

Nama : Murni Anita Wulandari

NIM : 20.01.013.033

Kelas : Pemrograman Mobile B

- **INTRODUCTIO :**

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. Jika biasanya sebuah framework itu berbasis MVC (Model View Controller), pada django ini ada 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.

- **WORKFLOW :**

Client melakukan Request. Misalnya meminta URL [Zulhilmi.id/profile](http://Zulhilmi.id/profile). Lalu dicek oleh URLconf, apakah Request Profile ada di aplikasi Django? Jika ada, Maka akan langsung memilih View yang sesuai. 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. Selanjutnya View memberikan data tersebut ke Template untuk Selanjutnya di Respons kepada Client.

- **APA SAJA YANG DIBUTUHKAN :**

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

- **MEMBUAT PROJECT :**

cd Desktop Untuk menyimpan Project di Halaman Desktop

Django-admin startproject perpus. Perintah ini untuk membuat Project baru Bernama Perpus

- 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.

- **BASIC ROUTING :**

Client memberikan Request untuk mengakses Halaman Buku pada Server Django namun PAGE NOT FOUND. Karena URL Buku yang di Request belum ada di Server. Untuk dapat melayani Request dari Client, harus dibuatkan URL baru seperti pada gambar berikut. 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.

- **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.

- **TEMPLATES :**

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

- **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.

- **TEMPLATE EXTENDING :**

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

- **STATIC FILES :**

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.

- **SETUP BOOTSTRAP :**

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

- **SETUP DATABASE :**

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

- **SETUP DATABASE MySQL :**

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

- **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. 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:**

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

- **DJANGO ADMIN :**

Django admin adalah salah satu fitur yang power full ada pada djargon dikatakan power full karena bisa melakukan kru sederhana untuk mengolah data pada model yang kita buat. Biasanya menggunakan django ini untuk mengetes model yang kita buat apakah sudah selesai, sesuai, relasinya sudah benar. Sederhana tetapi sangat membantu dan lagi dajango admin ini bersifat private tidak semabarang orang bisa mengaksesnya harus log in dulu. Pertanyaannya ada dua, Bagaimana cara kita log in django admin ini? Dan bagaimana model-model kita di daftarkan ke dalam django admin ini? Mari kita praktikkan. 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 : ModelAdmin:**

Masih membahas django admin, bagian kedua diepisode kali 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:**

ORM (object relational mapping) merupakan teknik yang digunakan dalam pemrograman untuk menggunakan basisdata 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.

- **FORM: ModelForm:**

Membuat file baru di perpustakaan dengan nama form. Lalu membuat viewsnya.

- **FORM: WIDGETS:**

Didalam forms.py ditambahkan widgets,textInput merupakan typenya tambahkan atribut kelas didalamnya.

- **CRUD: MENAMBAH DATA:**

Episode kali ini kita sudah masuk ke queryset, melakukan menambah data,menampilkan,mengubah dan menghapus data. Dari data yang sudah dibuat, 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, apabila sudah maka akan di simpan ke database. Lalu buka text editor buka file views. Fungsi dari csrf adalah untuk mngamankan form yang dibuat.

- **CRUD: MENAMPILKAN DATA:**  
Menampilkan data buku yang sudah dibuat, seperti menampilkan nilai 90/100. Dan menampilkan kelompok seperti produktif maka hasilnya pasti eror karena kelompok\_id merupakan type integer. Apabila menambah \_\_nama maka akan bisa mengeluarkan hasilnya. Penggunaan queryset tidak sepanjang query. Untuk menampilkan limit pada queryset cukup menambah diujungnya [:3], sedangkan dalam query limit 3.
- **CRUD: MENGUBAH DATA:**  
Mengubah data, disini perannya sangat penting karena saat kita salah dalam melakukan input data maka kita harus mengubahnya atau mengeditnya dengan menggunakan fitur update ini atau fitur ubah data. Ini lebih efektif ya, dibandingkan dengan saat kita hapus input ulang begitu seterusnya saat melakukan kesalahan. 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: MENGHAPUS DATA:**  
Hapus data juga penting dalam aplikasi, karena apabila kita sudah perlu data tersebut mau di gunakan apalagi. Fungsinya untuk menghapus data yang sudah tidak digunakan oleh user. Sebelum menggunakan action/aksi hapus sebelumnya terlebih dahulu membuat konfirmasi yang berisi apakah yakin ingin menghapus atau tidak.
- **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:**  
Episode kali ini membahas tentang mengakses user yang sedang aktif untuk ditampilkan ke dalam template. Biasanya kita login ke dalam aplikasi ada terdapat informasi user atau siapa yang saat ini sedang login atau yang sedang menggunakan aplikasi. Apabila saat kita log out dan tidak terdapat informasi maka kita bisa menggunakan attributes is\_authenticated. Jika user sudah melakukan log in maka tampilkan informasi user nya.
- **SIGN UP/ TAMBAH USER:**  
Episode kali ini kita akan membuat form sign up agar membuat user bisa login ke aplikasi perpustakaan kita. Sebenarnya ini berhubungan dengan episode django admin karena di dalamnya terdapat username django adminnya. Kalau kita lihat di dokumentasi resmi ini ada sebuah user form yang bernama UserCreationForm untuk membuat user baru.

- **UPLOAD FILE:**

Episode kali ini kita akan membahas tentang upload file seperti misalnya kita upload 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// :**

Episode kali ini kita akan membuat export file ini biasa digunakan untuk membuat report dalam 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.

- **VIRTUAL ENVIRONMENT//:**

Kali ini kita 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//:**

Kita akan mendeploy aplikasi kita ke server. Dengan awalan mengcoding di laptop kita 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.