

## CHAPTER II

### *Pemrograman II*



Andi Nur Utari Aminuddin

1184039

D4 TI 2B

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

*Politeknik Pos Indonesia*

2019

# DAFTAR ISI

<b>I</b>	<b>Pemrograman Dasar</b>	<b>1</b>
1.1	Jenis-Jenis Variabel dan Cara Pemakaian Variabel Tersebut di Kode Pyhton . . . . .	1
1.1.1	Jenis-Jenis Variabel . . . . .	1
1.1.2	Cara pemakaian variabel di pyhton . . . . .	1
1.1.3	Tipe data . . . . .	3
1.2	Bagaimana kode untuk meminta input dari user dan output ke layar .	4
1.3	Operator dasar aritmatika dan bagaimana mengubah string ke integer dan integer ke string . . . . .	4
1.3.1	Operator dasar aritmatika . . . . .	4
1.3.2	Mengubah tipe data . . . . .	6
1.4	Jenis-Jenis Perulangan . . . . .	6
1.5	Jenis-jenis kondisi . . . . .	7
1.6	Jenis-jenis error yang ditemukan dalam mengerjakan tugas ini . . . .	9
1.7	Cara memakai Try Except . . . . .	10
1.8	Keterampilan Pemrograman . . . . .	11
1.9	Keterampilan Penanganan Error . . . . .	14
1.9.1	Jenis-Jenis Error dan Penanganannya . . . . .	14
1.9.2	Penanganan Error . . . . .	15

# DAFTAR GAMBAR

I.1	Variabel Global . . . . .	2
I.2	Variabel Lokal . . . . .	2
I.3	Tipe data . . . . .	3
I.4	Input dan Output . . . . .	4
I.5	Pengurangan . . . . .	4
I.6	Pertambahan . . . . .	4
I.7	Perkalian . . . . .	5
I.8	Pembagian . . . . .	5
I.9	Modulus . . . . .	5
I.10	Perpangkatan . . . . .	5
I.11	Pembulatan . . . . .	6
I.12	Kode mengubah tipe data string ke integer . . . . .	6
I.13	Kode mengubah tipe data integer ke string . . . . .	6
I.14	Kode perulangan while . . . . .	6
I.15	Kode perulangan for . . . . .	7
I.16	Kode perulangan nested . . . . .	7
I.17	Kode kondisi if . . . . .	7
I.18	Kode Kondisi Ifelse . . . . .	7
I.19	Kode Kondisi Elif . . . . .	8
I.20	Kode Kondisi dalam Kondisi . . . . .	8
I.21	Cara menggunakan Try Except . . . . .	10
I.22	Nomor1 . . . . .	11
I.23	Nomor2 . . . . .	11
I.24	Nomor3 . . . . .	11
I.25	Nomor4 . . . . .	11
I.26	Nomor5 . . . . .	12
I.27	Nomor6 . . . . .	12
I.28	Nomor7 . . . . .	12

I.29	Nomor8 . . . . .	13
I.30	Nomor9 . . . . .	13
I.31	Nomor10 . . . . .	13
I.32	Nomor11 . . . . .	14
I.33	Penanganan Error . . . . .	15

# DAFTAR TABEL

# BAB I

## Pemrograman Dasar

### 1.1 Jenis-Jenis Variabel dan Cara Pemakaian Variabel Tersebut di Kode Python

#### 1.1.1 Jenis-Jenis Variabel

Variabel adalah sebuah tempat untuk menampung data pada memori dimana variabel dapat menampung sebuah nilai yang bersifat dinamis sesuai kebutuhan program. Sedangkan tipe data yaitu jenis data yang tersimpan pada variabel. Pada python terdapat dua variabel yaitu variabel lokal dan global.

1. variabel global yaitu variabel yang di definisikan diluar fungsi. dimana variabel global ini dapat dikenali di semua lingkungan program yang dibuat
2. variabel lokal yaitu variabel yang dapat diakses dari dalam fungsi dimana variabel tersebut didefinisikan.

