**LAPORAN JOBSHEET 4**

**Struktur Perulangan**



Disusun Oleh :

DWI RINAYATUS SA’ADAH

3.34.18.1.10

IK- 1B

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI SEMARANG**

**2018**

**BAB 4**

**STRUKTUR PERULANGAN**

### TUJUAN

* 1. Mengetahui dan memahami tentang perintah perulangan *(loop/repitive)* dalam bahasa C.
  2. Menjelaskan tentang bentuk umum atau struktur dari perintah-perintah perulangan tersebut.
  3. Membuat progrma yang melibatkan permasalahan-permasalahan yang membutuhkan proses perulangan.

### DASAR TEORI

Salah satu proses di dalam suatu program adalah proses perulangan (loop).

Dalam bahasa C ada beberapa antara lain:

1. Do-while

Pada pernyataan do-while, tubuh loop (berupa pernyataan) berada diawal, sedangkan kondisi berada dibelakang. Berikut ini low chart dan bentuk umum pernyataan do-while:

Struktur diagram alirnya:

Bentuk umum struktur DO-WHILE

*do*

*aksi /tubuh perulangan*

Benar

Kondisi ?

Pernyataan

tubuh loop

Salah

keluar loop

### Catatan :

* + - KONDISI adalah ungkapan yang digunakan untuk menyatakan suatu yang diisyaratkan (ungkapan akan bernilai boolean (kondisi true atau false))
    - Pernyataan adalah perintah-perintah yang dikerjakan selama loop berlangsung. Biasanya perulangan dalam d0-while pasti berupa pernyataan jamak.

Contoh penggunaan do-while, untuk mengunci pemakai agar yang ditekan hanya tombol ‘Y’, ‘y’, ‘T’, atau ‘t’ (bila tombol lain yang ditekan maka loop akan berjalan terus):

cacahß1 { inisialisasi variable counter } do

output (“Selamat Belajar Algoritma”) cacahßcacah+1 {increment variabel counter} while(cacah ≤ 100)

Contoh 2 penggunaan DO-WHILE, untuk mengunci pemakai agar yang ditekan hanya tombol ‘Y’,’y’, ‘T’, atau ‘t’ ( bila tombol lain yang ditekan maka loop akan berjalan terus):

### do

**pilih  input(karakter) /\* baca tombol \*/ sdh\_bnr  (pilih =’Y’) OR (pilih =’y’) OR**

**(pilih =’T’) OR (pilih =’t’) while(NOT sdh\_bnr)**

* 1. Pernyataan while

Berbeda dengan *do-while*, pada pernyataan while pengujian terhadap *loop* dilakukan dibagian awal ( sebelum tubuh *loop*), untuk lebih jelasnya perhatikan *flow chart* dan bentuk umum pernyataan while berikut :

Struktur diagram alirnya :

Kondisi ?

Benar

Salah

Pernyataan

Bentuk umum struktur WHILE

*while(kondisi)*

*aksi/tubuh perulangan*

keluar loop

**Keterangan :**

Dari gambar disamping terlihat bahwa kemungkinan pernyataan sebagai tubuh loop tidak akan pernah dijalankan selama kondisi yang disyaratkan oleh while tidak terpenuhi.

Contoh 1. Penggunaan WHILE untuk proses perulangan yang jumlah perulangannya dapat diketahui.

cacahß1 { inisialisasi variable counter } while(cacah ≤ 100)

output (“Selamat Belajar Algoritma”) cacahßcacah+1 {increment variabel counter}

endwhile

Contoh 2 penggunaan *WHILE*, untuk kasus yang sama seperti permasalahan di atas:

sdh\_bnr ß 0 { diberi nilai awal dahulu } while(NOT sdh\_bnr)

pilihßinput(karakter) { baca tombol } sdh\_bnr ß (pilih =’Y’) OR (pilih =’y’) OR

(pilih =’T’) OR (pilih =’t’)

Endwhile

* 1. Struktur REPEAT

Secara prinsip kerja struktur REPEAT sama dengan struktur Do-WHILE. Perbedaan mendasar antara keduanya adalah pada masalah logika pengulangan, yaitu perulangan dalam struktur REPEAT akan berjalan selama kondisi bernilai salah sedang pada struktur Do-WHILE adalah sebaliknya.

* 1. Struktur FOR

Struktur perulangan yang lain adalah struktur FOR. Bentuk umum dan flow chart dari struktur FOR adalah sebagai berikut :

Struktur diagram alirnya :

Ungkapan 2

Benar

Salah

Ungkapan 3

Pernyataan

Ungkapan 1

Keluar loop

Bentuk umum Struktur FOR

*for pencacah nilai\_awal to atau downto nilai\_akhir do aksi*

Keterangan :

* Pencacah : dipakai sebagai variabel pengendali loop (variabel counter).
* Nilai\_awal: diberikan untuk nilai awal variabel counter
* Nilai\_akhir: dipakai sebagai terminator proses perulangan

Contoh pemakaian FOR untuk kasus di atas:

**for cacah  1 to 100 do**

**output (“Selamat Belajar Algoritma”)**

* 1. Nested loop ( loop dalam loop)

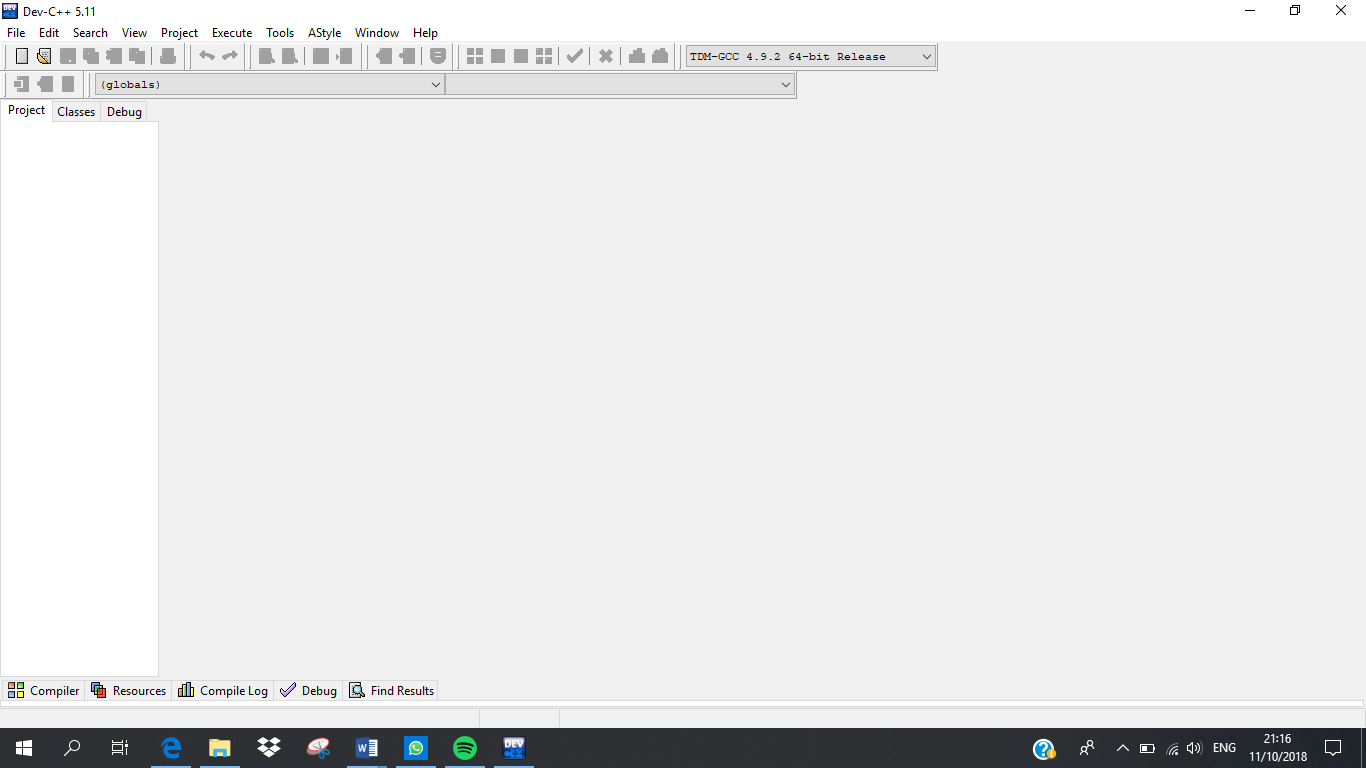
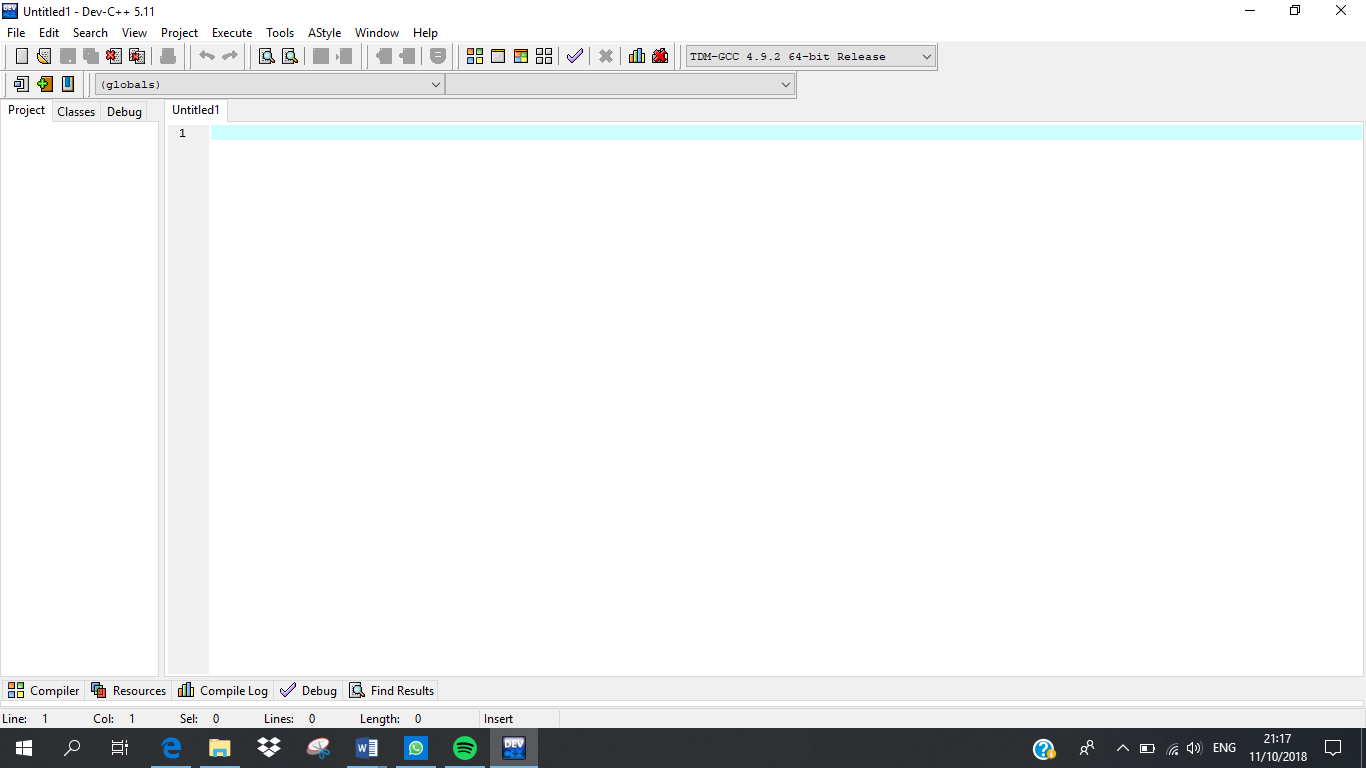
Dalam suatu loop bisa terkandung loop yang lain. Loop seperti ini dikenal dengan istilah loop dalam loop ( *nested loop*). Berikut ini contoh permasalahan yang membutuhkan *nested loop* untuk membuat tabel perkalian :

**for baris  1 to maks\_baris do for kolom 1 to maks\_kolom do**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
| 3 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 |
| 4 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 |
| 5 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
| 6 | 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 |
| 7 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 |
| 8 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 |

**Hasil\_kali = baris \* kolom output(hasil\_kali)**

### LANGKAH KERJA

1. **Langkah menampilkan “TURBO C” sebanyak 10 kali :**
   1. Jalankan Dev C++
   2.  Alt+F  New
   3. Ketik :

/\* ---------------------------------------------\*

\* Program Lat4a.c \*

\* contoh pemakaian do-while \*

\* menampilkan tulisan "Turbo C" 10 kali \*

\*----------------------------------------------\*/

#include<stdio.h>

#include<conio.h>

main()

{

int pencacah;

pencacah=0;

do

{

printf("%d", pencacah);

puts(" - TURBO C");

pencacah++;

//pencacah=pencacah + 1

}

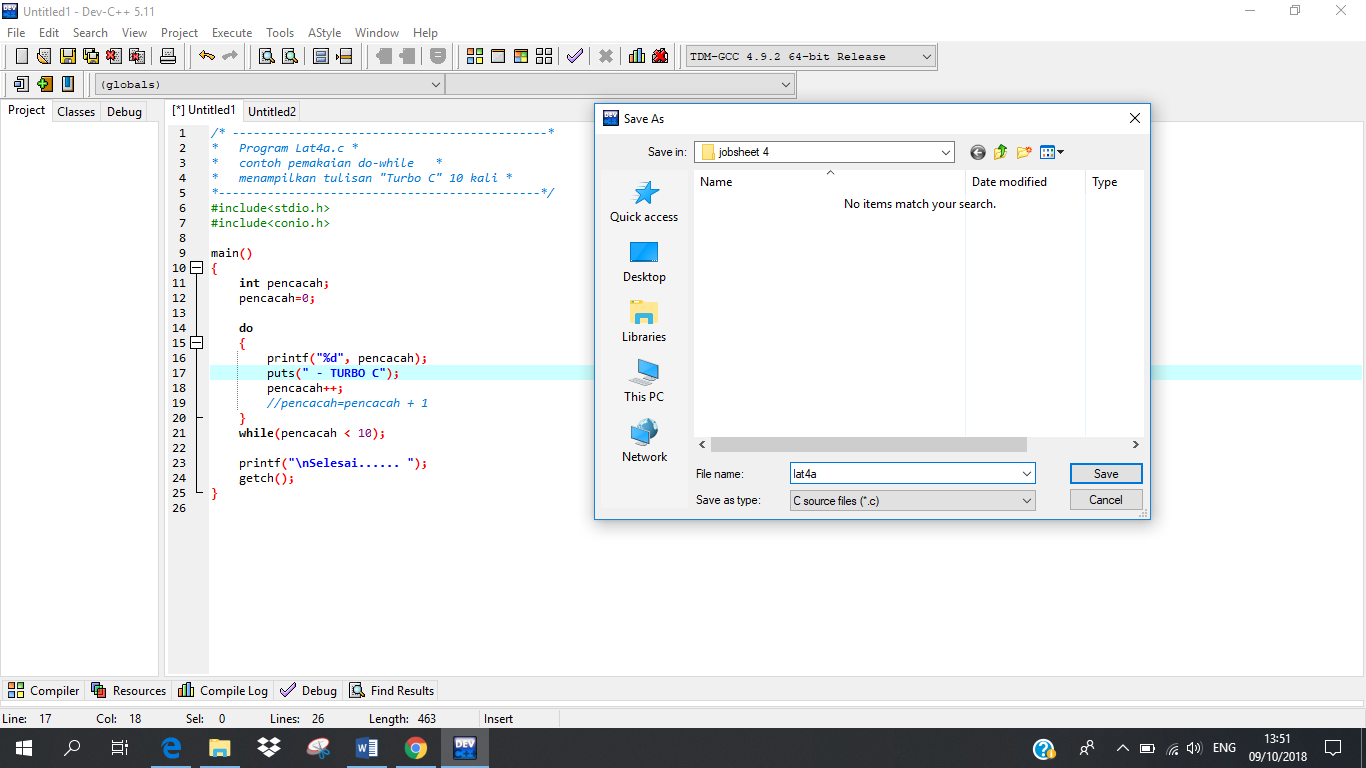
while(pencacah < 10);

printf("\nSelesai...... ");

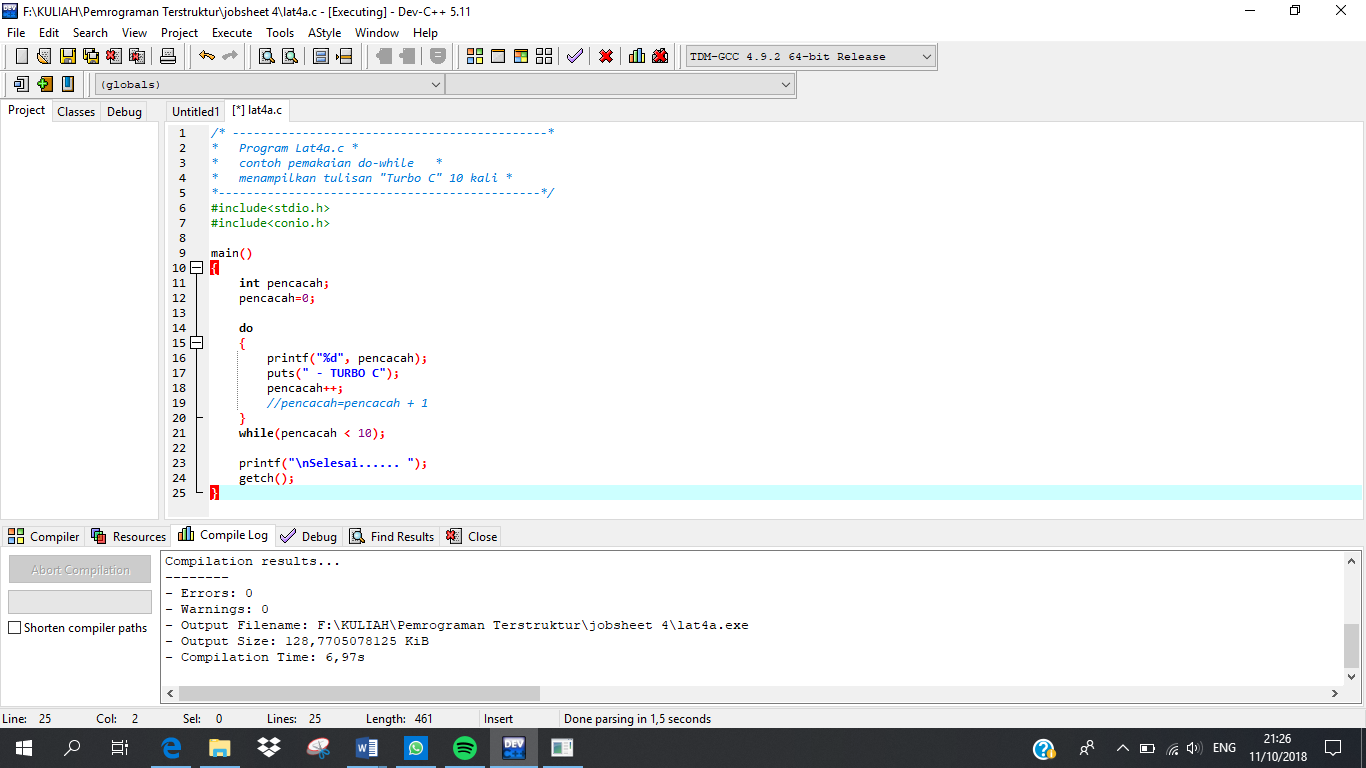
getch();

}

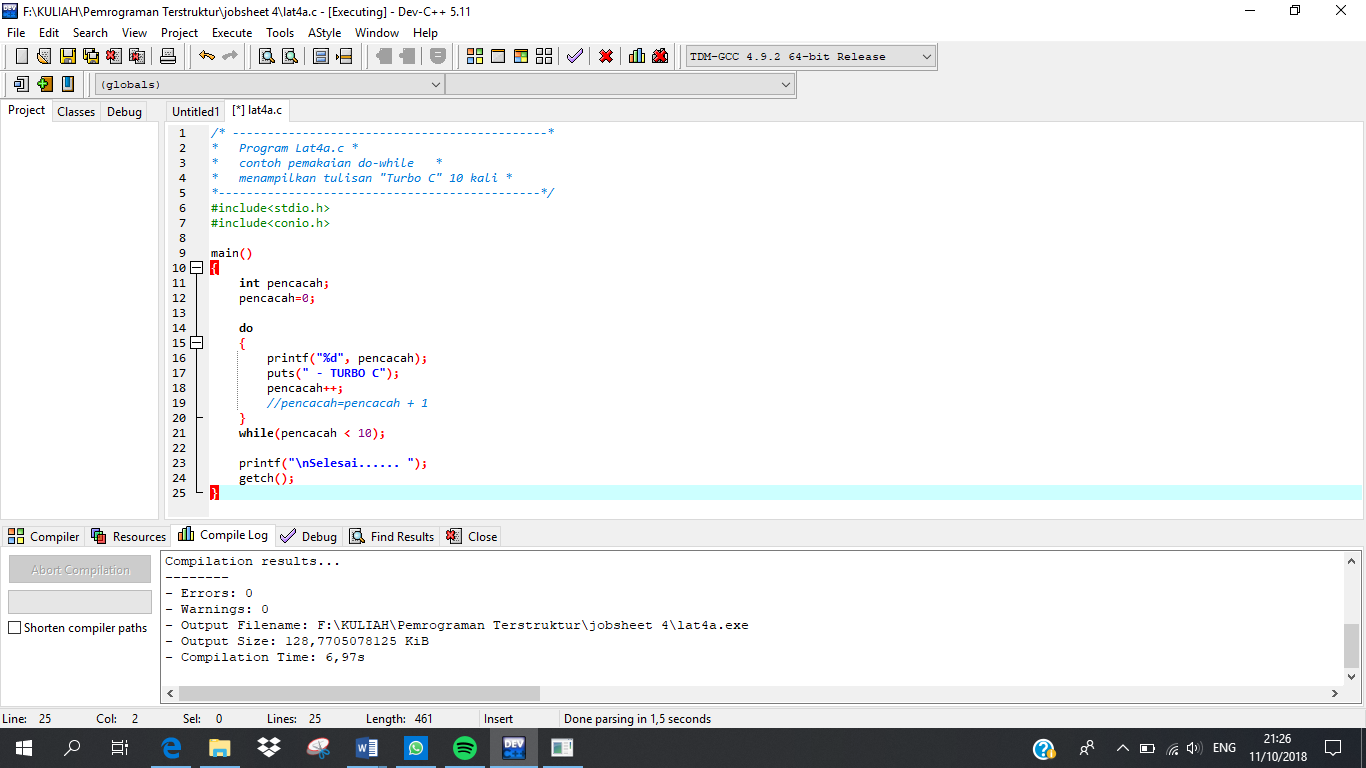
* 1. Alt+F, save dengan nama Lat4a.c

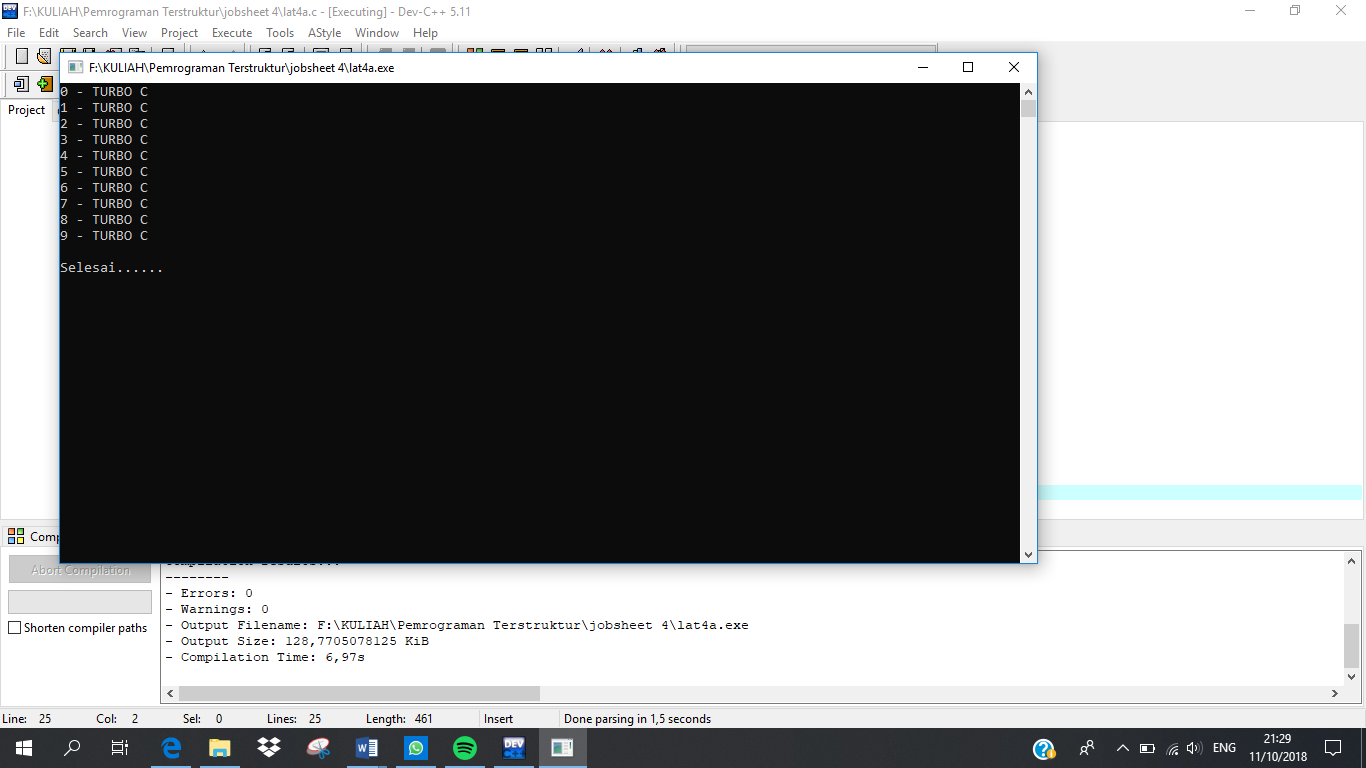


* 1. Lalu Compile hingga tidak terdapat error dan dapat di run.



