Modul PDP-10
Jam

Dalam Tugas ini anda harus menyelesaikan dengan standar 3 file (d**rvjam.c,jam.h,jam.c dan boolean.h**). Masalah yang harus anda selesaikan adalah mengenai pengolahan suatu penanganan waktu (jam). Semua fungsi dan prosedur (jam.h dan jam.c) yang anda buat harus di test dalam driver (**drvjam.c**). Dibawah ini seluruh header fungsi dan spesifikasi dari masalah yang harus anda selesaikan.

Keterangan :

I.S. = Initial State (keadaan awal)
F.S. = Final State (keadaan akhir)

```
/*Variabel Global jam,menit dan detik*/
    int HH; //jam
    int MM; //menit
    int SS; //detik
extern const int maxdetik=86400;
/*
 DEFINISI PRIMITIF
 KELOMPOK VALIDASI TERHADAP TYPE
*/

function IsJValid (Input H, M, S : integer) → boolean
/* Mengirim true jika H,M,S dapat membentuk J yang valid }
{ dipakai untuk mentest SEBELUM membentuk sebuah Jam }
*/

/* Konstruktor: Membentuk sebuah JAM dari komponen-komponennya */
procedure MakeJam (Input HH, MM, SS : integer)
/* Membentuk sebuah JAM dari komponen-komponennya yang valid
   Prekondisi : HH, MM, SS valid untuk membentuk JAM
*/
/*  Operasi terhadap komponen : selektor Get dan Set *** } */
/* { *** Selektor *** } */
function GetHour () → integer
/*{ Mengirimkan bagian HH (Hour) dari JAM }*/
function GetMinute () → integer
/*{ Mengirimkan bagian Menit (MM) dari JAM }*/
function GetSecond () → integer
/*{ Mengirimkan bagian Second (SS) dari JAM }*/
/*{ *** Pengubah nilai Komponen *** }*/
procedure SetHH (input/output HH : Integer, input newHH :
integer)
/*{ Mengubah nilai komponen HH dengan newHH }*/
```

```
procedure SetMM (input/output MM : Integer, input newMM :
integer)
/*{ Mengubah nilai komponen MM dengan newMM }*/
procedure SetSS (input/output SS : Integer, input newSS :
integer)
/*{ Mengubah nilai komponen SS dg newSS}*/

/*{ KELOMPOK BACA/TULIS }*/

procedure BacaJam (output HH,MM,SS : Integer)
/*I.S. : HH,MM,SS tidak terdefinisi
  F.S. : HH,MM,SS terdefinisi di Variabel global
  Proses :
  mengulangi membaca komponen H,M,S sehingga membentuk Jam yang
  valid. Tidak mungkin menghasilkan Jam yang tidak valid.
*/
procedure TulisJam ()
/*
{ I.S. : sembarang }
{ F.S. : Nilai variable global ditulis dg format HH:MM:SS }
{ Proses : menulis nilai setiap komponen Jam ke layar }
*/

/*{ KELOMPOK KONVERSI TERHADAP TYPE }*/
function JamToDetik (Input HH,MM,SS : Integer) → integer
/*
{ Diberikan sebuah JAM, mengkonversi menjadi Detik }
{ Rumus : detik = 3600*hour+menit*60 + detik }
{ nilai maksimum = 3600*23+59*60+59*60 }
{ Hati-hati dengan representasi integer pada bahasa implementasi
{ kebanyakan sistem mengimplementasi integer, }
{ bernilai maksimum kurang dari nilai maksimum hasil konversi }
*/
procedure DetikToJam (Input N : integer)
/*
{ Mengirim konversi detik ke JAM }
{ Catatan: Jika N ≥ 86400, maka harus dikonversi dulu menjadi
jumlah detik yang mewakili jumlah detik yang mungkin dalam 1
