

Lembar Contekan Python: Intermediate

Bab 16-43 | kode rapi, data, statistik, aplikasi pertama

OOP: kelas dan pewarisan

Bab 16

```
class Pohon: # cetakan objek
    def __init__(self, jenis, d):
        self.jenis, self.d = jenis, d
    def __str__(self):
        return f'{self.jenis} {self.d}'
str(Pohon('Jati', 32.5))
# buat objek -> 'Jati 32.5'
class Besar(Pohon): # pewarisan
    def __str__(self):
        return 'BESAR ' + super().__str__()
str(Besar('Jati', 45)) # -> 'BESAR Jati 45'
class Kolam: # properti: tanpa ()
    @property
    def ph(self):
        return 7.0
Kolam().ph # -> 7.0
# dataclass
from dataclasses import dataclass
@dataclass
class Ikan:
    nama: str
Ikan('lele')
# __repr__ otomatis -> Ikan(nama='lele')
```

Lingkungan, pytest, ruff, logging

Bab 17-19

```
$ python -m venv .venv
# kamar Python sendiri [cek]
$ .venv\Scripts\activate
# Linux: source .venv/bin/activate [ref]
$ uv add pandas # tambah dependensi [cek]
$ uv run pytest # jalan di lingkungan proyek [cek]
$ uv sync # samakan dengan uv.lock [cek]
$ pytest -x -k nama
# berhenti di gagal pertama [cek]
import pytest # tes *.py
P = [(1, 1), (2, 4)]
@pytest.mark.parametrize('n,h', P)
def test_k(n, h):
    assert n * n == h
def test_e():
    with pytest.raises(ZeroDivisionError):
        1 / 0 # -> 3 passed
$ ruff check . --fix
# periksa gaya + perbaiki [cek]
$ ruff format . # rapikan kode [cek]
import logging, sys # log berjenjang
logging.basicConfig(level=20, force=True,
                    stream=sys.stdout)
logging.info('mulai') # -> INFO:root:mulai
```

Git dan GitHub

Bab 20

```
$ git init -b main # mulai repo (a.txt sudah ada)
$ git add -A # siapkan perubahan
$ git commit -m 'pesan' # simpan satu titik waktu
$ git switch -c fitur # cabang baru
$ git switch main # kembali
$ git merge fitur # gabung cabang
$ git log --oneline # riwayat singkat
$ git push -u origin main # kirim ke GitHub [cek]
# .gitignore: .venv/ .env (rahasia!)
```

API dengan requests

Bab 21

```
import requests
# url = alamat API (di sini: server lokal)
# selalu beri timeout
r = requests.get(url, params={'q': 1},
                timeout=10)
r.status_code, r.json() # -> (200, {'n': 3})
# tak ada = 404
r2 = requests.get(url + '/x', timeout=10)
r2.status_code # -> 404
r.raise_for_status() # error bila 4xx/5xx
```

Regex

Bab 22

```
import re
t = 'JT-0042, MR-0107, xx-12'
re.findall(r'\b[A-Z]{2}-\d{4}\b', t)
# semua kode -> ['JT-0042', 'MR-0107']
re.search(r'd=(\d+)', 'd=45 cm')[1]
# ambil grup () -> '45'
re.sub(r'\s+', ' ', 'a b c')
# ganti pola -> 'a b c'
# grup bernama
m = re.search(r'(?P<k>\w+)=(?P<n>\d+)', 't=14')
m.groupdict() # -> {'k': 't', 'n': '14'}
re.fullmatch(r'\d{3}', '123') is not None
# cocok seluruhnya? -> True
```

numpy dan pandas

Bab 23

```
import numpy as np
a = np.array([480, 500, 510])
a[a > 490] # topeng boolean -> array([500, 510])
import pandas as pd # tabel
df = pd.DataFrame({'k': list('ABC'),
                  'j': ['lele', 'nila', 'lele'],
                  'n': [800, 1250, 600]})
df.shape # -> (3, 3)
df['kg'] = df.n * 0.5 # kolom baru
df[df.kg > 300].k.tolist()
# saring baris -> ['A', 'B']
df.groupby('j')['kg'].sum().to_dict()
# ringkas per kelompok -> {'lele': 700.0, 'nila': 625.0}
df.sort_values('kg').k.iloc[-1] # terbesar -> 'B'
```

numpy dan pandas ...

