Autentikasi :

1. Implementasi fungsi Login & Logout

• Implementasi fungsi Login

*Saat membuka page lab-9, url akan menjalankan fungsi index pada views.py. Pada fungsi ini, akan di cek apakah key ‘user\_login’ sudah ada pada session.*

*- Jika sudah, berarti user sudah login sebelumnya dan fungsi akan redirect ke url lab-9/profile. Url ini akan memanggil fungsi profile. Pada fungsi profile dicek apakah user sudah login (supaya tidak error saat url dijalankan sebelum login). Jika belum login maka, akan di-redirect ke url index. Jika ternyata sudah login, maka fungsi akan memanggil fungsi set\_data\_for\_session. Fungsi set\_data\_for\_session akan set data-data pada response dari request session yang sudah diambil sebelumnya saat login. Kemudian, data drones pada response akan diambil dari fungsi get\_drones pada api\_enterkomputer.py yang akan diubah dalam bentuk json. Setelah itu dilakukan pengecekan apakah key ‘drones’ sudah ada pada request session, jika ada maka key ‘fav\_drones’ pada response akan diisi dengan ‘drones’ pada request session. Jika tidak ada maka ‘fav\_drones’ akan diisi kosong (supaya tidak error saat pertama kali login). Kemudian akan ditampilakan halaman profile.html*

*- Jika belum, maka akan ditampilkan halaman login.html. Isi username dan password, kemudian klik submit. Form akan memanggil url auth\_login yang akan memanggil fungsi auth\_login pada custom\_auth.py. Pada fungsi ini, akan mengambil username dan password yang diinput user. Kemudian memanggil fungsi get\_access\_token pada csui\_helper.py untuk mendapatkan access\_token (jika username dan password benar). Lalu akan memanggil fungo verify\_user pada csui\_helper.py yang akan mengembalikan access\_token dan client\_id. Kemudian request session keys akan diset, kemudian redirect ke url index.*

• Implementasi fungsi Logout

*Pada profile.html,terdapat button logout yang jika diklik akan memanggil url auth\_logout. Url ini akan memanggil fungsi auth\_logout pada custom\_auth.py. Fungsi ini akan memanggil fungsi flush() untuk request session untuk menghapus semua session, kemudian redirect ke url index.*

2. Jelaskan Bagaimana Django Apps kalian mengetahui bahwa setiap user yang menagkses Apps kalian, akan dilayani secara berbeda? (Data milik pengguna tidak akan saling tertukar). Tunjukkan didalam baris kode

*Akses apps akan dilayani secara berbeda karena tiap user memiliki session id masing-masing yang berbeda yang ada pada baris kode 116 get\_data\_session dan baris 120 set\_data\_for\_session.*

*Get\_data\_session berguna untuk mengembalikan data id session user.*

*Set\_data\_for\_session berguna untuk apabila user sudah login, maka akan di set datanya dan dimasukkan ke response. Digunakan dalam fungsi dashboard. Akan get\_data\_user yang dimasukkan berdasarkan user\_login.*

Dashboard :

1. Mendaftar dan menggunakan OMDb API untuk menampilkan daftar film

*Daftar dari web omdb yang akan mengembalikan API key, dan apabila diakses url nya dengan judul dan tahun tertentu, akan mengembalikan JSON.*

2. Menggunakan DataTables untuk menampilkan daftar film dan filtering

*Pada list.html, terdapat javascript DataTables yang membuat tampilan daftar film dan filtering ada pada omdb.api*

3. Membuat fungsi pencarian film berdasarkan judul dan tahun

4. Jelaskan Bagaimana Django Apps kalian bisa melakukan pagination dan melakukan pencarian?

*Fungsi pagination ada pada omdb.api di mana mengambil 30 data dan dikelompokkan 10 demi 10.*

Koleksi Film

1. Membuat daftar film "tonton nanti" (disimpan di session)

2. Membuat daftar film "sudah ditonton" (disimpan di model)

3. Jelaskan Bagaimana Django Apps kalian bisa menyimpan film yang ingin ditonton, tanpa Login terlebih dahulu?

4. Jelaskan Bagaimana Django Apps kalian bisa menyimpan film yang ingin ditonton kedalam database, dan membedakan kepemilikan dari massing - masing daftar film? Tunjukkan dalam baris kode