**BAB V**

**FUNCTION DAN METHOD**

# ****Tujuan****

1. Dapat mengerti dan memahami pengertian dari *function dan method*.
2. Dapat memahami konsep dari *function dan method*.
3. Dapat mengerti dan memahami tujuan penggunaan *function dan method* pada bahasa pemrograman Python, C#, Java, dan PHP.
4. Dapat mengimplementasikan dengan benar penggunaan *function dan method* pada bahasa pemrograman Python, C#, Java, dan PHP.
5. Dapat mengetahui perbedaan penggunaan *function dan method* pada bahasa pemrograman Python, C#, Java, dan PHP.
6. Dapat mengerti dan mengetahui susunan penggunaan *function dan method* pada bahasa pemrograman Python, C#, Java, dan PHP.
7. Dapat membuat program sederhana dalam bahasa pemrograman apapun dengan menyertakan penggunaan *function dan method*.

# ****Dasar** **Teori****

* + 1. **Function**

Fungsi merupakan sub program yang sering juga disebut sub rutin program yang terdiri dari sekumpulan perintah yang akan melaksanakan suatu tugas tertentu. Fungsi memungkinkan program menjadi lebih terstruktur dan juga memudahkan pelacakan kesalahan. Menggunakan fungsi akan mempersingkat waktu pemrogram untuk menuliskan sekumpulan perintah yang sama berkali-kali. *Function* atau fungsi adalah satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang di eksekusi ketika dipanggil dari bagian lain dalam suatu program.

Tujuan utama dari penggunaan *function* atau fungsi adalah memudahkan dalam penggunaan program dan menghemat ukuran program yang dibuat. Keuntungan-keuntungan lain menggunakan atau memakai *function* adalah :

* Dapat menguraikan tugas pemrograman rumit menjadi langkah-langkah yang lebih sederhana atau lebih kecil.
* Dapat mengurangi duplikasi kode (kode yang sama ditulis berulang-ulang) dalam program.
* Dapat menggunakan kode yang ditulis dalam berbagai program yang berbeda.
* Dapat memecah program besar menjadi kecil, sehingga dapat dikerjakan oleh *programmer-programmer* atau dipecah menjadi beberapa tahap sehingga mempermudah pengerjaan dalan sebuah proyek.
* Dapat menyembunyikan informasi dari *user* sehingga mencegah adanya pembuatan iseng seperti memodifikasi atau mengubah program yang kita buat.
* Dapat meningkatkan kemampuan pelacakan kesalahan. Apabila terjadi suatu kesalahan, kita hanya mencari fungsi yang bersangkutan saja dan tidak perlu mencari kesalahan tersebut di seluruh program.

Dalam bahasa pemrograman, terdapat dua jenis fungsi, yaitu :

* *Return type* atau *non-void type* (mengembalikan suatu nilai)

Fungsi ini akan mengembalikan suatu nilai baru dari proses yang telah dilakukan. Ciri utama dari jenis fungsi ini adalah terdapat *keyword return* di dalam *block code*. Nilai yang dikembalikan oleh jenis fungsi ini akan sesuai dengan tipe data yang digunakan pada saat mendeklarasikan fungsi tersebut.

* *Non-return type* atau *void type* (tidak mengembalikan suatu nilai)

Fungsi ini dideklarasikan dengan tipe data *void,* yang artinya fungsi ini tidak akan mengembalikan suatu nilai baru setelah proses atau pekerjaannya selesai. Sehingga, fungsi tipe *void* tidak memerlukan *keyword return* dalam pemakaiannya.

(Sumber : [*http://ryanmuhammad-bahrudin.blogspot.com/2015/01/pengertian-function-beserta-contoh-nya.html*](http://ryanmuhammad-bahrudin.blogspot.com/2015/01/pengertian-function-beserta-contoh-nya.html))

* + 1. **Method**

Perlu kita ketahui bahwa *method* merupakan sebuah fungsi. *Method* merupakan sebuah blok program terpisah (di luar program utama) yang kita gunakan atau yang kita pakai untuk menyelesaikan suatu masalah atau suatu *problem* khusus. Tujuan dari penggunaan *method* ini adalah untuk memecah program kompleks menjadi bagian-bagian kecil sehingga nantinya dapat kita gunakan secara berulang-ulang tanpa harus menulis baris kode yang sama.

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

(Sumber : [*https://jogjaweb.co.id/blog/pengertian-class-object-property-dan-method*](https://jogjaweb.co.id/blog/pengertian-class-object-property-dan-method))

* + 1. **Penggunaan Function**

Secara umum, suatu *function* atau fungsi mempunyai beberapa unsur, yaitu :

* *Return type*

Fungsi dapat mengembalikan suatu nilai. *Return type* merupakan tipe data dari nilai fungsi yang dikembalikan. Beberapa fungsi melakukan operasi yang diinginkan tanpa mengembalikan suatu nilai.

* *Function name*

Ini merupakan nama sebenarnya dalam suatu fungsi. Nama fungsi dan daftar *parameter* bersama merupakan *function signature*.

* *Parameter list*

*Parameter* seperti *placeholder.* Saat sebuah fungsi dipanggil, anda melewatkan sebuah nilai sebagai *parameter*. Nilai ini disebut sebagai *parameter* atau *argument.* Daftar *parameter* mengacu pada tipe, urutan, dan jumlah *parameter* fungsi. *Parameter* bersifat *optional*, yang artinya fungsi tidak mengandung *parameter*.

* *Function body*

Berisi kumpulan pernyataan yang mendefinisikan fungsi yang dilakukannya.

* Ruang lingkup penggunaan (*optional*)

Ruang lingkup atau biasa kita sebut dengan istilah *access modifier/access specifier* berfungsi untuk menentukkan *class/object* mana saja yang menggunakan *variable* maupun *function/method* yang berada di dalam *class* tersebut.

Terdapat tiga (3) jenis *modifier,* yaitu :

1. *Public*

Mengindentifikasi suatu *class* atau *variable* suatu *function* atau *method* dengan *modifier* ini dapat dipergunakan oleh semua *class* yang tersedia.

1. *Private*

Mengidentifikasi suatu *class* atau *variable* suatu *function* atau *method* dengan *modifier* ini dapat dipergunakan hanya oleh *class* itu sendiri.

1. *Protected*

Mengidentifikasi suatu *class* atau *variable* suatu *function* atau *method* dengan *modifier* ini dapat dipergunakan hanya oleh *class* itu sendiri dan *sub-class.*

(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))

* + 1. **Penggunaan Method**

*Method* adalah sarana bagi *programmer* untuk memodularisasi, artinya mem-*break* atau memecah program kompleks menjadi bagian yang kecil-kecil sehingga nantinya dapat digunakan berulang-ulang, daripada menulis beberapa baris kode yang sama. Dalam penggunaannya, perlu diperhatikan aturan penamaannya.

Penamaan pada *method* mengikuti kaidah penamaan seperti pada penamaan *variable*. Bagian *parameter* pada *method* diisi dengan *parameter-parameter* fungsi yang diperlukan atau yang dibutuhkan. *Passing parameter* dilakukan dengan cara *pass by value* dan *pass by reference*. Dengan analogi tersebut, otomatis beberapa hal umum, seperti *local variable*, pemanggilan *method,* pendefinisian *return value type*, dan sebagainya adalah sama.

(Sumber : [*https://irfanisprayudhi.wordpress.com/2012/11/13/method-pada-java/*](https://irfanisprayudhi.wordpress.com/2012/11/13/method-pada-java/))

# ****Hasil dan Analisa Hasil****

* + 1. **Function pada Java**

Pada percobaan pertama, buat *project* dengan nama modul4\_kel4**.** Setelah itu masukkan *source code* ke dalam editornya.

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 XX");

//penggunaan function non-return type tanpa parameter

cetak1();

//penggunaan function return type dengan mengisikan nilai pada parameter text

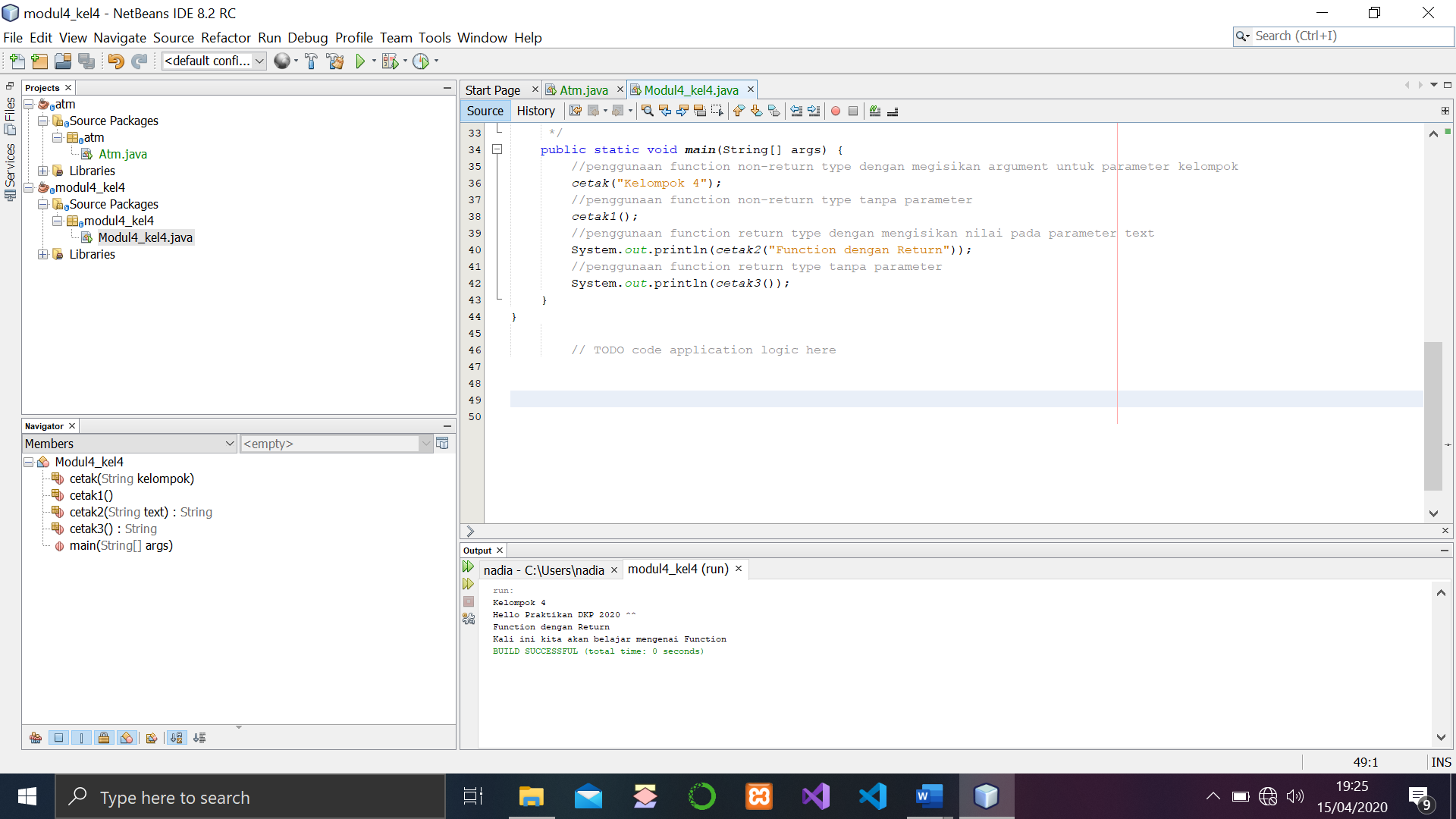
System.out.println(cetak2("Function dengan Return"));

//penggunaan function return type tanpa parameter

System.out.println(cetak3());

}

}



Gambar 5.1 Output function pada java

Pada program di atas ada beberapa hal yang perlu diperhatikan. Terdapat fungsi cetak() yang merupakan *non-return type*. Fungsi *non-return type* merupakan fungsi yang tidak mengembalikan nilai. Fungsi ini mempunyai sebuah *parameter* dengan tipe *string* dan *variable*-nya kelompok. Selanjutnya terdapat fungsi cetak1(). Fungsi cetak1() ini juga merupakan *non-return type*. Fungsi *non-return type* merupakan fungsi yang tidak mengembalikan nilai. Namun, berbeda dengan fungsi cetak() tadi, fungsi ini tidak mempunyai *parameter*.

