**RECOMMENDATION AND PLAY (RE-PLAY)**

Disusun untuk Memenuhi Tugas Mata Kuliah Pemrograman Dekstop yang Dibimbing oleh Moh. Kautsar Sophan, S.Kom., M.MT.

****

Disusun oleh:

Achmad Farid Alfa Waid (190411100073)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2021**

**PEMBAHASAN**

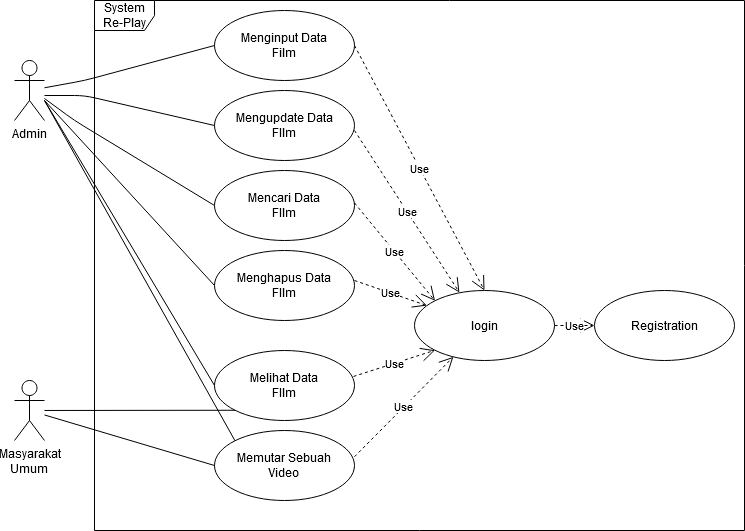
1. **DESKRIPSI APLIKASI**

Dikarenakan adanya pandemic covid – 19, semua kegiatan yang awalnya dilakukan di luar rumah, sekarang dialihkan untuk dilakukan di dalam rumah atau biasa disebut dengan Work From Home (WFH). Hal ini membuat banyak orang mulai jenuh dan stress sehingga banyak yang mencari film untuk dijadikan hiburan selama di rumah, bagi sebagian orang yang merupakan penikmat film sudah pasti memiliki list film yang akan mau di tonton, tetapi bagi orang lain yang jarang menonton film akan kesulitan mencari film yang bagus untuk di tonton, karena jika film yang ditonton jelek, maka akan semakin jenuh dan stress.

Banyak cara yang dilakukan oleh orang – orang agar terhindar dari hal tersebut, seperti mencari referensi dari internet ataupun memebuka sebuah forum di media social untuk meminta saran sebaiknya harus menonton film apa. Hal tersebut dirasa kurang efisien karena harus membuka sebuah forum dan menunggu tanggapan dari orang lain. Dan ada kemungkinan juga tidak akan ada yang menanggapi forum yang di buat orang tersebut. Agar lebih efisien dan tidak perlu membuka forum lagi dibuatkan sebuah aplikasi yang berisi data – data film yang dinilai bagus berdasarkan review dan mendapatkan rating tinggi dari orang – orang. Tidak hanya dari 1 genre film, ada beberapa genre sehingga orang yang akan mau menonton tidak bosan dengan alur cerita yang sama. Beberapa fitur yang tersedia pada aplikasi seperti berikut:

1. Menampilkan data – data film sesuai genre dengan rank rating tertinggi sampai terendah.
2. Menambah data film dengan informasi lengkap seperti judul film, tahun rilis film, synopsis film, review film tersebut, dan rating yang diberikan untuk film tersebut.
3. Mencari data film yang dinginkan.
4. Mengedit data film untuk menambahkan review tambahan atau merubah rating jika dirasa rating film tersebut terlalu rendah.
5. Menghapus data film (hanya untuk admin)
6. Menonton film yang tersedia
7. **USE CASE DIAGRAM**

Use case diagram merupakan suatu gambaran atau representasi dari interaksi yang tejadi antara sistem dengan lingkuangannnya(pemakainya). Use case termasuk kedalam salah satu tipe dari diagram UML (Unified Modeling Language), dimana use case didefinisikan dari sudut pandangaktor yang terlibat. Istilah aktor disini merujuk pada orang atau alat yang menggunakan sistem tersebut atau bisa juga siapapun yang berinteraksi dengan system.

****

Aplikasi memiliki 2 aktor, yaitu Admin dan Masyarakat Umur (User). Admin dapat mengelola data pada aplikasi, melihat tampilan data film, dan menonton film. Sedangkan untuk User hanya dapat melihat tampilan data film dan menonton film yang sudah tersedia. Untuk mengakses masing – masing fitur, Admin dan User harus melakukan login terlebih dahulu, jika belum mempunai akun untuk login, harus registrasi akun terlebih dahulu.