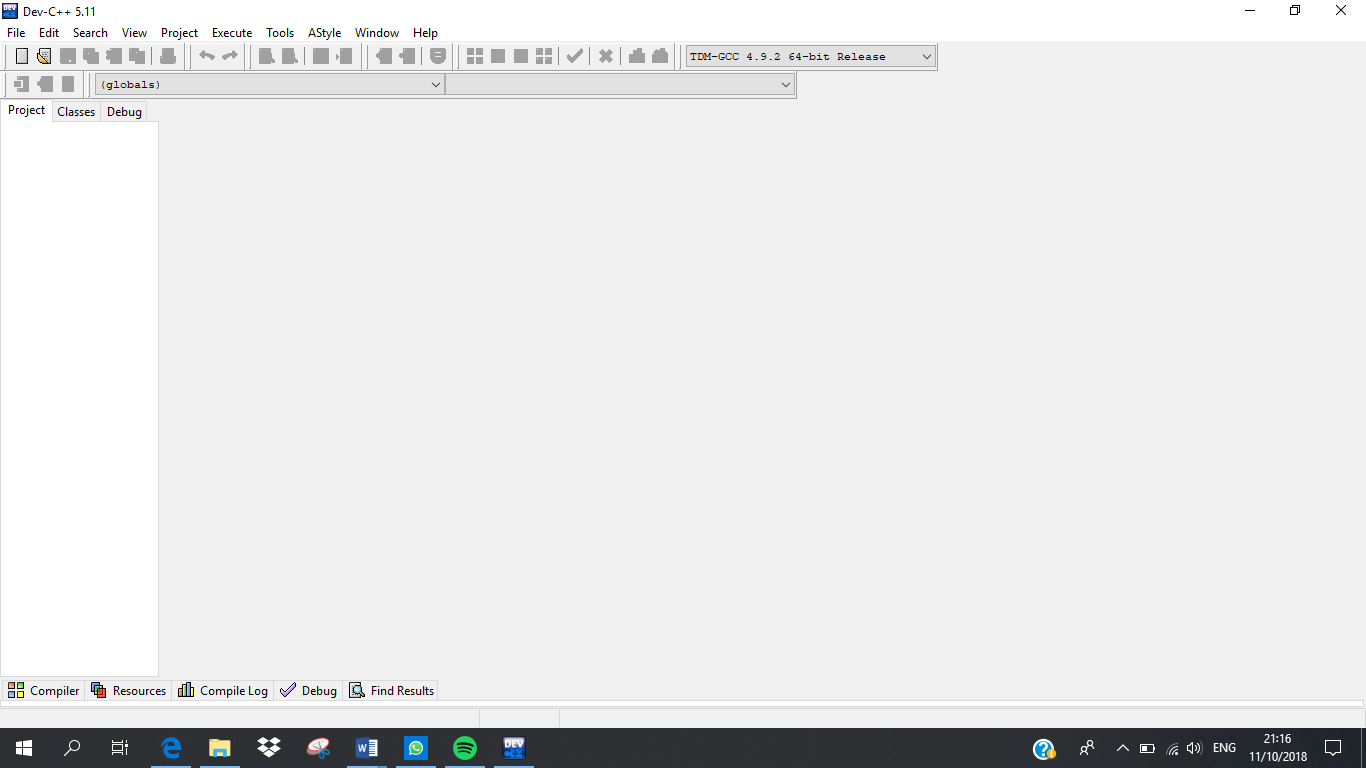
Setelah di compile tidak terdapat error

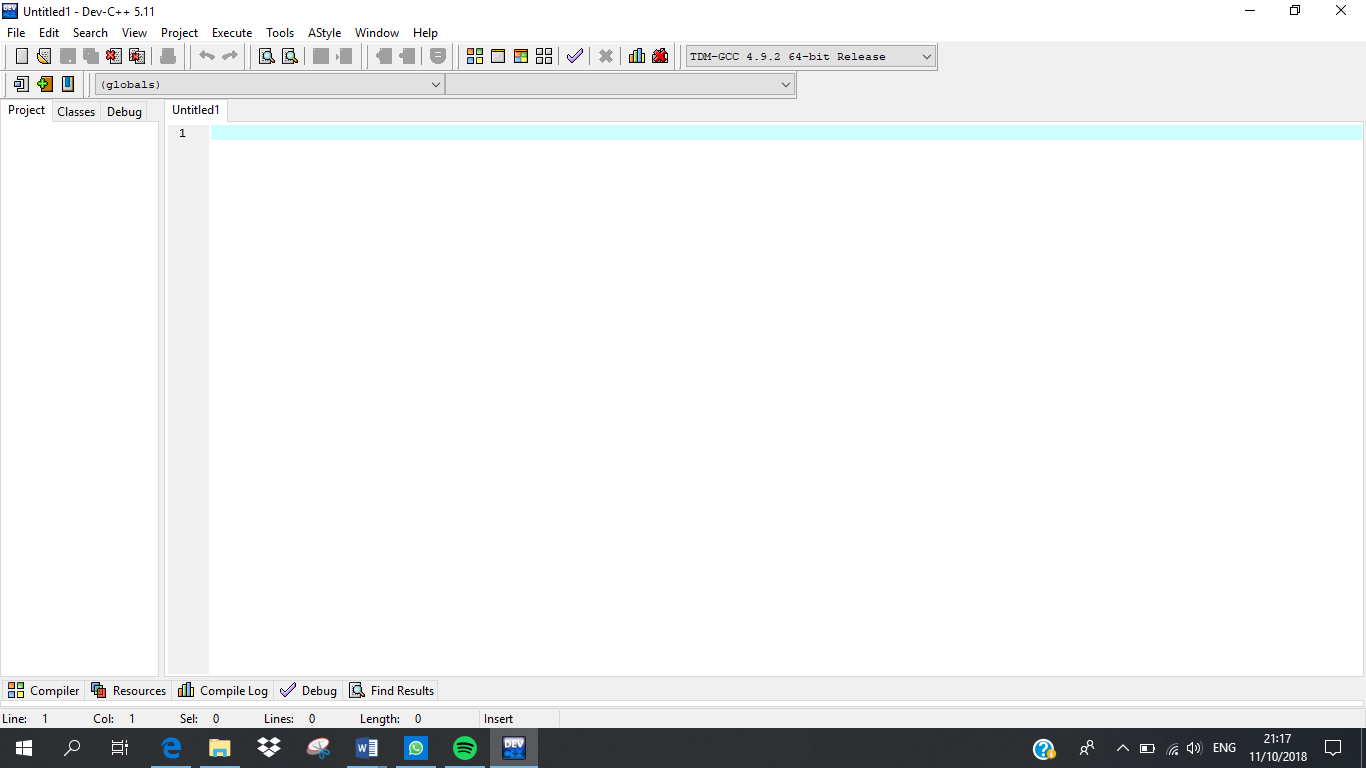
* 1.  Run program tersebut.

Hasil Program

### Membaca tombol Y atau T :

* 1. Jalankan Dev C++



* 1. Alt+F  New
  2. Ketik :

/\* ---------------------------------------\*

\* Program Lat4b.c \*

\* contoh pemakaian do-while \*

\* Untuk membaca tombol Y atau T \*

\*----------------------------------------\*/

#include <stdio.h>

#include <conio.h>

main()

{

int sudah\_benar;

char pilihan;

printf("Pilihlah Y atau T : ");

/\*program hanya mau dilanjutkan kalau tombol Y, y, T atau t yang ditekan\*/

do

{

pilihan = getch (); /\*baca tombol\*/

sudah\_benar = (pilihan=='Y') || (pilihan=='y') || (pilihan=='T') || (pilihan=='t');

}

while(!sudah\_benar);

/\*memberi keterangan tentang pilihan\*/

switch (pilihan)

{

case 'Y' :

case 'y' :

puts("\nPilihan anda adalah Y");

break;

case 'T' :

case 't' :

puts("\nPilihan anda adalah T");

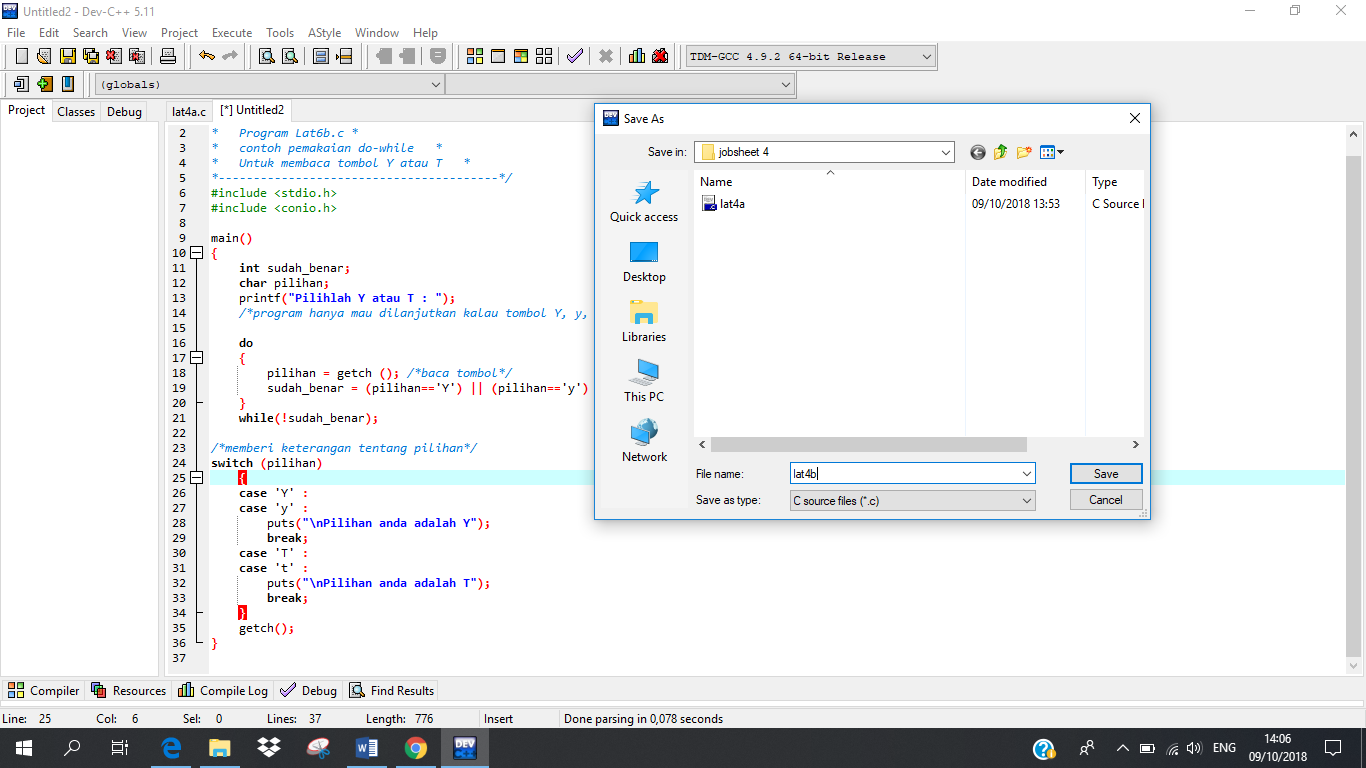
break;

}

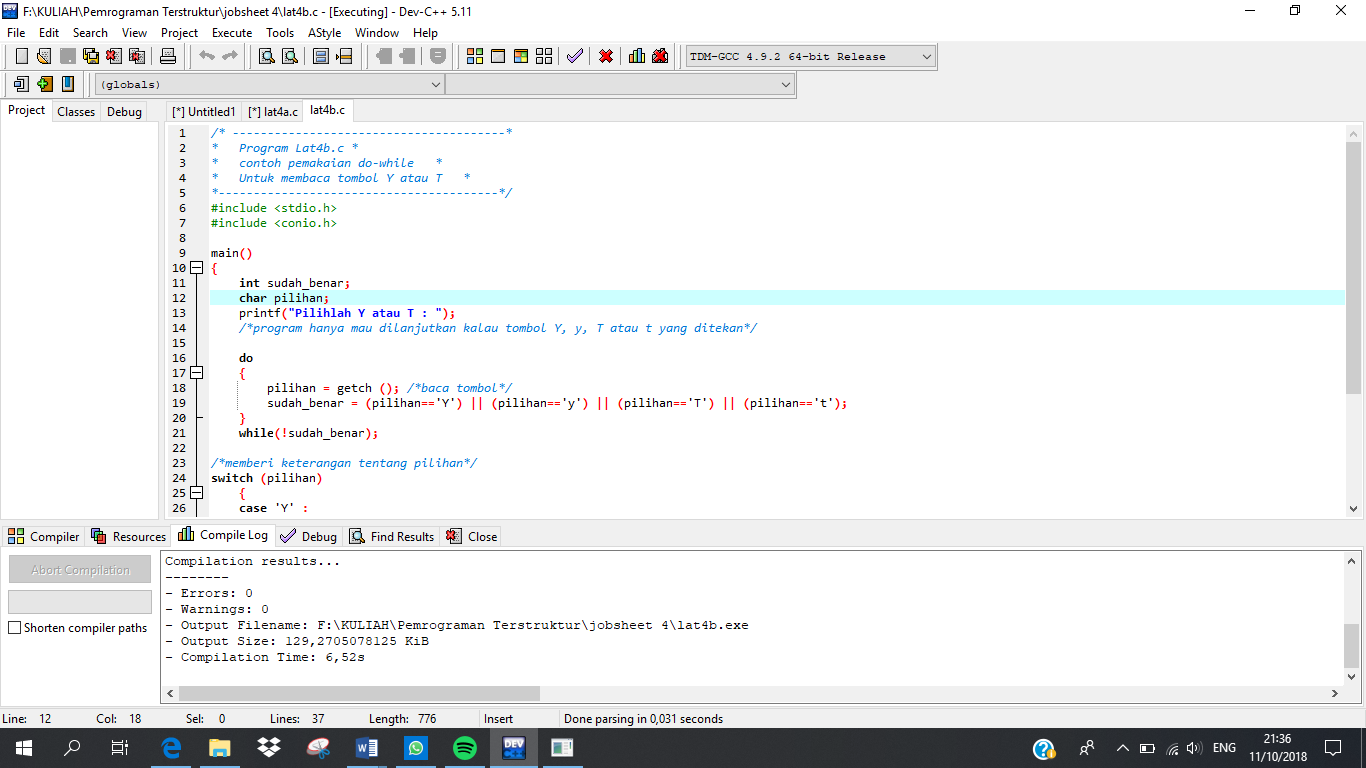
getch();

}

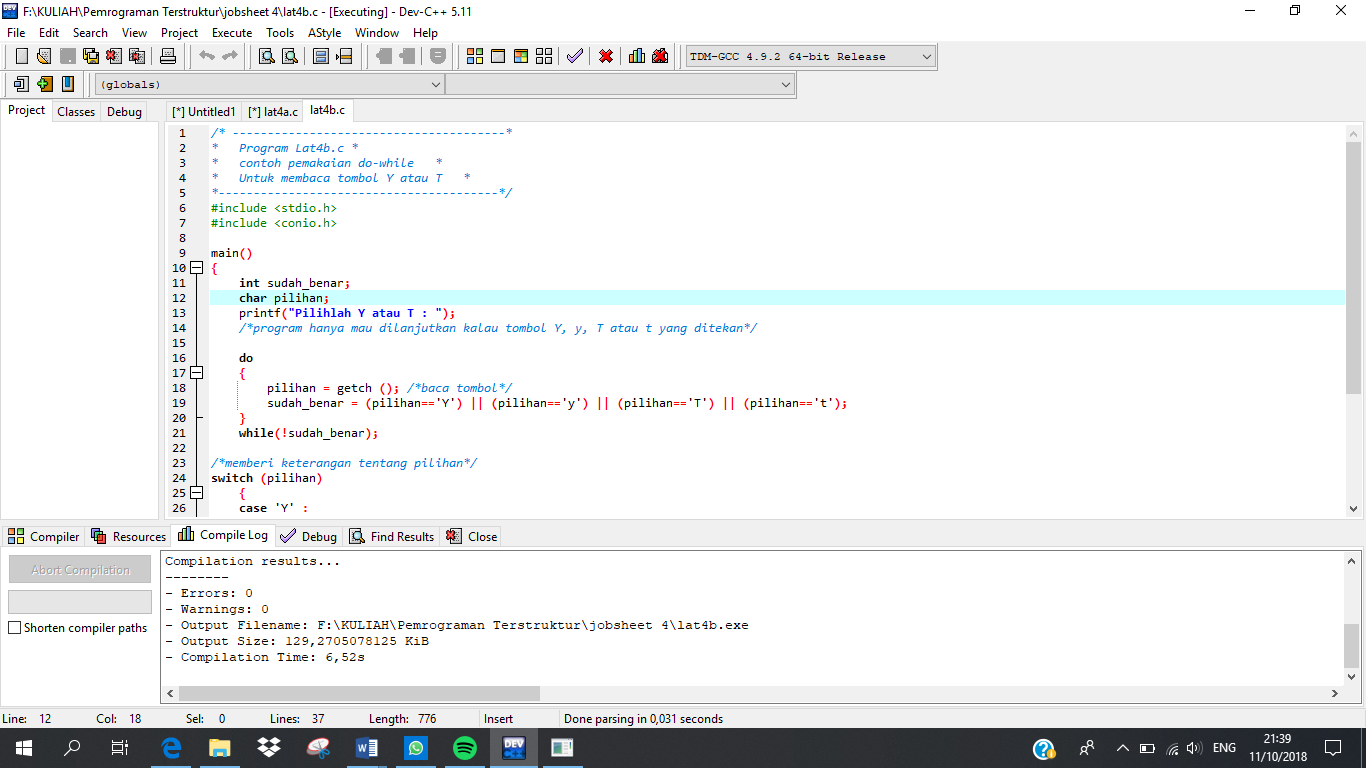
* 1. Alt+F, save dengan nama Lat4b.c

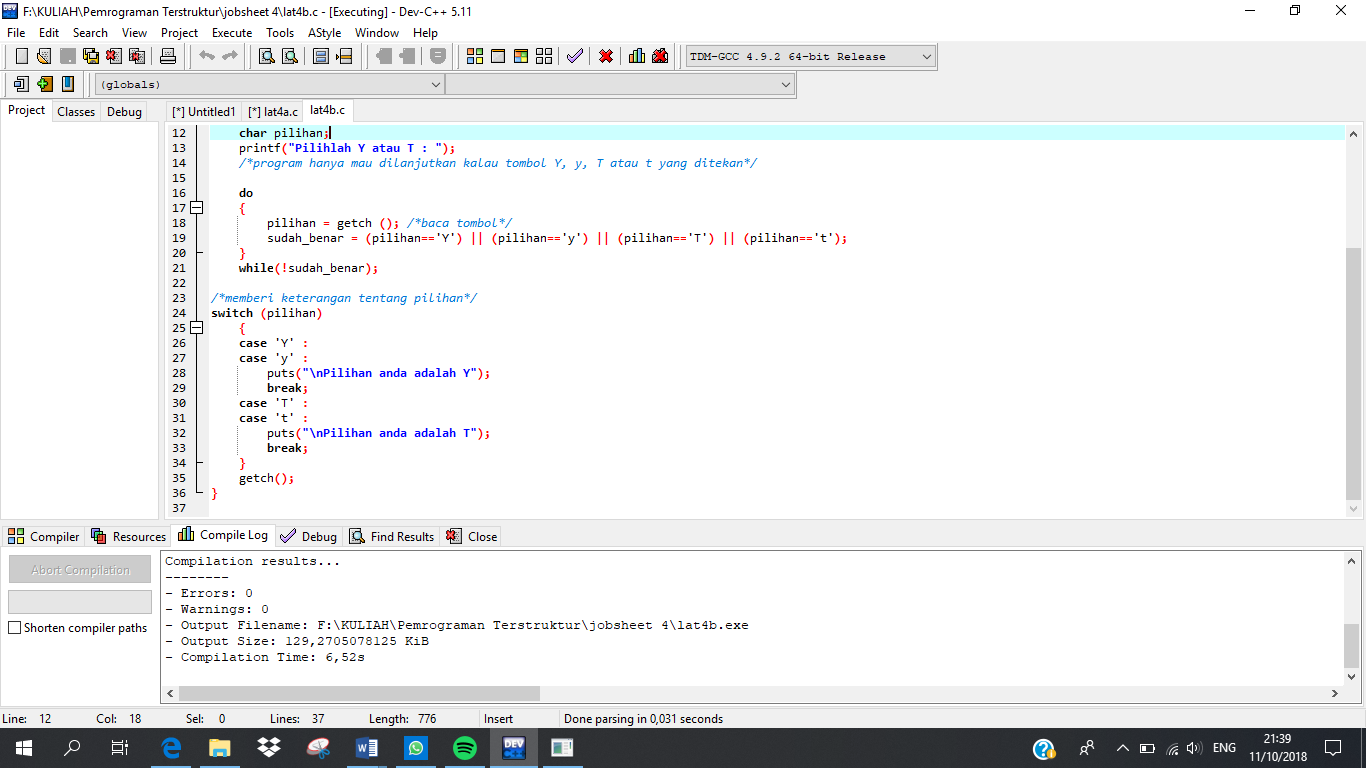


* 1. Lalu Compile hingga tidak terdapat error dan dapat di run.

