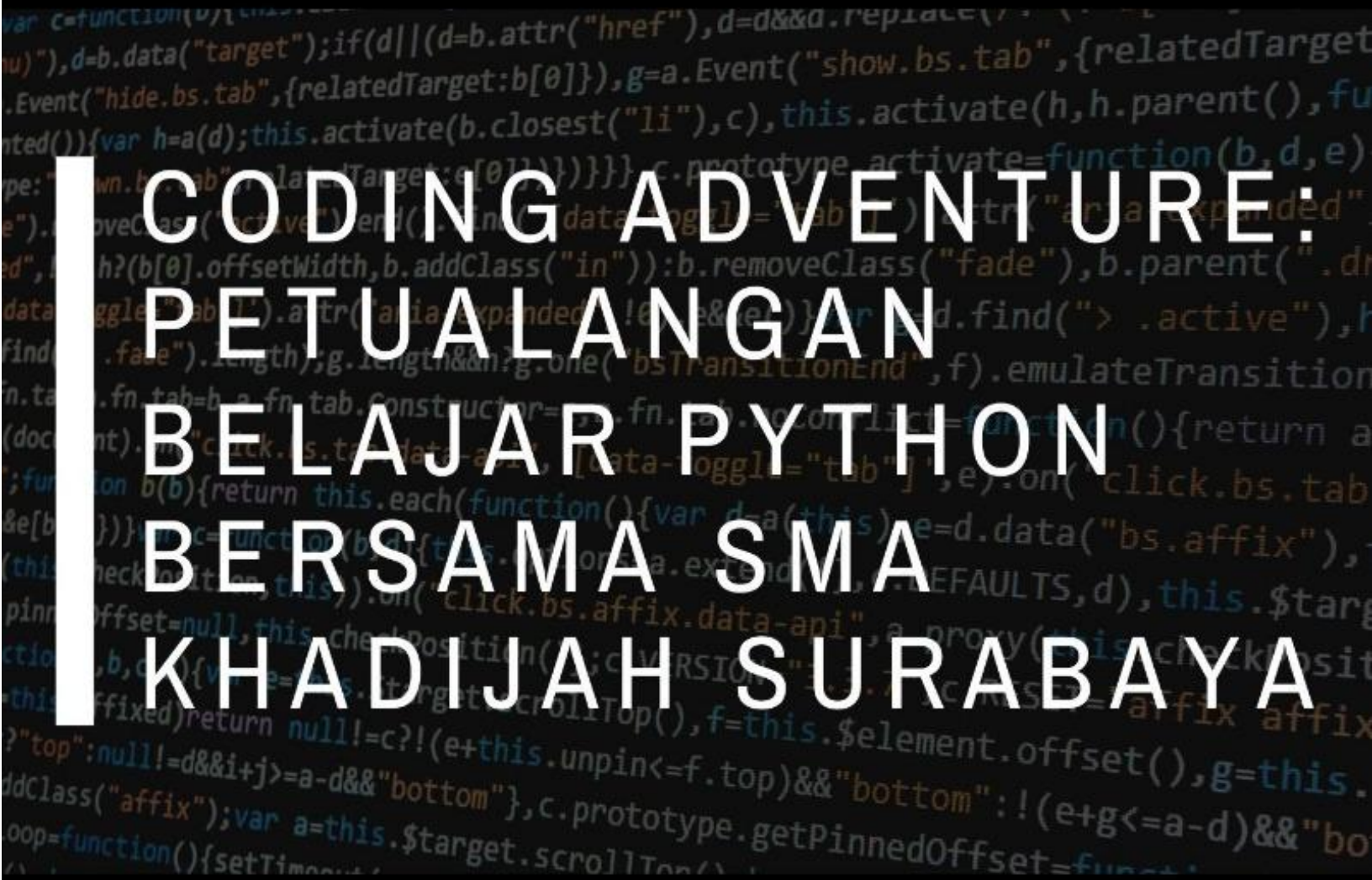


CODING ADVENTURE: PETUALANGAN BELAJAR PYTHON BERSAMA SMA KHADIJAH SURABAYA

The code is life. Never mess with the code.

