**LAPORAN JOBSHEET 1**

**Dasar - Dasar Bahasa C**



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 1

DASAR-DASAR BAHASA C

1. **TUJUAN INSTRUKSIONAL KHUSUS**
2. Mengetahui dan memahami tentang Pemroraman Terstruktur
3. Mengetahui dan memahami tentang perintah dasar pemrograman bahasa c.
4. Membuat program yang melibatkan permasalahan-permasalahan yang sederhana
5. **DASAR TEORI**

Program bahasa C adalah suatu program yang terdiri dari satu atau lebih fungsi-fungsi.

1. **Pengertian Pemrograman Terstruktur**

Pemrograman Terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Selain pengertian diatas Pemrograman Terstruktur adalah suatu aktifitas pemrograman dengan memperhatikan urutan langkah-langkah perintah secara sistematis, logis , dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami.

Prinsip dari pemrograman terstruktur adalah Jika suatu proses telah sampai pada suatu titik / langkah tertentu , maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya / kembali lagi ke baris sebelumnya, kecuali pada langkah – langkah untuk proses berulang (Loop).

Bahasa pemrograman yang mendukung pemrograman terstruktur C, Pascal, Delphi.

1. **Program C Dasar**

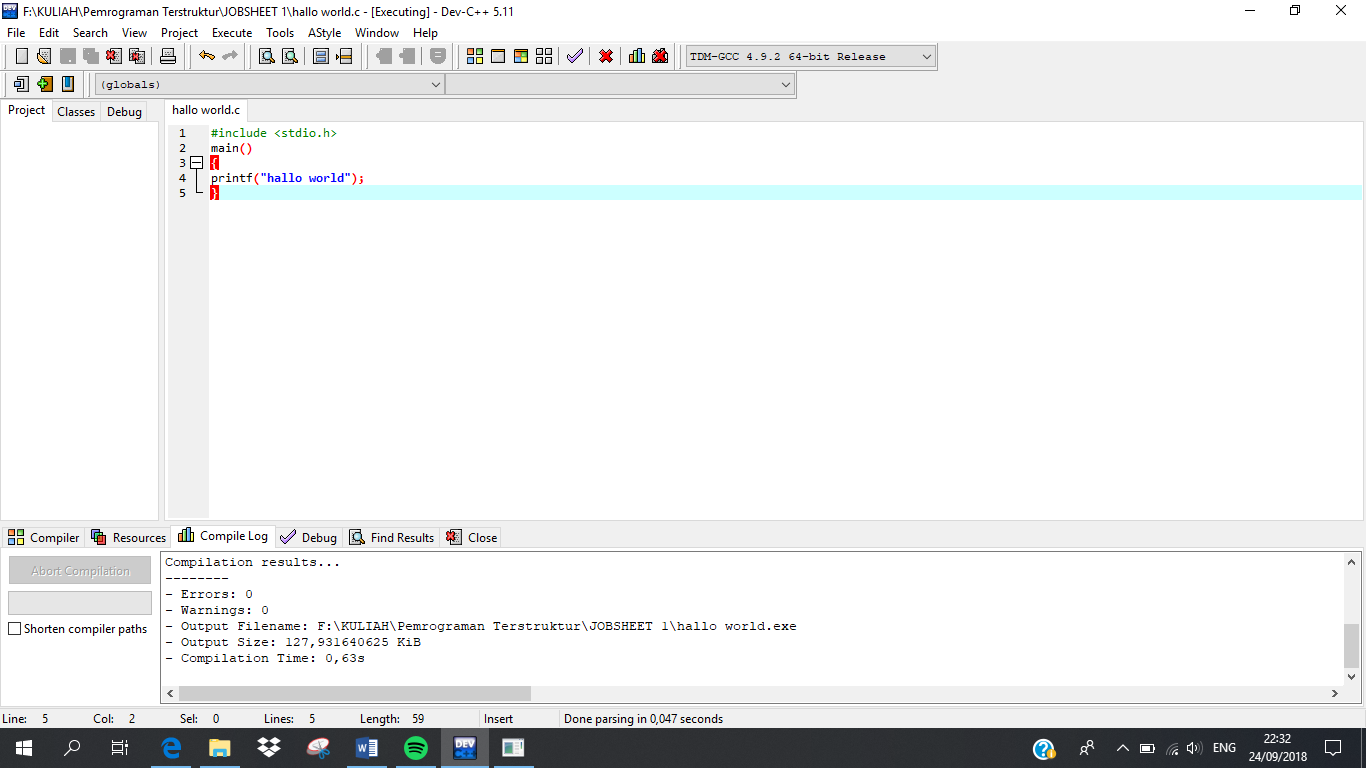
Berikut ini adalah contoh potongan program sederhana yang ditulis dalam bahasa C untuk menampilkan kalimat hallo world pada layar monitor

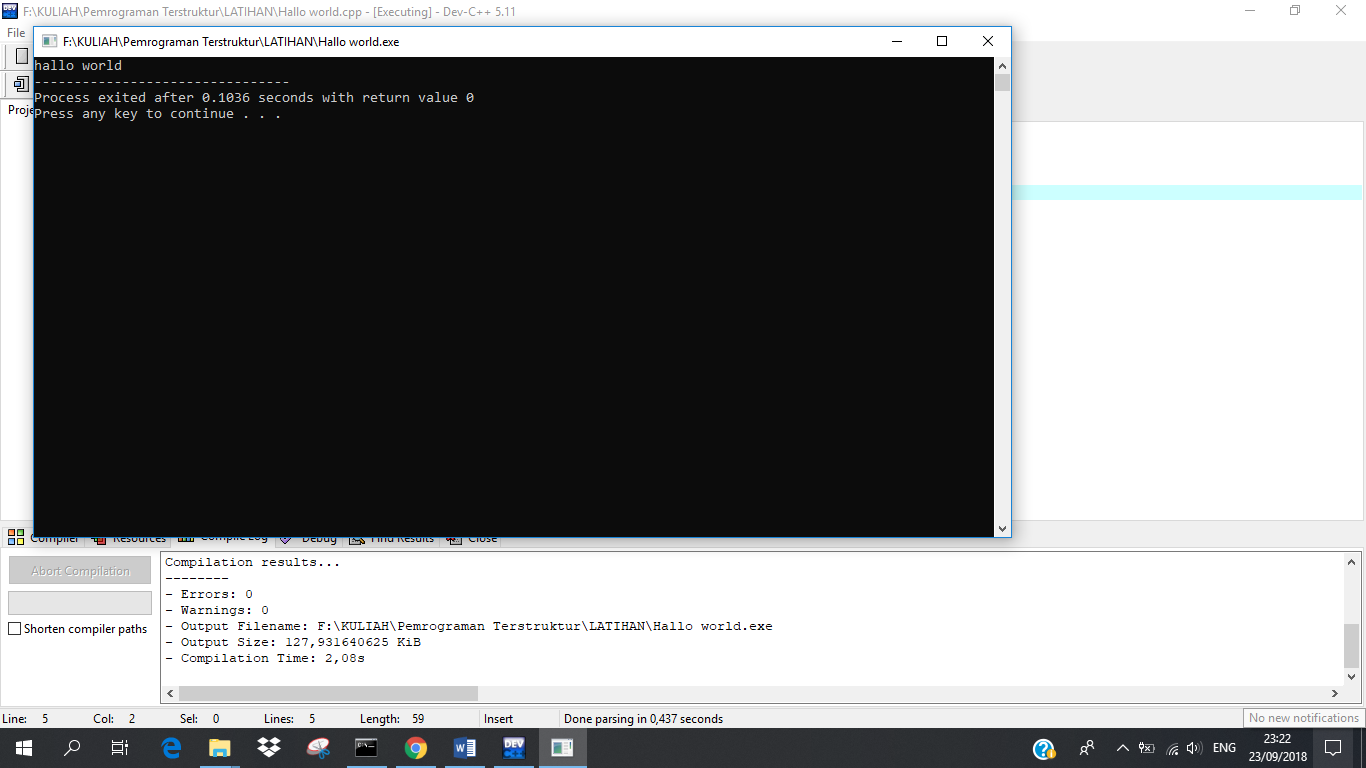
#include <stdio.h>

main()

{

printf(“hallo world”);

}



Dalam hal ini main() adalah sebuah fungsi yang harus selalu ada pada setiap program C. sedangkan program utama diletakan didalam tanda brace {}. Pada dasarnya program utama ini berupa pemanggilan fungsi-fungsi standar yang tersedia pada library (misal dalam hal ini printf adalah fungsi library yang digunakan untuk menampilkan hasil pada layar) atau fungsi-fungsi yang dibuat oleh programmer.

**Komponen-Komponen Program bahasa C**

Setiap program C terdiri dari beberapa komponen yang dikombinasikan dengan susunan tertentu. Untuk mendapatkan pemahaman yang menyeluruh berkaitan dengan struktur program C dan komponen-komponen programnya, dibawah ini akan diberikan program yang relative lengkap walaupun ukurannya kecil

/\* Program untuk menghitung perkalian dua bilangan.

\*/ #include <stdio.h>

int a,b,c;

int kali(int x, int y);

main()

{

/\* masukan bilangan pertama \*/

printf("Masukan bilangan antara 1 sampai 100: ");

scanf("%d", &a);

/\* masukan bilangan kedua \*/

printf("Masukan bilangan lain antara 1 sampai 100: ");

scanf("%d", &b);

/\* hitung dan tampilkan hasil \*/ c = kali(a, b);

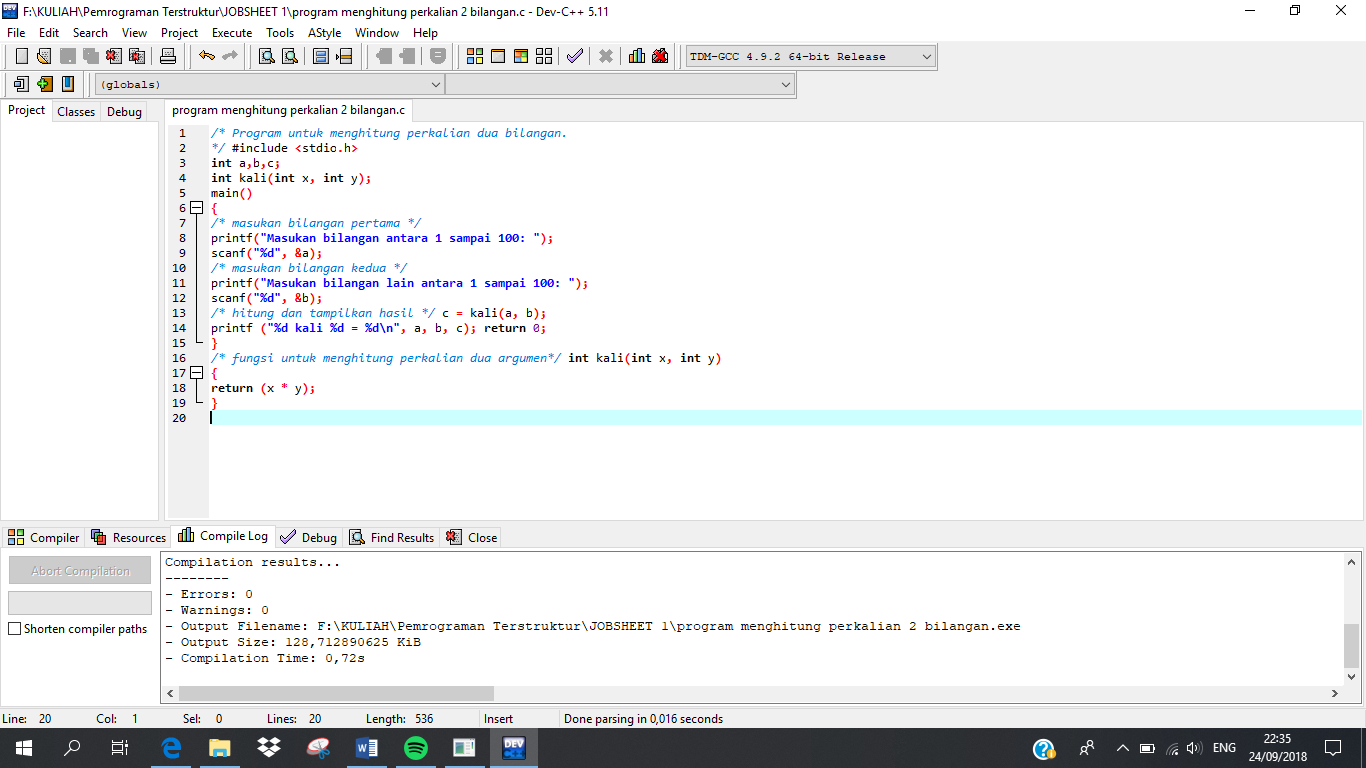
printf ("%d kali %d = %d\n", a, b, c); return 0;

}

/\* fungsi untuk menghitung perkalian dua argumen\*/ int kali(int x, int y)

{

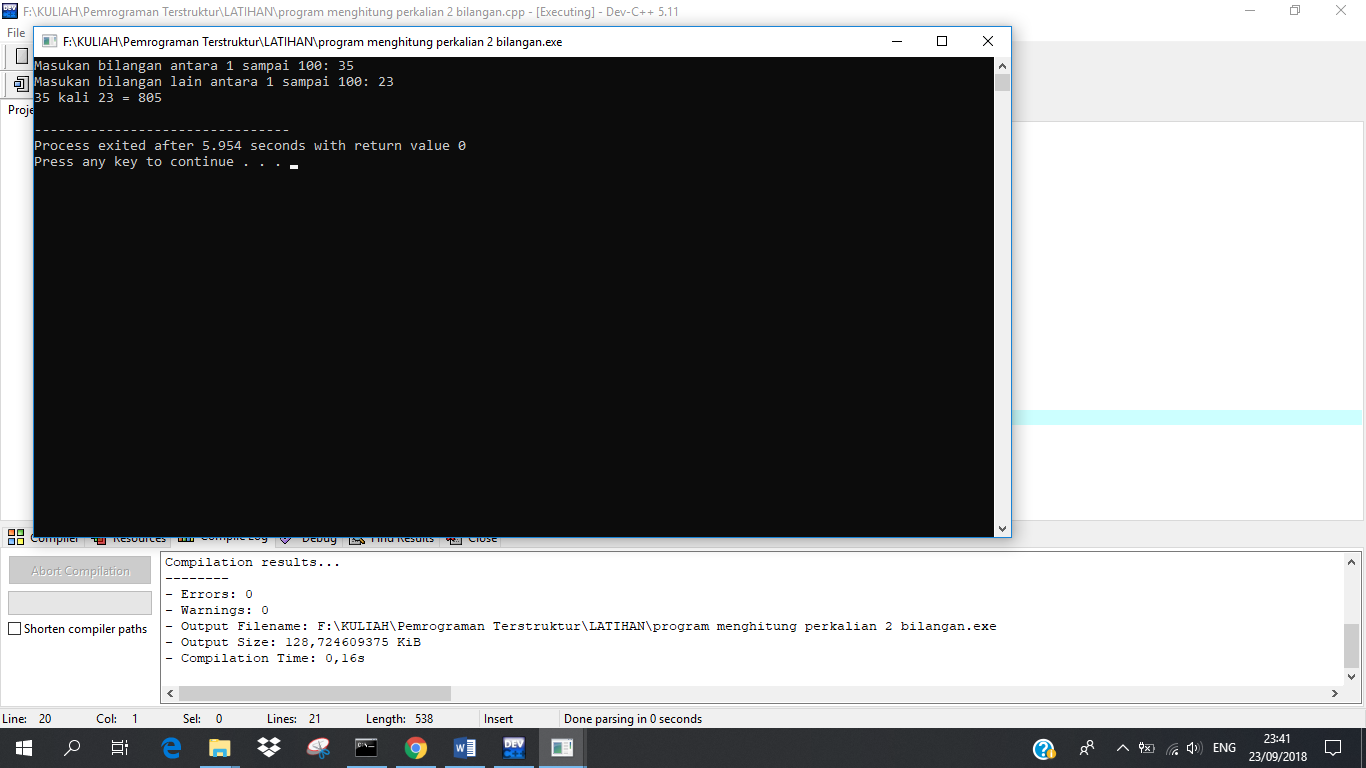
return (x \* y);

}

Masukan bilangan antara 1 sampai 100: 35

Masukan bilangan lain antara 1 sampai 100: 23

35 kali 23 = 805



(keterangan: pada program sebenaranya nomer bilangan tidak ditulis, penulisan disini semata-matas digunakan hanya untuk memudahkan penjelasan)

* **Fungsi main() (baris 8 sampai 28)**

Seperti yang telah disinggung diatas secara sekilas, komponen yang harus pada setiap program C adalah fungsi main(). Eksekusi program dimulai pada statement pertama pada main() dan berakhir di statemen terakhir dalam main().

