

RANGKUMAN PLAYLIST DJANGO



**UNIVERSITAS
TEKNOLOGI
SUMBAWA**

**MATA KULIAH : PEMROGRAMAN MOBILE
DOSEN PENGAMPUH : HERFANDI, A.Md.,S.Kom.,M.Kom**

DISUSUN OLEH :

Bayu Septian

NIM : 20.01.013.040

**PROGRAM STUDI TEKNIK
INFORMATIKA FAKULTAS
REKAYASA SISTEM UNIVERSITAS
TEKNOLOGI SUMBAWA 2022**

RANGKUMAN PLAYLIST DJANGO

Django merupakan salah satu web framework yang menggunakan bahasa pemrograman python, django berbasis MVT adalah kependekan dari Model, View, dan Template. Pengertian Web framework adalah sebuah tools yang berguna untuk memudahkan programmer ketika membuat aplikasi berbasis web.

Apa itu MVT ?

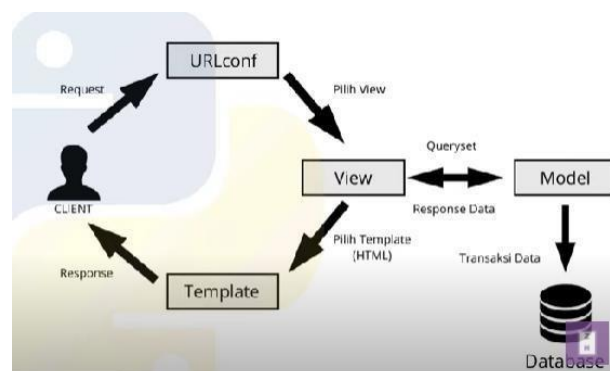
Jika biasanya sebuah framework itu berbasis MVC (Model View Controller), pada django ini aga sedikit berbeda yaitu menggunakan model MTV dimana pengertian dari MTV adalah sebagai berikut :

1. Model, merupakan bagian yang berfungsi untuk melakukan iteraksi dengan basis data.
2. View, merupakan bagian yang memuat logika biasanya digunakan untuk mengolah data dari model kemudian dapat dikirimkan ke dalam Template.
3. Template, merupakan bagian yang berfungsi untuk mengatur tampilan dalam bentuk XML atau HTML.

Ada beberapa skill dasar yang diperlukan untuk belajar Django seperti :

1. Memahami Python
2. Memahami Konsep OOP / PBO
3. Memahami Query SQL
4. Mengerti HTML

WORKFLOW Django



1. Client melakukan Request. Misalnya meminta URL Zulhilmi.id/profile
2. Lalu dicek oleh URLconf, apakah Request Profile ada di aplikasi Django?
3. Jika ada, Maka akan langsung memilih View yang sesuai
4. Dari View, jika Request membutuhkan data dari database maka View akan melakukan Queryset ke Model dan Model mengambil data dari Database yang selanjutnya direspons Kembali ke View
5. Selanjutnya View memberikan data tersebut ke Template untuk Selanjutnya di Respons kepada Client

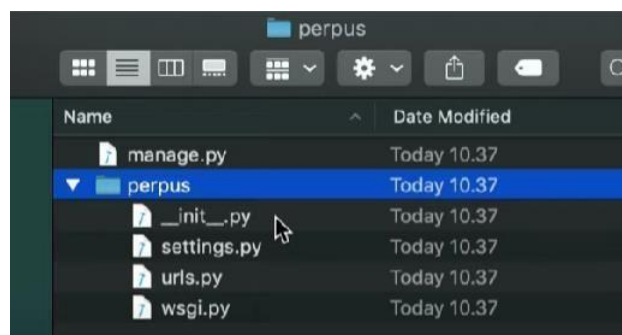
Catatan : Queryset bersifat Opsional, tergantung Request dari Client. Jika Request tidak membutuhkan database, maka dari View bisa langsung dijalankan ke Template.

PERALATAN YANG DIBUTUHKAN Belajar Django

1. CMD / Terminal Berfungsi untuk menginstall Django, membuat project, menjalankan server, dll
2. Text Editor (VSCode) Untuk menulis kode program
3. Web Browser Untuk menampilkan Aplikasi web yang telah dibuat
4. Install Python
5. PIP adalah sebuah tool yang akan kita gunakan untuk manajemen paket python. Termasuk juga menginstall Django.

