**LAPORAN JOBSHEET 10**

**Operasi File**



Disusun Oleh :

DWI RINAYATUS SA’ADAH

3.34.18.1.10

IK- 1B

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI SEMARANG**

**TAHUN 2018/2019**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc533150669)

[BAB 10 4](#_Toc533150670)

[OPERASI FILE 4](#_Toc533150671)

[I. TUJUAN 4](#_Toc533150672)

[II. DASAR TEORI 4](#_Toc533150673)

[a) Pendahuluan 4](#_Toc533150674)

[b) Jenis Operasi File 4](#_Toc533150675)

[c) Menutup File 6](#_Toc533150676)

[d) Operasi Penyimpanan File 6](#_Toc533150677)

[e) Operasi Pembacaan File 6](#_Toc533150678)

[f) Jenis File 7](#_Toc533150679)

[g) Operasi Baca dan Tulis File Per-Int 7](#_Toc533150680)

[h) Operasi Baca dan Tulis File Per-Blok 7](#_Toc533150681)

[i) Operasi Baca dan Simpan Data String pada File 8](#_Toc533150682)

[j) Akses File Biner Secara Acak 8](#_Toc533150683)

[k) Operasi Menghapus File 9](#_Toc533150684)

[l) Operasi Mengganti Nama File 9](#_Toc533150685)

[III. LANGKAH KERJA 10](#_Toc533150686)

[1. Latihan 1 Dasar Operasi File 10](#_Toc533150687)

[a. Latihan 1a : program menciptakan file 10](#_Toc533150688)

[b. Latihan 1b : program baca file 13](#_Toc533150689)

[c. Latihan 1c : program create & baca file 16](#_Toc533150690)

[2. Latihan 2 Operasi Tulis & Baca File Per-Int 19](#_Toc533150691)

[a. Latihan 2a : program menciptakan file 19](#_Toc533150692)

[b. Latihan 2b : program membaca file 22](#_Toc533150693)

[3. Latihan 3 Operasi Tulis & Baca File Per-Blok 22](#_Toc533150694)

[a) Latihan 3a : program menciptakan file 22](#_Toc533150695)

[b) Latihan 3b : program baca file 26](#_Toc533150696)

[4. Latihan 4 Operasi Tulis & Simpan Data String 29](#_Toc533150697)

[a. Latihan 4a : program tulis file2 29](#_Toc533150698)

[b. Latihan 4b : program tulis file2 31](#_Toc533150699)

[c. Latihan 4c : program tulis file2 34](#_Toc533150700)

[5. Latihan 5 Contoh Program fseek() 36](#_Toc533150701)

[a) Latihan 5a : program mencari karakter 36](#_Toc533150702)

[b) Latihan 5b : program mencari string 39](#_Toc533150703)

[6. Latihan 6 Contoh Program fseek 42](#_Toc533150704)

[IV. PERTANYAAN 47](#_Toc533150705)

[V. QUIZ 49](#_Toc533150706)

# BAB 10

# OPERASI FILE

## TUJUAN

1. Menjelaskan tentang struktur file
2. Menjelaskan tentang tahap-tahap operasi pada file
3. Menjelaskan tentang fungsi untuk penyimpanan dan pembacaan file per- karakter
4. Menjelaskan tentang file biner dan file teks
5. Menjelaskan tentang operasi penyimpanan dan pembacaan file per-int
6. Menjelaskan tentang operasi penyimpanan dan pembacaan file per-blok
7. Menjelaskan cara membaca dan menyimpan data string pada file
8. Menjelaskan cara mengakes file biner secara acak
9. Menjelaskan cara menghapus file
10. Menjelaskan cara mengganti nama file

## DASAR TEORI

### Pendahuluan

* 1. Penyimpanan suatu data dalam disk berupa suatu file.
  2. Gambar struktur file:
  3. Tahapan Operasi File:
     + Membuka / mengaktifkan file
     + Melaksanakan proses file
     + Menutup file
* Bentuk deklarasi:

FILE \*fopen(char \*namafile, char \*mode);

Keterangan :

* **namafile** berupa nama dari file yang akan diaktifkan
* **mode** berupa jenis oprasi yang akan dilakukan terhadap file
* **prototipe** ada pada file stdio.h

### Jenis Operasi File

1. **r** menyatakan file hanya akan dibaca, jika file belum ada maka tidak akan berhasil.
2. **w** menyatakan bahwa file baru diciptakan. Jika file tersebut sudah ada dalam disk, isinya yang lama akan terhapus.
3. **a** untuk membukanfile yang sudah ada untuk ditambah dengan data, jika file belum ada akan dibuat yang baru
4. **r+** sama dengan **“r”** tetapi selain file dapat dibaca, file juga dapat ditulisi.
5. **w+** sama dengan **“w”** tetapi selain file dapat ditulisi, file juga dapat dibaca.
6. **a+** sama dengan **“r”** tetapi selain file dapat ditulisi, file juga dapat dibaca.
7. Berhasil tidaknya operasi pengaktifan file dapat dilihat pada keluaran fungsi *fopen()*.
8. Jika keluaran fungsi berupa NULL (suatu makro yang didefinisikan pada file stdio.h), berarti operasi pengaktifan file gagal  misal membuka file dengan mode “r” tapi filenya belum ada
9. Contoh :

FILE \*pf; //deklarasi variabel pf pf = fopen(“COBA.TXT” “W”)

* Menciptakan dan mengaktifkan file bernama “COBA.TXT”
* Dengan mode yaitu “w” (mode penulisan ke file)
* Dan menempatkan pointer-ke-FILE e variabel pointer pf

if(pf = fopen(“COBA.TXT” “w”) == NULL)

{

printf(“File tidak dapat diciptakan !\n”);

exit(1); //keluar dari program

}

Keterangan :

* pf akan diisi dengan keluaran dari fungsi *fopen().*
* Jika nilainya NULL, maka akan mencetak “File tidak dapat diciptakan”, setelah itu program dihentikan

### Menutup File

1. Apabila sudah tidak diproses lagi, maka file tersebut ditutup, karena adanya keterbatasan jumlah file yang dapat dibuka secara serentak.
2. Perintah yang digunakan : fclose();
3. Bentuk deklarasi :

int fclose(FILE\*pf);

1. Bentuk deklarasi yang lain :

int fcloseall(void) fcloseall();

Prototype yang digunakan : stdio.h

### Operasi Penyimpanan File

1. Penyimpanan karakter ke file menggunakan perintah : fputc();
2. Bentuk deklarasi :

int fputc(charkar, FILE ptr\_file)

* **ptr\_file** adalah pointer-ke-FILE yang berisi keluaran dari *fopen(),*
* kar berupa karakter yang kan disimpn dalam file.

### Operasi Pembacaan File

1. Pembacaan karakter dari suatu file memakai perintah : fgetch().
2. Bentuk deklarasi :

int fgetc(FILE \*ptr\_file)

PROSES PEMBACAAN FILE PER KARAKTER :

1. Buka file COBA.TXT dengan mode “r” Jika tidak berhasil dibuka maka
   * beri keterangan pada layar bahwa file tak ada
   * selesai
2. Baca sebuah karakter dari file

jika karakter sama dengan EOF (tanda akhir file) maka ke langkah 4

1. Tampilkan karakter ke layar dan kembali ke langkah 2
2. Tutup file
3. Selesai

### Jenis File

1. File Biner : file yang pola penyimpanan di dalam disk berbentuk biner, yaitu seperti bentuk pada memori RAM (komputer). Dipakai untuk menyimpan data kompleks, mis : struct.
2. File Teks : file yang pola penyimpanan datanya dalam bentuk karakter. Dipakai untuk menyimpan data seperti karakter atau string.
3. Penentuan mode teks dan mode biner :

t : untuk mode teks

b : untuk mode biner

Contoh :

“rt” : mode file teks dan file hendak dibaca

“rt+” : mode file teks dan file bisa dibaca dan ditulisi.

Bisa juga ditulis : “r+t”

“rb” : mode file biner dan file hendak dibaca

### Operasi Baca dan Tulis File Per-Int

* 1. Perintah yang digunakan : \_putw(), \_getw().
  2. Bentuk deklarasi :

int\_putw(int nilai, FILE \*ptr\_file); int\_getw(FILE \*ptr\_file);

Kegunaan :

*\_getw()* untuk membaca sebuah data bertipe int dari file

*\_putw()* untuk menyimpan sebuah data bertipe *int* ke file

### Operasi Baca dan Tulis File Per-Blok

1. Fungsi : untuk menyimpan atau membaca data file dalam bentuk kesatuan blok(sejumlah byte), misal float atau struct.
2. Perintah yang digunakan : fread() dan fwrite();
3. Bentuk deklarasi :

int fread(void \*buffer, int n, FILE \*ptr\_file);

int fwrite(void \*buffer, int jum\_byte, FILE \*ptr\_file);

dengan :

buffer : pointer yang menunjuk ke alamat memori

jum\_byte : jumlah byte yang akan dibaca atau disimpan

n : banyaknya blok data berukuran jum\_byte yang akan ditulis/dibaca

ptr\_file : pointer-ke-FILE yang berisi nilai keluaran dari *fopen()*

### Operasi Baca dan Simpan Data String pada File

1. Perintah yang digunakan : *fgets()* dan *fputs()*

int fputs(char \*str, FILE \*ptr\_file);

char fgets(char \*str, int n, FILE \*ptr\_file);

1. Bentuk deklarasi :

Kegunaan:

fputs() : menyimpan string str ke dalam file.

fgets() : membaca string dari file sampai ditemukannya karakter baris baru ‘\n’ atau setelh (n-1) karakter, dengan n adalah panjang maksimal string yang dibaca per waktu-baca.

**Note :**

* Saat *simpan, fputs()* tidak menambahkan karakter baris-baru(‘\n’) dengan sendirinya, dan karakter tidak ikut disimpan.
* Baik *fgets()* maupun *fputs* digunakan untuk file teks.

### Akses File Biner Secara Acak

1. Tujuan : membaca data di tengah file scr cepat.
2. Perintah yang digunakan : fseek().

int fseek(FILE \*ptr\_file, long int offset, int posisi);

1. Bentuk deklarasi :

dengan :

**ptr\_file** adalah pointer yang berasal dari keluaran *fopen()*

**offset** menyatakan jumlah byte terhadap posisi

**posisi** dapat diisi suatu nilai tertera pada tabel

**Prototype : stdio.h**

|  |
| --- |
| **Konstanta Nilai Lokasi file** |
| **SEEK\_SET 0 Awal**  **file SEEK\_CUR 1 Posisi penunjuk file saat ini**  **SEEK\_END 2 Akhir file** |

Contoh Aplikasi fseek()

### Operasi Menghapus File

Bentuk deklarasi :

int remove(char \*namafile);

Namafile : pointer yang menunjuk ke nama file yang akan dihapus

* Jika operasi hapus berhasil, akan menghasilkan output = 0 Prototype : stdio.h

### Operasi Mengganti Nama File

Bentuk deklarasi :

int rename(char \*namafilelama, char \*namafilebaru);

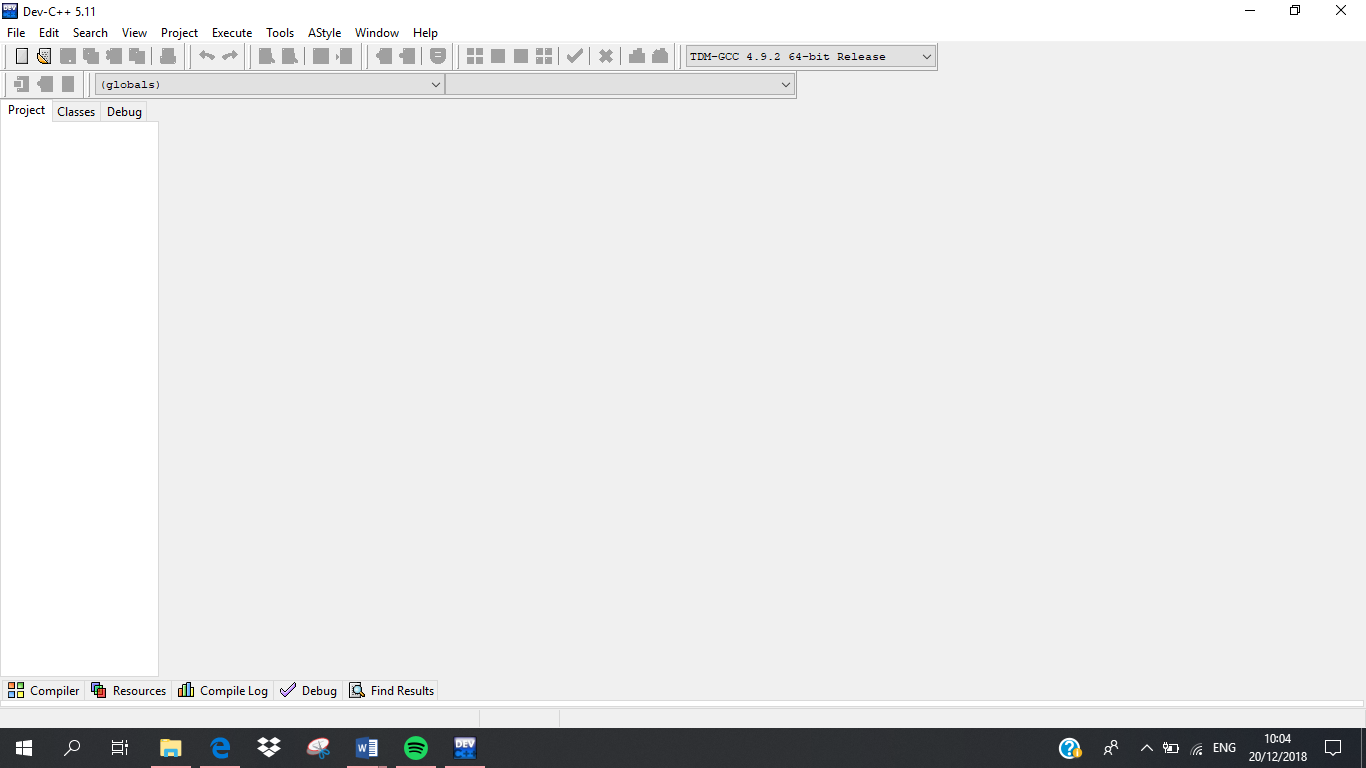
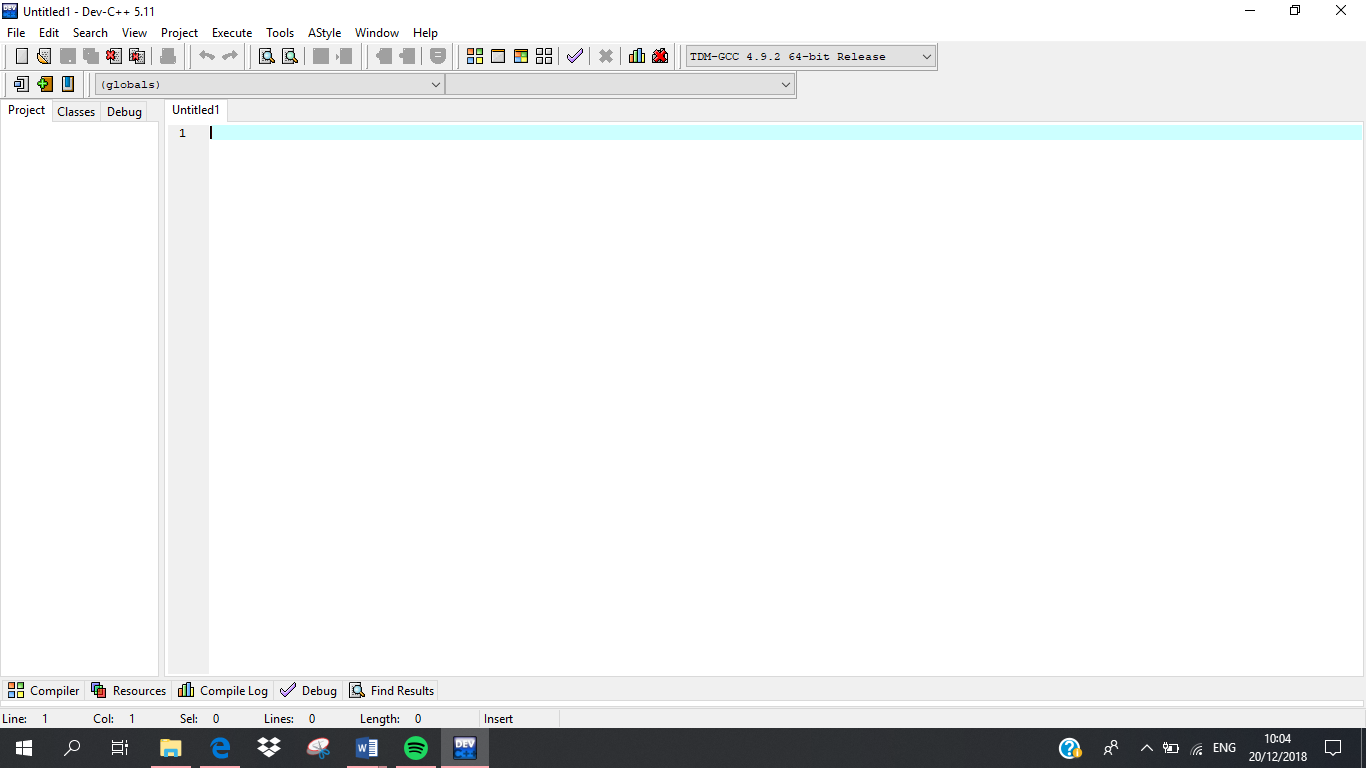
Namafile : pointer yang menunjuk ke nama file yang akan dihapus

* Jika operasi hapus berhasil, akan menghasilkan output = 0 Prototype : stdio.h

## LANGKAH KERJA

#### Latihan 1 Dasar Operasi File

##### Latihan 1a : program menciptakan file

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

#include<stdio.h>

#include<conio.h>

main(){

FILE \*pf; /\* Pointer-ke-FILE \*/

char kar;

/\* Ciptakan file \*/

if((pf=fopen("COBA.TXT","w")) == NULL)

{

printf("file tak dapat diciptakan!\r\n");

exit(1);

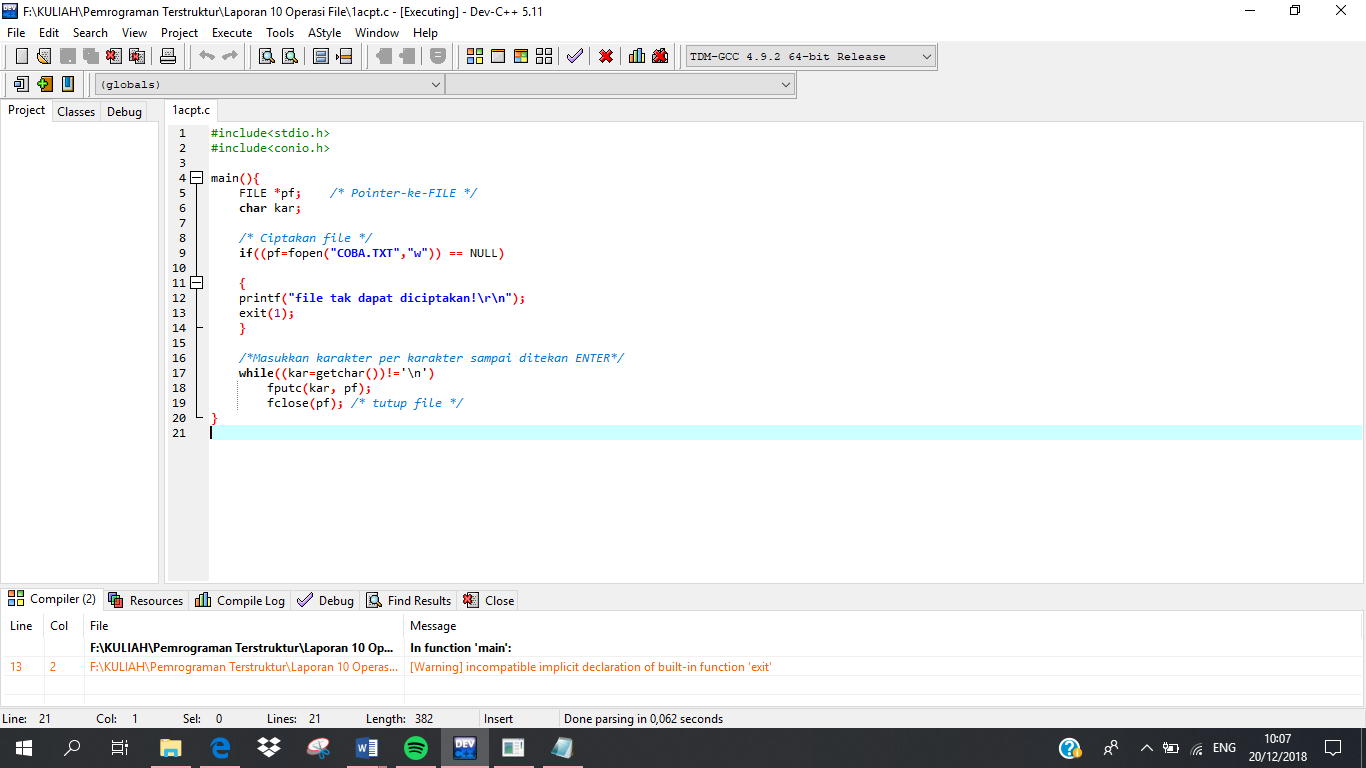
}

/\*Masukkan karakter per karakter sampai ditekan ENTER\*/

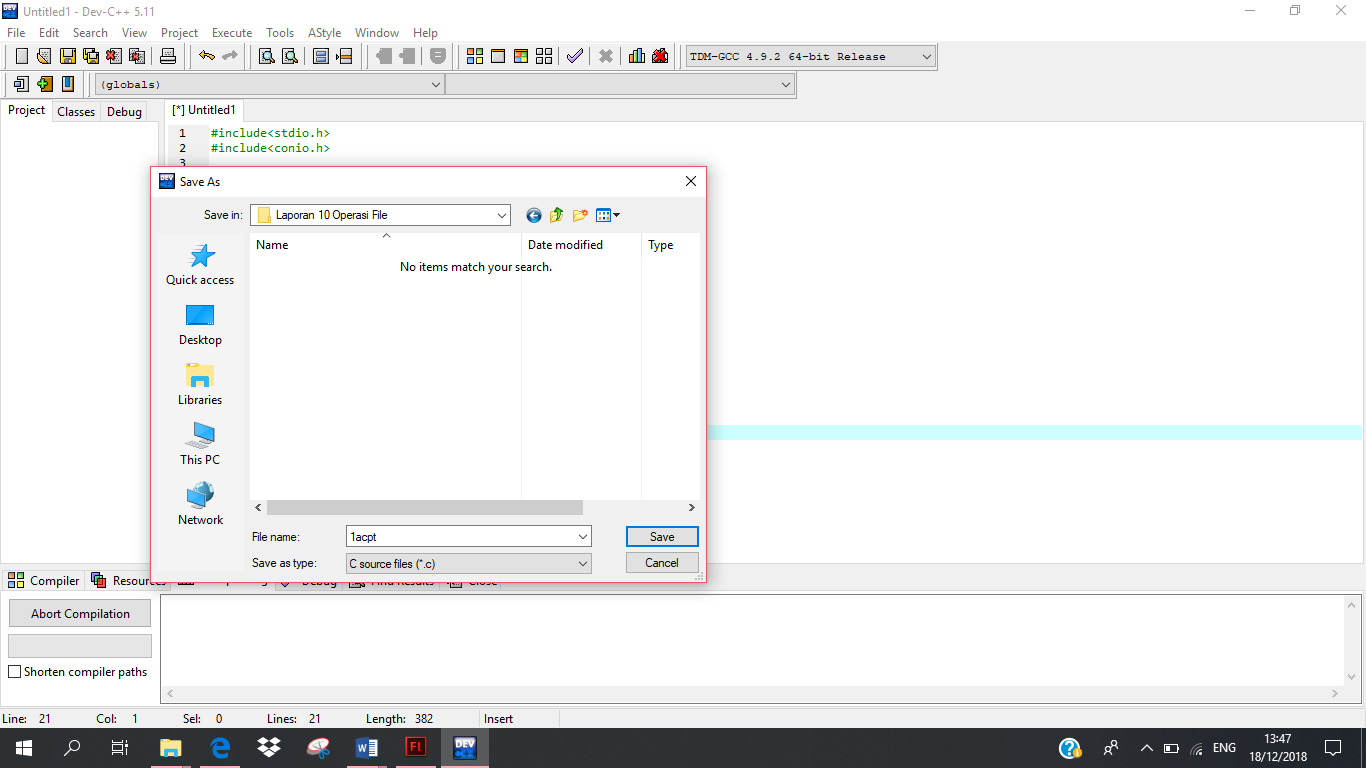
while((kar=getchar())!='\n')

fputc(kar, pf);

