

BAB I

PENGENALAN PYTHON

I. TUJUAN

- Mahasiswa dapat mengenal Python
- Mahasiswa dapat menggunakan dasar-dasar operasi Python
- Mahasiswa mampu mengenali tipe data dan struktur data dasar Python
- Mahasiswa memahami struktur sintaks Python, termasuk penulisan komentar, indentasi, dan aturan dasar lainnya
- Mahasiswa dapat menggunakan fungsi-fungsi bawaan Python

II. PENDAHULUAN

Pengenalan Python

Python adalah bahasa pemrograman multifungsi yang dirilis pada tahun 1991 oleh Guido van Rossum (GvR). Beliau membuat Python sebagai bahasa pemrograman yang mudah dibaca dan dimengerti (*readable*) serta memiliki kemampuan penanganan kesalahan (*exception handling*). Python menggunakan interpreter, sehingga program dapat dijalankan langsung tanpa perlu proses kompilasi. Python dilengkapi dengan berbagai fungsi bawaan dan pustaka (*library*) yang mempermudah pemrograman. Data yang dikelola di Python bisa memiliki berbagai tipe, seperti angka (*integer, float*), teks (*string*), hingga struktur data yang lebih kompleks (*list, dictionary*). Selain itu, Python mempunyai fungsi-fungsi matematika yang sangat lengkap, misalkan *abs*, *pow*, *round*, dst. Beberapa contoh penerapan Python adalah:

- Analisis Data, Visualisasi Data, dan Komputasi
- Web Development (*server-side*)
- Pembelajaran Mesin (*Machine Learning*) dan Kecerdasan Buatan
- Otomatisasi dan Skrip Sistem

Komponen utama Python

1. *Interpreter*. Lingkungan interaktif yang memungkinkan pengguna untuk menulis dan menjalankan perintah Python secara langsung melalui terminal atau *command prompt*, sehingga memudahkan eksperimen dengan sintaks dan fungsi Python tanpa perlu membuat *file script*.
2. *Python Script*. File kode dengan ekstensi *.py* memungkinkan pengguna menulis program yang lebih kompleks dan terstruktur, mempermudah pengelolaan, kolaborasi, serta pengujian aplikasi tanpa perlu mengetik ulang kode.
3. *Python Library*. Python memiliki beberapa pustaka, baik bawaan seperti *math*, *os*, dan *random*, maupun eksternal seperti *numpy*, *pandas*, dan *matplotlib*, memperluas fungsionalitas Python untuk komputasi, analisis data, dan visualisasi, yang dapat diinstal melalui *PyPI* menggunakan *pip*.
4. *Integrated Development Environment (IDE)*. Python didukung oleh berbagai IDE seperti *Visual Studio Code*, *PyCharm*, *Jupyter Notebook*, dan *Google Colab*, yang mempermudah penulisan, *debugging*, dan pengelolaan kode sesuai kebutuhan pengguna, dari proyek sederhana hingga aplikasi besar.
5. *Struktur Data Python*. Python dilengkapi dengan berbagai struktur data bawaan seperti *list* untuk elemen yang dapat diubah, *tuple* untuk data yang bersifat tetap, *dictionary* untuk pasangan kunci-nilai yang efisien, dan *set* untuk koleksi elemen unik tanpa urutan, yang memudahkan pengelolaan data dalam berbagai jenis pemrograman.

Tipe Data Dasar dalam Python

Python mendukung berbagai tipe data dasar yang memudahkan pengguna dalam menyimpan dan memanipulasi data. Tipe data dasar yang sering digunakan pada Python diantaranya adalah:

- Integer (*int*): menyimpan bilangan bulat.
- Float (*float*): menyimpan bilangan desimal.
- String (*str*): menyimpan teks atau karakter.
- Boolean (*bool*): menyimpan nilai logika *True* atau *False*.
- List (*list*): menyimpan kumpulan data yang terurut.
- Dictionary (*dict*): menyimpan data dalam pasangan *key-value*.

