



W E B P R O G R A M M I N G P Y T H O N

Silabus

- Python programming
- OOP & Introduction to Django
- HTML,CSS programming
- Django Admin
- Database and CRUD Operation
- Create Home Page
- Implementing Bootstrap 5 for styling UI
- Wrapping UP

Python Basic

- IDLE (Integrated Development and Learning)
- Visual Studio Code
- Tipe Data (Integer, Float, Strings)
- Built-in function
- Membuat Function
- Operator Matematika
- Variabel
- List, Tuples, Dictionaries
- Percabangan
- Perulangan
- Object Oriented Programming



Sejarah Python

- Diciptakan Guido van Rossum seorang pria berkebangsaan Belanda
- mulai mengembangkan bahasa pemrograman Python tersebut sejak tahun 1989 di Centrum Wiskunde & Informatica (CWI) dan akhirnya bahasa pemrograman Python secara resmi dirilis kepada publik pada tahun 1991.
- Nama Python sendiri sebenarnya bukan diambil dari nama ular piton.
- Nama Python tersebut diambil dari sebuah acara humor yang ditayangkan oleh channel BBC pada tahun 1980-an yang berjudul "Monty Python's Flying Circus"
- Pada tahun 1994, akhirnya Python versi 1.0 dirilis kepada publik

<https://diskominfo.sp.lebakkab.go.id/sejarah-singkat-bahasa-pemrograman-python/>

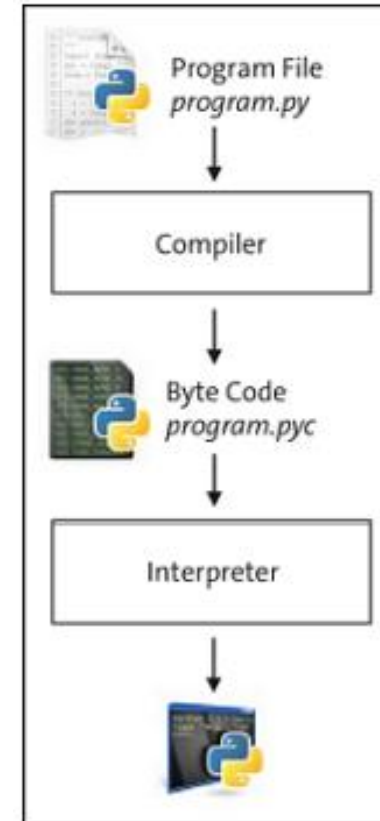
Implementasi Python

- Pengembangan Website
- Pengembangan IOT
- Data Mining
- Pengembangan Machine Learning
- Pengembangan game

<https://glints.com/id/lowongan/pentingnya-python/#.Y4Oq2XZBzIU>

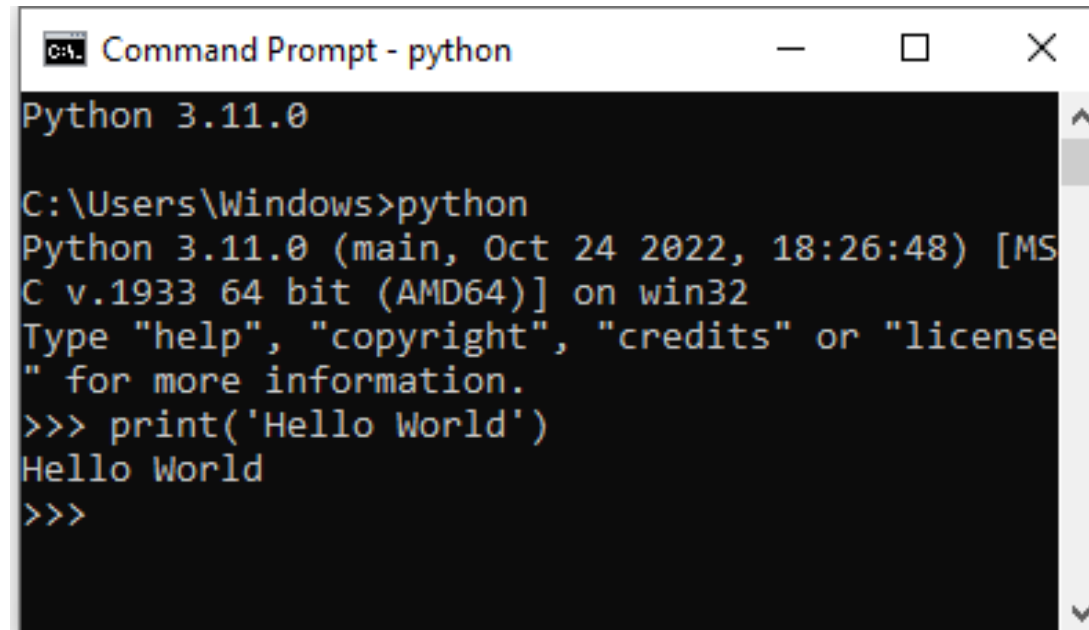
Internal Proses

- Kode python (.py)
- Dicompile menjadi bytecode disimpan di dalam memory atau dalam file .pyc
- Interpreter melakukan eksekusi bytecode sesuai OS



IDLE (Integrated Development and Learning)

- *Interactive Mode*

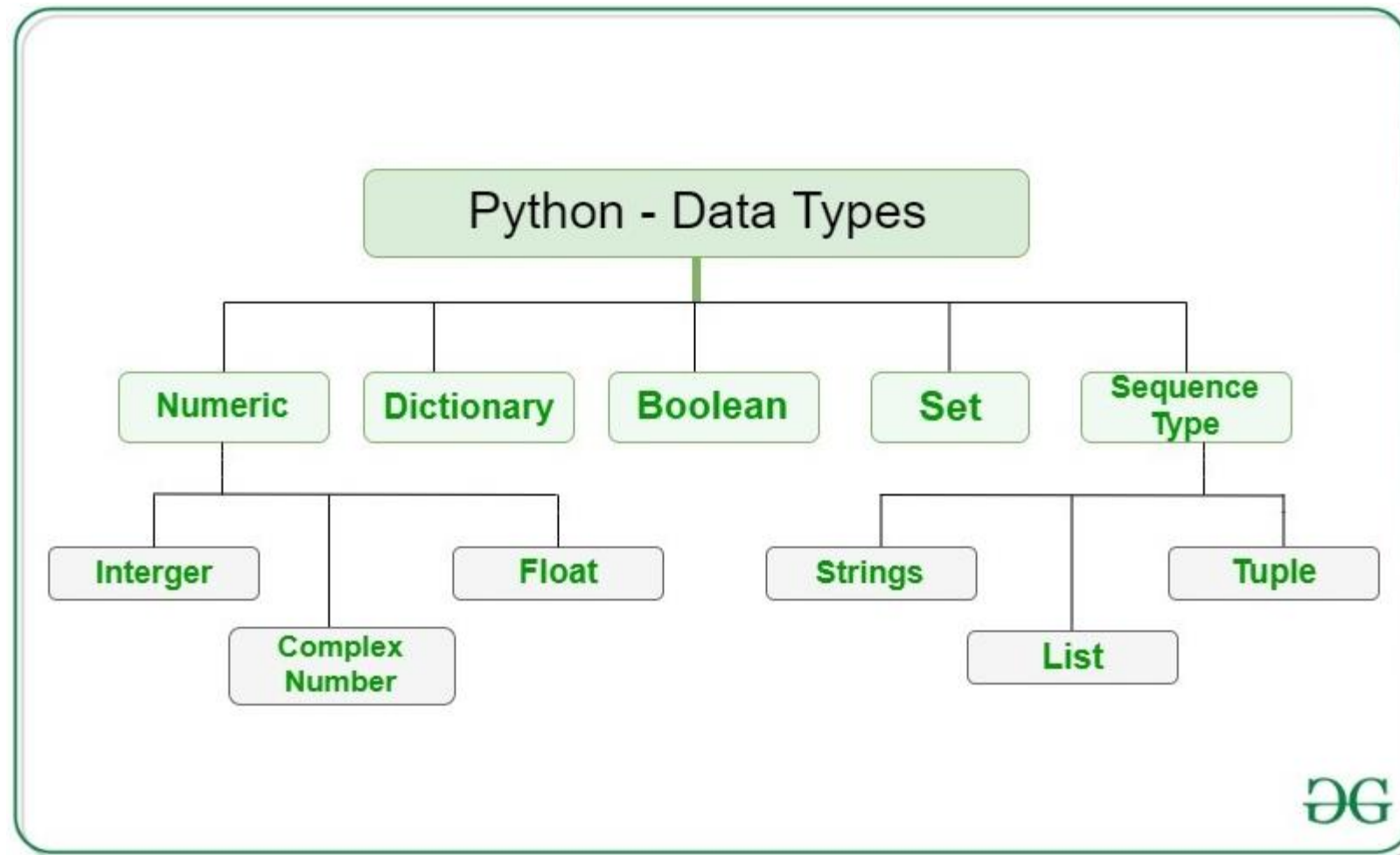


```
Command Prompt - python

Python 3.11.0

C:\Users\Windows>python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC
v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license
" for more information.
>>> print('Hello World')
Hello World
>>>
```

Tipe Data



<https://media.geeksforgeeks.org/wp-content/uploads/20191023173512/Python-data-structure.jpg>

Integers

Integers (int) — an integer number to represent an object such as “number 3”.

Integers -2, -1, 0, 1, 2, 3, 4, 5

```

Command Prompt - python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MS
C v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license
" for more information.
>>> 1 + 1
2
>>> 1 * 1
1
>>> 1 * (1 + 1)
2
>>>
  
```

Operator

1. Arithmetic Operators
2. Relational Operators
3. Bitwise Operators
4. Assignment operators
5. Identity operators
6. Membership operators

<https://makemeanalyst.com/python-programming/operators-and-expressions/>

Arithmetic/Math Operator

Math Operators

For reference, here's a list of other math operations you can apply towards numbers:

Operators	Operation	Example
<code>**</code>	Exponent	<code>2 ** 3 = 8</code>
<code>%</code>	Modulus/Remainder	<code>22 % 8 = 6</code>
<code>//</code>	Integer division	<code>22 // 8 = 2</code>
<code>/</code>	Division	<code>22 / 8 = 2.75</code>
<code>*</code>	Multiplication	<code>3 * 3 = 9</code>
<code>-</code>	Subtraction	<code>5 - 2 = 3</code>
<code>+</code>	Addition	<code>2 + 2 = 4</code>

[Python Cheat Sheet \(Updated in 2021\) | websitesetup.org](https://www.websitesetup.org/2021/02/python-cheat-sheet/)

Relational Operators

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Logical Operators

Python - Logical Operators

- not

x	not x
False	True
True	False
- and

x	y	x and y
False	False	False
False	True	False
True	False	False
True	True	True
- or

x	y	x or y
False	False	False
False	True	True
True	False	True
True	True	True

Operator Priority

<http://inderpsingh.blogspot.com/>

<https://inderpsingh.blogspot.com/2019/09/PythonTutorial5.html>

Bitwise Operators

Operator	Description
&	Bitwise AND
	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
>>	Bitwise right shift
<<	Bitwise left shift

Assignment operators

Operator	Example	Equals To
=	a = 10	a = 10
+=	a += 10	a = a+10
-=	a -= 10	a = a-10
*=	a *= 10	a = a*10
/=	a /= 10	a = a / 10
%=	a %= 10	a = a % 10
//=	a //= 10	a = a // 10
**=	a **= 10	a = a ** 10
&=	a &= 10	a = a & 10
=	a = 10	a = a 10
^=	a ^= 10	a = a ^10
>>=	a >>= 10	a = a >> 10
<<=	a <<= 10	a = a << 10

Identity operators

Operator	Description
is	It returns true if two variables point the same object and false otherwise
is not	It returns false if two variables point the same object and true otherwise

Membership operators

Operator	Description
in	It returns true if value/variable is found in the sequence and false otherwise
not in	It returns true if value/variable is not found in the sequence and false otherwise

Floats

Floating-point numbers (float) — use them to represent floating-point numbers.

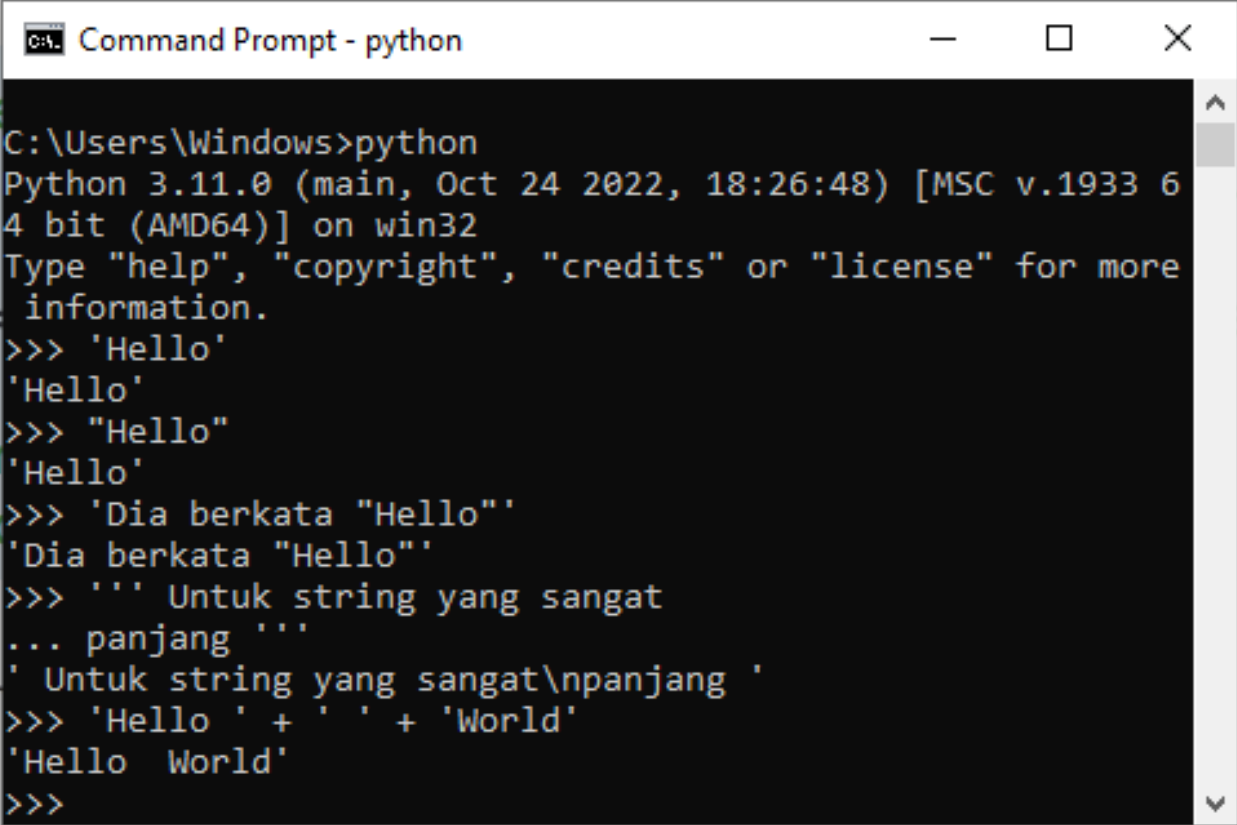
Floating-point numbers `-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25`

```

C:\Users\Windows>python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1
933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for
more information.
>>> 3/2
1.5
>>> 3//2
1
>>> 3.14 * 4 * 4
50.24
>>>
  
```

Character String/String

- Sequence of character
- Immutable



```
C:\Users\Windows>python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 6
4 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
>>> 'Hello'
'Hello'
>>> "Hello"
'Hello'
>>> 'Dia berkata "Hello"'
'Dia berkata "Hello"'
>>> ''' Untuk string yang sangat
... panjang '''
' Untuk string yang sangat\npanjang '
>>> 'Hello ' + ' ' + 'World'
'Hello World'
>>>
```

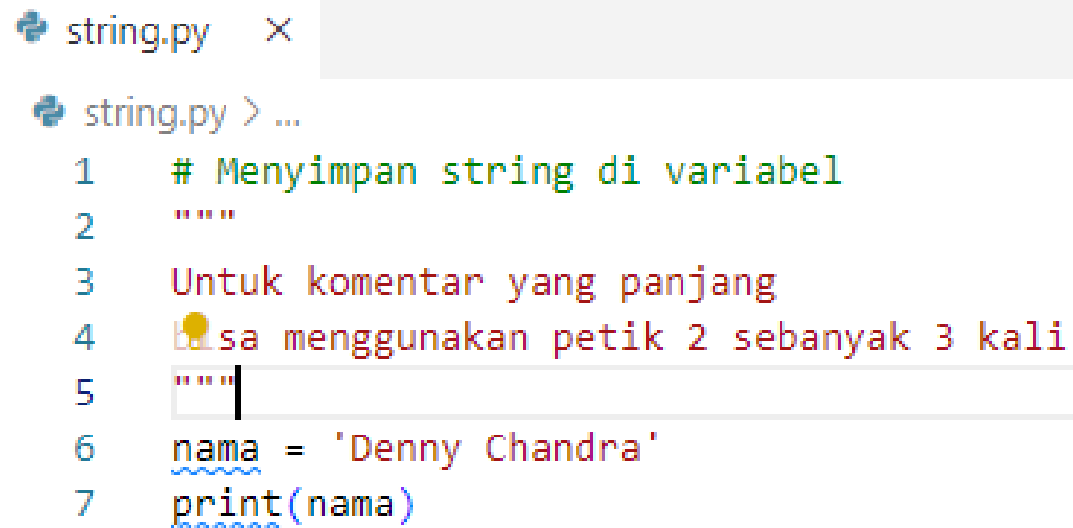
Latihan

- Buat program menggunakan console menghitung luas segitiga dengan alas dan tinggi

```
<>>> print(f"luas segitiga dari {alas} dan tingg {tinggi} adalah {luas}")
luas segitiga dari 10.0 dan tingg 5.0 adalah 25.0
```
- Buat program menggunakan console menghitung luas persegipanjang : $\text{luas} = \text{panjang} * \text{lebar}$

Menggunakan Visual Studio Code

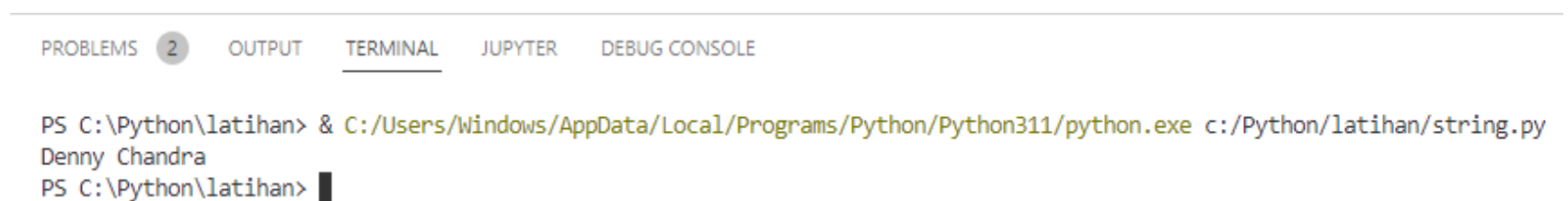
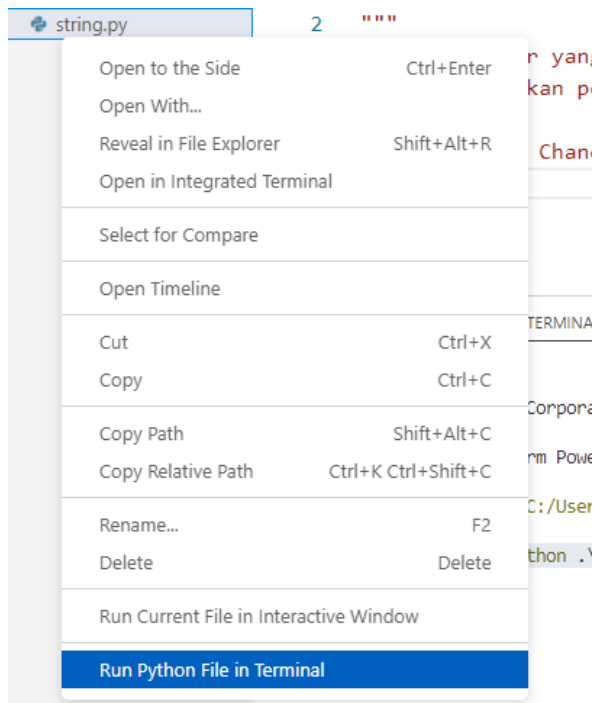
- Buat folder 'latihan'
- Di Visual Studio Code: File -> open folder 'latihan'
- Buat file string.py



```
string.py ×
string.py > ...
1  # Menyimpan string di variabel
2  """
3  Untuk komentar yang panjang
4  bisa menggunakan petik 2 sebanyak 3 kali
5  """
6  nama = 'Denny Chandra'
7  print(nama)
```

