

VERSION 1.0

JULI , 2022



[PRAKTIKUM PEMROG. FUNGSIONAL]

MODUL 1 – Pengenalan Paradigma
Pemrograman Fungsional dan Bahasa Python.

DISUSUN OLEH :
Andi Shafira Dyah K.
Mutaqhin Dean

DIAUDIT OLEH
Fera Putri Ayu L., S.Kom., M.T.

PRESENTED BY: TIM LAB-IT
UNIVERSITAS MUHAMMADIYAH MALANG

[PRAKTIKUM PEMROG. FUNGSIONAL]

PERSIAPAN MATERI

Praktikan diharapkan paham konsep fungsi dalam matematika.

TUJUAN PRAKTIKUM

- Mahasiswa mampu memahami paradigma pemrograman fungsional.
- Mahasiswa mampu memahami tipe data dan variabel pada python.
- Mahasiswa mampu memahami dan mengimplementasikan perulangan, percabangan dan fungsi pada python.

PERSIAPAN SOFTWARE/APLIKASI

- Komputer/Laptop
- Sistem operasi Windows/Linux/Max OS
- Pycharm/Google Collab/ Jupyter Notebook

MATERI POKOK

Karena hampir semua materi pada praktikum pemrograman fungsional langsung diimplementasikan dalam *source code*, maka semua materi bisa anda akses pada *Google Collab* melalui tautan [ini](#).

1 Paradigma Pemrograman Fungsional

Pada semester sebelumnya, kalian telah mempelajari mengenai pemrograman berbasis objek (PBO/OOP). Pada pemrograman tersebut, kalian menggunakan objek sebagai komponen untuk menulis sebuah program. Jika pada PBO kalian familiar dengan penggunaan kelas dan objek, pada pemrograman fungsional ini kalian akan mengenal lebih dalam mengenai fungsi (function/method). Sesuai dengan namanya, pemrograman fungsional merupakan paradigma pemrograman yang menggunakan fungsi (function/method) sebagai komponen utama dalam menulis sebuah program. Jika dalam PBO, kalian menggunakan bahasa pemrograman *Java*. Maka dalam pemrograman fungsional ini kalian akan menggunakan bahasa pemrograman *python*. Selain *python*, sebenarnya banyak bahasa pemrograman yang bisa digunakan untuk menerapkan pemrograman fungsional yaitu Javascript, Clojure, Haskell, Ruby, dll.

2 Pengenalan Bahasa Python

Python adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (interpretatif) dengan metode orientasi objek (Object Oriented Programming) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax. Sebagian lain mengartikan Python sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python sendiri menampilkan fitur-fitur menarik sehingga layak untuk dipelajari. Pertama, Python memiliki tata bahasa dan script yang sangat mudah untuk dipelajari. Python juga memiliki sistem pengelolaan data dan memori otomatis. Selain itu modul pada Python selalu diupdate. Python banyak diaplikasikan pada berbagai sistem operasi seperti Linux, Microsoft Windows, Mac OS, Android, Symbian OS, Amiga, Palm dan lain-lain.

Python diciptakan oleh Guido van Rossum pertama kali di Centrum Wiskunde & Informatica (CWI) di Belanda pada awal tahun 1990-an. Bahasa python terinspirasi dari bahasa pemrograman ABC. Sampai sekarang, Guido masih menjadi penulis utama untuk python, meskipun bersifat open source sehingga ribuan orang juga berkontribusi dalam mengembangkannya.

Selain karena mendukung pemrograman fungsional. Python merupakan bahasa pemrograman yang sangat populer hingga saat tulisan ini ditulis, berdasarkan artikel ini, python menduduki posisi pertama sebagai bahasa pemrograman paling populer di dunia. sebagai orang yang bergelut di dunia IT, kita dituntut untuk belajar sesuatu yang baru dengan cepat. Termasuk belajar mengenai teknologi yang populer pada saat ini. Karena alasan itulah python dipilih dipelajari dalam mata praktikum pemrograman fungsional pada semester ganjil 2022 ini.

