**TUGAS PRAKTIKUM KONSEP PEMPROGRAMAN**

**JILID 9 part 4**



**Oleh :**

**Nama : Rosi Arif Mulyadi**

**NRP : 3121522021**

**Prodi : D3 Teknik Informatika PENS PSDKU Sumenep**

**Kelas : 1 ITA D3 Sumenep**

**Dosen :**

**Lusiana Agustien M.Kom**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**Praktikum 5 (4/5)**

**FUNGSI**

**TUJUAN**

1. Mengetahui perbedaan antara variabel lokal, eksternal, statis dan register.

**DASAR TEORI**

Penggolongan Variabel berdasarkan Kelas Penyimpanan Suatu variabel, di samping dapat digolongkan berdasarkan jenis/tipe data juga dapat diklasifikasikan berdasarkan kelas penyimpanan (storage class). Penggolongan berdasarkan kelas penyimpanan berupa :

* Variabel lokal
* Variabel eksternal
* Variabel statis
* Variabel register
* **Variabel Lokal**

Variabel lokal adalah variabel yang dideklarasikan dalam fungsi, dengan sifat :

* Secara otomatis diciptakan ketika fungsi dipanggil dan akan sirna (lenyap) ketika eksekusi terhadap fungsi berakhir.
* Hanya dikenal oleh fungsi tempat variabel tersebut dideklarasikan
* Tidak ada inisialisasi secara otomatis (saat variabel diciptakan, nilainya tak menentu). Dalam banyak literatur, variabel lokal disebut juga dengan variabel otomatis. Variabel yang termasuk dalam golongan ini bisa dideklarasikan dengan menambahkan kata kuci auto di depan tipe-data variabel. Kata kunci ini bersifat opsional, biasanya disertakan sebagai penjelas saja.
* **Variabel Eksternal**

Variabel eksternal merupakan variabel yang dideklarasikan di luar fungsi, dengan sifat :

* Dapat diakses oleh semua fungsi
* Kalau tak diberi nilai, secara otomatis diinisialisasi dengan nilai sama dengan nol.

Variabel eksternal haruslah dideklarasikan sebelum definisi fungsi yang akan mempergunakannya. Untuk memperjelas bahwa suatu variabel dalam fungsi merupakan variabel eksternal, di dalam fungsi yang menggunakannya dapat mendeklarasikan variabel itu kembali dengan menambahkan kata kunci extern di depan tipe data variabel.

Kalau dalam suatu program terdapat suatu variabel eksternal, suatu fungsi bisa saja menggunakan nama variabel yang sama dengan variabel eksternal, namun diperlakukan sebagai variabel lokal.

* **Variabel Statis**

Variabel statis dapat berupa variabel internal (didefinisikan di dalam fungsi) maupun variabel eksternal. Sifat variabel ini : void fung\_x(void) { int x; . . . } x adalah variabel lokal bagi fungsi fung\_x() 71

* Kalau variabel statis bersifat internal, maka variabel hanya dikenal oleh fungsi tempat variabel dideklarasikan
* Kalau variabel statis bersifat eksternal, maka variabel dapat dipergunakan oleh semua fungsi yang terletak pada file yang sama, tempat variabel statis dideklarasikan
* Berbeda dengan variabel lokal, variabel statis tidak akan hilang sekeluarnya dari fungsi (nilai pada variabel akan tetap diingat).
* Inisialisasi akan dilakukan hanya sekali, yaitu saat fungsi dipanggil yang pertama kali.

Kalau tak ada inisialisasi oleh pemrogram secara otomatis akan diberi nilai awal nol Variabel statis diperoleh dengan menambahkan kata kunci static di depan tipe data variabel.

* **Variabel Register**

Variabel register adalah variabel yang nilainya disimpan dalam register dan bukan dalam memori RAM. Variabel yang seperti ini hanya bisa diterapkan pada variabel yang lokal atau parameter formal, yang bertipe char atau int. Variabel register biasa diterapkan pada variabel yang digunakan sebagai pengendali loop. Tujuannya untuk mempercepat proses dalam loop. Sebab variabel yang dioperasikan pada register memiliki kecepatan yang jauh lebih tinggi daripada variabel yang diletakkan pada RAM.

**TUGAS PERCOBAAN**

**Lakukan percobaan-percobaan untuk bisa menjawab semua pertanyaan di bawah ini, analisislah dan tuliskan alasannya**

1. Adakah sesuatu yang salah pada sebuah fungsi yang tidak mempunyai return value ? Jelaskan analisismu tentang sebuah fungsi yang tidak memiliki return value!

Jawab :

#include <stdio.h>

#include <stdlib.h>

void

fungsi(void);

main()

{

fungsi();

}

void fungsi(void)

{

puts("Tidak ada masalah");

}

Analisa :

: Tidak. Setiap fungsi tidak harus memiliki return value. Hanya fungsi yang digunakan kembali dalam fungsi main() atau fungsi lain sebagai suatu nilai output yang harus memiliki return value.

2. Apakah yang terjadi jika sebuah fungsi memberikan return value tetapi tidak diassign ke variabel apapun ?

Jawab :

#include <stdio.h>

#include <stdlib.h>

int fungsi(void);

main()

{

fungsi(); //tanpa assignment printf("nilai dari return value adalah %d\n", fungsi());

printf("nilai dari return value adalah %d\n", fungsi());

//output return value

}

int fungsi(void)

{

return (7); //return value

}

Analisa :

Tidak terjadi apa-apa. Fungsi tetap menjalankan proses di dalamnya. Tetapi, hasil dari proses itu tidak bisa digunakan dalam fungsi main() atau fungsi lain.