Fakultas Informatika
School of Computing
Telkom University

Telkom University

DAFTAR ISI

Pendahuluan

Python adalah salah satu bahasa pemrograman paling populer di dunia dan digunakan secara luas dalam berbagai bidang seperti ilmu data, kecerdasan buatan, keamanan siber, pengembangan aplikasi web, otomasi sistem, hingga pembuatan game. Python pertama kali diperkenalkan oleh Guido van Rossum pada tahun 1991 dan sejak saat itu terus berkembang menjadi bahasa yang sederhana, fleksibel, dan memiliki komunitas terbesar di dunia pemrograman. Pada level pengenalan dasar, sangat penting untuk memahami apa itu Python, bagaimana cara kerjanya, serta mengapa bahasa ini menjadi pilihan utama bagi banyak pengembang, peneliti, maupun pemula yang baru memulai perjalanan dalam dunia pemrograman.

1. Apa Itu Python?

Python adalah bahasa pemrograman *interpreted*, *high-level*, dan *general-purpose*.

- *Interpreted*: Python dieksekusi baris per baris oleh *interpreter* tanpa perlu proses kompilasi terlebih dahulu. Ini memudahkan proses *debugging* dan pembuatan program.
- *High-level*: Sintaks Python dekat dengan bahasa manusia sehingga lebih mudah dipahami.
- *General-purpose*: Python tidak terbatas pada satu jenis aplikasi saja dapat digunakan untuk *web development*, *data science*, analisis data, *machine learning*, *scripting*, dan lain-lain.

Selain itu, Python mendukung berbagai paradigma pemrograman, seperti:

- Pemrograman prosedural
- Pemrograman berorientasi objek (OOP)
- Pemrograman fungsional

1.1 Sejarah Singkat Python

Python dikembangkan oleh Guido van Rossum pada akhir tahun 1980-an di Centrum Wiskunde & Informatica (CWI), Belanda. Tujuan utamanya adalah menciptakan bahasa yang mudah dibaca, minimalis, namun tetap kuat digunakan untuk berbagai kebutuhan. Nama "Python" sendiri terinspirasi dari acara komedi Inggris *Monty Python's Flying Circus*. Seiring perkembangan zaman, Python mengalami berbagai pembaruan menjadi standar utama dan terus mengalami update secara berkala.

1.2 Mengapa Python Populer dan Banyak Dipelajari?

Ada banyak alasan mengapa Python sangat digemari oleh pemula maupun profesional. Beberapa alasan utamanya adalah:

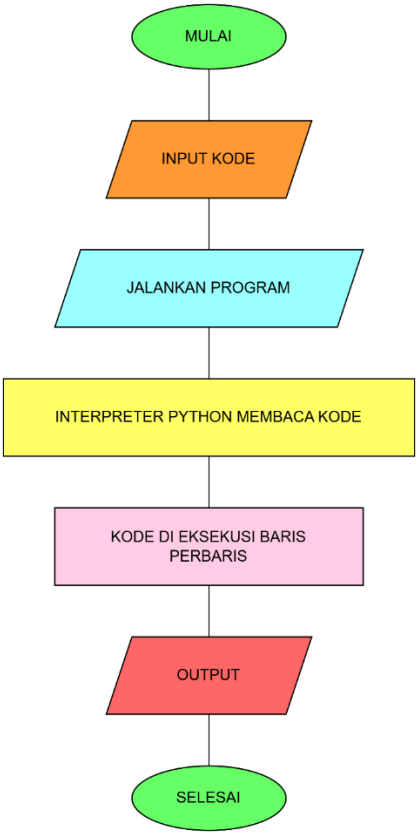
- **Sintaks yang Sederhana dan Mudah Dipahami**
  Python dirancang agar mudah dibaca. Bahkan pemula dapat memahami struktur dasar program hanya dalam beberapa menit. Contoh: print ("Hello, World!")
- **Memiliki Banyak Library dan Framework**

Python kaya akan modul dan library yang dapat digunakan untuk berbagai kebutuhan, seperti:
  - NumPy, Pandas digunakan dalam proses analisis data
  - Matplotlib, Seaborn digunakan dalam proses visualisasi data
  - TensorFlow, PyTorch digunakan dalam proses *machine learning* dan *AI*
  - Flask, Django digunakan dalam proses pengembangan web
  - OpenCV digunakan dalam proses pengolahan citra
- **Komunitas yang Sangat Besar**
  Dengan komunitas global, Python memiliki dokumentasi lengkap, tutorial di mana-mana, dan forum diskusi aktif. Ini mempermudah pemula belajar dari mana saja.
- **Fleksibel dan Bisa Digunakan untuk Banyak Bidang**
  Python cocok digunakan untuk:
  - Aplikasi web
  - Sistem otomasi (*automation*)
  - *Data science & AI*
  - Pembuatan aplikasi *desktop*
  - *IoT*
  - *Game development*

1.3 Cara Kerja Python

Saat menjalankan program Python, interpreter atau program akan membaca kode dari atas ke bawah dan mengeksekusinya baris demi baris. Prosesnya:

1. Python membaca file atau input program.
2. Python menerjemahkan kode ke dalam *bytecode*.
3. Bytecode dijalankan oleh *Python Virtual Machine (PVM)*.



