

# Algoritme dan Pemrograman



ALL YOU NEED TO KNOW BEFORE CODING

---

Gusti Ahmad Fanshuri Alfarisy, M.Kom.

## Daftar Isi

Kata Pengantar.....	6
Minggu 1 Pengantar.....	7
Tujuan .....	7
Instalasi PyCharm.....	7
Minggu 2 Variabel, Konsol I/O, dan Konversi Tipe Data .....	19
Tujuan .....	19
Variabel .....	19
Konsol I/O – Print .....	20
Konsol I/O – Input .....	21
Praktikum 3 : Input.....	21
Operasi Aritmatika .....	21
Praktikum 4 : Aritmatika .....	22
Konversi Tipe Data .....	22
Praktikum 4 : Konversi Tipe Data .....	23
Latihan.....	23
Minggu 3 Eksekusi Kondisional .....	24
Tujuan .....	24
Eksekusi Kondisional .....	24
Praktikum 1 : Program Luas Bidang .....	25
Praktikum 2 : Permainan Tebak Angka .....	26
Eksekusi Kondisional Bersarang.....	26
Praktikum 3 : Program Tahun Kabisat.....	27
Eksekusi Kondisional Majemuk.....	27
Praktikum 4 : IMT .....	28
Praktikum 5 : Permainan Suit/Pingsut .....	29
Latihan.....	30
Minggu 4 Perulangan dengan While.....	32
Tujuan .....	32
While .....	32

	3
Praktikum 1 : Bilangan Genap.....	32
Praktikum 2 : Daftar Perkalian .....	33
Praktikum 3 : Program Luas dan Keliling Lingkaran .....	33
Praktikum 4: Tebak Angka .....	33
Latihan.....	34
Minggu 5 Perulangan dengan For .....	35
Tujuan .....	35
Definite Loop dengan For.....	35
Praktikum 1 : Program Bilangan Ganjil dan Jumlahnya .....	35
Praktikum 2 : Program Prediksi Investasi.....	36
Break dan Continue.....	36
Praktikum 3 : Penjumlahan Bilangan Positif .....	37
Perulangan Bersarang.....	37
Praktikum 4 : Pencetakan Segitiga Siku-siku.....	39
Latihan.....	39
Minggu 6 List dan Dictionary .....	42
Tujuan .....	42
List.....	42
Praktikum 1: Statistik sederhana dari list .....	43
Praktikum 2: Penjumlahan Matriks (List dua dimensi) .....	43
Dictionary.....	44
Praktikum 3: Toko Buah .....	44
Latihan.....	45
Minggu 7 String.....	46
Tujuan .....	46
String sebagai Ururan Karakter .....	46
Kunjungan Karakter pada String .....	47
Operator pada String .....	47
Metode pada String .....	48
Praktikum 1: Singkatan .....	48
Praktikum 2: RNA .....	49

	4
Praktikum 3: Tanda Kurung.....	49
Latihan.....	50
Minggu 8 Ujian Tengah Semester .....	52
Minggu 9 Fungsi dan Rekursif.....	53
Tujuan .....	53
Pemanggilan Fungsi .....	53
Membuat Fungsi .....	53
Default Argumen.....	54
Praktikum 1: Contoh Fungsi.....	54
Fungsi dengan Nilai Kembali .....	55
Praktikum 2: Contoh Fungsi dengan Nilai Kembali .....	55
Latihan Belajar Mandiri.....	56
Fungsi Rekursif.....	56
Praktikum 3: Faktorial dan Fibonacci .....	57
Latihan Belajar Mandiri.....	57
Latihan.....	57
Minggu 10 Penanganan Kesalahan .....	59
Tujuan .....	59
Mengangkat Kesalahan.....	59
Praktikum 1: Harga Diskon.....	59
Latihan Belajar Mandiri.....	59
Praktikum 2: Raise Error .....	60
Latihan Belajar Mandiri.....	60
Blok Try dan Except.....	60
Praktikum 3: Program Luas Segitiga dengan Penanganan Kesalahan .....	61
Latihan.....	62
Minggu 11 Menulis dan Membaca Berkas .....	63
Tujuan .....	63
Menulis dan Membaca Berkas.....	63
Praktikum 1 : Membaca dan Menulis File Teks .....	63
Latihan Belajar Mandiri.....	64

	5
Latihan.....	64
Minggu 12 PyQt5: Widget.....	65
Tujuan .....	65
Instalasi PyQt5 .....	65
Widget.....	66
Praktikum 1: Hello World.....	69
Praktikum 2: ComboBox.....	69
Praktikum 3: Kedai Kopi .....	71
Latihan.....	73
Minggu 13 PyQt5: Layout .....	74
Tujuan .....	74
Layout.....	74
Praktikum 1: Data Mahasiswa dengan Horizontal dan Vertikal Layout.....	74
Praktikum 2: Dummy Kalkulator dengan Grid Layout .....	76
Latihan.....	77
Minggu 14 Progress Tugas Besar .....	78
Tujuan .....	78
Petunjuk Asisten dan Praktikan .....	78
Minggu 15 Progress Tugas Besar .....	79
Tujuan .....	79
Petunjuk Asisten dan Praktikan .....	79
Minggu 16 Ujian Akhir Semester – Persentasi Tugas Besar .....	80
Lampiran LPTB .....	81

## Kata Pengantar

Segala puji bagi Allah ﷻ yang telah memberikan karunia-Nya sehingga modul praktikum ini dapat diselesaikan sebagaimana mestinya sesuai target yang telah ditentukan. Penulis menyadari bahwa seluas-luasnya ilmu adalah ilmu dari Allah ﷻ. Tanpa melalui izin dan bimbingan-Nya penulis tidak mungkin dapat menyelesaikan modul ini dengan sabaik-baiknya. Selain itu, penulis menulis kata pengantar ini ditengah-tengah pandemi global Covid-19 yang menjadi bencana non-alam nasional di Indonesia. Penulis berharap dan berdoa semoga wabah ini cepat berakhir dan keadaan kembali ke keadaan normal seperti sediakala.

Modul praktikum ini diperuntukan terutama bagi mahasiswa di lingkungan Institut Teknologi Kalimantan (ITK) sebagai acuan dalam mengikuti praktikum Algoritme Pemrograman. Dengan keberagaman program studi yang ada di ITK, modul ini sebisa mungkin dirancang untuk semua kalangan namun memberikan tantangan bagi mahasiswa yang antusias dan tertarik mengkaji masalah-masalah komputasi. Modul ini sebisa mungkin tidak hanya menyajikan kode program, namun juga memberikan penjelasan tentang dasar penggunaan sintaksis dari bahasa pemrograman Python. Sebagai tambahan, modul ini diharapkan membuat mahasiswa mampu belajar mandiri melalui latihan-latihan belajar mandiri yang disediakan dimodul dan dikerjakan selama praktikum. Tujuannya adalah untuk meningkatkan kemampuan mahasiswa sebagai pembelajar mandiri untuk memenuhi tantangan dunia kerja masa depan yang memiliki *problem* rumit dan tidak diketahui sebelumnya oleh mahasiswa. Sejatinya, *programmer* yang handal adalah pembelajar sejati.

Penulis menyadari apa yang telah ditulis dalam modul ini tidak luput dari kekurangan. Sehingga, penulis membuka seluas-luasnya kritik dan saran dari dosen pengajar, instruktur, asisten maupun praktikan yang membangun demi kebaikan bersama. Kritik dan saran tersebut dapat disampaikan ke [gusti.alfarisy@itk.ac.id](mailto:gusti.alfarisy@itk.ac.id).

Semoga modul praktikum ini yang telah disusun secara bertahap sejak tahun 2019 dapat memberikan manfaat yang seluas-luasnya bagi civitas akademika ITK ataupun masyarakat pada umumnya.

Kalimat pengantar terakhir yang ingin saya tekankan adalah,

“Apabila dengan ilmu ini saudara memperoleh kebaikan di Akhirat maka gunakanlah, namun apabila menjauh darinya apalagi memberikan keburukan bagi sesama, maka Penulis berlepas diri dari apa yang telah saudara lakukan.”

Banjarmasin, 6 April 2020

Gusti Ahmad Fanshuri Alfarisy, S.Kom., M.Kom.

# Minggu 1

## Pengantar

### Tujuan

Setelah menyelesaikan pertemuan ini, praktikan mampu melakukan instalasi IDE PyCharm dan diharapkan siap melakukan praktikum pada minggu berikutnya

### Instalasi PyCharm

Terdapat dua perangkat lunak yang perlu disiapkan sebelum memprogram aplikasi yaitu Python sebagai interpreter untuk menerjemahkan sintaksis pemrograman kedalam program yang dapat dijalankan dan PyCharm sebagai Integrated Development Environment (IDE). Bagi yang belum mengenal IDE, kerjanya mirip seperti editor teks (Ms. Word, Libre Writer, Google Docs, dan lain-lain) yang dapat kita ketikkan teks (kode program) disertai fitur yang lengkap.

Langkah pertama adalah instalasi Python, praktikan dapat mengunjungi <https://www.python.org/downloads/> atau menyalin program dari asisten praktikum. Tampilan laman web Python sejak modul ini ditulis (2020) dapat dilihat pada Gambar 1. Setelah memasuki laman tersebut, praktikan dapat mengklik "Download Python...."



**Gambar 1 Laman web Python**

Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		f9f3768f757e34b342dbc06b41cbc844	24007411	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		e9d5ebc92183a177b8e8a58cad5b8d67	17869888	<a href="#">SIG</a>
<a href="#">macOS 64-bit installer</a>	Mac OS X	for OS X 10.9 and later	f12203128b5c639dc08e5a43a2812cc7	30023420	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		7506675dcbb9a1569b54e600ae66c9fb	8507261	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	1a98565285491c0ea65450e78afe6f8d	8017771	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	b5df1cbb2bc152cd70c3da9151cb510b	27586384	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	2586cdad1a363d1a8abb5fc102b2d418	1363760	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		1b1f0fc5ee8601f160cfad5b560e3a7	7147713	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		6f0ba59c7d8eba7bb0ee21682fe39748	26481424	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		04d97979534f4bd33752c183fc4ce680	1325416	<a href="#">SIG</a>

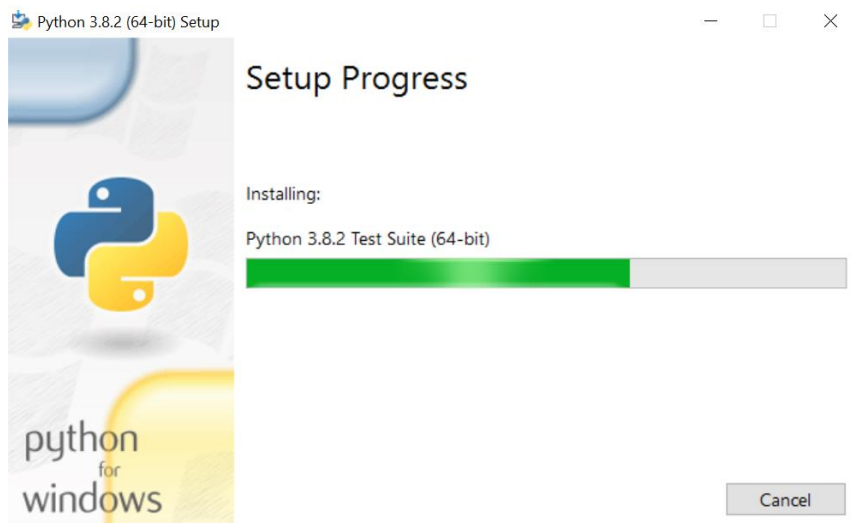
**Gambar 2 Laman web installer Python**

Atau dapat scroll ke bawah, terdapat daftar installer, download sesuai kebutuhan. Pada umumnya, jika sistem operasi yang digunakan adalah windows 64 bit, maka Windows x86-64 executable installer yang diunduh. Laman yang telah di scroll dapat dilihat pada Gambar 2.



**Gambar 3 Installer Python**

Setelah berhasil diunduh, klik installer python dan pastikan PATH telah terpilih seperti terlihat pada Gambar 3. Selain itu, direktori instalasi dapat diingat dengan baik atau dicatat. Selanjutnya, klik “Install Now”.



**Gambar 4 Progress Instalasi Python**

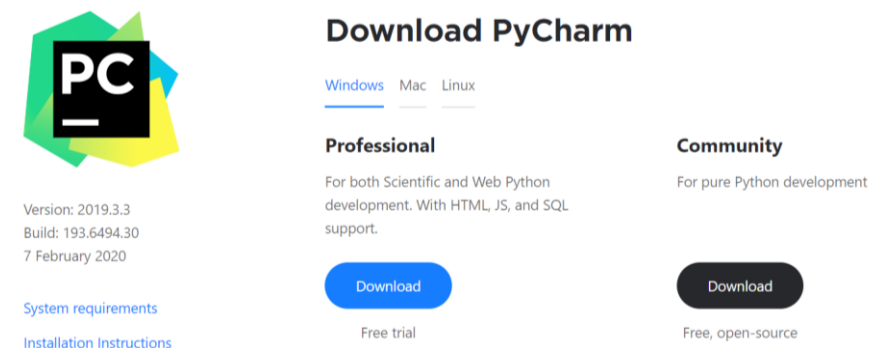
Tunggu instalasi selesai seperti pada Gambar 4.





**Gambar 5 Instalasi Berhasil**

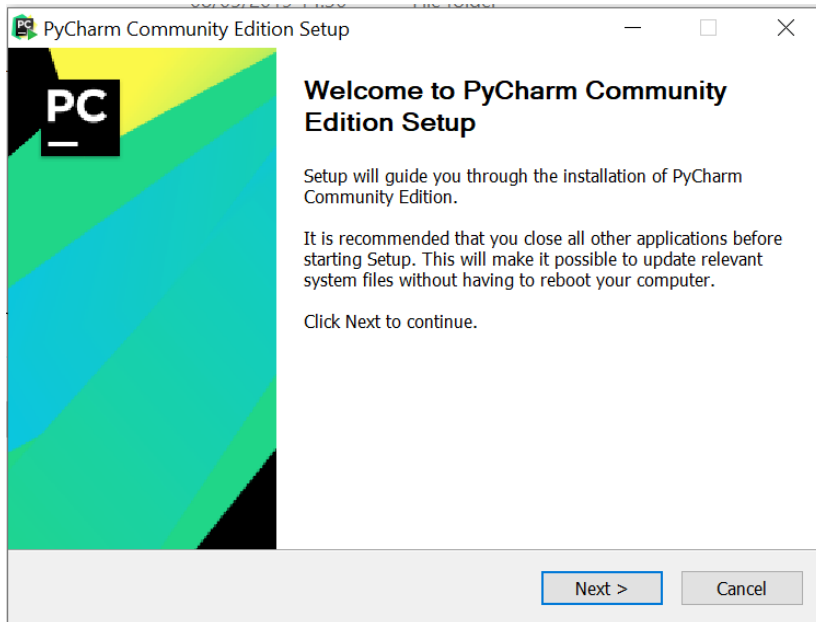
Setelah instalasi selesai, anda akan dihadapkan pada jendela Gambar 5. Informasi tersebut menunjukkan bahwa instalasi telah selesai, klik “close”.



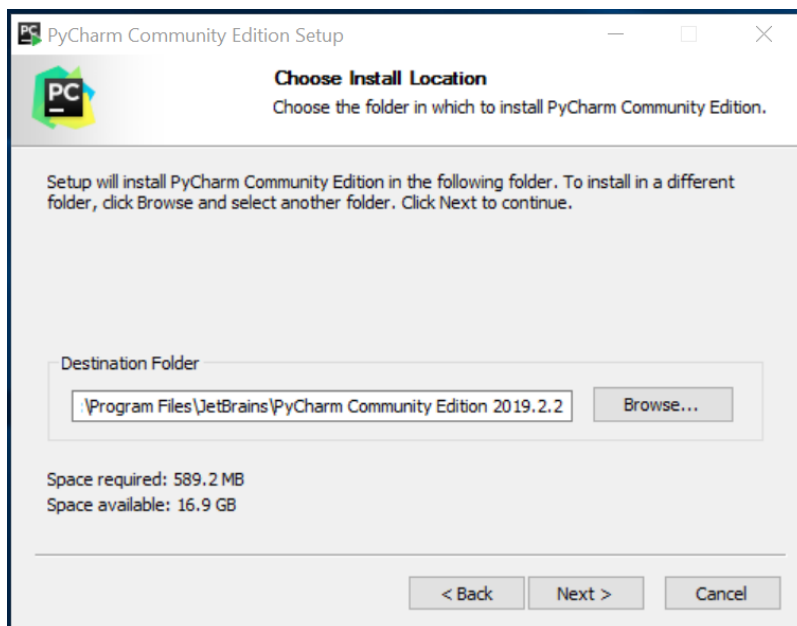
**Gambar 6 Laman web installer PyCharm IDE**

Langkah selanjutnya adalah mengunduh IDE PyCharm di <https://www.jetbrains.com/pycharm/download/#section=windows>. Lalu pilih sesuai sistem operasi yang anda gunakan dan pilih opsi *community* (komunitas) selanjutnya klik “Download”. Lebih jelasnya dapat dilihat pada Gambar 6.

