

# LEARNING PROGRESS REVIEW

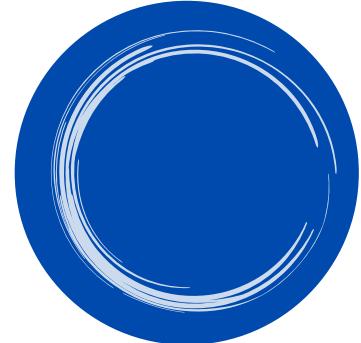
## “WEEK 2”

BY : DREAM (KELOMPOK 1)

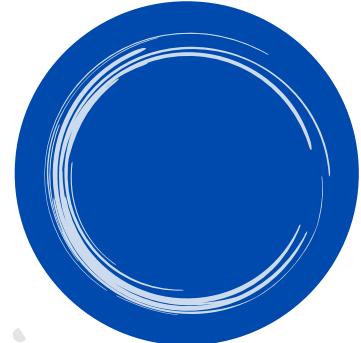
*Data Realm Engineers And Maestros*



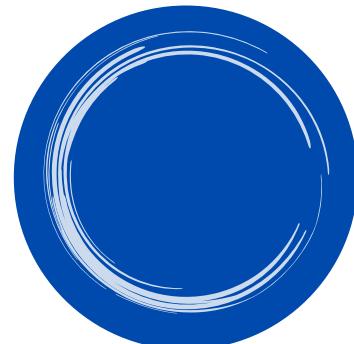
# **ANGGOTA KELOMPOK**



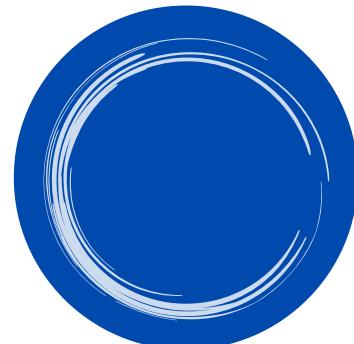
**Afroh Fauziah**



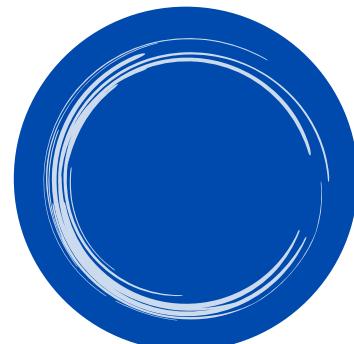
**Althaf Taqiyyah**



**Andrew Bintang Pratama**

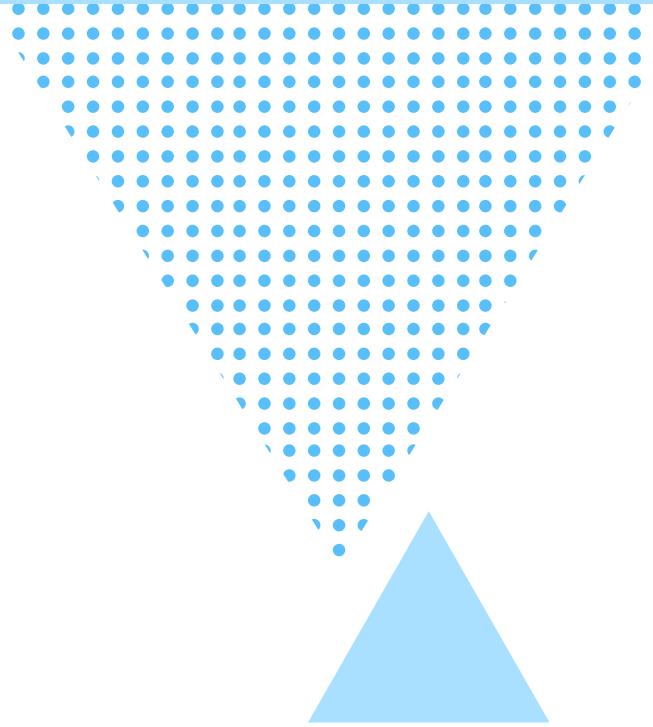


**Andrew Suadnya**



**Andi Rosilala**





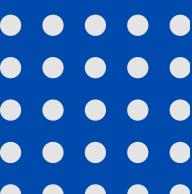
# PYTHON EV



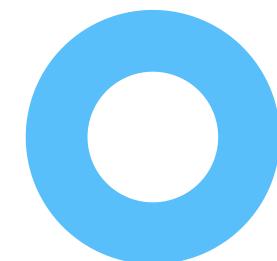
# Virtual Env using Conda



Virtual environment memungkinkan kita untuk mengisolasi proyek Python kita, memungkinkan penggunaan versi paket yang berbeda tanpa mengganggu paket yang sudah ada secara global.



## Perintah - Perintah Conda



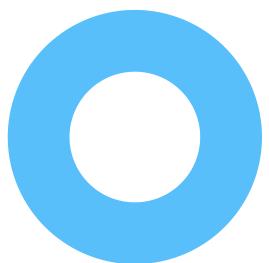
### Membuat Env Virtual

```
▼ TERMINAL
PS C:\Users\ASUS\VirtualBox VMs\staroffice> conda create --Bintang myenv python=3.9 numpy pandas
>>
```

### Membuat Lingkungan Virtual dengan Paket Tertentu

```
▼ TERMINAL
PS C:\Users\ASUS\VirtualBox VMs\staroffice> conda create --bintang myenv
```

## Perintah - Perintah Conda



### Mengaktifkan Lingkungan Virtual:

▼ TERMINAL

```
PS C:\Users\ASUS\VirtualBox VMs\staroffice> conda activate myenv
```

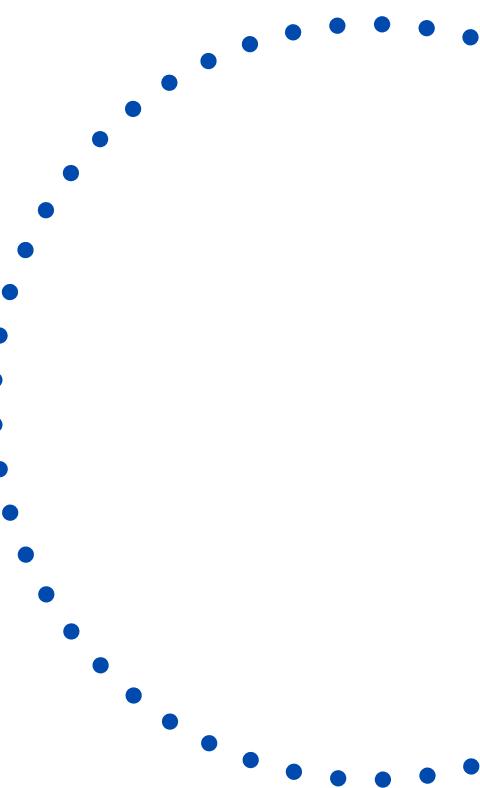
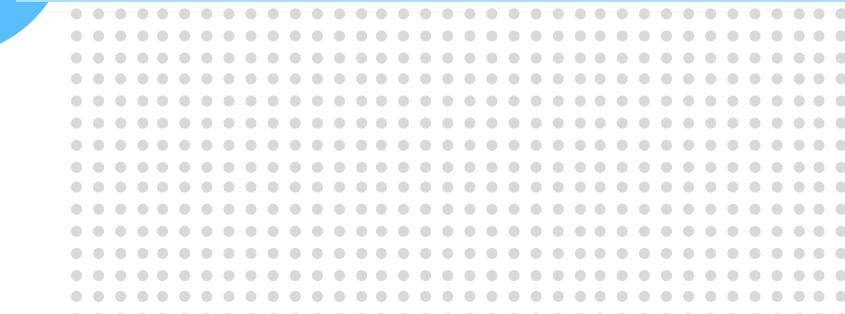
### Menonaktifkan Lingkungan Virtual

▼ TERMINAL

```
PS C:\Users\ASUS\VirtualBox VMs\staroffice> conda deactivate
```

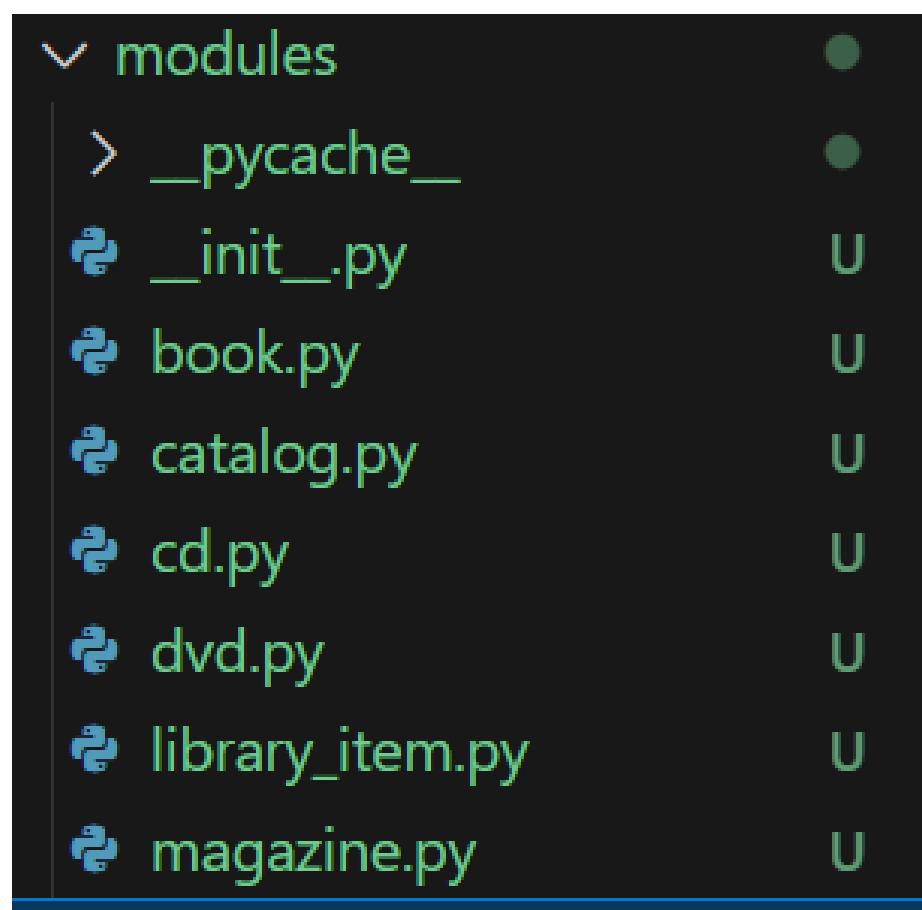
# Python Module

Modul Python adalah file yang berisi definisi dan pernyataan Python. File-file ini sering kali berisi kode yang dibuat oleh pengembang atau pihak ketiga yang dapat digunakan kembali dalam program Python. Modul memungkinkan Anda untuk mengatur kode Anda menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola, serta memungkinkan penggunaan kode yang sama di berbagai proyek.



