

01153 - ALGORITMA DAN STRUKTUR DATA 1

SUB PROGRAM

OLEH : ANDRI HERYANDI, M.T.



08

SUB PROGRAM

01153 - Algoritma dan Struktur Data 1

■ Part #1

- Definisi sub program
- Algoritma
 - Procedure
 - Pendeklarasian Procedure
 - Pemanggilan Procedure
 - Function
 - Pendeklarasian Function
 - Pemanggilan Function
- Python
 - Deklarasi Function
 - Pemanggilan Function
 - Urutan Penulisan Argument
 - Built-in Function

■ Part #2

- Recursive Function
- Passing Argument
- Default Value
- Module
 - Import Module
 - `import` Modules
 - `import` Modules `as`
 - `from` Modules `import`
 - Menggunakan Module
 - Install Module
 - UnInstall Module
 - Membuat Module



Oleh : Andri Heryandi, M.T.

01153 - ALGORITMA DAN STRUKTUR DATA 1

SUB PROGRAM (PART #1)

OLEH : ANDRI HERYANDI, M.T.



DEFINISI

01153 - Algoritma dan Struktur Data 1

- Sub Program adalah bagian dari suatu program yang dapat melakukan suatu pekerjaan tertentu.
- Sub program disebut juga sebagai module, sub rutin
- Sub program digunakan untuk “memecah” suatu program besar menjadi bagian-bagian yang lebih kecil yang lebih mudah untuk dikelola (manage). Kemampuan ini disebut sebagai **dekomposisi**. [lebih lanjut, silahkan explore istilah Computational Thinking]
- Sub program terdiri dari serangkaian instruksi (statement) yang mempunyai suatu kegunaan tertentu yang dipanggil dengan menggunakan namanya saja.
- Dalam dunia programming, dikenal 2 jenis sub program, yaitu :
 - Procedure : sub program yang berisi serangkaian instruksi khusus
 - Function : sub program yang berisi serangkaian instruksi khusus dan mengembalikan (return) suatu nilai.



DEFINISI

01153 - Algoritma dan Struktur Data 1

■ Keuntungan menggunakan subprogram

- Mempermudah membaca program, karena sekumpulan perintah bisa dibaca dari nama subprogramnya.
- Membuat program menjadi lebih terstruktur (*structured-code*)
- Memperpendek source code, karena aktivitas/pekerjaan yang sering dilakukan cukup disusun menjadi satu subprogram dan boleh digunakan/dipanggil berulang kali.
- Mudah dalam pemeliharaan (*maintainable*).



Oleh : Andri Heryandi, M.T.

CONTOH SUB PROGRAM

01153 - Algoritma dan Struktur Data 1

■ Contoh kasus :

- Buatlah sebuah program yang akan menerima input data dari pengguna dan kemudian melakukan pencarian nilai terbesar dari data-data tersebut dan kemudian menampilkan nilai terbesar tersebut.

■ Dekomposisi :

- Sub program untuk input data
- Sub program untuk pencarian nilai terbesar
- Sub program untuk menampilkan nilai terbesar



Oleh : Andri Heryandi, M.T.

PENULISAN SUB PROGRAM

01153 - Algoritma dan Struktur Data 1

■ Di mana sub program ditulis?

1. Sub program ditulis di luar program utama.
2. Sub program ditulis sebelum bagian program yang memanggilnya/menggunakannya.
3. Jika program utama memiliki pendefinisian identifier global, maka :
 1. Tulislah sub program sebelum pendefinisian identifier global, jika anda tidak ingin identifier global dikenal di dalam sub program.
 2. Tulislah sub program setelah pendefinisian identifier global, jika anda ingin identifier global bisa dikenal di dalam sub program.



Oleh : Andri Heryandi, M.T.

STRUKTUR PROGRAM YANG MENGGUNAKAN SUB PROGRAM

01153 - Algoritma dan Struktur Data 1

Program Nama_Program

{I.S :}

{F.S :}

Kamus (Global):

{deklarasi identifier global, jika ada}

Procedure nama_prosedur(<daftar parameter>)

{I.S. :}

{F.S. :}

Kamus: {kamus lokal/deklarasi lokal}

{deklarasi identifier lokal}

Algoritma:

{aksi-aksi yang akan dilakukan oleh prosedur}

EndProcedure

Function nama_function(<daftar parameter>)→Tipe_Return

{I.S. :}

{F.S. :}

Kamus: {kamus lokal/deklarasi lokal}

{deklarasi identifier lokal}

Algoritma:

{aksi-aksi yang akan dilakukan oleh function}

EndFunction

Algoritma (Program Utama):

{instruksi_program_utama}



ALUR PROGRAM YANG MENGGUNAKAN SUB PROGRAM

01153 - Algoritma dan Struktur Data 1

Algoritma (Program Utama):

↓
instruksi_1
instruksi_2
↓
pemanggilan_procedure()
←
↓
instruksi_3
instruksi_4
↓
val=pemanggilan_function()
←
↓
instruksi_5
instruksi_...