Selesai mengunduh, klik installer dan anda akan melihat jendela seperti pada Gambar 7. Klik Next dan anda akan dibawa ke jendela Gambar 8. Pastikan direktori sudah selesai (jika anda belum mahir menggunakan windows, lebih baik biarkan direktori tersebut/default sebagai alamat penyimpanan hasil instalasi). Selain itu, anda juga perlu memastikan kapasitas penyimpanan sekunder anda (Hardisk/SSD) mencukupi, yaitu sekitar 600 MB. Jika selesai, klik “Next”

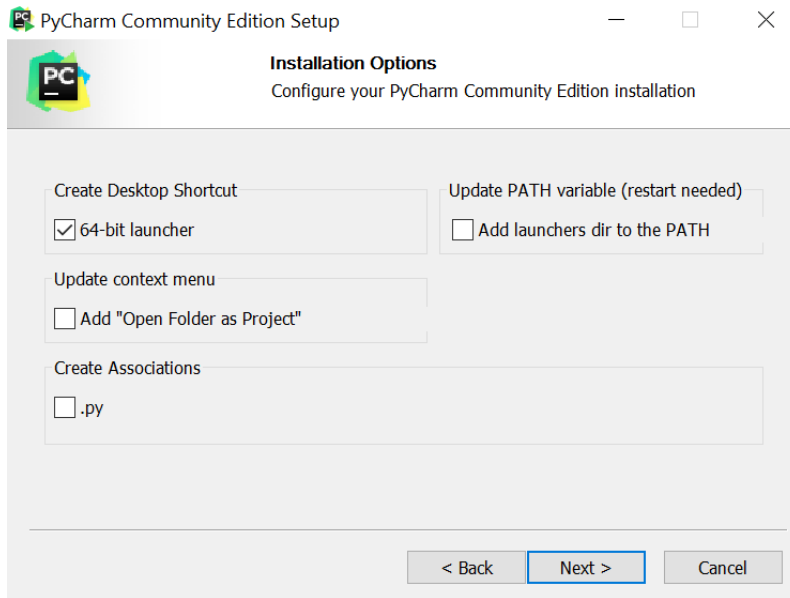


**Gambar 7 Setup pertama installer PyCharm**

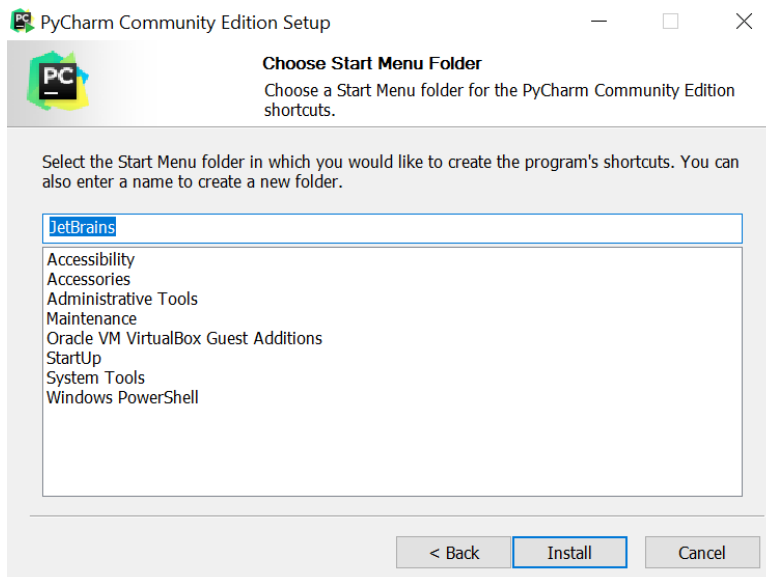


**Gambar 8 Pemilihan Lokasi Instalasi**

Selanjutnya, anda mengatur instalasi. Jika arsitektur perangkat berbasis 64 bit, maka beri tanda centang pada “64-bit launcher” seperti terlihat pada Gambar 9. Klik “next” dan anda akan dibawa pada Jendela *start menu* pada Gambar 10, klik “next”.

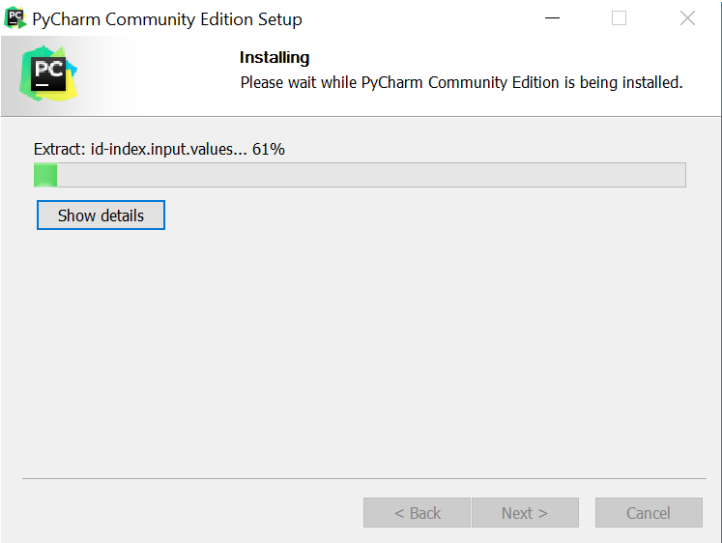


**Gambar 9 Opsi Instalasi**

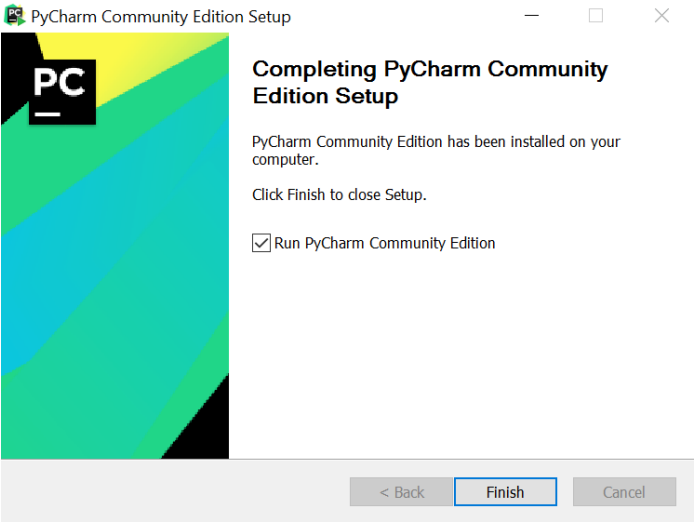


**Gambar 10 Pemilihan Berkas Menu Awal**

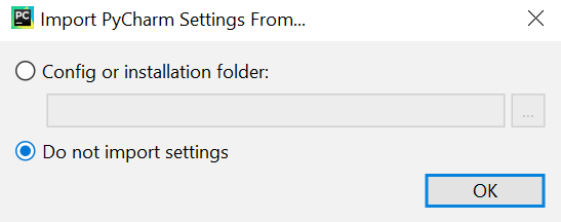
Tunggu proses *loading* selesai, anda dapat melihat jendela progress pada Gambar 11. Setelah memuat data selesai, jendela pemberitahuan instalasi selesai akan muncul seperti pada Gambar 12. Untuk mencoba IDE PyCharm secara langsung, beri tanda centeng pada “Run PyCharm Community Edition”. Selanjutnya, anda akan dihadapkan pada Gambar 13, karena PyCharm anda gunakan pertama kali, pilih opsi “Do not import settings”.



**Gambar 11 Progress Instalasi**

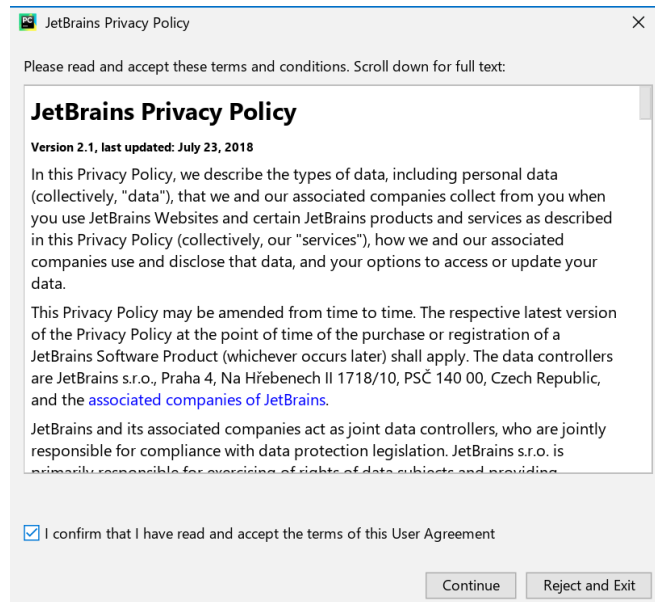


**Gambar 12 Jendela Pemberitahuan Instalasi Selesai**

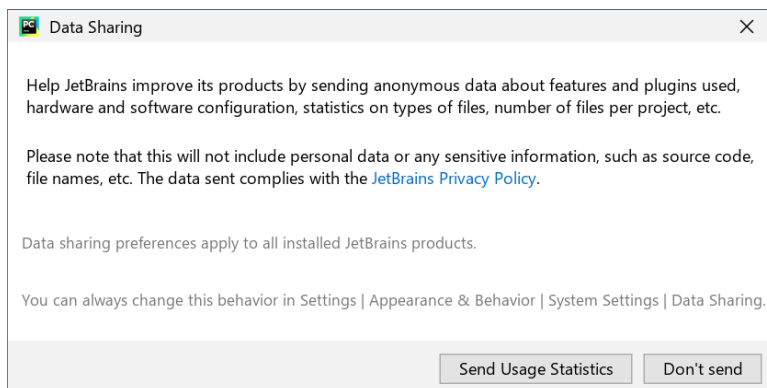


**Gambar 13 Jendela Impor Pengaturan**

Setelah selesai klik “OK”, jendela kebijakan privasi akan muncul seperti terlihat pada gambar 14. Lalu centang pada box “*I confirm that I have....*” dan klik “Continue”



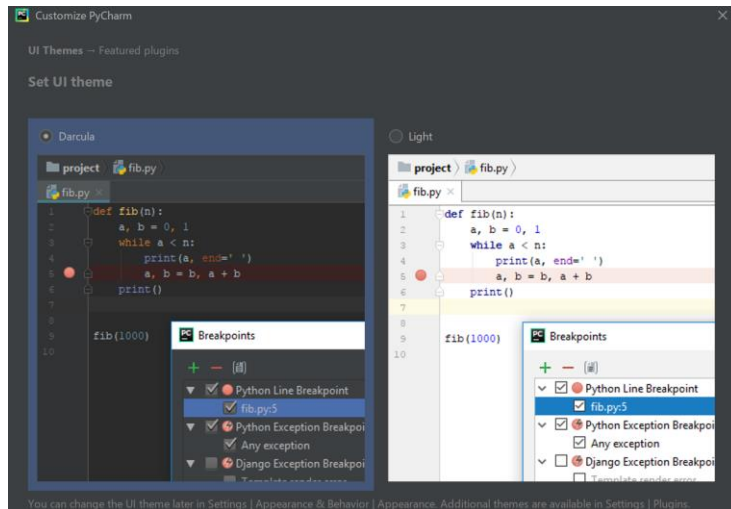
**Gambar 14 Kebijakan Privasi JetBrains**



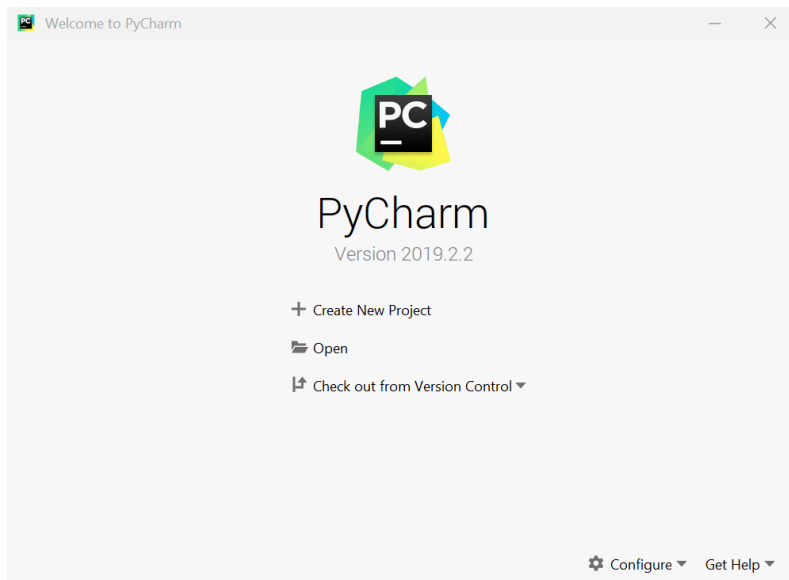
**Gambar 15 Opsi Berbagi Data**

Anda akan dibawa kepada jendela berbagi data seperti terlihat pada Gambar 15. Jika anda ingin membantu JetBrains untuk berbagi data terkait fitur dan plugin yang digunakan, konfigurasi perangkat keras dan luna, statistik jenis file, jumlah file per proyek, dan lain-lain anda dapat klik “*Send Usage Statistics*”. Namun jika dirasa tidak perlu dapat klik “*Don't send*”

Setelah itu, anda diminta untuk memilih tema/gaya/warna tampilan dari PyCharm. Pilih sesuai selera anda seperti terlihat pada Gambar 16.

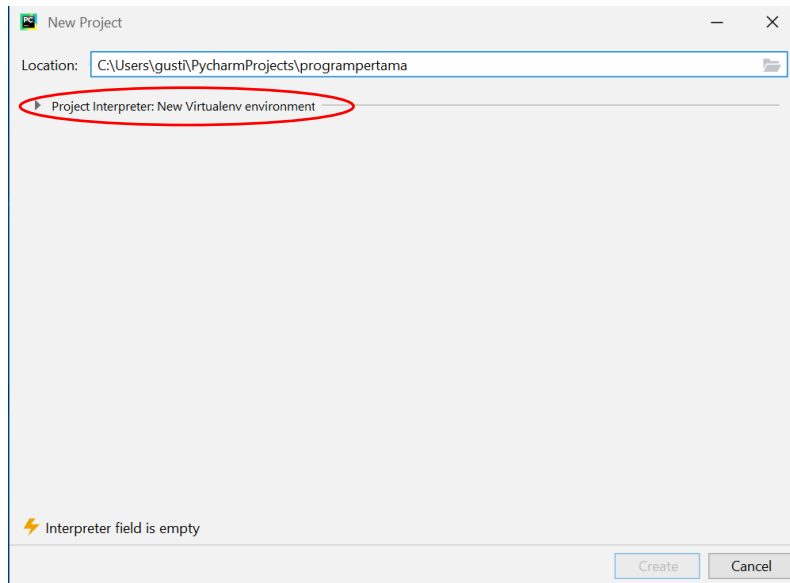


**Gambar 16 Pemilihan Tema PyCharm**



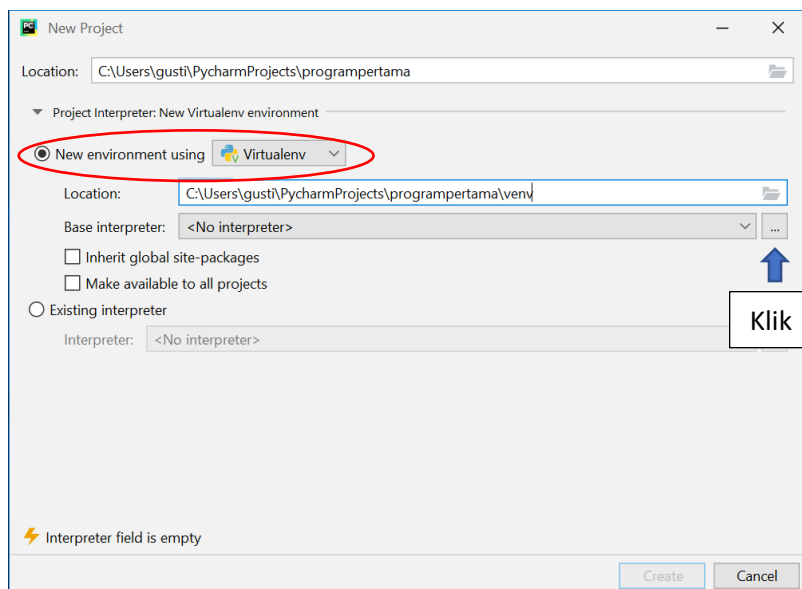
**Gambar 17 Jendela Awal PyCharm saat Tidak Ada Proyek**

Lalu anda akan dibawa pada jendela awal PyCharm. Jendela ini muncul jika anda belum memiliki proyek sama sekali sebelumnya (Gambar 17). Klik “+ Create New Project”



**Gambar 18 Jendela Proyek Baru**

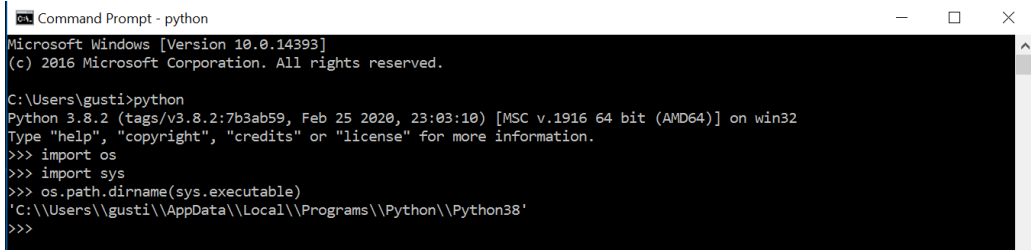
Selanjutnya anda akan melihat jendela lokasi proyek baru seperti pada Gambar 18. Anda dapat menyesuaikan lokasi proyek. Namun, pada tahapan awal, PyCharm belum mengetahui lokasi interpreter agar dapat menjalankan program. Untuk itu pengaturan lanjutan diperlukan. Klik “> Project Interpreter”



**Gambar 19 Pengaturan Environment**

Seperti terlihat pada Gambar 19, pilih opsi *New environment using /Virtualenv/*. Lalu klik tanda ... pada Base interpreter dan masukkan direktori interpreter Python. Anda dapat

melihat direktori instalasi Python pada Gambar 3. Namun jika anda sudah lupa, anda dapat mengecek direktori Python dengan masuk ke *command prompt* seperti pada Gambar 20. Ikuti langkah sesuai gambar tersebut (Hal ini dapat berfungsi jika tanda contend diberikan pada *add python .. path* seperti pada Gambar 3).