#### 1.1.2 Cara pemakaian variabel di python

1. Variabel Global variabel global bisa diakses dari seluruh tempat dimanapun di dalam program. Untuk menggunakan variabel global dalam suatu fungsi, variabel global harus dideklarasikan didalam fungsi. Global dapat diakses oleh fungsi apa pun, tetapi hanya dapat dimodifikasi jika Anda secara langsung di deklarasikan dalam fungsi dengan kata kunci 'global' di dalam fungsi. Deklarasi eksplisit dengan menggunakan global (variable name) di dalam suatu fungsi.

```

1 # membuat variabel global
2 a = "Tari"
3
4 def help():
5     # ini variabel lokal
6     nama = "Lisa"
7     # mengakses variabel lokal
8     print (nama)
9     print (a)
10 # mengakses variabel global
11 print (a)
12 # memanggil fungsi help()
13 help()

```

Gambar I.1: Variabel Global

2. Variabel Lokal variabel lokal hanya bisa diakses dari dalam fungsi di mana ia di definisikan. Jika ada variabel yang dideklarasikan didalam suatu fungsi, variabel ini tidak ada kaitannya dengan variabel lain dengan nama yang sama diluar fungsi, dengan kata lain nama variabel hanya lokal untuk fungsi.

```

1 total = 0
2 # Variabel global
3 # Definisi fungsi
4 def sum( arg1, arg2 ):
5     """Menambahkan variabel dan mengembalikan hasilnya."""
6     total = arg1 + arg2;
7     # total di sini adalah variabel lokal
8     print ("Di dalam fungsi nilai total : ", total)
9     return total
10
11 # Pemanggilan fungsi sum
12 sum( 10, 20 )
13 print ("Di luar fungsi, nilai total : ", total ) |

```

Gambar I.2: Variabel Lokal

### 1.1.3 Tipe data

```
1 #Boolean
2 answer = 25 < 10
3 print(answer)
4 answer = 25 > 10
5 print(answer)
6 answer = "A" == "a"
7 print(answer)
8
9 #String
10 nama = "Andi Nur Utari Aminuddin"
11 print (nama)
12
13 #Integer
14 var = 10*25
15 print (var)
16
17 #Set
18 angka = {1,2,3,4,5}
19 print (angka)
20
21 #float
22 angka = 10.25
23 print (angka)
24
25 #heksadesimal
26 angka = 10*0.5
27 print (angka)
28
29 #complex
30 a = 2000j
31 print (a)
32
33 #dictionary
34 d = {1:'satu', 2:'dua', 'tiga':3}
35 print(type(d))
36 print("d[1] = ", d[1])
37 print("d['tiga'] = ", d['tiga'])
38 # Error
39 print("d[3] = ", d[3])

41 #tuple
42 a = ( 1,2,3,4,5 )
43 print (a)
44 print (a[0])
45
46 #list
47 b = ( 1,2,3,4,5 )
48 print (b)
49 print (b[0])
```

Gambar I.3: Tipe data



## 1.2 Bagaimana kode untuk meminta input dari user dan output ke layar

```
1 #Input dan Output
2 nama = input ("Masukkan nama anda :")
3 npm = input ("Masukkan npm anda :")
4 print ('Nama saya' + nama)
5 print ('npm saya' + npm)
```

Gambar I.4: Input dan Output

## 1.3 Operator dasar aritmatika dan bagaimana mengubah string ke integer dan integer ke string

### 1.3.1 Operator dasar aritmatika

1. Pengurangan, operator ini digunakan untuk melakukan operasi pengurangan

```
1 #pengurangan
2 a = 25
3 b = 10
4 hasil = a-b
5 print (hasil)
```

Gambar I.5: Pengurangan

2. Pertambahan, operator ini digunakan untuk melakukan operasi pertambahan

```
1 #pertambahan
2 a = 25
3 b = 10
4 hasil = a+b
5 print (hasil)
6 .
```

Gambar I.6: Pertambahan

3. Perkalian, operator ini digunakan untuk melakukan operasi perkalian

```
1 #perkalian
2 a = 25
3 b = 10
4 hasil = a*b
5 print (hasil)
6
```

