* pendefinsi ini deklarasikan beberapa *method. Method* pertama dinamakan greeting yang isinya System.out.println("Hai, saya method dari class pendefinisi. Salam kenal :)");. *Method* kedua dinamakan kelompok dengan parameter String kelompok akan mencetak variabel kelompok dengan tipe data *string*. Lalu ada *method* dengan nama kenalan parameternya ada 2 yaitu String nama, String hobi. return "Hai, Nama saya " + nama + " hobi saya " + hobi; jadi ketika dipanggil akan menggambil parameter nama dan hobi. Pada *class* pemanggil pada *main class* untuk memanggil *method* tadi perlu instansiasi objek yang berisi *command* pendefinisi objek = new pendefinisi();*.* Nama *class*nya pendefinisi dan objek adalah nama objeknya lalu = new pendefinisi(); adalah nama *class* tadi yang sudah dibuat. Untuk pemanggilan *method*nya caranya adalah nama objek (.) nama *method* yang sudah dibuat. Yang pertama objek.greeting(); berarti memanggil *method* greeting yang *source code* berisi perintah untuk mencetak sehingga *output*nya Hai, saya method dari class pendefinisi. Salam kenal. Yang kedua objek.kelompok("Kelompok 53"); berarti memanggil *method* kelompokdengan mengisi *String* Kelompok 53 pada variabel (kelompok) lalu akan dicetak Kelompok 53 karena pada *method* tadi sudah ada perintah untuk mencetaknya dengan perintah System.out.println .Yang ketiga String print = objek.kenalan("Erika", "Baca Novel"); berarti memanggil *method* kenalan dengan mengisikan parameter pertamanya Erika dan parameter keduanya Baca Novel. Yang sudah dideklarasikan dengan tipe data *string* lalu akan mengembalikan nilai Hai, Nama saya " + nama + " hobi saya " + hobi dimana + nama akan diganti dengan Erika dan + hobi akan digantikan dengan Baca Novel.

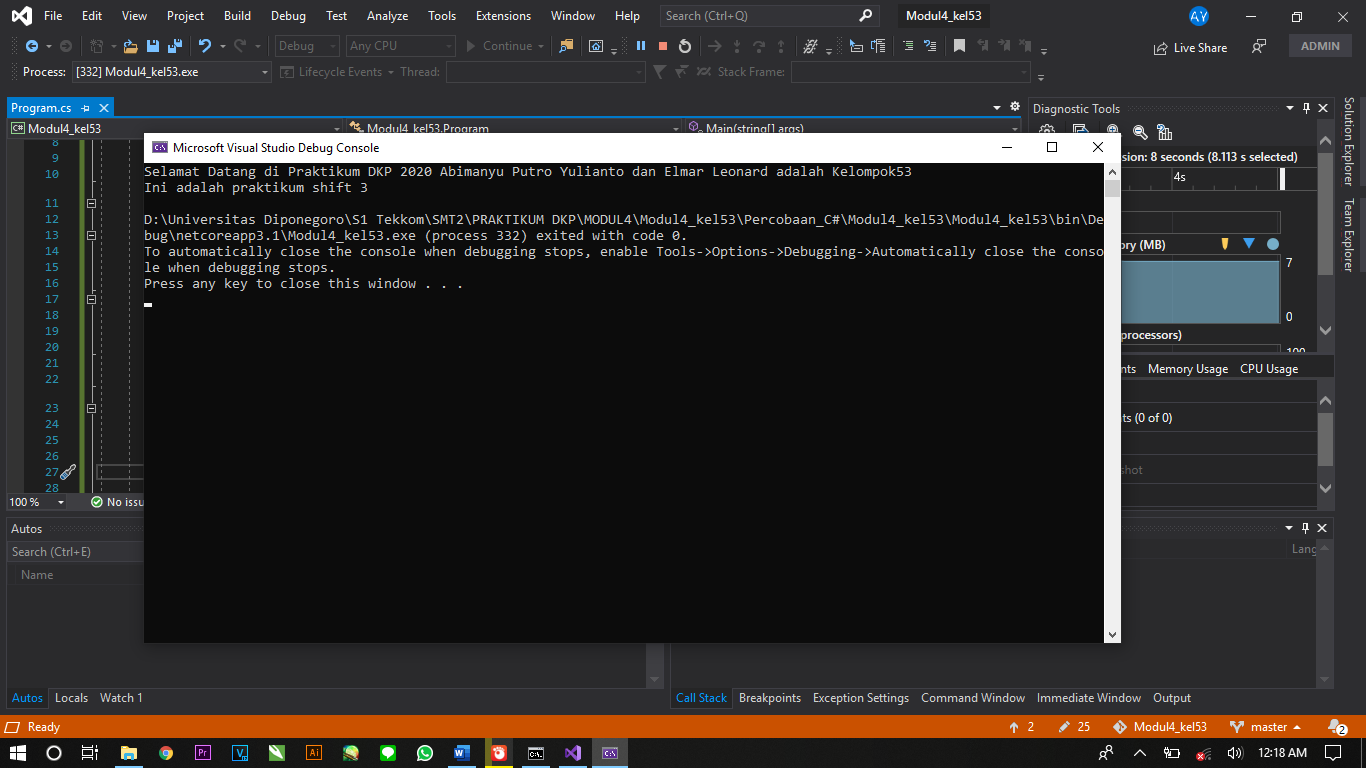
**2.3.3. Function pada C#**

Dibawah ini

*Source code:*

|  |
| --- |
| 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("Abimanyu Putro Yulianto", "Elmar Leonard", "Kelompok53");  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();  }  } |

