

Start



Table of contents

← Back

Next →

Table of contents

01

Django & MVT

- Create Project
- Create App
 - MVT

02

URLS

- Urls?
- How to Use Urls?
- URL Parameter

03

View & Template

- Show String
- Create Template
- Show HTML

04

Model

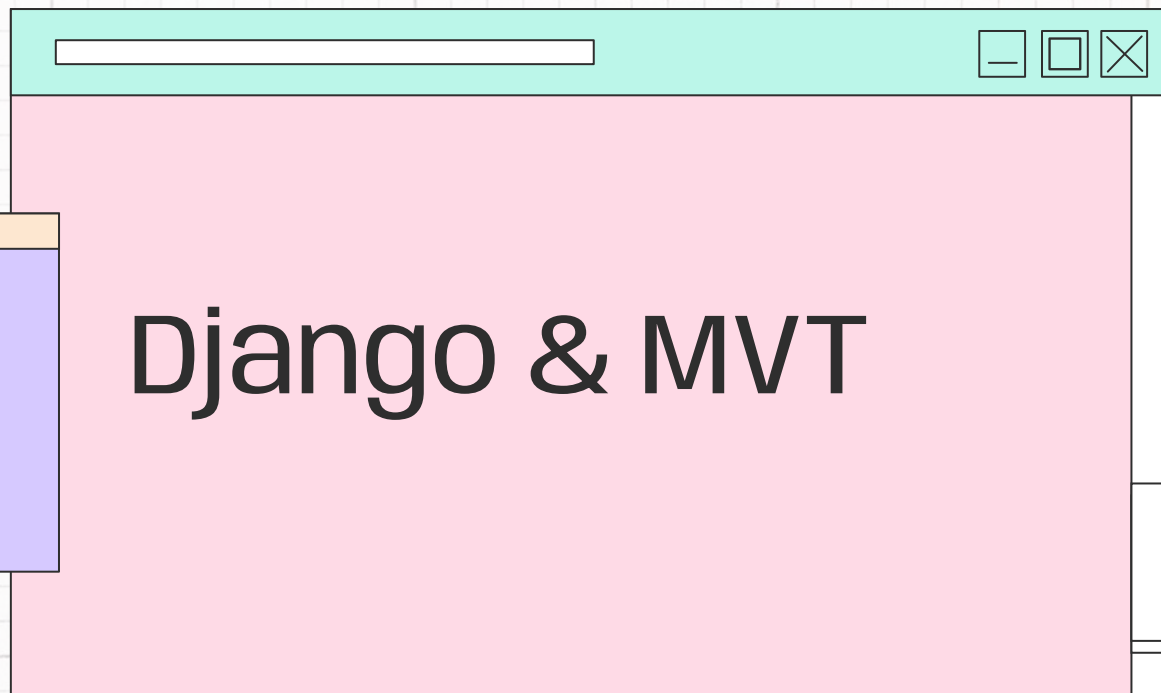
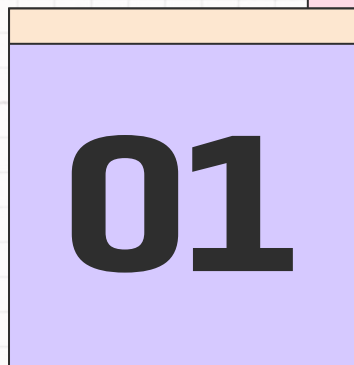
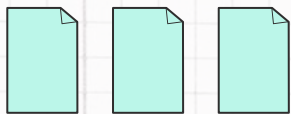
- Data Fields
- Makemigrations & Migrate

05

Practice

- Create a Django App





Prerequisite

1

Pastikan python sudah di install di device yang digunakan

<https://www.python.org/downloads/>

`python --version`

`pip --version`

2

Buat sebuah venv (Virtual Environment) lalu aktifkan

`python -m venv env`

`env\Scripts\activate.bat (Windows)`

`source env/bin/activate (MacOS)`

3

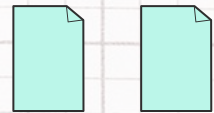
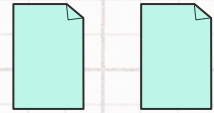
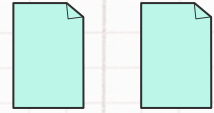
Install Django

`python -m pip install Django`



Project vs App

Project adalah kumpulan dari berbagai app. Sempelnya jika **Google** merupakan sebuah project, maka **Drive, Gmail, Assistant, Translate, dan lainnya** adalah sebuah app

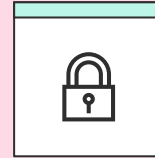


Create Project and App



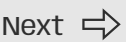
Create Django Project

`django-admin startproject <project_name>`



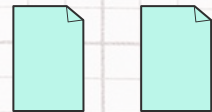
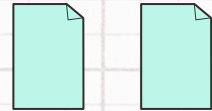
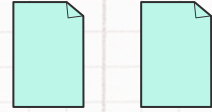
Create Django App

`python manage.py startapp <app_name>`



requirements.txt

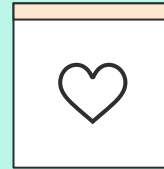
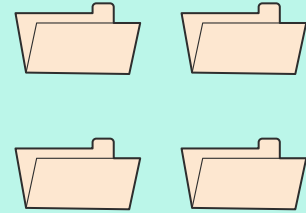
Digunakan untuk menyimpan dependensi proyek seperti library apa saja yang digunakan, kita perlu membuat file ini secara manual di root folder project Django kita