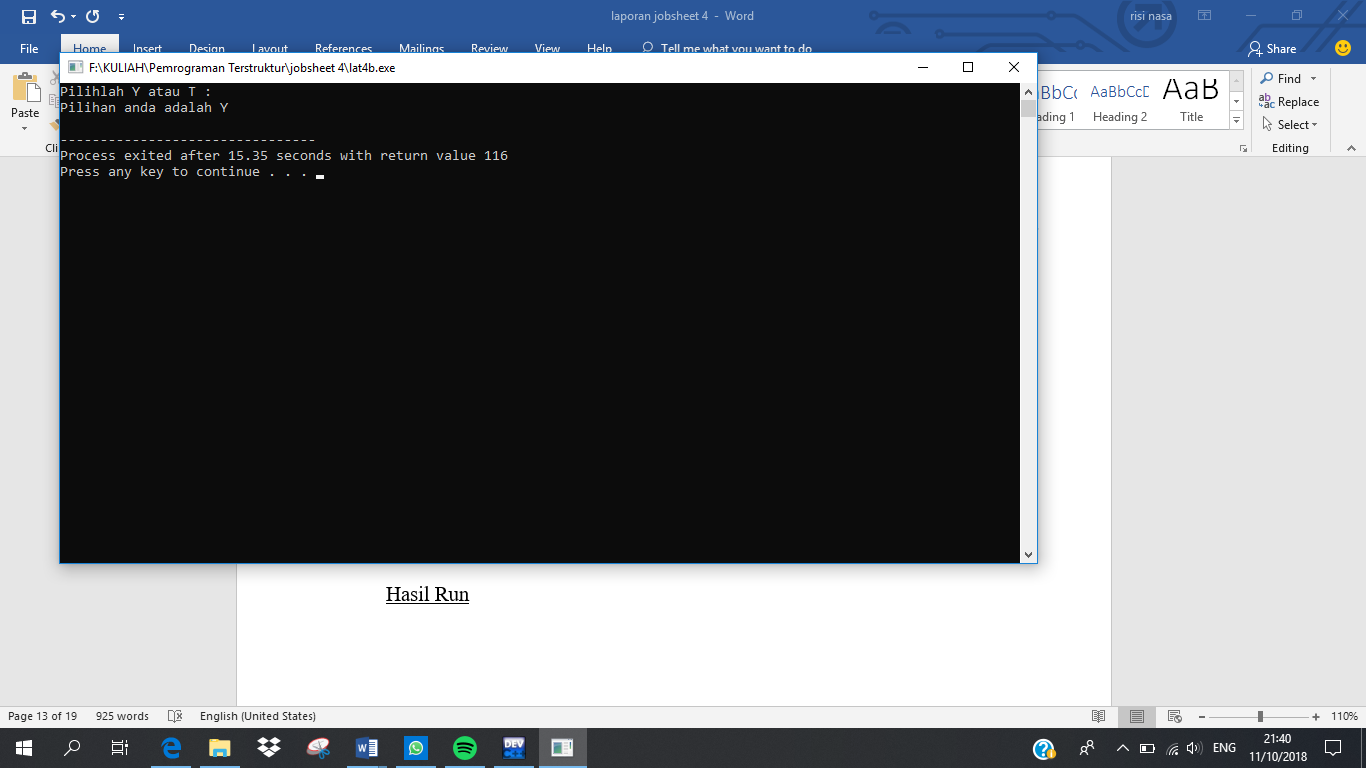


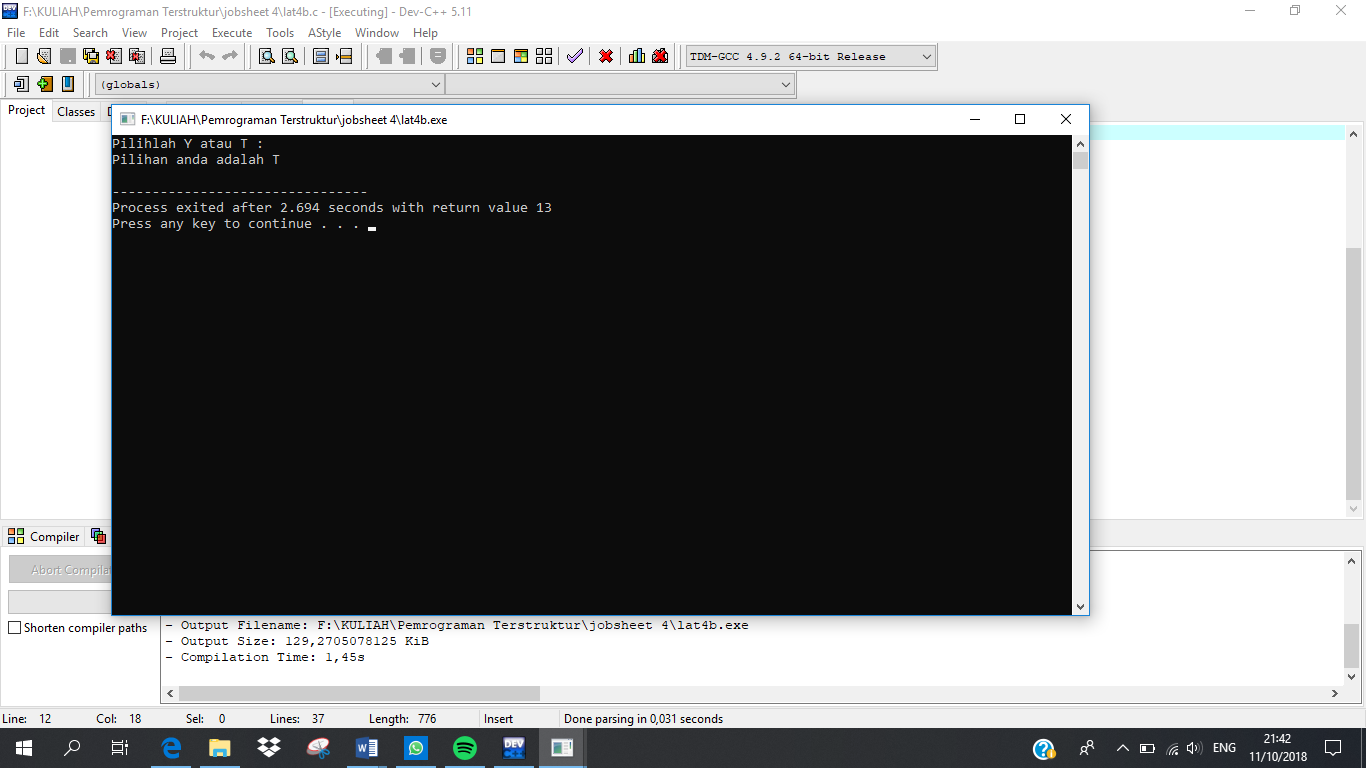
Setelah decompile tidak terdapat error

* 1.  Run program tersebut.

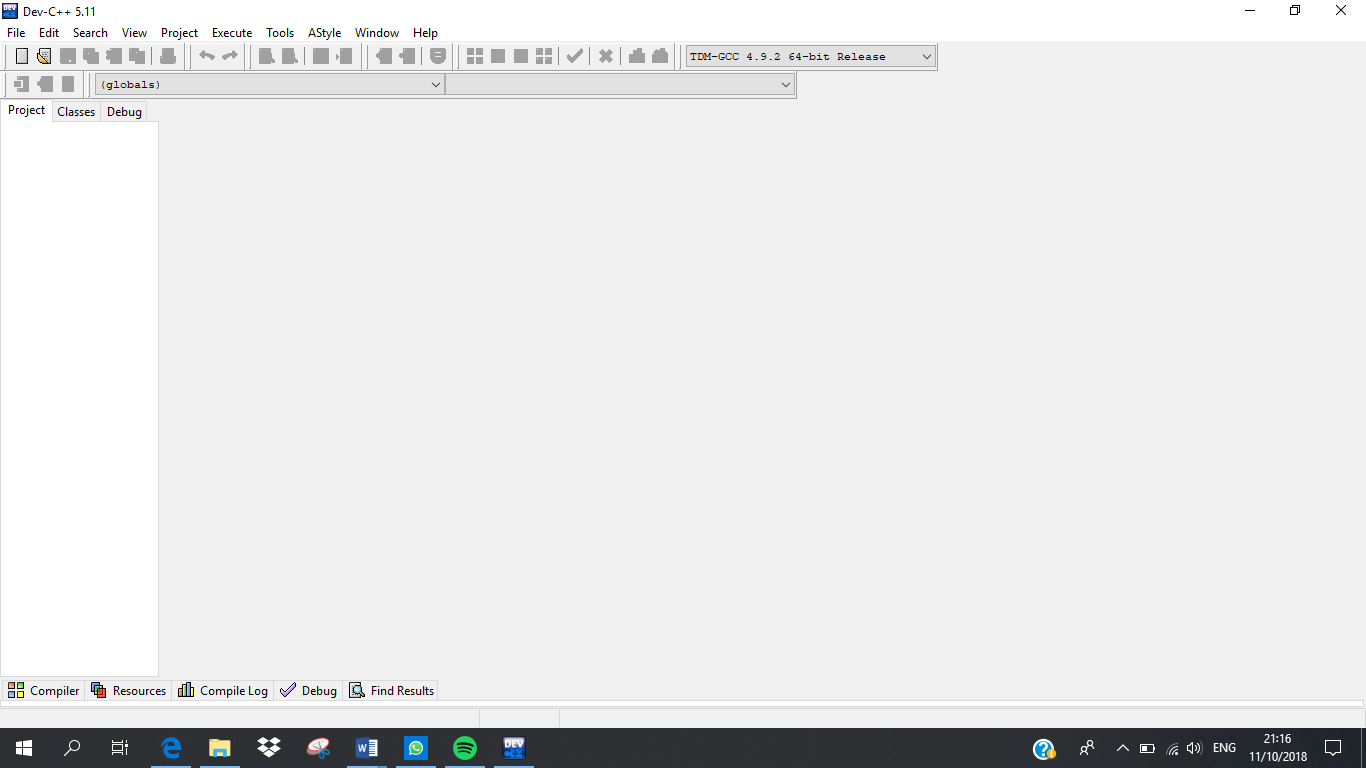
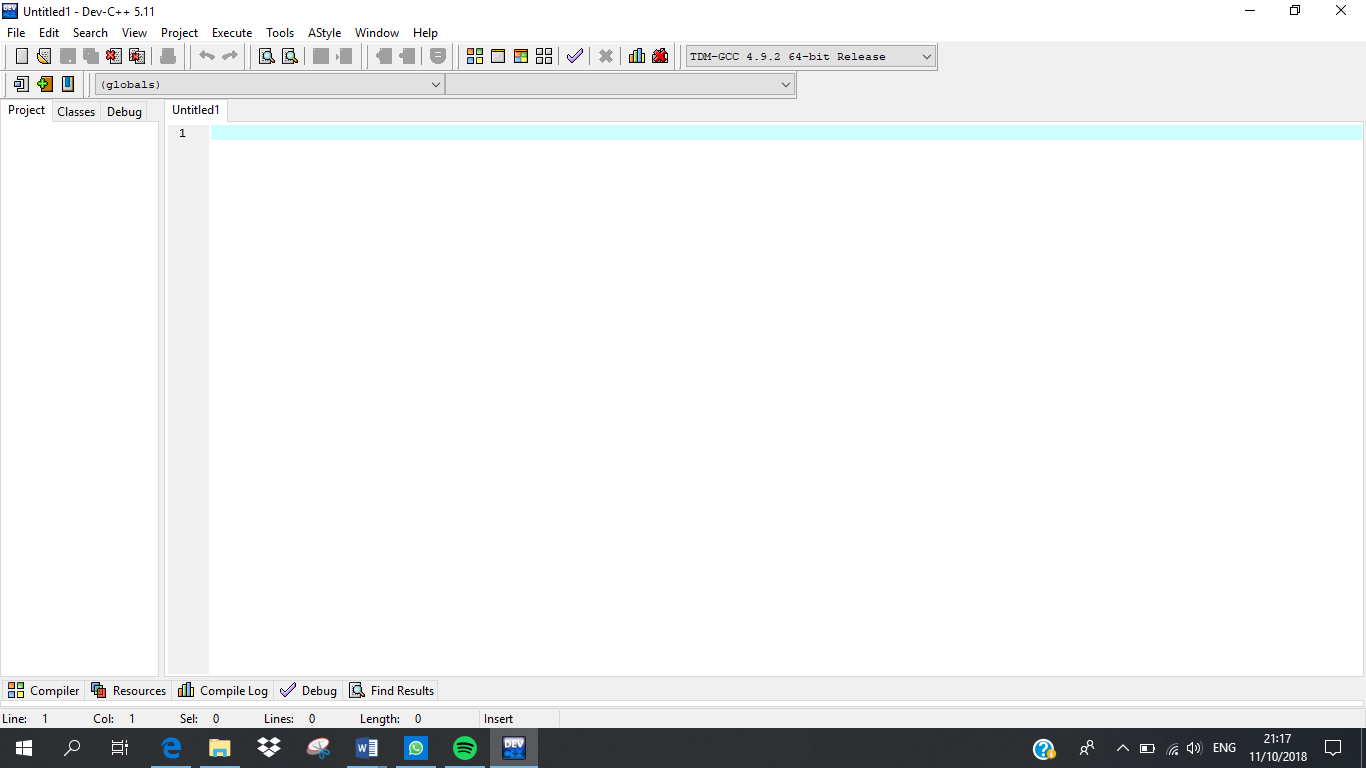


Hasil Run





### Menghitung jumlah kata dan karakter dalam suatu kaliamt

* 1. Jalankan Dev C++
  2. Alt+F New
  3. Ketik :

/\* ---------------------------------------------\*

\* Program Lat4c.c \*

\* contoh pemakaian do-while \*

\* menghitung jumlah kata dan karakter \*

\* dalam suatu kalimat \*

\*----------------------------------------------\*/

#include <stdio.h>

#include <conio.h>

#define ENTER '\r' /\*karakter CR\*/

#define SPASI ' ' /\*karakter spasi\*/

main()

{

char karakter;

int jumkar = 0;

int jumspasi= 1;

puts("Ketikkan sebuah kalimat dan akhiri dengan ENTER");

puts("Saya akan menghitung jumlah karakter dan kata pada kalimat tersebut\n");

while( (karakter = getche () ) !=ENTER)

{

jumkar++;

if(karakter == SPASI) jumspasi++;

}

printf("\nJumlah karakter : %d",jumkar);

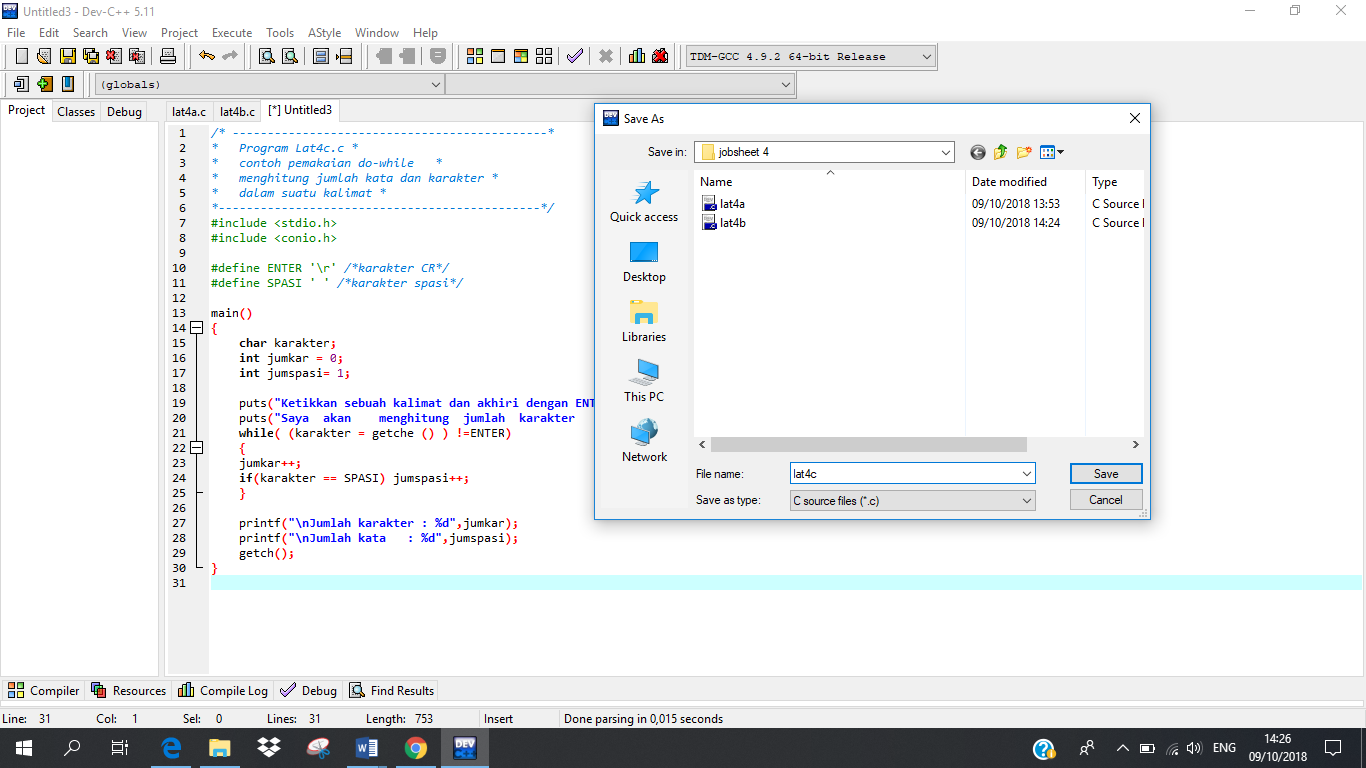
printf("\nJumlah kata : %d",jumspasi);

printf("\nJumlah spasi : %d",jumspasi-1);

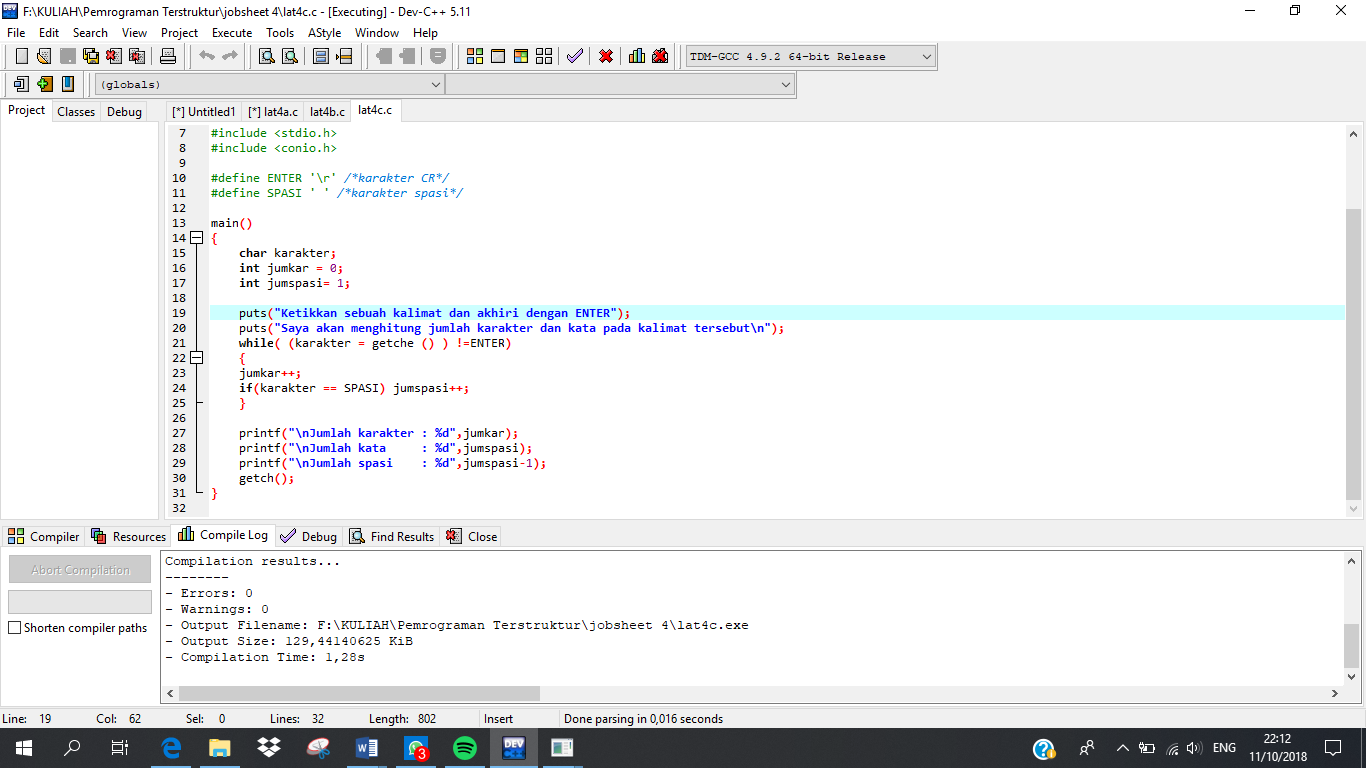
getch();

}

* 1. Alt+F, save dengan nama Lat4c.c

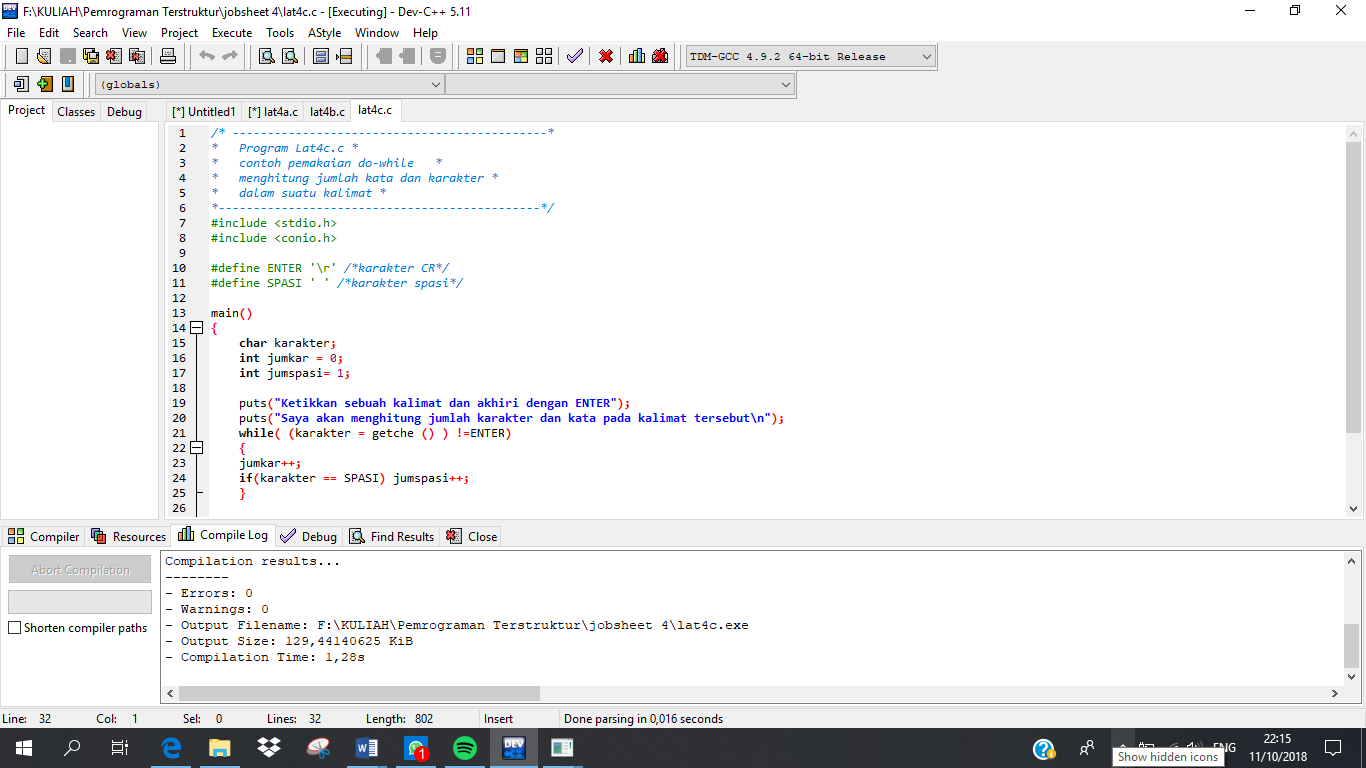


* 1. Lalu Compile hingga tidak terdapat error dan dapat di run.



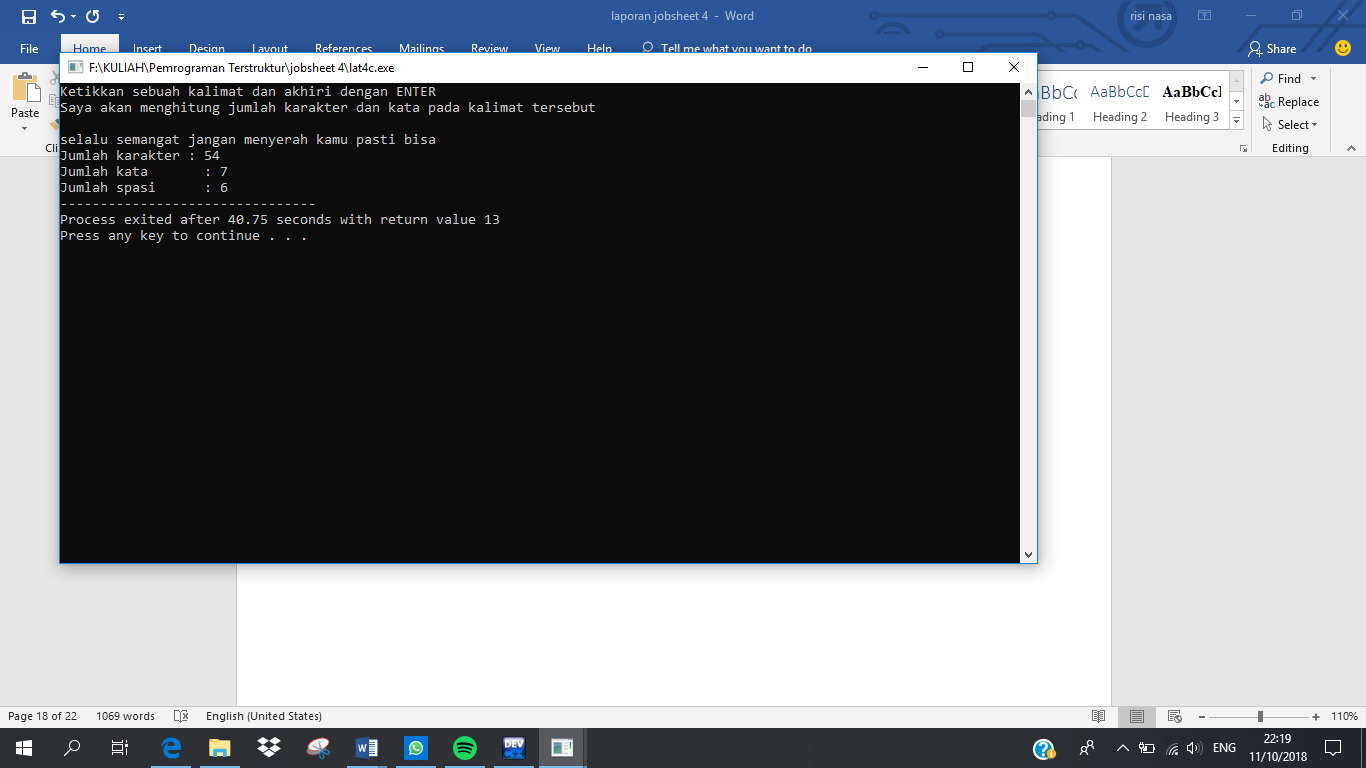
Setelah decompile tidak terdapat error.