2.1 Instalasi Python di pycharm IDE (optional)

Sebelum melakukan pemrograman menggunakan python, terlebih dahulu harus melakukan instalasi python pada laptop/komputer anda. Untuk langkah instalasi python, dapat mengikuti tutorial pada link [berikut](#). Setelah berhasil menginstall python, selanjutnya anda membutuhkan IDE untuk menulis dan menjalankan program python yang dibuat. Terdapat banyak IDE yang dapat anda gunakan, tetapi kami merekomendasikan untuk menggunakan IDE pycharm. Untuk link panduan instalasi IDE Pycharm bisa kalian lihat [disini](#)

2.2 Google Collab

Collaboratory, atau disingkat “Collab”, adalah produk dari Google Research. Collab memungkinkan siapa saja untuk menulis dan mengeksekusi kode python arbitrer melalui browser, dan sangat cocok untuk pembelajaran mesin, analisis data, dan pendidikan. Secara lebih teknis, Collab adalah layanan notebook Jupyter yang dihosting yang tidak memerlukan penyiapan untuk digunakan, sekaligus memberikan akses gratis ke sumber daya komputasi termasuk GPU. Oleh karena itu Collab sangat direkomendasikan untuk kalian yang memiliki komputer / pc dengan spesifikasi yang rendah.

Dengan menggunakan google collab, kalian dapat mengakses kode yang sudah kalian tulis, dimana saja dan kapan saja selama kalian punya akses internet. untuk memulai menggunakan google collab, kalian bisa klik tombol connect / hubungkan pada bagian kanan atas dari halaman ini. Setelah tanda centang berwarna hijau sudah muncul, berarti google collab siap untuk digunakan.

3 Tipe Data pada Python

Python memiliki banyak tipe data. Diantaranya adalah berikut :

Text Type:	<code>str</code>
Numeric Types:	<code>int</code> , <code>float</code> , <code>complex</code>
Sequence Types:	<code>list</code> , <code>tuple</code> , <code>range</code>
Mapping Type:	<code>dict</code>
Set Types:	<code>set</code> , <code>frozenset</code>
Boolean Type:	<code>bool</code>
Binary Types:	<code>bytes</code> , <code>bytearray</code> , <code>memoryview</code>

untuk mengetahui tipe data yang sedang digunakan, dapat menggunakan function `type()`. Untuk lebih jelasnya, kalian bisa menjalankan baris kode dibawah ini:

3.1 Text Type

3.1.1 String

```
[2]: # coba jalankan kode dibawah ini agar lebih memahami
      tipe data print("Hello World")
      print(type("Hello Colab"))
```

```
Hello World
<class 'str'>
```

pada baris pertama pada kode diatas, kita melakukan output dengan menggunakan fungsi print() yang kita isikan dengan sebuah string “Hello Colab”, sehingga ketika kita membungkus “Hello Colab” dengan fungsi type() akan menampilkan tipe data dari “Hello Colab”, yaitu string (str)

3.2 Numeric Type

3.2.1 int dan float

```
[3]: # tipe data numerik
      print(type(10)) #int
      print(type(10.0)) #float
```

```
<class 'int'>
<class 'float'>
```

3.3 Sequence Type

sequence merupakan kumpulan dari sebuah nilai yang bertipe data sama ataupun berbeda (seperti array)

3.3.1 list

Dalam bahasa pemrograman Python, struktur data yang paling dasar adalah urutan atau lists. Setiap elemen-elemen berurutan akan diberi nomor posisi atau indeksinya. Indeks pertama dalam list adalah nol, indeks kedua adalah satu dan seterusnya.

```
[4]: # deklarasi list pada bahasa pemrograman python
      list1 = ['kimia', 'fisika', 1993, 2017]
      list2 = [1, 2, 3, 4, 5 ]
      list3 = ["a", "b", "c", "d"]

      print(type(list1))
      print(type(list2))
      print(type(list3))
```

```
<class 'list'>
<class 'list'>
<class 'list'>
```

```
[5]: #Cara mengakses nilai di dalam list Python
      print ("list1[2]: ", list1[2])
      print ("list2[1:5]: ", list2[1:5])
```

```
list1[2]: 1993
```

```
list2[1:5]: [2, 3, 4, 5]
```

```
[6]: list1[2] = "biologi" #mengganti nilai pada indeks ke 2
      print ("Nilai baru ada pada index 2 : ", list1[2])
```

Nilai baru ada pada index 2 : biologi

Selain cara diatas, terdapat banyak cara untuk memodifikasi nilai atau isi dari suatu list, salah satunya yakni dengan metode `append()`. Untuk mempelajari method-method list lainnya, silahkan mengunjungi tautan [berikut](#).

3.3.2 tuple

Sebuah tuple adalah urutan objek Python yang tidak berubah. Tuple adalah urutan, seperti daftar. Perbedaan utama antara tuple dan daftarnya adalah bahwa tuple tidak dapat diubah (*immutable*) tidak seperti List Python. Tuple menggunakan tanda kurung, sedangkan List Python menggunakan tanda kurung siku.