# Python Module

## Python Module



## Importing module

A code editor window showing a Python script named 'Main.py'. The code imports modules from the 'modules' package:

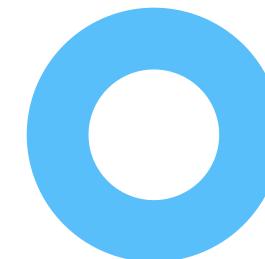
```
1 import json
2 from modules.book import Book
3 from modules.magazine import Magazine
4 from modules.cd import Cd
5 from modules.dvd import Dvd
6 from modules.catalog import Catalog
```



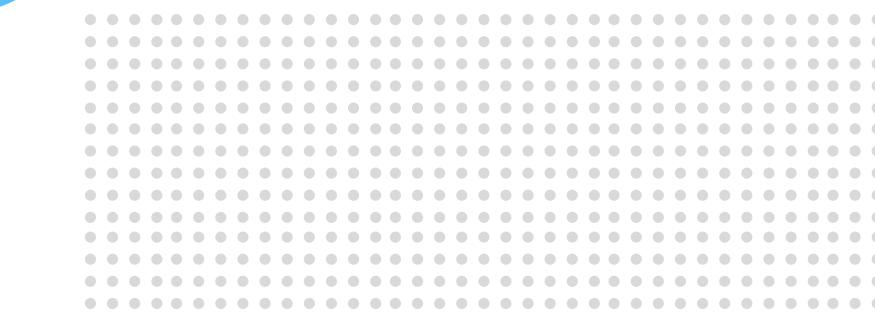


# Built Package

## Pypi (python package index)



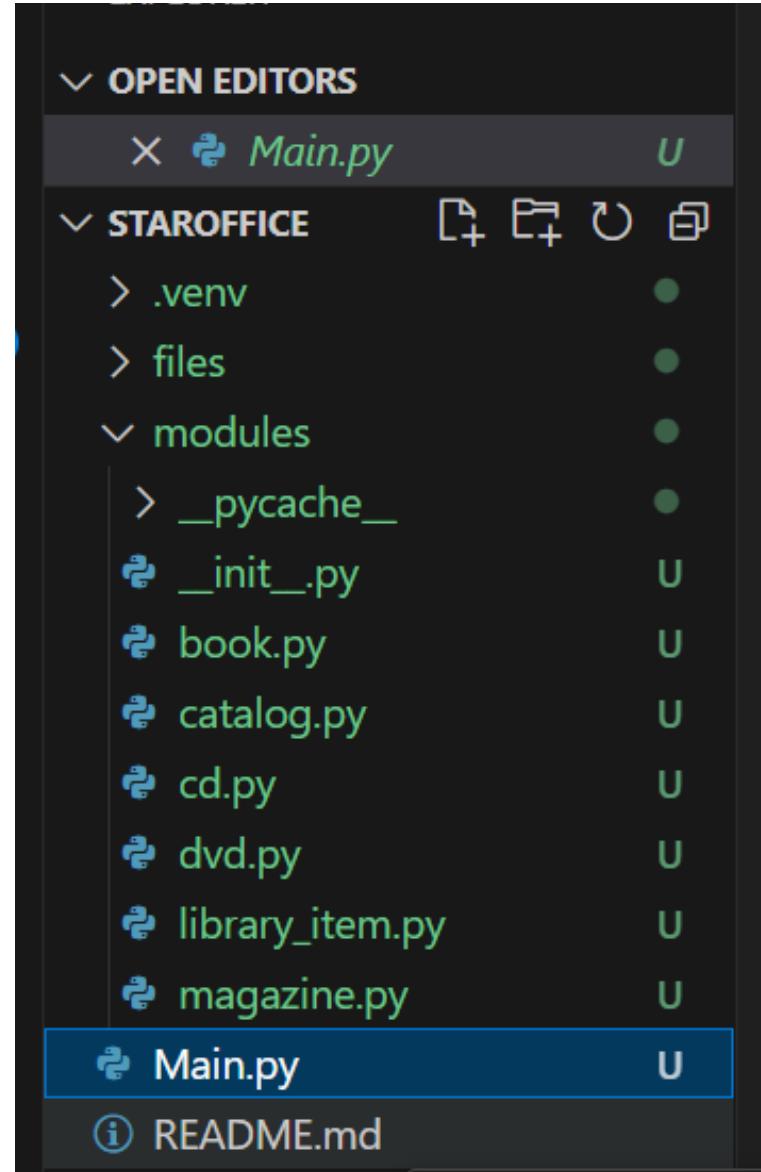
PyPI adalah singkatan dari Python Package Index. Ini adalah repositori online yang berisi ribuan paket perangkat lunak yang ditulis dalam bahasa pemrograman Python. PyPI memungkinkan pengembang Python untuk membagikan, menemukan, dan menginstal perangkat lunak Python yang dibuat oleh komunitas Python.





# Built Package

## Python package

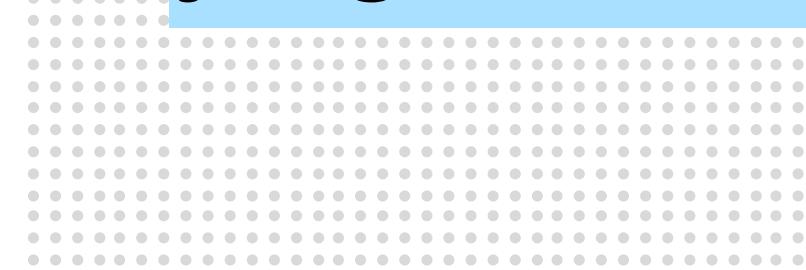
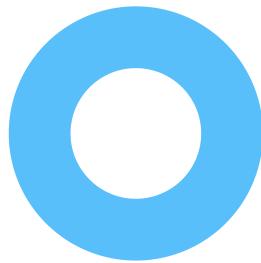


1. File `setup.py` digunakan untuk mengonfigurasi proyek Anda dan memberikan informasi tentang bagaimana menginstal dan mendistribusikannya.
2. File `README.md` adalah file Markdown yang berisi informasi tentang proyek Anda, termasuk penggunaan, instalasi, dan dokumentasi lainnya.
3. Buat direktori `tests` yang berisi file-file pengujian untuk menguji kode Anda. Ini penting untuk memastikan bahwa kode Anda berfungsi dengan benar.
4. File `tox.ini` digunakan untuk mengkonfigurasi dan menjalankan tes otomatis dengan menggunakan perangkat lunak tox. Ini membantu untuk memastikan bahwa proyek Anda dapat diuji di berbagai lingkungan Python yang berbeda.



# Flake8

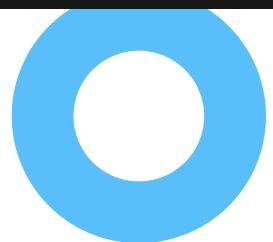
Flake8 adalah alat linting yang sangat populer untuk Python. Ini menggabungkan beberapa alat linting seperti PyFlakes, pycodestyle (sebelumnya dikenal sebagai pep8), dan Ned Batchelder's McCabe script menjadi satu paket. Flake8 membantu Anda menjaga kode Python Anda konsisten, membantu dalam mengidentifikasi potensi bug, memperbaiki gaya kode yang tidak konsisten, serta memastikan kualitas kode yang lebih baik secara keseluruhan.



# Flake8

## Install flake8

```
▽ TERMINAL  
PS C:\Users\ASUS\VirtualBox VMs\staroffice> pip install flake8  
Requirement already satisfied: flake8 in c:\python310\lib\site-packages (7.0.0)  
Requirement already satisfied: mccabe<0.8.0,>=0.7.0 in c:\python310\lib\site-packages (from flake8) (0.7.0)  
Requirement already satisfied: pycodestyle<2.12.0,>=2.11.0 in c:\python310\lib\site-packages (from flake8) (2.11.  
1)  
Requirement already satisfied: pyflakes<3.3.0,>=3.2.0 in c:\python310\lib\site-packages (from flake8) (3.2.0)  
PS C:\Users\ASUS\VirtualBox VMs\staroffice>
```



# Flake8

## Jalankan flake8

```
PS C:\Users\ASUS\VirtualBox VMs\staroffice> flake8
```

## Setelah dijalankan flake8

```
TERMINAL
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:58:80: E501 line too long (93 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:59:80: E501 line too long (159 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:60:80: E501 line too long (100 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:61:80: E501 line too long (100 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:62:80: E501 line too long (90 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:63:80: E501 line too long (88 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:64:80: E501 line too long (92 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:65:80: E501 line too long (86 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:66:80: E501 line too long (90 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:67:80: E501 line too long (88 > 79 characters)
.\.venv\Lib\site-packages\pip\_vendor\rich\table.py:86:80: E501 line too long (91 > 79 characters)
```