* 1. Run program tersebut.



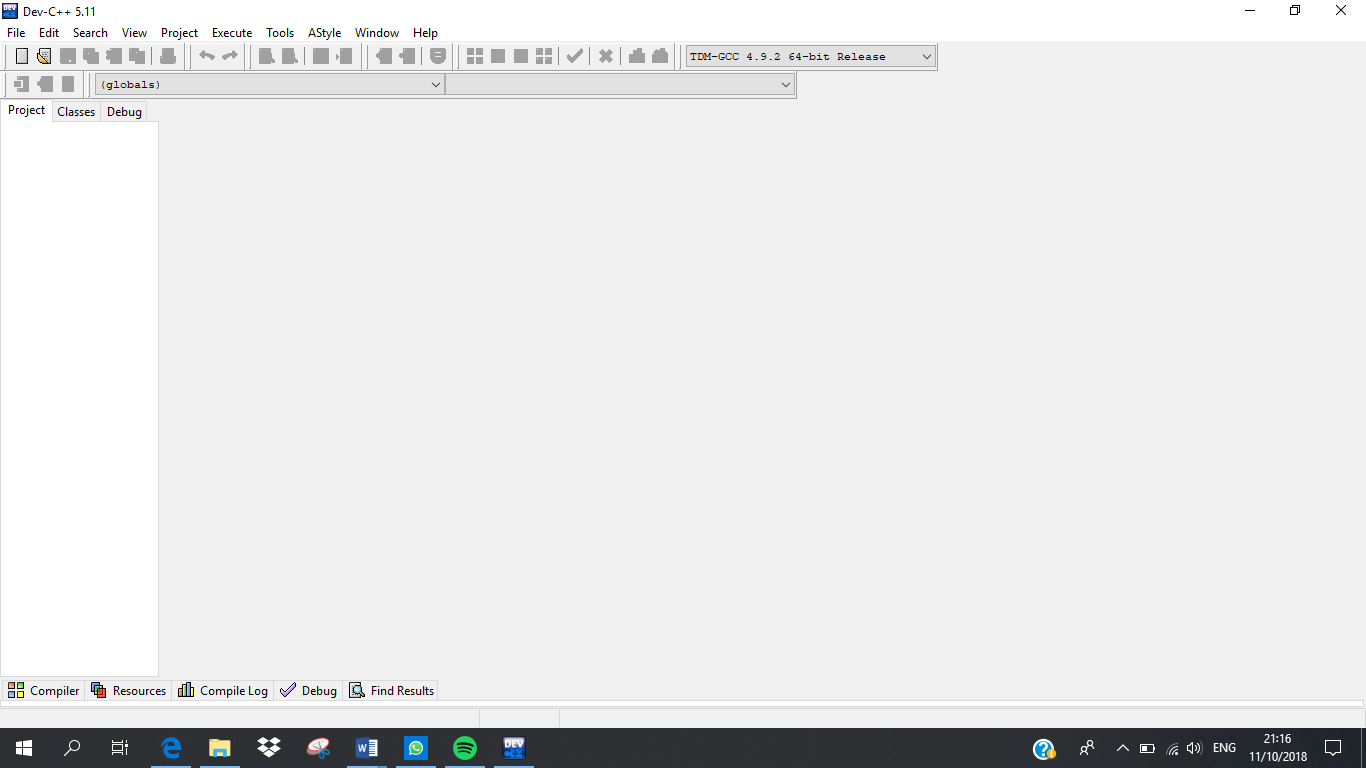


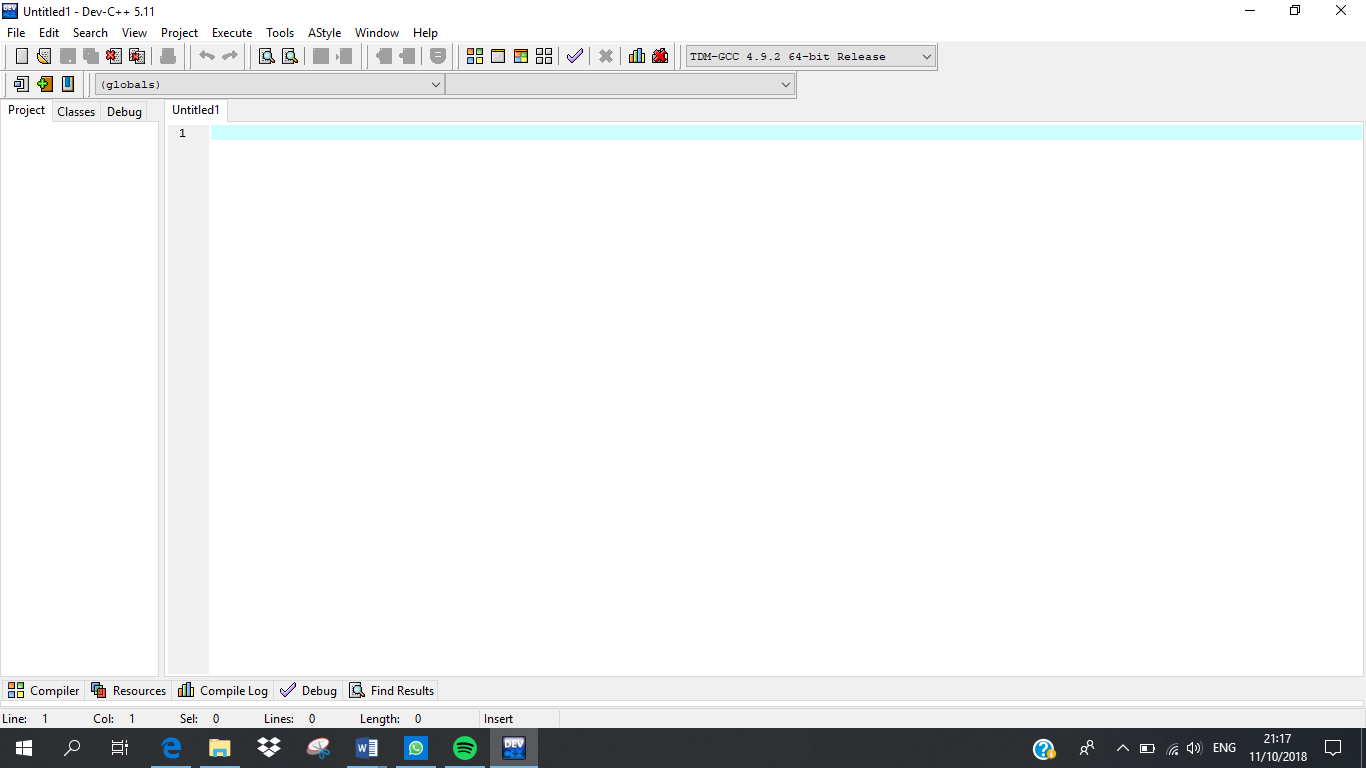
Hasil Run



### Mencari rata-rata dari sejumlah bilangan denga struktur *for* :

* 1. Jalankan Dev C++



* 1. Alt+F  New
  2. Ketik :

/\* ---------------------------------------------\*

\* Program Lat4d.c \*

\* contoh pemakaian for \*

\* Mencari rata-rata dari sejumlah bilangan \*

\*----------------------------------------------\*/

#include<stdio.h>

#include<conio.h>

main()

{

int i,n; /\*n adalah jumlah bilangan\*/

float bil,rata,total=0; /\*bil adalah bilangan yang akan anda masukkan\*/

printf("\nJumlah bilangan : ");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

printf("Bilangan ke %d : ",i);

scanf("%f",&bil);

total+=bil;

printf("total = %.lf\n",bil +total );

//total = total + bil

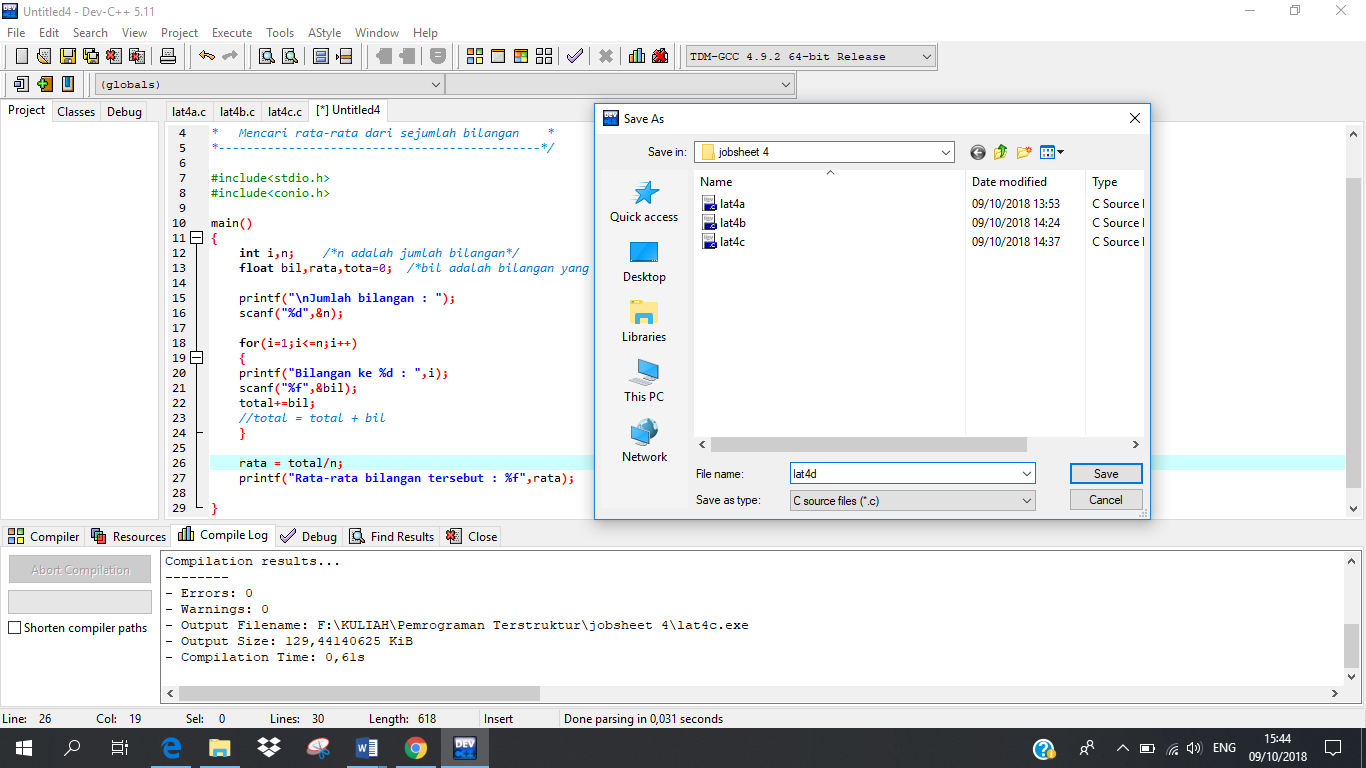
}

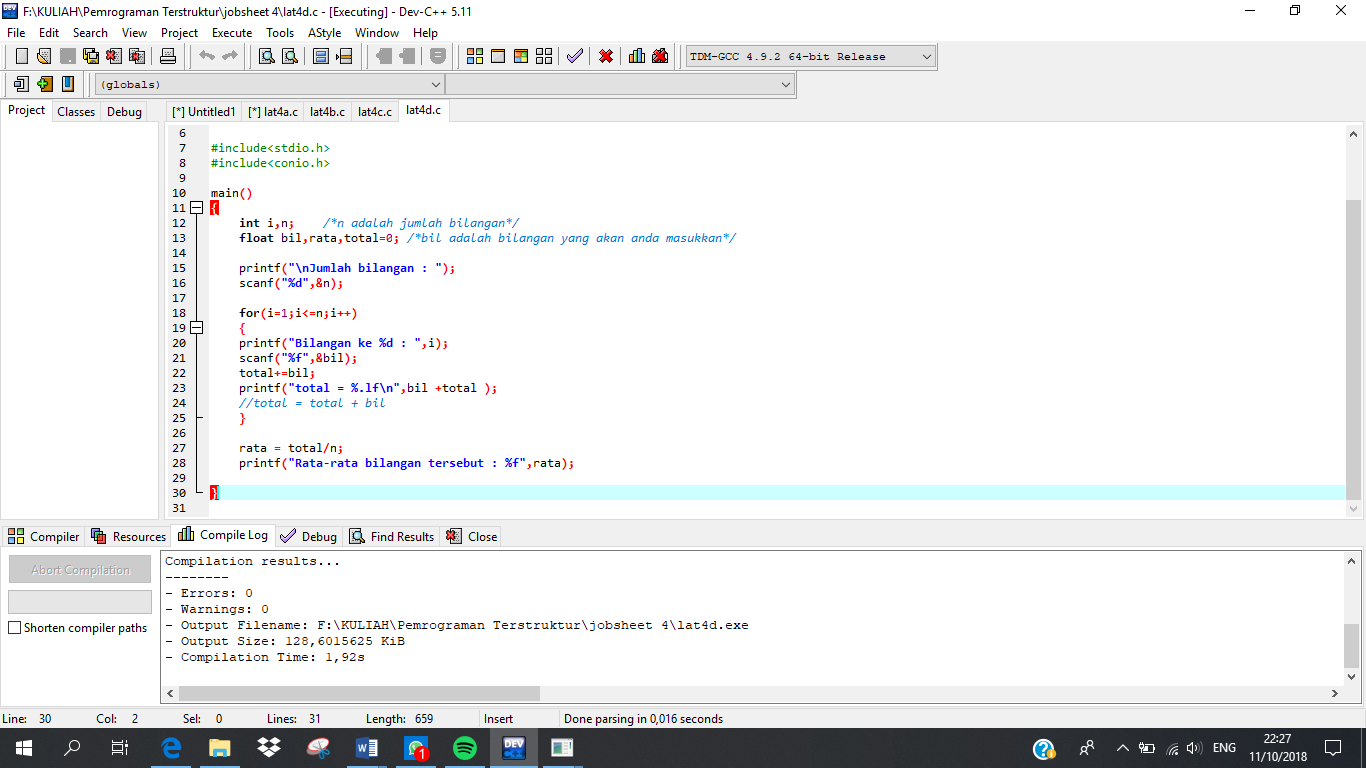
rata = total/n;

printf("Rata-rata bilangan tersebut : %f",rata);

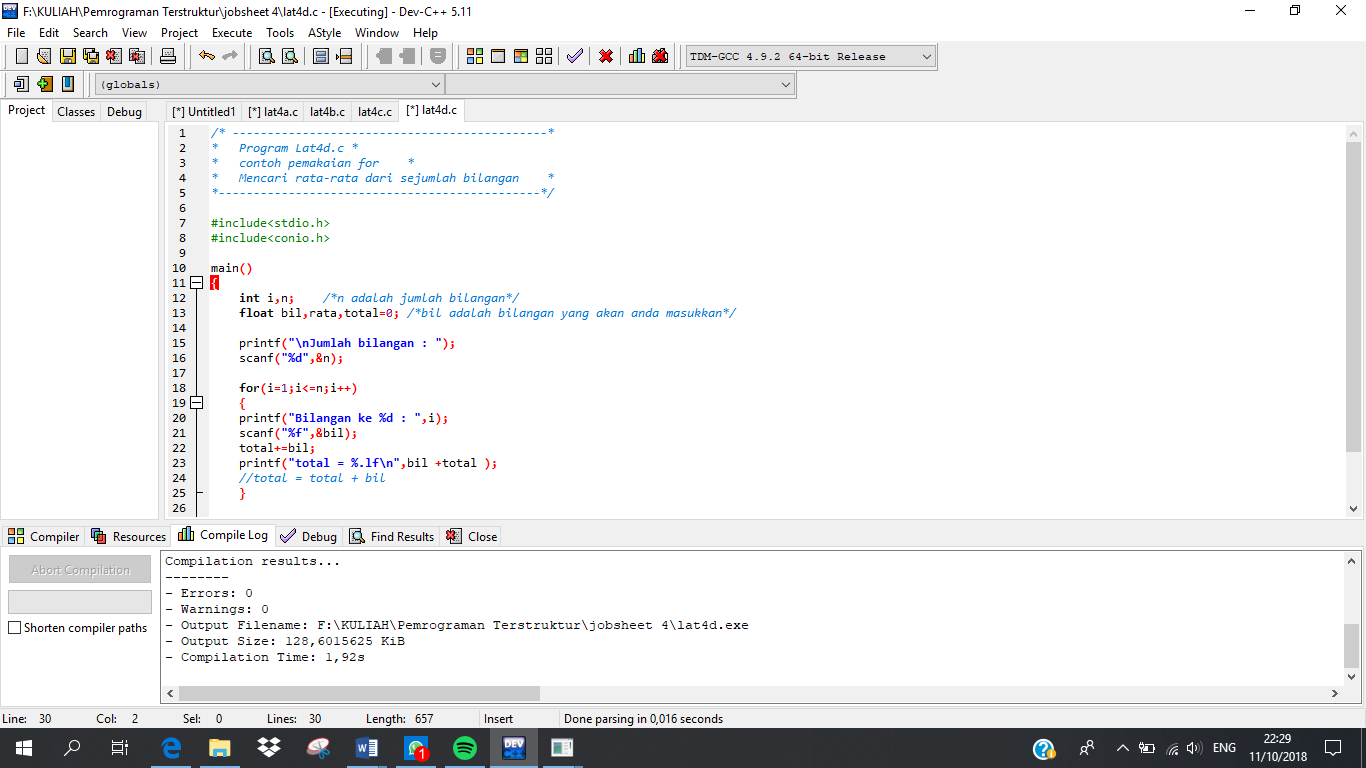
}

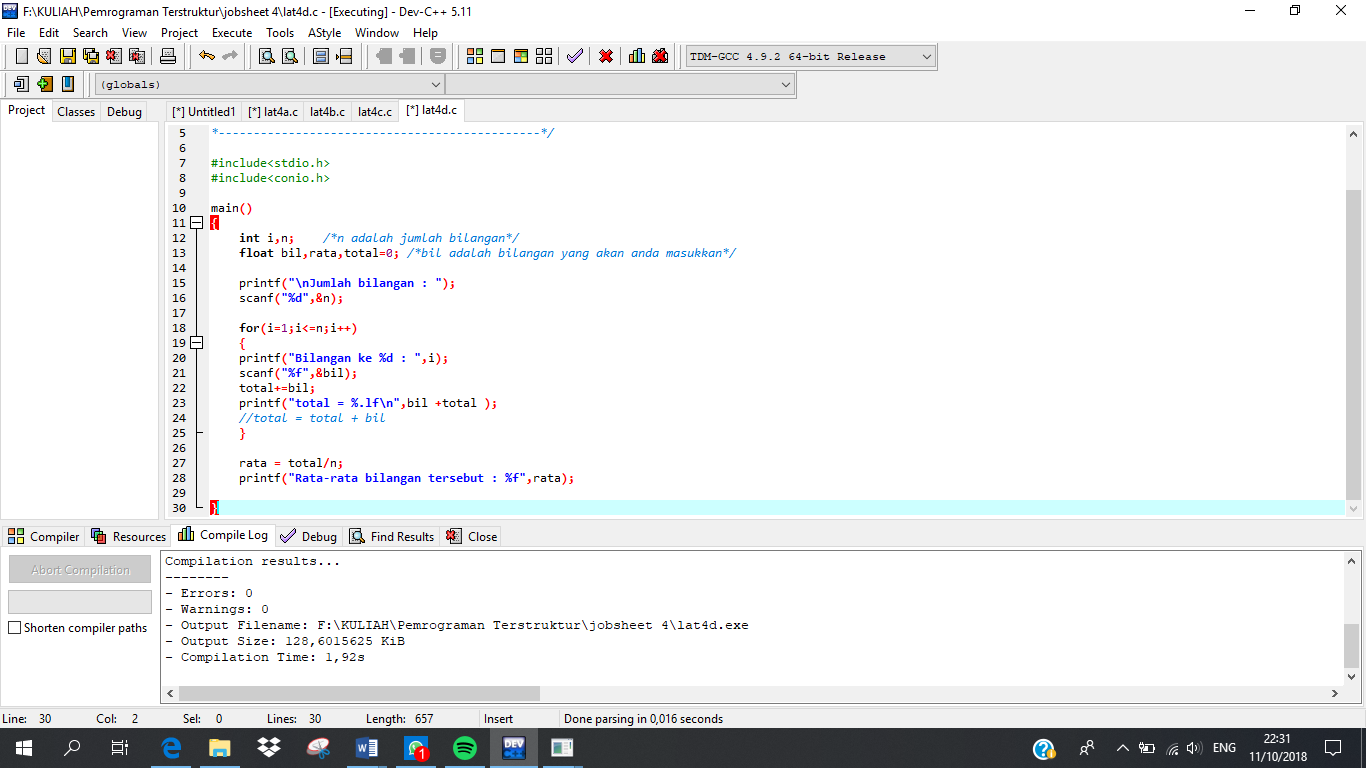
* 1. Alt+F, save dengan nama Lat4d.c

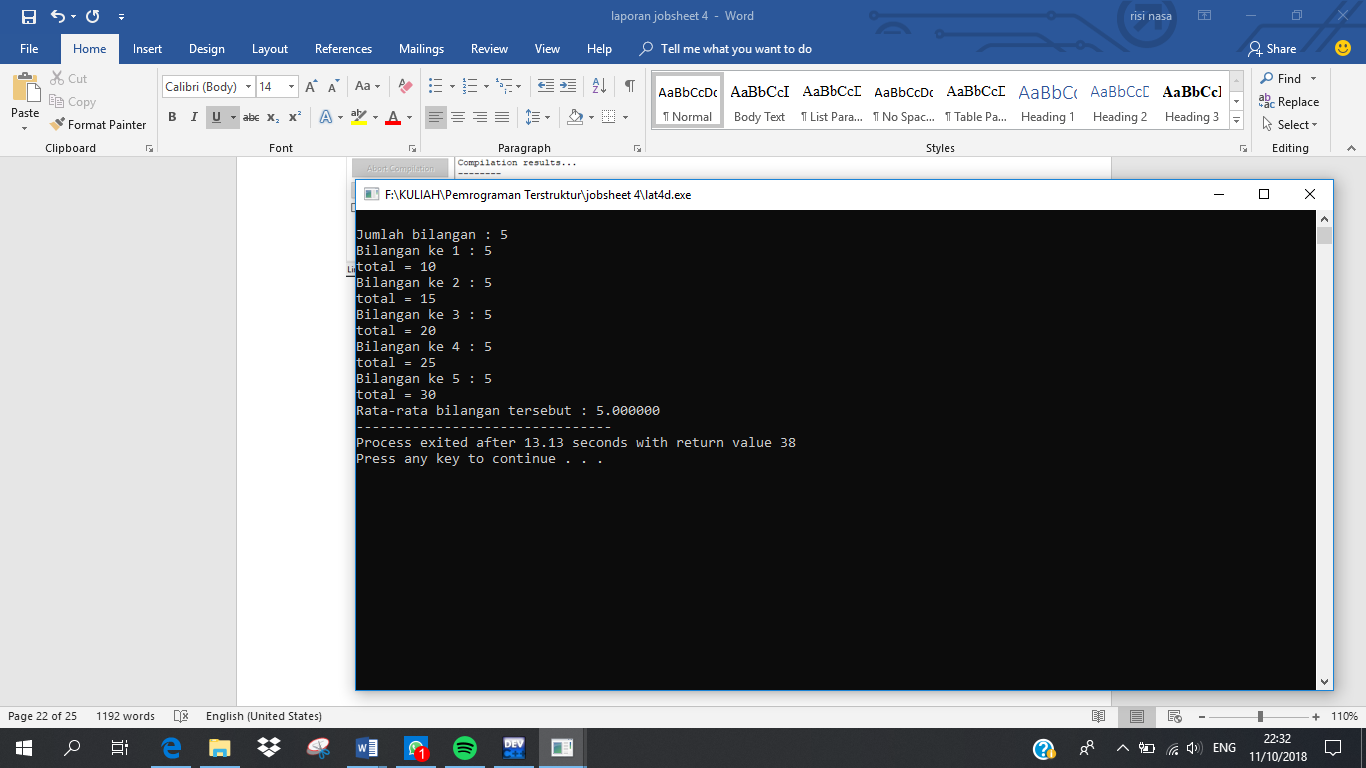


* 1.  Lalu Compile hingga tidak terdapat error dan dapat di run.

