

Algoritma dan Python

SURYO HANDOYO 2025



do the
laundry



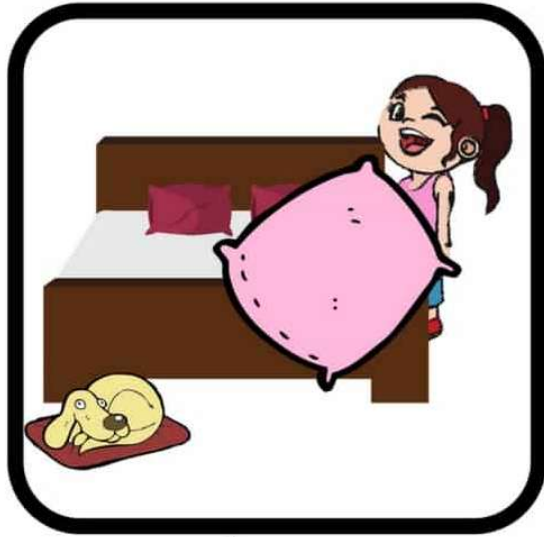
hang the
clothes



iron the
clothes



vacuum the
floor



make my
bed



go to bed



wake up



brush your
teeth



drive to
work



get home



take a bath



brush your
hair



get dressed



put on makeup



surf the net



play with
friends



go to school



go shopping



exercise



wash the car



go out
with a friend



take
pictures



play the
guitar



water the
plant



go for a walk



work



have breakfast



have lunch



have dinner



make dinner

Rundown Materi Algoritma

1. Pengenalan Algoritma & Dasar Python
2. Tipe Data & Operasi Dasar
3. Struktur Kontrol Percabangan (If-Else)
4. Struktur Perulangan
5. Fungsi dalam Algoritma
6. Struktur Data List dan String
7. Algoritma Pencarian
8. Algoritma Pengurutan
9. Studi Kasus Algoritma Dunia Nyata
10. Review & Mini Project

1. Pengenalan Algoritma dan Python

- **Algoritma** adalah **langkah-langkah atau urutan instruksi yang logis dan sistematis** untuk menyelesaikan suatu masalah atau melakukan suatu tugas. Misalnya memasak **mie instan**.
- Struktur dasar algoritma (input – proses – output)
- PRAKTEK 1 : Instalasi aplikasi python (download www.python.org)
- PRAKTEK 2 : Instalasi Visual Studio Code



- IDLE = Integrated Development and Learning Environment.
- Menulis dengan print untuk komentar dan variable (konstan atau dinamis)
- PRAKTEK 3 : Buatlah program untuk menulis nama dan usia anda

2. Tipe Data & Operasi Dasar

- Tipe data int, float, str, dan bool merupakan tipe data dasar dalam pemrograman. int untuk bilangan bulat, float untuk bilangan desimal, str untuk teks (string), dan bool untuk nilai boolean (true atau false).
- Operasi aritmatika adalah operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian. Operasi logika, di sisi lain, melibatkan manipulasi nilai benar atau salah, dan contohnya termasuk AND, OR, dan NOT
- Dalam Python, ada tujuh operator dasar: + (penjumlahan), - (pengurangan), * (perkalian), / (pembagian), % (modulo), // (pembagian dengan pendekatan), dan ** (eksponen)
- Konversi tipe data
- PRAKTEK : Buat program kalkulator sederhana (tambah, kurang, kali, bagi)

Contoh tipe data (variabel atau konstanta)

- **STRING** (huruf, angka, symbol, atau gabungan semuanya)
- tempat_sampah="Plastik, Kertas, Dus, Botol Aqua, Tissue"
- **FLOAT** (semua angka dan mengandung decimal)
- suhu=37,5 celcius
- **INT** (semua angka bilangan bulat dan tanpa desimal)
- Berat_kendaraan=2 ton
- **BOOL (logika, umumnya untuk berisi True dan False)**
- Lulus=True
- Male=False

CONTOH

Operator	Fungsi	Contoh
+	Penjumlahan	$5 + 2 \rightarrow 7$
-	Pengurangan	$5 - 2 \rightarrow 3$
*	Perkalian	$5 * 2 \rightarrow 10$
/	Pembagian	$5 / 2 \rightarrow 2.5$
//	Pembagian bulat	$5 // 2 \rightarrow 2$
%	Modulus (siswa bagi)	$5 \% 2 \rightarrow 1$
**	Pangkat	$2 ** 3 \rightarrow 8$

3. Struktur Kontrol Percabangan (If-Else)

1. Percabangan if, elif, dan else

Digunakan untuk membuat keputusan berdasarkan kondisi tertentu.