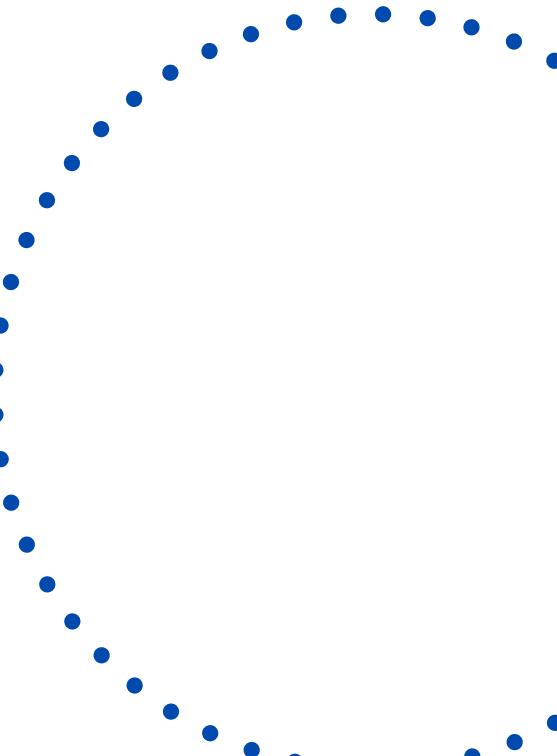
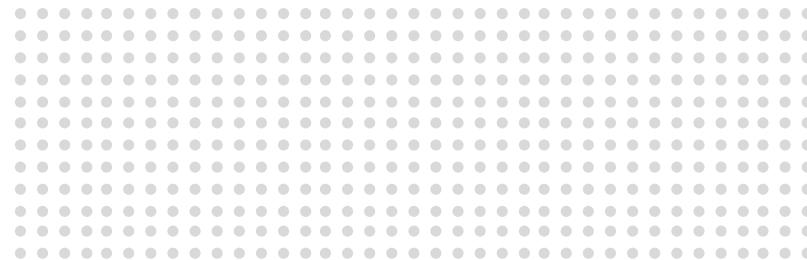
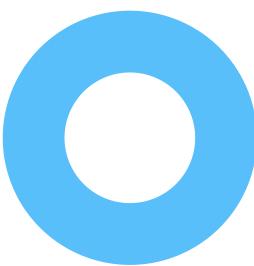
Ln 32, Col 23 Spaces: 4 UTF-8 CRLF { } P





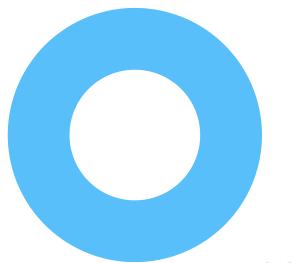
# TOX

Tox adalah alat yang digunakan untuk otomatisasi pengujian dan lingkungan virtual di proyek Python. Ini membantu pengembang Python memastikan bahwa kode mereka berfungsi dengan baik di berbagai versi Python dan lingkungan yang berbeda.

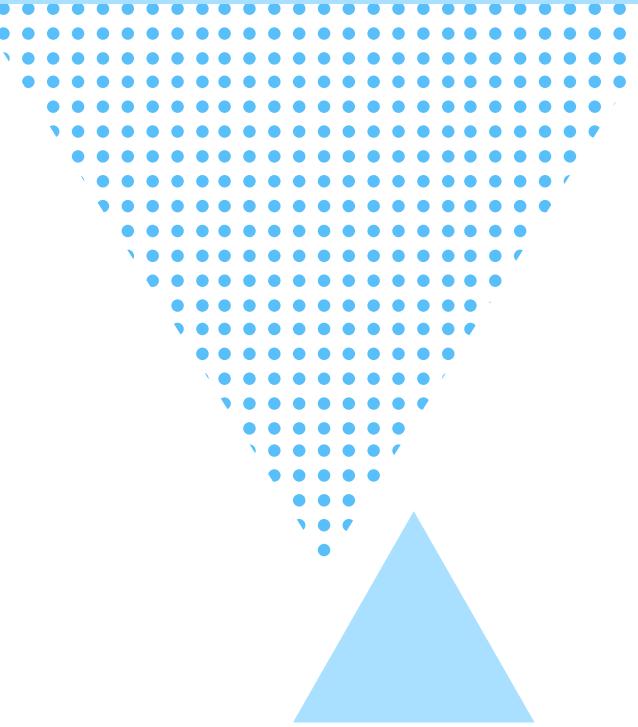


# TOX

```
n ... ← → ⌂ py_calculator  
tox.ini  
tox.ini  
1 [tox]  
2 envlist = py37, py38, py39, flake8, cov  
3 skipsdist = true  
4  
5 [testenv]  
6 deps =  
7 | pytest  
8 | pytest-cov  
9 commands =  
10 | pytest --cov=py_calculator {posargs:[--cov-report=xml]}  
11  
12 [testenv:flake8]  
13 deps =  
14 | flake8  
15 commands =  
16 | flake8 py_calculator tests  
17  
18 [py37]  
19 deps =  
20 | python=3.7  
21  
22 [py38]  
23 deps =  
24 | python=3.8  
25  
26 [py39]  
27 deps =  
28 | python=3.9
```



**tox.ini**



# GET CONCEPTS





# GIT



Git adalah sebuah sistem yang membantu kita melacak perubahan pada kode kita. Sistem ini bekerja dengan cara merekam perubahan tersebut ke dalam sebuah database khusus yang disebut "Repository".

Berkat Git, kita bisa melihat riwayat perubahan kode kita dengan mudah, termasuk siapa yang melakukan perubahan, kapan perubahan itu terjadi, dan alasan di balik perubahan tersebut.



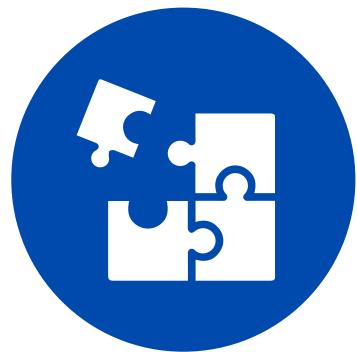
# KEUNTUNGAN MENGGUNAKAN GIT



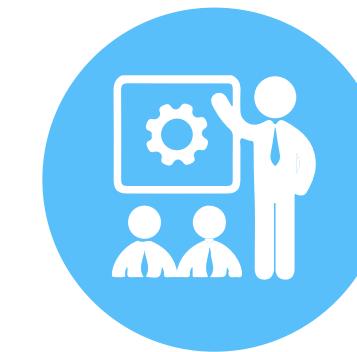
Melacak perubahan  
kode dengan mudah



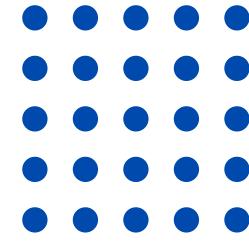
Melihat perkembangan  
kode dari waktu ke waktu



Mengembalikan kode  
ke versi sebelumnya

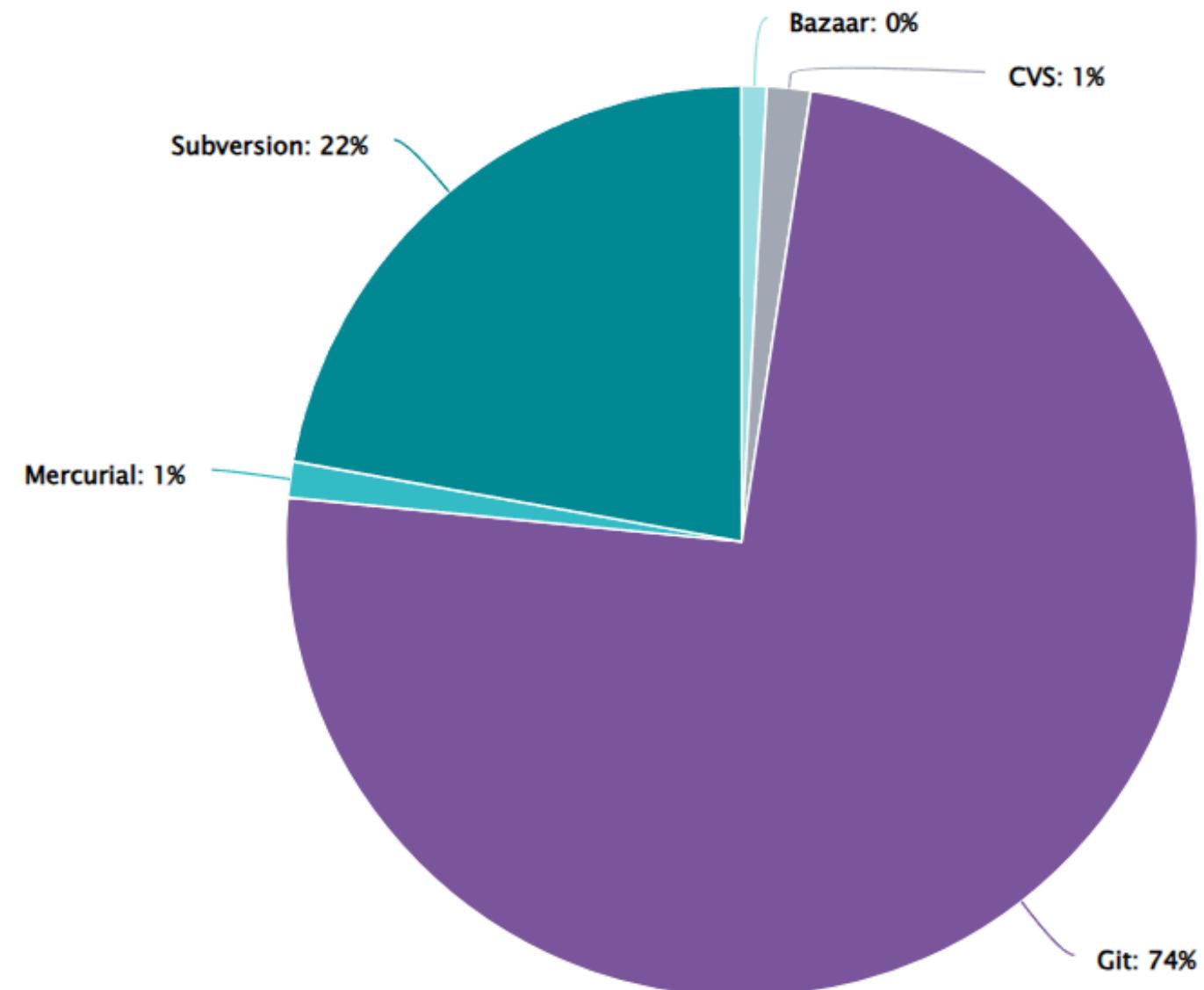


Berkolaborasi dengan  
tim lain



# PENGUASAAN GIT DI KALANGAN DEVELOPER

Compare Repositories



Berdasarkan survei, Git mendominasi sebagai repository yang paling banyak digunakan oleh developer (74%), jauh mengungguli Subversion (22%), Mercurial (1%), Bazaar (0%), dan CVS (1%). Hal ini menunjukkan bahwa Git telah menjadi pilihan utama bagi developer dalam mengelola kode mereka.