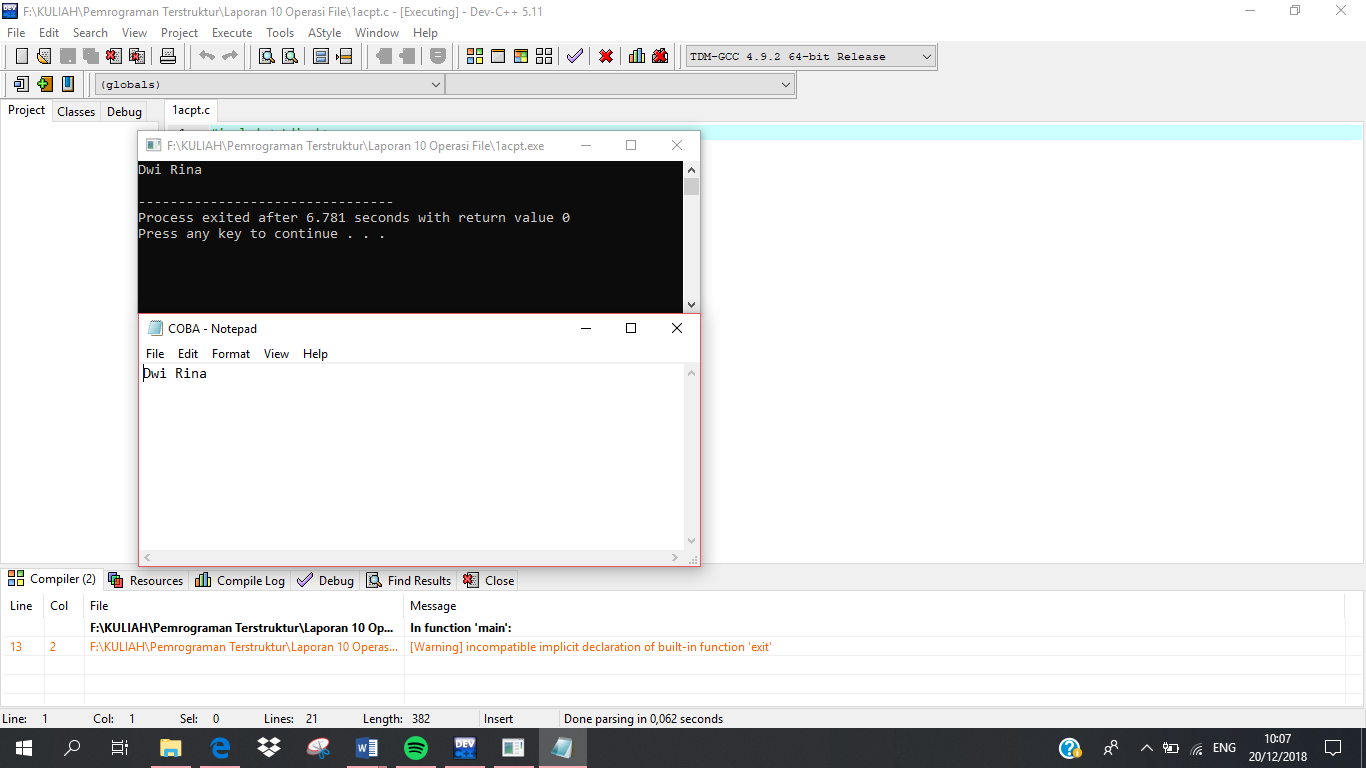
fclose(pf); /\* tutup file \*/

}

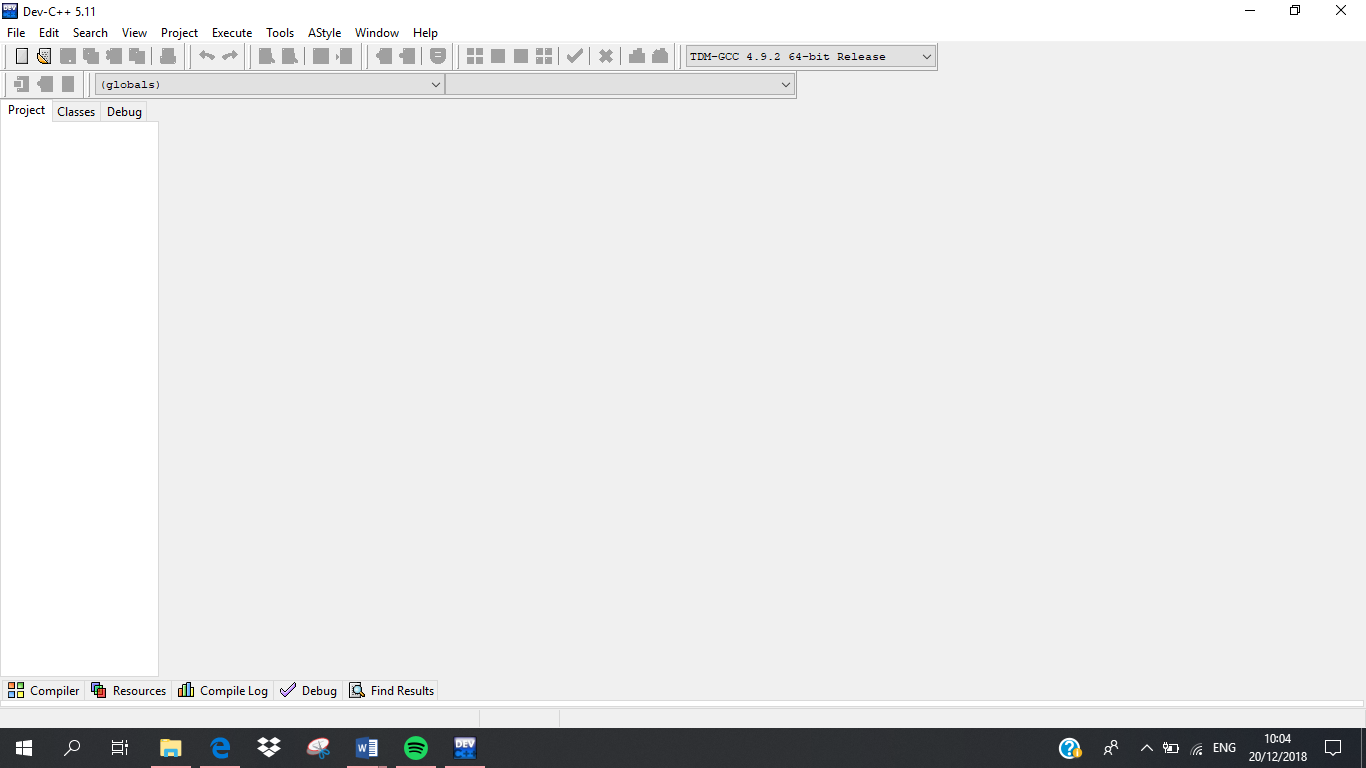
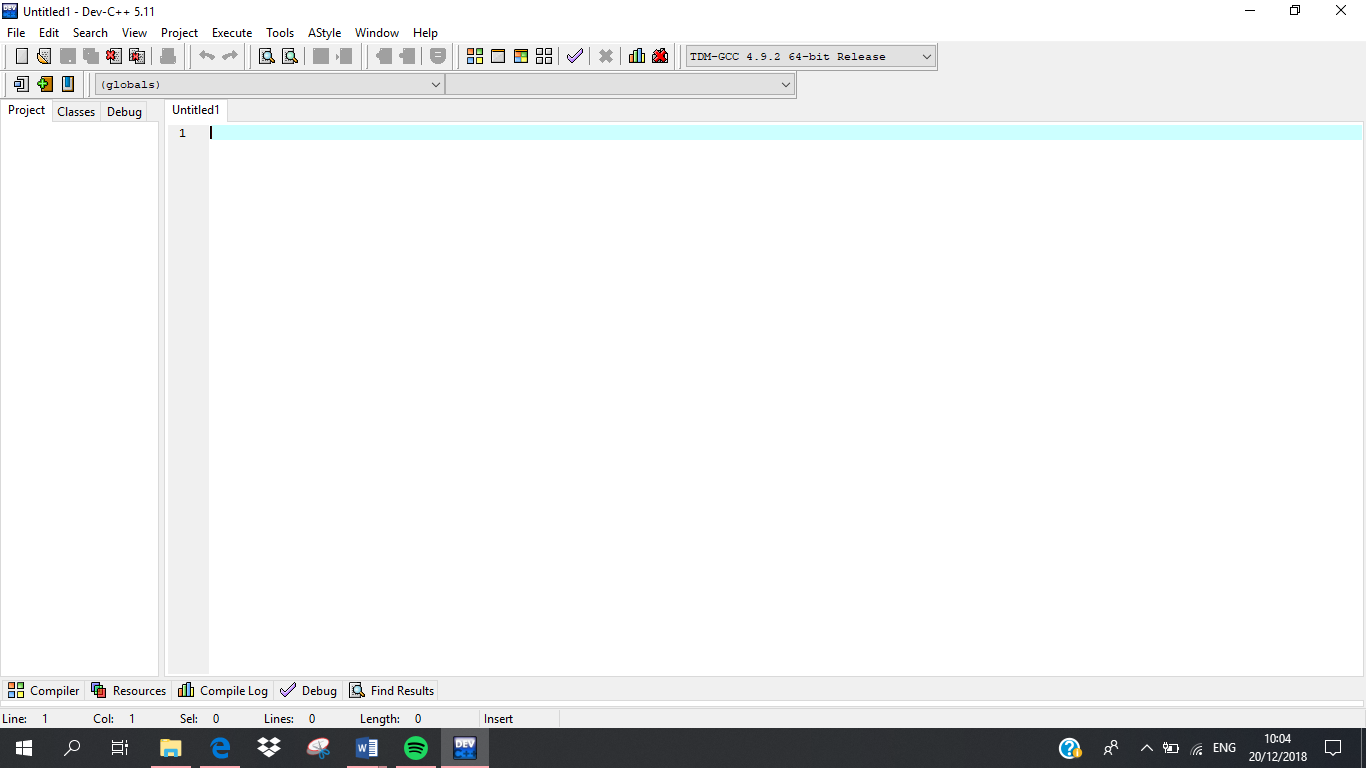
1. Alt+F, pilih save ketik nama 1acpt.c



1. Alt+O, pilih directories
2. Alt+R
3. Hasil Run Program



##### Latihan 1b : program baca file

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

#include<stdio.h>

#include<stdlib.h>

main(){

FILE \*pf;

char kar;

if((pf=fopen("COBA.TXT","r")) == NULL) /\* buka file \*/

{

printf("file tak dapat dibuka !\r\n");

exit(1);

}

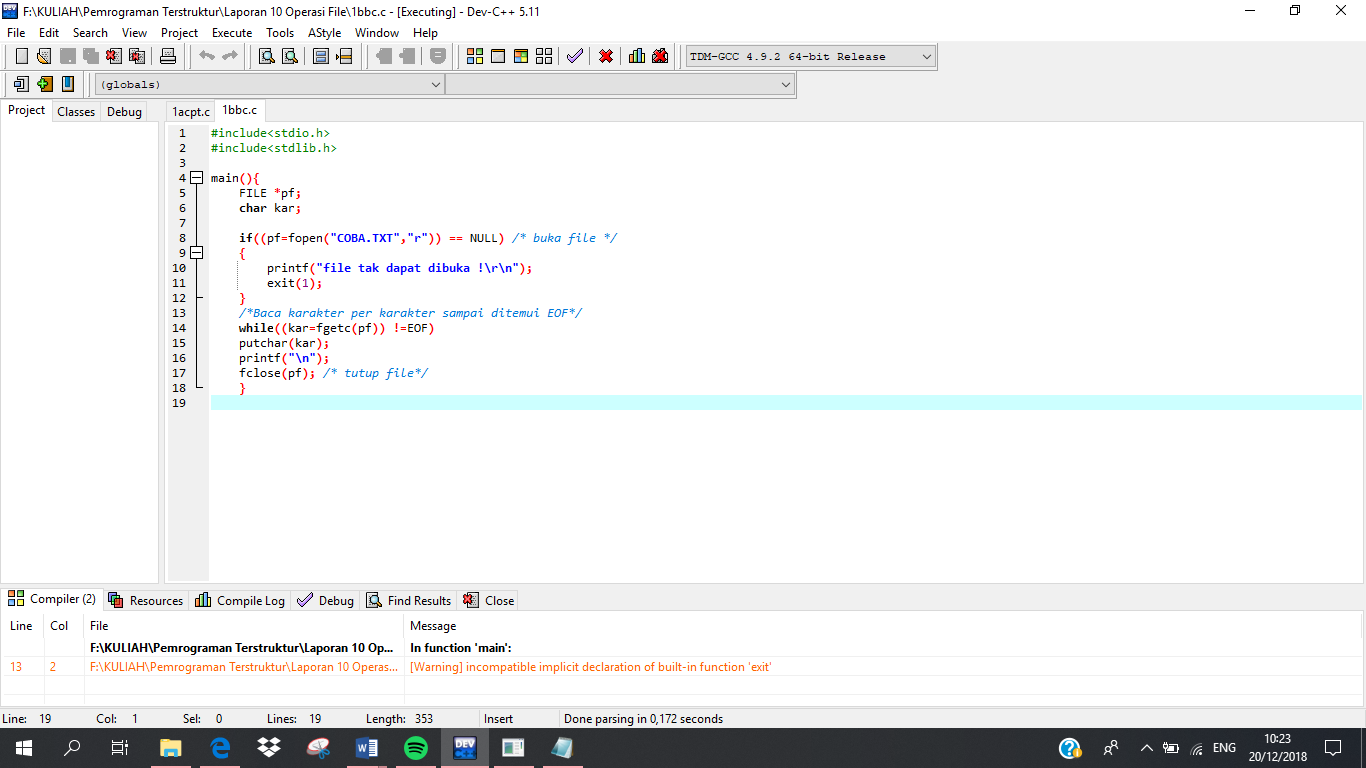
/\*Baca karakter per karakter sampai ditemui EOF\*/

while((kar=fgetc(pf)) !=EOF)

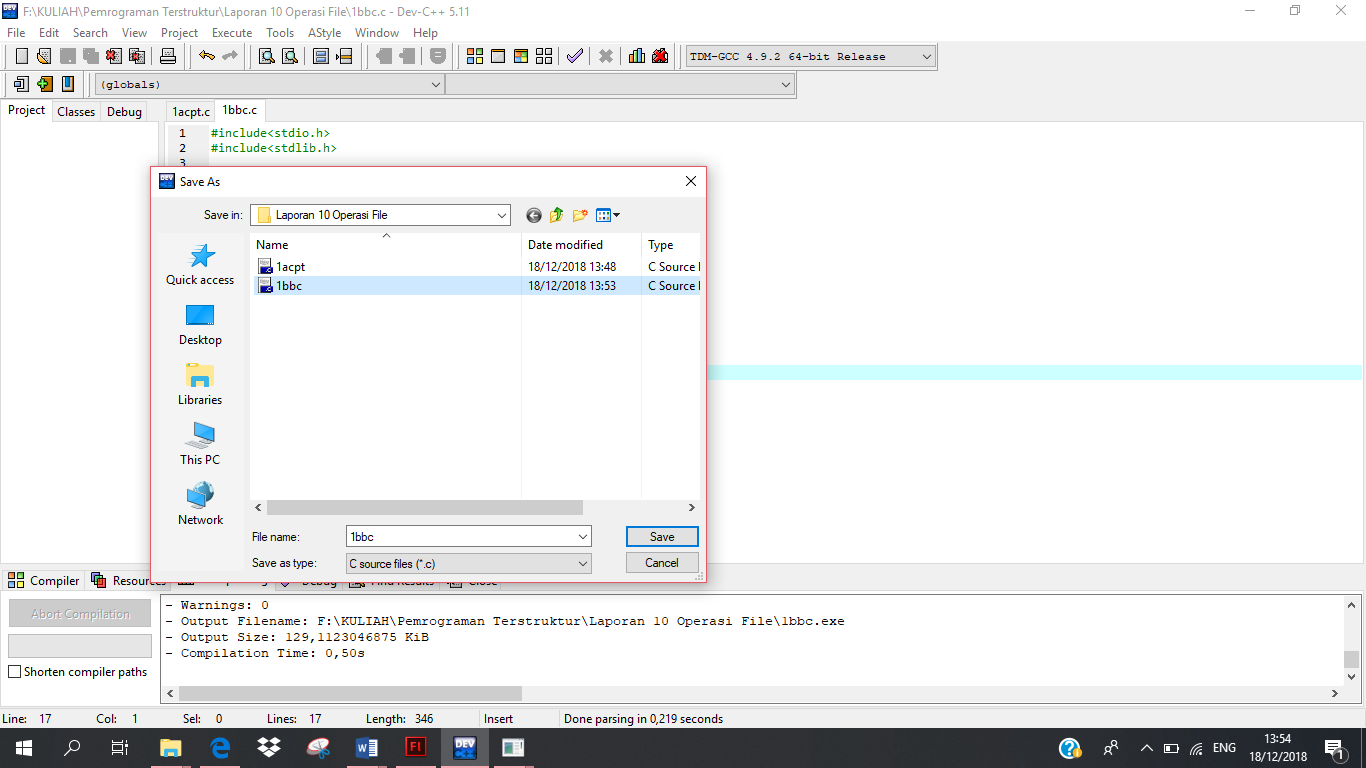
putchar(kar);

printf("\n");

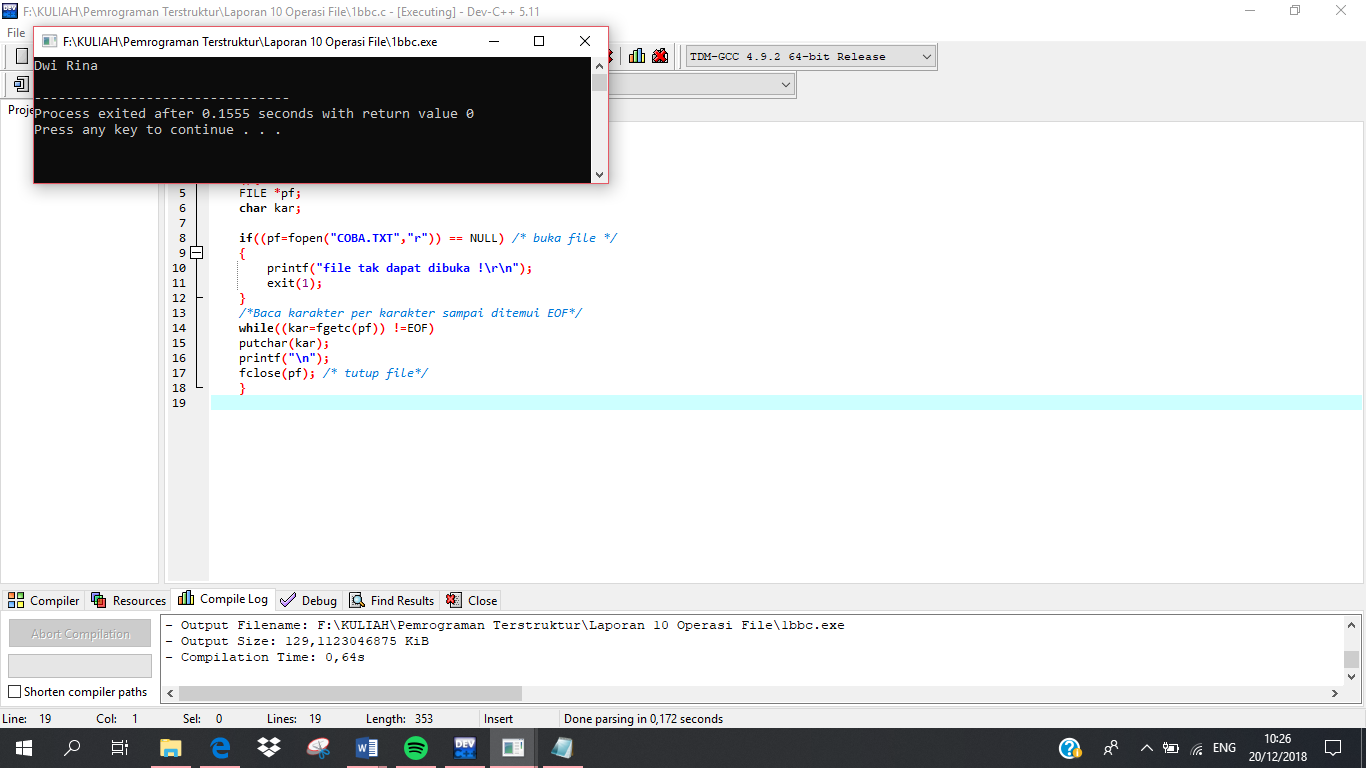
fclose(pf); /\* tutup file\*/

 }

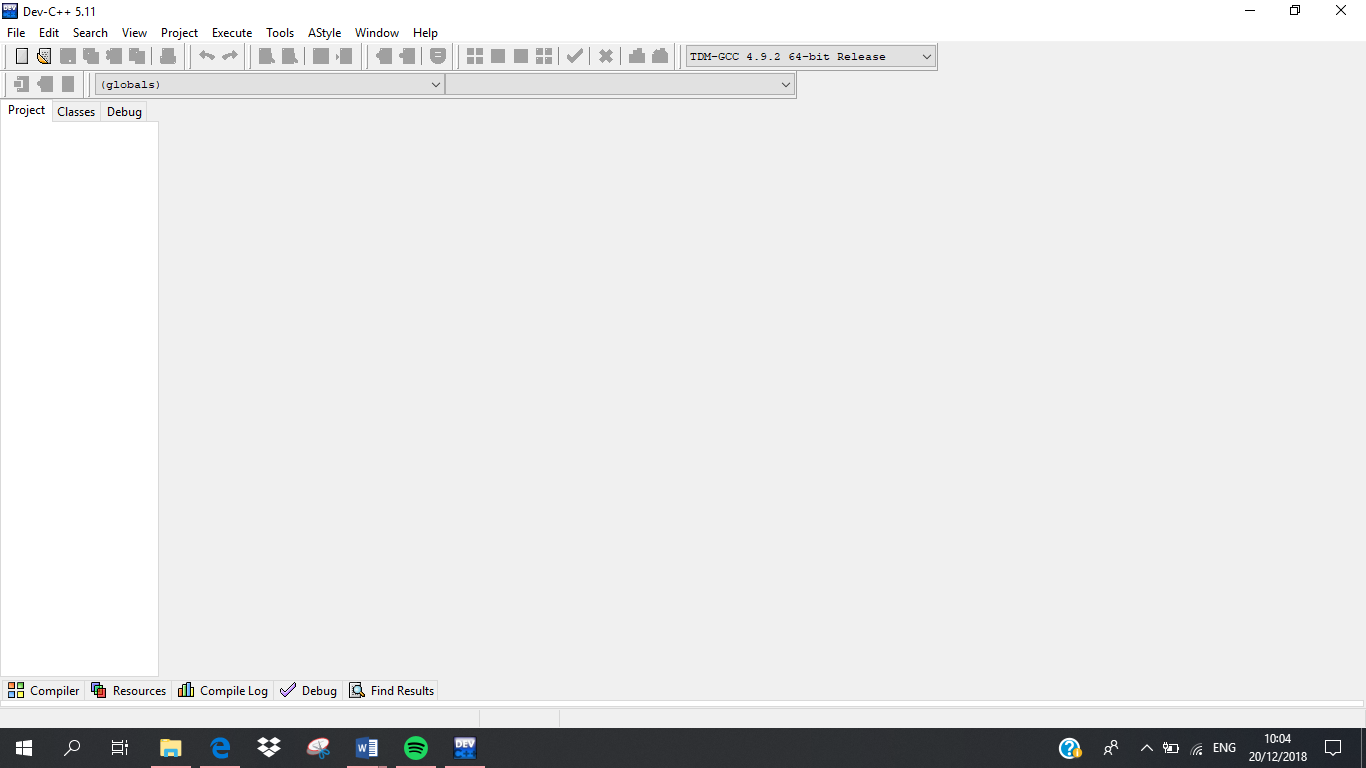
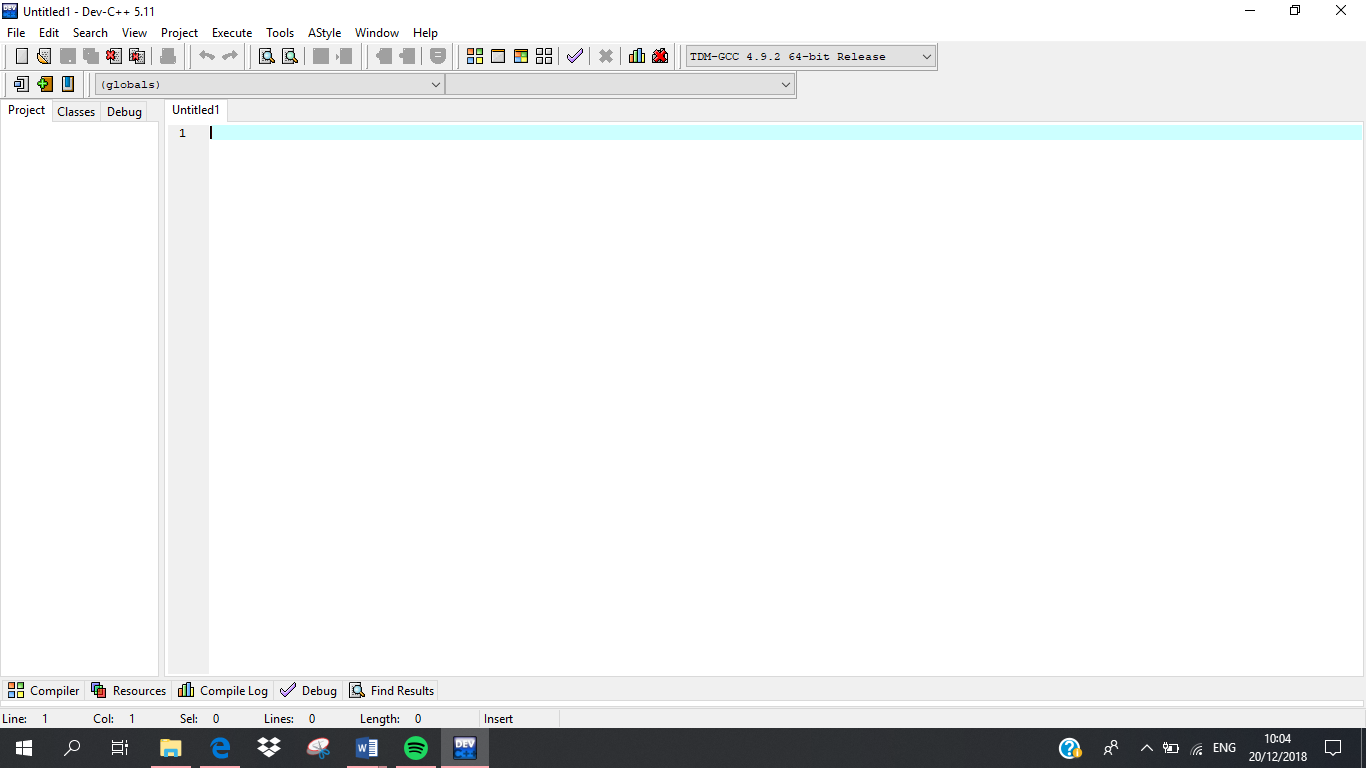
1. Alt+F, pilih save ketik nama 1bbc.c



1. Alt+O, pilih directories
2. Alt+R
3. Hasil Run Program



##### Latihan 1c : program create & baca file

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

#include<stdio.h>

#include<stdlib.h>

main(){

FILE \*pf; /\* Pointer-ke-FILE \*/

char kar;

/\* Ciptakan file \*/

if((pf = fopen("COBA.TXT","r+")) == NULL )

{

printf("file tak dapat diciptakan !\r\n"); exit(0);

}

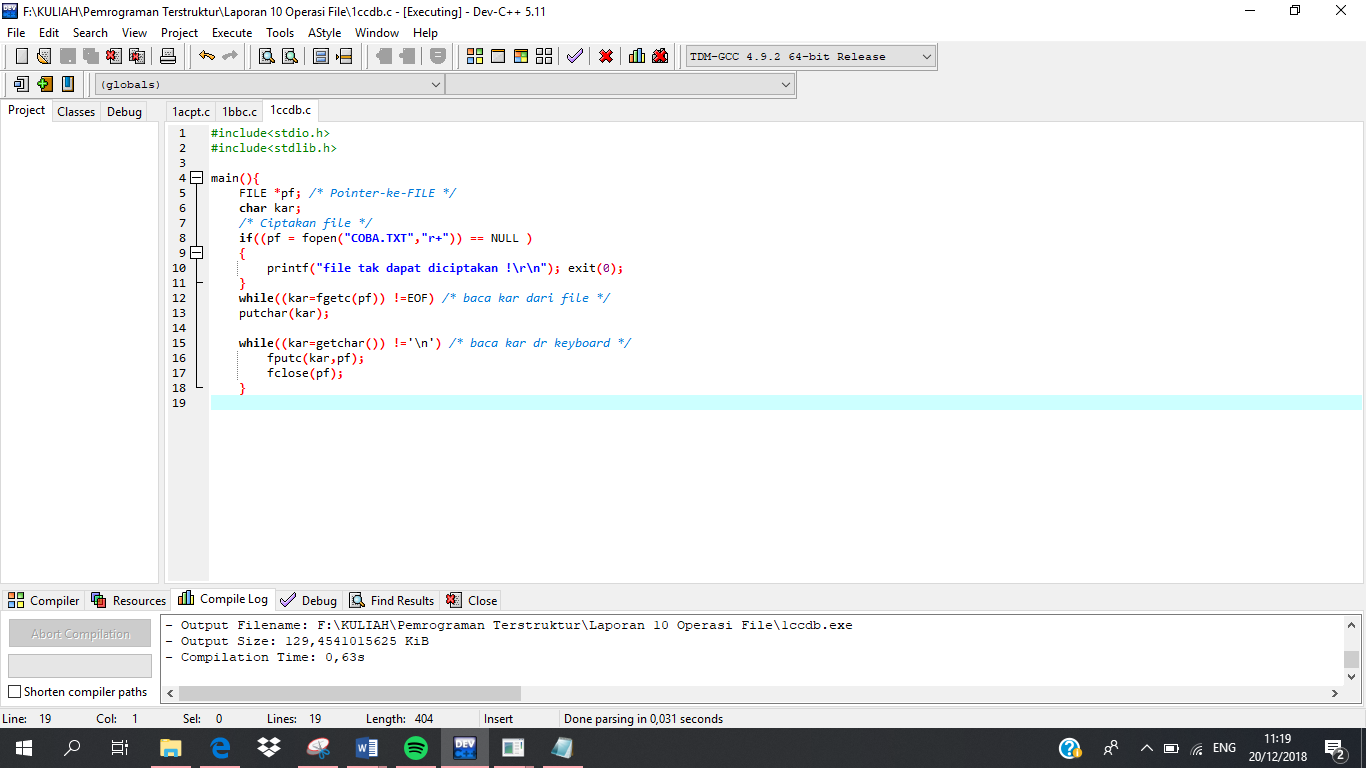
while((kar=fgetc(pf)) !=EOF) /\* baca kar dari file \*/

putchar(kar);

