Nama:Linda Mutiara

Nim:19.01.013.005

Tugas: Rangkuman Belajar Django Python

**VIDEO 1 (INTRODUCTION)**

Django merupakan web framework gratis (free open source) yang mengikuti pola arsitektur MVT dan ditulis dengan Bahasa python.Django adalah sebuah framework full-stack untuk membuat aplikasi web dengan bahasa pemrograman Python.

* Web framework merupakan software kerangka kerja yang membantu membuat pengembangan situs secara mudah dan cepat.
* Web framework atau Web application framework adalah sebuah perangkat lunak kerangka kerja yang membantu membuat dalam pengembangan situs web secara mudah dan cepat.
* MVT (Model View Template)
* Model merupakan representasi dari database.
* Views berisi fungsi-fungsi mengatur request dan response.
* Template berhubungan dengan user interface atau berkas-berkas HTML.

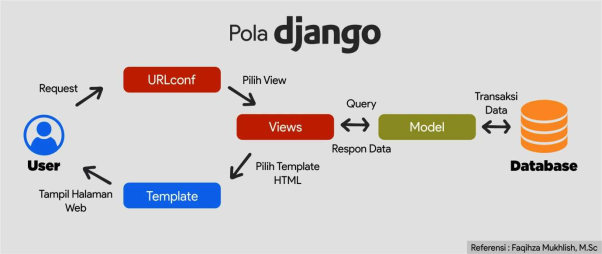
Wesite yang menggunakan teknologi django:

* Codesaya.com
* Kelaskita.com
* Gojek Indonesia

Skill dasar yang dibutuhkan untuk belajar django:

* Python
* OOP/PBO
* Paham query SQL
* HTML

**VIDEO 2 (WORKFLOW)**



Pertama, kita mulai dari user (client) melakukan request urlconf. Jika profil ada maka akan memilih view yang sesuai lalu akan dicek lagi apakah request dari client membutuhkan data dari database. Jika membutuhkan data dari database makan akan melakukan cursed ke model. Model akan mengambil data dari database kemudian di respon view datanya dan akan di lanjutkan ke template.

Namun, sifatnya opsional jadi tergantung dari request jika requestnya tidak membutuhkan database dari view bisa langsung memiloh template dan direspon ke user (client).

**VIDEO 3 (APA SAJA YANG DIBUTUHKAN?)**

Tools yang dibutuhkan untuk memulai dengan django:

1. CMD/Terminal

CMD digunakan untuk menginstall django, membuat project, menjalankan server, dan lain-lain.

1. Text Editor

Text Editor digunakan untuk menulis program. Bisa menggunakan visual studio code atau yang lain.

1. Web Browser

Karena kita membuat aplikasi web, yang hasilnya harus dilihat melalui browser.

1. Python 3.7.x

Saat installasi python jangan lupa mencentang add python to PATH.

**VIDEO 4 (MEMBUAT PROJECT BARU DJANGO)**

Divideo dijelaskan bahwa akan membuat aplikasi perpus

Ketikkan perintah Django-admin startproject perpus

Distruktur maka akan muncul folder dengan nama perpus. Berikut penjelasannya:

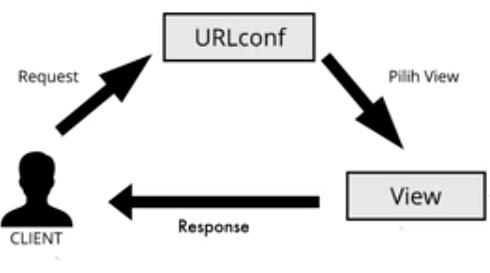
* Perpus - merupakan folder root atau utama yang menampung semua kebutuhan project web yang dibuat.
* Manage.py – file ini merupakan server yang akan mengolah project web yang dibuat. Agar server web dapat berjalan file inilah yang pertama kali dieksekusi ke CMD.
* \_\_init\_\_.py – file ini merupakan file kosong yang menandai bahwa direktori ini adalah sebuah paket (python package).
* settings.py – file ini berisi konfigurasi project seperti database,auth, hingga language code.
* url.py – tempat untuk mendeklarasikan url fungsinya sama dengan web.php pada routes laravel.
* wsgi.py –

Setelah project baru selsai dibuat maka selanjutnya menjalankan project yang dibuat. Caranya menjalankan file manage.py dengan mengetikkan perintah python manage.py runserver. Selanjutnya buka browser dan masukkan url “”.

**Video 5 Basic routing**

Mengenal besik routing Django dengan pola URL

Alur sederhana aplikasi perpus



1. Clien melakukan request kemudian di cek oleh URLconf pilih viewnya kemudian di response
2. Clien meminta reques meminta ke aplikasi perpus halaman buku URLbuku kemudian di cek, apalida sudah di cek dan sudah ada dipilih viewnya lalu direspon dimana halaman buku

**Video 6.1**

Apps merupakan aplkash pada django yang mempunyai model databases view template dan urlconf.Dimana sh pekerja di handle oleh app,dan memiliki lebih dari 1 model app,

Membuuat apps :

1. Masuk ke cd/desktop/perpus/
2. Ada dua cara membuat app
3. Django - admin dengan perintah startapp perpustakaan
4. Menjalankan file python manange.py diikuti sub perintah startapp perpustakaan
5. Setelah app dibuat sekarang kita buka code editor dengan perintah code.
6. App pada django merupaan sebuah pages
7. Jadi disana terdapat beberapa isi dari modul perpustakaan
8. Admin.py yang digunakan untuk meregistrasikan model yang kita buat untuk ditampilkan di njago admin
9. App.py file membantu user melakukan configurasi
10. Model.py untuk memdelkan table ke databases
11. Test.py digunakan untuk testing
12. Views.py berisi fungsi” untuk melakukan untuk melakukan proses dan akses dari client.yang nnti nya views ini akan disimpan di file views yang sebenarnya.

