

Contents

Minggu 0: Siap-Siap Sebelum Prak □	1
Konsep Utama & Kegiatan Lab	2
Cheatsheet	3
Tugas	3
Konsep Utama & Kegiatan Lab	4
Live Coding: give them the many ways and ciri2 bahwa a line is a comment	5
Cheatsheet	5
Soal	5
Minggu 2: Typcasting dan Perhitungan Dasar	5
Konsep Utama & Kegiatan Lab	6
Live Coding: ASCII outputs, one path storybook?	7
Cheatsheet	7
Soal	8
Minggu 3: Pengambilan Keputusan (Selection) □	8
Konsep Utama & Kegiatan Lab	9
Live Coding:	9
Cheatsheet	9
Soal	10
Konsep Utama & Kegiatan Lab	11
Live Coding:	11
Cheatsheet	11
Soal	11
Konsep Utama & Kegiatan Lab	12
Tantangan 2 (Palindrome Checker): Buat program yang memeriksa apakah sebuah kata dibaca sama dari depan dan belakang. Ini akan memaksa mereka untuk menggunakan perulangan pada sebuah string.	13
Cheatsheet	13
Soal	13
Konsep Utama & Kegiatan Lab	14
Cheatsheet	15
Soal	15
Konsep Utama & Kegiatan Lab	16
Cheatsheet	17
Soal	17

pandoc 1sampe7.md -o final-booklet.pdf --toc --title="WOAW alpro" --pdf-engine=xelatex --highlight-style=pygments

Minggu 0: Siap-Siap Sebelum Prak □

Tujuan minggu ini adalah menyiapkan kamu untuk menulis kode dan memperkenalkan kamu pada alat-alat dasar yang akan digunakan di prak ALPRO

```
minggu-0/
├── CONFIDENTIAL_MATERI/          # Soal Materi mingguan. cuma ada di repo semuaan.
└── M0_test_hello_world.md
```

```

|   └─ M0_test_hello_world.pdf
|
|─ CONFIDENTIAL_PEMBAHASAN/      # Pembahasan. publish klo nilai udh keluar.
|   └─ kunci_T0_test_hello_world.md
|       |
|       └─ kunci_M0_test_hello_world.md
|
|─ KONTEN_TUTOR/
|   └─ tutor0_setup_lab.md      # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
|       └─ tutor0_setup_lab.pdf
|           |
|           └─ tes_awal0.pdf      # Tes_Awal mingguan.
|
|─ README_assets/
|   └─ git-flowchart.png
|       └─ diagram-venv.png
|
|─ README.md                    # penjelasan singkat prak mingguan
|
|─ tests/
|   └─ test_T0_hello_world.py    # Unit test supaya kamu bisa cek jawabanmu sendiri.
|
|─ TUGAS/
|   └─ T0_test_hello_world.py    # Lembar Tugas minggu 0. boleh kerjain di sini.

```

Konsep Utama & Kegiatan Lab

1. Mengatur Area Kerja

- **Mengapa:** Sebelum menulis kode, siswa perlu menyiapkan lingkungan kerja.
- **Bagaimana:** Tawarkan dua opsi: menginstal Python secara lokal atau menggunakan IDE online (seperti Replit).
- **Live Coding:** Bimbing siswa untuk menginstal atau membuka IDE, memastikan semuanya berfungsi.

2. Mengetahui GitHub dan juga Git

- **Mengapa:** Infrastruktur untuk menghasilkan dan berbagi kode
- **Bagaimana:** Fokus pada empat perintah inti: `git clone`, `git add`, `git commit`, dan `git push`.
- **Live Coding:** Bimbing siswa untuk mengkloning repositori template mingguan, membuat perubahan kecil pada file, dan meng `git push` kembali ke GitHub.

3. Menjalankan Program Pertama Anda

- **Mengapa:** Ini adalah momen puncak, di mana mereka berhasil membuat dan menjalankan program pertama.
 - **Bagaimana:** Ajarkan sintaks dasar untuk mencetak teks.
 - **Live Coding:** Minta siswa untuk menulis dan menjalankan program “Halo, Dunia!”.
-

Cheatsheet

Screenshot Kegunaan Area Kerja:

- (notebook? vscode extension? thonny? notepad? notepad++ ?) jujur bingung ajarin yg mana.