Hasil (*Output*):



Gambar 2.3.3. Function pada C#

Analisis:

Pertama ada *function* *non-return ­*dengan parameter kita deklarasikan 3 variabel yaitu String a, String b, String c ketika *function* ini dipanggil maka akan menajalan perintah Console.WriteLine("Selamat Datang di Praktikum DKP 2020 {0} dan {1} adalah {2}", a, b, c); artinya parameter {0} dan {1} adalah {2} digunakan untuk memasukan variabel {0} misal akan diganti menjadi *string* a dan selanjutnya sesuai dengan urutannya. Parameter a, b, c ini akan diisi di *class main* ketika memanggil *function* tersebut. Untuk memanggilnya pertama tulis *function*nya non\_return lalu isi parameter pertama, kedua, dan ketiga dengan ("Abimanyu Putro Yulianto", "Elmar Leonard", "Kelompok53"); Ketika *function* ini dipanggil maka nilai ("Abimanyu Putro Yulianto", "Elmar Leonard", "Kelompok53" akan dilempar ke parameter String a, String b, String c dan disesuaikan lalu akan mengeksekusi perintah didalam *function* tersebut yaitu Console.WriteLine jadi hasilnya akan menjadi Selamat Datang di Praktikum DKP 2020 Abimanyu Putro Yulianto dan Elmar Leonard adalah kelompok53.

Untuk *return* berfungsi untuk mengembalikan nilai maka akan mengembalikan nilai(int a) ke pemanggilan fungsi itu yaitu return\_func(1) dengan nilai parameter 1 dengan tipe data *integer*. Karena ada *if else statement* akan memproses angka 1 pada if (a > 0 && a < 3) akan mengembalikan nilai return a \* 3 karena tertulis angka 1 maka yang akan dieksekusi return a \* 3 maka nilai 1 ini akan dikalikan dengan 3 lalu hasil perkalian tersebut 3 akan dikembalikan ke return\_func(1) jadi ketika di *run* akan menghasilkan Ini adalah praktikum shift 3 karena ada perintah Console.WriteLine s

