

# **Learning Progress Review III**

11-16 Oktober 2025 | Group 5, Seru!

# TABLE OF CONTENTS

## WEEK III

01 →   
**Git and Version  
Control  
System**

02 →   
**Exploring SQL  
& GitHub**

03 →   
**Basic to  
Programming**



# **01 GIT AND VERSION CONTROL SYSTEM**

# VERSION CONTROL SYSTEM

## What is a version control system?

A **Version Control System** (VCS) is a tool used in software development and collaborative projects to track and manage changes to source code. It allows developers to:

- Record and track every update to the codebase
- Collaborate on code without overwriting each other's work
- Revert to earlier states of the project if needed
- Maintain a detailed and structured history of the project's evolution

## Benefits of Version Control Software

- History and traceability
- Collaboration and parallel development
- Quality and faster delivery
- Risk management and audit trails

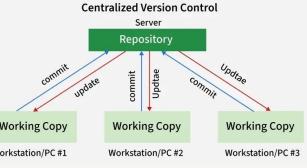
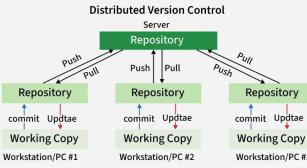
## Core Concepts in Version Control System

- Repository
- Commit and commit Message
- Branching and Merging
- Tag and Releases
- Pull request and code review



# Types of Version Control Systems

Ada 3 Tipe Utama Version Control System

Tipe VCS	Deskripsi	Contoh	Keterangan Gambar Model VCS
<b>Local Version Control</b>	Semua versi disimpan di satu komputer lokal.	RCS (Revision Control System)	
<b>Centralized Version Control (CVCS)</b>	Semua versi disimpan di satu server pusat; pengguna mengambil dan mengunggah perubahan ke sana.	CVS, Subversion (SVN), Perforce	
<b>Distributed Version Control (DVCS)</b>	Setiap pengguna memiliki salinan penuh dari repositori, termasuk seluruh riwayatnya.	Git, Mercurial, Bazaar	

Source : <https://www.geeksforgeeks.org/git/version-control-systems/>

## Popular VCS Tools in 2025



Source : <https://trio.dev/version-control-systems/>



## What is it?

GitHub is a **web-based company** that provides a service for storing **Git repositories** and includes a **Version Control System**. It is a social networking site for software developers, making it popular for **open-source** projects.



## Business model

GitHub's business model is based on the **freemium model**. While it provides a robust free plan, it generates revenue through several paid services: **Paid Plans (SaaS)**, **GitHub Marketplace**, and **Additional Service**.



## Key benefits

- **Repository Storage:** Use it to store your code.
- **Collaboration:** It's a platform and tool for collaborating on projects with others.
- **Networking:** It includes a profile page where you can connect and communicate with other programmers.
- **Version Control:** It helps you save and track all the changes made to your repository, which is essential for managing project updates.



*Git is an open-source, distributed version control tool that helps track changes made in computer files/ codes.*

*It facilitates collaboration and coordination amongst teams of developers while maintaining data integrity.*

## Instalasi

Git :

<https://git-scm.com/downloads>

VS Code :

<https://code.visualstudio.com/download>

## The main components



- Working Directory
- Staging Area (Index)
- .git Directory (Local Repository)



## Git glossary

- 📁 Repository: Tempat penyimpanan code dan file.
- 📄 Clone: Menyalin repository dari server ke komputer.
- ✚ Add: Menambahkan perubahan ke staging area.
- 📁 Commit: File siap disimpan.
- ⬆ Push: Mengupload file ke repository.
- ⬇ Pull: Mendownload content dari repository ke komputer.

# KONFIGURASI GIT AWAL

## Persiapan Awal

- Setelah menginstal Git, lakukan konfigurasi awal sebelum digunakan.

## Perintah Konfigurasi

- `git config --global user.name "My Name"`
- `git config --global user.email myEmail@example.com`

## Persiapan Awal

- Setelah menginstal Git, lakukan konfigurasi awal sebelum digunakan.

## Mengecek Konfigurasi

- `git config -list`

\* Gunakan username & email yang sama dengan akun GitHub / GitLab / Bitbucket agar mudah diintegrasikan.