Project Structure

/root_folder

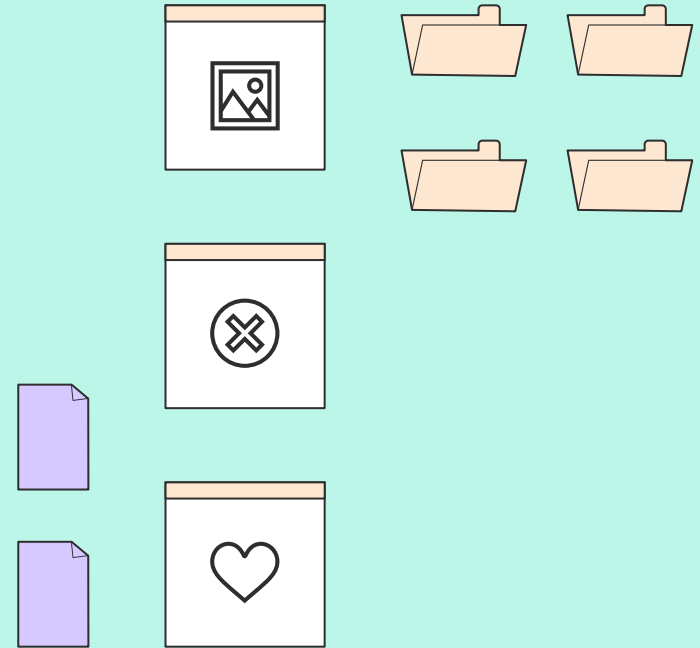
- <project_folder> (folder pusat project)
- <app_1_folder> (folder suatu app)
- <app_2_folder> (folder suatu app)
- ...
- db.sqlite3 (database project)
- manage.py (program menjalankan django command)
- requirements.txt (menyimpan dependensi proyek)



Project Structure

/root_folder/<project_folder>

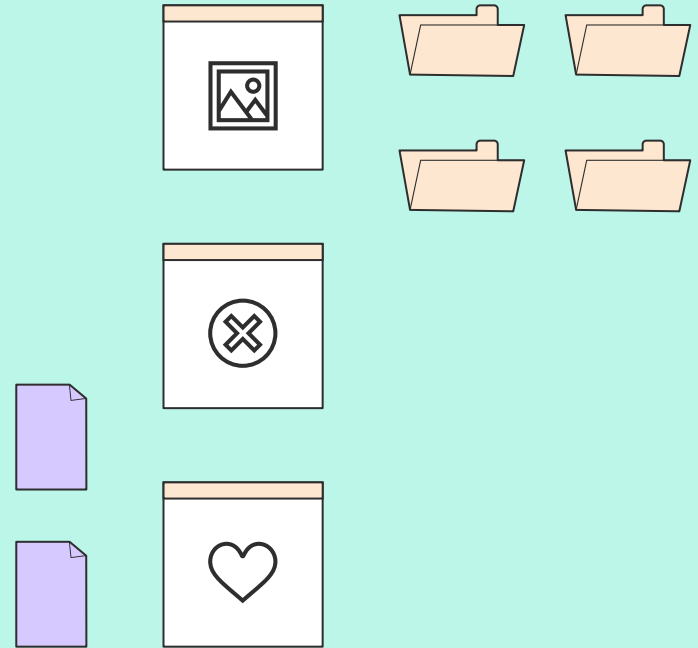
- asgi.py (mengatur asgi)
- settings.py (mengatur proyek)
- urls.py (memetakan alur proyek)
- wsgi.py (mengatur wsgi)



Project Structure

/root_folder/<app_folder>

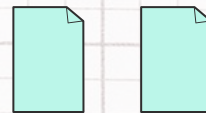
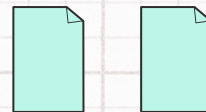
- /migrations (menyimpan hasil migrasi)
- /templates (meletakkan file template html)
- admin.py (registrasi model ke admin)
- apps.py (file tentang app)
- models.py (membuat model object)
- tests.py (melakukan pengujian unit)
- urls.py (menyusun path (alur halaman))
- views.py (membuat logika dari database ↔ template)



Run Django

```
python manage.py runserver
```

Merupakan command yang digunakan untuk menjalankan proyek Django (defaultnya berjalan di port 8000 dalam computer)



Model

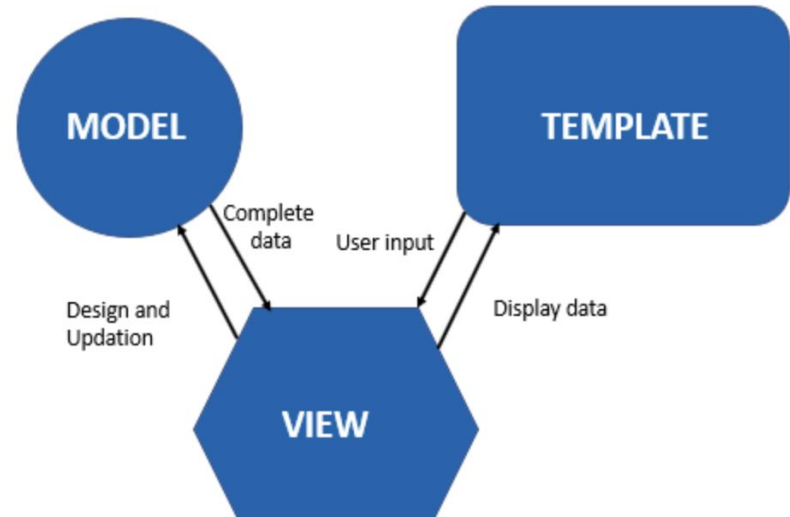
Berfungsi untuk mendefinisikan object ke dalam table database dan mengatur relasi diantara object-objectnya. Contohnya object "Mata Kuliah" didefinisikan memiliki field nama, jumlah_sks, dan jadwal.

View

Berfungsi untuk mengatur logika dan alur program serta data yang dikirimkan oleh pengguna ke dalam database maupun dari database ke pengguna. View berfungsi menjembatani Model dan Template. Contohnya mengatur total mata kuliah yang diambil tidak boleh lebih dari 24 sks di dalam sistem.