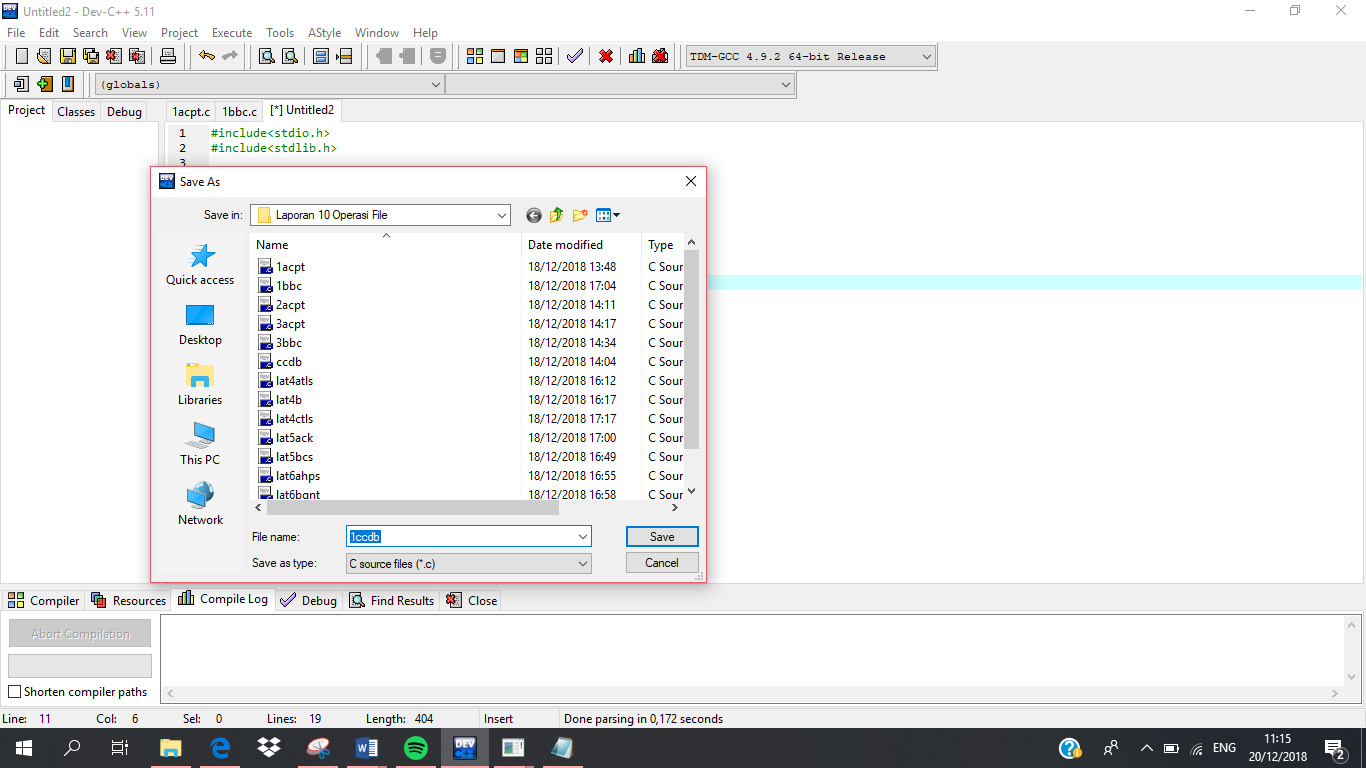
while((kar=getchar()) !='\n') /\* baca kar dr keyboard \*/

fputc(kar,pf);

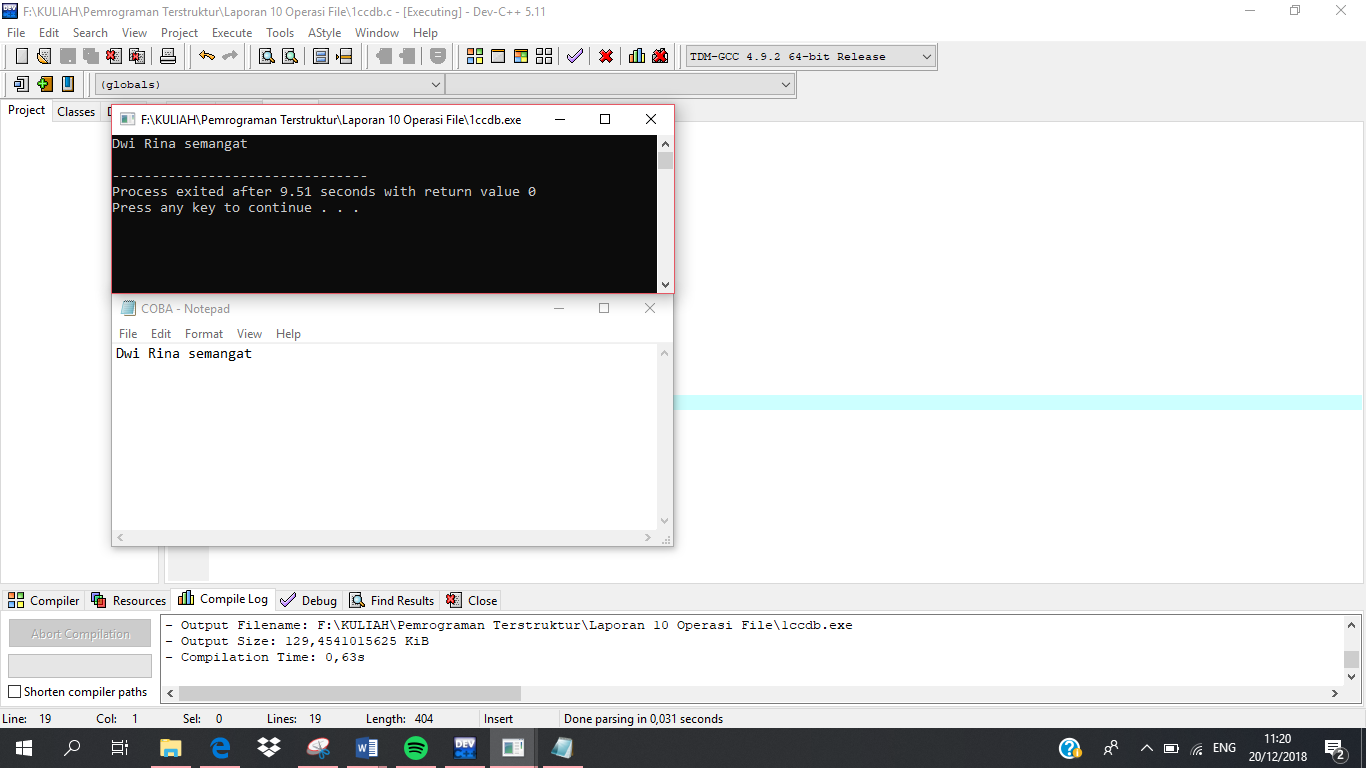
fclose(pf);

 }

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

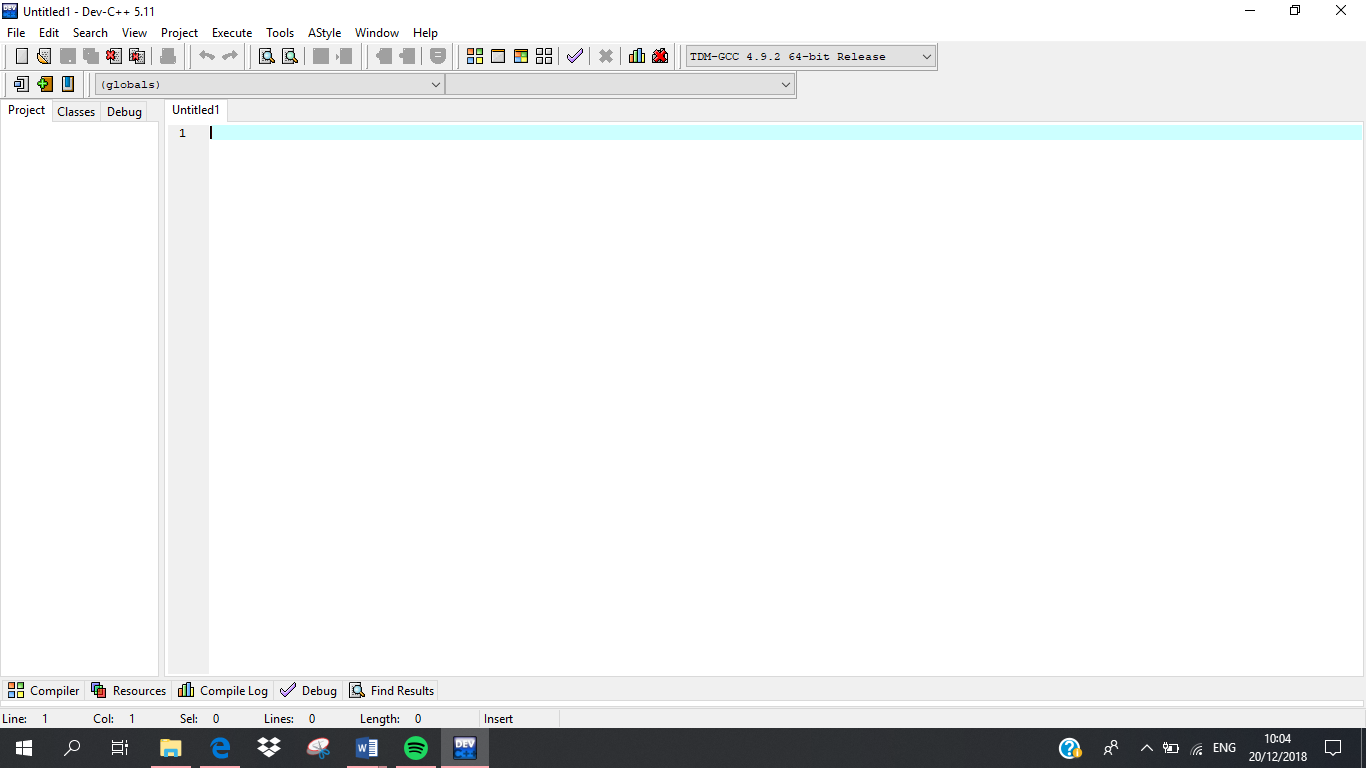
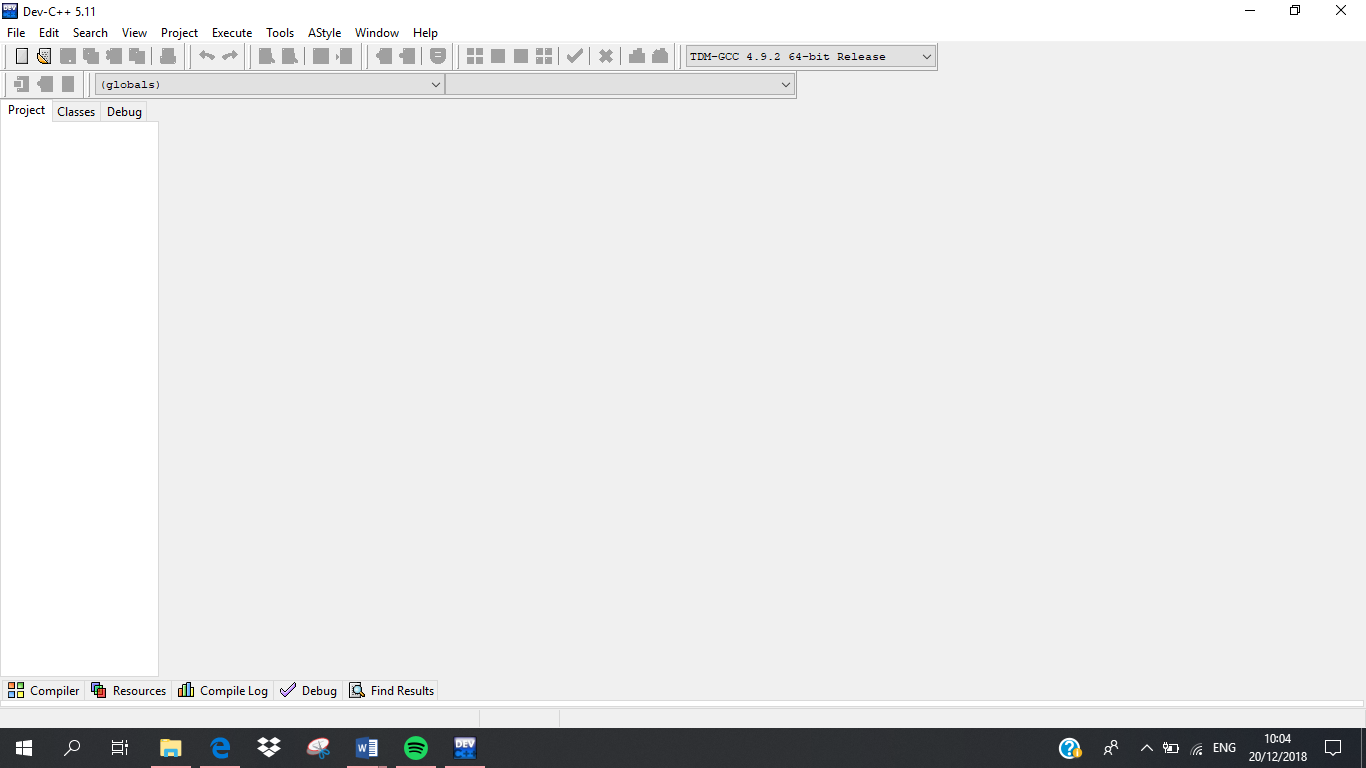


1. Alt+O, pilih directories
2. Alt+R
3. Hasil Run Program



#### Latihan 2 Operasi Tulis & Baca File Per-Int

###### Latihan 2a : program menciptakan file

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

#include <stdio.h>

#include <stdlib.h>

main(){

FILE \*pf; /\* ptr-ke-FILE \*/

int nilai, data, i;

char jawab;

if((pf=fopen("BILANGAN.DAT", "wb")) == NULL)

{

printf("file gagal diciptakan!\n");

exit(1);

}

printf("Masukkan banyaknya data : ");

scanf("%d",&data);

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

{

printf("\nBilangan yang disimpan : ");

scanf("%d",&nilai);

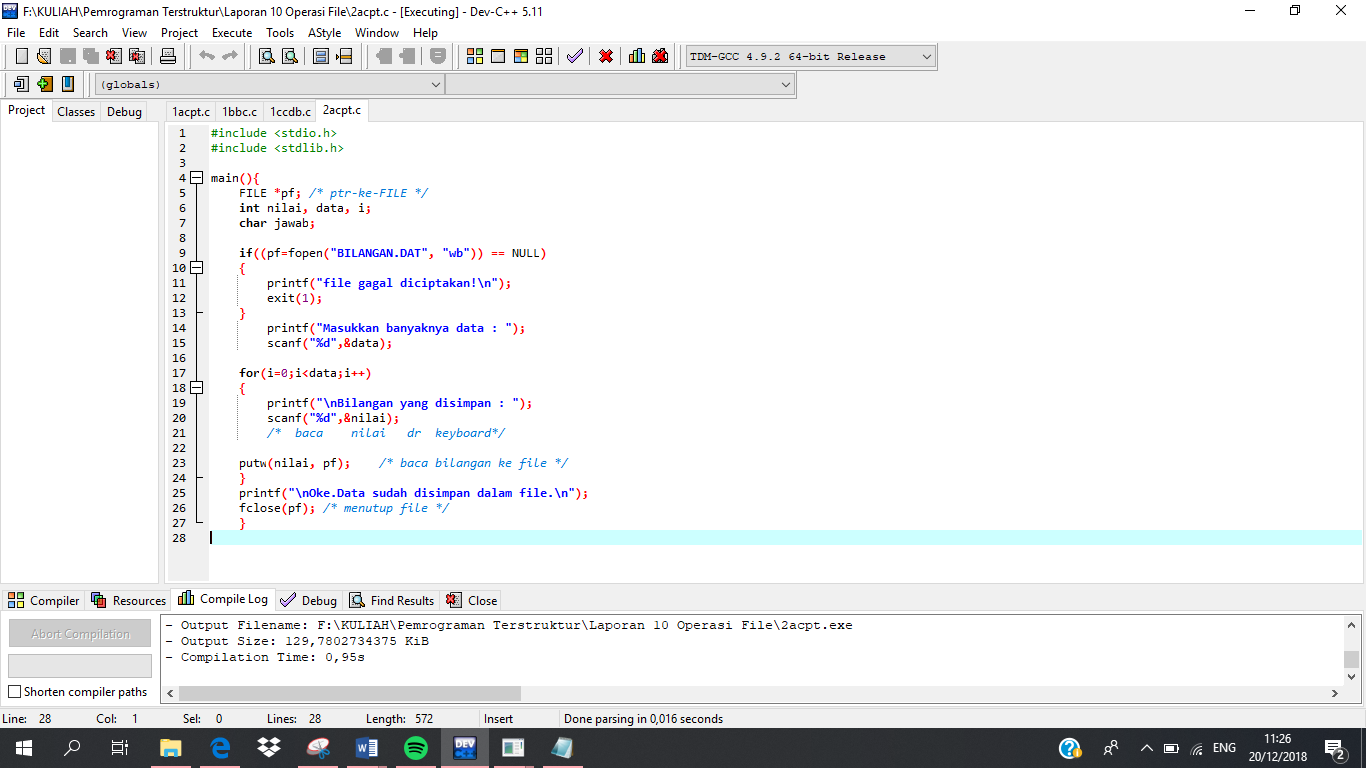
/\* baca nilai dr keyboard\*/

putw(nilai, pf); /\* baca bilangan ke file \*/

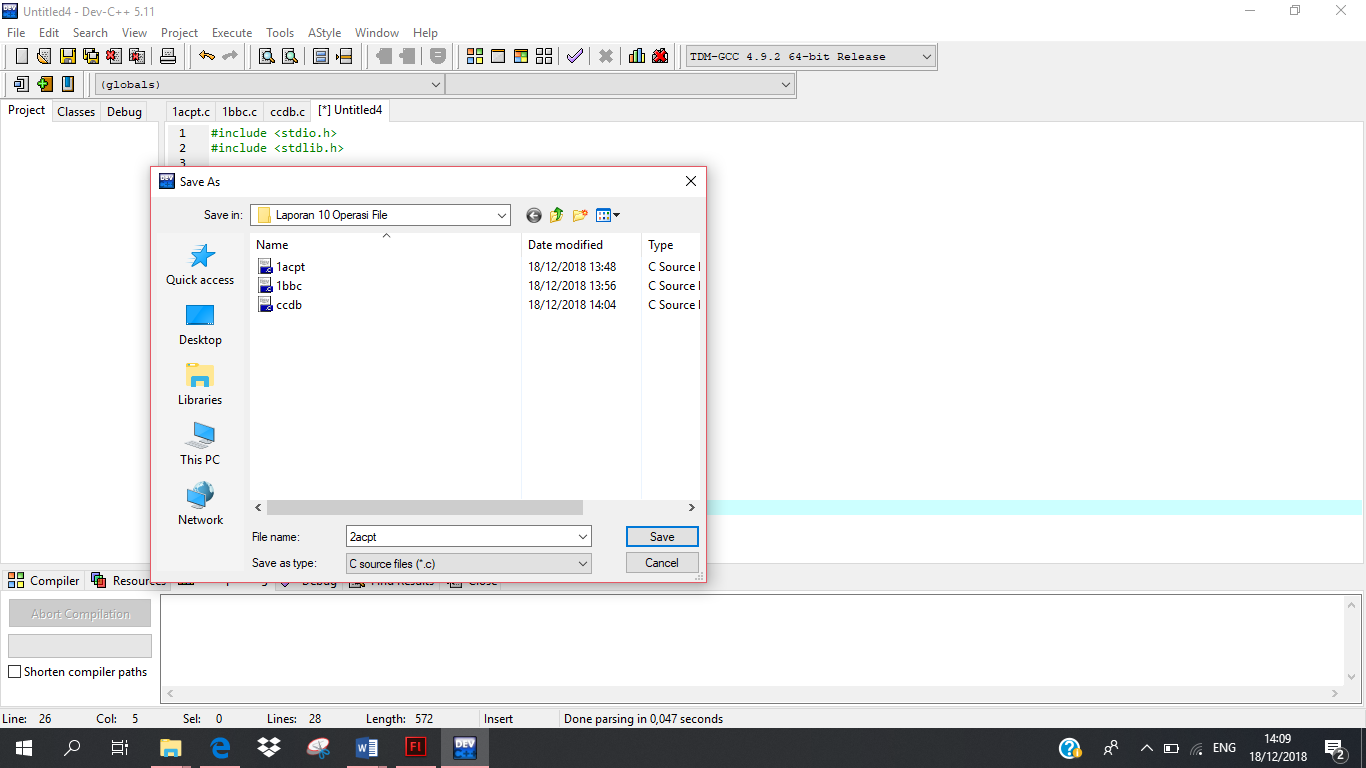
}

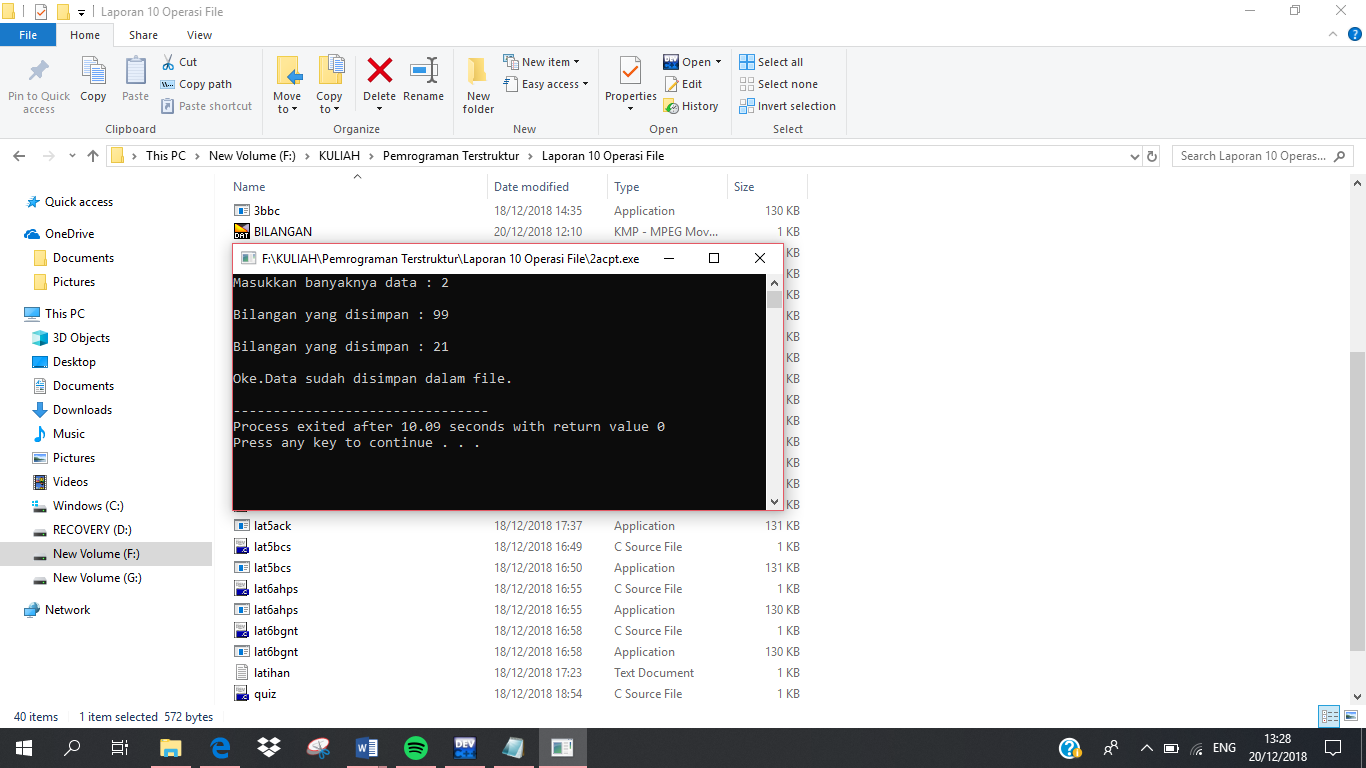
printf("\nOke.Data sudah disimpan dalam file.\n");