* **Pengarah (directive) #include (baris 2)**

Pengarah #include mengintruksikan pada compiler C untuk menambahkan isi file include (file header) pada program yang dibuat saat compilasi. Pengarah #include pada contoh program diatas berarti “tambahkan isi file stdio.h”. Umumnya program C selalu membutuhkan satu atau lebih file header.

* **Definisi Variabel (baris 4)**

Varibel adalah sebuah nama yang digunakan sebagai petunjuk lokasi tempat simpan data. Dalam C, sebuah varibel harus didefinisikan terlebih dahulu sebelum digunakan. Sebuah definisi varibabel menginformasikan pada compiler nama varibel dan tipe data yang bisa disimpan. Dalam program contoh, definisi varibel baris 4 int a,b,c mendefinisikan tiga variable dengan nama a, b dan c yang masing-masing akan menyimpan tipe data integer.

* **Prototipe Fungsi (baris 6)**

Prototipe Fungsi berguna untuk menginformasikan pada compiler C nama dan argument-argumen fungsi yang ada pada program. Prototype ini harus ditulis sebelum fungsi tersebut digunakan, harus ditekankan disini bahwa prototype fungsi ini berbeda dengan definisi fungsi (function definition).

* **Statemen-Statemen Program (baris 11,12,15,16,19,20,22, dan 28)**

Perintah – perintah operasi yang dijalankan suatu program C pada dasarnya merupakan statemen-statemen program: Statemen-stamen ini dapat berupa operasi matematika, pemanggilan fungsi, membaca input keyboard, menampilkan informasi pada layar, dan lain-lain. Perlu diingat disini bahwa akhir statemen pada C harus selalu diakhiri oleh tanda baca semicolon.

* **printf()**

Statemen printf() (baris 11, 15, dan 20) adalah fungsi library untuk menampilkan informasi pada layar monitor. Statemen ini dapat menampilkan text sederhana (seperti pada baris 11 dan 15), atau pesan dan nilai satu atau lebih variable program (seperti pada baris 20)

* **scanf()**

Statemen scanf() (baris 12 dan 16) adalah fungsi library juga yang berguna untuk membaca data dari keybord dan memberikan data tersebut pada satu atau lebih variable program.

Statemen program pada baris 19 adalah pemanggilan fungsi kali(). Dengan kata lain statemen ini mengeksekusi fungsi kali, dan mengirimkan argument a dan b pada fungsi, setelah fungsi selesai dipanggil, kali() mengembalikan return sebuah nila pada program yang selanjutnya disimpan pada variable c.

* **return**

Pada baris 22 dan 28 terdapat statemen return, statemen return pada baris 28 merupakan bagian dari fungsi kali(). Yaitu untuk menghitung perkalian x dan y dan mengembalikan hasil pada statemen program yang memanggil kali(). Statemen return pada baris 22 mengembalikan nilai 0 pada system operasi setelah program selesai dilaksanakan.

* **Definisi Fungsi**

Fungsi adalah bagian program yang bersifat independent yang ditulis atau dibuat untuk melakukan tugas-tugas tertentu. Setiap fungsi harus memiliki nama yang unik yang mencerminkan tugas yang akan dilaksanakan oleh fungsi tersebut.

Fungsi dengan nama kali(), pada baris 26 sampai 29, adalah fungsi yang didefinisikan programmer (user-defined function).

Bahasa C juga telah menyediakan fungsi-fungsi library yang sering digunakan oleh programmer, misal untuk operasi-operasi input/output, operasi matematika, dll. Yang perlu dilakukan oleh programmer untuk menggunakan fungsi-fungsi library ini adalah cukup meng include-kan file header yang berisi informasi fungsi-fungsi yang tersedia beserta argumennya, misal untuk memanggil fungsi trigonometri seperti sin, cos, dan lain-lain pada awal program harus kita tulis : #include <math.h>

* **Komentar Program**

Komentar program bertujuan untuk dokumentasi listing sumber. Komentar pada C harus dimulai dengan /\* dan diakhiri dengan \*/. Setiap bagian program yang berada diantara tanda tersebut tidak akan dicompile oleh compiler (tidak mempengaruhi ukuran file exe).

* **Braces (Baris 9, 23, 27, dan 29)**

Setiap fungsi yang dibuat pada C harus berada didalam tanda kurung kurawal (brace), hal ini juga termasuk fungsi main(). Satu atau lebih statemen yang berada dalam tanda kurung kurawal dinamakan block.

1. **TIPE DATA**

Tipe data digunakan untuk mendeklarasikan suatu pengenal/variabel atau konstanta. Bahasa C menyediakan lima macam tipe data dasar, yaitu :



Selanjutnya int, float, double dan char dapat dikombinasikan dengan pengubah (modifier) signed, unsigned, long dan short. Hasil kombinasi tipe data tersebut secara keseluruhan adalah sebagai berikut :



Pengubah signed berarti nilai variabel dapat mempunyai tanda, sehingga dapat bernilai positif atau negatif.

1. **VARIABEL**

Variabel adalah suatu pengenal yang digunakan untuk menyimpan suatu nilai dan nilai dari variabel tersebut dapat berubah-ubah selama proses program.

1. **Mendeklarasikan Variabel**

Variabel belum dapat digunakan didalam program sebelum dideklarasikan terlebih dahulu. Dengan kata lain, deklarasi variabel harus dilakukan terlebih dahulu sebelum variabel tersebut digunakan.

Deklarasi variabel berarti memberitahukan kepada C tentang tipe data dari suatu variabel, sehingga C dapat mempersiapkan terlebih dahulu tempat dari variabel tersebut di memori.

* Bentuk umum deklarasi

***typedata*** namavariabel;

* Contoh :

**int** jumlah;

**float** rata\_rata;

**char** nama;

1. **Menentukan tipe data variable**

Untuk menentukan tipe dari pada variabel tergantung dari jenis data yang akan di tampung pada variabel.

1. **Penugasan variabel**

Memberikan nilai awal kepada suatu variabel

* Bentuk umum

Namavariabel = *nilai;*

* *Contoh :*

jumlah = 5;

rata\_rata = 2.5;

1. **Pendeklarasian dan penugasan variabel**

Nilai awal suatu variabel dapat langsung diberikan pada saat mendeklarasikannya.

* Bentuk umum

***typedata*** namavariabel = nilai;

* *Contoh* :

int jumlah = 5;

float rata\_rata = 2.5;

1. **Tipe data string**

Bahasa C tidak menyediakan deklarasi variabel untuk tipe string. Nilai string adalah kumpulan dari nilai-nilai karakter yang berurutan dalam bentuk larik dimensi satu. Dengan demikian, untuk nilai string dapat dideklarasikan sebagai larik bertipe char.

* Bentuk umum deklarasi

***char*** nama[*elemen\_karakter*]

* Contoh deklarasi dan penugasan

char nama[17] = ‘Jhon Travolta’

1. **KONSTANTA**
2. **Konstanta Bernama**

Konstanta bernama adalah suatu pengenal yang berhubungan dengan nilai tetap (variabel konstan). C++ memungkinkan pendefinisian suatu konstanta bernama. Hal ini dilakukan dengan menggunakan kata kunci const.

* Bentuk umum :

**const** *tipe\_data* nama\_konstanta = *nilai*;

* *Contoh :*

const int bulan = 12;

const float phi = 3.14285;

* Keterangan :

1. variabel bulan adalah konstanta bernama bertipe integer dengan nilai 12.

2. variabel phi adalah konstanta bernama bertipe float dengan nilai 3.141592.

Konstanta bernama berbeda dengan variabel, suatu konstanta bernama nilai yang ada padanya tidak dapat diubah setelah didefinisikan.

Keuntungan pemakaian konstanta bernama adalah untuk menghindari salah tulis terhadap nilai yang terkandung didalam variabel.

Contoh :

#include <stdio.h>

main( )

{

const float phi = 3.14285;

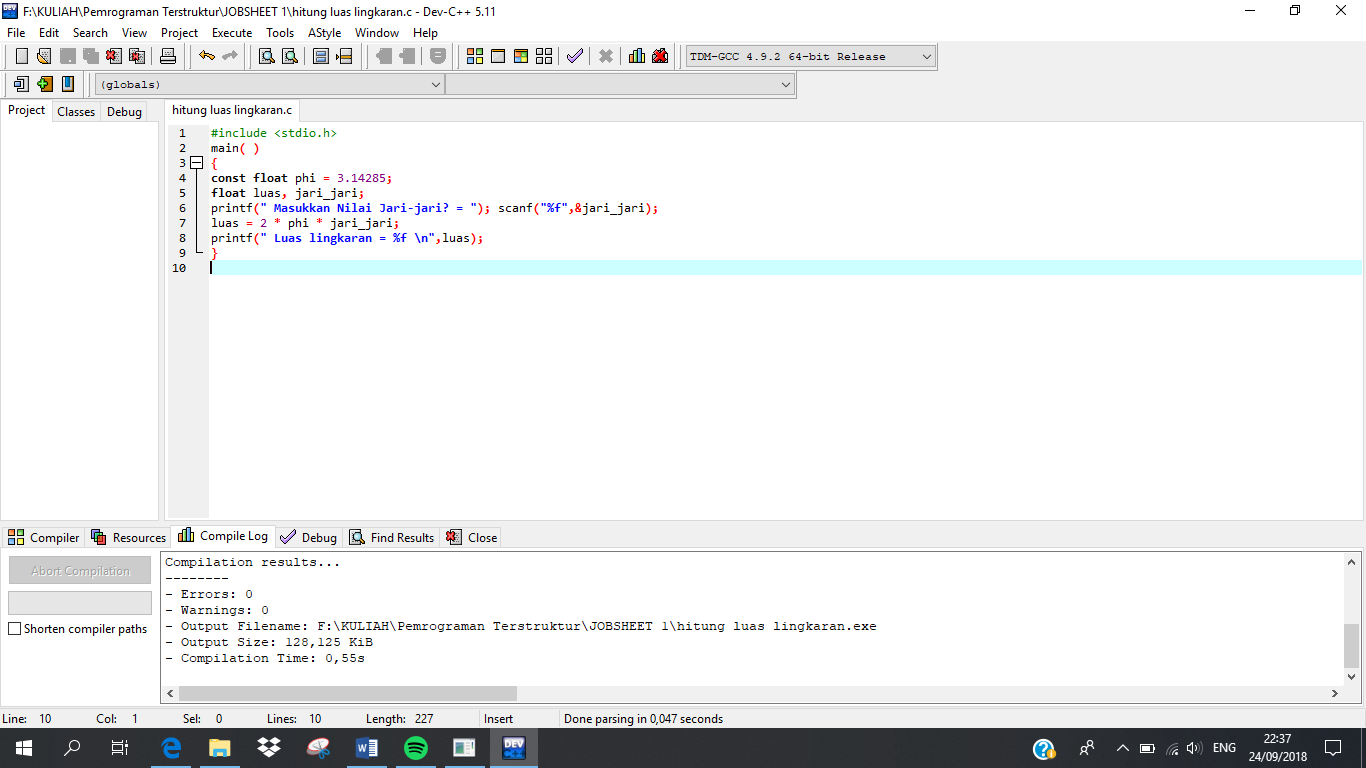
float luas, jari\_jari;