# MEMBUAT REPOSITORY

1 Access this link and make your account : <https://github.com/>

2 Click "New repository" 

3 Name your repo and click "Create repository" 

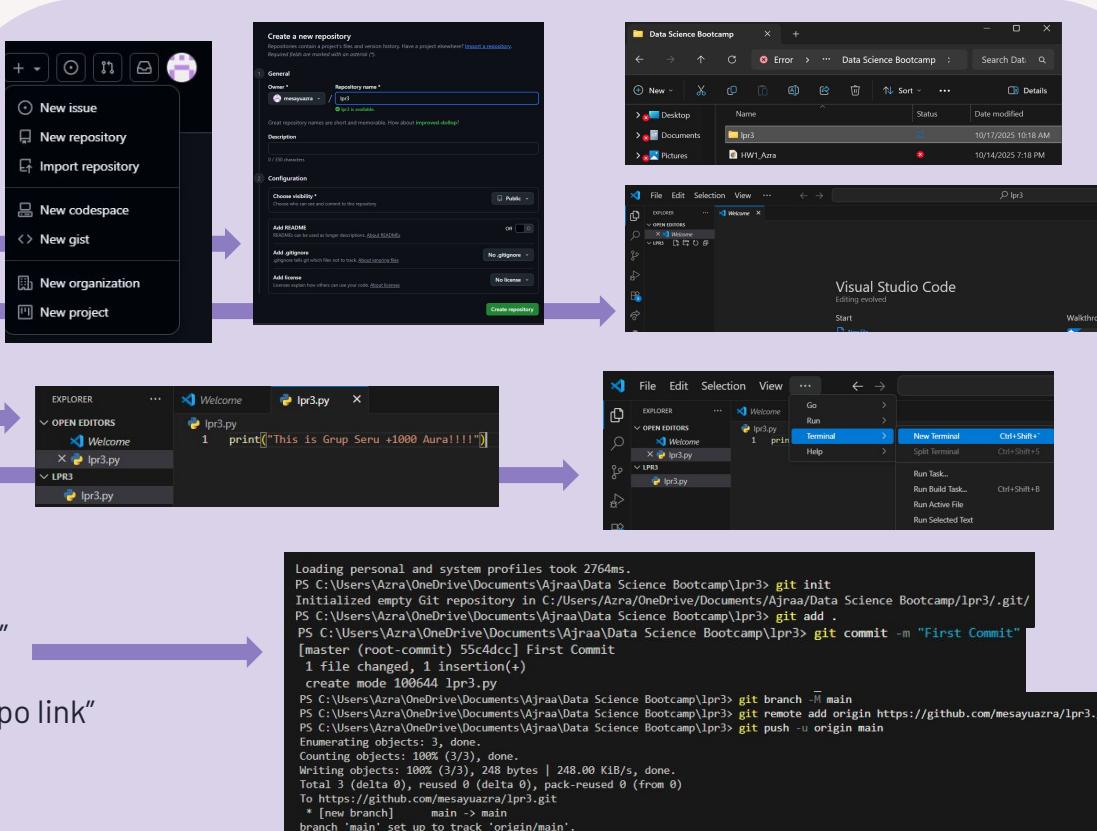
4 Make a folder and open it in VSCode 

5 Create a new file 

6 Open "New Terminal" 

7 Run this commands

- git init
- git add .
- git commit -m "insert your message"
- git branch -M main
- git remote add origin "insert your repo link"
- git push -u origin



# Your Repository Has Been Created



A screenshot of a GitHub repository page. The repository name is `lpr3`, owned by `mesayuazra`. The page shows a single commit from `mesayuazra` titled "First Commit" made 3 hours ago. The commit includes a file named `lpr3.py`. Below the commit, there is a section for the `README` file, which currently contains a placeholder icon and the text "Add a README". A green button labeled "Add a README" is visible at the bottom of this section. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.

# Penyimpanan Perubahan



## Step 1

git add .