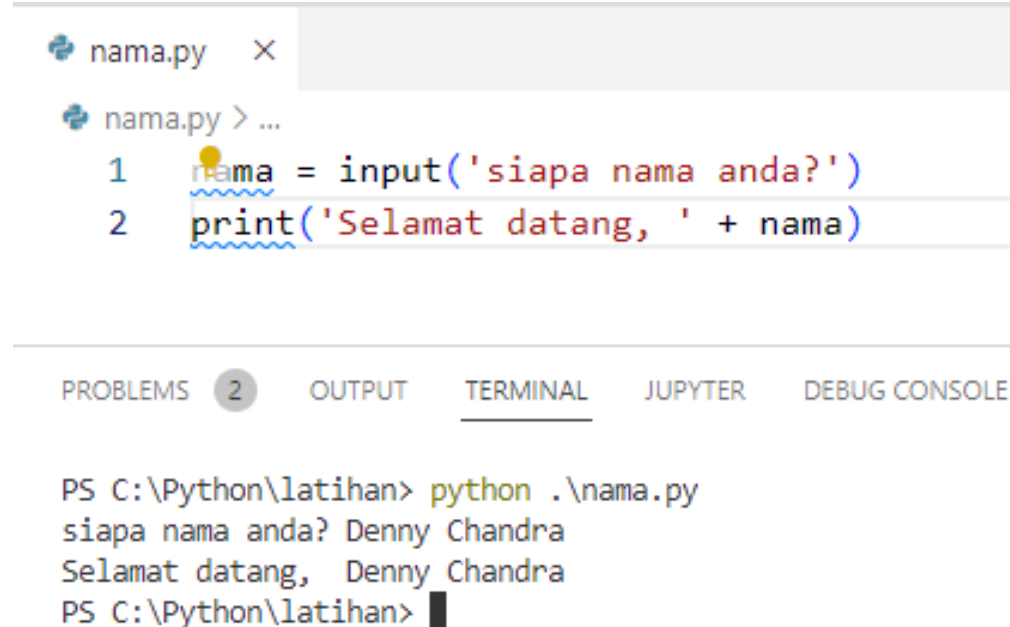
Menggunakan Visual Studio Code

- Terminal
- Ketik : `python .\string.py`
- Atau klik kanan file -> Run Python File in Terminal



Built-in Function

- Function = Bagian kode yang dapat digunakan kembali yang dibuat untuk menjalankan tindakan yang ditentukan
- `print()`
- `input()`



```
nama.py  X
nama.py > ...
1  nama = input('siapa nama anda?')
2  print('Selamat datang, ' + nama)

PROBLEMS 2 OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

PS C:\Python\latihan> python .\nama.py
siapa nama anda? Denny Chandra
Selamat datang, Denny Chandra
PS C:\Python\latihan> 
```

Latihan

- Buat program menggunakan Visual Studio Code untuk menghitung luas segitiga dengan

```
>>> print(f"luas segitiga dari {alas} dan tingg {tinggi} adalah {luas}")
luas segitiga dari 10.0 dan tingg 5.0 adalah 25.0
```

- Buat program menggunakan Visual Studio Code untuk console menghitung luas persegi panjang : $luas = panjang * lebar$

Built-in Function

Built-in Functions			
A abs() aiter() all() any() anext() ascii()	E enumerate() eval() exec()	L len() list() locals()	R range() repr() reversed() round()
B bin() bool() breakpoint() bytearray() bytes()	F filter() float() format() frozenset()	M map() max() memoryview() min()	S set() setattr() slice() sorted() staticmethod() str() sum() super()
C callable() chr() classmethod() compile() complex()	G getattr() globals()	N next()	T tuple() type()
D delattr() dict() dir() divmod()	H hasattr() hash() help() hex()	O object() oct() open() ord()	V vars()
	I id() input() int() isinstance() isinstance() iter()	P pow() print() property()	Z zip()
			_ __import__()


<https://docs.python.org/3.11/library/functions.html>

Built-in Function

```
# len()
nama = 'Denny Chandra'
print("Panjang string", len(nama))
#filter
umur = [5, 12, 17, 18, 24, 32]
def myFunc(x):
    if x < 18:
        return False
    else:
        return True
dewasa = filter(myFunc, umur)
print("kategori dewasa : ")
for x in dewasa:
    print(x)
```

- Methods = function di dalam kelas

```
# Basic Python method
class class_name
    def method_name () :
        .....
        # method body
        .....
```



String Operations

Python

Created by Hanzel Godinez
@GodinezHanzel

General operations

split	Returns a list of substrings in the string
isalpha	Returns True or False if a string containing just alphabetic characters
isdigit	Returns True or False if a string containing just numbers
*	Replicates a string

Capitalization operations

upper	Converts a string to uppercase
lower	Converts a string to lowercase
title	Capitalizes the first letter of each word in string
capitalize	Capitalize converts only the first character of a string to uppercase

Concatenation operations

join	Takes a list of strings and puts them together to form a single string
+	Add two strings together, less efficient than join method

Searching operations

find	Searches for the target in a string, returns -1 if substring does not exist in the string
index	Searches for the target in a string, raises a ValueError exception if substring does not exist in the string
rfind	Searches for the target in a string, from right to left or from last position to 0 position
rindex	Searches for the target in a string, from right to left or from last position to 0 position
startswith	Checks the beginning of a string for a match
endswith	Checks the end of a string for a match
replace	Replaces the target with a new string
strip	Returns a new string without any whitespace or other characters at the beginning or end of the string.
rstrip	Returns a new string without any whitespace or other characters at the end of the string

Built-in Methods

#string method

```
nama = " denny chandra "  
print("upper",nama.upper())  
print("capitalize",nama.capitalize())  
print("title",nama.title())  
value = "100"  
print("isdigit",value.isdigit())  
print("isaplha","abcdef".isalpha())
```

- Modul Python adalah file yang berisi definisi dan statemen Python.
- Sebuah modul dapat mendefinisikan fungsi, kelas, dan variabel.
- Modul juga dapat menyertakan kode yang dapat dijalankan.
- Pengelompokan kode terkait ke dalam modul membuat kode lebih mudah dipahami dan digunakan.
- Itu juga membuat kode terorganisir secara logis.

Contoh menggunakan Modules : array

- Modul ini mendefinisikan tipe objek yang secara ringkas dapat mewakili array nilai dasar: characters, integers, angka floating point.
- Array sangat mirip dengan lists, kecuali bahwa jenis objek yang disimpan di dalamnya dibatasi.
- Tipe ditentukan pada waktu pembuatan objek dengan menggunakan kode tipe

```
from array import array  
myarray = array('i', [5, 6, 7, 2, 3, 5])  
print(myarray)
```

Type Pada Array

Type code	C Type	Python Type	Minimum size in bytes	Notes
'b'	signed char	int	1	
'B'	unsigned char	int	1	
'u'	wchar_t	Unicode character	2	(1)
'h'	signed short	int	2	
'H'	unsigned short	int	2	
'i'	signed int	int	2	
'I'	unsigned int	int	2	
'l'	signed long	int	4	
'L'	unsigned long	int	4	
'q'	signed long long	int	8	
'Q'	unsigned long long	int	8	
'f'	float	float	4	
'd'	double	float	8	

Membuat Modul

```
# A simple module, calc.py
```

```
"""add x,y."""
```

```
def add(x, y):  
    return (x+y)
```

```
"""subtract x,y."""
```

```
def subtract(x, y):  
    return (x-y)
```

```
#test_calc.py
```

```
# importing module calc.py
```

```
import calc
```

```
print(calc.add(10, 2))
```

- Python memungkinkan untuk merangkum beberapa modul dalam satu package.
- Merupakan best practice jika modul-modul ini disatukan secara tematis.
- Berbeda dengan modul, package dapat berisi sejumlah package lain, yang dapat berisi modul atau package.
- Untuk membuat package, subfolder harus dibuat dalam directory
- Nama folder harus sesuai dengan nama package
- Folder ini dapat berisi file program bernama `__init__.py`
- File ini berisi kode inisialisasi yang dieksekusi sekali ketika package diimpor.

```
|— admin
|   |— __init__.py
|   |— company_admin.py
|   |— kiosk_admin.py
|   |— pages_admin.py
|   |— screens_admin.py
|— forms
|   |— __init__.py
|   |— company_form.py
|   |— kiosk_form.py
|   |— pages_form.py
|   |— screens_form.py
|— models
|   |— __init__.py
|   |— company.py
|   |— kiosk.py
|   |— pages.py
|   |— screens.py
|— views
|   |— __init__.py
|   |— company_view.py
|   |— kiosk_view.py
|   |— pages_view.py
|   |— screens_view.py
```

- Digunakan untuk *ordered sequence of elements*
- Singkatnya, list ini untuk membantu menyimpan data terkait dan melakukan operasi yang sama pada beberapa nilai sekaligus.
- Tidak seperti string, lists sifatnya mutable (bisa berubah)

```
my_list = [1, 2, 3]
my_list2 = ["a", "b", "c"]
my_list3 = ["4", 2, "book", 5]
alpha_list = list((1, 2, 3))
print(alpha_list)
```

Python List Methods	
Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

<https://idkuu.com/list-1-2-3-4-if-list-2-0-printlist>

```
my_list = [1, 3, 4, 2]
my_list.append(5)
print(my_list)
my_list.reverse()
print(my_list)
my_list.sort()
print(my_list)
my_list.sort(reverse=True)
print(my_list)
#combine lists
my_list2 = ["a", "b", "c"]
combo_list = my_list + my_list2
print(combo_list)
combo_list.sort()
```

- Digunakan untuk *ordered sequence of elements*
- Singkatnya, list ini untuk membantu menyimpan data terkait dan melakukan operasi yang sama pada beberapa nilai sekaligus.
- Tidak seperti lists, tuple sifatnya immutable (tidak bisa berubah)
- Operasinya lebih cepat dari lists

```
my_tuple = (0,1,2,3,4,5,6,7,8,9,10)
print(my_tuple)
print(my_tuple[0])
print(my_tuple[0:3])
print(my_tuple[0:10:2])
```

```
PS C:\Python\latihan> python .\tuple.py
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
0
(0, 1, 2)
(0, 2, 4, 6, 8)
PS C:\Python\latihan> █
```


- Dictionaries menyimpan indeks dengan kunci yang dipetakan ke nilai tertentu.
- Sifatnya mutable

```
new_dict = {  
    "brand": "Honda",  
    "model": "Civic",  
    "year": 1995  
}  
print(new_dict)  
new_dict["year"] = 2000  
print(new_dict)  
#menampilkan berdasarkan key  
print(new_dict["brand"])
```

- If
- Elif
- If Else
- If Not

```
nilai = 50
#if
if nilai > 60:
    print("lulus")
#elif, utk kondisi lain
elif nilai == 60:
    print("remedial")
#jika tidak ada di kondisi diatas
else:
    print("sampai jumpa di semester pendek")

new_list = [1, 2, 3, 4]
x = 1
if x not in new_list:
    print("'x' bukan di list, so this is True!")
```

- Tidak ada bentuk switch case
- Alternati menggunakan Dictionary Mapping

```
# This code runs only in python 3.10 or above versions
```

```
def number_to_string(argument):  
    match argument:  
        case 0:  
            return "zero"  
        case 1:  
            return "one"  
        case 2:  
            return "two"  
        case default:  
            return "something"
```

```
result = number_to_string(0)  
print(result)
```

- while

```
# while loop
count = 0
while (count < 3):
    count = count + 1
    print("Hello Geek")
```

```
a = [1, 2, 3, 4]
while a:
    print(a.pop())
else:
    print("done")
```

Contoh Lain (while)

- Do-While

```
while True:
    # Code to be executed
    user_input = input("Enter 'q' to quit: ")
    if user_input == 'q':
        break
```

- While

```
user_input = input("Enter 'q' to quit: ")

while user_input != 'q':
    # Code to be executed
    user_input = input("Enter 'q' to quit: ")
```

- for

```
# Iterating over a list
```

```
l = ["a", "b", "c"]
```

```
for i in l:
```

```
    print(i)
```

```
#continue
```

```
for letter in 'abcdef':
```

```
    if letter == 'b' or letter == 'd':
```

```
        continue
```

```
    print('Current Letter without b,d :', letter)
```

```
#break
```

```
for letter in 'abcdef':
```

```
    if letter == 'd':
```

```
        break
```

```
    print('Current Letter before break :', letter)
```

```
#pass
```

```
for letter in 'abcdef':
```

```
    pass
```

```
print('last letter', letter)
```

```
new_dict = {  
    "brand": "Honda",  
    "model": "Civic",  
    "year": 1995  
}  
  
#print all key names in the dictionary  
for x in new_dict:  
    print(x)  
  
#print all values in the dictionary  
for x in new_dict:  
    print(new_dict[x])  
  
#loop through both keys and values  
for x, y in new_dict.items():  
    print(x, y)
```

- Buat program untuk menampilkan bilangan genap dari 2 s.d 100 menggunakan for dan range (range([start], stop, [step]))
- Buat program untuk menampilkan bilangan ganjil dari 99 s.d 1

- Hampir semua elemen di Python adalah Object
- Class = blue print untuk membuat object
- Object = instansiasi/perwujudan dari Class

OOP – membuat class

#person.py

```
class Person(object):  
    def __init__(self,nama,alamat):  
        self.nama = nama  
        self.alamat = alamat  
  
    def tampil(self):  
        print('nama = ',self.nama,'alamat = ',self.alamat)
```

OOP – membuat object

```
#test_person.py
```

```
from person import Person
```

```
person = Person("denny", "bandung")  
person.tampil()
```

OOP – membuat subclass

```
#siswa.py
from person import Person
class Siswa(Person):
    def __init__(self, nama, alamat, nim):
        super().__init__(nama, alamat)
        self.nim = nim

    def tampil(self):
        print('nama = ', self.nama, 'alamat = ', self.alamat, 'nim = ', self.nim)
```

OOP – membuat object dari subclass

```
#test_siswa.py
```

```
from siswa import Siswa
```

```
siswa = Siswa("denny", "bandung", "12345")  
siswa.tampil()
```

Latihan

- Buat class : Vehicle dengan property make,model,year dan 1 function start_engine.function start_engine melakukan print di console untuk menampilkan propertynya
- Buat class Car extends Vehicle, definisikan ulang function start_engine
- Buat class Motorcycle extends Vehicle, definisikan ulang function start_engine
- Buat object dari Car dan Motorcycle kemudian panggil function start_engine

Tanya Jawab, diskusi



<https://p92.hu/techdetails/html-css-and-javascript>

HTML, CSS, JavaScript

Istilah Pada Web

- **Halaman web (*web page*)**

Halaman yang digunakan untuk menampilkan informasi-informasi yang terdapat di web.

- **Situs web (*web site*)**

Sekumpulan halaman web milik seseorang, perusahaan atau instansi yang dikumpulkan dan diletakkan dalam web server.

- ***Homepage***

Istilah untuk menyebut halaman pertama yang akan muncul jika sebuah situs web diakses.

Istilah Pada Web

- ***Hyperlink*** atau ***Link***

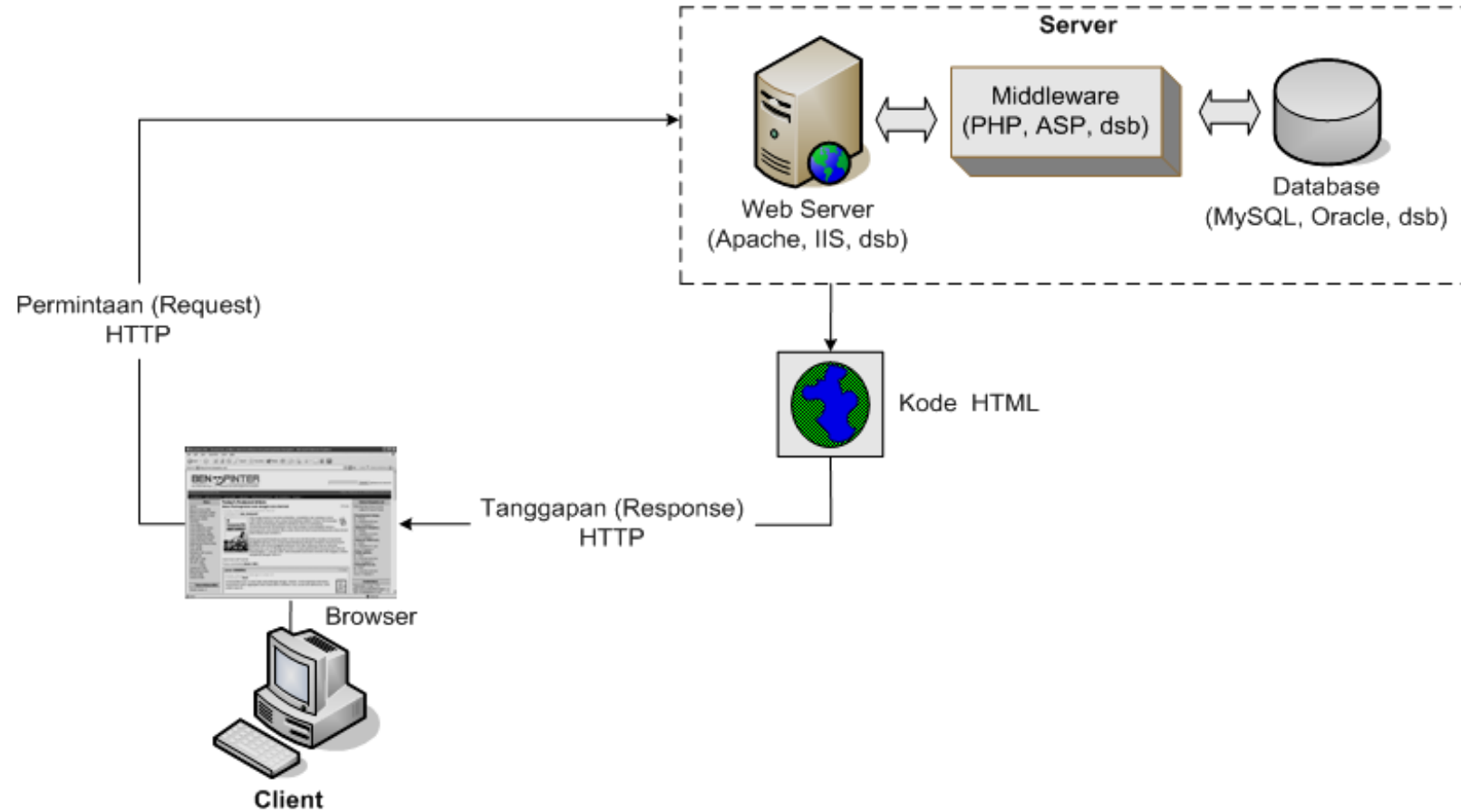
Penghubung antar halaman web.

- **URL (*Uniform Resource Locator*)**

Alamat suatu dokumen atau situs web yang bersifat unik dan khas, contoh :

- <http://www.google.com>
- <http://103.134.152.1>
- <https://telkompcc.co.id/neo>

Arsitektur Client-Server



Teknologi Web Client - Server

- **Teknologi Web Pada Sisi Client**

Pemrosesan kode program dilakukan di client.

Contoh : JavaScript, TypeScript, Angular, ReactJS, VueJS

- **Teknologi Web Pada Sisi Server**

Pemrosesan kode program dilakukan di server.

Contoh : PHP, JSP, ASP, NodeJS, Python, Java Servlet

HTML (Hypertext Markup Language)

- HTML Merupakan bahasa standard yang di gunakan untuk membuat dokumen web .
- Kemampuan HTML :
 - Membuat tampilan dari halaman web.
 - Membuat form.
 - Menambahkan objek-objek seperti gambar, audio, video, animasi, dll.

Tag HTML

- Tag adalah kode-kode yang digunakan untuk men-setting dokumen web.
- Nama tag ditulis di dalam tanda “<” dan “>” dan untuk tag penutup ditambahkan tanda “/”.
- Boleh ditulis dengan huruf besar maupun kecil.
- Digunakan sepasang, contoh : **<form>** dan **</form>**.
- Ada tag yang tidak perlu sepasang, contoh : **
, **<hr/>, ****, dll.

