BUKU RESEP UNTUK KODE

```
for photo in photos:
    rename(photo)

print("Beres!")
```

Python buat Kreator Sibuk

Empat bab pendek untuk mengotomatiskan bagian membosankan dari mengelola akun masakan

DITULIS UNTUK NADIA · DIBUAT BERSAMA CLAUDE

Daftar Isi

1 Kode itu cuma resep **1**
Apa itu program 1
Install Python 2
Baris kode pertamamu 2
Barusan itu apa yang terjadi 3

2 Caption otomatis **5**
Variabel itu toples berlabel 5
Mengisi template 6
Ngapain repot? 7
Dua trik kecil untuk teks 7

3 Rename 50 foto dalam 2 detik **9**
Daftar berisi banyak hal 9
For loop 9
Rename file: dry run dulu 11
Sekarang beneran 12

4 Post mana yang paling bagus? **14**
Apa itu file CSV 14
Membacanya dengan Python 15
Urutkan berdasarkan saves 16
Rata-rata 17
Langkah berikutnya 19

BAB 1

Kode itu cuma resep

Minggu malam. Ada 40 foto baru dari syuting minggu ini, tiga caption yang belum ditulis, dan perasaan samar bahwa video Es Cendol bulan lalu performanya bagus banget. Seberapa bagus? Nggak tahu, karena ngeceknya harus scroll insights satu per satu.

Semua ini nggak susah. Cuma membosankan. Dan kerjaan membosankan yang diulang-ulang itu justru keahlian komputer.

Buku ini ngajarin Python secukupnya supaya kerjaan-kerjaan itu bisa kamu serahkan ke laptop. Nggak perlu kuliah ilmu komputer. Empat bab pendek, satu ide dalam satu waktu.

Apa itu program

Kamu sudah biasa ngikutin resep. "Panaskan minyak. Masukkan bawang merah. Aduk dua menit." Tiap langkahnya kecil dan jelas, dan urutannya penting.

Program itu resep untuk komputer. Bedanya cuma satu: komputer ngikutin resepnya *persis*. Dia nggak pernah nebak maksudmu. Kalau resepnya bilang "tambahkan garam" tapi lupa bilang berapa, koki manusia pakai feeling. Komputer berhenti dan protes.

Kedengarannya nyebelin, tapi justru ini kabar baiknya. Begitu

programmu jalan sekali, dia akan jalan dengan cara yang sama setiap kali, mau rename 5 foto atau 5.000 foto.

Python adalah bahasa yang kita pakai untuk menulis resepnya. Gratis, populer, dan bacanya hampir kayak bahasa Inggris biasa. Makanya banyak pemula mulai dari sini.

Install Python

Ini cukup sekali saja.

- **Windows:** buka python.org, klik *Downloads*, lalu jalankan installer-nya. Di layar pertama, centang **"Add python.exe to PATH"** sebelum klik *Install*. (PATH itu daftar tempat komputer mencari program. Dengan dicentang, Python bisa dijalankan dari mana saja.)
- **Mac:** buka python.org dan install versi terbaru untuk macOS.

Untuk cek apakah berhasil, buka **terminal**. Itu aplikasi tempat kamu mengetik perintah, bukan klik-klik. Di Windows namanya *Terminal* atau *Command Prompt*. Di Mac namanya *Terminal*. Ketik ini lalu tekan Enter:

```
python3 --version
```

Di Windows, kalau itu nggak jalan, coba `python --version`. Harusnya muncul sesuatu seperti `Python 3.12.3`. Versi berapa pun yang depannya 3 sudah oke.

Baris kode pertamamu

Bikin folder baru di desktop namanya `creator-scripts`. Buka text editor biasa (VS Code gratis dan ramah pemula), lalu bikin

file di folder itu namanya `halo.py`. Akhiran `.py` memberi tahu komputer bahwa ini resep Python.

Ketik dua baris ini di file-nya:

```
print("Halo, Nadia!")
print("Dapur Nadia: masakan simpel, rasa juara")
```

Simpan. Lalu di terminal, masuk ke folder itu dan jalankan:

```
cd Desktop/creator-scripts
python3 halo.py
```

Output:

```
Halo, Nadia!
Dapur Nadia: masakan simpel, rasa juara
```

Udah. Kamu barusan nulis program dan komputernya ngikutin.

Barusan itu apa yang terjadi

`print()` itu **fungsi**: resep mini siap pakai yang sudah dikenal Python. Kamu kasih sesuatu di dalam kurungnya, dan dia menampilkannya di layar.