fclose(pf); /\* menutup file \*/

 }

1. Alt+F, pilih save ketik nama 2acpt.c

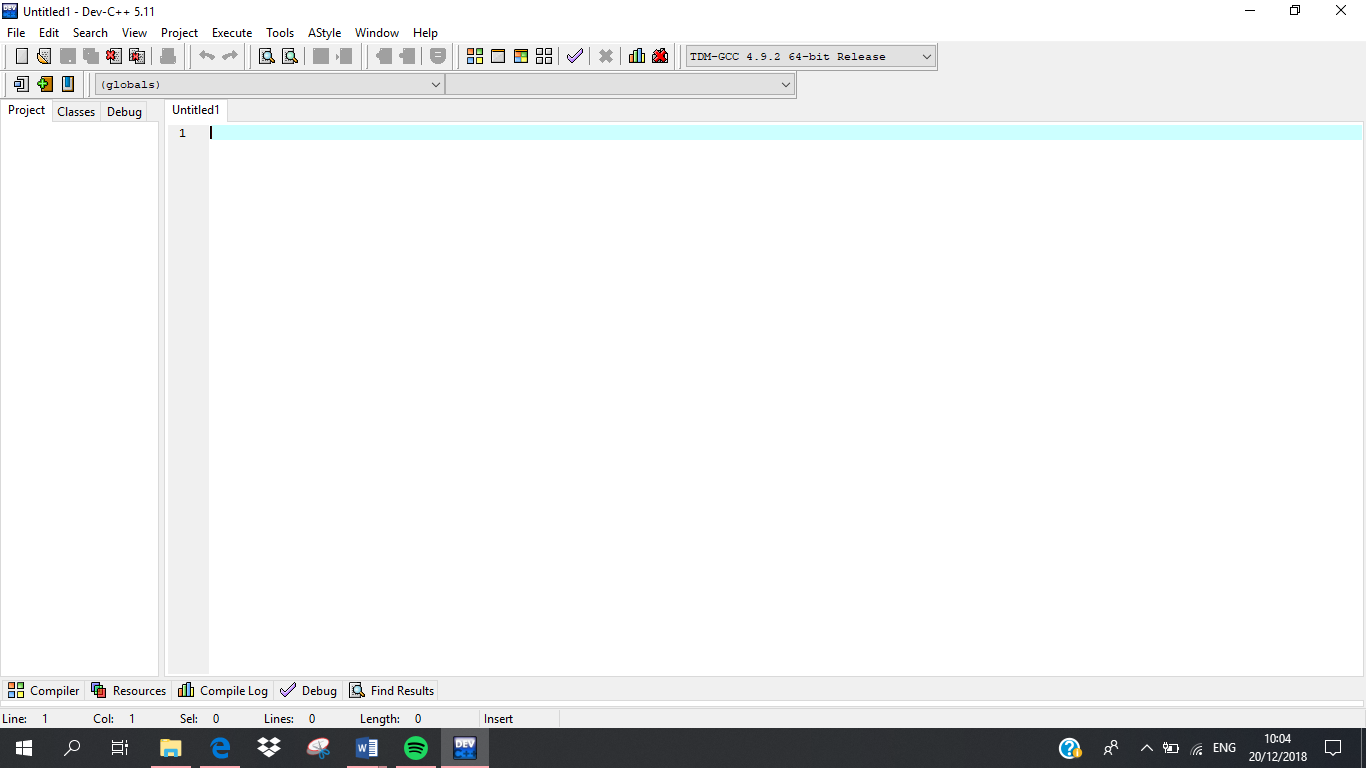


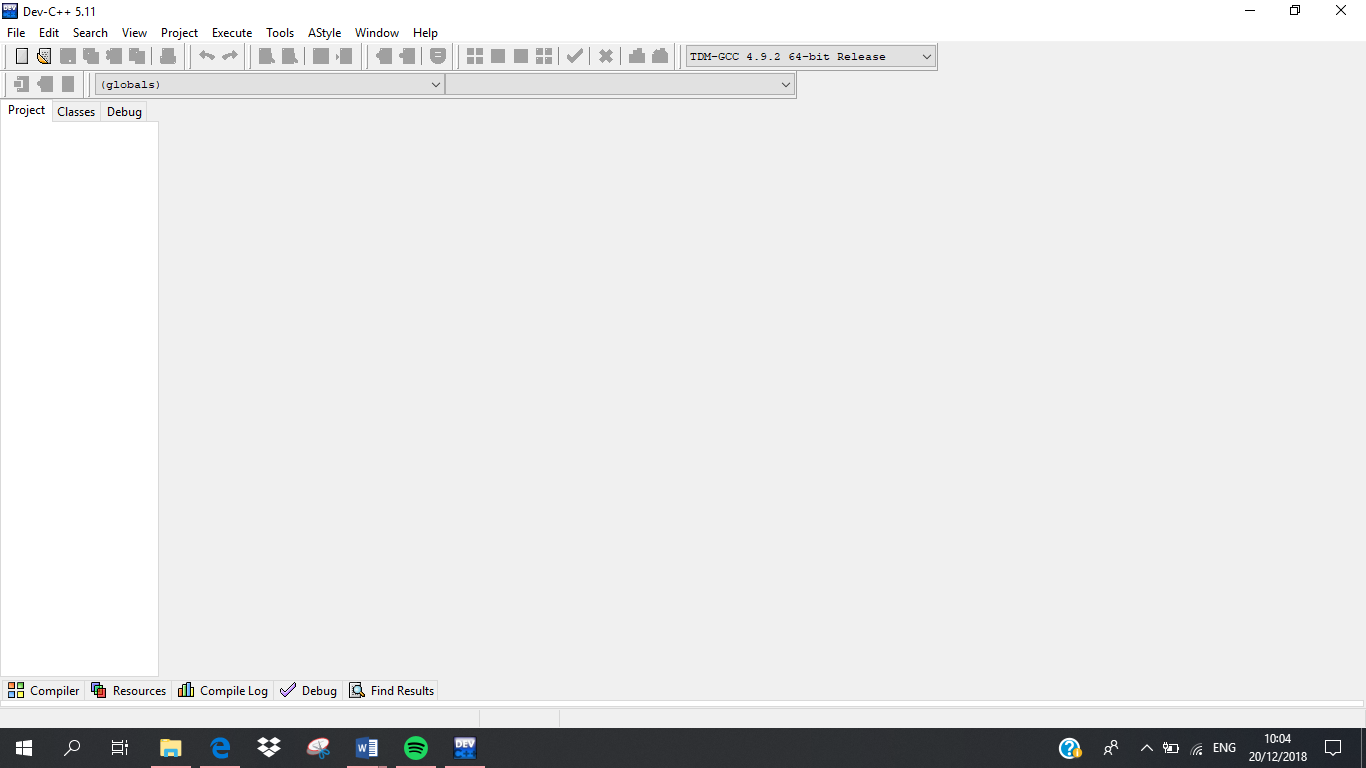
1. Alt+O, pilih directories
2. Alt+R
3. Hasil Run Program

###### Latihan 2b : program membaca file

1. Jalankan progam Dev C++
2. Alt + F, pilih New
3. Ketik :
4. Alt+F, pilih save ketik nama 2bbc.c
5. Alt+R
6. Hasil Run Program

#### Latihan 3 Operasi Tulis & Baca File Per-Blok

1. Latihan 3a : program menciptakan file
2. Jalankan progam Dev C++
3. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<stdlib.h>

main()

{

FILE \*f\_struktur;

char jawaban;

int sudah\_benar;

struct

{

char judul[20];

char pengarang[20];

int jumlah;

} buku; /\* variabel buku bertipe struktur \*/

if((f\_struktur = fopen("DAFBUK.DAT","wb")) == NULL)

{

printf("File tidak dapat diciptakan!\n");

exit(1);

}

do

{

fflush(stdin); /\* Hapus isi penampung keyboard \*/

printf("Judul buku :");

gets(buku.judul);

printf("Nama pengarang :");

gets(buku.pengarang); printf("Jumlah buku :");

scanf("%d", &buku.jumlah);

fflush(stdin); /\* Hapus isi penampung keyboard \*/

/\* Rekam sebuah data betipe struktur \*/

fwrite(&buku,sizeof(buku), 1, f\_struktur);

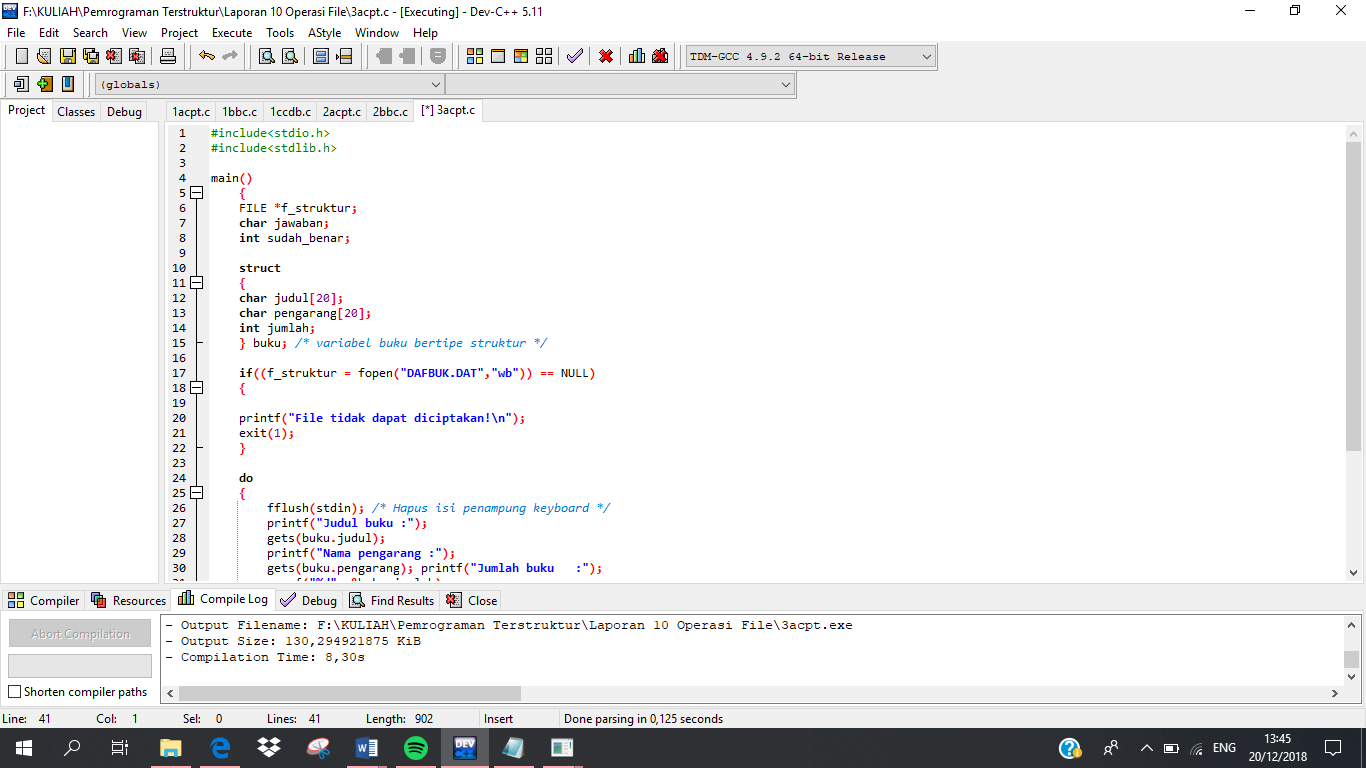
printf("\nMau merekam data lagi[Y/T]?");

jawaban = getchar();

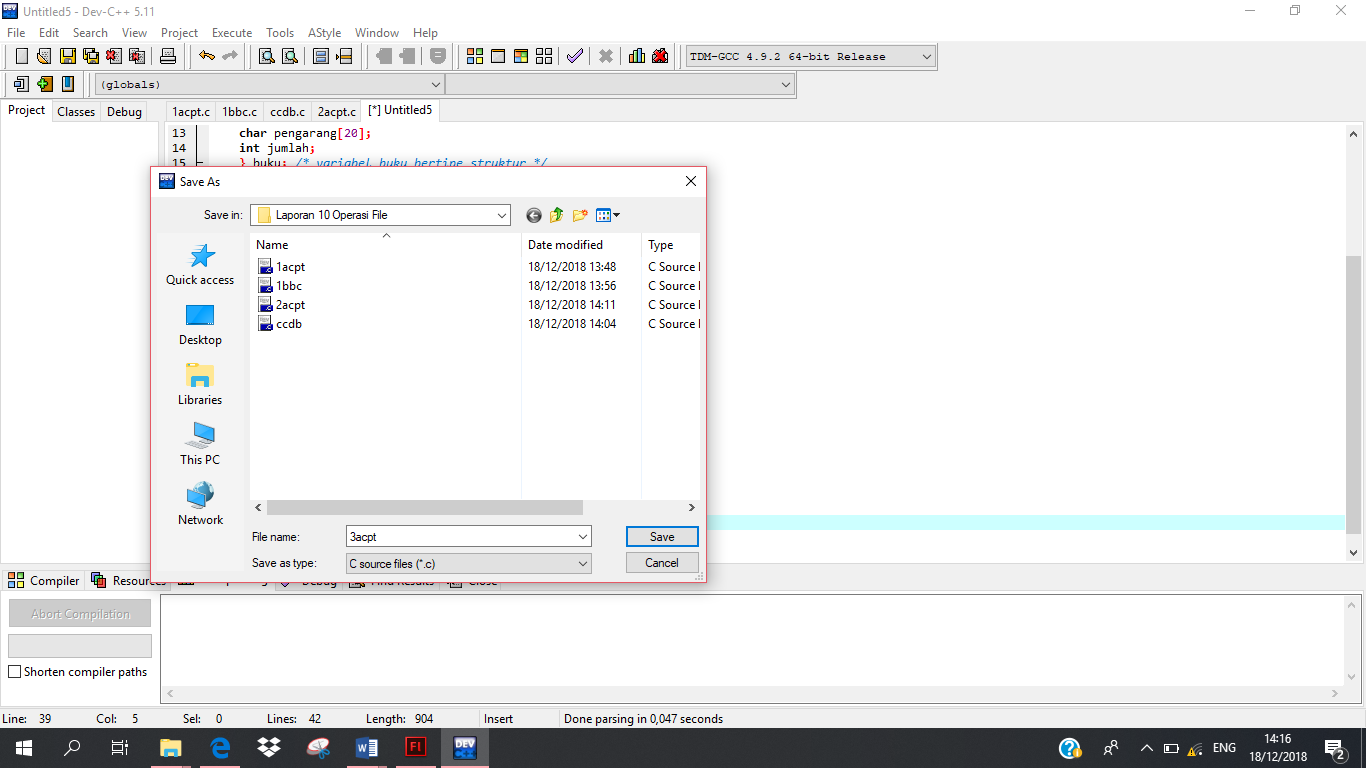
printf("\n");

}while(jawaban == 'Y'|| jawaban == 'y');

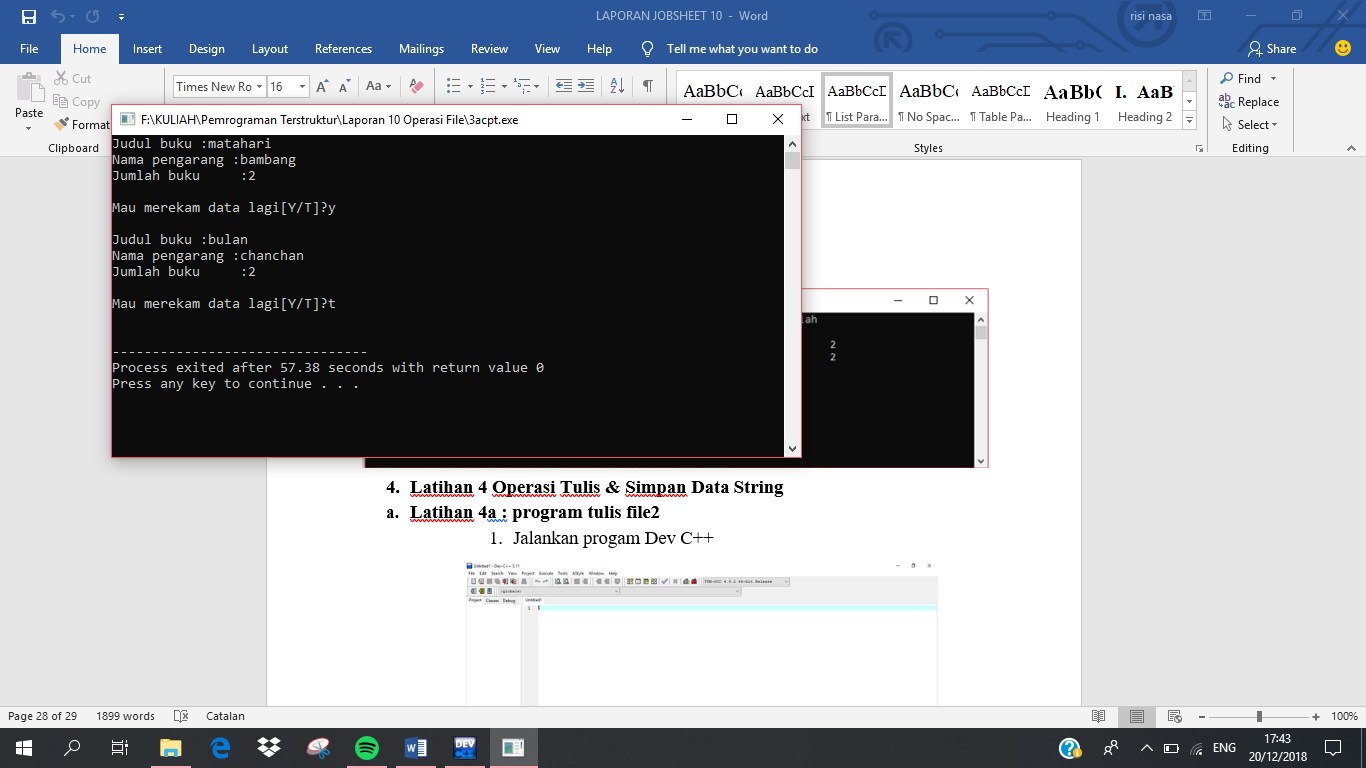
fclose(f\_struktur); /\* Tutup file \*/

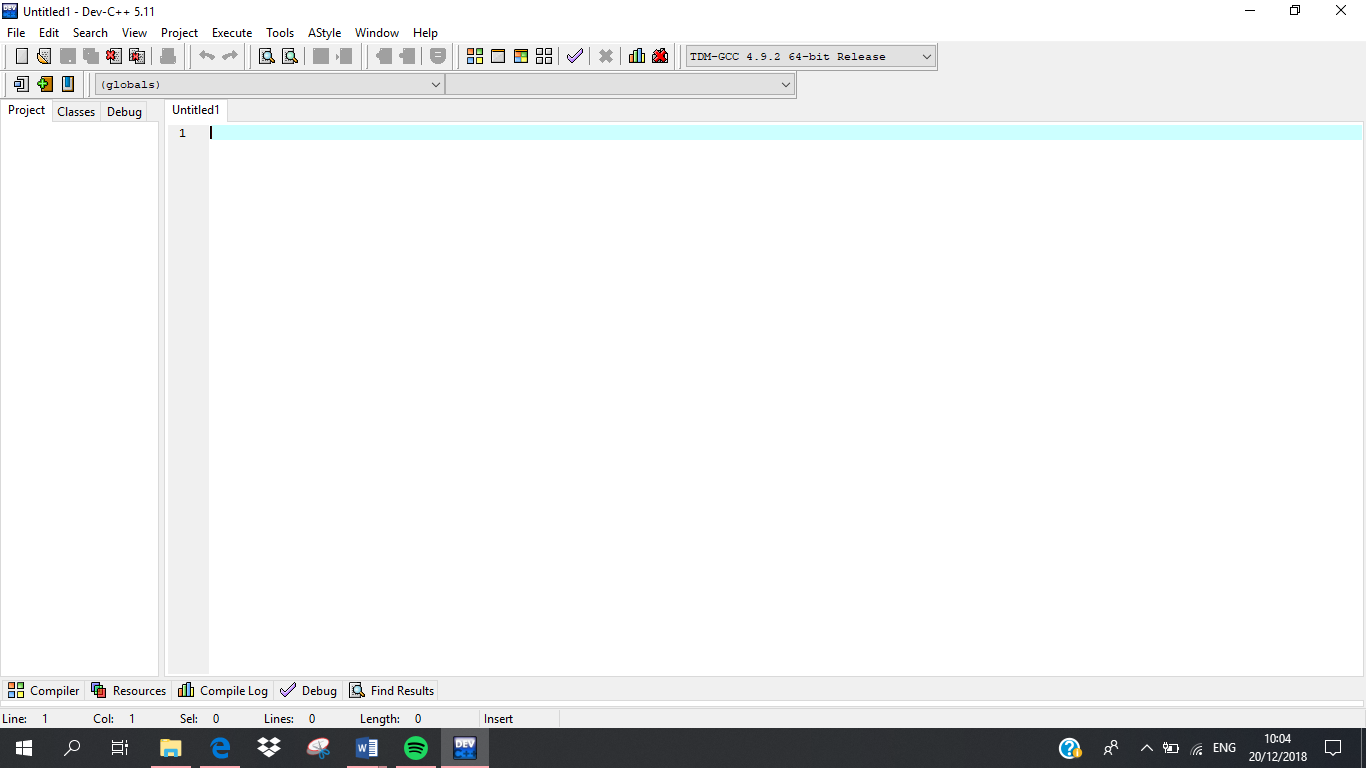
 }

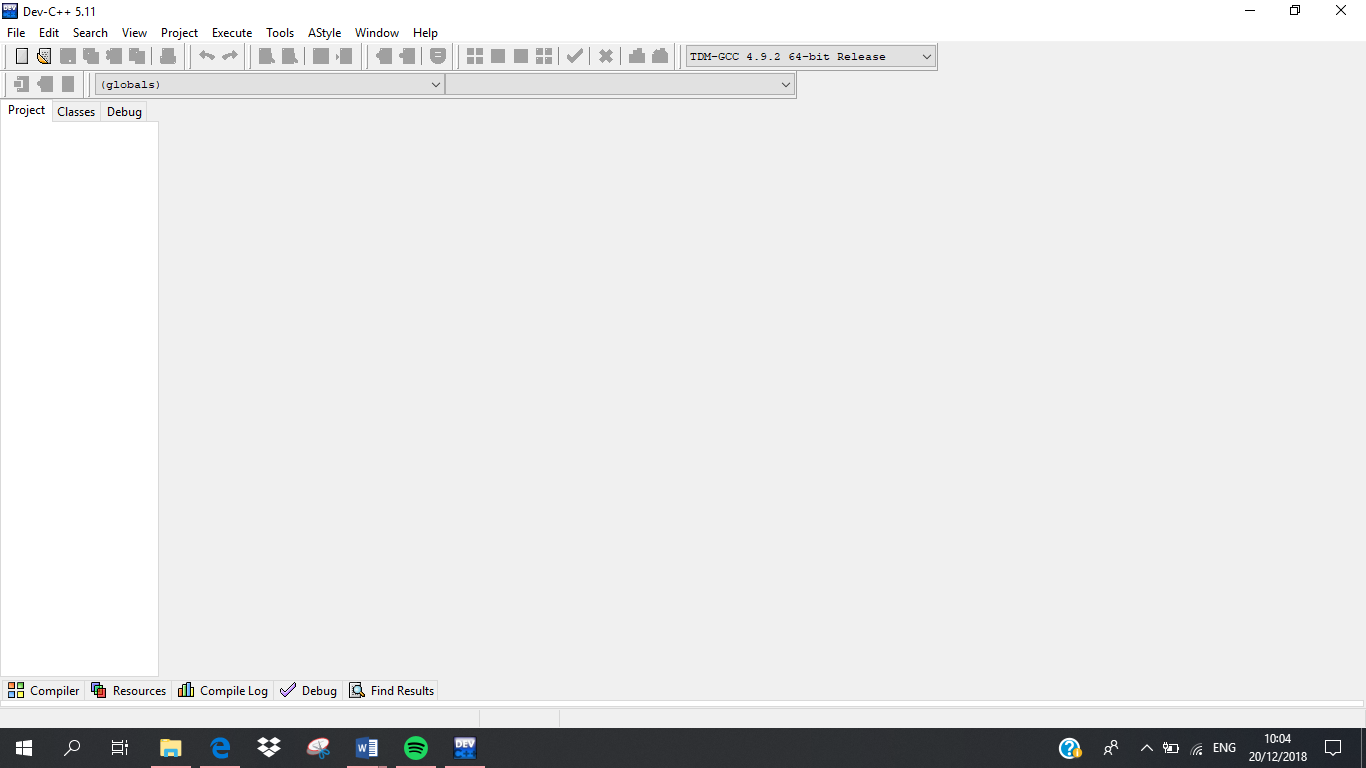
1. Alt+F, pilih save ketik nama 3acpt.c



1. Alt+R
2. Hasil Run Program



1. Latihan 3b : program baca file
2. Jalankan progam Dev C
3. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<stdlib.h>

main(){

FILE \*f\_struktur;

int i = 1;

struct

{

char judul[20];

char pengarang[20];

int jumlah;

} buku; /\* variabel buku bertipe struktur \*/

if((f\_struktur = fopen("DAFBUK.DAT","rb")) != NULL)

{

printf("%2s. %-30s %-30s %s\n\n", "No", "Judul Buku", "Nama Pengarang", "Jumlah");

/\* diulang selama masih ada record yg terbaca dlm file\*/

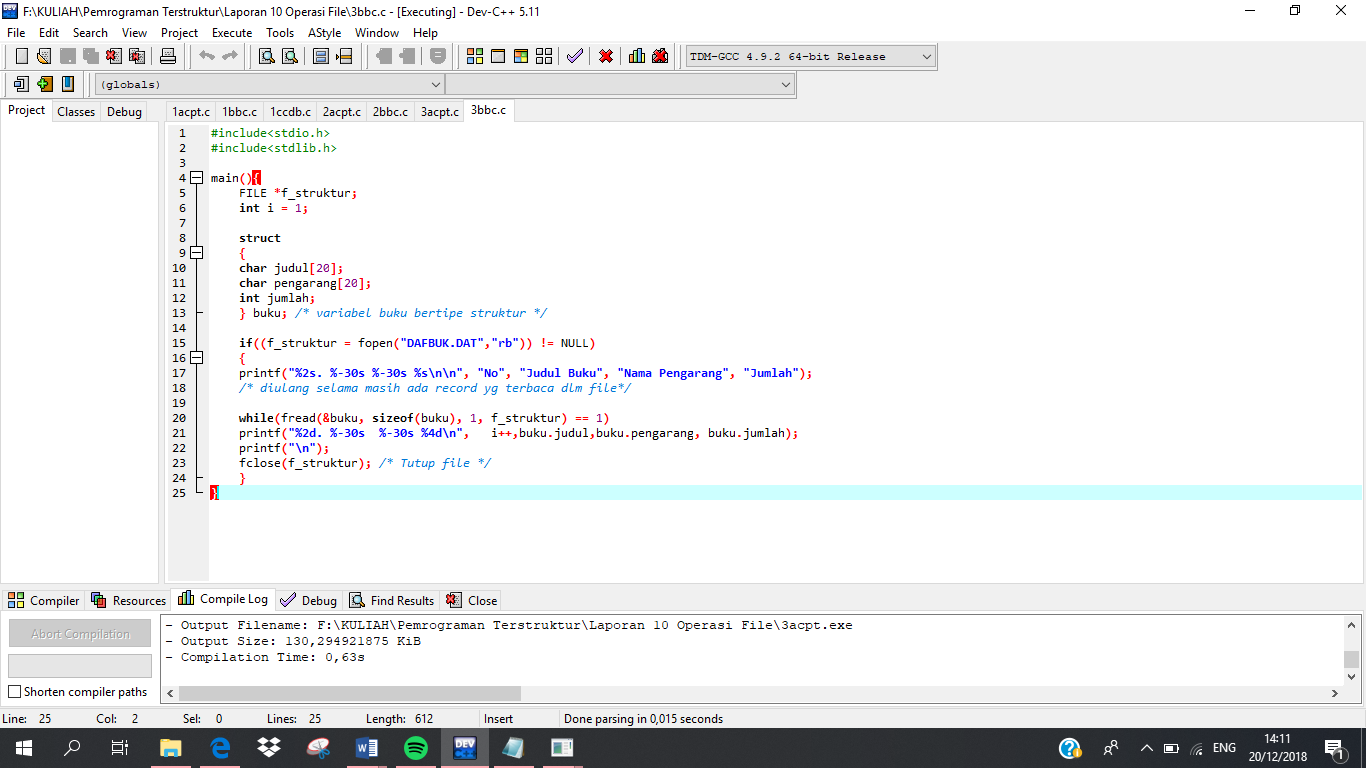
while(fread(&buku, sizeof(buku), 1, f\_struktur) == 1)

printf("%2d. %-30s %-30s %4d\n", i++,buku.judul,buku.pengarang, buku.jumlah);