Popularitas Git ini dapat dikaitkan dengan berbagai kelebihan yang ditawarkannya, seperti keamanan yang tinggi, kemudahan penggunaan bersama tim, dan kemampuannya dalam melacak riwayat perubahan kode dengan detail. Git membantu developer untuk bekerja dengan lebih efisien dan profesional, sehingga menjadikannya alat yang sangat penting dalam pengembangan perangkat lunak modern.



# GLOSARIUM GIT



## Repository

Dasar dari kontrol versi Git, repositori adalah database terpusat yang menyimpan riwayat lengkap perubahan pada sekumpulan file.

Repositori dapat berupa:

- **Local repository:** Di komputer developer.
- **Remote repository:** Dihosting secara online (contoh: GitHub, GitLab).

## Branch

Branch adalah fitur Git yang memungkinkan developer untuk membuat jalur pengembangan terpisah.

## Commit

Commit adalah tindakan menyimpan snapshot yang telah distaged ke dalam repositori.

Setiap commit menyimpan metadata seperti:

- Pesan yang menjelaskan perubahan yang dilakukan.
- Siapa yang melakukan perubahan tersebut.

## Master/Main

Secara default, repositori Git memiliki branch utama yang disebut master atau main. Branch ini biasanya berisi kode yang dianggap stabil dan siap untuk dirilis.

## Snapshot

Snapshot mewakili keadaan sebuah file atau folder pada titik waktu tertentu. Dalam Git, snapshot ini sering disebut sebagai versi atau revisi.

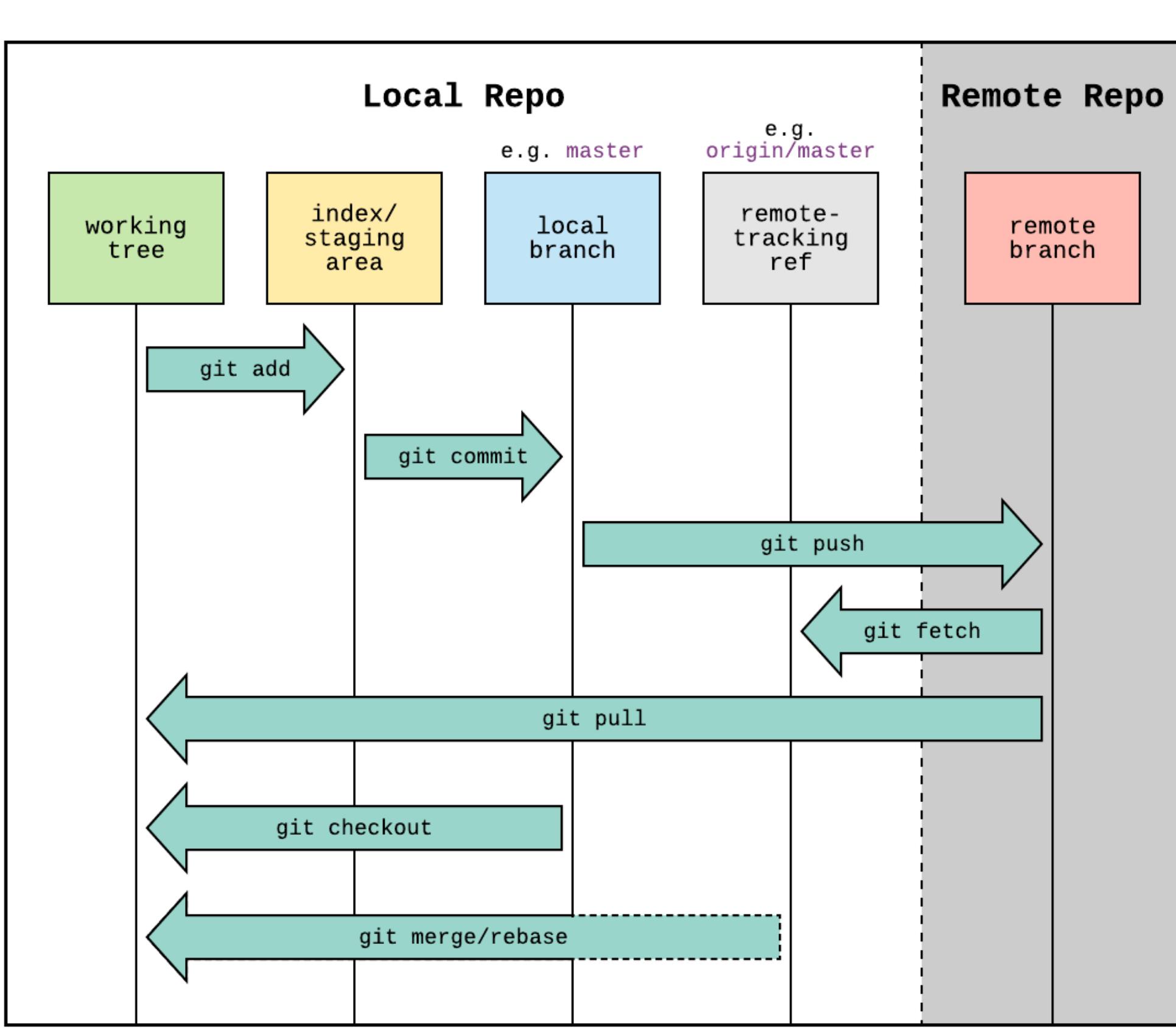
## Snapshot

Staging adalah proses memilih revisi spesifik yang akan disertakan dalam commit berikutnya.

# PERINTAH DASAR GIT

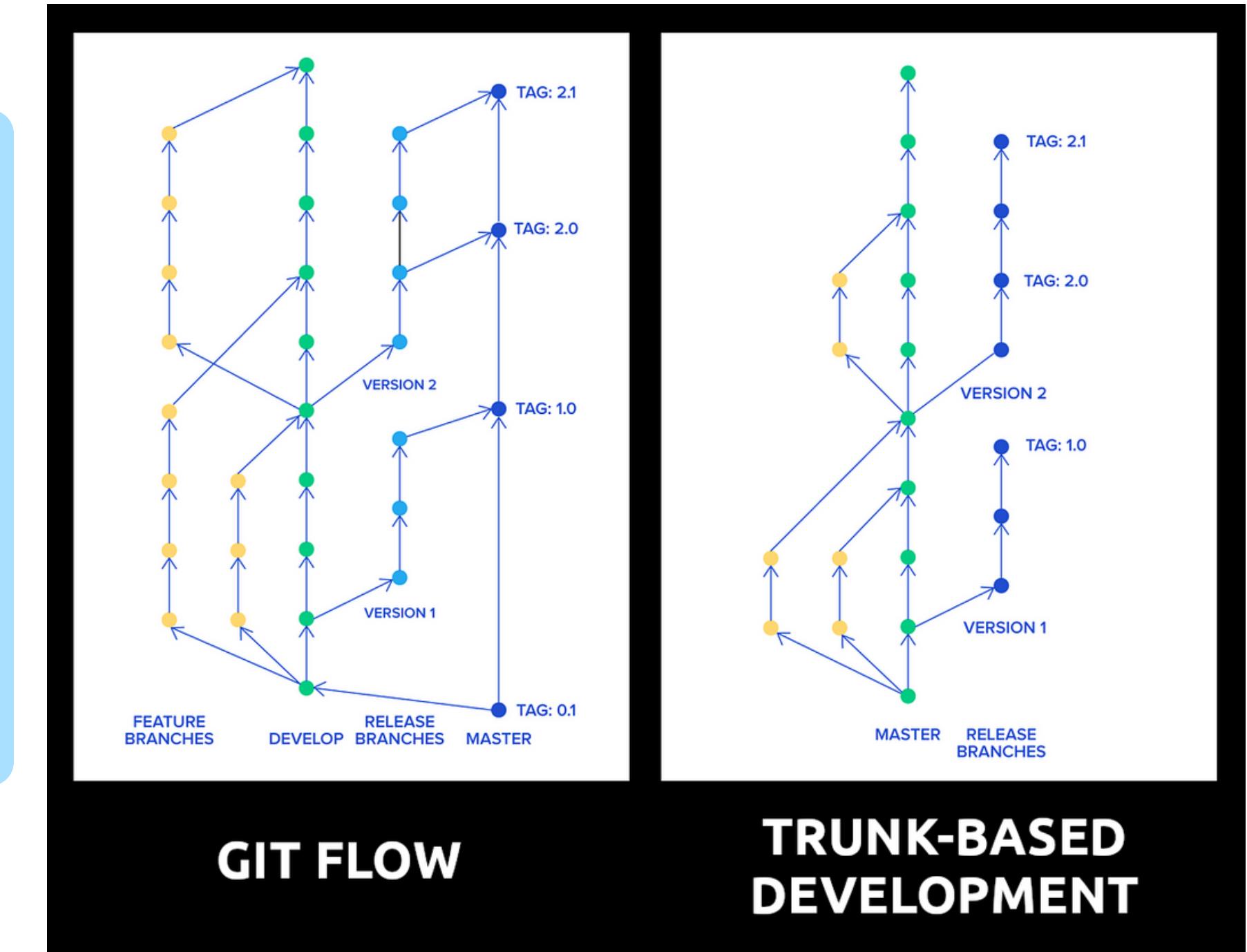
- “git branch -a” melihat branch yang ada pada suatu repository (local & remote).
- “git status” melihat status perbedaan dari branch kita dan current HEAD commit.
- “git diff” melihat perubahan/perbedaan file (line by line) dari branch kita dan current HEAD commit. Biasanya hal-hal seperti ini lebih mudah dilakukan pada IDE (VS Code/IntelliJ).
- “git add nama\_file” menyiapkan snapshot/status dari suatu file (“staged”) agar bisa digunakan pada “commit”.
- “git commit -m ‘pesan dalam commit’” menyimpan suatu perubahan dalam git. (Perubahan yang disimpan adalah perubahan yang sudah ada didalam “staged”).
- “git log” melihat history atau log pada suatu branch.
- “git push origin nama\_branch” mengubah suatu branch di remote repository sesuai dengan perubahan yang terjadi pada local repository.
- “git merge nama\_branch” menggabungkan suatu branch dengan branch utama.