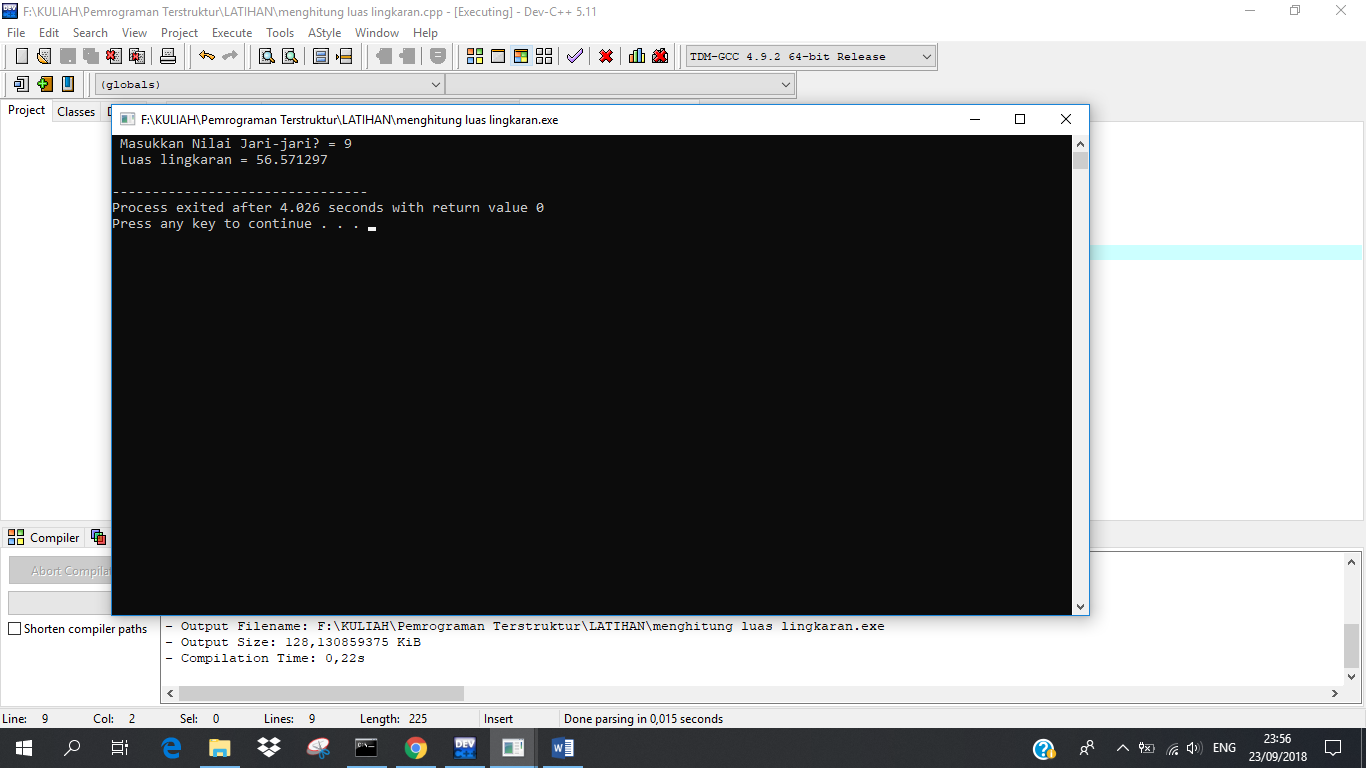
printf(“ Masukkan Nilai Jari-jari? = “); scanf(“%f”,&jari\_jari);

luas = 2 \* phi \* jari\_jari;

printf(“ Luas lingkaran = %f \n”,luas);

}



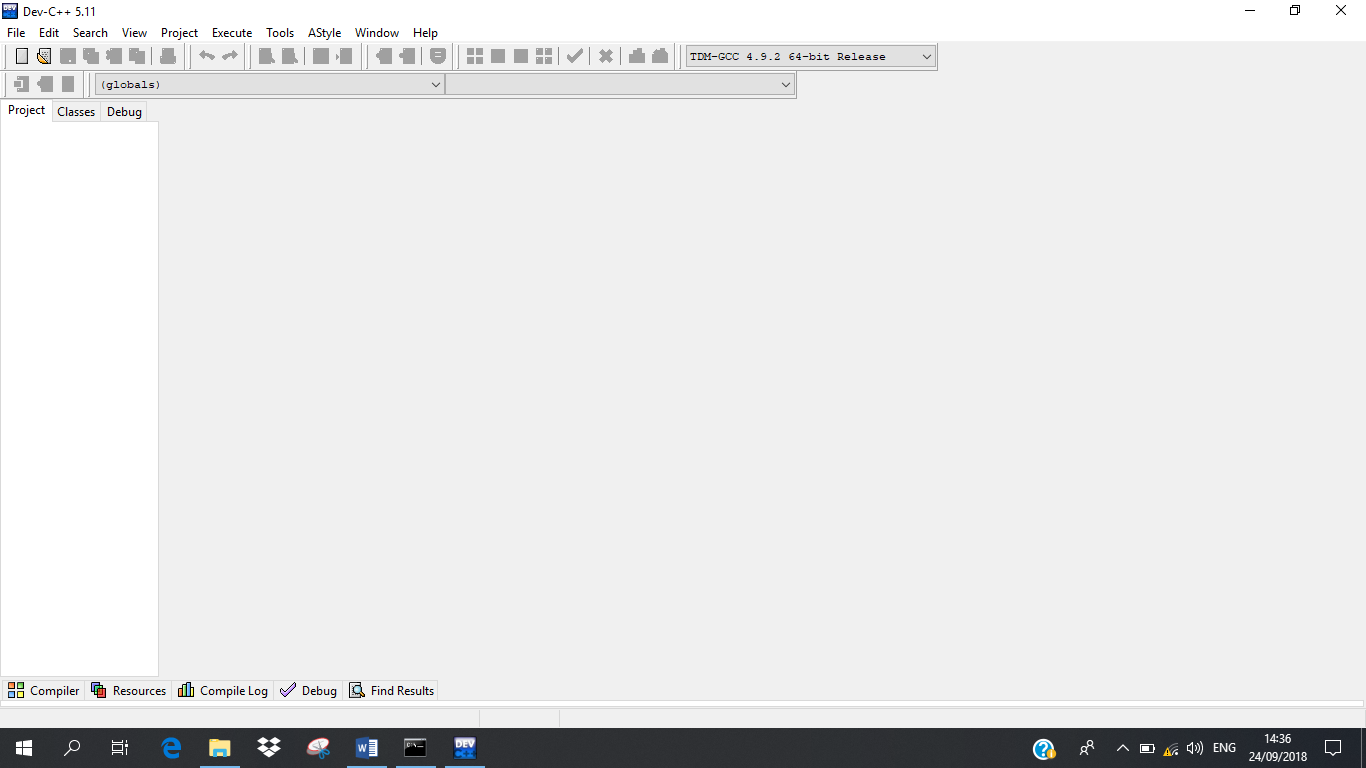


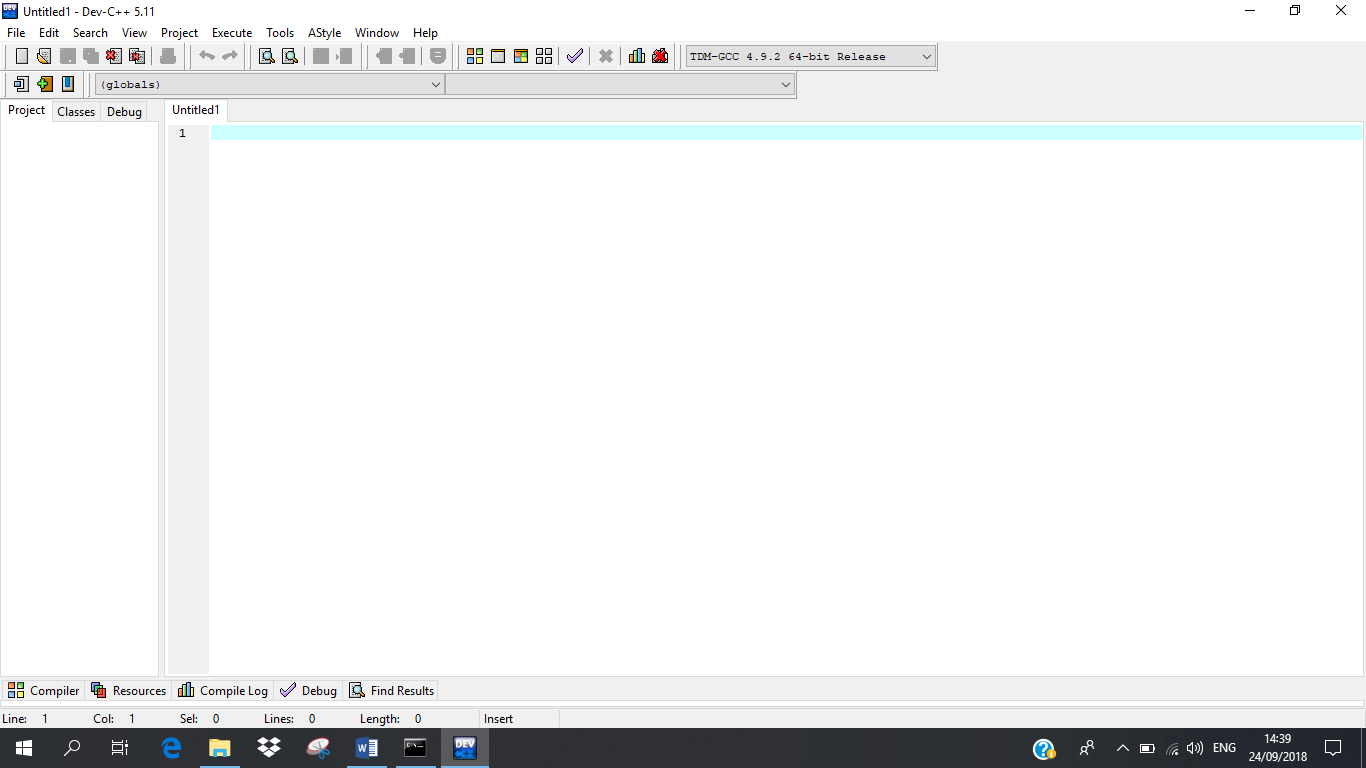
1. **Konstanta Karakter dan Konstanta String**

Konstanta karakter merupakan nilai sebuah karakter yang ditulis diantara tanda petik tunggal (‘ ‘), Contoh : ‘a’, ‘7’, dll. Konstanta string (literal string) merupakan nilai sebuah atau lebih karakter yang ditulis diantara tanda petik ganda (“ “), Contoh : “a”, “bahasa c”, dll.

1. **Konstanta Karakter Escape**

Konstanta karakter escape banyak digunakan di statemen-statemen untuk menampilkan hasil, misal menggeser kursor ke baris berikutnya (ganti baris baru) , membuat kursor kembali ke kolom pertama (carriage return) atau membunyikan bel dan lain sebagainya. Suatu konstanta karater escape diawali dengan karakter atau tanda ‘ \’.

1. **LANGKAH KERJA**
2. **Program pertama**
3. Jalankan program DevC++
4. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<conio.h>

main()

{

//clrscr();

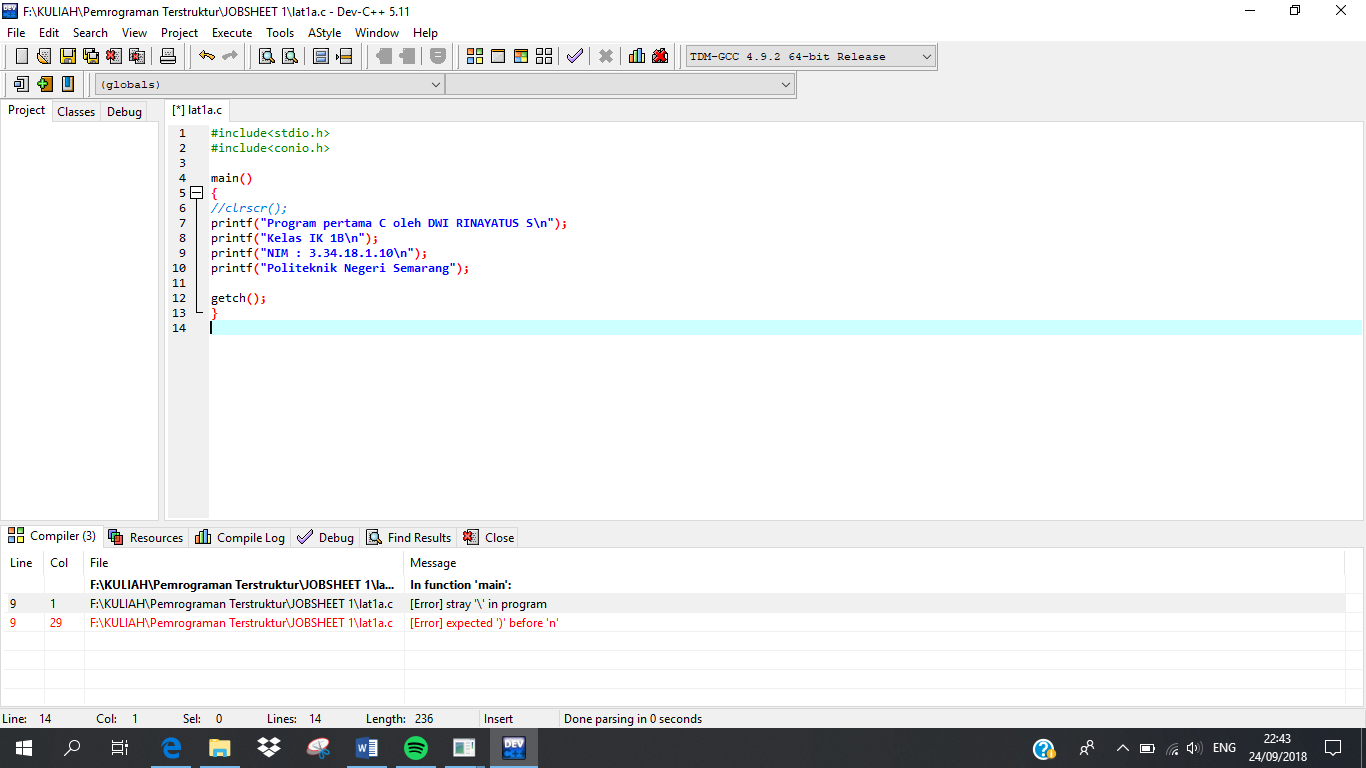
printf("Program pertama C oleh DWI RINAYATUS S\n");

printf("Kelas IK 1B\n");

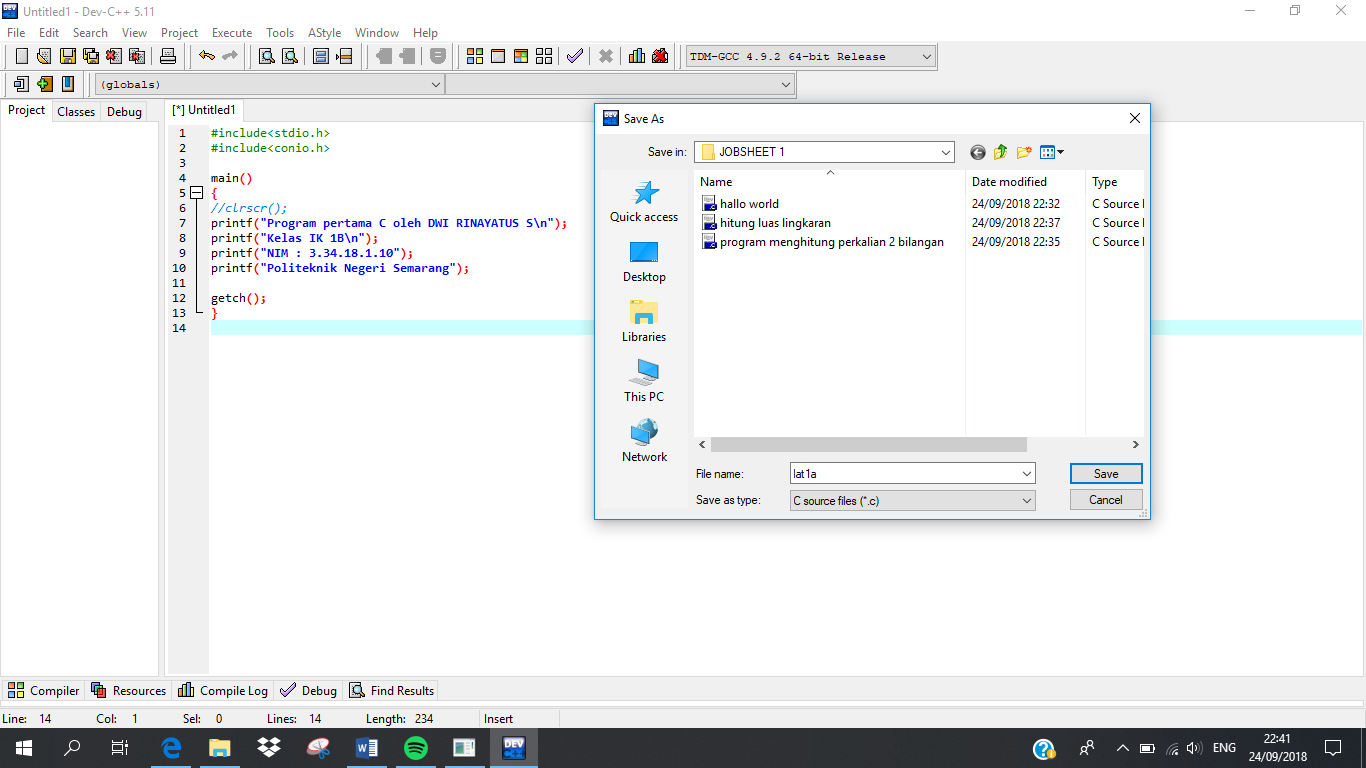
printf("NIM : 3.34.18.1.10\n");

printf("Politeknik Negeri Semarang");