Terdapat pula fungsi cetak2(). Fungsi ini merupakan *return type*. Fungsi *return type* merupakan fungsi yang mengembalikan nilai. Fungsi ini mempunyai *parameter* dengan tipe data *string* dan *variable*-nya *text*. Dan yang terakhir ada fungsi cetak3(). Fungsi cetak3() ini juga merupakan *return type*. Fungsi *return type* merupakan fungsi yang mengembalikan nilai. Namun, fungsi ini tidak mempunyai *parameter*.

* + 1. **Method pada Java**

Pada percobaan kedua, membuat *class* dengan nama pendefinisi. Setelah itu isikan editor dengan *source code* seperti pada dibawah ini.

*Class* pendefinisi *method* :

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;

}

}

Setelah itu, klik *file* modul4\_kel4 yang tadi digunakan pada percobaan. Tambahkan komentar (//) pada tiap baris kode dibawah public static void main(String[] args) {, kemudian tambahkan *source code* pada *file* modul4\_kel4 seperti gambar berikut.

*Class* pemanggil *method* :

public static void main(String[] args) {

pendefinisi objek = new pendefinisi();

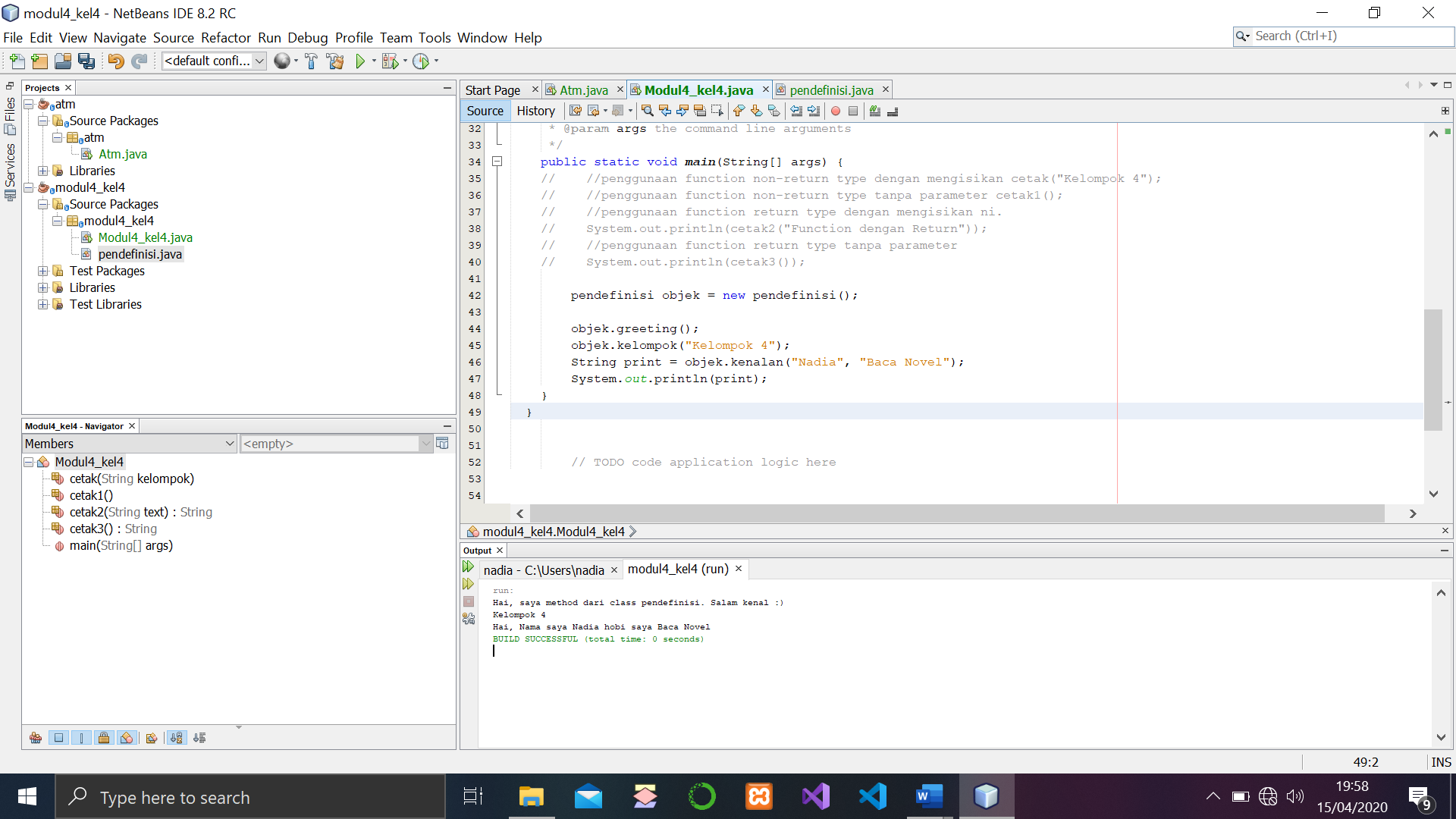
objek.greeting();

objek.kelompok("Kelompok xx");

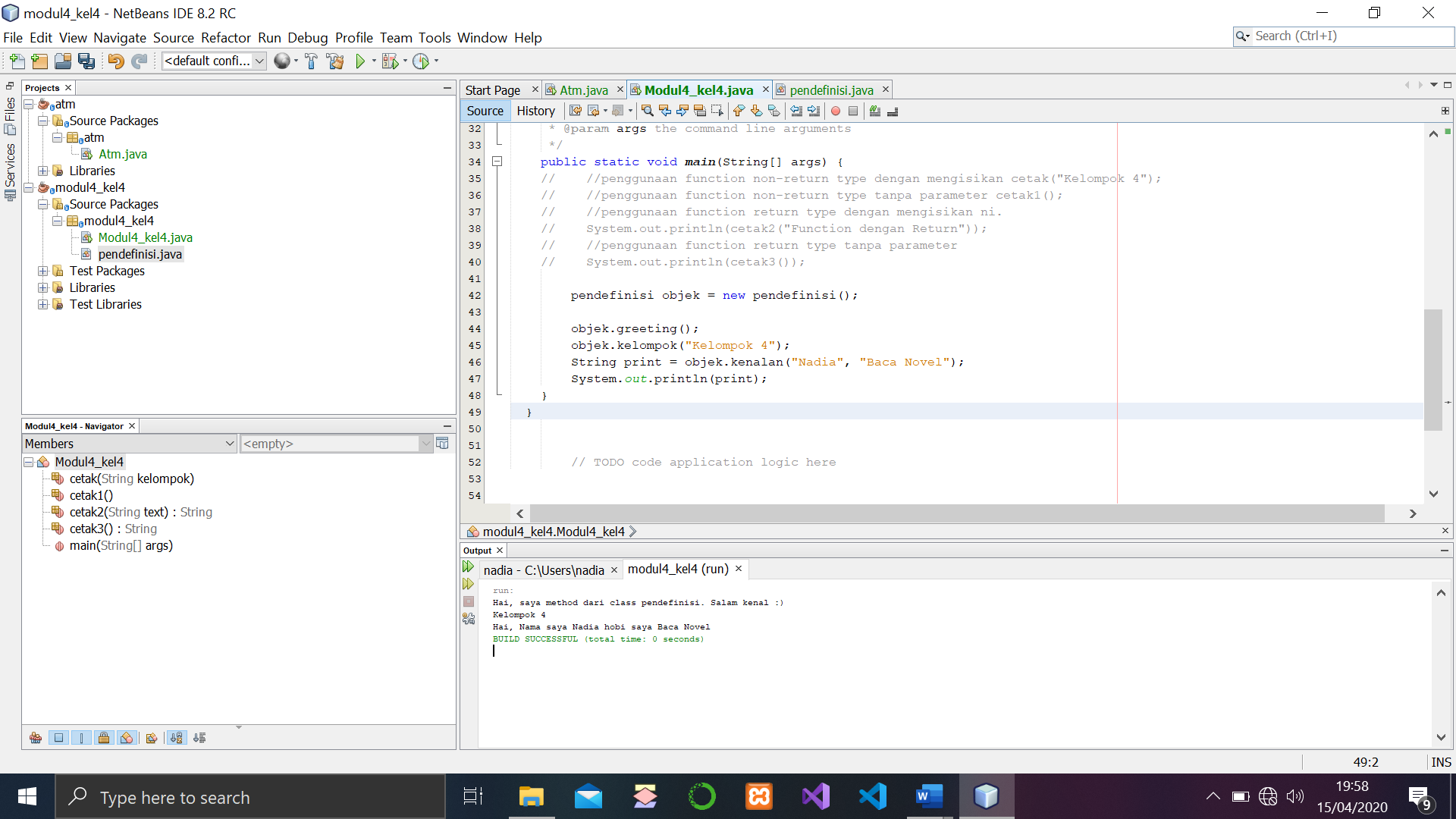
String print = objek.kenalan("Erika", "Baca Novel");

System.out.println(print);

}



Gambar 5.2 source code pada file modul4\_ kel4



Gambar 5.3 output method pada java

Pada program tersebut kita membuat *class* pada *file* modul4\_kel4. *Class* tersebut diberi nama pendefinisi. Pada *source code class* pendefinisi *method,* terdapat System.out.println("Hai, saya method dari class pendefinisi. Salam kenal :)"); yang mempunyai fungsi untuk mengeluarkan atau menyajikan *output* atau keluaran. Untuk *class* pemanggil *method,* klik file modul4\_kel4 yang tadi digunakan pada percobaan pertama, lalu tambahkan komentar (//) pada tiap baris kode di bawah public static void main(String[] args) {, kemudian tambahkan *source code* pada *file* modul4\_kel4.

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

Untuk percobaan pertama, buat *project* dengan nama modul4\_kel4. Setelah itu masukkan *source code* ke dalam editornya.