Kata-kata di dalam tanda kutip disebut **string**. String itu ya teks biasa. Tanda kutipnya memberi tahu Python di mana teks mulai dan selesai, supaya tagline-mu nggak dibaca sebagai perintah.

Python menjalankan barisnya dari atas ke bawah, satu per satu. Sama kayak resep.

> **Coba sendiri.** Ubah `halo.py` supaya baris pertama menampilkan nama akunmu dan baris kedua tagline-mu sendiri. Jalankan lagi. Lalu hapus salah satu tanda kutipnya dan jalankan sekali lagi, cuma untuk lihat bentuk

error itu kayak apa. (Santai aja. Error itu cara komputer bilang "aku nggak ngerti langkah yang ini.")

Rangkuman

- Program itu resep yang diikuti komputer persis, dari atas ke bawah.
- Python ditulis di file `.py` dan dijalankan dengan `python3 namafile.py`.
- `print()` menampilkan sesuatu di layar, dan teks ditulis di dalam tanda kutip.

BAB 2

Caption otomatis

Lihat sepuluh caption terakhirmu. Kemungkinan polanya sama: nama masakan, berapa lama masaknya, ajakan untuk save, dan hashtag yang itu-itu juga. Pola itu kamu tulis ulang manual setiap kali, dan kadang hashtag-nya lupa.

Di bab ini kamu nulis polanya sekali, lalu biar Python yang ngisi.

Variabel itu toples berlabel

Di dapur kamu nyimpan bahan di toples berlabel: *gula*, *garam*, *cabai bubuk*. Waktu resep bilang "masukkan gula", kamu nggak perlu ingat ada di rak mana. Cukup cari labelnya.

Variabel itu toples berlabel untuk satu informasi. Kamu isi sekali, setelah itu cukup panggil labelnya.

```
account = "Dapur Nadia"
dish = "Sambal Matah"
minutes = 15

print(account)
print(dish)
print(minutes + 5)
```

Output:

```
Dapur Nadia
```

```
Sambal Matah
20
```

Tanda `=` di sini bukan "sama dengan" kayak di pelajaran matematika. Artinya "masukkan ini ke toples". Jadi `dish = "Sambal Matah"` artinya: ambil teks *Sambal Matah* dan masukkan ke toples berlabel `dish`.

Perhatikan, `minutes` nggak pakai tanda kutip. Itu bikin dia jadi **angka**, bukan teks, jadi Python bisa ngitung pakai dia. `minutes + 5` hasilnya 20.

(Nama toplesnya sengaja pakai bahasa Inggris, karena hampir semua contoh kode di internet begitu. Isinya boleh bahasa apa saja.)

Mengisi template

Sekarang bagian yang berguna. Python punya yang namanya **f-string**: string dengan huruf `f` di depannya, di mana apa pun yang ada di dalam kurung kurawal `{ }` akan diganti dengan isi toples itu.

```
account = "Dapur Nadia"
dish = "Sambal Matah"
minutes = 15

caption = f"Resep baru di {account}: {dish}! Jadi dalam
↪  {minutes} menit. Save dulu biar nggak lupa."
print(caption)
print(len(caption), "karakter")
```

Output:

```
Resep baru di Dapur Nadia: Sambal Matah! Jadi dalam 15
↪  menit. Save dulu biar nggak lupa.
```

```
88 karakter
```

Template caption-mu sekarang tinggal satu baris. Besok cukup ganti `dish` dan `minutes`, jalankan file-nya, lalu copy hasilnya.

Baris terakhir pakai fungsi lain, `len()`, yang menghitung panjang sesuatu. Berguna kalau platformnya punya batas karakter.

Ngapain repot?

Pertanyaan wajar. Ngetik satu caption manual cuma semenit.

Tapi template adalah awal dari otomasi. Begitu caption-mu jadi resep, bukan sesuatu yang kamu ketik ulang, kamu bisa bikin 30 sekaligus, hashtag-nya selalu konsisten, atau ganti ajakan di semua caption cukup dengan satu edit. Di bab berikutnya kita belajar cara mengulang resep berkali-kali, dan di situlah mulai terasa hemat waktunya.

Dua trik kecil untuk teks

String punya alat bawaan. Tulis titik setelah nama toplesnya, lalu nama alatnya:

- `dish.upper()` mengubah *Es Teler* jadi *ES TELER*. Cocok untuk kata pertama yang mencolok.
- `dish.lower()` kebalikannya. Cocok untuk hashtag.

Kamu juga bisa bikin baris baru di dalam string dengan `\n` (garis miring terbalik dan huruf n).