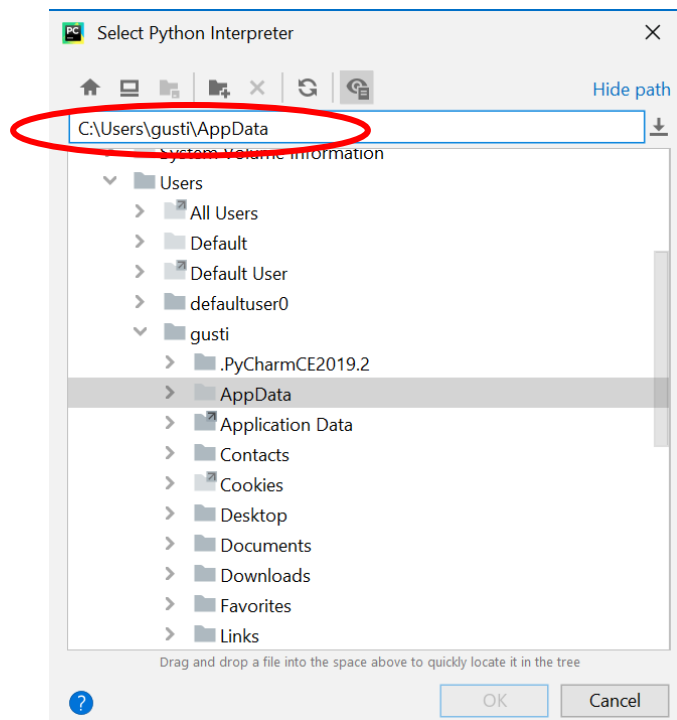


```

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\gusti>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> import sys
>>> os.path.dirname(sys.executable)
'C:\Users\gusti\AppData\Local\Programs\Python\Python38\'
>>>
  
```

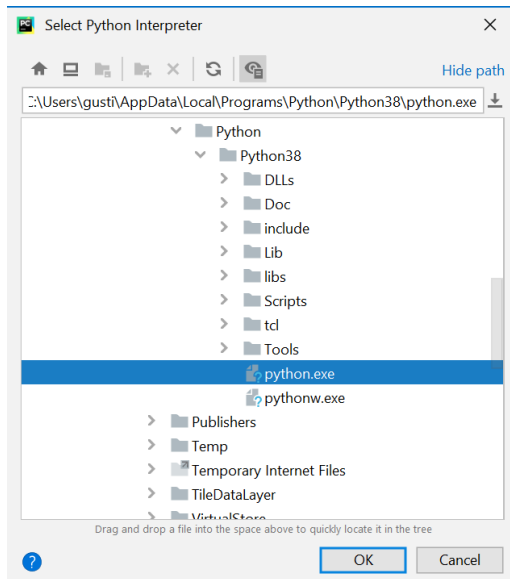
**Gambar 20 Direktori Python dengan Command Prompt**



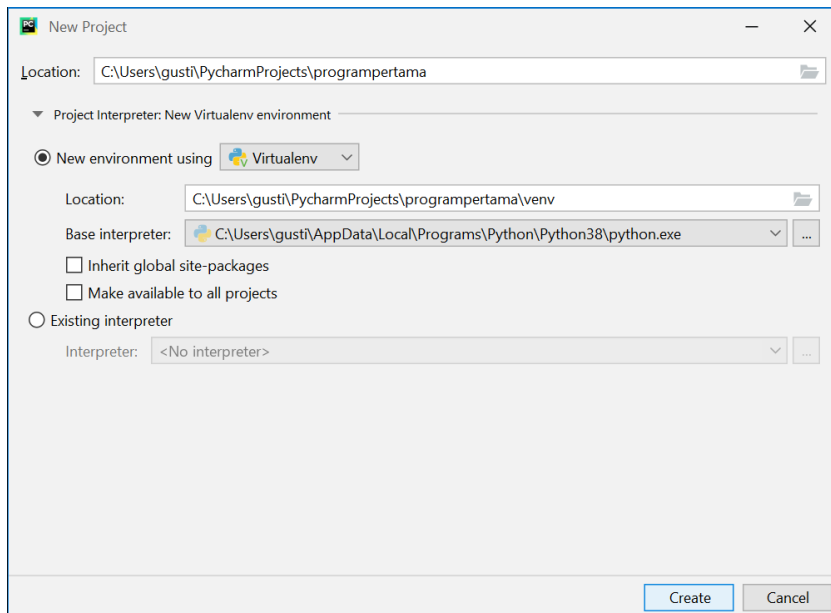
**Gambar 21 Masuk AppData, pemilihan Python Interpreter**

Selanjutnya, setelah anda mengetahui alamat direktorinya dan ketika anda mencoba menjelajah direktori tersebut tidak ditemukan (pada AppData), maka anda dapat langsung menuliskannya di kolom direktori seperti terlihat pada Gambar 21. Setelah itu, anda tinggal mengikuti direktori sesuai dengan interpreter Python seperti terlihat pada Gambar 22.





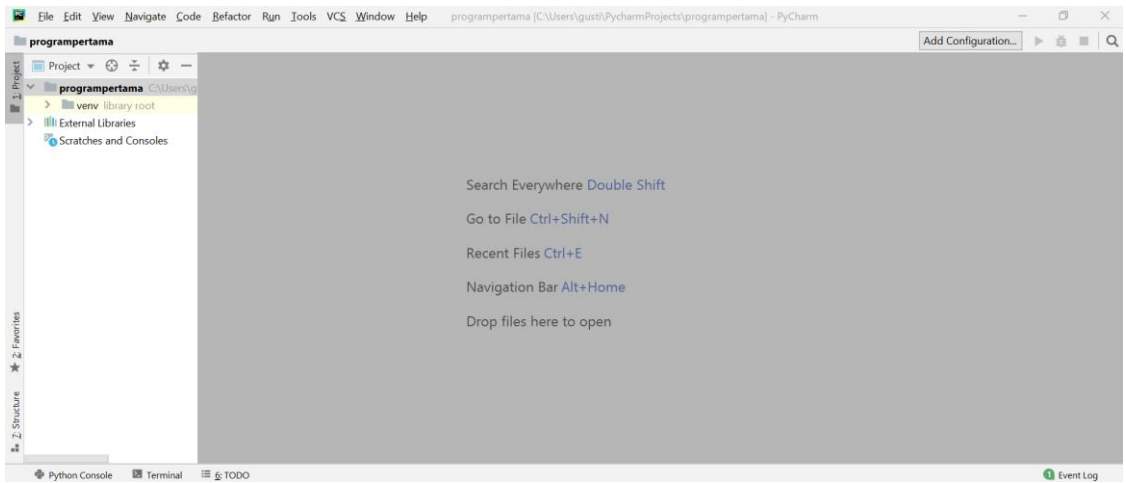
**Gambar 22 Pemilihan python.exe sebagai interpreter**



**Gambar 23 Jendela Selesai Pemilihan Interpreter**

Jika selesai, anda dapat melihat hasil pemilihan interpreter tersebut pada Gambar 23. Lalu klik “Create” dan jendela PyCharm akan muncul seperti pada Gambar 24.

Akhirnya, selamat anda telah berhasil menginstall dan menjalankan PyCharm untuk pertama kali. Selamat ber-eksplorasi....!!



**Gambar 24 Jendela Awal PyCharm**

## Minggu 2

### Variabel, Konsol I/O, dan Konversi Tipe Data

#### Tujuan

Setelah menyelesaikan praktikum ini, praktikan diharapkan mampu menggunakan variabel, operasi aritmatika, masukan dan keluaran dengan konsol, dan konversi tipe data.

#### Variabel

Apa itu variabel ? variabel merupakan suatu tempat yang tersedia di memori komputer untuk menyimpan data baik itu huruf, rangkaian huruf (ekuivalen dengan kata/kalimat), angka (bilangan bulat/decimal), atau karakter khusus. Variabel dapat dinyatakan dengan suatu string (rangkaihan karakter) dan memiliki aturan khusus dalam pendefinisianya. Berikut hal-hal yang perlu diperhatikan saat membuat variabel :

1. Deklarasi nama variabel hanya diperbolehkan untuk dimulai oleh karakter huruf atau simbol underscore (a-z, A-Z, \_), selanjutnya karakter dapat berupa huruf, nomor, atau \_.  
Contoh : `_panjang` (dibolehkan), `2panjang` (**tidak dibolehkan, error**).
2. Nama variabel di Python bersifat *case sensitive*, maksudnya adalah huruf kecil dan besar dianggap berbeda. Contoh : **Panjang** dengan **panjang** adalah dua variabel yang berbeda dikarenakan huruf "p" pada variabel pertama adalah huruf kapital sedangkan pada variabel kedua menggunakan huruf kecil.
3. Terdapat nama-nama atau kode khusus yang tidak dapat dijadikan variabel dikarenakan karakter tersebut merupakan fungsi atau pernyataan standar pada Python untuk menjalankan fungsi tertentu. Contoh : **in**, **for**, **if**, dll...
4. Operator yang digunakan dalam operasi aritmatika atau yang lainnya tidak dapat dijadikan sebagai unsur pembentuk variabel. Contoh : **+**, **-**, **/**, dll...

#### Praktikum 1 : Penggunaan Variabel

```

1  #Contoh definisi variabel untuk menampung karakter
2  kalimat = "Halo semua, saya adalah program komputer"
3  print(kalimat)
4  #Contoh definisi variabel untuk menampung bilangan bulat
5  bil_bulat = 7800
6  print(bil_bulat)
7  #Contoh definisi variabel untuk menampung bilangan desimal
8  bil_desimal = 10.087
9  print(bil_desimal)
10 #Contoh definisi dua variabel sekaligus dengan integer
11 a = b = 123
12 print(a, b)
13 #Contoh lainnya, nama variabel dapat mengandung angka

```

```

14 kalimat2 = "Bye bye..."
15 print(kalimat2)

```

### Latihan Belajar Mandiri

1. Cari tau makna dari # pada program diatas. Apakah pernyataan yang terdapat karakter tersebut dijalankan oleh program ?
2. Bagaimana kode program dapat dijalankan ? Dari atas ke bawah ?
3. Cari tau apa yang fungsi print lakukan pada program anda.
4. Cari tau dari berbagai sumber, nilai apa saja yang dapat ditampung oleh variabel ?

### Konsol I/O – Print

Konsol I/O merupakan singkatan dari konsol *input/output* yang menunjukkan masukan dan keluaran data atau informasi melalui konsol. Sedangkan konsol, yang sering dirujuk dengan kata sistem konsol atau terminal, sendiri adalah komputer standar yang terdiri dari monitor dan keyboard dimana hanya dapat menerima masukan berupa karakter dari perangkat masukan tersebut. Seperti program yang anda jalankan diatas hanya memunculkan teks tanpa melibatkan unsur grafis apapun.

Untuk dapat melakukan *output* / keluaran berupa teks di konsol, kita dapat menggunakan fungsi print. Berikut potongan kode program yang menggunakan fungsi tersebut.

#### Praktikum 2 : Print

```

1  jum_motor = 3
2  jum_rumah = 3
3  umur_saya = 18
4  rambut_saya = "Hitam"
5  print("Saya punya", jum_motor, "motor dan", jum_rumah, "rumah")
6  print(f"umur saya {umur_saya} tahun dan saya punya rambut warna
7  {rambut_saya}")
8
9  jenis_manusia = 2
10 biner = "tau bilangan biner"
11 not_biner = "tidak tau bilangan biner"
   x = f"Di dunia ini ada {jenis_manusia} jenis manusia yaitu yang {biner}
   dan yang {not_biner}"
12 print(x)
13
14 manusia = False
15 who = "program"
16 teks = "Lalu saya siapa ? Apakah saya manusia ?! {} , lalu saya apa ?
   {}"
17 print(teks.format(manusia, who))
18
19 print("""
20 Pergi wisata ke kota solo
21 Singgah sebentar di kota bantul

```

22	<code>Sungguh malang nasib si jomblo</code>
23	<code>Setiap malam hanya memeluk dengkul</code>
24	<code>""")</code>

### Latihan Belajar Mandiri

1. Buatlah komentar (“#”) untuk menjelaskan fungsionalitas dari masing-masing baris program diatas!
2. Carilah disemua baris pada program diatas, dimana terdapat string yang ditaruh kedalam string. Berapa jumlahnya ?

### Konsol I/O – Input

Contoh sebelumnya, untuk mencetak data anda menggunakan fungsi print, sedangkan untuk menerima masukkan data, kita dapat menggunakan fungsi input. Fungsi tersebut dapat ditaruh informasi berupa teks terkait data apa yang akan dimasukkan. Seperti, `input(“Nama anda : “)`.

#### Praktikum 3 : Input

1	<code>umur = input("Umur anda berapa ? ")</code>
2	<code>tinggi = input("Tinggi anda ? ")</code>
3	<code>weight = input("Kalau berat badan ? ")</code>
4	
5	<code>print(f"Jadi, umur anda {umur} tahun, tinggi {tinggi} cm. dan berat {weight} kg.")</code>

### Operasi Aritmatika

Dalam program, kita dapat melakukan perhitungan terhadap nilai atau variabel. Hasil operasi antar nilai dapat melalui penjumlahan, pengurangan, perkalian, pembagian, dan lain-lain. Operasi perhitungan di Python membutuhkan simbol yang mirip dengan simbol matematis yang disebut dengan operator. Sedangkan nilai yang dikomputasikan menggunakan operator disebut dengan operand. Berikut operator aritmatika yang dapat digunakan saat memprogram:

+	Penambahan
-	Pengurangan
*	Perkalian
/	Pembagian
**	Pangkat, 5 pangkat 2 = 5**2
//	Fungsi floor, 5//2 = 2
%	Modulus, 6 % 2 = 0, 4 % 3 = 1,

$2 \% 7 = 2$
--------------

#### Praktikum 4 : Aritmatika

```

1  # pustaka math berisi fungsi-fungsi dan konstanta
2  # matematika
3  import math
4
5  # Persegi Panjang
6  panjang = 1.10
7  lebar   = 2.20
8  luas    = panjang * lebar
9  print("Luas persegi panjang : " , luas)
10
11 #Luas Lingkaran
12 jari2   = 2.20
13 luas    = math.pi * jari2**2
14 print("Luas lingkaran : " , luas)
15
16 #Luas Permukaan Tabung
17 jari2Tabung = 2.20
18 tinggiTabung = 10
19 luasPermukaan = 2*math.pi*jari2Tabung*(jari2Tabung+tinggiTabung)
20 print("Luas Permukaan Tabung : " , luasPermukaan)

```

#### Latihan Belajar Mandiri

1. Manfaatkan internet, carilah konstanta dan fungsi lain yang terdapat pada modul *math* di dokumentasi python, <https://docs.python.org/3.8/index.html>. Kemukakan hasil temuan anda kepada asisten praktikum.

#### Konversi Tipe Data

Pada kode program sebelumnya, anda telah memprogram komputer untuk menerima masukan atau data dari pengguna. Sejatinya, apa yang anda dapatkan saat fungsi input dijalankan adalah berupa string yang mana operasi dengan bilangan tidak dapat dilakukan. Tidak semua operand dengan tipe data berbeda dapat dilakukan operasi dengan operator tertentu.

Tipe data dapat dikonversi ke tipe data lain agar dapat dilakukan operasi terhadap tipe data yang sama. Sebagai contoh variabel dengan tipe data string dengan nilai "2" tidak dapat dilakukan operasi penambahan dengan variabel bertipe data integer. Lihatlah potongan kode program berikut :

#ERROR

X = "234"

$$Y = 100 + X$$

Beberapa fungsi konversi tipe data di Python dapat dilihat pada tabel berikut :

int()	Merubah nilai ke tipe data integer
float()	Merubah nilai ke tipe data float / double
str()	Merubah nilai ke tipe data string

Praktikum 4 : Konversi Tipe Data	
1	<code>print(".:Program Luas Persegi Panjang:.\n")</code>
2	
3	<code>panjang = input("Masukkan panjang : ")</code>
4	<code>lebar = input("Masukkan lebar : ")</code>
5	<code>panjang = int(panjang)</code>
6	<code>lebar = int(lebar)</code>
7	<code>luas = panjang*lebar</code>
8	
9	<code>print("Luas Persegi Panjang : ", luas)</code>

#### *Latihan Belajar Mandiri*

1. Cobalah buat program diatas menjadi lebih singkat !

#### Latihan

1. Buatlah program untuk mencari luas permukaan tabung dimana jari-jari dan tinggi berasal dari masukkan pengguna !
2. Buatlah program untuk mengkonversi suhu dari celcius ke fahrenheit !
3. Buatlah program untuk menghitung gaji pegawai dalam satu minggu dimana masukkan pengguna berupa upah pegawai per jam dengan asumsi satu hari kerja adalah 8 jam!

## Minggu 3

### Eksekusi Kondisional

---

#### Tujuan

Setelah menyelesaikan praktikum ini, praktikan diharapkan mampu :

1. Menerapkan logika berpikir kondisional sederhana ke dalam Python melalui sintaks If, elif, dan else melalui operator komparasi.
2. Menerapkan logika berpikir kondisional bersarang dan majemuk.

#### Eksekusi Kondisional

Eksekusi kondisional merupakan suatu cara memprogram untuk membuat komputer dapat mengambil langkah ketika diberikan kondisi tertentu. Anda dapat bayangkan terdapat dua jenis potongan kode program, sebut saja A dan B. Potongan kode program A dapat dijalankan ketika suatu variabel melebihi nilai tertentu (misal x lebih dari 50), sebaliknya B dijalankan ketika kondisi tersebut tidak terpenuhi. Contoh dalam aplikasi nyata adalah saat anda membuat program konversi nilai angka ke nilai huruf sesuai peraturan akademik di ITK. Nilai masukkan angka dari konsol akan ditransformasikan kedalam nilai huruf tertentu jika memenuhi rentang nilai yang telah didefinisikan sebelumnya.

Dengan eksekusi kondisional, kita dapat menulis program yang berguna melalui pengecekan kondisi kapan suatu pernyataan akan dijalankan. Selain itu kita dapat memberikan perilaku pada program tergantung dari masukan pengguna. Sehingga kita dapat memberikan efek fleksibilitas pada program yang akan kita buat.