Procedure nama_procedure()

↓
instruksi_procedure
...
↓
instruksi_procedure
End procedure

function nama_function()

↓
instruksi_function
...
↓
return value
End function

STRUKTUR PENGULANGAN

01153 - Algoritma dan Struktur Data 1

Algoritma



Oleh : Andri Heryandi, M.T.

PROCEDURE

OLEH : ANDRI HERYANDI, M.T.



PENGERTIAN PROCEDURE

01153 - Algoritma dan Struktur Data 1

- Procedure adalah sekumpulan perintah yang digunakan untuk menjalankan suatu pekerjaan tertentu
- Procedure HANYA mengeksekusi perintah-perintah tapi TIDAK mereturnkan data/nilai.



Oleh : Andri Heryandi, M.T.

PENGERTIAN PROCEDURE

01153 - Algoritma dan Struktur Data 1

■ Struktur Procedure

- **Header Procedure.** Header procedure terdiri dari keyword **procedure**, nama procedure dan daftar parameter. Parameter-parameter yang ditulis di dalam header procedure disebut sebagai parameter formal (**formal parameter**). Parameter sifatnya optional. Anda boleh membuat procedure tanpa parameter.
- **Deklarasi Lokal.** Pendefinisian identifier (tipe, konstanta, variable) yang hanya berlaku di dalam badan procedure. Deklarasi lokal sifatnya adalah optional.
- **Badan Procedure.** Berisi instruksi-instruksi yang akan dieksekusi. Bagian ini adalah bagian yang wajib ditulis.



STRUKTUR PROCEDURE

01153 - Algoritma dan Struktur Data 1

■ Notasi algoritmik

Procedure nama_prosedur(<daftar parameter>)

{I.S. :}

{F.S. :}

Kamus: {kamus lokal/deklarasi lokal}

 {deklarasi identifier lokal}

Algoritma:

 {aksi-aksi yang akan dilakukan oleh prosedur}

EndProcedure



STRUKTUR PROCEDURE

01153 - Algoritma dan Struktur Data 1

■ Contoh Procedure

Procedure tulis_nama()

Algoritma:

output("Nama Saya Cecep Gorbachev")

output("Saya tinggal di Bandung")

EndProcedure

Procedure tulis_nama_berulang(N:integer)

Kamus:

i:integer

Algoritma:

for i ← 1 to N do

output(i, "Nama Saya Cecep Gorbachev")

end for

EndProcedure



PEMANGGILAN PROCEDURE

01153 - Algoritma dan Struktur Data 1

- Pemanggilan procedure bisa dimana saja, asalkan procedure yang akan dipanggil telah didefinisikan sebelumnya (ditulis di atas bagian pemanggilan).
- Pemanggilan procedure boleh di program utama atau di dalam procedure / function lain.
- Procedure dipanggil dengan menggunakan nama procedurennya.
- Pemanggilan procedure dilakukan dengan memanggil nama procedurennya dilengkapi dengan parameternya (kalau ada).
- Konsistenkan pemanggilan nama procedure dengan nama procedure yang dibuat. Ada Bahasa pemrograman yang pemanggilan procedurennya bersifat case sensitive dan ada juga yang case insensitive.
- Nilai yang dikirim ke parameter formal dari suatu procedure disebut sebagai **parameter aktual** (***actual parameter***). Parameter aktual disebut juga sebagai ***argument***. Tapi dalam banyak referensi, banyak juga yang menyebut parameter adalah argument (atau pun sebaliknya).
- Parameter aktual ditulis setelah nama procedure. Parameter aktual diawali dengan tanda buka kurung dan ditutup dengan tanda tutup kurung. Jika memiliki lebih dari 1 parameter, maka pemisah parameternya menggunakan tanda koma.