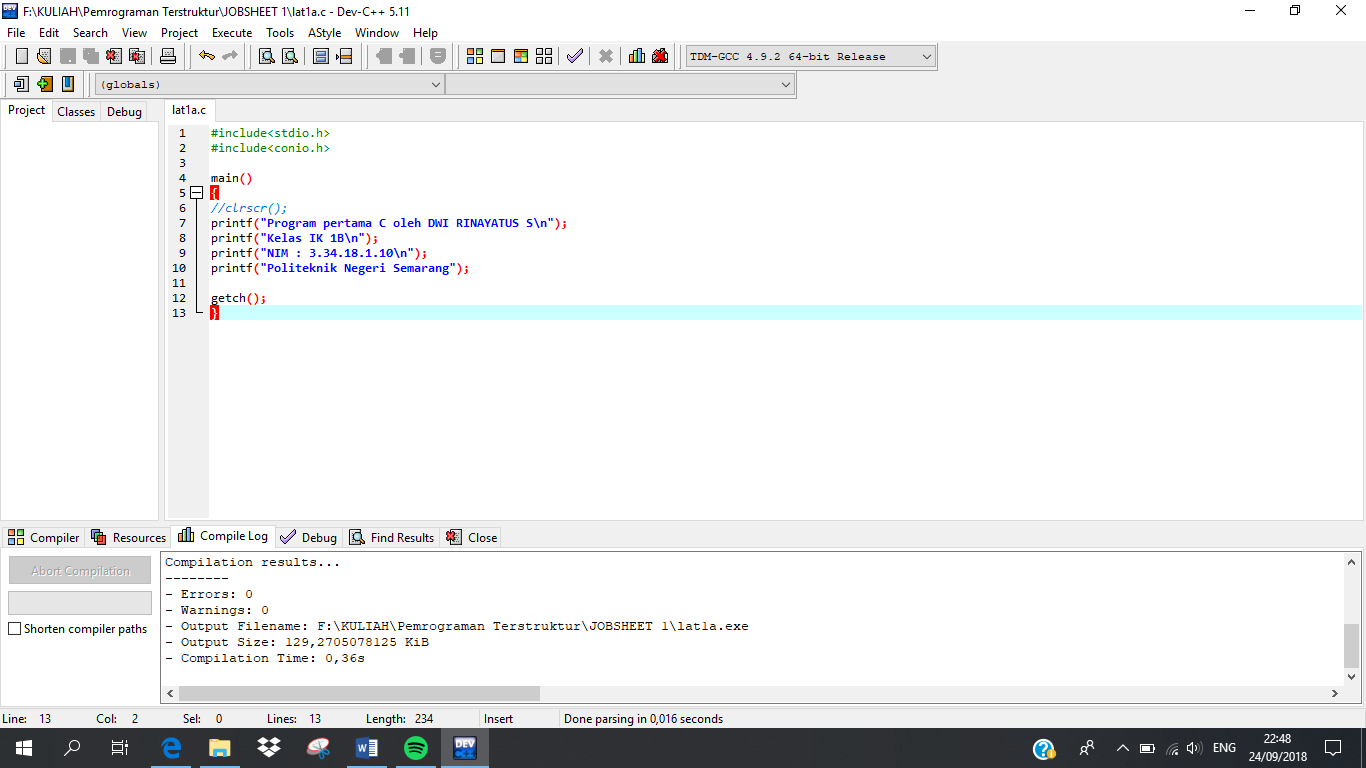
getch();



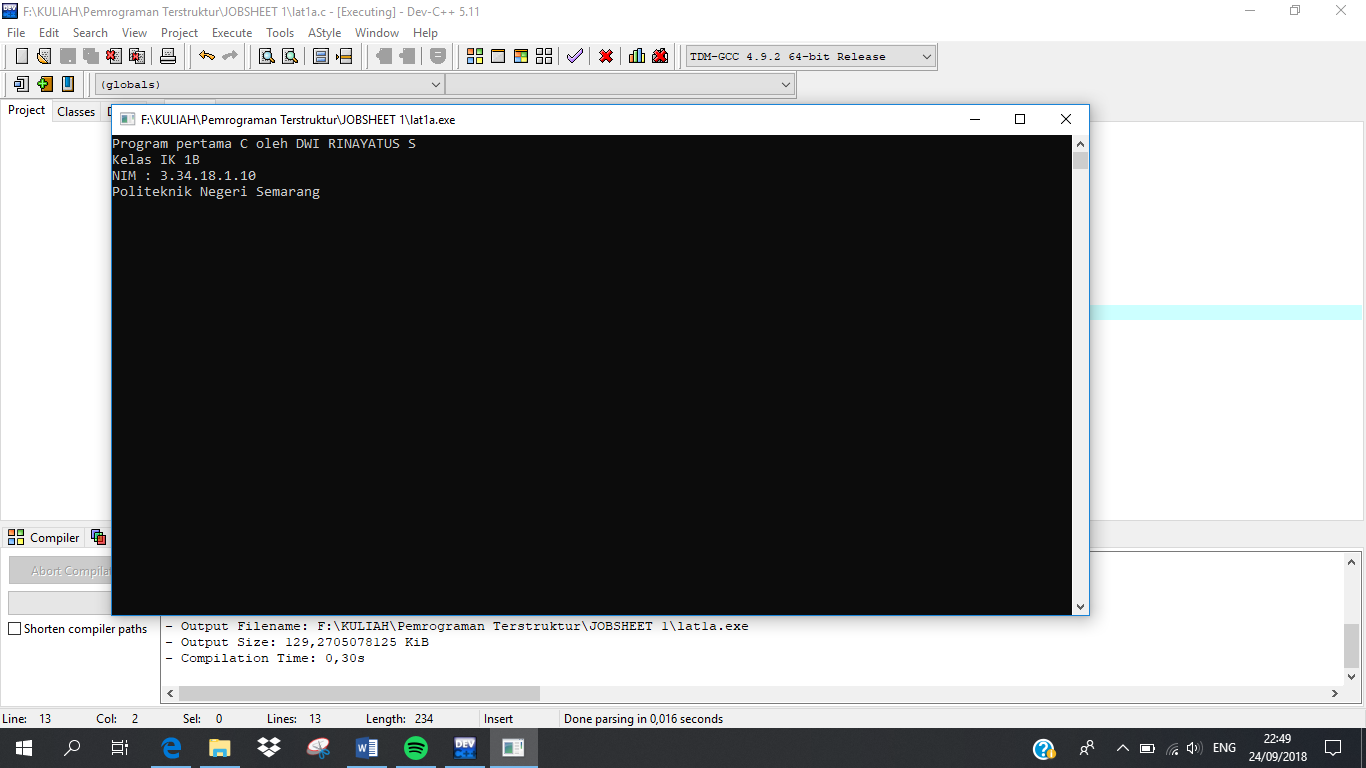
1. Alt+F, pilih save ketik nama lat1a.c



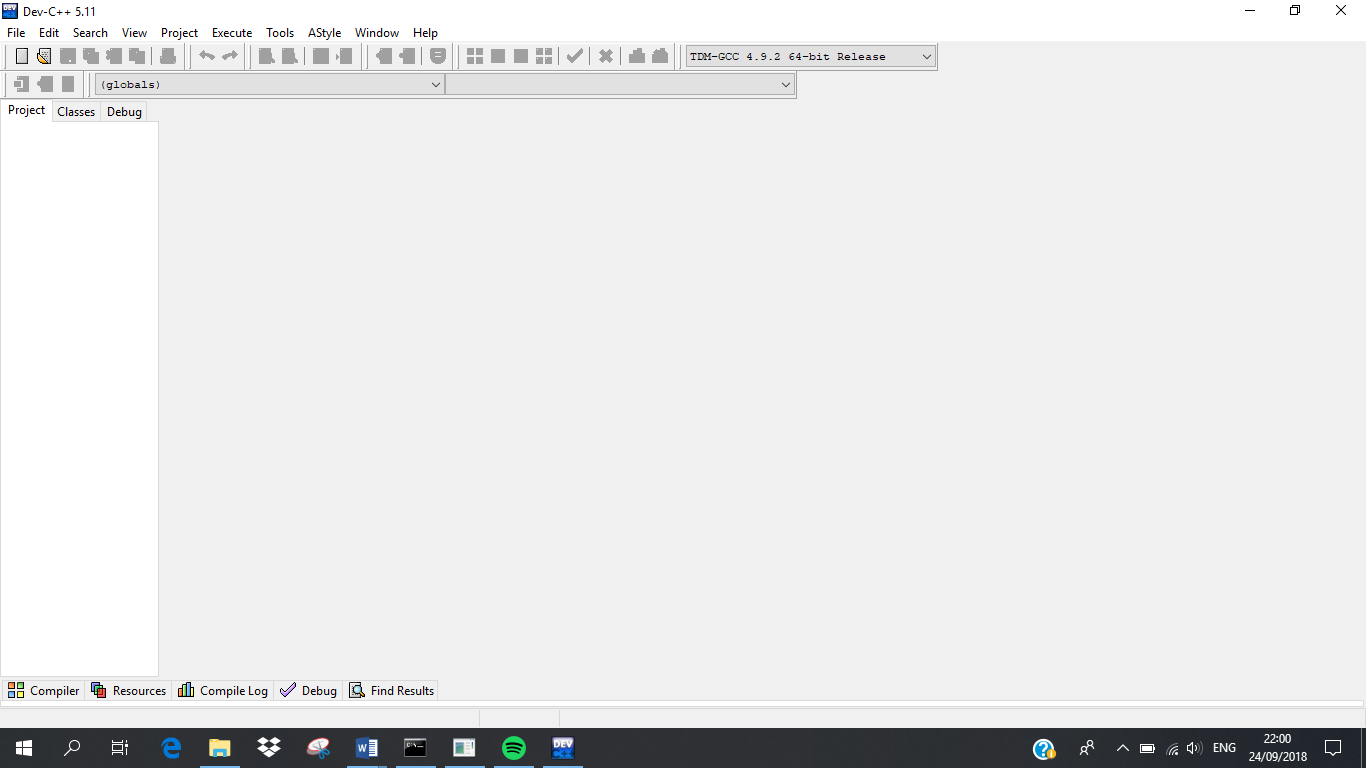
1. Lalu compile apabila masih terdapat error benarkan hingga tidak terdapat error.



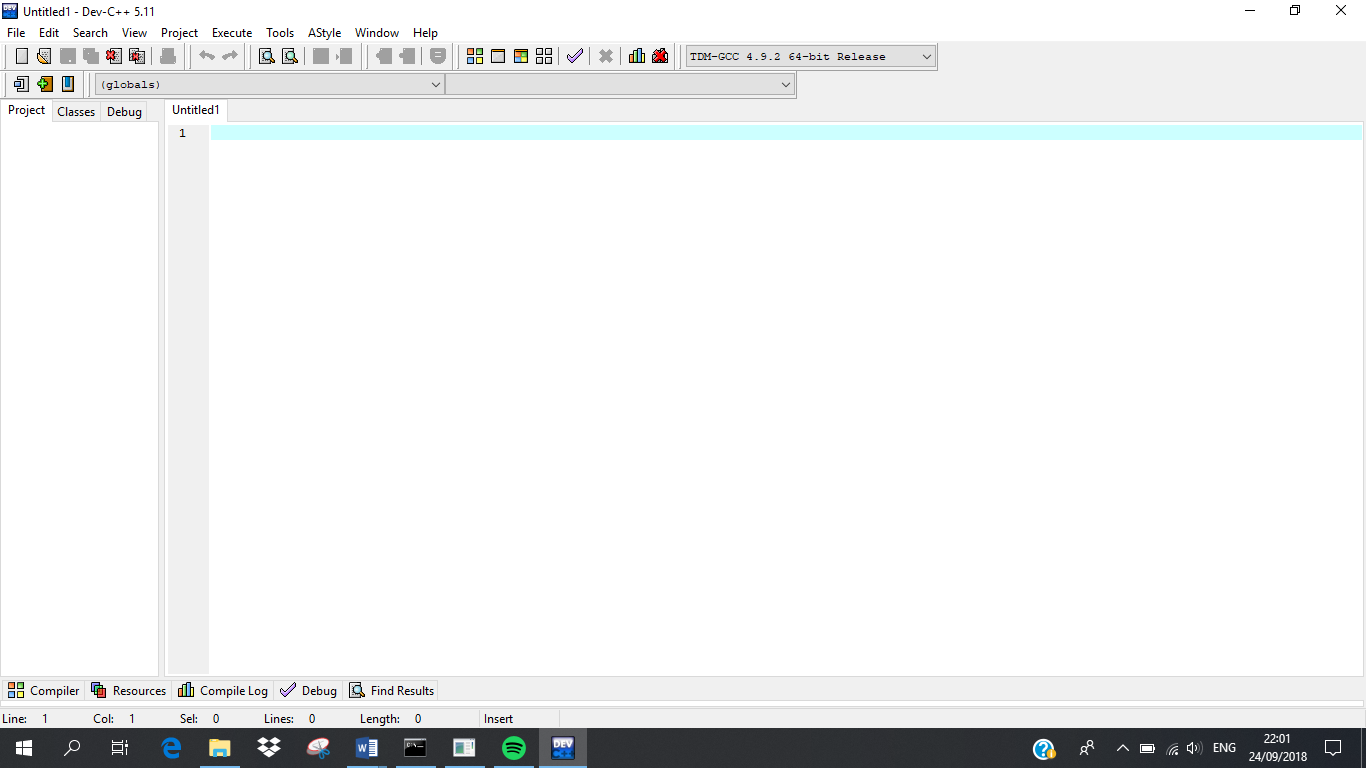
1. Lalu klik Run maka akan muncul programnya.