**Video 6.2 Membuat views,dimana nnatinya views akan disimpan di dalam apps**

1. Jadi disini untuk code view yang ada di dalam file urls pada folder perpus kita copy lalu dihapus dan di pindahkan ke folder perpustakaan pada file views lalu di save

From django.http import HttResponse

def buku (request) :

return httpResponse(‘Halaman Buku’)

1. Nah setelah sudah sekarang kita masuk lagi ke file urls dimana jika kita jalankan akan terjadi eror dimana disana terdapat

path (‘buku/’, buku ),

Jadi harus dilakukan pemanggilan data dari folder perpustakaan dengan code

From perpustakaan.Views import buku

Dimana disini kita mengimport method buku yang ada di file views.py yang terdapat di dalam folder perpustakaan lalu save.

1. Untuk membuktikan hasil dari views bisa dengan cara buka 127.0.0.0:8000/buku/

Dan nnti nya akan muncul hasil dari view disana seperti pada contoh

“Halaman Buku”

1. Langkah selanjudnya yaitu menambahkan URL penerbit dengan cara yang sama

def penerbit (request) :

return httpResponse(‘Halaman penerbit’)

1. Dan di import.py nya kita taro seperti code dibawah lalu save dan jalankan sehingga sudah ada 2 view yang dibuat.

From perpustakaan.Views import buku,penerbit

1. Jadi di views juga dapat kita sisipkan kode HTML yaitu

“h1 (header) “ yang digunakana untuk mengatur besar tulisan pada output

**Video 7.1 Templates**

1. Pada file setting tambahkan templates di dalam code DIRS yang bermaksud sebagai tempat file HTML kita disimpan yaitu di file directori.

'DIRS': ['templates'

1. Kemudian kita buat folder Templates dan didalamnya kita buat file HTMLnya yaitu (Buku.HTML)
2. Untuk langkah selanjudnya kita buat code HTML sederhana

<!DOCTYPE html>

<title> perpustakaan </title>

</head>

<body>

<center>

<h1> perpustakaan </h1>

</center>

<h1> Daftar Buku </h2>

</body>

</html>

1. Kemudian kita masuk ke views lalu hapus

From django.http import HttResponse

1. Lalu isi kan dengan code dibawah,save lalun liat hasilnya.

def home(request):

  template = 'buku.html'

    return render(request, template)

1. Jadi untuk alur kerja dari program ini yaitu dimana client melakukan request untuk membuka halaman buku,lalu dipilih view nya buku sehingga nanti akan uncul views dari perpustakaan dan pilihan daftar buku.

**Video 7.2**

Bahasa template pada django di bagi menjadi 3 yaitu

1. Subtitusi variable
2. Yang digunakan untuk menampilkan variable ke template yang dilemparkan dari views.

def buku (request) :

judul = “belajar django”

konteks{

‘title’ = judul

1. return render (reques ,’buku.html’, konteks)

kemudian tambahkan {{ title }} pada buku .html dan liat hasilnya.

Jadi disini juga dapat mengubah filter pada tulisa yang digunakan dengan cara

{{title | upper}} untuk tulisan besar semua dan lower untuk tulisan kecil semua.

1. Melakukan perulangan dengan menggunakan template django dimana ada 3 judul

“belajar django” “belajar python” dan “belajar boostrap”

Lalu ditampilkan didalam html (template) dengan code

{% for judul in title%}

<li>{{ judul }} </li>

{% endfor %} = digunakan untuk menampilkan kode

{{ penulis }} = digunakan untuk menampilkan isi dari variabel

1. Filter

Dugunakan untuk memodifikasi variable yang akan ditampilkan

1. Tags

Untuk melakukan control vlow seperti lupping,login,if,else,include untuk mengambil file eksternal

**Video 7.3 Template Extending**

Dimana disini kita membuat template dengan konfigurasi sendiri tanpa menggunakan apps dan disimpan di folder projek perpus,jadi disini nanti nya aka nada template utama dengan nama base html yang isiya akan menampung konten” yang dilempar dari templateyang ada di dalam apps,jadi di dalam folder app akan dibuat sebuah folder baru yang isinya hanya ada konte saja dan konten” yg ada nnantinya akan dimasukan ledalam base.html.

1. Untuk langkah pertama kita buat file base.html di dlam forlder template

<!DOCTYPE html>

<title> perpustakaan sekolah</title>

</head>

<body>

{% block content %} = jadi untuk block content ini merupakan block yang akan diisi oleh conten” dari apps

{% andblock %}

</body>

</html>

1. Untuk buku.html dan penerbit .html yang ada di template di pindahkan pada folder perpustakaan dengan membuat folder template terlebih dahulu

{% extand “base.html” %}

{% block content %}

<center>

<h1> perpustakaan </h1>

</center>

<h1> Daftar Buku </h2>

{% for judul in title%}

<li>{{ judul }} </li>

{% endfor %}

{{ penulis }}

{% andblock %}

Buat code yang sama juga pada penerbit.html tinggal diganti bagian header degan daftar penerbit.