MEMBUAT PROJECT APLIKASI PERPUSTAKAAN

1. cd Desktop / #Untuk menyimpan Project di Halaman Desktop
2. Django-admin startproject perpustakaan #Perintah ini untuk membuat Project baru Bernama Perpustakaan
3. Didalam Folder Project Perpustakaan terdapat :



- Manage.py #Ini adalah file Perintah untuk berinteraksi dengan Project Django
 - __init__.py #Untuk memberitahukan Python bahwa Perpus adalah sebuah Package
 - Setting.py #Berisikan File Konfigurasi / Pengaturan seperti Konfigurasi Template, Database, dll
 - Urls.py #Berisikan pola – pola URL
 - Wsgi.py #Untuk development Project yang melibatkan Web server yang kompetibel dengan Wsgi / Mengonline-kan Project
4. Python manage.py runserver #Untuk menjalankan Project dan mendapatkan alamat Project Django yang akan dibuat

BASIC ROUTING

1. Client memberikan Request untuk mengakses Halaman Buku pada Server Django namun PAGE NOT FOUND. Karena URL Buku yang di Request belum ada di Server.



2. Untuk dapat melayani Request dari Client, harus dibuatkan URL baru seperti pada gambar berikut.

```
from django.http import HttpResponse

def buku(request):
    return HttpResponse('Halaman Buku')

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

3. Jika sudah, maka Request dari Client akan dapat terpenuhi



MEMBUAT APPS

Apps adalah sebuah aplikasi pada Django yang mempunyai Model Database, View, Template, dan URLCons. Setiap Project di Django mempunyai Apps dan bisa lebih dari satu Apps

```
cd Desktop/perpus/  
django-admin startapp perpustakaan  
python manage.py startapp perpustakaan  
code .
```

MEMBUAT VIEWS

Alur yang digunakan untuk membuat Views ini adalah Client, URLConf, View dan memberikan Response kepada Client. Views diciptakan untuk memenuhi Request dari Client.

```
urls.py  views.py  X  
perpustakaan > views.py > penerbit  
1  from django.shortcuts import render  
2  from django.http import HttpResponse  
3  
4  def buku(request):  
5      return HttpResponse('Halaman Buku')  
6  
7  def penerbit(request):  
8      return HttpResponse('<h1>Halaman Penerbit</h1>')  
9
```

TEMPLATE

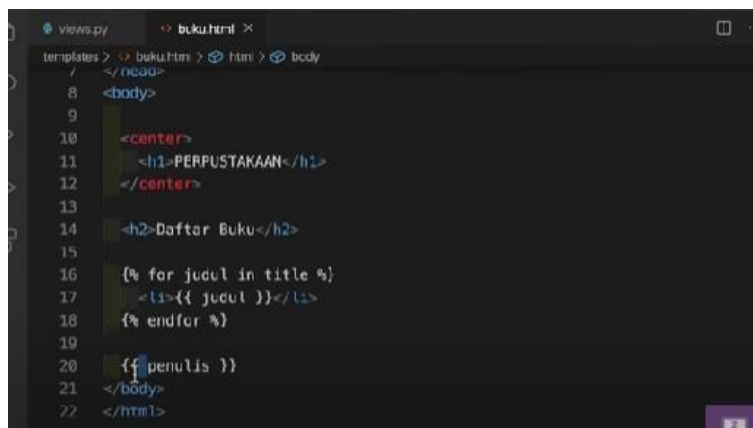
1. Pada tahap ini, Alur yang digunakan mulai berkembang. Dimulai dari Client, URLConf, View, dan Template.
2. Pada bagian Settings.py Line 58. Tambahkan 'DIRS' : ['template']
3. Buat Folder templates satu Level dengan settings, lalu buat buku.html
4. Membuat HTML sederhana
5. Lalu ubah Views.py menjadi seperti berikut

