

Nama: Nurmi Santi

Nim:19.01.013.056

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.

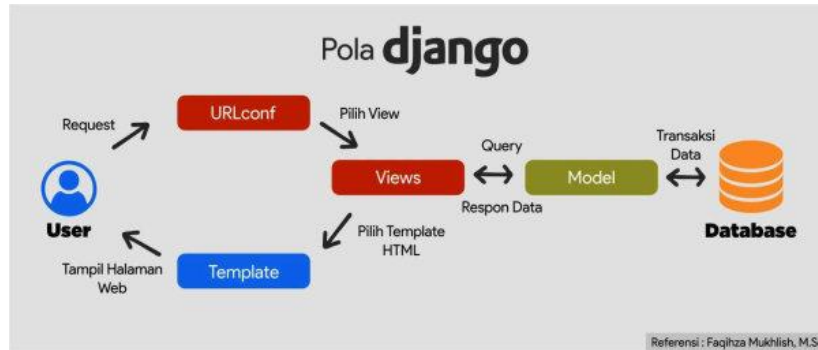
Website yang menggunakan teknologi django:

- 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 maka akan melakukan query 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 memilih 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.

2. Text Editor

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

3. Web Browser

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

4. Python 3.7.x

Saat instalasi python jangan lupa mencentang add python to PATH.

VIDEO 4 (MEMBUAT PROJECT BARU DJANGO)

Divideo dijelaskan bahwa akan membuat aplikasi perpustakaan

Ketikkan perintah Django-admin startproject perpustakaan

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

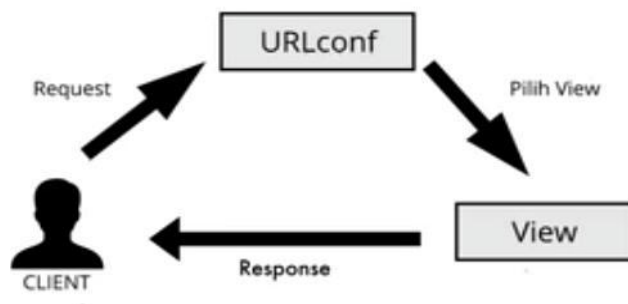
- Perpustakaan - merupakan folder root atau utama yang menampung semua kebutuhan project web yang dibuat.
- Manage.py – file ini merupakan server yang akan mengolah project web yang dibuat. Agar server web dapat berjalan file inilah yang pertama kali dieksekusi ke CMD.
- __init__.py – file ini merupakan 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 selesai 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 basic routing Django dengan pola URL

Alur sederhana aplikasi perpustakaan



1. Client melakukan request kemudian dicek oleh URLconf pilih viewnya kemudian di response

2. Client meminta request meminta ke aplikasi perpustakaan halaman buku URL buku kemudian di cek, apabila sudah di cek dan sudah ada dipilih viewnya lalu direspon dimana halaman buku

Video 6.1

Apps merupakan aplikasi pada Django yang mempunyai model database view template dan urlconf. Dimana setiap pekerja di handle oleh app, dan memiliki lebih dari 1 model app,

Membuat apps :

1. Masuk ke cd/desktop/perpus/
2. Ada dua cara membuat app
 - a. Django - admin dengan perintah startapp perpustakaan
 - b. Menjalankan file python manage.py diikuti sub perintah startapp perpustakaan
3. Setelah app dibuat sekarang kita buka code editor dengan perintah code.
4. App pada Django merupakan sebuah pages
5. Jadi disana terdapat beberapa isi dari modul perpustakaan
 - a. Admin.py yang digunakan untuk meregistrasikan model yang kita buat untuk ditampilkan di Django admin
 - b. App.py file membantu user melakukan konfigurasi
 - c. Model.py untuk mendefinisikan table ke database
 - d. Test.py digunakan untuk testing
 - e. Views.py berisi fungsi untuk melakukan proses dan akses dari client. yang nantinya views ini akan disimpan di file views yang sebenarnya.