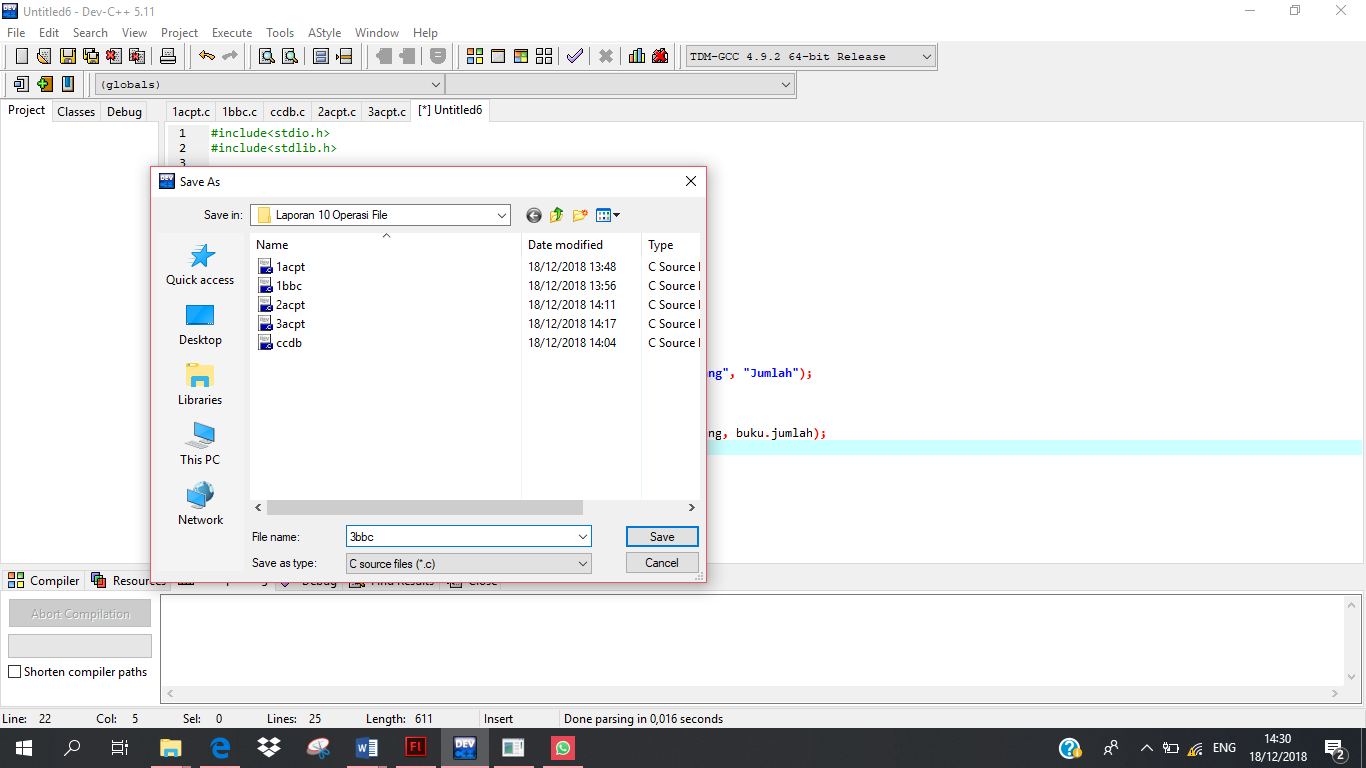
printf("\n");

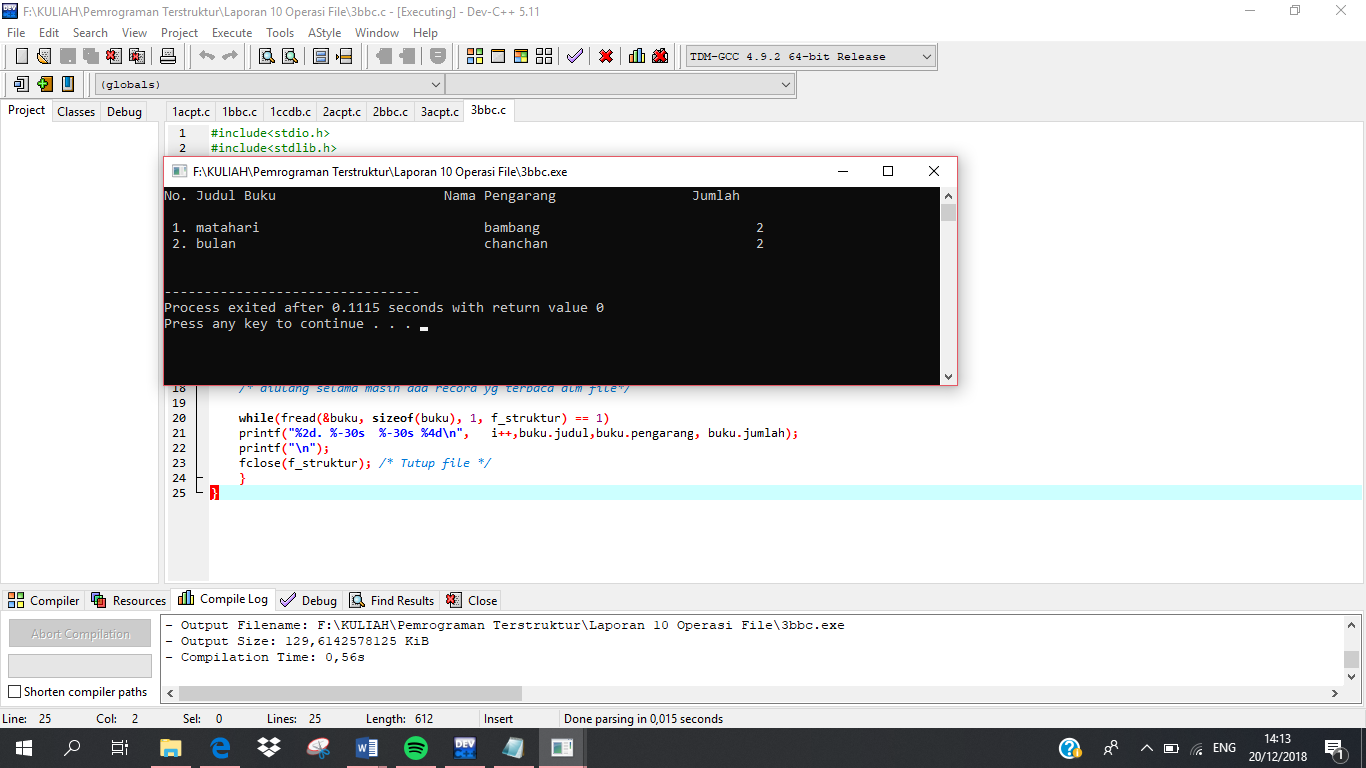
fclose(f\_struktur); /\* Tutup file \*/

}

}

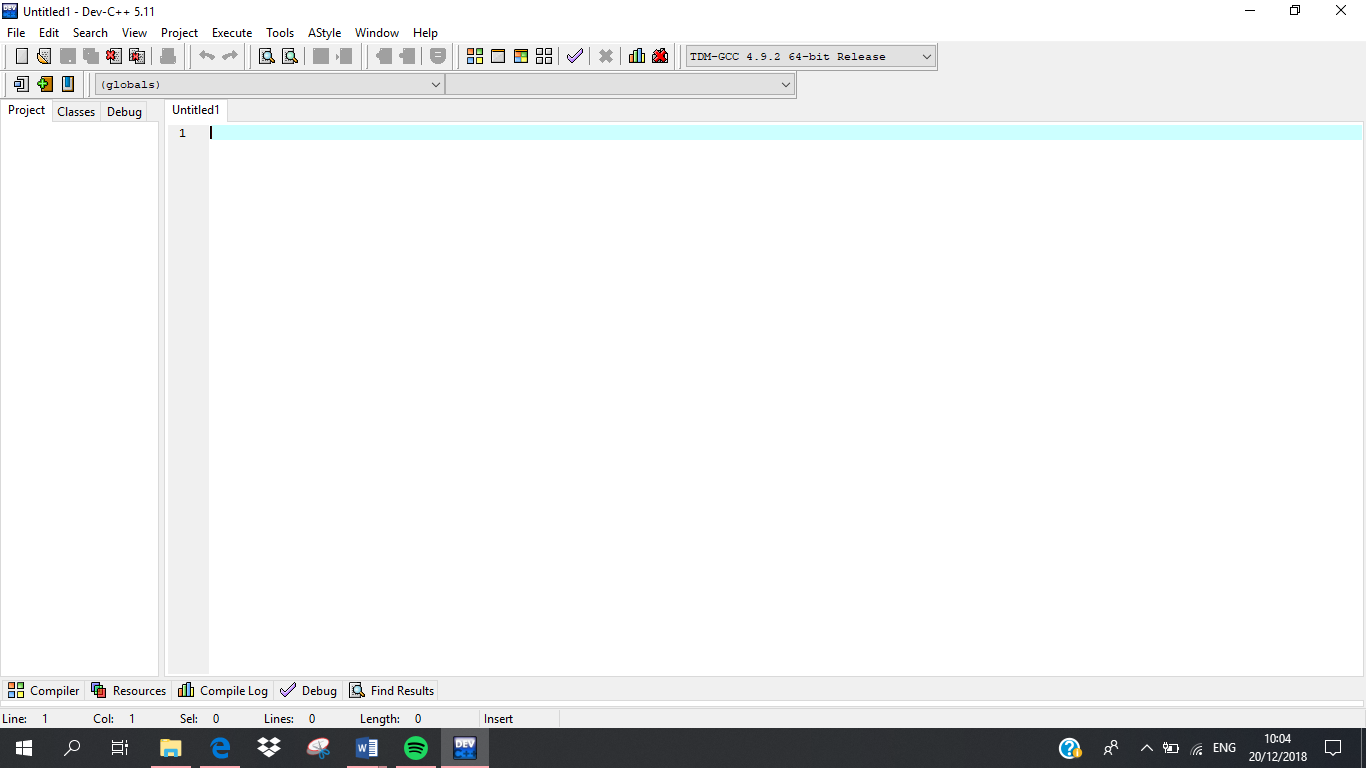
1. Alt+F, pilih save ketik nama 3bbc.c



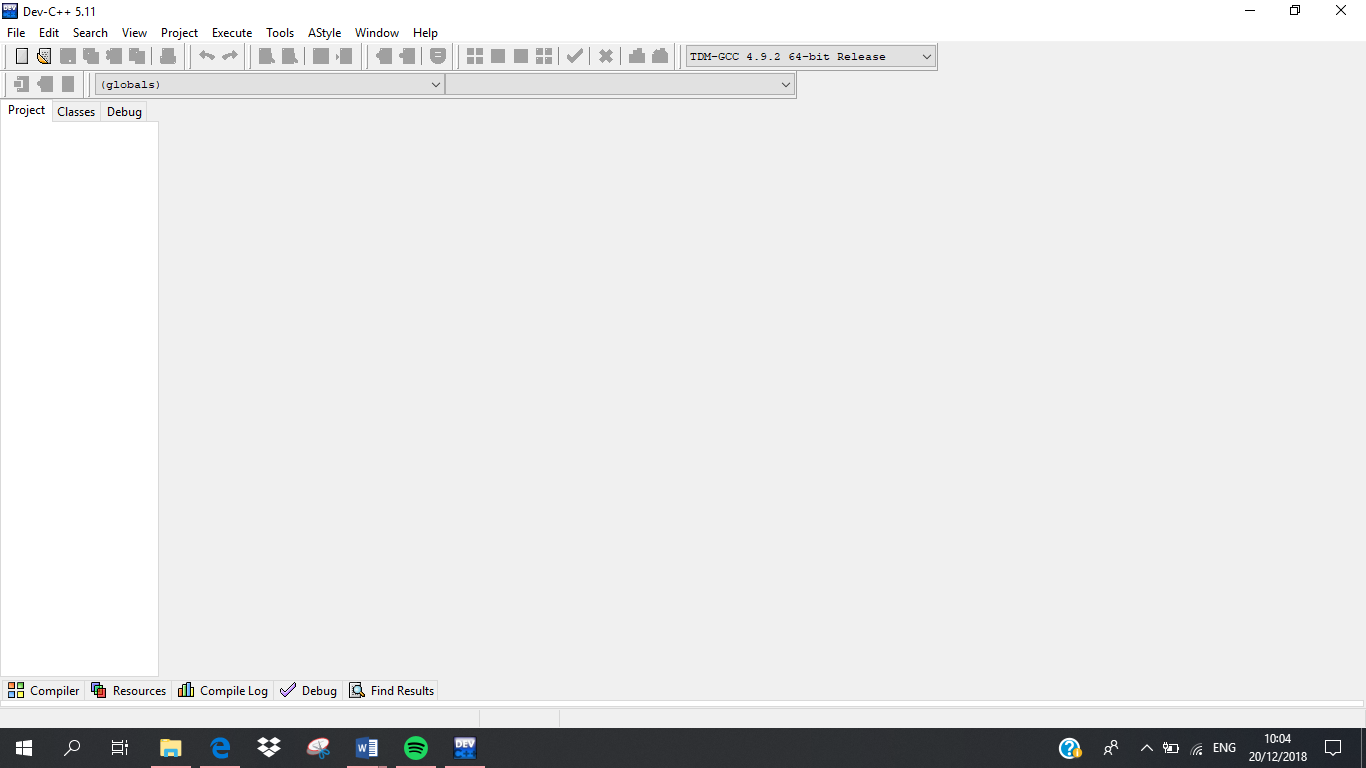
1. Alt+R
2. Hasil Run Program

#### Latihan 4 Operasi Tulis & Simpan Data String

1. Latihan 4a : program tulis file2
2. Jalankan progam Dev C++



1. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<stdlib.h>

main(){

FILE \*pf; /\* pointer-ke-FILE\*/

int data,i;

char nama[40];

system("cls");

if((pf=fopen("latihan.txt","w"))==NULL)

{

printf("File gagal diciptakan...!\n");

exit(1);

}

printf("Masukkan banyaknya data :");

scanf("%d", &data);

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

{

printf("\nNama ke-%d : ", i);

fflush(stdin);

gets(nama);

strcat(nama, "\n");

fputs(nama,pf);

}

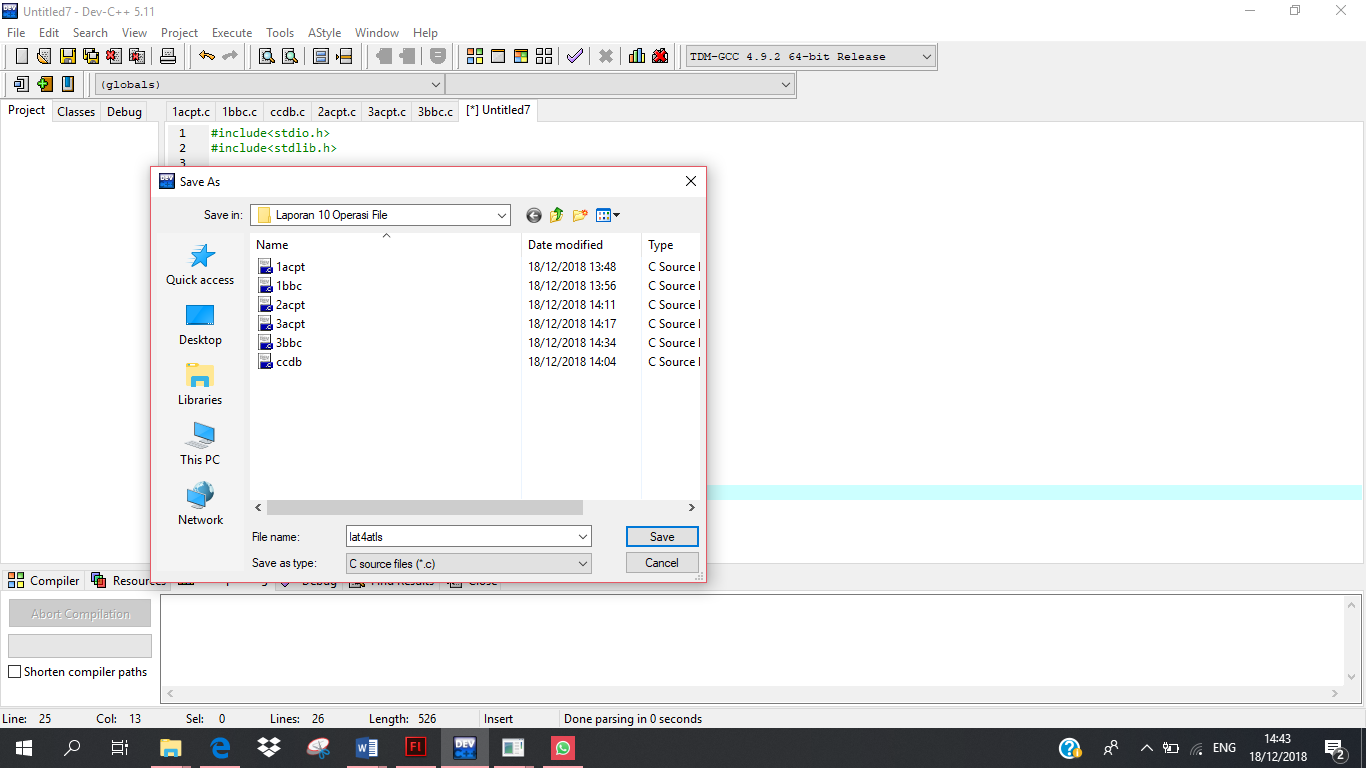
printf("\nOke.Data sudah disimpan dalam file,,,,,\n");

fclose(pf);

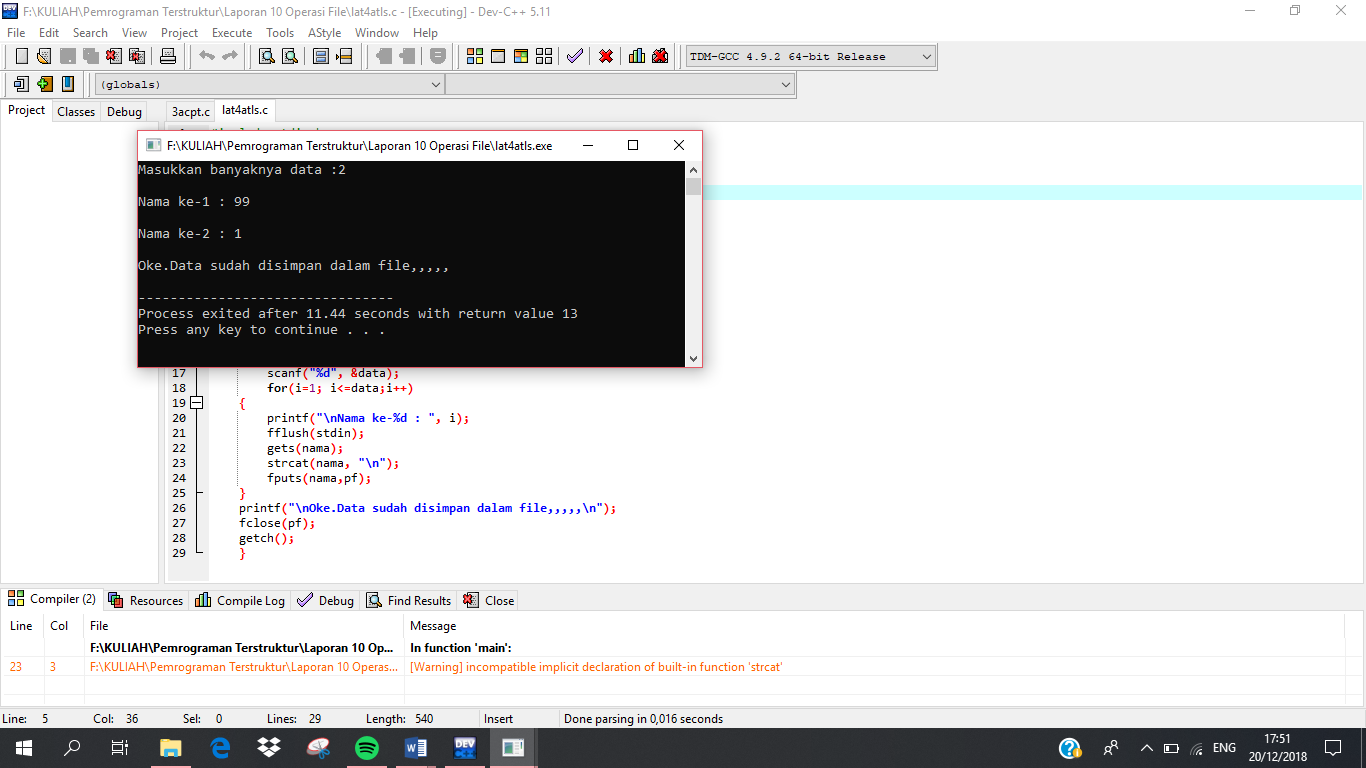
getch();

}

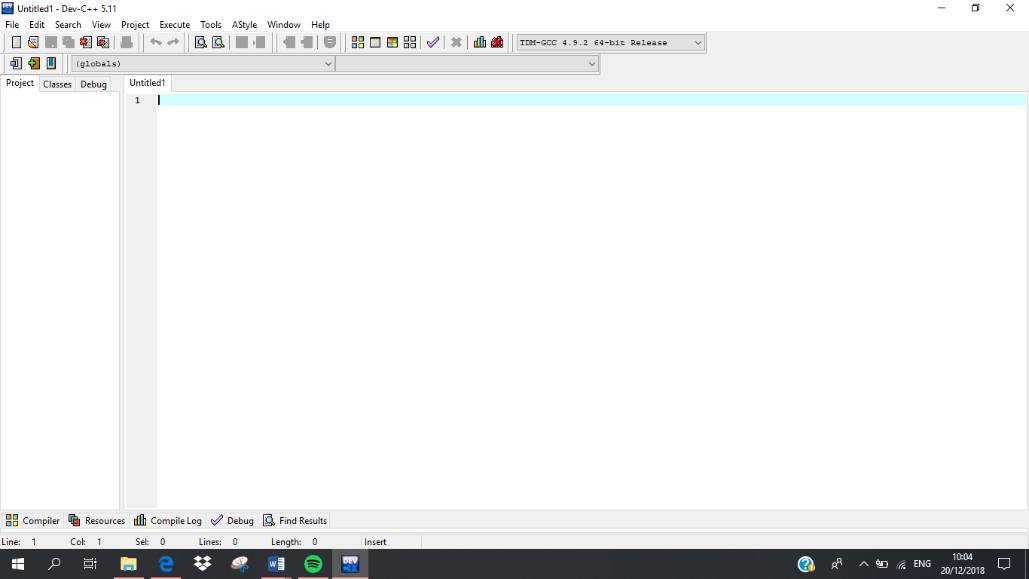
1. Alt+F, pilih save ketik nama 4atls.c



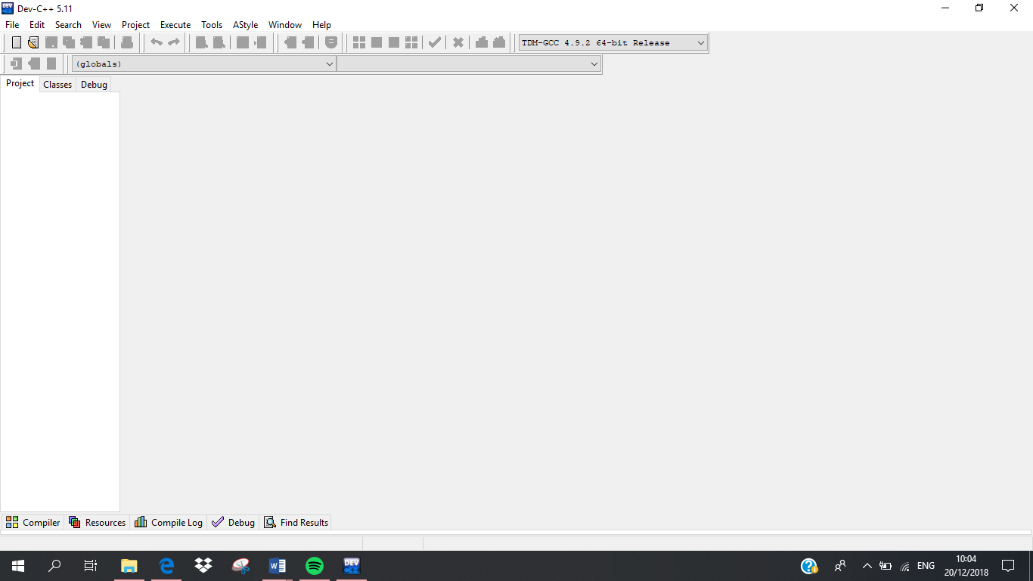
1. Alt+R
2. Hasil Run Program



1. Latihan 4b : program tulis file2
2. Jalankan progam Dev C++



1. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<conio.h>

main(){

FILE \*pf; /\*pointer-ke-FILE\*/

int data,i=1;

char nama[40];

system("cls");

if((pf=fopen("latihan.txt","w"))==NULL)

{

printf("File gagal diciptakan. !\n");

exit(1);

}

printf("\nNama ke-%d : ",i);

fflush(stdin);

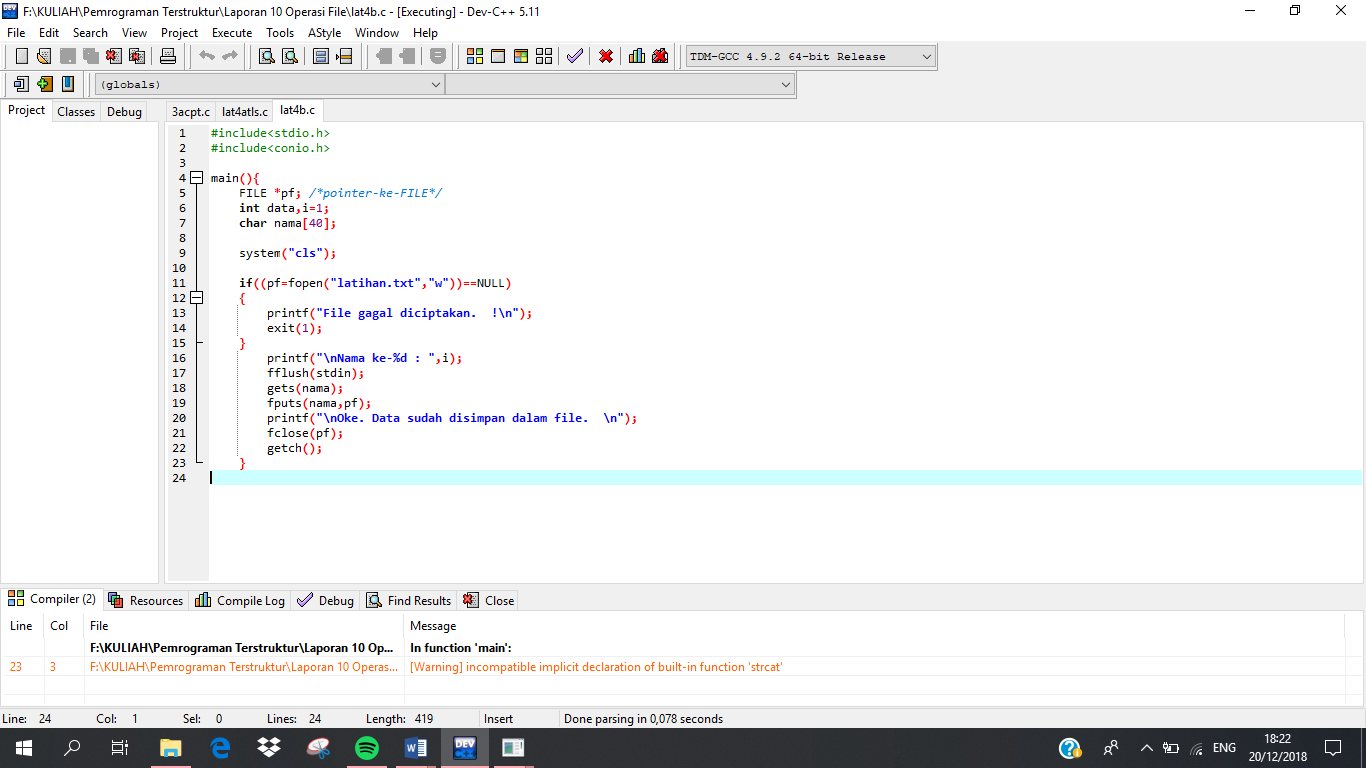
gets(nama);

fputs(nama,pf);