1. Untuk bagian judul juga bisa dipisahkan untuk template nya dengan cara buat judul.html pada fordel template di alam base.html buat code

{% include ‘judul.html’ %}

**Vidio 8 Static files**

Static file adalah kumpulan file css, js, dan gambar

Bagaimana konvigurasi static filenya untuk memanggil fil css, js daan gambaar yang ditampilkan di aplikasi

Membuka file setting menambahkan

STATIC\_URL = '/static/'

STATICFILES\_DIRS = [

    os.path.join(BASE\_DIR, 'static'),

]

Static file ini untuk menyimpan file js, file css dan gambar sebuah folde yang Bernama “static”

Sedangkan STATIC\_URL akses untuk broser untuk mengakseses membuka file css js dan gambar yang ada di folder static

Kemudian membuat folder baru dengan nama static dimana isi file ini untuk menyimpan file css,imagnes dan cs

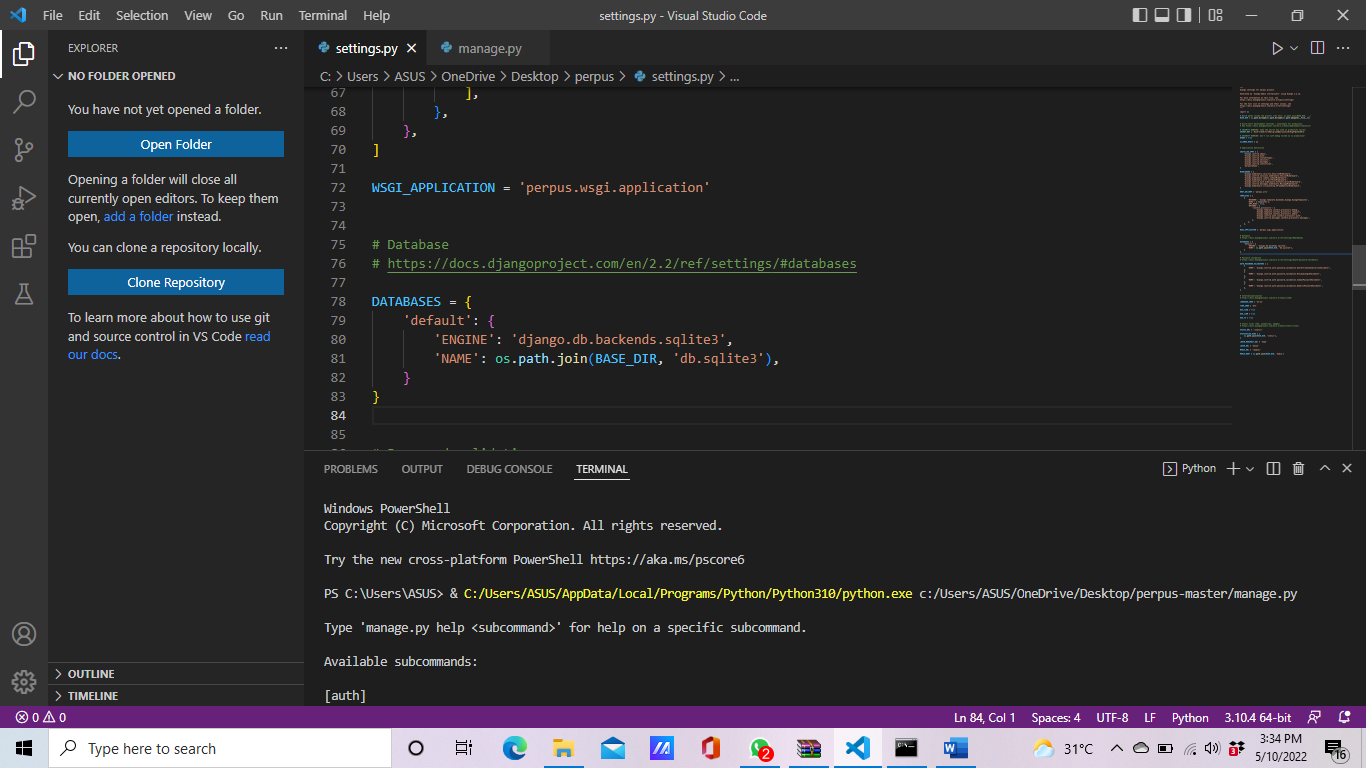
**Video 8.2 Setup bootstrap**

Hanya memindahkan file css,js ke dalamfolder static Untuk yang belum ada Bootstrap downlod terlebi dahulu di getbootstrap.com dan juga jQuery downlod di code.jquery.com

Kemudian file booststrap.min.css dimasukkan ke folder css, file booststrap.min.jc dan jquery.js dimasukkan ke folder js. Kemudian panggil file bootstrap.min.css tersebut di base.html dengan codingan “ <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}"> ”. untuk pemanggilan file file booststrap.min.jc “ <script src="{% static 'js/bootstrap.min.js' %}"></script> ” dan jquery.js “ <script src="{% static 'js/jquery.js' %}"></script>”

**VIDEO 9.1 (SETUP DATABASE)**

1. Lihat konfigurasi database default pada settings.py.



1. Lakukan migrasi untuk menerapkan model ke database.
2. Jika belum mebuat model, maka table-tabel default akan di-apply ke database.