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("Praktikan 1", "Praktikan 2", "Kelompok4");

Console.WriteLine("Ini adalah praktikum shift {0}", return\_func(1));

//percobaan2 objek = new percobaan2();

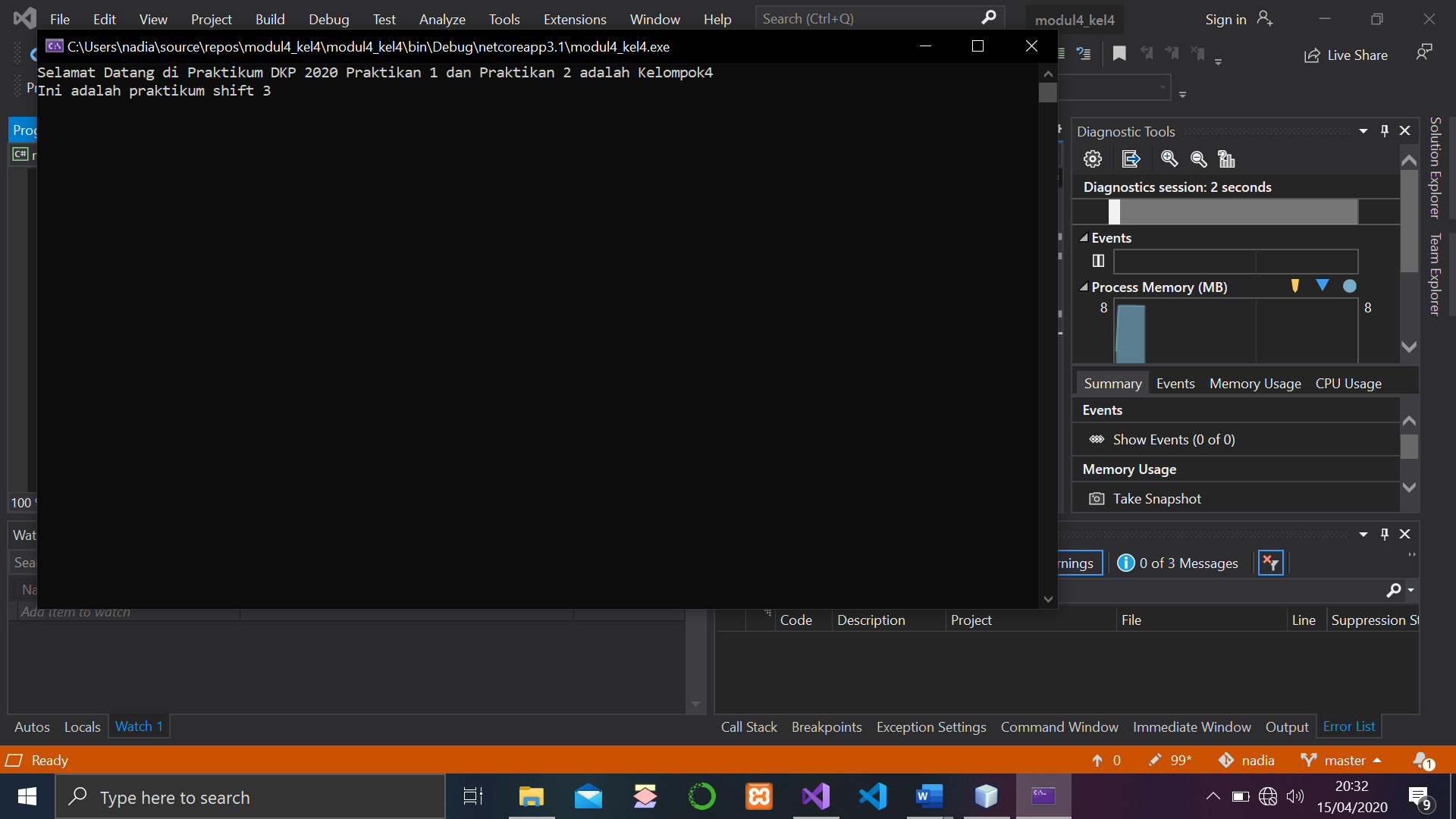
//objek.pembagian(39, 35);

//Console.WriteLine("Hasil pengurangan {0} dengan {1} adalah {2}", 39, 35, objek.pengurangan(39, 35));

Console.ReadKey();

}

}



Gambar 5.4 output function pada C#

Pada program diatas, yaitu *function* pada C# ada 2 *function* yang digunakan dalam program tersebut yaitu non\_return() dan return\_func(). Pada fungsi non\_return() perlu diperhatikan bahwa fungsi akan mencetak *string* ke konsol dengan *parameter* berupa 3 string yaitu a, b, dan c. Lain hal nya pada return\_func(). Pada return\_func()fungsi akan mengembalikan nilai. Fungsi akan mengembalikan nilai dari *parameter int* a. Nilai yang dikembalikan dari *parameter int* a tersebut tergantung dari *parameter* yang diberikan. Apabila nilai berada pada rentang 1 – 3, maka fungsi akan mengembalikan nilai dikali 3 dan jika nilai di luar rentang tersebut, fungsi akan mengembalikan nilai 0.

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

Untuk percobaan kedua tambahkan *file* baru sebagai *class* C# yang akan menampung *method* yang akan digunakan. Lalu langkah selanjutnya adalah klik kanan pada nama *project* yang ada pada panel *Solution Explorer*. Kemudian pilih *add* -> *new item*. Pilih tipe file *class,* kemudian beri nama percobaan2.cs. Pemberian nama ada dibagian bawah. Kemudian klik *add.* Setelah itu masukkan *source code* ke dalam editornya dan hilangkan *comment* yang ada pada *file* pertama.

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)

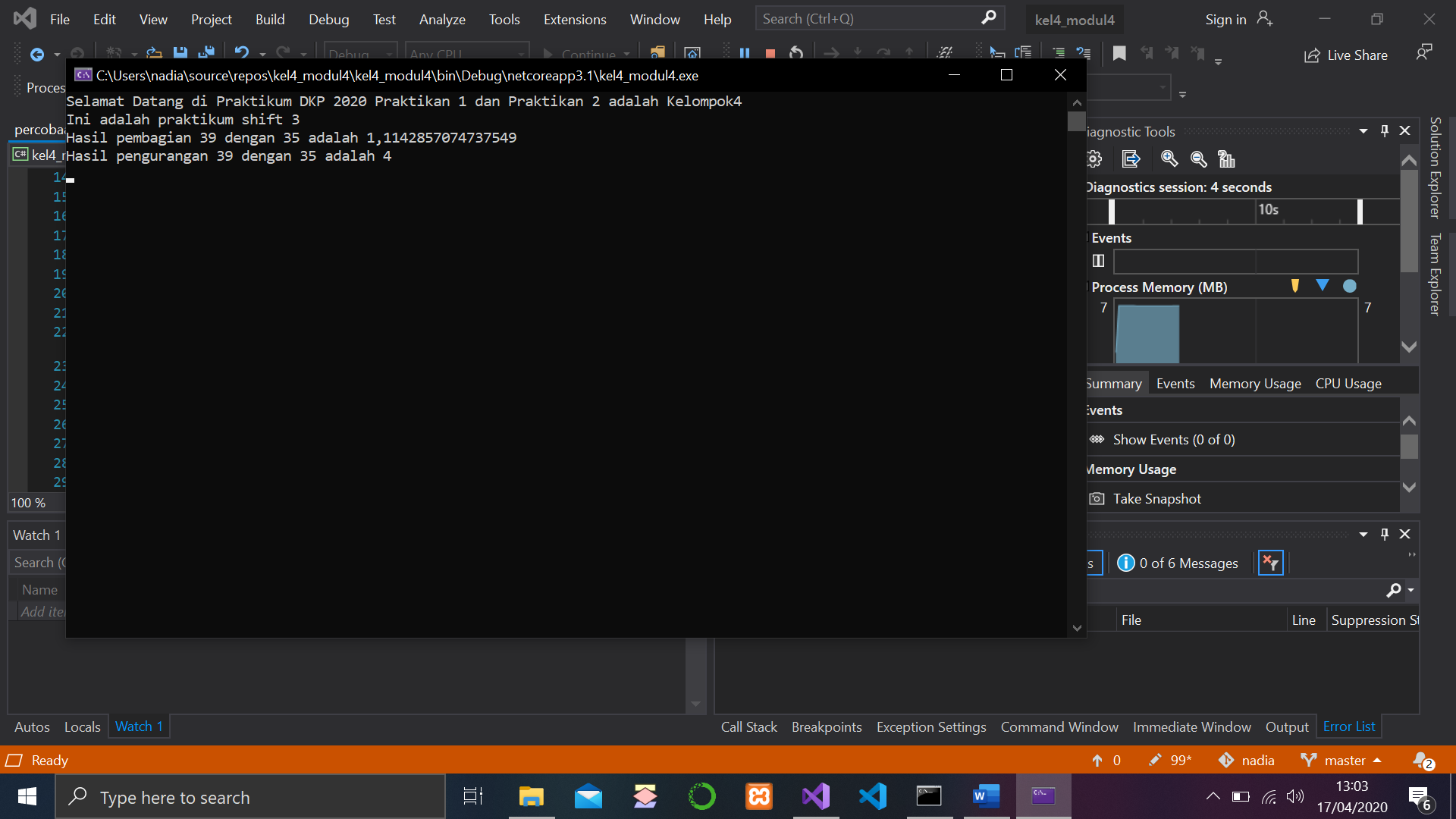
{

int c = a - b;

return c;

}

}



Gambar 5.5 method dalam C#

Dapat kita perhatikan, pada program di atas terdapat 1 *class* baru yang digunakan atau dipakai untuk menampung *method.* Di dalam *class* tersebut, terdapat method pembagian()dengan parameter berupa *float* (*float* a dan *float* b) dan juga method pengurangan() dengan parameter *integer* (*integer* a dan *integer* b). Perlu diperhatikan bahwa untuk memanggil *method* pada *main class*, *class* yang mengandung *method* tersebut harus dideklarasikan terlebih dahulu sebagai objek.

Perlu diperhatikan apabila menggunakan *method –* nya. Penggunaan *method* dapat dilakukan dengan memanggil nama percobaan2 objek = new percobaan2(); digunakan untuk menginisiasikan objek baru yang merujuk pada *class* percobaan2. objek.pembagian(39, 35); digunakan untuk memanggil *method* pembagian yang ada pada *class* percobaan2. Pada percobaan pembagian ini, menggunakan angka 39 dan 35. Hasil dari program tersebut (program pembagian) adalah 1,11428. Kemudian, objek.pengurangan(39, 35) digunakan untuk memanggil *method* pengurangan. Pada percobaan pengurangan ini, digunakan angka 39 dan 35 pula. Hasil dari program tersebut (program pengurangan) adalah 4.

* + 1. **Function pada Python**

Pada percobaan pertama, buat *project* dengan nama modul4\_kel4. Setelah itu masukkan *source code* ke dalam editornya.