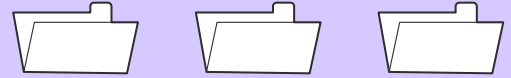
Template

Berfungsi untuk menampilkan halaman website ke pengguna. Tampilan depan yang dapat dilihat oleh pengguna website berupa HTML yang diproses oleh browser. Contohnya menampilkan pengambilan mata kuliah pengguna dan pesan dari server jika pengguna tidak memenuhi kriteria pengambilan mata kuliah,



Quiz – Django & MVT

View Django ngapain?



A

Mendefinisikan object ke dalam database

B

Menampilkan tampilan data kepada pengguna website

C

Menjembatani Template dan Model



Start



Table of contents



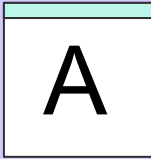
Back



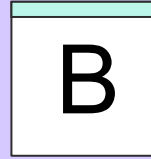
Next

Quiz – Django & MVT

View Django ngapain?



Mendefinisikan object ke dalam database



Menampilkan tampilan data kepada pengguna website



Menjembatani Template dan Model



Start



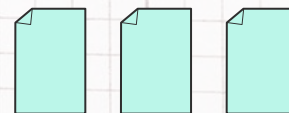
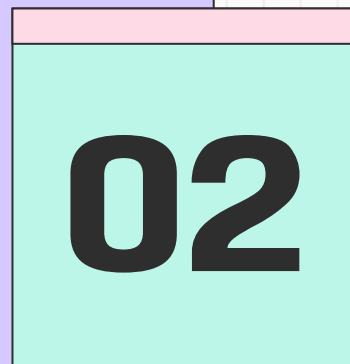
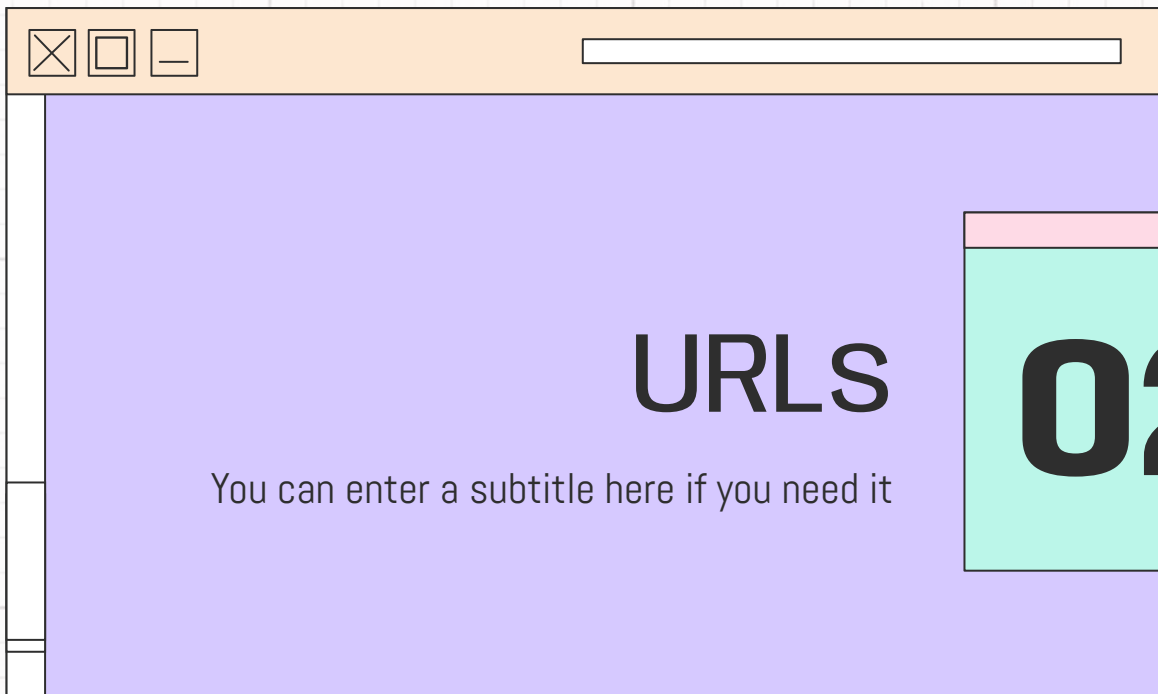
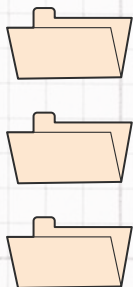
Table of contents



Back




Next





Path?



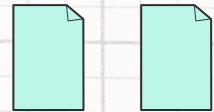
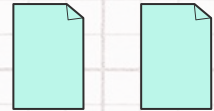
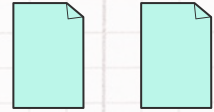