> **Coba sendiri.** Bikin file `caption.py` yang menyusun caption untuk post besok. Pakai empat toples: `dish`, `main_ingredient`, `minutes`, dan `hashtags`. Bikin nama

masakannya mencolok pakai `.upper()` dan taruh hashtag-nya di baris sendiri.

Salah satu jawabannya:

```
dish = "Es Teler"
main_ingredient = "kelapa muda"
minutes = 10
hashtags = "#esteler #resepmudah #minumansegar"

caption = f"{dish.upper()} cuma {minutes} menit!
↪  Rahasianya {main_ingredient} yang segar.\n{hashtags}"
print(caption)
```

Output:

```
ES TELER cuma 10 menit! Rahasianya kelapa muda yang
↪  segar.
#esteler #resepmudah #minumansegar
```

Rangkuman

- Variabel itu toples berlabel: `dish = "Sambal Matah"` memasukkan teks ke toples bernama `dish`.
- Teks butuh tanda kutip. Angka nggak, dan bisa dihitung.
- f-string seperti `f"Hari ini: {dish}"` mengisi template dari toples-toplesmu.

BAB 3

Rename 50 foto dalam 2 detik

Kamera kasih nama foto kayak `IMG_0412.jpg`, `IMG_0413.jpg`, `IMG_0415.jpg`. Enam bulan kemudian kamu nyari foto sambal matah dan semua file kelihatan sama.

Rename satu-satu itu definisi kerjaan membosankan: klik, ketik, Enter, lima puluh kali. Yuk ajari komputer melakukannya.

Daftar berisi banyak hal

Sejauh ini satu toples cuma isi satu hal. **List** adalah toples yang isinya banyak hal, berurutan. Ditulis pakai kurung siku dan koma:

```
dishes = ["nasi goreng", "soto ayam", "es teler"]
```

Itu list berisi tiga string. Bisa juga list berisi 50 nama foto.

For loop

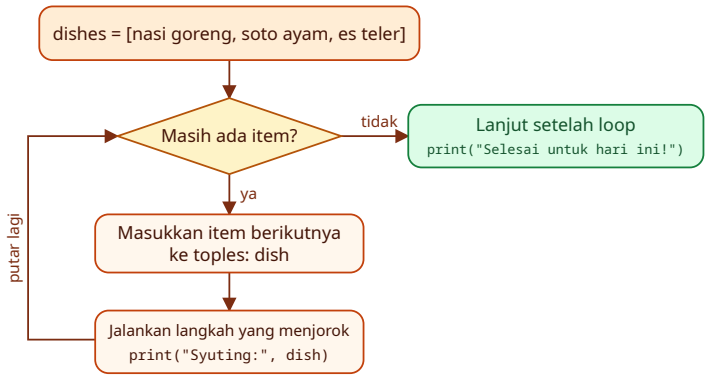
Loop mengulang langkah yang sama untuk setiap item di list. Kayak daftar belanja: untuk setiap barang di daftar, cari, masukkan ke keranjang.

```
dishes = ["nasi goreng", "soto ayam", "es teler"]

for dish in dishes:
    print("Syuting:", dish)

print("Selesai untuk hari ini!")
```

Output:

```
Syuting: nasi goreng
Syuting: soto ayam
Syuting: es teler
Selesai untuk hari ini!
```

Baca loop-nya keras-keras: "untuk setiap dish di dishes, tampilkan *Syuting:* dan nama masakannya." Setiap putaran, Python memasukkan item berikutnya ke toples `dish` lalu menjalankan baris yang menjorok ke dalam lagi.



Gambar 3.1: Cara kerja for loop: ambil item berikutnya, jalankan langkah yang menjorok, ulangi sampai list habis, lalu lanjut.

Dua detail penting:

- Baris `for` diakhiri **titik dua** `:`.

- Langkah yang diulang ditulis **menjorok** (digeser ke kanan 4 spasi). Dari situ Python tahu mana yang ada di dalam loop. `print("Selesai untuk hari ini!")` nggak menjorok, jadi cuma jalan sekali, setelah loop selesai.

Rename file: dry run dulu

Sekarang kerjaan aslinya. Taruh *salinan* beberapa foto di folder bernama `photos` di dalam folder `creator-scripts`. (Salinan ya. Kita hati-hati sama file asli.)

Kalau kode mengubah file, selalu lakukan **dry run** dulu: versi yang cuma *menampilkan* apa yang akan dilakukan, tanpa benar-benar melakukannya.

```
from pathlib import Path

folder = Path("photos")
photos = sorted(folder.glob("IMG_*.jpg"))

for number, photo in enumerate(photos, start=1):
    new_name = f"sambal-matah-{number:02}.jpg"
    print(photo.name, "->", new_name)
```

Output:

```
IMG_0412.jpg -> sambal-matah-01.jpg
IMG_0413.jpg -> sambal-matah-02.jpg
IMG_0415.jpg -> sambal-matah-03.jpg
IMG_0418.jpg -> sambal-matah-04.jpg
IMG_0421.jpg -> sambal-matah-05.jpg
```

Kata-kata baru, satu per satu:

- `from pathlib import Path` meminjam alat untuk mengurus file. Python punya banyak alat bawaan;

`import` mengambil satu dari raknya.

- `folder.glob("IMG_*.jpg")` mencari semua file yang namanya diawali `IMG_` dan diakhiri `.jpg`. Tanda `*` artinya "apa saja di sini".
- `sorted(...)` mengurutkannya, supaya foto 1 memang foto pertama yang kamu ambil.
- `enumerate(photos, start=1)` memberi nomor urut di samping tiap foto: 1, 2, 3...
- `{number:02}` menulis angkanya dua digit, jadi `1` jadi `01`. Supaya urutan file tetap benar kalau fotonya lebih dari sembilan.

Sekarang beneran

Daftarnya sudah benar. Sekarang ganti baris `print` dengan baris yang me-rename, lalu tampilkan isi folder di akhir supaya hasilnya kelihatan:

```
from pathlib import Path

folder = Path("photos")
photos = sorted(folder.glob("IMG_*.jpg"))

for number, photo in enumerate(photos, start=1):
    new_name = f"sambal-matah-{number:02}.jpg"
    photo.rename(folder / new_name)

print(sorted(p.name for p in folder.iterdir()))
```

Output:

```
['sambal-matah-01.jpg', 'sambal-matah-02.jpg',
↪  'sambal-matah-03.jpg', 'sambal-matah-04.jpg',
↪  'sambal-matah-05.jpg']
```

Lima file atau lima ratus, waktunya sama-sama dua detik.

> **Coba sendiri.** Bikin folder tes berisi beberapa salinan fotomu. Ubah script-nya supaya nama barunya pakai nama masakan hari ini dan tanggalnya, misalnya `es-teler-2026-10-26-01.jpg`. Dry run dulu. Rename cuma kalau daftar yang tampil sudah benar.

Rangkuman

- List menyimpan banyak hal berurutan: `["nasi goreng", "soto ayam"]`.
- `for` loop menjalankan langkah yang menjorok sekali untuk setiap item di list.
- Kalau kode mengubah file, dry run dulu (cuma menampilkan), baru jalankan beneran.

BAB 4

Post mana yang paling bagus?

Setiap kreator punya feeling soal apa yang jalan. “Minuman biasanya rame.” “Hari Minggu bagus buat posting.” Kadang feeling-nya benar. Kadang dia cuma ingat satu post yang pernah viral.

Di bab ini, biar angkamu sendiri yang menjawab.

Apa itu file CSV

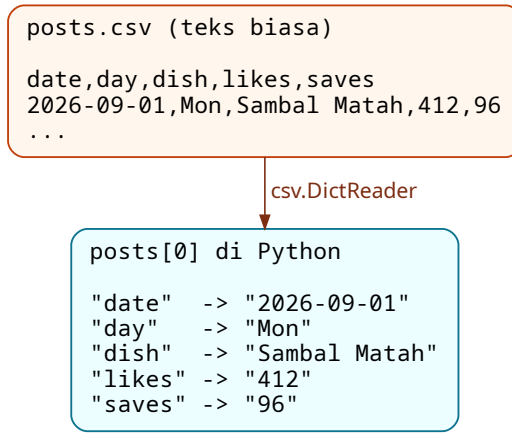
Kebanyakan platform bisa export insights, dan format yang umum adalah **CSV**. Singkatan dari *comma-separated values*, dan sebenarnya itu cuma spreadsheet yang disimpan sebagai teks biasa. Tiap baris adalah satu baris tabel, dan koma memisahkan kolomnya.

Ini contoh export kecil untuk delapan post, disimpan sebagai `posts.csv`:

```
date,day,dish,likes,saves
2026-09-01,Mon,Sambal Matah,412,96
2026-09-03,Wed,Soto Ayam,388,71
2026-09-05,Fri,Es Teler,951,240
2026-09-07,Sun,Nasi Goreng Kampung,603,158
2026-09-08,Mon,Tempe Mendoan,297,44
2026-09-10,Wed,Martabak Telur,520,133
```

```
2026-09-12,Fri,Es Cendol,1104,301
2026-09-14,Sun,Ayam Geprek,734,187
```

Kalau dibuka di aplikasi spreadsheet, kelihatannya tabel biasa. Baris pertama adalah **header**: nama-nama kolom.