Struktur HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Title</title>
  <meta name="Keywords" content="" />
  <meta name="Description" content="" />
  <link rel="icon" href="" type="image/x-
icon"/>
</head>
<body>
  <div>Hello World</div>
</body>
</html>
```

Struktur HTML

- Tag **<html>...</html>** berfungsi untuk menyatakan bahwa dokumen tersebut adalah dokumen HTML.
- Tag **<head>...</head>** berisi kode-kode untuk menuliskan keterangan/informasi tentang dokumen HTML tersebut.
- Tag **<title>...</title>** digunakan untuk membuat judul halaman Web (*web page*).
- Tag **<body>...</body>** berisi tag-tag untuk isi atau layout tampilan halaman web.
- Tag **<meta>** mendefinisikan metadata tentang dokumen HTML. Metadata adalah data (informasi) tentang data.

CSS (Cascading Style Sheet)

- CSS adalah kumpulan format yang digunakan untuk mengatur tampilan atau isi dari sebuah halaman web menjadi lebih fleksibel.

- Elemen dasar CSS :

1. Selector

Adalah tag/elemen normal pada HTML

2. Deklarasi

Adalah satu atau beberapa perintah/nilai dari CSS yang menunjukkan type/bentuk yang diaplikasikan pada selector.

Pengkodean CSS

- Inline style

```
<div style="font-family: verdana;  
font-size: 12px">  
  Hello World  
</div>
```

- Internal style

```
<head>  
<style type="text/css">  
  div { font-family: verdana; font-size:  
    20px }  
  body { background-color: yellow }  
</style>  
</head>
```

Pengkodean CSS

- Eksternal style

```
<head>
```

```
  <link
```

```
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLA SjC" crossorigin="anonymous">
```

```
</head>
```

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

Selector CSS

- Class menggunakan titik

```
.embedded{  
    font-size: 20px;  
    color: red;  
}
```

- Id menggunakan #

```
#div1{  
    font-size: 20px;  
    color: green;  
}
```

JavaScript

- JavaScript diciptakan oleh Brendan Eich pada tahun 1995, dan menjadi standar ECMA pada tahun 1997.
- ECMA-262 adalah nama resmi dari standar. ECMAScript adalah nama resmi dari bahasa tersebut.
- TypeScript adalah *strongly typed* programming language yang dibuat diatas JavaScript.

Library/Framework pada Front End

- JQuery = Library JavaScript
- Bootstrap = Framework HTML, CSS, Javascript
- Angular = Framework JavaScript
- ReactJS = Library JavaScript
- VueJS = Framework JavaScript
- Dan masih banyak lagi yang lainnya

What's the Difference?



HTML

Hypertext Markup Language

Create the structure

- Controls the layout of the content
- Provides structure for the web page design
- The fundamental building block of any web page



CSS

Cascading Style Sheet

Stylize the website

- Applies style to the web page elements
- Targets various screen sizes to make web pages responsive
- Primarily handles the "look and feel" of a web page



Javascript

Increase interactivity

- Adds interactivity to a web page
- Handles complex functions and features
- Programmatic code which enhances functionality

<https://brytdesigns.com/html-css-javascript-whats-the-difference>

Tugas

- Buat 3 buah halaman html dalam 1 folder
 - index.html
 - about.html
 - contact.html
- Tambahkan style (slide 67)
- Buat navigasi dari index.html ke about.html dan ke contact.html dan sebaliknya
- Buat menjadi halaman profile masing-masing.
 - index.html adalah halaman utama
 - about.html mendeskripsikan dirimu
 - contact.html adalah kontakmu

Django makes it easier to build better web
apps more quickly and with less code.

Get started with Django

Django

- Django adalah framework web open source gratis untuk membangun Aplikasi web berbasis Python yang modern.
- Django membantu developer dengan cepat dalam membangun aplikasi web dengan mengabstraksikan banyak pekerjaan yang berulang dalam membangun situs web, seperti menghubungkan ke database, menangani keamanan, mengaktifkan otentikasi pengguna, membuat rute URL, menampilkan konten pada halaman melalui template dan form,
- Mendukung banyak basis data dan menyiapkan antarmuka admin.
- Dengan Django maka developer akan fokus kepada fungsional aplikasi daripada mengerjakan hal yang berulang-ulang dalam membuat web yang standar

Django

- Django merupakan salah satu framework yang banyak digunakan perusahaan besar seperti Instagram, Pinterest, Mozilla dan National Geographic, DropBox dan lainnya.
- Django sangat mudah untuk digunakan oleh start up untuk membangun sebuah project

<https://www.gamelab.id/news/810-7-website-terkenal-yang-dibuat-dengan-python-django#:~:text=django%20lebih%20dulu.-,Pengenalan%20Django,dan%20cepat%20untuk%20melakukan%20pengembangan.>

The Power Of Django

- Object-relational mapper
- Automatic admin interface
- Robust template system
- Quick internationalization

Instalasi Django

- Menggunakan command prompt ketik : **pip** (untuk mac **pip3**)
- pip adalah package manager untuk python package
- pip sudah otomatis ada di dalam python versi 3.4 ke atas

☞ Select Command Prompt

```
C:\Python\latihan>pip
```

Usage:

```
pip <command> [options]
```

Commands:

install	Install packages.
download	Download packages.
uninstall	Uninstall packages.
freeze	Output installed packages in requirements format.
inspect	Inspect the python environment.
list	List installed packages.
show	Show information about installed packages.
check	Verify installed packages have compatible dependencies.
config	Manage local and global configuration.

Instalasi Django

- Menggunakan command prompt ketik : **pip install Django==4.1.3**
- Kemudian cek versi django yang terinstall

Command Prompt - python

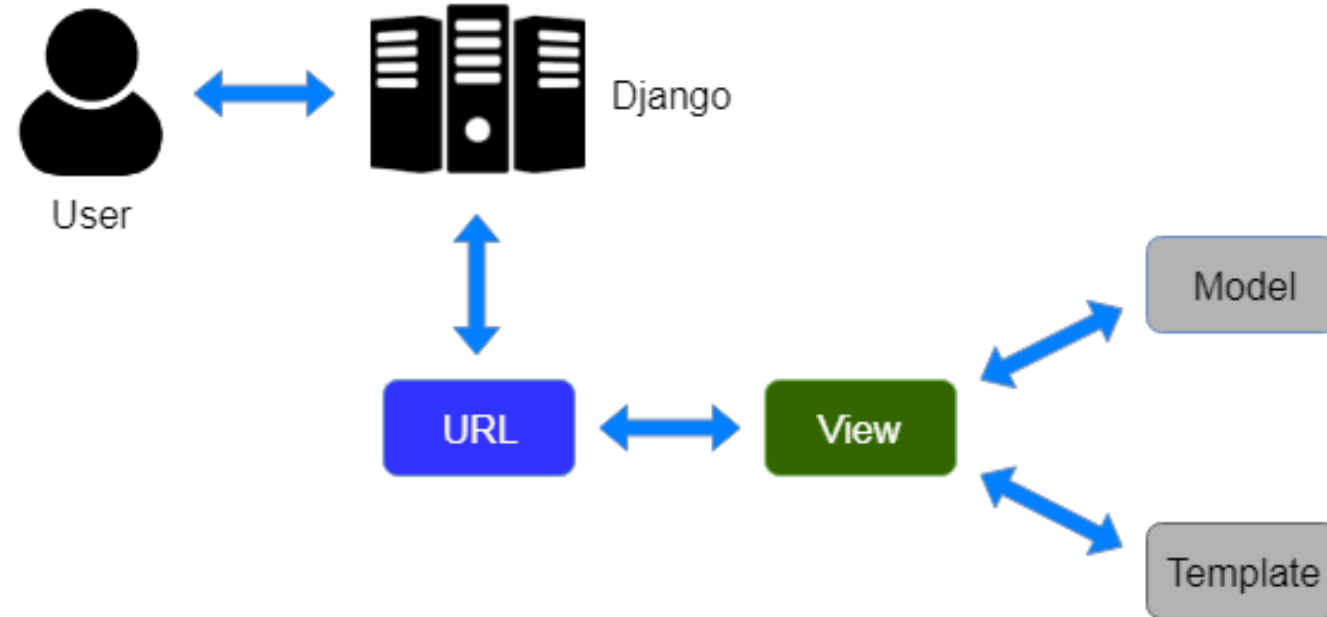
```
C:\Python\latihan>python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> print(django.get_version())
4.1.3
>>>
```

Django MVT

- MVT (Model View Template) adalah software design pattern.
- Ada tiga komponen penting yaitu Model, View dan Template.
- Model membantu menangani database, Template menangani UI (User Interface) dan View digunakan untuk menangani proses bisnis dan menghubungkan antara model dan template

<https://www.javatpoint.com/django-mvt>

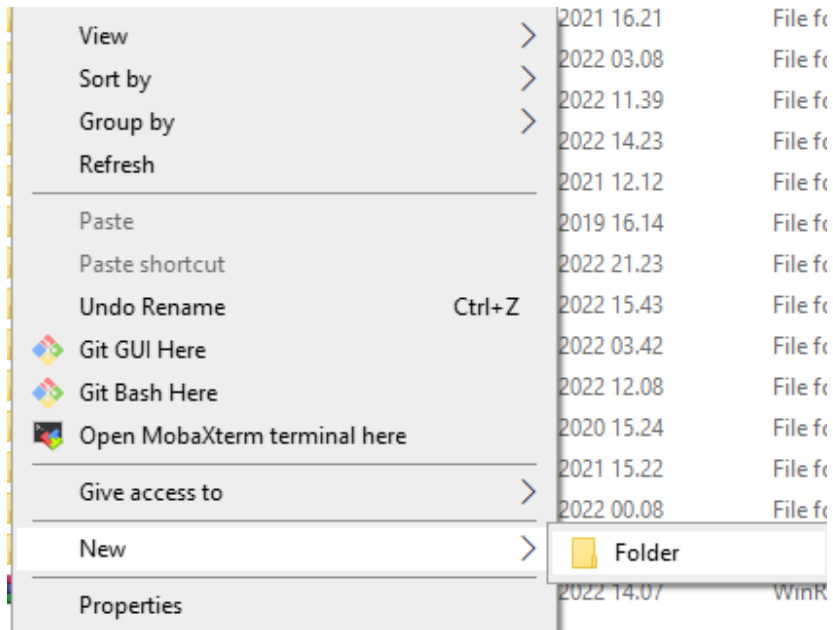
Django MVT



<https://www.javatpoint.com/django-mvt>

Membuat Project

- Buat folder di drive c atau drive lain misal c:\projectdjango
- Klik kanan di c , New -> Folder



- Rename jadi projectdjango 

Membuat Project

- Menggunakan command prompt arahkan ke folder tersebut

```
Command Prompt
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd..

C:\Users>cd..

C:\>cd projectdjango
```

- Ketik : **python -m django startproject bookreviews** (ini perintah untuk membuat project)




```
C:\projectdjango>python -m django startproject bookreviews

C:\projectdjango>_
```

Membuat Project

- Rename folder di c:\projectdjango\bookreviews menjadi c:\projectdjango\bookreviewspj
- Berikut ini strukturnya:

is PC > System (C:) > projectdjango > bookreviewspj

Name	Date modified	Type	Size
 bookreviews	29/11/2022 03.29	File folder	
 db.sqlite3	29/11/2022 03.29	SQLITE3 File	0 KB
 manage.py	29/11/2022 03.27	Python File	1 KB

Membuat Project

- Menggunakan command prompt menuju ke folder bookreviews : **cd bookreviewsproject**
- Ketik : **python manage.py runserver** (Ini perintah untuk menjalankan project)

```
C:\projectdjango>cd bookreviewsproject

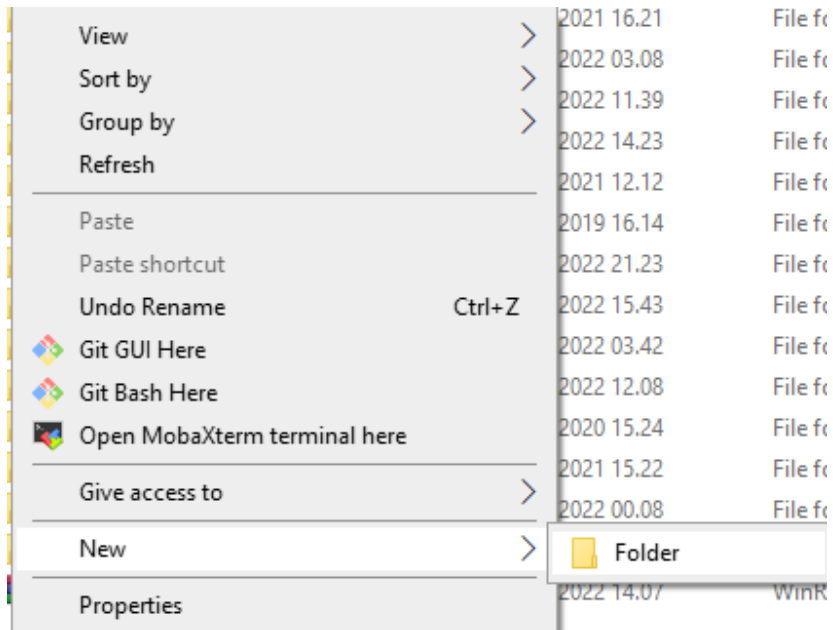
C:\projectdjango\bookreviewsproject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
November 29, 2022 - 03:43:05
Django version 4.1.3, using settings 'bookreviews.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Jika Membuat Project selain di C (misal di drive F)


- Buat folder di drive F atau drive lain misal F:\projectdjango
- Klik kanan di F , New -> Folder



- Rename jadi projectdjango 

Membuat Project (misal di F)

- Menggunakan command prompt arahkan ke folder tersebut

 Command Prompt

```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>F:

F:\>cd projectdjango
```

- Ketik : **python -m django startproject bookreviews**

```
F:\projectdjango>python -m django startproject bookreviews

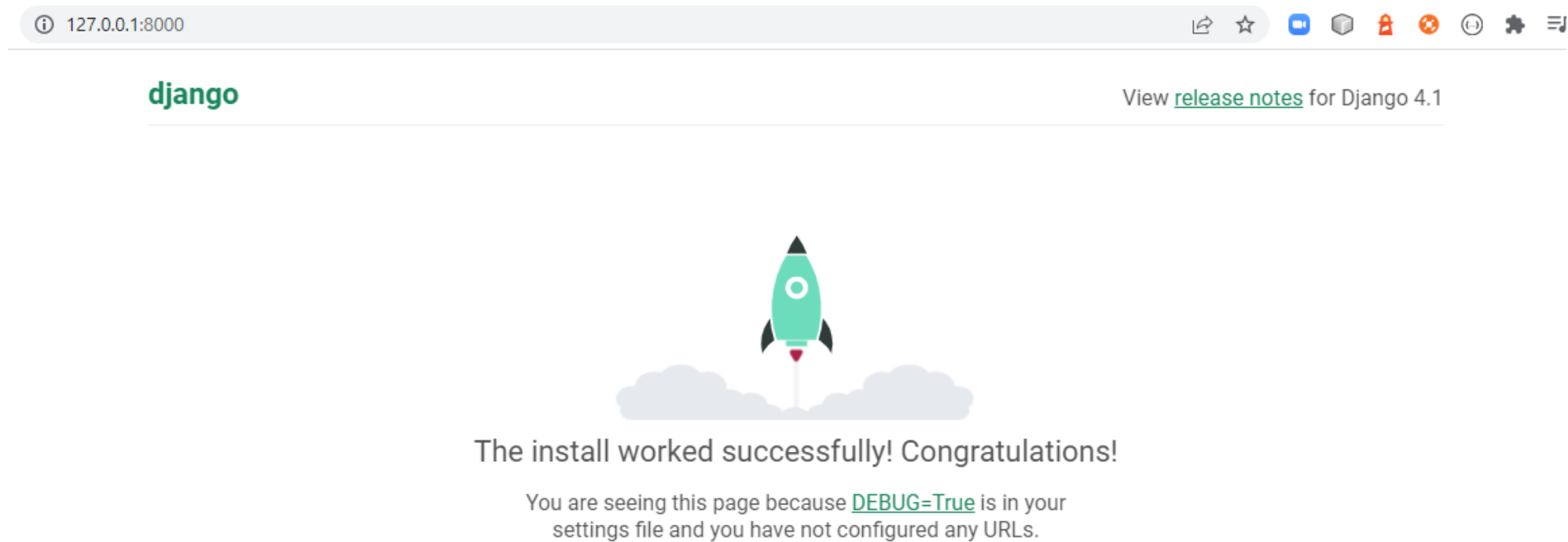
F:\projectdjango>
```

Membuat Project

- Rename folder di F:\projectdjango**bookreviews** menjadi F:\projectdjango**bookreviewsproject**
- Menggunakan command prompt menuju ke folder bookreviews : **cd bookreviewsproject**
- Ketik : **python manage.py runserver** (Ini perintah untuk menjalankan project)

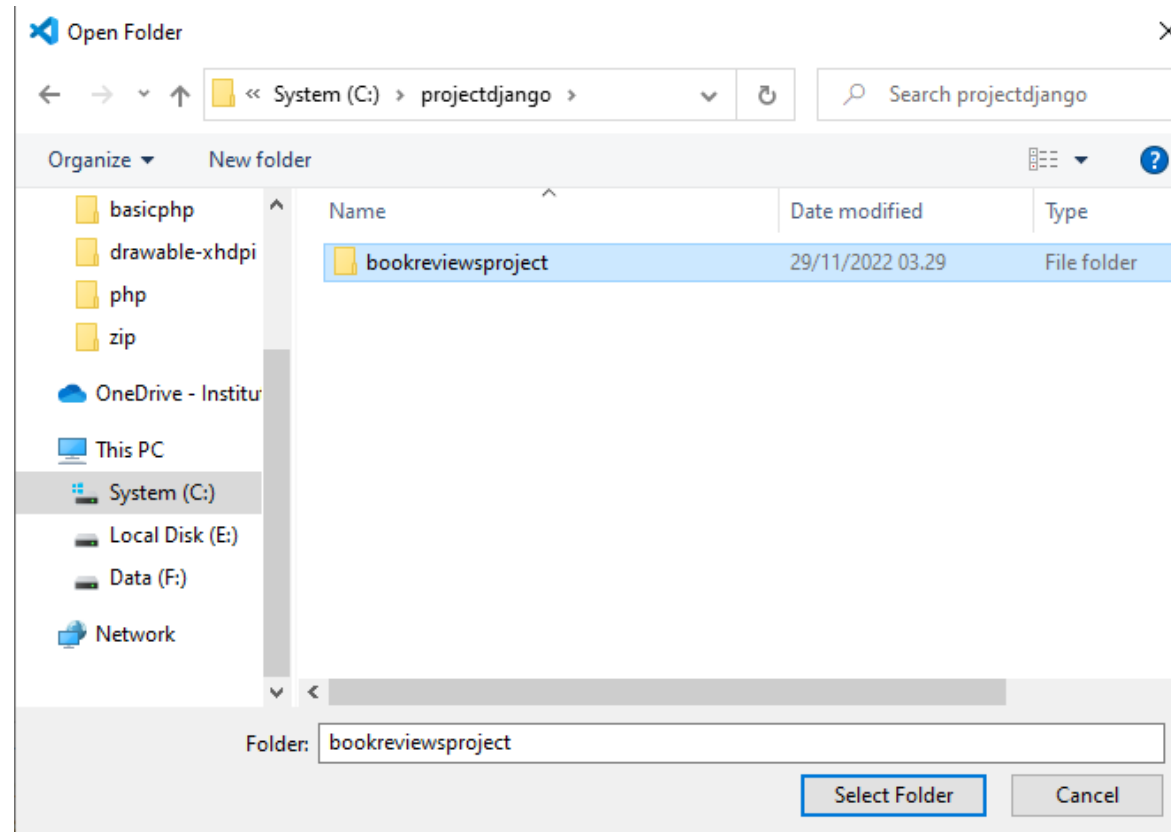
Membuat Project

- Buka chrome ketik : <http://127.0.0.1:8000>

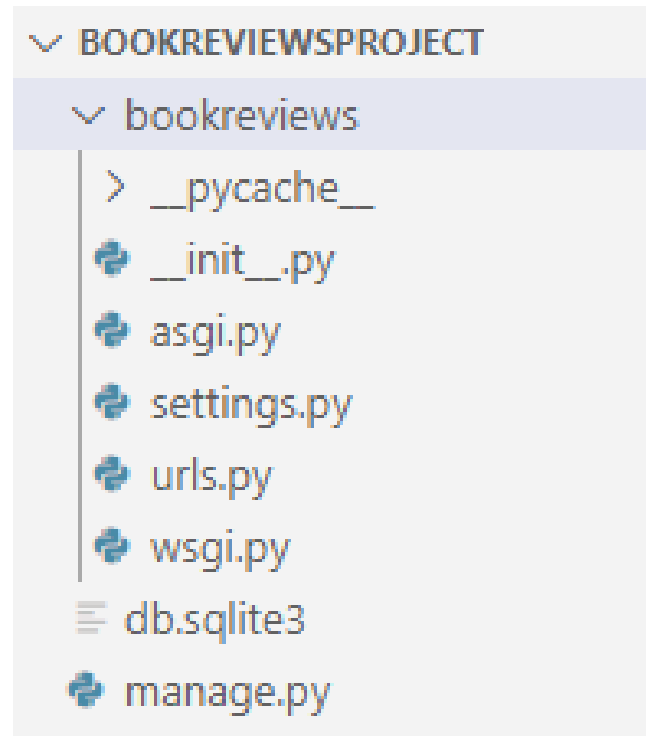


Visual Studio Code

- Buka Visual Studio Code
- File -> Open Folder arahkan ke folder bookreviewsproject
- Klik Select Folder



Struktur Direktori



Struktur Direktori

- **__pycache__**: Folder ini menyimpan bytecode yang dikompilasi saat kita membuat proyek. Kita dapat mengabaikan folder ini. Tujuan folder ini adalah untuk membuat proyek lebih cepat dijalankan dengan men-cache kode yang dikompilasi yang kemudian dapat dengan mudah dieksekusi.
- **__init__.py**: File ini menetapkan apa yang harus dijalankan ketika Django diluncurkan untuk pertama kali
- **asgi.py**: File ini memungkinkan Antarmuka Gateway Server Asynchronous opsional (ASGI) untuk dijalankan. ASGI menyediakan antarmuka antara server web dan kerangka kerja Python yang bekerja secara asinkron.