1. **Menampilkan tipe data string, integer, floating, dan karakter**
2. Jalankan progam Dev C++



1. Alt + F, pilih New



1. 4. Ketik :

#include<stdio.h>

main()

{

printf("\tNama Mahasiswa:%s\n", "DWI RINAYATUS S");

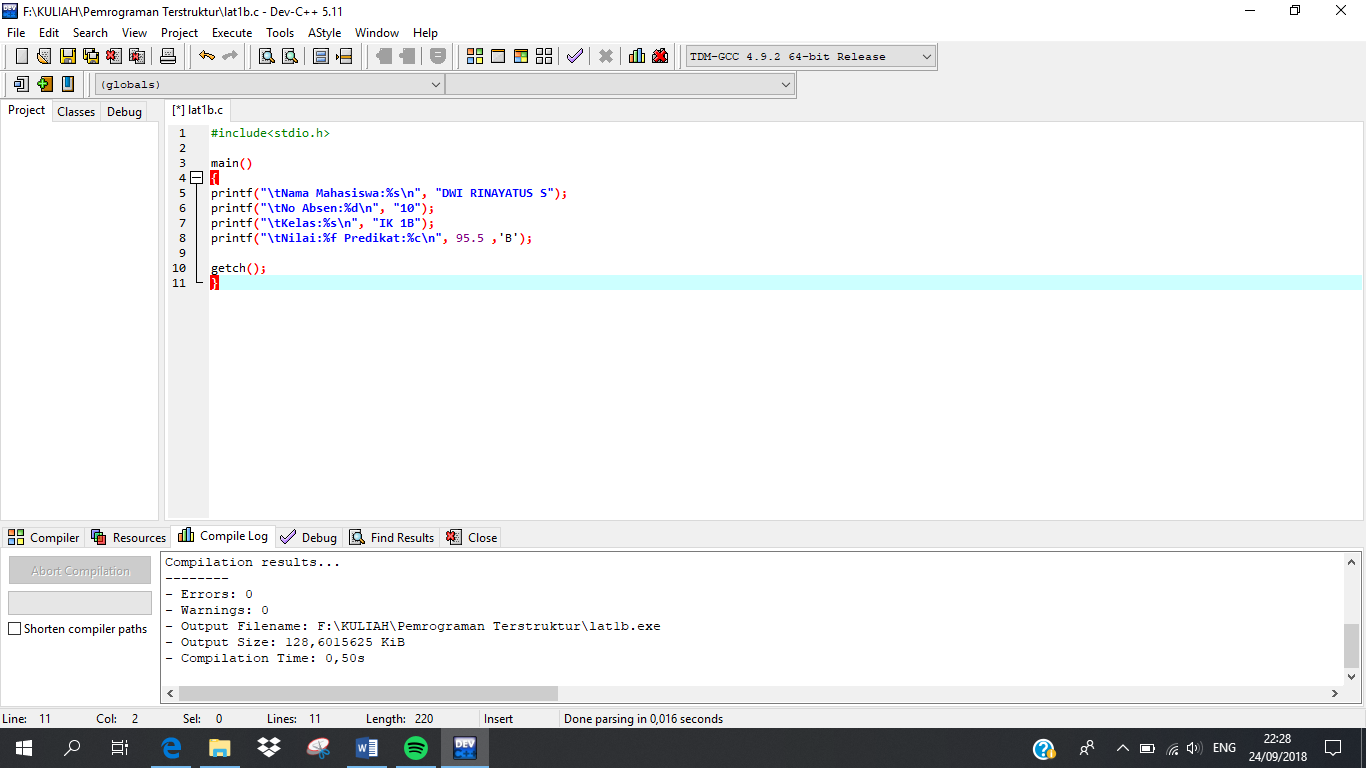
printf("\tNo Absen:%d\n", "10");

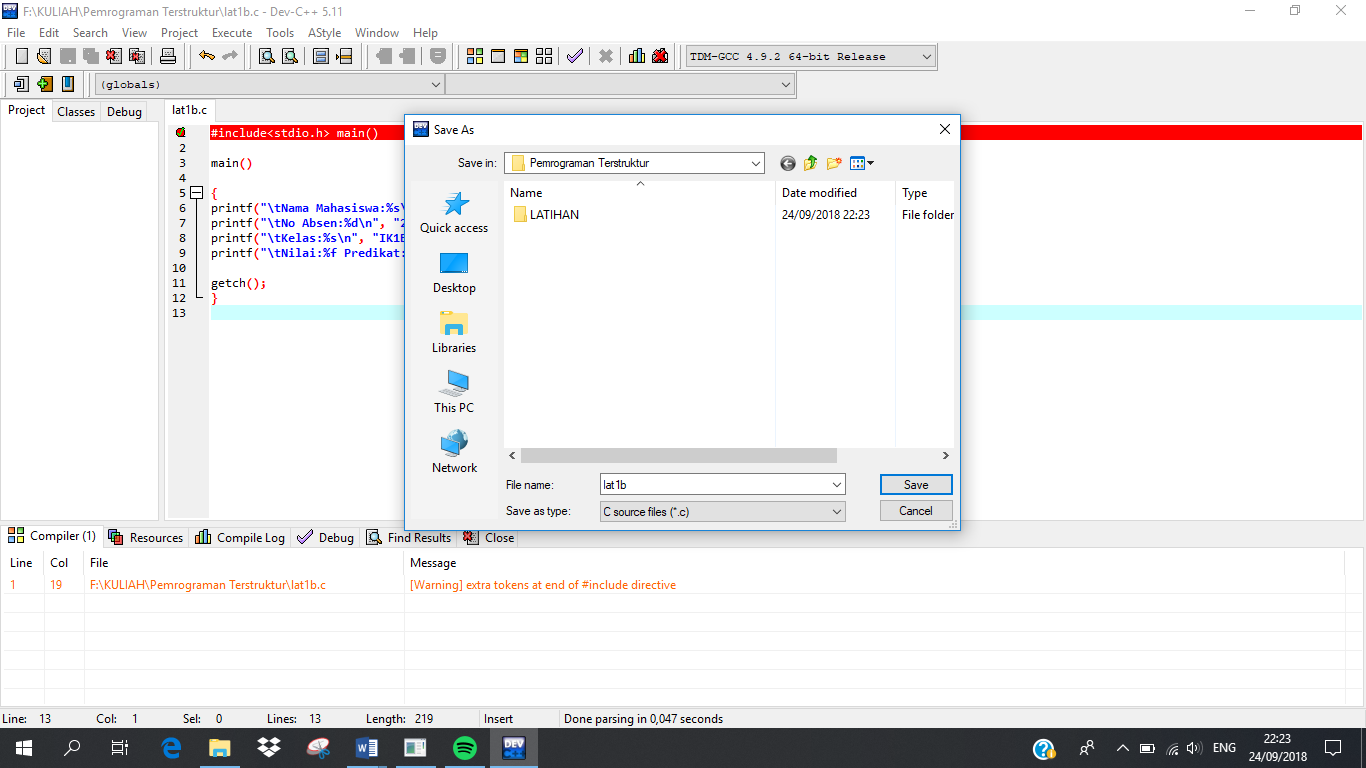
print("\tKelas:%s\n", "IK 1B");

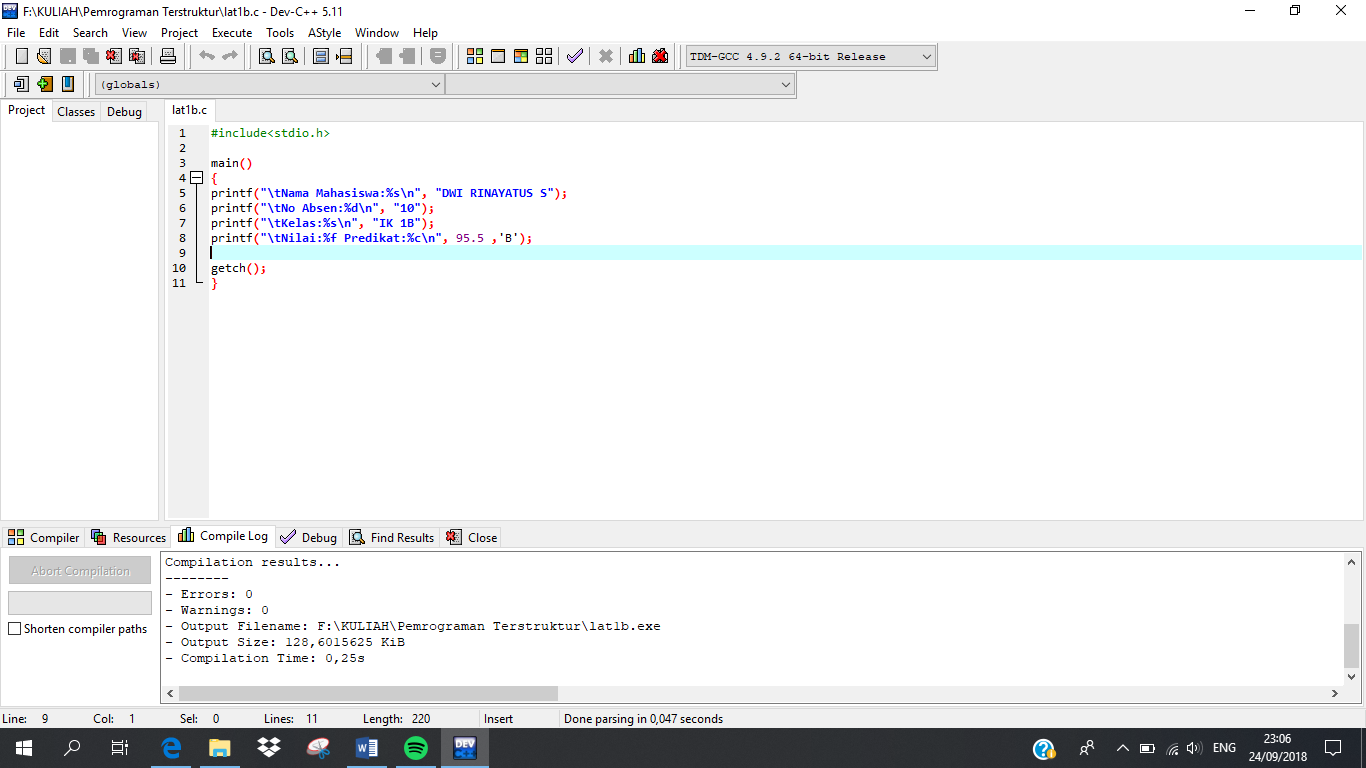
printf("\tNilai:%f Predikat:%c\n", 95,5 ,'B');

getch();

}

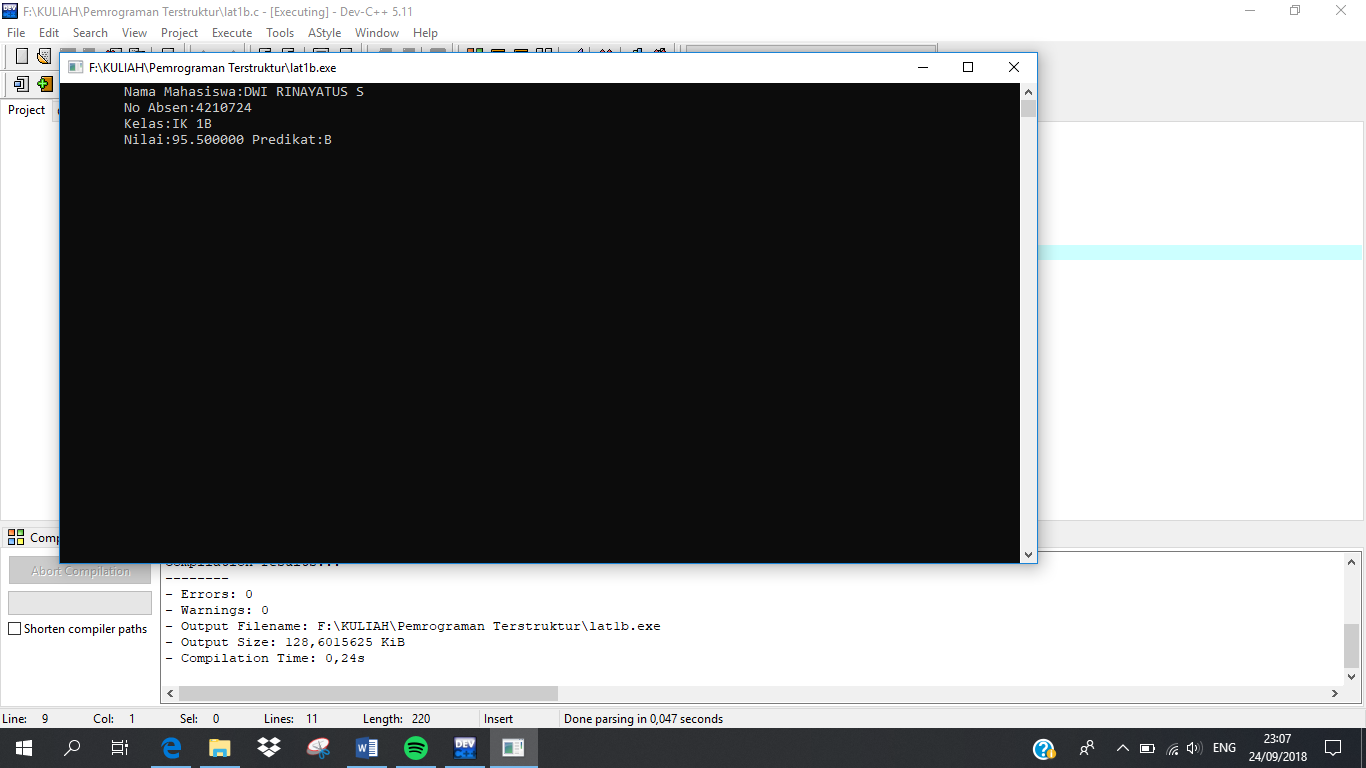


1. Alt+F, pilih save ketik nama lat1b.c
2. Lalu compile apabila masih terdapat error benarkan hingga tidak terdapat error.