```
views.py  settings.py  buku.html  
perpustakaan > views.py > buku  
1  from django.shortcuts import render  
2  
3  def buku(request):  
4      return render(request, 'buku.html')  
5  
6  def penerbit(request):  
7      return HttpResponse('<h1>Halaman Penerbit</h1>')  
8
```

DJANGO TEMPLATE LANGUAGE

Terbagi menjadi 3 :

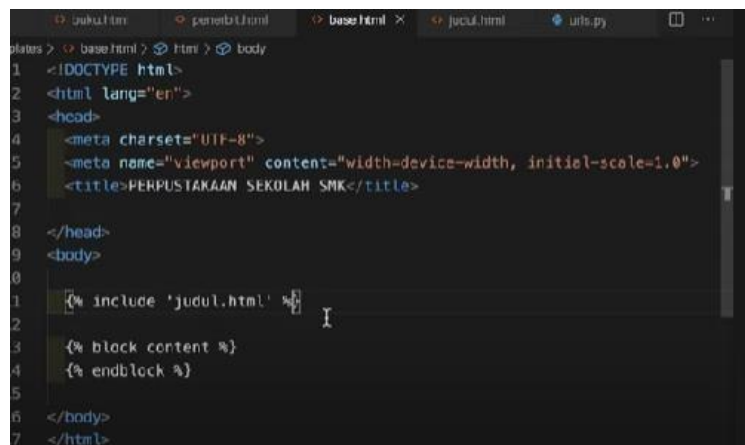
1. Substitusi Variabel : Untuk menampilkan Variabel ke Template yang dilemparkan dari Views
2. Filter : Untuk memodifikasi Variabel yang akan ditampilkan
3. Tags : Untuk melakukan control Flow seperti Looping, If Else, Include, dll untuk mengambil file eksternal



```
templates > > buku.html > html > body
1  </head>
2
3  <body>
4
5  <center>
6    <h1>PERPUSTAKAAN</h1>
7  </center>
8
9  <h2>Daftar Buku</h2>
10
11  {% for judul in title %}
12    <li>{{ judul }}</li>
13  {% endfor %}
14
15  {{ penulis }}
16 </body>
17 </html>
```

TEMPLATE EXTENDING

1. Pada tahap ini, kita akan membuat Base Template / Template dasar untuk semua Halaman.
2. Template utama yang isinya Base.html adalah file html utama yang akan menampung konten – konten dari Template Apps
3. Didalam Apps akan dibuat folder template yang isinya hanya bagian – bagian konten
4. Bagian Konten akan diextend / dimasukkan kedalam Base.html



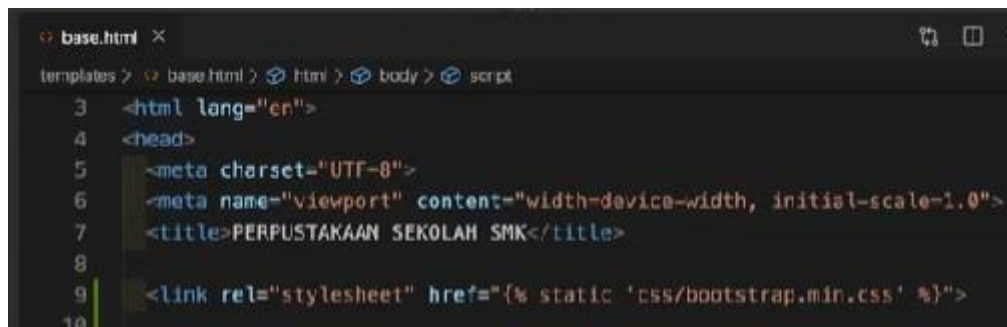
```
templates > > base.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>PERPUSTAKAAN SEKOLAH SMK</title>
7  </head>
8  <body>
9
10  {% include 'judul.html' %}
11
12  {% block content %}
13  {% endblock %}
14 </body>
15 </html>
```

STATIC FILE

Static File adalah kumpulan File CSS, Java Script, dan gambar. Static File ini digunakan untuk mempercantik / memperindah tampilan Aplikasi yang dibuat dan memberikan pengalaman kenyamanan saat Aplikasi digunakan.

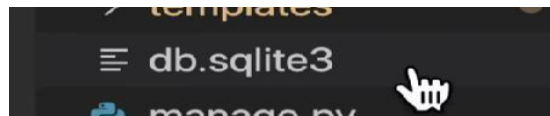
SET UP BOOTSTRAP

1. Tahap ini cukup mudah dilakukan karena kita hanya perlu memindahkan file – file CSS dan Java Script kedalam Folder static yang telah di Set Up.
2. Download Bootstrap dan JQuery yang akan digunakan.
3. Selanjutnya panggil file – file tersebut di Base.html

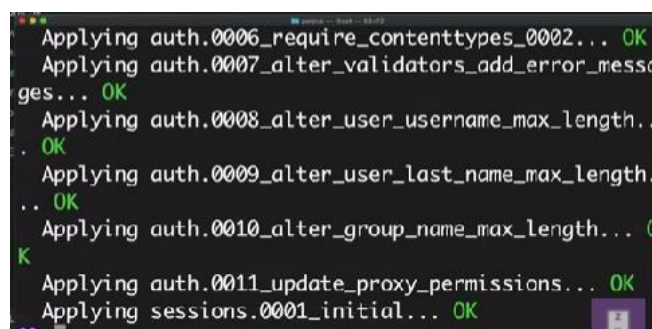