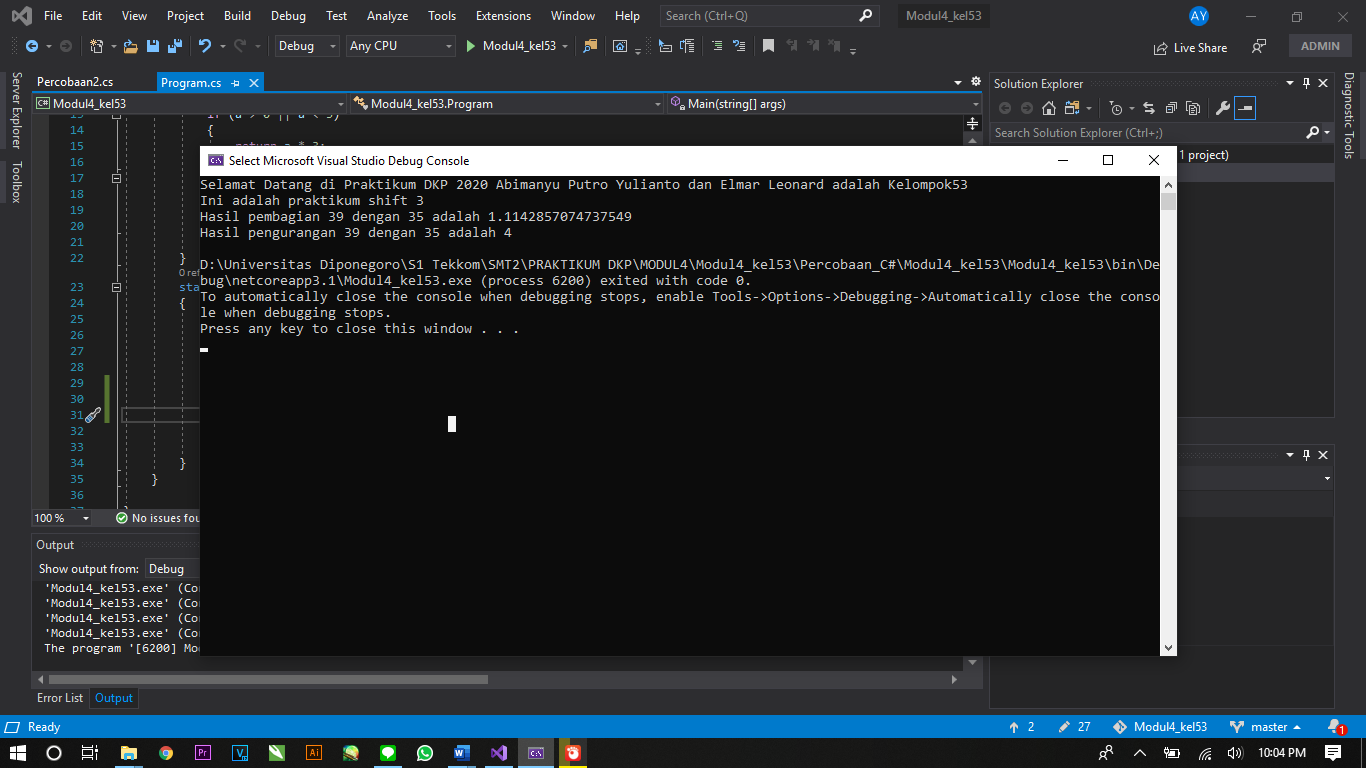
**2.3.4. Method pada C#**

Dibawah ini

*Source code:*

|  |
| --- |
| 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;  }  } |

Hasil (*Output*):



Gambar 2.3.4. Method pada C#

Analisis:

Pada *class* baru berisikan *class* pendefinisi membuat 2 *method*. *Method* pertama bernama pembagian yang dideklarasikan 2 variabel yang berberntuk *float* yaitu float a, float b lalu ada perintah double c = a / b; yang artinya variabel a akan dibagi dengan variabel b yang meghasilkan variabel c. lalu ada perintah Console.WriteLine("Hasil pembagian {0} dengan {1} adalah {2}", a, b, c); yang akan mencetak hasilnya jika *method* ini dipanggil. Lalu *method* kedua bernama pengurangan yang dideklarasikan 2 variabel berbentuk *integer* yaitu int a, int b yang berisikan perintah int c = a - b; yang artinya variabel a akan dikurangi dengan variabel b yang hasilnya adalah variabel c nanti ketika dipanggil fungsi ini akan menjalankan perintah tersebut.

Pada *main class* untuk memanggil *method* tersebut menggunakan perintah yang pertama percobaan2 objek = new percobaan2(); yang artinya adalah percobaan2 merupakan nama *class* pendefinisi yang berisi *method* tadi, objek merupakan nama objek. Lalu dengan objek.pembagian(39, 35); untuk memanggil *method* pembagian dengan parameter 39 dan 35. Lalu nilai parameter itu akan dimasukan ke dalam *method* pembagian. Dan yang terakhir Console.WriteLine("Hasil pengurangan {0} dengan {1} adalah {2}", 39, 35, objek.pengurangan(39, 35));. Pada objek.pengurangan(39, 35)); untuk memanggil *method* pengurangan dan nilai 39 dan 35 akan diproses pada perintah yang sudah dibuat pada *method* pengurangan pada class pendefinisi. Karena ada perintah Console.WriteLine akan dicetak yang hasil *output*nya Hasil pengurangan 39 dengan 35 adalah 4.