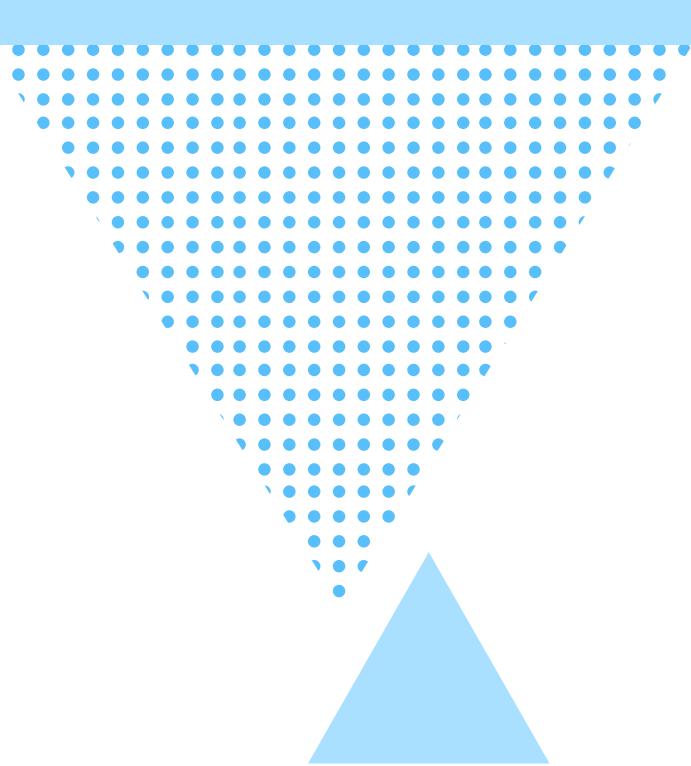
# WORKFLOW GIT



# GIT FLOW VS TRUNK-BASED DEVELOPMENT

Gitflow dan Trunk-based Development adalah dua alur kerja populer dalam Git. Gitflow menggunakan branch "long-lived" untuk fitur dan environment, cocok untuk proyek besar dengan tim terstruktur. Trunk-based Development menggunakan branch "short-lived" dengan pull request ke branch utama, cocok untuk tim kecil yang bergerak cepat.



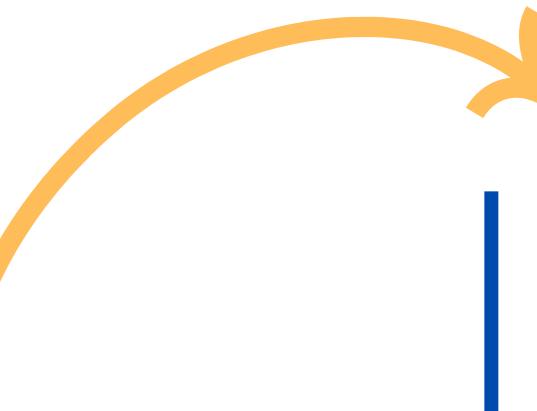


# GITHUB





# GITHUB



GitHub dikembangkan oleh Chris Wanstrath, P. J. Hyett, Tom Preston-Werner, dan Scott Chacon menggunakan Ruby on Rails, dan dimulai pada bulan Februari 2008.

Ini adalah anak perusahaan dari Microsoft, yang mengakuisisi perusahaan pada tahun 2018 dengan harga \$ 7,5 miliar

GitHub.com adalah situs untuk penyimpanan online repositori Git.

# Git



First developed in 2005

Git is installed and maintained on your local system (rather than in the cloud)

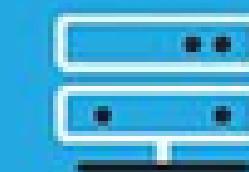


One thing that really sets Git apart is its branching model

Git is a high quality version control system

VS.

# GitHub



GitHub is designed as a Git repository hosting service



You can share your code with others, giving them the power to make revisions or edits



GitHub is a cloud-based hosting service



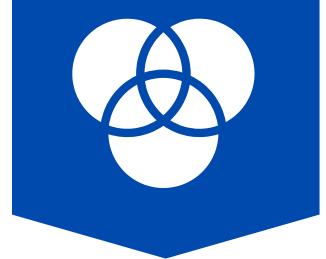


# Review Pull Requests pada GitHub



## Definisi

Permintaan untuk menggabungkan perubahan dari satu branch ke branch lain di GitHub.



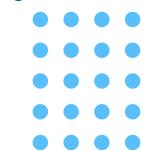
## Manfaat

- Meningkatkan kualitas kode.
- Menemukan dan memperbaiki bug.
- Meningkatkan kolaborasi tim.



## Alur Kerja

1. Fork repository yang ingin diubah.
2. Buat branch baru
3. Lakukan perubahan pada kode.
4. Buat pull request ke branch target di repository utama.
5. Tunggu approval dari reviewer.

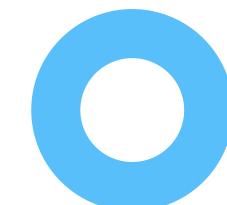




# Resolve Merge Conflicts

## Definisi

- Terjadi saat dua branch memiliki perubahan pada baris kode yang sama.
- Terjadi saat merge pull request.

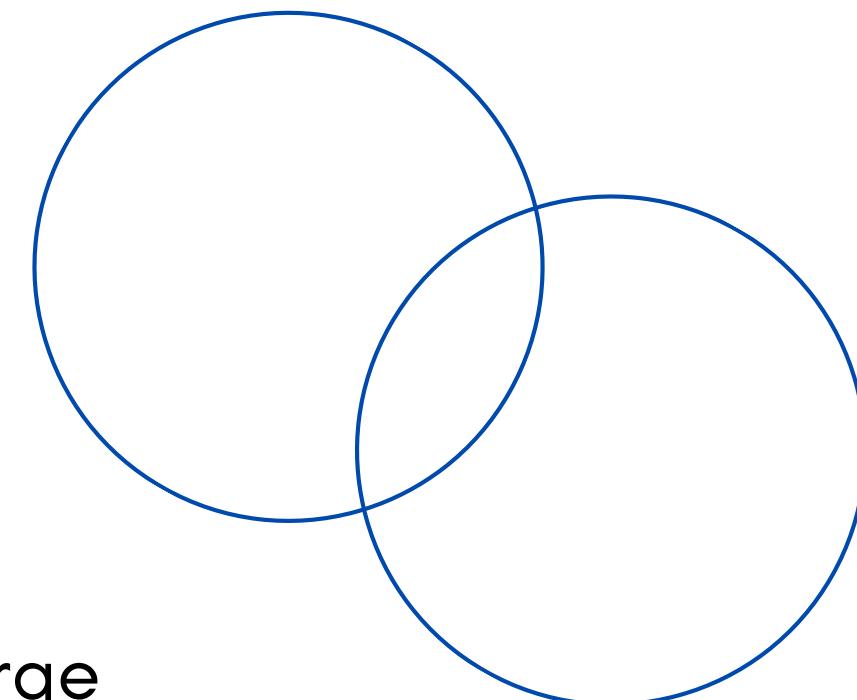


## Menemukan Merge Conflict

- GitHub menunjukkan pesan error saat merge pull request.
- File yang berkonflik ditandai dengan warna merah.
- Cari perbedaan antara perubahan di kedua branch.

## Cara Resolve Merge Conflicts

1. Buka file yang berkonflik.
2. Bandingkan perubahan
3. Pilih perubahan
4. Hapus tanda merge conflict.
5. Commit dan merge pull request.