Gambar I.7: Perkalian

4. Pembagian, operator ini digunakan untuk melakukan operasi pembagian

```
1 #pembagian
2 a = 2000
3 b = 25
4 hasil = a/b
5 print (hasil)
6
```

Gambar I.8: Pembagian

5. Modulus, operator ini digunakan untuk melakukan operasi modulus

```
1 #modulus
2 a = 17
3 b = 2
4 hasil = a%b
5 print (hasil)
6
```

Gambar I.9: Modulus

6. Perpangkatan, operator ini digunakan untuk melakukan operasi perpangkatan

```
1 #perpangkatan
2 a = 25
3 b = 2
4 hasil = a**b
5 print (hasil)
6
```

Gambar I.10: Perpangkatan

7. Pembulatan, operator ini digunakan untuk melakukan operasi pembulatan pada hasil pembagian

```
1 #pembulatan
2 a = 25
3 b = 7
4 hasil = a//b
5 print (hasil)
6
```

Gambar I.11: Pembulatan

### 1.3.2 Mengubah tipe data

1. Mengubah string ke integer

```
1 #mengubah string ke integer
2 a_str = '25'
3 b_int = int (a_str)
4 print (b_int)
5
```

Gambar I.12: Kode mengubah tipe data string ke integer

2. Mengubah integer ke string

```
1 #mengubah integer ke string
2 a_int = 100
3 b_str = str (a_int)
4 print (b_str)
5
```

Gambar I.13: Kode mengubah tipe data integer ke string

## 1.4 Jenis-Jenis Perulangan

1. While Looping

```
1 #while perulangan
2 a = 1
3 while (a<25):
4     print (a)
5     a +=1
6
```

Gambar I.14: Kode perulangan while

## 2. for Looping

```
1 #forloop
2 bilangan = {0,1,2,3}
3 for b in bilangan:
4     print (bilangan)
```

Gambar I.15: Kode perulangan for

## 3. Nested Looping

```
1 #Nested Loop
2 a = 1
3 while ( a < 3 ) :
4     b = 1
5     while ( b <= 3 ) :
6         print ( ' abc ' )
7         b+= 1
8     a+= 1
```

Gambar I.16: Kode perulangan nested

# 1.5 Jenis-jenis kondisi

## 1. Kondisi IF

```
1 #kondisi IF
2 a = 9
3 if a > 0:
4     print (a,"adalah angka terakhir npm saya")
5
```

Gambar I.17: Kode kondisi if

## 2. Kondisi Ifelse

```
1 # I f E l s e
2 a = 25
3 if a > 0 :
4     print (a, "adalah bilangan positif")
5 else:
6     print (a, "adalah bilangan negatif")
```

Gambar I.18: Kode Kondisi Ifelse

### 3. Kondisi Elif

```
1 # E l i f
2 a = -10
3 if a < 0 :
4     print (a, "adalah bilangan negatif")
5 elif a > 0 :
6     print (a, "adalah bilangan positif")
7 else:
8     print (a, "adalah bilangan heksadesimal")
```

Gambar I.19: Kode Kondisi Elif

### 4. Kondisi dalam Kondisi

```
1 #kondisi didalam kondisi
2 a = -25
3 if a < 0 :
4     print (a, "adalah bilangan negatif")
5     if a < 0 :
6         print (a, "adalah bilangan negatif yang lebih kecil dari -5")
7     else:
8         print (a, "adalah bilangan kurang dari 100")
9 elif a > 0:
10    print ( a,"adalah bilangan positif")
11 else:
12    print (a, "adalah bilangan prima")
```