Video 6.2 Membuat views, dimana nantinya views akan disimpan di dalam apps

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


```
from django.http import HttpResponse
def buku(request):
    return HttpResponse('Halaman Buku')
```
2. Nah setelah sudah sekarang kita masuk lagi ke file urls dimana jika kita jalankan akan terjadi error 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.
3. Untuk membuktikan hasil dari views bisa dengan cara buka 127.0.0.1:8000/buku/ Dan nantinya akan muncul hasil dari view disana seperti pada contoh

“Halaman Buku”

- Langkah selanjutnya yaitu menambahkan URL penerbit dengan cara yang sama
def penerbit (request) :
return httpResponse('Halaman penerbit')
- 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
- Jadi di views juga dapat kita sisipkan kode HTML yaitu
“h1 (header) “ yang digunakana untuk mengatur besar tulisan pada output

Video 7.1 Templates

- Pada file setting tambahkan templates di dalam code DIRS yang bermaksud sebagai tempat file HTML kita disimpan yaitu di file directori.
'DIRS' 'templates'
- Kemudian kita buat folder Templates dan didalamnya kita buat file HTMLnya yaitu (Buku.HTML)
- Untuk langkah selanjutnya kita buat code HTML sederhana
<!DOCTYPE html>
<title> perpustakaan </title>
</head>
<body>
<center>
<h1> perpustakaan </h1>
</center>
<h1> Daftar Buku </h2>
</body>
</html>
- Kemudian kita masuk ke views lalu hapus
From django.http import HttpResponse
- Lalu isi kan dengan code dibawah,save lalun liat hasilnya.

```
def home(request):  
  
    template = 'buku.html'  
  
    return render(request, template)
```

- 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. Substitusi variable

- a. Yang digunakan untuk menampilkan variable ke template yang dilemparkan dari views.

```
def buku (request) :  
    judul = "belajar django"
```

```
    konteks {  
        'title' = judul
```

- b. `return render (request, 'buku.html', konteks)`

kemudian tambahkan `{{ title }}` pada buku .html dan lihat hasilnya.

Jadi disini juga dapat mengubah filter pada tulisan yang digunakan dengan cara `{{title | upper}}` untuk tulisan besar semua dan `lower` untuk tulisan kecil semua.

- c. Melakukan perulangan dengan menggunakan template django dimana ada 3 judul "belajar django" "belajar python" dan "belajar bootstrap"

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
```

2. Filter

Digunakan untuk memodifikasi variable yang akan ditampilkan

3. Tags

Untuk melakukan control view seperti looping, 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 proyek perpus, jadi disini nanti nya akan ada template utama dengan nama `base.html` yang isinya akan menampilkan konten yang dilempar dari template yang ada di dalam apps, jadi di dalam folder app akan dibuat sebuah folder baru yang isinya hanya ada konten saja dan konten yg ada nantinya akan dimasukkan kedalam `base.html`.

- a. Untuk langkah pertama kita buat file `base.html` di dalam folder template

```
<!DOCTYPE html>  
<title> perpustakaan sekolah</title>  
</head>  
<body>
```

```

{% block content %} = jadi untuk block content ini merupakan block yang
akan diisi oleh konten” dari apps
{% andblock %}
</body>
</html>

```

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

```

{% extend "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 dengan daftar penerbit.

- c. Untuk bagian judul juga bisa dipisahkan untuk template nya dengan cara buat judul.html pada folder template di dalam base.html buat code
- ```

{% include 'judul.html' %}

```

## Vidio 8 Static files

Static file adalah kumpulan file css, js, dan gambar

Bagaimana konfigurasi static filenya untuk memanggil file css, js dan gambar 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 folder yang bernama “static”

Sedangkan STATIC\_URL akses untuk browser untuk mengakses 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, images dan js

## Video 8.2 Setup bootstrap

Hanya memindahkan file css, js ke dalam folder static Untuk yang belum ada Bootstrap download terlebih dahulu di [getbootstrap.com](http://getbootstrap.com) dan juga jQuery download di [code.jquery.com](http://code.jquery.com)

Kemudian file bootstrap.min.css dimasukkan ke folder css, file bootstrap.min.js 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 bootstrap.min.js “ `<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.

```
DATABASES = {
 'default': {
 'ENGINE': 'django.db.backends.sqlite3',
 'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
 }
}
```