3. Apakah yang terjadi jika sebuah fungsi diassign ke sebuah variabel padahal fungsi tersebut tidak memiliki return value ?

Jawab :

#include <stdio.h>

#include <stdlib.h>

void

fungsi(int, int);

main()

{

int a = 5, b = 10, x;

x = fungsi(a, b);

printf("%d\n", x);

}

void fungsi(m, n) //tanpa return value

{

m++;

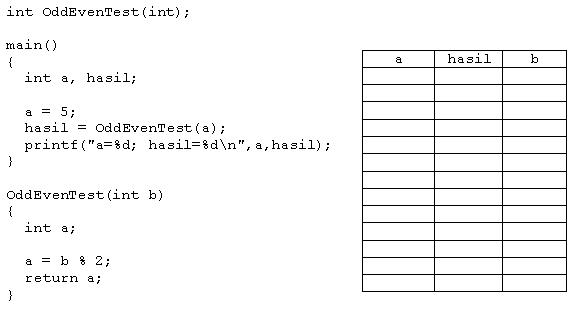
n--;

}

Analisa :

Selama tipe data fungsi tersebut bukan void, fungsi tetap bisa memberi return value tanpa deklarasi return pada fungsi secara eksplisit. Dengan syarat ada suatu variable dalam fungsi itu yang bertipe data sama dengan return fungsi tersebut dalam deklarasinya.

4. Trace secara manual semua program di bawah ini baris per barisnya, dan tampilkan nilai semua variabel pada setiap baris prosesnya. Selain itu, tebaklah tampilkan keluaran programnya

a. 

Jawab :

#include <stdio.h>

#include <stdlib.h>

int OddEvenTest(int);

main()

{

int a, hasil;

a = 5;

hasil = OddEvenTest(a);

printf("a=%d; hasil=%d\n",a,hasil);

}

OddEvenTest(int b)

{

int a;

a = b % 2;

return a;

}

|  |  |  |
| --- | --- | --- |
| A | hasil | b |
| tak tentu | tak tentu | undefined |
| tak tentu | tak tentu | undefined |
| 5 | tak tentu | undefined |
| 5 | 1 | Undefined |
| 5 | 1 | Undefined |
| undefined | undefined | undefined |
| undefined | undefined | undefined |
| undefined | undefined | 5 |
| undefined | undefined | 5 |
| tak tentu | undefined | 5 |
| 1 | undefined | 5 |
| 1 | undefined | 5 |
| undefined | undefined | undefined |

b.

void demo(void);

main()

{

int i=0;

while(i < 3)

{

demo();

i++;

}

}

void demo(void)

{

auto int var\_auto = 0;

static int var\_static = 0;

printf("auto = %d, static = %d\n", var\_auto, var\_static);

++var\_auto;

++var\_static;

}

|  |  |  |
| --- | --- | --- |
| I | Var\_auto | Var\_static |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Jawab :

#include <stdio.h>

#include <stdlib.h>

void demo(void);

main()

{

int i=0;

while(i < 3)

{

demo();

i++;

}

}

void demo(void)

{

auto int var\_auto = 0;

static int var\_static = 0;

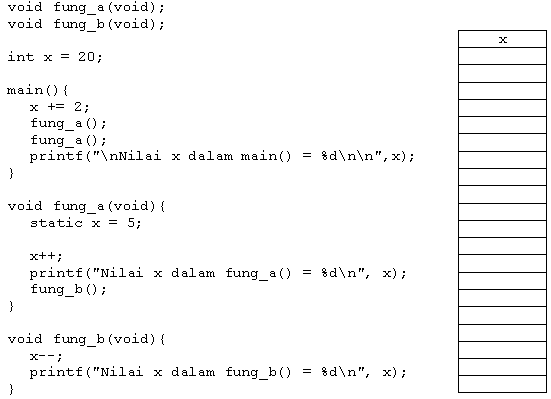
printf("auto = %d, static = %d\n", var\_auto, var\_static);

++var\_auto;

++var\_static;

}

|  |  |  |
| --- | --- | --- |
| I | Var\_auto | Var\_static |
| 0 | undefined | Undefined |
| 0 | undefined | Undefined |
| 0 | 1 | 1 |
| 1 | undefined | undefined |
| 1 | undefined | undefined |
| undefined | undefined | undefined |
| undefined | undefined | undefined |
| undefined | undefined | undefined |
| undefined | undefined | undefined |
| undefined | 0 | undefined |
| undefined | 0 | 0 |
| undefined | 0 | 0 |
| undefined | 0 | 0 |
| undefined | 0 | 0 |
| undefined | 1 | 0 |
| undefined | 1 | 1 |
| undefined | undefined | undefined |

c. 