Penggunaan Fungsi dan Variabel dalam Python

Seperti halnya kalkulator, Python menyediakan berbagai fungsi dasar yang mempermudah operasi matematika dan logika. Python memiliki ratusan fungsi bawaan (*built-in functions*) yang dirancang untuk menyelesaikan berbagai permasalahan pemrograman. Setiap fungsi di Python memiliki peran tertentu. Misalnya, fungsi matematika seperti *abs*, *pow*, *round*, dan *math.sqrt* sangat bermanfaat dalam melakukan perhitungan numerik. Selain itu, Python juga memiliki konstanta seperti *math.pi* dan *math.e* yang umum digunakan dalam kalkulasi ilmiah. Fungsi-fungsi dasar yang terdapat pada Python diantaranya adalah:

- *abs(x)* : mengembalikan nilai mutlak dari x.
- *round(x, n)* : membulatkan angka x ke n tempat desimal.
- *pow(x, y)* : menghitung x pangkat y.
- *divmod(x, y)* : hasil bagi dan sisa dari pembagian x oleh y.

Variabel khusus seperti:

- *math.pi* : ekuivalen dengan nilai 3.14.
- *math.e* : ekuivalen dengan nilai 2.71 (bilangan *euler*).
- *math.inf* : bilangan tak berhingga, *inf*.
- *math.nan* : *not a number*, 0/0.
- *i* dan *j* : bilangan *complex*.

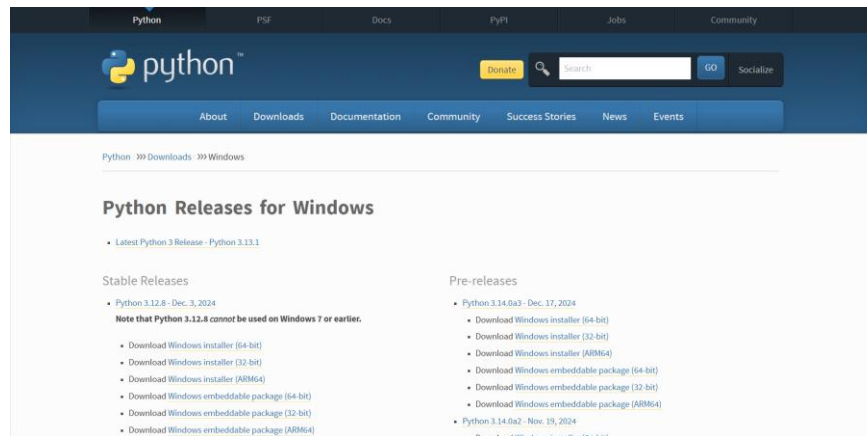
III. LANGKAH PRAKTIKUM

Instalasi dan Konfigurasi Python

Berikut langkah-langkah menginstal Python 3 pada sistem operasi Windows.

1. Kunjungi situs berikut: <https://www.python.org/downloads/windows/>.

Setelah itu akan muncul tampilan seperti gambar dibawah.



2. Pilih Python versi 3.13.1.
3. Pilih Windows installer (32-bit) atau Windows installer (64-bit). Pemilihan ini disesuaikan dengan spesifikasi sistem pada perangkat masing-masing.

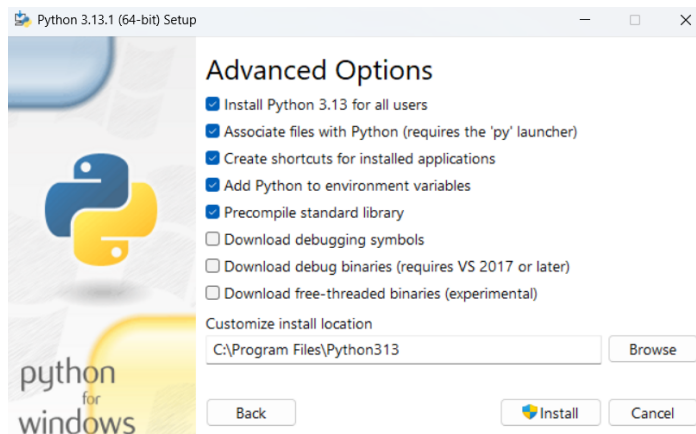
Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	SBOM
Gzipped source tarball	Source release		6820ac52d77af870f795dabc64583234	27.9 MB	SIG	.sigstore	SPDX
XZ compressed source tarball	Source release		80c16badb94ffe235280d4d9a099b8bc	21.5 MB	SIG	.sigstore	SPDX
macOS 64-bit universal2 installer	macOS	for macOS 10.13 and later	19e5a1d9e8264c88706ac9604c526e9b	68.2 MB	SIG	.sigstore	
Windows installer (64-bit)	Windows	Recommended	901760cfa29327ab08c6083dcdcc210	27.4 MB	SIG	.sigstore	SPDX
Windows installer (32-bit)	Windows		12455257e0eaf8c7a3b0af6522647638	26.1 MB	SIG	.sigstore	SPDX
Windows installer (ARM64)	Windows	Experimental	f35c98b7b0924865021e89160f208cdb	26.7 MB	SIG	.sigstore	SPDX
Windows embeddable package (64-bit)	Windows		d5c8030976b5eaf55ed6b321c073dda7	10.3 MB	SIG	.sigstore	SPDX
Windows embeddable package (32-bit)	Windows		87e2f278da8dd60e681ad1b5bb9e555	9.2 MB	SIG	.sigstore	SPDX
Windows embeddable package (ARM64)	Windows		7848dbb812b79e61146a592bb19075ee	9.6 MB	SIG	.sigstore	SPDX