* python manage.py makemigrations
* python manage.py migrate

**VIDEO 9.2 (SETUP DATABASE: MySQL)**

1. Konfigurasi database pada settings.py.

DATABASE = {

‘default’ : {

‘ENGINE’ : ‘django.db.backends.mysql’,

‘NAME’ : ‘perpus’

‘USER’ : ‘root’

‘PASWORD’ : ‘ranger46’ ,

‘HOST’ : ‘localhost’

‘PORT’ : ‘3306’,

}

1. Install mysqlclient.
2. Lakukan migrasi untuk menerapkan model ke database

* pip install mysqlclient
* python manage.py makemigrations
* python manage.py migrate

**Video 11 Django Admin**

jenggo admin adalah salah satu fitur yang powerful yang ada pada jenggot dikatakan powerful karena bisa melakukan grup sederhana untuk mengolah data pada model yang kita buat Hai ini adalah tampilan Jinggo buku penerbitan lain itu merupakan model-model model nanti yang kita buat akan didaftarkan ke dalam Jinggo admin disini kita bisa mengolahnya bisa menambahkan data buku data kelompok mengubah dan menghapus data Hai biasanya

Mari kita praktekkan sebenarnya saat kita membuat Project ada url yang sudah dibuatkan oleh jenggotnya yaitu admin untuk mengakses jenggo kalau kita akses langsung flash.bin cap jago admin login

Mari kita buat akunnya buka

germinal python manage.py createsuperuser username(leave blank to use ‘purpele’) admin

Email address : [admine@sekolah.sch.id](mailto:admine@sekolah.sch.id)

Password (again)

Python manager.py runserver

jika sudah membuat akun hai lalu kita coba login

selanjutnya kita menampilkan model yang kita buat ke dalam Jinggo admin buka Project dengan text editor lalu mengedit file admin.py

from django.contrib import admin

from perpustakaan.models import Buku, Kelompok

# Register your models here.

admin.site.register(Buku, BukuAdmin)

admin.site.register(Kelompok)

model-model yang kita buat sudah masuk sudah terdaftar perpustakaan adalah nama esnya buku kelompok adalah model powerful nya disini Di sini ada tombol X kalau kita klik kita lihat ada sebuah form untuk menambahkan data ke dalam databook ke dalam model buku kita hanya membuat kelas modal saja kelas buku itu didaftarkan ke dalamnya

**video 11.2 Django admin:model admin**

kita akan melakukan kustomisasi sederhana Hai terhadap tampilan data buku kira-kira seperti ini I feel feel apa saja yang akan kita tampilkan sebagai informasi seperti judul penerbit penulis dan lain-lain mengungkit akan menampilkan kotak pencarian dan juga filter-filter kelompok buku adaptif normatif atau produktif baik buka test drivernya kita masih mengedit file admin

from django.contrib import admin

from perpustakaan.models import Buku, Kelompok

# Register your models here.

class BukuAdmin(admin.ModelAdmin):

  list\_display = ['judul', 'penulis', 'kelompok\_id', 'jumlah']

  search\_fields = ['judul', 'penulis', 'penerbit']

  list\_filter = ('kelompok\_id',)

  list\_per\_page = 4

admin.site.register(Buku, BukuAdmin)

admin.site.register(Kelompok)

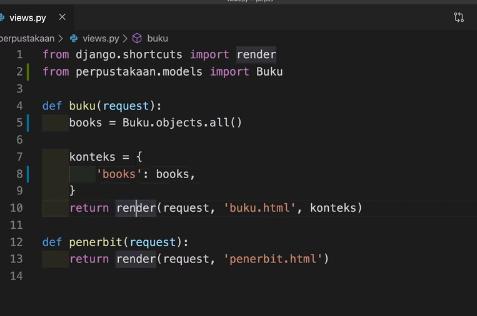
sekarang tampilannya lebih informatif judul buku penulis ini siapa penerbit dan jumlahnya ada berapa ini sudah terbagi menjadi dua halaman lakukan pencarian web berdasarkan judul pemograman web jenggo Coba kita akan lakukan filter buku produktif saja buku normatif saja adaptif semua kelompok ID Apa yang akan terjadi berhubungan dengan metode STR. metode STR itu akan mengembalikan Pada representasi model adalah nama kelompok maka ini akan tampil saat kita memanggil film yang maka dia akan mengembalikan Nama judulnya apa Nama kelompoknya .