#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 )

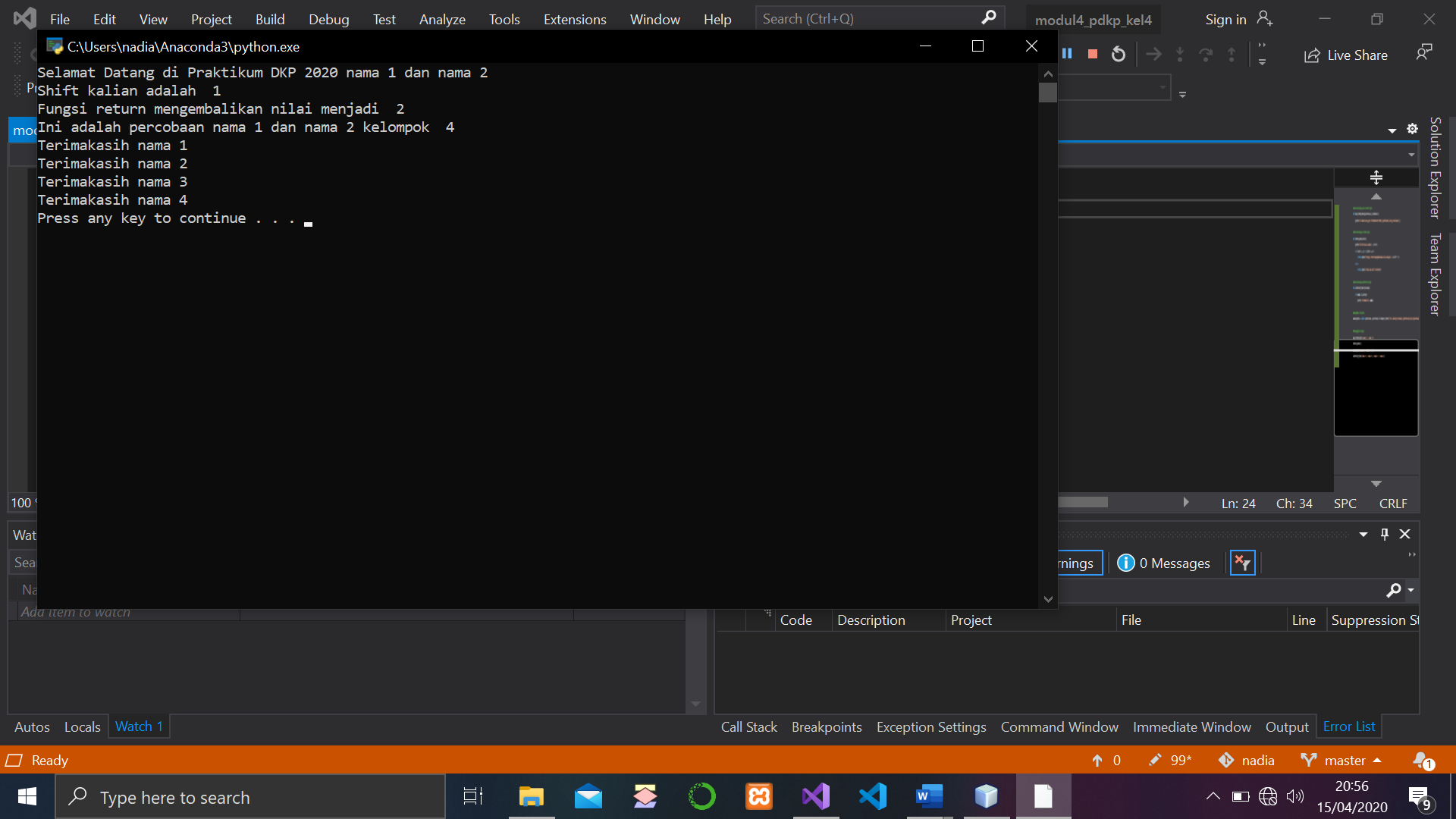
#Pemanggilan Fungsi

non\_return\_func("nama 1", "nama 2")

return\_func(1)

anonim\_func("nama 1", "nama 2", 5)

arbitrary\_func("nama 1", "nama 2", "nama 3", "nama 4")



Gambar 5.6 output function pada python

Pada program tersebut, dapat kita lihat ada empat tipe *function. Type function* tersebut adalah *non – return function, return function, arbitrary function,* dan *anonymous function. Non – return function* merupakan sebuah fungsi yang dideklarasikan dengan tipe data *void,* yang artinya fungsi ini tidak akan mengembalikan suatu nilai baru setelah proses atau pekerjaannya selesai. Sehingga, fungsi tipe *void* tidak memerlukan *keyword return* dalam pemakaiannya. *Return function* merupakansebuah fungsi yang akan mengembalikan suatu nilai baru dari proses yang telah dilakukan. Ciri utama dari jenis fungsi ini adalah terdapat *keyword return* di dalam *block code*. Nilai yang dikembalikan oleh jenis fungsi ini akan sesuai dengan tipe data yang digunakan pada saat mendeklarasikan fungsi tersebut.

*Arbitrary function* merupakan fungsi yang jumlah *parameter* -nya fleksibel. *Parameter* dideklarasikan dengan menambahkan simbol *asterisk* (\*) pada awal parameter. *Arbitrary function* digunakan apabila kita tidak mengetahui secara pasti *parameter* yang akan digunakan pada fungsi. Dan yang terakhir, *anonymous function,* merupakan fungsi yang dapat dideklarasikan tanpa memberikan nama fungsi. Fungsi ini memanfaatkan *lambda* yang ada pada python.

* + 1. **Method pada Python**

Pada percobaan kedua tambahkan *file* baru yang akan menampung *method* yang akan digunakan. Cara menambahkan *file* baru pada python sama seperti penambahan *class* pada C#. Kemudian pada opsi *add new item,* pilih *empty python file,* kemudian beri nama percobaan2.py. Langkah selanjutnya adalah klik *add.* Setelah itu masukkan *source code* ke dalam editornya.

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:

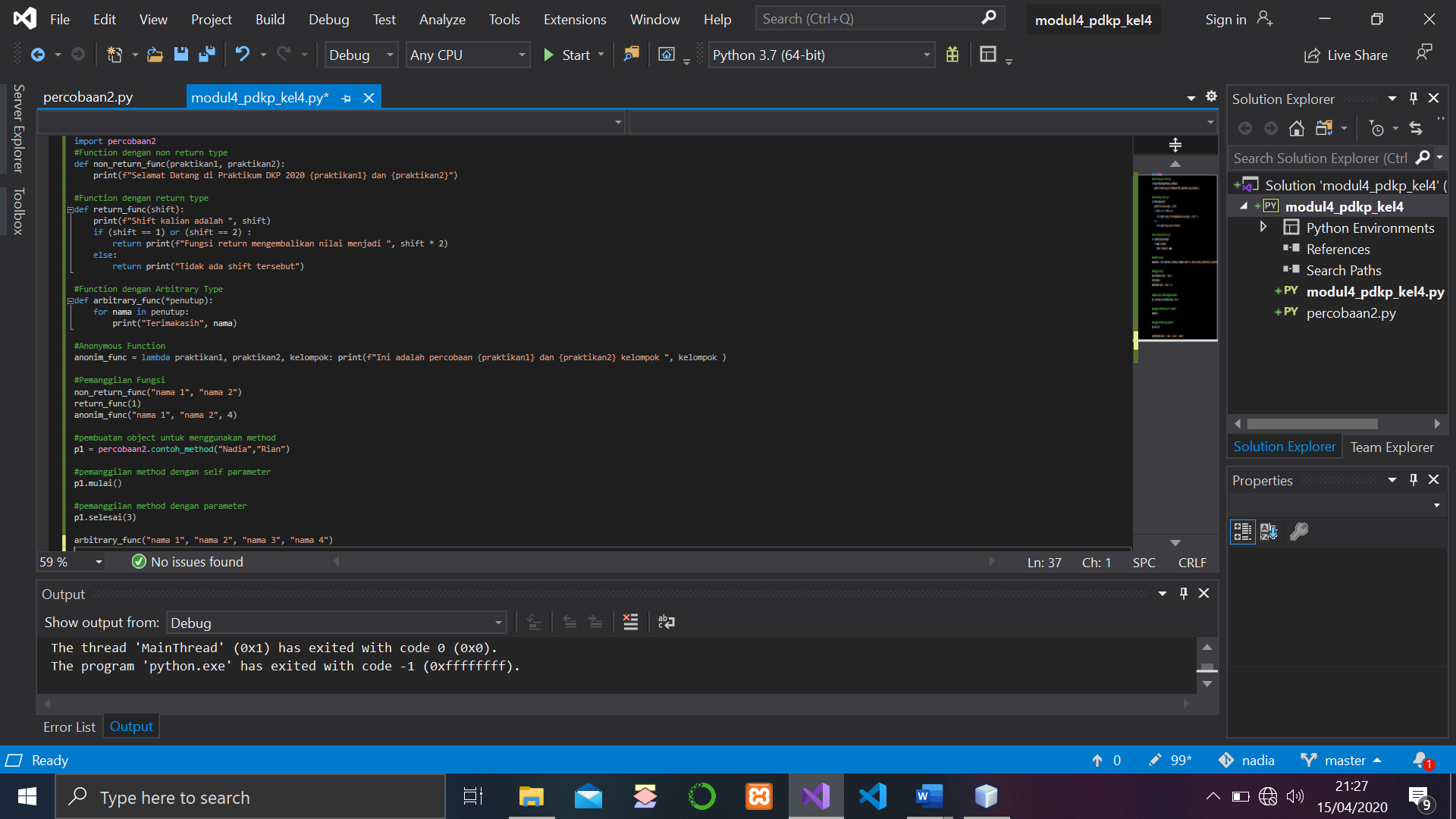
print(waktu)

