

Lembar Contekan Python: Basic

Bab 1-15 | dari nol sampai program kecil yang berguna

Terminal dan Python

Bab 3-4

```
$ cd folder_proyek # pindah folder [ref]
$ python halo.py # jalankan skrip [ref]
$ python --version # cek versi
print('Halo', 3, sep=', ', end='!')
# cetak -> Halo, 3!
# komentar: catatan untuk manusia
```

Variabel dan tipe data

Bab 5

```
n, h, ok = 42, 1.5, True
# int, float, bool
nama = 'Jati' # str (teks)
type(h).__name__ # nama tipe -> 'float'
int('42') + 1 # teks ke angka -> 43
float('2.5') # teks ke desimal -> 2.5
str(7) + ' kg' # angka ke teks -> '7 kg'
bool(0), bool('a')
# kosong/nol = False -> (False, True)
f'{3.14159:.2f}' # dua desimal -> '3.14'
f'{1234567:,}'
# pemisah ribuan -> '1,234,567'
x = None # tidak ada nilai
```

Input

Bab 5-6

```
umur = int(input('Umur? '))
# input() selalu teks!
umur + 1 # setelah diubah ke int -> 43
```

Operator

Bab 6

```
7 / 2, 7 // 2, 7 % 2
# bagi, bulat, sisa -> (3.5, 3, 1)
2 ** 10 # pangkat -> 1024
divmod(1000, 30)
# (hasil bagi, sisa) -> (33, 10)
round(3.14159, 2) # pembulatan -> 3.14
1 < 2 < 3 # perbandingan berantai -> True
(1 > 2) or not (3 > 4)
# and, or, not -> True
'at' in 'Jati' # ada di dalam? -> True
n = 5 # tambah lalu simpan
n += 2
```

Kondisi

Bab 7

```
n = 520 # jika / atau jika / selain itu
if n >= 500:
    s = 'siap'
elif n >= 400:
    s = 'hampir'
else:
    s = 'belum'
s # -> 'siap'
'siap' if n >= 500 else 'belum'
# satu baris -> 'siap'
jenis = 'sapi' # pencocokan pola
match jenis:
    case 'sapi' | 'kambing':
        k = 'ruminansia'
    case _:
        k = 'lain'
k # -> 'ruminansia'
```

Perulangan

Bab 8

```
total = 0 # penampung
for b in [3, 5, 7]:
    total += b
total # -> 15
list(range(0, 10, 3))
# awal, akhir (tak ikut), langkah -> [0,
# 3, 6, 9]
list(enumerate('ab', 1))
# nomor + isi -> [(1, 'a'), (2, 'b')]
g = 1800 # while + break
while True:
    g += 150
    if g >= 2400:
        break
g # -> 2400
[i for i in range(5) if i % 2 == 0]
# list comprehension -> [0, 2, 4]
```

List dan slicing

Bab 9

```
xs = [3, 1, 2] # list
xs.append(4) # tambah di akhir
xs[0], xs[-1]
# pertama, terakhir -> (3, 4)
xs[1:3] # irisan [awal:akhir] -> [1, 2]
sorted(xs)
#urut (asli tetap) -> [1, 2, 3, 4]
```

List dan slicing ...

Bab 9

```
xs.sort(reverse=True) # urutkan di tempat
xs[::-1] # dibalik -> [1, 2, 3, 4]
len(xs), sum(xs), max(xs)
# banyak, jumlah, terbesar -> (4, 10, 4)
```

Dict, tuple, set

Bab 9

```
d = {'Jati': 32.5, 'Mahoni': 28.0}
# dict: kunci -> nilai
d['Meranti'] = 45.2 # tambah/ubah
d.get('Sengon', 0)
# nilai bawaan bila tak ada -> 0
{k: v for k, v in d.items() if v > 30}
# dict comprehension -> {'Jati': 32.5,
# 'Meranti': 45.2}
lintang, bujur = (-6.6, 106.8)
# tuple: tak bisa diubah
{1, 2, 3} & {2, 3, 4}
# set: irisan -> {2, 3}
{1, 2} | {2, 3}
# set: gabungan -> {1, 2, 3}
```