PEMANGGILAN PROCEDURE

01153 - Algoritma dan Struktur Data 1

■ Contoh Pemanggilan Procedure

Program Demo_Procedure

Kamus:

Procedure tulis_nama()

Algoritma:

```
output("Nama Saya Cecep Gorbachev")  
output("Saya tinggal di Bandung")
```

EndProcedure

Procedure tulis_nama_berulang(N:integer)

Kamus:

i:integer

Algoritma:

```
for i ← 1 to N do  
    output(i, "Nama Saya Cecep Gorbachev")  
end for
```

EndProcedure

Algoritma (Program Utama) :

```
tulis_nama()  
tulis_nama_berulang(10)
```

Hasil eksekusi

```
Nama Saya Cecep Gorbachev  
Saya tinggal di Bandung  
1Nama Saya Cecep Gorbachev  
2Nama Saya Cecep Gorbachev  
3Nama Saya Cecep Gorbachev  
4Nama Saya Cecep Gorbachev  
5Nama Saya Cecep Gorbachev  
6Nama Saya Cecep Gorbachev  
7Nama Saya Cecep Gorbachev  
8Nama Saya Cecep Gorbachev  
9Nama Saya Cecep Gorbachev  
10Nama Saya Cecep Gorbachev
```



FUNCTION

OLEH : ANDRI HERYANDI, M.T.



PENGERTIAN FUNCTION

01153 - Algoritma dan Struktur Data 1

- Function adalah sekumpulan perintah yang digunakan untuk menjalankan suatu pekerjaan tertentu dan mengembalikan/memberikan/menghasilkan (return) sebuah nilai.
- Function TIDAK HANYA mengeksekusi perintah-perintah saja tapi function harus mereturnkan data/nilai.
- Dalam Bahasa Indonesia, function disebut sebagai fungsi.
- Konsep function di pemrograman diambil dari ilmu matematika, contoh :
 - Pernahkah anda melihat rumus $F(x)=x^2+5$. Ini artinya kalau kita menemukan rumus $A=F(3)$ maka rumus yang ada dalam fungsi tersebut akan dieksekusi sehingga akan menghasilkan 3^2+5 yaitu 14. Ini artinya $F(3)$ akan menghasilkan/mereturnkan nilai 14 jika X -nya adalah 3.
 - Contoh lain : $F(x,y)= 2x+y$, maka kalau ada ada $F(5,2)$ akan menghasilkan $2*5+2 = 12$.
- Tetapi di dunia pemrograman, nilai tidak selalu berupa angka. Begitu juga dengan function di dunia pemrograman yang dapat mereturnkan/menghasilkan nilai bertipe angka, karakter, string, Boolean, atau tipe data lain



STRUKTUR FUNCTION

01153 - Algoritma dan Struktur Data 1

- **Header Function.** Header function terdiri dari keyword **function**, nama function, daftar parameter formal dan tipe data nilai yang akan di-return-kan. Parameter sifatnya optional. Anda boleh membuat function tanpa parameter.
- **Deklarasi Lokal.** Pendefinisian identifier (tipe, konstanta, variable) yang hanya berlaku di dalam badan function. Deklarasi lokal sifatnya adalah optional. Anda hanya membuatnya jika di dalam function membutuhkan identifier khusus yang terpisah dari identifier global.
- **Badan Function.** Berisi instruksi-instruksi yang akan dieksekusi. Bagian ini adalah bagian yang wajib ditulis. Berbeda dengan procedure, di dalam badan function harus ada baris yang me-return-kan nilai dari functionnya (yaitu **return nilai_return**).



STRUKTUR FUNCTION

01153 - Algoritma dan Struktur Data 1

■ Notasi algoritmik

Function nama_function(<daftar parameter>) → tipe_data_return

{I.S. :}

{F.S. :}

Kamus: {kamus lokal/deklarasi lokal}

{deklarasi identifier lokal}

Algoritma:

{aksi-aksi yang akan dilakukan oleh function}

return nilai_return // atau return nilai_return

EndFunction



STRUKTUR FUNCTION

01153 - Algoritma dan Struktur Data 1

■ Contoh Function

Function jumlah_deret(N:integer) → integer

Kamus :

i, jumlah : integer

Algoritma:

jumlah ← 0

for i ← 1 to N do

 jumlah ← jumlah + i

return jumlah

EndFunction

Function besar_bunga(awal, persen_bunga:real, waktu:integer) → real

Algoritma:

return awal * (1+persen_bunga/100) ^ waktu

EndFunction



PEMANGGILAN FUNCTION

01153 - Algoritma dan Struktur Data 1

- Pemanggilan function bisa dimana saja, asalkan function yang akan dipanggil telah didefinisikan sebelumnya (ditulis di atas bagian pemanggilan).
- Pemanggilan function boleh di program utama atau di dalam procedure / function lain.
- Function dipanggil dengan menggunakan namanya dilengkapi dengan parameternya.
- ***HARAP DIINGAT***. Function mewaliki sebuah nilai, sehingga anda dapat memperlakukan hasil pemanggilan function seakan-akan itu adalah variable/nilai, dimana :
 - Nilai return dari pemanggilan function bisa disimpan dulu ke variable kemudian variable tersebut digunakan, atau pun
 - Nilai return dari pemanggilan function bisa langsung digunakan (misalkan langsung ditulis ke layar, dikirim sebagai parameter ke procedure/parameter lain)



PEMANGGILAN FUNCTION

01153 - Algoritma dan Struktur Data 1

■ Contoh Pemanggilan Function

Program Demo_Function

Kamus:

total : integer
bunga_didapatkan : real

Function jumlah_deret(N:integer) → integer

Kamus :

i, jumlah : integer

Algoritma:

```
jumlah ← 0
for i ← 1 to N do
    jumlah ← jumlah + i
return jumlah
```

EndFunction

Function besar_bunga(awal, persen_bunga:real, waktu:integer) → real

Algoritma:

```
return awal * (1+persen_bunga/100) ^ waktu
```

EndFunction

Algoritma (Program Utama) :

```
{nilai return disimpan ke variable}
total ← jumlah_deret(10)
output("Jumlah deret 1 s.d 10 : ", total)
{nilai return function dioutputkan langsung}
output("Jumlah deret 1 s.d 5 : ", jumlah_deret(5))
{nilai return disimpan ke variable}
bunga_didapatkan ← besar_bunga(100000,10,2)
output("Bunga Didapatkan : ", bunga_didapatkan)
{nilai return function dioutputkan langsung}
output("Bunga Didapatkan : ", besar_bunga(200000,10,1))
```

Hasil eksekusi

```
Jumlah deret 1 s.d 10 : 55
Jumlah deret 1 s.d 5 : 15
Bunga Didapatkan : 121000
Bunga Didapatkan : 220000
```



STRUKTUR PENGULANGAN

01153 - Algoritma dan Struktur Data 1

Python



Oleh : Andri Heryandi, M.T.

SUB PROGRAM DI PYTHON

01153 - Algoritma dan Struktur Data 1

- Dalam Bahasa Python sub program (baik procedure atau pun function) diimplementasikan dalam bentuk function.
- Untuk membedakan procedure dengan function di python yaitu dengan melihat nilai returnnya. Jika nilai returnnya **NoneType**, bisa disebut bahwa function tersebut adalah procedure.

- Sintak pembuatan function :

```
def nama_function([parameter]):  
    badan_function  
    return value_yang_direturnkan
```

- Keterangan :

- **def** adalah keyword untuk membuat function.
- Parameter sifatnya optional. Jika diwajibkan, parameter ditulis sesuai urutan dalam deklarasi function.
- Badan function diisi dengan statement-statement yang diinginkan
- **return** digunakan untuk mendefinisikan nilai yang akan dikembalikan. Jika anda membuat procedure, maka return tidak usah ditulis (akan mereturnkan None).



CONTOH DEKLARASI DAN PEMANGGILAN FUNCTION

01153 - Algoritma dan Struktur Data 1

■ Contoh Pendeklarasian dan Pemanggilan Function

```
# deklarasi identifiier global
nama = "Cecep Gorbachevs"

def tulis_nama():
    print(f>Nama Saya {nama}")

def tulis_nama_berulang(N):
    for i in range(1,N+1):
        print(f">{i}Nama Saya {nama}")

def jumlah_deret(N):
    jumlah = 0
    for i in range(1,N+1):
        jumlah = jumlah + i
    return jumlah

def besar_bunga(awal, persen_bunga, waktu):
    return awal * (1 + persen_bunga/100) ** waktu
```

```
# Program Utama
tulis_nama()
tulis_nama_berulang(5)

total_deret = jumlah_deret(10)
print(f>Total Deret 1 s.d 10 : {total_deret}")
print("Total Deret 1 s.d 5 : {}".format(jumlah_deret(5)))

bunga_didapatkan = besar_bunga(100000,10,3)
print("Tabungan setelah 3 bulan {:.2f}".format(bunga_didapatkan))
```