Setelah di compile tidak terdapat error

1. Lalu klik Run maka akan muncul programnya.



1. **Komentar program**
2. Jalankan progam Dev C++
3. Alt + F, pilih New
4. Ketik :

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

\* Program Prl\_3.c \* \* Dibuat oleh : DWI RINAYATUS S \*

\* Tanggal : 22 September 2018 \*

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

#include<stdio.h>

main()

{

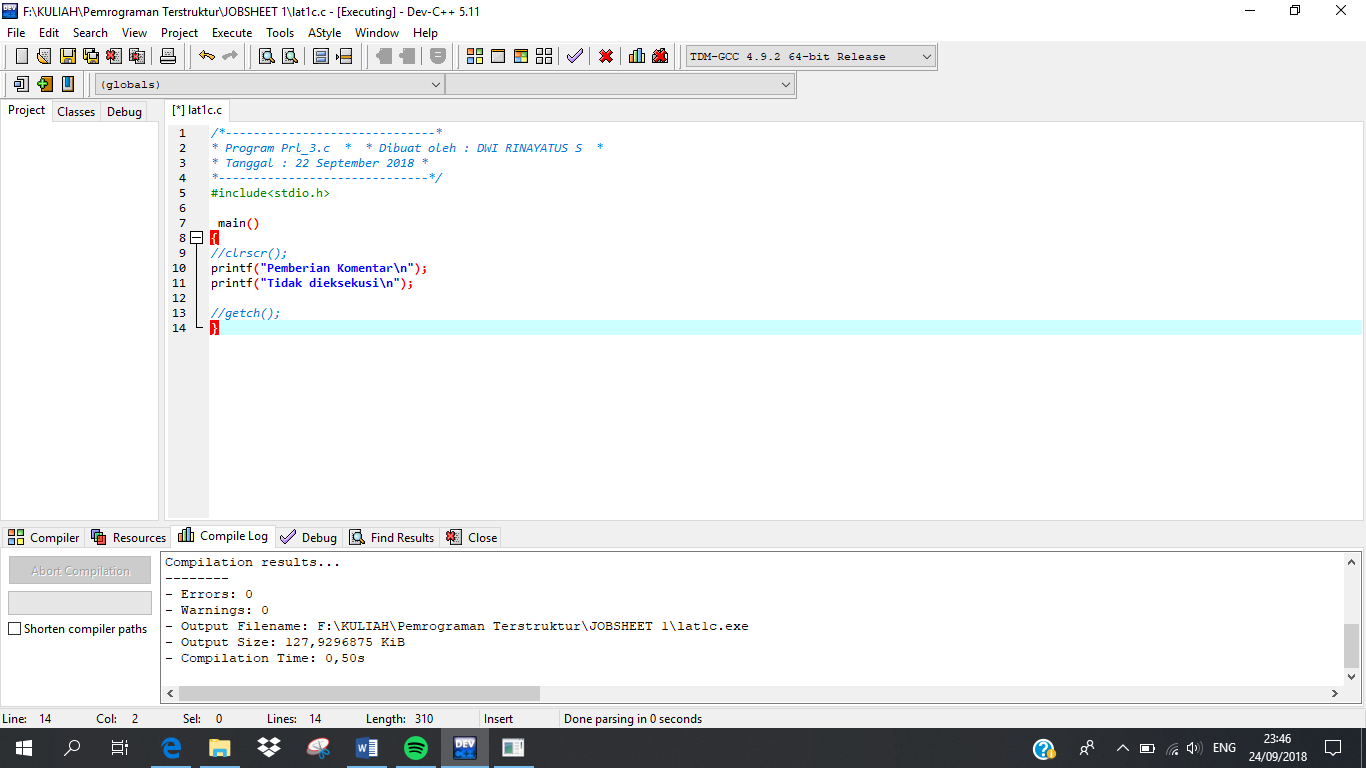
//clrscr();

printf("Pemberian Komentar\n");

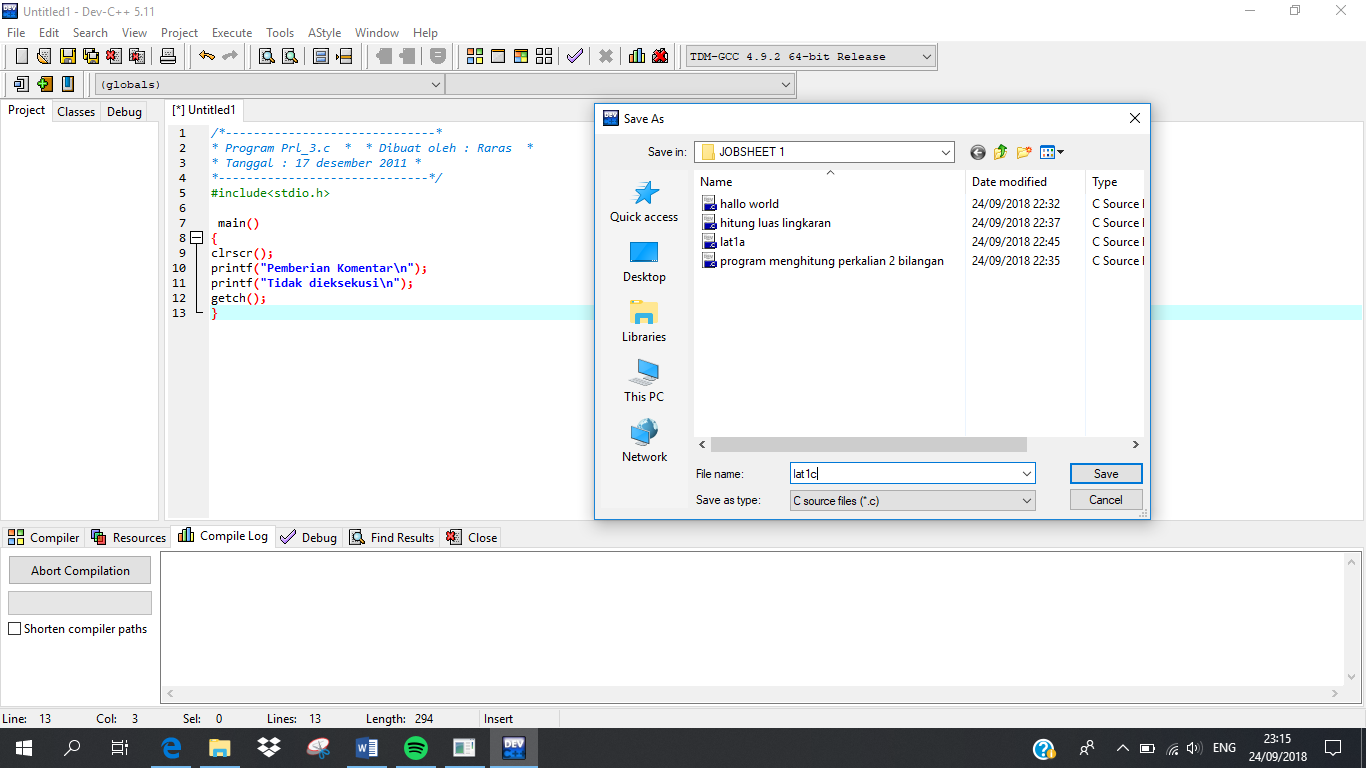
printf("Tidak dieksekusi\n");

//getch();

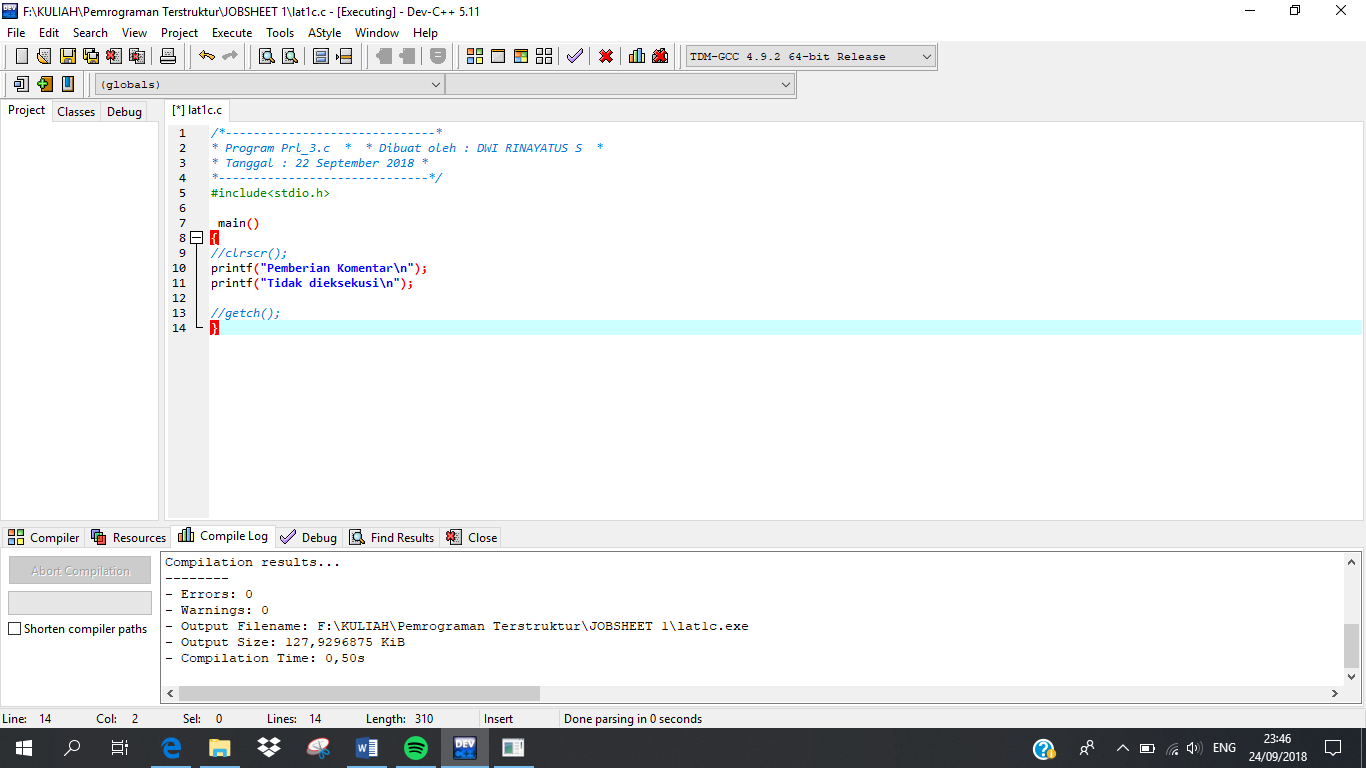
}



1. Alt+F, pilih save ketik nama lat1c.c

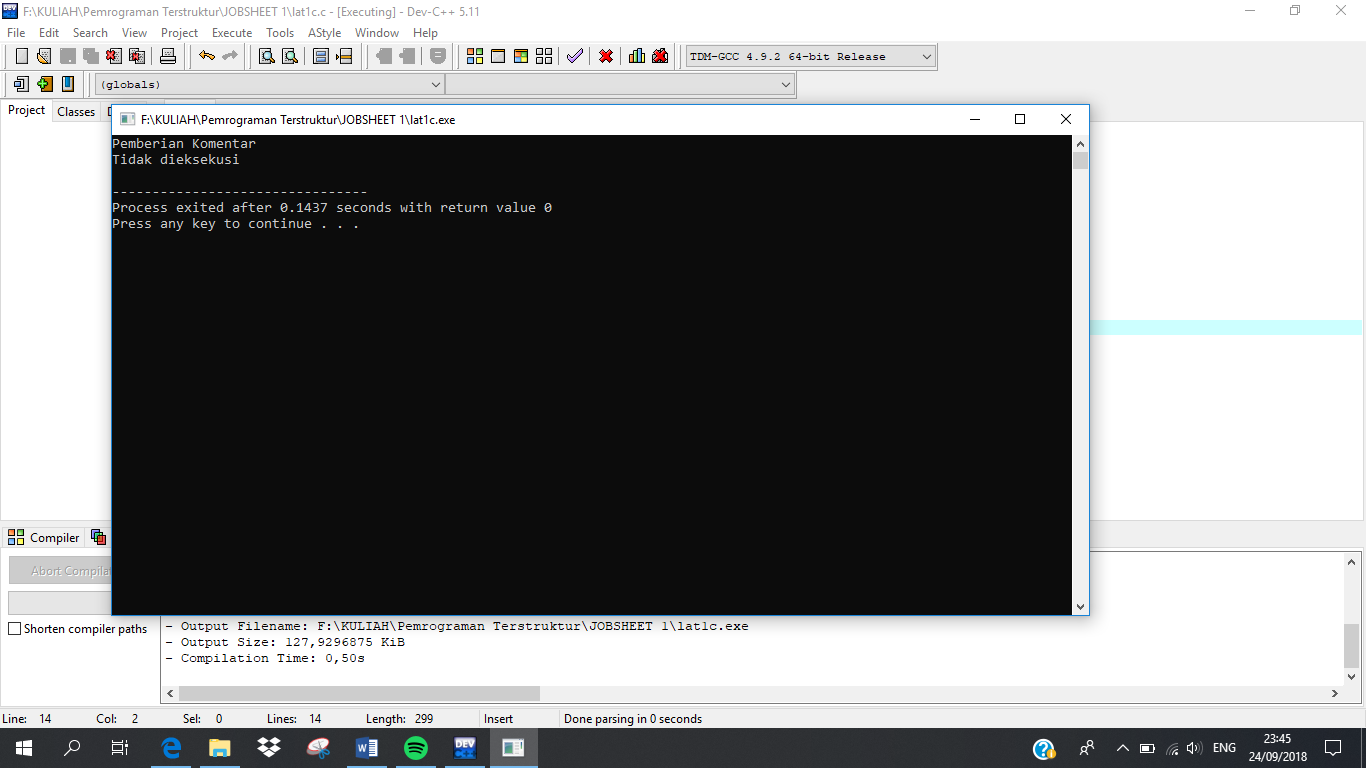


1. Lalu compile apabila masih terdapat error benarkan hingga tidak terdapat error.



Setelah dicompile tidak ada error.

1. Lalu klik Run maka akan muncul programnya.



1. PERTANYAAN
2. Apa yang dimaksud dengan interpreter dan compiler.

Interpreter adalah perangkat lunak yang mampu mengeksekusi code program (yang ditulis oleh programmer) lalu menterjemahkannya ke dalam bahasa mesin, sehingga mesin melakukan instruksi yang diminta oleh programmer tersebut. Perintah-perintah yang dibuat oleh programmer tersebut dieksekusi baris demi baris, sambil mengikuti logika yang terdapat di dalam kode tersebut. Proses ini sangat berbeda dengan compiler, dimana pada compiler, hasilnya sudah langsung berupa satu kesatuan perintah dalam bentuk bahasa mesin, dimana proses penterjemahan dilaksanakan sebelum program tersebut dieksekusi.

Compiler sendiri adalah program sistem yang digunakan sebagai alat bantu dalam pemrogaman.Perangkat lunak yang melakukan proses penterjemahan code (yang dibuat programmer) ke dalam bahasa mesin. Hasil dari terjemahan ini adalah bahasa mesin. Pada beberapa compiler, output berupa bahasa mesin dilaksanakan dengan proses assembler yang berbeda.