Membuat tuple semudah memasukkan nilai-nilai yang dipisahkan koma. Secara opsional, Anda dapat memasukkan nilai-nilai yang dipisahkan koma ini di antara tanda kurung juga. Sebagai contoh :

```
[7]: #Contoh sederhana pembuatan tuple pada bahasa pemrograman python
```

```
tup1 = ('fisika', 'kimia', 1993, 2017)
tup2 = (1, 2, 3, 4, 5 )
tup3 = "a", "b", "c", "d"
```

```
print(type(tup1))
print(type(tup2))
print(type(tup3))
```

```
<class 'tuple'>
<class 'tuple'>
<class 'tuple'>
```

```
[8]: #Cara mengakses nilai di dalam tuple
```

```
print ("tup1[0]: ", tup1[0])
print ("tup3[1:4]: ", tup3[1:4])
```

```
tup1[0]: fisika
tup3[1:4]: ('b', 'c', 'd')
```

Karena tuple bersifat *immutable*, maka isi dari suatu tuple tidak dapat dimodifikasi seperti pada list. Untuk mempelajari tuple lebih lanjut, anda dapat mengunjungi tautan [berikut](#).

3.3.3 range

`range` adalah suatu fungsi pada python yang mereturn bilangan sequence dimulai dari 0 hingga mencapai batas tertentu. Biasanya `range` digunakan pada perulangan `for` dan `while`.

```
[9]: a = range(0,10)

print(a)      # Untuk menampilkan semua nilai dari range,
print(list(a)) # perlu casting ke list dahulu

print(type(a))

range(0, 10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
<class 'range'>
```

3.4 Mapping Type

3.4.1 Dictionary

Dictionary seperti buku alamat, dengan buku alamat anda bisa mencari alamat atau detail kontak hanya menggunakan nama orang yang anda cari. Kita mengasosiasikan key (nama) dengan value (detail). Catatan key harus bersifat unik, anda tidak bisa menemukan informasi yang tepat jika ada dua orang yang mempunyai nama yang sama dalam buku alamat anda.

Anda hanya bisa menggunakan objek immutable (seperti string) untuk key/ kunci dictionary. Anda bisa menggunakan obyek mutable atau immutable untuk value dalam dictionary. angka, atau tuple.

```
[10]: #membuat dictionary
dict1 = {'Nama': 'Seneca', 'Umur': 15,
        'Kelas': '10'} print(type(dict1))

#mengkases value dictionary berdasarkan
key-nya. print("dict1['Nama']: ",
dict1['Nama']) print("dict1['Umur']: ",
dict1['Umur'])

<class 'dict'>
dict1['Nama']: Seneca
dict1['Umur']: 15
```

```
[11]: dict1['Hobby'] = "Merenung" # Menambah entri baru
dict1['Umur'] = 20; # Mengubah value yang sudah ada berdasarkan keynya
print(dict1)

{'Nama': 'Seneca', 'Umur': 20, 'Kelas': '10', 'Hobby': 'Merenung'}
```

Untuk mempelajari dictionary lebih lanjut, anda dapat mengunjungi tautan [berikut](#).

3.5 Boolean Type

Menyatakan benar **True** yang bernilai 1, atau salah **False** yang bernilai 0

```
[12]: print(type(True))
print(type(False))
#penulisan true dan false wajib diawali huruf kapital
```

```
<class 'bool'>
<class 'bool'>
```

4 Variabel pada Python

Pada python, untuk menuliskan sebuah variabel tidak perlu mendefinisikan tipe data yang ingin disimpan di sebuah variabel. Sebuah variabel dapat berganti tipe data sedinamis mungkin. Jalankan baris kode di bawah agar kalian lebih memahami penggunaan variabel pada python

```
[13]: a = 5
      print(a) #5
      print(type(a)) #mencetak tipe data dari variabel a yaitu int
      a = "Ganti Variable" #pergantian nilai dari variabel a beserta
      tipe datanya print(a) #"Ganti Variable"
      print(type(a)) #mencetak tipe data dari variabel a yang baru yaitu str
      (string)
```

```
5
<class 'int'>
Ganti Variable
<class 'str'>
```

Di dalam python, untuk menggabungkan antara variabel satu dengan variabel lainnya, ada sebuah aturan, yaitu variabel dengan tipe data yang berbeda tidak dapat digabungkan kecuali melalui casting. Agar lebih memahami mengenai penggabungan variabel dan casting kalian bisa menjalankan baris kode di bawah ini

```
[14]: # akan menampilkan 12 karena 5 dan 7 merupakan tipe data int
      5 + 7
```

```
[14]: 12
```