**Video 12 ORM(objek relation mapingg)**

Tehnik yang digunakan dalam pemogromanuntuk menggunakan bassdata relational sebagai penyimpanan data dengan bentuk objek. Dhango menggunakan ORM agar kode python ditulis tidak campur aduk dengan query sql. ORM bertugas sebagai penghubung aplikasi yang dibuat menggunakan database relasional

seperti berikut ini ada kode Paketan jadinya ini akan mengambil data dari tabel buku dan ditampilkan kita soroti bagian kuenya select Bintang from buku ini sama dengan buku titik objek titik Hal inilah yang disebut dengan kue riset itu kalua

kita kembali lagi ke sini ini alur kita sudah berkembang di WeChat buku titik objek itu ekornya nih Yang nanti akan disimpan reviews lalu akan meminta mengambil kebo Delhi model Saya butuh data dari model buku dari tabel buku nanti question itu akan Meneruskan ke query SQL select Bintang from buku ngambil dari database kemudian direspon datanya dan ditampilkan ke ke layar kita jadi bertugas sebagai penghubung aplikasi yang dibuat menggunakan database relasional ini hanya penghubung baik kita coba contohnya ya untuk menampilkan data buku Ayo kita ke views



akan menampilkan judul buku dari model buku-buku adalah sebuah objek itu adalah sebuah atributnya itu filmnya apa nih yang akan ditampilkan judul dengan begitu data-data yang ada dalam model sudah bisa Tampilkan ke dalam template sudah bisa dirender

**video 13.1 from:modelfrom**

pembuatan aplikasi tidak terlepas dari yang namanya file baik itu form untuk menambah data ataupun mengubah data nah form inilah yang biasa ditulis dengan html input type text input type number input tipe Day Dan lain-lain inputtype inilah yang harus disesuaikan dengan tipe data di tabel database membuat form dengan menggunakan model volt djengo kita tidak perlu repot-repot untuk membuat form dengan html karena kita bisa membuatnya dari model yang kita buat ini opsional membuat form saja tidak sampai ke menyimpan data ke database

from django.forms import ModelForm

from perpustakaan.models import Buku

class FormBuku(ModelForm):

  class Meta:

    model = Buku

    fields = '\_\_all\_\_'

fields ini kita akan menggunakan semua file yang ada didalam model buku yang akan ditampilkan ke form input. kita buat views.py jangan lupa impor kan from perpustakaan forms import from

tipe input ini Hai yang sesuai dengan tipe data pada tabel di database teks number integer dan lain-lain Nah di sini juga kita bisa kita buka Firefox disini jadi kita bisa memfilter memperbolehkan view mana saja yang hanya ditampilkan yang boleh ditampilkan

**vido 13.2 froms:widgets**

menambahkan atribut menggunakan Gadget pada input yang sudah kita buat di episode sebelumnya Sekarang kita akan mencoba menambahkan atribut kelas dan memanggil class but saat Hai dan lihat apa yang terjadi terhadap form input ini baik kita kembali ke teks editor masih di file forms.py disini pertama-tama kita impor kanguru foxnya from jenggot impor pom-pom di bagian Meta kita tambahkan widget judul film apa yang akan kita ubah yang akan kita sisipkan atribut ke dalam input teks nya kedalam form input di sini kita coba judul dulu form Static text input atau siput ini adalah tipe yaitu input type text Karena judul adalah percaya sama dengan inputtype tekstur maka disini inputtype teks kalau misalkan tipenya text area nanti kita tulis text area ataupun sale kita tulis baru-baru di sini kita tulis atribut dan value-nya akibatnya adalah kelas Hai input teks ya tambahkan atribut kelas value-nya VOC kontrol kita Panggil kelas kontrol yang ada pada bootstrap nanti kita lihat apa yang terjadi

cara menggunakan app busrat ya Bagaimana cara menyusun struktur plus bootloop seperti ini kontainer bawa kemudian kolom playlistnya ada di channel ini ya nanti silahkan lihat ada di deskripsi lalu formnya yang sudah tadi yang tadi kita kap kita masukkan kesini ke dalam kolam D4 lalu kita tambahkan

{% extends 'base.html' %}

{% block content %}

<div class="container">

  <div class="row">

    <div class="col-md-4">

      <h3>Tambah Buku</h3>

      {% if pesan %}

        <div class="alert alert-success">{{ pesan }}</div>

      {% endif %}

      <form action="{% url 'tambah\_buku' %}" method="post" enctype="multipart/form-data">

        {% csrf\_token %}

        {{ form.as\_p }}

        <button class="btn btn-primary">Tambah Buku</button>

        <a href="{% url 'buku' %}" class="btn btn-danger">Kembali</a>

      </form>

    </div>

  </div>

</div>

**Video 14.1 Crud:menambah data**

mengisi form input ini lalu menyimpannya ke database itu prosesnya ada di views reviews nanti akan mengecek apakah data yang diisi oleh user sudah benar atau sudah valid kalau sudah makan Simpati database

**14.2 Crud:menampilkan data**

mengubah tampilan daftar bukunya ke dalam bentuk tabel ya agar sedikit lebih rapi kita hanya menampilkan judul saja jadi baik kita ke file buku.html kita ubah dulu tampilannya

di bagian body kita lakukan looking untuk mengambil data dari model buku lalu kita tampilkan datanya disini baris ke-1 kolom ke-1 buku titik judul terus penulis,penerbit, jumlah dan kelompok Underscore ID