```
base.html x
templates > base.html > html > body > script
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>PERPUSTAKAAN SEKOLAH SMK</title>
8
9   <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
10
```

SET UP DATABASE PADA DJANGO



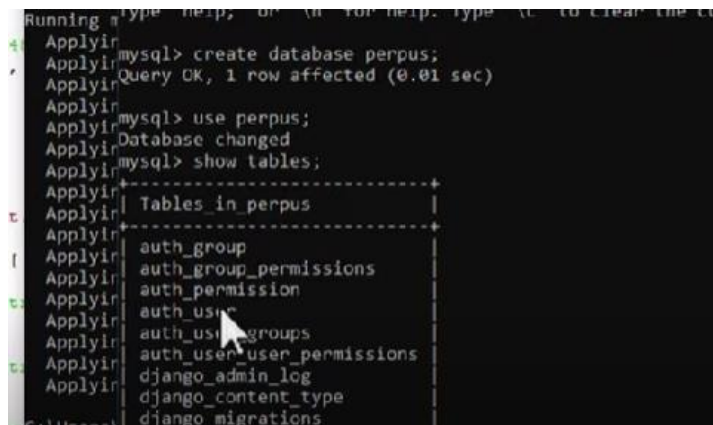
1. Secara default Django menggunakan DBMS Sqlite 3 dengan nama 'db.sqlite3'. Ini bisa di Rename sesuai dengan keinginan kita seperti 'perpustakaan.sqlite3'
2. Pada saat pertama kali melakukan Runserver, ini akan mengcreate database saja tidak termasuk tabel – tabel nya database
3. Selanjutnya dilakukan Migrasi dilakukan untuk menyebarkan / menginisialisasi tabel – tabel kedalam db.sqlite3 terhadap database project yang akan dibuat.
4. Jika berhasil melakukan Migrasi, maka akan tampil seperti dibawah ini



```
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_mes... 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 sessions.0001_initial... OK
```

SET UP DATABASE : MYSQL

1. Pada tahap ini, melakukan Konfigurasi MySql sebagai DBMS untuk project Django yang akan dibuat.
2. Tahap ini bersifat Opsional, bisa tidak digunakan jika ingin menggunakan sqlite3 sebagai DBMS
3. Jika ingin menggunakan MySql sebagai DBMS, selanjutnya install MySql Installer
4. Jika berhasil diinstall, selanjutnya buka MySql Command Line lalu create Database
5. Konfigurasikan Database ke settings.py dan akan tampil seperti gambar berikut



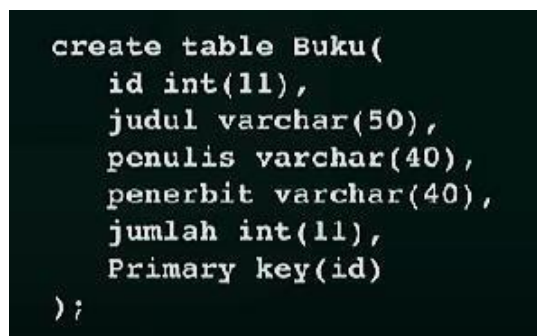
```
Running 'type help, or (h) for help. type (c) to clear the cu
Applyin mysql> create database perpus;
Applyin Query OK, 1 row affected (0.01 sec)
Applyin mysql> use perpus;
Applyin Database changed
Applyin mysql> show tables;
Applyin +-----+
Applyin | Tables in perpus |
Applyin +-----+
Applyin | auth_group
Applyin | auth_group_permissions
Applyin | auth_permission
Applyin | auth_user
Applyin | auth_user_groups
Applyin | auth_user_user_permissions
Applyin | django_admin_log
Applyin | django_content_type
Applyin | django_migrations
```

MODELS

Models merupakan definitive dari database atau representasi tabel pada database. Dengan menggunakan models ini, kita tidak perlu lagi menggunakan Query SQL untuk membuat tabel di database.

Adapun perbedaan SQL dengan Models adalah sebagai berikut :

SQL



```
create table Buku(
  id int(11),
  judul varchar(50),
  penulis varchar(40),
  penerbit varchar(40),
  jumlah int(11),
  Primary key(id)
);
```


Model