Lebih rincinya, suatu pernyataan akan dijalankan ketika kondisi terpenuhi atau ketika ekspresi boolean bernilai True. Sebaliknya, pernyataan tidak akan dieksekusi jika ekspresi bernilai False. Apa itu ekspresi boolean? Ekspresi boolean merupakan suatu ekspresi yang mengembalikan/memiliki suatu nilai boolean. Boolean sendiri memiliki nilai berupa Benar atau Salah yang dalam Python disebut dengan True atau False. Sebagai contoh :

```
X = 5 is 3
print(X)
print(type(X))
```

**Cobalah** kode program diatas sebelum masuk kedalam praktikum utama. Apa yang dapat anda simpulkan ?

Nilai boolean sendiri dapat dihasilkan melalui operator komparasi yang membandingkan dua nilai atau variabel. Berikut beberapa operator komparasi :

==	Sama dengan (nilai)
----	---------------------



!=	Tidak sama dengan (nilai)
>	Lebih besar dari
<	Kurang dari
>=	Lebih besar sama dengan
<=	Kurang dari sama dengan
is	Sama dengan (objek/referensi)
Is not	Tidak sama dengan (objek/referensi)

Untuk membuat program dapat melakukan pengecekan sesuai kondisi, diperlukan sintaks khusus. Penulisan sintaks kondisi dimulai dengan `if` yang diikuti dengan (...) yang berisi ekspresi boolean. Ketika ekspresi bernilai `True`, maka pernyataan (*statement*) setelahnya yang merupakan bagian dari eksekusi kondisional akan dieksekusi. Perlu dipastikan bahwa suatu pernyataan yang merupakan bagian dari `if` selalu merujuk ke dalam (1 tab kedepan dibandingkan dengan sintaks `if`). Sebagai contoh :

```
if(True) :
    print("Statement ini selalu dijalankan")

if(False) :
    print("Statement ini tidak akan pernah dijalankan")

print("Diluar dari eksekusi kondisional, selalu dijalankan")

if(10 == 10) :
    print("Statement ini selalu dijalankan karena 10 sama dengan 10")

if(10 == 1) :
    print("Statement ini tidak akan pernah dijalankan karena ekspresi bernilai False")
```

**Cobalah** potongan kode program diatas? Cetakkan mana yang tidak dieksekusi? Kenapa?

#### Praktikum 1 : Program Luas Bidang

```
1 import math
2
3 print(".: Program Luas Bidang :.\n")
4 print("1. Persegi")
5 print("2. Persegi Panjang")
6 print("3. Lingkaran")
7 pil = int(input("Pilihan anda ? "))
8
9 if(pil == 1):
10     sisi = float(input("Berapa sisi persegi ? "))
11     luasPersegi = sisi*sisi
12     print("Luas Persegi = ", luasPersegi)
13 elif(pil == 2):
14     panjang = float(input("Berapa panjang persegi panjang ? "))
```

```

15 lebar = float(input("Berapa lebar persegi panjang ? "))
16 luasPersegiPanjang = panjang*lebar
17 print("Luas Persegi Panjang = ", luasPersegiPanjang)
18 elif(pil == 3):
19     jari2 = float(input("Berapa jari-jari lingkaran ? "))
20     luasLingkaran = math.pi * jari2 * jari2
21     print("Luas Lingkaran = ", luasLingkaran)
22 else:
23     print("Masukan pilihan yang benar !")

```

### *Latihan Belajar Mandiri*

1. Apa makna dari **if**, **elif**, dan **else** pada program diatas ? Manfaatkanlah literatur yang tersedia bebas di internet.
2. Berilah komentar pada program diatas pada masing-masing kondisi.

### Praktikum 2 : Permainan Tebak Angka

```

1 import random, sys
2 print("::: Permainan Tebak Angka :::.\n")
3
4 input = int(input("Masukan angka 1 - 5 ? "))
5 if(input > 5):
6     sys.exit("Angka yang anda masukkan
    tidak valid! Masukan angka 1 - 5!")
7
8 com = random.randint(0, 5)
9 if(input == com):
10     print("Tebakkan anda BENAR!!")
11 else:
12     print("Tebakkan anda SALAH!!")

```

### *Latihan Belajar Mandiri*

1. Carilah maksud dari sys.exit dan random pada dokumentasi python yang telah dibahas pada pertemuan sebelumnya!
2. Apa saja fungsi yang tersedia pada modul random? Kemukakan temuan anda kepada asisten praktikum!

## Eksekusi Kondisional Bersarang

Sebelumnya kita mencoba kondisi yang sederhana tanpa adanya tingkatan seleksi. Pada eksekusi kondisional bersarang, suatu kondisi memiliki kondisi lainnya didalamnya. Dengan kata lain, terdapat percabangan kondisi yang mengharuskan kondisi lainnya untuk dipenuhi. Satu statement if memiliki statement if lain di dalamnya. Contoh potongan kode program untuk kondisi bersarang dapat dilihat sebagai berikut :

```

if 0 < x:
    if x < 10:
        print("X adalah bilangan bulat positif kurang dari 10")

```

Potongan kode program diatas menunjukkan eksekusi kondisional bersarang untuk pengecekan nilai bilangan bulat positif yang kurang dari 10. Kondisi pertama dilakukan dengan ekspresi boolean berupa  $0 < x$  yang menunjukkan bahwa  $x$  harus bernilai lebih dari 0 atau positif untuk dapat menjalankan statement didalamnya. Selanjutnya dilakukan pengecekan kondisi kembali dengan ekspresi boolean berupa  $x < 10$  yang menunjukkan nilai  $x$  harus kurang dari 10. Sehingga dengan seleksi bersarang tersebut statement terdalam dijalankan ketika kedua kondisi terpenuhi.

#### Praktikum 3 : Program Tahun Kabisat

```

1 print("::: Program Tahun Kabisat :::\n")
2
3 tahun = int(input("Tahun : "))
4
5 if tahun % 4 == 0:
6     if tahun % 100 == 0:
7         if tahun % 400 == 0:
8             print("Tahun ", tahun, " adalah tahun kabisat")
9         else:
10            print("Tahun ", tahun, " bukan tahun kabisat")
11    else:
12        print("Tahun ", tahun, " bukan tahun kabisat")
13 else:
14    print("Tahun ", tahun, " bukan tahun kabisat")

```

#### Eksekusi Kondisional Majemuk

Eksekusi kondisional majemuk merupakan suatu ekspresi boolean yang melibatkan operator logika seperti and (dan), or (atau), dan not (tidak/bukan). Dengan operator tersebut, banyak kondisi dapat diekspresikan dalam satu statement if.

Cara kerja operator logika mirip dengan penalaran manusia. Sebagai contoh, and mengharuskan suatu pernyataan bernilai benar semua untuk dikatakan benar. Dalam program, diharuskan nilai boolean bernilai True sebagai operand untuk operator and. Pada or, hanya salah satu saja bernilai True dapat dikatakan ekspresi tersebut bernilai benar. Sedangkan not merupakan negasi nilai boolean. Jika nilai boolean bernilai True, maka not menghasilkan False. Hasil operasi pada operator logika dengan boolean sebagai operand dapat dilihat pada tabel dibawah :

Nilai boolean 1	Nilai boolean 2	Hasil operator <b>and</b>
True	True	<b>True</b>
True	False	<b>False</b>
False	True	<b>False</b>

False	False	False
-------	-------	-------

Nilai boolean 1	Nilai boolean 2	Hasil operator <b>or</b>
True	True	<b>True</b>
True	False	<b>True</b>
False	True	<b>True</b>
False	False	<b>False</b>

Nilai boolean	Hasil operator <b>not</b>
True	<b>False</b>
False	<b>True</b>

```
if 0 < x and x < 10:
    print("X adalah bilangan bulat kurang dari 10")
```

Pada potongan kode program diatas, seleksi dilakukan dengan melibatkan operator and dimana kedua kondisi tersebut harus terpenuhi atau bernilai True agar statement di dalamnya dapat dieksekusi. Jika x lebih dari 0 dan x kurang dari 10, maka cetak x adalah bilangan bulat kurang dari 10. Akan tetapi, ketika kondisi hanya terpenuhi salah satu saja, misal x = 100, maka kondisi kedua x < 10 bernilai False. Menyebabkan nilai ekspresi boolean menjadi False diakibatkan salah satu ekspresi bernilai False (True and False = False)

Contoh penerapan dari eksekusi kondisional majemuk adalah program indeks massa tubuh yang merupakan program untuk menentukan kriteria kegemukan berdasarkan Indeks Massa Tubuh (IMT) dengan rumus  $IMT = \frac{\text{berat badan}}{\text{tinggi}^2}$  dimana berat badan dalam kg. dan tinggi dalam m. Pada program ini pengguna diminta memasukkan data yang familiar di masyarakat dengan menggunakan data berat badan dalam kilogram dan data tinggi badan dalam centimeter.

#### Praktikum 4 : IMT

```
1 print("::: Program Indeks Massa Tubuh :::")
2
3 beratBadan = int(input("Berat Badan (kg.) : "))
4 #Tinggi Badan dalam cm
5 tinggiBadanCM = int(input("Tinggi Badan (cm.) : "))
6 #Tinggi Badan dalam meter
7 tinggiBadanM = tinggiBadanCM/100
8 imt = beratBadan/(tinggiBadanM**2)
9 kriteria = ""
10 if imt <= 18.5:
11     kriteria = "Kurus"
12 elif 18.5 < imt and imt <= 25:
13     kriteria = "Normal"
14 elif 25 < imt and imt <= 30:
```

```

15 kriteria = "Gemuk"
16 else:
17     kriteria = "Kegemukan (Obesitas)"
18
19 print("Kriteria tubuh anda berdasarkan IMT adalah", kriteria)

```

### Tips 💡 | Gaya Pada Konsol

Untuk membuat tampilan program tidak monoton, gaya konsol dapat diubah dengan karakter spesial. Karakter spesial \033[xxm merupakan karakter untuk memberikan instruksi pada konsol untuk merubah gaya teks seperti warna, background, huruf tebal, dan lain-lain. Sedangkan kode 0m pada \033 menunjukkan gaya yang dimurnikan / reset style. Kode umum yang sering digunakan dapat dilihat dibawah :

\033[30m hitam	\033[40m 	\033[1m bright
\033[31m merah	\033[41m 	\033[22m normal
\033[32m hijau	\033[42m 	\033[0m reset all
\033[33m kuning	\033[43m 	
\033[34m biru	\033[44m 	
\033[35m magenta	\033[45m 	
\033[36m cyan	\033[46m 	
\033[37m "putih"	\033[47m 	

Contoh aplikasi lain adalah permainan suit/pingsut yang merupakan permainan dimana pengguna diminta untuk memasukkan pilihan telunjuk (manusia), jempol (gajah), atau kelingking (semut). Pilihan komputer didapatkan berdasarkan nilai random integer dengan rentang antara 1 - 3 seperti terlihat pada tabel berikut :

Nilai random integer (X)	Pilihan
X == 1	jempol (gajah)
X == 2	telunjuk (manusia)
X == 3	kelingking (semut)

### Praktikum 5 : Permainan Suit/Pingsut

```

1 import random, sys
2
3 print("::: Permainan Suit/Pingsut :::")
4
5 print("1. Jempol (Gajah)")
6 print("2. Telunjuk (Manusia)")
7 print("3. Kelingking (Semut)")
8 pil = int(input("Pilihan anda ? "))
9 if(pil < 1 or pil > 3):
10     sys.exit("Masukkan pilihan yang benar!! Pilihan antara 1 - 3.")
11 #Pilihan komputer
12 kom = random.randint(1,3)

```

```

13 if(kom == 1):
14     if(pil == 1):
15         print("Sama-sama Gajah! sesama gajah saling membantu...")
16     if(pil == 2):
17         print("Diinjek gajah.. kamu kalah!")
18     if(pil == 3):
19         print("Kamu gigit gajah, kamu menang!")
20 elif(kom == 2):
21     if(pil == 1):
22         print("Kamu abis nginjek manusia, kamu menang!")
23     if(pil == 2):
24         print("Sama-sama Manusia! Jangan berantem lah...")
25     if(pil == 3):
26         print("Kamu dibunuh manusia, kamu kalah!")
27 elif(kom == 3):
28     if(pil == 1):
29         print("Kamu abis dikerjain sama semut, kamu kalah!")
30     if(pil == 2):
31         print("Kamu gak sengaja injek semut, kamu menang!")
32     if(pil == 3):
33         print("Sesama semut saling membahu..!")

```

## Latihan

1. Modifikasilah kode program pada Praktikum 5 diatas dengan melibatkan gaya konsol agar lebih menarik.
2. Buatlah program penerjemah protein yang menerima masukkan berupa kodon dan memberikan keluaran berupa nama protein. Sebagai contoh jika masukkan berupa "UUU" atau "UUC" maka menghasilkan "Phenylalanine". Terjemahan protein dapat dilihat dibawah :

Kodon	Protein
AUG	Methionine
UUU, UUC	Phenylalanine
UUA, UUG	Leucine
UCU, UCC, UCA, UCG	Serine
UAU, UAC	Tyrosine
UGU, UGC	Cysteine
UGG	Tryptophan

3. Buatlah program untuk menghitung akar-akar persamaan kuadrat dengan rumus :

$$D = b^2 - 4 \times a \times c$$

Jika  $D < 0$ , akar imajiner

Jika  $D = 0$ ,  $x_1 = x_2$  yang didapat dari  $-b/(2 \times a)$

Jika  $D > 0$ , maka terdapat dua akar yaitu  $x_1 = \frac{-b+\sqrt{d}}{2a}$  dan  $x_2 = \frac{-b-\sqrt{d}}{2a}$

4. Buatlah program chatbot dimana anda dapat bertukar pesan / chat dengan komputer terkait bahasa sehari-hari.

## Minggu 4

### Perulangan dengan While

---

#### Tujuan

Setelah menyelesaikan praktikum ini, praktikan diharapkan mampu membuat program sederhana dengan melibatkan perulangan while.

#### While

Pada suatu kasus, untuk menyelesaikan masalah tertentu, kita perlu menuliskan beberapa perintah yang mirip berulang kali. Ketika pernyataan yang kita tulis saat memprogram membutuhkan lebih dari 100 kali perulangan maka hal tersebut akan memakan waktu. Oleh sebab itu, bahasa pemrograman memiliki kemampuan dalam mengulang perintah berkali-kali sesuai dengan kondisi pemrogram. Salah satu pernyataan yang dapat kita gunakan adalah dengan menggunakan `while`.

While merupakan *syntax* untuk membuat suatu perulangan berdasarkan kondisi tertentu. Sesuai dengan maknanya yaitu “selama” (*while*), statement yang masuk dalam blok while akan terus diulang selama kondisi terpenuhi. While memerlukan ekspresi boolean sebagai penentu perulangan terus dijalankan atau tidak. Jika ekspresi bernilai True, maka *statement* dijalankan. Setelah semua *statement* didalamnya selesai dieksekusi, program akan kembali mengeksekusi ekspresi boolean pada while (berulang). Begitu sebaliknya jika False, maka *statement* didalamnya tidak akan dijalankan dan akan berlanjut pada eksekusi *statement* selanjutnya.

#### Praktikum 1 : Bilangan Genap

```
1 print("::: Program Bilangan Genap :::\n")
2
3 batas = int(input("Batas Atas Bilangan Genap : "))
4
5 i = 1
6 while(i <= batas):
7     if(i % 2 == 0):
8         print(i, end=" ")
9     i+=1;
```

#### Latihan Belajar Mandiri

1. Berilah komentar tiap baris pada kode program diatas!
2. Cari tau maksud dari `end=" "` pada fungsi `print`, beserta parameter lain yang terdapat pada fungsi tersebut!



## Praktikum 2 : Daftar Perkalian

```

1 print("::: Program Daftar Perkalian :::.\n")
2
3 bil = int(input("Masukkan Bilangan : "))
4
5 i = 1
6 while(i <= 10):
7     print(f"{bil} x {i} = {bil*i}")
8     i+=1

```

## Praktikum 3 : Program Luas dan Keliling Lingkaran

```

1 import math
2 print("::: Program Luas dan Keliling Lingkaran :::.\n")
3
4 pil = -1
5 while(pil != 3):
6     print("MENU")
7     print("1. Luas Lingkaran")
8     print("2. Keliling Lingkaran")
9     print("3. Keluar")
10    pil = int(input("Pilihan Anda ? "))
11
12    if(pil == 1):
13        jari2 = float(input("Masukkan Jari-Jari : "))
14        luas = math.pi * jari2**2
15        print("\033[1mLuas Lingkaran = ", luas, "\033[0m")
16    elif(pil == 2):
17        jari2 = float(input("Masukkan Jari-Jari : "))
18        keliling = 2*math.pi*jari2
19        print("\033[1mKeliling Lingkaran = ", keliling, "\033[0m")
20    elif(pil == 3):
21        print("Bye bye... program dimatikan")
22    else:
23        print("\033[31mMasukkan pilihan yang benar!\033[0m")

```

*Latihan Belajar Mandiri*

1. Bisakah anda tebak, sebelum program dijalankan, kapan program diatas akan berhenti ?
2. Dapatkah anda menambahkan satu menu tambahan dari program diatas terkait luas/keliling bidang ?