tambahkan table border

<div class="alert alert-success">{{ message }}</div>

       <div class=”row”>

<div class”col-md-12>

      <table class="table table-bordered table-striped">

        <thead>

          <tr>

            <th>COVER</th>

            <th>JUDUL</th>

            <th>PENULIS</th>

            <th>PENERBIT</th>

            <th>JUMLAH</th>

            <th>KELOMPOK</th>

          </tr>

        </thead>

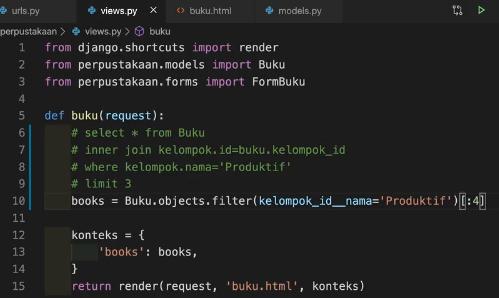
        <tbody>

          {% for buku in books %}

          <tr>

            <td>

tabel kelompok yaitu id buku kelompok ID itu ya seperti itu yang hai eh ini bukan ya saya lupa lagi where i Hai kelompok Hai nama = produktif ya pokoknya gitulah panjang ya kalau kita menggunakan kuota SQL kita tidak perlu lagi kita hanya menggunakan query set DJ Kau seperti ini



terakhir limit data limit data itu Hai seperti ini ya kalau bsql limit misalkan hanya 3data saja yang ditampilkan di dalam periset kita tambahkan di bagian akhir itu bracket titik dua tiga ini adalah limit kalau kita refresh Hai hanya menampilkan data sebanyak tiga dua sebanyak dua setempat produktif 4

**14.3 Crud:mengubah data**

mengubah data mengubah data di sini perannya sangat penting dalam aplikasi karena saat kita salah melakukan input data maka kita harus mengubahnya atau mengoreksinya dengan menggunakan fitur update ini atau fitur ubah data ini lebih efektif dibanding dengan saat kita hapus input ulang hapus itu lanjutnya nubah data pengubah data di sini Sebenarnya sama dengan for menambah data bedanya song ubah data sudah terisi oleh data Apa yang akan kita ubah seperti berikut ini kita akan mengubah data buku bahasa Indonesia ubah lalu simpan perubahan selesai Bagaimana cara

def ubah\_buku(request, id\_buku):

    buku = Buku.objects.get(id=id\_buku)

    template = 'ubah-buku.html'

    if request.POST:

        form = FormBuku(request.POST)

 if form.is\_valid():

            form.save()

            messages.success(request, "Data Berhasil diperbaharui!")

            return redirect('ubah\_buku', id\_buku=id\_buku)

    else:

        form = FormBuku(instance=buku)

        konteks = {

            'form':form,

            'buku':buku,

        }

    return render(request, template, konteks)

setelah perubahan disimpan apa yang terjadi biasanya akan diri direct akan diarahkan kembali kehalaman ubah data itu sendiri lalu muncul pesan data Brasil diperbarui kira-kira seperti itu berarti di sini kita butuh redirect bukan render

**video 14.4 Crud:hapus data**

ini kita akan membuat fitur hapus data dan kita akan menambahkan kolom action atau aksi ya d e ah data buku nanti setiap record itu bisa kita hapus bisa kan kita akan menghapus bpo ada tombolnya hapus hilang datanya oke nah sebenarnya hapus data ini sangat-sangat simple ya enggak ga 2kali sangat ya simple aja simpel aja pertama-tama kit

a buat kolom dulu link untuk membuat actionnya kita ke

template buku.html di bagian header tambahkan setelah kelompok  <th>ACTION</th>

lalu dibawah kita tambahkan  <a href="{% url 'hapus\_buku' buku.id %}" class="btn btn-danger">Hapus</a>

selanjutnya membuat actionnya buat viewnya proses hapusnya

def hapus\_buku(request, id\_buku):

    buku = Buku.objects.filter(id=id\_buku)

    buku.delete()

    return redirect('buku')

terakhir membuat urls.py

 path('buku/hapus/<int:id\_buku>', hapus\_buku, name="hapus\_buku"),

data berhasil dihapus secara 10 tangkap misalnya di bagian di bawah judul buku di sini ya hal ini sama dengan ubah buku ya dengan ubah buku mana itu babuku for message dicopy saja sama Hai Pasti sangat Hai medives Coba kita hapus buku PPL apakah naik akan menghapus pemodelan perangkat lunak ya data berhasil dihapus secara hai hapus baik Nah itu bagian hapus data simple simple banget ya maunya Jadi kita cuma seperti ini jadi yang agak pusingnya agak ribet nya di bagian hapus data ini mungkin di bagian ah membuat modal ya itu ya karena dirinya cuma sedikit

**Vidio 15 Authentication login dan logut**

Proses verifikasi / validasi identitas user yang terdaftar sebelum mengakses systemdengan ini kita dapat membatasi usr mana saja yang boleh menabah kata mengubah kata.

Untuk membuat usernya baru bisa menggunakan

Python manage.py createsuperuser

Usename(leave blank to use purple):html

Password:

Password(again)

Createsuperuser created successfully

Langkah pertama membuat urlnya

From djanggo.control auth.views.import loginview

Kemudian untuk pola urlnya

Path(‘masuk/”,LoginView.as\_View() name=’masuk’)

Kemudian untuk menu login

{% extends 'base.html' %}

{% block content %}

{% if user.is\_authenticated %}

  <meta http-equiv="refresh" content="0; url={% url 'buku' %}">

{% else %}

<div class="container">

  <div class="row justify-content-center">

    <div class="col-md-4">

      <div class="card shadow-lg">

        <div class="card-header">

          LOGIN.