Perintah Git Dasar:

- `git clone [URL]`
- `git add .`
- `git commit -m "pesan"`
- `git push`

Sintaks Python Dasar:

- `print("teks")`
-

Tugas

- **Soal:** Buat program Python yang mencetak kalimat “Halo, Dunia! Saya [Nama Anda].” di terminal.
- **Kegiatan Tes Materi (Mandiri):** Berikan siswa file kode dengan kesalahan sederhana dan minta mereka memperbaikinya. Ini mendorong mereka untuk belajar cara membaca pesan kesalahan. **### Minggu 1: Input Output Sempel** □

Tujuan minggu ini adalah mengajarkan siswa cara menyimpan dan memanipulasi informasi dalam program mereka. Mereka akan belajar bagaimana program bisa menjadi interaktif dengan menerima masukan dari pengguna.

INI BLOM DIGANTI KABEH YO ISINYA. BARU README

minggu-0/

```
├─ CONFIDENTIAL_MATERI/          # Soal Materi mingguan. cuma ada di repo semuanya.
|   ├── M0_test_hello_world.md
|   └─ M0_test_hello_world.pdf
|
└─ CONFIDENTIAL_PEMBAHASAN/      # Pembahasan. publish klo nilai udh keluar.
```

```

|   ├── kunci_T0_test_hello_world.md
|   |
|   └── kunci_M0_test_hello_world.md
|
|── KONTEN_TUTOR/
|   ├── tutor0_setup_lab.md      # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
|   ├── tutor0_setup_lab.pdf
|   |
|   └── tes_awal0.pdf            # Tes_Awal mingguan.
|
|── README_assets/
|   ├── git-flowchart.png
|   └── diagram-venv.png
|
|── README.md                    # penjelasan singkat prak mingguan
|
|── tests/
|   └── test_T0_hello_world.py    # Unit test supaya kamu bisa cek jawabanmu sendiri.
|
|── TUGAS/
|   └── T0_test_hello_world.py    # Lembar Tugas minggu 0. boleh kerjain di sini.

```

Konsep Utama & Kegiatan Lab

1. Tipe Data dan Variabel

- **Mengapa:** Jelaskan bahwa setiap informasi punya jenisnya sendiri (teks, angka, True/False) dan program perlu tahu perbedaannya.
- **Bagaimana:** Perkenalkan variabel sebagai “kotak berlabel” untuk menyimpan data. Ajarkan cara mendeklarasikan variabel untuk tiga tipe data utama: string (teks), integer (bilangan bulat), dan float (bilangan desimal). tapi python aslinya gaperlu dideklarasikan ???
- **Live Coding:** Tunjukkan cara membuat dan mencetak variabel dari setiap tipe.

2. Masukan Pengguna (input())

- **Mengapa:** Masukan pengguna memungkinkan program berinteraksi dengan orang lain, tidak hanya berjalan otomatis.
- **Bagaimana:** Perkenalkan fungsi input() sebagai cara untuk mendapatkan informasi dari pengguna. Jelaskan bahwa input() selalu mengembalikan string, ini adalah poin penting yang akan mereka hadapi.
- **Live Coding:** playing around with data types. linear storybook. then trying to see what data type.

3. Membuat Komentar di Skrip

- **Mengapa:** perlu comments.
- **Bagaimana:** Perkenalkan fungsi input() sebagai cara untuk mendapatkan informasi dari pengguna. Jelaskan bahwa input() selalu mengembalikan string, ini adalah poin penting yang akan mereka hadapi.
-

Live Coding: give them the many ways and ciri2 bahwa a line is a comment

Cheatsheet

Tipe Data Utama:

- str (string): 'teks'
- int (integer): 100
- float (float): 3.14

Sintaks Python Dasar:

- input("tulisannn:")
 - print(nilai) (expand on this)
 - comments syntax
-

Soal

- **Tugas:**
 - Buat program yang meminta nama depan, nama belakang, dan tahun lahir pengguna. Kemudian, cetak kalimat yang menyapa mereka dan tampilkan data yang mereka masukkan. minta mereka kasih comment.
 - Buat Program kalkulasi matematika simpel?
 - **Materi:** Debugging kode. analisa program. intinya nguji mereka bisa baca & ngedebug kode simpel ga.

Minggu 2: Typecasting dan Perhitungan Dasar

Tujuan minggu ini adalah mengajarkan siswa cara melakukan perhitungan dengan data yang diterima dari pengguna.