Gambar 4.1: Satu baris CSV jadi dictionary di Python: setiap nama kolom jadi label yang bisa kamu cari.

Membacanya dengan Python

Python punya alat bawaan untuk file CSV. `csv.DictReader` membaca tiap baris jadi **dictionary**: kumpulan toples di dalam satu toples besar, masing-masing berlabel nama kolom.

```
import csv

with open("posts.csv", newline="") as file:
    posts = list(csv.DictReader(file))
```

```
print(len(posts), "post")
print(posts[0])
```

Output:

```
8 post
{'date': '2026-09-01', 'day': 'Mon', 'dish': 'Sambal
↪  Matah', 'likes': '412', 'saves': '96'}
```

`posts[0]` artinya "post pertama" (programmer mulai ngitung dari 0, maaf ya). Untuk ambil satu nilai, pakai labelnya: `posts[0]["dish"]` hasilnya `Sambal Matah`.

Perhatikan `'saves': '96'`. Tanda kutipnya berarti Python membaca angka itu sebagai *teks*. Supaya bisa dihitung, ubah jadi angka pakai `int()`, singkatan dari *integer* (bilangan bulat).

Urutkan berdasarkan saves

Saves itu sinyal bagus: orang menyimpan resep yang mau mereka masak nanti. Yuk cari tiga teratas.

```
import csv

with open("posts.csv", newline="") as file:
    posts = list(csv.DictReader(file))

def saves(post):
    return int(post["saves"])

posts.sort(key=saves, reverse=True)

for post in posts[:3]:
    print(post["dish"], "-", post["saves"], "saves")
```

Output:

```
Es Cendol - 301 saves
Es Teler - 240 saves
Ayam Geprek - 187 saves
```

`def` bikin fungsi kecil buatanmu sendiri. `saves(post)` menerima satu post dan mengembalikan jumlah saves-nya sebagai angka. `posts.sort(key=saves, reverse=True)` mengurutkan list berdasarkan angka itu, dari yang terbesar. `posts[:3]` artinya "tiga yang pertama".

Dua dari tiga teratas adalah minuman. Feeling-mu mungkin memang benar.

Rata-rata

Satu angka lagi yang berguna: rata-rata saves per post, supaya kamu tahu angka "normal"-nya berapa.

```
import csv

with open("posts.csv", newline="") as file:
    posts = list(csv.DictReader(file))

total = 0
for post in posts:
    total = total + int(post["saves"])

print("Rata-rata saves per post:", total / len(posts))
```

Output:

```
Rata-rata saves per post: 153.75
```

Post yang jauh di atas 154 saves layak dipelajari.

> **Coba sendiri.** Cari hari terbaik untuk posting. Kelompokkan saves berdasarkan kolom `day` dan tampilkan rata-rata tiap

hari.

Salah satu jawabannya:

```
import csv

with open("posts.csv", newline="") as file:
    posts = list(csv.DictReader(file))

by_day = {}
for post in posts:
    by_day.setdefault(post["day"],
↪   []).append(int(post["saves"]))

for day, numbers in by_day.items():
    print(day, sum(numbers) / len(numbers))
```

Output:

```
Mon 70.0
Wed 102.0
Fri 270.5
Sun 172.5
```

Dengan cuma dua post per hari, ini baru petunjuk, belum bukti. Terus kumpulkan datanya dan jalankan lagi bulan depan.

Rangkuman

- File CSV itu spreadsheet yang disimpan sebagai teks biasa, satu baris per baris tabel.
- `csv.DictReader` mengubah tiap baris jadi dictionary yang dibaca lewat nama kolom.
- Angka dari CSV datang sebagai teks. Pakai `int()` sebelum menghitung.

Langkah berikutnya

Sekarang kamu sudah kenal variabel, string, list, loop, file, dan dictionary. Itu fondasi yang beneran. Beberapa langkah berikutnya yang cocok untuk minggu seorang kreator:

- Bikin caption untuk seminggu penuh dari CSV kecil berisi daftar masakan.
- Resize foto untuk tiap platform pakai library gratis `Pillow`.
- Minta asisten AI menjelaskan kode yang belum kamu pahami, baris demi baris. Sekarang kamu sudah cukup paham untuk mengikutinya.