

Running Modul Struktur Data

Latihan Pengenalan C++

CREATE PROJECT\

Untuk membuat project baru, silahkan ikuti langkah-langkah berikut.

1. Buka aplikasi Code:Blocks yang sudah terinstal
2. Klik tombol "Create a new project" atau ke **File >> New >> Project...**
3. Pilih "Console Application" pada jendela **New From Template** lalu klik **Go**
4. Selanjutnya klik "Next" dan pilih Bahasa **C++** lalu klik "Next" lagi
5. Tulis judul project anda dengan nama "**RunMod_NIM**" pada field "Project Title" dan tentukan lokasi dimana project anda akan disimpan dengan cara mengklik tombol titik tiga pada field "Folder to create project in:" Jika sudah, klik "Next".
6. Terakhir klik "Finish"

SOAL RUNMOD\

Pada latihan soal ini, kita akan mempelajari tentang himpunan. Himpunan adalah kumpulan obyek yang berbeda, dimana tidak ada duplikasi obyek dan obyek tidak harus dalam urutan tertentu. Diberikan dua buah himpunan, program dibawah ini akan membuat himpunan irisan dari kedua himpunan.

Seperti yang kalian ketahui, irisan adalah operasi himpunan yang akan menghasilkan himpunan baru yang anggota-anggota himpunannya ada di kedua himpunan yang lama.

Contoh:

Himpunan A: {1, 3, 5, 6}
Himpunan B: {2, 3, 4, 5, 7}
Irisan A dan B: {3, 5}

Himpunan A: {4, 2, 3, 1}
Himpunan B: {2, 5, 6, 1}
Irisan A dan B: {2, 1}

ISI PROJECT\

Setelah anda membuat project baru, silahkan buka file main.cpp. Silahkan ikuti instruksi berikut ini dengan menuliskannya kembali dalam Bahasa **C++**:

1. Tulis tipe data bentukan berikut. Disini kita akan menggunakan array dinamis, oleh karena itu tidak diperlukan konstanta maksimal untuk arraynya. Kode berikut ditulis sebelum function main()

```
using namespace std;

const nMax: integer = 10
Type Himpunan <
    anggota : array[0..nMax-1] of integer
    panjang: integer
>

int main();
```

2. Tuliskan subprogram berikut pada baris selanjutnya (masih sebelum function main).
Penulisan function pada **C++** harus diawali oleh return type kemudian baru nama function-nya. Contoh **bool** anggotaHimpunan(...)

```
function anggotaHimpunan(himp: Himpunan, bil: integer) -> Boolean
/* mencari bil di dalam himpunan himp. True jika ada, False jika
tidak ditemukan atau himpunannya kosong - SEQUENTIAL SEARCH */

kamus
    found : boolean
    i : integer
algoritma
    found <- false
    i <- 0
    while (i <= himp.panjang and not found) do
        found <- bil == himp.anggota[i]
        i <- i + 1
    endwhile
    return found
endfunction
```

Untuk **parameter output** dan **input/output** gunakan symbol ampersand "&" sebelum nama variabel.
Contohnya &himp. Untuk perintah input/read gunakan **cin >>**. Contohnya input(N) ditulis **cin >> N**.

```
procedure inputAnggotaHimpunan(output himp: Himpunan)
/* IS.      : Data tersedia di dalam piranti masukan
Proses     : proses input akan berhenti apabila bilangan sudah pernah ada di
            dalam himpunan atau sudah mencapai batas kapasitas maksimum dari
            arraynya
FS.        : Terdefinisi himpunan bilangan bulat beserta panjangnya */
kamus
    bilangan: integer
algoritma
    himp.panjang <- -1
    input(bilangan)
    while (himp.panjang < nMax-1 and not anggotaHimpunan(himp, bilangan)) do
        himp.panjang <- himp.panjang + 1
        himp.anggota[himp.panjang] <- bilangan
        input(bilangan)
    endwhile
endprocedure
```

Kita juga bisa menjadikan tipe data bentukan sebagai return dari fungsi. Penulisan dua buah parameter yang memiliki tipe data yang sama harus ditulis terpisah. Contoh: Himpunan himp1, Himpunan himp2.

```
function irisan (himp1, himp2: Himpunan) -> Himpunan
/* IS.      : terdapat dua buah himpunan yang sudah terdefinisi sebelumnya
   Proses   : himp1 dan himp2 dilakukan irisan untuk menghasilkan himpunan baru
   FS.      : terdapat himpunan baru yang anggotanya merupakan bagian dari himp1
              dan himp2 */

kamus
    intersect: Himpunan
    i, j: integer
algoritma
    intersect.panjang = -1;
    i = 0;
    do
        j = 0;
        do
            if (himp1.anggota[i] = himp2.anggota[j]) then
                intersect.panjang <- intersect.panjang + 1
                intersect.anggota[intersect.panjang] = himp1.anggota[i]
            endif
            j = j + 1
        while (j <= himp2.panjang)
        i = i + 1
    while (i <= himp1.panjang)
    return intersect
endfunction
```

Selanjutnya kita buat prosedur untuk mengoutputkan isi di dalam himpunannya

```
procedure printHimp(himp: Himpunan)
/* IS.      : Himpunan himp tidak kosong
   FS.      : Seluruh anggota himpunan himp sudah ditampilkan */
kamus
    i: integer
algoritma
    i = 0
    do
        output (himp.anggota[i])
    while (i <= himp.panjang)
endprocedure
```

3. Untuk melengkapi program anda, silahkan ketikkan kode berikut pada main() dengan bahasa C++ di project anda

```
program HimpunanIrisan
kamus
    set1, set2, set3: Himpunan
algoritma
    output("Anggota himpunan 1: ")
    inputAnggotaHimpunan(set1)

    output("Anggota himpunan 2: ")
    inputAnggotaHimpunan(set2)

    printHimp(set1)
    printHimp(set2)

    set3 = irisan(set1, set2)
    printHimp(set3)
endprogram
```

4. Setelah selesai, coba jalankan program dengan klik tombol "Build and Run" (tombol bergambar gir dan segitiga) atau dengan menekan tombol **F9** di keyboard
5. **Pahami setiap baris kode yang telah anda ketik**
6. Selesai

<=====SELAMAT MENGERJAKAN RUNNING MODUL=====>