# SOCIAL ASPECT OF GITHUB



Social network-like interface

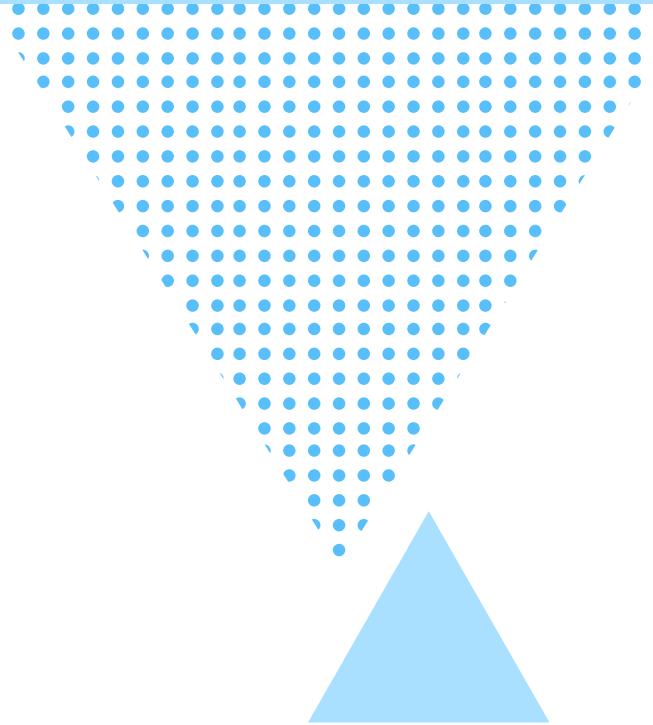
- Follow other people work (“friends”)
- Favors sharing , commenting
- Statistics, various useful plots

# BASIC GIT & GITHUB COMMAND

- git init - Initialize a Git repository/working directory
- git status - Status of your working directory
- git add <filename> or git add . (for all files in your working directory)
- git commit - Stash changes in your working directory
- git log -oneline - View your commit history
- git clone - Create an identical copy



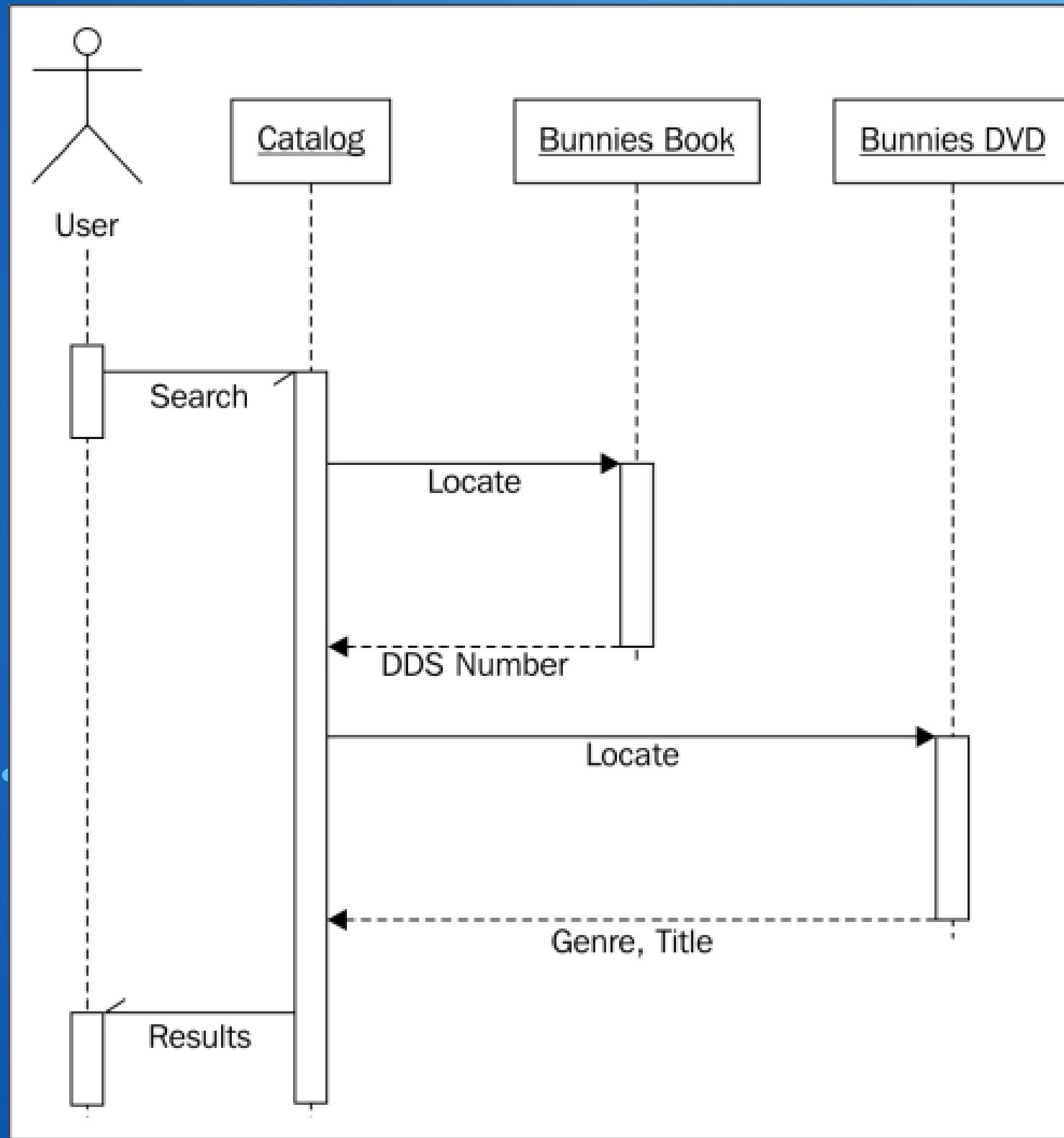
git pull-request  
git hub compare  
git hub create  
git hub release  
git hub fork  
git push  
git pull  
git branch  
git checkout  
git merge