## Step 2

git status



## Step 3

git commit -m "Nama  
File.txt"



## Step 4

git status



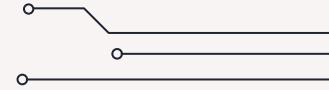
## Step 5

git push origin master



## Step 6

Check in github



# Merge Perubahan (1/2)

**Merge** artinya menggabungkan perubahan kode dari satu cabang (branch) ke cabang lainnya.

## ■ Tahapan Proses Merge di GitHub (Versi Sederhana)

- 1. Buat Cabang (Branch) Baru** → Tujuannya: agar kamu bisa bekerja tanpa mengganggu kode utama.
- 2. Lakukan Perubahan di Branch-mu** → Ubah, tambah, atau hapus kode sesuai kebutuhan. Setelah selesai, **commit**
- 3. Upload ke GitHub (Push)** → Kirim perubahannya dari komputer lokal ke repositori GitHub.
- 4. Buat Pull Request (PR)** → Pull Request = permintaan untuk menggabungkan hasil kerjamu ke branch utama ([main](#)).
- 5. Review & Diskusi** → Tim lain akan membaca perubahannya. Kalau ada masalah, kamu bisa perbaiki di branch yang sama.
- 6. Merge ke Branch Utama** → Jika sudah disetujui, klik tombol “**Merge Pull Request**” di GitHub. Semua perubahannya akan otomatis masuk ke [main](#).



# Ilustrasi Melakukan Merge Perubahan (2/2)

## 1. Main Branch = Buku Asli

Ada satu buku utama yang berisi semua tulisan yang



## 2. Branch = Fotokopi Buku

Kamu memfotokopi buku utama untuk menulis bab baru



## 3. Commit = Menandai Halaman yang Diubah

Setiap kali kamu menulis perubahan, kamu memberi catatan kecil



## 4. Push = Mengirim Fotokopimu ke Perpustakaan

Kirim fotokopimu dari rumah ke perpustakaan



## 8. Delete Branch = Buang Fotokopi Lama

Setelah selesai, buang fotokopi



## 7. Merge = Naskahmu Masuk ke Buku Utama

Naskahmu digabungkan ke dalam buku utama



## 6. Review = Editor Membaca dan Memberi Masukan

Editor membaca dan memberi komentar pada naskahmu



## 5. Pull Request = Mengajukan Naskah ke Editor

Buat perminian ke editor untuk menggabungkan tulisanmu





# Tips Sederhana

- Selalu buat **branch baru** untuk setiap fitur atau perbaikan.
- Jangan langsung ubah di **main**.
- Selalu **buat pull request** agar perubahannya bisa direview.
- Gunakan **pesan commit** yang jelas, agar orang lain tahu apa yang kamu ubah.

02

# EXPLORING SQL & GITHUB

# Codespaces dan SQL Tools

## Codespaces

GitHub **Codespaces** adalah **lingkungan pengembangan terkelola sepenuhnya yang dihosting di cloud**. Ini mengatasi masalah **perbedaan lingkungan** antar anggota tim data scientist yang menggunakan OS berbeda (macOS, Windows, Linux). Dengan Codespaces, setiap orang hanya butuh **browser dan akses internet** untuk mengakses proyek, yang menjamin **lingkungan kerja yang konsisten dan identik** bagi semua, **mempercepat** kolaborasi tim.

## How to use codespaces?

Untuk menjalankan Codespaces dari proyek yang sudah ada (milik orang lain):

- Pastikan Anda sudah login ke GitHub.
- Buka repository proyek tersebut.
- Klik tombol Code, lalu pilih tab Codespaces.
- Pilih Create codespace on [nama branch] (misalnya, main).

Setelah berhasil, hal pertama yang wajib dilakukan adalah membaca README. README adalah panduan atau keterangan yang disediakan pemilik proyek untuk membantu Anda menjalankan proyek tersebut.

## SQL Tools

Dalam **Codespaces**, Anda juga dapat menjalankan **query SQL** (Structured Query Language) melalui **SQL tools**. Anda bisa melakukan operasi dasar seperti **Create, Read, Update, Delete (CRUD)**, dan *query SQL* lainnya.



# 03 **BASIC PROGRAMMING**

# Python Programming



## Filosofi

- Bahasa pemrograman open-source, interpretatif dan multiguna.
- Dirancang tahun 1991 oleh Guido van Rossum.
- Didesain dengan penekanan pada keterbacaan kode dan sintaksis yang intuitif.