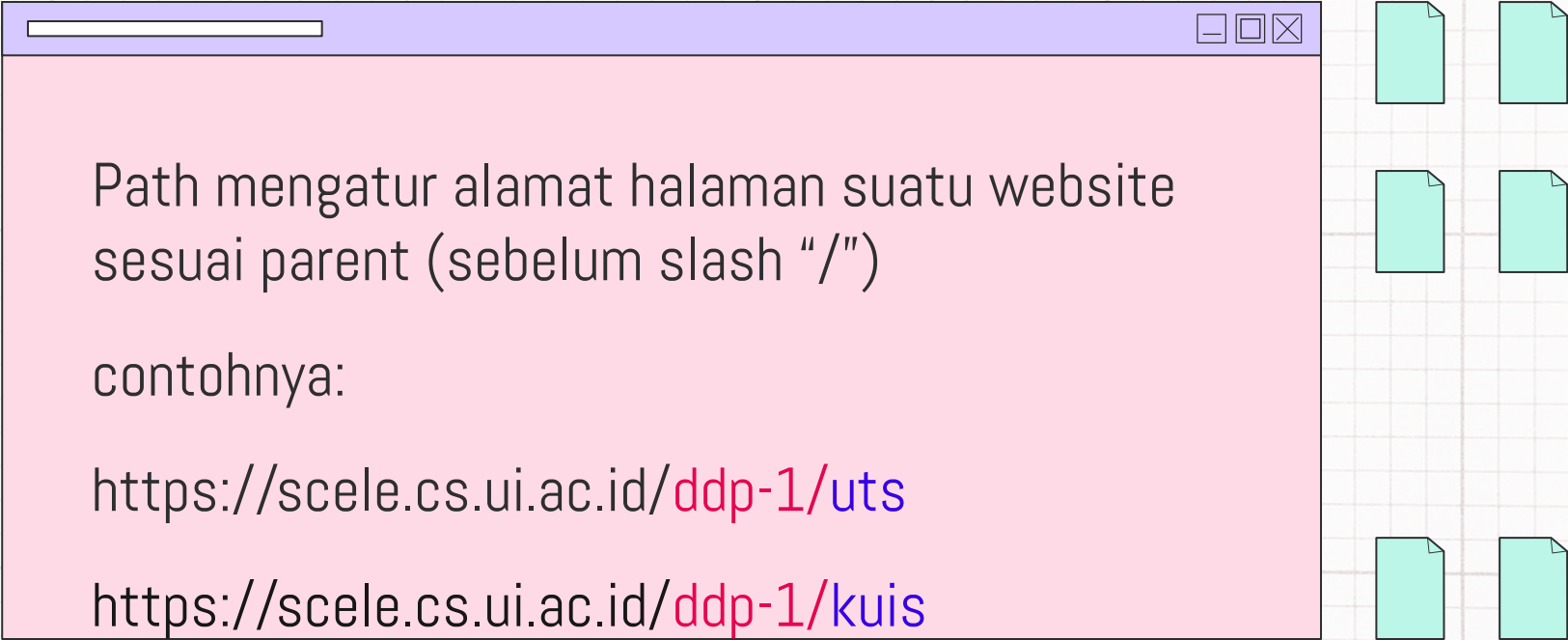
Path adalah sebuah jalur akses halaman website
(biasa dikenal dengan url address)

contohnya:

<https://scele.cs.ui.ac.id/ddp-1>

<https://scele.cs.ui.ac.id/kalkulus-1>






Path mengatur alamat halaman suatu website
sesuai parent (sebelum slash "/")

contohnya:

<https://scele.cs.ui.ac.id/ddp-1/uts>

<https://scele.cs.ui.ac.id/ddp-1/kuis>

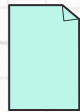
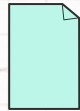
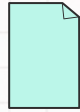


Path juga dapat disusun secara dinamis (dapat. berubah) dengan menggunakan URL parameter

contohnya:

<https://scele.cs.ui.ac.id/kalkulus-1/kuis/1>

<https://scele.cs.ui.ac.id/kalkulus-1/kuis/2>



project/urls.py

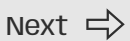
```
from django.contrib import admin
from django.urls import path, include
```

```
urlpatterns = [
    path("", include('homepage.urls')),
    path('admin/', admin.site.urls),
    path('app/', include('app.urls'))
]
```

merah : merupakan jalur yang disusun

hijau : merupakan file berisi path yang ingin ditambahkan dengan parent app/ (jalur yang disusun di sebelumnya), lihat slide halaman berikutnya

include : merupakan fungsi untuk menambahkan semua path di dalam aplikasi tertentu



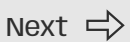
app/urls.py (buat file secara manual dengan nama urls.py di dalam app)

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path("", views.a),
    path('<int:angka>', views.b),
    path('<str:nama>', views.c)
]
```

merah : merupakan jalur yang disusun, '<>' berarti jalur tersebut dinamis (bisa berubah-ubah sesuai fieldnya <type_data:variabel_argument>

hijau : merupakan fungsi di dalam file views yang dijalankan ketika pengguna membuka jalur di sebelah kirinya



Default port yang digunakan Django adalah 8000, artinya kita bisa jalanin Django dengan url localhost:8000

contohnya:

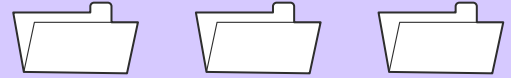
localhost:8000/ (render homepage)

localhost:8000/app (render template oleh app views.a)

localhost:8000/app/1 (render template oleh app views.b)



Quiz - URLs



localhost:8000/app/pti/ jalanin path yang mana?

A

`path("", views.a)`

B

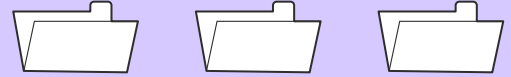
`path('<str:nama>/', views.b)`

C

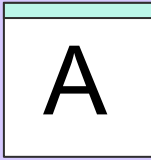
`path('<int:angka>/', views.c)`



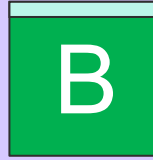
Quiz - URLs



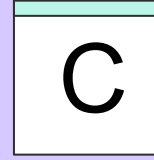
localhost:8000/app/pti/ jalanin path yang mana?



`path("", views.a)`



`path('<str:nama>/', views.b)`



`path('<int:angka>/', views.c)`



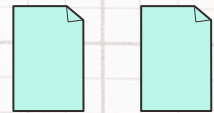
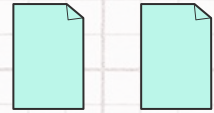
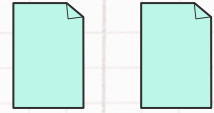
03

View & Template



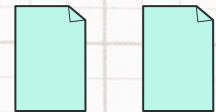
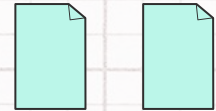
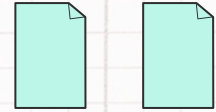
HttpResponse()

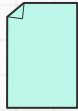
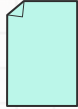
Fungsi yang mengembalikan tampilan **argumennya** berupa **string**
ke tampilan depan **website secara langsung**



Render()

Fungsi yang mengembalikan **data** yang sudah diolah ke dalam **template HTML** melalui **context**





project/urls.py

REVIEW

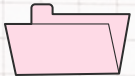
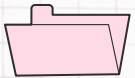
```
from django.contrib import admin
from django.urls import path, include
```

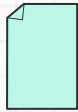
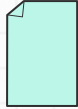
```
urlpatterns = [
    path("", include('homepage.urls')),
    path('admin/', admin.site.urls),
    path('app/', include('app.urls')) # untuk app/urls masuknya ke app/
]
```

merah : merupakan jalur yang disusun

hijau : merupakan file berisi path yang ingin ditambahkan dengan parent app/ (jalur yang disusun di sebelumnya), lihat slide halaman berikutnya

include : merupakan fungsi untuk menambahkan semua path di dalam aplikasi tertentu





app/urls.py

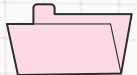
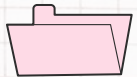
```
from django.urls import path  
from . import views
```