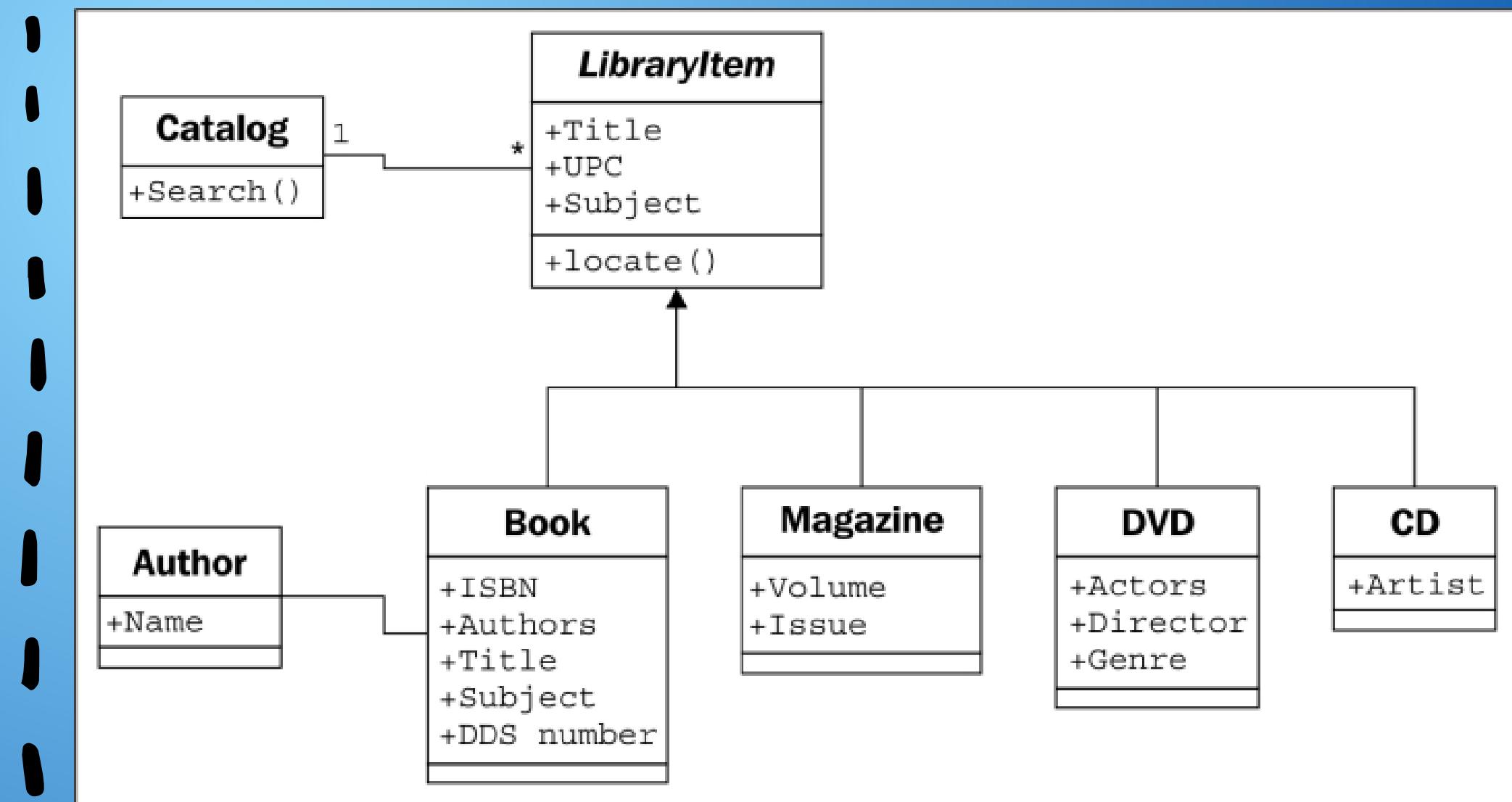
# PROJECT 1



# CASE 1

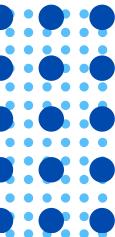


## Create Automation

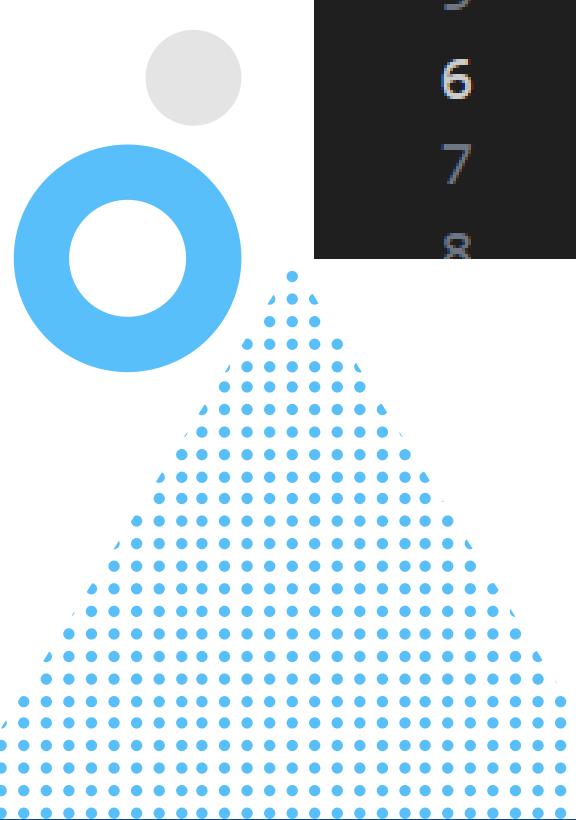


## Create OOP





# KELAS LIBRARY ITEM



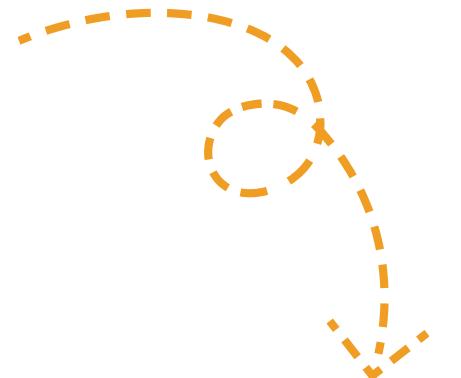
```
lib library_item.py X
modules > lib library_item.py > LibraryItem > __init__
1 class LibraryItem:
2     def __init__(self, title=None, upc=None, subject=None, volume=None, issue=None):
3         self.title = title
4         self.upc = upc
5         self.subject = subject
6         self.volume = volume
7         self.issue = issue
8
```



Membuat kelas library\_item sebagai parent class.  
Atribut dan metode dari class ini akan diwariskan ke  
class item book, cd, dvd dan magazine.

# KELAS BOOK

```
book.py U X  
modules > book.py > ...  
1  from modules.library_item import LibraryItem  
2  
3  class Book(LibraryItem):  
4      def __init__(self, title, upc, subject, isbn, authors, dds_number):  
5          super().__init__(title, upc, subject)  
6          self.isbn = isbn  
7          self.authors = authors  
8          self.dds_number = dds_number  
9  
10 book1 = Book(  
11     title='Test book',  
12     upc='',  
13     subject='this is subject',  
14     isbn='',  
15     authors='',  
16     dds_number=''  
17 )  
18  
19 print(book1.title)  
20
```

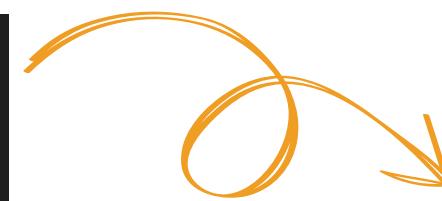


Pembuatan kelas book dengan dengan atribut dan operasi yang diwarisi dari kelas library\_item. Terdapat variabel khusus pada kelas book, yaitu subject, isbn, authors dan dds\_number.



# KELAS CD, DVD, & MAGAZINE

```
modules > cd.py > Cd
1   from modules.library_item import LibraryItem
2
3   class Cd(LibraryItem):
4       def __init__(self, title, upc, subject, artist):
5           super().__init__(title, upc, subject)
6           self.artist = artist
```



**Kelas cd diisi atribut title, upc, subject dan artist.**



**Kelas dvd diisi atribut title, upc, subject, genre, actors dan directors.**

```
class Dvd(LibraryItem):
    def __init__(self, title, upc, subject, actors, directors, genre):
        super().__init__(title, upc, subject)
        self.actors = actors
        self.directors = directors
```



**Kelas magazine diisi atribut title, upc, subject, volume dan issue.**

```
class Magazine(LibraryItem):
    def __init__(self, title, upc, subject, volume, issue):
        super().__init__(title, upc, subject)
        self.volume = volume
        self.issue = issue
```

# KELAS CATALOG & CATALOG.JSON

```
catalog.json U X  
files > {} catalog.json > {} 1  
1 [  
2 {  
3     "source": "cd",  
4     "title": "ini cd 1",  
5     "subject": null,  
6     "upc": null,  
7     "artist": null  
8 },  
9 {  
10    "source": "book",  
11    "title": "Title test",  
12    "subject": "ini subject test",  
13    "upc": null,  
14    "isbn": "1345-123",  
15    "authors": "Febri",  
16    "dds_number": 123412  
17 },  
18 {  
19     "source": "book",  
20     "title": "Title test",  
21     "subject": "ini subject test",  
22     "upc": null,  
23     "isbn": "1345-123",  
24     "authors": "Febri",  
25     "dds_number": 123412  
26 },  
27 {  
28     "source": "book",  
29     "title": "Title test",  
30     "subject": "ini subject test".
```

File catalog.json berisi data katalog dalam format JSON, objeknya mewakili item dalam katalog, yakni untuk Book, Magazine, CD & DVD

Kelas catalog.py berfungsi menyediakan fungsionalitas pencarian dalam katalog item dengan memasukkan semua variabel yang dibutuhkan.

```
s > catalog.py > ...  
class Catalog():  
    def search(self, input_search):  
        for catalog_item in self.catalog:  
            for item in catalog_item:  
                if input_search.lower() in item.title.lower():  
                    if type(item) is Magazine:  
                        list_result.append(f'Title: {item.title}, Volume: {item.volume}, Type: Magazine')  
                    elif type(item) is Book:  
                        list_result.append(f'Title: {item.title}, DDS Number: {item.dds_number}, Type: Book')  
                    elif type(item) is Cd:  
                        list_result.append(f'Title: {item.title}, Artist: {item.artist}, Type: Cd')  
                    elif type(item) is Dvd:  
                        list_result.append(f'Title: {item.title}, Genre: {item.genre}, Type: Dvd')  
        return list_result
```

# CODE PROGRAM MAIN.PY



```
main.py > ...
import json
from modules.book import Book
from modules.magazine import Magazine
from modules.cd import Cd
from modules.dvd import Dvd
from modules.catalog import Catalog

# Membuka file catalog.json
with open('files/catalog.json') as f:
    data_json = json.load(f)

# Inisialisasi daftar untuk menyimpan instance objek yang sesuai dengan jenis item
books = []
magazines = []
cds = []
dvds = []

# Iterate melalui setiap item dalam data JSON dan membuat instance objek yang sesuai
for item in data_json:
    source = item['source']
    if source == 'book':
        books.append(Book(
            title=item['title'],
            subject=item['subject'],
            upc=item['upc'],
            isbn=item['isbn'],
            authors=item['authors'],
            dds_number=item['dds_number']
        ))
    elif source == 'magazine':
        magazines.append(Magazine(
            title=item['title'],
            subject=item['subject'],
            upc=item['upc'],
            volume=item['volume'],
            issue=item['issue']
        ))
    elif source == 'cd':
        cds.append(Cd(
            title=item['title'],
            subject=item['subject'],
            upc=item['upc'],
            artist=item['artist']
        ))
    elif source == 'dvd':
        dvds.append(Dvd(
            title=item['title'],
            subject=item['subject'],
            upc=item['upc'],
            genre=item['genre']
        ))

# Menyusun katalog dengan semua item yang sudah dibuat
catalog_all = [books, magazines, cds, dvds]

# Mencari item dengan input tertentu
input_search = 'test'
results = Catalog(catalog_all).search(input_search)
```

File main.py sebagai logika utama program, seperti membuka & membuat instance objek

berdasarkan file JSON, menyusun katalog, mencari & menampilkan hasil pencarian, jadi filenya mengandung kode hingga bertanggung jawab untuk berjalannya program.

# OUTPUT

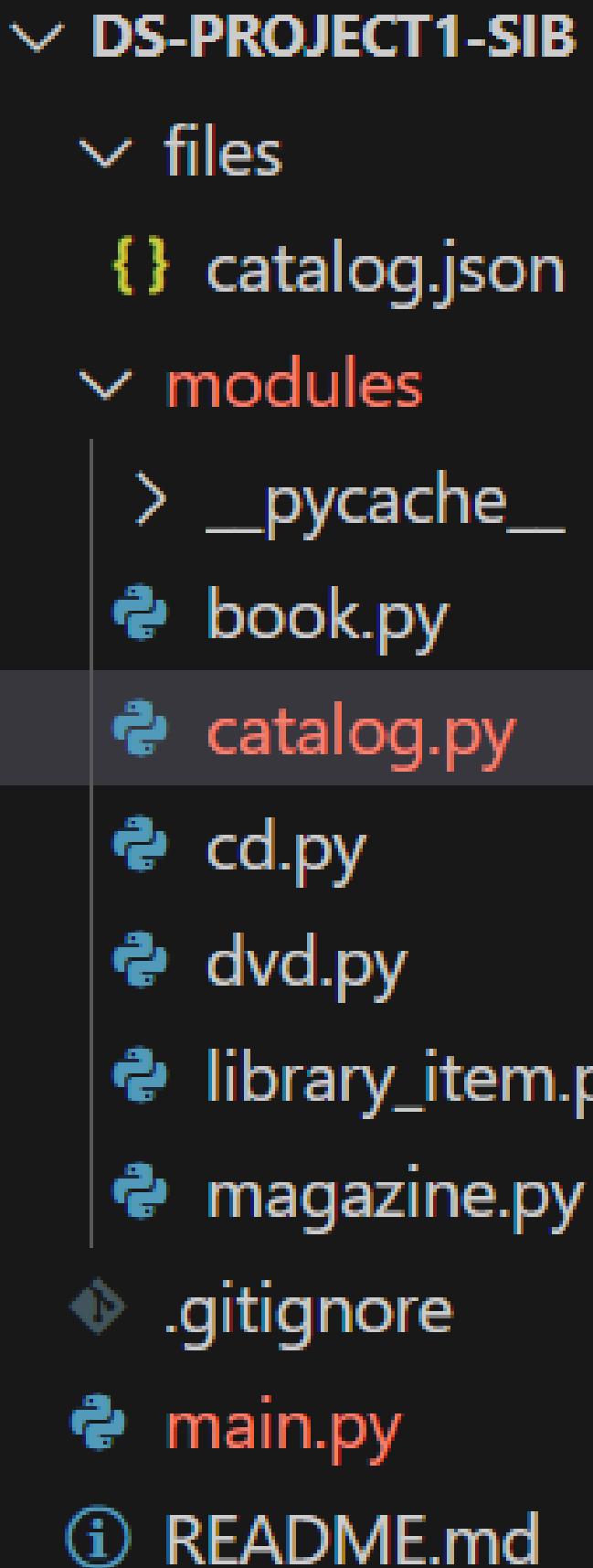
## keseluruhan poin / file yang dibuat

note: file .gitignore dibuat agar file tertentu (seperti file biner hasil kompilasi) dapat diabaikan saat kita menyimpan file ke repository github.