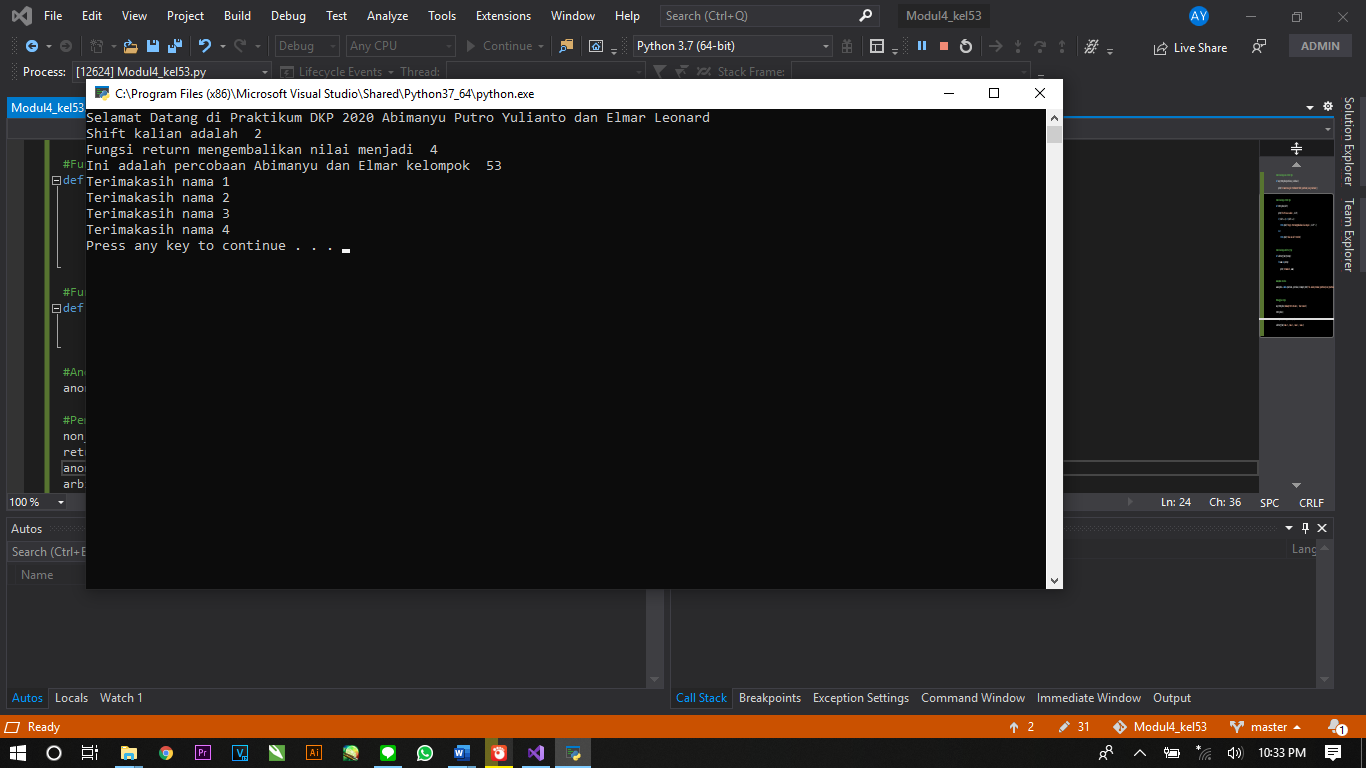
**2.3.5. Function pada Python**

Dibawah ini

*Source code:*

|  |
| --- |
| #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("Abimanyu Putro Yulianto", "Elmar Leonard")  return\_func(2)  anonim\_func("Abimanyu", "Elmar", 53)  arbitrary\_func("nama 1", "nama 2", "nama 3", "nama 4") |

Hasil (*Output*):



Gambar 2.3.5. Function pada Python

Analisis:

Di *Python* untuk menuliskan *function* harus menggunakan def didepannya. Lalu nama fungsinya non\_return\_func lalu dengan parameter praktikan1, praktikan2 lalu didalamnya ada perintah print(f"Selamat Datang di Praktikum DKP 2020 {praktikan1} dan {praktikan2}") untuk menulis seperti itu harus menggunakan format *string* yang baru yaitu f *string.* Lalu dengan perintah non\_return\_func("Abimanyu Putro Yulianto", "Elmar Leonard")maka parameter yang berada didalamnya akan dimasukkan kedalam parameter pada fungsinya yaitu praktikan1 menjadi Abimanyu Putro Yulianto dan praktikan2 menjadi Elmar Leonard.Untuk *return function* karena ada perintah return\_func(2) maka angka 2 akan dimasukkan ke dalam def return\_func(shift): yang didalamnya ada *if else* nya. Karena yang dimasukkan adalah angka 2 artinya memenuhi syaratnya dan akan dicetak angka 2 pada *output*nya. Lalu ada *Arbitrary function* yang jumlah parameternya fleksibel jadi bisa bebas jumlah parameternya. Dengan perintah arbitrary\_func("nama 1", "nama 2", "nama 3", "nama 4") maka ada 4 parameter yang tertuliskan dan akan dimasukkan kedalam def arbitrary\_func(\*penutup): dengan penambahan perintah print("Terimakasih", nama) yang nantinya sebelum nama 1 akan tercetak tulisan Terimakasih. Lalu ada *Anonymous function* yang tidak perlu dideklarasikan sebagai function dengan menggunakan lambda dan karena perintah anonim\_func("Abimanyu", "Elmar", 53) nilai parameter didalamnya akan dimasukkan kedalam anonim\_func = lambda praktikan1, praktikan2, kelompok: print(f"Ini adalah percobaan {praktikan1} dan {praktikan2} kelompok ", kelompok ) yang akan dimasukkan kedalam lambda praktikan1, praktikan2, kelompok dan ada perintah untuk mencetak ini Ini adalah percobaan {praktikan1} dan {praktikan2} kelompok ", kelompok dimana parameter {praktikan1} akan diganti dengan Abimanyu, parameter {praktikan2} diganti dengan Elmar, dan kelompok akan ditambahkan dengan angka 53 karena ada , kelompok.