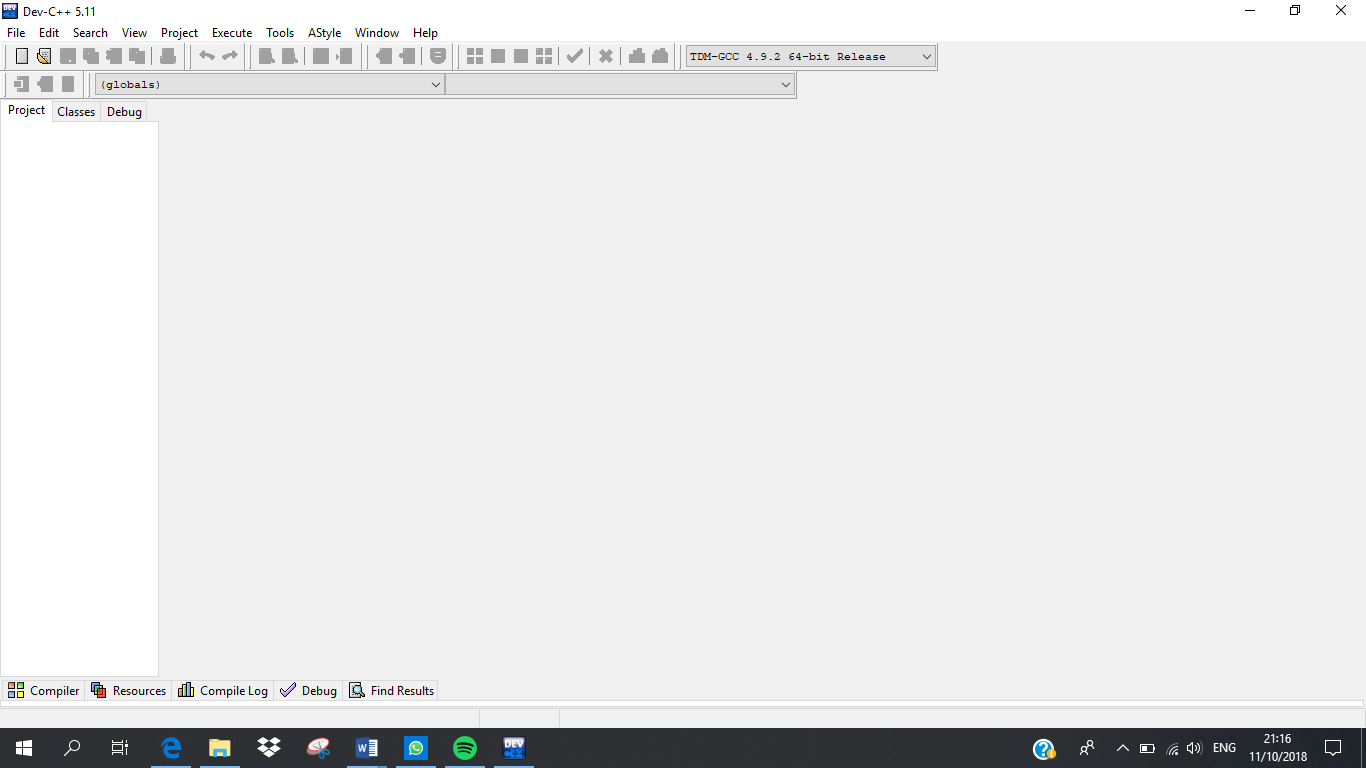
Setelah decompile tidak terdapat error

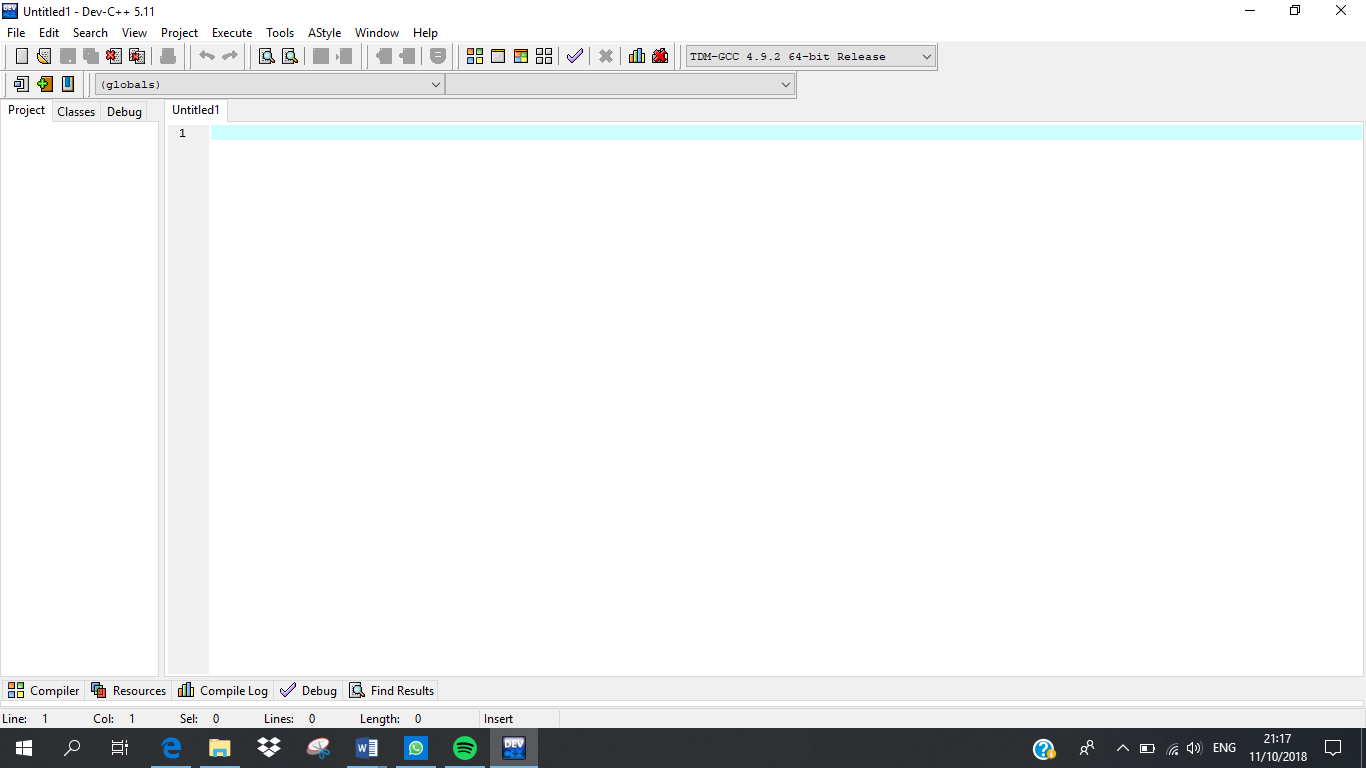
* 1.  Run program tersebut.

Hasil Run



1. **Membuat tabel perkalian studi kasus *nested loop* dengan struktur *for*** 
   1. Jalankan Dev C++



* 1. Alt+F  New
  2. Ketik :

/\* -------------------------------------\*

\* Program Lat4e.c \*

\* contoh pemakaian nested loop \*

\* membuat tabel perkalian \*

\*--------------------------------------\*/

#include <stdio.h>

#include <conio.h>

main()

{

int baris, kolom, hasil\_kali, maks = 0;

printf("masukkan jumlah maksimal baris / kolomnya : ");

scanf("%d",&maks);

/\*kalang nested loop\*/

for(baris = 1;baris <=maks ; baris++)

{

for(kolom = 1;kolom <=maks ; kolom++)

{

hasil\_kali = baris \* kolom; printf("%2d \t", hasil\_kali);

}

printf("\n"); /\*pindah baris\*/

}

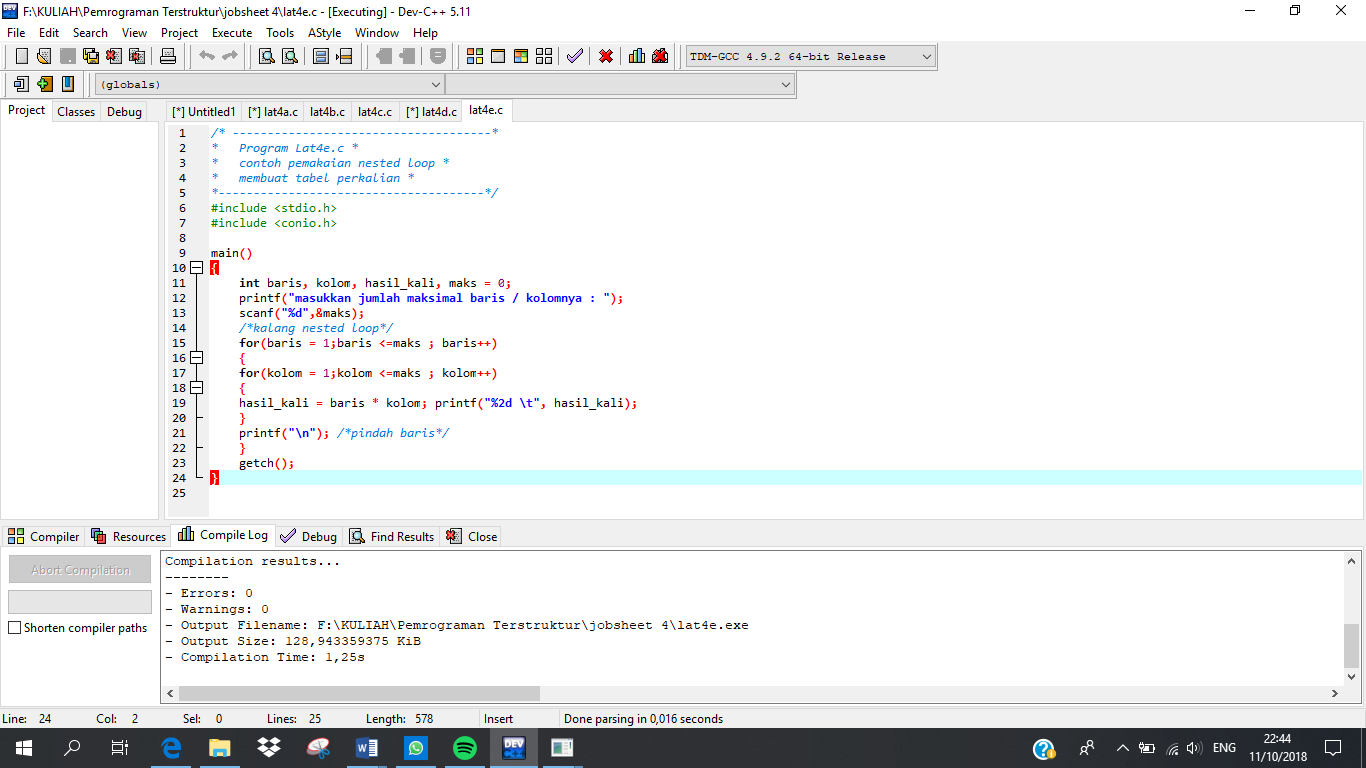
getch();

}

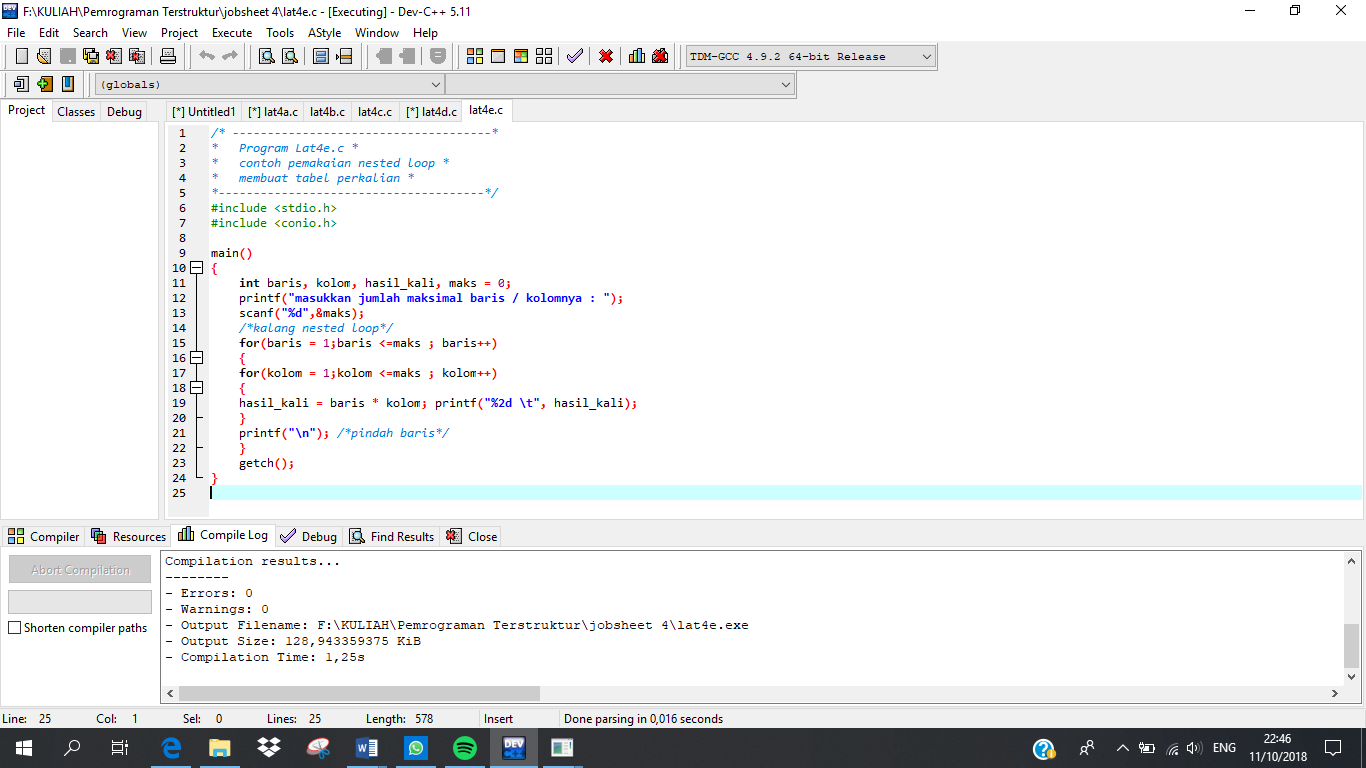
* 1. Alt+F, save dengan nama Lat4e.c

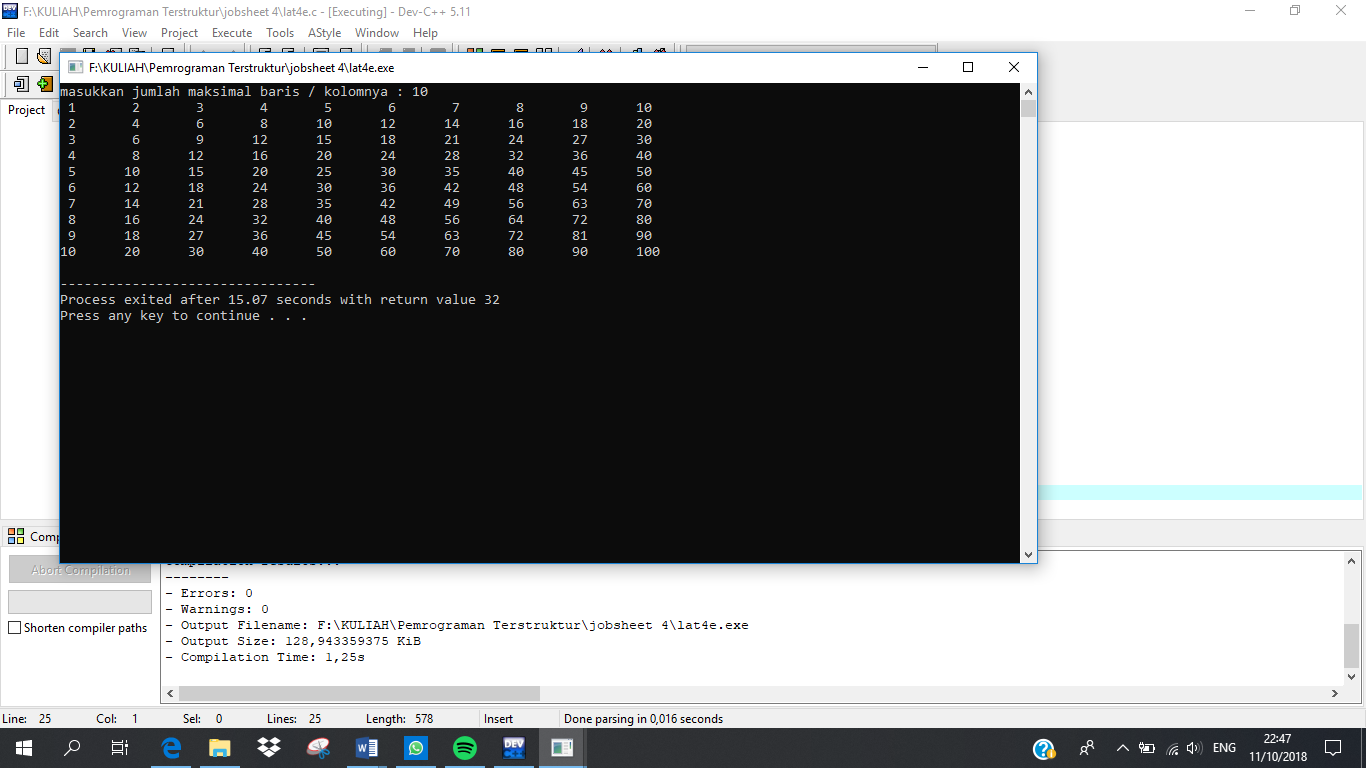


* 1. Lalu Compile hingga tidak terdapat error dan dapat di run.



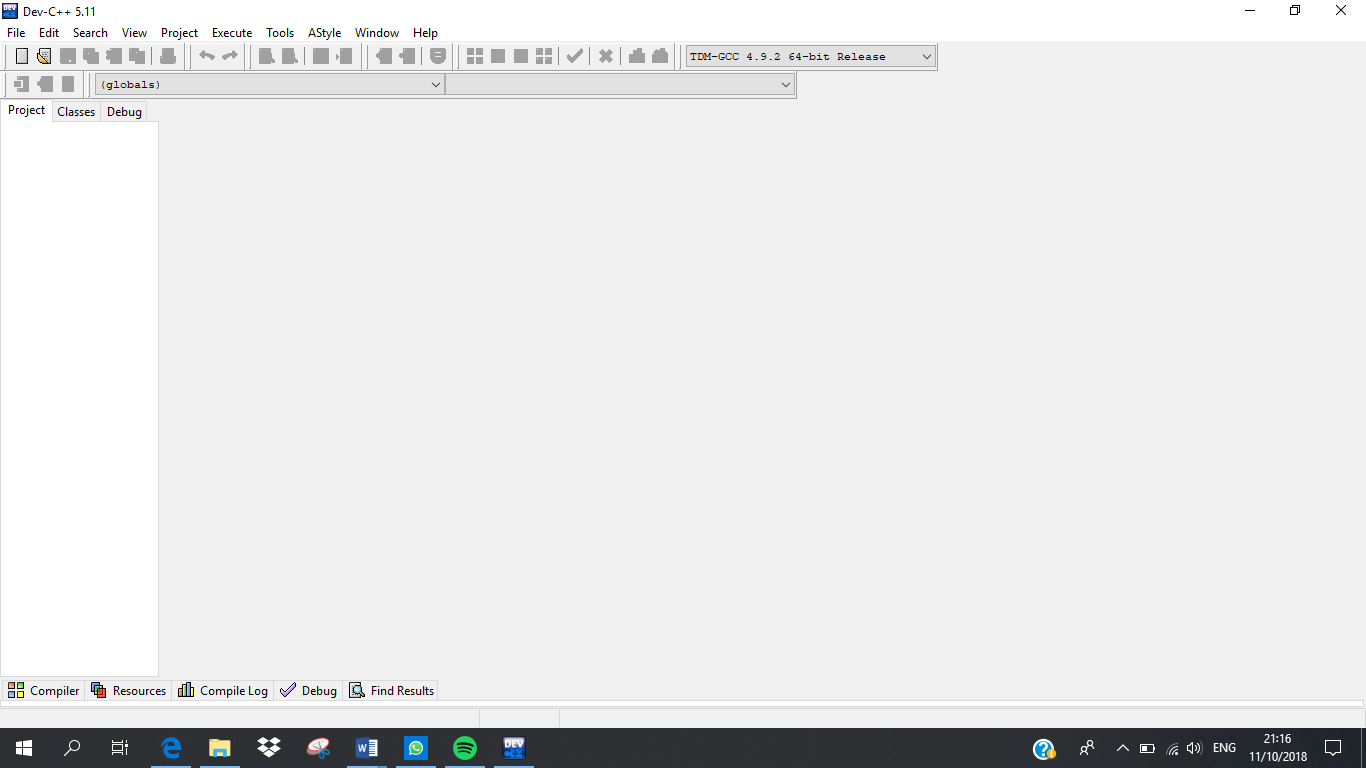
Setelah program di compile tidak terdapat error.

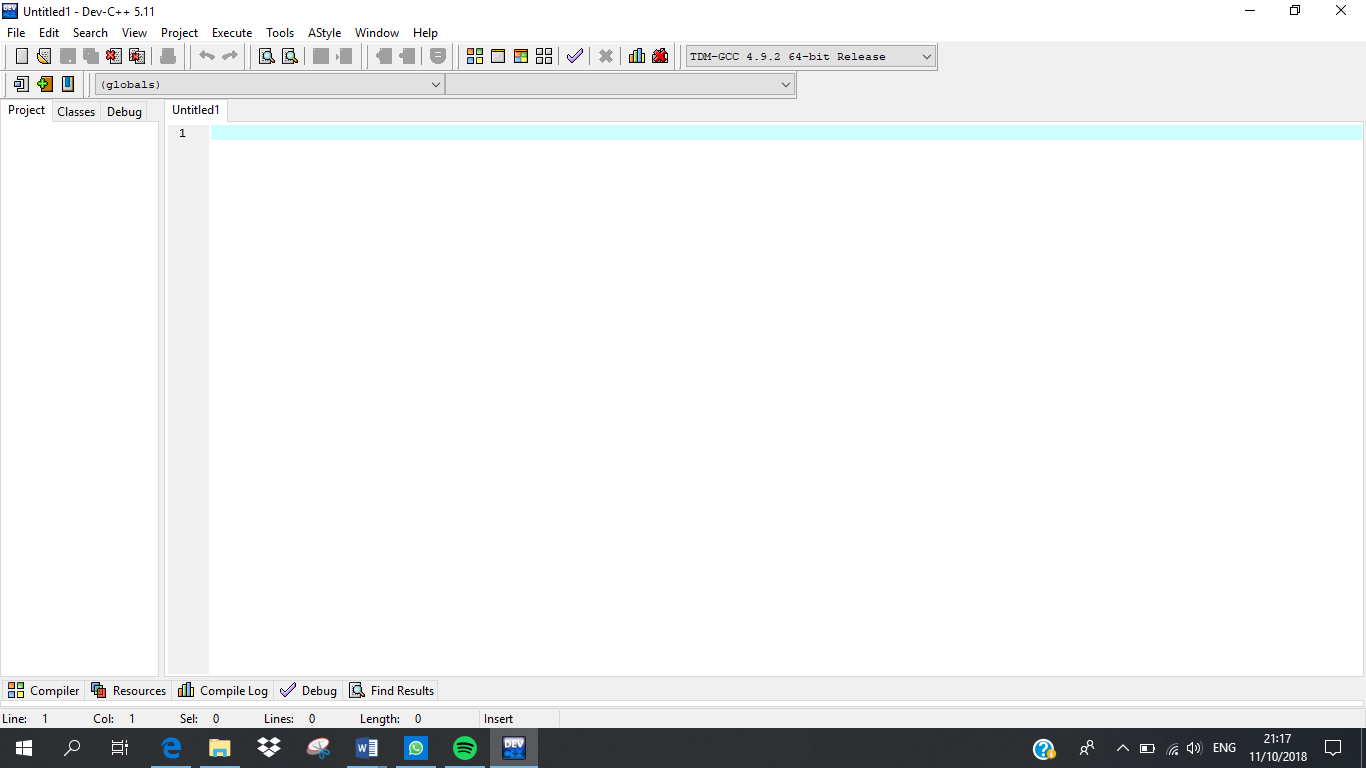
* 1.  Run program tersebut.