Struktur Direktori

- **settings.py:** File settings.py adalah file penting yang mengontrol pengaturan proyek. File ini berisi beberapa properti.
 - **BASE_DIR:** Menentukan lokasi proyek di drive komputer.
 - **SECRET_KEY:** Digunakan ketika kita memiliki data yang mengalir masuk dan keluar dari aplikasi web. Jangan pernah berbagi ini dengan orang lain.
 - **DEBUG:** Situs dapat berjalan dalam mode debug atau tidak. Dalam mode debug, kita mendapatkan informasi terperinci tentang kesalahan – misalnya, jika mencoba menjalankan `http://localhost:8000/123` di browser, kita akan melihat halaman tidak ditemukan(404)
 - **INSTALLED_APPS:** Tempat untuk modul/aplikasi di register di aplikasi web
 - **MIDDLEWARE:** Mengacu pada fungsi bawaan Django untuk memproses aplikasi request/response, yang mencakup otentikasi, session, dan keamanan.

Struktur Direktori

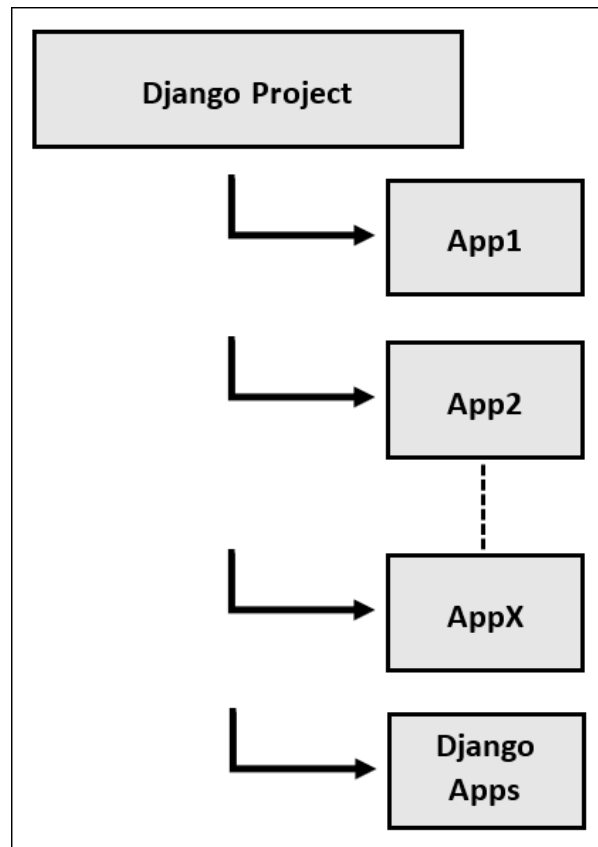
- **settings.py:** File settings.py adalah file penting yang mengontrol pengaturan proyek. File ini berisi beberapa properti.
 - **ROOT_URLCONF:** Menentukan di mana URL berada.
 - **TEMPLATES:** Mendefinisikan kelas template, daftar direktori tempat komputer harus mencari file template, dan pengaturan template tertentu.
 - **AUTH_PASSWORD_VALIDATORS:** menentukan validasi yang diinginkan pada password – misalnya, panjang minimum. Ada beberapa properti lain di settings.py, seperti **LANGUAGE_CODE** dan **TIME_ZONE**, tetapi kita hanya fokus pada properti yang lebih penting.

Struktur Direktori

- **urls.py**: File ini memberitahu Django halaman mana yang akan dirender sebagai response terhadap browser atau permintaan URL. Misalnya, ketika seseorang memasuki `http://localhost:8000/123` URL, permintaan masuk ke `urls.py` dan dialihkan ke halaman berdasarkan jalur yang ditentukan di sana.
- **wsgi.py**: File ini adalah singkatan dari Web Server Gateway Interface (WSGI) dan membantu Django melayani halaman web.
- **manage.py**: File ini digunakan internal oleh Django, kita tidak perlu melakukan apapun di file ini
- **db.sqlite3** : Adalah File database SQLITE yang merupakan bawaan dan digunakan selama development

Django Project Structure

- Sebuah project django dapat memiliki beberapa apps



Django Project Structure

- Setiap apps kita buat, kemudian di daftarkan di file settings.py pada bagian **INSTALLED_APPS**

```
settings.py X
bookreviews > settings.py > ...
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40 ]
```


App book

- Sebagai awal, kita akan membuat apps Buku di dalam project BookReviews
- Kita gunakan terminal pada Visual Studio Code dengan memilih menu View -> Terminal
- Ketik : **python manage.py startapp book**

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

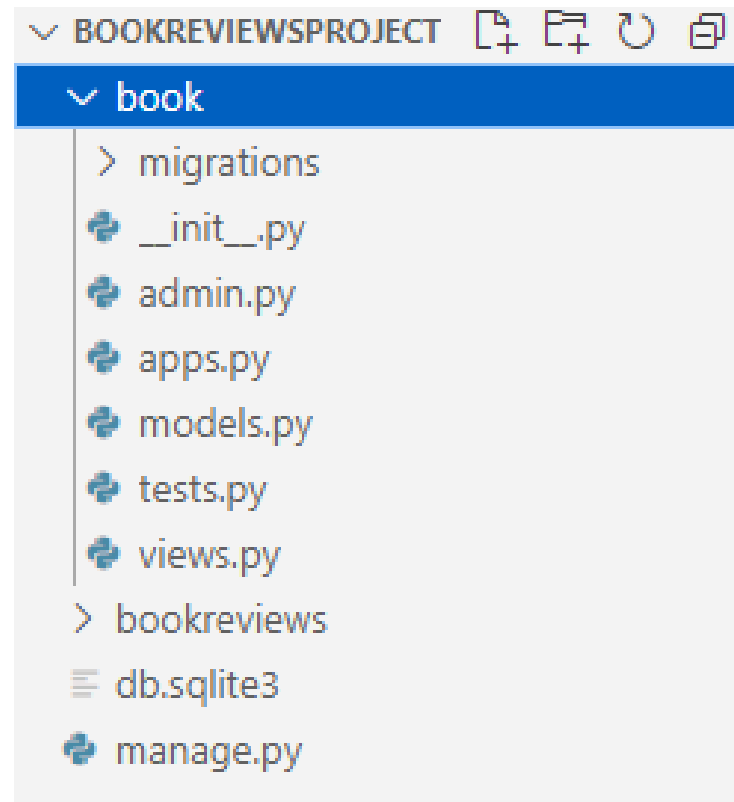
Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\projectdjango\bookreviewsproject> python manage.py startapp book
```

```
PS C:\projectdjango\bookreviewsproject> █
```

App book

- Setelah perintah tersebut maka akan terbentuk folder book sebagai berikut:



App book

- Daftarkan app book pada bookreviews/settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'book'  
]
```

Struktur App book

- Folder **migrations** adalah tempat Django menyimpan migrasi, atau perubahan pada basis data.
- **__init__.py** memberi tahu Python bahwa app book adalah sebuah package.
- **admin.py** adalah tempat untuk mendaftarkan **model** dengan aplikasi admin Django.
- **apps.py** adalah file konfigurasi umum untuk aplikasi Django.
- **models.py** adalah modul yang berisi model untuk aplikasi.
- **test.py** berisi prosedur pengujian yang dijalankan saat menguji aplikasi.
- **views.py** adalah modul yang berisi tampilan untuk aplikasi.

Membuat halaman Home Page

- Buka file views.py di folder book

```
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.
def home(request):
    return HttpResponse('<h1>Welcome to Home Page</h1>')
```

Membuat halaman Home Page

- Buka file urls.py di folder bookreviews

```
from django.contrib import admin
from django.urls import path
from book import views as bookView

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', bookView.home)
]
```

Membuat halaman Home Page

- Pada terminal jalankan perintah : **python .\manage.py runserver**

```
PS C:\projectdjango\bookreviewsproject> python .\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have 18 unapplied migration(s). Your project may not work properly
ns.
```

```
Run 'python manage.py migrate' to apply them.
```

```
November 29, 2022 - 05:16:50
```

```
Django version 4.1.3, using settings 'bookreviews.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CTRL-BREAK.
```

Membuat halaman Home Page

- Buka browser dan ketik : `http://127.0.0.1:8000/`



Welcome to Home Page

Latihan : membuat halaman About



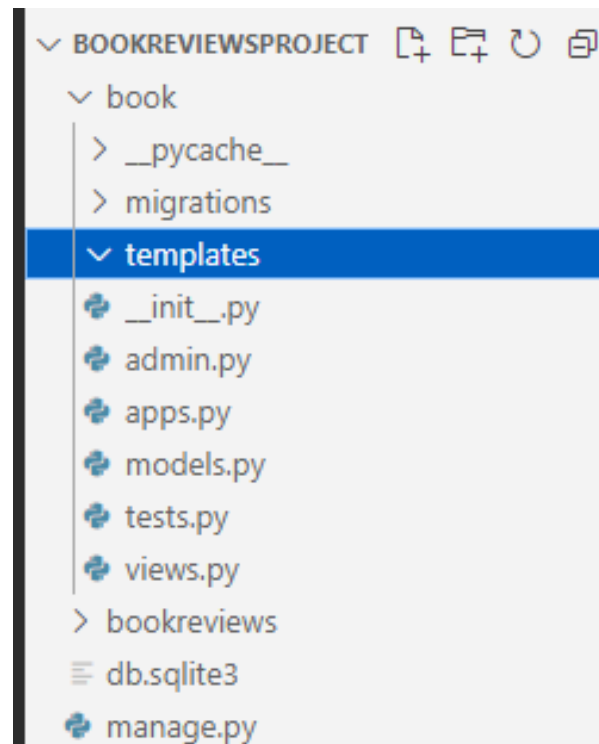
← → ↻ ⓘ 127.0.0.1:8000/about/

Welcome to About Page

- Buka file views.py di folder book
- Tambahkan function about
- Tambahkan url di urls.py

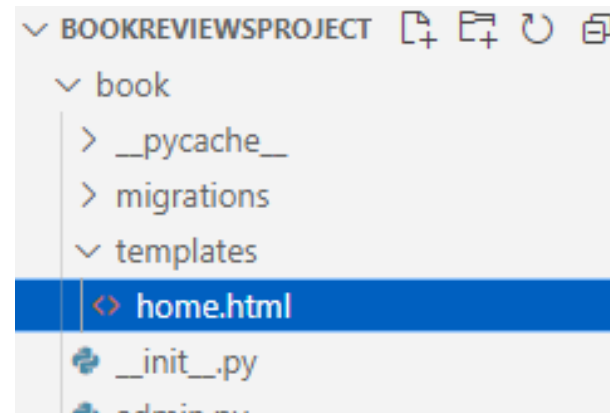
Membuat Templates (HTML)

- Di dalam folder book, buat folder templates



Membuat Templates (HTML)

- Di dalam folder templates buat file home.html, klik kanan folder templates -> New File



Membuat Templates (HTML)

- Isi home.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Movies App</title>
    <meta name="description" content="Aplikasi BookReview"/>
    <meta name="keywords" content="book review,katalog buku"/>
  </head>
  <body>
    <h1>Welcome to Home Page</h1>
    <div>Konten disini</div>
  </body>
</html>
```

Membuat Templates (HTML)

- Buka file book/views.py

```
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.
def home(request):
    return render(request, 'home.html')

def about(request):
    return HttpResponse('<h1>Welcome to About Page</h1>')
```

Latihan : membuat Templates (HTML) untuk halaman about/

- Buat file about.html di dalam folder book/templates
- Isi about.html dengan html seperti home.html, ubah isinya
- Ubah book/views.py supaya method about me-render about.html

Passing data ke template

- Buka file book/views.py

```
from django.shortcuts import render  
from django.http import HttpResponse
```

```
# Create your views here.
```

```
def home(request):  
    return render(request, 'home.html', {'name' : 'Denny Chandra'})
```

```
def about(request):  
    return render(request, 'about.html')
```

Passing data ke template

- Buat file book/templates/home.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Movies App</title>
    <meta name="description" content="Aplikasi BookReview"/>
    <meta name="keywords" content="book review,katalog buku"/>
  </head>
  <body>
    <h1>Welcome to Home Page {{ name }}</h1>
    <div>Konten disini</div>
  </body>
</html>
```


Latihan Passing data ke template about

← → ↻ ⓘ 127.0.0.1:8000/about/

Welcome to About Page BookReview

Konten disini

Membuat Hyperlink

- Buka file bookreviews/urls.py

```
from django.contrib import admin
from django.urls import path
from book import views as bookView

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', bookView.home, name='home'),
    path('about/', bookView.about, name='about')
]
```

Membuat Hyperlink

- Buka file book/templates/home.html dan about.html
- Beri potongan kode berikut setelah <body>

```
<a href="{% url 'home' %}">Home</a>
```

```
<a href="{% url 'about' %}">About</a>
```

Menambahkan bootstrap di templates

- Ke alamat <https://getbootstrap.com/>
- Klik Icon disk sebelah kanan

Include via CDN

When you only need to include Bootstrap's compiled CSS or JS, you can use [jsDelivr](#). See it in action with our simple [quick start](#), or [browse the examples](#) to jumpstart your next project. You can also choose to include Popper and our JS [separately](#).

```
<!-- CSS only -->
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/"
```

Copy to clipboard



Menambahkan bootstrap di templates

- Paste di templates home.html dan about.html

```
home.html ×  urls.py  about.html
jok > templates > <> home.html
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <title>Movies App</title>
5          <meta name="description" content="Aplikasi BookReview"/>
6          <meta name="keywords" content="book review,katalog buku"/>
7          <!-- CSS only -->
8          <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
9      </head>
```

Menambahkan bootstrap di templates

- Tambahkan elemen `<div class="container">` setelah body

```
<body>
```

```
    <div class="container">
```

```
        <a href="{% url 'home' %}">Home</a>
```

```
        <a href="{% url 'about' %}">About</a>
```

```
        <h1>Welcome to Home Page {{ name }}</h1>
```

```
    <div>Konten disini</div>
```

```
</div>
```

```
</body>
```

Menambahkan bootstrap di templates



Menambahkan form search di HomePage

- Menggunakan VSCode buka book/templates/home.html

```
<body>
  <div class="container">
    <a href="{% url 'home' %}">Home</a>
    <a href="{% url 'about' %}">About</a>
    <form action="">
      <div class="mb-3">
        <label class="form-label">Cari Buku:</label>
        <input type="text" name="judul" class="form-control" />
      </div>
      <button type="submit" class="btn btn-primary">Search</button>
    </form>
  </div>
</body>
```


Menambahkan form search di HomePage

127.0.0.1:8000



[Home](#) [About](#)

Cari Buku:

Search

Menambahkan form search di HomePage

- Menggunakan VSCode buka book/views.py

```
from django.shortcuts import render
from django.http import HttpResponse
```

```
# Create your views here.
```

```
def home(request):
    searchJudul = request.GET.get('judul')
    return render(request, 'home.html', {'searchJudul':searchJudul})
```

```
def about(request):
    return render(request, 'about.html', {'appName' : 'BookReview'})
```

Menambahkan form search di HomePage

- Menggunakan VSCode buka book/templates/home.html

```
<div class="container">
  <a href="{% url 'home' %}">Home</a>
  <a href="{% url 'about' %}">About</a>
  <form action="">
    <div class="mb-3">
      <label class="form-label">Cari Buku:</label>
      <input type="text" name="judul" class="form-control" />
    </div>
    <button type="submit" class="btn btn-primary">Search</button>
  </form>
  {% if searchJudul %}
    Pencarian untuk <strong>{{ searchJudul }}</strong>
  {% endif %}
</div>
```

Menambahkan form search di HomePage

127.0.0.1:8000/?judul=Pemrograman+Python



[Home](#) [About](#)

Cari Buku:

Search

Pencarian untuk **Pemrograman Python**

Referensi Django Template Language

- <https://docs.djangoproject.com/en/4.1/ref/templates/builtins/>

Resume command

- **python -m django startproject bookreviews** (untuk membuat project)
- **python manage.py runserver** (untuk menjalankan project)
- **python manage.py startapp book** (untuk membuat app)

Model

- Membuat model Buku
- Manage migrations
- Mengakses Django admin
- Mengkonfigurasi gambar
- Menyimpan gambar (upload)
- Menambah model di admin

Model

- Model adalah object yang dipetakan ke database
- Ketika membuat model, Django melakukan eksekusi SQL untuk membuat tabel yang sesuai dalam database tanpa kita harus menulis satu baris pun dari SQL

Membuat Model Buku

- Buka menggunakan VSCode book/models.py

```
from django.db import models
```

```
# Create your models here.
```

```
class Book(models.Model):
```

```
    judul = models.CharField(max_length=250)
```

```
    isbn = models.CharField(max_length=13)
```

```
    pengarang = models.CharField(max_length=250)
```

```
    sinopsis = models.TextField(blank=True, null=True) #null = DB, blank = form
```

```
    tanggal_rilis = models.DateField()
```

```
    jumlah_halaman = models.IntegerField()
```

```
    gambar = models.ImageField(upload_to='movie/images/')
```

```
def __str__(self) -> str:
```

```
    return self.judul
```

Referensi Django Model (Field)

Field Name	Class	Particular
AutoField	class AutoField(**options)	It An IntegerField that automatically increments.
BigAutoField	class BigAutoField(**options)	It is a 64-bit integer, much like an AutoField except that it is guaranteed to fit numbers from 1 to 9223372036854775807.
BigIntegerField	class BigIntegerField(**options)	It is a 64-bit integer, much like an IntegerField except that it is guaranteed to fit numbers from -9223372036854775808 to 9223372036854775807.
BinaryField	class BinaryField(**options)	A field to store raw binary data.
BooleanField	class BooleanField(**options)	A true/false field. The default form widget for this field is a CheckboxInput.

Referensi Django Model (Field)

CharField	<code>class DateField(auto_now=False, auto_now_add=False, **options)</code>	It is a date, represented in Python by a <code>datetime.date</code> instance.
DateTimeField	<code>class DateTimeField(auto_now=False, auto_now_add=False, **options)</code>	It is a date, represented in Python by a <code>datetime.date</code> instance.
DateTimeField	<code>class DateTimeField(auto_now=False, auto_now_add=False, **options)</code>	It is used for date and time, represented in Python by a <code>datetime.datetime</code> instance.
DecimalField	<code>class DecimalField(max_digits=None, decimal_places=None, **options)</code>	It is a fixed-precision decimal number, represented in Python by a <code>Decimal</code> instance.
DurationField	<code>class DurationField(**options)</code>	A field for storing periods of time.
EmailField	<code>class EmailField(max_length=254, **options)</code>	It is a <code>CharField</code> that checks that the value is a valid email address.
FileField	<code>class FileField(upload_to=None, max_length=100, **options)</code>	It is a file-upload field.

Referensi Django Model (Field)

FloatField	<code>class FloatField(**options)</code>	It is a floating-point number represented in Python by a float instance.
ImageField	<code>class ImageField(upload_to=None, height_field=None, width_field=None, max_length=100, **options)</code>	It inherits all attributes and methods from FileField, but also validates that the uploaded object is a valid image.
IntegerField	<code>class IntegerField(**options)</code>	It is an integer field. Values from -2147483648 to 2147483647 are safe in all databases supported by Django.
NullBooleanField	<code>class NullBooleanField(**options)</code>	Like a BooleanField, but allows NULL as one of the options.
PositiveIntegerField	<code>class PositiveIntegerField(**options)</code>	Like an IntegerField, but must be either positive or zero (0). Values from 0 to 2147483647 are safe in all databases supported by Django.

Referensi Django Model (Field)

SmallIntegerField	<code>class SmallIntegerField(**options)</code>	It is like an IntegerField, but only allows values under a certain (database-dependent) point.
TextField	<code>class TextField(**options)</code>	A large text field. The default form widget for this field is a Textarea.
TimeField	<code>class TimeField(auto_now=False, auto_now_add=False, **options)</code>	A time, represented in Python by a datetime.time instance.

Referensi Django Model (Field Options)

Field Options	Particulars
Null	Django will store empty values as NULL in the database.
Blank	It is used to allowed field to be blank.
Choices	An iterable (e.g., a list or tuple) of 2-tuples to use as choices for this field.
Default	The default value for the field. This can be a value or a callable object.
help_text	Extra "help" text to be displayed with the form widget. It's useful for documentation even if your field isn't used on a form.
primary_key	This field is the primary key for the model.
Unique	This field must be unique throughout the table.