INI BLOM DIGANTI KABEH YO ISINYA. BARU README

minggu-0/

|— CONFIDENTIAL_MATERI/ # Soal Materi mingguan. cuma ada di repo semuaan.

```

|   ├── M0_test_hello_world.md
|   └── M0_test_hello_world.pdf
|
|   ├── CONFIDENTIAL_PEMBAHASAN/          # Pembahasan. publish klo nilai udh keluar.
|   │   ├── kunci_T0_test_hello_world.md
|   │   │   |
|   │   └── kunci_M0_test_hello_world.md
|   |
|   ├── KONTEN_TUTOR/
|   │   ├── tutor0_setup_lab.md          # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
|   │   ├── tutor0_setup_lab.pdf
|   │   │   |
|   │   └── tes_awal0.pdf                # Tes_Awal mingguan.
|   |
|   ├── README_assets/
|   │   ├── git-flowchart.png
|   │   └── diagram-venv.png
|   |
|   ├── README.md                        # penjelasan singkat prak mingguan
|   |
|   ├── tests/
|   │   ├── test_T0_hello_world.py      # Unit test supaya kamu bisa cek jawabanmu sendiri.
|   |
|   └── TUGAS/
|       ├── T0_test_hello_world.py      # Lembar Tugas minggu 0. boleh kerjain di sini.

```

Konsep Utama & Kegiatan Lab

1. Typecasting sebagai Solusi

- **Mengapa:** Ingatkan siswa tentang masalah di mana `input()` selalu mengembalikan string. Perlihatkan lagi bagaimana Python akan menggabungkan teks ("`5`" + "`5`" = "`55`") daripada menjumlahkan angka. Typecasting adalah solusi untuk mengubah data dari satu jenis ke jenis lain.
- **Bagaimana:** Perkenalkan fungsi `int()` dan `float()`. Gunakan analogi mesin konverter data yang mengubah teks angka menjadi angka asli, sehingga dapat digunakan untuk perhitungan.
- **Live Coding:** Tunjukkan cara typecasting lewat interpreter shell dan di program.

2. Melakukan Perhitungan

- **Mengapa:** Ini adalah inti dari sebagian besar program. Dengan menguasai perhitungan, mereka dapat membuat program yang melakukan tugas yang lebih berguna

- **Bagaimana:** Setelah mengonversi data, ajarkan mereka cara menggunakan operator aritmatika (+, -, *, /, mod, ^)
- **Live Coding:** playing around with data types. python interpreter shell.

3. Membuat Pesan yang Bermakna

- **Mengapa:** Program yang baik tidak hanya memberikan jawaban, tetapi juga menyajikannya dengan jelas.
- **Bagaimana:** Ajarkan cara menggabungkan string dan variabel yang sudah dihitung untuk membuat pesan keluaran yang informatif.
-

Live Coding: ASCII outputs, one path storybook?

Cheatsheet

Fungsi Typecasting:

- `int(nilai)`: Mengubah ke bilangan bulat.
- `float(nilai)`: Mengubah ke bilangan desimal.
- `str(nilai)`: Mengubah ke teks.

Operator Aritmatika:

- + (penjumlahan)
- - (pengurangan)
- * (perkalian)
- / (pembagian)

Contoh Kode:

```
# Menerima masukan dari pengguna
panjang_str = input("Masukkan panjang: ")
lebar_str = input("Masukkan lebar: ")

# Mengubah tipe data
panjang = float(panjang_str)
lebar = float(lebar_str)

# Melakukan perhitungan
luas = panjang * lebar

# Mencetak hasilnya
print(f"Luas persegi panjang adalah: {luas}")
```

Soal

- **Tugas:**
 - Buat program yang meminta panjang sisi persegi, lalu hitung keliling dan luasnya. Cetak kedua hasilnya dalam kalimat yang berbeda.
- **Pluspoint:**
 - Buat ASCII art yg ke-print sama hasil cetakan kalkulasinya
- **Materi:**
 - Debugging type error code. Berikan potongan kode dengan kesalahan TypeError dan minta siswa untuk memperbaikinya menggunakan typecasting yang benar.
 -

Minggu 3: Pengambilan Keputusan (Selection) □

Tujuan minggu ini adalah mengajarkan siswa cara membuat program mengambil jalur yang berbeda berdasarkan suatu kondisi. Ini adalah salah satu konsep paling fundamental untuk menciptakan program yang dinamis dan “pintar.”