```
class Buku(models.Model):  
    judul = models.CharField(max_length=50)  
    penulis = models.CharField(max_length=40)  
    penerbit = models.CharField(max_length=40)  
    jumlah = models.IntegerField()
```

Ketika melakukan Migrasi pada model buku, maka Django akan melakukan Create Tabel Buku sesuai dengan field – field yang ada pada model buku ini. Maka jadilah Tabel pada Database yaitu Tabel Buku.

MODELS : FOREIGN KEY

1. Foreign Key digunakan untuk membuat Relasi antar tabel dalam database Relational.
2. Kelompok_id pada gambar dibawah merupakan Foreign Key yang nanti nya akan diisi oleh id tabel kelompok



DJANGO ADMIN

Django Admin merupakan salah satu fitur yang powerfull yang ada pada Django. Dikatakan Powerfull karena dapat melakukan CRUD sederhana untuk mengelola data pada model yang kita buat.

Model – Model yang kita buat akan ditambahkan kedalam Django Admin. Django admin ini bersifat Private karena diperlukan Login terlebih dahulu untuk dapat mengakses nya. Django admin Sederhana namun sangat membantu.

Pertanyaannya ada dua :

1. Bagaimana cara kita log in django admin ini?

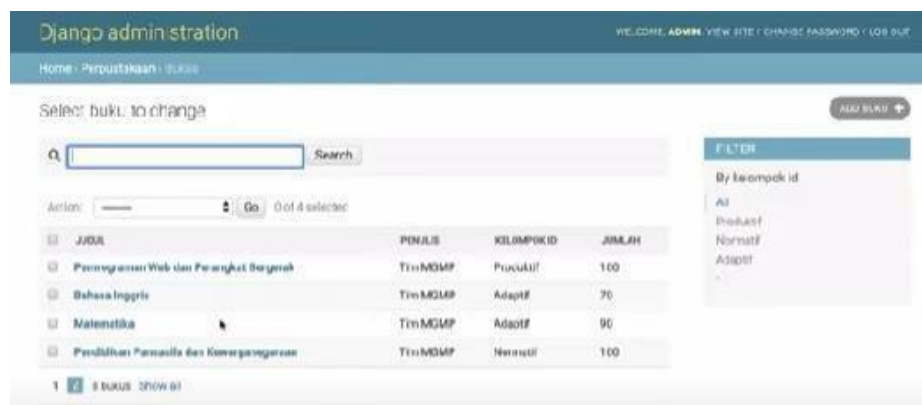
2. Bagaimana model-model kita di daftarkan ke dalam django admin ini?

Sebenarnya saat kita membuat projek ada URL yang sudah dibuatkan oleh djangonya yaitu admin, URL inilah untuk mengakses django admin, apabila ingin akses langsung /admin tampilannya langsung django admin log in. Lalu buat akun buka terminal buat username nya dengan nama admin, lalu email addres, dan password 2 kali. Jalankan kembali servernya jika sudah membuat akun. Akun tersebut di masukkan ke dalam log in.

Apabila sudah masuk ke dalam akun selanjutnya kita menampilkan model yang sudah kita buat di dalam django admin. Lalu save, model-model yang kita sudah buat sudah masuk sudah terdaftar. Perpustakaan adalah nama X nya, Buku dan kelompok adalah model nya. Power fullnya terdapat di tombol add kalau di klik akan menampilkan ada sebuah form untuk menambahkan data dalam Buku, kita hanya membuat kelas model saja di datakan di dalam django admin. Mengisi judul,penulis,penerbit,jumlah,dan kelompok id dengan mengisi nama dan keterangan dan save. Ini udah termasuk ke database. Kita buka lagi di model kelompok yang sudah terdapat 3 kelompok seperti adaptif,normatif,dan produktif.

DJANGO ADMIN : MODEL ADMIN

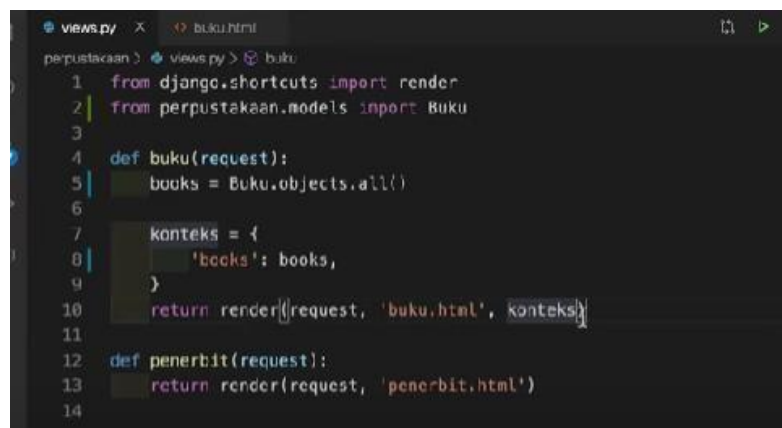
Pada tahap ini, kita akan melakukan custom sederhana terhadap tampilan data buku, fill-fill apa saja yang ditampilkan sebagai informasi seperti judul,penulis,penerbit dan lain-lain. Kita akan menampilkan kotak pencarian dan filter kelompok buku.



ORM (Object Relational Mapping)

ORM (object relational mapping) merupakan teknik yang digunakan dalam pemrograman untuk menggunakan basis data relasional sebagai penyimpanan data dengan bentuk objek. Perlu diketahui django menggunakan teknik ini untuk menggunakan database relasional.

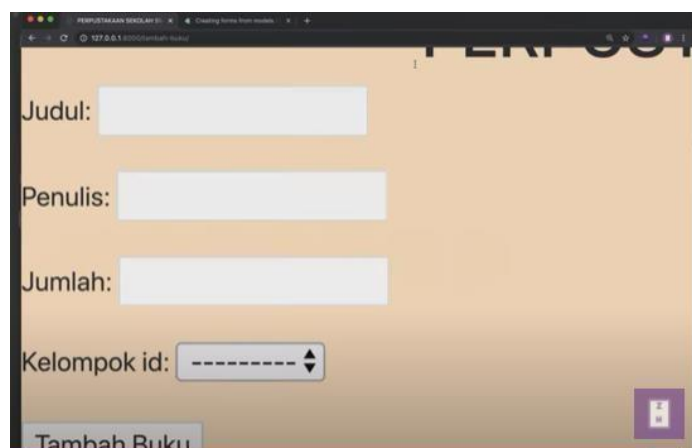
Kenapa? Agar kode python yang ditulis tidak campur aduk dengan query sql. Jadi, ORM ini bertugas sebagai penghubung aplikasi yang dibuat menggunakan database relasional.

A screenshot of a code editor showing a Django view function. The file is named 'views.py' and the function is named 'buku'. It imports 'render' from 'django.shortcuts' and 'Buku' from 'perpustakaan.models'. The function 'buku(request)' retrieves all books from the database using 'Buku.objects.all()', creates a context dictionary with the key 'books' and the value 'books', and then renders the 'buku.html' template with this context. Another function 'penerbit(request)' is also shown, which renders the 'penerbit.html' template.