## Praktikum 4: Tebak Angka

```

1 import random
2 print("::: Permainan Tebak Angka :::.\n")
3
4 komp = random.randint(1,10)

```

```

5
6 print("Komputer telah memikirkan suatu angka dari 1 - 10.
  Tebak angka tersebut!")
7
8 kesempatan = 3
9
10 while(kesempatan > 0):
11     angka = int(input("Tebakkan saya : "))
12     if(angka < 1 or angka > 10):
13         print("Masukkan angka yang benar!
  Angka yang dipikirkan komputer antara 1 - 10")
14     elif(angka == komp):
15         print("Tebakkan anda benar!, komputer memikirkan angka ", angka)
16         kesempatan = 0
17     else:
18         kesempatan-=1
19         if(kesempatan == 0):
20             print("Salah!!, angka yang dipilih komputer adalah", komp)
21         else:
22             print("Tebakkan anda salah! Coba lagi.")
23             print("Sisa kesempatan = ", kesempatan)

```

### *Latihan Belajar Mandiri*

1. Cobalah modifikasi program diatas agar terlihat lebih menarik dengan menambahkan warna pada konsol.

### Latihan

1. Buatlah program untuk mencetak bilangan ganjil dari N sampai dengan 1 dimana N adalah bilangan bulat masukkan pengguna. Sebagai contoh ketika pengguna memasukkan 10, maka komputer akan mencetak 9 7 5 3 1.
2. Buatlah program luas dan keliling bidang persegi, persegi panjang, dan segitiga dengan pilihan menu. Selama menu keluar tidak dipilih, program terus berjalan.
3. Buatlah program untuk mencari nilai faktorial. Misal dimasukkan angka 4 maka hasilnya adalah 24 ( $4! = 4 \times 3 \times 2 \times 1 = 24$ ). Keluaran harus mencetak penjabaran dari faktorial tersebut. Contoh :  
 Input : 4  
 Output :  $4! = 4 \times 3 \times 2 \times 1 = 24$
4. Buatlah sebuah program pembangkit password otomatis dengan panjang karakter sebagai masukkan pengguna.

## Minggu 5

### Perulangan dengan For

---

#### Tujuan

Setelah menyelesaikan praktikum ini, praktikan diharapkan mampu membuat program sederhana dengan melibatkan perulangan for.

#### Definite Loop dengan For

Praktikum sebelumnya mendemonstrasikan bahwa perulangan dapat dilakukan dengan memberikan kondisi tertentu. Selama kondisi tersebut terpenuhi, perulangan akan terus dilakukan. Perulangan lain yang dapat dilakukan selain adanya kondisi adalah dengan menentukan jumlah perulangan berapa kali akan dilakukan. Cara ini mengharuskan programmer mengetahui jumlah perulangan. Untuk melakukannya, kita dapat memanfaatkan *For*.

*For* merupakan syntax untuk membuat suatu perulangan dengan jumlah yang telah diketahui. Perulangan for akan mengulang blok kode program disertai nilai pada rentang tertentu secara berurutan. Rentang nilai tersebut didapatkan melalui fungsi *range* yang menghasilkan nilai bilangan bulat antara nilai awal sampai dengan batas akhir. Walaupun fungsi *range* sering digunakan dalam perulangan for, fungsi tersebut bukanlah satu-satunya cara untuk melakukan perulangan. For juga dapat digunakan untuk mengunjungi setiap elemen dalam list. Berikut contoh penggunaan for dalam berbagai macam kasus :

```
for i in range(0,10):
    print(i)
```

Pada potongan kode program diatas, fungsi *range(0,10)* menghasilkan rentang nilai antara nol sampai sembilan. Nilai 10 pada parameter kedua dalam fungsi tersebut merupakan batas atas yang tidak akan dilibatkan dalam rentang nilai. Secara *default* nilai dari parameter pertama adalah nol, sehingga jika nilai 10 saja yang dilibatkan sebagai parameter pertama, maka rentang yang dimaksud dimulai dengan nilai nol. Berikut contoh hasil rentang nilai bilangan bulat yang dihasilkan oleh fungsi *range* :

```
range(0,10) => 0,1,2,3,4,5,6,7,8,9
range(10) => 0,1,2,3,4,5,6,7,8,9
range(2,7) => 2,3,4,5,6
range(4) => 0,1,2,3
```

Praktikum 1 : Program Bilangan Ganjil dan Jumlahnya	
1	<code>print(".: Program Bilangan Ganjil dan Jumlahnya :.\n")</code>
2	
3	<code>batas = int(input("Batas Atas Bilangan Ganjil : "))</code>
4	

```

5 print("Bilangan Ganjil : ", end = "")
6 sum = 0
7 for i in range(batas+1):
8     if(i % 2 == 1):
9         print(i, end=" ")
10        sum+=i
11
12 print("\nJumlah : ", sum)

```

#### Praktikum 2 : Program Prediksi Investasi

```

1 print("::: Program Prediksi Investasi :::\n")
2
3 saldo = int(input("Masukkan Saldo Awal (Rp.) : "))
4 persentase = float(input("Masukkan
Persentase Keuntungan Per Tahunnya (%) : "))
5
6 tahun = int(input("Masukkan Waktu Investasi (Tahun) : "))
7 print("Tahun Ke-\t Saldo Awal\t\t Laba Investasi\t\t Saldo Akhir")
8
9 for i in range(tahun):
10     laba = saldo*(persentase/100)
11     saldoAkhir = saldo+laba
12     print(f"{i}\t\t {saldo}\t\t\t {laba}\t\t\t {saldoAkhir}")
13     saldo = saldoAkhir

```

#### Latihan Belajar Mandiri

1. Cobalah buat hasil tampilan rincian investasi pada program diatas terlihat lebih rapi dengan cara membuat nilai hanya mengandung dua angka dibelakang koma.

#### Break dan Continue

*Break* merupakan suatu pernyataan untuk keluar dari perulangan. Pernyataan ini dapat digunakan baik dalam perulangan dengan *while* maupun *for*. Sedangkan *continue* merupakan suatu pernyataan untuk melanjutkan perulangan tanpa harus mengeksekusi seluruh blok kode program yang ada dalam perulangan.

Sebagai contoh pada potongan kode program dibawah, perulangan seharusnya dilakukan sebanyak lima kali. Namun, ketika kondisi *i* sama dengan dua terpenuhi maka perulangan dihentikan.

```

for i in range(5):
    print("Perulangan Ke-", i+1)
    if(i == 2):
        break

```

Cobalah program diatas sebelum masuk pada praktikum utama!

Sedangkan contoh penggunaan continue dapat dilihat pada potongan kode program dibawah. Perulangan dilakukan sebanyak tiga kali dan saat i sama dengan 1 maka perulangan dilanjutkan tanpa mengeksekusi pernyataan dibawahnya.

```
for i in range(3):
    print("Perulangan Ke-", i+1)
    if(i == 1):
        continue;
    print("Perulangan Ke-", i+1)
```

Cobalah program diatas sebelum masuk pada praktikum utama!

#### Praktikum 3 : Penjumlahan Bilangan Positif

```
1 print(".: Program Penjumlahan 5 Bilangan Positif :.\n")
2
3 sum = 0
4 for i in range(5):
5     n = float(input(f"Bilangan ke-{i+1} : "))
6     if(n <= 0):
7         continue
8     sum += n
9 print("Hasil Penjumlahan : ", sum)
```

#### Latihan Belajar Mandiri

1. Ubahlah program diatas agar hanya dapat menerima masukan berupa bilangan bulat positif, jika terdapat bilangan negatif atau desimal maka perulangan otomatis berhenti.

#### Perulangan Bersarang

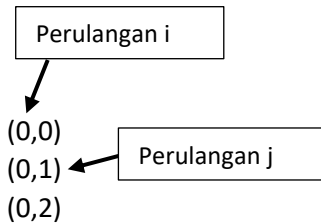
Perulangan bersarang (nested loop) merupakan suatu perulangan yang memiliki perulangan di dalamnya. Sebagai contoh perulangan sebanyak tiga kali dapat diulang sebanyak tiga kali pula pada tiap perulangannya.

```
for i in range(3):
    for j in range(3):
        print(f"({i}, {j})")
```

Cobalah program diatas sebelum masuk pada praktikum utama!

Pada potongan kode program diatas, perulangan bagian luar (*outer loop*) yang ada pada variabel i melakukan pengulangan pada blok kode program di dalamnya sebanyak tiga kali.

Sedangkan pada perulangan bagian dalam (*inner loop*) melakukan hal yang sama yaitu perulangan sebanyak tiga kali tergantung dari *outer loop*. Proses eksekusi program dapat dilihat dari tiap langkah eksekusi program dijalankan. Pertama-tama perulangan ke *i* akan masuk dengan nilai  $i = 0$ , lalu mengulang sebanyak 3 kali dengan nilai  $j = 0, 1, 2$ . Sehingga pada perulangan pertama untuk  $i = 0$  akan menghasilkan :



Untuk perulangan *i* selanjutnya menyesuaikan dengan banyaknya perulangan sehingga hasil akhir yang akan didapatkan adalah sebagai berikut :

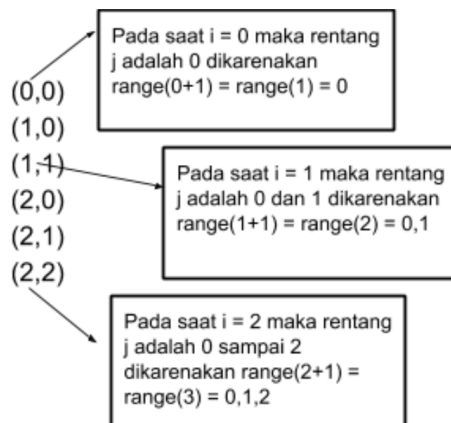
```

(0,0)
(0,1)
(0,2)
(1,0)
(1,1)
(1,2)
(2,0)
(2,1)
(2,2)
  
```

Perulangan pada bagian dalam (*inner loop*) dapat dipengaruhi oleh variabel perulangan yang ada pada perulangan bagian luar (*outer loop*). Sebagai contoh jika inner loop dengan variabel *j* dependen atau tergantung pada variabel *outer loop* sebanyak *i*. Maka hasilnya akan menyesuaikan dengan perulangan luarnya. Contoh kode program dan hasil :

```

for i in range(3):
    for j in range(i+1):
        print(f"({i}, {j})")
  
```



Cobalah program diatas sebelum masuk pada praktikum utama!

#### Praktikum 4 : Pencetakan Segitiga Siku-siku

```

1 print("::: Program Segitiga Siku-Siku :::\n")
2
3 n = int(input("Masukkan sisi : "))
4
5 for i in range(n):
6     for j in range(i+1):
7         print("x", end="")
8     print()

```

#### Latihan

1. Buatlah program untuk mencetak deret harmonik berdasarkan inputan pengguna (N). Misalkan N adalah 5 maka program akan mencetak :  
 $1 + (1/2) + (1/3) + (1/4) + (1/5) = 2.283333333$
2. Buatlah program untuk menampilkan deret bilangan fibonacci sampai suku ke N ! Misalkan N adalah 7 maka program akan mencetak (tanda baca koma juga muncul pada layar namun pada urutan terakhir tidak muncul) :  
1, 1, 2, 3, 5, 8, 13
3. Buatlah program untuk mencari cicilan rumah per tahun dengan diketahui harga rumah asal, harga rumah yang dijual ke klien, dan lama waktu cicilan dengan skema 20, 15, 10, dan 5 tahun.

4. Buatlah program dengan perulangan bersarang untuk mencetak (misal yang dimasukkan = 5):

a.

```
XXXXXX
XXXXX
XXX
XX
X
```

b.

```
xxxxxx
-----
xxx
--
x
```

c.

```
1
12
123
1234
12345
1234
123
12
1
```

d.

```
x
xx
xxx
xxxx
xxxxx
xxxxxx
xxxxx
xxx
xx
x
```

e.

```
. . . . 5
. . . 4
```



. . 3  
. 2  
1

## Minggu 6

### List dan Dictionary

---

#### Tujuan

Setelah menyelesaikan praktikum ini, praktikan diharapkan mampu membuat program sederhana dengan melibatkan struktur data list dan dictionary.

#### List

List merupakan struktur data yang dapat menampung rangkaian nilai atau objek. Anda dapat bayangkan jika terdapat daftar nilai ipk mahasiswa yang cukup banyak, maka membuat variabel satu per satu untuk menampung nilai tersebut akan menjadi tugas yang membosankan. Dengan variabel yang ada pun, kita tidak dapat memanfaatkan perulangan yang nantinya akan berguna dalam membangun aplikasi. Sehingga, list memberikan opsi kepada memprogram untuk dapat menampung banyak variabel/nilai kedalam suatu container.

Untuk dapat membuat list, yang menampung banyak nilai, kurung siku ([]) menjadi syntax utama dimana nilai akan ditaruh didalamnya dengan pemisah berupa tanda koma (,). Berikut contoh pembuatan list :

```
kopi = ["Cappucino", "Mocacino", "Espresso"]
angka = [17, 200]
kosong = []
gabungan = [100, "huruf", ["list dalam list"]]
print(kopi, angka, kosong, gabungan)
```

Cobalah potongan kode program diatas, sebelum praktikum utama dimulai!

List sifatnya adalah *mutable* artinya dapat dimutasi atau dirubah nilainya. Untuk dapat merubah nilai pada list, kita memerlukan indeks tertentu yang akan kita rubah. Misal kita akan merubah nilai list angka pada indeks pertama dengan memasukkan angka 1 pada kurung siku. Indeks list dimulai dari 0 sampai panjang list yang dapat diakses melalui fungsi `len()`. Dengan cara tersebut, kita dapat memanfaatkan perulangan terhadap indeks dari list. Berikut contoh perubahan nilai pada list :

```
angka[1] = 999
```

## Praktikum 1: Statistik sederhana dari list

```

1  jum_bil = int(input("Masukkan banyaknya jumlah bilangan? "))
2  daftar_bil = []
3  total = 0
4  for i in range(jum_bil):
5      bil = float(input(f"Bilangan ke-{i+1}: "))
6      daftar_bil.append(bil)
7      total += bil
8
9  print("Bilangan yang telah terdata", daftar_bil)
10 print("Rata-rata:", total/len(daftar_bil))
11 print("Maksimum: ", max(daftar_bil))
12 print("Minimum: ", min(daftar_bil))

```

*Latihan Belajar Mandiri*

1. Cari taulah apa saja fungsi yang dapat digunakan terhadap list selain max dan min ! Kemukakan temuan anda kepada asisten.
2. Tambahlah kode program diatas secara manual agar mampu menghasilkan standard deviasi pada data di list!

## Praktikum 2: Penjumlahan Matriks (List dua dimensi)

```

1  a = [[1, 2, 3], [4, 5, 6]]
2  b = [[7, 8, 9], [10, 11, 12]]
3  c = []
4
5  print('Matriks A')
6  for x in a:
7      print(x)
8  print('Matriks B')
9  for x in b:
10     print(x)
11  for i in range(len(a)):
12     e1 = []
13     for j in range(len(a[0])):
14         jum = a[i][j] + b[i][j]
15         e1.append(jum)
16     c.append(e1)
17  print("Hasil Penjumlahan")
18  for x in c:
19     print(x)

```

*Latihan Belajar Mandiri*

1. Tambahlah kode program diatas agar dapat menghitung selisih dan nilai mutlak dari selisih antar dua matriks tersebut!

## Dictionary

Pada praktikum sebelumnya anda telah menggunakan struktur data list yang menggunakan bilangan bulat untuk dapat mengakses nilai dari list tersebut. Dictionary tidak jauh berbeda dengan list, namun yang membedakan adalah cara aksesnya yang tidak hanya terpaku pada bilangan bulat. Cobalah potongan kode program berikut sebelum masuk pada praktikum utama.

```
a = dict()
a['10111'] = 'Agus'
a['11222'] = 'Rohan'
print(a)
b = {1: 'Januari', 2: 'Februari', 'Januari' : 1, 'Februari' : 2}
print(b[2])
print(b['Januari'])
print(b)
```

### Praktikum 3: Toko Buah

```
1 stock = {
2     'pisang': 10,
3     'jambu': 5,
4     'apel': 0,
5     'durian': 7,
6     'manggis': 1
7 }
8 harga = {
9     'pisang': 10000,
10    'jambu': 15000,
11    'apel': 20000,
12    'durian': 15000,
13    'manggis': 5000
14 }
15 print('-----TOKO BUAH-----')
16 print('Daftar Buah')
17 for key in stock:
18     print(key, end=", ")
19 print()
20 buah = input('Buah apa yang ingin anda beli? ')
21 jumlah = int(input('Berapa jumlah buah yang anda beli? '))
22 if(stock[buah]>= jumlah):
23     stock[buah] = stock[buah] - jumlah
24     print("Total harga pembelian", harga[buah]*jumlah)
25     print(f"Sisa stock buah {buah} : {stock[buah]}")
26 else:
27     print('Maaf, stock kami tidak mencukupi')
```

## Latihan

1. **Buatlah program dimana pengguna bisa membuat matriks dan mengubah nilai matriks berdasarkan kolom dan barisnya. Program disertai dengan manu dan akan berhenti ketika pengguna memilih menu keluar!**
2. **Buatlah program perkalian antar matriks!**

## Minggu 7

### String

#### Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu menggunakan fungsi, operator, dan pemformatan pada string

#### String sebagai Urutan Karakter

Sama seperti list yang merupakan urutan dari suatu nilai / objek, string merupakan urutan dari suatu karakter. Sehingga cara akses terhadap suatu karakter menggunakan sintaksis yang sama dengan list. Sebagai contoh :

```
pulau = 'kalimantan'
print(pulau[2])
-> l
print(pulau[-2])
-> a
```

Potongan kode program diatas menunjukkan cara akses karakter dalam suatu string. Akses karakter menggunakan indeks yang dimulai dari 0 sebelah kiri string. Sedangkan untuk indeks yang dimulai dari sebelah kanan dimulai dengan -1. Gambaran hubungan antara index dengan karakter pada string dapat dilihat pada gambar berikut :

k	a	l	i	m	a	n	t	a	n
0	1	2	3	4	5	6	7	8	9
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Pemotongan pada string memiliki aturan yang sama dengan pemotongan pada list. Contoh :

```
pulau = 'kalimantan'
print(pulau[2:7])
-> liman

print(pulau[4:])
-> mantan

print(pulau[:4])
-> kali
```

```
print(pulau[:])
-> kalimantan
```

### Kunjungan Karakter pada String

Karakter dapat dikunjungi satu per satu melalui perulangan langsung terhadap nama string.  
Contoh :

```
for k in pulau:
    print(k)

for i in range(len(pulau)):
    print(pulau[i])
```

Fungsi len pada string menghasilkan jumlah karakter termasuk spasi.

