

1. TYPE BENTUKAN TIME

```
#Nama File : type time.py
#Deskripsi : Membuat tipe time beserta konstruktor dan selektornya
#Pembuat : Abyan A.
#Tanggal : 9 Oktober 2020

#DEFINISI TYPE
# type time : < j:integer[0...23],m:integer[0...59],d:integer[0...59] >
# {j:integer[0...23],m:integer[0...59],d:integer[0...59] adalah jam, meni, detik pada suatu time}

#DEFINISI DAN SPESIFIKASI KONSTRUKTOR
# makeTime : integer[0...23],integer[0...59],integer[0...59] --> time
# {makeTime(x,y,z) membentuk time dengan jam x, menit y, dan detik z}

#DEFINISI DAN SPESIFIKASI SELEKTOR DENGAN FUNGSI
# jam : time --> integer[0...23]
# {jam(D)menentukan nilai jam dari sebuah time}
# menit : time --> integer[0..59]
# {menit(D)menentukan nilai menit dari sebuah time}
# detik : time --> integer[0...59]
# {detik(D) menentukan nilai detik dari sebuah time}

#DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP TIME
# KonversiDetikToTime : integer>0 --> time
# {KonversiDetikToTime(x) mengkonversi detik x menjadi sebuah time}
# JumlahDetik : time --> detik
# {JumlahDetik(T) menghitung jumlah detik dari sebuah time}
# AddTime : 2 time --> time
# {AddTime(T1,T2) menjumlahkan 2 buah time menghasilkan sebuah time}
# AddTimeDetik : time, integer>0 --> time
# {AddTimeDetik(T,N) menjumlahkan sebuah time dengan detik N menghasilkan sebuah time}
#DEFINISI DAN SPESIFIKASI PREDIKAT
# isBefore : 2 time --> boolean
# {isBefore(T1,T2) bernilai benar jika T1 adalah sebelum T2}
# isAfter : 2 time --> boolean
# {isAfter(T1,T2) bernilai benar jika T1 adalah setelah T2}

#####

#REALISASI TYPE
class time:
    def __init__(self,a,b,c):
        self.j = a
        self.m = b
        self.d = c

#REAKISASI KONSTRUKTOR
def makeTime(j,m,d):
    return (j,m,d)

#REALISASI SELEKTOR
def jam(t):
    return t.j
def menit(t):
    return t.m
def detik(t):
    return t.d

#REALISASI OPERATOR TERHADAP TIME
def KonversiDetikToTime(N):
    a= 0 if N//3600 >= 24 else N//3600
    b= (N % 3600) // 60
    c=((N % 3600) % 60)
    return (a,b,c)

def jumlahdetik(T):
    return jam(T)*3600 + menit(T)*60 + detik(T)

def AddTime(t1,t2):
    return KonversiDetikToTime(jumlahdetik(t1)+jumlahdetik(t2))

def AddTimeDetik(T,N):
    return KonversiDetikToTime(jumlahdetik(T)+ N)

#REALISASI PREDIKAT
def isBefore(T1,T2):
    return(jumlahdetik(T1) < jumlahdetik(T2))

def isAfter(T1,T2):
    return(jumlahdetik(T1) > jumlahdetik(T2))
```

```
#####
#APLIKASI TYPE
A=time(23,11,30)
B=time(9,11,40)

#APLIKASI SELRKTOR DAN KONSTRUKTOR
print (jam(A))
print (menit(B))
print (menit(B))
print (makeTime(23,59,59))

#APLIKASI OPERATOR TERHADAP TYPE
print(KonversiDetikToTime(86500))
print(KonversiDetikToTime(4000))
print(jumlahdetik(A))
print(jumlahdetik(B))
print(AddTime(A,B))
print(AddTimeDetik(A,4000))

#APLIKASI PREDIKAT
print(isBefore(A,B))
print(isAfter(A,B))
```

TYPE TIME.py - E:\KULIAH\DAASPRO\FUNGSIONAL\TYPE TIME.py (3.7.4)
File Edit Format Run Options Window Help

```
#REALISASI OPERATOR TERHADAP TIME
def KonversiDetikToTime(N):
    a= 0 if N//3600 >= 24 else N//3600
    b= (N % 3600) // 60
    c=( (N % 3600) % 60)
    return(a,b,c)

def jumlahdetik(T):
    return jam(T)*3600 + menit(T)*60 + detik(T)

def AddTime(t1,t2):
    return KonversiDetikToTime(jumlahdetik(t1)+jumlahdetik(t2))

def AddTimeDetik(T,N):
    return KonversiDetikToTime(jumlahdetik(T)+ N)

#REALISASI PREDIKAT
def isBefore(T1,T2):
    return(jumlahdetik(T1) < jumlahdetik(T2))

def isAfter(T1,T2):
    return(jumlahdetik(T1) > jumlahdetik(T2))

#####
#APLIKASI TYPE
A=time(23,11,30)
B=time(9,11,40)

#APLIKASI SELRKTOR DAN KONSTRUKTOR
print (jam(A))
print (menit(B))
print (menit(B))
print (makeTime(23,59,59))

#APLIKASI OPERATOR TERHADAP TYPE
print(KonversiDetikToTime(86500))
print(KonversiDetikToTime(4000))
print(jumlahdetik(A))
print(jumlahdetik(B))
print(AddTime(A,B))
print(AddTimeDetik(A,4000))
```