Fungsi

Bab 10

```
def luas(r, pi=3.14): # nilai bawaan pi
    return pi * r * r
luas(2) # pakai nilai bawaan -> 12.56
luas(r=2, pi=3) # argumen bernama -> 12
kali = lambda a, b: a * b # fungsi mini
kali(3, 4) # -> 12
sorted(['bb', 'a'], key=len)
# kunci urutan -> ['a', 'bb']
# tanpa return, hasilnya None
```

Modul

Bab 11

```
import math # import modul
round(math.pi, 3) # -> 3.142
from datetime import date, timedelta
# ambil sebagian
date(2026, 1, 15) + timedelta(days=270)
# hitung tanggal -> datetime.date(2026,
# 10, 12)
date(2026, 10, 12).strftime('%d-%m-%Y')
# format tanggal -> '12-10-2026'
import random # acak berbenih
random.seed(7)
random.sample(range(1, 13), 3)
# -> [6, 3, 7]
```

Error

Bab 12

```
def ke_angka(t): # tangkap error
    try:
        return float(t)
    except ValueError:
        return None
ke_angka('12.5'), ke_angka('abc')
# aman dari teks salah -> (12.5, None)
try: # raise, except, finally
    raise ValueError('rusak')
except (ValueError, KeyError) as e:
    pesan = str(e)
finally:
    pass
pesan # -> 'rusak'
# error buatan sendiri
class DataError(Exception):
    pass
assert 1 + 1 == 2, 'harus 2'
# periksa asumsi
import logging # catat peringatan
logging.warning('pH aneh')
```

File dan Path

Bab 13

```
from pathlib import Path # jalur berkas
p = Path('a.txt')
# w timpa, a tambah, r baca
with open(p, 'w', encoding='utf-8') as f:
    f.write('1\n2\n')
# baca per baris
with open(p, encoding='utf-8') as f:
    total = sum(int(b) for b in f)
total # -> 3
p.read_text(encoding='utf-8')
# baca semua -> '1\n2\n'
q = Path('data/b.txt') # bagian nama jalur
q.name, q.suffix, q.stem
# -> ('b.txt', '.txt', 'b')
p.exists() # ada berkasnya? -> True
```

CSV dan JSON

Bab 14

```
import csv, json # modul bawaan
# newline='' wajib
with open('t.csv', 'w', newline='',
          encoding='utf-8') as f:
    w = csv.DictWriter(f, ['k', 'n'])
    w.writeheader()
    w.writerow({'k': 'A', 'n': 800})
# semua nilai jadi teks!
with open('t.csv', newline='',
          encoding='utf-8') as f:
    baris = list(csv.DictReader(f))
    baris # -> [{'k': 'A', 'n': '800'}]
json.dumps({'a': '0'}, ensure_ascii=False)
# objek ke teks JSON -> '{"a": "0"}'
json.loads('{"a": [1, 2]}')['a']
# teks JSON ke objek -> [1, 2]
# JANGAN pickle.load dari sumber asing!
```

Excel dan PDF

Bab 15

```
import openpyxl as xl # pustaka Excel
wb = xl.Workbook() # tulis .xlsx
ws = wb.active
ws.append(['kolam', 'kg'])
ws.append(['A', 432.4])
wb.save('a.xlsx')
wb2 = xl.load_workbook('a.xlsx')
# baca sel
wb2.active['B2'].value # -> 432.4
# tulis PDF
from reportlab.pdfgen import canvas
c = canvas.Canvas('a.pdf')
c.drawString(72, 750, 'Laporan')
c.save()
from pypdf import PdfReader # baca PDF
len(PdfReader('a.pdf').pages) # -> 1
```