INI BLOM DIGANTI KABEH YO ISINYA. BARU README

```
minggu-0/
├── CONFIDENTIAL_MATERI/          # Soal Materi mingguan. cuma ada di repo semuanya.
│   ├── M0_test_hello_world.md
│   └── M0_test_hello_world.pdf
│
├── CONFIDENTIAL_PEMBAHASAN/      # Pembahasan. publish klo nilai udh keluar.
│   ├── kunci_T0_test_hello_world.md
│   |
│   └── kunci_M0_test_hello_world.md
│
├── KONTEN_TUTOR/
│   ├── tutor0_setup_lab.md       # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
│   ├── tutor0_setup_lab.pdf
│   |
│   └── tes_awal0.pdf             # Tes_Awal mingguan.
│
├── README_assets/
│   ├── git-flowchart.png
│   └── diagram-venv.png
│
├── README.md                     # penjelasan singkat prak mingguan
│
└── tests/
    └── test_T0_hello_world.py    # Unit test supaya kamu bisa cek jawabanmu sendiri.
```



```
|
└─ TUGAS/
    └─ T0_test_hello_world.py      # Lembar Tugas minggu 0. boleh kerjain di sini.
```

Konsep Utama & Kegiatan Lab

1. Mental Model: Pengambilan Keputusan

- **Mengapa:** Jelaskan bahwa program sering kali harus membuat pilihan. Gunakan contoh sehari-hari seperti “jika saya lapar, maka saya akan makan.” Ini menciptakan kebutuhan yang jelas akan pernyataan kondisional.
- **Bagaimana:** Gunakan diagram alur (flowchart) untuk menunjukkan alur logika. Gunakan simbol berlian untuk mewakili sebuah keputusan, dengan dua jalur keluarannya (True atau False).

2. Pernyataan if, elif, dan else

- **Mengapa:** Ini adalah cara Python untuk menerjemahkan logika pengambilan keputusan
- **Bagaimana:** Ajarkan sintaks if untuk satu kondisi, lalu tambahkan else untuk memberikan opsi cadangan. Terakhir, kenalkan elif (singkatan dari “else if”) untuk menciptakan rantai kondisi yang lebih panjang dan efisien (seperti “switch-case” di bahasa lain)
- **Live Coding:** make ur own adventure storybook.

3. Operator Logika (AND, OR, NOT)

- **Mengapa:** Operator logika memungkinkan program untuk membuat keputusan yang lebih kompleks.
- **Bagaimana:** Jelaskan cara menggunakan and (semua kondisi harus benar) dan or (salah satu kondisi harus benar) untuk menggabungkan dua atau lebih kondisi.
-

Live Coding:

Cheatsheet

Sintaks Kondisional: * if condition: * elif condition: * else:

Operator Perbandingan: * == (sama dengan) * != (tidak sama dengan) * > (lebih besar dari), < (lebih kecil dari) * >= (lebih besar dari atau sama dengan) * <= (lebih kecil dari atau sama dengan)

Operator Logika: * and * or * not

Soal

- **Tugas:**

- Buat program classifier biasa.
- Artiiin flowchart ke kode.

- **Materi:**

- Debugging logical error. Berikan skrip dengan logika yg salah tapi programnya bisa jalan.
- Soal cerita mirip logika kombinasional hahahaha. ### Minggu 4: Pengulangan (Iteration) □

Tujuan minggu ini adalah mengajarkan siswa cara membuat program yang dapat mengotomatiskan tugas-tugas berulang. Siswa akan belajar untuk membedakan dan memilih antara perulangan for dan while berdasarkan kebutuhan spesifik.

INI BLOM DIGANTI KABEH YO ISINYA. BARU README

minggu-0/

```
├─ CONFIDENTIAL_MATERI/          # Soal Materi mingguan. cuma ada di repo semuaan.
|   ├── M0_test_hello_world.md
|   └─ M0_test_hello_world.pdf
|
├─ CONFIDENTIAL_PEMBAHASAN/      # Pembahasan. publish klo nilai udh keluar.
|   ├── kunci_T0_test_hello_world.md
|   |
|   └─ kunci_M0_test_hello_world.md
|
├─ KONTEN_TUTOR/
|   ├── tutor0_setup_lab.md      # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
|   ├── tutor0_setup_lab.pdf
|   |
|   └─ tes_awal0.pdf            # Tes_Awal mingguan.
|
├─ README_assets/
|   ├── git-flowchart.png
|   └─ diagram-venv.png
|
├─ README.md                    # penjelasan singkat prak mingguan
|
├─ tests/
|   └─ test_T0_hello_world.py    # Unit test supaya kamu bisa cek jawabanmu sendiri.
|
└─ TUGAS/
    └─ T0_test_hello_world.py    # Lembar Tugas minggu 0. boleh kerjain di sini.
```