Bab 23

```
# gabung tabel
h = pd.DataFrame({'j': ['lele', 'nila'],
                  'rp': [3, 2]})
df.merge(h, on='j').shape # -> (3, 5)
df.to_csv('a.csv', index=False) # simpan CSV
pd.read_csv('a.csv').shape # baca CSV -> (3, 4)
```

SQL: sqlite3 dan SQLAlchemy

Bab 24

```
import sqlite3 # tanda ? anti injeksi
c = sqlite3.connect(':memory:')
c.execute('CREATE TABLE p (j TEXT, d REAL)')
r = [(('Jati', 32.5), ('Mahoni', 28.0))]
c.executemany('INSERT INTO p VALUES (?,?)', r)
# kueri berparameter
q = 'SELECT j FROM p WHERE d > ?'
c.execute(q, (30,)).fetchall() # -> [('Jati',)]
# ORM: kelas = tabel
import sqlalchemy as sa
from sqlalchemy.orm import (DeclarativeBase,
                             Mapped, Session, mapped_column)
class B(DeclarativeBase):
    pass
class Sektor(B):
    tablename = 's'
    id: Mapped[int] = mapped_column(
        primary_key=True)
    nama: Mapped[str]
e = sa.create_engine('sqlite:///')
B.metadata.create_all(e)
with Session(e) as s:
    s.add(Sektor(nama='Ikan'))
    s.commit()
n = len(s.scalars(sa.select(Sektor)).all())
n # -> 1
with e.connect() as k: # SQL mentah
    q = sa.text('SELECT COUNT(*) FROM s')
    jml = k.execute(q).scalar()
jml # -> 1
```

Statistik dan statsmodels

Bab 26-27

```
# uji t dua kelompok
from scipy import stats
a, b = [30, 32, 31, 29], [35, 36, 34, 37]
stats.ttest_ind(a, b).pvalue < 0.05 # -> True
round(stats.norm.cdf(30, 25, 5), 3)
# peluang normal -> 0.841
round(stats.pearsonr(a, b)[0], 2)
# korelasi -> -0.4
# OLS: [konstanta, lereng]
import statsmodels.api as sm
X = sm.add_constant([1, 2, 3, 4, 5])
y = [2.1, 3.9, 6.2, 7.8, 10.1]
res = sm.OLS(y, X).fit()
# ekspresi generator
res.params.round(2).tolist() # -> [0.05, 1.99]
round(res.rsquared, 3) # R kuadrat -> 0.997
# logistik: rasio odds
import numpy as np, pandas as pd
import statsmodels.formula.api as smf
d = pd.DataFrame({'x': range(1, 9),
                  'y': [0, 0, 1, 0, 1, 0, 1, 1]})
r = smf.logit('y ~ x', d).fit(disps=0)
round(float(np.exp(r.params['x'])), 2) # -> 1.81
```

Generator, decorator, closure

Bab 30-31

```
def genap(n): # yield = satu per satu
    for i in range(2, n + 1, 2):
        yield i
list(genap(10)) # -> [2, 4, 6, 8, 10]
# ekspresi generator
g = (i * i for i in range(3))
next(g), next(g) # -> (0, 1)
def pengali(k): # closure mengingat k
    return lambda x: x * k
pengali(3)(7) # -> 21
import functools # decorator + wraps
def hitung(f):
    @functools.wraps(f)
    def b(*a):
        b.n += 1
        return f(*a)
        b.n = 0
    return b
@hitung
def tambah(a, b):
    return a + b
tambah(1, 2), tambah.n
# n = jumlah panggilan -> (3, 1)
```

Type hint dan Pydantic

Bab 33-34

```
# int atau None
def cari(xs: list[str], x: str) -> int | None:
    return xs.index(x) if x in xs else None
cari(['a', 'b'], 'b'), cari(['a'], 'z')
# -> (1, None)
$ mypy --strict app.py
# periksa tipe sebelum jalan [cek]
# model = penjaga pintu
from pydantic import BaseModel, Field
class Pohon(BaseModel):
    jenis: str
    d: float = Field(gt=0)
Pohon(jenis='Jati', d=32.5)
# '32.5' diubah ke angka -> Pohon(jenis='Jati',
# d=32.5)
try: # galat rapi
```

Type hint dan Pydantic ...

