

CONTOH MODULAR PROGRAMMING dalam BAHASA C

Part – 2

```
/* File      : maxmodul.cpp */
/* Deskripsi  : Contoh program yang mengandung FUNGSI dan Procedure */
/*           : Untuk mencari posisi bilangan maksimum pada suatu array*/
/* Dibuat oleh : Dosen DDP / 132 231 591 */
/* Tanggal   : 29-10-2012 */

#include <stdio.h>
// deklarasi modul yang digunakan

int max(int x[5], int t, int *p);

int main()
{
    /* program utama */
    /* Kamus */
    int a[5]={5, 9, -3, 2, 11};
    int b[7]={4, 8, 1, 7, -9, 2, 10};
    int j, k, n, m;

    j=0;
    k=0;
    n=5;
    m=7;
    printf("Mencari posisi bilangan terbesar pada array A : \n");
    j=max(a,n,&j);
    printf("Posisi bilangan dengan nilai terbesar pada array A = %d,
dengan nilai bilangan = % d \n", j, a[j]);
    printf("Mencari posisi bilangan terbesar pada array B : \n");
    k=max(b,m,&k);
    printf("Posisi bilangan dengan nilai terbesar pada array B = %d,
dengan nilai bilangan = % d \n", k, b[k]);
    printf("Hasil kali bilangan terbesar pada array A dan B = %d \n",
a[j]*b[k]);

    return 0;
}

/* BODY realisasi prosedur/fungsi */

int max(int x[], int t, int *p)    // parameter x,y merupakan passing
parameter by value
{
    /* Algoritma mencari posisi bilangan terbesar pada array 1 dimensi */
    int i;
    *p = 0;
    i = 1;
    while (i < t)
    {
        if ( x[*p] < x[i])
        { *p = i; }
        i = i + 1;
    }
    return *p;
} /* end function max */
```

```

/* File      : subprg.cpp */
/* Deskripsi  : Contoh program yang mengandung FUNGSI dan Procedure */
/*           : Ada dua cara untuk deklarasi prototype. Yang dipakai
cara2*/
/* Dibuat oleh : Dosen DDP / 132 231 591 */
/* Tanggal    : 29-10-2012 */

#include <stdio.h>
// deklarasi modul yang digunakan
int maxab (int x, int y);
void tukar(int *x, int *y);

int main()
{
/* program utama */
/* Membaca dua bilangan bulat */
/* Menuliskan maksimum dua bilangan yang dibaca dengan memanggil fungsi */
/* Menukar kedua bilangan dengan Procedure */
/* Kamus */
int a, b, c;
printf("Maksimum dua bilangan : \n");
printf("Ketikkan dua bilangan, pisahkan dengan RETURN : \n");
scanf("%d %d", &a, &b);
printf("Kedua bilangan : a = %d, b = %d \n", a,b);
// menampilkan maksimum dari 2 bilangan dapat dilakukan dengan 2 cara
:
// cara ke-1 :
//c = maxab(a,b); printf("Maksimum = %d \n", c);
//cara ke-2 :
printf("Maksimum = %d \n", maxab (a,b));
printf("Menukar kedua bilangan di atas : \n");
tukar (&a,&b);
printf("Kedua bilangan setelah ditukar : a = %d, b = %d \n",a,b);
return 0;
}

/* BODY realisasi prosedur/fungsi */

int maxab (int x, int y) // parameter x,y merupakan passing parameter by
value
{
/* Algoritma mencari maksimum dua bilangan bulat */
if (x >= y)
{ return x; }
else
{ return y; }

/* cara lain :
return ((x >= y) ? x : y);
*/
} /* end function maxab */

void tukar (int *x, int *y) // parameter x,y merupakan passing parameter by
address
{
int temp;

/* menukar 2 bilangan integer */
temp = *x;
*x = *y;
*y = temp;
} /* end procedure tukar */

```

LATIHAN !

Alat hitung (ALU (Aritmethic Logical Unit)) yang ada di dalam mesin komputer hanya mengenal operasi tambah (+) dan kurang (-). Semua perintah matematik yang berhubungan dengan proses kali, bagi, pangkat, akar diterjemahkan/dihitung dengan menggunakan proses tambah dan kurang. Contoh untuk melaksanakan perintah perkalian bilangan 2 dengan 5, dilakukan dengan cara menambahkan bilangan 2 dengan 2 sebanyak 5 kali.

$$(2 * 5 = 2 + 2 + 2 + 2 + 2).$$

Berikut ini adalah algoritma yang belum selesai (belum lengkap) dalam bentuk modular untuk melakukan proses perkalian dua buah bilangan bulat dengan proses penjumlahan.

Lengkapi algoritma tersebut lalu buat programnya dalam bahasa C.

Diketahui

Bil, pengali, hasil adalah variabel global bertipe integer

Algoritma

Procedure kali(b,p,h)

```
{ b,p   parameter input passing by value bertipe integer.
  h   parameter output passing by address/reference bertipe integer.
  bp, n variabel lokal bertipe integer. }
bp ← p
if (b = 0) OR (p = 0)
  then operasi-1
  else
    if kondisi-1
      then operasi-2
    endif
    n ← 1
    while kondisi-2 do
      operasi-3
      n ← n + 1
    endwhile
  endif
if p < 0
  then h ← ekspresi-1
endif
end_procedure_kali
```

Modul Utama {pemanggilan procedure}

```
  hasil ← 0
  Read(bil)
  Read(pengali)
  Kali(bil,pengali,hasil)
  Write(hasil)
end_modul_utama.
```