**Modul 2**

**Program C dengan header dan include : SORTING**

Nama : ..................................................... NIM : ................................. Tgl : ................ Nilai : ..................

**Skill C Programming :**

* Membuat file-file header;

**Deskripsi program :**

Dari program yang berisi prosedur (dan fungsi), kita dapat membaginya menjadi beberapa bagian sehingga program yang besar dapat lebih terstruktur.

Sebuah file header adalah sebuah file berekstensi .h yang mengandung deklarasi fungsi C dan definisi macro yang dapat dipakai oleh file program lainnya. Ada dua tipe file header : file yang ditulis programmer dan file-file yang ikut serta dalam compiler.

Cara menggunakan atau memanggil file header adalah dengan menuliskannya pada bagian preprosesor C yaitu dengan direktif #include. Meng-include kan sebuah file header sama dengan meng-copy semua isi dari file header tersebut. Jika kita ingin membangun sebuah program yang terdiri atas beberapa file source code, ada baiknya semua konstanta, macro, variabel global dan prototipe fungsi ada di file header. “include” kan file ini apabila diperlukan.

Sebagai catatan :

Sebuah program utama dapat meng-include kan file berekstensi .h maupun .c (jika berupa fungsi atau prosedur). Untuk file .h cukup di-save saja file tersebut sebelum bisa digunakan. Untuk file .c harus di-compile dulu sebelum digunakan. Akibatnya, jika ada konstanta-konstanta yang diperlukan di file .c maka file yang berisi konstanta atau deklarasi lainnya juga harus di-include dalam file .c tersebut.

1. Copy-lah file di bawah ini, beri nama sorting.c, kemudian compile dan run. Ini adalah file asal yaitu program lengkap dalam satu file.

**sorting.c**

#include <stdlib.h>

#include <stdio.h>

#define ArraySize 11

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Sorting

Deskripsi : Mengurutkan array A dengan berbagai metode Sorting

IS : Array A telah berisi nilai sembarang

FS : Array A terurut

Dibuat oleh : Winangsari

Tanggal : 4 April 2016

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void BubbleSort (int AkhirTakTerurut , int A[ArraySize-1]);

void SelectionSort (int AkhirTakTerurut, int A[ArraySize-1]);

int main ()

// KAMUS \*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

int A[ArraySize-1]; //A array acak

int i;

//ALGORITMA PROGRAM UTAMA \*\*\*\*\*\*\*\*\*\*

printf ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n");

printf ("Program SORTING - BUBBLE SORT dan SELECTION SORT \n");

printf ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n\n");

// Mengisi array dengan bilangan random

printf ("Isi Array A (acak) Mula-mula adalah ..... \n");

for (i=1; i<=ArraySize-1; i++)

{

A[i] = rand() % 1000; //random antara 0 - 999

printf ("%d ", A[i]);

}

printf ("\n\n");

// BubbleSort (ArraySize-1, A);

SelectionSort (ArraySize-1, A);

printf ("\*\*\*\*\*\*\*\* Hasil Akhir sorting \*\*\*\*\*\*\*\*\*\*\* \n");

for (i=1; i<=ArraySize-1; i++)

{

printf ("%d ", A[i]);

}

printf ("\n\n");

system ("pause");

}

void BubbleSort (int AkhirTakTerurut, int A[ArraySize-1])

// Deskripsi : Mengurutkan array A dari elemen ke-1 hingga ke-AkhirTakTerurut

// secara ascending dengan metode Bubble Sort

// IS : Array A telah berisi nilai sembarang

// FS : Array A terurut

// Dibuat oleh : Winangsari

// Tanggal : 4 April 2016

{

int aw, ak, i, j, temp;

printf ("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BUBBLE SORT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

aw = 1;

ak = AkhirTakTerurut;

while (ak >= 2)

{

printf ("\*\*\*Sekarang cari nilai yang tepat untuk lokasi terbelakang A[%d] \n", ak);

i = aw;

j = i + 1;

while (j <= ak)

{

printf ("bandingkan A[%d]=%d dan A[%d]=%d ", i, A[i], j, A[j]);

if (A[i] > A[j])

{

printf ("----> Tukar....\n");

temp = A[i];

A[i] = A[j];

A[j] = temp;

//Tampilkan urutan array saat ini

for (i=1; i<=ArraySize-1; i++)

{

printf ("%d ", A[i]);

}

printf ("\n\n");

}

else

printf ("----> Sudah benar urutannya, biarkan....\n\n");

i = j;

j = i + 1;

}

ak = ak - 1;

printf ("\n");

}

}

void SelectionSort (int AkhirTakTerurut, int A[ArraySize-1])

// Deskripsi : Mengurutkan array A dari elemen ke-1 hingga ke-AkhirTakTerurut

// secara ascending dengan metode Selection Sort

// IS : Array A telah berisi nilai sembarang

// FS : Array A terurut

// Dibuat oleh : Winangsari

// Tanggal : 4 April 2016

{

int aw, ak, i, j, indeks, MAX;

printf ("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SELECTION SORT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

aw = 1;

ak = AkhirTakTerurut;

while (ak >= 2) //Lakukan pengulangan hingga tinggal 2 elemen

{

printf ("Sekarang mencari nilai yang tepat untuk lokasi terbelakang A[%d] \n", ak);

//Cari MAX di ruang yang belum terurut, yaitu dari aw hingga ak

MAX = A[aw]; //MAX diisi oleh nilai array pertama

printf ("Mula-mula MAX[%d] = %d \n", aw, MAX);

i = aw + 1; //mulai bandingkan dengan elemen ke 2

while (i <= ak)

{

//bandingkan MAX dan A[i]

if (A[i] > MAX)