Gambar 1. Alur Kerja Python

2. Apa itu variabel?

Variabel adalah objek dengan nama tertentu yang berfungsi untuk menyimpan suatu nilai. Nilai pada objek tersebut dapat diakses kembali dengan cara menulis atau memanggil nama variabel tersebut.

Ciri-ciri variabel di Python:

- Tidak perlu deklarasi tipe (*Python dynamic typing*).
- Tipe data variabel dapat berubah sesuai nilai yang diberikan.
- Nama variabel bersifat *case-sensitive* (contoh: `Nama` dan `nama` dianggap berbeda.).

Dalam pembuatan nama variabel, terdapat beberapa aturan penamaan yang menjadi standar pada bahasa python, yaitu:

- Berisi huruf `(a-z, A-Z)`, angka `(0–9)`, dan garis bawah `_`.
- Tidak boleh diawali angka.
- Tidak boleh mengandung spasi → gunakan `_`
  Contoh: `nama_pengguna`, `umur_siswa`.
- Harus bukan kata kunci Python, misalnya *while*, *for*, *class*, dll.

- Ikuti standar penamaan PEP 8 → gunakan nama variabel yang jelas dan bermakna.

Contoh-contoh penulisan nama variabel yang benar serta contoh penulisan yang salah dalam bahasa python dapat dilihat pada kode berikut:

```
# contoh penamaan yang VALID

umur = 25
nama_pengguna = "Ash Ketchum"
total_nilai = 90.5
info = "Pokemon"


# contoh penamaan yang TIDAK VALID

4umur = 25                    # dimulai dengan angka
nama pengguna = "Ash Ketchum"   # Mengandung spasi
while = "Pikachu"             # Menggunakan kata kunci Python
```

2.1 Apa itu Tipe Data?

Tipe data adalah kategori atau jenis data yang menentukan bentuk nilai yang bisa disimpan dalam sebuah variabel dan operasi apa saja yang bisa dilakukan pada data tersebut. Setiap variabel pada python memiliki tipe datanya masing-masing. Variabel pada python menerapkan tipe dinamis (d*inamic typing*), artinya tidak perlu mendeklarasikan tipe data variabel secara eksplisit. Python secara otomatis menentukan tipe data variabel berdasarkan nilai yang diberikan.

Kenapa tipe data penting?

1. Menentukan bagaimana komputer mengolah data tersebut (misalnya angka bisa dijumlahkan, teks tidak bisa dijumlahkan seperti angka).
2. Membantu *programmer* memilih struktur data yang tepat.
3. Mencegah *error* dalam program.

Berikut merupakan tipe data yang sering digunakan:

A. Tipe Data Besar

Tipe Data	Nama	Penjelasan	Contoh
int	Integer	Angka bulat, tanpa desimal	10, 2025, -7
float	Float	Angka pecahan/desimal	3.14, 10.5, -2.7
str	String	Teks atau karakter	"Hera", 'Hello'
bool	Boolean	Nilai logika benar/salah	True, False
list	List	Kumpulan data yang dapat diubah (mutable)	[1, 2, 3], ["a", "b"]
tuple	Tuple	Kumpulan data yang tidak dapat diubah (immutable)	(1, 2, 3), ("a", "b")

set	Set	Kumpulan data unik, tanpa urutan	{1, 2, 3}, {"a", "b"}
dict	Dictionary	Pasangan key–value	{"nama": "Hera", "umur": 20}

1. Integer (int)

   Angka bulat tanpa desimal.

```
umur = 20
tahun = 2025
```

2. Float

   Angka dengan desimal.

```
tinggi = 160.5
nilai = 3.14
```

3. String (str)

   Teks dalam tanda kutip.

```
nama = "Hera"
pesan = 'Hello Python'
```

4. Boolean (bool)

   Nilai logika: ***True*** atau ***False***

```
is_active = True
lulus = False
```

5. List

   Menyimpan banyak nilai, bisa berbeda tipe, dapat diubah (*mutable*).

```
buah = ["apel", "mangga", "pisang"]
angka = [1, 2, 3, 4]
campuran = [1, "hello", True]
```