1. **DATABASE**

Untuk database yang ada di dalam aplikasi, mengunakan database sqlite yang diberi nama test. Terdapat 2 table di dalam database tersebut, yaitu table users dan table listfilm.

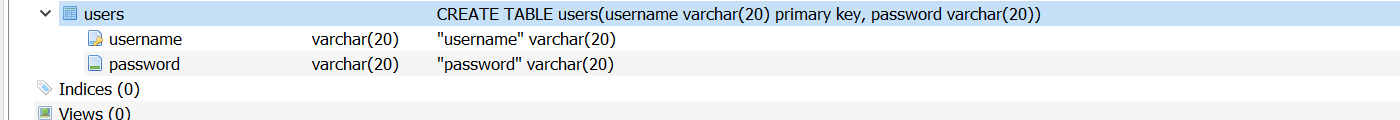
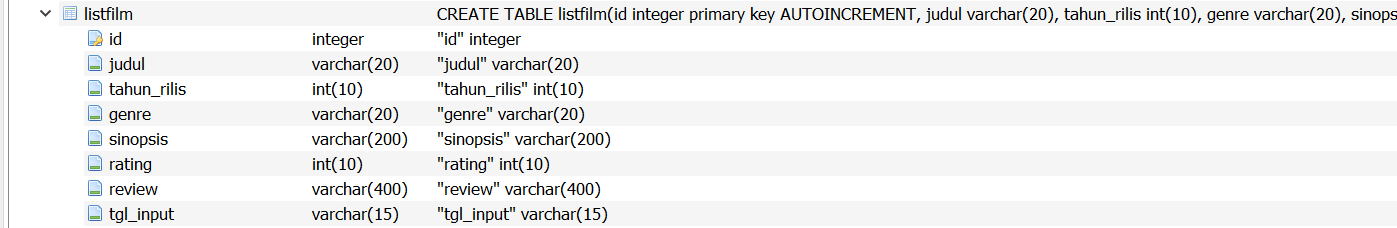


Table users memiliki 2 kolom, yang berisi username dan password kedua kolom tersebut menggunakan type data varchar. Table ini digunakan user untuk mengakses aplikasi.



Sedangkan untuk table listfilm memiliki 8 kolom yaitu sebagai berikut:

1. id : Merupakan id dari data film dan menggunakan type data integer
2. judul : Merupakan judul dari film dan menggunakan type data varchar
3. tahun\_rilis : Merupakan tahun perilisan film dan menggunakan type data

integer

1. genre : Merupakan genre dari film dan menggunakan type data varchar
2. sinopsis : Merupakan synopsis dari film dan menggunakan type data varchar
3. rating : Merupakan rating yang diperoleh film dan menggunakan type data