Struktur Dasar

if kondisi:

 # kode ini dijalankan jika kondisi benar

elif kondisi_lain:

 # kode ini dijalankan jika kondisi elif benar

else:

 # kode ini dijalankan jika semua kondisi di atas salah

CONTOH

- Bayangkan kamu ingin keluar rumah, dan kamu berpikir: "Kalau hujan, saya akan membawa payung."
- Kalimat itu adalah bentuk logika if!
- Dalam Python, kamu bisa menuliskannya seperti ini:

```
hujan = True

if hujan:
    print("Bawa payung")
```

```
hujan = False

if hujan:
    print("Bawa payung")
else:
    print("Tidak perlu bawa payung")
```

```
nilai = 75
```

```
if nilai >= 90:  
    print("Nilai A")  
elif nilai >= 75:  
    print("Nilai B")  
else:  
    print("Nilai C")
```

Penjelasan:

Jika nilai lebih dari atau sama dengan 90, akan muncul "Nilai A".

Jika tidak, tetapi nilai lebih dari atau sama dengan 75, muncul "Nilai B".

Jika tidak memenuhi semua, muncul "Nilai C"

2. Operator Logika

a. not

Membalikkan nilai logika (True menjadi False dan sebaliknya).

```
hujan = False
```

```
if not hujan:
```

```
    print("Tidak hujan, bisa jalan-jalan.")
```

Penjelasan: Karena `is_raining` adalah False, maka `not is_raining` menjadi True, dan kode di dalam if dijalankan.

b. or

Menghasilkan True jika salah satu kondisi bernilai True.

```
umur = 16
```

```
punya_sim = False
```

```
if umur >= 17 or punya_sim:
```

```
    print("Boleh mengemudi")
```

```
else:
```

```
    print("Tidak boleh mengemudi")
```

Penjelasan: Karena umur kurang dari 17 dan tidak punya SIM, maka hasilnya False dan output-nya "Tidak boleh mengemudi".

3. Contoh Gabungan

usia = 20

pelajar = False

if usia < 18:

 print("Masih anak-anak")

elif not pelajar or usia > 18:

 print("Bukan pelajar atau sudah dewasa")

else:

 print("Pelajar dewasa")

4. Struktur Pengulangan

- Di Python, struktur pengulangan digunakan untuk menjalankan sekelompok kode berulang kali selama kondisi tertentu masih terpenuhi. Ada dua jenis pengulangan utama seperti berikut.

1. Pengulangan for

Digunakan untuk mengulang sesuatu sebanyak jumlah tertentu, biasanya digunakan untuk iterasi terhadap list, string, atau range (rentang angka).

Struktur dasar:

```
for variabel in iterable:  
    # kode yang diulang
```

Iterasi/iterable, secara umum, adalah proses pengulangan atau perulangan suatu tindakan atau serangkaian langkah untuk mencapai hasil yang diinginkan.

Contoh 1 – Mengulang angka dengan range()

```
for i in range(5):  
    print("Halo ke-", i)
```

Output:

Halo ke- 0
Halo ke- 1
Halo ke- 2
Halo ke- 3
Halo ke- 4

Contoh 2 – Mengulang isi list python

```
buah = ["apel", "jeruk", "pisang"]  
  
for item in buah:  
    print(item)
```

Output:

apel
jeruk
pisang

2. Pengulangan while

Digunakan untuk mengulang selama kondisi bernilai True. Cocok digunakan saat jumlah pengulangan tidak diketahui sebelumnya.