waktu -= 1

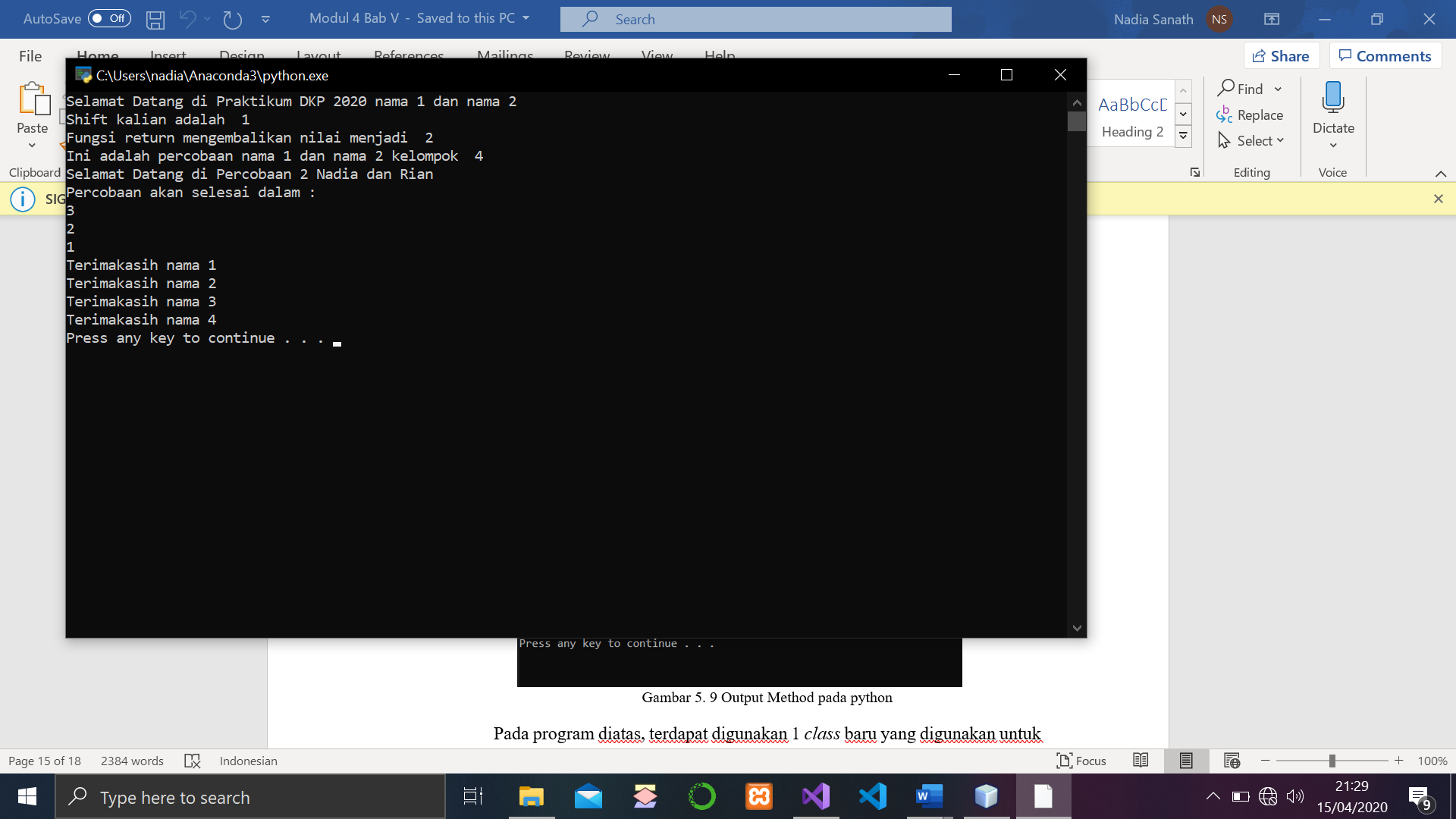
}

}

Pada *file* yang pertama, tambahkan beberapa kode sehingga akan menjadi seperti di bawah ini :



Gambar 5.7 kode method pada python



Gambar 5.8 output method pada python

Pada program di atas, dapat diperhatikan terdapat 1 *class* baru yang digunakan untuk menampung *method.* Pada bahasa pemrograman python, terdapat *method* init(). *Method* init() merupakan *method* yang akan dieksekusi secara otomatis ketika *class* dari *method* tersebut dipanggil pada sebuah objek. Kemudian, terdapat *self parameter* yaitu *parameter* yang mengacu pada *class* itu sendiri. *Self parameter* ini harus dideklarasikan sebagai *parameter* pertama dari sebuah *method.* Perlu diperhatikan, aturan untuk penamaannya. Penamaan bebas namun harus diletakkan sebagai *parameter* pertama.

* + 1. **Function pada PHP**

Pada percobaan pertama, nyalakan Apache dari Xampp Control Panel dan buatlah *file* bernama modul4\_kel4.php dalam *folder* Xampp > htdocs.

<?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>";

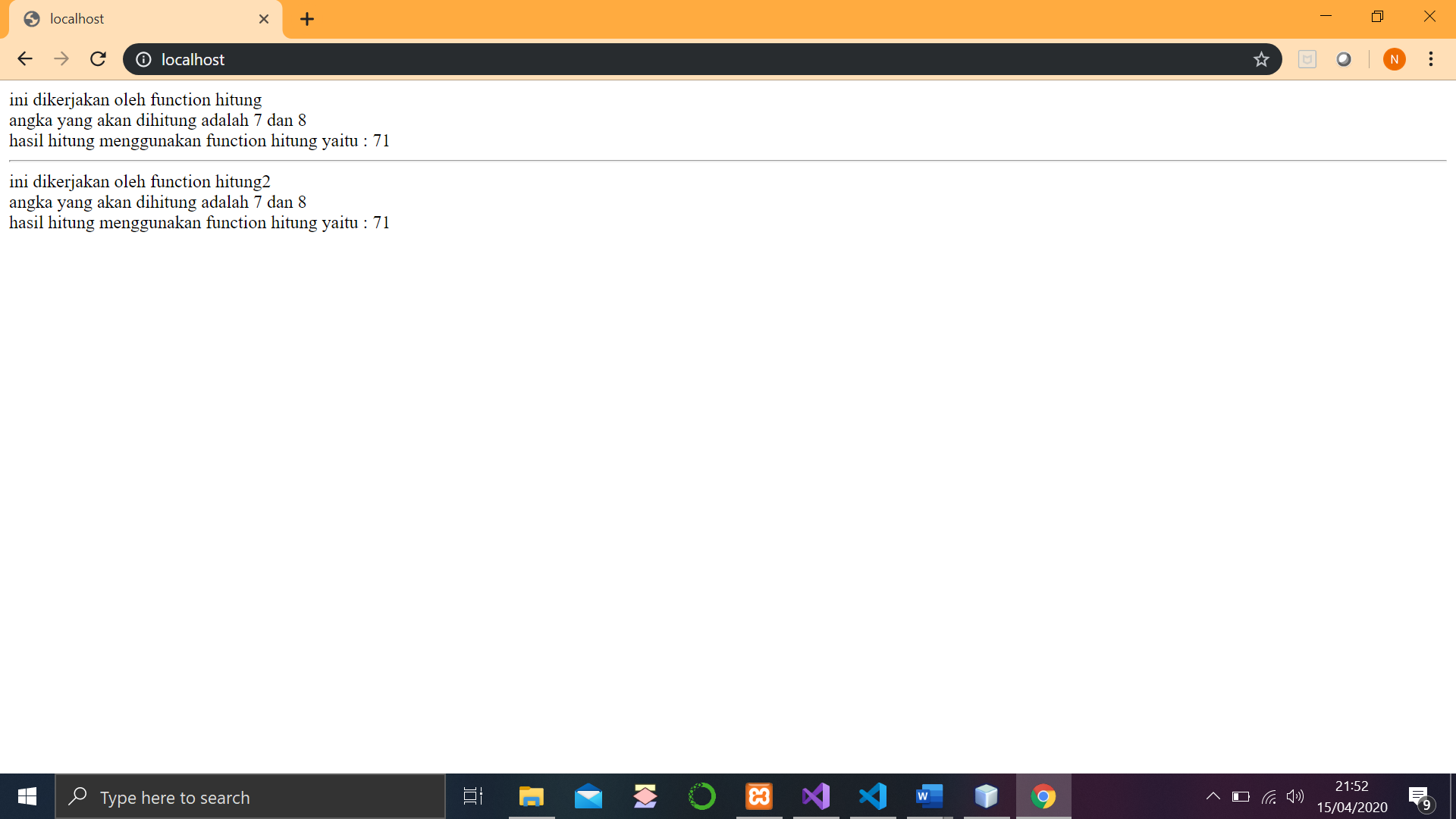
echo "hasil hitung menggunakan function hitung yaitu : $hasil";

}

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

hitung2($bil1, $bil2);

?>



Gambar 5.9 output function pada PHP

Pada pemrograman di atas menggunakan *return type* dan *non return type*. Pada program di atas menggunakan fungsi aritmatika (dalam matematika). Fungsi tersebut berupa penjumlahan dan perkalian. Dalam program tersebut terdapat bil1 dan bil2. Bil1 dan bil2 sudah ditentukan nilainya. Bil1 nilainya adalah 7 dan bil2 nilainya adalah 8. Operasi aritmatika-nya adalah bil1+bil2\*bil2. Sehingga hasil hitung menggunakan *function* hitung adalah 71, sesuai dengan operasi atau rumusnya.

* + 1. **Method pada PHP**

Pada percobaan kedua di bahasa pemrograman PHP yaitu *method*, terdapat sedikit perbedaan dengan bahasa pemrograman lain, OOP pada bahasa pemrograman PHP ini menggunakan notasi berbeda untuk menyatakan *method.* Notasi yang dimaksud yaitu notasi panah (->) bukan titik.

<?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();

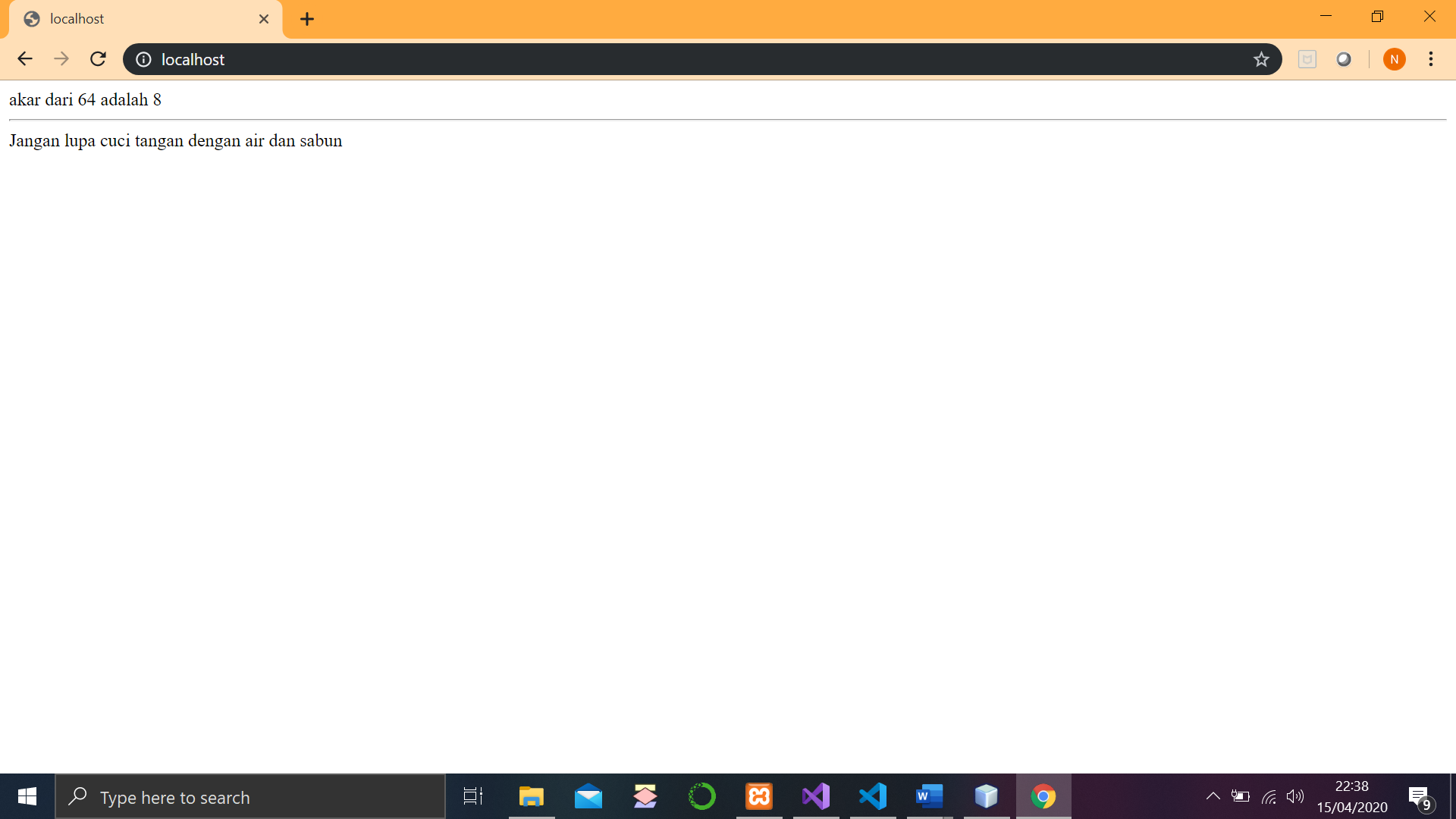
// OOP pada PHP menggunakan -> bukan .