Konsep Utama & Kegiatan Lab

1. Mental Model: Pengulangan

- **Mengapa:** Jelaskan bahwa ada banyak tugas yang berulang-ulang dalam pemrograman, seperti mencetak setiap item dalam daftar, atau menjalankan kode sampai kondisi tertentu terpenuhi. Menggunakan baris kode yang sama berulang kali tidak efisien dan rentan kesalahan.
- **Bagaimana:** Jelaskan bahwa ada dua jenis utama perulangan: perulangan `for` yang digunakan saat Anda tahu berapa kali perulangan perlu berjalan (misalnya, untuk setiap item dalam daftar), dan perulangan `while` yang digunakan saat Anda ingin perulangan berjalan sampai kondisi tertentu tidak lagi terpenuhi.

2. Repetisi `for`

- **Mengapa:** alat yang sempurna untuk mengulangi setiap item dalam sebuah urutan. Ini sangat berguna untuk memproses daftar, string, atau range angka.
- **Bagaimana:** Ajarkan sintaks `for`
- **Live Coding:** show database printing?

3. Repetisi `while`

- **Mengapa:** Ajarkan sintaks `while`
- **Bagaimana:** Buat game tebak-tebakan sederhana di mana perulangan berlanjut hingga pengguna memasukkan tebakan yang benar
-

Live Coding:

Cheatsheet

Sintaks Looping: `* for item in sequence: * while condition:`

Soal

- **Tugas:**
-
- Artikan flowchart ke kode.
- **Materi:**

- Debugging logical error. Berikan skrip dengan logika yg salah tapi programnya bisa jalan.
- Soal cerita mirip logika kombinasional hahahaha. ### Minggu 5: (Review Buffer) Pemecahan Masalah & Berpikir Algoritmik □

Tujuan minggu ini adalah membantu siswa mengembangkan keterampilan inti dalam pemrograman: kemampuan untuk memecah masalah besar menjadi bagian-bagian yang lebih kecil dan menggunakan alat yang mereka miliki untuk menyelesaikannya.

INI BLOM DIGANTI KABEH YO ISINYA. BARU README

```
minggu-0/
├── CONFIDENTIAL_MATERI/          # Soal Materi mingguan. cuma ada di repo semuaan.
│   ├── M0_test_hello_world.md
│   └── M0_test_hello_world.pdf
│
├── CONFIDENTIAL_PEMBAHASAN/      # Pembahasan. publish klo nilai udh keluar.
│   ├── kunci_T0_test_hello_world.md
│   |
│   └── kunci_M0_test_hello_world.md
│
├── KONTEN_TUTOR/
│   ├── tutor0_setup_lab.md       # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
│   ├── tutor0_setup_lab.pdf
│   |
│   └── tes_awal0.pdf             # Tes_Awal mingguan.
│
├── README_assets/
│   ├── git-flowchart.png
│   └── diagram-venv.png
│
├── README.md                     # penjelasan singkat prak mingguan
│
├── tests/
│   └── test_T0_hello_world.py    # Unit test supaya kamu bisa cek jawabanmu sendiri.
│
└── TUGAS/
    └── T0_test_hello_world.py    # Lembar Tugas minggu 0. boleh kerjain di sini.
```

Konsep Utama & Kegiatan Lab

1. Kerangka Pemecahan Masalah

- **Mengapa:** Ajarkan siswa cara berpikir sistematis, bukan hanya menulis kode. Ini adalah keterampilan yang bisa diterapkan pada masalah apa pun.

- **Bagaimana:** Kenalkan kerangka kerja 4 langkah:
 1. **Pahami (Decomposition):** Pecah masalah menjadi bagian-bagian kecil.
 2. **Rencanakan (Pattern Recognition):** Cari pola yang bisa diulang.
 3. **Kode (Abstraction):** Tulis kode yang rapi.
 4. **Uji (Algorithm Design):** Pastikan kode berfungsi untuk semua kasus.