{

MAX = A[i];

indeks = i;

};

printf ("Banding dg elemen ke-%d : MAX = A[%d] = %d \n", i, indeks, MAX);

i++;

}

printf ("\n");

//Taruh nilai MAX di akhir array yang belum terurut

A[indeks] = A[ak];

A[ak] = MAX;

for (i=1; i<=ArraySize-1; i++)

{

printf ("%d ", A[i]);

}

printf ("\n");

ak = ak - 1;

}

}

1. Bagi-bagilah program menjadi beberapa bagian :
2. Main Program, dengan nama file : main\_sorting.c
3. File header yang berisi deklarasi variabel / konstanta, nama file : header\_sorting.h
4. File header yang berisi prosedur bubble sort, nama file : prosedur\_sorting\_bubble.h
5. File source code yang berisi prosedur selection sort, nama file : prosedur\_sorting\_selection.c

Save lah file-file header. Compile source code prosedur\_sorting\_selection.c. Terakhir compile dan run main\_sorting.c

**main\_sorting.c**

#include <stdlib.h>

#include <stdio.h>

#include "header\_sorting.h"

#include "prosedur\_sorting\_bubble.h"

#include "prosedur\_sorting\_selection.c"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Sorting

Deskripsi : Mengurutkan array A dengan berbagai metode Sorting

IS : Array A telah berisi nilai sembarang

FS : Array A terurut

Dibuat oleh : Winangsari

Tanggal : 4 April 2016

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main ()

// KAMUS \*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

int A[ArraySize-1]; //A array acak

int i;

//ALGORITMA PROGRAM UTAMA \*\*\*\*\*\*\*\*\*\*

printf ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n");

printf ("Program SORTING - BUBBLE SORT dan SELECTION SORT \n");

printf ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n\n");

// Mengisi array dengan bilangan random

printf ("Isi Array A (acak) Mula-mula adalah ..... \n");

for (i=1; i<=ArraySize-1; i++)

{

A[i] = rand() % 1000; //random antara 0 - 999

printf ("%d ", A[i]);

}

printf ("\n\n");

// BubbleSort (ArraySize-1, A);

SelectionSort (ArraySize-1, A);

printf ("\*\*\*\*\*\*\*\* Hasil Akhir sorting \*\*\*\*\*\*\*\*\*\*\* \n");

for (i=1; i<=ArraySize-1; i++)

{

printf ("%d ", A[i]);

}

printf ("\n\n");

system ("pause");

}

**header\_sorting.h**

#define ArraySize 11

**prosedur\_sorting\_bubble.h**

void BubbleSort (int AkhirTakTerurut, int A[ArraySize-1])

// Deskripsi : Mengurutkan array A dari elemen ke-1 hingga ke-AkhirTakTerurut

// secara ascending dengan metode Bubble Sort

// IS : Array A telah berisi nilai sembarang

// FS : Array A terurut

// Dibuat oleh : Winangsari

// Tanggal : 4 April 2016

{

int aw, ak, i, j, temp;

printf ("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BUBBLE SORT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

aw = 1;

ak = AkhirTakTerurut;

while (ak >= 2)

{

printf ("\*\*\*Sekarang cari nilai yang tepat untuk lokasi terbelakang A[%d] \n", ak);

i = aw;

j = i + 1;

while (j <= ak)

{

printf ("bandingkan A[%d]=%d dan A[%d]=%d ", i, A[i], j, A[j]);

if (A[i] > A[j])

{

printf ("----> Tukar....\n");

temp = A[i];

A[i] = A[j];

A[j] = temp;

//Tampilkan urutan array saat ini

for (i=1; i<=ArraySize-1; i++)

{

printf ("%d ", A[i]);

}

printf ("\n\n");

}

else

printf ("----> Sudah benar urutannya, biarkan....\n\n");

i = j;

j = i + 1;

}

ak = ak - 1;

printf ("\n");

}

}

**prosedur\_sorting\_selection.c**

#include "header\_sorting.h"

void SelectionSort (int AkhirTakTerurut, int A[ArraySize-1])

// Deskripsi : Mengurutkan array A dari elemen ke-1 hingga ke-AkhirTakTerurut

// secara ascending dengan metode Selection Sort

// IS : Array A telah berisi nilai sembarang

// FS : Array A terurut

// Dibuat oleh : Winangsari

// Tanggal : 4 April 2016

{

int aw, ak, i, j, indeks, MAX;

printf ("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SELECTION SORT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

aw = 1;

ak = AkhirTakTerurut;

while (ak >= 2) //Lakukan pengulangan hingga tinggal 2 elemen

{

printf ("Sekarang mencari nilai yang tepat untuk lokasi terbelakang A[%d] \n", ak);

//Cari MAX di ruang yang belum terurut, yaitu dari aw hingga ak

MAX = A[aw]; //MAX diisi oleh nilai array pertama

printf ("Mula-mula MAX[%d] = %d \n", aw, MAX);

i = aw + 1; //mulai bandingkan dengan elemen ke 2

while (i <= ak)

{

//bandingkan MAX dan A[i]

if (A[i] > MAX)

{

MAX = A[i];

indeks = i;

};

printf ("Banding dg elemen ke-%d : MAX = A[%d] = %d \n", i, indeks, MAX);

i++;

}

printf ("\n");

//Taruh nilai MAX di akhir array yang belum terurut

A[indeks] = A[ak];

A[ak] = MAX;

for (i=1; i<=ArraySize-1; i++)

{

printf ("%d ", A[i]);

}

printf ("\n");

ak = ak - 1;

}

}

Tugas tambahan : beberapa bagian diganti, kemudian analisis hasilnya / apa yang terjadi