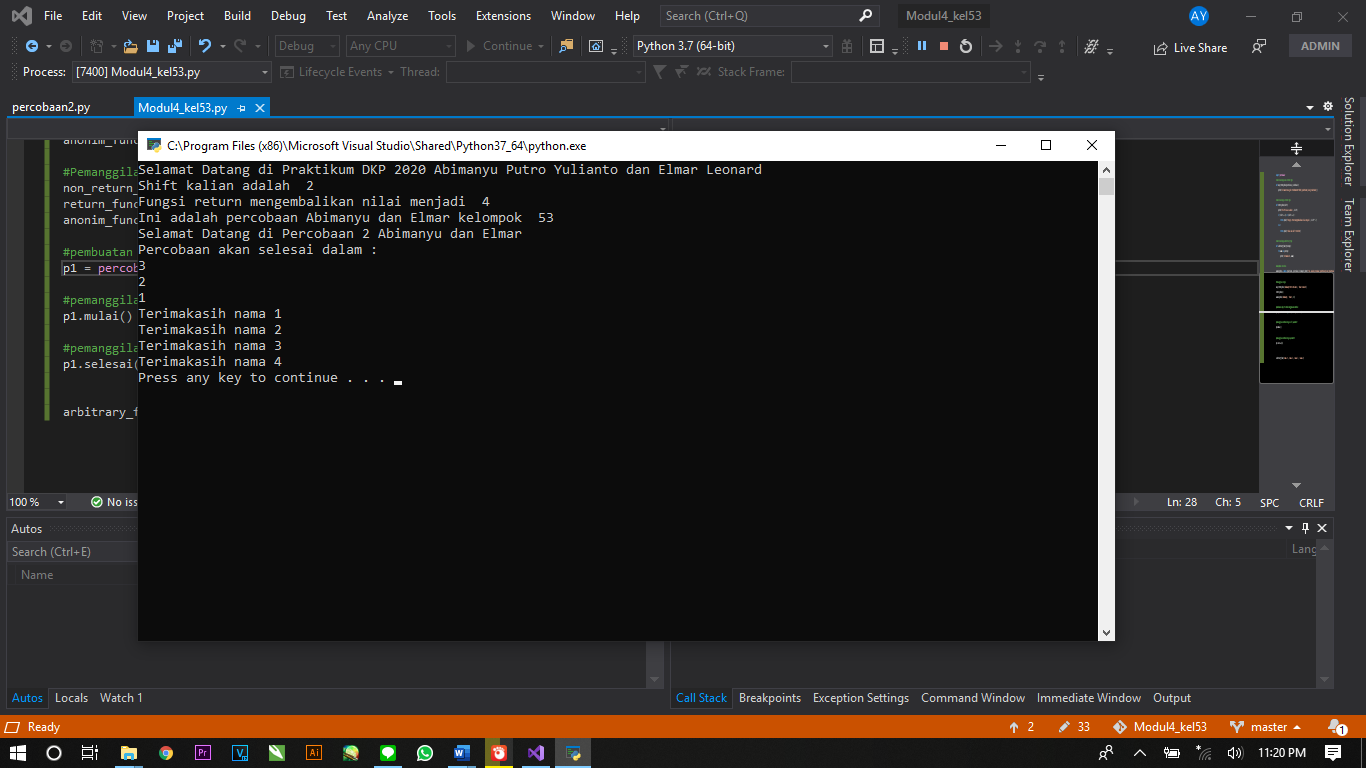
**2.3.6. Method pada Python**

Dibawah ini

*Source code:*

|  |
| --- |
| 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  }  } |

Hasil (*Output*):



Gambar 2.3.6. Method pada Python

Analisis:

Untuk memanggil/memasukkan *calss* untuk *method* tersebut dengan menambahkan import percobaan2 pada awal program. Pertama deklarasikan init method yang akan otomatis akan dijalankan ketika *class* itu dipakai dengan perinth p1 = percobaan2.contoh\_method(“Abimanyu”, “Elmar”) dimana parameter yang terisi itu akan dimasukkan kedalam def \_\_init\_\_(self, praktikan1, praktikan2):. Untuk mendeklarasikan parameter di *method* harus menggunakan self parameter. Perintah p1.mulai() untuk memanggil *method* dengan *self* parameter yang berisi perintah print(f"Selamat Datang di Percobaan 2 {self.praktikan1} dan {self.praktikan2}") dimana perintah itu akan menggambil parameter dari init method yang akan dimasukkan kedalamnya yaitu parameter (“Abimanyu”, “Elmar”) yang akan langsung dicetak karena ada perintah print. Lalu ada perintah p1.selesai(3) untuk menjalankan *method* dengan parameter yang bernama selesai. Angka 3 dalam parameternya itu akan dimasukkan ke dalam *method* dengan parameter yang berisikan perintah jika parameter yang dimasukan memenuhi while waktu > 0: maka angka 3 akan diproses dengan mencetak angka itu yang akan diproses lagi dengan perintah waktu -= 1 yang artinya angka 3 akan dikurangi sampai samadengan 1 dan akan berhenti.