```
views.py x buku.html
perpustakaan > views.py > buku
1 from django.shortcuts import render
2 from perpustakaan.models import Buku
3
4 def buku(request):
5     books = Buku.objects.all()
6
7     konteks = {
8         'books': books,
9     }
10    return render(request, 'buku.html', konteks)
11
12 def penerbit(request):
13     return render(request, 'penerbit.html')
14
```

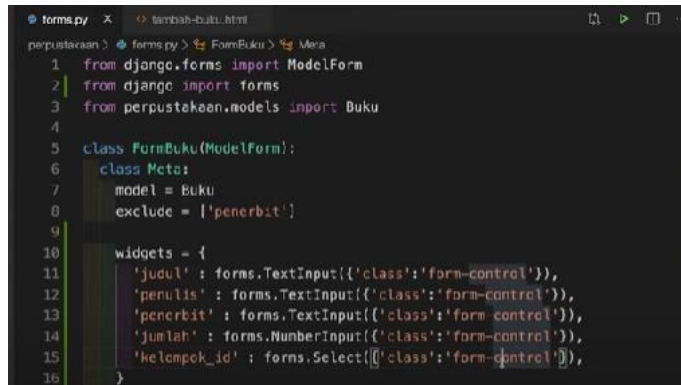
FORMS : MODEL FORM

1. Form biasanya ditulis dengan HTML. Namun di Django tidak wajib menggunakan HTML, bisa saja menggunakan Models
2. Input Type harus disesuaikan dengan Type data pada tabel database.
3. Selanjutnya Membuat file baru di perpustakaan dengan nama form. Lalu membuat viewsnya untuk menghasilkan seperti gambar dibawah ini

A screenshot of a web browser showing a form titled 'PERPUSTAKAAN SIKOLAN'. The form has four input fields: 'Judul:', 'Penulis:', 'Jumlah:', and 'Kelompok id:'. The 'Kelompok id:' field is a dropdown menu. At the bottom of the form is a button labeled 'Tambah Buku'.

FORMS : WIDGED

1. Pada tahap ini, kita menambahkan atribut dengan menggunakan widged pada output yang telah dibuat sebelumnya.
2. Didalam forms.py ditambahkan widgets, text Input merupakan typenya.
3. Tambahkan atribut kelas didalamnya.