Gambar I.20: Kode Kondisi dalam Kondisi

## 1.6 Jenis-jenis error yang ditemukan dalam mengerjakan tugas ini

1. Syntax Error Syntax error adalah suatu keadaan atau kondisi ketika ada kesalahan penulisan kode pada program python hal ini menyebabkan program tidak dapat dijalankan. contohnya kesalahan pemberian titik dua atau tanda kutip. Output pemberitahuan error nya yaitu invalid syntax. Yang harus dilakukan saat terjadi syntax error pada kode program yaitu memperbaiki penulisan kodenya.
2. Name Error Name error, yaitu exception yang muncul ketika melakukan eksekusi pada suatu program terhadap lokal name dan global name tidak terdefinisi. error ini terjadi saat pemanggilan variabel yang tidak di definisikan atau memanggil sebuah function yang tidak ada. Output pemberitahuan error nya yaitu name 'a' is not defined. Untuk mengatasi terjadi name error yaitu dengan memastikan variabel dan function yang akan dipanggil benar-benar ada dalam kode program dan tidak terjadi kesalahan penulisan.
3. Type Error Type Error, yaitu suatu keadaan yang terjadi ketika melakukan eksekusi pada suatu operasi atau fungsi yang tipe datanya berbeda atau tidak sesuai dengan operasi yang akan dilakukan. Contoh kasusnya pada kesalahan tipe data antara string dan integer, kesalahan dalam input list, tuple dan dictionary. Cara penanganannya yaitu dengan mengkonversi variabel yang digunakan sesuai dengan tipe datanya.
4. Indentation error Indentation error, yaitu tulisan kode program yang menjerok. indentation error akan terjadi ketika mengetik kode program namun tidak memperhatikan indentasinya. Jika terjadi indentasi maka program akan error. cara mengatasinya yaitu memperhatikan indentasi saat menuliskan suatu program.

## 1.7 Cara memakai Try Except

Salah satu bentuk penanganan error pada program python yaitu dengan menggunakan statement Try Except

```
1 #Try Except
2 nama = 'tari'
3 try :
4     print (nama)
5 except :
6     print ("Terdapat kesalahan pada kode program anda")
7 finally:
8     print ("Guanakan Try except untuk mengatasi error")
```

Gambar I.21: Cara menggunakan Try Except

## 1.8 Keterampilan Pemrograman

### 1. Jawaban soal nomor 1

```
1 #Menggunakan pagar untuk NPM
2 print (1184039%3)
3 #mod3 = 1
4
5 print (" ##  ##  #####      ##  #####  #####  #####")
6 print ("#### #### ##  ##  ##### ##  ##  ##  ##  ##")
7 print (" ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##")
8 print (" ##  ##  #####  #####  ##  ##  #####  #####")
9 print (" ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##")
10 print (" ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##")
11 print (" ##  ##  #####  ##  #####  #####  #####")
```

Gambar I.22: Nomor1

### 2. Jawaban soal nomor 2

```
1 # Jawaban No 2
2
3 npm = input ("Masukkan NPM anda:")
4 ulang = 1
5 while ( ulang <= 9 ) :
6     print ( "Halo , "+ str (npm) + "Apa kabar ?" )
7     ulang += 1
```

Gambar I.23: Nomor2

### 3. Jawaban soal nomor 3

```
1 # Jawaban No . 3
2 npm = input ("Masukkan NPM anda :")
3 ulang = 1
4 while ( ulang <= 9 ) :
5     print ( "Halo, "+ str (npm[4:7]) + "Apa kabar?")
6     ulang += 1
```

Gambar I.24: Nomor3

### 4. Jawaban soal nomor 4

```
1 # Jawaban Nomor 4
2
3 npm = input ("Masukkan NPM anda :")
4 print ("Halo, "+ str(npm[-3]) + "Apa kabar?")
```

Gambar I.25: Nomor4



5. Jawaban soal nomor 5

```
1 a = 1
2 b = 1
3 c = 8
4 d = 4
5 e = 0
6 f = 3
7 g = 9
8
9 npm = [ a , b , c , d , e , f , g ]
10 for n in npm :
11     print (n,end = " ")
```

Gambar I.26: Nomor5

6. Jawaban soal nomor 6

```
1 a = 1
2 b = 1
3 c = 8
4 d = 4
5 e = 0
6 f = 3
7 g = 9
8
9 npm = [ a , b , c , d , e , f , g ]
10 for n in npm :
11     print (n,end = " ")
12 print (a+b+c+d+e+f+g)
```