6. Tuple

   Menyimpan banyak nilai tapi tidak dapat diubah (*immutable*).

```
koordinat = (10, 20)
bulan = ("Jan", "Feb", "Mar")
```

7. Set

   Koleksi data *unique* (tidak ada duplikasi) dan tidak berurutan.

```
angka_unik = {1, 2, 3, 3, 4}   # hasilnya {1, 2, 3, 4}
```

8. Dictionary(dict)

   Pasangan *key–value*.

```
mahasiswa = {
        "nama": "Hera",
```

```
    "nim": "123456",
    "jurusan": "Data Science"
}
```

B. Melihat Type Data

Gunakan fungsi type()

```
print(type("Hello"))    # str
print(type(12))         # int
print(type(3.14))       # float
print(type([1,2,3]))    # list
print(type({"a": 1}))   # dict
```

C. Konversi Tipe Data (Type Casting)

1. String ke Integer

```
x = int(input("Masukkan angka: "))
print(x + 10)
```

2. Float ke Int

```
angka_float = 12.78
angka_int = int(angka_float)
print("Angka float:", angka_float)
print("Setelah dikonversi ke int:", angka_int)
```

D. Operator

1. Operator Aritmatika

Digunakan untuk melakukan operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya.

```
x = 5
y = 3

print(x + y)   # Penjumlahan
print(x - y)   # Pengurangan
print(x * y)   # Perkalian
print(x / y)   # Pembagian
print(x % y)   # Sisa bagi
print(x ** y)  # Pangkat
```

2. Operator Perbandingan

Digunakan untuk membandingkan dua nilai. Hasilnya akan berupa True atau False.

```
print(x == y)  # Sama dengan
print(x != y)  # Tidak sama dengan
print(x > y)   # Lebih besar dari
print(x <= y)  # Kurang dari atau sama
```

3. Operator Logika

Digunakan untuk menggabungkan ekspresi logika yang hasilnya berupa True atau False. Sangat berguna dalam pengambilan keputusan yang melibatkan lebih dari satu kondisi.

```
a = True
b = False
print(a and b)  # False
print(a or b)   # True
print(not a)    # False
```

4. Operator Penugasan

Digunakan untuk memberi nilai ke variabel, sekaligus dapat digabungkan dengan operasi lain.

```
x = 5
x += 4   # Sama dengan x = x + 4
x -= 3   # Sama dengan x = x - 3
x *= 2   # Sama dengan x = x * 2
x /= 1   # Sama dengan x = x / 1
```

Contoh Program Lengkap

```
# 1. Membuat variabel berisi data diri
nama = "Rara"          # Tipe data string (teks)
umur = 20              # Tipe data integer (angka bulat)
tinggi = 155.0         # Tipe data float (angka desimal)
lulus = True           # Tipe data boolean (benar/salah)

# 2. Menggunakan operator untuk menghitung sesuatu
tahun_sekarang = 2025
tahun_lahir = tahun_sekarang - umur   # Menghitung tahun lahir

# 3. Menggabungkan semua data ke dalam satu list
data = [nama, umur, tinggi, lulus, tahun_lahir]

# 4. Menampilkan data ke layar
print("=== DATA DIRI ===")
print("Nama          :", nama)
print("Umur          :", umur)
print("Tinggi Badan  :", tinggi, "cm")
print("Status Lulus  :", lulus)
print("Tahun Lahir   :", tahun_lahir)

# 5. Menampilkan tipe data dari setiap variabel
print("\n=== TIPE DATA ===")
print("Tipe 'nama'       :", type(nama))
print("Tipe 'umur'       :", type(umur))
print("Tipe 'tinggi'     :", type(tinggi))
print("Tipe 'lulus'      :", type(lulus))
```

```
print("Tipe 'data'        :", type(data))
```

Output :

```
=== DATA DIRI ===
Nama          : Rara
Umur          : 20
Tinggi Badan   : 155.0 cm
Status Lulus   : True
Tahun Lahir    : 2005

=== TIPE DATA ===
Tipe 'nama'        : <class 'str'>
Tipe 'umur'        : <class 'int'>
Tipe 'tinggi'      : <class 'float'>
Tipe 'lulus'       : <class 'bool'>
Tipe 'data'        : <class 'list'>
```

3. Operator Aritmatika

Operator aritmatika adalah simbol atau karakter khusus yang digunakan untuk melakukan operasi matematis seperti penjumlahan, pengurangan, perkalian, pembagian, dan lain-lain. Python mendukung berbagai operator aritmatika yang dapat bekerja dengan tipe data numerik: ***integer***, ***float***, dan ***complex number***.