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

```
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\KULIAH\DAASPRO\FUNGSIONAL\TYPE TIME.py =====
23
11
11
(23, 59, 59)
(0, 1, 40)
(1, 6, 40)
83490
33100
(0, 23, 10)
(0, 18, 10)
False
True
>>>
```

Ln: 17 Col: 4

Activate Windows
Go to Settings to activate Windows.

Ln: 89 Col: 24

2. TYPE BENTUKAN POINT

```
#Nama File : type point.py
#Deskripsi : membuat tipe point beserta konstruktor dan selektornya
#Pembuat : Abyan A.
#Tanggal : 9 Oktober 2020

#DEFINISI TYPE
# type point : <x=real, y=real>
# (x=real, y=real adalah sebuah point dimana x sebagai absis dan y sebagai ordinat)

#DEFINISI DAN SPESIFIKASI KONSTRUKTOR
# make_point : 2 real --> point
# (make_point(a,b) membentuk point dari a dan b dimana a sebagai absis dan b sebagai ordinat)

#DEFINISI DAN SPESIFIKASI SELEKTOR DENGAN FUNGSI
# absis : point --> real
# (absis(P) menentukan nilai absis pada point P)
# ordinat : point --> real
# (ordinat(P) menentukan nilai ordinat pada point P)

#DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP POINT
# gradien : 2 point --> integer
# (gradien(P1,P2) menghitung gradien suatu garis dari 2 buah point)
# isSejajar : 4 point --> integer
# (isSejajar(Q1,Q2,Q3,Q4) menentukan apakah gradien garis pertama(Q1,Q2) sama dengan gradien garis kedua(Q3,Q4))
# isTegakLurus : 4 point --> integer
# (isTegakLurus(R1,R2,R3,R4) menentukan apakah gradien garis pertama(R1,R2) tegak lurus dengan garis kedua(R3,R4))

#####

#REALISASI TYPE
class point:
    def __init__(self,a,b):
        self.x=a
        self.y=b

#REALISASI KONSTRUKTOR
def make_point(a,b):
    return (a,b)

#REALISASI SELEKTOR
def absis(P):
    return P.x

def ordinat(P):
    return P.y

#REALISASI OPERATOR TERHADAP POINT
def gradien(P1,P2):
    a= ordinat(P2)-ordinat(P1)
    b= absis(P2) - absis(P1)
    return (a//b)

def isSejajar(Q1,Q2,Q3,Q4):
    return gradien(Q1,Q2) == gradien(Q3,Q4)

def isTegakLurus(R1,R2,R3,R4):
    return gradien(R1,R2) * gradien(R3,R4) == -1

#####

#APLIKASI TYPE
P = point(3,6)
Q = point(5,7)
R = point(10,5)
S = point(4,7)

#APLIKASI OPERATOR
print (absis(P))
print (ordinat(Q))
print (make_point(2,4))
print (gradien(P,Q))
print (gradien(R,S))
print (isSejajar(P,Q,R,S))
print (isTegakLurus(P,Q,R,S))
```

```
#####  
#REALISASI TYPE  
class point:  
    def __init__(self,a,b):  
        self.x=a  
        self.y=b  
  
#REALISASI KONSTRUKTOR  
def make_point(a,b):  
    return (a,b)  
  
#REALISASI SELEKTOR  
def absis(P):  
    return P.x  
  
def ordinat(P):  
    return P.y  
  
#REALISASI OPERATOR TERHADAP POINT  
def gradien(P1,P2):  
    a= ordinat(P2)-ordinat(P1)  
    b= absis(P2) - absis(P1)  
    return (a//b)  
  
def isSejajar(Q1,Q2,Q3,Q4):  
    return gradien(Q1,Q2) == gradien(Q3,Q4)  
  
def isTegakLurus(R1,R2,R3,R4):  
    return gradien(R1,R2)* gradien(R3,R4) == -1  
  
#####  
#APLIKASI TYPE  
P = point(3,6)  
Q = point(5,7)  
R = point(10,5)  
S = point(4,7)  
  
#APLIKASI OPERATOR  
print (absis(P))  
print (ordinat(Q))  
print (make_point(2,4))
```

Python 3.7.4 (tags/v3.7.4:009359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: E:\KULIAH\DAISPRO\FUNGSIONAL\TYPE POINT.py =====

```
3  
7  
(2, 4)  
0  
-1  
False  
False  
>>>
```

Ln: 12 Col: 4

Activate Windows
Go to Settings to activate Windows.

Ln: 70 Col: 21