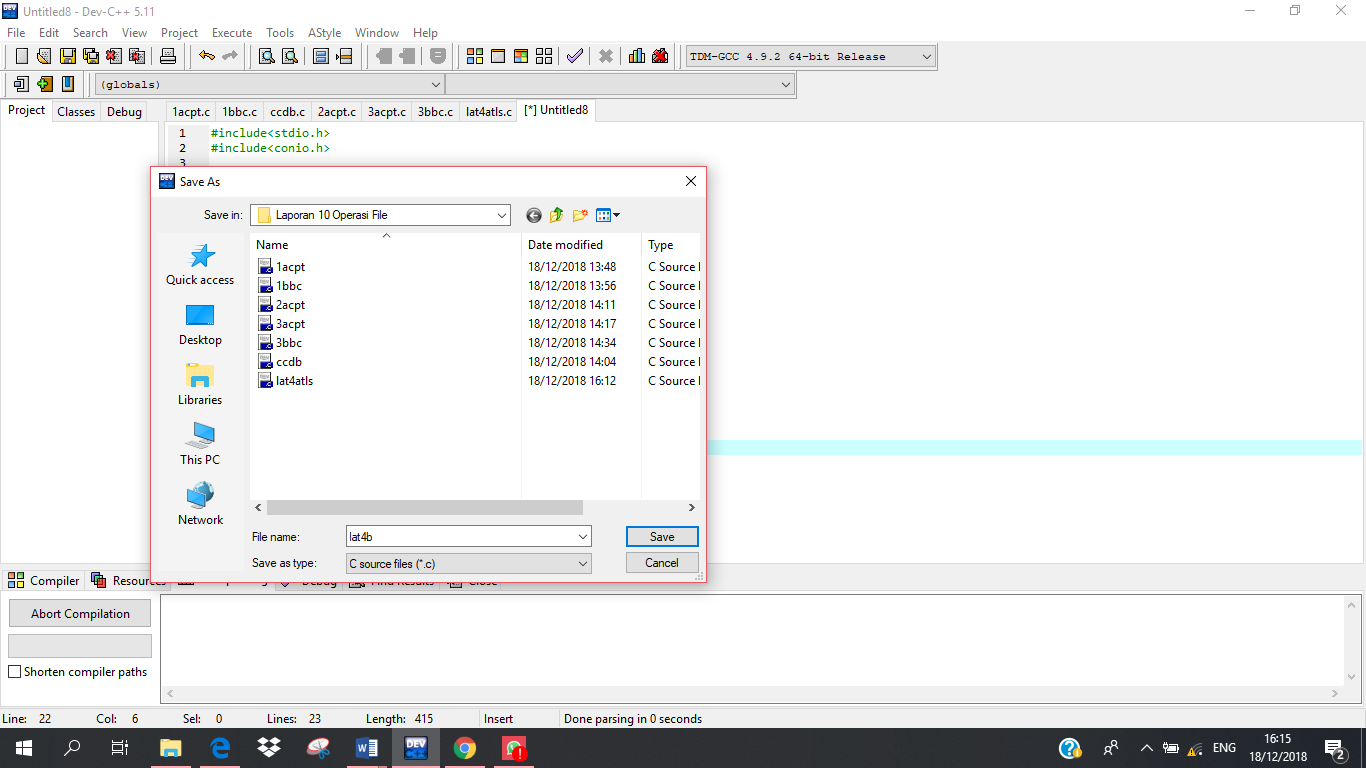
printf("\nOke. Data sudah disimpan dalam file. \n");

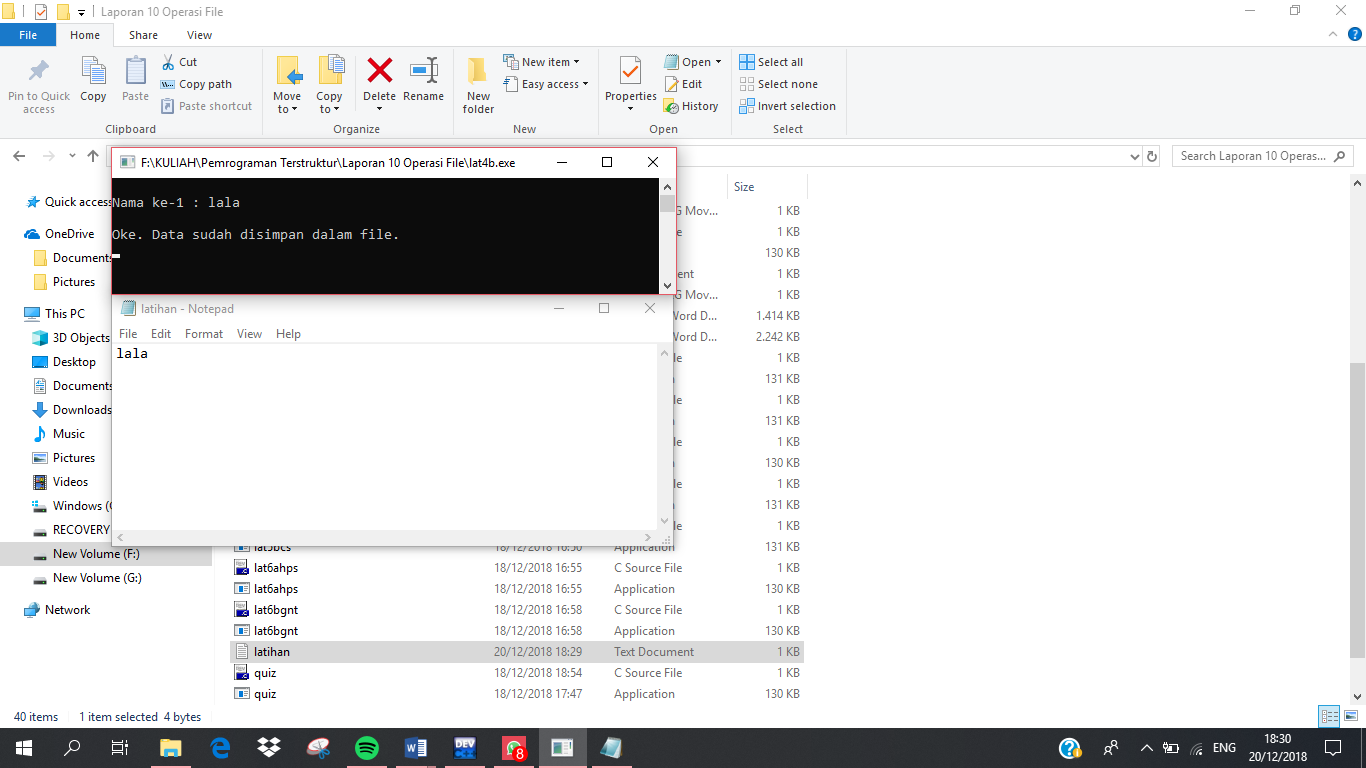
fclose(pf);

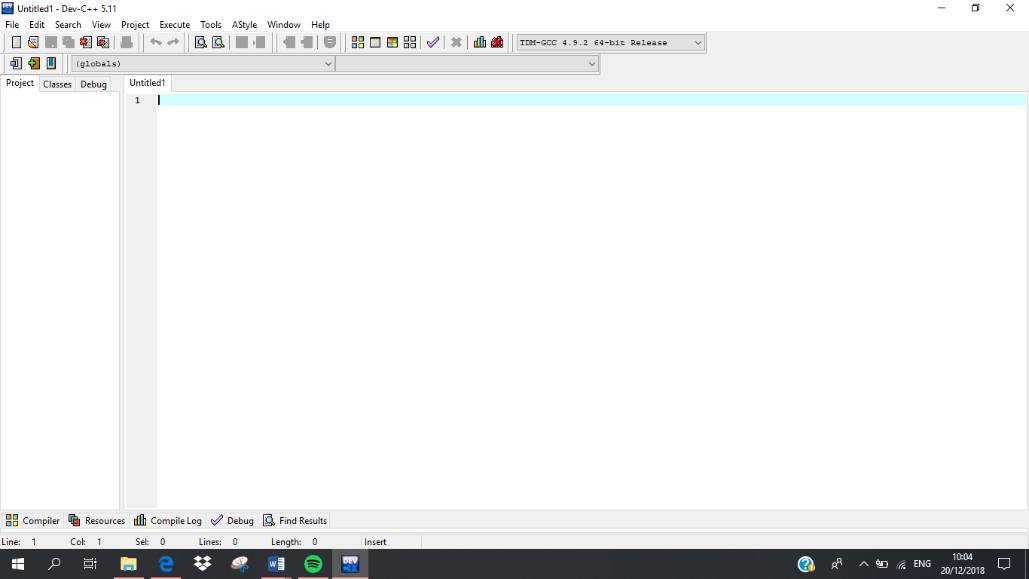
getch();

 }

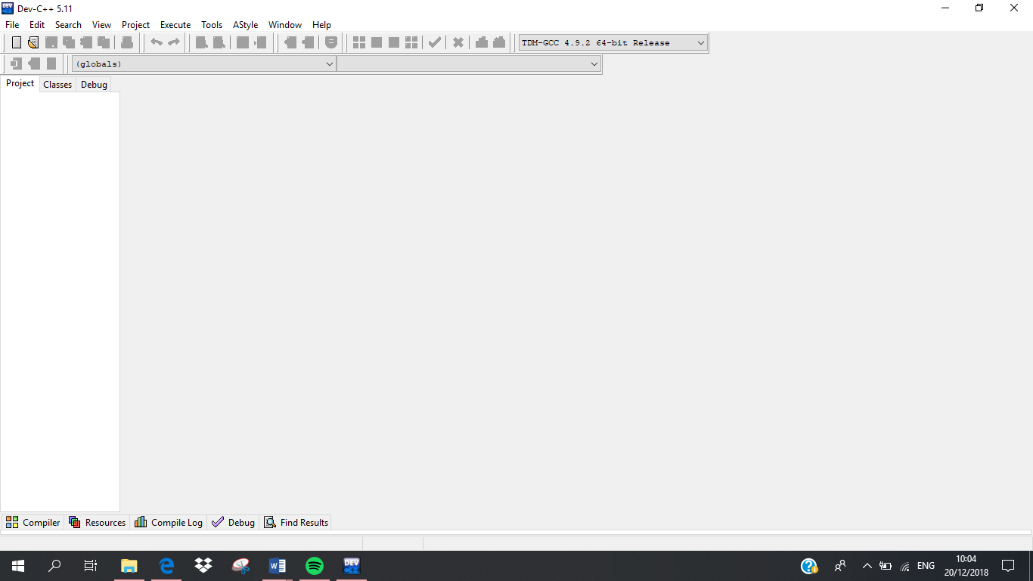
1. Alt+F, pilih save ketik nama lat4b.c



1. Alt+O, pilih directories, ganti semua dengan C:\TC, lalu esc
2. Alt+C, compile to OBJ
3. Alt+C, compile to EXE
4. Alt+R
5. Latihan 4c : program tulis file2
6. Jalankan progam Dev C++



1. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<stdlib.h>

main(){

FILE \*pf; /\*pointer-ke-FILE\*/

int data,i=1;

char nama[40];

system("cls");

if((pf=fopen("latihan.txt","r"))==NULL)

{

printf("File gagal dibuka...!\n");

exit(1);

}

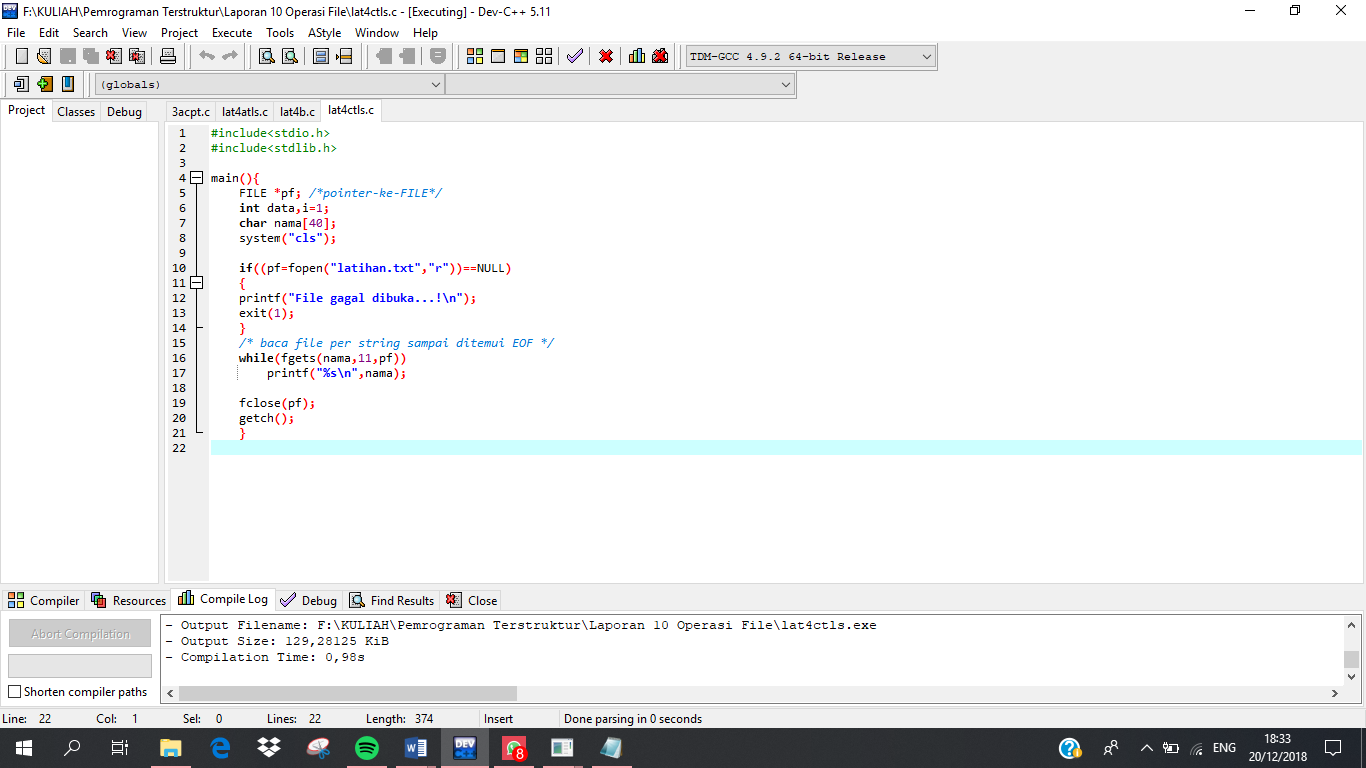
/\* baca file per string sampai ditemui EOF \*/

while(fgets(nama,11,pf))

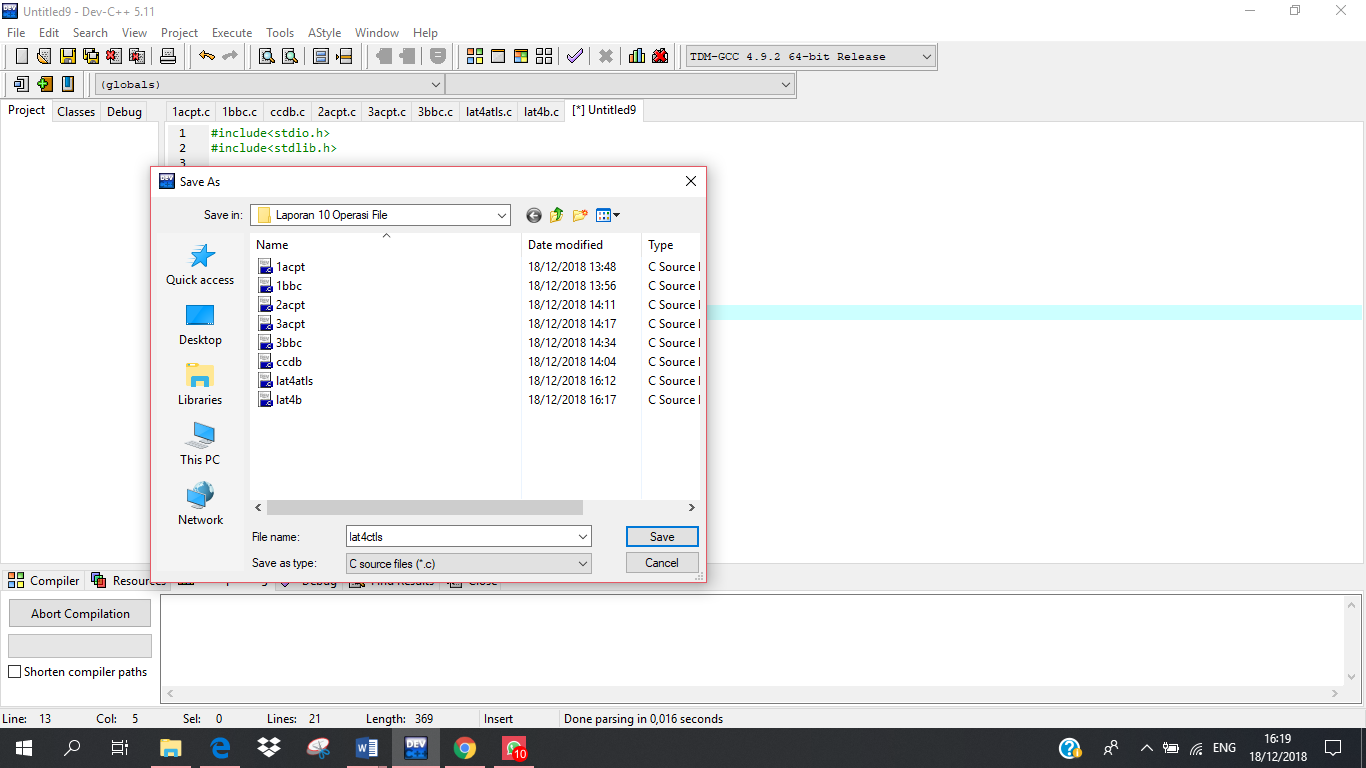
printf("%s\n",nama);

fclose(pf);

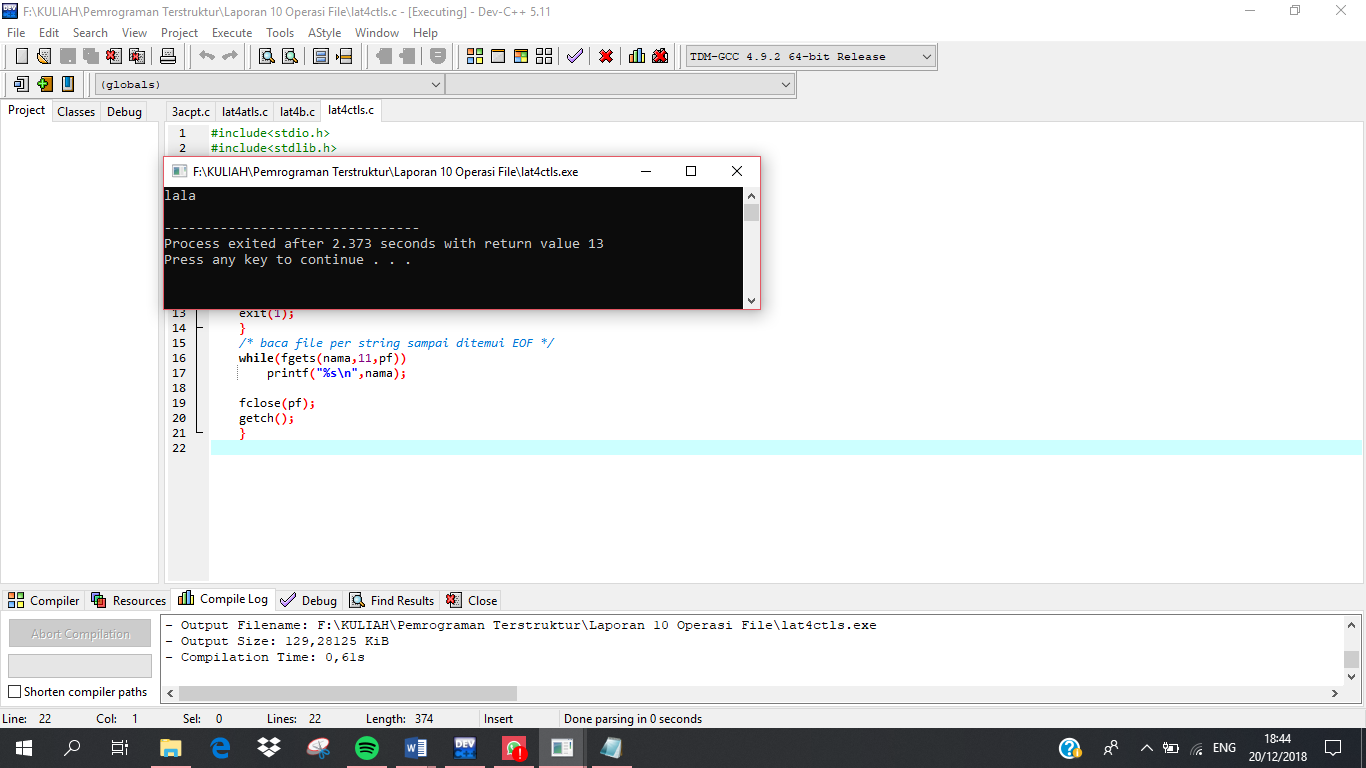
getch();

 }

1. Alt+F, pilih save ketik nama lat4ctls.c

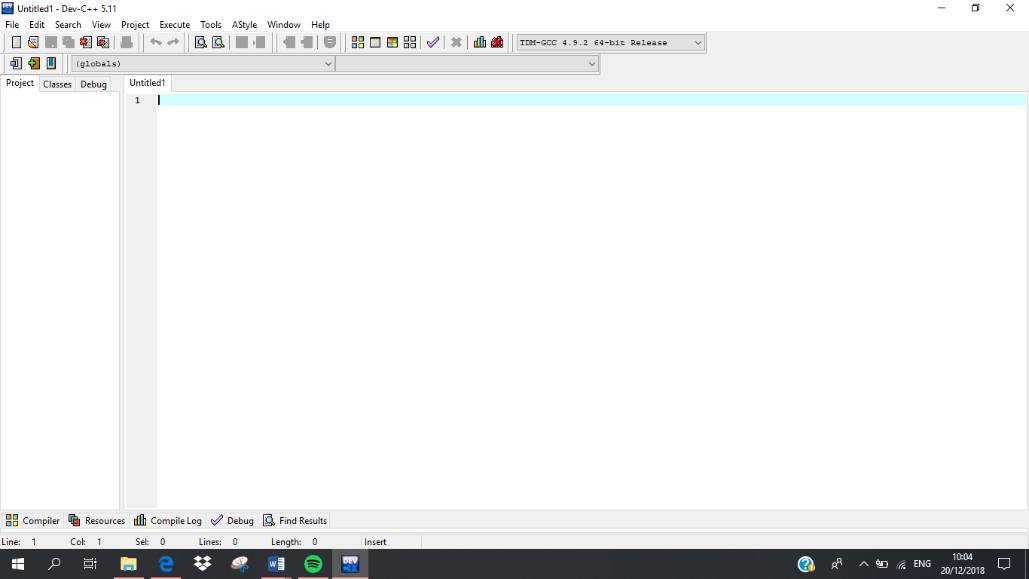


1. Alt+O, pilih directories, ganti semua dengan C:\TC, lalu esc
2. Alt+C, compile to OBJ
3. Alt+C, compile to EXE
4. Alt+R

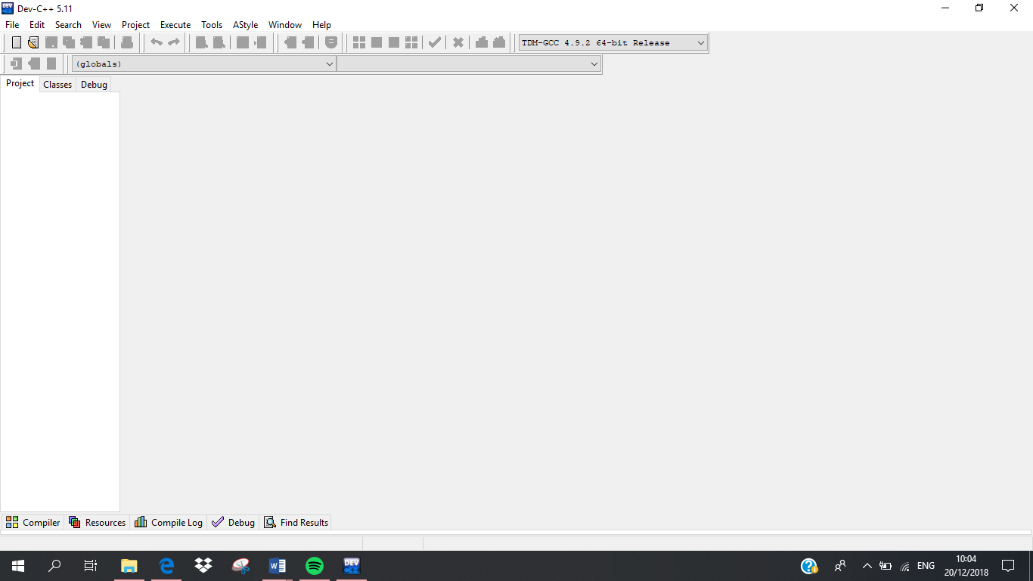


#### Latihan 5 Contoh Program fseek()

1. Latihan 5a : program mencari karakter
2. Jalankan progam Dev C++



1. Alt + F, pilih New



1. Ketik :

#include<stdlib.h>

#include<string.h>

main()

{

FILE \*pf;

char kar, jawab;

int i, no\_record;

long int offset\_byte;

if((pf =fopen("latihan.txt", "r")) == NULL)

{

printf("File tidak dapat dibuka !\n");

exit(1);