### Operator pada String

Operator yang dapat digunakan pada string dapat dilihat pada tabel berikut :

in	<p>Operator boolean untuk melakukan pengecekan antara dua string dimana suatu string ada atau merupakan subset dari string yang lain. Contoh :</p> <pre>'a' in 'kalimantan' -&gt; True  'minta' in 'kalimantan' -&gt; False</pre>
==	<p>Operator komparasi yang membandingkan kesamaan antara dua string.</p> <pre>if pulau == 'kalimantan':     print("Pulau Kalimantan merupakan pulau terbesar di Indonesia") else:     print("Bukan pulau")</pre>
>	<p>Operator komparasi membandingkan string yang kurang dari string pembandingnya</p> <pre>print('kali' &lt; pulau) -&gt; True</pre>
<	<p>Operator komparasi membandingkan string yang lebih dari string pembandingnya</p> <pre>print('kalimantana' &gt; pulau) -&gt; True</pre>

## Metode pada String

Daftar metode yang dapat digunakan pada string dapat dilihat pada tabel berikut :

<code>capitalize()</code>	Mengembalikan nilai berupa kapitalisasi dari string. Setiap karakter awal pada kata akan menjadi huruf besar.
<code>upper()</code>	Mengembalikan nilai berupa huruf kapital untuk semua karakter.
<code>lower()</code>	Mengembalikan nilai berupa huruf kecil dari string.
<code>startswith(str)</code>	Metode yang mengembalikan nilai berupa boolean yang didapat dari apakah suatu string dimulai dari string argumen pertama.
<code>endswith(str)</code>	Metode yang mengembalikan nilai berupa boolean yang didapat dari apakah suatu string diakhiri dari string argumen pertama.
<code>split(arg)</code>	Membagi string menjadi beberapa bagian yang dipisahkan dengan argumen pertama.
<code>find(sub, start, end)</code>	Mengembalikan index dari sub string (argumen pertama) yang dapat dimulai melalui index pada argumen kedua dan batas index pada argumen ketiga. Argumen dua dan tiga tidak diharuskan untuk dispesifikasikan.
<code>strip()</code>	Menghilangkan spasi yang berlebih dari suatu string.

### Praktikum 1: Singkatan

```

1 print(".: Program Singkatan .:")
2
3 kalimat = input("Masukkan Kalimat : ")
4 singkatan = ""
5 daftar_kata = kalimat.strip().split(" ")
6 for kata in daftar_kata:
7     singkatan += kata[0].upper()
8
9 print(f"Singkatan : {singkatan}")

```

**Program Komplemen RNA** : Program untuk memetakan kode gen DNA kedalam bentuk urutan komplemen nukleotida RNA.

Masing-masing gen DNA dipetakan dengan aturan berikut :

G -> C

C -> G

T -> A



A -> U

Keterangan :

4 Nukleotida yang ada pada DNA adalah Adenine (A), Cytosine (C), Guanine (G), Thymine (T)

4 Nukleotida yang ada pada RNA adalah Adenine (A), Cytosine (C), Guanine (G), Urasil (U)

#### Praktikum 2: RNA

```

1 import sys
2 print("::: Program Transkripsi RNA ::.")
3
4 dna = input("masukkan string DNA : ")
5 dna = dna.upper()
6 rna = ""
7 for x in dna:
8     if(x == 'G'):
9         rna += 'C'
10    elif(x == 'C'):
11        rna += 'G'
12    elif(x == 'T'):
13        rna += 'A'
14    elif(x == 'A'):
15        rna += 'U'
16    else:
17        sys.exit("Bukan DNA !")
18        break;
19
20 print(f"string RNA : {rna}")

```

#### Praktikum 3: Tanda Kurung

```

1 import sys
2 print("::: Program Pengecekan Tanda Kurung ::.")
3
4 stack = []
5 input_string = input("Masukkan kalimat : ")
6 for i in input_string:
7     if i in '[((':
8         stack.append(i)
9     elif i in ')])':
10        if stack:
11            if i == ']' and '[' != stack.pop():
12                sys.exit("Tanda [] salah !")
13            if i == '}' and '{' != stack.pop():
14                sys.exit("Tanda {} salah !")

```

```

15         if i == ')' and '(' != stack.pop():
16             sys.exit("Tanda ( ) salah !")
17     else:
18         sys.exit("Tidak ada tanda kurung di awal !")
19
20 print("Kalimat Sudah Benar...")

```

## Latihan

1. Buatlah program untuk membalik suatu kalimat dan hitung jumlah huruf vokalnya !
2. Buatlah program translasi protein dari RNA yang dimasukkan !

RNA dapat dibagi menjadi 3 urutan nukleotida yang disebut kodon dan selanjutnya diubah ke polipeptida

RNA : "AUGUUUUCU" => diterjemahkan ke

Kodon : "AUG", "UUU", "UCU" yang menjadi polipeptida dengan urutan sebagai berikut =>

Protein : "Methionine", "Phenylalanine", "Serine".

Aturan penerjemah protein adalah sebagai berikut :

Kodon	Protein
AUG	Methionine
UUU, UUC	Phenylalanine
UUA, UUG	Leucine
UCU, UCC, UCA, UCG	Serine
UAU, UAC	Tyrosine
UGU, UGC	Cysteine
UGG	Tryptophan
UAA, UAG, UGA	STOP

Ketika UAA, UAG, atau UGA ditemukan dalam urutan RNA, maka program dihentikan.

3. Buatlah program untuk memverifikasi ISBN-10 yang terdiri dari 9 digit + 1 karakter yang dapat berupa angka (0-9) atau X (representasi dari 10). Rumus pengecekan kode ISBN-10 adalah sebagai berikut :

$$(x_1 * 10 + x_2 * 9 + x_3 * 8 + x_4 * 7 + x_5 * 6 + x_6 * 5 + x_7 * 4 + x_8 * 3 + x_9 * 2 + x_{10} * 1) \bmod 11 == 0$$

Jika program memenuhi kriteria diatas maka ISBN-10 dianggap valid.

Contoh ISBN-10 dengan kode : 3-598-21508-8. Dimasukkan pada kriteria diatas menjadi :

$$(3 * 10 + 5 * 9 + 9 * 8 + 8 * 7 + 2 * 6 + 1 * 5 + 5 * 4 + 0 * 3 + 8 * 2 + 8 * 1) \bmod 11 == 0$$

Jika hasilnya 0, maka dianggap valid !

4. Buatlah program untuk mengevaluasi aritmatika sederhana melalui kalimat. Ketika pengguna memasukkan 'selesai' maka program berhenti. Contoh I/O :

Berapa 5 ditambah 10 ?

-> 15

Berapa 10 dibagi 2 ?

-> 5

Berapa 5 dikurangi dengan -5

-> 10

Berapa 2 dikali 10 ?

-> 20

Selesai

--program berhenti--

## Minggu 8

Ujian Tengah Semester

---

## Minggu 9

### Fungsi dan Rekursif

---

#### Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu menggunakan prinsip fungsi dan rekursif pada konteks pemrograman dengan Python.

#### Pemanggilan Fungsi

Dalam konteks pemrograman, fungsi merupakan suatu urutan argumen yang memiliki nama dalam melakukan komputasi. Fungsi didefinisikan dengan nama fungsi yang diikuti dengan urutan argumen yang nantinya fungsi dapat dipanggil melalui namanya. Anda dapat membayangkan bahwa program yang dibuat dapat dibungkus dalam suatu tempat yang memiliki nama tersendiri. Sehingga, program yang anda buat dapat dipanggil cukup melalui nama itu saja. Sebagai contoh python telah menyediakan fungsi-fungsi yang menjalankan pernyataan/program tertentu sehingga kita tidak perlu melakukan *hard code* dari awal. Contoh pemanggilan fungsi yang tersedia di Python:

```
type(2)
-> <class 'int'>
```

Nama dari fungsi diatas adalah type. Fungsi tersebut diikuti dengan argumen yang berada pada tanda kurung. Hasil dari pemanggilan fungsi tersebut merupakan tipe dari argumen.

#### Membuat Fungsi

Selain kita dapat menggunakan fungsi-fungsi yang tersedia pada Python seperti fungsi yang ada pada math dan random, kita dapat membuat fungsi kita sendiri. Setelah kita mendefinisikan suatu fungsi, kita dapat memanggil berulang kali fungsi tersebut sesuai kebutuhan. Contoh pembuatan fungsi baru :

```
def cetak_lirik():
    print("Indonesia Raya..")
    print("Merdeka Merdeka..")
    print("Tanahku, Negeriku yang kucinta!")
```

Hasil dari pencetakan suatu fungsi :

```
print(cetak_lirik)
-> <function cetak_lirik at 0x000001E5FAD81D08>

print(type(cetak_lirik))
-> <class 'function'>
```

Pada saat nama fungsi dicetak, hasil yang didapatkan adalah suatu fungsi yang disimpan pada memori tertentu. Sedangkan pada saat jenis “cetak\_lirik” dicetak, menghasilkan tipe berupa fungsi. Untuk memanggil fungsi yang telah dibuat, cukup dengan memanggil nama beserta argumennya :

```
cetak_lirik()

Indonesia Raya..
Merdeka Merdeka..
Tanahku, Negeriku yang kucinta!
```

Setelah fungsi dibuat, kita dapat memanggil fungsi tersebut di dalam fungsi yang lain.

```
def ulang_lirik():
    cetak_lirik()
    cetak_lirik()

Indonesia Raya..
Merdeka Merdeka..
Tanahku, Negeriku yang kucinta!
Indonesia Raya..
Merdeka Merdeka..
Tanahku, Negeriku yang kucinta!
```

## Default Argumen

Kita dapat memberikan argumen secara default jika argumen tersebut tidak tersedia saat pemanggilan fungsi. Argumen default dinyatakan dengan tanda “=” setelah parameter. Contoh :

```
def print_twice(bruce = "Tidak ada"):
    print(bruce)
    print(bruce)

print_twice()

-> Tidak ada
    Tidak ada
```

Tanpa diberikan suatu argumen, secara default parameter bruce akan bernilai “Tidak ada”.

### Praktikum 1: Contoh Fungsi

```
1 def asal(kota = "Balikpapan"):
2     print(f"Saya berasal dari {kota}")
3
4 def cetak_list(list = ["kosong"]):
```

```

5     for x in list:
6         print(x)
7
8     def cetak_segi_tiga(n = 5):
9         for i in range(n):
10            for j in range(i+1):
11                print("*", end = "")
12            print()
13
14 asal("Banjarmasin")
15 asal("Samarinda")
16 asal("Palangkaraya")
17 asal("Pontianak")
18 asal()
19
20 cetak_list([1,2,3,4,5])
21 cetak_list()
22
23 cetak_segi_tiga(3)
24 cetak_segi_tiga()

```

### Fungsi dengan Nilai Kembali

Beberapa fungsi yang tersedia memiliki nilai kembali atau menghasilkan suatu hasil berupa nilai. Contoh-contoh fungsi diatas dapat dikatakan suatu prosedur atau fungsi void yang tidak memiliki nilai kembali. Prosedur tersebut hanya mengeksekusi pernyataan-pernyataan dalam suatu fungsi. Contoh fungsi dengan nilai kembali :

```

x = math.cos(radians)
golden = (math.sqrt(5) + 1) / 2

```

`Math.cos` merupakan fungsi untuk menghasilkan nilai cos dari nilai radian. Fungsi tersebut menghasilkan suatu nilai yang disimpan pada variabel `x`. Pernyataan berikutnya merupakan ekspresi yang melibatkan fungsi dimana `Math.sqrt` menghasilkan nilai berupa hasil akar yang selanjutnya dioperasikan dengan operand lainnya.

#### Praktikum 2: Contoh Fungsi dengan Nilai Kembali

```

1  def tambah(a,b):
2      return a + b
3
4  def kali(a,b,c):
5      return a*b*c
6
7  def jumlah(list):
8      jumlah = 0
9      for x in list:
10         jumlah+=x
11     return jumlah

```

```

12
13 def maksimum(a,b,c):
14     if(a > b and a > c):
15         return a
16     elif(b > c):
17         return b
18     else:
19         return c
20
21 hasil1 = tambah(5,10)
22 print(f"hasil1 = {hasil1}")
23 hasil2 = kali(2,3,4)
24 print(f"hasil2 = {hasil2}")
25 hasil3 = jumlah([1,2,3,4,5])
26 print(f"hasil3 = {hasil3}")
27 hasil4 = maksimum(5,10,7)
28 print(f"hasil4 = {hasil4}")

```

### Latihan Belajar Mandiri

1. Cari taulah bagaimana anda dapat memasukkan argumen tak terhingga tanpa harus mendefinisikan parameter satu per satu pada fungsi! contoh : kali(2,3,4), kali(1,4,7,8), kali(7,9,10,2,56,99,100). (Manfaatkan internet!)

### Fungsi Rekursif

Seperti diketahui pada praktikum sebelumnya, suatu fungsi dapat memanggil fungsi lainnya berkali-kali. Jika fungsi yang dipanggil adalah nama fungsi itu sendiri (memanggil dirinya sendiri), maka fungsi tersebut merupakan fungsi rekursif. Pemanggilan tersebut akan menghasilkan perulangan yang mirip dengan perulangan while dan diperlukan kondisi khusus untuk membuat fungsi tersebut berhenti.

Sebagai contoh, terdapat suatu permasalahan sederhana untuk mencetak angka mundur dari n sampai dengan 1. Permasalahan ini dapat diselesaikan dengan mudah menggunakan perulangan while atau for. Tetapi permasalahan ini juga dapat diselesaikan secara rekursif. Contoh fungsi rekursif untuk mencetak angka mundur :

```

def cetak_mundur(n):
    if(n > 0):
        print(n)
        cetak_mundur(n-1)

```

Keluaran dari cetak\_mundur(5) adalah

```

5
4
3
2

```



## Praktikum 3: Faktorial dan Fibbonaci

```

1 def faktorial(n):
2     if n < 2:
3         return 1
4     else:
5         return n * faktorial(n - 1)
6
7
8 def fib(n):
9     if (n > 1):
10        return fib(n - 1) + fib(n - 2)
11    else:
12        return n
13
14
15 print(faktorial(10))
16 print(faktorial(5))
17 print(faktorial(2))
18
19 print(fib(11))
20 print(fib(5))
21 print(fib(1))

```

## Latihan Belajar Mandiri

1. Buatlah fungsi rekursif untuk mencetak suatu string sebanyak n !
2. Buatlah fungsi rekursif untuk mencetak jumlah bilangan sebanyak n! Contoh ketika n = 4 maka keluaran adalah 10 didapat dari 1 + 2 + 3 + 4.

## Latihan

1. Buatlah fungsi untuk mencari nilai faktorial dari suatu angka. Contoh dimasukkan 5, output : 5 x 4 x 3 x 2 x 1 = 120
2. Buatlah fungsi yang merepresentasikan fungsi matematika berikut :

$$y=6x^2+3x+2$$

Yang memberikan keluaran berupa nilai y dimana x adalah bilangan bulat antara -10 sampai 10.

3. Buatlah fungsi yang dapat mengekstrak suatu string kedalam tiga bagian yaitu angka, huruf, dan simbol. Contoh masukkan berupa "ab13@\*\*5df#" akan menghasilkan list dengan elemen ["abdf", 135, "@\*\*#"].

4. Buatlah fungsi untuk mendeteksi apakah suatu angka yang dimasukkan merupakan bilangan prima atau bukan !
5. Buatlah fungsi rekursif untuk mencetak jumlah bilangan genap dari rentang a sampai b! Misal  $a = 2$  dan  $b = 8$  maka jumlah bilangan genap adalah 30 (didapat dari  $2 + 4 + 6 + 8$ )
6. Buatlah fungsi rekursif untuk mencari huruf kapital pertama pada string (hint: `isupper` merupakan fungsi untuk mengecek apakah suatu string atau karakter berupa kapital atau bukan). Fungsi dideklarasikan seperti pada nama fungsi dibawah. `Str` merupakan string sedangkan `i` merupakan indeks dari string yang dimulai dengan 0. Contoh pemanggilan fungsi `kapital_pertama("instiTut", 0)` yang menghasilkan keluaran nilai berupa "T"

```
def kapital_pertama(str, i) :
```

```
    ... *
```

```
    ... *
```

```
    ... *
```

## Minggu 10

### Penanganan Kesalahan

---

#### Tujuan

Setelah mengikuti praktikum ini, praktikan diharapkan mampu menangani kesalahan yang muncul selama program berjalan yang memungkinkan dapat mengganggu pengguna.

#### Mengangkat Kesalahan

Pada program, Python akan langsung menghentikan aplikasi yang dijalankan ketika error atau kesalahan terjadi. Terdapat dua jenis kesalahan secara umum yaitu kesalahan sintaksis dan kesalahan eksepsi. Kesalahan pada sintaksis terjadi ketika kita menuliskan program yang tidak sesuai dengan aturan python sedangkan eksepsi terjadi pada saat kesalahan terjadi walaupun secara sintaksis tidak terdapat kesalahan apapun.

Kita dapat mengangkat kesalahan pada program dengan menggunakan sintaks `raise` dan `assert` agar pesan kesalahan lebih informatif bagi programmer.

#### Praktikum 1: Harga Diskon

```

1 def harga_diskon(harga_asal, diskon):
2     assert diskon <= 100, "Diskon harus kurang dari 100%"
3     assert diskon > 0, "Diskon harus bernilai positif"
4     harga_diskon = harga_asal - (harga_asal*diskon/100)
5     return harga_diskon
6
7 harga = input("Masukkan harga: ")
8 assert harga.isdigit(), "Harga yang dimasukkan harus berupa angka"
9 harga = int(harga)
10 diskon = input("Masukkan diskon (tanpa persen/%): ")
11 assert diskon.isdigit(), "Masukkan diskon hanya berupa angka"
12 diskon = float(diskon)
13
14 print("Harga diskon menjadi: ",
15       harga_diskon(harga, diskon), "rupiah")

```

#### Latihan Belajar Mandiri

1. Cari tau apa yang dilakukan pada fungsi `isdigit()` setelah program menerima masukan!
2. Buatlah pengujian tambahan bahwa harga diskon hanya berlaku untuk barang diatas 150.000 rupiah dengan maksimal diskon 20%, sedangkan diatas 250.000 maksimal diskon 25%, dan diatas 500.000 maksimal diskon 40%.