## Fitur Utama

- High readability, mudah diinterpretasikan
- Multiguna, mendukung banyak paradigma pemrograman
- Portabilitas tinggi, multi OS
- Alat analisis paling populer sehingga mudah menemukan komunitas diskusi



## Penerapan

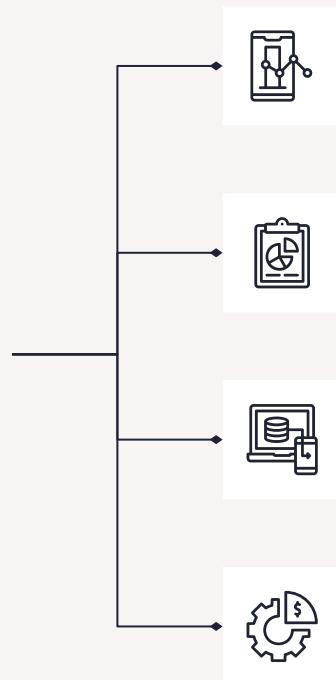
- Menjadi salah satu tools wajib bagi Data Scientist dan Data Analyst.
- Didukung banyak package berlimpah untuk analisis multi tema

# Google Collab



## Fitur

Notebook berbasis cloud untuk menulis dan mengeksekusi kode Python melalui browser.



### 1. Akses gratis ke GPU & TPU

Menawarkan sumber daya komputasi secara gratis untuk mempercepat pekerjaan machine learning.

### 2. Environment siap pakai

Sudah terinstal banyak library penting untuk machine learning seperti PyTorch, Keras, dan TensorFlow.

### 3. Kolaborasi tim

Memungkinkan beberapa orang untuk mengedit satu notebook secara bersamaan, sama seperti Google Docs.

### 4. Berbasis cloud

Semua pekerjaan tersimpan otomatis di Google Drive, menghemat penyimpanan lokal dan memudahkan akses dari mana saja.

# Variabel Python

## Definisi

- **Variabel** adalah lokasi memori yang dicadangkan untuk menyimpan nilai-nilai
- Variabel bersifat **mutable**, artinya nilainya bisa berubah-ubah
- **Syntax:**

Nama\_variabel = <nilai>

Contoh :

nilai = 20

nama = "Muhimah"

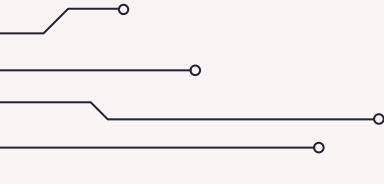
## Aturan Penulisan Variabel

1. Nama variabel boleh diawali menggunakan huruf atau garis bawah (\_), contoh: `nama`, `_nama`, `namaKu`, `nama_variabel`.
2. Karakter selanjutnya dapat berupa huruf, garis bawah (\_) atau angka, contoh: `--nama`, `n2`, `nilai`.
3. Karakter pada nama variabel bersifat sensitif (*case-sensitif*). Artinya huruf besar dan kecil dibedakan. Misalnya, `variabel_Ku` dan `variabel_ku`, keduanya adalah variabel yang berbeda.
4. Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti `if`, `while`, `for`, dsb.

# Data Type Python

Tipe Data	Contoh	Penjelasan
Boolean	True atau False	Menyatakan benar <code>True</code> yang bernilai 1, atau salah <code>False</code> yang bernilai 0
String	"Ayo belajar Python"	Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda " atau ')
Integer	25 atau 1209	Menyatakan bilangan bulat
Float	3.14 atau 0.99	Menyatakan bilangan yang mempunyai koma
Hexadecimal	9a atau 1d3	Menyatakan bilangan dalam format heksa (bilangan berbasis 16)
Complex	1 + 5j	Menyatakan pasangan angka real dan imaginer
List	[xyz, 786, 2.23]	Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
Tuple	(xyz, 768, 2.23)	Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah
Dictionary	{'nama': 'adi', 'id': 2}	Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

# Contoh Variabel dan Data Type Pada Program Sederhana



```
#boolean
laki = False
perempuan = True

print(perempuan)
print(laki)

True
False

string
nama = "Muhibah"
jabatan = "staf"
bagian = "logistik"
tugas = "Data dan Pelaporan Logistik"

print("memiliki jabatan sebagai ",jabatan," di bagian ",bagian," dan mempunya tugas mengolah ",tugas)

Muhibah memiliki jabatan sebagai staf di bagian logistik dan mempunya tugas mengolah Data dan Pelaporan Logistik

#integer
umur = 24
tahun_kelahiran = 1977
tahun_sekarang = 2025

print(umur)
print(tahun_sekarang - tahun_kelahiran)

24
24
```



```
#float
gbr = 3.24 Radu komi decimal

print("IPK adalah ",gbr)

IPK adalah 3.24

#list type
print(type(laki))
print(type(gbr))
print(type(umur))
print(type(lak))

<class 'bool'>
<class 'float'>
<class 'int'>
<class 'float'>
```



# Module, Package, & Library

## Module

**Pengertian:** Sebuah file tunggal berekstensi .py yang berisi kode Python (fungsi, class, atau variabel).

**Tujuan:** Mengelompokkan kode terkait agar lebih terorganisir dan mudah digunakan kembali.

**Contoh:** digitalskola.py bisa menjadi sebuah module yang berisi fungsi, class, variable, atau bahkan sebuah kode atau sekumpulan kode yang telah dikelompokkan.