Selain bisa menggabungkan tipe data numerik, anda juga bisa menggabungkan tipe data teks dan sekuens.

```
[15]: #Menggabungkan tipe data String
      "Hallo" + " dunia"
```

```
[15]: 'Hallo dunia'
```

```
[16]: #menggabungkan tipe data list
      [1,2,3,4] + [5,6,7,8]
```

```
[16]: [1, 2, 3, 4, 5, 6, 7, 8]
```

```
[17]: a = 5
      b = 10
      print(a+b)
```

```
[17]: 15
```

```
[18]: # akan menampilkan error karena dua tipe data berbeda tidak dapat digabung
      5 + "Halo"
```



```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-18-0dd7ff18140d> in <module>()
    akan menampilkan error karena dua tipe data berbeda
    1          #tidak dapat digabung
    .
    ----> 2 5+ "Halo"

TypeError: unsupported operand type(s) for +: 'int' and 'str'

```

```

[19]: c=5
      d="Halo"
      print(c+d)

```

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-19-519beec7e877> in <module>()
    1          c = 5
    2          d = "Halo"
    ----> 3 print(c+d)

TypeError: unsupported operand type(s) for +: 'int' and 'str'

```

4.1 Casting

Casting berguna untuk mengubah tipe data yang berbeda agar dapat digabungkan/dioperasikan secara bersamaan.

```

[20]: #merubah tipe data int menjadi str
      print("Contoh casting int ke string")
      x = 1 #pada awalnya tipe data dari x adalah int, karena x memiliki
            nilai sebuah bilangan integer
      print(type(x))

      x = str(x) #fungsi str() digunakan untuk merubah variabel di
                 dalamnya (yaitu variabel x) untuk diubah menjadi string
      print(type(x)) #setelah diubah dengan method str(), variabel x sekarang
                     bertipe data string / str

```

Contoh casting int ke string

```

<class 'int'>
<class 'str'>

```

```
[21]: #merubah tipe data float menjadi int
print("Contoh casting int ke string")
y = 20.3 #pada awalnya tipe data dari y adalah float, karena y
        memiliki nilai sebuah float (bilangan yang memiliki koma / floating
        point)
print(type(y))

y = int(y) #fungsi int() digunakan untuk merubah variabel di
           dalamnya (yaitu variabel y) untuk diubah menjadi integer
print(type(y)) #setelah diubah dengan method int(), variabel y sekarang
               bertipe data integer / int
```

Contoh casting int ke string

```
<class 'float'>
<class 'int'>
```

```
[22]: #setelah mempelajari beberapa metode untuk melakukan casting tipe data,
       #kalian bisa mencoba program dibawah ini untuk menerapkan casting tipe
       data dengan penggabungan variabel

a = "hello"
b = 10

print(a + " Colab, kamu berumur " + str(b) + " tahun")
```

hello Colab, kamu berumur 10 tahun

Selain bisa mengubah tipe data primitive, casting juga bisa mengubah type data sequence.

```
[23]: print("contoh casting list menjadi tuple")
       ini_list = [1,1,2,3,4,5]
       print(type(ini_list))

       ini_tuple = tuple(ini_list) #proses casting
       print(type(ini_tuple))
```

contoh casting list menjadi tuple

```
<class 'list'>
<class 'tuple'>
```

5 Percabangan dalam Python

Sama seperti bahasa pemrograman lainnya, python juga memiliki percabangan seperti if, else, elif, dkk. Penerapan percabangan ini mirip dengan bahasa pemrograman yang telah kalian pelajari. Untuk lebih memahaminya, kalian bisa menjalankan baris program dibawah ini:

```
[24]: # if
a = 33
b = 200
if b > a:
    print("b is greater than a")
```

b is greater than a

```
[25]: # elif
a = 33
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
```

a and b are equal

```
[26]: # if, elif, else
a = 200
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
```

a is greater than b

untuk lebih lengkapnya mengenai percabangan pada python, kalian bisa membuka link [berikut](#)

6 Perulangan pada Python

6.1 For Loop

Berbeda dari bahasa pemrograman lain, for loop pada python digunakan untuk looping terhadap sequence (list, tuple, dictionary, set atau string)