## Praktikum 2: Raise Error

```

1 def hapusProduk(akses, produk):
2     if akses != "admin":
3         raise AssertionError("Anda harus punya akses
admin untuk bisa menghapus barang")
4
5     if produk not in ["tas", "jam", "baju", "celana", "buku"]:
6         raise ValueError("Produk tidak ada")
7
8 hapusProduk("pegawai", "komputer")

```

## Latihan Belajar Mandiri

1. Ubahlah argumen pada fungsi hapusProduk agar tidak memunculkan kesalahan!
2. Cari taulah melalui internet, jenis error apa saja yang dapat diangkat oleh Python? Kemukakan temuan anda kepada asisten!

## Blok Try dan Except

Pada contoh sebelumnya, `assert` digunakan untuk menguji suatu kondisi dan `raise` untuk mengangkat kesalahan dengan informasi yang berguna untuk programmer. Pada blok `try` dan `except` kita dapat menangani suatu kesalahan yang diangkat oleh program. Sebagai contoh pada program penghitung luas lingkaran, pengguna secara tidak sengaja memasukkan string yang berisi karakter berupa huruf sehingga tidak dapat diproses lebih lanjut dan dapat menyebabkan program berhenti. Kesalahan tersebut dapat kita tangkap menggunakan blok `except` dan diantisipasi berdasarkan fungsi / pernyataan lain yang terdapat pada blok tersebut. Berikut aturan yang digunakan untuk menggunakan blok penanganan kesalahan :

```

try:
    Buat pernyataan anda disini;
    .....
except Exception1:
    Jika terdapat exception1, maka blok ini akan dieksekusi.
except Exception2:
    Jika terdapat exception2, maka blok ini akan dieksekusi.
    .....
except Exception3, Exception4, ...:
    Jika terdapat lebih dari satu exception, maka blok ini akan dieksekusi.
    .....
else:
    Jika tidak terdapat exception, maka blok ini akan dieksekusi.

Finally:
    Ada exception atau tidak, blok ini akan dijalankan.

```

## Praktikum 3: Program Luas Segitiga dengan Penanganan Kesalahan

```

1 print("::: Program Luas Segitiga :::")
2
3 pil = True
4 while (pil == True):
5     try:
6         alas = input("Alas: ")
7         alas = float(alas)
8         if alas < 0:
9             raise AssertionError()
10    except ValueError:
11        print("Nilai yang anda masukkan bukan berupa angka,
nilai 1 akan dimasukkan sebagai nilai alas")
12        alas = 1
13    except AssertionError:
14        print("Nilai alas tidak boleh negatif!,
nilai akan dimutlakkan!")
15        alas = abs(alas)
16    else:
17        print("Alas yang anda masukkan sudah
benar dan akan diproses")
18
19    try:
20        tinggi = input("Tinggi: ")
21        tinggi = float(tinggi)
22        if tinggi < 0:
23            raise AssertionError()
24    except ValueError:
25        print("Nilai yang anda masukkan bukan berupa angka,
nilai 1 akan dimasukkan sebagai nilai tinggi")
26        tinggi = 1
27    except AssertionError:
28        print("Nilai tinggi tidak boleh negatif!,
nilai akan dimutlakkan!")
29        tinggi = abs(alas)
30    else:
31        print("Tinggi yang anda masukkan sudah benar
dan akan diproses")
32
33    luas = alas*tinggi*0.5
34    print(f"Alas ({alas}), Tinggi ({tinggi}), Luas = {luas}")
35
36    pilihan = '?'
37    while(pilihan != 'y' and pilihan != 't'):
38        try:
39            pilihan = input("Lanjut (y) atau berhenti (t)? ").lower()
40            if(pilihan != 'y' and pilihan != 't'):
41                raise ValueError()
42        except ValueError:
43            print("Masukkan hanya karakter y atau t")
44        else:

```

```
45     if(pilihan == 'y'):
46         pil = True
47     else:
48         pil = False
49         print("bye..bye..")
```

## Latihan

1. Buatlah program anda pada praktikum minggu-minggu sebelumnya agar dapat menangani kesalahan dari masukkan pengguna!

# Minggu 11

## Menulis dan Membaca Berkas

---

### Tujuan

Setelah mengikuti praktikum ini, praktikan diharapkan mampu membuat, menulis, dan membaca *file* melalui program.

### Menulis dan Membaca Berkas

Pada Python, kita dapat menyimpan suatu teks atau data kedalam *persistent storage* (tempat penyimpanan yang tetap dapat dimuat walaupun tidak ada listrik/dimatikan, hardisk, ssd, dan lain-lain.). Untuk dapat menyimpan atau menulis berkas, fungsi utama yang digunakan adalah `open()` yang disertai dengan argumen berupa nama file dan jenis aksi yang akan dilakukan.

#### Praktikum 1 : Membaca dan Menulis File Teks

```

1 fwrite = open("file_baru.txt", "w")
2 fwrite.write("Hallo, Baris pertama file yang saya buat\n")
3 fwrite.write("Ini baris ke-2\n")
4 fwrite.write("Ini baris ke-3\n")
5 fwrite.write("Ini baris ke-4\n")
6 fwrite.write("Ini baris ke-5\n")
7 fwrite.write("Ini baris ke-6\n")
8 fwrite.close()
9
10 with open("file_baru2.txt", "w") as f:
11     f.write("Hi, ini file yang lain\n")
12     f.write("Tapi menggunakan cara yang berbeda\n")
13     f.write("Kira2 apa ya bedanya\n")
14
15 with open("file_baru.txt", "r") as f:
16     counter = 0
17     for l in f:
18         counter+=1
19         print(f"{counter}: {l}", end="")
20     print(f'Jumlah baris dari file "{f.name}" adalah {counter}')
21
22 # Isi Program
23 with open("minggu11/write_read.py", "r") as f:
24     counter = 1
25     for l in f:
26         print(f"{counter}: {l}", end="")
27     counter+=1

```

## Latihan Belajar Mandiri

1. Bukalah <https://docs.python.org/3/library/functions.html#open>, dan cari tau apa saja mode yang dapat digunakan pada fungsi `open()` ? Kemukakan temuan anda kepada asisten.
2. Carilah pada dokumentasi python, properti apa saja yang bisa kita gunakan selain “name” pada objek file yang telah kita buka.
3. Menurut anda, apa perbedaan baris 1 dengan baris 10 ?
4. Modifikasilah program diatas dengan menambahkan fungsi agar dapat menemukan kata yang paling banyak karakternya dari suatu file.

### Latihan

1. **Buatlah program untuk mengkonversi berkas teks menjadi teks yang tidak memiliki baris baru (tanpa *enter*)!**
2. **Buatlah program untuk menghasilkan teks dari suatu berkas teks lain yang berisi daftar semua kata pada teks disertai dengan posisi kata keberapa dari berkas tersebut!**



# Minggu 12

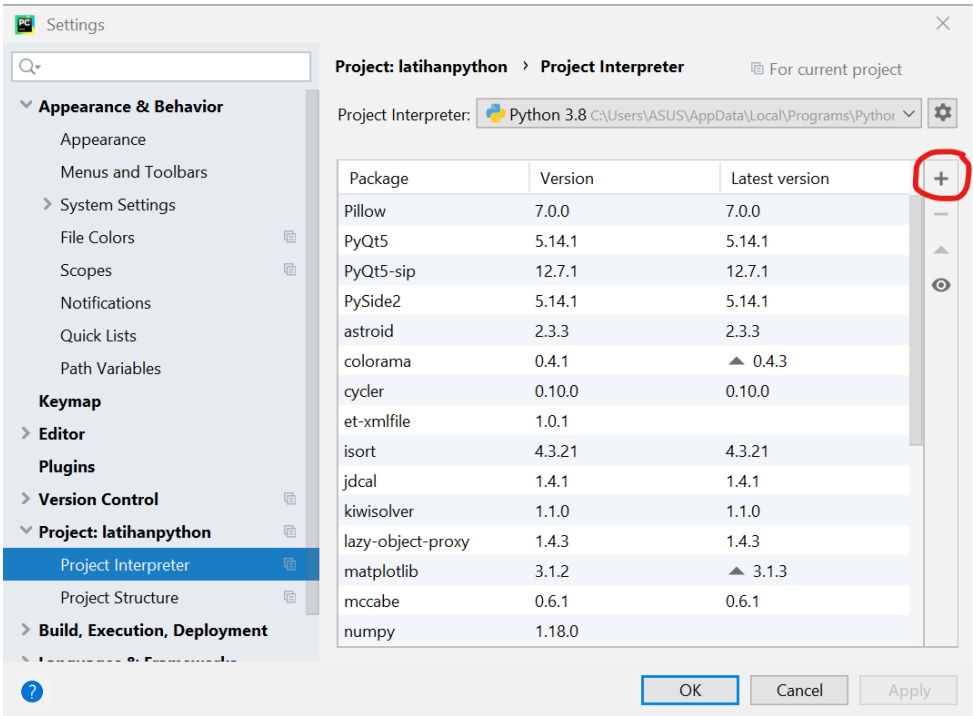
## PyQt5: Widget

### Tujuan

Setelah mengikuti praktikum ini, praktikan diharapkan mampu membangun antarmuka pengguna grafis yang disediakan oleh pustaka python yaitu PyQt5 melalui widget.

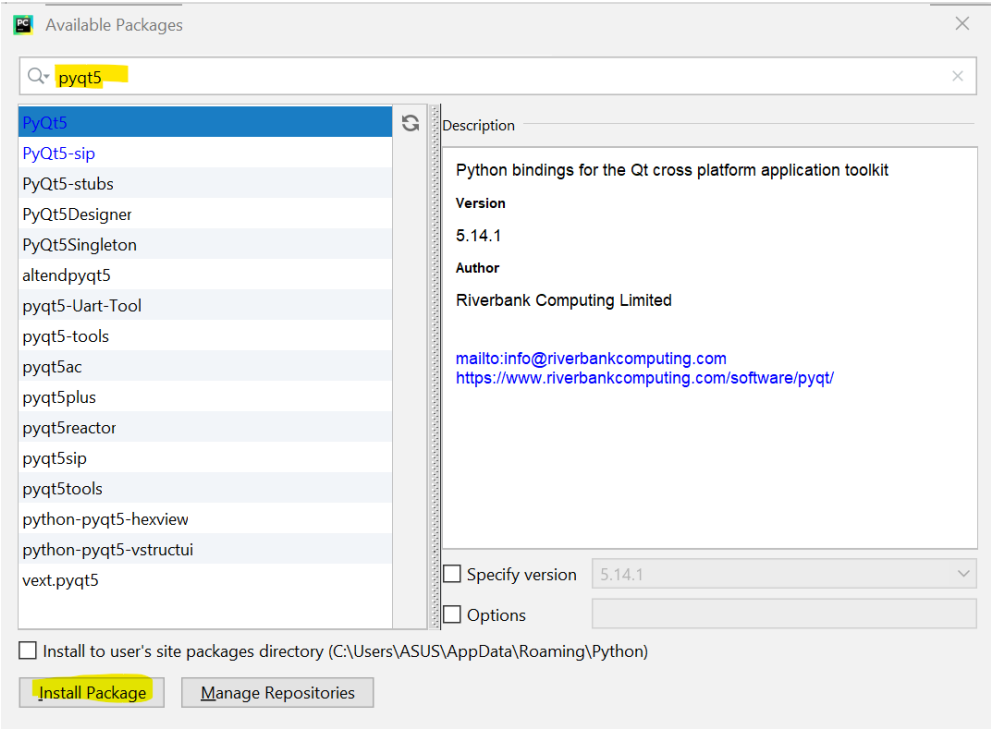
### Instalasi PyQt5

Sebelum kita dapat memprogram aplikasi desktop dengan PyQt5, kita perlu melakukan instalasi modul tersebut. Cara instalasi dapat menggunakan perintah PIP pada python, namun PyCharm menyediakan antarmuka yang sederhana untuk bisa menginstal modul yang diinginkan. Langkah pertama adalah masuk ke FILE > SETTINGS dan pastikan menu yang terpilih adalah PROJECT INTERPRETER (disebelah kiri). Selanjutnya, klik tombol dengan icon tambah (+) pada pojok kanan, seperti terlihat pada Gambar 12.1.



Gambar 12.25

Selanjutnya, PyCharm akan memunculkan jendela modul yang tersedia. Masukkan “pyqt5” pada field pencarian, lalu pilih PyQt5 pada daftar modul (sebelah kiri) dan klik install package. Lebih jelasnya dapat dilihat pada Gambar 12.2


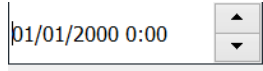


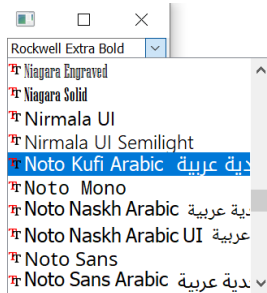

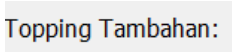


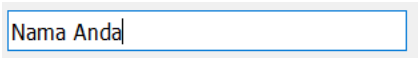

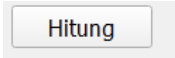
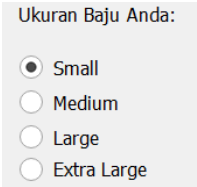
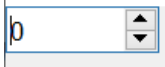

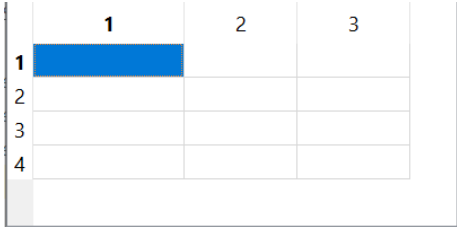
Gambar 12.26

Widget

Widget merupakan istilah yang digunakah oleh Qt sebagai objek utama antarmuka. Pengguna dapat berinteraksi langsung dengan antarmuka grafis melalui widget yang telah disediakan oleh PyQt. Antarmuka dapat terdiri dari beberapa widget yang disematkan pada jendela utama. Berikut daftar widget yang dapat digunakan oleh programmer :

Widget	Fungsi
QCheckbox	Checkbox, memungkinkan pengguna untuk memilih berbagai macam opsi <div><div>Topping Tambahan:</div><div><input checked="" type="checkbox"/> Bubuk Coklat (Rp. 5.000)</div><div><input checked="" type="checkbox"/> Serpihan Coklat (Rp. 8.000)</div><div><input type="checkbox"/> Biskuit (Rp. 8.000)</div></div>
QComboBox	Sebuah kotak yang berisi daftar nilai, membuat pengguna dapat memilih dari banyak pilihan yang tersedia

	Jenis Kopi: Espresso (Rp. 25.000) ▾
QDateEdit	Untuk dapat menyunting/memilih tanggal 
QDateTimeEdit	Untuk dapat menyunting/memilih tanggal disertai dengan waktu 
QDial	Penyesuaian nilai melalui rotasi 
QDoubleSpinBox	Sebuah antarmuka yang dapat diisi pengguna (berupa bilangan desimal) yang disertai dengan spin untuk dapat menurunkan atau menaikkan nilai 
QFontComboBox	Pemilihan jenis font 
QLCDNumber	LCD Display 
QLabel	Sebuah label yang memberikan informasi teks, tidak terdapat interaksi apapun yang dapat dilakukan oleh pengguna 
QLineEdit	Antarmuka untuk dapat memasukkan string

	
QProgressBar	<p>Progress bar yang menunjukkan progress/kemajuan suatu operasi</p> 
QPushButton	<p>Sebuah tombol</p> 
QRadioButton	<p>Sebuah himpunan pilihan dengan satu pilihan yang dapat dipilih.</p> 
QSpinBox	<p>Sebuah antarmuka yang dapat diisi pengguna (berupa bilangan bulat) yang disertai dengan spin untuk dapat menurunkan atau menaikkan nilai</p> 
QTimeEdit	<p>Antarmuka untuk menyunting/memasukkan waktu</p> 
QTableWidget	<p>Antarmuka untuk menampilkan table yang dapat disunting</p> 
<p>Dan masih banyak lagi, daftar widget dapat dilihat di <a href="https://doc.qt.io/qtforpython/PySide2/QtWidgets/">https://doc.qt.io/qtforpython/PySide2/QtWidgets/</a></p>	

## Praktikum 1: Hello World

```

1  from PyQt5.QtWidgets import *
2  from PyQt5.QtGui import *
3
4  app = QApplication([])
5
6  window = QMainWindow()
7  window.setGeometry(500, 300, 500, 150)
8  window.setWindowTitle("Aplikasi Pertama")
9
10 label_masuk = QLabel(window)
11 label_masuk.setText("Masukkan Nama : ")
12 label_masuk.setFixedWidth(150)
13 label_masuk.move(20,20)
14
15 #ledit = Line Edit
16 ledit_nama = QLineEdit(window)
17 ledit_nama.setText('Nama Anda')
18 ledit_nama.setToolTip('silakan masukan nama anada')
19 ledit_nama.move(150,20)
20 ledit_nama.setFixedWidth(300)
21
22 # ledit_nama.setEchoMode(QLineEdit.Password)
23 button_ok = QPushButton(window)
24 button_ok.setText("Ok")
25 button_ok.move(350, 70)
26
27 def button_act_par(nama):
28     print("Terclick", nama)
29
30 def button_act():
31     isi_field = ledit_nama.text()
32     message_box = QMessageBox(window)
33     message_box.setWindowTitle("Informasi")
34     message_box.setText(f>Nama anda adalah : {isi_field})
35     message_box.exec_()
36     # Atau bisa disingkat seperti ini
37     # QMessageBox.about(window, "Informasi",
38     f>Nama anda adalah : {isi_field}")
39
40 button_ok.clicked.connect(button_act)
41
42 window.show()
43 app.exec_()

```

## Praktikum 2: ComboBox

```

1  from PyQt5.QtWidgets import *
2  from PyQt5.QtGui import QFont
3  app = QApplication([])