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

echo "<hr>";

$objek -> staysafe('air', 'sabun');

?>



Gambar 5.10 output method dalam php

Pada program tersebut, menggunakan fungsi aritmatika (dalam matematika). Fungsi tersebut berupa akar pangkat dua. Dalam program tersebut terdapat angka dalam bentuk *float.* Operasi aritmatika-nya adalah sqrt($angka);. Dalam operasi tersebut, sqrt dalam bahasa pemrograman PHP mempunyai fungsi sebagai akar pangkat dua. Angka dalam pemrograman tersebut sudah ditentukan, yaitu 68. Sehingga *output* dari proses aritmatika tersebut adalah 8.

# ****Tugas****

using System;

namespace Modul\_4\_Kel4

{

class Program

{

static void Main(string[] args)

{

string input;

string k = "PRAKTIKUM DKP KELOMPOK 4 SHIFT 1"; Console.SetCursorPosition((Console.WindowWidth) / 2, Console.CursorTop);

Console.WriteLine(k);

Console.WriteLine(" ");

Console.WriteLine("NAMA PRAKTIKAN 1\t : NADIA");

Console.WriteLine("NIM\t : 21120119120023");

Console.WriteLine("NAMA PRAKTIKAN 2\t : AGRIAN RIZKY HIBATULLAH");

Console.WriteLine("NIM\t : 21120119130106");

Console.WriteLine(" ");

Class2 objek = new Class2();

{

string S = "SELAMAT DATANG DI TOKO ELEKRONIK SINAR JAYA. KAMI SIAP MELAYANI ANDA DENGAN SEPENUH HATI"; Console.SetCursorPosition((Console.WindowWidth) / 2, Console.CursorTop);

Console.WriteLine(S);

string L = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; Console.SetCursorPosition((Console.WindowWidth) / 2, Console.CursorTop);

Console.WriteLine(L);

Console.WriteLine();

}

int hp, byr, kmb, p, jml, i, j;

String[,] smp = new String[10, 4];

//MENU :

Console.WriteLine("-----------------------------------------------------------------------");

Console.WriteLine("|NO | NAMA ELEKTRONIK \t\t | HARGA(Rp)|");

Console.WriteLine("-----------------------------------------------------------------------");

Console.WriteLine("|1 | TELEVISI PANASONIC\t\t | 1,300,000|");

Console.WriteLine("|2 | KULKAS \t\t\t | 1,600,000|");

Console.WriteLine("|3 | RADIO\t\t\t | 2,750,000|");

Console.WriteLine("|4 | LAPTOP ASUS\t\t\t | 1,250,000|");

Console.WriteLine("|5 | KOMPUTER\t\t\t\t | 350,000 |");

Console.WriteLine("|6 | KIPAS ANGIN \t\t\t | 700,000 |");

Console.WriteLine("|7 | AC \t | 2,717,000|");

Console.WriteLine("|8 | JAM DINDING | 115,000 |");

Console.WriteLine("-----------------------------------------------------------------------");

for (i = 0; i < 9; i++)

Console.WriteLine("\n-----------------------------------------------------------------------------");

{

for (j = 0; j < 3; j++)

{

Console.Write(smp[i, j] + "\t|");

}

Console.WriteLine("\n-----------------------------------------------------------------------------"); //pindah baris

}

Console.WriteLine("MASUKKAN NO PILIHAN ELEKTRONIK YANG ANDA INGINKAN : ");

int.TryParse(Console.ReadLine(), out hp);

Console.WriteLine("MASUKKAN JUMLAH ELEKTRONIK YANG AKAN ANDA BELI : ");

int.TryParse(Console.ReadLine(), out jml);

Console.WriteLine("MASUKKAN NOMINAL PEMBAYARAN : ");

int.TryParse(Console.ReadLine(), out byr);

System.Console.WriteLine();

switch (hp)

{

case 1:

p = 1300000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[1, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[1, 1], jml); }

break;

case 2:

p = 1600000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[2, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[2, 1], jml); }

break;

case 3:

p = 2750000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[3, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[3, 1], jml); }

break;

case 4:

p = 1250000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[4, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[4, 1], jml); }

break;

case 5:

p = 350000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[5, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[5, 1], jml); }

break;

case 6:

p = 700000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[6, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[6, 1], jml); }

break;

case 7:

p = 2717000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[7, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[7, 1], jml); }

break;

case 8:

p = 115000;

kmb = byr - (p \* jml);

if (kmb >= 0)

{

objek.tx();

Console.WriteLine(smp[8, 1] + " SEJUMLAH " + jml + " BUAH");

Console.WriteLine("UANG KEMBALIAN ANDA " + kmb + " RUPIAH");

}

else { objek.k(smp[8, 1], jml); }

break;

default:

Console.WriteLine("MASUKAN SALAH");

break;

}

Console.ReadKey();

}

}

}

Main *Class*

using System;

using System.Collections.Generic;

using System.Text;

namespace Modul\_4\_Kel4

{

class Class2

{

public void tx()

{

Console.WriteLine("TERIMA KASIH TELAH BERBELANJA DI TOKO ELEKTRONIK SINAR JAYA ");

}

public void k(String nama, int jumlah)

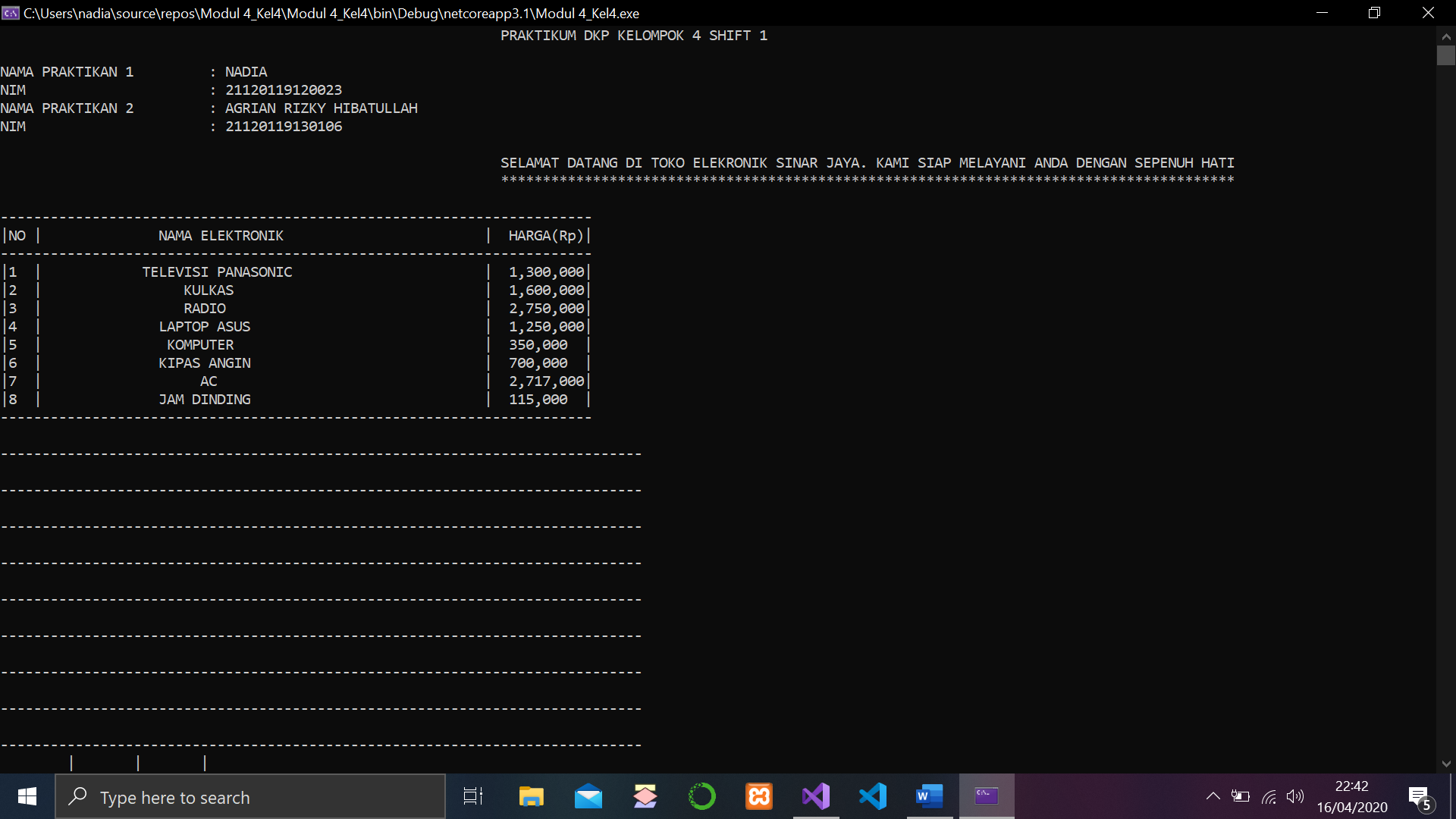
{

Console.WriteLine("MAAF UANG ADA TIDAK CUKUP UNTUK MEMBELI " + nama + "sejumlah" + jumlah + "buah");