Operator aritmatika sangat mudah digunakan dan bekerja secara langsung pada dua operand, contoh:

```
a = 10
b = 3
hasil = a + b
```

tanda + adalah operator aritmatika.

3.1. Jenis-Jenis Operator Aritmatika

Python memiliki 7 operator aritmatika utama:

Operator	Nama operator	Contoh	Hasil
+	Penjumlahan	7 + 3	10
-	Pengurangan	7 – 3	4
*	Perkalian	7 * 3	21
/	Pembagian	7 / 3	2.3333
//	Pembagian Bulat	7 // 3	2
%	Modulus (sisa bagi)	7 % 3	1
**	Perpangkatan	7 ** 2	49

3.2. Penjelasan Lengkap Setiap Operator

- Operator Penjumlahan (+)

Digunakan untuk menjumlahkan dua nilai numerik.

```
x = 8
y = 5
print(x + y) # 13
```

- Operator Pengurangan (-)

Digunakan untuk mengurangi nilai dari operand kiri dengan operand kanan.

```
print (10 - 4) # 6
```

- Operator Perkalian (*)

Selain untuk perkalian angka, operator ini dapat mengulang teks.

```
print (6 * 3) # 18
print ("Hi" * 3) # HiHiHi
```

- Operator Pembagian (/)

Hasil selalu berupa float.

```
print (10 / 4) # 2.5
```

- Operator Pembagian Bulat (//)

Menghasilkan angka bulat terdekat ke bawah.

```
print (10 // 4) # 2
```

- Operator Modulus (%)

Menghasilkan sisa pembagian.

Modulus sering digunakan untuk:

1. menentukan bilangan ganjil/genap
2. pola perulangan
3. logika pembagian kelompok data

```
print (10 % 4) # 2
```

- Operator Perpangkatan (**)

Digunakan untuk menghitung pangkat.

```
print (2 ** 5) # 32
```

3.3. Contoh Program Lengkap Operasi Aritmatika

```
a = int(input("Masukkan angka pertama: "))
b = int(input("Masukkan angka kedua: "))

print("Hasil Penjumlahan:", a + b)
print("Hasil Pengurangan:", a - b)
print("Hasil Perkalian:", a * b)
print("Hasil Pembagian:", a / b)
print("Hasil Pembagian Bulat:", a // b)
print("Sisa Bagi:", a % b)
print("Pangkat:", a ** b)
```

Studi Kasus Mini: Menghitung Rata-Rata Nilai

Masalah: Luffy memperoleh nilai ujian Matematika, IPA, Bahasa Indonesia dan Bahasa Inggris sebesar: 85, 90, 78, 92. Luffy ingin mengetahui rata-rata nilai ujian yang di peroleh kali ini, hitunglah rata-rata hasil ujian Luffy dengan persamaan aritmatika yang telah di berikan

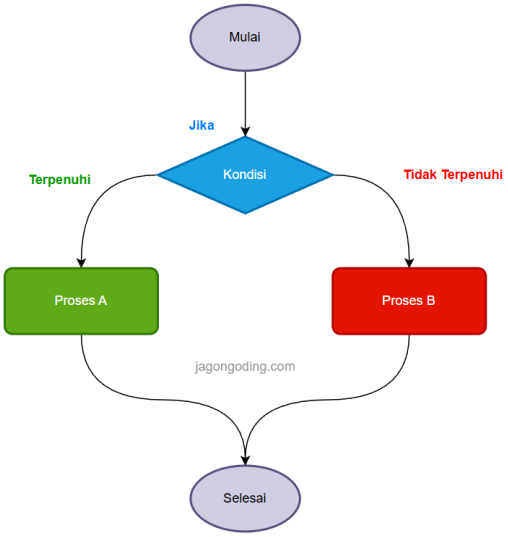
```
nilai = [85, 90, 78, 92, 88]
rata_rata = sum(nilai) / len(nilai)

print("Rata-rata nilai:", rata_rata)
```

4. Percabangan dan Perulangan

4.1 Percabangan

Percabangan adalah proses penentuan keputusan atau dalam bahasa inggris ini biasa disebut sebagai *conditional statement*. Percabangan berfungsi untuk mengontrol aliran sehingga dapat mengeksekusi blok kode yang berbeda untuk setiap kondisi yang berbeda. Secara umum alur percabangan menggunakan if, elif dan else adalah sebagai berikut:

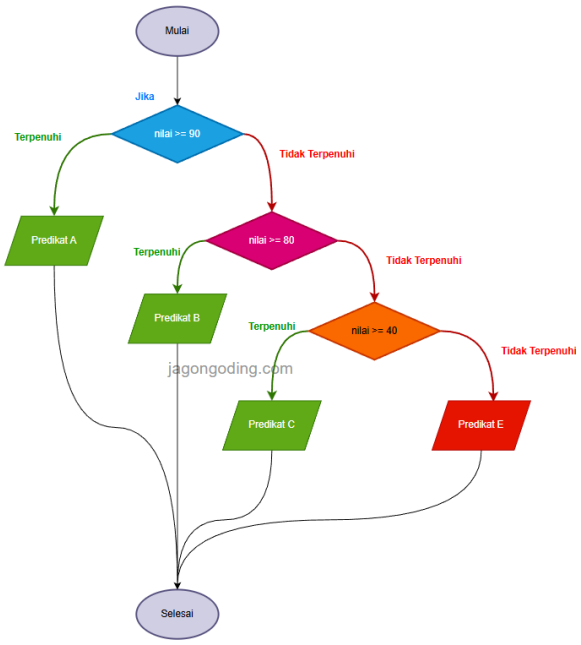
- Cek kondisi pada cabang if, jika kondisi bernilai True maka hanya blok if yang akan dieksekusi
- Jika bernilai False maka akan dilanjutkan dengan mengecek blok-blok elif secara berurutan, jika ada blok elif yang bernilai True, maka blok tersebutlah yang akan dieksekusi
- Jika tidak ada satupun diantara if dan elif yang bernilai True maka blok else akan dieksekusi



Contoh:

1. Jika nilai siswa lebih dari atau sama dengan 90, maka dia dapat predikat A.
2. Jika nilai siswa lebih dari atau sama dengan 80, maka dia dapat predikat B.
3. Jika nilai siswa lebih dari atau sama dengan 40, maka dia dapat predikat C.

4. Dan bila tidak ada satupun yang terpenuhi maka akan mengambil bagian else.



nilai = int(input("Masukkan nilai: "))
 	
if nilai >= 90:
  predikat = "Predikat A"
elif nilai >= 80:
  predikat = "Predikat B"
elif nilai >= 40:
  predikat = "Predikat C"
else:
  predikat = "Predikat E"
 	
print("Hasil:", predikat)
	Masukkan nilai: 38
	Hasil: Predikat E

4.2. Perulangan

Perulangan dalam dunia pemrograman adalah baris kode atau instruksi yang dieksekusi oleh komputer secara berulang-ulang sampai suatu kondisi tertentu terpenuhi . Dengan perulangan, kita bisa mengeksekusi suatu kode program berkali-kali dengan jumlah tertentu, atau selama sebuah kondisi tertentu terpenuhi.

Python memiliki dua jenis loop utama:

- for loop digunakan ketika jumlah perulangan sudah diketahui.
- while loop digunakan ketika perulangan bergantung pada kondisi (berhenti saat kondisi False).

A. For Loops

For digunakan untuk mengulangi sesuatu berdasarkan **urutan** (sequence), misalnya: list, string, range angka, tuple, dictionary, dll.

Contoh:

1. Bentuk umum pengunaan For loops

Fruits=["apple", "banana", "cherry"] for x in fruits: print(x)	**Output :** apple banana cherry

2. Looping pada String

for huruf in "BUAH": print(huruf)	**Output :** B U A H

3. Looping mengunakan statement Break

fruits=["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
  break
	Output :
	apple
	banana

4. Looping mengunakan Statement Continue

fruits=["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
  continue
  print(x)
	Output :
	apple
	cherry

5. Looping dengan parameter range

for x in range(2, 6): print(x)	**Output :** 2 3 4 5

B. While Loops

While digunakan ketika perulangan berjalan selama kondisi bernilai True.

Contoh:

1. Bentuk umum while loops

i = 1
while i < 6:
  print(i)
  i += 1
	Output :
	1
	2
	3
	4
	5

2. While dengan statement Continue

i = 0
while i < 6:
  i += 1
  if i == 3:
  continue
  print(i)
	Output :
	1
	2
	4
	5
	6

3. While dengan statement Break

i = 1
while i < 6:
  print(i)
  if i == 3:
  break
  i += 1
	Output :
	1
	2
	3