        </div>

        <div class="card-body">

          <form method="post">

            {% csrf\_token %}

            {{ form.as\_p }}

            <button class="btn btn-primary">Masuk</button>

          </form>

        </div>

      </div>

    </div>

  </div>

</div>

{% endif %}

{% endblock %}

Saat login berhasil maka beralih ke file setting tambahkan code di paling bawa“LOGIN\_REDIRECT\_URL=’buku’ Ketika login berhasil akan diarahkan ke halaman buku. Apabila menu belum login maka kita membutuhkan dekulator pada menu setting ditambahkan lagi LOGIN\_REDIRECT\_URL=’masuk’

cara menggunakan LOGIN\_REDIRECT\_URL=’masuk’ di file view, mengimporkan file setting dengan code”from Django.conf.import setting” dengan begitu “@login\_required(login\_url=setting.LOGIN\_URL”

Utuk logout kita membuat viewnya dulu “From djanggo.control auth.views.import loginview,logoutView, untuk pola urlnya Path(‘keluar/”,LoginView.as\_View() name=’keluar’)

**Video 15.2 mengakses username di template**

1. Mengakses user yang sedang aktif untuk diakses kedlam template
2. Dimana disini ada user yang sdang aktif tapi kita tidak tau user mana yang sedang aktif dan tidak mempunyai nama
3. Pada bagian template di program judul.html kita tambahkan,dimana variable userini yang nantinya akan mencetak user yangsedang aktif

<p> {{user }} </p>

1. Dan ketika kita keluar maka akan muncul anonymous user secara otomatis
2. cara menghilangkan anonymous user

{%if user.is-autenticated%}

<p> {{user }} </p>

{%endif%}

**Video 16 Sign up/ tambah user**

Membuat view

Langkah 1

“ from Django.contrib.auth.from import usercreationform

Def signup(request):

If request.POST:

From= usercreationfrom(request.POST)

From.save()

Messeger.success(request.”user berhasil dibuat!)

Retrun redirect(‘signup”)

Else:

Messages.error(request.”terjadi kesalahan!)

Retrun redirect(‘signup”)

Else:

From = usecreationfrom()

Konteks={

‘from’=from,

}

Retrun render(request,’signub.html,konteks)

Langkah 2 membuat file temlets

{% extends 'base.html' %}

{% block content %}

<div class="container">

  <div class="row">

    <div class="col-md-6">

      <div class="card">

        <div class="card-header">

          Tambah User

        </div>

        <div class="card-body">

          {% for message in messages %}

            {% if message.tags == 'success' %}

              <div class="alert alert-success">{{ message }}</div>

            {% else %}

            <div class="alert alert-danger">{{ message }}</div>

            {% endif %}

          {% endfor %}

          <form action="{% url 'signup' %}" method="post">

            {% csrf\_token %}

            {{ form.as\_p }}

            <button class="btn btn-primary">Simpan</button>

            <a href="{% url 'users' %}" class="btn btn-danger">Kembali</a>

          </form>

        </div>

      </div>

    </div>

  </div>

</div>

{% endblock %}

Langkah 3

Menambah pada file url path('user/', users, name='users'),

    path('user/add/', signup, name='signup'),

from django.contrib import admin

from django.urls import path

from perpustakaan.views import \*

from django.contrib.auth.views import LoginView, LogoutView

from django.conf.urls.static import static

urlpatterns = [

    path('', home, name='home'),

    path('admin/', admin.site.urls),

    path('penerbit/', penerbit, name='penerbit'),

    path('buku/', buku, name='buku'),

    path('buku/tambah/', tambah\_buku, name='tambah\_buku'),

    path('buku/ubah/<int:id\_buku>', ubah\_buku, name='ubah\_buku'),

    path('buku/hapus/<int:id\_buku>', hapus\_buku, name="hapus\_buku"),

    path('buku/export/xls/', export\_xls, name='export\_xls'),

    path('auth/masuk/', LoginView.as\_view(), name='masuk'),

    path('auth/keluar/', LogoutView.as\_view(next\_page='masuk'), name='keluar'),

    path('user/', users, name='users'),

    path('user/add/', signup, name='signup'),

]

**Video 17 Upload fille**

membahas upload file pada django sebagai contoh kita akan melakukan upload cover buku kita akan menambahkan file baru ya pada model buku yang bernama cover sehingga default tambah Bogor nanti akan muncul form input baru yaitu cover seperti ini ataupun di form ubah buku gitu dan didata Bukit akan menambahkan kolom baru yaitu cover ini sebagai informasi Buku Bagaimana caranya pertama kita buka file setting kita lakukan pengaturan dulu di baris paling bawah

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

“MEDIA\_URL = '/media/' ”untuk mengakses url file gambar kita file cover, “MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media' “ dimana file gambar tersebut akan disimpan akan diupload di folder di media

agar di media url bisa diakses di template kita tambahkan konteks prosesor media di dalam tempel di sini kita duplikat di baris request lalu kita ubah request nya menjadi media save