<https://www.javatpoint.com/django-model>

Referensi Django Model

- <https://docs.djangoproject.com/en/4.1/ref/models/>

Instalasi Pillow (library Image)

- Pillow adalah library untuk melakukan image processing
- Pada command Prompt ketik : **pip install pillow**

```
C:\projectdjango\bookreviewsproject>pip install pillow
Requirement already satisfied: pillow in c:\users\windows\appdata\local\programs\python\python311\lib\site-packages (9.3.0)

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```


Managing Migrations

- Migrations bertujuan untuk men-generate skema database
- Perintah migrate akan membuat database berdasarkan default setting
- Ketik : **python manage.py migrate**

```
C:\projectdjango\bookreviewsproject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

Managing Migrations

- Membuat migrations untuk Book
- Ketik : **python manage.py makemigrations**

```
C:\projectdjango\bookreviewsproject>python manage.py makemigrations
Migrations for 'book':
  book\migrations\0001_initial.py
    - Create model Book
```

Managing Migrations

- Jalankan migrations
- Ketik : **python manage.py migrate**

```
C:\projectdjango\bookreviewsproject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, book, contenttypes, sessions
Running migrations:
  Applying book.0001_initial... OK
```

Managing Migrations

- Jadi setiap penambahan model harus melakukan command
 - **python manage.py makemigrations**
 - **python manage.py migrate**

Mengakses Django admin

- Alamat admin sudah definisikan di awal (default) di bookreviews/url.py

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', bookView.home, name='home'),  
    path('about/', bookView.about, name='about')  
]
```

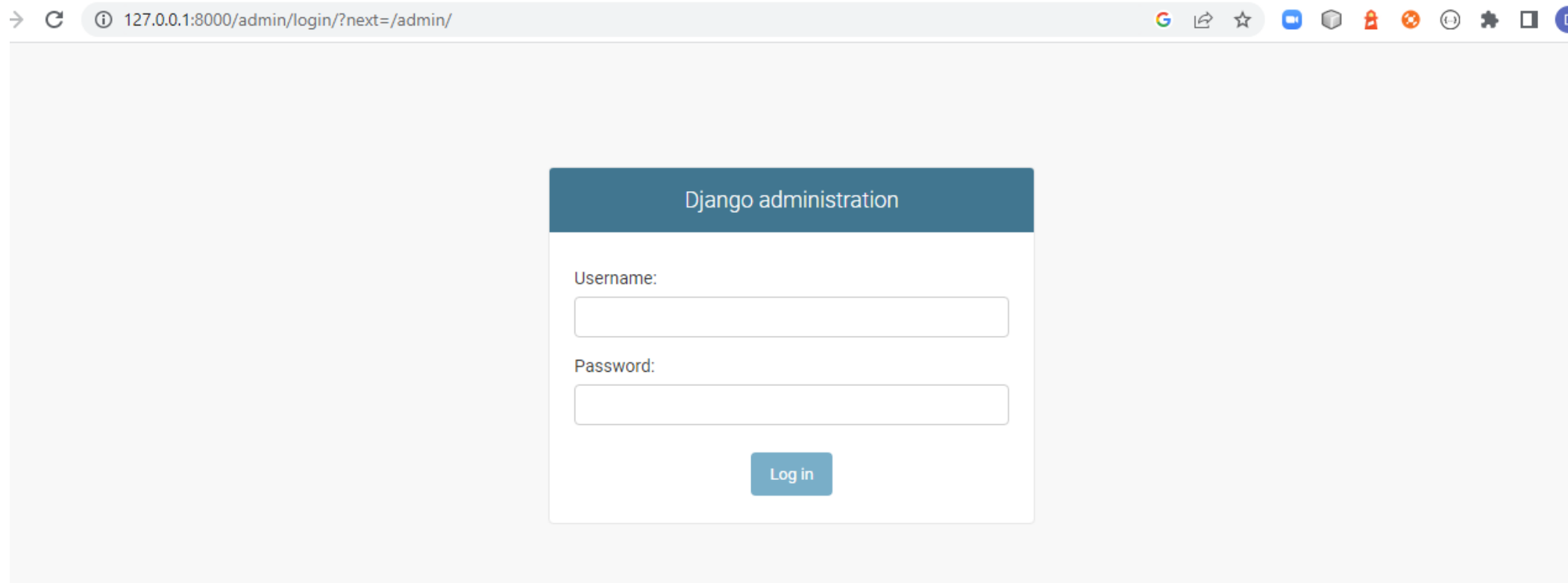
- Untuk mengakses maka jalankan aplikasi : **python manage.py runserver**

```
PS C:\projectdjango\bookreviewsproject> python .\manage.py runserver  
Watching for file changes with StatReloader  
Performing system checks...
```

```
System check identified no issues (0 silenced).  
November 30, 2022 - 08:10:15  
Django version 4.1.3, using settings 'bookreviews.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```

Mengakses Django admin

- Buka browser ketik : <http://127.0.0.1:8000/admin>



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/admin/login/?next=/admin/`. The page content is the Django administration login interface. It features a dark blue header with the text "Django administration". Below the header, there are two input fields: "Username:" and "Password:". A blue "Log in" button is positioned below the password field. The browser's address bar also shows standard navigation icons (back, forward, refresh) and a search bar.

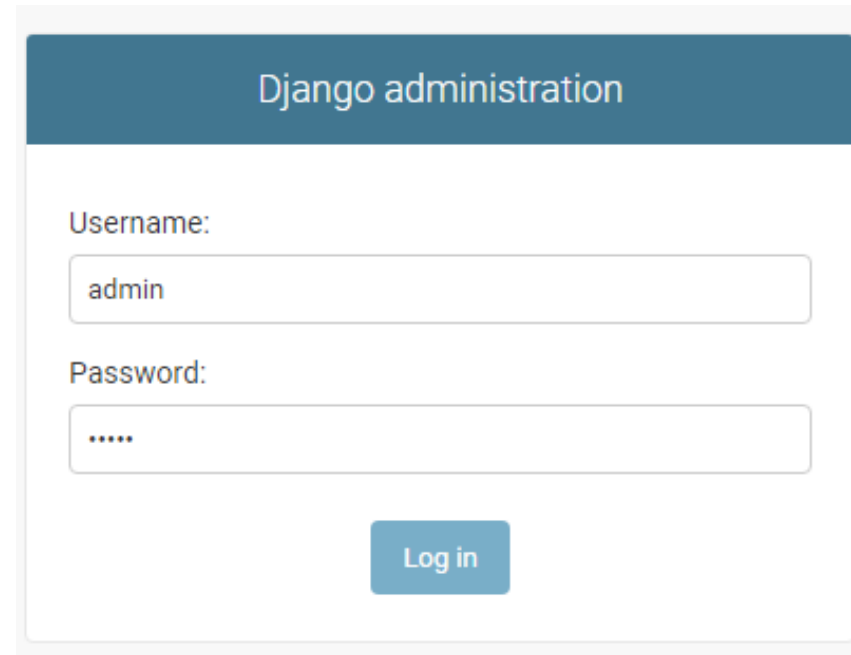
Mengakses Django admin

- Membuat superuser yang akan digunakan untuk login admin
- Ketik : **python manage.py createsuperuser**
- Isi username,email,password

```
C:\projectdjango\bookreviewsproject>python manage.py createsuperuser
Username (leave blank to use 'windows'): admin
Email address: dennyc.dev@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Mengakses Django admin

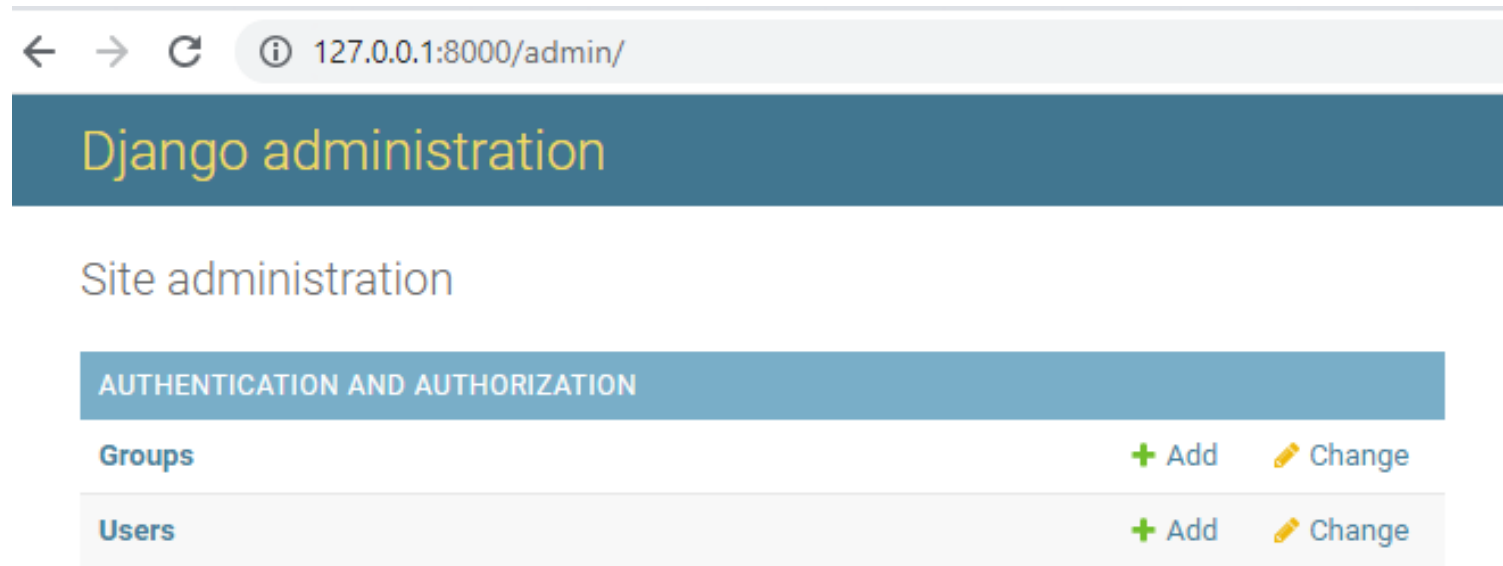
- Login menggunakan user yang dibuat tadi



The image shows a screenshot of the Django administration login interface. At the top, there is a dark blue header with the text "Django administration" in white. Below the header, the form is white. It contains two labels: "Username:" and "Password:". Under "Username:", there is a text input field containing the word "admin". Under "Password:", there is a password input field with five dots. At the bottom of the form, there is a blue button with the text "Log in" in white.

Mengakses Django admin

- Tampilan Django admin



Mengkonfigurasi setting gambar

- Buka menggunakan VSCode buka bookreviews/settings.py
- Tambahkan kode berikut paling bawah

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')  
MEDIA_URL = '/media/'
```

- Tambahkan paling atas

```
import os
```

Mengkonfigurasi setting gambar

- Buka menggunakan VSCode buka bookreviews/urls.py
- Sesuaikan kode berikut

```
from django.contrib import admin
from django.urls import path
from book import views as bookView
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', bookView.home, name='home'),
    path('about/', bookView.about, name='about')
]

urlpatterns += static(settings.MEDIA_URL,
    document_root=settings.MEDIA_ROOT)
```

Menambahkan model Book ke admin

- Buka menggunakan VSCode file : book/admin.py

```
from django.contrib import admin  
from .models import Book
```

```
# Register your models here.  
admin.site.register(Book)
```

CRUD Buku

- List Buku

Select book to change

ADD BOOK +

Action: 0 of 1 selected

☐ BOOK

☐ Pemrograman Python

1 book

CRUD Buku

- Input Buku

Add book


Judul:

Isbn:

Pengarang:

Sinopsis:

Tanggal rilis:

Today | 

Note: You are 7 hours ahead of server time.

Jumlah halaman:

Gambar:

Choose File

No file chosen

Save and add another

Save and continue editing

SAVE

Resume command

- **python manage.py migrate** (untuk melakukan inisialisasi database)
- **python manage.py makemigrations** (untuk melakukan migrasi model)
- **python manage.py createsuperuser** (untuk setting superuser)

Resume command setiap ada model baru

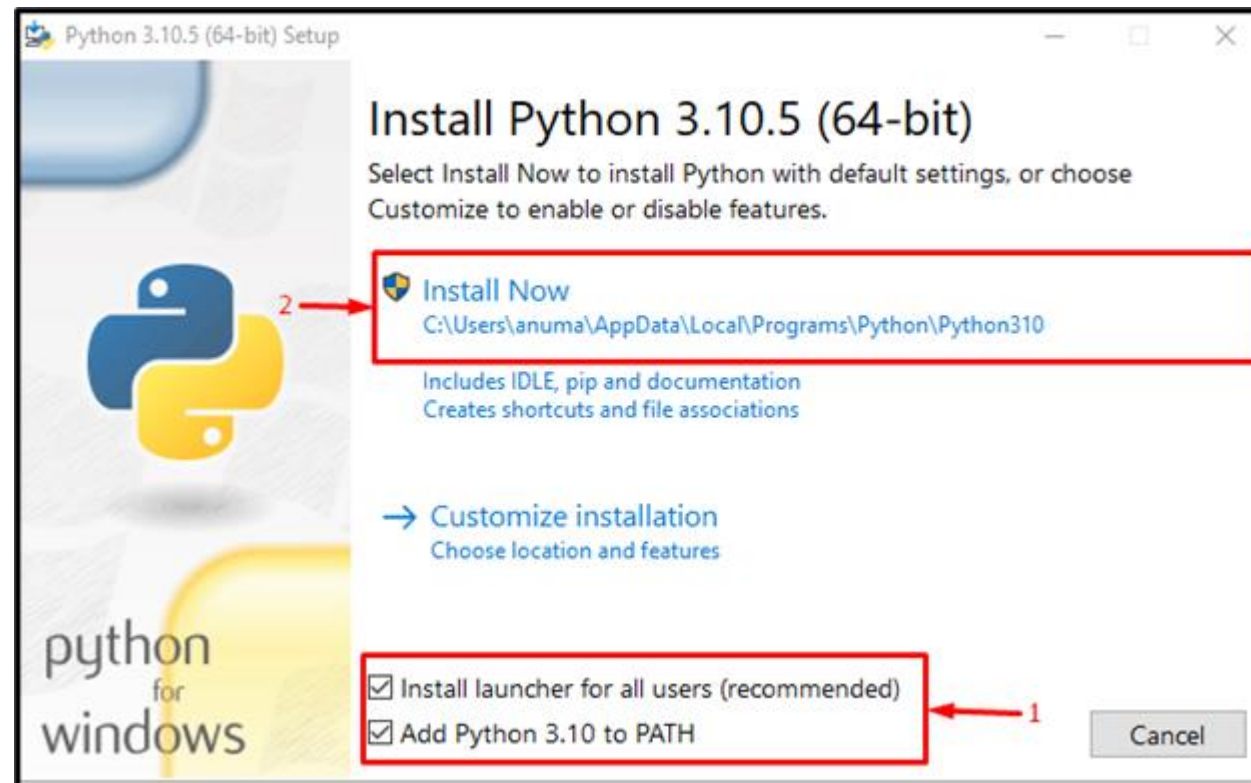
- **python manage.py makemigrations** (untuk melakukan migrasi model)
- **python manage.py migrate** (untuk melakukan inisialisasi database)

Resume command instalasi

- Instalasi ini hanya dilakukan sekali di komputer, kecuali jika ingin update
- **pip install Django==4.1.3** (untuk install Django)
- **pip install pillow** (untuk install library pillow)

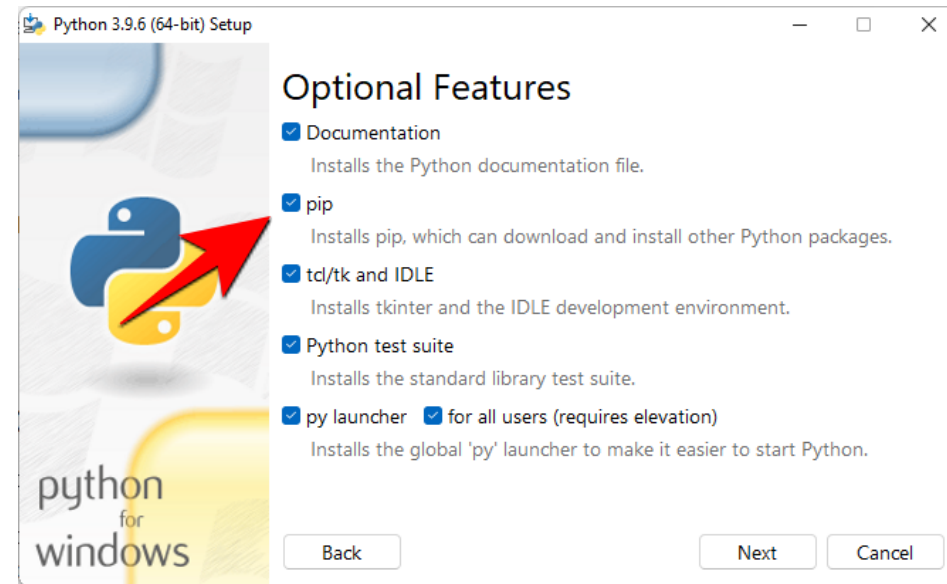
Troubleshooting

- Command pyton atau py tidak dikenali, pastikan ketika instalasi seperti ini:



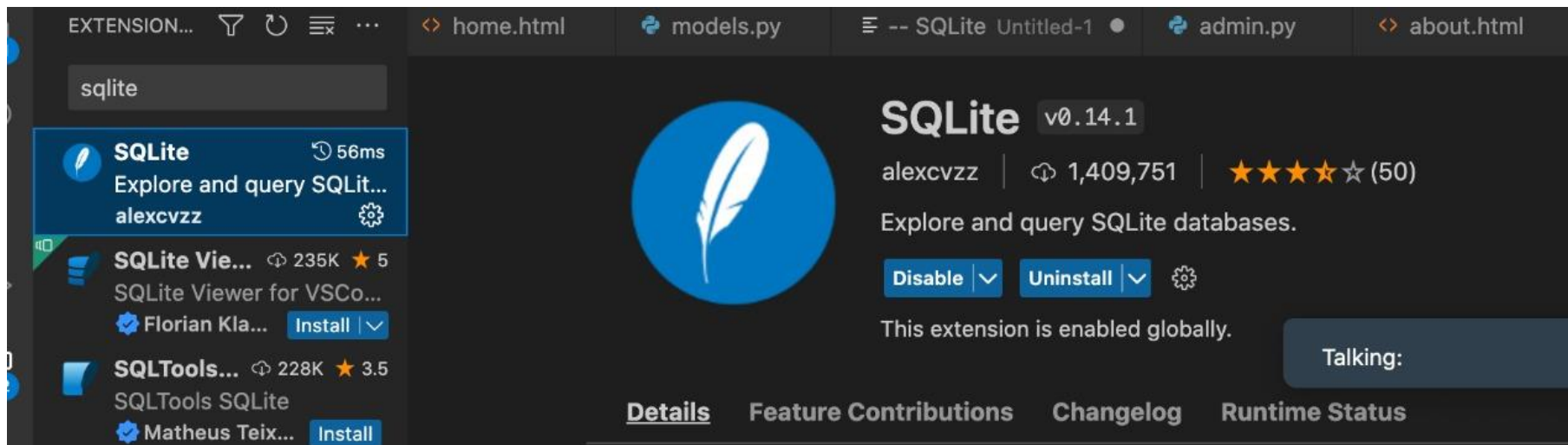
Troubleshooting

- Command pip tidak dikenali, bisa dengan custom install seperti gambar dibawah



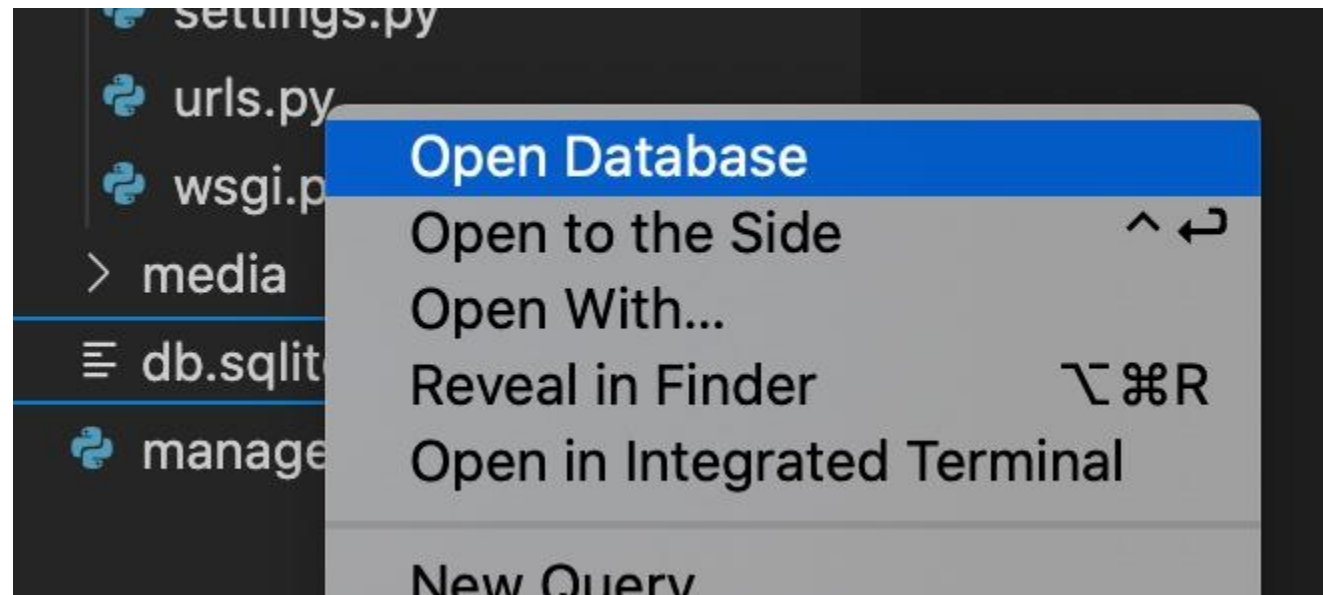
VSCode Extension: SQLite

- Untuk melihat isi dari db.sqlite3 maka bisa menggunakan extension SQLite



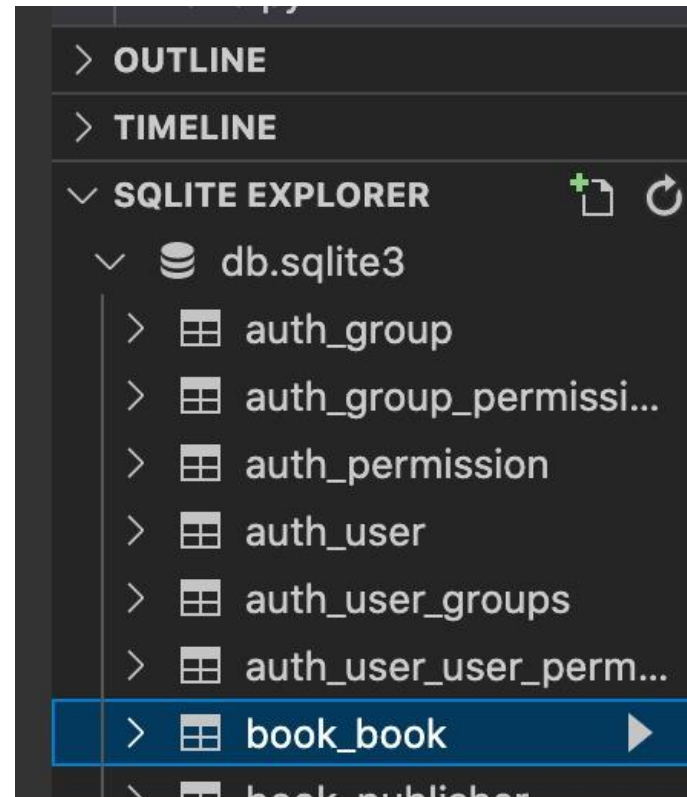
VSCode Extension: SQLite

- Klik kanan db.sqlite3 -> Open Database



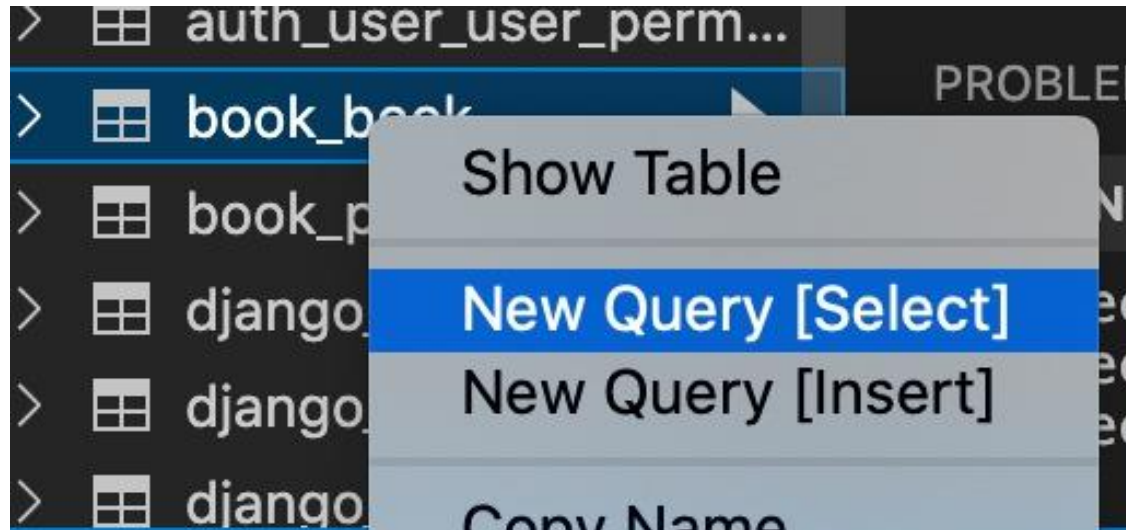
VSCode Extension: SQLite

- Akan muncul SQLite Explorer



VSCode Extension: SQLite

- Klik Kanan pada tabel book_book -> New Query [Select]



VSCode Extension: SQLite

- Akan muncul SQL Editor lalu ketikkan kode SQL untuk operasi ke tabel book_book
- Berikut SQL untuk menghapus semua data di tabel book_book

```
1  -- SQLite
2  DELETE FROM book_book;
```

Change All Occurrences Ctrl+F2

Refactor... Ctrl+Shift+R

Commit Changes >

Cut Ctrl+X

Copy Ctrl+C

Paste Ctrl+V

Run Query Ctrl+Shift+Q

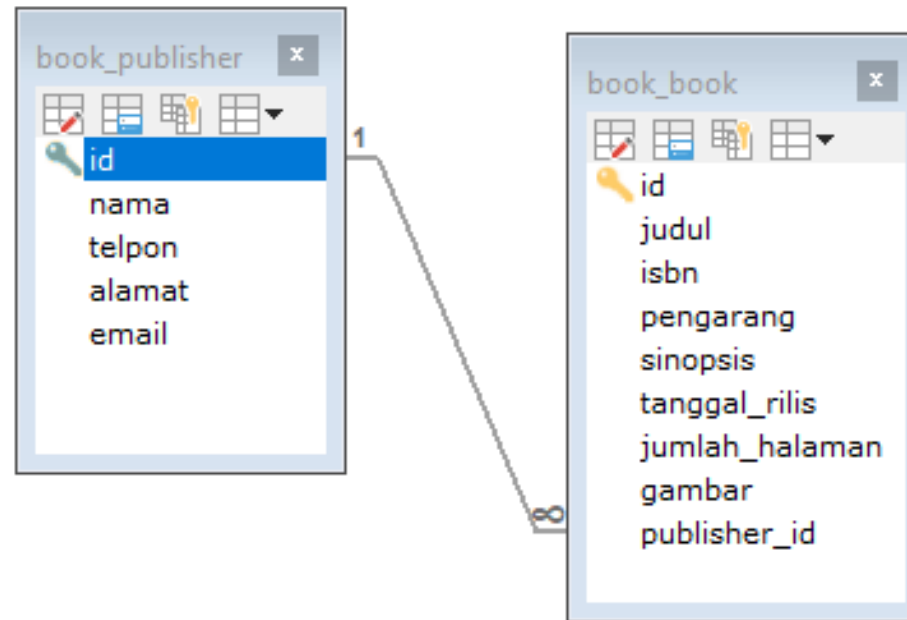
Run Selected Query

Use Database

Command Palette... Ctrl+Shift+P

Relasi antar Model

- Aplikasi BookReviews memiliki 2 tabel publisher dan book
- Relasinya adalah 1 publisher dapat menerbitkan banyak buku (one to Many) atau banyak buku diterbitkan oleh 1 publisher (many to one)



Model

- Membuat model Publisher
- Melakukan edit model Buku
- Melakukan migrations

File book/models.py

```
from django.db import models
from django.core.validators import RegexValidator

# Create your models here.
class Publisher(models.Model):
    nama = models.CharField(max_length=250)
    alamat = models.TextField(blank=True, null=True)
    phone_regex = RegexValidator(regex=r'^\+?1?\d{9,15}$', message="min 8 digit")
    telpon = models.CharField(validators=[phone_regex], max_length=15, blank=True)
    email = models.EmailField(blank=True, null=True)
    def __str__(self) -> str:
        return self.nama
```

File book/models.py (lanjutan)

```
class Book(models.Model):
    judul = models.CharField(max_length=250)
    isbn = models.CharField(max_length=13)
    pengarang = models.CharField(max_length=250)
    sinopsis = models.TextField(blank=True, null=True) #null = DB, blank = form
    tanggal_rilis = models.DateField()
    jumlah_halaman = models.IntegerField()
    gambar = models.ImageField(upload_to='movie/images/')
    publisher = models.ForeignKey(Publisher, on_delete=models.CASCADE)

    def __str__(self) -> str:
        return self.judul
```

File book/admin.py (lanjutan)

```
from django.contrib import admin
from .models import Book
from .models import Publisher
```

```
# Register your models here.
admin.site.register(Publisher)
admin.site.register(Book)
```

Lakukan Migration

- Hapus semua data book jika masih ada datanya
- Lakukan eksekusi berikut:
 - **python manage.py makemigrations**
 - Jika muncul warning seperti disamping, maka ketik 1 lalu ketik lagi 1 seperti gambar disamping
 - **python manage.py migrate**

```
C:\projectdjango\bookreviewsproject>python manage.py makemigrations
Migrations for 'book':
  book\migrations\0003_alter_book_publisher.py
  - Alter field publisher on book

C:\projectdjango\bookreviewsproject>python manage.py migrate
Operations to perform:
```

```
machintos@192 bookreviewsproject % python3 manage.py makemigrations
It is impossible to add a non-nullable field 'publisher' to book without specifying a default. This is because the database needs something to populate existing rows.
Please select a fix:
  1) Provide a one-off default now (will be set on all existing rows with a null value for this column)
  2) Quit and manually define a default value in models.py.
Select an option: 1
Please enter the default value as valid Python.
The datetime and django.utils.timezone modules are available, so it is possible to provide e.g. timezone.now as a value.
Type 'exit' to exit this prompt
>>> 1
Migrations for 'book':
  book/migrations/0002_publisher_book_publisher.py
  - Create model Publisher
  - Add field publisher to book
machintos@192 bookreviewsproject %
```

Hasil di Halaman Admin

←
→
↻
127.0.0.1:8000/admin/

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	✎ Change
Users	+ Add	✎ Change

BOOK

Books	+ Add	✎ Change
Publishers	+ Add	✎ Change

Hasil di Halaman Admin

Add publisher

Nama:

Alamat:

Telpon:

Email:

Save and add another

Save and continue editing

SAVE

Hasil di Halaman Admin

Add book


Judul:

Isbn:

Pengarang:

Sinopsis:

Tanggal rilis:

Today | 

Note: You are 7 hours ahead of server time.

Jumlah halaman:

Gambar:

Choose File

No file chosen

Publisher:



PT.Elex Media Komputindo
Pustaka Setia

Save and add another

Save and continue editing

SAVE

Halaman Pencarian

- Mengubah book/views.py
- Mengubah template book/templates/home.html

File book/views.py (potongan kode)

```
def home(request):  
    searchJudul = request.GET.get('judul')  
    if searchJudul:  
        books = Book.objects.filter(judul__icontains=searchJudul)  
    else:  
        books = Book.objects.all()  
    return render(request, 'home.html', {'searchJudul':searchJudul,'books' : books})
```

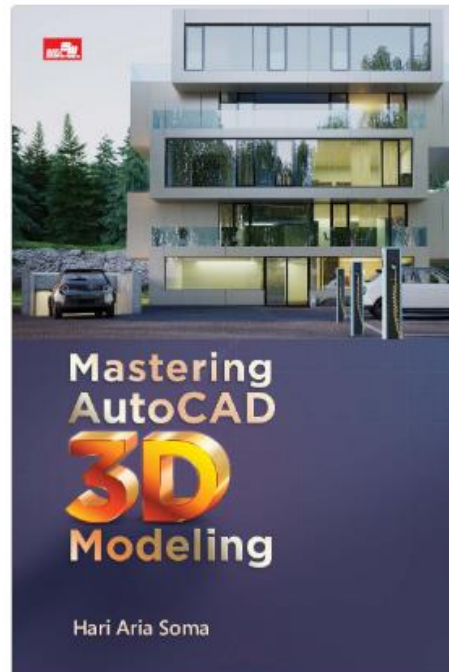
File book/templates/home.html (potongan kode)

```
{% if searchJudul %}Pencarian untuk <strong>{{ searchJudul }}</strong>{% endif %}
<br/>
<div class="row row-cols-1 row-cols-md-3 g-4">
  {% for book in books %}
    <div class="col">
      <div class="card">
        
        <div class="card-body">
          <h5 class="card-title fw-bold">{{ book.judul }} | {{ book.pengarang }} | {{ book.tanggal_rilis|date:"d-m-Y" }}</h5>
          <p class="card-text">
            {{ book.sinopsis }}
          <br/>
            {{ book.isbn }} | {{ book.jumlah_halaman }} | {{ book.publisher.nama }}
          <a href="{% url 'detail' book.id %}">
            <h5 class="card-title fw-bold">Detail</h5>
          </a>
        </p>
      </div>
    </div>
  </div>
</div>
{% endfor %}
</div>
```

Hasil halaman home

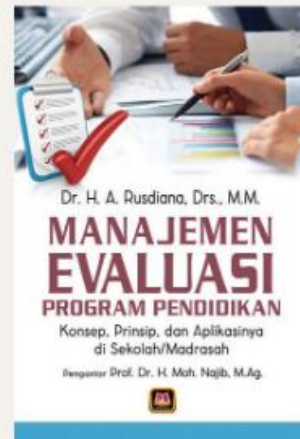
Cari Buku:

Search



**Mastering AutoCAD 3D Modeling | Hari
Aria Soma | 09-11-2022**

Buku yang sangat bagus, Berisi kumpulan soal latihan
dari berbagai model 3
9786230036439 | 240 | PT.Elex Media Komputindo



**Manajemen Evaluasi Program
Pendidikan Konsep Prinsip dan
Aplikasinya di Sekolah/Madrasah |
Rusdiana | 01-07-2017**

dd
9789790766549 | 246 | Pustaka Setia

[Detail](#)

Hasil pencarian di halaman home

127.0.0.1:8000/?judul=mastering

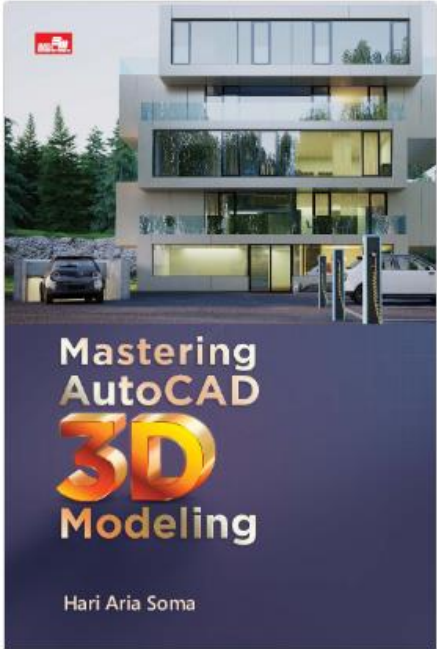
About

Cari Buku:

mastering

Search

Pencarian untuk **mastering**



Mastering AutoCAD 3D Modeling | Hari Aria Soma | 09-11-2022

Buku yang sangat bagus, Berisi kumpulan soal latihan dari berbagai model 3

9786230036439 | 240 | PT.Elex Media Komputindo

Membuat Halaman Detail (klik detail dari home)

- Buat template book/templates/detail.html
- Buat method detail di book/views.py
- Buat path di bookreviews/urls.py

Buat template book/templates/detail.html

```
<div class="card">
  
  <div class="card-body">
    <h5 class="card-title fw-bold">
      {{ book.judul }} | {{ book.pengarang }} | {{ book.tanggal_rilis|date:"d-m-Y" }}
    </h5>
    <p class="card-text">
      {{ book.sinopsis }}
      <br/>
      {{ book.isbn }} | {{ book.jumlah_halaman }} | {{ book.publisher.nama }}
    </p>
  </div>
</div>
```


Buat method detail di book/views.py (potongan kode)

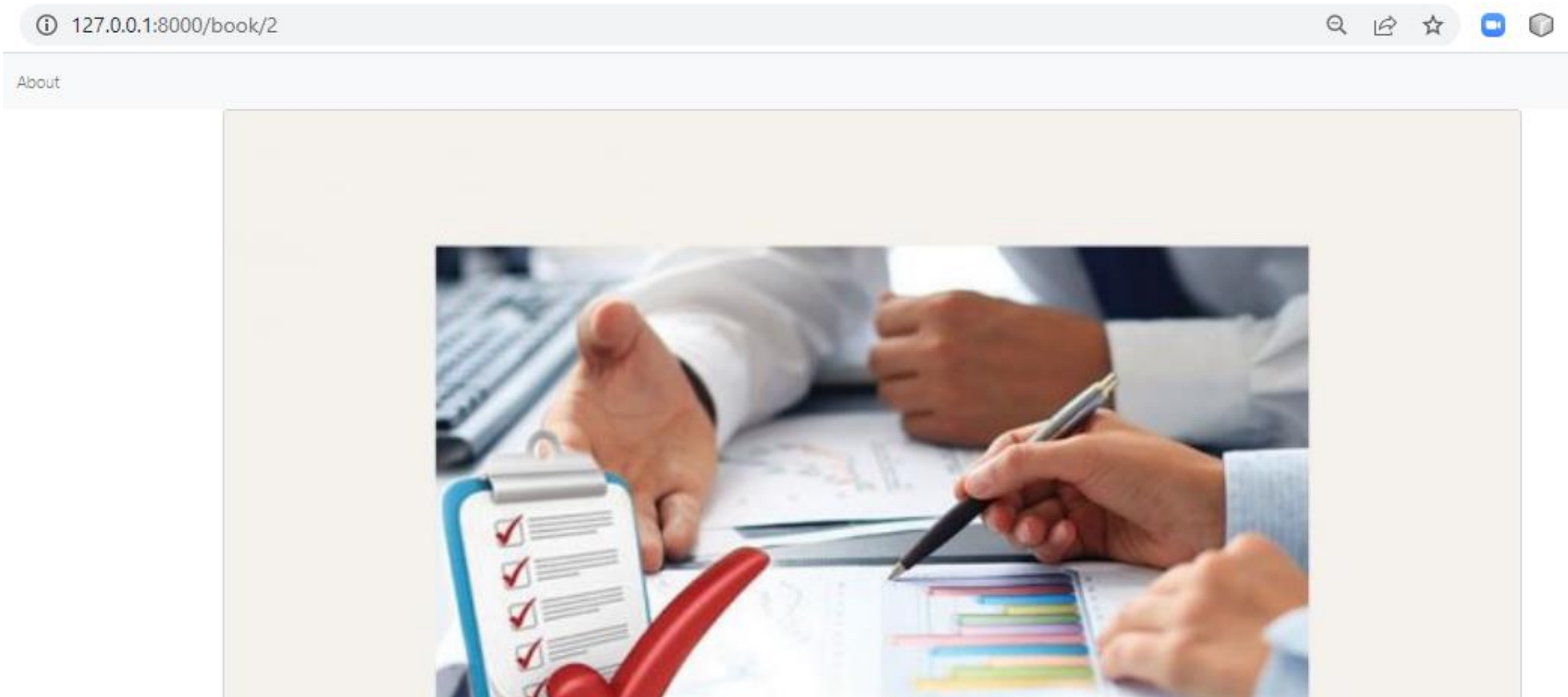
- Tambab import `from django.shortcuts import get_object_or_404`
- Tambah method

```
def detail(request,book_id):  
    book = get_object_or_404(Book,pk=book_id)  
    return render(request,'detail.html',{'book' : book})
```

Tambah path di bookreviews/urls.py

```
path('book/<int:book_id>', bookView.detail, name='detail')
```

Hasil halaman detail



Membuat Base Template (untuk menu)

- Membuat base template untuk menu untuk menghilangkan duplikasi kode
 - Buat file di book/templates/base.html
 - Masukkan base template ke setiap halaman (home,about,detail)

Buat file di book/templates/base.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>BookReviews App</title>
    <!-- CSS only -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
          rel="stylesheet"
          integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
          crossorigin="anonymous"/>
    <!-- JavaScript Bundle with Popper -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpyxD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
crossorigin="anonymous"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
  </head>
```

Buat file di book/templates/base.html (lanjutan)

```
<body>
  <nav class="navbar navbar-expand-lg bg-light">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">BookReview</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
```

Buat file di book/templates/base.html (lanjutan)

```
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
  <li class="nav-item">
    <a class="nav-link" href="{% url 'home' %}">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="{% url 'about' %}">About</a>
  </li>
</ul>
<ul class="navbar-nav mb-2 mb-lg-0 d-flex">
  <li class="nav-item d-flex justify-content-end">
    <a class="nav-link active" aria-current="page" href="{% url 'admin:index' %}">Login</a>
  </li>
</ul>

</ul>
</div>
</div>
</nav>
<div class="container">
  {% block content %}
  {% endblock content %}
</div>
</body>
</html>
```

Masukkan base template ke setiap halaman (home,about,detail)

- Ketik paling atas

```
{% extends 'base.html' %}  
{% block content %}
```

- Ketik paling bawah

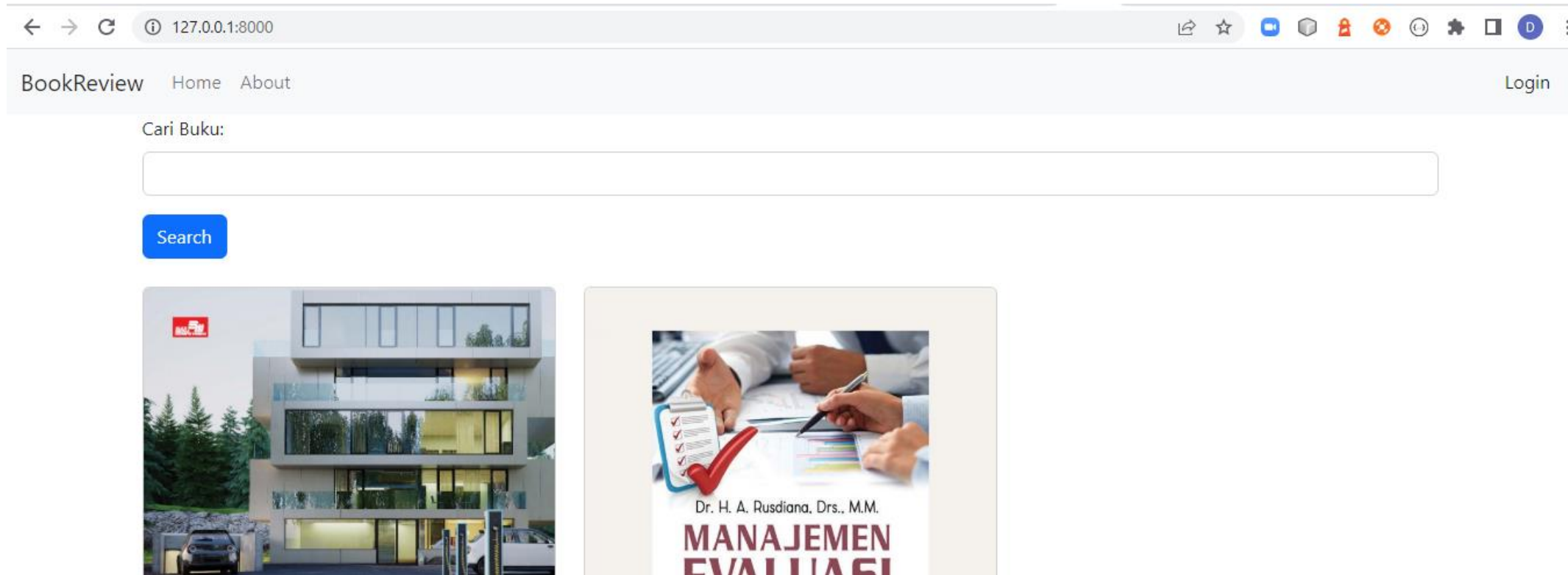
```
{% endblock content %}
```


Masukkan base template book/detail.html

```
{% extends 'base.html' %}
{% block content %}

<div class="card">
  
  <div class="card-body">
    <h5 class="card-title fw-bold">
      {{ book.judul }} | {{ book.pengarang }} | {{ book.tanggal_rilis|date:"d-m-Y" }}
    </h5>
    <p class="card-text">
      {{ book.sinopsis }}
      <br/>
      {{ book.isbn }} | {{ book.jumlah_halaman }} | {{ book.publisher.nama }}
    </p>
  </div>
</div>
{% endblock content %}
```

Hasil implementasi base template



Latihan menambah halaman dan menu

- Tambahkan menu contact
- Buat templates/contact.html
- Buat method ke halaman contact di book/views.py
- Buat path baru di bookreviews/urls.py
- Ubah menu di base.html

Latihan mini project

- Aplikasi Kantin Sekolah
- Kantin
 - nama (input text)
 - telpon (input telpon)
- Makanan
 - nama (input text)
 - harga (input int)
 - gambar (input text)
 - Stok (input int)
 - tersedia (radio button [Tersedia,Tidak Tersedia])
 - Kantin ForeignKey(Kantin,on_delete=models.CASCADE)

Latihan mini project

- Aplikasi Kantin Sekolah
- Halaman admin CRUD kantin,makanan
- Halaman Home menampilkan makanan kantin bisa search
- Halaman detail adalah detail dari makanan

Langkah-langkah

1. `python -m django startproject appkantin`

```
Command Prompt

Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd..