Hasil eksekusi

```
Nama Saya Cecep Gorbachevs
1Nama Saya Cecep Gorbachevs
2Nama Saya Cecep Gorbachevs
3Nama Saya Cecep Gorbachevs
4Nama Saya Cecep Gorbachevs
5Nama Saya Cecep Gorbachevs
Total Deret 1 s.d 10 : 55
Total Deret 1 s.d 5 : 15
Tabungan setelah 3 bulan 133100.00
```



URUTAN ARGUMENT DALAM PEMANGGILAN FUNCTION

01153 - Algoritma dan Struktur Data 1

- Dalam Bahasa python, ada 2 cara penulisan urutan argument ketika pemanggilan function yaitu :
 - Berdasarkan posisi (positional argument), yaitu penulisan urutan argument berdasarkan posisi parameter dalam pendefinisian/deklarasi function.
 - Berdasarkan keyword/nama (keyword (named) argument), yaitu penyulisan urutan argument berdasarkan nama dari parameter.

```
def getBunga(awal, persen_bunga, waktu):  
    return awal * (1 + persen_bunga/100) ** waktu
```

```
# positional argument
```

```
hasil = getBunga(100000, 10, 5)  
print("1. Bunga : {:.2f}".format(hasil))
```

```
#keyword argument
```

```
print("2. Bunga : {:.2f}".format(getBunga(waktu=5, awal=100000, persen_bunga=10)))  
print("3. Bunga : {:.2f}".format(getBunga(awal=100000,waktu=5,persen_bunga=10)))  
print("4. Bunga : {:.2f}".format(getBunga(persen_bunga=10,awal=100000,waktu=5)))
```

Oien : Anari Heryandi, M.I.



BUILT-IN FUNCTION DI PYTHON

01153 - Algoritma dan Struktur Data 1

- Built-in function adalah function-function bawaan dari python yang bisa anda gunakan tanpa melakukan import terlebih dahulu.
- Daftar built-in function (Ref. <https://docs.python.org/3.8/library/functions.html>)

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min() ✓	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted() ✓
bin()	enumerate()	input() ✓	oct()	staticmethod()
bool()	eval()	int() ✓	open()	str() ✓
breakpoint()	exec()	isinstance()	ord()	sum() ✓
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float() ✓	iter()	print() ✓	tuple()
callable()	format() ✓	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	import _()
complex()	hasattr()	max() ✓	round()	

CONTOH PENGGUNAAN BUILT-IN FUNCTION

01153 - Algoritma dan Struktur Data 1

```
print("Nilai Absolut : ", abs(-5))
print("Nilai Terbesar : ",max([4,7,5,9,2]))
print("Nilai Terkecil : ",min([4,7,5,9,2]))
print("Total Data : ",sum([4,7,5,9,2]))
print("Banyak Data : ",len([4,7,5,9,2]))
print("Urutan Ascending : ",sorted([4,7,5,9,2]))
print("Urutan Descending : ",sorted([4,7,5,9,2],reverse=True))
print("Integer dari \"123\" : ", int("123"))
print("Biner 1234 : ",bin(1234))
print("Octal 1234 : ",oct(1234))
print("Hexadesimal 1234 : ",hex(1234))
print("Eksekusi Rumus \"1+3*5\" : ",eval("1+3*5"))
```

Hasil eksekusi

```
Nilai Absolut : 5
Nilai Terbesar : 9
Nilai Terkecil : 2
Total Data : 27
Banyak Data : 5
Urutan Ascending : [2, 4, 5, 7, 9]
Urutan Descending : [9, 7, 5, 4, 2]
Integer dari "123" : 123
Biner 1234 : 0b10011010010
Octal 1234 : 0o2322
Hexadesimal 1234 : 0x4d2
Eksekusi Rumus "1+3*5" : 16
```



Latihan

LATIHAN 1

APLIKASI PERHITUNGAN LUAS DAN KELILING

01153 - Algoritma dan Struktur Data 1

- **Buatlah aplikasi yang dapat melakukan perhitungan luas dan keliling dari bangun 2 dimensi berikut :**
 - Lingkaran
 - Bujursangkar
 - Persegipanjang
 - Segitiga
- **Buatlah dalam bentuk menu yang menawarkan pilihan bangun apa yang ingin dihitung.**
- **Susunlah sub program untuk setiap perhitungan yang tersedia.**
- **Setiap pengguna memilih pilihan, maka akan mengeksekusi sub program yang sesuai.**



LATIHAN 1

APLIKASI PERHITUNGAN LUAS DAN KELILING

01153 - Algoritma dan Struktur Data 1

■ Contoh Eksekusi :