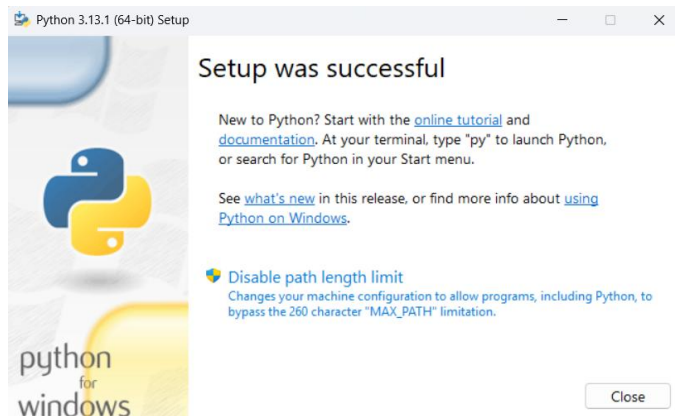
4. Setelah *installer* berhasil diunduh, lakukan instalasi dan pastikan mencentang “Add python.exe to PATH”. Lalu klik *Customize Installation*.



5. Pada “Optional Features”, pastikan semuanya sudah tercentang, lalu klik “Next”.
6. Pada “Advanced Options”, centang bagian “Install Python 3.13 for all users”, lalu klik install.



7. Setelah proses instalasi, sebaiknya klik *Disable path length limit* untuk mengatasi *problem path* yang melebihi 255 karakter (nama folder yang terlalu panjang).



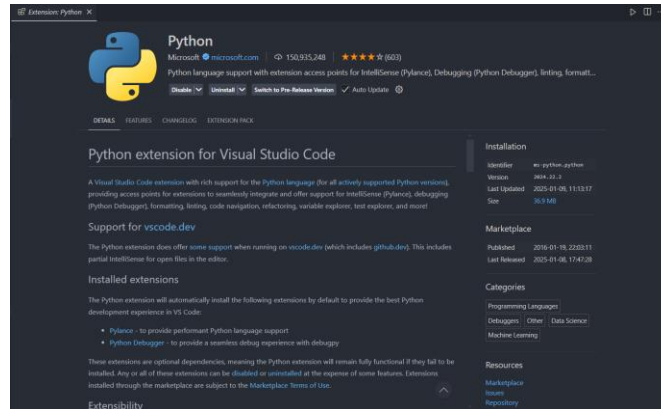
Sekarang pastikan Python sudah terinstal pada lokal masing-masing. Untuk Mac OS atau Ubuntu umumnya Python telah terinstal secara otomatis, namun untuk mengecek, ikuti langkah-langkah berikut.

1. Masuk ke terminal atau *command prompt*.
 - a. Untuk pengguna Windows, klik **windows button** lalu ketik “Command Prompt”.
 - b. Untuk pengguna Mac OS, tekan **command + spacebar**. Lalu ketik “Terminal”.
 - c. Untuk pengguna Ubuntu, menggunakan *keyboard shortcut* dengan menekan **Ctrl + Alt + T**.
2. Masukkan perintah “python --version” lalu tekan **Enter**.
3. Baik Windows, Mac OS, dan Ubuntu akan menampilkan versi Python yang telah terinstal.

```
C:\Users\Admin>python --version
Python 3.13.1
```

Sekarang Python telah terinstal, sekarang kita siapkan IDE yang diperlukan, yaitu Visual Studio Code. Berikut adalah tahapan-tahapan melakukan Instalasi Visual Studio Code.