Bab 33-34

```
Pohon(jenis='Jati', d=-5)
except Exception as e:
    pesan = e.errors()[0]['msg']
pesan # -> 'Input should be greater than 0'
# pengaturan (bisa dari env)
from pydantic.settings import BaseSettings
class Atur(BaseSettings):
    folder: str = 'hasil'
Atur().folder # -> 'hasil'
```

Async dan thread

Bab 35

```
import asyncio # serempak
async def f(n):
    await asyncio.sleep(0.01)
    return n * 2
async def main():
    return await asyncio.gather(f(1), f(2))
asyncio.run(main()) # -> [2, 4]
async def g(): # TaskGroup
    async with asyncio.TaskGroup() as tg:
        t = tg.create_task(f(1))
        return t.result()
asyncio.run(g()) # -> 2
# thread untuk tunggu I/O
import concurrent.futures as cf
with cf.ThreadPoolExecutor(2) as p:
    hasil = list(p.map(abs, [-1, -2]))
hasil # -> [1, 2]
```

Web: FastAPI dan Flask

Bab 36

```
# jalur + parameter
from fastapi import FastAPI
from fastapi.testclient import TestClient
app = FastAPI()
@app.get('/kolam/{n}')
def kolam(n: str, ph: float = 7.0):
    return {'n': n, 'ph': ph}
c = TestClient(app)
c.get('/kolam/?ph=6.8').json()
# -> {'n': 'A', 'ph': 6.8}
c.get('/kolam/?ph=x').status_code
# tipe salah = 422 -> 422
$ uvicorn app:app --reload # jalankan server [cek]
# Flask: yang ringkas
from flask import Flask
w = Flask(__name__)
@w.get('/halo')
def halo():
    return {'p': 'hai'}
w.test_client().get('/halo').json
# -> {'p': 'hai'}
```

GeoPandas dan Rasterio

Bab 38-39

```
import geopandas as gpd # titik + CRS
from shapely.geometry import Point
pts = [Point(110, -7), Point(110.01, -7.01)]
g = gpd.GeoDataFrame({'n': [1, 2]}, geometry=pts,
                    crs='EPSG:4326')
g.crs.to_epsg() # -> 4326
u = g.to_crs(32749) # ke meter (UTM 49S)
round(u.geometry[0].distance(u.geometry[1]))
# jarak (m) -> 1563
round(u.buffer(100).area[0])
# luas penyangga (m2) -> 31365
# simpan/baca GeoPackage
g.to_file('a.gpkg')
gpd.read_file('a.gpkg').shape # -> (2, 2)
# tulis GeoTIFF
import numpy as np, rasterio
from rasterio.transform import from_origin
a = np.arange(4, dtype='float32').reshape(2, 2)
t = from_origin(110, -7, 0.1, 0.1)
with rasterio.open('a.tif', 'w', driver='GTiff',
                  height=2, width=2, count=1,
                  dtype='float32', crs='EPSG:4326',
                  transform=t) as d:
    d.write(a, 1)
# baca pita 1 + CRS
with rasterio.open('a.tif') as s:
    b, e = s.read(1), s.crs.to_epsg()
(b.mean().item(), e) # -> (1.5, 4326)
# NDVI
n, r = np.array([.8, .6]), np.array([.2, .3])
np.round((n - r) / (n + r), 2).tolist()
# -> [0.6, 0.33]
```

Context manager dan automasi

Bab 32, 41-43

```
# abaikan satu jenis error
from contextlib import suppress
with suppress(FileNotFoundError):
    open('tidak_ada').close()
# jalankan program lain
import subprocess, sys
cmd = [sys.executable, '-c', 'print(2)']
p = subprocess.run(cmd, text=True,
                  capture_output=True)
p.stdout.strip() # -> '2'
import argparse # argumen baris perintah
a = argparse.ArgumentParser()
a.add_argument('--n', type=int, default=3)
a.parse_args(['--n', '5']).n # -> 5
$ streamlit run app.py # dashboard browser [cek]
```

Buku "Python dari Nol sampai Sangat Advanced", Lampiran B. Python 3.14. Setiap baris kode dijalankan sungguhan; "->" = hasil asli. Angka numpy ditulis tanpa pembungkus np.float64(...).

[cek] = opsi atau fungsi diperiksa ada (tidak dijalankan penuh). [ref] = referensi, tidak dijalankan di sini. \$ = perintah terminal.