Aplikasi Perhitungan Luas dan Keliling

-
1. Lingkaran
 2. Bujursangkar
 3. Persegipanjang
 4. Segitiga
 0. Keluar
-

Pilihan anda ? **1**

Perhitungan Luas dan Keliling Lingkaran

Radius : **5**
Luas Lingkaran : 78.54 [output]
Keliling Lingkaran : 31.42 [output]

Aplikasi Perhitungan Luas dan Keliling

-
1. Lingkaran
 2. Bujursangkar
 3. Persegipanjang
 4. Segitiga
 0. Keluar
-

Pilihan anda ? **2**

Perhitungan Luas dan Keliling Persegipanjang

Panjang : **5**
Lebar : **4**
Luas Persegipanjang : 20
Keliling Persegipanjang : 18



LATIHAN 2

FUNCTION TAHUN KABISAT

01153 - Algoritma dan Struktur Data 1

- **Buatlah sebuah function yang akan memeriksa suatu tahun apakah tahun kabisat atau tidak.**
- **Parameter**
 - Tahun : integer
- **Return status kabisat (True/False) : Bool**
- **Contoh pemanggilan function :**
 - kabisat(2004) # akan mereturnkan True
 - kabisat(1900) # akan mereturnkan False
 - kabisat(1998) # akan mereturnkan True
 - kabisat(2000) # akan mereturnkan True
 - kabisat(1995) # akan mereturnkan False



LATIHAN 3

FUNCTION TANGGAL AKHIR BULAN

01153 - Algoritma dan Struktur Data 1

- **Buatlah sebuah function yang akan mereturnkan tanggal terakhir di suatu bulan**
- **Parameter**
 - Tahun,bulan : integer;
- **Return**
 - tanggal akhir bulan : bilangan bulat.
- **Contoh pemanggilan function :**
 - akhirBulan(2004,2) # akan mereturnkan 29 karena tahun kabisat
 - akhirBulan(2020,1) # akan mereturnkan 31
 - akhirBulan(2006,2) # akan mereturnkan 28 karena 2006 bukan tahun kabisat.



LATIHAN 4

FUNCTION MEMVALIDASI TANGGAL

01153 - Algoritma dan Struktur Data 1

- **Buatlah suatu function yang memvalidasi (True/False) dari suatu tanggal. Valid tidaknya suatu tanggal sesuaikan dengan aturan tanggal.**
- **Parameter**
 - tahun, bulan, tanggal : integer
- **Return**
 - status_valid : Boolean
- **Contoh pemanggilan :**
 - validasiTanggal(2020,12,20) # 20 Desember 2020, mereturnkan True (karena valid)
 - validasiTanggal(2020,4,31) # 31 April 2020, mereturnkan False (karena April hanya sampai 30)
 - validasiTanggal(2020,12,32) # 32 Desember 2020, mereturnkan False
 - validasiTanggal(1990,13,20) # mereturnkan False, karena bulan tidak valid
 - validasiTanggal(2008,2,29) # 10 Februari 2008, mereturnkan True
 - validasiTanggal(2010,2,29) # 29 Februari 2010, mereturnkan False (bukan kabisat, february hanya sampai 28)



LATIHAN 5

(PERHITUNGAN HARI)

01153 - Algoritma dan Struktur Data 1

- Buatlah suatu function untuk menghitung hari ke berapa sejak 01 Januari 1900.
- Tanggal 01 Januari 1900 menghasilkan angka 1, tanggal 02 Januari 1900 menghasilkan angka 2, dst.
- Parameter : Tahun, bulan, tanggal (integer)
- Return : Banyak hari sejak 01 Januari 1900 (integer) jika tanggal valid, dan mereturnkan None jika tanggal tidak valid
- Contoh Pemanggilan :
 - HariKe(1900,1,1) # mereturnkan 1
 - HariKe(1900,1,2) # mereturnkan 2
 - HariKe(1900,2,1) # mereturnkan 32
 - HariKe(1945,8,17) # mereturnkan 16665
 - HariKe(2021,2,29) # mereturnkan None
 - HariKe(2021,11,10) # mereturnkan 44509



FORUM DISKUSI

01153 - Algoritma dan Struktur Data 1



LMS UNIKOM

<https://lms.unikom.ac.id>



**Group Whatsapp
Perkuliahahan**



Youtube Playlist

<https://unikom.id/YT-ASD1>



Oleh : Andri Heryandi, M.T.