```
1 from django.forms import ModelForm
2 from django import forms
3 from perpustakaan.models import Buku
4
5 class FormBuku(ModelForm):
6     class Meta:
7         model = Buku
8         exclude = ['penerbit']
9
10    widgets = {
11        'judul': forms.TextInput({'class': 'form-control'}),
12        'penerbit': forms.TextInput({'class': 'form-control'}),
13        'jumlah': forms.NumberInput({'class': 'form-control'}),
14        'kelompok_id': forms.Select({'class': 'form-control'}),
15    }
```

CRUD : Menambah Data

1. Pada tahap ini, kita masuk ke dalam queryset. Disini kita menambah data, menampilkan, mengubah dan menghapus data Dari data yang sudah dibuat.
2. Selanjutnya kita akan menyimpannya ke dalam database. Prosesnya ada di views yang nantinya akan mengecek apakah data yang di submit oleh user sudah benar atau sudah valid.
3. Apabila sudah maka akan di simpan ke database.
4. Lalu buka text editor, buka file views. Fungsi dari csrf sendiri adalah untuk mngamankan form yang dibuat.

CRUD : Menampilkan Data

1. Menampilkan data buku yang sudah dibuat, seperti menampilkan nilai 90/100 dan menampilkan kelompok seperti produktif maka hasilnya pasti error karena kelompok_id merupakan type integer.
2. Namun Apabila menambah nama, maka akan bisa mengeluarkan hasilnya.
3. Penggunaan queryset tidak sepanjang query. Untuk menampilkan limit pada queryset cukup ditambah diujungnya [:3], sedangkan dalam query limit 3.

CRUD : Mengubah Data

1. Tahap Mengubah data ini perannya sangat penting karena saat kita salah dalam melakukan input data, maka kita harus mengubahnya atau mengeditnya dengan menggunakan fitur update atau fitur ubah data ini. Ini lebih efektif dibandingkan dengan hapus, lalu input ulang
2. Ubah data sebenarnya sama dengan form menambah data. Bedanya form ubah data sudah terisi oleh data yang ingin kita ubah, misalnya seperti ingin mengubah judul yang dari Bhs.Indonesia menjadi IPA, lalu klik simpan dan selesai.

CRUD : Hapus Data

1. Hapus data juga penting dalam aplikasi, karena apabila data tersebut tidak lagi digunakan kita dapat menghapus nya. Karena Fungsi fitur ini sendiri adalah untuk menghapus data yang sudah tidak digunakan oleh user.
2. Sebelum menggunakan action/aksi hapus, terlebih dahulu membuat konfirmasi yang berisi apakah yakin ingin menghapus atau tidak untuk menghindari ketidaksengajaan user dalam menghapus data.

AUTHENTICATION : LOGIN & LOGOUT

Authentication merupakan proses verifikasi/validasi identitas user yang terdaftar sebelum mengakses system. Dengan ini kita jadi bisa membatasi user mana saja yang boleh menambahkan data, mengubah data, dan menghapus data. Jadi tidak sembarang user mengakses halaman-halaman tersebut . django ada system authentication yang akan kita gunakan yang bernama class LoginView untuk membuat namanya sama seperti membuat log in django admin. Berarti saat ingin membuat yang baru bisa membuat di terminal.

MENGAKSES USERNAME DI TEMPLATE

Pada tahap ini, kita mencoba untuk mengakses user yang sedang aktif untuk ditampilkan ke dalam template. Biasanya saat kita login ke dalam aplikasi, terdapat informasi user atau siapa saja yang saat ini sedang login atau yang sedang menggunakan aplikasi. Dan Apabila kita log out dan tidak terdapat informasi maka kita bisa menggunakan attributes `is_authenticated`.



	PENULIS	PENERBIT	JUMLAH
	Tim MGMP	Kemendikbud	100

SIGN UP

Pada tahap ini kita akan membuat form sign up, agar user bisa login ke aplikasi perpustakaan yang dibuat. Tahap ini berhubungan dengan django admin karena di dalamnya terdapat username django adminnya. Jika lihat di dokumentasi resmi Django, ada sebuah user form yang bernama UserCreationForm untuk membuat user baru.



UPLOAD FILE

Pada tahap ini kita membuat tools upload file, seperti mengupload cover buku. Kita akan menambahkan file baru pada model buku yang bernama cover sehingga di dalam tambah buku nanti akan muncul yang baru seperti cover atau muncul di ubah buku. Di buku kita akan menambahkan kolom baru yaitu cover sebagai informasi buku yang kita masukkan.

EXPORT FILE

Pada tahap ini akan membuat export file. Export file ini biasa digunakan untuk membuat report atau laporan ke dalam bentuk file excel, pdf, atau file lainnya. Dalam aplikasi

biasanya terdapat fitur laporan yang harus di export data yang terdapat dalam database ke dalam file excel contohnya. Cara menggunakan export ini bisa menggunakan `django-import-export`.