1. Sebutkan arti dari fungsi main(), printf(), dan #include.

* Arti fungsi main()

Sebagai salah satu dasar penyusunan blok pada C++. Fungsi ini menjadi awal dan akhir eksekusi program C++. Sedangkan Tubuh fungsi dimulai dari tanda { hingga tanda }. Lebih spesifik lagi, semua yang terletak didalam tanda { } disebut blok. Tanda () digunakan untuk mengapit argumen fungsi, yaitu nilai yang akan dilewatkan ke fungsi. Pada fungsi main() seperti pada contoh, tidak ada argumen yang diberikan. Oleh karena itu tidak ada entri di dalam ().

* Arti fungsi printf()

Arti fungsi printf() adalah sebagai fungsi library untuk menampilkan informasi pada layar monitor. Statemen ini dapat menampilkan text sederhana atau pesan dan nilai satu atau lebih variable program.

* Arti fungsi #include

Pengarah #include mengintruksikan pada compiler C untuk menambahkan isi file include (file header) pada program yang dibuat saat compilasi. Pengarah #include pada contoh program diatas berarti “tambahkan isi file stdio.h”. Umumnya program C selalu membutuhkan satu atau lebih file header.

1. Apa kegunaan dari perintah \n,\t,\f,\d, dan \c.

\n (Newline) : untuk meletakkan baris baru(ganti baris )

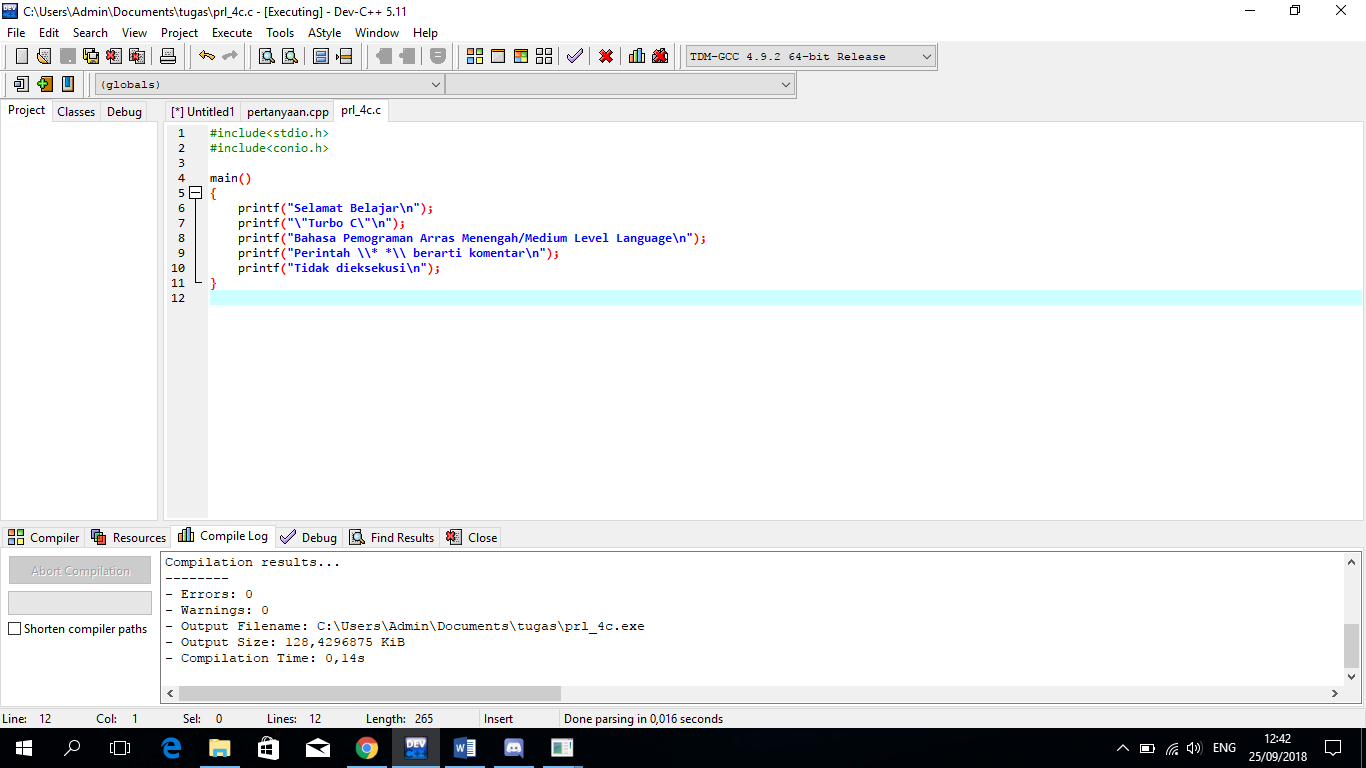
\t (Horizontal tab) : untuk meletakkan tab horizontal

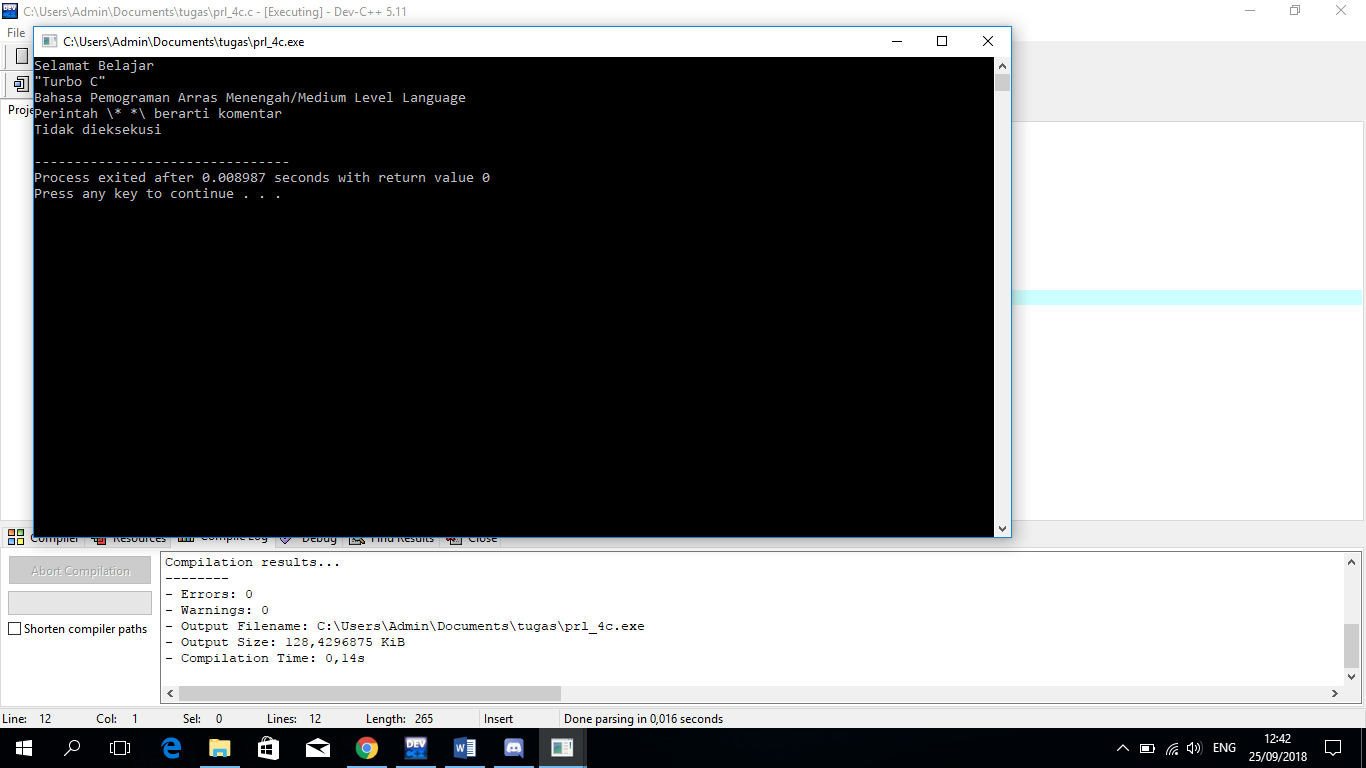
\f (Formfeed) : untuk meletakkan formfeed

\d

\c

4. Buatlah program dan simpan dengan nama prl\_4c : Selamat Belajar “Turbo C” Bahasa Pemograman Arras Menengah/Medium Level Language Perintah \\* \*\ berarti komentar Tidak dieksekusi





1. **KUIS**

“Program Hitung Luas Segitiga”

= = = = = = = = = = = = = = = = =

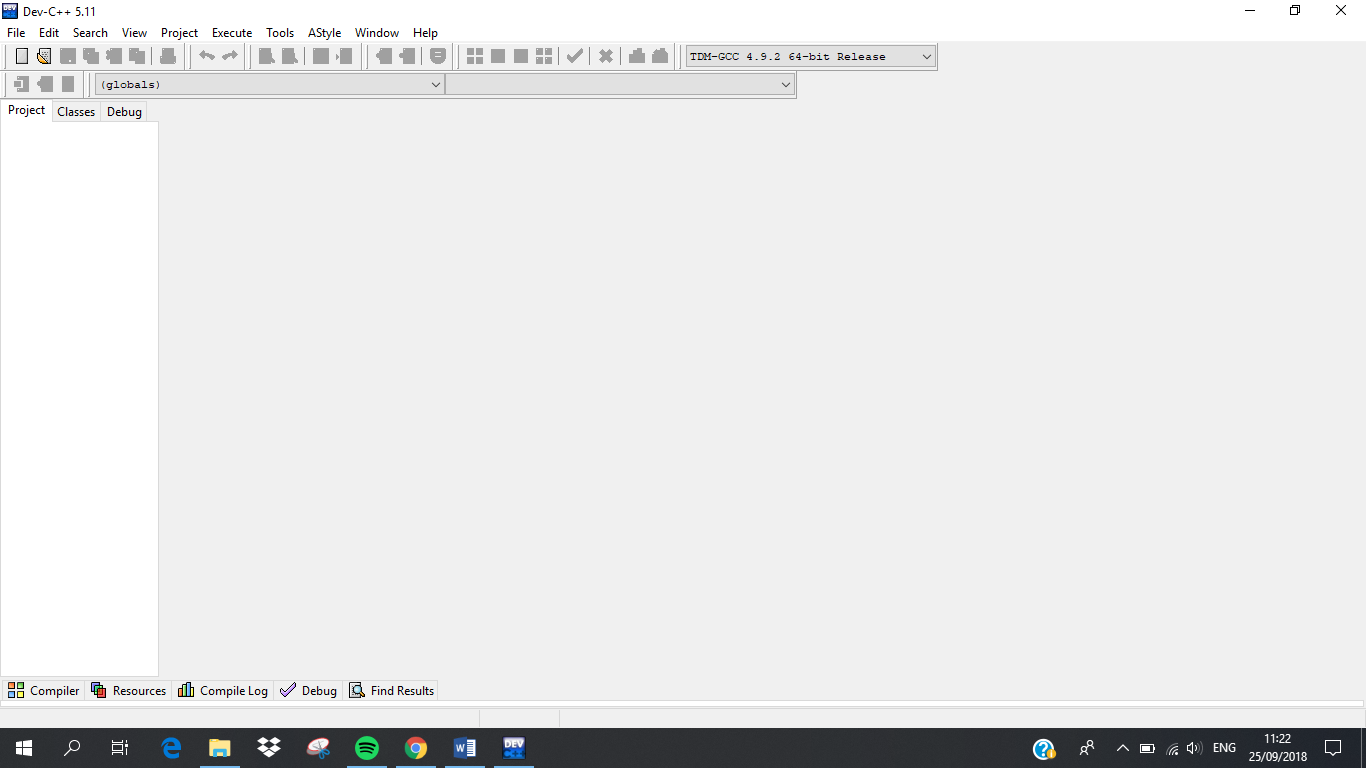
Masukkan alas segitiga : … cm

Masukkan tinggi segitiga : … cm

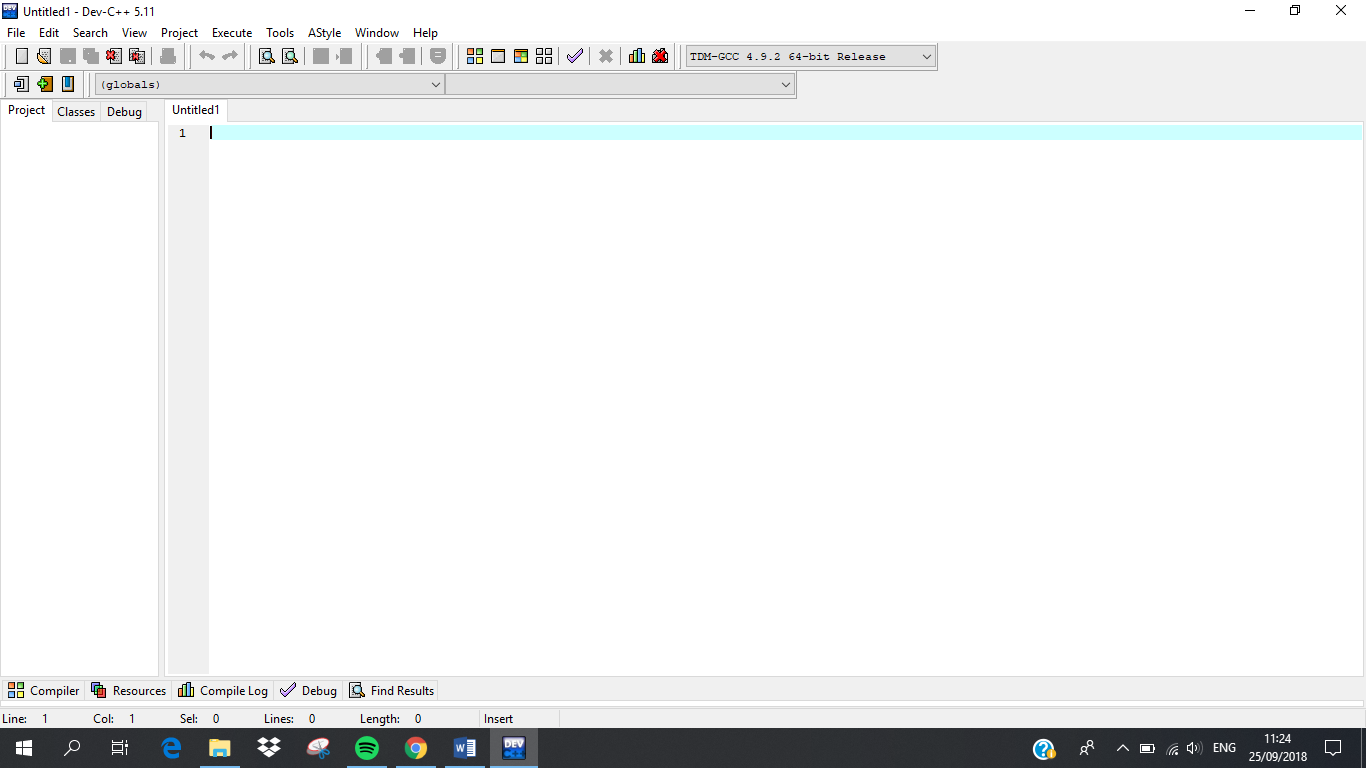
Luas Segitiga : 1/2\*alas\*tinggi

Jadi luas segitiga dengan alas = … dan tinggi … adalah

1. Buka aplikasi Dev C++



1. Alt + F, pilih New



1. Ketik :

#include <stdio.h>

#include <conio.h>

int alas,tinggi,luas;

main()