1. Kunjungi situs berikut: <https://code.visualstudio.com/>.
2. Unduh Visual Studio Code berdasarkan sistem operasi masing-masing.
3. Instal Visual Studio Code.
4. Jika sudah selesai, pergi ke tab *extensions*, atau bisa menggunakan *keyboard shortcut* dengan menekan **Ctrl + Shift + X**.
5. Lalu cari ekstensi Python, dan install.



Menjalankan Kode Program Python

Berikut langkah-langkah untuk menjalankan kode Python pada Visual Studio Code.

1. Buat atau buka folder/direktori baru.
2. Buat file Python baru dengan memberi “.py” pada penamaan file.
3. Jalankan kode berikut.

```
print("Hello, World!")
```

Kode di atas akan menampilkan teks “Hello, World!” pada terminal Visual Studio Code.

Membuat Variabel

Dalam Python, variabel digunakan untuk menyimpan berbagai jenis nilai, termasuk angka maupun teks. Berikut adalah contoh cara mendeklarasikan dan menggunakan variabel dalam Python:

```
a = 100
b = 200
nama = "Pemrograman Python"
total = a + b
rata2 = total / 2
```

Untuk menampilkan hasil “total”, “rata2” dan “nama”, dapat dilakukan dengan memanggil variabel tersebut menggunakan fungsi “print()”.

```
print("Total:", total)
print("Rata-rata:", rata2)
print("Selamat Datang ke", nama)
```

Penamaan Variabel

Pemberian nama variabel mengikuti aturan sebagai berikut:

1. Harus dimulai dengan huruf atau garis bawah (_).

Contoh:

```
umur = 19
_umur = 20
```

2. Tidak boleh mengandung spasi, tanda baca, atau karakter khusus.

Contoh:

```
!umur = ...
tanggal lahir = ...
```

3. *Case-sensitive* (besar-kecil huruf berpengaruh).

Contoh:

```
Tempat = "Babarsari"
tempat = "Condong Catur"
# Kedua variable diatas dianggap berbeda
```

Tipe Data Python

Python secara otomatis menentukan tipe data variabel berdasarkan nilai yang diberikan. Tidak perlu mendeklarasikan tipe data seperti dalam beberapa bahasa pemrograman lainnya. Selain itu, Python juga mendukung berbagai tipe data seperti integer, float, string, list, dan dictionary. Beberapa variabel yang ada dalam Python yaitu sebagai berikut:

Contoh	Tipe Data
x = "Hello, World"	String (<i>str</i>)
x = 20	Integer (<i>int</i>)
x = 20.5	Float (<i>float</i>)
x = True	Boolean (<i>bool</i>)
x = [1, 2, 3, 4, 5]	List (<i>list</i>)
x = {"nama": "Budi", "umur": 25}	Dictionary (<i>dict</i>)

Operator Dasar

Operator Aritmatika

Operator	Arti
+	Penjumlahan

-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus
**	Eksponensial
//	Pembagian bulat

Operator Perbandingan

Operator	Arti
==	Sama dengan
!=	Tidak sama dengan
<	Kurang dari
>	Lebih dari
<=	Kurang dari sama dengan
>=	Lebih dari sama dengan

Operator Logika

Operator	Arti
and	Akan menghasilkan nilai <i>true</i> jika kedua statement memiliki nilai <i>true</i> dan <i>false</i> untuk lainnya
or	Akan menghasilkan nilai <i>true</i> jika salah satu statement memiliki nilai <i>true</i> .
not	Membalikkan nilai statement, jika <i>false</i> akan menjadi <i>true</i> .

Contoh:

```
print(2**5 + (9/3) - 5) # Output: 30.0
x = 5
y = 3
print(x != y) # Output: True
```

Indentasi Pada Python

Indentasi di Python adalah cara menandai blok kode, seperti yang digunakan dalam fungsi, perulangan, pernyataan bersyarat, dan struktur lainnya. Berbeda dengan banyak bahasa pemrograman lain yang menggunakan kurung kurawal {} untuk menentukan blok kode, Python menggunakan **spasi atau tab** untuk tujuan tersebut. Indentasi bukan hanya konvensi gaya dalam Python namun merupakan persyaratan sintaksis. Jika indentasi tidak konsisten atau tidak digunakan dengan benar, Python akan menampilkan error.

Contoh Benar:

```
if 5 > 2:
    print("Lima lebih besar dari dua!")
```

Contoh Salah:

```
if 5 > 2:
    print("Lima lebih besar dari dua!")
# IndentationError: expected an indented block
```

Fungsi Formatted String (F-String)

Fungsi F-string adalah cara untuk menyisipkan ekspresi atau variabel langsung ke dalam string dengan menggunakan kurung kurawal {}.