2. Lakukan migrasi untuk menerapkan model ke database.
3. Jika belum membuat 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'

'PASSWORD' : 'ranger46' ,

'HOST' : 'localhost'

'PORT' : '3306',

}
```

2. Install mysqlclient.
3. 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 Jenggo buku penerbitan lain itu merupakan model-model model nanti yang kita buat akan didaftarkan ke dalam Jenggo 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 Django 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

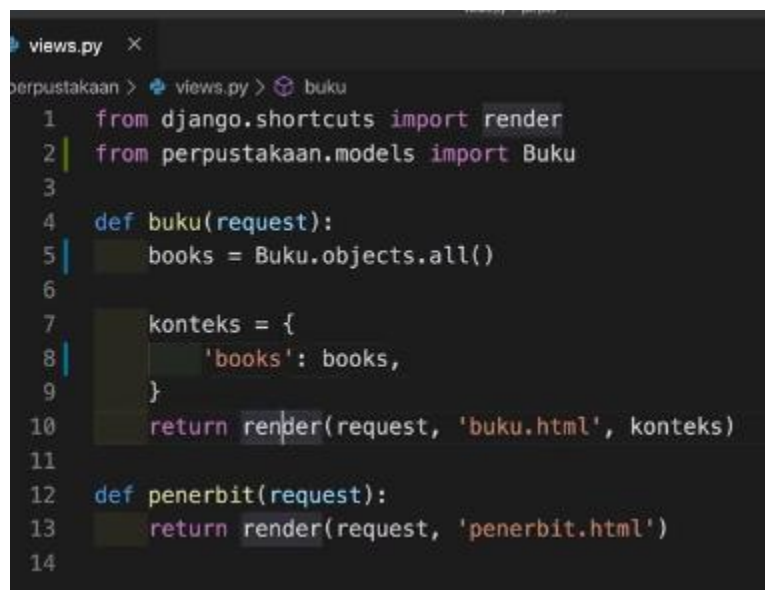
adalah nama kelompok maka ini akan tampil saat kita memanggil film yang maka dia akan mengembalikan Nama judulnya apa Nama kelompoknya .

## Video 12 ORM(objek relation mapping)

Teknik yang digunakan dalam pemograman untuk menggunakan basis data relational sebagai penyimpanan data dengan bentuk objek. Django 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 jadi ini akan mengambil data dari tabel buku dan ditampilkan kita soroti bagian kuerinya select Bintang from buku ini sama dengan buku titik objek titik Hal inilah yang disebut dengan kueri itu kalau

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



```
views.py x
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
```

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 django 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 kolom 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>
 Kembali
 </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

```

perpustakaan > views.py > buku
1 from django.shortcuts import render
2 from perpustakaan.models import Buku
3 from perpustakaan.forms import FormBuku
4
5 def buku(request):
6 # select * from Buku
7 # inner join kelompok.id=buku.kelompok_id
8 # where kelompok.nama='Produktif'
9 # limit 3
10 books = Buku.objects.filter(kelompok_id__nama='Produktif')[:4]
11
12 konteks = {
13 'books': books,
14 }
15 return render(request, 'buku.html', konteks)

```

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 di 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 kita 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 logout

Proses verifikasi / validasi identitas user yang terdaftar sebelum mengakses system dengan ini kita dapat membatasi user mana saja yang boleh menambah kata mengubah kata. Untuk membuat usernya baru bisa menggunakan Python manage.py createsuperuser

Username(leave blank to use purple):html

Password:

Password(again)

Createsuperuser created successfully

Langkah pertama membuat urlnya

From django.contrib.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 bawah “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, mengimpor file setting dengan code “from Django.conf import setting” dengan begitu “@login\_required(login\_url=setting.LOGIN\_URL)”

Untuk logout kita membuat viewnya dulu “From django.conf.urls import loginview,logoutView, untuk pola urlnya Path(’keluar/’,LoginView.as\_View() name=’keluar’)

## Video 15.2 mengakses username di template

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

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

- Dan ketika kita keluar maka akan muncul anonymous user secara otomatis
- cara menghilangkan anonymous user

```

{%if user.is-autenticated%}
<p> {{user }} </p>
{%endif%}

```