}

do

{

printf("\n Nomor record dr data yg mau ditampilkan : ");

scanf("%d", &no\_record);

offset\_byte = (no\_record-1);

fseek(pf, offset\_byte, SEEK\_SET);

kar=fgetc(pf); /\* baca kar dari file \*/

putchar(kar); /\* tampilkan ke layar \*/

printf("\nMau mencoba lagi (Y/T)? ");

jawab=getche();

} while (jawab == 'y' || jawab == 'Y');

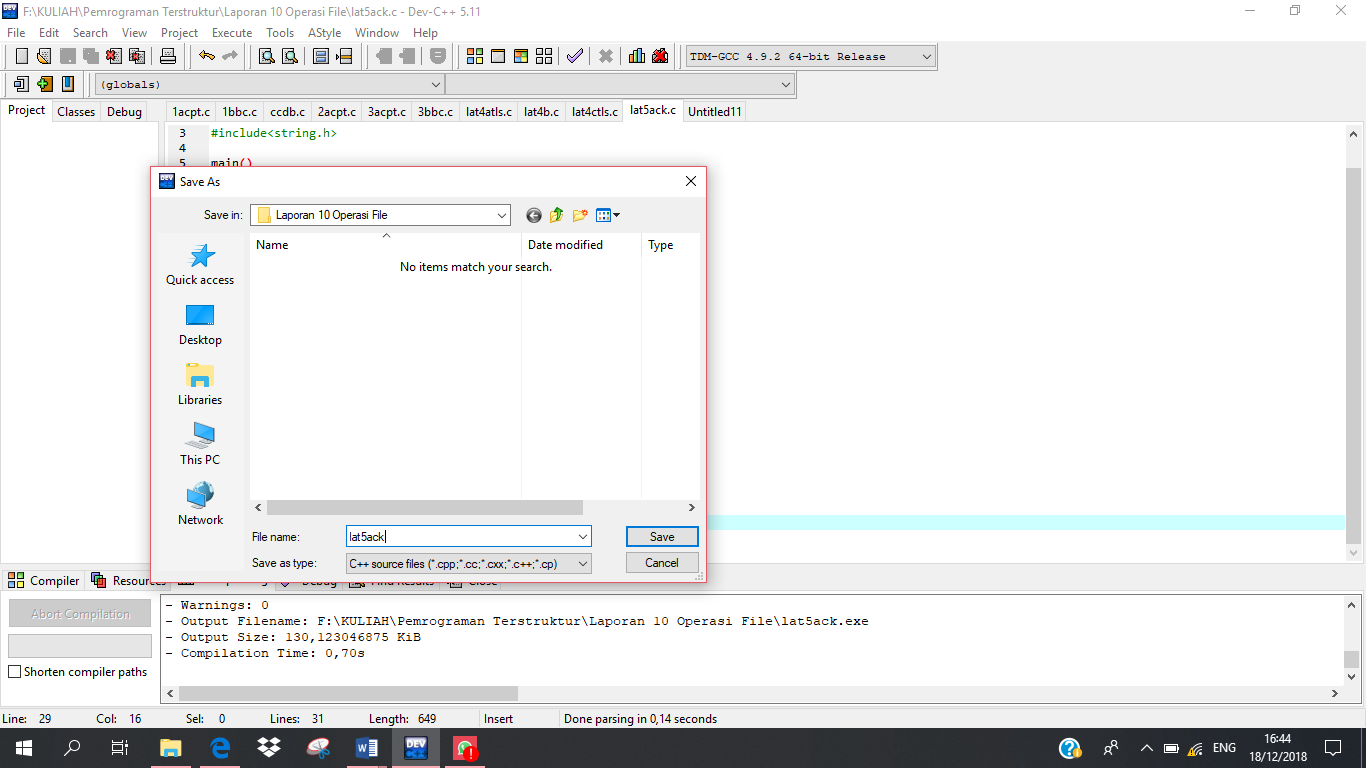
printf("\n");

fclose(pf);

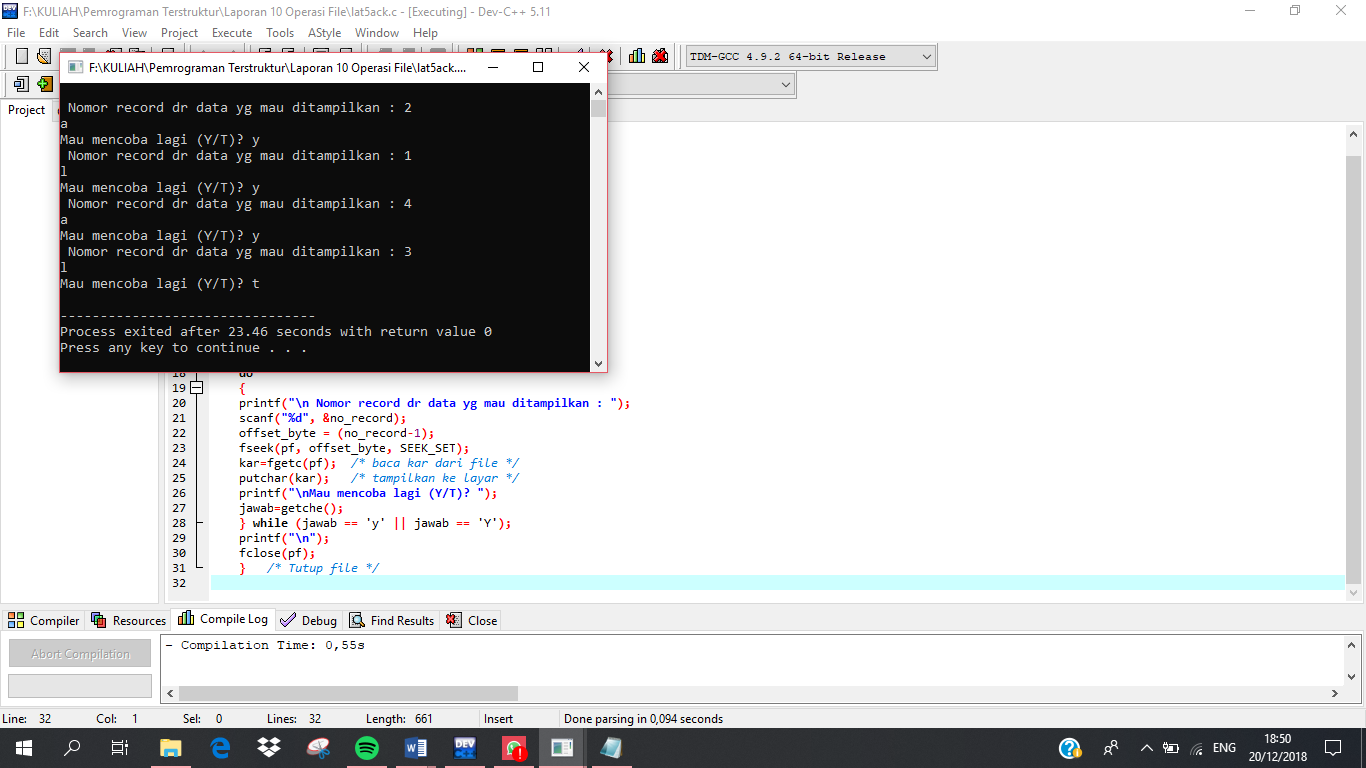
} /\* Tutup file \*/



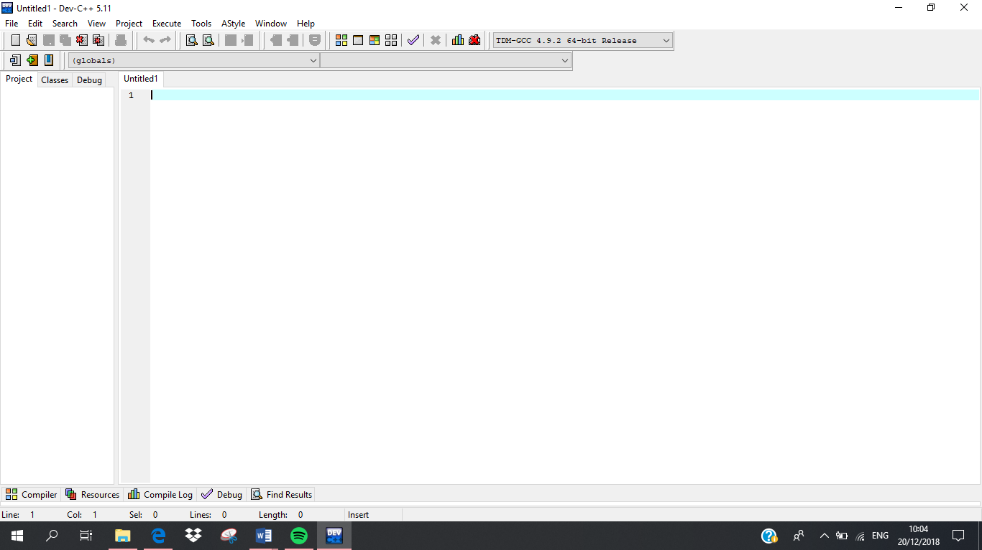
1. Alt+F, pilih save ketik nama lat5ack.c



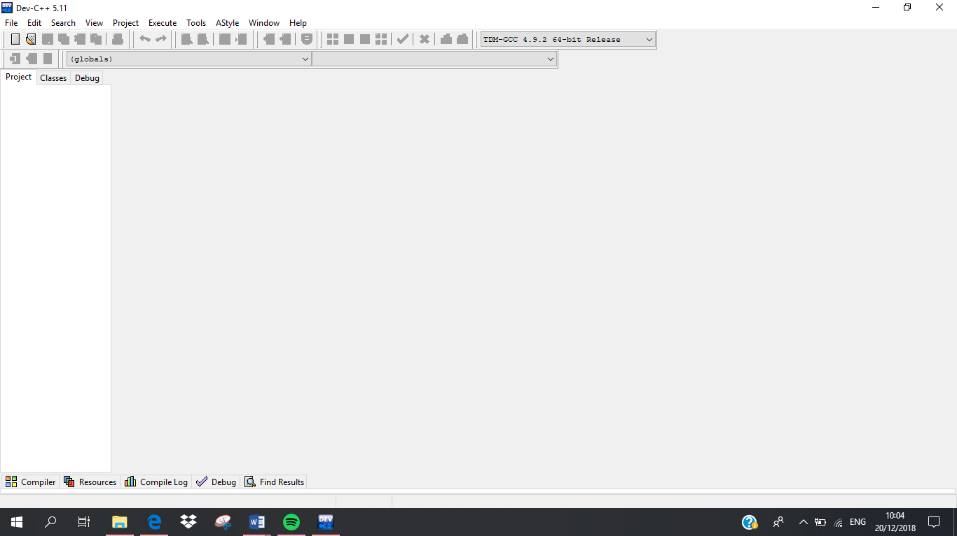
1. Alt+O, pilih directories, ganti semua dengan C:\TC, lalu esc
2. Alt+C, compile to OBJ
3. Alt+C, compile to EXE
4. Alt+R



1. Latihan 5b : program mencari string
2. Jalankan progam Dev C++



1. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

main()

{

FILE \*pf;

char jawab, nama[20];

int i, no\_record;

long int offset\_byte;

if((pf =fopen("latihan.txt", "r")) == NULL) {

printf("File tidak dapat dibuka !\n");

exit(1);

}

do

{

printf("\n Nomor record dr data yg mau ditampilkan : ");

scanf("%d", &no\_record);

offset\_byte = (no\_record-1);

fseek(pf, offset\_byte, SEEK\_SET);

printf("%s\n",fgets(nama,20,pf));

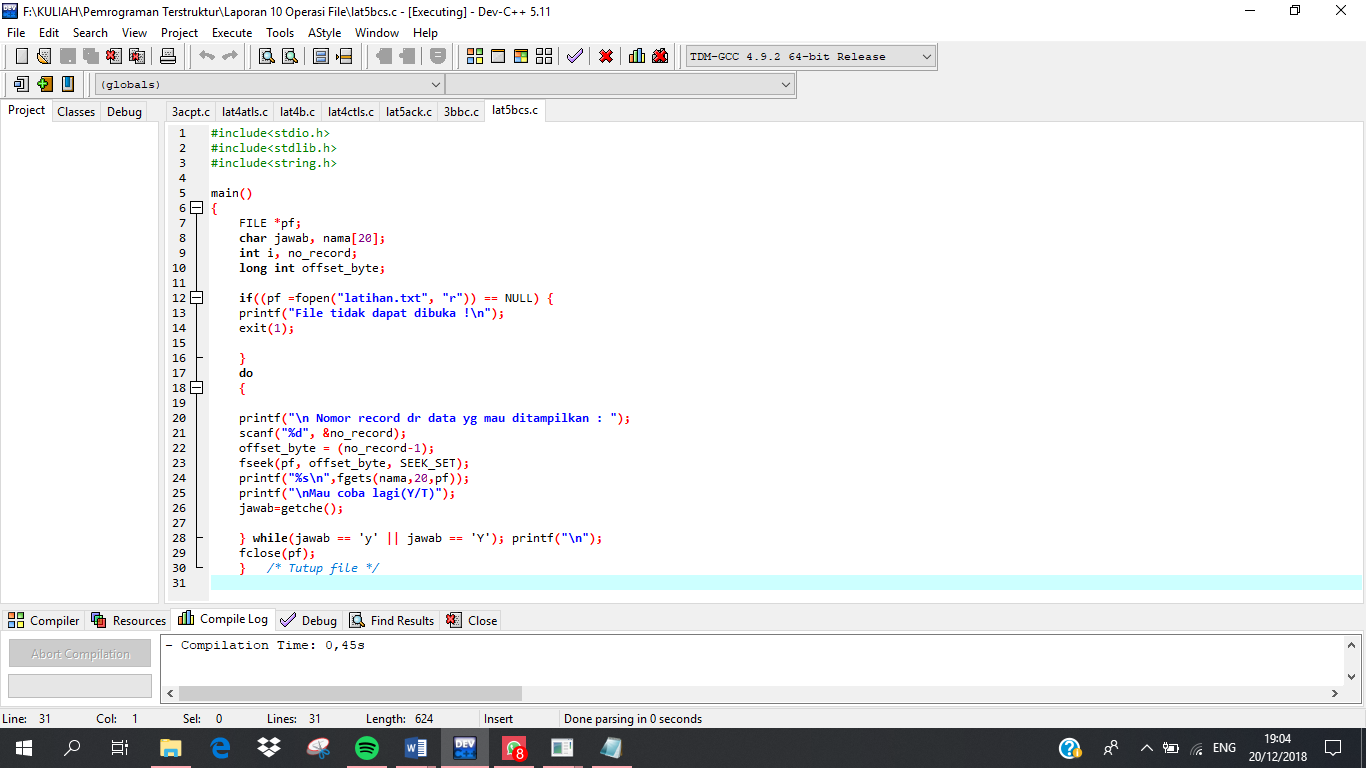
printf("\nMau coba lagi(Y/T)");

jawab=getche();

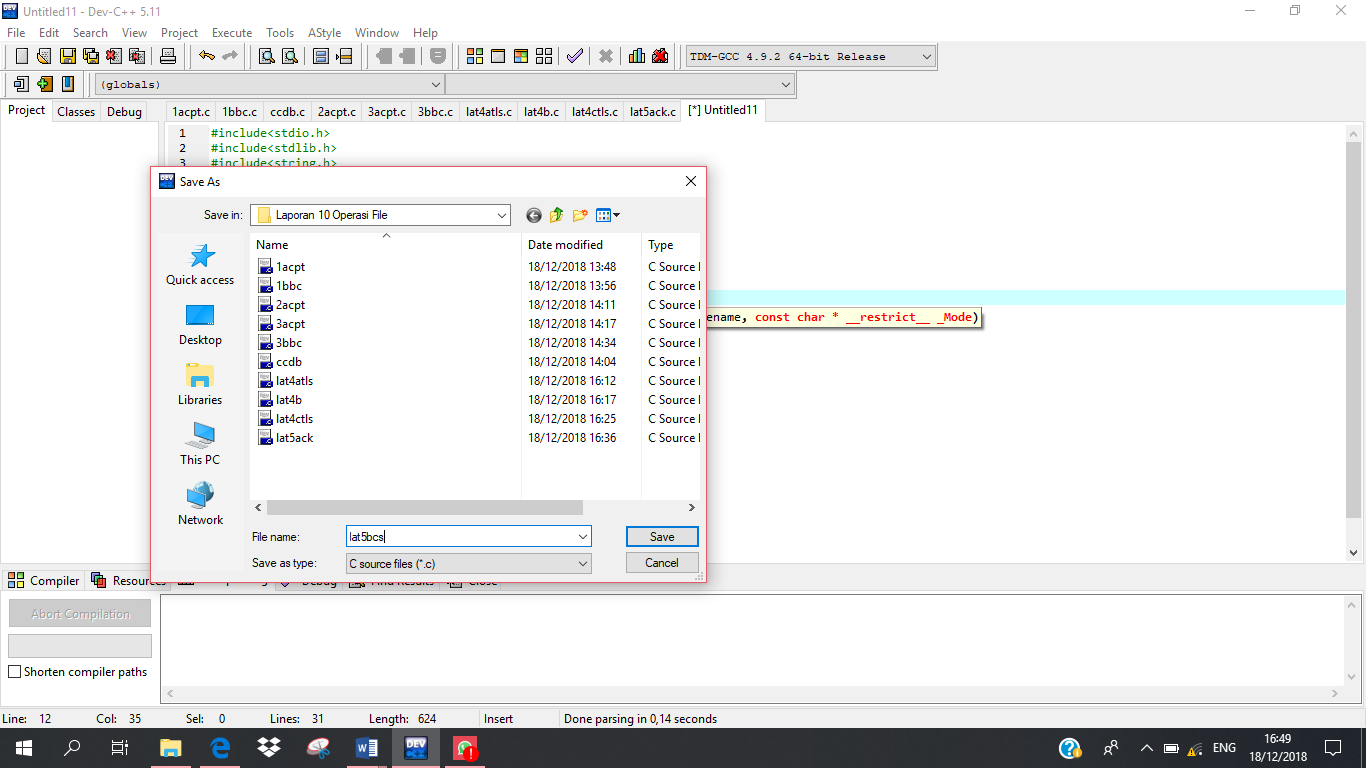
} while(jawab == 'y' || jawab == 'Y'); printf("\n");

fclose(pf);

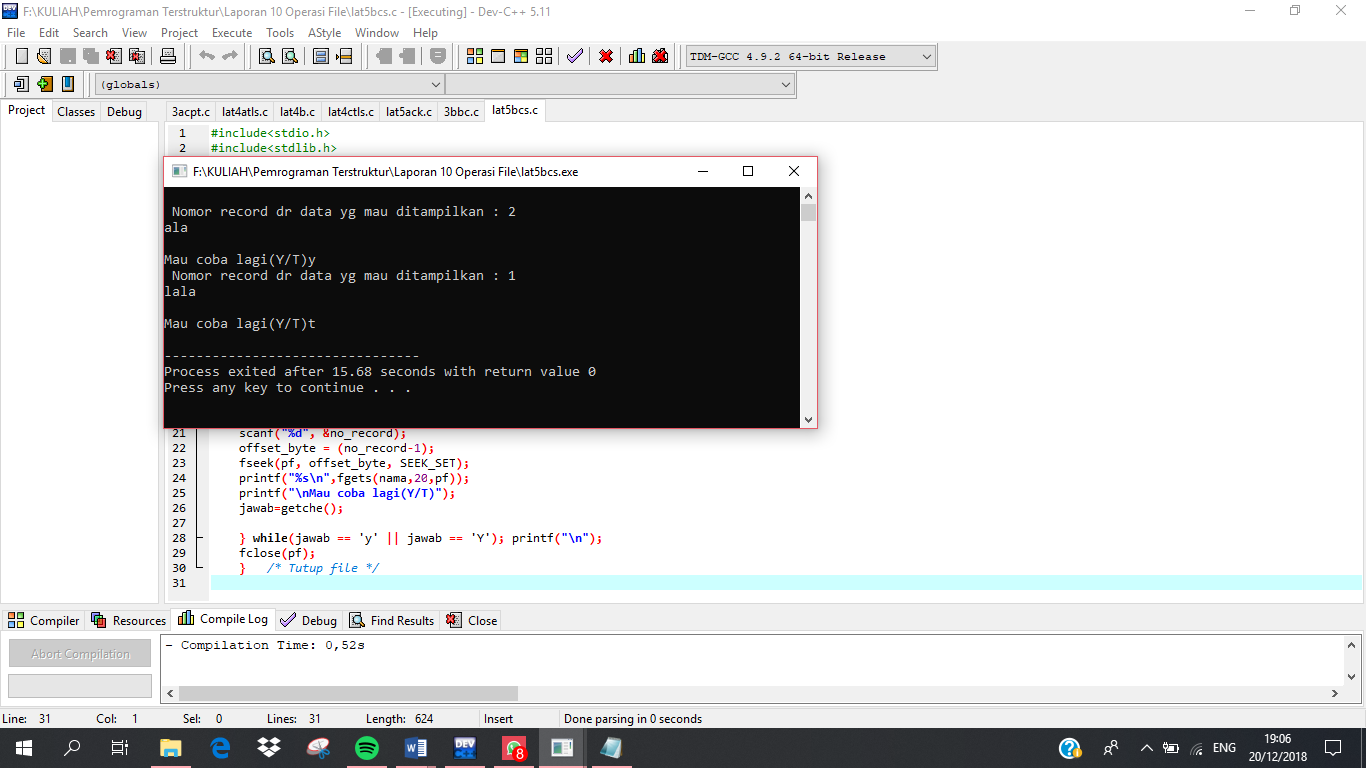
} /\* Tutup file \*/



1. Alt+F, pilih save ketik nama lat5bcs.c

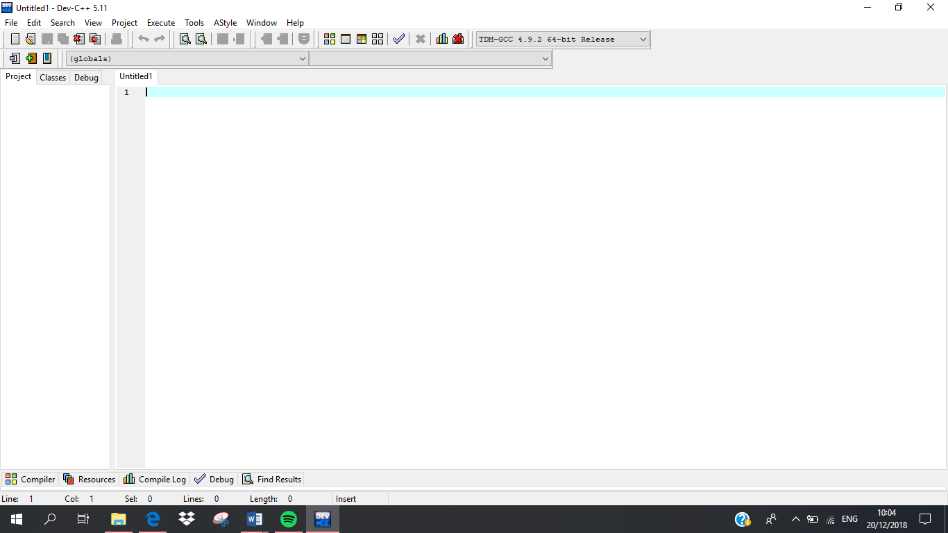


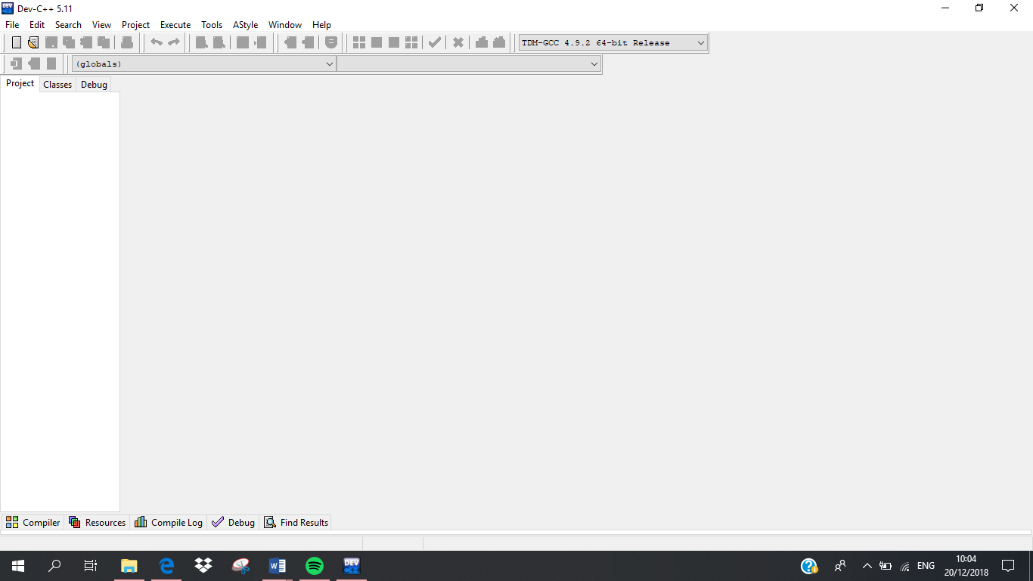
1. Alt+O, pilih directories, ganti semua dengan C:\TC, lalu esc
2. Alt+C, compile to OBJ
3. Alt+C, compile to EXE
4. Alt+R



#### Latihan 6 Contoh Program fseek

1. Latihan 6a : program untuk mnghapus file
2. Jalankan progam Dev C++





1. Alt + F, pilih New
2. Ketik :

#include<stdio.h>

#include<stdlib.h>

#define PJG 65

main()

{

int kode;

char namafile[PJG];

printf("Nama file yang akan dihapus : ");

gets(namafile);

kode = remove(namafile);

