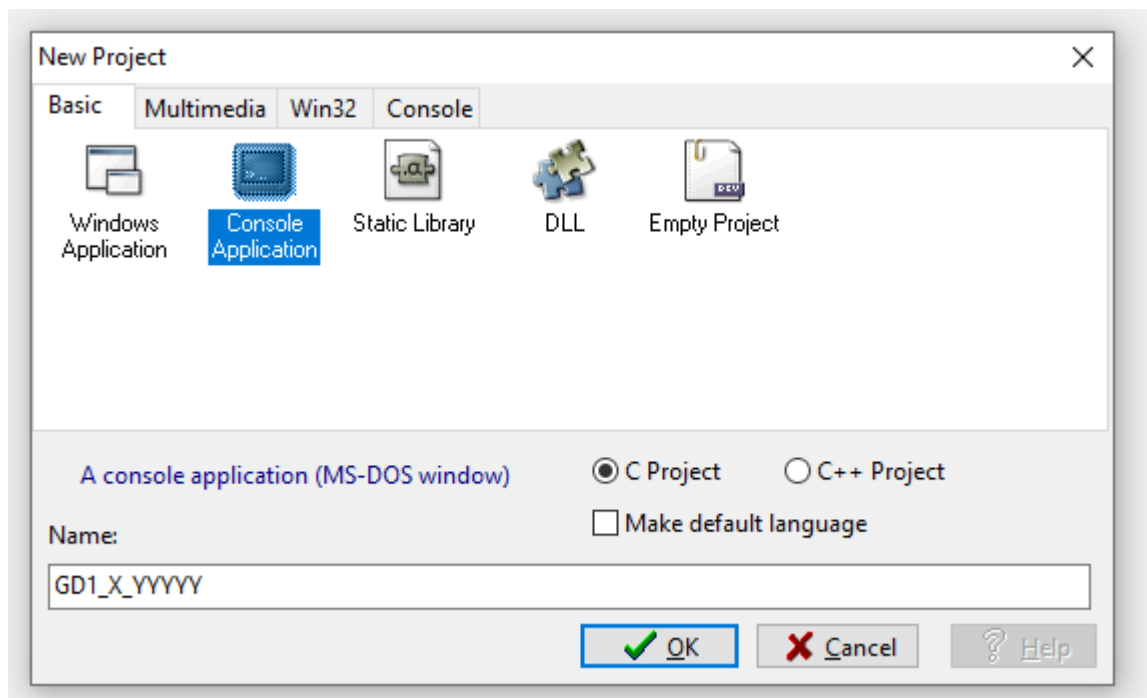


# Guided 1

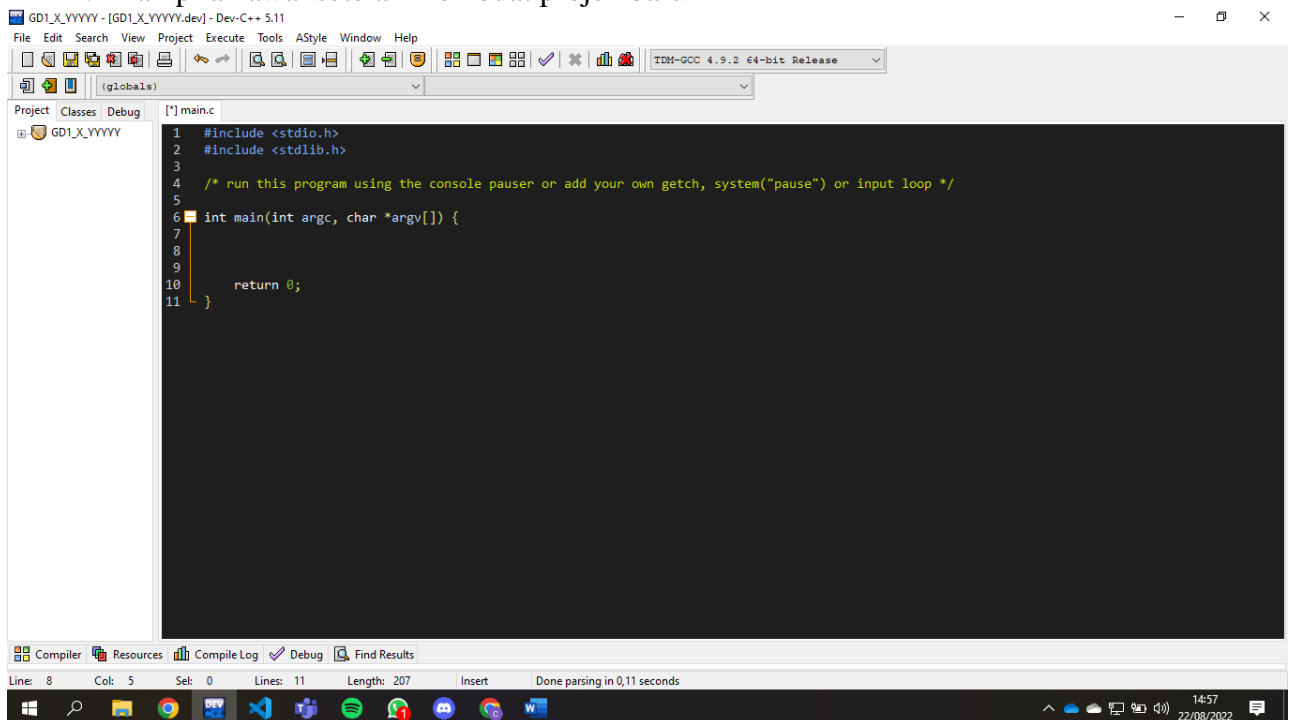
## Abstract Data Types

### Membagi file C menjadi 3 bagian

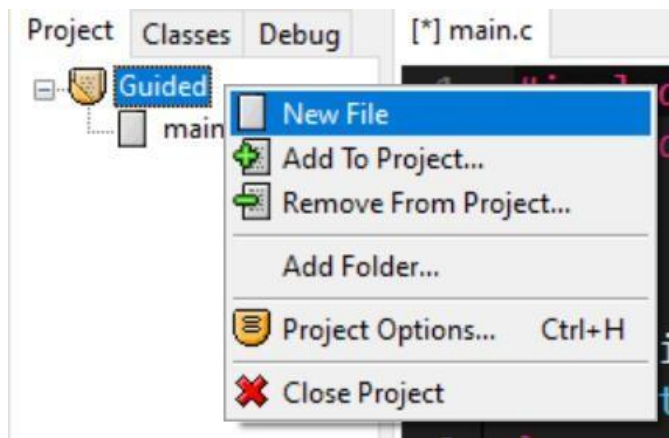
1. Buat project baru dengan template Console Application