```

```

4
5 window = QMainWindow()
6 window.setGeometry(300,300, 700, 500)
7 window.setWindowTitle("Aplikasi Radio Button")
8
9 label_ukuran = QLabel("Ukuran Baju Anda:", window)
10 label_ukuran.adjustSize()
11 label_ukuran.move(50,10)
12
13 radio_ukuran_S = QRadioButton("Small", window)
14 radio_ukuran_M = QRadioButton("Medium", window)
15 radio_ukuran_L = QRadioButton("Large", window)
16 radio_ukuran_XL = QRadioButton("Extra Large", window)
17 radio_ukuran_S.setChecked(True)
18 radio_ukuran_XL.adjustSize()
19 radio_ukuran_S.move(50, 50)
20 radio_ukuran_M.move(50, 80)
21 radio_ukuran_L.move(50, 110)
22 radio_ukuran_XL.move(50, 140)
23
24 radio_group_ukuran = QButtonGroup(window)
25 radio_group_ukuran.addButton(radio_ukuran_S)
26 radio_group_ukuran.addButton(radio_ukuran_M)
27 radio_group_ukuran.addButton(radio_ukuran_L)
28 radio_group_ukuran.addButton(radio_ukuran_XL)
29
30 def click_ukuran(btn):
31     label_hasil.setText(f"Ukuran : {btn.text()}
& Metode Pembayaran : {radio_group_metode.checkedButton().text()}")
32     label_hasil.adjustSize()
33
34 def click_metode(btn):
35     label_hasil.setText(f"Ukuran : {radio_group_ukuran
.checkedButton().text()} & Metode Pembayaran : {btn.text()}")
36     label_hasil.adjustSize()
37
38 radio_group_ukuran.buttonClicked.connect(click_ukuran)
39
40 label_metode = QLabel("Metode Pembayaran:", window)
41 label_metode.move(50, 200)
42 label_metode.adjustSize()
43
44 radio_metode_debcre = QRadioButton("Debit/Kartu Kredit", window)
45 radio_metode_transfer = QRadioButton("Transfer Bank", window)
46 radio_metode_cod = QRadioButton("Cash On Delivery", window)
47 radio_metode_debcre.setChecked(True)
48 radio_metode_debcre.adjustSize()
49 radio_metode_transfer.adjustSize()
50 radio_metode_cod.adjustSize()
51
52 radio_group_metode = QButtonGroup(window)
53 radio_group_metode.addButton(radio_metode_debcre)

```

```

54 radio_group_metode.addButton(radio_metode_transfer)
55 radio_group_metode.addButton(radio_metode_cod)
56
57 radio_metode_debcre.move(50, 230)
58 radio_metode_transfer.move(50, 260)
59 radio_metode_cod.move(50, 290)
60
61 radio_group_metode.buttonClicked.connect(click_metode)
62
63 font_ku = QFont("Verdana")
64 font_ku.setPointSize(8)
65 font_ku.setBold(True)
66 label_hasil = QLabel("", window)
67 label_hasil.move(50, 360)
68 label_hasil.setFont(font_ku)
69
70 window.show()
71 app.exec_()

```

### Praktikum 3: Kedai Kopi

```

1  from PyQt5.QtWidgets import *
2  from PyQt5.QtGui import *
3
4  app = QApplication([])
5  window = QMainWindow()
6  window.setWindowTitle("Pemesanan Kopi")
7  window.setGeometry(300, 300, 350, 400)
8
9  #Jenis Kopi
10 lbl_kopi = QLabel("Jenis Kopi:", window)
11 lbl_kopi.move(10, 10)
12 lbl_kopi.adjustSize()
13
14 cbx_kopi = QComboBox(window)
15 cbx_kopi.addItem('Espresso (Rp. 25.000)', 'Macchiato (Rp. 30.000)',
16                  'Cappuccino (Rp. 35.000)', 'Latte (Rp. 35.000)')
17 harga_kopi_indeks = [25000, 30000, 35000, 35000]
18 cbx_kopi.adjustSize()
19 cbx_kopi.move(100, 10)
20
21 #Topping tambahan
22 lbl_toppings = QLabel("Topping Tambahan:", window)
23 lbl_toppings.adjustSize()
24 lbl_toppings.move(10, 50)
25
26 chx_coklat = QCheckBox('Bubuk Coklat (Rp. 5.000)', window)
27 chx_coklat2 = QCheckBox('Serpihan Coklat (Rp. 8.000)', window)
28 chx_biskuit = QCheckBox('Biskuit (Rp. 8.000)', window)
29 chx_keju = QCheckBox('Serutan Keju (Rp. 7.000)', window)
30 chx_kayu_manis = QCheckBox('Kayu Manis (Rp. 4.500)', window)

```

```

30 chx_vanilla_cream = QCheckBox('Es Krim Vaninlla (Rp. 15500)', window)
31
32 chx_coklat.adjustSize()
33 chx_coklat2.adjustSize()
34 chx_biskuit.adjustSize()
35 chx_keju.adjustSize()
36 chx_kayu_manis.adjustSize()
37 chx_vanilla_cream.adjustSize()
38
39 chx_coklat.move(10, 80)
40 chx_coklat2.move(10, 110)
41 chx_biskuit.move(10, 140)
42 chx_keju.move(10, 170)
43 chx_kayu_manis.move(10, 200)
44 chx_vanilla_cream.move(10, 230)
45
46 #Fungsi menghitung total harga
47 def get_total_topping():
48     total = 0
49     if(chx_coklat.isChecked()):
50         total += 5000
51     if(chx_coklat2.isChecked()):
52         total += 8000
53     if(chx_biskuit.isChecked()):
54         total += 8000
55     if(chx_keju.isChecked()):
56         total += 7000
57     if(chx_kayu_manis.isChecked()):
58         total += 4500
59     if(chx_vanilla_cream.isChecked()):
60         total += 15500
61     return total
62
63 def hitung():
64     harga_kopi = harga_kopi_indeks[cbx_kopi.currentIndex()]
65     harga_topping = get_total_topping()
66     total_harga = harga_kopi+harga_topping
67     lbl_total.setText(f"Rp. {total_harga}")
68     lbl_total.adjustSize()
69
70 #Tombol Harga
71 btn_hitung = QPushButton("Hitung", window)
72 btn_hitung.move(70, 270)
73 btn_hitung.clicked.connect(hitung)
74
75 #Label Harga
76 lbl_harga = QLabel("Harga:", window)
77 lbl_harga.adjustSize()
78 lbl_harga.move(10, 310)
79
80 font_total = QFont()
81 font_total.setBold(True)

```



```
82 font_total.setPointSize(14)
83
84 lbl_total = QLabel(window)
85 lbl_total.setFont(font_total)
86 lbl_total.move(10, 350)
87
88 window.show()
89 app.exec_()
```

## Latihan

1. Buatlah tampilan sederhana untuk menambahkan data barang beserta harganya kedalam sebuah tabel.
2. Buatlah program penghitung investasi dalam jangka waktu tertentu dengan bunga sekian persen pertahunnya menggunakan antarmuka grafis.

# Minggu 13

## PyQt5: Layout

### Tujuan

Setelah mengikuti praktikum ini, praktikan diharapkan mampu menggunakan layout pada PyQt5.

### Layout

Pada praktikum sebelumnya, kita menggunakan posisi absolute (melalui posisi x dan y pada layar), secara spesifik, untuk menempatkan widget yang telah kita buat. Otomatis, penentuan panjang dan lebar pun perlu penyesuaian. Hal tersebut diperlukan ketelitian yang berkesinambungan, perlu mempertimbangkan posisi widget lain. Untuk dapat membuat posisi dan geometri menyesuaikan secara otomatis, kita dapat memanfaatkan layout. Ada beberapa jenis layout yang disediakan oleh PyQt, namun pada praktikum ini akan dikenalkan tiga jenis layout utama yaitu horizontal, vertikal, dan grid.

Dengan memanfaatkan layout horizontal melalui QHBoxLayout, kita dapat mengatur dengan rapi widget-widget yang kita masukkan kedalam layout tersebut secara horizontal, sebaliknya QVBoxLayout digunakan untuk mengatur kumpulan widget secara vertikal. Sedangkan Grid akan menata widget berdasarkan kolom dan baris. Lebih jelasnya, silakan ketik kode program praktikum dibawah.

#### Praktikum 1: Data Mahasiswa dengan Horizontal dan Vertikal Layout

```

1  from PyQt5.QtWidgets import *
2  from PyQt5.QtCore import *
3  from PyQt5.QtGui import QIcon
4
5  import sys
6  app = QApplication([])
7
8  window = QMainWindow()
9  window.setWindowTitle("Layout Horizontal dan Vertikal")
10 window.setGeometry(300, 300, 400, 200)
11
12 lbl_nama = QLabel("Nama", window)
13 lbl_nama.adjustSize()
14 lbl_nim = QLabel("NIM", window)
15 lbl_nim.adjustSize()
16 lbl_tgl_lahir = QLabel("Tgl. Lahir", window)
17 lbl_tgl_lahir.adjustSize()
18
19 ledit_nama = QLineEdit(window)
20 ledit_nim = QLineEdit(window)
21

```

```

22 calender_lahir = QCalendarWidget(window)
23
24 btn_simpan = QPushButton("Simpan", window)
25 btn_simpan.setIcon(QIcon("simpan.png"))
26 btn_simpan.setMinimumSize(QSize(200,50))
27 btn_simpan.setIconSize(QSize(40,40))
28
29 layout_hor1 = QHBoxLayout()
30 layout_hor1.addWidget(lbl_nama)
31 layout_hor1.addWidget(ledit_nama)
32 layout_hor1.setSpacing(75)
33
34 layout_hor2 = QHBoxLayout()
35 layout_hor2.addWidget(lbl_nim)
36 layout_hor2.addWidget(ledit_nim)
37 layout_hor2.setSpacing(87)
38
39 layout_hor3 = QHBoxLayout()
40 layout_hor3.addWidget(lbl_tgl_lahir)
41 layout_hor3.addWidget(calender_lahir)
42 layout_hor3.setSpacing(50)
43
44 layout_hor4 = QHBoxLayout()
45 layout_hor4.addWidget(btn_simpan, alignment=Qt.AlignRight)
46 # layout_hor4.set
47
48 layout_ver = QVBoxLayout()
49 layout_ver.addLayout(layout_hor1)
50 layout_ver.addLayout(layout_hor2)
51 layout_ver.addLayout(layout_hor3)
52 layout_ver.addLayout(layout_hor4)
53
54 widget = QWidget()
55 widget.setLayout(layout_ver)
56 window.setCentralWidget(widget)
57
58 #Message Box
59 def show_message():
60     # sd = selected date
61     sd = calender_lahir.selectedDate()
62     message_box = QMessageBox(window)
63     message_box.setWindowTitle("Berhasil")
64     message_box.setText(f>Nama: {ledit_nama.text()}\nNIM:
{ledit_nim.text()}\nTanggal Lahir: {sd.toString()} atau
{sd.day()}/{sd.month()}/{sd.year()}")
65     message_box.exec_()
66
67 btn_simpan.clicked.connect(show_message)
68
69 window.show()
70 sys.exit(app.exec_())

```

## Praktikum 2: Dummy Kalkulator dengan Grid Layout

```

1  from PyQt5.QtWidgets import *
2
3  app = QApplication([])
4
5  window = QMainWindow()
6  window.setGeometry(300,300, 500, 300)
7  window.setWindowTitle("Grid Layout")
8
9  grid = QGridLayout()
10
11  #Text
12  ledit_num = QLineEdit(window)
13  ledit_num.setFixedHeight(50)
14  ledit_num.setEnabled(False)
15
16  def buat_tombol(label):
17      qbtn = QPushButton(label, window)
18      qbtn.setFixedHeight(100)
19      return qbtn
20
21  def set0():
22      ledit_num.setText(f"{ledit_num.text()}0")
23  def set1():
24      ledit_num.setText(f"{ledit_num.text()}1")
25  def set2():
26      ledit_num.setText(f"{ledit_num.text()}2")
27  def set3():
28      ledit_num.setText(f"{ledit_num.text()}3")
29  def set4():
30      ledit_num.setText(f"{ledit_num.text()}4")
31  def set5():
32      ledit_num.setText(f"{ledit_num.text()}5")
33  def set6():
34      ledit_num.setText(f"{ledit_num.text()}6")
35  def set7():
36      ledit_num.setText(f"{ledit_num.text()}7")
37  def set8():
38      ledit_num.setText(f"{ledit_num.text()}8")
39  def set9():
40      ledit_num.setText(f"{ledit_num.text()}9")
41  def clear():
42      ledit_num.setText("")
43
44  deflist = [set0, set1, set2, set3, set4,
45             set5, set6, set7, set8, set9]
46
47  #init Button
48  btn: QPushButton = []

```

```

48 for i in range(10):
49     qbtn = buat_tombol(str(i))
50     qbtn.clicked.connect(deflist[i])
51     btn.append(qbtn)
52
53 btn_clear = buat_tombol("C")
54 btn_clear.clicked.connect(clear)
55
56 btn_kurang = buat_tombol("-")
57 btn_tambah = buat_tombol("+")
58 btn_tambah.setFixedHeight(200)
59
60 grid.addWidget(ledit_num, 0,0,1,4)
61 grid.addWidget(btn_clear, 1, 3)
62 grid.addWidget(btn_kurang, 2, 3)
63 grid.addWidget(btn_tambah, 3, 3, 2, 1)
64
65 grid.addWidget(btn[7], 1, 0)
66 grid.addWidget(btn[8], 1, 1)
67 grid.addWidget(btn[9], 1, 2)
68
69 grid.addWidget(btn[4], 2, 0)
70 grid.addWidget(btn[5], 2, 1)
71 grid.addWidget(btn[6], 2, 2)
72
73 grid.addWidget(btn[1], 3, 0)
74 grid.addWidget(btn[2], 3, 1)
75 grid.addWidget(btn[3], 3, 2)
76
77 grid.addWidget(btn[0], 4, 1)
78
79 widget = QWidget()
80 widget.setLayout(grid)
81 window.setCentralWidget(widget)
82
83 window.show()
84 app.exec_()

```

## Latihan

1. Rapikanlah latihan program pada praktikum sebelumnya menggunakan layout yang telah anda pelajari!
2. Buatlah kalkulator sederhana yang memiliki fitur aritmatika sederhana seperti penjumlahan, pengurangan, perkalian, pembagian, pangkat, dan modulus.

## Minggu 14

### Progress Tugas Besar

---

#### Tujuan

Tujuan dari minggu ke-14 adalah untuk memberikan arahan, bimbingan, dan konsultasi terkait permasalahan selama tugas besar dikerjakan. Selain itu asisten dapat memantau progress mahasiswa terkait.

#### Petunjuk Asisten dan Praktikan

Asisten melakukan penyalinan Lembar Progres Tugas Besar (LPTB) yang ada pada google sheet melalui link <https://docs.google.com/spreadsheets/d/1fijBMJz7hHuKcRwgtYa3VEh94MTkPL6IjzLv7uQA8T0/edit?usp=sharing> ke gdrive masing-masing. Tautan yang disalin tersebut diserahkan kepada pengampu untuk dapat dipantau lebih lanjut. Proses monitoring dan pengisian LPTB dilakukan secara daring.

Selanjutnya meminta praktikan untuk mengisi biodata diri sesuai kelompok masing-masing yang di bedakan melalui *sheet*.

Asisten atau Praktikan mengisi hari/tanggal, materi/pokok bahasan, serta keterangan yang diikuti dengan tanda tangan digital (disisipkan via Image atau drawing). Setelah selesai mengisi, asisten harus melakukan proteksi terhadap data yang telah diisi sehingga hanya asisten dan pengampu yang dapat melakukan perubahan.

Asisten harus memastikan bahwa konsultasi kelompok mahasiswa minimal terjadi sebanyak 4x untuk memastikan progress tugas besar berlangsung. Asisten diminta untuk aktif dalam memonitor perkembangan kelompok praktikan.

## Minggu 15

### Progress Tugas Besar

---

#### Tujuan

Tujuan dari minggu ke-15 adalah untuk memberikan arahan, bimbingan, dan konsultasi terkait permasalahan selama tugas besar dikerjakan. Selain itu asisten dapat memantau progress mahasiswa terkait.

#### Petunjuk Asisten dan Praktikan

Asisten atau Praktikan mengisi hari/tanggal, materi/pokok bahasan, serta keterangan yang diikuti dengan tanda tangan digital (disisipkan via Image atau drawing). Setelah selesai mengisi, asisten harus melakukan proteksi terhadap data yang telah diisi sehingga hanya asisten dan pengampu yang dapat melakukan perubahan.

Asisten harus memastikan bahwa konsultasi kelompok mahasiswa minimal terjadi sebanyak 4x untuk memastikan progress tugas besar berlangsung. Asisten diminta untuk aktif dalam memonitor perkembangan kelompok praktikan.

## Minggu 16

Ujian Akhir Semester – Persentasi Tugas Besar

---



# Lampiran

## LPTB

### Tampilan LPTB

Template LPTB Alpro

File Edit View Insert Format Data Tools Add-ons Help

100% \$ % .0 .00 123 Default (Arial) 10 B I A

Share

	A	B	C	D	E	F	
1		INSTITUT TEKNOLOGI KALIMANTAN					
		TAHAP PERSIAPAN BERSAMA					
		ALGORITME PEMROGRAMAN					
2		Lembar Progres Tugas Besar (LPTB)					Nama Anggota Kelompok
3	Mata Kuliah	: Algoritme Pemrograman		Semester	: Genap/Ganjil	NIM	NAMA
4	Kode MK	: -		Tahun Akademik	: 2019/2020		
5	Kredit	: 3 SKS		Dosen Pengampu	:		
6				Asisten	: Asisten 1		
7					: Asisten 2		
8							
9							
10							
11							
12	Hari/Tanggal	Materi/Pokok Bahasan	Tanda Tangan Asisten	Tanda Tangan Mahasiswa	Keterangan		
13							

Kel. 1 Kel. 2 Kel. 3 Kel. 4 Kel. 5 Kel. 6 Kel. 7 Kel. 8 Kel. 9