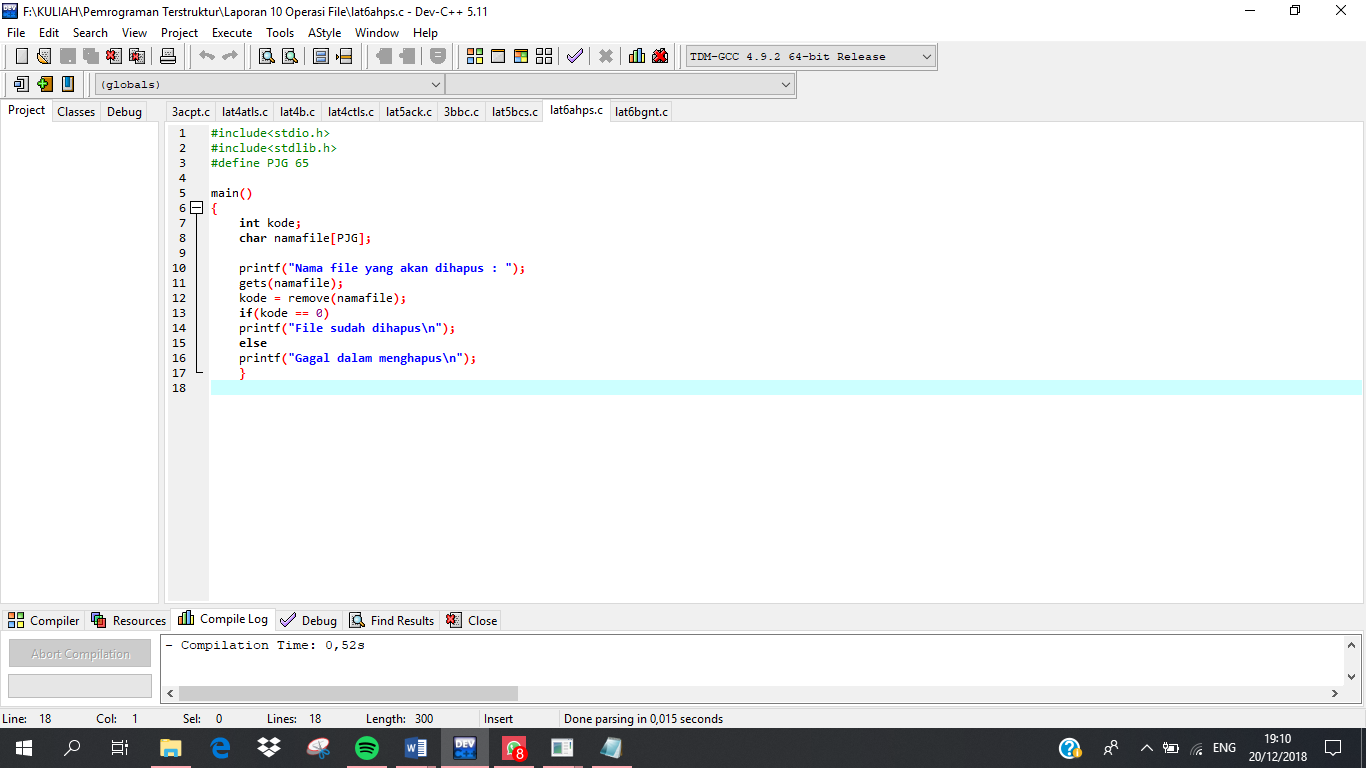
if(kode == 0)

printf("File sudah dihapus\n");

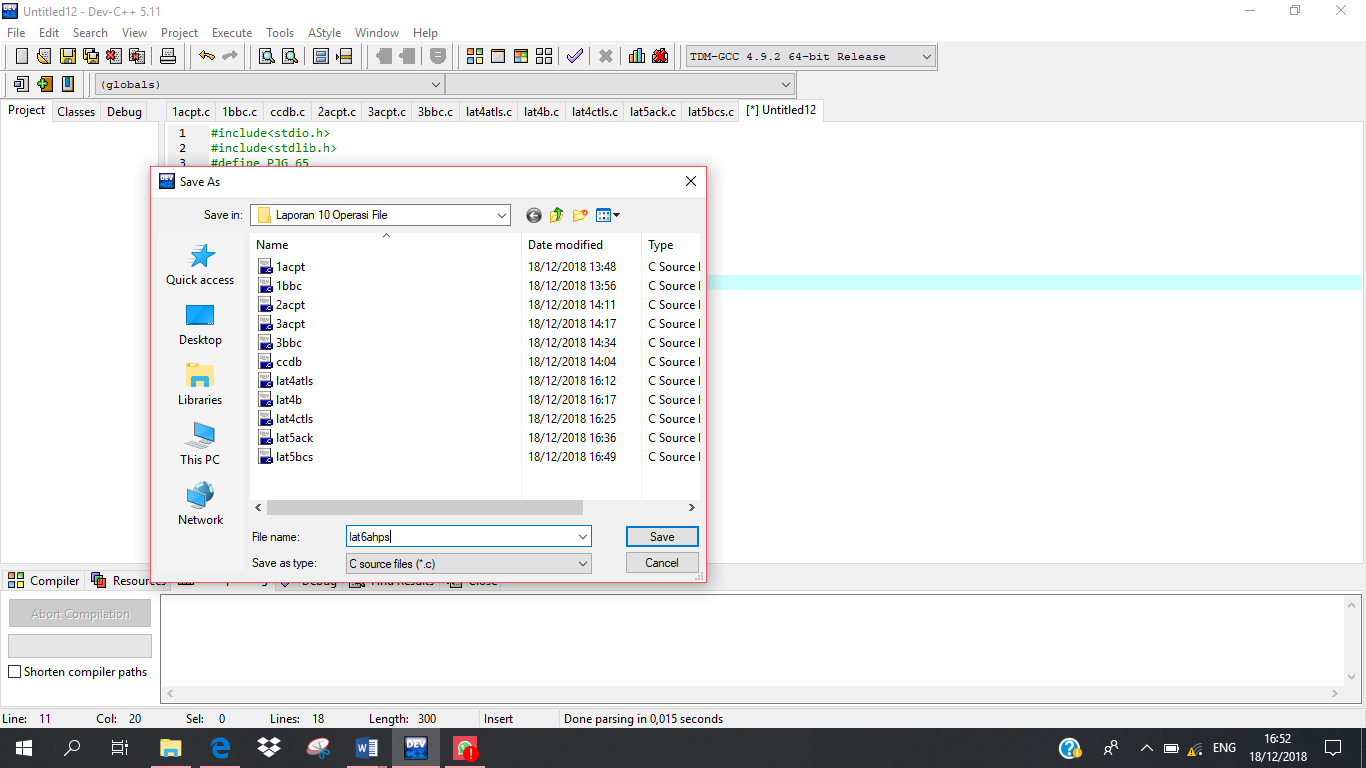
else

printf("Gagal dalam menghapus\n");

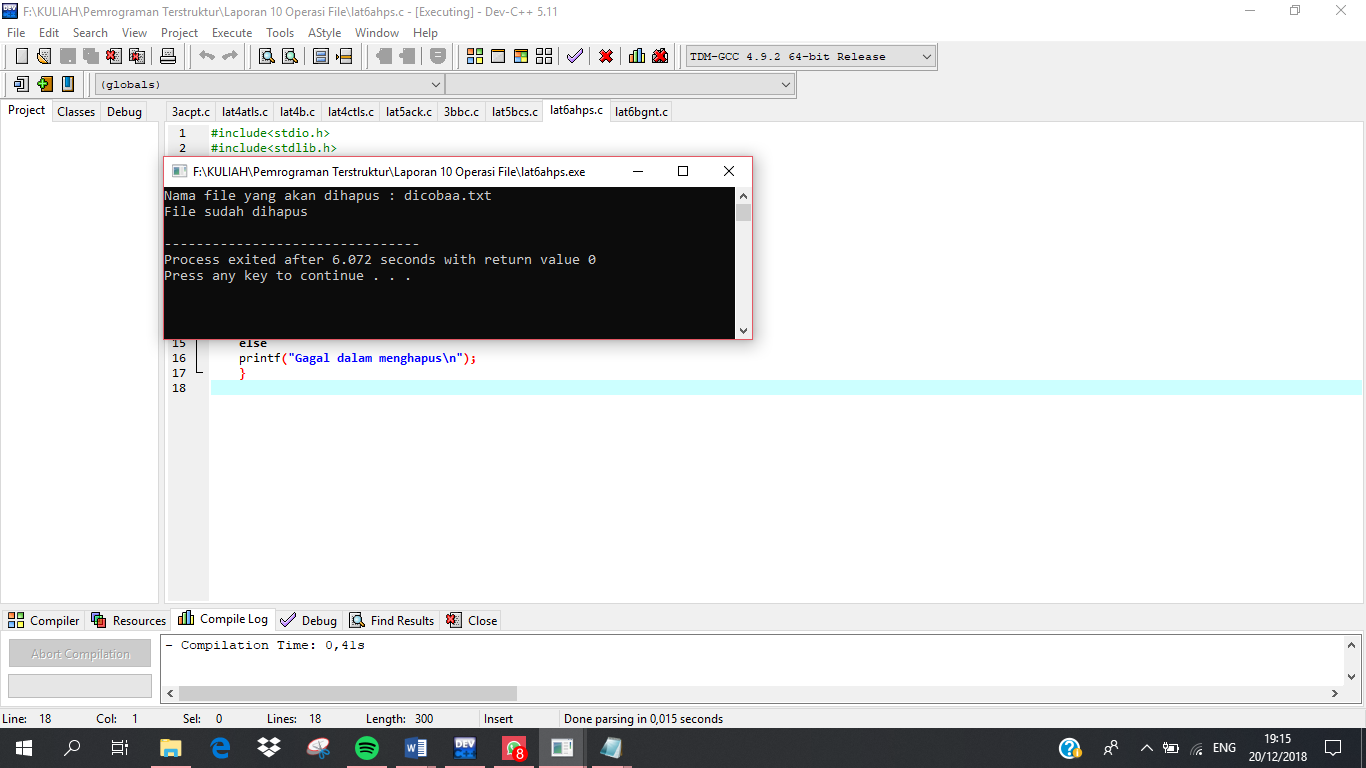
}



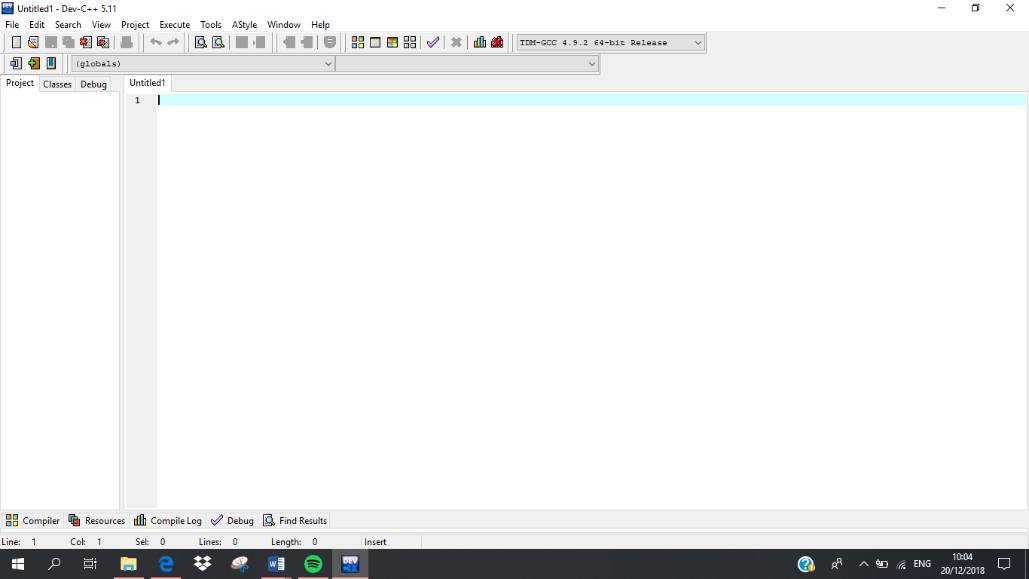
1. Alt+F, pilih save ketik nama lat6ahps.c



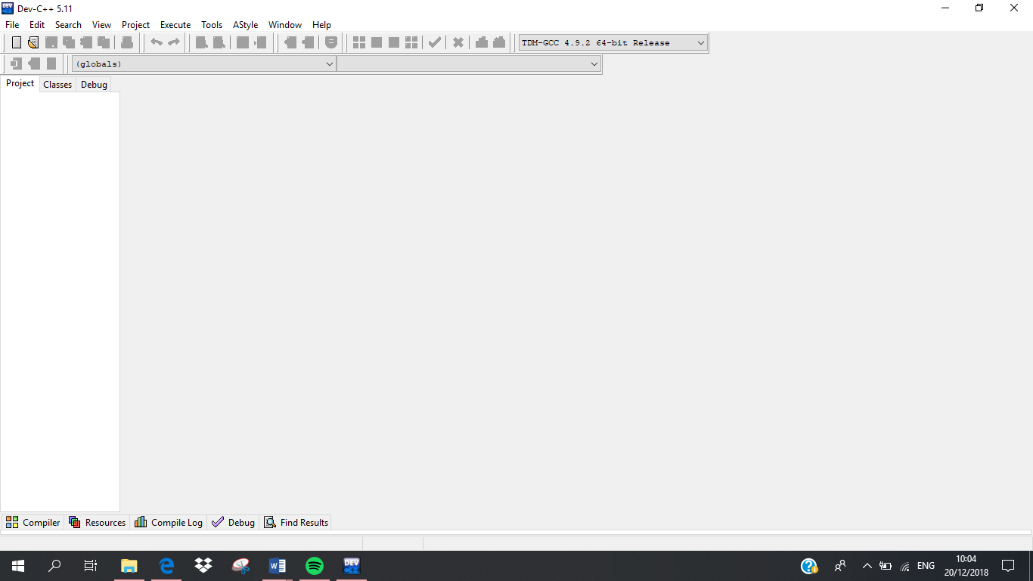
1. Alt+O, pilih directories, ganti semua dengan C:\TC, lalu esc
2. Alt+C, compile to OBJ
3. Alt+C, compile to EXE
4. Alt+R



1. Latihan 6b : program untuk mengganti nama file
2. Jalankan progam Dev C++



1. Alt + F, pilih New



1. Ketik :

#include<stdio.h>

#include<stdlib.h>

#define PJG 65

main()

{

int kode;

char namafilelama[PJG], namafilebaru[PJG];

printf("Nama file yang akan diganti : ");

gets(namafilelama);

printf("Nama file yang baru : ");

gets(namafilebaru);

kode = rename(namafilelama, namafilebaru);

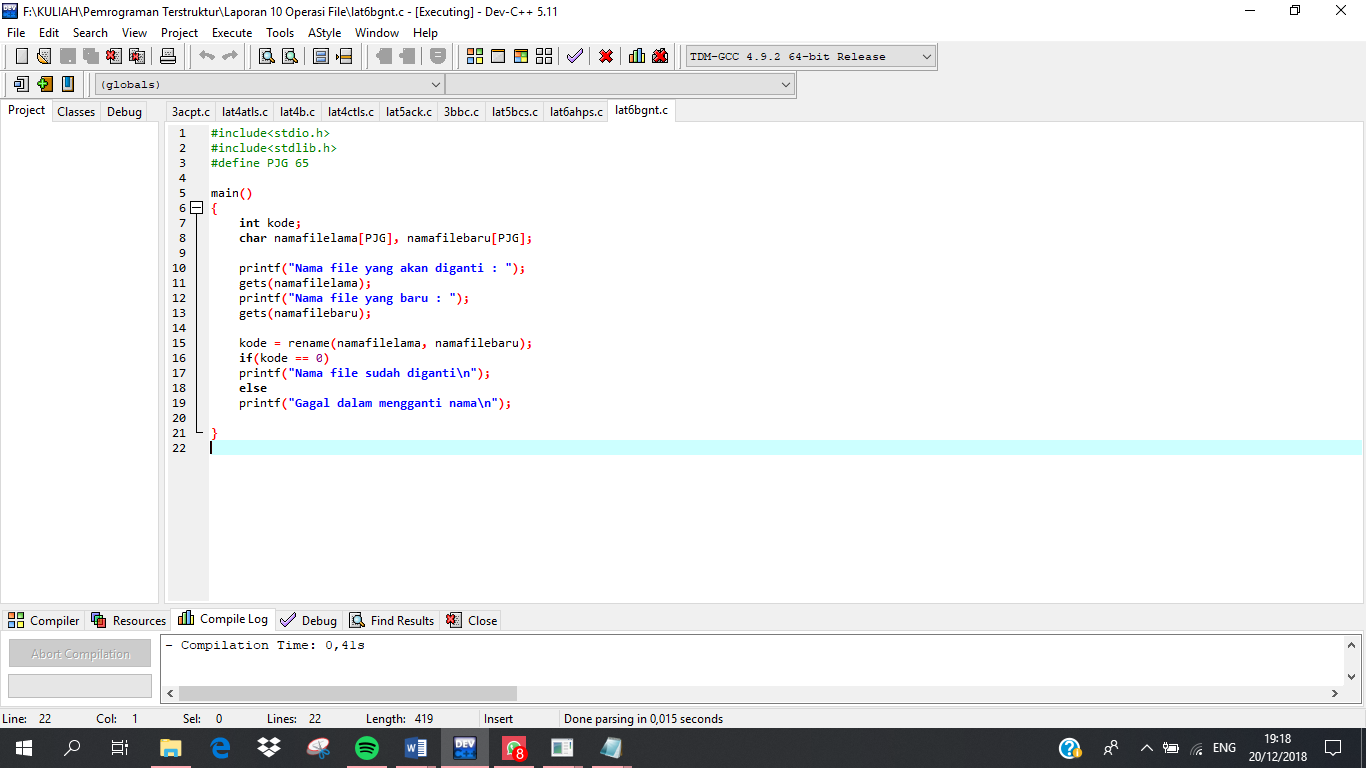
if(kode == 0)

printf("Nama file sudah diganti\n");

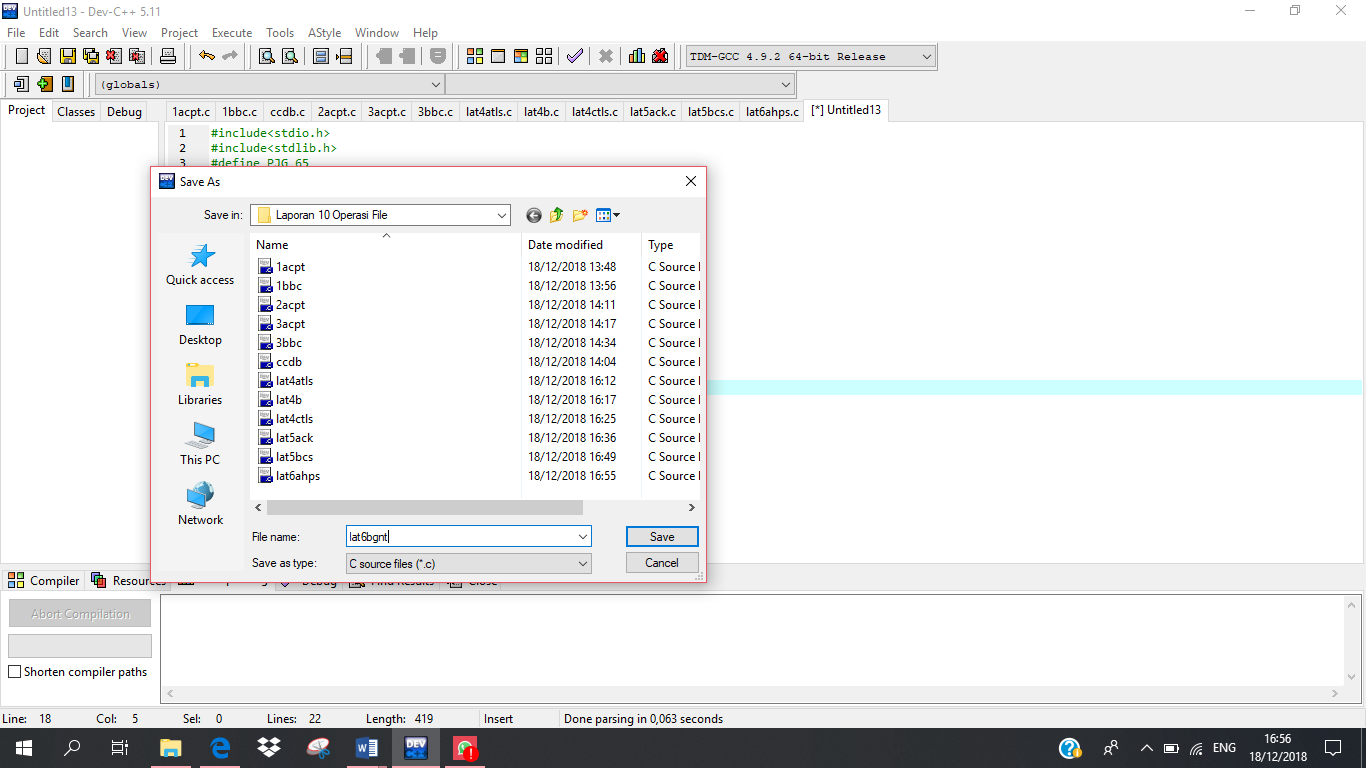
else

printf("Gagal dalam mengganti nama\n");

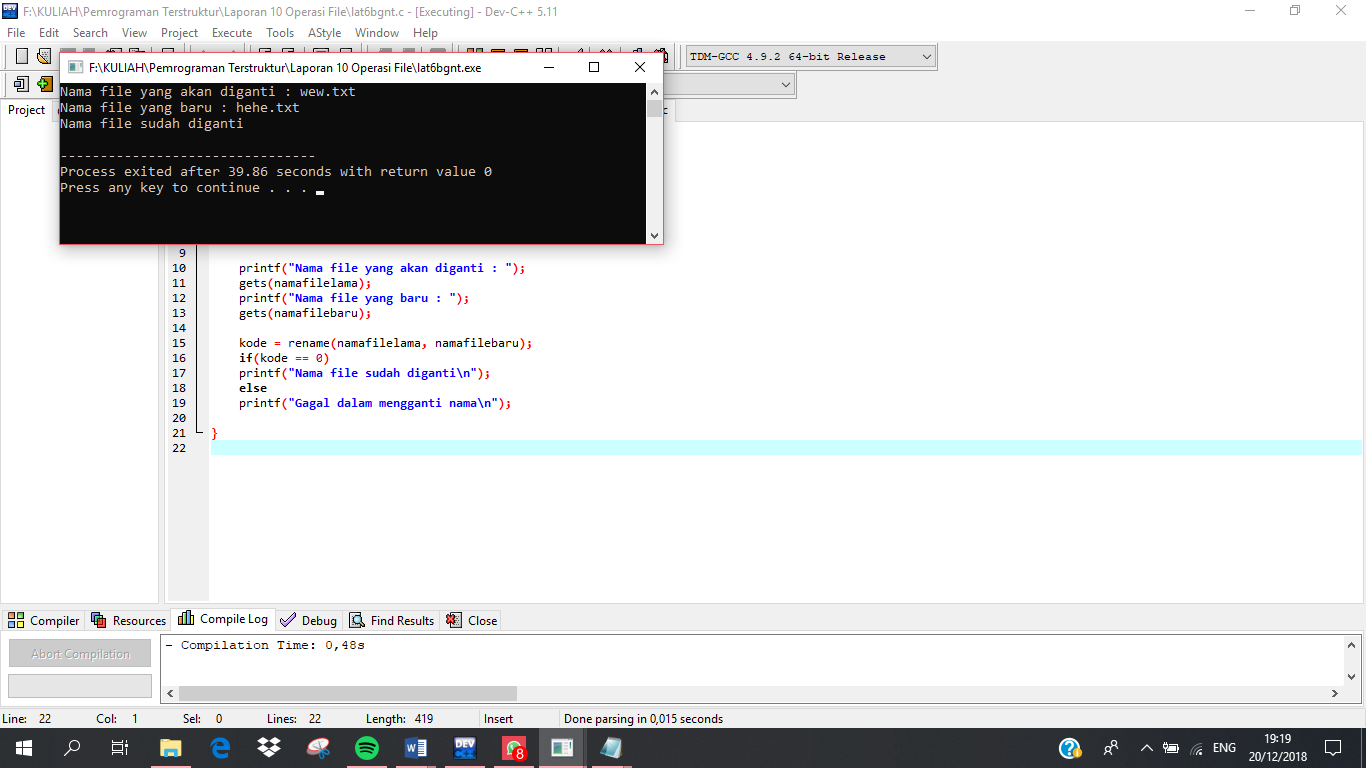
}



1. Alt+F, pilih save ketik nama lat6bgnt.c



1. Alt+O, pilih directories, ganti semua dengan C:\TC, lalu esc
2. Alt+C, compile to OBJ
3. Alt+C, compile to EXE
4. Alt+R



## PERTANYAAN

1. Sebutkan dan jelaskan struktur suatu file !

Jawab :

Struktur suatu file merupakan sebuah penyimpanan suatu data dalam disk yang berupa suatu file.

1. Sebutkan dan jelaskan secara garis besar tahapan operasi suatu file, sertakan contohnya untuk masing-masing tahapan.

Jawab :

* Membuka / mengaktifkan file

Contoh :

if(pf = fopen(“COBA.TXT” “w”) == NULL)

{

printf(“File tidak dapat diciptakan !\n”);

exit(1); //keluar dari program

}

Keterangan :

* pf akan diisi dengan keluaran dari fungsi fopen().
* Jika nilainya NULL, maka akan mencetak “File tidak dapat diciptakan”, setelah itu program dihentikan
* Melaksanakan proses file

Contoh :

* Operasi Baca dan Tulis File Per-Int

int\_putw(int nilai, FILE \*ptr\_file);

int\_getw(FILE \*ptr\_file);

Kegunaan :

\_getw() untuk membaca sebuah data bertipe int dari file

\_putw() untuk menyimpan sebuah data bertipe int ke file

* Menutup file

Contoh :

int fclose(FILE\*pf);

Keterangan :

Apabila sudah tidak diproses lagi, maka file tersebut ditutup, karena adanya keterbatasan jumlah file yang dapat dibuka secara serentak.

1. Sebutkan dan jelaskan jenis-jenis file.

Jawab :

* File Biner : file yang pola penyimpanan di dalam disk berbentuk biner, yaitu seperti bentuk pada memori RAM (komputer). Dipakai untuk menyimpan data kompleks, mis : struct.
* File Teks : file yang pola penyimpanan datanya dalam bentuk karakter. Dipakai untuk menyimpan data seperti karakter atau string.

1. Perhatikan potongan kode program berikut ini :

FILE \*pf;

pf =fopen(“LATIHAN.TXT”, “w+”);

Jelaskan maksud dari kode program tersebut !

Jawab :

* Menciptakan dan mengaktifkan file bernama “LATIHAN.TXT”
* Dengan mode yaitu “w” (mode penulisan ke file)
* Dan menempatkan pointer-ke-FILE e variabel pointer pf

1. Buatlah kesimpulan dari praktik yang telah anda lakukan!

**Kesimpulan**

Operasi file pada dasarnya terbagi menjadi tiga yaitu membuka / mengaktifkan file, melaksanakan proses file, dan menutup file. Untuk melaksanakan perintah tersebut terdapat fungsi yang berbeda. Fopen() berfungsi untuk membuka file, Fclose() berfungsi untuk menutup file, sedangkan fungsi yang digunakan untuk menjalankan file terdapat jenis-jenis operasi file yang berbeda-beda. Jenis file dibedakan menjadi dua yaitu file biner dan file teks. File Biner adalah file yang pola penyimpanan di dalam disk berbentuk biner. File Teks : file yang pola penyimpanan datanya dalam bentuk karakter. Dipakai untuk menyimpan data seperti karakter atau string.

## QUIZ

Buat program dengan langkah membaca perkategori dan hitung jumlah datanya

Dengan data 1-121-1901-123-1911-123-192

2-123-1902-123-1912-123-1922-123-193

3-123-191

Kategori A

Kategori B

Kategori C

**Script Program**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int kategori1=0, kategori2=0, kategori3=0;

char angka[25], data;

FILE \*pf;

if((pf =fopen("hehe.txt","r")) == NULL)

{

printf(" file tidak dapat dibuka ");

exit(0);

}

while(fgets(angka, 10,pf ))

{

printf("%s\n", &angka);

data = angka[0];

if (data == '1')

kategori1++;

else

if(data == '2')

kategori2++;

else

if(data == '3')

kategori3++;

}

printf("kategori 1 jumlahnya %d\n", kategori1);

printf("kategori 2 jumlahnya %d\n", kategori2);

printf("kategori 3 jumlahnya %d\n", kategori3);

fclose(pf);

}

**Run Hasil Program**