2. Sesi Pemecahan Masalah Langsung

- **Mengapa:** Demonstrasi langsung adalah cara terbaik untuk menunjukkan kerangka kerja ini beraksi.
- **Bagaimana:** Pimpin siswa melalui dua atau tiga tantangan coding klasik yang menggabungkan perulangan dan kondisional.
 - **Tantangan 1 (FizzBuzz):** Sebuah program yang mencetak angka dari 1 hingga 100, tetapi mengganti kelipatan 3 dengan “Fizz”, kelipatan 5 dengan “Buzz”, dan kelipatan 3 dan 5 dengan “FizzBuzz”.

Tantangan 2 (Palindrome Checker): Buat program yang memeriksa apakah sebuah kata dibaca sama dari depan dan belakang. Ini akan memaksa mereka untuk menggunakan perulangan pada sebuah string.

Cheatsheet

Kerangka Pemecahan Masalah:

1. Pahami masalahnya.
2. Rencanakan solusinya (pseudocode/flowchart).
3. Tulis kode.
4. Uji kode dengan berbagai masukan.

Soal

- **Tugas:**
-
- Artiin flowchart ke kode.
- **Materi:**
- FizzBuzz ### Minggu 6: Struktur Data (Lists & Tuples) □

Tujuan minggu ini adalah mengajarkan siswa cara menyimpan dan mengelola banyak item dalam satu wadah. Ini adalah langkah penting untuk menangani data skala besar.

INI BLOM DIGANTI KABEH YO ISINYA. BARU README

```

minggu-0/
├── CONFIDENTIAL_MATERI/          # Soal Materi mingguan. cuma ada di repo semuanya.
│   ├── M0_test_hello_world.md
│   └── M0_test_hello_world.pdf
│
├── CONFIDENTIAL_PEMBAHASAN/      # Pembahasan. publish klo nilai udh keluar.
│   ├── kunci_T0_test_hello_world.md
│   │
│   └── kunci_M0_test_hello_world.md
│
├── KONTEN_TUTOR/
│   ├── tutor0_setup_lab.md      # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
│   ├── tutor0_setup_lab.pdf
│   │
│   └── tes_awal0.pdf            # Tes_Awal mingguan.
│
├── README_assets/
│   ├── git-flowchart.png
│   └── diagram-venv.png
│
├── README.md                    # penjelasan singkat prak mingguan
│
├── tests/
│   └── test_T0_hello_world.py    # Unit test supaya kamu bisa cek jawabanmu sendiri.
│
└── TUGAS/
    └── T0_test_hello_world.py    # Lembar Tugas minggu 0. boleh kerjain di sini.

```

Konsep Utama & Kegiatan Lab

1. Lists: Koleksi yang Fleksibel

- **Mengapa:** Jelaskan bahwa list adalah cara Python untuk menyimpan daftar item yang dapat diubah dan diurutkan. Pikirkan sebagai **daftar belanjaan** — Anda bisa menambah atau menghapus barang sesuka hati.
- **Bagaimana:** Tunjukkan cara membuat list dengan tanda kurung siku `[]` dan metode manipulasi paling relevan:
 - `.append()` untuk menambahkan item ke akhir daftar.
 - `.remove()` untuk menghapus item berdasarkan nilai.
 - `.pop()` untuk menghapus item berdasarkan indeks.
 - `.sort()` untuk mengurutkan item.
- **Live Coding:** Tunjukkan cara membuat list dan memanipulasinya, seperti menambahkan nama baru atau menghapus nama yang sudah ada

2. Tuples: Koleksi yang Tetap

- **Mengapa:** tuple juga menyimpan daftar item, tetapi bersifat **immutable** (tidak dapat diubah). Gunakan ini saat Anda memiliki data yang tidak boleh berubah.
- **Bagaimana:** Tunjukkan cara membuat tuple dengan tanda kurung biasa () dan demonstrasikan bahwa Anda **tidak dapat** memanipulasinya.

3. Kapan Menggunakan yang Mana?

- Berikan panduan sederhana: Gunakan list saat Anda perlu mengubah data, dan gunakan tuple saat data Anda harus tetap konstan.