Hasil Run

### Membuat tabel perkalian studi kasus pemakaian pernyataan *break* :

* 1. Jalankan Dev C++



* 1. Alt+F New
  2. Ketik :

/\* ---------------------------------------------\*

\* Program Lat4f.c \*

\* contoh pemakaian break dan continue \*

\* Program menu sederhana \*

\* Operasi dua bilangan \*

\*----------------------------------------------\*/

#include <stdio.h>

#include<conio.h>

main()

{

char x ='1',ket;

float bil1,bil2,hasil; int i;

while(1)

{

/\*Cetak menu\*/

for(i=1;i<20;++i)

printf("\xDB");

printf("\n\n");

printf("\rOPERASI DUA BILANGAN");

printf("\n1.Penjumlahan");

printf("\n2.Selisih");

printf("\n3.Perkalian");

printf("\n4.Pembagian");

printf("\n5.Selesai");

printf("\n\n");

for(i=1;i<20;++i)

printf("\xDB");

printf("\n\nPilihan Anda :");

x=getche();

if(x=='5') //selesai

break;

/\*Pilihan yang salah\*/

if (x<'1' || x>'5')

{

printf("Anda salah memilih");

printf("\nTekan sembarang tombol");

getch();

continue;

/\*Kembali ke Menu\*/

}

/\*Pilihan yang benar\*/

printf("\nInputkan Bilangan pertama : ");

scanf("%f",&bil1);

printf("\nInputkan Bilangan kedua : ");

scanf("%f",&bil2);

switch(x)

{

case '1':

hasil=bil1+bil2;

ket='+';

break;

case '2':

hasil=bil1-bil2;

ket='-';

break;

case '3':

hasil=bil1\*bil2;

ket='\*';

break;

case '4':

hasil=bil1/bil2;

ket='/';

break;

}

/\*Cetak hasil\*/

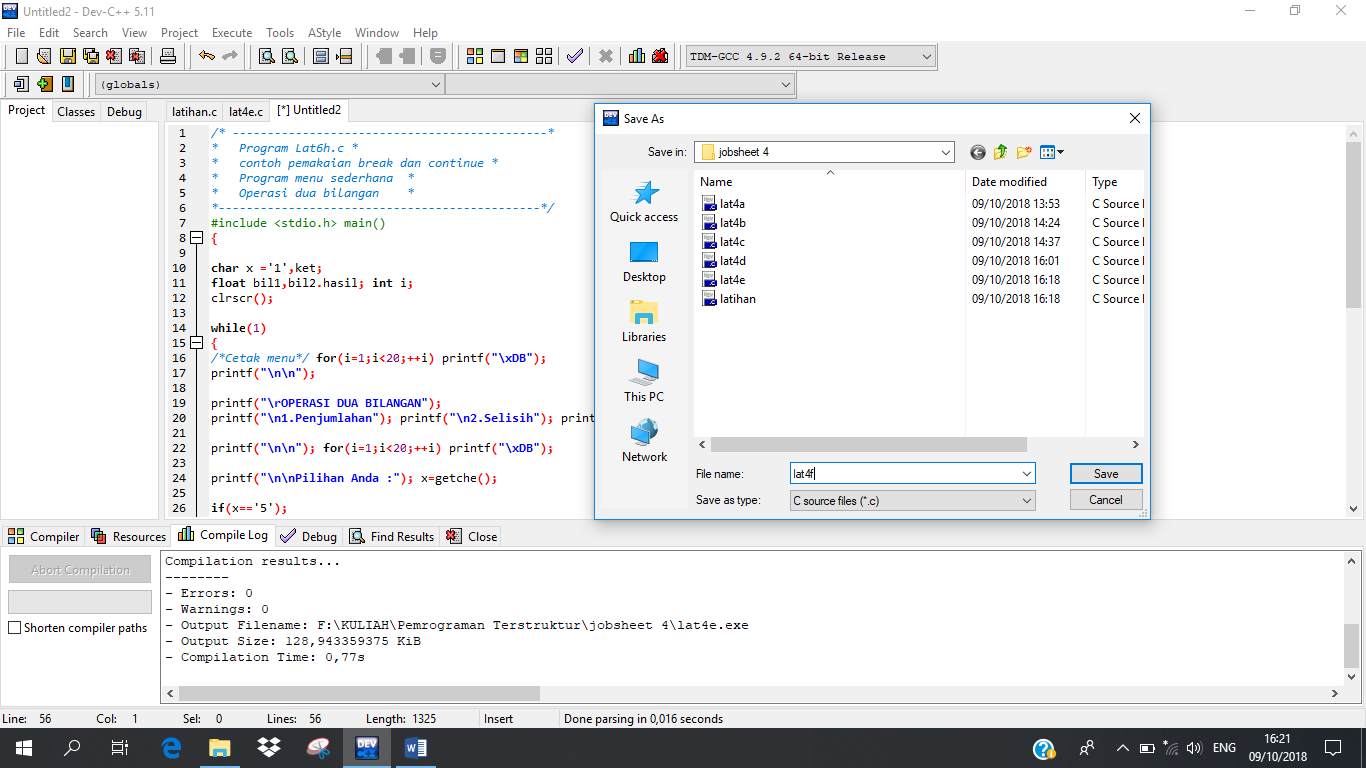
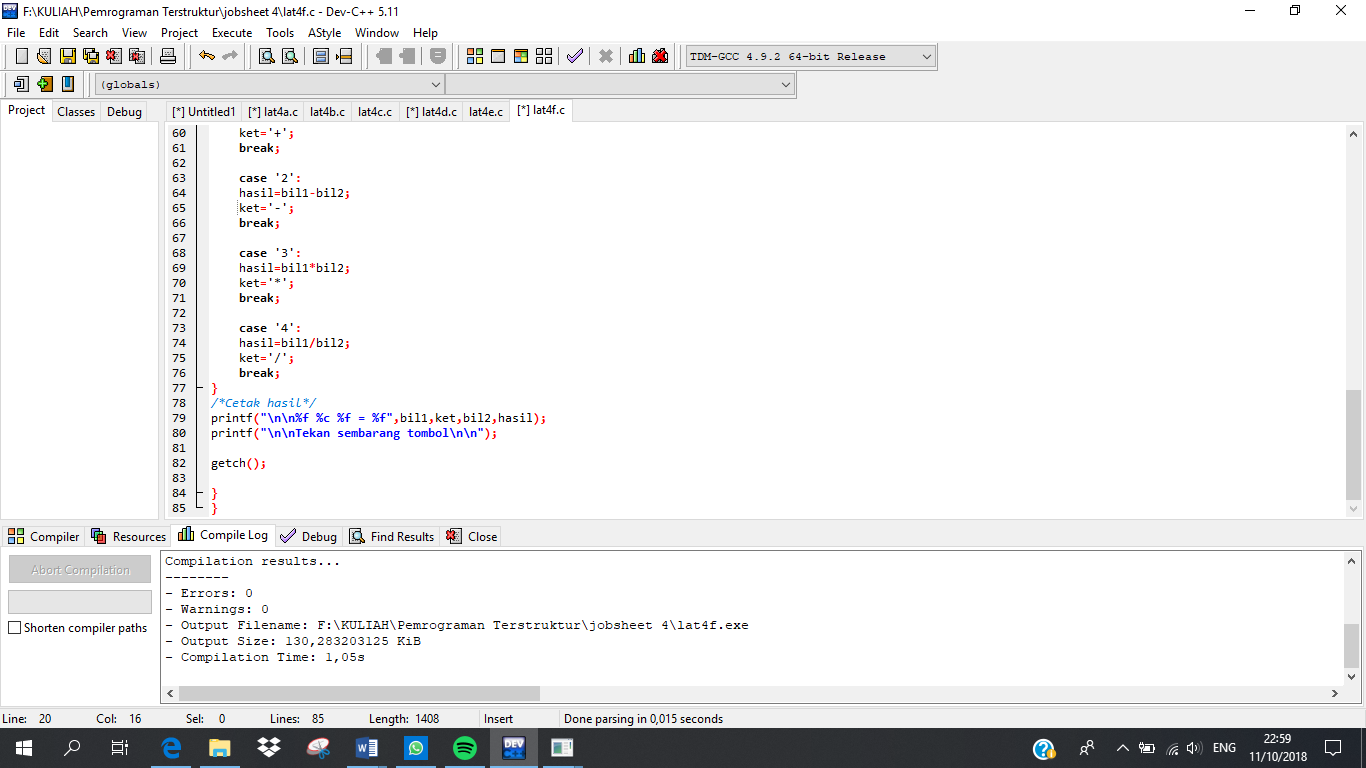
printf("\n\n%f %c %f = %f",bil1,ket,bil2,hasil);

printf("\n\nTekan sembarang tombol\n\n");

getch();

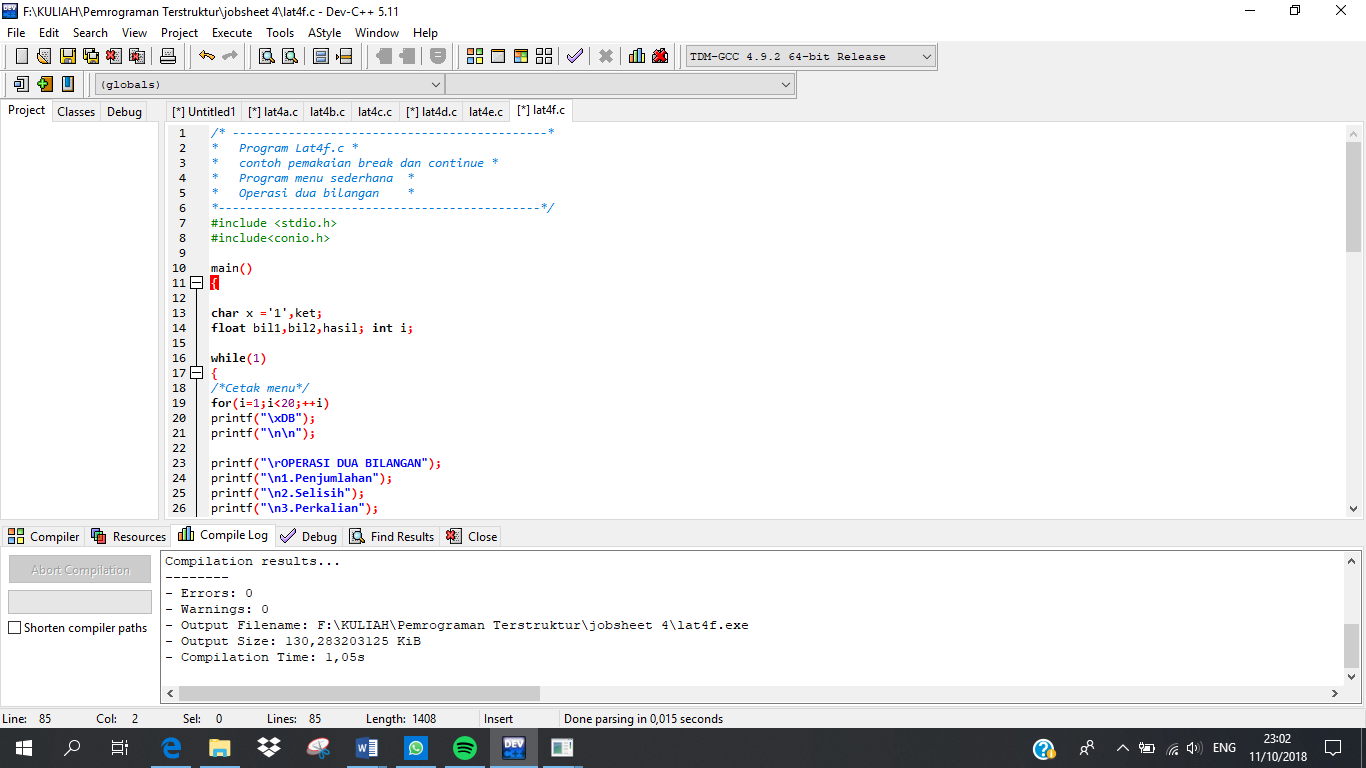
}

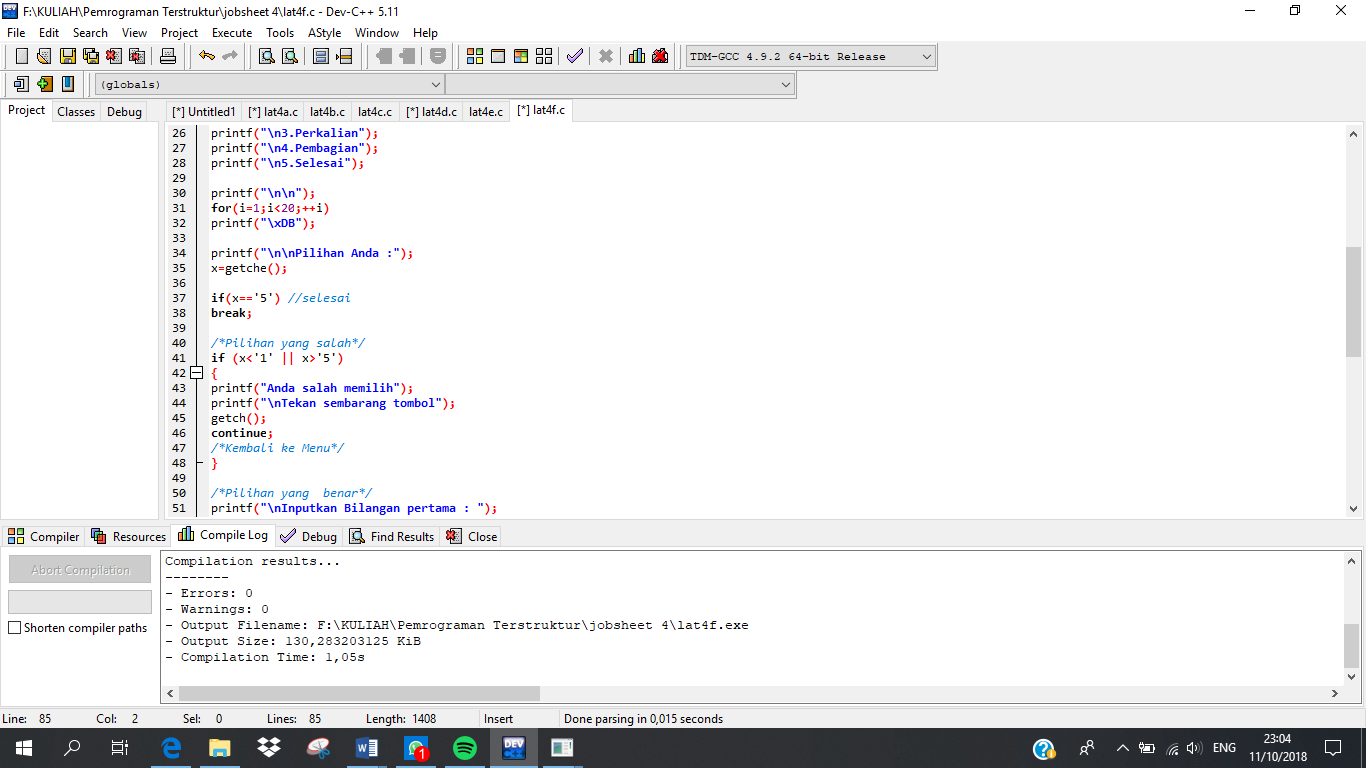
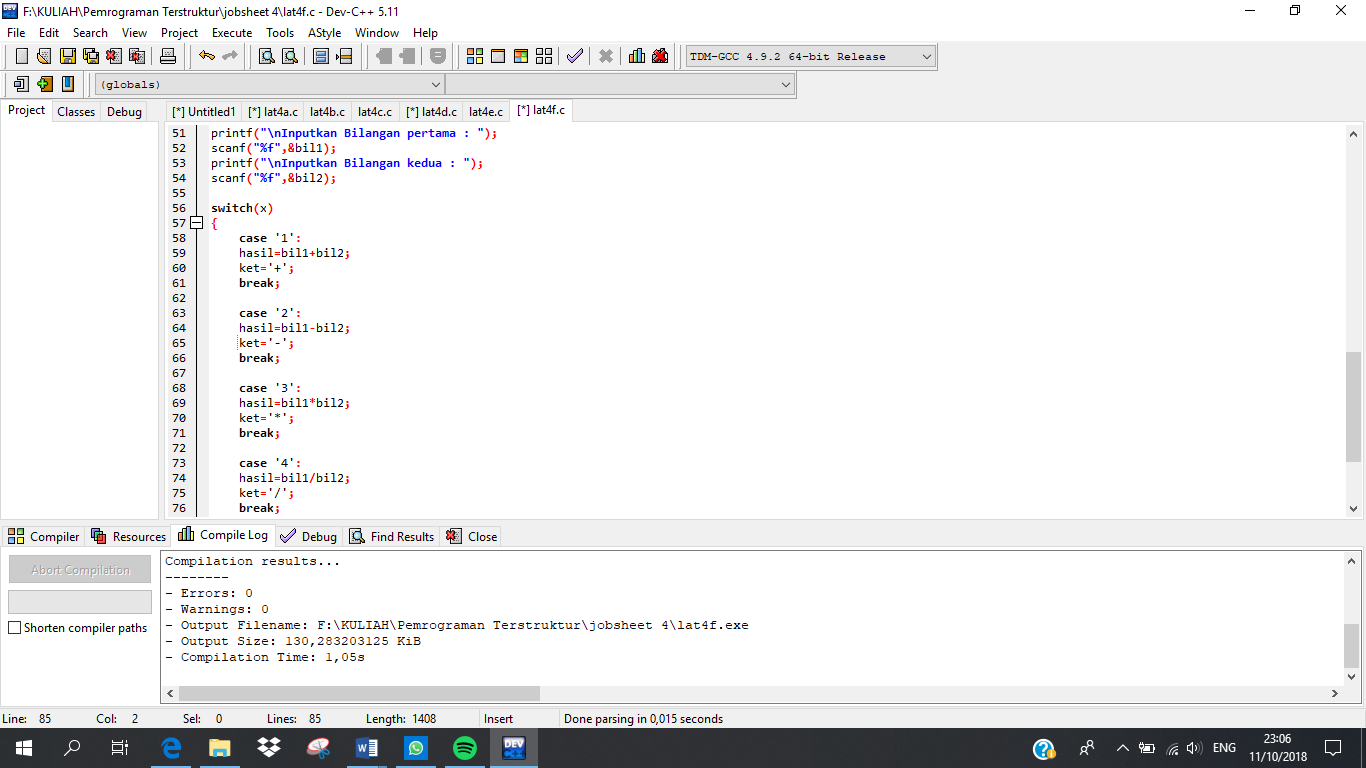
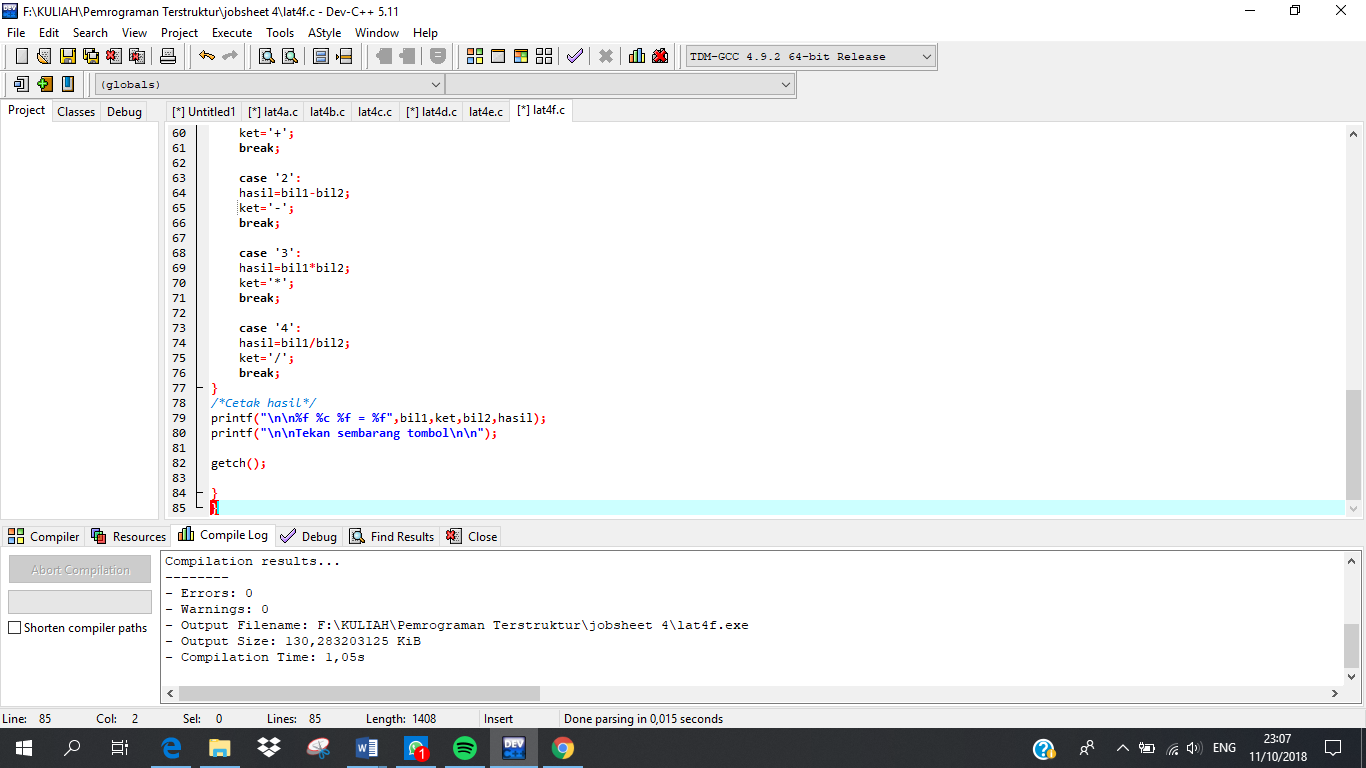
}