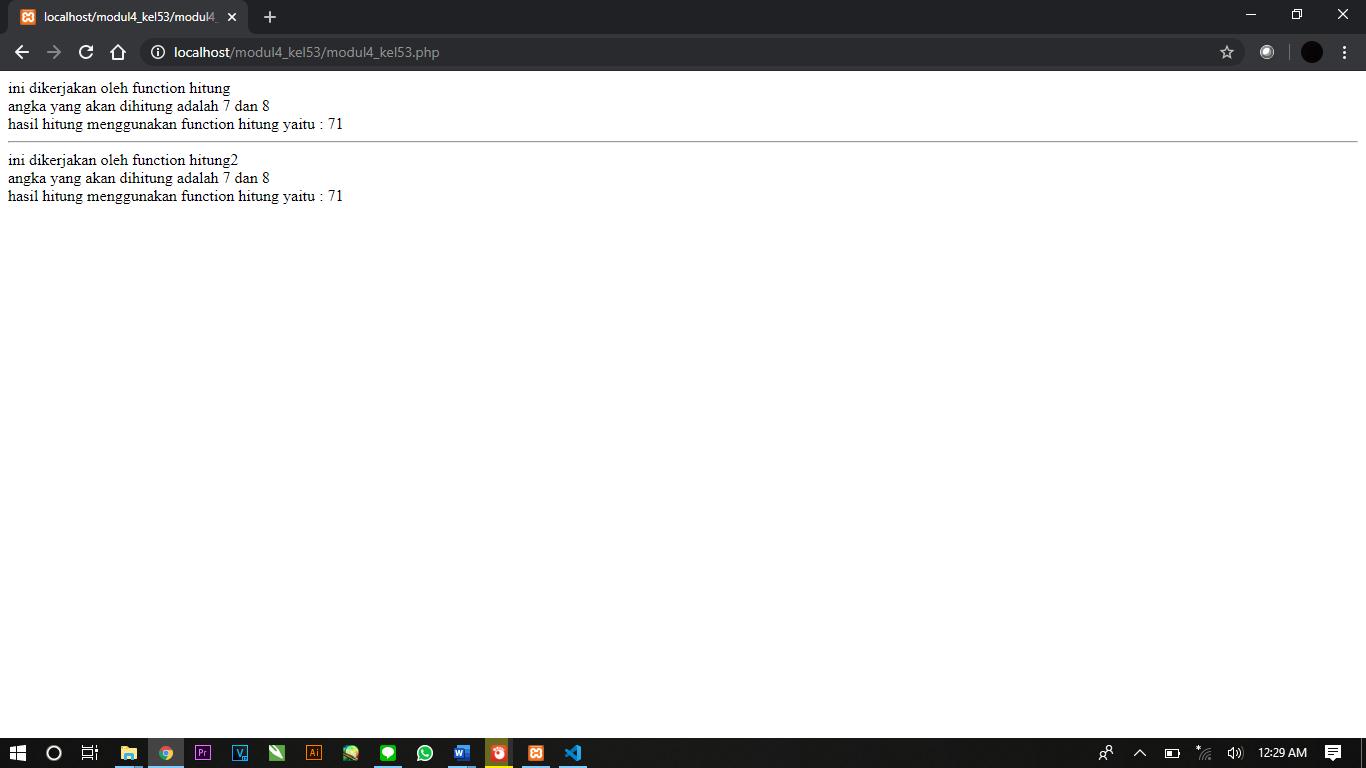
**2.3.7. Function pada PHP**

Dibawah ini

*Source code:*

|  |
| --- |
| <?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);  ?> }  } |

Hasil (*Output*):



Gambar 2.3.7. Function pada PHP

Analisis:

Yang pertama *function* dengan *return type* yang diberi nama function hitung dengan parameternya ($bil1, $bil2) dengan nilai return $bil1 + $bil2 \* $bil2; lalu deklarasikan $bil1 dengan angka 7, deklarasikan $bil2 dengan angka 8 dan ada $hasil dengan parameter memanggil function hitung ($bil1, $bil2). Lalu ada perintah untuk mencetak output echo "ini dikerjakan oleh function hitung <br>";. Lalu perintah echo "angka yang akan dihitung adalah $bil1 dan $bil2 <br>"; untuk mencetak output dengan memasukkan nilai parameter dari $bil1 = 7 dan $bil2 = 8; yang juga akan ikut dicetak. Dan terakhir ada echo "hasil hitung menggunakan function hitung yaitu : $hasil"; yang artinya akan dicetak yang hasil outputnya akan dimasuki hasil hitung dari parameter $hasil.

Yang kedua ada *function non-return type* yang diberi nama function hitung2 degan parameyternya ($bil1, $bil2). Langsung menuju perintah $hasil yang berisi perintah memanggil function hitung ($bil1, $bil2) jadi walaupun tidak ada *return type* nya tetap bisa memanggil function hitung jadi hasil *output*nya akan tetap sama dengan yang pertama.

