

Modul PDP-10

Jam

Petunjuk :

Dengan menggunakan standard 2 file (file driver dan library), selesaikan masalah penanganan jam atau waktu. Program utama/driver bernama **testJam.py** dan file library **jam.py**

```
#File : testJam.py
-*- coding: utf-8 -*-
from jam import *
def main():
    print("Test MakeJam: ",end='');MakeJamTest(23,45,50)
    j1=MakeJam(23,45,50)
    print("Getter dan Setter")
    print("Jam      : {}".format(GetHour(j1)))
    print("Menit   : {}".format(GetMinute(j1)))
    print("Detik    : {}".format(GetSecond(j1)))
    print("Jam j1      : ",end='');TulisJam(j1)
    j2=MakeJam(0,0,0)
    print("Jam j2      : ",end='');TulisJam(j2)
    print("Setting ulang j1 dan j2")
    SetHH(j1,2);SetMM(j1,20);SetSS(j1,20)
    print("Jam j1      : ",end='');TulisJam(j1)
    SetHH(j2,3);SetMM(j2,20);SetSS(j2,20)
    print("Jam j2      : ",end='');TulisJam(j2)
    print("Konversi");jconv=MakeJam(6,45,35)
    print("Konversi jam-detik 6:45:35 = {}".format(Jam2Detik(jconv)))
    d2j=Detik2Jam(Jam2Detik(jconv))
    TulisJam(d2j)

    print("j3=j1+j2")
    j3=MakeJam(21,12,59)
    j4=MakeJam(12,9,51)
    JPlus(j3,j4)
    j5=MakeJam(12,34,55)
    j6=MakeJam(8,12,15)
    print("Selisih j4-j3      : ",end='');TulisJam(JMinus(j6,j5))

    print("j1==j2 : %r"%JEQ(j1,j2))
    print("j1!=j2 : %r"%JNEQ(j1,j2))
    print("j1<j2  : %r"%JLT(j1,j2))
    print("j1>j2  : %r"%JGT(j1,j2))

    print("Next 1 detik j1 : %d"%Jam2Detik(NextDetik(j1)))
    print("Next 1 detik j2 : %d"%Jam2Detik(NextDetik(j2)))
    print("Next 15 detik j1: %d"%Jam2Detik(NextNDetik(j1,15)))
    print("Next 15 detik j2: %d"%Jam2Detik(NextNDetik(j2,15)))
    print("Prev 1 detik j1 : %d"%Jam2Detik(PrevDetik(j1)))
    print("Prev 1 detik j2 : %d"%Jam2Detik(PrevDetik(j2)))
    print("Prev 15 detik j1: %d"%Jam2Detik(PrevNDetik(j1,15)))
    print("Prev 15 detik j2: %d"%Jam2Detik(PrevNDetik(j2,15)))
    print("Durasi      : %d"%Durasi(j1,j2))
```

```

    print("Ta da ! ")

if __name__ == '__main__':
    main()

```

```

#File library : jam.py
-*- coding: utf-8 -*-
from math import *
# J,M,D mewakili suatu sistem penanggalan hari, bulan, tahun
# representasi dengan struktur data Dictionary
# Jam={"jam":None,"menit":None,"detik":None}
# jam : 0..23, menit:0..59, detik:0..59
# Konstruktor

#Validasi Konstant jumlah detik maksimal
MAXDETIK=86400

Function ResetJam(J)→ Jam
''' Mereset isi J menjadi jam=menit=detik=0'''

Function IsJValid(H,M,S)→ Boolean
'''mengembalikan True H : 0..23, M:0..59, S:0..59, selain itu False'''

#Konstruktor
Function MakeJam(HH,MM,SS)→ Jam
'''
Menghasilkan Jam dengan nilai HH, MM, SS setelah dites dengan IsJValid,
dan menghasilkan pesan 'Invalid Jam', jika tidak valid

'''

#test konstruktor
Function MakeJamTest (HH,MM,SS)
'''
Melakukan test terhadap MakeJam dan menuliskannya di layar dengan TulisJam
'''

#Selektor*/
Function GetHour(J) → Integer
''' Menghasilkan nilai jam dari J '''

Function GetMinute(J)→ Integer
''' Menghasilkan nilai menit dari J '''

Function GetSecond(J)→ Integer
''' Menghasilkan nilai detik dari J '''

Function SetHH(J, newHH)→ Integer
''' memberi nilai jam dengan newHH pada J dan mengembalikan J'''

Function SetMM(J,newMM)→ Integer