```
urlpatterns = [  
    path("", views.a), # berarti 'app/' hasil include dari project urls.py  
    path('<int:angka>', views.b),  
    path('<str:nama>', views.c)  
]
```

merah : merupakan jalur yang disusun, '<>' berarti jalur tersebut dinamis (bisa berubah-ubah sesuai fieldnya <type_data:variabel_argument>

hijau : merupakan fungsi di dalam file views yang dijalankan ketika pengguna membuka jalur di sebelah kirinya

REVIEW



app/views.py

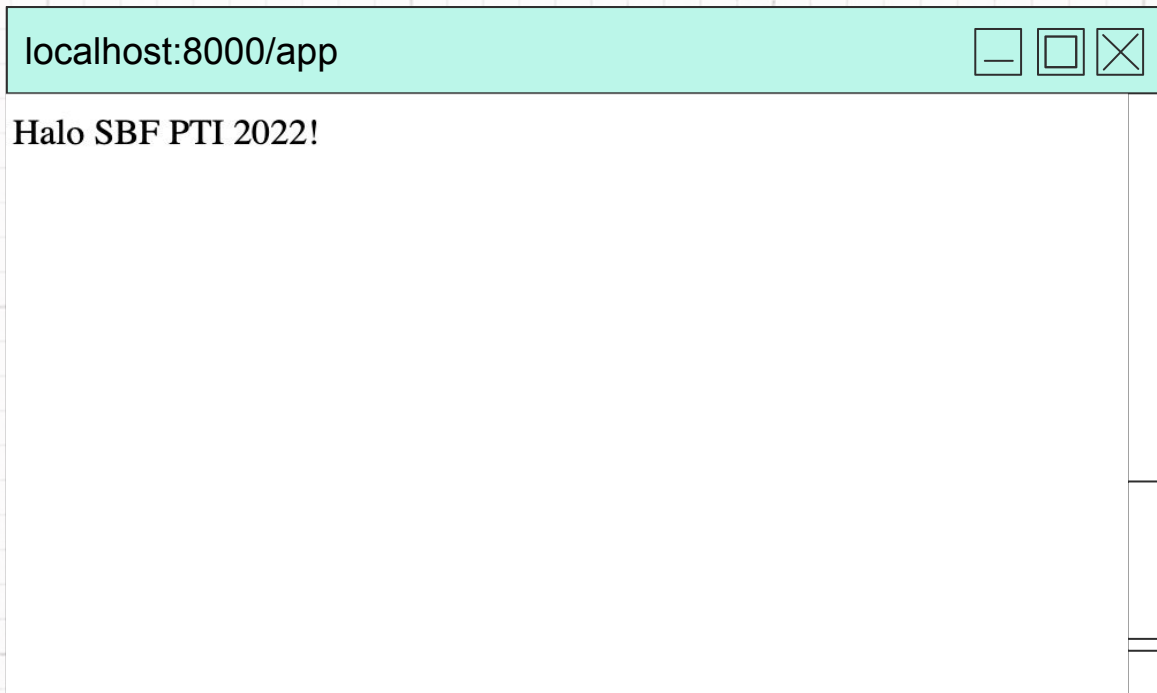
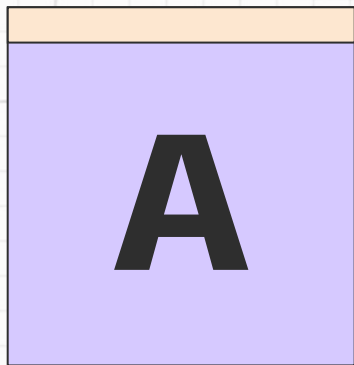
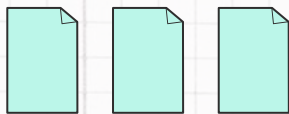
```
from django.shortcuts import render
from django.http import HttpResponse

def a(request):
    return HttpResponse("Halo SBF PTI 2022!")