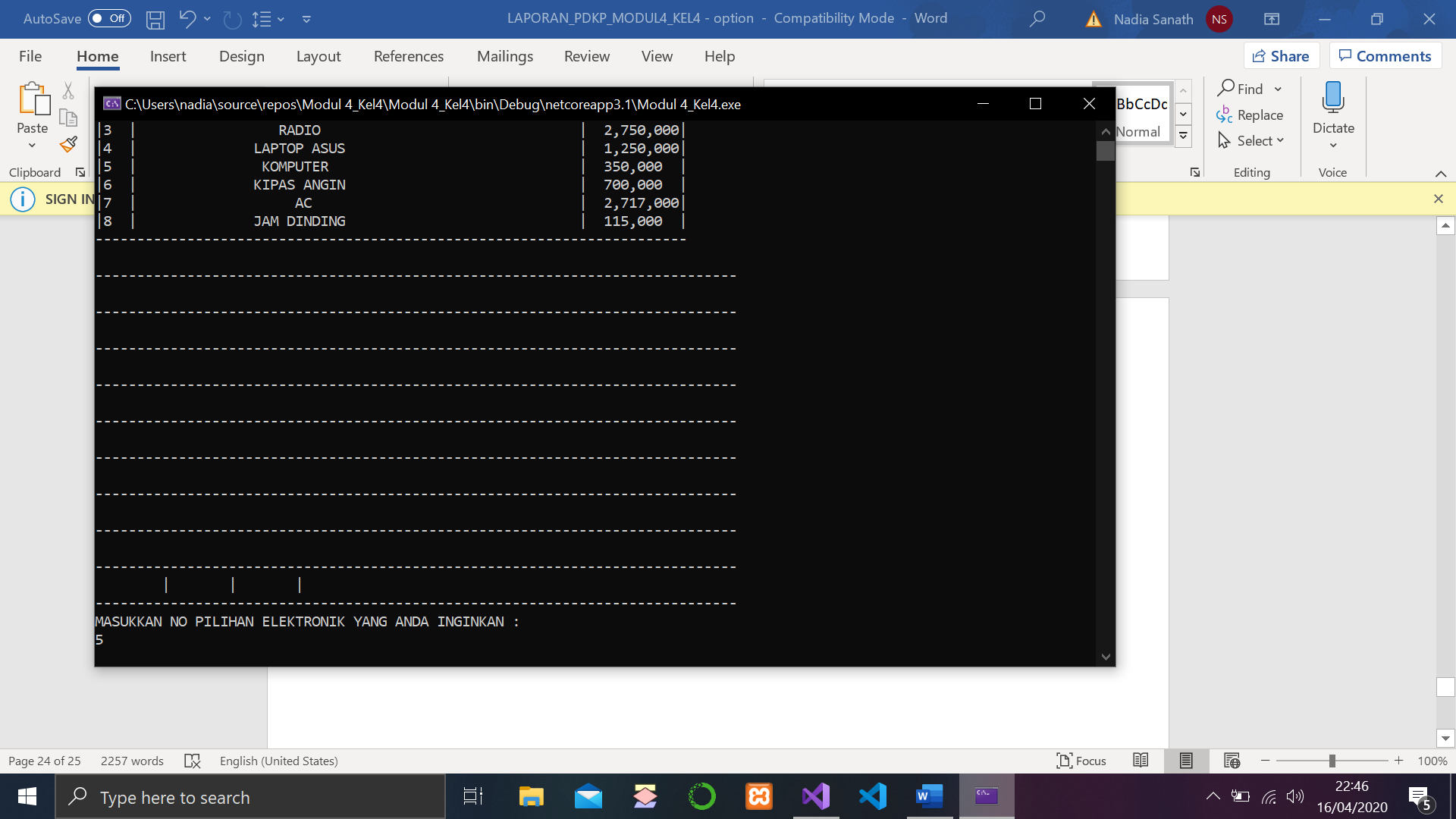
}

}

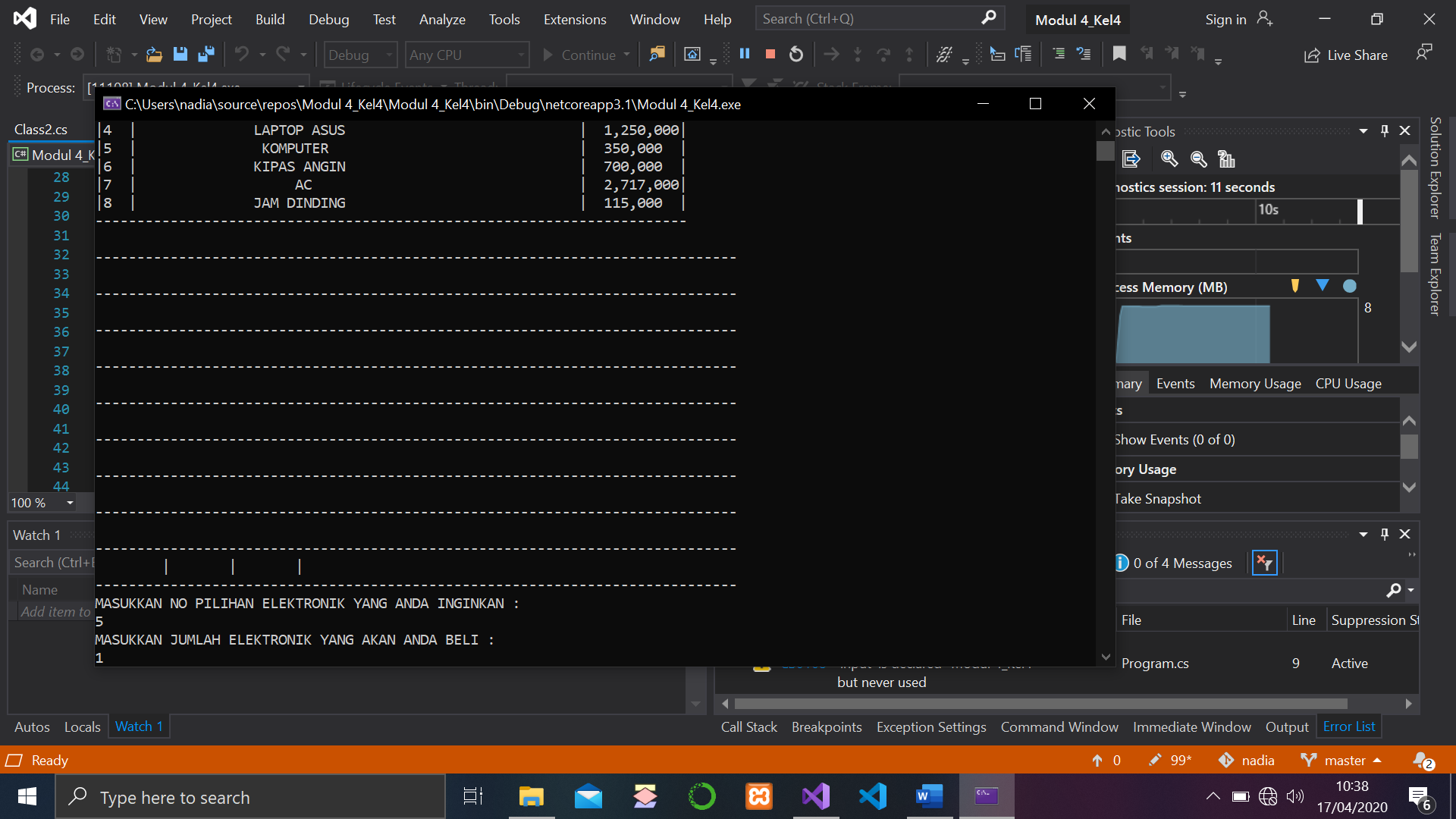
}



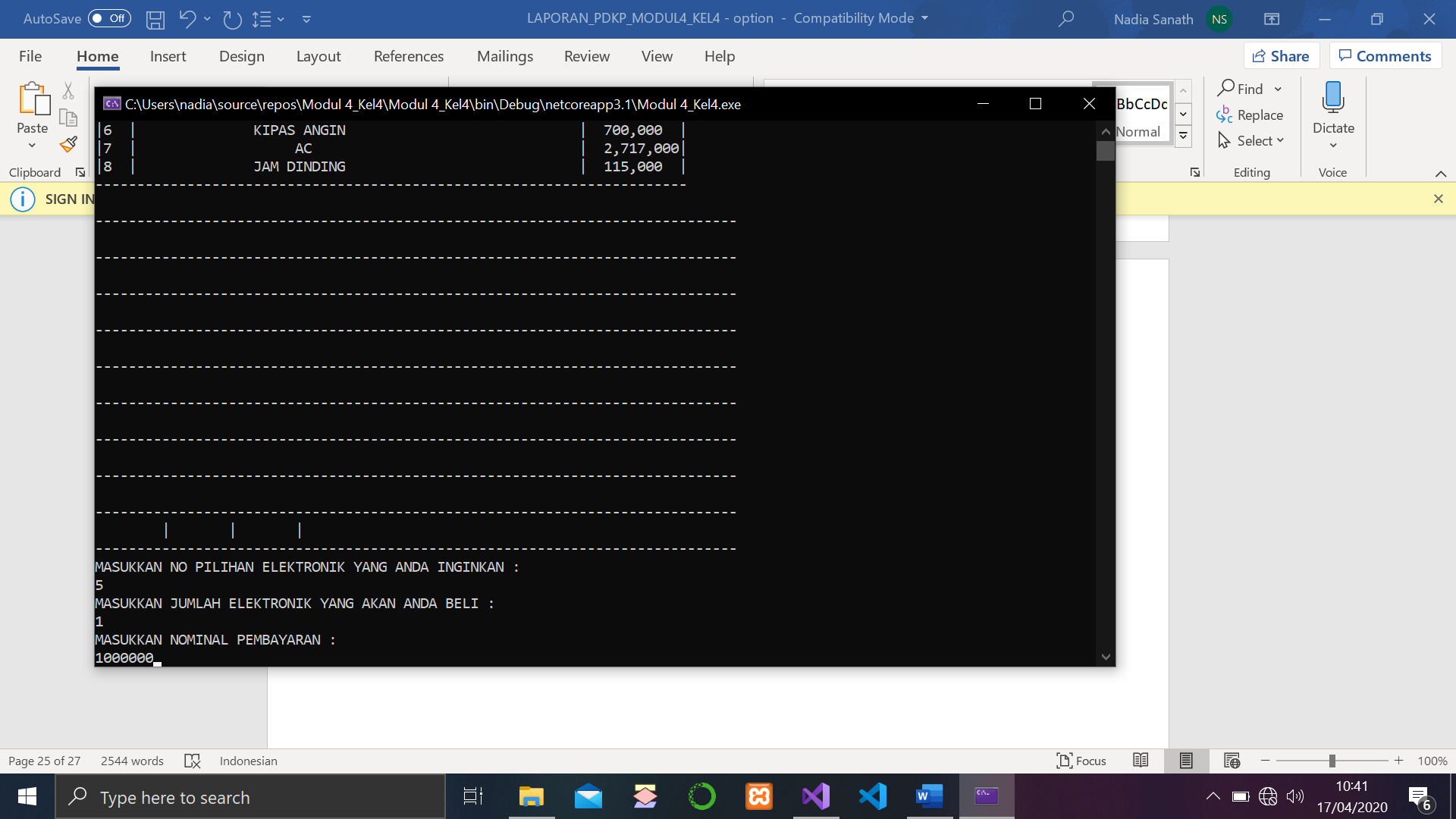
Gambar 5.11 output 1 dengan C#



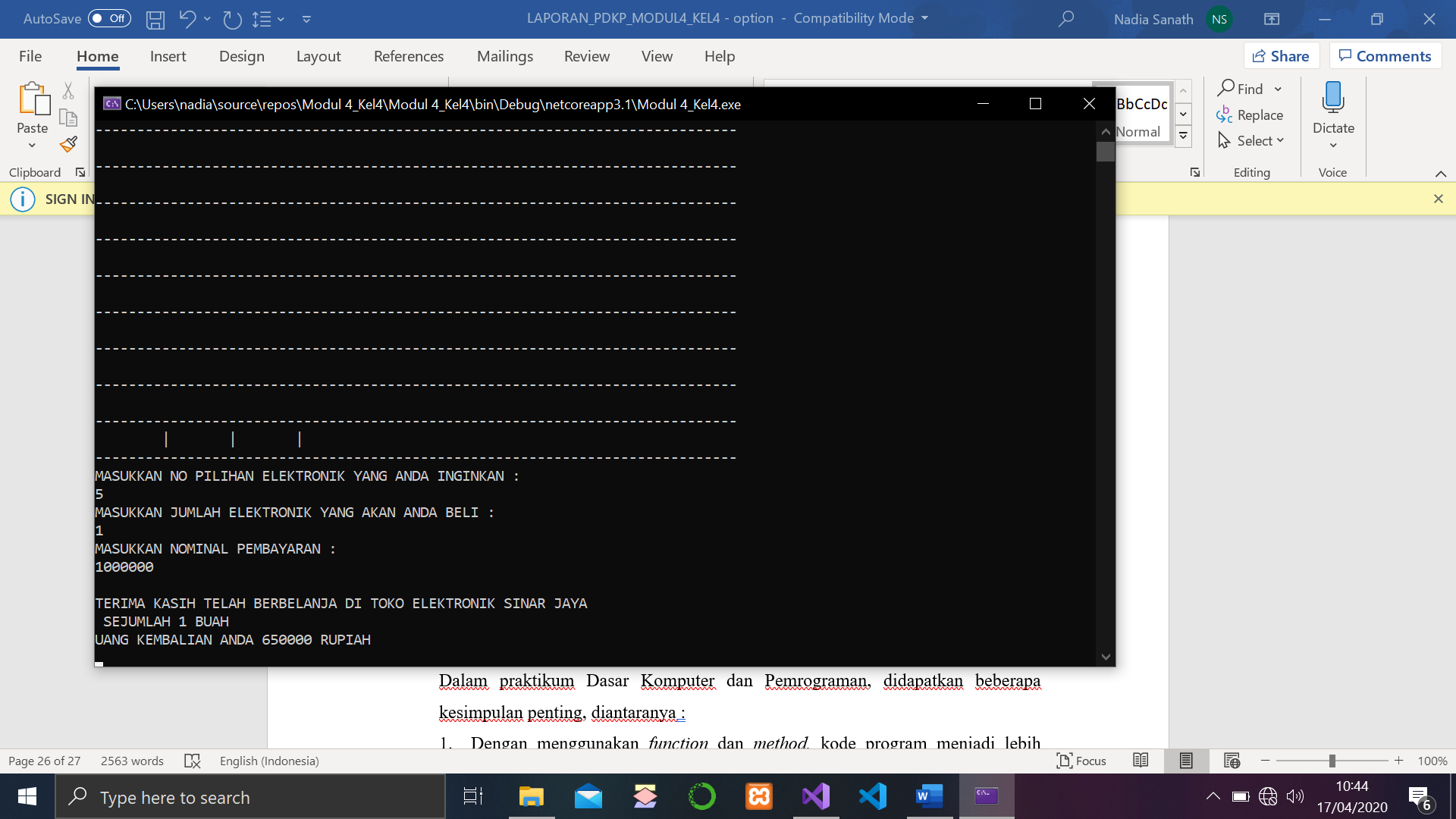
Gambar 5.12 output 2 dengan C#



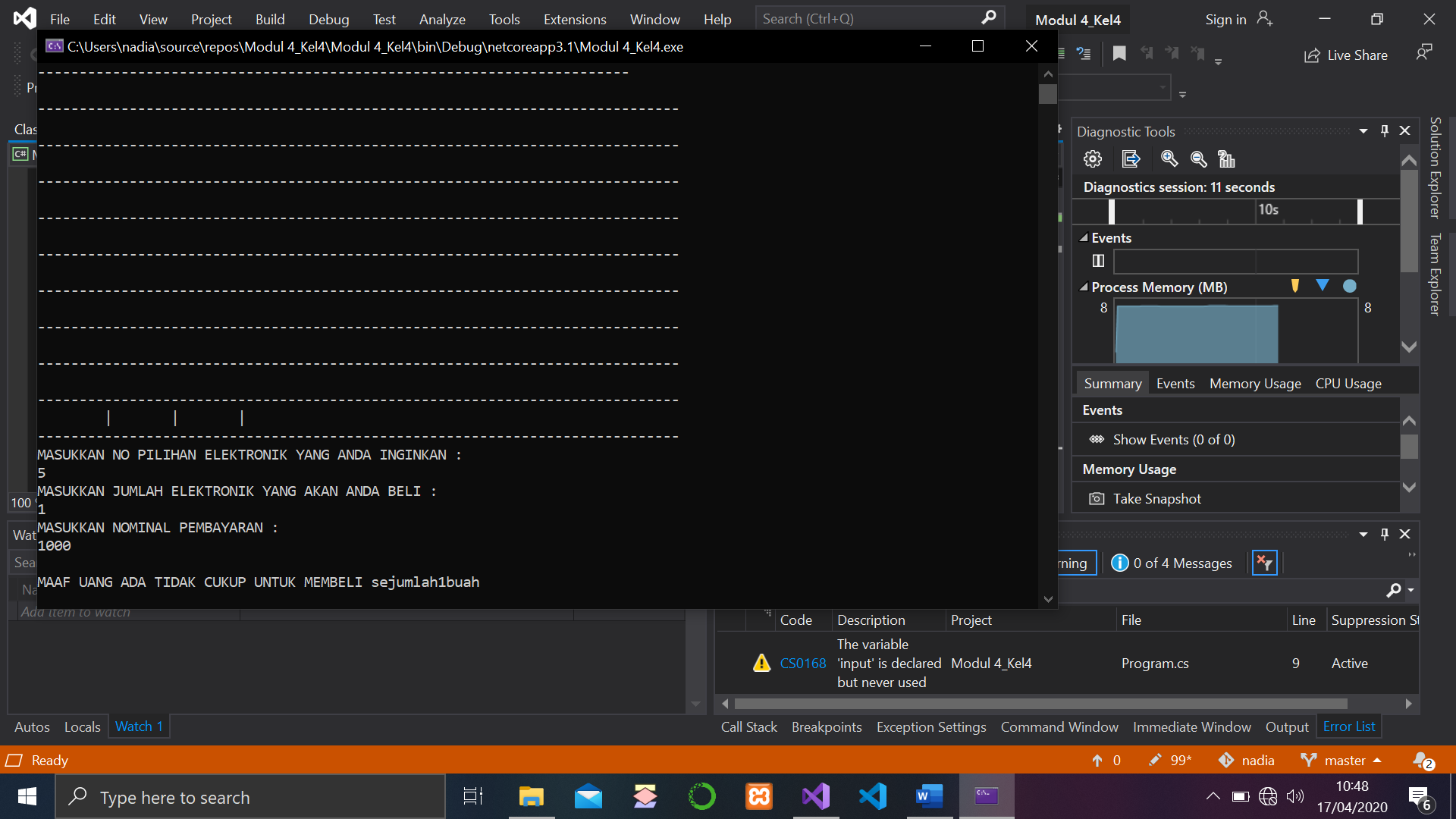
Gambar 5.13 output 3 dengan C#



Gambar 5.14 output 4 tugas



Gambar 5.15 output 5 tugas



Gambar 5.16 output 6 tugas

Pada program tersebut, kami buat dengan menggunakan bahasa pemrograman C# dengan nama folder Program Cs adalah Modul\_4\_Kel4 dan untuk *main class* nya dengan nama *Class2*. Pada program Cs tersebut diawali dengan static void Main(string[] args)yaitu merupakan sebuah *main method* atau yang dikenal sebagai *function* utama dalam pemrograman. Di mana ketika program tersebut dieksekusi yang pertama kali baca adalah static void Main(string[] args), setelah itu baru dilanjutkan pada pembacaan *function-function* lainnya yang ada di-*include* atau disertakan dalam *function* utama tersebut. Serta tidak dapat dirubah supaya dapat jalan programnya. Kemudian, *line* 13 yaitu string input; adalah tipe data untuk teks untuk angka, huruf, spasi, dan berbagai karakter, serta berfungsi untuk *identifier input*. Kemudian, Console.CursorTop berfungsi sebagai menengahkan kalimat, Console.WriteLine untuk menampilkan kalimat dan untuk berpindah baris ke kalimat selanjutnya. Int adalah tipe data untuk angka numerik yang tidak menggunakan koma, untuk tipe data 32 bit. Class2 objek = new Class2(); untuk memanggil yang ada atau terdapat di *Class2*

Pada pemrograman tersebut kami memberikan contoh *input*-an nya. Baik *input*-an yang benar maupun *input*-an yang salah. Pada saat menjalankan program tersebut, keluaran atau *output* pertama yaitu perintah untuk memasukkan no pilihan elektronik yang diinginkan. Kemudian, selanjutnya terdapat perintah untuk memasukkan jumlah elektronik atau barang yang akan dibeli. Apabila *input*-an yang diberikan salah, maka program akan berhenti.