```

```

''' memberi nilai menit dengan newMM pada J dan mengembalikan J'''

Function SetSS(J,newSS)→ Integer
''' memberi nilai detik dengan newSS pada J dan mengembalikan J'''

#Baca Tulis*/
Function BacaJam(J)→ Integer
'''
membaca input 3 variabel dalam 1 baris untuk variabel j,m,d
sebagai parameter argumen MakeJam, lalu menuliskannya di layar
dengan TulisJam, serta mengembalikan J
'''

Function TulisJam(J)
'''
menulis J dengan format HH:MM:SS, setelah J diperiksa dengan IsJValid
untuk mendapatkan nilai jam,menit dan detik dengan selektor sebelumnya
'''

#Konversi Tipe*/
Function Jam2Detik(J)→ Integer
'''
mengembalikan nilai integer hasil konversi jam ke detik dengan rumus
jam*3600+menit*60+detik
'''

Function Detik2Jam(N)→ Jam
'''
merubah N (suatu bilangan integer) menjadi Jam dengan algoritma:
Buat variabel sisa dan N1←N
Jika N>MAXDETIK maka
    N1 ← N1 mod 3600
    jam ← N1 div 3600
    sisa ← N1 mod 3600
    menit ← sisa div 60
    detik ← sisa div 60
selain itu
    jam ← N1 div 3600
    sisa ← N1 mod 3600
    menit ← sisa div 60
    detik ← sisa mod 60
'''

#operasi type
Function JEQ(J1,J2)→ Boolean
'''
mengembalikan nilai True jika J1 sama dengan J2
gunakan fungsi Jam2Detik, untuk membantu
'''

Function JNEQ(J1,J2)→ Boolean
'''
mengembalikan True jika J1 tidak sama dengan J2
kebalikan dari JEQ
'''

```

```

Function JLT(J1,J2)→ Boolean
'''
mengembalikan True jika J1 lebih kecil dari J2
'''

Function JGT(J1,J2)→ Boolean
'''
mengembalikan True jika J1 lebih besar dari J2
'''

#operasi aritmatik
Function JPlus(J1,J2)→ String
'''
Menghasilkan format string dengan TulisJam, dari penjumlahan J1 dengan J2
dengan algoritma :
sum ← MakeJam
s ← h ← m ← day ← 0
s ← detik dari J1 + detik dari J2
Jika s>60 maka
    m ← s div 60; s ← s mod 60
    m ← m + menit dari J1 + menit dari J2
Jika m>60 maka
    h ← m div 60; m ← m mod 60
    h ← h + jam dari J1+ jam dari J2
Jika h>24 maka
    day ← 1; h ← h mod 24
jika day=0 maka
    TulisJam(MakeJam(h,m,s))
selain itu
    Output(day)
    TulisJam(MakeJam(h,m,s))
'''

Function JMinus(J1,J2) → Integer
'''
menghitung selisih dari J1 dengan J2, dengan algoritma :
selisih ← MakeJam(0,0,0)
jika detik J2 > detik J1 maka
    menit J1 ← menit J1 - 1
    detik J1 ← detik J1 +60
    detik selisih ← detik J1 - detik J2

jika menit J2 > menit J1 maka
    jam J1 ← jam J1 - 1
    menit J1 ← menit J1 +60

menit selisih ← menit J1- menit J2
jam selisih ← jam J1-jam J2
→ selisih
'''

Function NextDetik(J)→ Integer
'''
mengembalikan Jam setelah 1 detik setelah J
gunakan fungsi Detik2Jam dan Jam2Detik untuk membantu menyelesaikan
'''

```

```
Function NextNDetik(J,N)→ Integer
```

```
'''
```

```
mengembalikan Jam setelah N detik setelah J
```

```
gunakan fungsi Detik2Jam dan Jam2Detik untuk membantu menyelesaikan
```

```
'''
```

```
Function PrevDetik(J) → Jam2Detik
```

```
'''
```

```
mengembalikan Jam sebelum 1 detik sebelum J
```

```
gunakan fungsi Detik2Jam dan Jam2Detik untuk membantu menyelesaikan
```

```
'''
```

```
Function PrevNDetik(J,N)→ Jam
```

```
'''
```

```
mengembalikan Jam sebelum N detik dari J
```

```
gunakan fungsi Detik2Jam dan Jam2Detik untuk membantu menyelesaikan
```

```
'''
```

```
Function Durasi(Jaw,Jakh)→ Integer
```

```
'''
```

```
mengembalikan selisih Jaw dengan Jakh
```

```
gunakan Jam2Detik untuk mendapatkan nilai awal dan akhir
```

```
jika akhir<awal maka mengembalikan None
```

```
selain itu mengembalikan selisih akhir dengan awal
```

```
'''
```