{

printf("Program Hitung Luas Segitiga\n");

printf("======================\n");

printf("\n");

printf("Masukkan Alas Segitiga:\t\t");

scanf("%d", &alas);

printf("Masukkan Tinggi Segitiga :\t");

scanf("%d", &tinggi);

luas = alas\*tinggi\*1/2;

printf("\n");

printf("Jadi Luas Segitiga Dengan :\n");

printf("Alas = %d cm\n", alas );

printf("Tinggi = %d cm\n", tinggi );

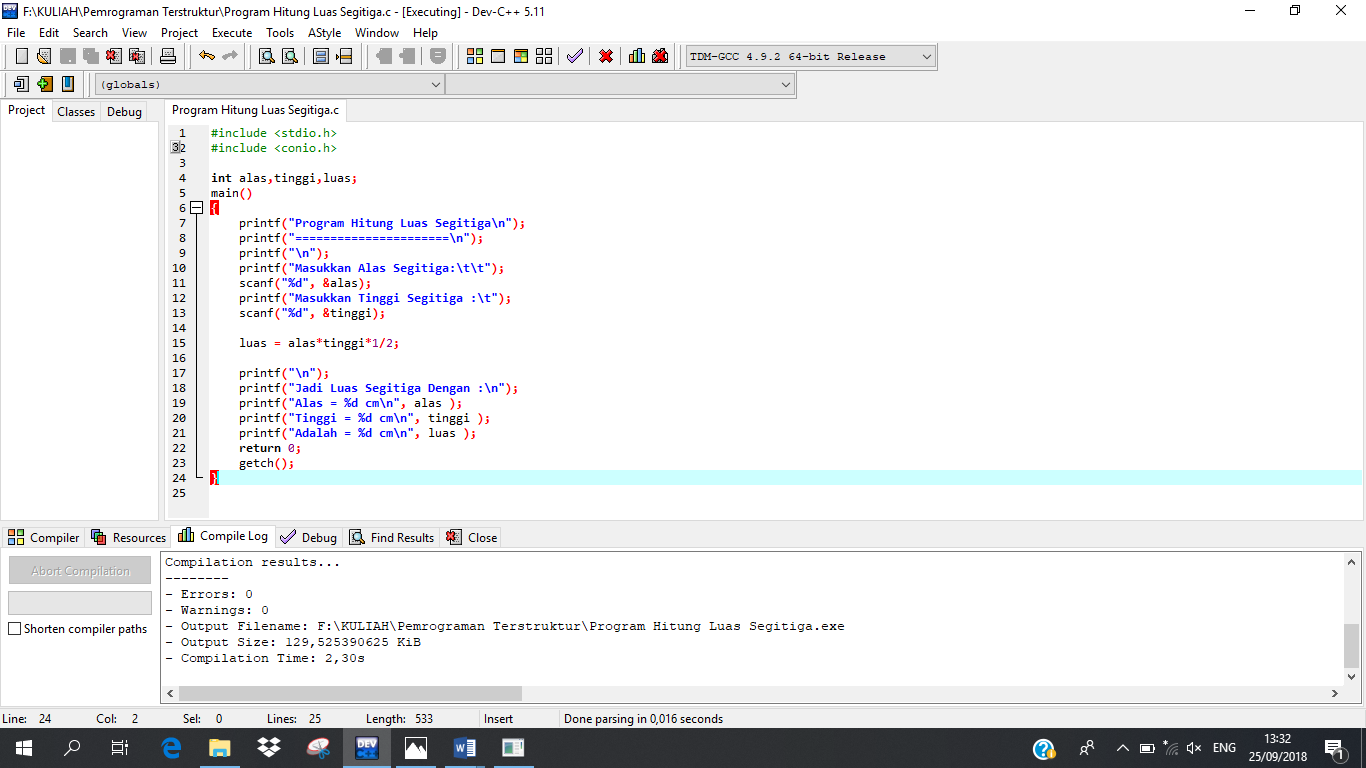
printf("Adalah = %d cm\n", luas );

return 0;

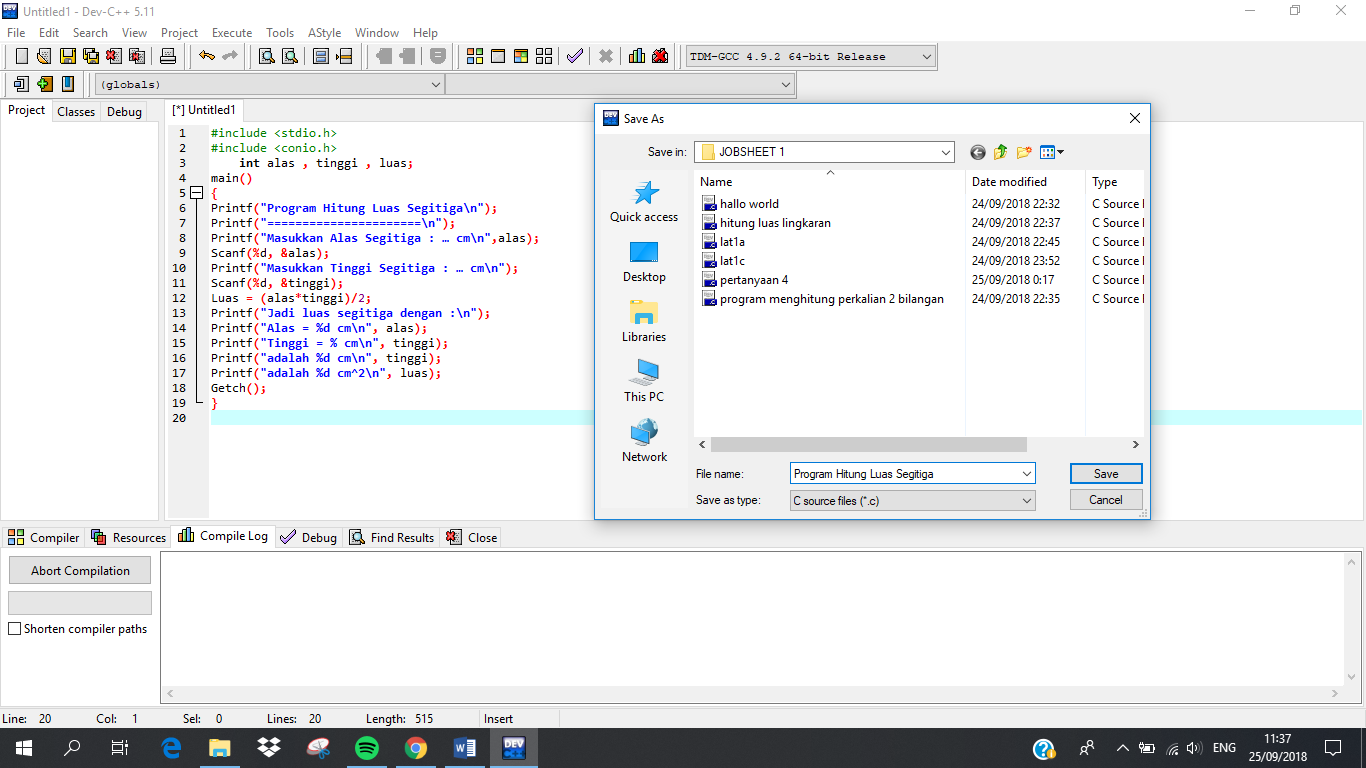
getch();

} Return(0.5\* x\* y);

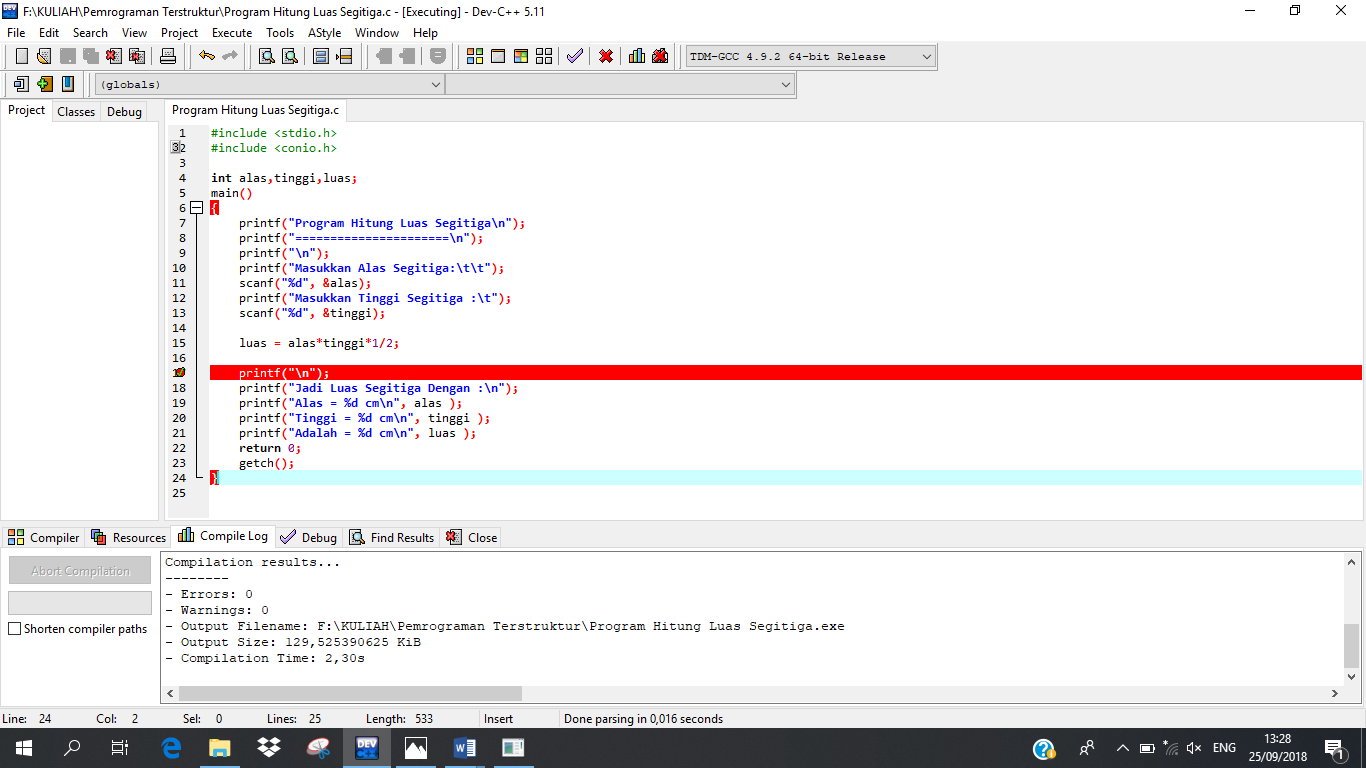
}

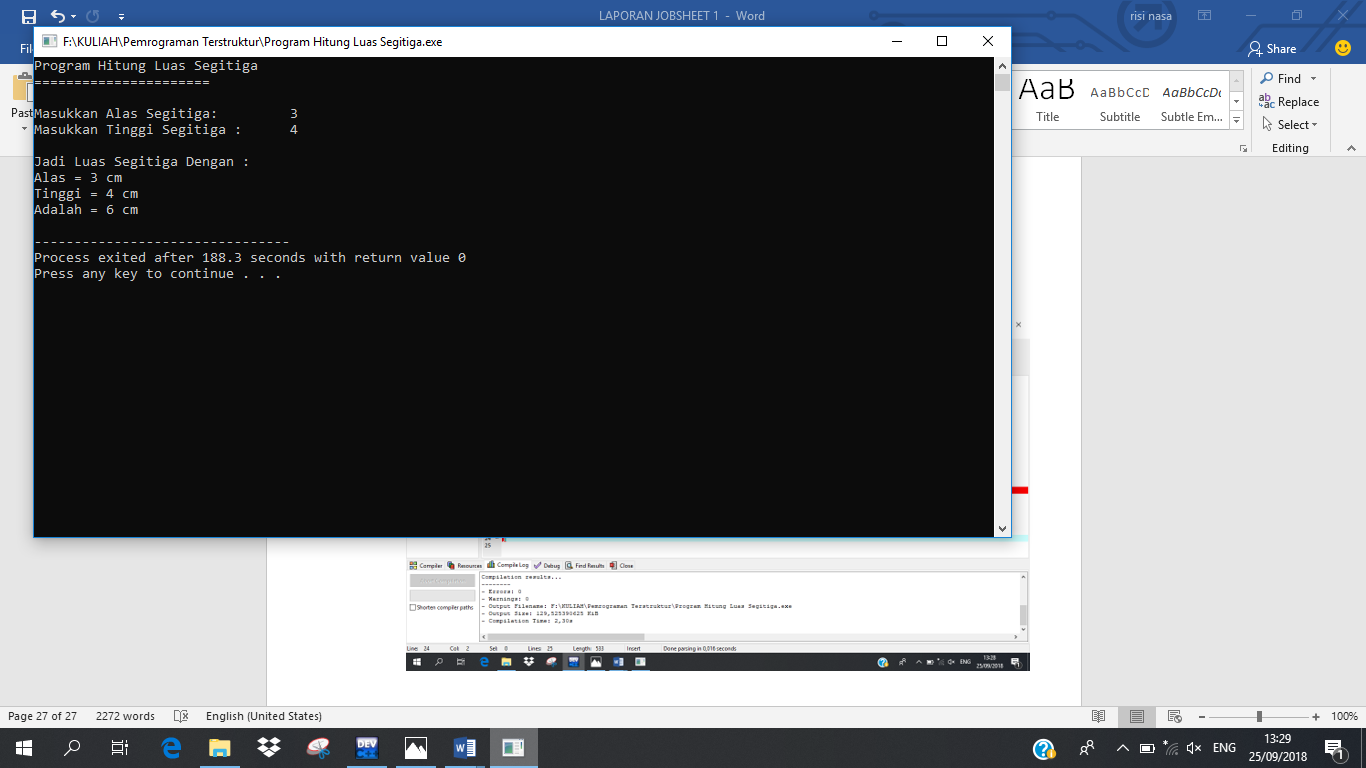


1. Alt+F, pilih save ketik nama Program Hitung Luas Segitiga.c



1. Lalu compile apabila masih terdapat error benarkan hingga tidak terdapat error.





1. KESIMPULAN

Jadi kesimpulan dari praktikum ini adalah banyak sekali kode atau symbol yang harus dipahami dengan baik dan bendar.