Contoh penerapan pada program tersebut adalah pada awal perintah yaitu memasukkan no elektronik yang diinginkan, kami meng-*input* kan angka atau nomor 5 yaitu elektronik berupa komputer. Kemudian, perintah kedua yaitu perintah untuk meng-*input* kan jumlah elektronik atau barang (komputer) yang akan dibeli, kami meng-*input* kan angka atau no satu (1). Dan yang terakhir terdapat perintah untuk memasukkan atau meng-*input* kan jumlah nominal atau uang yang akan diberikan sebagai pembayaran barang elektronik, kami meng-*input* kan nominal 1000000. Sehingga, uang kembalian yang akan diberikan oleh penjual kepada konsumen adalah 650000, karena harga satu *unit* komputer tersebut adalah 350000.

Kami juga melakukan penerapan pada program kami dengan memasukkan atau meng-*input* kan angka atau nilai yang salah, sehingga program tidak dapat berjalan atau bereksekusi dengan baik dan lancar.

# ****Kesimpulan****

Dalam praktikum Dasar Komputer dan Pemrograman, didapatkan beberapa kesimpulan penting, diantaranya :

* + - 1. Dengan menggunakan *function* dan *method,* kode program menjadi lebih pendek, mudah dibaca, dan tentunya mudah untuk dipahami.
      2. Dengan menggunakan *function* dan *method,* program dalam diselesaikan dengan koordinasi yang baik dan mudah.
      3. Dengan menggunakan *function* dan *method,* sangat mudah dalam mencari kesalahan-kesalahan karena alur logika jelas dan sederhana.
      4. Fungsi-fungsi dapat menjadikan program mempunyai struktur yang jelas.
      5. Dengan memisahkan langkah-langkah detail ke satu atau lebih fungsi-fungsi, maka fungsi utama akan menjadi lebih pendek, kelas, dan mudah dimengerti.
      6. Fungsi-fungsi digunakan untuk menghindari penulisan program yang sama yang ditulis secara berulang-ulang. Langkah-langkah tersebut dapat dituliskan sekali saja secara terpisah dalam bentuk fungsi. Selanjutnya bagian program yang membutuhkan langkah-langkah ini tidak perlu selalu menuliskannya.
      7. Dengan menggunakan *function* dan *method,* akan mempermudah dokumentasi.
      8. Terdapat 2 fungsi dalam bahasa pemrograman, yaitu *return type* (mengembalikan suatu nilai baru dari proses yang dilakukannya) dan *non-return type* (tidak akan mengembalikan suatu nilai baru setelah pekerjaannya).
      9. Perbedaan antara *function* dan *method* adalah pada cara pemanggilannya. Fungsi dipanggil dengan langsung menyebutkan nama fungsi dan menyertakan *argument.* Sedangkan *method* dipanggil melalui *instance object* sebagai perwujudan dari suatu *class.*