Gambar I.27: Nomor6

7. Jawaban soal nomor 7

```
1 a = 1
2 b = 1
3 c = 8
4 d = 4
5 e = 0
6 f = 3
7 g = 9
8
9 npm = [ a , b , c , d , e , f , g ]
10 for n in npm :
11     print (n,end = " ")
12 print (a*b*c*d*e*f*g)
```

Gambar I.28: Nomor7

8. Jawaban soal nomor 8

```
1 #Jawaban nomor 8
2 a = 1
3 b = 1
4 c = 8
5 d = 4
6 e = 0
7 f = 3
8 g = 9
9
10 npm = [ a , b , c , d , e , f , g ]
11 for n in npm:
12     print(n)
```

Gambar I.29: Nomor8

9. Jawaban soal nomor 9

```
1 #Jawaban nomor 9
2 a = 1
3 b = 1
4 c = 8
5 d = 4
6 e = 0
7 f = 3
8 g = 9
9
10 npm = [ a , b , c , d , e , f , g ]
11 for n in npm:
12     if (n%2 == 0 ) :
13         if (n!=0) :
14             print (n, end = " ")
```

Gambar I.30: Nomor9

10. Jawaban soal nomor 10

```
1 #Jawaban nomor 10
2 a = 1
3 b = 1
4 c = 8
5 d = 4
6 e = 0
7 f = 3
8 g = 9
9
10 npm = [ a , b , c , d , e , f , g ]
11 for n in npm:
12     if (n%2 != 0 ) :
13         print (n, end = " ")
```

Gambar I.31: Nomor10

11. Jawaban soal nomor 11

```
1 #Jawaban nomor 11
2 a = 1
3 b = 1
4 c = 8
5 d = 4
6 e = 0
7 f = 3
8 g = 9
9
10 npm = [ a , b , c , d , e , f , g ]
11 print (f)
```

Gambar I.32: Nomor11

## 1.9 Keterampilan Penanganan Error

### 1.9.1 Jenis-Jenis Error dan Penanganannya

1. Syntax Error Syntax error adalah suatu keadaan atau kondisi ketika ada kesalahan penulisan kode pada program python hal ini menyebabkan program tidak dapat dijalankan. contohnya kesalahan pemberian titik dua atau tanda kutip. Output pemberitahuan error nya yaitu invalid syntax. Yang harus dilakukan saat terjadi syntax error pada kode program yaitu memperbaiki penulisan kodenya.
2. Name Error Name error, yaitu exception yang muncul ketika melakukan eksekusi pada suatu program terhadap lokal name dan global name tidak terdefinisi. error ini terjadi saat pemanggilan variabel yang tidak di definisikan atau memanggil sebuah function yang tidak ada. Output pemberitahuan error nya yaitu name 'a' is not defined. Untuk mengatasi terjadi name error yaitu dengan memastikan variabel dan function yang akan dipanggil benar-benar ada dalam kode program dan tidak terjadi kesalahan penulisan.
3. Type Error Type Error, yaitu suatu keadaan yang terjadi ketika melakukan eksekusi pada suatu operasi atau fungsi yang tipe datanya berbeda atau tidak sesuai dengan operasi yang akan dilakukan. Contoh kasusnya pada kesalahan tipe data antara string dan integer, kesalahan dalam input list,tupl dan dictionary. Cara penanganannya yaitu dengan mengkonversi variabel yang digunakan sesuai dengan tipe datanya.

4. Indentation error Indentation error, yaitu tulisan kode program yang menjorok. indentation error akan terjadi ketika mengetik kode program namun tidak memperhatikan identasinya. Jika terjadi indentasi maka program akan error. cara mengatasinya yaitu memperhatikan indentasi saat menuliskan suatu program.

### 1.9.2 Penanganan Error

```
1 a = "11"  
2 b = 11  
3 try :  
4     print (a+b)  
5 except TypeError :  
6     print ("Program error terdapat perbedaan tipe data")
```

Gambar I.33: Penanganan Error