Jawab :

#include <stdio.h>

#include <stdlib.h>

void fung\_a(void);

void fung\_b(void);

int x = 20;

main()

{

x += 2;

fung\_a();

fung\_a();

printf("\nNilai x dalam main()=%d\n\n",x);

}

void fung\_a(void)

{

static x = 5;

x++;

printf("Nilai x dalam fung\_a() = %d\n", x);

fung\_b();

}

void fung\_b(void)

{

x--;

printf("Nilai x dalam fung\_b() = %d\n", x);

}

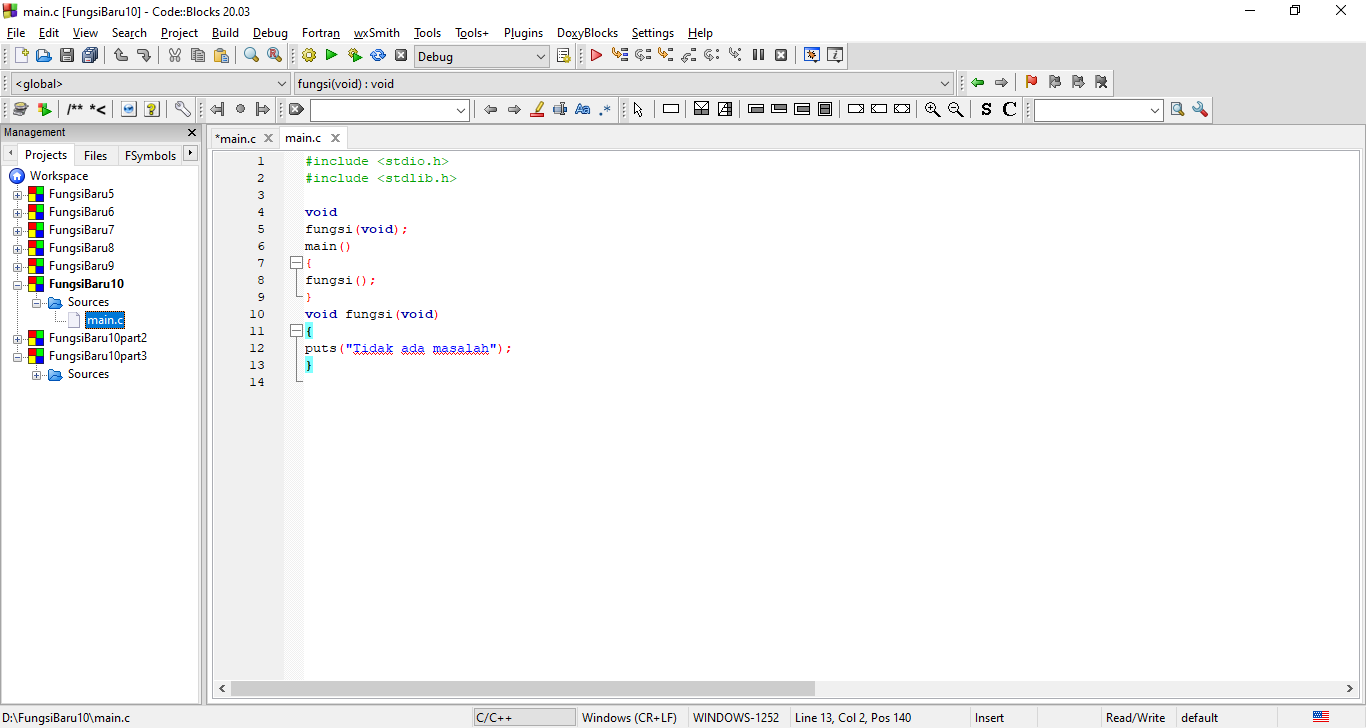
|  |
| --- |
| X |
| 20 |
| 20 |
| 20 |
| 20 |
| 22 |
| 6 dan 21 |
| 7 dan 20 |
| 20 |
| 20 |
| 20 |
| 20 |
| 20 |
| 5 |
| 5 |
| 6 |
| 6 |
| 22 |
| 22 |
| 22 |
| 22 |
| 22 |
| 21 |
| 21 |
| 21 |

**LAPORAN RESMI**

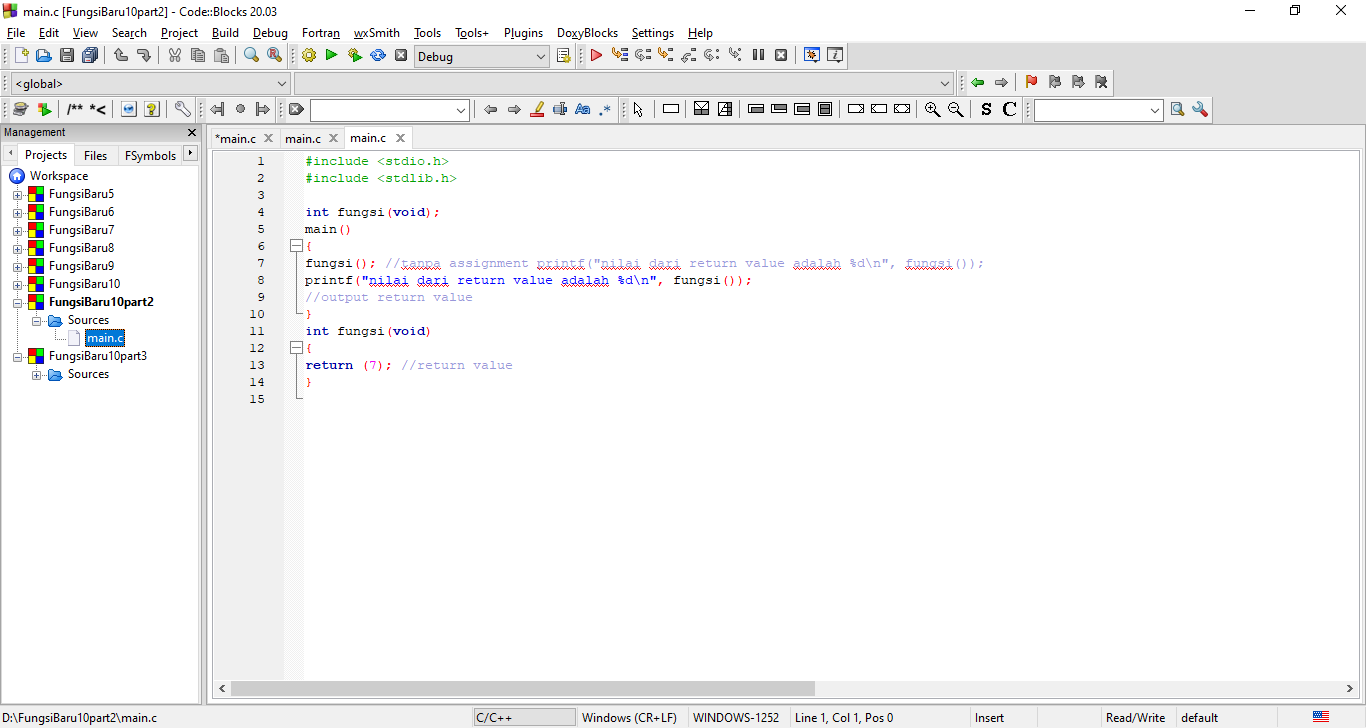
1. Tulis listing program dari semua percobaan yang dilakukan.

Jawab :

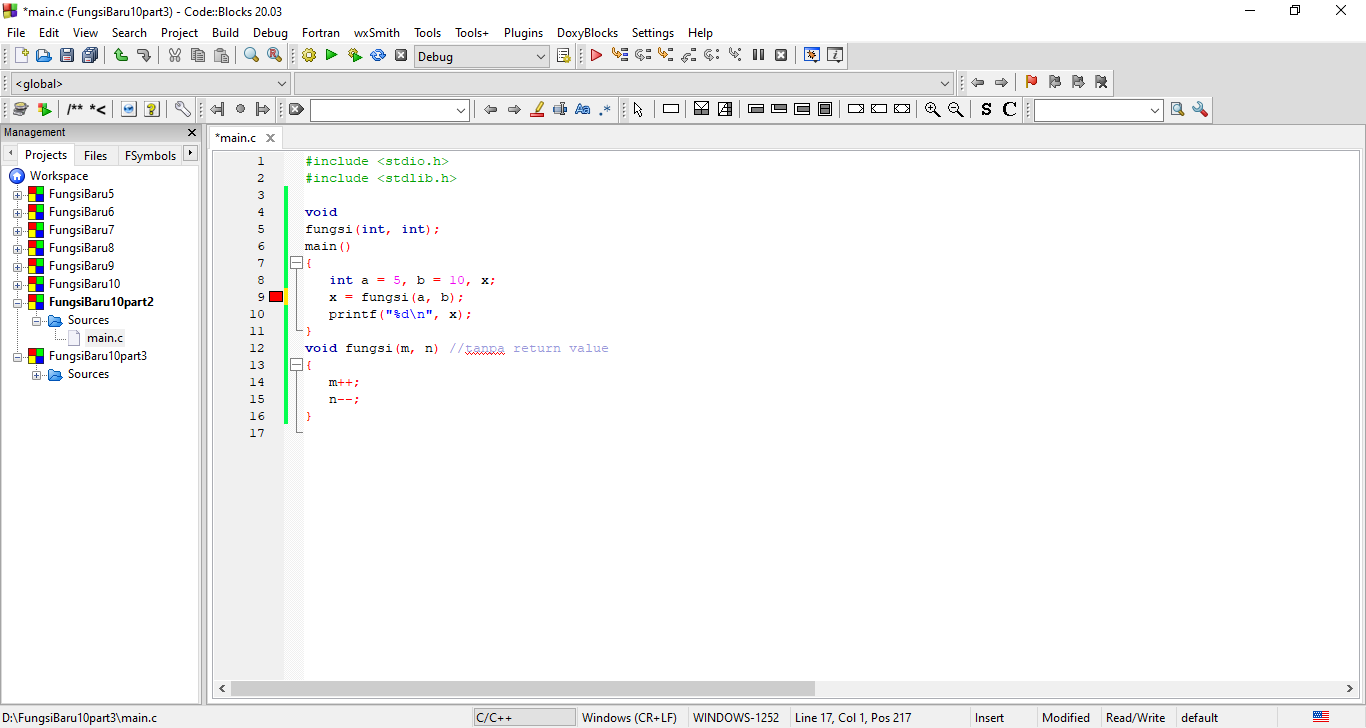
A) Listing Program :



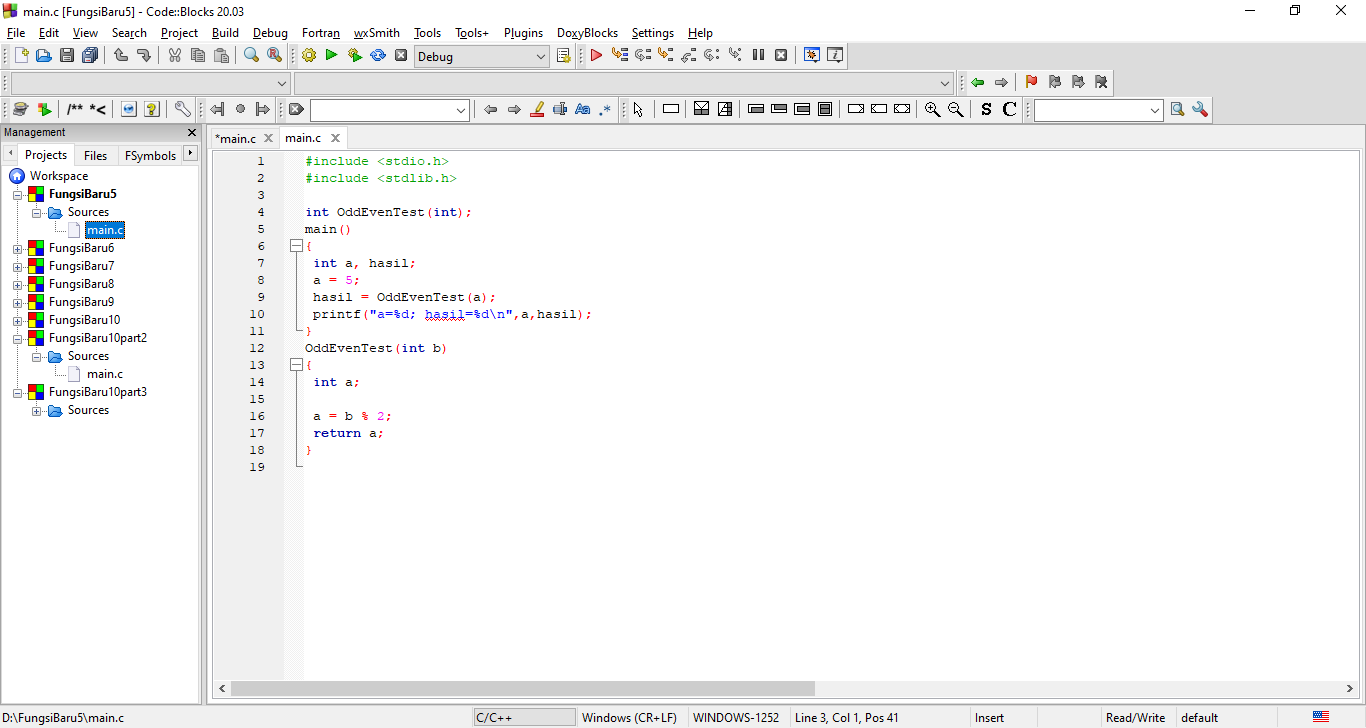
B) Listing Program :



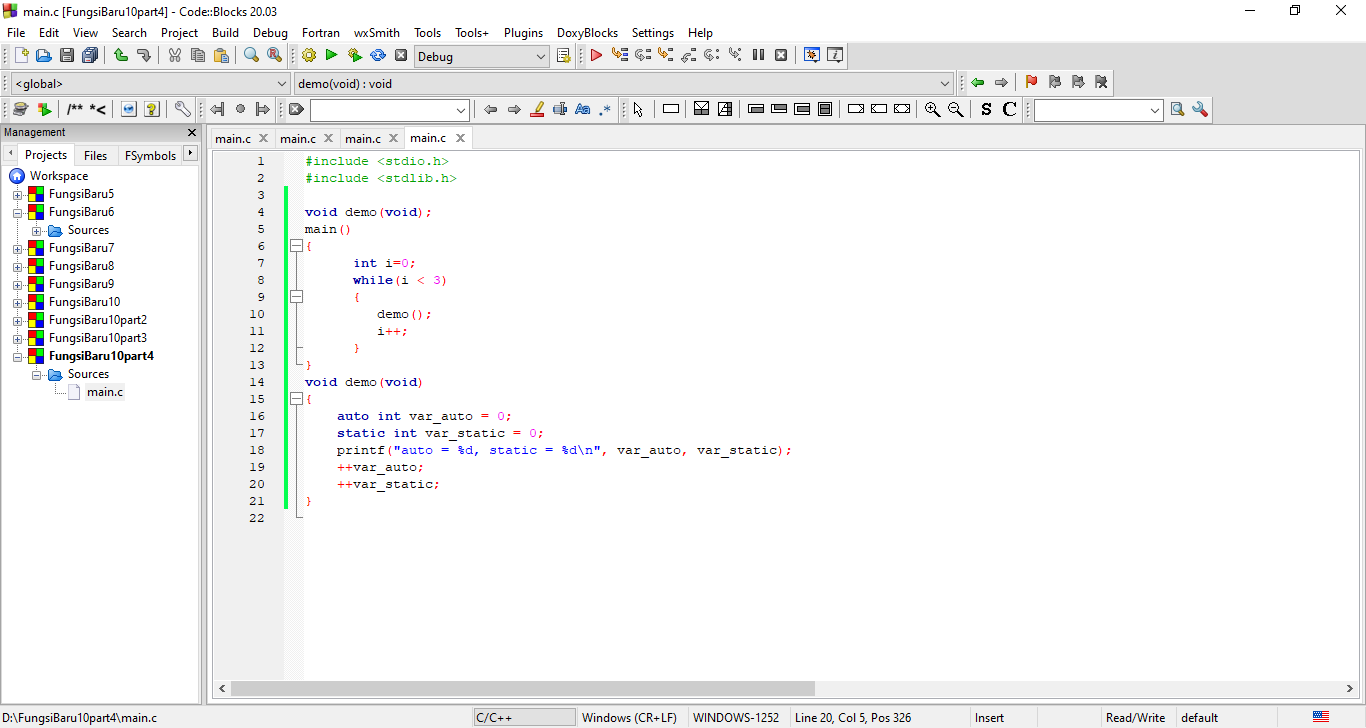
C) Listing Program :



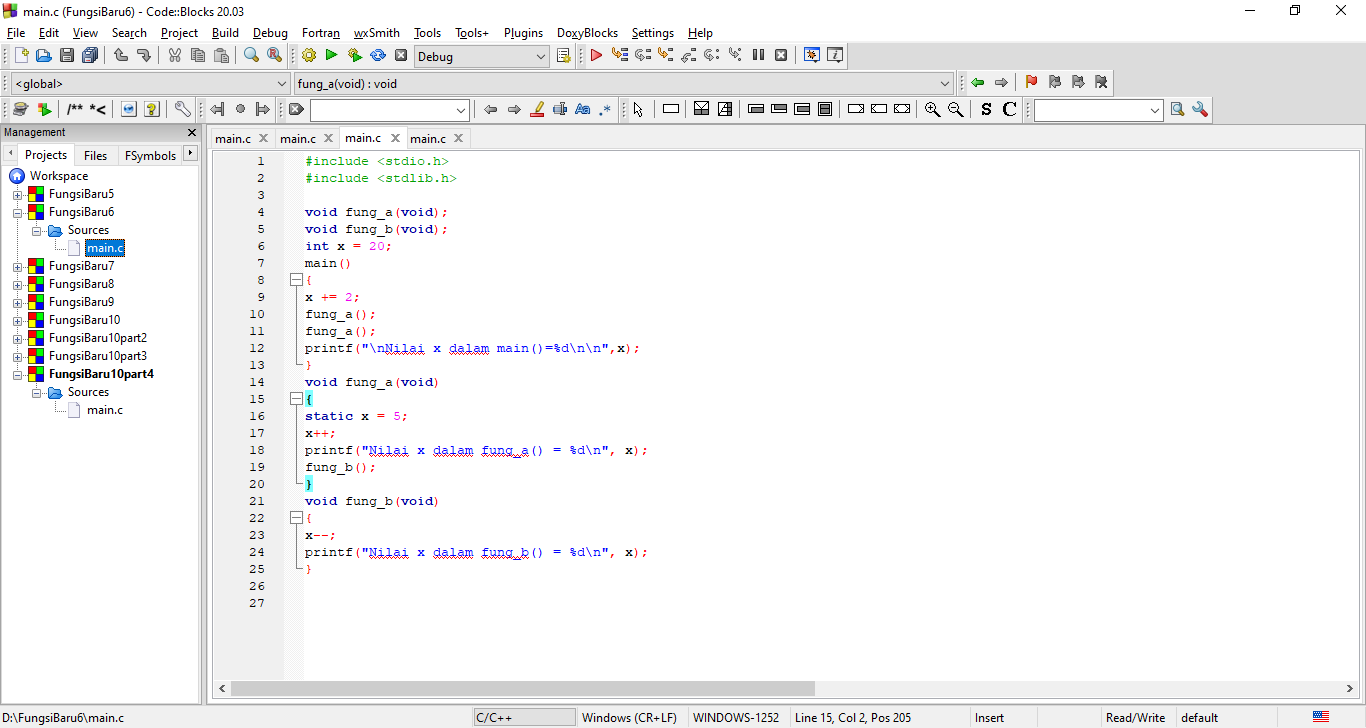
D) 1. Listing Program :



2. Listing Program :



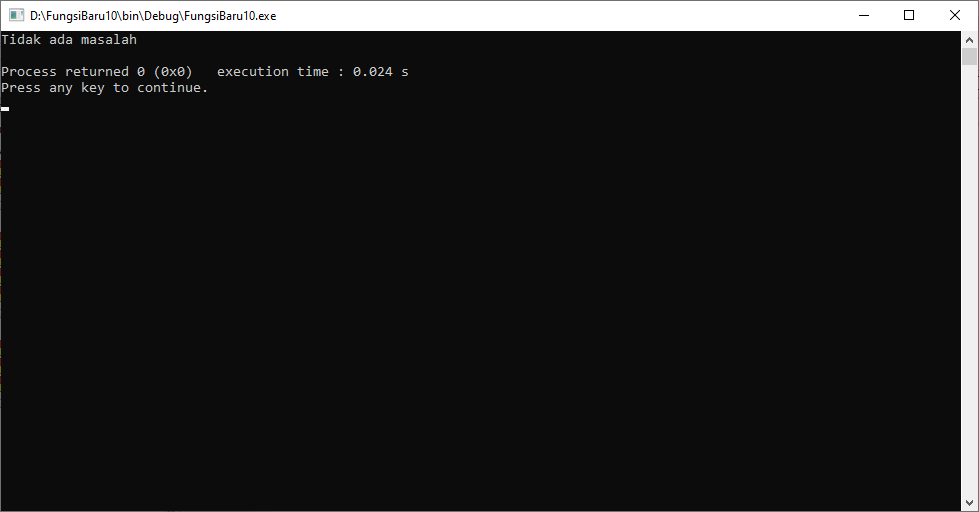
3. Listing Program :



2. Kemudian tuliskan outputnya. Terangkan mengapa demikian.

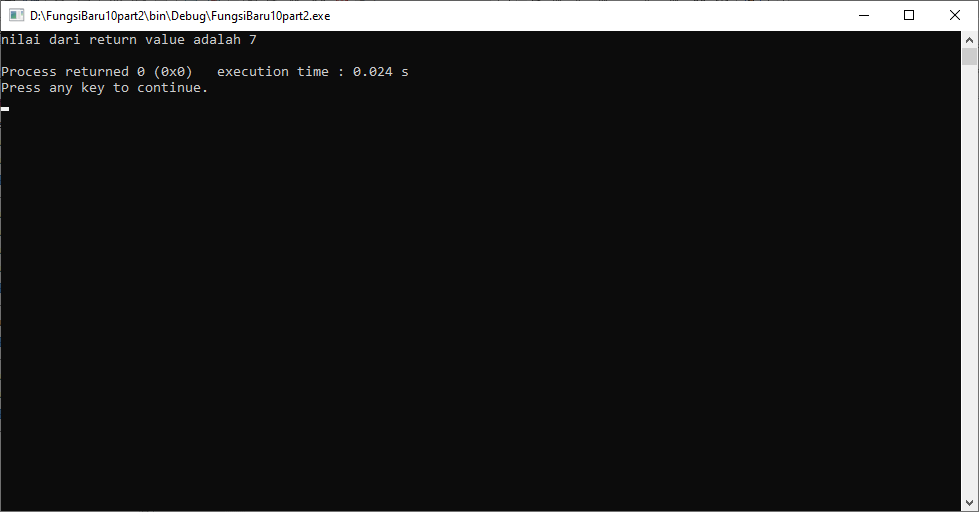
Jawab :

A) Output :



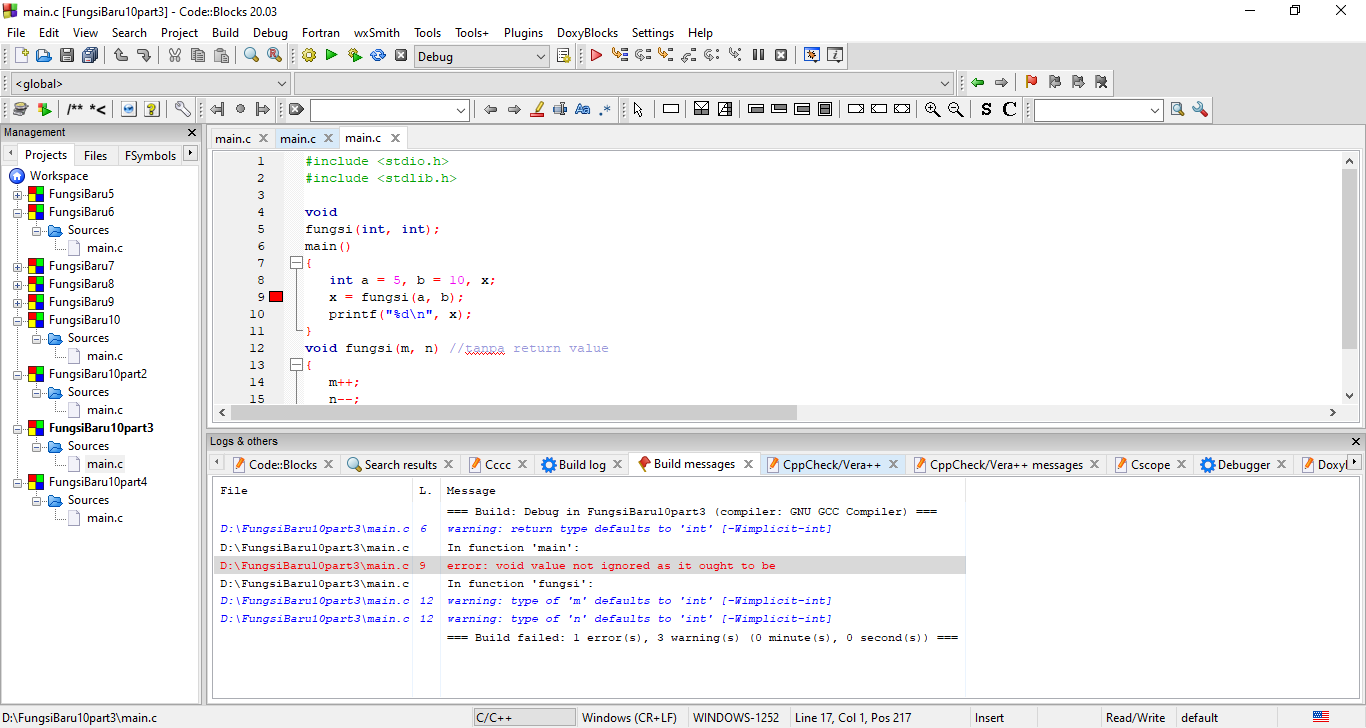
Alasan : Setiap fungsi tidak harus memiliki return value. Hanya fungsi yang digunakan kembali dalam fungsi main() atau fungsi lain sebagai suatu nilai output yang harus memiliki return value.

B) Output :



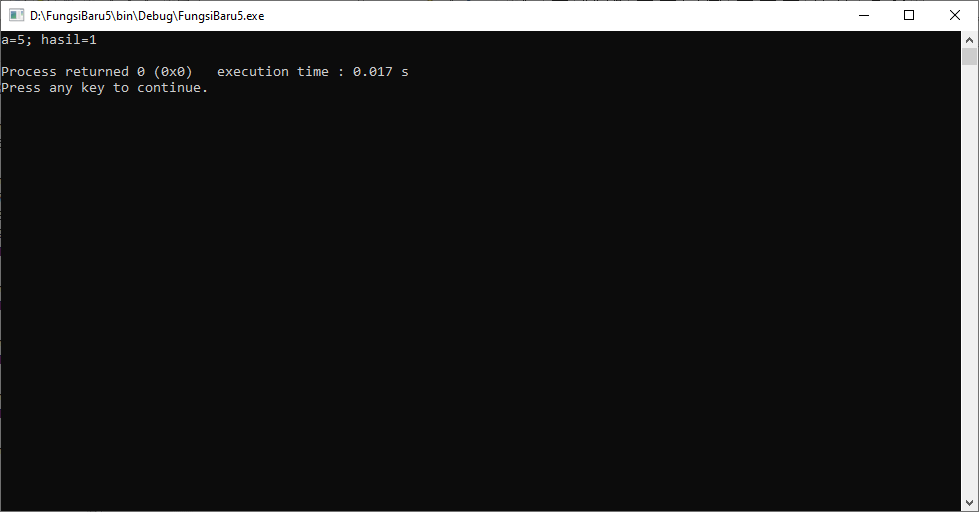
Alasan : Fungsi tetap menjalankan proses di dalamnya. Tetapi, hasil dari proses itu tidak bisa digunakan dalam fungsi main() atau fungsi lain.

C) Output :



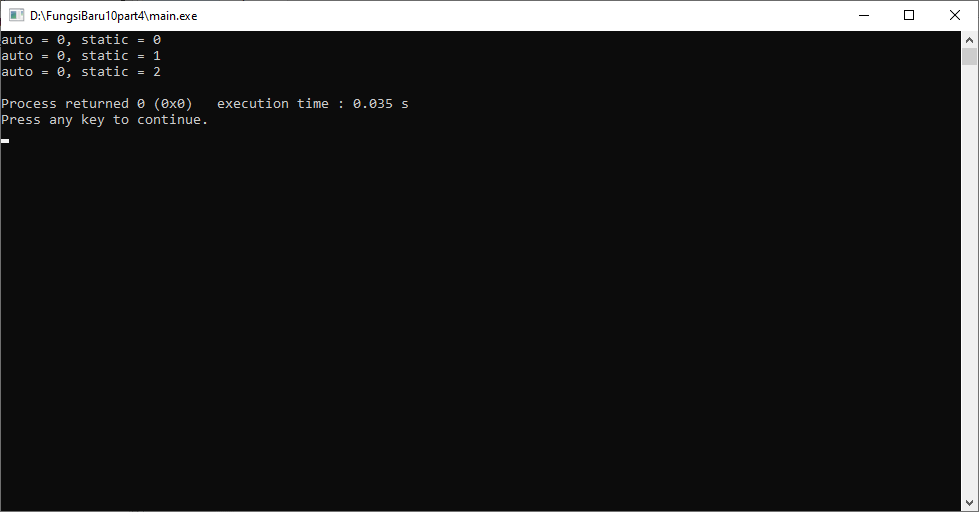
Alasan : Selama tipe data fungsi tersebut bukan void, fungsi tetap bisa memberi return value tanpa deklarasi return pada fungsi secara eksplisit. Dengan syarat ada suatu variable dalam fungsi itu yang bertipe data sama dengan return fungsi tersebut dalam deklarasinya.

D) 1. Output :



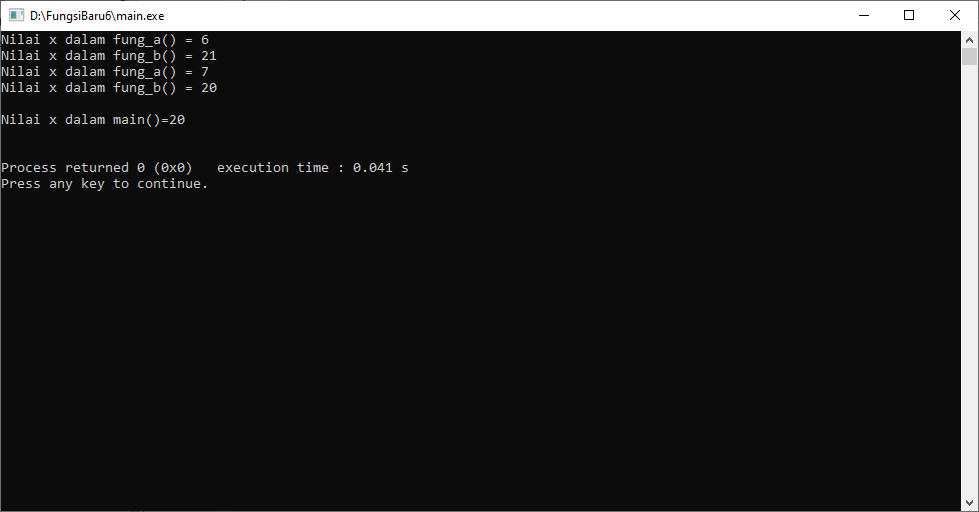
Alasan : -

2. Output :



Alasan : -

3. Output :



Alasan : -