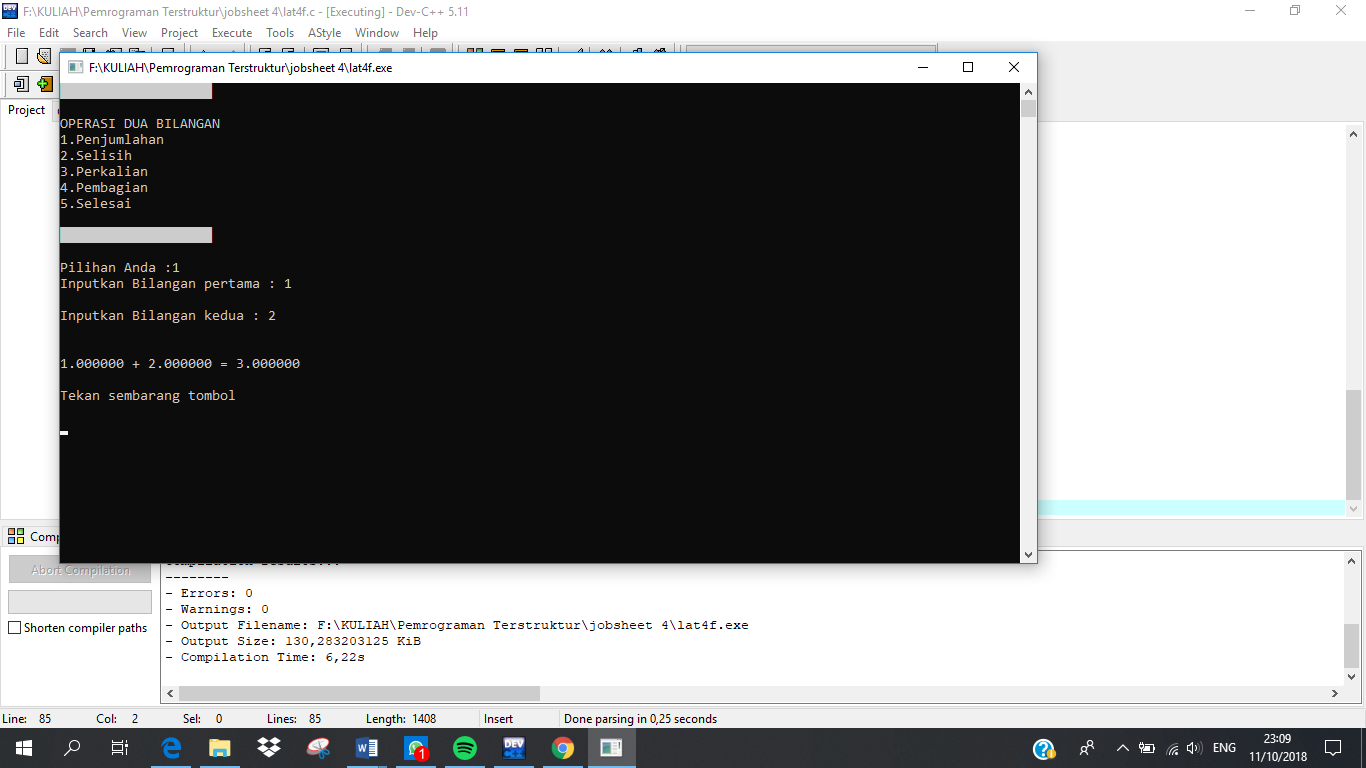
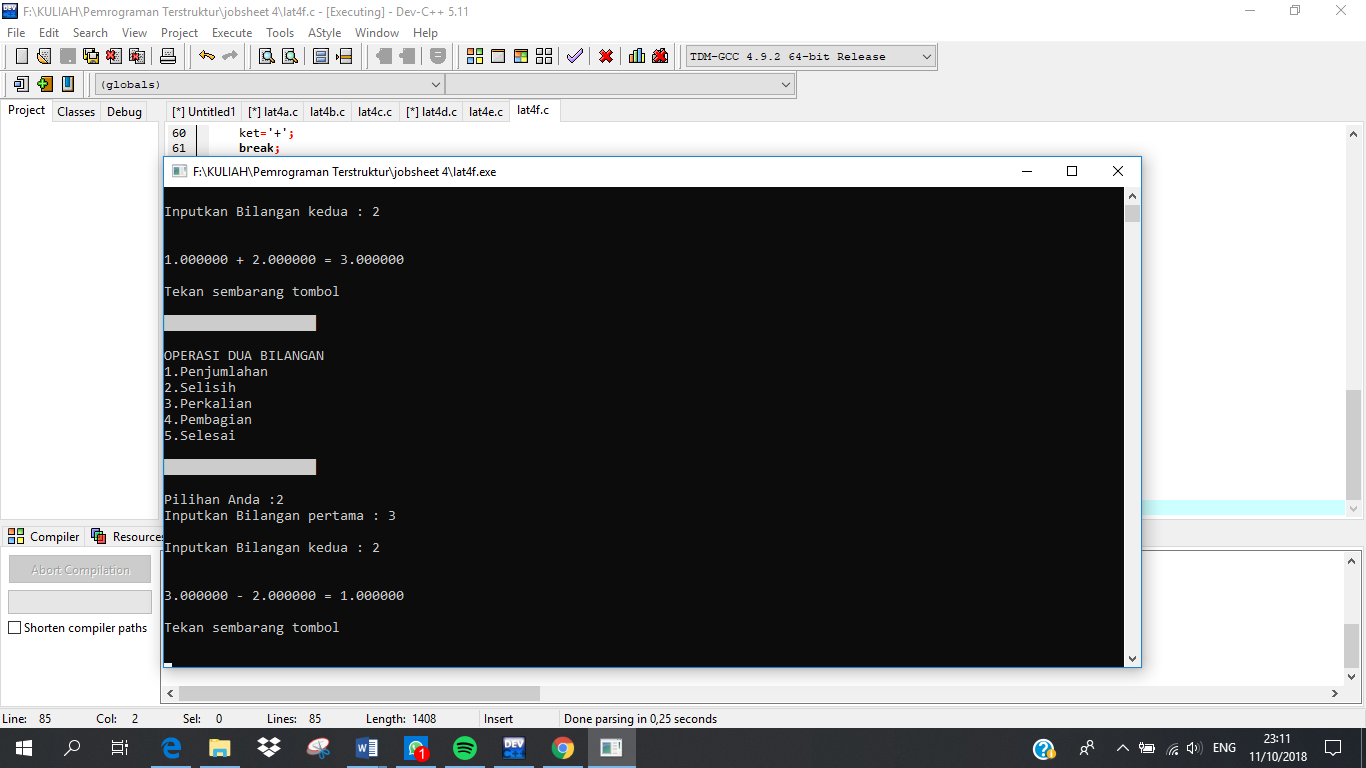
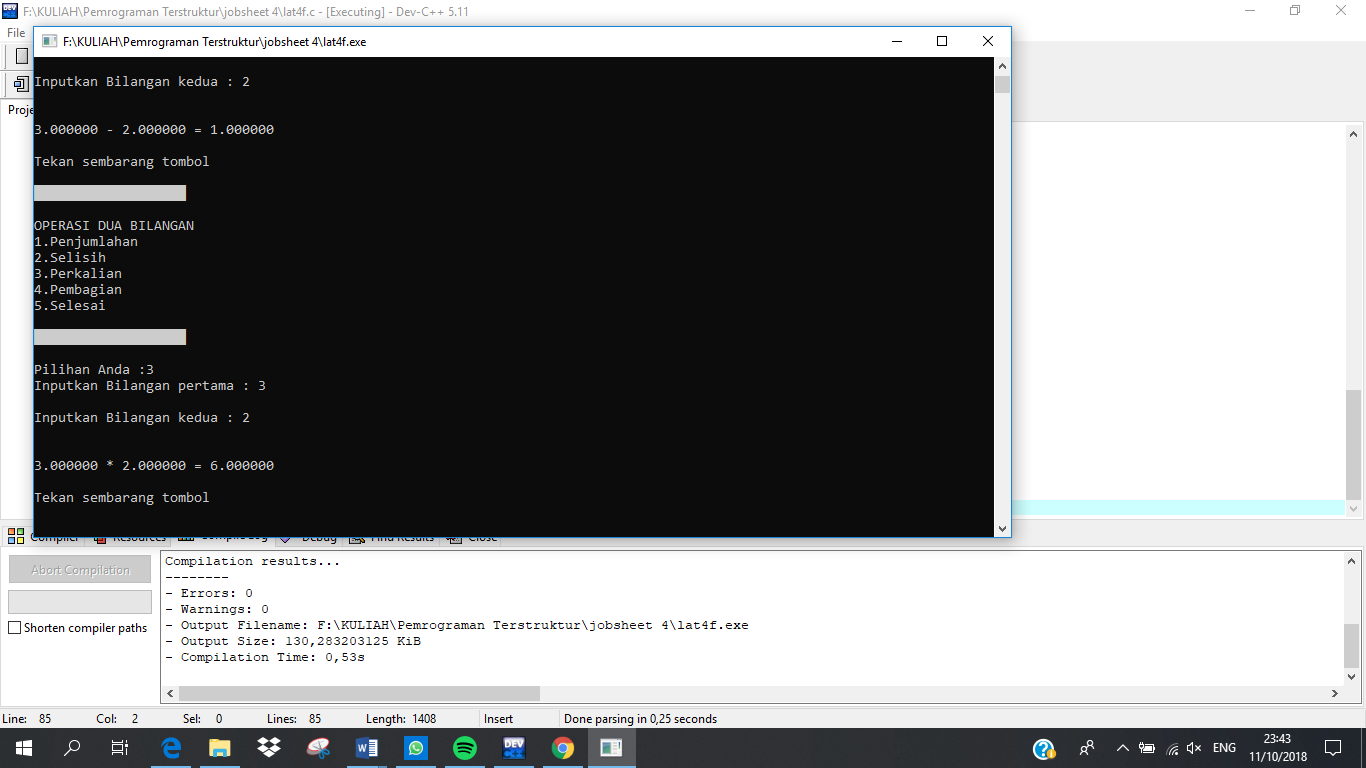
* 1. Alt+F, save dengan nama Lat4f.c
  2.  Lalu Compile hingga tidak terdapat error dan dapat di run.

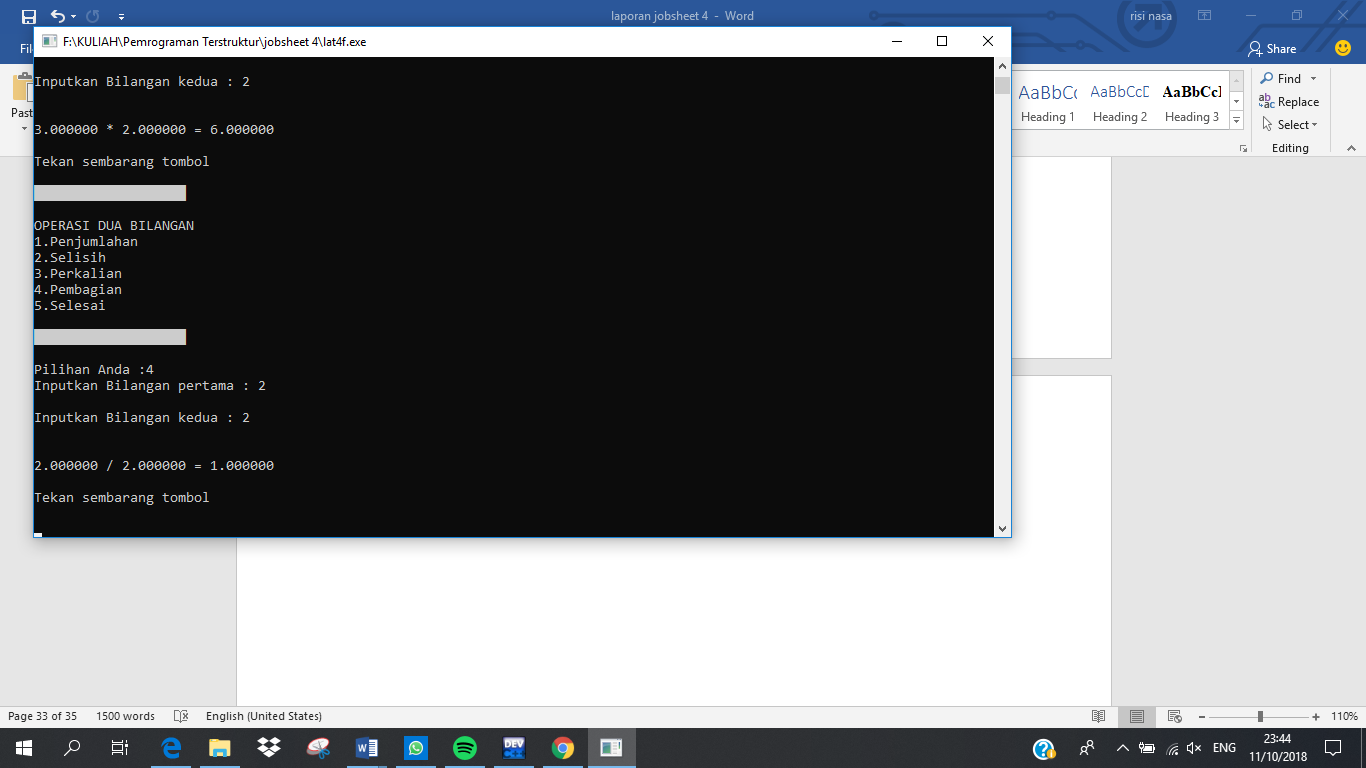
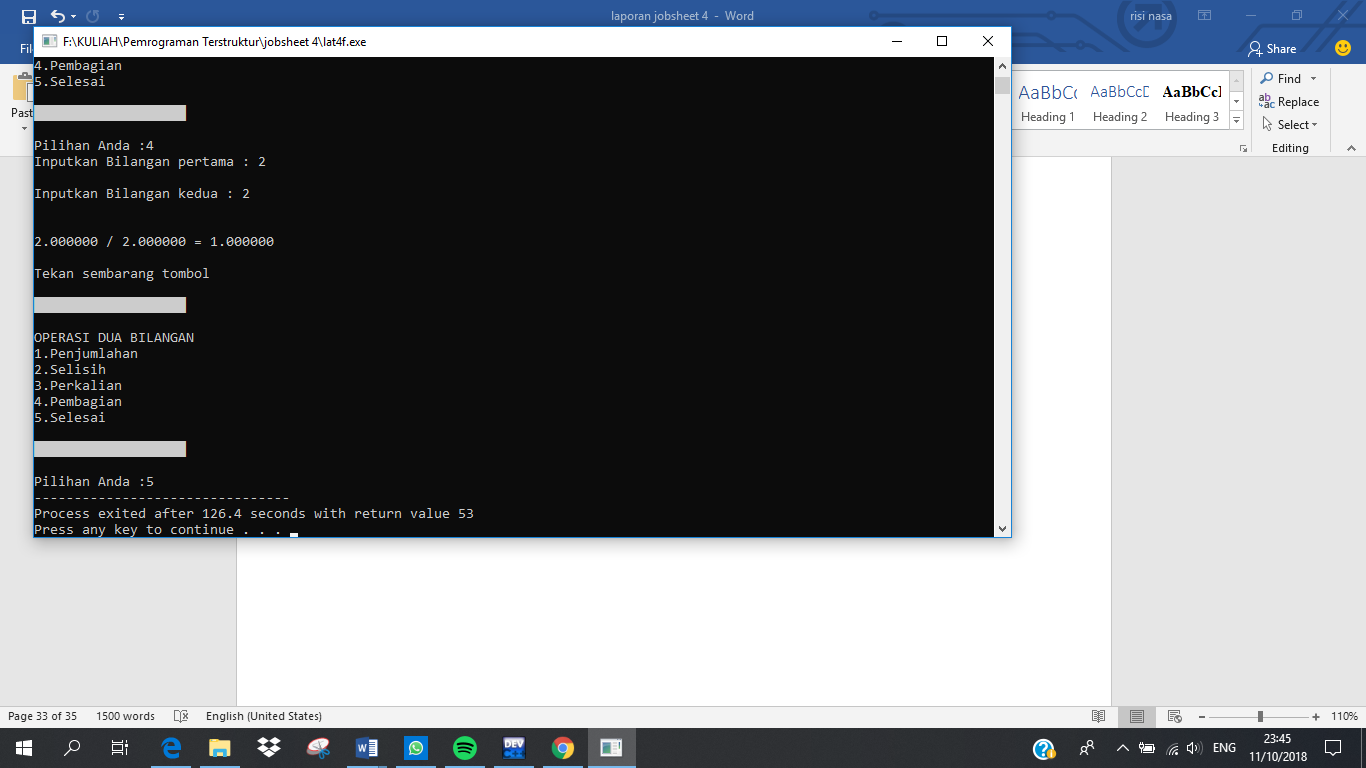
Setelah decompile tidak ada error

* 1. Run program tersebut.





 Hasil Run



### PERTANYAAN

### Apa yang anda ketahui tentang iterasi? Jelaskan dengan singkat dan jelas !

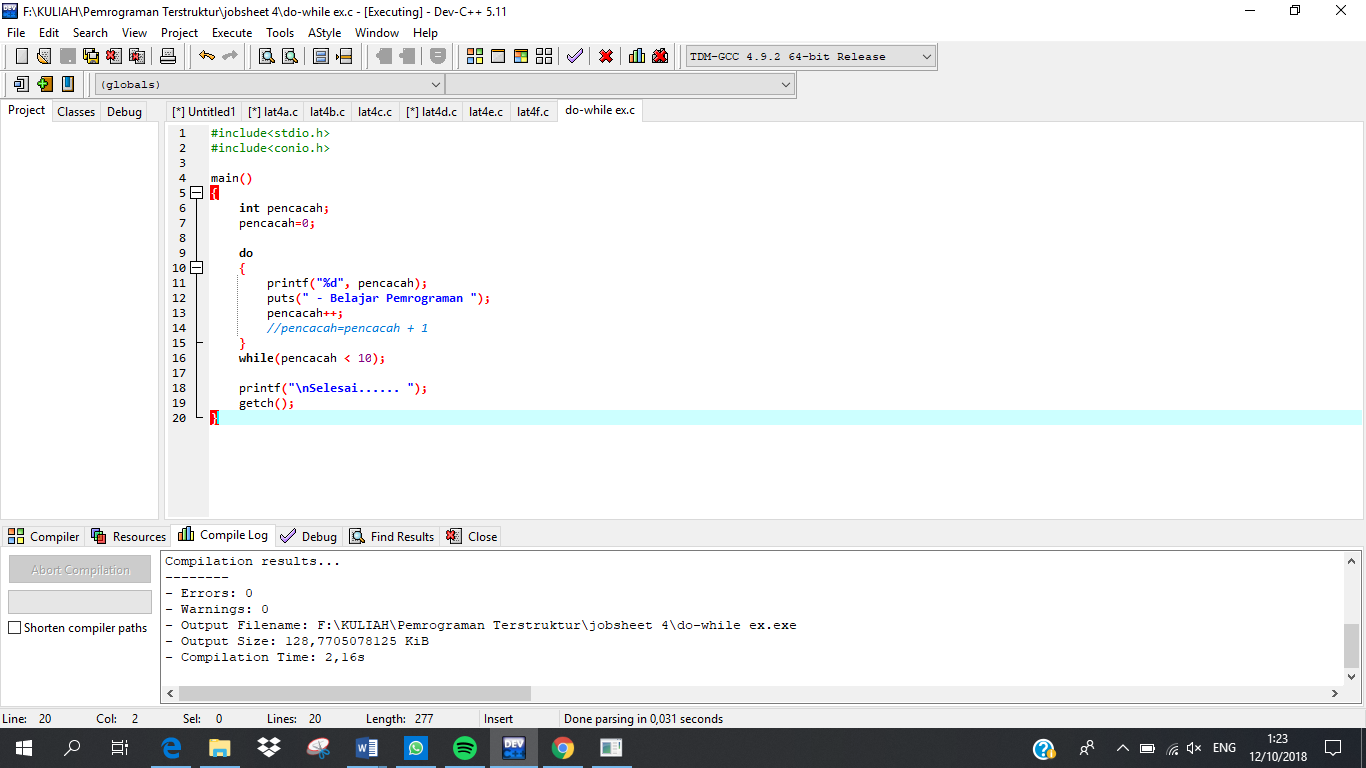
### 

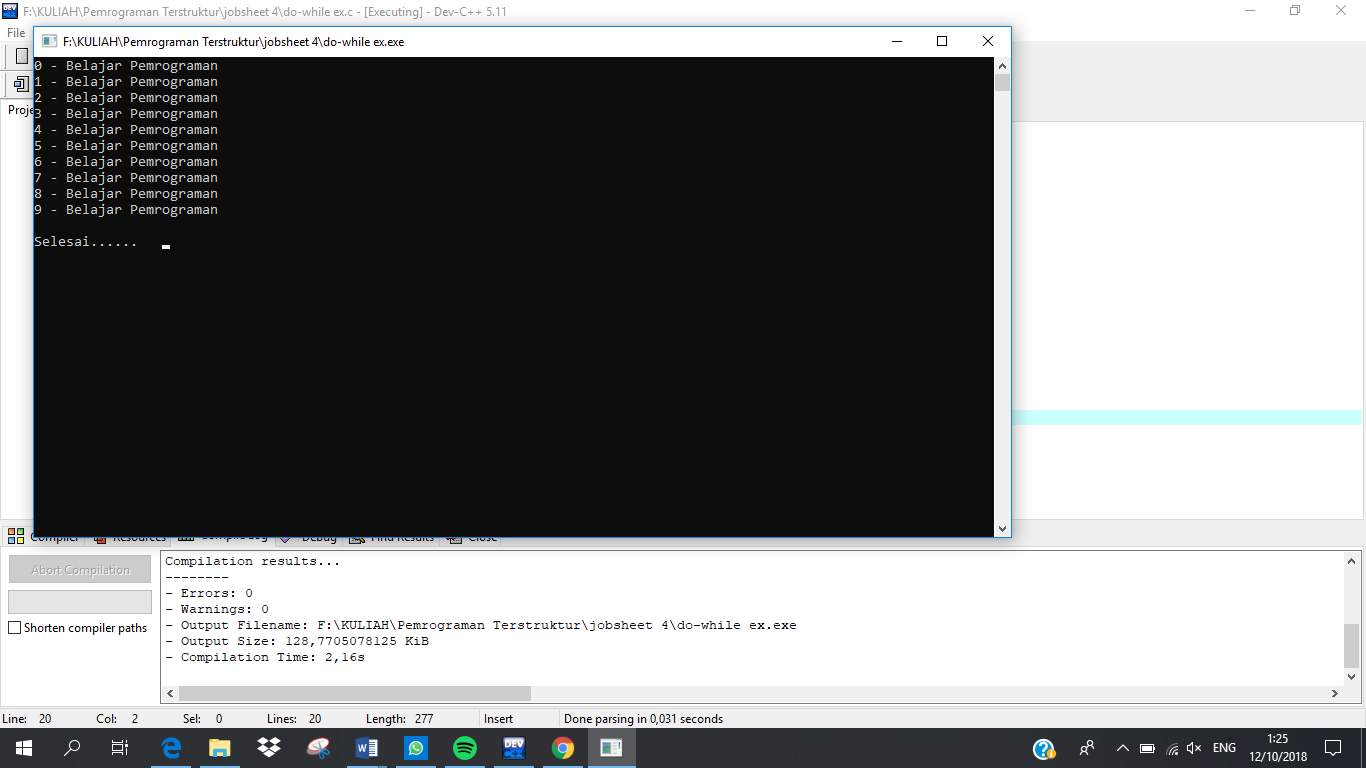
### Iterasi adalah suatu proses perulangan dan dapat diartikan sebagai suatu proses atau metode yang digunakan secara berulang-ulang (pengulangan) dalam menyelesaikan suatu permasalahan matematik.

1. Penyataan *do-while* dan *while* mempunyai kesamaan yaitu sama-sama mengetes kondisi. Jelaskan perbedaan keduanya dan disertai dengan contoh program!

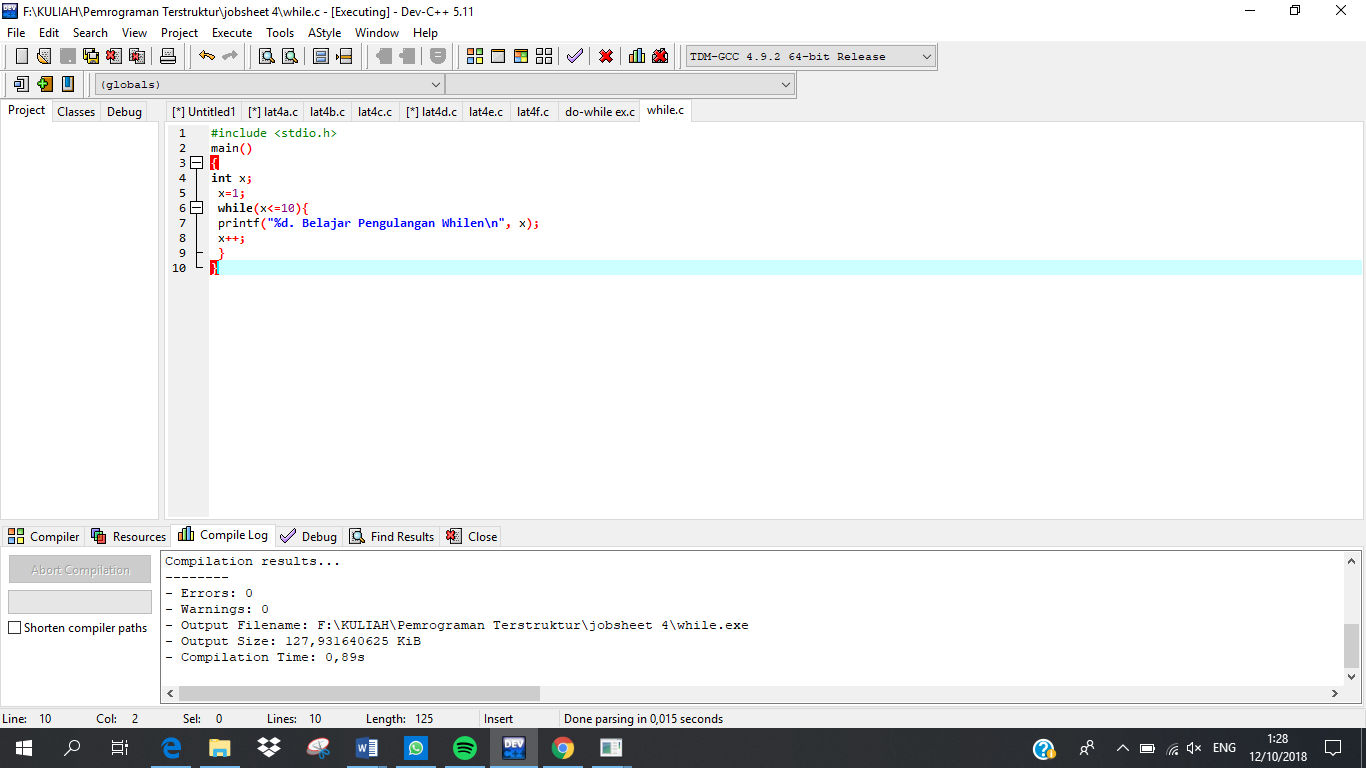
Perbedaan

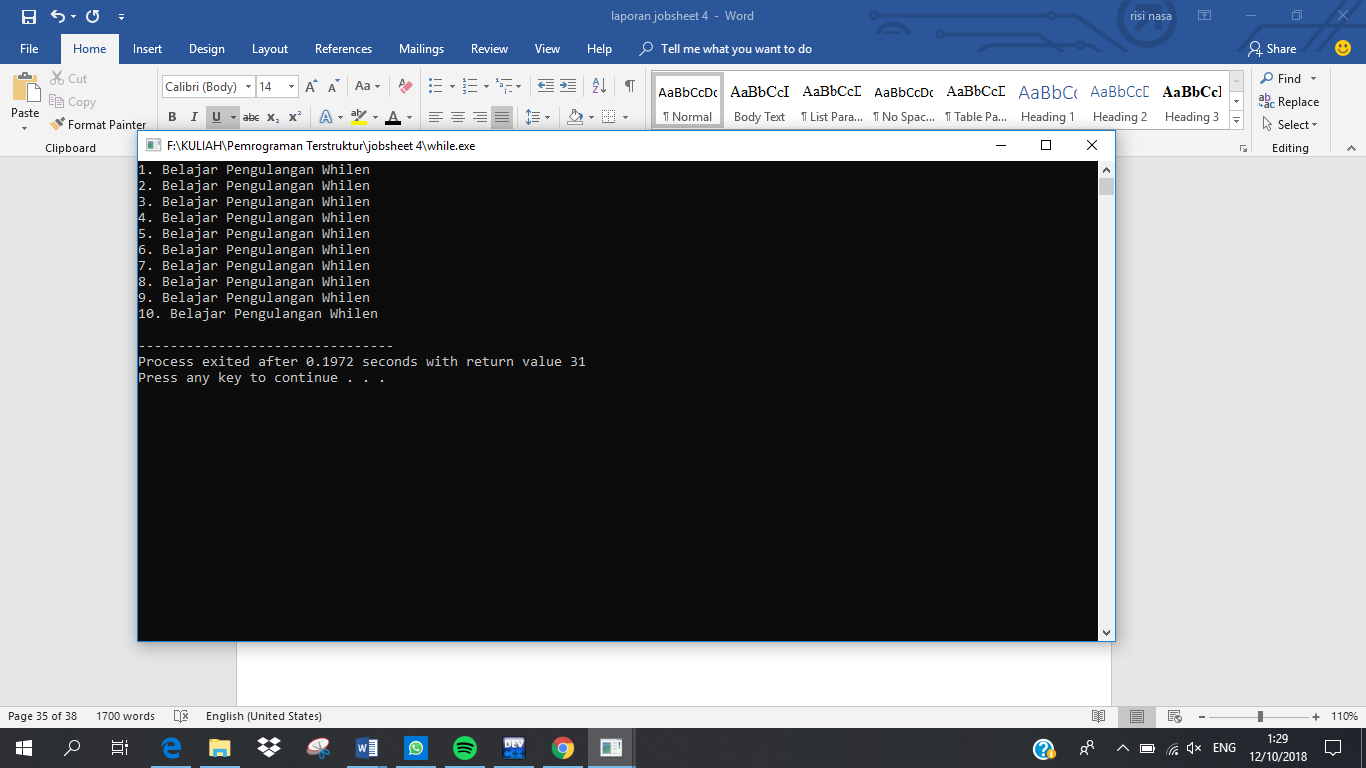
* do-while

 do-while metode perulanganya di jalankan dulu satu kali, dan di cek pada baris while, jika true maka perulangan di kerjakan jika false maka perulangan di hentikan.