hari, yaitu dengan rumus: N1 = N mod 86400, baru N1 dikonversi
menjadi JAM }
*/
/*
{ KELOMPOK OPERASI TERHADAP TYPE }
{ *** Kelompok Operator Relational *** }
*/
function JEQ (Input HH1,MM1,SS1,HH2,MM2,SS2 : Integer) → boolean
/*{ Mengirimkan true jika JAM1=JAM2, false jika tidak }*/
```

```
function JNEQ (Input HH1,MM1,SS1,HH2,MM2,SS2 : Integer) → boolean
/*{ Mengirimkan true jika J1 tidak sama dengan J2 }*/
function JLT (Input HH1,MM1,SS1,HH2,MM2,SS2 : Integer) → boolean
/*{ Mengirimkan true jika J1<J2 , false jika tidak }*/
function JGT (Input HH1,MM1,SS1,HH2,MM2,SS2 : Integer) → boolean
/*{ Mengirimkan true jika J1>J2, false jika tidak }*/
/*{ *** Operator aritmatika JAM *** }*/
procedure JPlus (Input HH1,MM1,SS1,HH2,MM2,SS2 : Integer)
/*{ Menghasilkan JAM1+JAM2, dalam bentuk JAM di variable
Global }*/
procedure JMinus (Input HH1,MM1,SS1,HH2,MM2,SS2 : Integer)
/*{ Menghasilkan JAM1-JAM2, dalam bentuk JAM }*/
/*{ Prekondisi : JAM1<=JAM2 }*/
procedure NextDetik (Input HH,MM,SS)
/*{ Mengirim 1 detik setelah HH,MM,SS dalam bentuk JAM di */
/*variable global }*/
procedure NextNDetik (Input HH,MM,SS, N : integer)
/* Mengirim N detik setelah HH,MM,SS dalam bentuk JAM di */
/* variabel global }*/
function PrevDetik (Input HH,MM,SS  : integer)
/*{ Mengirim 1 detik sebelum HH,MM,SS dalam bentuk JAM di */
/* variable global }*/
function PrevNDetik (Input HH,MM,SS , N : integer)
/*{ Mengirim N detik sebelum J dalam bentuk JAM }*/

/*{ *** Kelompok Operator Aritmetika *** }*/
function Durasi (Input HH1,MM1,SS1,HH2,MM2,SS2:Integer)
/*{ Mengirim JAM2-JAM1 dlm Detik, dengan kalkulasi }*/
/*{ Hasilnya negatif jika JAM1 > JAM2 }*/
```

```
/*
File : drvjam.c
Driver jam  untuk menguji jam.c,jam.h dan boolean.h
*/
#include"jam.h"
#include"boolean.h"
int HH1=16,MM1=43,SS1=13;
int HH2=23,MM2=30,SS2=33;
int detik=65378;
int main()
{
    //fungsi jam
    printf("IsJValid : %d\n",IsJValid(HH,MJ,SS));
    printf("%d:%d:%d =%d detik\n",HH,MJ,SS,Jam2Detik(HH,MJ,SS));
    Detik2Jam(detik);
    printf("%d detik sama dengan %d:%d:%d \n",detik,HH,MJ,SS);
```

```
    JPlus(HH1,MM1,SS1,HH2,MM2,SS2);
    printf("%d:%d:%d ditambah %d:%d:%d sama dengan %d:%d:%d \n",
               HH1,MM1,SS1,HH2,MM2,SS2,HH,MJ,SS);
    JMinus(HH1,MM1,SS1,HH2,MM2,SS2);
    printf("%d:%d:%d dikurang %d:%d:%d sama dengan %d:%d:%d \n",
               HH2,MM2,SS2,HH1,MM1,SS1,HH,MJ,SS);
    NextDetik(21,49,24);
    printf("Next Detik dari 21:49:24 = %d:%d:%d\n",HH,MJ,SS);
    NextNDetik(21,49,24,15);
    printf("Next %d Detik dari 21:49:24 = %d:%d:%d\n",
               15,HH,MJ,SS);
    PrevDetik(21,49,24);
    printf("Prev Detik dari 21:49:24 = %d:%d:%d\n",HH,MJ,SS);
    PrevNDetik(21,49,24,15);
    printf("Next %d Detik dari 21:49:24 = %d:%d:%d\n",
               15,HH,MJ,SS);
    printf("Durasi dari 24:10:00 - 23:10:00 = %d detik\n",
               Durasi(24,10,0,23,10,0));
   return 0;
}
```