C:\Users>cd..

C:\>cd projectdjango

C:\projectdjango>python -m django startproject appkantin
```

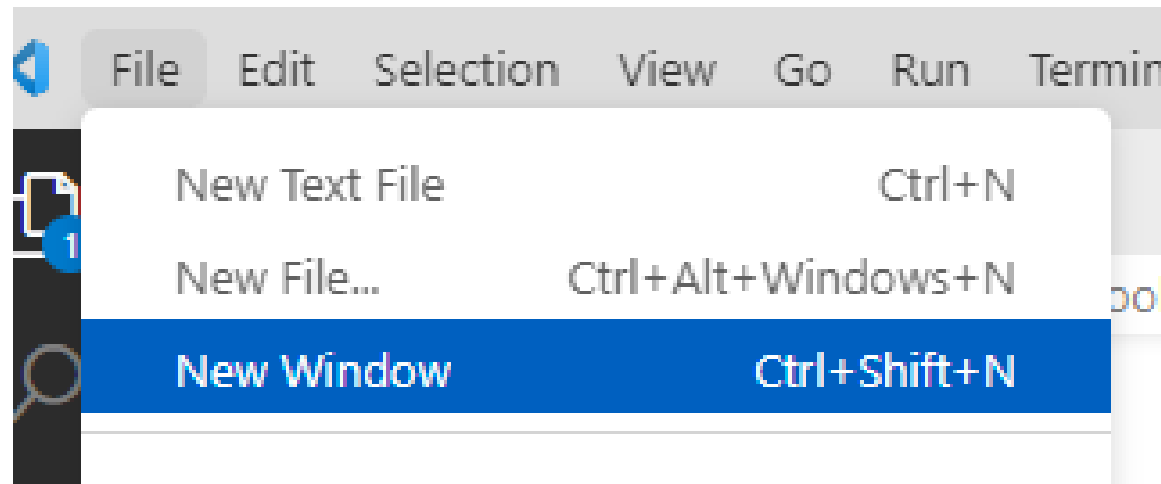
2. Rename folder (perhatikan : appkantinproject)

This PC > System (C:) > projectdjango > appkantinproject >

Name	Date modified	Type	Size
 appkantin	01/12/2022 08.18	File folder	
 manage.py	01/12/2022 08.18	Python File	1 KB

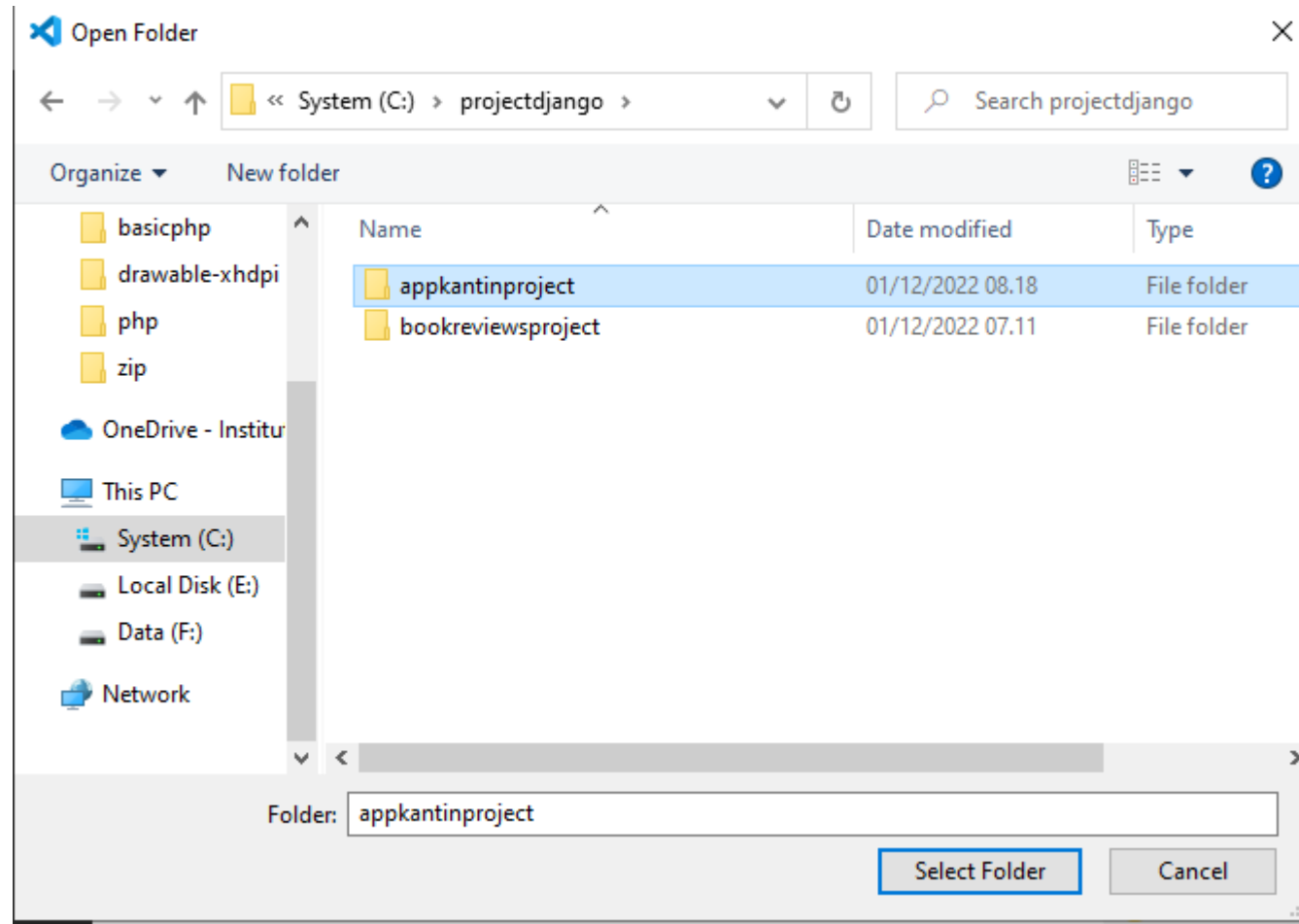
Langkah-langkah

3. Buka Visual Studio Code dengan New Window untuk dapat membuka 2 project (bookreviews dan appkantin). Membuka 2 window project memudahkan untuk melihat project bookreviews sebagai referensi



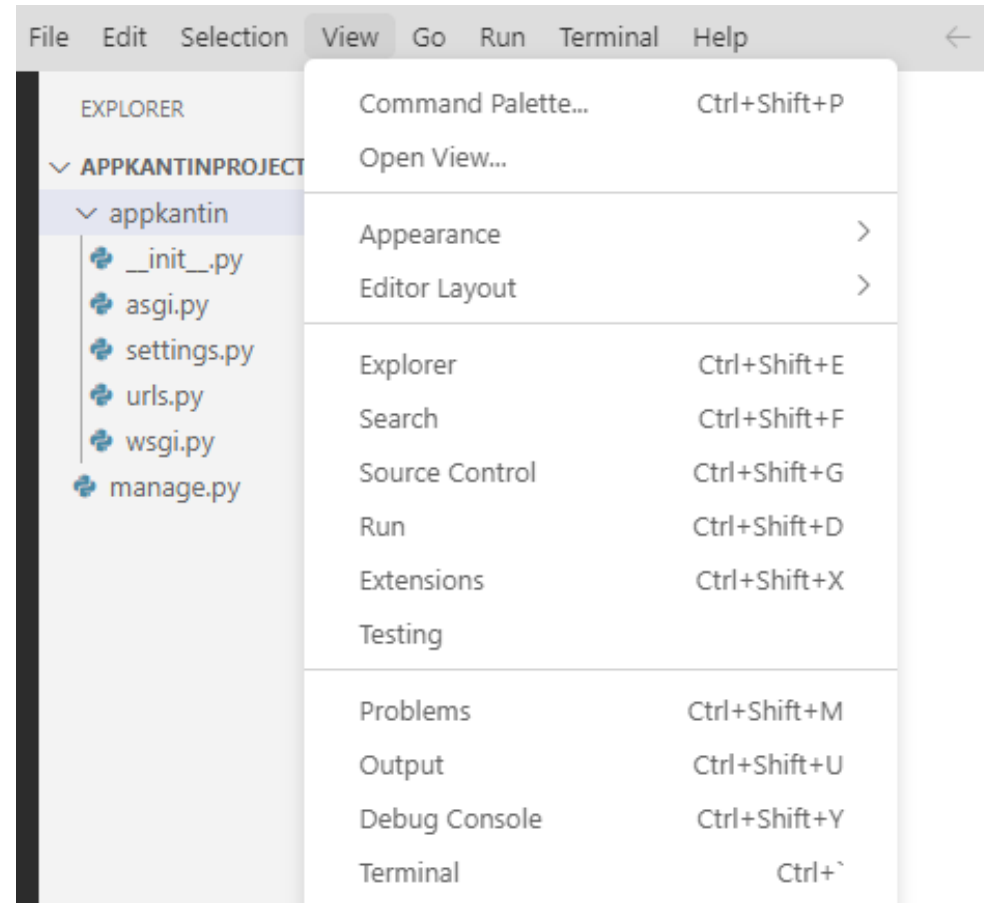
Langkah-langkah

4. Open Folder



Langkah-langkah

5. Pada VSCode View -> Terminal



Langkah-langkah

6. Ketik di terminal : **python manage.py runserver**. Sebelumnya pastikan bookreviews sudah dimatikan (ctrl + c) pada terminal di project bookreviews

```
PS C:\projectdjango\appkantinproject> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have 18 unapplied migration(s). Your project may not work properly.
ns.
```

```
Run 'python manage.py migrate' to apply them.
```

```
December 01, 2022 - 08:27:26
```

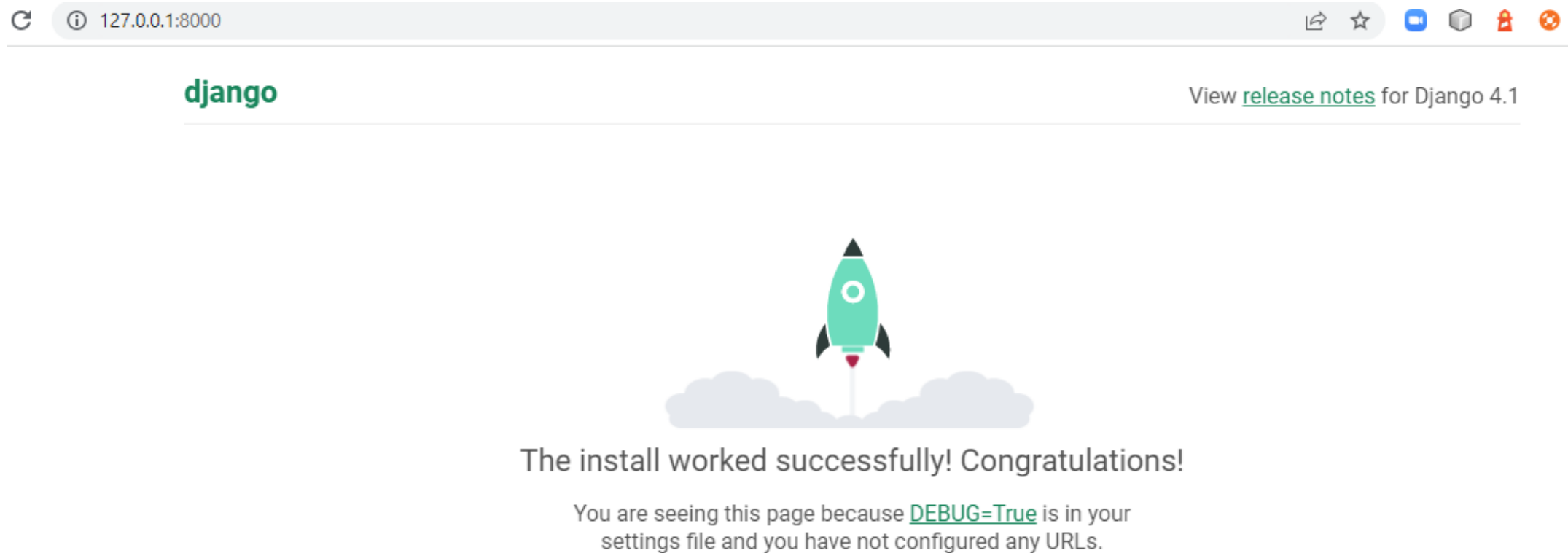
```
Django version 4.1.3, using settings 'appkantin.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CTRL-BREAK.
```

Langkah-langkah

7. Buka browser



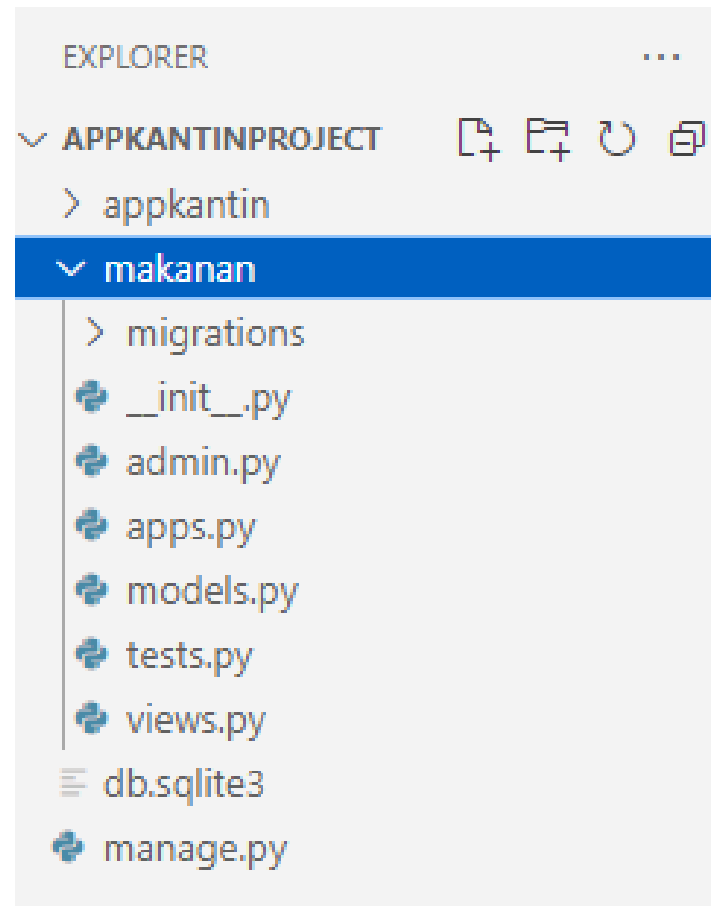
Langkah-langkah

8. Buka cmd, buat app makanan di folder appkantinproject

```
C:\projectdjango>cd appkantinproject  
C:\projectdjango\appkantinproject>python manage.py startapp makanan  
C:\projectdjango\appkantinproject>
```

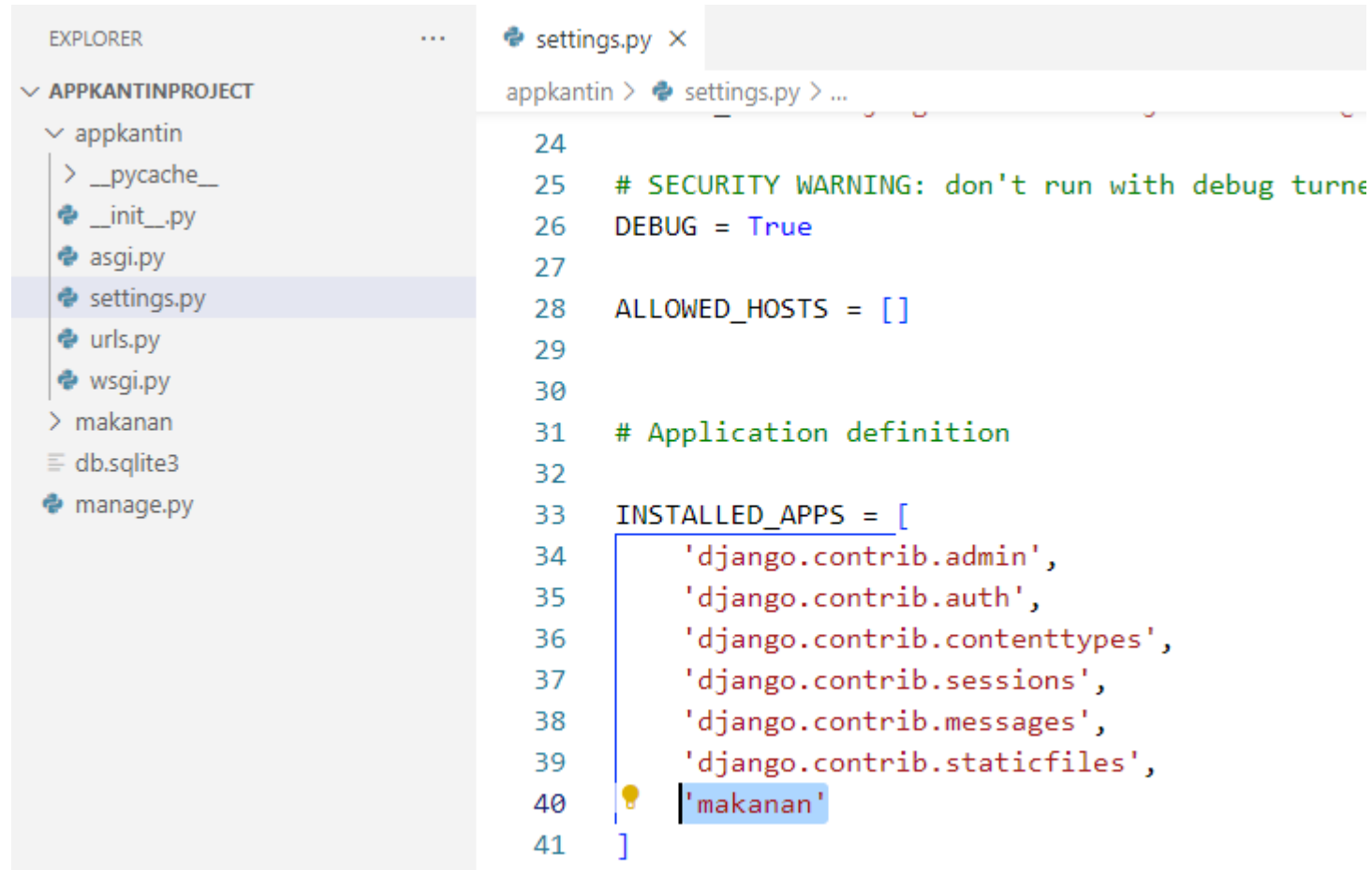
Langkah-langkah

9. Cek di VSCode



Langkah-langkah

10. Daftarkan app 'makanan' di settings.py