dan pilihlah tipe berupa C Project



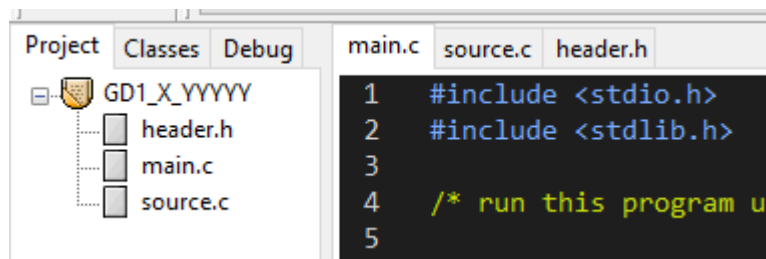
2. Tampilan awal setelah membuat projek baru



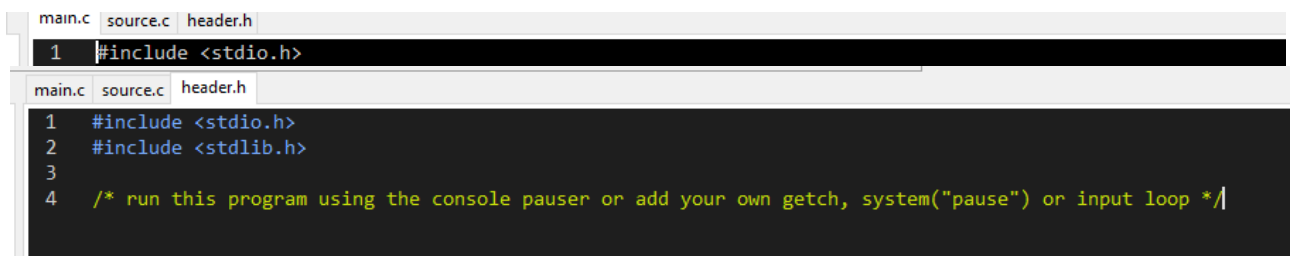
3. Klik kanan pada file `.dev` anda pada tab project lalu pilih new file