Struktur dasar:

while kondisi:

 # kode yang diulang

Contoh:

```
angka = 1
```

```
while angka <= 5:
```

```
    print("Angka sekarang:", angka)
```

```
    angka += 1
```

Output:

Angka sekarang: 1

Angka sekarang: 2

Angka sekarang: 3

Angka sekarang: 4

Angka sekarang: 5

Kontrol Pengulangan

break → Menghentikan pengulangan

```
for i in range(10):  
    if i == 5:  
        break  
    print(i)
```

Output:

0 sampai 4 (karena berhenti di 5)

continue → Melewati satu iterasi, lalu lanjut

```
for i in range(5):  
    if i == 2:  
        continue  
    print(i)
```

Output: 0, 1, 3, 4 (karena 2 dilewati)

else dalam pengulangan

```
for i in range(3):  
    print("Iterasi ke-", i)  
else:  
    print("Selesai!")
```

Output:

Iterasi ke- 0

Iterasi ke- 1

Iterasi ke- 2

Selesai!

TIPS:

Gunakan for jika kamu tahu berapa kali ingin mengulang.

Gunakan while jika pengulangan tergantung suatu kondisi yang bisa berubah saat program berjalan.

Jangan lupa untuk menghindari while tanpa akhir (infinite loop)

6. Struktur Data LIST dan String

- List adalah kumpulan nilai (elemen) yang tersusun secara berurutan dan dapat diubah (mutable).

Ciri-ciri List:

- Dapat menyimpan banyak nilai.
- Tertata berdasarkan indeks (mulai dari 0).
- Bisa berisi campuran tipe data.
- Bisa diubah: ditambah, dihapus, diubah elemennya

Cara Membuat List

List kosong

```
data = []
```

List dengan isi awal

```
angka = [1, 2, 3, 4, 5]
```

```
buah = ["apel", "jeruk", "mangga"]
```

```
campuran = [10, "sapi", True, 3.14]
```

Mengakses Elemen List

```
buah = ["apel", "jeruk", "mangga"]
```

```
print(buah[0]) # 'apel'
```

```
print(buah[2]) # 'mangga'
```

Mengubah Nilai di List

```
buah[1] = "semangka"  
print(buah) # ['apel', 'semangka', 'mangga']
```

Menambahkan dan Menghapus Elemen
Tambah:

```
buah.append("pisang") # Tambah di akhir  
buah.insert(1, "nanas") # Tambah di indeks ke-1
```

Hapus:

```
buah.remove("apel") # Hapus berdasarkan nilai  
buah.pop(2) # Hapus berdasarkan indeks (misal  
indeks 2)
```

Pengulangan pada List

```
for item in buah:  
    print(item)
```

Fungsi List yang Sering Dipakai

Fungsi / Method	Keterangan
-----------------	------------

append(x)	Menambah elemen di akhir
-----------	--------------------------

insert(i, x)	Menyisipkan x di posisi i
--------------	---------------------------

remove(x)	Menghapus elemen dengan nilai x
-----------	---------------------------------

pop(i)	Menghapus elemen di indeks i
--------	------------------------------

len(list)	Menghitung jumlah elemen
-----------	--------------------------

list.sort()	Mengurutkan list (jika tipe cocok)
-------------	------------------------------------

list.reverse()	Membalik urutan list
----------------	----------------------

list.index(x)	Mengembalikan indeks pertama dari x
---------------	-------------------------------------

list.count(x)	Menghitung jumlah kemunculan x
---------------	--------------------------------

Contoh Program Singkat

```
belanja = []
```

```
# Input data belanja
```

```
for i in range(3):
```

```
    barang = input("Masukkan nama barang: ")
```

```
    belanja.append(barang)
```

```
print("Daftar belanja Anda:", belanja)
```

1. List Dua Dimensi (Multi List)

Definisi:

List dua dimensi adalah list yang berisi list lain di dalamnya, seperti tabel atau matriks.

Contoh:

```
data_produk = [  
    ["Pensil", 10, 2000],  
    ["Pulpen", 5, 3500],  
    ["Penghapus", 7, 1500]  
]
```


Akses Elemen:

```
data_produk = [  
    "Pensil", 10, 2000,  
    "Pulpen", 5, 3500,  
    "Penghapus", 7, 1500  
]  
print(data_produk[0])      ['Pensil', 10, 2000]  
print(data_produk[0][0])   'Pensil'  
print(data_produk[1][2])   3500
```

Menampilkan seluruh data:

```
for item in data_produk:  
    print("Produk:", item[0], "- Qty:", item[1], "- Harga:", item[2])
```