   'django.template.context\_processors.media',

Langkah kedua kita tambahkan file baru yaitu cover di dalam model di sini kita membutuhkan bidang model yang bernama image file image Ini mesti kita install dulu karena itu berupa PX yang bernama pillor

kembali ke model cover model image file  “cover = models.ImageField(upload\_to='cover/', null=True)

menambahkan agar file yang di upload file gambar yang diupload nanti akan masuk ke dalam folder media folder cover baru file gambarnya kita tambahkan “null=True” karena ini adalah file baru yang ditambahkan kedalam model yang sudah dibuat, tanggal = models.DateTimeField(auto\_now\_add=True, null=True)

langkah selanjutnya kita ke views    form = FormBuku(request.POST, request.FILES)

files saat kita mengirim data mencubit data dengan memompa mengirim file-file tersebut akan ditempatkan di request file ini maka itu ini juga berlaku untuk view ubah buku kita copy     form = FormBuku(request.POST, request.FILES, instance=buku)

selanjutnya kita ke template tambah buku di sini penting tambahkan atribut enctype="multipart/form-data">atribut ini wajib dibubuhkan berlaku untuk form ubah buku

**Video 18 export file**

1. Ekport file menggunakan pekage dimana terlebih dahulu kita akan menginstal pekage terlebih dahulu di terminal dengan nama pekage

Django - import – eksport

1. Pada folder perpustakaan kita buat program baru dengan nama resource.py

From import – eksport import resource

From perpustakaan.model import Buku

Class meta :

Model = buku

1. Kemudian kita buat view untuk ekspport data

From perpustakaan.resource import Bukuresource

1. Ekspor data buku kedalam export\_xls

Def exsport\_xls (request) :

Buku = Bukuresource()

Dataset = buku.exsport()

Resfond = httpresponse (dataset.xls,content\_type=’ ‘)

1. Buat URL

Path (‘export/xls/’ , exsport\_xls, name=’ exsport\_xls’),

1. Memperbolehkan data buku yang dimunculkan

Fields = [ ‘judul ‘ , ‘ tangga; ‘, ‘ kelompok \_id\_nama’,’penerbit’,’jumlah’]

<href>= {%url ‘exsport\_xls ‘ %} class =“ btn btn - success ”> exsport\_xls </href>

**VIDEO 19 (VIRTUAL ENVIRONMENT)**

Setiap kita membuat projek didalam lingkungan virtual, projek tersebut akan terisolasi dari direktori system dan memiliki paket python sendiri yang terinstall di lingkungan virtual tersebut.

1. Buka terminal atau CMD (Command Prompt) dan arahkan ke dalam directory folder project Django.
2. Jika sudah selanjutnya ketikan perintah command line seperti dibawah ini:

python -m venv env

1. Setelah berhasil membuat virtual environment tersebut maka akan terdapat folder baru didalam folder project dengan nama env, selanjutnya dapat mengaktifkannya terlebih dahulu dengan mengetikan perintah command line seperti dibawah ini:

Env\Scripts\activate

**Video 20.2 defployment**

Mempersiapkn kebtuhan apa sajautuk mendiploy aplikasikita ke server

Dimana disini kita akan mendeploy aplikash perpustakaan kita ke server

Untuk proses yang pertama kita buat dulu aplikash kita di computer server lalu di upload ke github kemundian nntinya ny akan ditarik oleh computer server untuk di defploy disana.

1. Kita remote dulu user dengan degan cara masuk menggunaakan id dan password user yang dibuat contohnya

Ssh [zulx@192.168.100.8](mailto:zulx@192.168.100.8)

1. Ada 4 bagian dalam melakukan proses defloyment

* Update install python 3
* Intall app perpustakaan
* Membuat service gunicarn
* Configurasi nginx

1. Langkah pertama yaitu kita mengaupdate engan perintah

Sudo apt - get update

1. Kemudian intall python 3 di server

Sudo apt – get install python3-pip python3-dev nginx

1. Upgrade pip dan install venv

Sudo –H pip3 install --upgrade pip

Sudo –H pip3 install virtualenv

1. Install aplikasi github yang di clone dari github,untuk penyimpanan directori akan di

simpan di /home/zulx atau bisa juga diganti dengan nama user masing”

git clone https”/github.com /writerlab/perpus

1. Untuk melihat working directori

Cd perpus

Pwd

/home/zulx/perpus

1. Sekarang kita buat virtualenv dengan nama forder env dan mengaktifkan env

Virtualenv . env

Source .env/bin/active

1. Install pakage untuk kebutuhan projek

Pip install django ==2.2.12 pillow django – import – eksport gunicorn

1. Konfogurasi host

Nano perpus/ settings.py

1. Di allowed\_host kita isikan id server kita

192.168.100.8

1. lalu buat static root

statatic root = os.path.join (BASE\_DIR, ‘static/’)

1. menyimpan konfigurasi dan keluar

control o kemudian enter

control x

1. kemudian kumpulkan semua konten static kelokasi dir yang sudah di config

python manage.py collecstatic

1. buat axception untuk port 8000

sudo ufw allow 8000

1. tes gunicarn untuk melayani projek kita

python manage.py runserver 0.0.0.0:8000

gunicorn –bind 0.0.0.0:8000 perpus.wsgi

1. keluar dari env deactivate
2. buat file service gunicorn

sudo nano /etc/system/system/gunicorn.service

1. start dan enable gunicorn

sudo system start gunicorn

sudo system enable gunicorn