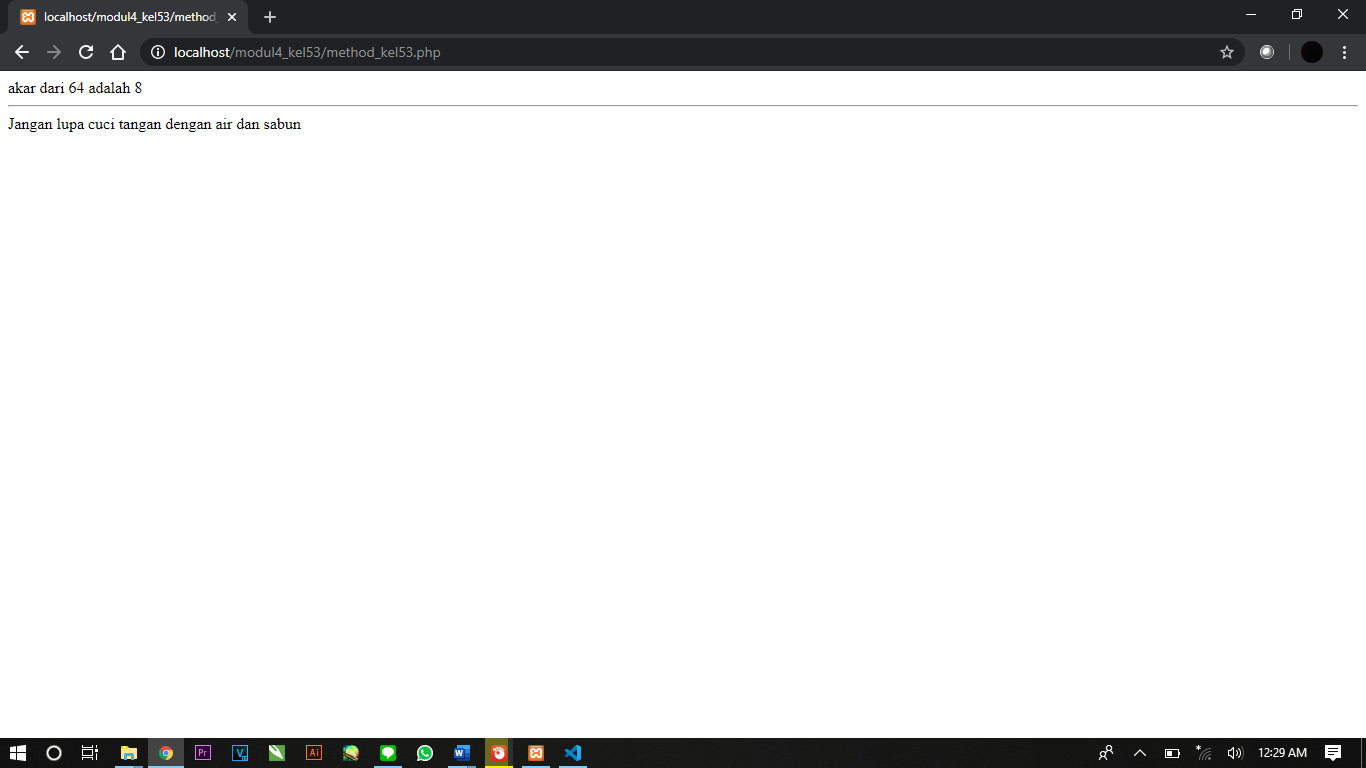
**2.3.8. Method pada PHP**

Dibawah ini

*Source code:*

|  |
| --- |
| <?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');    ?> |

Hasil (*Output*):



Gambar 2.3.8. Method pada PHP

Analisis:

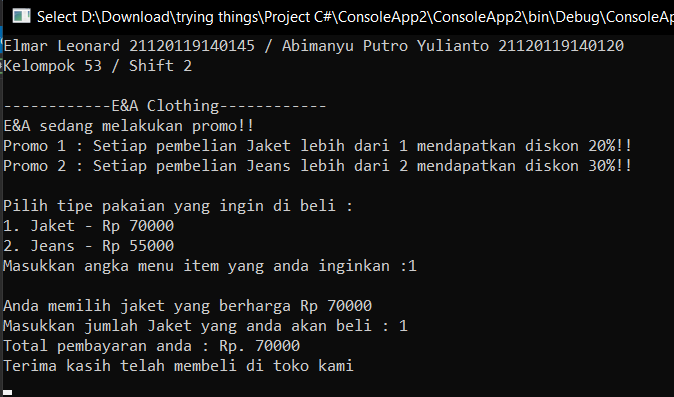
Pertama ada *class* pendefinisi *method* dengan membuat class pendefinisi yang berisi *method* bernama akar dengan variabel $angka yang berbentuk float lalu akan mengembalikan nilai sqrt($angka); dimana sqrt adalah rumus untuk akar. Lalu ada *method* bernama staysafe dengan parameter $noun1 dan $noun2 bertipe data *string.*

Kedua ada *code* untuk memanggil *method* dengan mendeklarasikan $angka = 64 lalu ada instansiasi objeknya ada $objek = new pendefinisi();. Lalu ada echo "akar dari $angka adalah ".$objek -> akar($angka); yang artinya akan dicetak hasil output dengan memasukan nilai dari variabel $angka yaitu 64 dan memanggil *method* akaryang sudah ada rumus untuk akarnya dengan parameternya dari $angka jadi akan menghasilkan akar dari 64 adalah 8. Terakhir ada $objek -> dengan memanggil *method* staysafe yang parameternya berisi ('air', 'sabun') akan dimasukkan kedalam $noun1 dan $noun2 pada perintah echo "Jangan lupa cuci tangan dengan $noun1 dan $noun2"; jadi hasil outputnya adalah Jangan lupa cuci tangan dengan air dan sabun.