```
# app/admin.py

from import_export import resources
from core.models import Book

class BookResource(resources.ModelResource):

    class Meta:
        model = Book
```

VIRTUAL ENVIRONMENT

Pada tahap ini akan membahas tentang virtual environment atau lingkungan virtual, dalam kasus ini lingkungan virtual berguna untuk mengisolasi proyek yang kita buat. Setiap kita membuat proyek didalam lingkungan virtual ini, proyek tersebut akan terisolasi dari direktori system dan memiliki paket python sendiri yang terinstal di lingkungan virtual tersebut. Lingkungan system operasi lebih besar yang didalam nya sudah ada django versi 2.2.12, pillow versi 7.1.2, dan `django-import-export` versi 2.2.0. kita juga sudah membuat aplikasi perpustakaan di lingkungan system tersebut dengan menggunakan django versi 2.2.12. apabila kita membuat proyek di dalam lingkungan virtual environment 1 dan virtual environment 2. Di dalam virtual env 1 terdapat aplikasi perpustakaan yang dibangun dengan django 1.11 sedangkan di dalam virtual env 2 terdapat juga aplikasi yang dibangun dengan django 3.0.8 mysqlclient.

PERSIAPAN DEPLOYMENT

Deployment adalah kegiatan untuk menyebarkan aplikasi yang telah dikerjakan oleh developer (pengembang). Artinya kita mempublish projek atau aplikasi yang kita buat untuk dilihat oleh orang lain. Kita juga melewati siklus pembuatan aplikasi yaitu :

- a. Development : mengerjakan proyek aplikasi kita atau proses pembuatan, pengembangan . kita di awal-awal akan membuat form,model,view,tampilan itu adalah tahapan development atau pengembangan.

- b. Testing : setelah dilakukan pembuatan otomatis kita akan melakukan uji coba agar mengetahui kekurangan dari kegiatan tersebut. Mengecek apakah sudah sesuai atau apakah ada bug. Testing ini juga dilakukan saat production.
- c. Production : aplikasi kita yang sudah di buat sudah di deploy / sudah bisa di publish / sudah bisa digunakan oleh orang lain. Bisa juga user melakukan testing pada saat tahap ini.

Alur kerja nya dari client yang melakukan request langsung ditangani oleh web server lalu ke wsgi lalu terakhir ke django atau perputaran yang sudah kita buat yaitu aplikasi perpustakaan tersebut.

Yang perlu disiapkan adalah app (github,writelab,perpus), terminal, server, domain.

DEPLOYMENT

Pada tahap ini, kita akan mendeploymen aplikasi yang telah dibuat ke server. Dengan awalan mengcoding di local lalu mengupload ke github lalu di clon atau ditarik ke computer server untuk dideploy disana.

Saat aplikasi kita dalam mode production disetting dan debug masih nilainya masih dalam true jadi kalau misalkan salah menuju ke halaman pasti akan eror .

Saat kita dalam mode production kita tidak boleh dibiasakan mengubah tampilan saat mode dalam production, apabila terjadi eror akan berakibat fatal maka itu kalau kita ingin mengembangkan aplikasi yang dalam production kita hanya perlu mengembangkannya di local dan kita masuk ke tahap development atau tahap pengembangan lagi seperti mengubah tampilan atau menambah fitur yang diinginkan.