Cheatsheet

Sintaks Lists: *daftar = ["apel", "jeruk"] *daftar.append("mangga") *daftar.pop(0)

Sintaks Tuples: *nilai_tes = (85, 92, 78)

Fungsi Penting: *len(koleksi): Mengembalikan jumlah item.

Soal

- **Tugas:**
 - Buat program database.
- **Materi:**
 - Buat program yang melacak skor harian. Minta pengguna untuk memasukkan skor selama lima hari berturut-turut. Simpan skor tersebut dalam sebuah list, hitung rata-rata, lalu cetak skor tertinggi dan terendah. ### Minggu 7: Fungsi & Prosedur □

Tujuan minggu ini adalah mengajarkan siswa cara menulis kode yang rapi, terorganisir, dan dapat digunakan kembali. Ini adalah langkah besar menuju penulisan program yang lebih kompleks dan mudah dikelola.

INI BLOM DIGANTI KABEH YO ISINYA. BARU README

```
minggu-0/
├── CONFIDENTIAL_MATERI/          # Soal Materi mingguan. cuma ada di repo semuaan.
│   ├── M0_test_hello_world.md
│   └── M0_test_hello_world.pdf
│
├── CONFIDENTIAL_PEMBAHASAN/      # Pembahasan. publish klo nilai udh keluar.
│   ├── kunci_T0_test_hello_world.md
│   │
│   └── kunci_M0_test_hello_world.md
│
```

```

├─ KONTEN_TUTOR/
│  ├─ tutor0_setup_lab.md      # Buku Prak. isinya link pretest + tutor + live coding + instruksi tugas.
│  └─ tutor0_setup_lab.pdf
│
│  └─ tes_awal0.pdf           # Tes_Awal mingguan.
│
├─ README_assets/
│  └─ git-flowchart.png
│  └─ diagram-venv.png
│
├─ README.md                  # penjelasan singkat prak mingguan
│
├─ tests/
│  └─ test_T0_hello_world.py  # Unit test supaya kamu bisa cek jawabanmu sendiri.
│
└─ TUGAS/
   └─ T0_test_hello_world.py  # Lembar Tugas minggu 0. boleh kerjain di sini.

```

Konsep Utama & Kegiatan Lab

1. Fungsi vs. Prosedur

- **Mengapa:** Jelaskan bahwa membedakan antara “melakukan sesuatu” dan “mengembalikan nilai” adalah konsep inti pemrograman. Sebuah **prosedur** melakukan tugas tanpa mengembalikan nilai (seperti `print()`), sementara sebuah **fungsi** menghitung atau memproses sesuatu dan **mengembalikan hasilnya** (seperti `len()` atau `max()`).
- **Bagaimana:** Tunjukkan kedua konsep secara berdampingan.
 - **Prosedur:** Buat kode yang hanya mencetak pesan, tanpa `return`.
 - **Fungsi:** Buat kode yang menjumlahkan dua angka dan **mengembalikan** hasilnya menggunakan `return`.

2. Parameter dan Argumen

- **Mengapa:** Fungsi dan prosedur membutuhkan informasi untuk bekerja. Informasi ini disebut **parameter**.
- **Bagaimana:** Jelaskan bahwa parameter adalah *input* yang diberikan saat fungsi dipanggil. Tunjukkan bagaimana membuat fungsi yang menerima parameter.

3. Menerapkan Fungsi

- **Live Coding:** Ambil tugas yang sudah mereka kerjakan sebelumnya (misalnya, kalkulator luas dari Minggu 2) dan tunjukkan cara mengubahnya menjadi sebuah fungsi yang dapat digunakan kembali. Ini menunjukkan manfaat nyata dari modularitas.
- **Contoh:**


```
# Prosedur: Tidak mengembalikan nilai
def sapa_pengguna(nama):
    print(f"Halo, {nama}!")

# Fungsi: Mengembalikan nilai
def hitung_luas_persegi(sisi):
    luas = sisi * sisi
    return luas
```

Cheatsheet

Sintaks Fungsi/Prosedur:

- `def nama_fungsi(parameter):`
- Gunakan `return` untuk mengembalikan nilai.

Contoh Praktis:

- **Prosedur:** `sapa_pengguna("Budi")` □ hanya mencetak di layar.
 - **Fungsi:** `luas_kamar = hitung_luas_persegi(5)` □ menyimpan nilai yang dikembalikan.
-

Soal

- **Tugas:**
- refactor?
- **Materi:**
- yg dibikin, jadiin function? procedure?