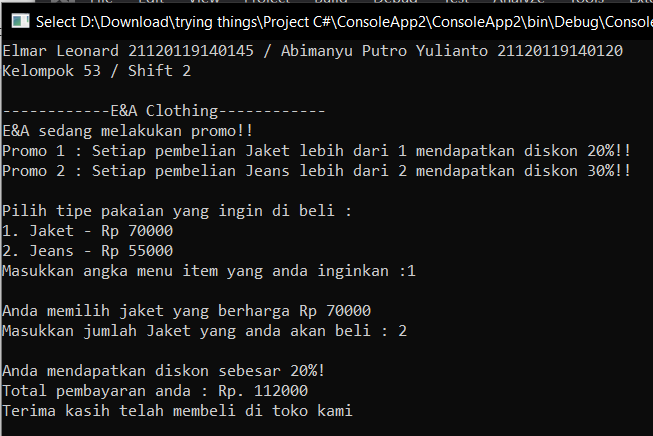
# **2.4. Tugas**

|  |
| --- |
| using System;  class MainClass  {  public static void Main(string[] args)  {  Console.WriteLine("Elmar Leonard 21120119140145 / Abimanyu Putro Yulianto 21120119140120");  Console.WriteLine("Kelompok 53 / Shift 2\n");  Console.ReadKey();  int tipe;  float hasil1, hasil2, hasil3, hasil4;  Console.WriteLine("------------E&A Clothing------------");  Console.WriteLine("E&A sedang melakukan promo!!");  Console.WriteLine("Promo 1 : Setiap pembelian Jaket lebih dari 1 mendapatkan diskon 20%!!");  Console.WriteLine("Promo 2 : Setiap pembelian Jeans lebih dari 2 mendapatkan diskon 30%!!\n");  Console.ReadKey();  Console.WriteLine("Pilih tipe pakaian yang ingin di beli : ");  Console.WriteLine("1. Jaket - Rp 70000");  Console.WriteLine("2. Jeans - Rp 55000");  Console.Write("Masukkan angka menu item yang anda inginkan :");  tipe = Convert.ToInt16(Console.ReadLine());  switch (tipe)  {  case 1:  Console.WriteLine("\nAnda memilih jaket yang berharga Rp 70000");  Console.Write("Masukkan jumlah Jaket yang anda akan beli : ");  int jumlah1 = Convert.ToInt16(Console.ReadLine());  int harga1 = 70000;  if (1 < jumlah1)  {  hasil1 = harga1 \* jumlah1; |

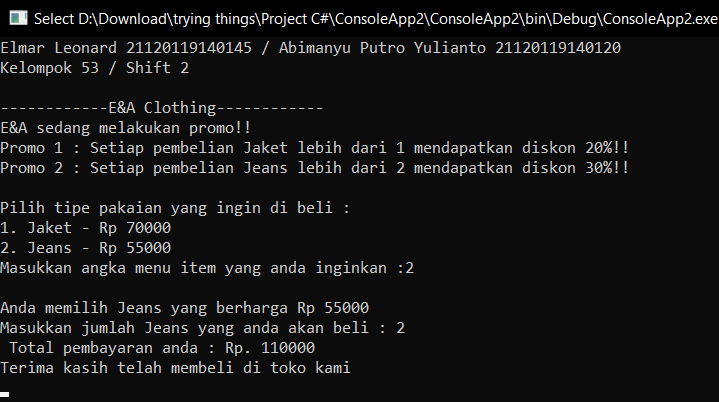
|  |
| --- |
| hasil2 = hasil1 - (hasil1 \* 20 / 100);  Console.WriteLine("\nAnda mendapatkan diskon sebesar 20%!");  Console.WriteLine("Total pembayaran anda : Rp. " + hasil2);  }  else  {  hasil1 = harga1 \* jumlah1;  Console.WriteLine("Total pembayaran anda : Rp. " + hasil1);  }  break;  case 2:  Console.WriteLine("\nAnda memilih Jeans yang berharga Rp 55000");  Console.Write("Masukkan jumlah Jeans yang anda akan beli : ");  int jumlah2 = Convert.ToInt16(Console.ReadLine());  int harga2 = 55000;  if (2 < jumlah2)  {  hasil3 = harga2 \* jumlah2;  hasil4 = hasil3 - (hasil3 \* 30 / 100);  Console.WriteLine("\nAnda mendapatkan diskon sebesar 30%!");  Console.WriteLine("Total pembayaran anda : Rp. " + hasil4);  }  else  {  hasil3 = harga2 \* jumlah2;  Console.WriteLine(" Total pembayaran anda : Rp. " + hasil3);  }  break;  }  Console.WriteLine("Terima kasih telah membeli di toko kami");  Console.ReadKey();  }  } |



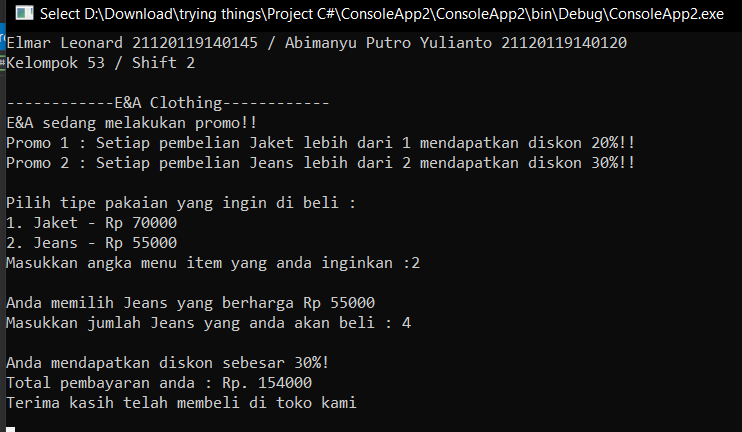
Gambar 3.4.1. Output hasil memilih pilihan 1 tanpa diskon



Gambar 3.4.2. . Output hasil memilih pilihan 1 dengan diskon



Gambar 3.4.3. . Output hasil memilih pilihan 2 tanpa diskon



Gambar 3.4.4. . Output hasil memilih pilihan 2 dengan diskon

Analisis:

# **2.5. Kesimpulan**

Setelah melakukan praktikum ini kami memiliki beberapa kesimpulan, yaitu: