

# **TUGAS PRAKTIKUM KONSEP PEMROGRAMAN**

## **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 kunci 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 memergunakannya. 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 sekluarnya 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.

```

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

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



```
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);
}
```

[illegible]

```
#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 :

The screenshot shows the Code::Blocks IDE interface. The title bar indicates the file is 'main.c [FungsiBaru10] - Code::Blocks 20.03'. The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar contains icons for file operations, compilation, and debugging. The 'Management' pane on the left shows a project structure with 'FungsiBaru10' selected, containing 'main.c'. The main editor displays the following C code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void
5 fungsi(void);
6 main()
7 {
8     fungsi();
9 }
10 void fungsi(void)
11 {
12     puts("Tidak ada masalah");
13 }
14
```

The status bar at the bottom shows the file path 'D:\FungsiBaru10\main.c', the compiler 'C/C++', the window title 'Windows (CR-LF)', the window type 'WINDOWS-1252', the cursor position 'Line 13, Col 2, Pos 140', and the editor mode 'Insert'.

## B) Listing Program :

The screenshot shows the Code::Blocks IDE interface. The title bar indicates the file is 'main.c [FungsiBaru10part2] - Code::Blocks 20.03'. The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar contains icons for file operations, compilation, and debugging. The 'Management' pane on the left shows a project structure with 'FungsiBaru10part2' selected, containing 'main.c'. The main editor displays the following C code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int fungsi(void);
5 main()
6 {
7     fungsi(); //sama assignment printf("nilai dari return value adalah %d\n", fungsi());
8     printf("nilai dari return value adalah %d\n", fungsi());
9     //output return value
10 }
11 int fungsi(void)
12 {
13     return (7); //return value
14 }
15
```

The status bar at the bottom shows the file path 'D:\FungsiBaru10part2\main.c', the compiler 'C/C++', the window title 'Windows (CR-LF)', the window type 'WINDOWS-1252', the cursor position 'Line 1, Col 1, Pos 0', and the editor mode 'Insert'.

## C) Listing Program :

\*main.c (FungsiBaru10part3) - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Management

Projects Files FSymbols

Workspace

- FungsiBaru5
- FungsiBaru6
- FungsiBaru7
- FungsiBaru8
- FungsiBaru9
- FungsiBaru10
- FungsiBaru10part2
  - Sources
  - main.c
- FungsiBaru10part3
  - Sources

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void
5  funksi(int, int);
6  main()
7  {
8      int a = 5, b = 10, x;
9      x = funksi(a, b);
10     printf("%d\n", x);
11 }
12 void funksi(m, n) //anggap return value
13 {
14     m++;
15     n--;
16 }
17

```

D:\FungsiBaru10part3\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 17, Col 1, Pos 217 Insert Modified Read/Write default

## D) 1. Listing Program :

main.c (FungsiBaru5) - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Management

Projects Files FSymbols

Workspace

- FungsiBaru5
  - Sources
  - main.c
- FungsiBaru6
- FungsiBaru7
- FungsiBaru8
- FungsiBaru9
- FungsiBaru10
- FungsiBaru10part2
  - Sources
  - main.c
- FungsiBaru10part3
  - Sources

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int OddEvenTest(int);
5  main()
6  {
7      int a, hasil;
8      a = 5;
9      hasil = OddEvenTest(a);
10     printf("a=%d; hasil=%d\n", a, hasil);
11 }
12 OddEvenTest(int b)
13 {
14     int a;
15
16     a = b % 2;
17     return a;
18 }
19

```

D:\FungsiBaru5\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 3, Col 1, Pos 41 Insert Read/Write default

## 2. Listing Program :

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void demo(void);
5 main()
6 {
7     int i=0;
8     while(i < 3)
9     {
10         demo();
11         i++;
12     }
13 }
14 void demo(void)
15 {
16     auto int var_auto = 0;
17     static int var_static = 0;
18     printf("auto = %d, static = %d\n", var_auto, var_static);
19     ++var_auto;
20     ++var_static;
21 }
22
```

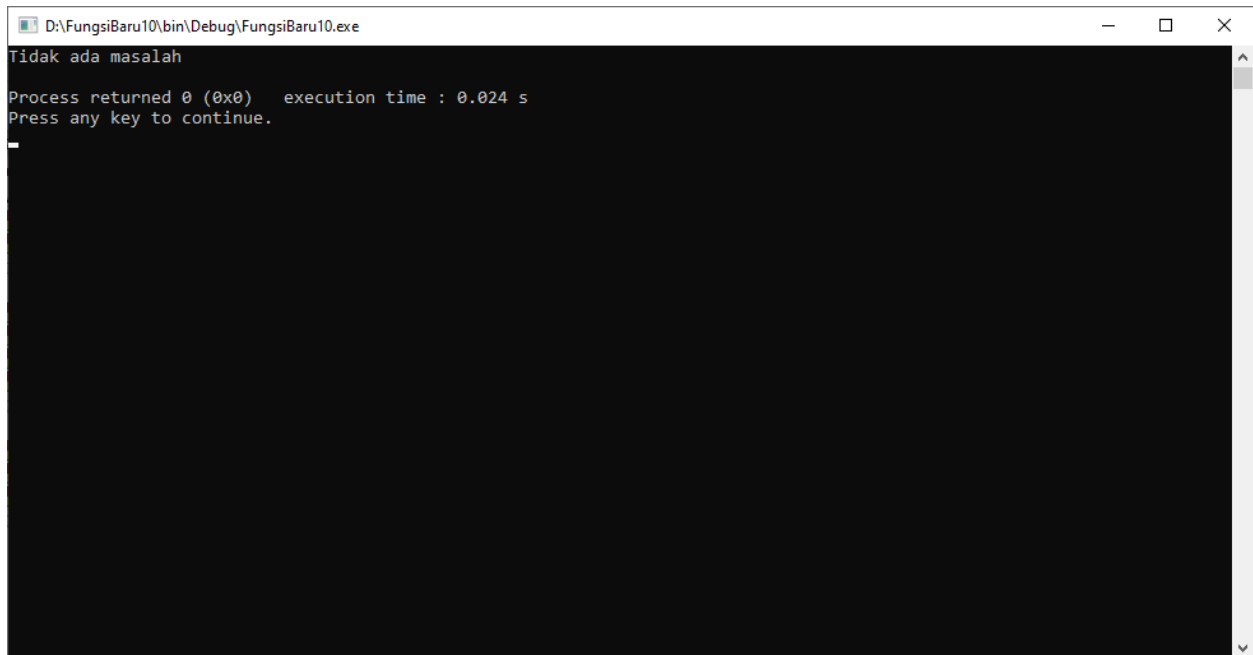
### 3. Listing Program :

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void fung_a(void);
5 void fung_b(void);
6 int x = 20;
7
8 main()
9 {
10     x += 2;
11     fung_a();
12     fung_a();
13     printf("\nNilai x dalam main() = %d\n", x);
14 }
15 void fung_a(void)
16 {
17     static x = 5;
18     x++;
19     printf("Nilai x dalam fung_a() = %d\n", x);
20     fung_b();
21 }
22 void fung_b(void)
23 {
24     x--;
25     printf("Nilai x dalam fung_b() = %d\n", x);
26 }
27
```

2. Kemudian tuliskan outputnya. Terangkan mengapa demikian.

Jawab :

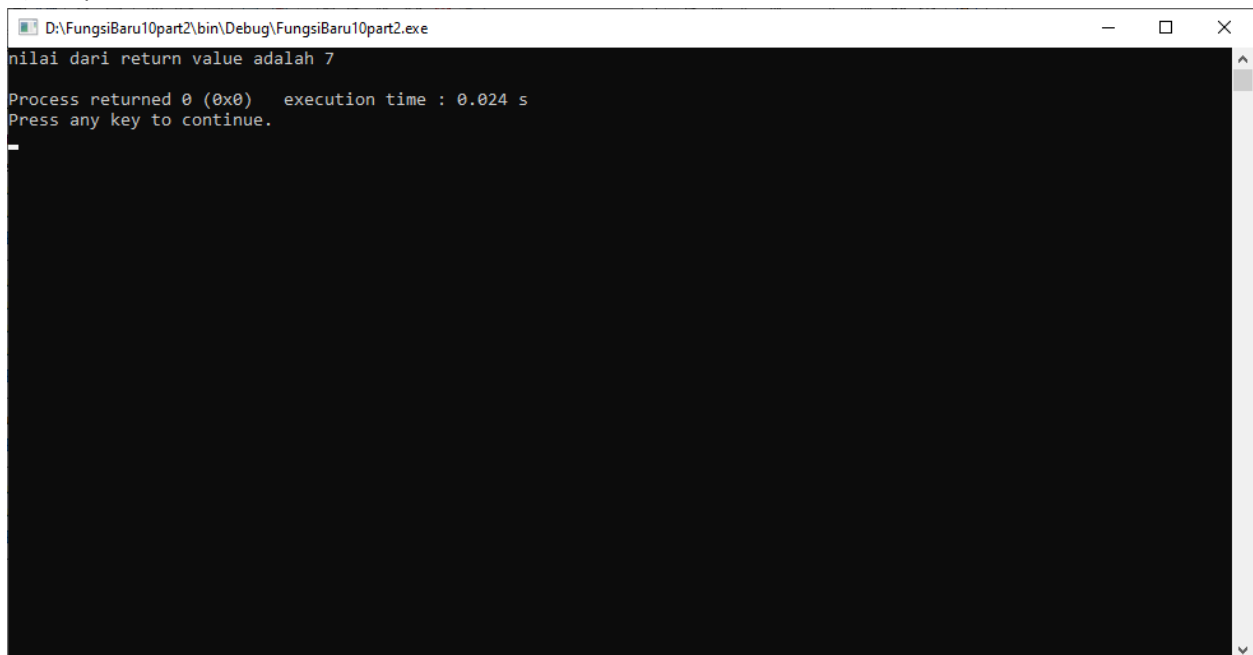
A) Output :



```
D:\FungsiBaru10\bin\Debug\FungsiBaru10.exe
Tidak ada masalah
Process returned 0 (0x0) execution time : 0.024 s
Press any key to continue.
```

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.

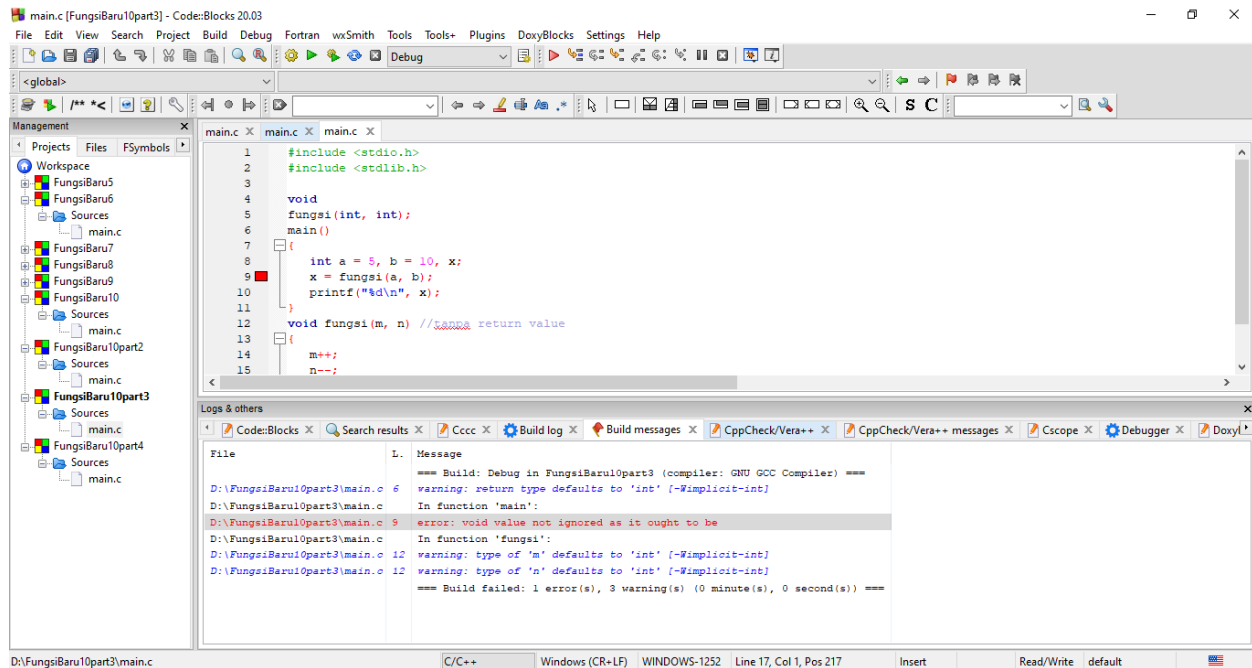
B) Output :



```
D:\FungsiBaru10part2\bin\Debug\FungsiBaru10part2.exe
nilai dari return value adalah 7
Process returned 0 (0x0) execution time : 0.024 s
Press any key to continue.
```

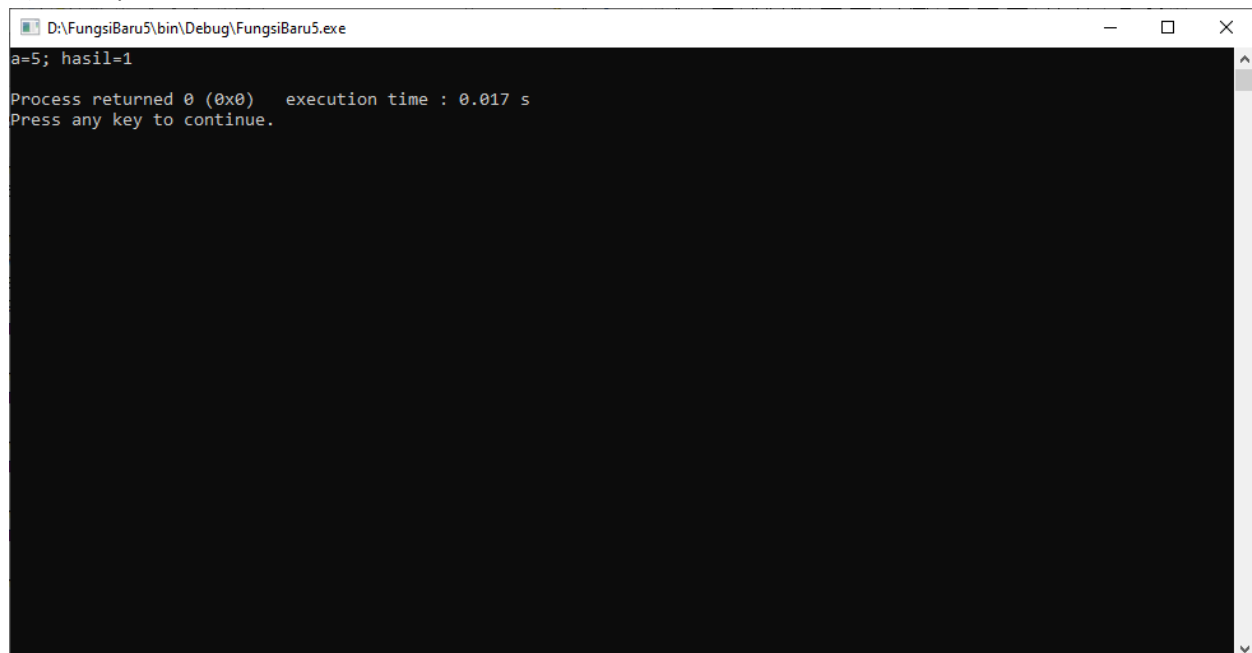
Alasan : Fungsi tetap menjalankan proses di dalamnya. Tetapi, hasil dari proses itu tidak bisa digunakan dalam fungsi main() atau fungsi lain.

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 :

```
D:\FungsiBaru10part4\main.exe
auto = 0, static = 0
auto = 0, static = 1
auto = 0, static = 2

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```

Alasan :-

### 3. Output :

```
D:\FungsiBaru6\main.exe
Nilai x dalam fung_a() = 6
Nilai x dalam fung_b() = 21
Nilai x dalam fung_a() = 7
Nilai x dalam fung_b() = 20

Nilai x dalam main()=20

Process returned 0 (0x0)   execution time : 0.041 s
Press any key to continue.
```

Alasan :-