## Package

**Pengertian:** Sebuah folder yang berisi sekumpulan module terkait dan sebuah file khusus \_\_init\_\_.py.

**Tujuan:** Mengelola dan mengorganisir module-module ke dalam satu direktori, memungkinkan struktur proyek yang lebih besar.

**Contoh:** Sebuah package analisis\_data bisa berisi module pembersihan.py dan visualisasi.py.

## Library

**Pengertian:** Kumpulan dari berbagai package dan module yang menyediakan fungsionalitas luas untuk tugas-tugas spesifik.

**Tujuan:** Menyediakan toolbox siap pakai sehingga programmer tidak perlu menulis kode dari nol.

**Contoh:**  
Pandas & Numpy (komputasi numerik).  
Matplotlib & Seaborn (visualisasi data)  
Scikit-Learn, TensorFlow, PyTorch (machine learning)

# Most common used library in python



**NumPy (Numerical Python):** Mengolah data dalam bentuk angka (matriks, vektor, array) dengan cepat.

**Pandas:** Mengolah data seperti di Excel — membaca, membersihkan, dan menganalisis tabel data

**Matplotlib:** Membuat grafik, diagram batang, garis, pie chart, dsb.

**Seaborn:** Visualisasi data yang lebih cantik dan interaktif (dibangun di atas Matplotlib).

**Scikit-learn:** Machine Learning — membuat model seperti klasifikasi, prediksi, dan clustering.

**TensorFlow:** Deep Learning — membuat dan melatih neural network.

**PyTorch:** Deep Learning seperti TensorFlow, tapi lebih fleksibel dan populer di kalangan peneliti.

**SciPy:** Perhitungan ilmiah seperti statistik, optimisasi, dan kalkulus

# Operator Aritmatika Python

Operator	Contoh Penggunaan	Penjelasan
+ (Penjumlahan)	$2 + 5 = 7$	Menjumlahkan dua nilai.
- (Pengurangan)	$5 - 2 = 3$	Mengurangkan nilai kanan dari kiri.
* (Perkalian)	$8 * 2 = 16$	Mengalikan dua nilai.
/ (Pembagian)	$16 / 2 = 8$	Membagi nilai kiri dengan kanan.
** (Pangkat)	$3 ** 3 = 27$	Memangkatkan nilai kiri dengan pangkat kanan.
% (Sisa bagi)	$22 \% 3 = 1$	Mengambil sisa hasil pembagian.
// (Pembagian Bulat)	$12 // 5 = 2$	Membagi lalu mengambil hasil bulatnya.

# Operator Logika Python

Operator	Penjelasan	Hasil	Contoh (A = 5, B = 10)
and	True and False	False	$A < 10 \text{ and } B < 10$  False
or	True or False	True	$A < 10 \text{ or } B < 10$  True
not	Not False	True	$A = \text{False}, \text{not } A$  True