```
[27]: #for loop pada list
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

apple
banana
cherry

```
[28]: #for loop pada string
for x in "banana":
    print(x)
```

b
a
n
a
n
a

```
[29]: #menggunakan keyword break untuk keluar dari for loop #program akan
      berhenti ketika nilai x adalah "banana"
      fruits = ["apple", "banana", "cherry"]
      for x in fruits:
          print(x)
          if x == "banana":
              break
```

apple
banana

```
[30]: #menggunakan keyword continue
      #ketika x bernilai "banana", maka akan di skip dan dilanjut iterasi
      selanjutnya
      fruits = ["apple", "banana", "cherry"]
      for x in fruits:
          if x == "banana":
              continue
          print(x)
```

apple
cherry

```
[31]: #menggunakan range pada for loop
      for x in range(6):
          print(x)
```

0
1
2
3
4
5

untuk lebih lengkapnya mengenai for loop, bisa mengunjungi link [berikut](#)

6.2 While Loop

Pada python, while loop mirip dengan while loop yang ada pada bahasa pemrograman lainnya. Tidak seperti for loop yang memiliki aturan khusus. While loop pada python cukup mudah di-pahami. Kalian bisa mencoba langsung while loop pada python dengan menjalankan program dibawah ini

```
[32]: i = 1
      while i < 6:
          print(i)
          i += 1
```

1
2
3
4
5

untuk lebih lengkapnya mengenai while loop, bisa mengunjungi link [berikut](#).

7 Function pada python

Jika pada paradigma pemrograman sebelumnya (OOP-Java) kalian lebih familiar dengan

istilah method, maka pada pemrograman fungsional dengan python kali ini kita akan banyak menggunakan function (di python juga ada method kok, cek [disini](#) untuk tau perbedaan keduanya).

Python sendiri memiliki banyak built-in function, diantaranya yang sering anda gunakan, antara lain print() berfungsi untuk mencetak objek, len() berfungsi mengembalikan (return) panjang suatu objek,

Sedangkan untuk membuat suatu function, perlu didefinisikan dengan keyword “def”. Untuk lebih memahami mengenai function pada python, anda dapat menjalankan program dibawah ini.

```
[33]: def my_function():  
      print("Hello from a function")  
  
      my_function()
```

Hello from a function

```
[34]: def my_function(fname):  
      print(fname + " Refsnes")  
  
      my_function("Emil")  
      my_function("Tobias")  
      my_function("Linus")
```

Emil Refsnes
Tobias Refsnes
Linus Refsnes

```
[35]: def my_function(fname, lname):  
      print(fname + " " + lname)  
      my_function("Emil", "Refsnes")
```

Emil Refsnes

8 Tugas Praktikum

Buatlah sebuah program tentang sistem informasi, bebas menggunakan google collab ataupun Pycharm atau IDE python lain. Ketentuan program tersebut adalah:

8.1 Soal NIM ganjil

1. Ada dua jenis akun, akun user dan akun admin
2. Akun admin dapat melakukan input buku yang akan dipinjam
3. Akun user dapat melakukan peminjaman buku yang tersedia
4. Buku yang sudah dipinjam tidak dapat dipinjam oleh user lain.
5. Akun user harus dapat mengembalikan buku.

6. Demokan skenario kepada asisten

8.2 Soal NIM genap

1. Terdapat dua akun, akun teacher dan akun student
2. Akun teacher dapat melakukan input data student dan edit nilai
3. Terdapat 4 kolom untuk data student, yakni ID (untuk mempermudah, id sebaiknya berurut dari 0 dst.), Nama, nilai UTS, nilai UAS
4. Akun student bisa menampilkan nilai akhir dari student itu sendiri
5. Nilai akhir diperoleh dari nilai UTS + UAS / 2
6. Demokan skenario kepada asisten

Buatlah program sekreatif kalian dengan syarat harus sesuai dengan ketentuan diatas dan gunakan semaksimal mungkin materi yang sudah kalian pelajari di modul 1 ini. **Dilarang menggunakan library tambahan (seperti pandas dll.).**

9 Rubrik Penilaian

1. Program dapat berfungsi (max. 20)
2. Program menggunakan materi yang sudah dipelajari pada modul (max. 20)
3. Menjelaskan dengan **baik dan lancar** kepada asisten saat demo:
 - Menjelaskan program yang dibuat (max. 20)
 - Menjelaskan yang sudah dipelajari dari modul (max. 20)
 - Menjawab pertanyaan dari asisten (max. 20)
4. Jika program identik dengan praktikan lain. Maka akan ada pengurangan nilai pada kedua praktikan.