Contoh:

```
nama = "Putri"
umur = 20

print(f"Namaku {nama} dan umurku {umur} tahun.")
# Output: Namaku Putri and umurku 20 tahun.
```

Statement Pengendali

Struktur kendali dalam Python digunakan untuk mengontrol aliran eksekusi program berdasarkan kondisi tertentu. Python memiliki tiga jenis utama struktur kendali:

1. **if**, digunakan untuk menjalankan kode jika kondisi bernilai *True*.

Contoh:

```
x = 10
if x > 5:
    print("x lebih besar dari 5")
```

2. **if – else**, menambahkan blok alternatif jika kondisi **if** tidak terpenuhi.

Contoh:

```
x = 3
if x > 5:
    print("x lebih besar dari 5")
else:
    print("x tidak lebih besar dari 5")
```

3. **if - elif – else**, digunakan untuk memeriksa beberapa kondisi.

Contoh:

```
x = 7
```

```
if x > 10:
    print("x lebih besar dari 10")
elif x > 5:
    print("x lebih besar dari 5 tetapi tidak lebih dari 10")
else:
    print("x lebih kecil atau sama dengan 5")
```

Python juga memiliki ekspresi ternary, yang digunakan untuk menulis pernyataan if sederhana dalam satu baris.

Contoh:

```
umur = 20
status = "Dewasa" if age >= 18 else "Belum Dewasa"
print(status) # Output: Dewasa
```

Perulangan (*Looping*)

Python memiliki dua jenis perulangan utama, yaitu **for loop** dan **while loop**, yang digunakan untuk mengulang blok kode berdasarkan kondisi tertentu.

Berikut adalah penjelasan dan contoh dari masing-masing jenis perulangan:

1. **for loop**, digunakan untuk mengiterasi elemen dalam koleksi seperti list, string, atau menggunakan fungsi `range()`.

Contoh:

```
for i in range(5):
    print(i)
# Output : 0, 1, 2, 3, 4
```

2. **while loop**, digunakan untuk menjalankan blok kode selama kondisi bernilai *True*.

Contoh:

```
count = 0
while count < 5:
    print(count)
    count += 1
# Output : 0, 1, 2, 3, 4
```

Pendefinisian Fungsi

Functions dalam Python adalah blok kode yang dirancang untuk menjalankan tugas tertentu. Dengan fungsi, kita dapat mengelompokkan logika program menjadi bagian yang lebih kecil, terstruktur, dan dapat digunakan kembali.

Functions yang dapat didefinisikan oleh pengguna adalah sebagai berikut:

1. Fungsi Tanpa Parameter

```
def say_hello():  
    print("Hello, World!")  
say_hello() # Output : Hello, World!
```

2. Fungsi Dengan Parameter

```
def penjumlahan(a, b):  
    return a + b  
print(penjumlahan(3, 5)) # Output: 8
```

3. Fungsi Dengan Parameter Default, jika parameter tidak diberikan, nilai default akan digunakan.

```
def salam(nama="Tamu"):  
    return f"Hello, {nama}!"  
print(salam()) # Output: Hello, Tamu!  
print(salam("Putri")) # Output: Hello, Putri!
```

IV. TUGAS PRAKTIKUM

John adalah seorang siswa yang baru saja menyelesaikan ujian akhir semesternya. Ia mengikuti tiga mata pelajaran, yaitu

- Matematika
- Bahasa Inggris
- Ilmu Pengetahuan Alam

Buatkan program Python untuk memasukkan nilai dari ketiga mata pelajaran tersebut dan kemudian menghitung rata-rata nilai berdasarkan nilai yang telah dimasukkan. Formula rata-rata adalah:

$$\text{Rata - rata} = \frac{\text{Total Nilai}}{\text{Jumlah Mata Pelajaran}}$$

Berdasarkan rata-rata yang dihitung, tentukan kategori nilai John menggunakan kriteria berikut:

- Jika rata-rata 90-100, kategori nilai adalah **Sangat Baik**.
- Jika rata-rata 80-89, kategori nilai adalah **Baik**.
- Jika rata-rata 70-79, kategori nilai adalah **Cukup**.
- Jika rata-rata di bawah 70, kategori nilai adalah **Perlu Perbaikan**,