def b(request, angka):
    return HttpResponse("Halo No %d, SBF PTI 2022! % angka")

def c(request, nama):
    context = {
        'nama': nama,
        'biro': "PTI"
    }
    return render(request, 'app/index.html', context)
```





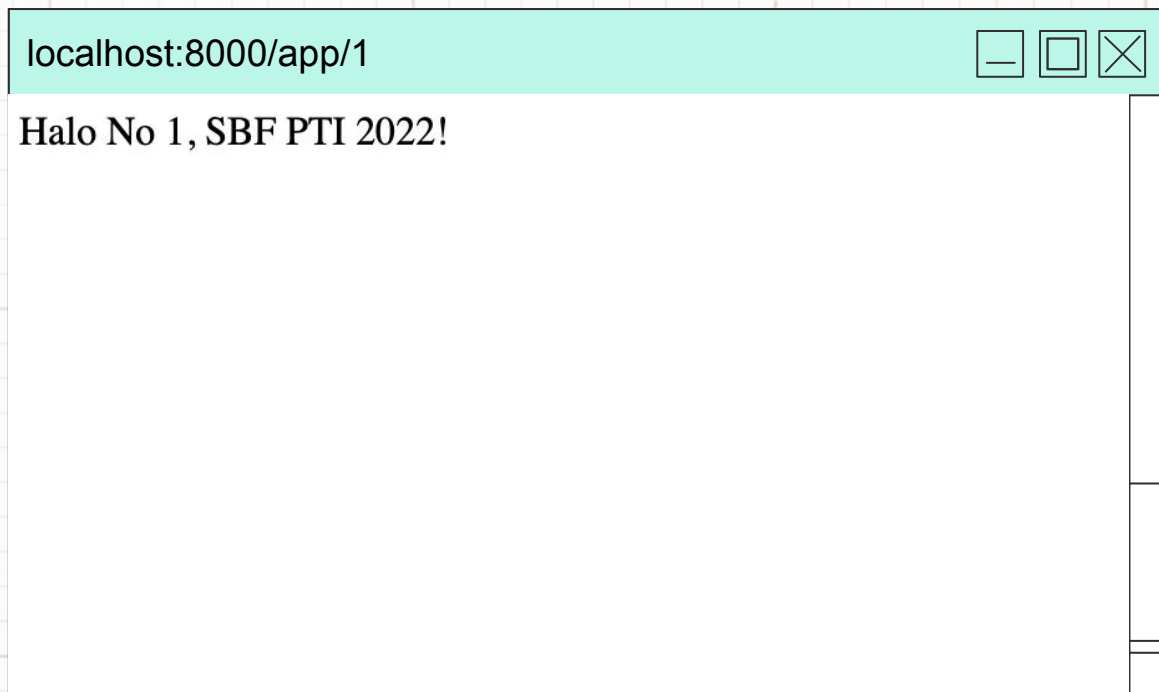
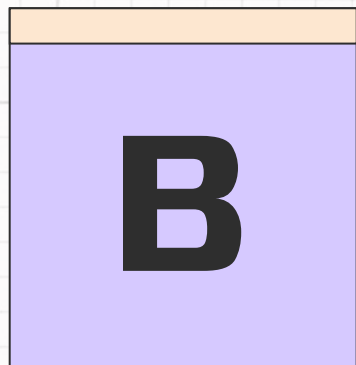
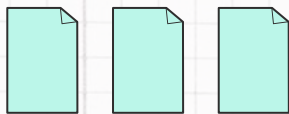
Start

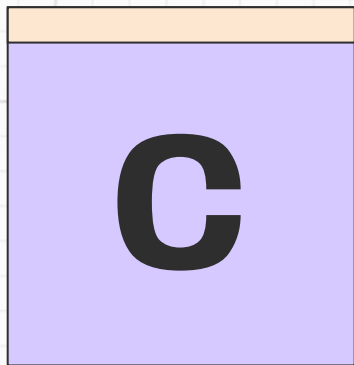
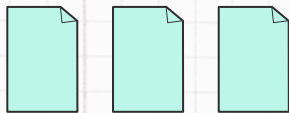


Table of contents

← Back

Next →

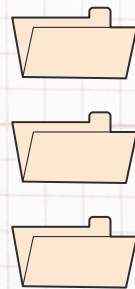




app/templates/app/index.html