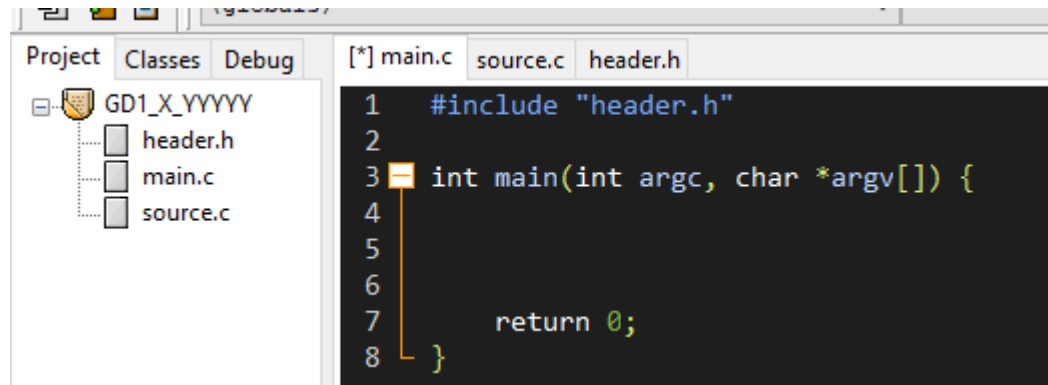
4. Buatlah 2 file baru lalu save dengan nama `source.c` dan `header.h`



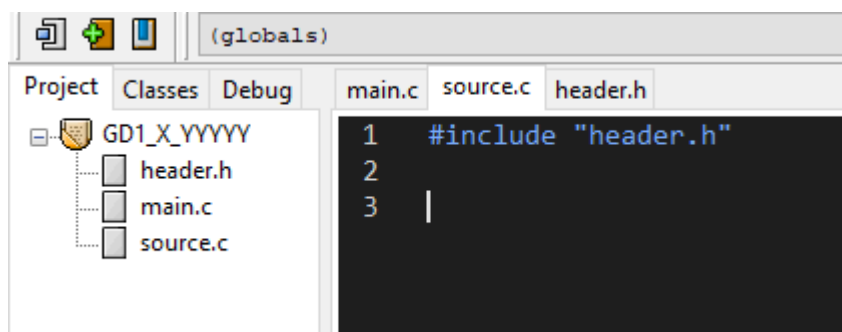
5. Masuk ke file `main.c` lalu cut semua code yang ada di atas `int main` dan pindahkan ke file yang bernama `header.h`. `Stdio.h` dan `stdlib.h` adalah library yang akan digunakan program dan biasanya semua library akan diletakan di file header



6. Masuk ke file `main.c` lalu tambahkan code berikut: `#include "header.h"`. Fungsi code tersebut adalah untuk memasukkan file header ke `main.c` sehingga terhubung dan code pada `header.h` dapat digunakan pada file `main.c`.



Tambahkan juga `#include "header.h"` pada file `source.c` agar saling terhubung. File `source` ini nantinya akan digunakan sebagai wadah pengimplementasian seluruh prosedur dan fungsi yang sudah dideklarasikan pada file `header.h` untuk selanjutnya bisa digunakan pada file `main.c`