```
24
25 # SECURITY WARNING: don't run with debug turned on in production
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'makanan'
41 ]
```

Langkah-langkah

11. Buat Model (makanan/models.py)

```
models.py X
makanan > models.py > ...
1  from django.db import models
2  from django.core.validators import RegexValidator
3
4  # Create your models here.
5  class Kantin(models.Model):
6      nama = models.CharField(max_length=250)
7      phone_regex = RegexValidator(regex=r'^\+?1?\d{9,15}$', message="min 8 digit")
8      telpon = models.CharField(validators=[phone_regex], max_length=15, blank=True)
9      def __str__(self) -> str:
10         return self.nama
11
12  class Makanan(models.Model):
13      nama = models.CharField(max_length=250)
14      harga = models.IntegerField()
15      gambar = models.ImageField(upload_to='movie/images/')
16      stok = models.IntegerField()
17      kantin = models.ForeignKey(Kantin, on_delete=models.CASCADE)
18      TERSEDIA = (
19         ('1', 'Tersedia'),
20         ('0', 'Tidak tersedia')
21     )
22      tersedia = models.CharField(max_length=1, choices=TERSEDIA)
23      def __str__(self) -> str:
24         return self.nama
```

Langkah-langkah

12. Command : `python manage.py migrate`

```
C:\projectdjango\appkantinproject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```


Langkah-langkah

13. Command : **python manage.py makemigrations**

```
C:\projectdjango\appkantinproject>python manage.py makemigrations
Migrations for 'makanan':
  makanan\migrations\0001_initial.py
    - Create model Kantin
    - Create model Makanan
```

Langkah-langkah

13. Command : **python manage.py migrate**

```
C:\projectdjango\appkantinproject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, makanan, sessions
Running migrations:
  Applying makanan.0001_initial... OK
C:\projectdjango\appkantinproject>
```

Langkah-langkah

14. Create super user untuk admin : **python manage.py createsuperuser**. Misal username : admin, password:admin

Command Prompt

```
C:\projectdjango\appkantiproject>python manage.py createsuperuser
Username (leave blank to use 'windows'): admin
Email address: dennyc.dev@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

C:\projectdjango\appkantiproject>_
```

Langkah-langkah

15. Setting image asset (appkantin/settings.py)

Buka menggunakan VSCode buka appkantin/settings.py

Tambahkan kode berikut paling bawah

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')  
MEDIA_URL = '/media/'
```

Tambahkan paling atas

```
import os
```

Langkah-langkah

16. Setting image asset (appkantin/urls.py)

```
from django.contrib import admin
from django.urls import path
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
]
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

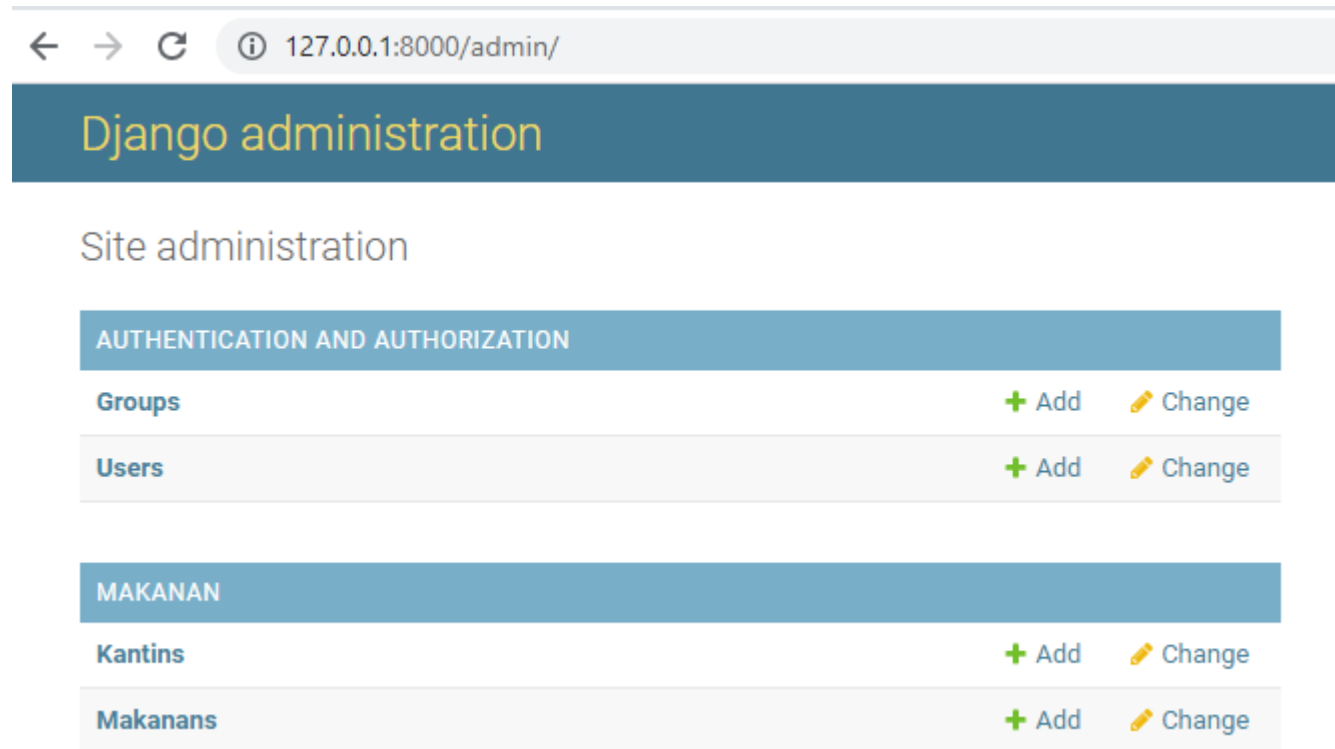
Langkah-langkah

17. Register model di makanan/admin.py

```
from django.contrib import admin
from .models import Kantin
from .models import Makanan
# Register your models here.
admin.site.register(Kantin)
admin.site.register(Makanan)
```

Langkah-langkah

18. Login (pastikan server jalan, jika tidak jalankan di terminal VSCode : `python manage.py runserver`). CRUD kantin, makanan sudah selesai.



Membuat View di Home

19. Membuat view di makanan/views.py

```
from django.shortcuts import render
from django.http import HttpResponse
from .models import Makanan
from django.shortcuts import get_object_or_404

# Create your views here.
def home(request):
    searchMakanan = request.GET.get('nama')
    if searchMakanan:
        makanans = Makanan.objects.filter(nama__icontains=searchMakanan)
    else:
        makanans = Makanan.objects.all()
    return render(request, 'home.html', {'searchMakanan':searchMakanan, 'makanans' : makanans})

def detail(request, makanan_id):
    makanan = get_object_or_404(Makanan, pk=makanan_id)
    return render(request, 'detail.html', {'makanan' : makanan})
```


Langkah-langkah

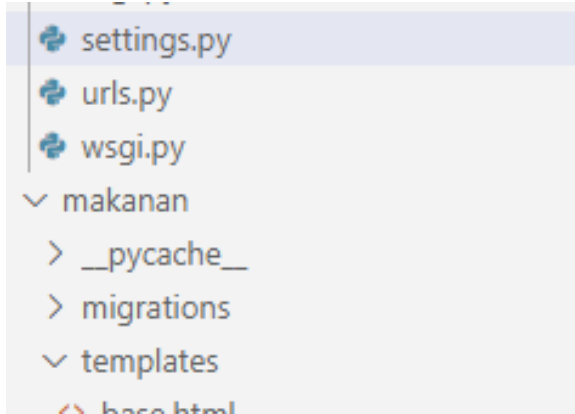
20. Siapkan url (appkantin/urls.py)

```
from django.contrib import admin
from django.urls import path
from django.conf.urls.static import static
from django.conf import settings
from makanan import views as makananView

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', makananView.home, name='home'),
    path('makanan/<int:makanan_id>', makananView.detail,
name='detail')
]
urlpatterns +=
static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Langkah-langkah

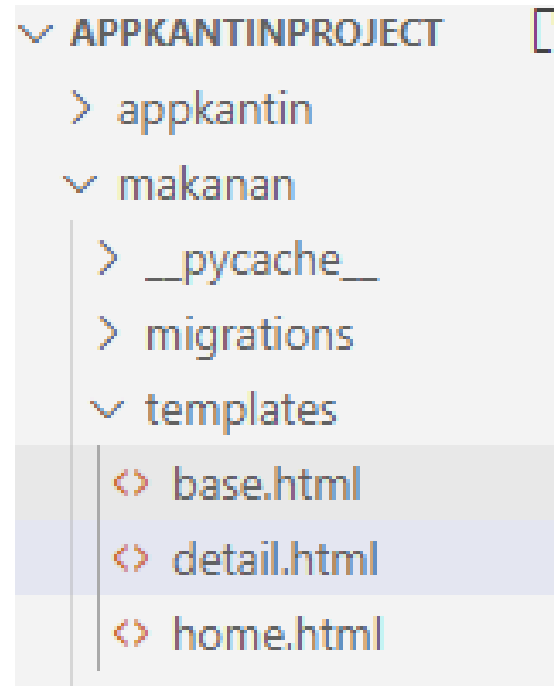
21. Seting template (appkantin/settings.py)



```
52 ]  
53  
54 ROOT_URLCONF = 'appkantin.urls'  
55  
56 TEMPLATES = [  
57     {  
58         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
59         'DIRS': [os.path.join(BASE_DIR, 'makanan/templates')],
```

Langkah-langkah

22. Buat template untuk base.html, home.html dan detail.html (File kosong dulu)



Langkah-langkah

23. Buat template untuk base.html

- Copy dari base.html di project **bookreviews**
- Sesuaikan

Langkah-langkah

24. Buat template untuk home.html

- Copy dari home.html di project **bookreviews**
- Sesuaikan

Langkah-langkah

25. Buat template untuk detail.html

- Copy dari detail.html di project **bookreviews**
- Sesuaikan

Menambah pemesanan makanan dan halaman Cart

- Di detail makanan terdapat form untuk memasukkan jumlah pesanan

Ayam bakar | 150000

10 | Kantin AA

Jumlah Pesanan:

Pesan

Menambah pemesanan makanan dan halaman Cart

- Klik Pesan maka kehalaman **Pesanan**

Kantin Home Pesanan Login

Item	Harga	Jumlah	Sub total	Aksi
Ayam bakar	Rp150000	3	Rp450000.00	Hapus
sate	Rp10000	1	Rp10000.00	Hapus

Total : Rp460000.00

- Jika klik hapus maka pesanan dihapus

Kode di makanan/models.py

- Tambahkan kode ini paling bawah

```
class CartItem(models.Model):  
    makanan = models.ForeignKey(Makanan, on_delete=models.CASCADE)  
    quantity = models.PositiveIntegerField(default=1)  
    item_total = models.DecimalField(max_digits=10, decimal_places=2,  
blank=True, null=True)
```

Kode CartItem di makanan/views.py

- Tambahkan kode ini paling bawah

```
def add_to_cart(request, makanan_id):
    makanan_id = int(makanan_id)
    makanan = Makanan.objects.get(pk=makanan_id)

    if request.method == 'POST':
        quantity = int(request.POST.get('quantity', 1)) # jumlah pesanan dari form
        if quantity < 1:
            quantity = 1 # minimal 1

        # cart_item jika sudah ada pesanan makanan tersebut
        # created jika belum ada pesanan makanan tersebut maka di simpan dan created = true
        cart_item, created = CartItem.objects.get_or_create(makanan=makanan)

        # If the item exists, update the quantity and item_total
        if not created:
            cart_item.quantity += quantity
            cart_item.item_total = cart_item.quantity * makanan.harga
            cart_item.save()
        else:
            cart_item.quantity = quantity
            cart_item.item_total = quantity * makanan.harga
            cart_item.save()

    return redirect('cart')
```

Kode CartItem di makanan/views.py

- Tambahkan kode ini paling bawah

```
def view_cart(request):
    cart_items = CartItem.objects.all()
    total_price = sum(item.item_total for item in cart_items)

    return render(request, 'cart.html', {'cart_items': cart_items, 'total_price': total_price})

def remove_from_cart(request, makanan_id):
    makanan_id = int(makanan_id)
    makanan = Makanan.objects.get(pk=makanan_id)

    try:
        cart_item = CartItem.objects.get(makanan=makanan)
        cart_item.delete()
    except CartItem.DoesNotExist:
        pass

    return redirect('cart')
```

Kode form pesan makanan di detail.html

```
<!-- Add to Cart form with quantity input -->
<form method="post" action="{% url 'add_to_cart' makanan.id %}">
  {% csrf_token %}
  <label for="quantity">Jumlah Pesanan:</label>
  <input type="number" name="quantity" id="quantity" value="1" min="1">
  <button type="submit" class="btn btn-success">Pesanan</button>
</form>
```

Kode di cart.html

```
{% extends 'base.html' %}
{% block content %}
<div class="table-responsive">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>Item</th>
        <th>Harga</th>
        <th>Jumlah</th>
        <th>Sub total</th>
        <th>Aksi</th>
      </tr>
    </thead>
    <tbody>
      {% for item in cart_items %}
      <tr>
        <td>{{ item.makanan.nama }}</td>
        <td>Rp{{ item.makanan.harga }}</td>
        <td>{{ item.quantity }}</td>
        <td>Rp{{ item.item_total }}</td>
        <td>
          {% if item.makanan.id %}
          <a href="{% url 'remove_from_cart' item.makanan.id %}">
            <button class="btn btn-danger">Hapus</button>
          </a>
          {% endif %}
        </td>
      </tr>
      {% endfor %}
    </tbody>
  </table>
</div>
<p>Total : Rp{{ total_price }}</p>
{% endblock content %}
```

Kode tambahan di urls.py

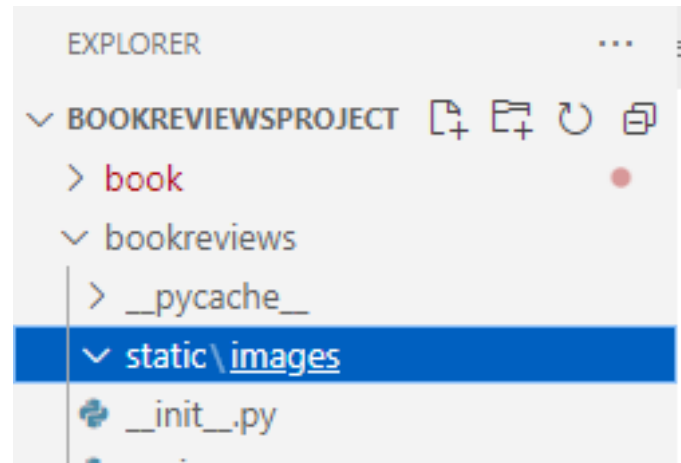
```
path('cart/', makananView.view_cart, name='cart'),  
path('add_to_cart/<int:makanan_id>', makananView.add_to_cart,  
name='add_to_cart'),  
path('remove_from_cart/<int:makanan_id>', makananView.remove_from_cart,  
name='remove_from_cart')
```

Tambah menu di base.html

```
<li class="nav-item">  
    <a class="nav-link" href="{% url 'cart' %}">Pesanan</a>  
</li>
```

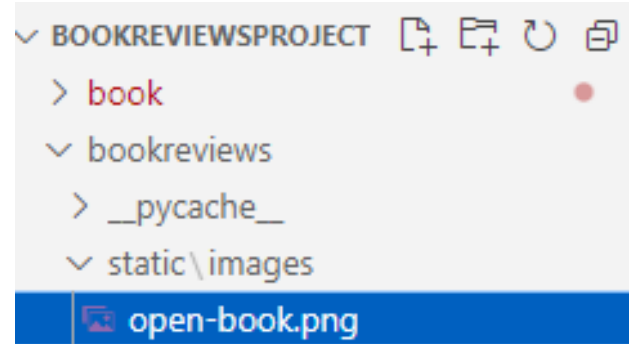
Materi Tambahan (static file) pada project (bookreviews)

- Menambahkan file statis (bisa gambar,css,javascript,pdf)
- Misal akan menambahkan icon dari :
https://www.flaticon.com/free-icon/open-book_81467?related_id=81467&origin=pack
- Setelah download nama filenya 'open-book.png'
- Buat folder di dalam bookreviews (static > images)



Materi Tambahan (static file)

- Letakkan file 'open-book.png' di dalam folder images



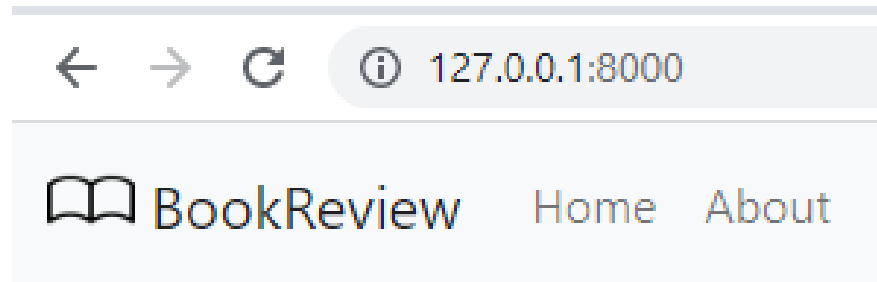
- Buka file bookreviews/settings.py, letakkan kode berikut paling bawah

```
STATICFILES_DIRS = [  
    BASE_DIR / 'bookreviews/static/',  
]
```

Materi Tambahan (static file)

- Mengakses file static di template (base.html)

```
<a class="navbar-brand" href="{% url 'home' %}">
    {% load static %}
    
    BookReview</a>
```



Cari Buku:

Materi Tambahan melakukan migrasi ke database MySQL

- Asumsi sudah ada database di MySQL dengan nama 'bookreviews'
- Install : `pip install pymysql`
- Letakkan kode berikut di `bookreviews/__init__.py`

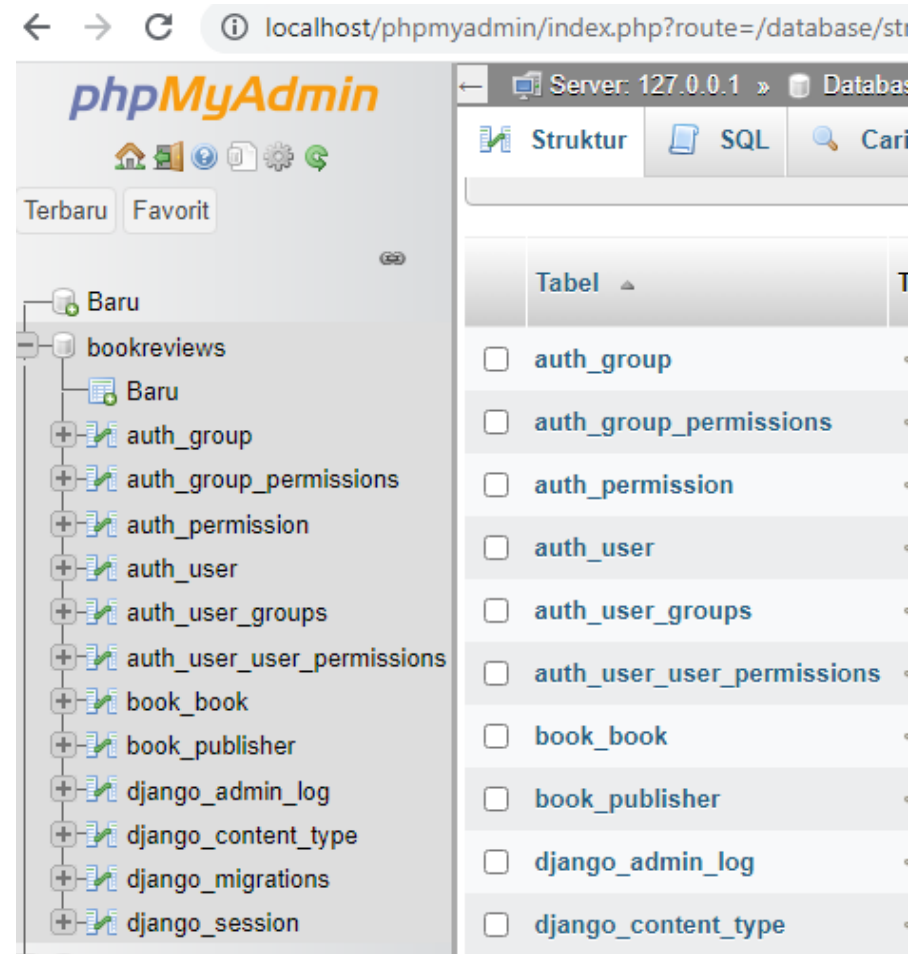
```
import pymysql  
pymysql.install_as_MySQLdb()
```

- Ganti property database di `settings.py`

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'bookreviews',  
        'USER': 'root',  
        'PASSWORD': '',  
        'HOST': 'localhost',    # Or an IP Address that your DB is hosted on  
        'PORT': '3306',  
    }  
}
```

Materi Tambahan melakukan migrasi ke database MySQL

- Lakukan migrasi : `python manage.py migrate`





TERIMA KASIH.