```
<html>
  <body>
    <p style="color:red;">Salam kenal {{ nama }} dari biro {{ }}!</p>
  </body>
</html>
```



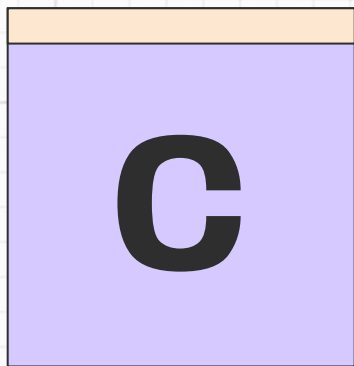
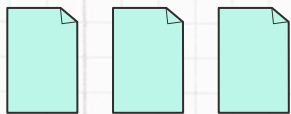
Start



Table of contents

← Back

Next →



localhost:8000/app/mario



Salam kenal Mario dari biro PTI!



Start

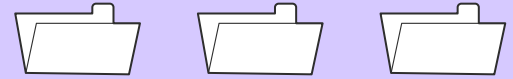


Table of contents

← Back

Next →

Quiz – View & Template



Fungsi mengembalikan string langsung ke website?

A

`HttpResponse()`

B

`HttpRequest()`

C

`render()`



Start

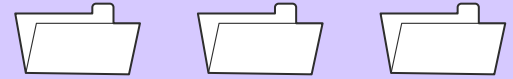


Table of contents

← Back

Next →

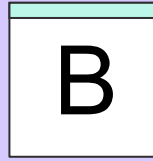
Quiz – View & Template



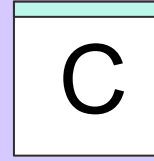
Fungsi mengembalikan string langsung ke website?



`HttpResponse()`



`HttpRequest()`



`render()`



Start



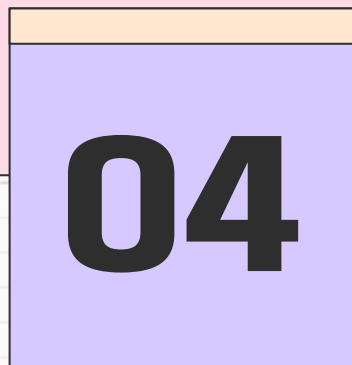
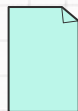
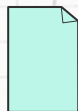
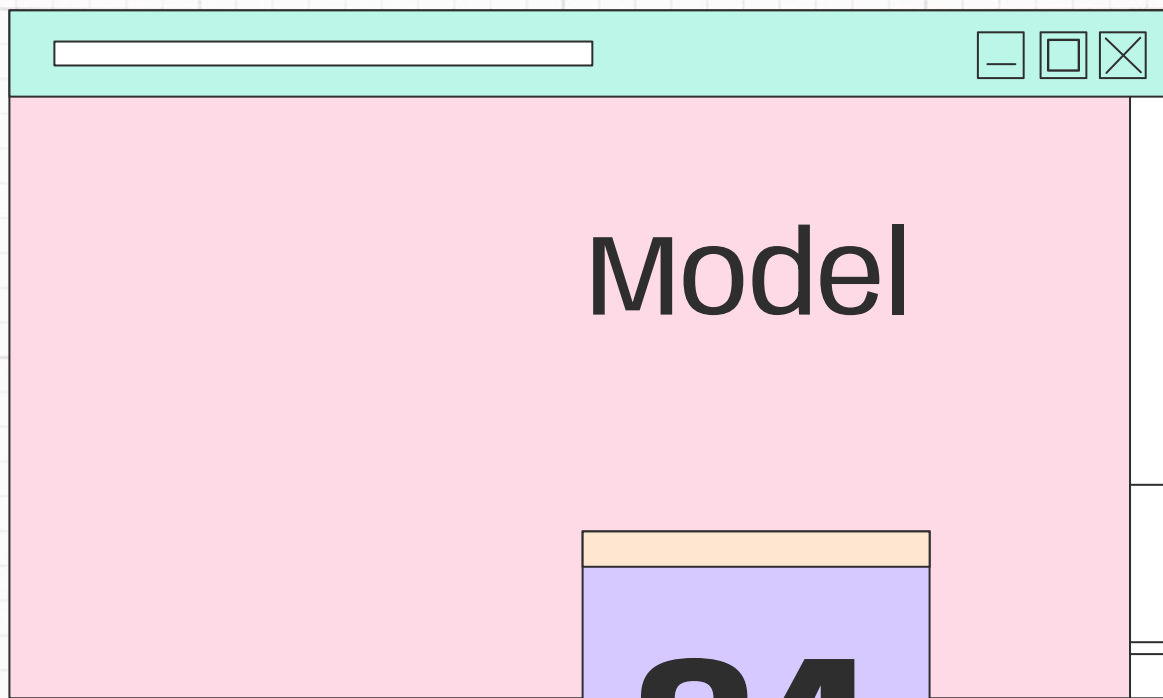
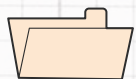
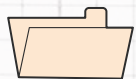
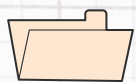
Table of contents



Back

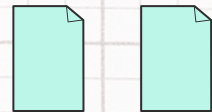
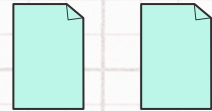
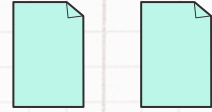


Next



Class & Attributes

Class merupakan cetakan dari suatu object sedangkan attributes adalah hal yang dimiliki oleh object tersebut



app/models.py

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

```
class Matakuliah(models.Model):  
    nama = models.CharField(max_length=200, unique=True)  
    jumlah_sks = models.IntegerField()  
    jadwal = models.DateField()
```

hijau : merupakan variabel atribut dari kelas Matakuliah

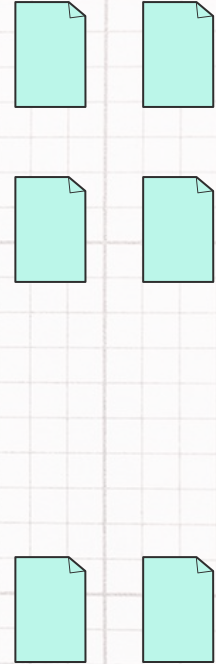
merah : merupakan tipe data atribut

biru : merupakan argument tipe data yang digunakan untuk mengatur batasan argument



Add Object -> Database Table

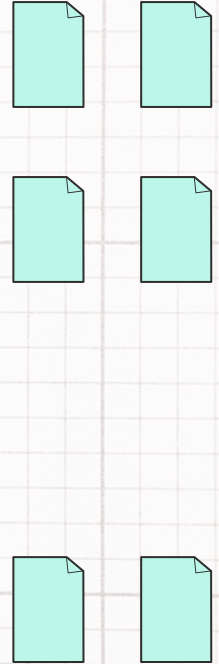
Object yang telah disusun berupa sebuah class di dalam app/models.py perlu dilakukan **migrasi** agar dapat tersimpan ke dalam **tabel database**



Makemigrations

`python manage.py makemigrations`

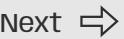
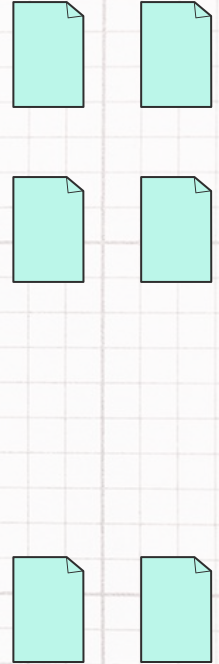
Berfungsi untuk mendaftarkan object yang disusun ke dalam proyek Django

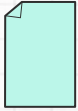
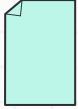


Migrate

`python manage.py migrate`

Berfungsi untuk menjalankan migrasi pada object yang telah terdaftar di dalam proyek Django



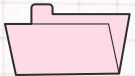
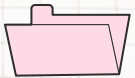


app/admin.py

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