# Conditional Statements

## If

**if text expression:**  
**statement(s)**

```
age = 25  
if age == 25:  
    print("+99 Aura")
```

## If - Else

**if text expression:**  
**Body of if**  
**else:**  
**Body of else**

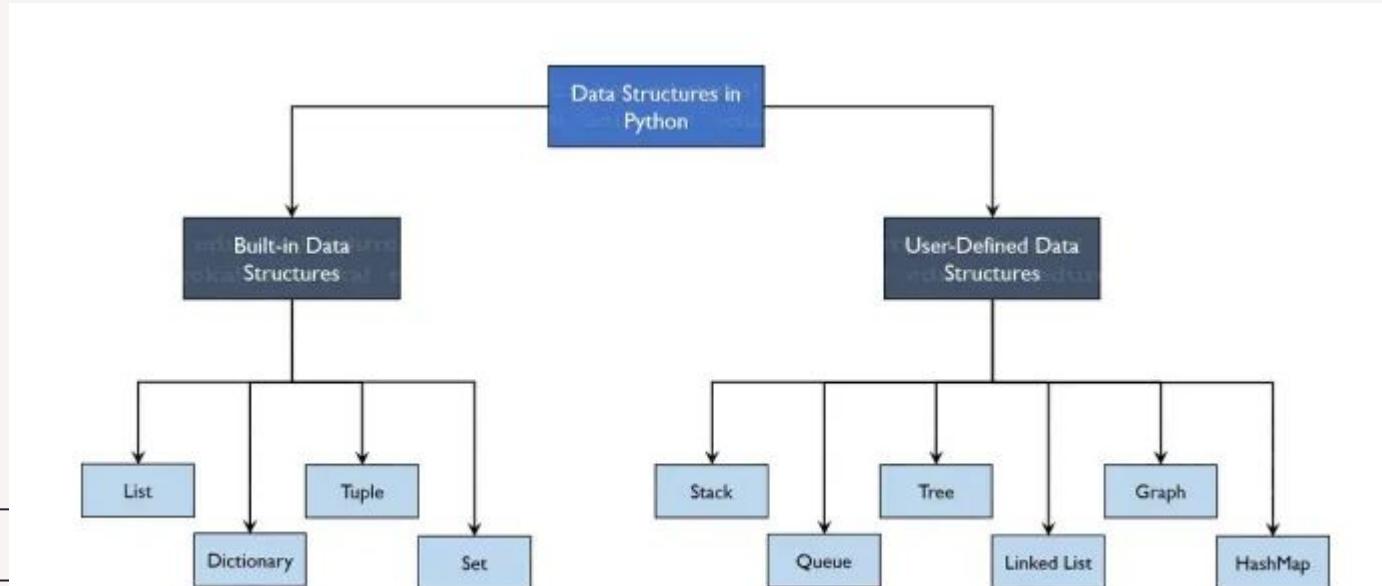
```
age = 25  
if age == 25:  
    print("+99 Aura")  
else:  
    print("-10 Aura")
```

## If - Elif

**if text expression:**  
**Body of if**  
**elif text expression:**  
**Body of elif**  
**else:**  
**Body of else**

```
age = 25  
if age >= 25:  
    print("+99 Aura")  
Elif age >=20:  
    print("+98 Aura")  
else:  
    print("-10 Aura")
```

# Data Structures in Python



# BUILT IN DATA STRUCTURES

Dalam section ini kita hanya fokus membahas built in data structures

Fitur	List	Tuple	Dictionary	Set
<b>Sintaks</b>	Ditulis dengan kurung siku []. Contoh: ['apel', 'pisang'].	Ditulis dengan kurung biasa (). Contoh: ('apel', 'pisang').	Ditulis dengan kurung kurawal {} dengan pasangan key: value. Contoh: {'nama': 'Adi', 'umur': 25}.	Ditulis dengan kurung kurawal {} dengan item tunggal. Contoh: {'apel', 'pisang'}.
<b>Keterurutan</b>	Terurut. Item memiliki urutan yang pasti dan akan tetap sama.	Item memiliki urutan yang pasti dan tidak berubah.	Terurut (mulai Python 3.7). Item memiliki urutan penyisipan.	Tidak terurut. Item tidak memiliki urutan yang pasti dan dapat berubah.
<b>Perubahan</b>	Bisa diubah (mutable). Anda dapat menambah, menghapus, atau mengubah item.	Tidak bisa diubah (immutable). Setelah dibuat, item tidak bisa diubah.	Bisa diubah (mutable). Anda dapat menambah, menghapus, atau mengubah pasangan key: value.	Bisa diubah (mutable). Anda dapat menambah atau menghapus item, tapi isi item tidak bisa diubah.
<b>Elemen Duplikat</b>	Diperbolehkan. Bisa berisi item dengan nilai yang sama.	Diperbolehkan. Bisa berisi item dengan nilai yang sama.	Tidak diperbolehkan untuk kunci (key). Nilai (value) boleh duplikat.	Tidak diperbolehkan. Hanya menyimpan item yang unik.
<b>Akses Item</b>	Dengan indeks (berbasis nol). Contoh: list[0].	Dengan indeks (berbasis nol). Contoh: tuple[0].	Dengan kunci (key). Contoh: dict['nama'].	Item tidak bisa diakses dengan indeks karena tidak terurut.



# **THANK YOU!**