Hasil Run

* while

 while metode perulangannya di cek dulu, dan jika benar (true) maka perulangannya dikerjakan, jika salah (false) maka perulangan dihentikan. Di jalankan dulu baru di cek, kalau benar dijalankan lagi.

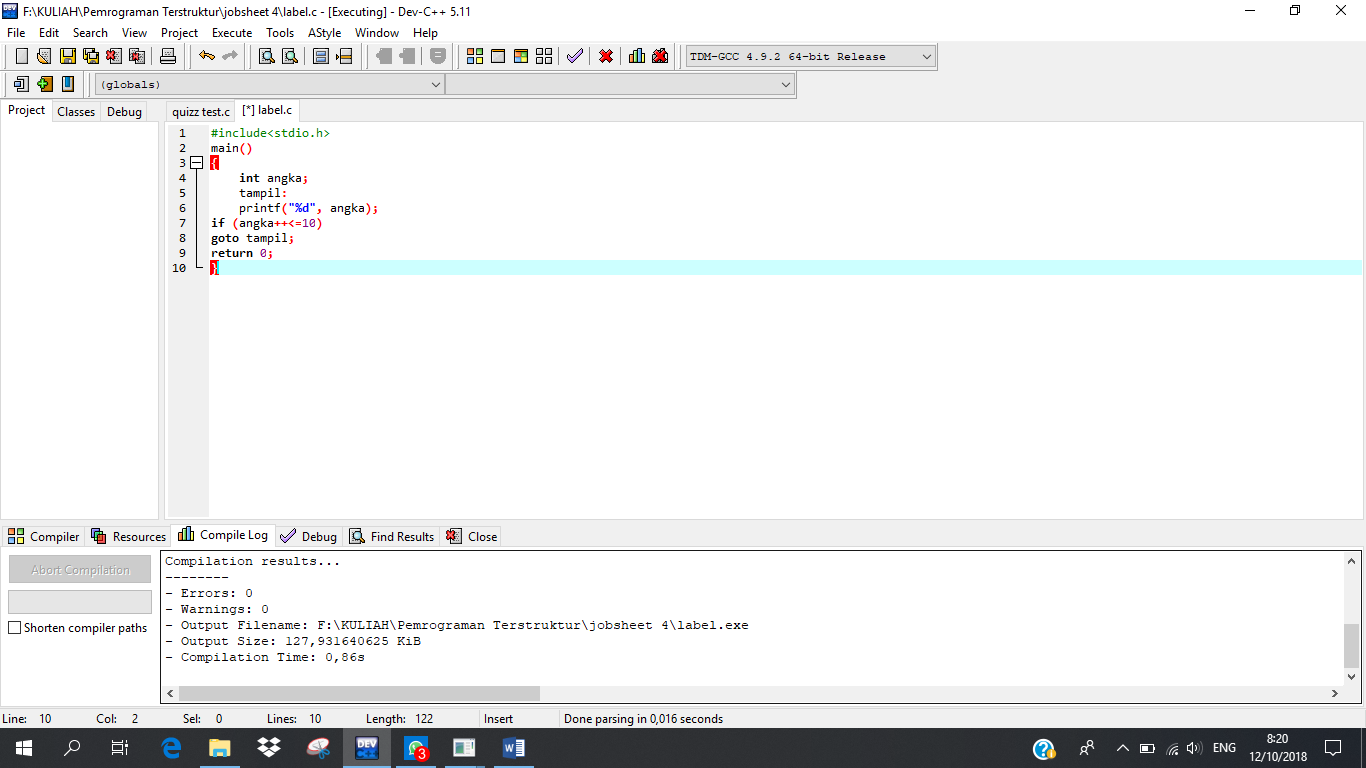
Hasil Run

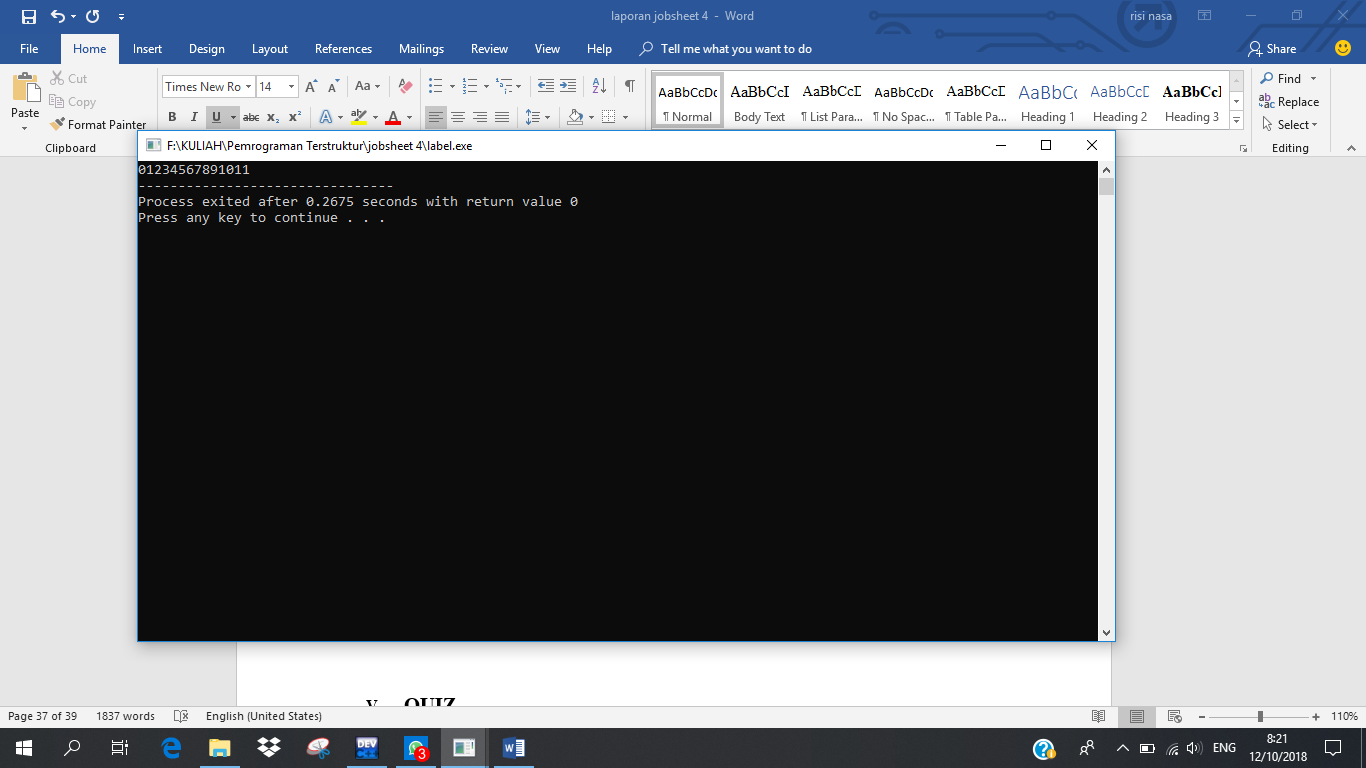
1. Jelaskan tentang kinerja pernyataan *break* dan continue yang dilibatkan dalam struktur perulangan!

* Untuk pernyataan **break** digunakan untuk melompati perintah sekaligus memberhentikan perulangan walau kondisinya masih bernilai true atau keluar dari perulangan. Jika pernyataan break dikerjakan, maka proses akan keluar dari loop kemudian dilanjutkan ke pernyataan selanjutnya.
* Sedangkan **continue** yaitu perintah yang digunakan untuk melompati perintah lainnya pada body loop dan akan memulai perulangan lagi. dengan kata lain mengembalikan proses yang sedang dilaksanakan ke-awal loop lagi, tanpa menjalankan sisa perintah dalam loop tersebut.

1. Apa yang kamu ketahui tentang label? Jelaskan dan disertai pula dengan contoh programnya !

Label yaitu tanda yang berfungsi untuk menandai suatu baris program, tempat dimana loncatan CPU digunakan GOTO mendarat.



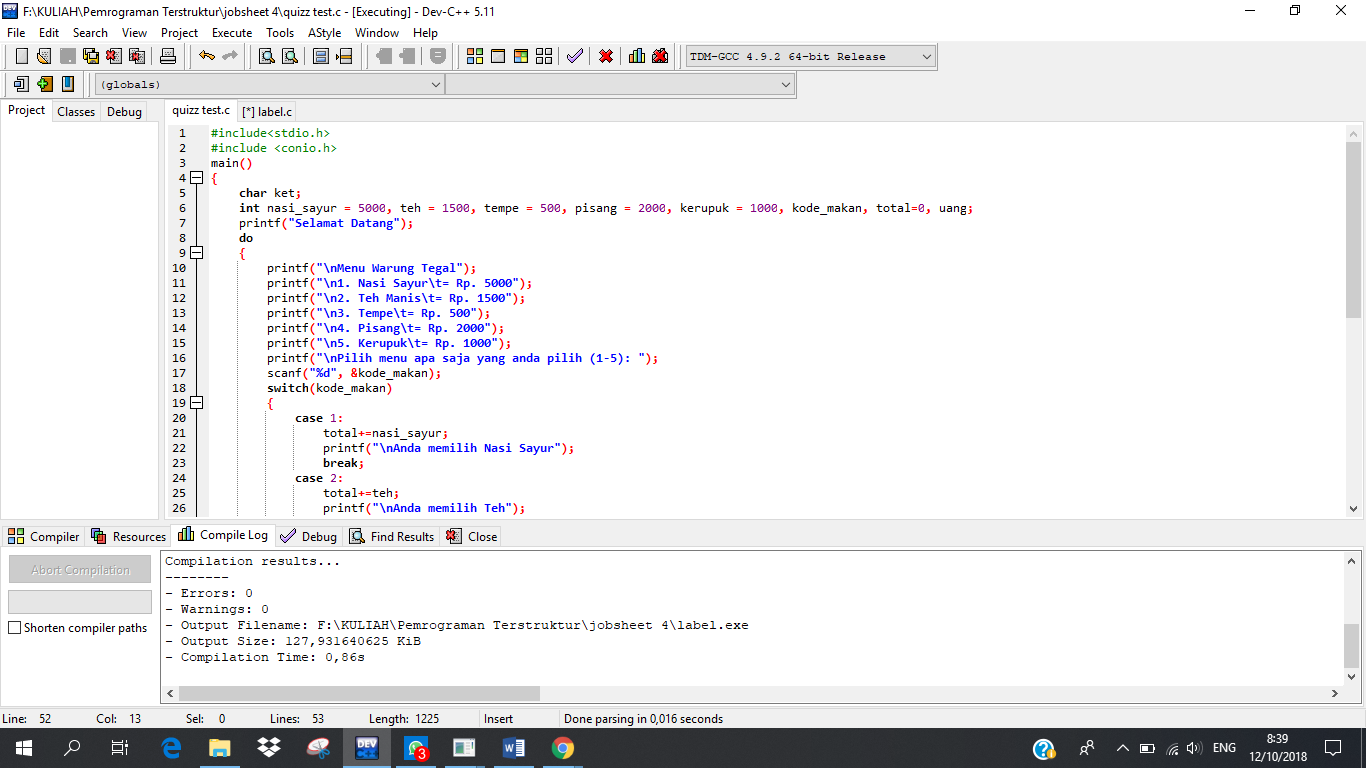
Hasil Run 

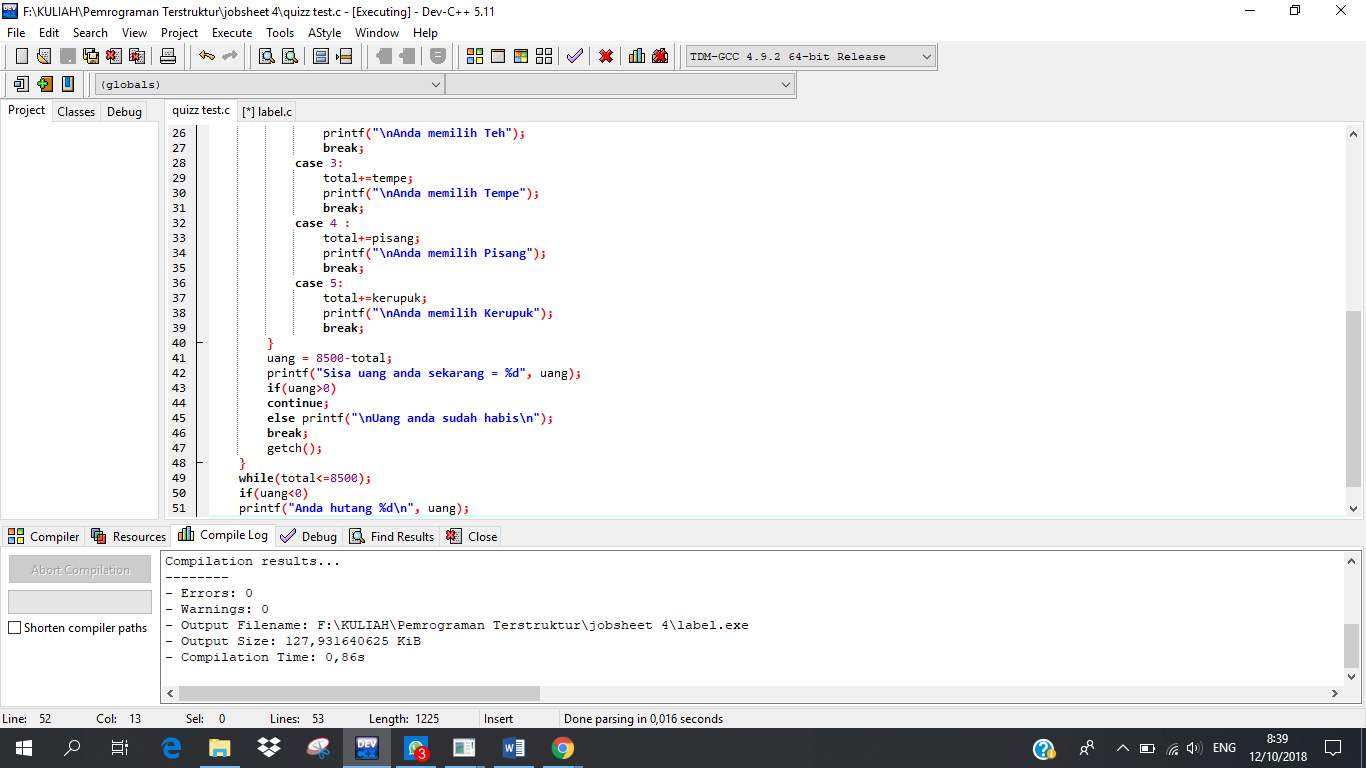
1. Apa guna dari fungsi *exit()* ?

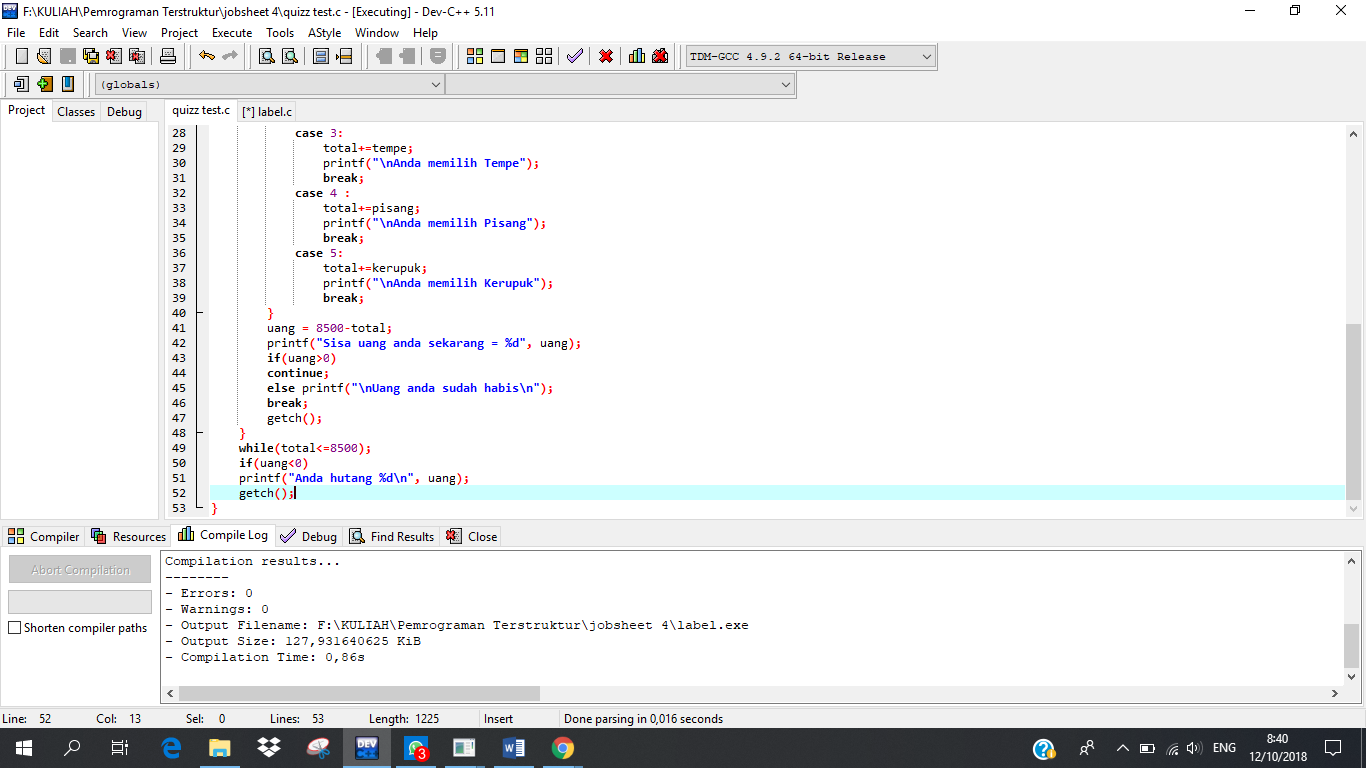
Perintah fungsi **exit()** yaitukeluar dari program dapat digunakan untuk menghentikan eksekusi program ketika dalam sebuah proses eksekusi dideteksi keadaan yang tidak dikehendaki. Kondisi tersebut bisa berupa data atau spesifikasi yang merupakan syarat eksekusi program berjalan normal. Fungsi exit()biasanya diberi nilai 0 saat diinisialisasi jika program dihentikan secara normal tanpa menunjukkan kesalahan atau deteksi.

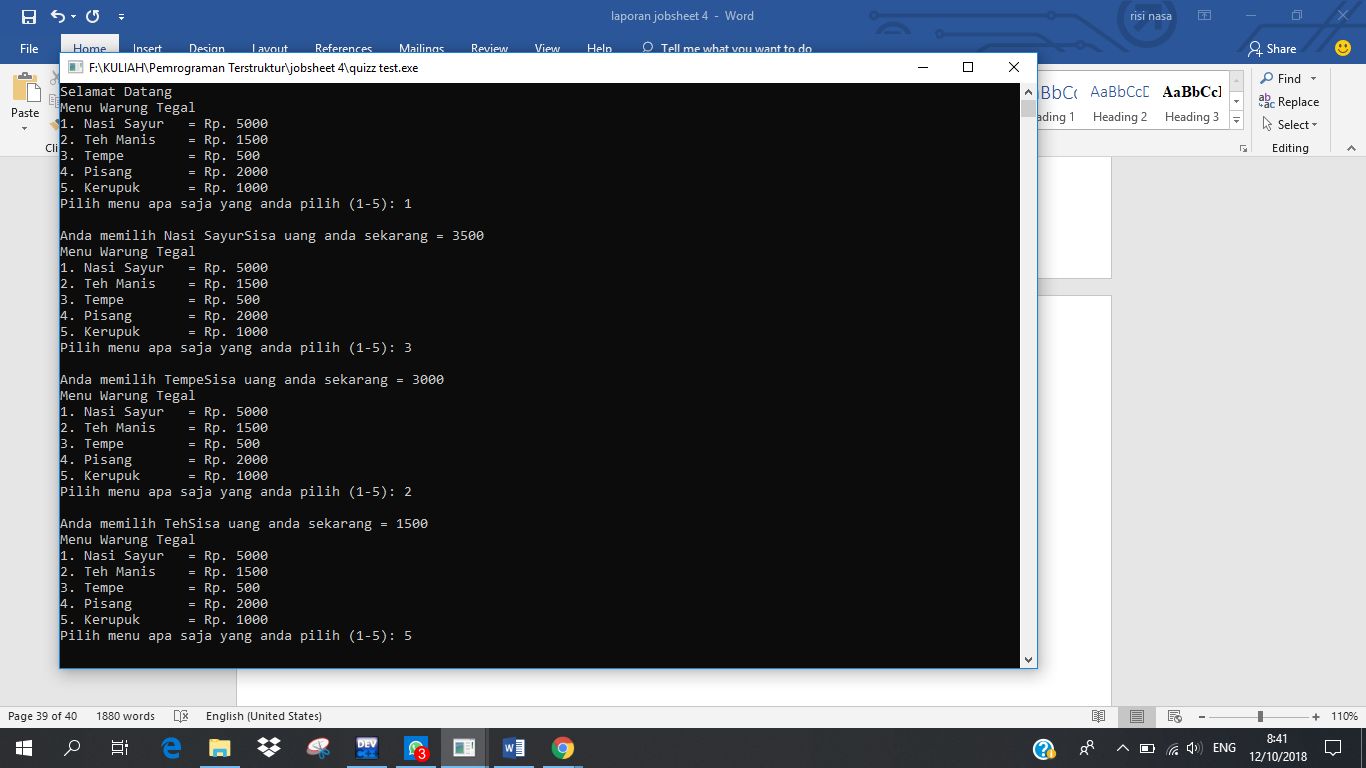
1. **QUIZ**

Buatlah suatu program tentang menu warteg dengan budget Rp. 8500 dengan menggunakan salah satu perintah perulangan yaitu do-while, while, atau for.







Hasil Run Program

1. **KESIMPULAN**

Setelah saya melakukan praktikum jobsheet 4 tentang struktur perulangan saya dapat menyimpulkan bahwa :

* 1. Macam proses perulangan dalam bahasa C antara lain : do-while, for.
  2. Perulangan dapat dilakukan dengan memasukkan batas perulangan, dan akan berhenti jika sudah mencapai batas perulangannya.
  3. Di struktur perulangan terdapat fungsi break, continue, exit()