```
admin.site.register(Matakuliah)
```

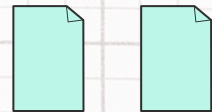
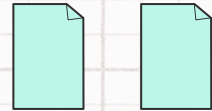
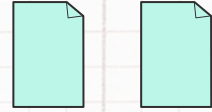
jangan lupa untuk menambahkan Matakuliah ke dalam admin.py agar model ini memiliki interface di dalam Django admin



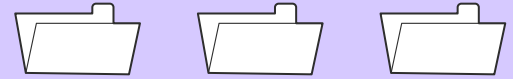
Create Admin (Superuser)

```
python manage.py createsuperuser
```

Kemudian mengisi username, email, dan password untuk login
admin localhost:8000/admin



Quiz – Model



Langkah menambahkan model ke proyek Django?

A

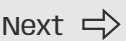
1. Susun model.py
2. python manage.py makemigrations
3. python manage.py migrate

B

1. python manage.py makemigrations
2. python manage.py migrate
3. Susun model.py

C

1. Susun model.py
2. python manage.py migrate
3. python manage.py makemigrations



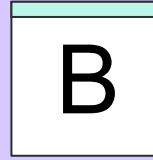
Quiz – Model



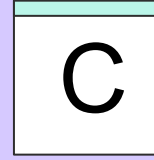
Langkah menambahkan model ke proyek Django?



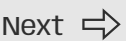
1. Susun model.py
2. python manage.py makemigrations
3. python manage.py migrate



1. python manage.py makemigrations
2. python manage.py migrate
3. Susun model.py



1. Susun model.py
2. python manage.py migrate
3. python manage.py makemigrations



Shortcut Django Command **keys**

Note	Command	Note	Command
Create venv	<code>python -m venv env</code>	Run Django	<code>python manage.py runserver</code>
Activate venv (Windows)	<code>env/Scripts/activate.bat</code>	Makemigrations	<code>python manage.py makemigrations</code>
Activate venv (MacOS)	<code>source env/bin/activate</code>	Migrate	<code>python manage.py migrate</code>
Install Django on venv	<code>python -m pip install Django</code>	Freeze Reqs	<code>pip freeze > requirements.txt</code>
Create Django Project	<code>django-admin startproject <project_name></code>	Install Reqs	<code>pip install -r requirements.txt</code>
Create Django App	<code>python manage.py startapp <app_name></code>		





Want to learn more?

<https://docs.djangoproject.com/en/4.1/>



Start



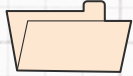
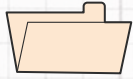
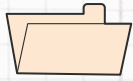
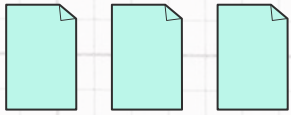
Table of contents



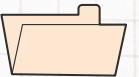
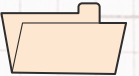
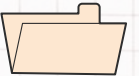
Back



Next



The main content area is a large teal rectangle. Overlaid on it is a purple square containing the number '05' in large black font. Below the purple square, the text 'Time to Practice!' is written in a large, black, sans-serif font. The entire content area is framed by a pink border with a white title bar at the top containing three window control icons (minimize, maximize, close).



Start



Table of contents

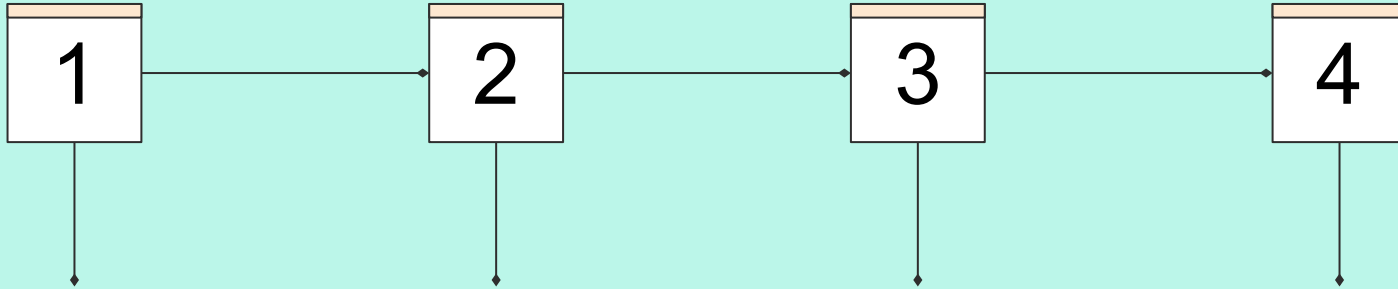


Back

Next



Practice



1 Membuat venv

Buat venv bernama "env" di dalam suatu folder

2 Aktifkan venv

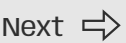
Aktifkan env menggunakan terminal sesuai OS computer, jika sudah aktif maka akan muncul tanda (env)

3 Install Django

Install Django setelah mengaktifkan env

4 Buat proyek

Buat proyek Django bernama "sbf_pti"



Practice

5

Buat app

Buat aplikasi Django bernama "absensi"

6

Jalankan Django

python manage.py runserver lalu cek browser ke localhost:8000

7

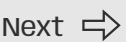
Susun model

Buat object class bernama "Peserta" yang memiliki atribut nama, angkatan, jurusan, dan divisi_sbf

8

Migrasi

Lakukan migrasi pada class yang telah dibuat



Practice



9

10

11

12

Menata urls.py

Susun urls.py yang ada di project folder dan app folder, tambahkan path menuju localhost:8000/daftar-peserta

Buat views

Buat views untuk menampilkan daftar peserta SBF PTI 2022 sesuai path di urls.py ke dalam template index.html

Buat template

Buat template index.html yang digunakan untuk menampilkan data peserta

Eksplorasi

Lakukan eksplorasi fitur-fitur framework Django