7. Dengan begitu, kalian sudah memisahkan 3 fungsi file yang berbeda namun tetap dapat berhubungan satu sama lain.

## header.h

```
// File header.h ini berisi spesifikasi ADT yang akan digunakan

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef char string[20];

// Mendefinisikan sebuah ADT baru dengan nama "Mahasiswa"
typedef struct {
    string nama;
    string npm;
} Mahasiswa;

// Dibawah ini adalah deklarasi prosedur atau fungsi yang kemudian akan digunakan
void initMahasiswa(Mahasiswa m[]);
    // Prosedur untuk menginisialisasi ADT Mahasiswa

Mahasiswa makeMahasiswa(string nama, string npm);
    // Fungsi yang digunakan untuk mengembalikan ADT Mahasiswa dengan nama dan npm sesuai pada parameter

void tampilMahasiswa(Mahasiswa m[]);
    // Prosedur yang digunakan untuk menampilkan seluruh data ADT Mahasiswa

void hapusMahasiswa(Mahasiswa m[], string npm);
    // Prosedur yang digunakan untuk menghapus ADT Mahasiswa

int isFull(Mahasiswa m[]);
    // Prosedur yang digunakan untuk mengecek apakah sebuah ADT terisi penuh atau belum
```

## source.c

```
// File ini berisi detail implementasi seluruh prosedur dan fungsi yang ada di header.h

#include "header.h"

void initMahasiswa(Mahasiswa m[]){
    int i;
    for(i = 0; i < 5; i++){
        strcpy(m[i].nama, "-");
        strcpy(m[i].npm, "-");
    }

    // (*m).nama ekuivalen dengan m->nama
    // Karena disini menggunakan Array, jadi masih menggunakan bentuk dasar yaitu m[i].nama
}

Mahasiswa makeMahasiswa(string nama, string npm){
    // Fungsi untuk mengembalikan ADT Mahasiswa dengan nama dan npm sesuai dengan parameter
    Mahasiswa m;
    strcpy(m.nama, nama);
    strcpy(m.npm, npm);

    return m;
}

void tampilMahasiswa(Mahasiswa m[]){
    int i;

    // Prosedur untuk menampilkan data ADT Mahasiswa
    for(i = 0; i < 5; i++){
        printf("===== DATA %d =====\n", i+1);
        printf("Nama Mahasiswa : %s\n", m[i].nama);
        printf("NPM Mahasiswa : %s\n\n", m[i].npm);
    }
}

void hapusMahasiswa(Mahasiswa m[], string npm){
    int i;

    // Prosedur untuk menghapus ADT Mahasiswa berdasarkan NPM
    for(i = 0; i < 5; i++){
        if(strcmp(m[i].npm, npm) == 0){
            m[i] = makeMahasiswa("-", "-");
            printf("\n[+] Data telah dihapus");
            return;
        }
    }
    printf("\n[!] Data tidak ditemukan!");
}

int isFull(Mahasiswa m[]){
    int i;
    for(i = 0; i < 5; i++){
        if(strcmp(m[i].nama, "-") == 0){
            return i;
        }
    }
    return -1;
}
```

## main.c

```
#include "header.h"

int main(int argc, char *argv[]) {

    Mahasiswa m[5];
    string nama, npm;

    int menu, index;
    initMahasiswa(m);

    do{
        system("cls");
        printf("1. Tambah data\n");
        printf("2. Tampil data\n");
        printf("3. Hapus data\n");
        printf(">>> "); scanf("%d", &menu);

        switch(menu){
            case 1:
                index = isFull(m);
                if(isFull(m) != -1){
                    printf("\nMasukkan nama mahasiswa : "); fflush(stdin); gets(nama);
                    printf("\nMasukkan npm mahasiswa : "); fflush(stdin); gets(npm);

                    // makeMahasiswa disini akan mengembalikan value berupa nama dan npm dan
                    // akan disimpan ke variabel m[index]
                    m[index] = makeMahasiswa(nama, npm);
                    printf("[+] Berhasil menambahkan data");
                } else{
                    printf("\n[!] Data sudah penuh!");
                }
                break;

            case 2:
                tampilMahasiswa(m);
                break;

            case 3:
                printf("\nMasukkan npm mahasiswa yang akan dihapus : "); fflush(stdin); gets(npm);
                hapusMahasiswa(m, npm);
                break;

            case 0:
                printf("\n\n==== Keluar dari program =====");
                break;

            default:
                printf("\n[!] Menu tidak tersedia");
                break;
        }
        getch();
    } while(menu != 0);

    return 0;
}
```

1. Penamaan folder Guided : GD1\_X\_YYYYY

- X = Kelas Praktikan
- YYYYYY = 5 Digit terakhir NPM Praktikan

2. Zip lalu upload ke situs kuliah