• • • • • • • • • • • • • • • •

▶ Hasil akhir dari semua kelas variabel dengan main.py yang lengkap

```
(.venv) C:\Users\ASUS\VirtualBox VMs\staroffice>python main.py
Test book
result ke-1 | Title: Title test, DDS Number: 123412, Type: Book
result ke-2 | Title: Title test, DDS Number: 123412, Type: Book
result ke-3 | Title: Title test, DDS Number: 123412, Type: Book
result ke-4 | Title: ini book 1 test, DDS Number: 123412, Type: Book
result ke-5 | Title: media_cnn test, Volume: volume 1, Type: Magazine
result ke-6 | Title: Test Dvd 1, Genre: Comedy, Type: Dvd
```

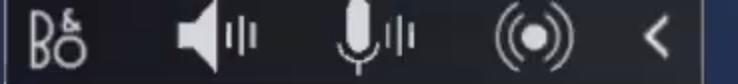




The background features a blue gradient with white abstract shapes. At the top left is a white square with a grid of dots. To its right is a large white triangle pointing upwards. In the bottom left corner, there are three concentric circles: a dark blue inner circle, a light blue middle circle, and a white outer circle.

**TERIMAKASIH**

-DREAM



Digital



# Python IV

## Advance Level

CC

Session 8 - Data Engineer

Tutor: M. Singgih Prastio

Session 6-Introduction to Git x +

File /Users/prakasab/OneDrive-DANAINDONESIA/NEW\_DANA/Documents/personal/DigitalSkola/DE16/SESI6-GIT/Session%206-Introduction%20to%20Git.pdf

2 / 15 | - 125% + | 🔍 ⚡ ⚡

Session 6-Introduction to Git.pdf



DigitalSkola  
Uncover The World Of Digital Skills With Us

# Table of Content

## What will We Learn Today?

- 1. What & Why Git?
- 2. Git Terms.
- 3. Github.
- 4. Git Config.
- 5. Create A Repository.
- 6. Basic Git Command.
- 7. Workflow in Git.
- 8. Git Use Cases.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

BAT 1% F -9:08 SSD 91% 214 KB/s RAM 83% CPU 26%

Github - Google Slides (2) WhatsApp

docs.google.com/presentation/d/1AwRkwn2wj0b223RjdEBzuy1lrjPKOyo5\_bxy7kdg1JM/edit#slide=id.p2

1 2 3 4 5 6 7 8 9

Bagas Prakasa

DigitalSkola Uncover The World Of Digital Skills With Us

**Table of Content**

**What will We Learn Today?**

1. Introduction to GitHub  
2. Communicate using Markdown  
3. GitHub Pages  
4. Review pull requests  
5. Resolve merge conflicts  
6. Release-based workflow

© Copyright by Digital Skola 2021

Session 10  
(28/02/2024)

The slide is part of a session titled 'Session 10' on '28/02/2024'. It displays a table of contents and a list of learning objectives. The sidebar shows nine steps, each with a screenshot of a digital skill course. The main area features a large orange question mark icon and a digital illustration of two people working on computers.

3 WhatsApp Post Attendee - Zoom Session 11 - Project 1

imminent-locust-045.notion.site/Session-11-Project-1-ff461a542a114f22914abab4c090f521

Labirin Urra DigitalSkola Intip DJP My Noted Dummy sample rest... API publik PIKOBAR... GraphQL and REST... XPath online real-ti... Starred GDrive Biznet Gi...

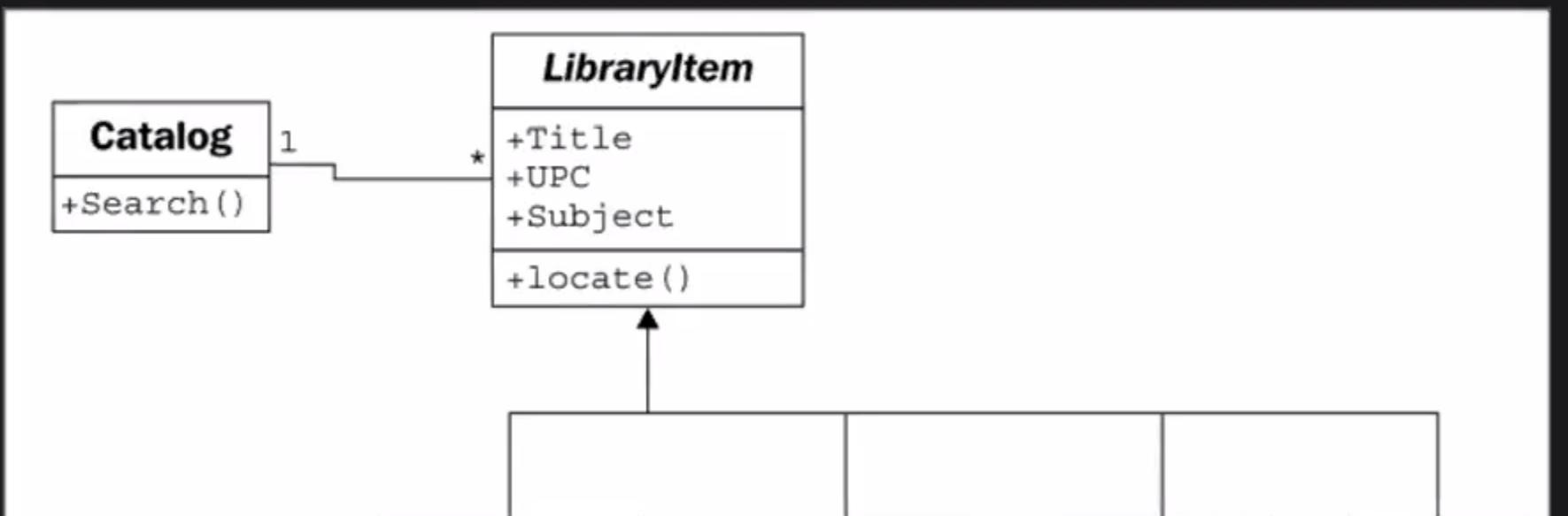
Muhammad Singgih Search Duplicate Open in notion

# Session 11 - Project 1

Requirement tools:

- Python ≥ 3.8 <https://www.python.org/downloads/>
- Visual Studio Code <https://code.visualstudio.com/download>
- Git <https://git-scm.com/downloads>
- Github <https://github.com/>
- SourceTree <https://www.sourcetreeapp.com/>

Create OOP:



File Edit Selection View Go Run ... ⏪ ⏩ 🔍 homework1 ⏷ ⏸ ⏹ ⏺ ⏻ ⏴

EXPLORER

HOMEWORK1

myhomework

homework.py

test

test\_homework.py

LICENSE

README.md

setup.py

tox.ini

setup.py > ...

```
25
26     setup(
27         name='homework1',
28         description="test package",
29         long_description=long_description,
30         long_description_content_type='text/markdown',
31         author="zyah",
32         author_email='nrfauh@xxx.com}',
33         package_dir={'': 'src'},
34         packages=find_packages(where='src'),
35         python_requires='>=3.6',
36         install_requires=install_requires,
37         extras_require={'dev': extra_requires_dev},
38         include_package_data=True
39     )
40
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Please avoid running `setup.py` and `easy\_install`. Instead, use pypa/build, pypa/installer or other standards-based tools.

See <https://github.com/pypa/setuptools/issues/917> for details.

```
!!
easy_install.initialize_options(self)
error: error in 'egg_base' option: 'src' does not exist or is not a directory
[end of output]
```

**note:** This error originates from a subprocess, and is likely not a problem with pip.

C:\Users\Fauziah\homework1\

Ln 15, Col 21 Spaces UTF-8 CRLF Python 3.12.1 64-bit Prettier

Mentor - Agung

Rico - Digital Skola

A\_Nicholas R S R Sirait

A\_Vina Alvina herawati

A\_Nur Fauziah

A\_Nur Fauziah

A\_Muhsyi Fadhilah Hirani

A\_froh Fauziah

A\_froh Fauziah

Session 12  
(01/03/2024)

Session 11 - Project 1

Create Automation:

The diagram illustrates a process flow for a library search system. It starts with a User performing a Search action on a Catalog. The Catalog then performs a Locate action to find a Bunnies Book. Simultaneously, the Catalog performs another Locate action to find a Bunnies DVD. Finally, the Catalog returns Results to the User. The process involves three objects: Catalog, Bunnies Book, and Bunnies DVD.

```
graph TD; User((User)) -- "Search" --> Catalog["Catalog"]; Catalog -- "Locate" --> Book["Bunnies Book"]; Catalog -- "Locate" --> DVD["Bunnies DVD"]; Catalog -- "Results" --> User;
```

References:

<https://subscription.packtpub.com/book/programming/9781784398781/1/ch01lvl1sec15/case-study>