integer

1. review : Merupakan review yang diberikan oleh user pada film dan

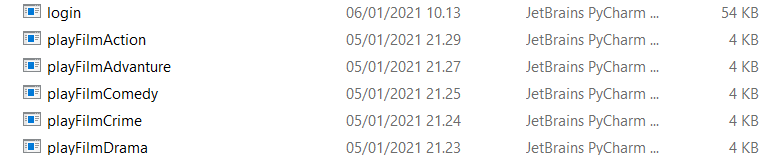
menggunakan type data varchar

1. tgl\_input : Merupakan tanggal input data film tersebut pada aplikasi dan

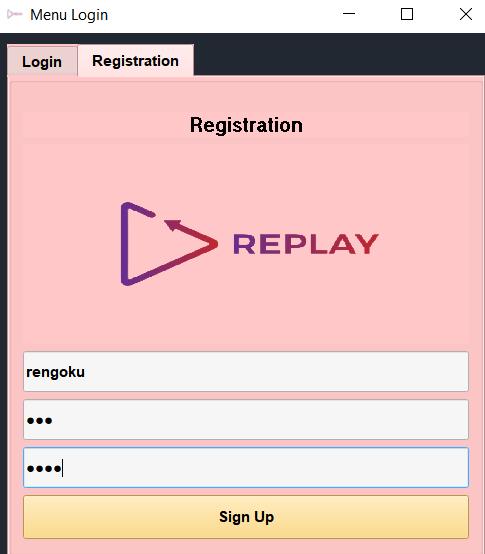
menggunakan type data varchar

1. **FILE PYTHON**

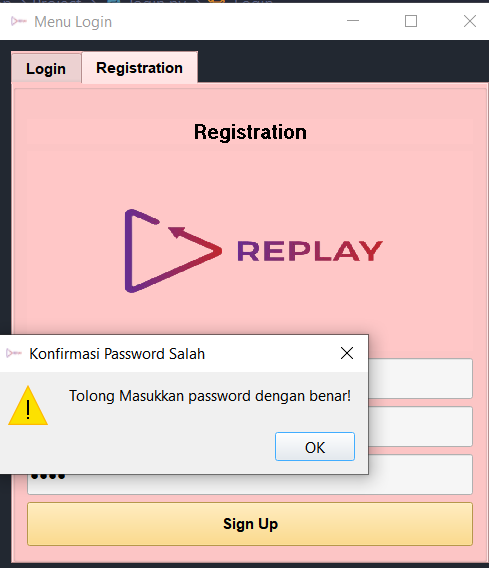
Terdapat 6 file python dari aplikasi saya, yang pertama adalah file python utama yang bernama login.py dan yang lain adalah file python playFilmAction.py, playFilmAdvanture.py, playFilmComedy.py, playFilmCrime.py, dan playFilmDrama.py yang digunakan untuk menonton film berdasarkan masing – masing genre.



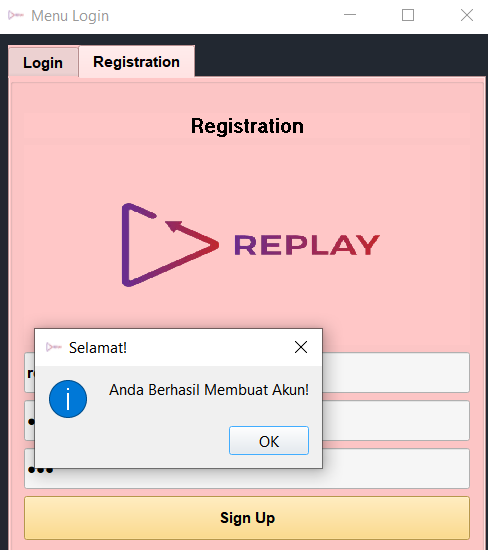
1. **CAPTURE HASIL RUNNING**
2. Melakukan Registrasi / Membuat Akun



Apabila pada konfirmasi password salah, maka akan menamilkan QMassageBox seperti berikut

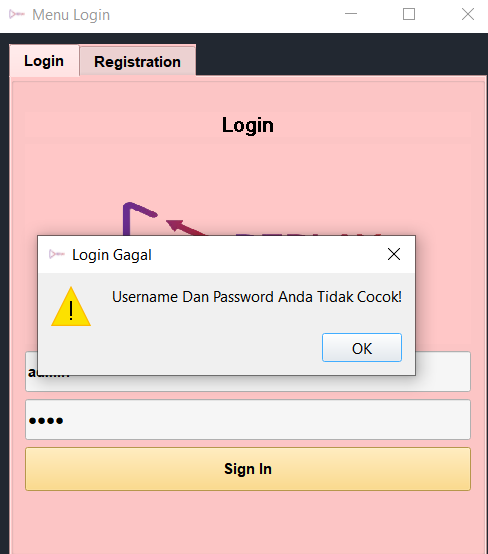


Apabila sudah berhasil melakukan konfirmasi dan berhasil membuat akan menampilkan QMassageBox seperti berikut

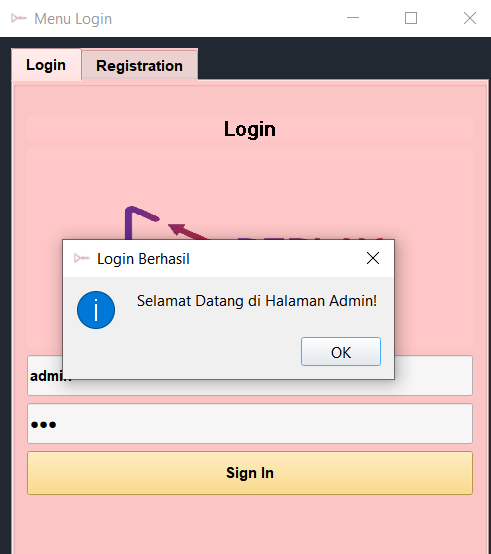


1. Melakukan Login

Bisa menggunakan akun admin dan akun user, dan bila saat pengisian username dan password terdapat kesalahan, maka akan menampilkan QMassageBox seperti berikut

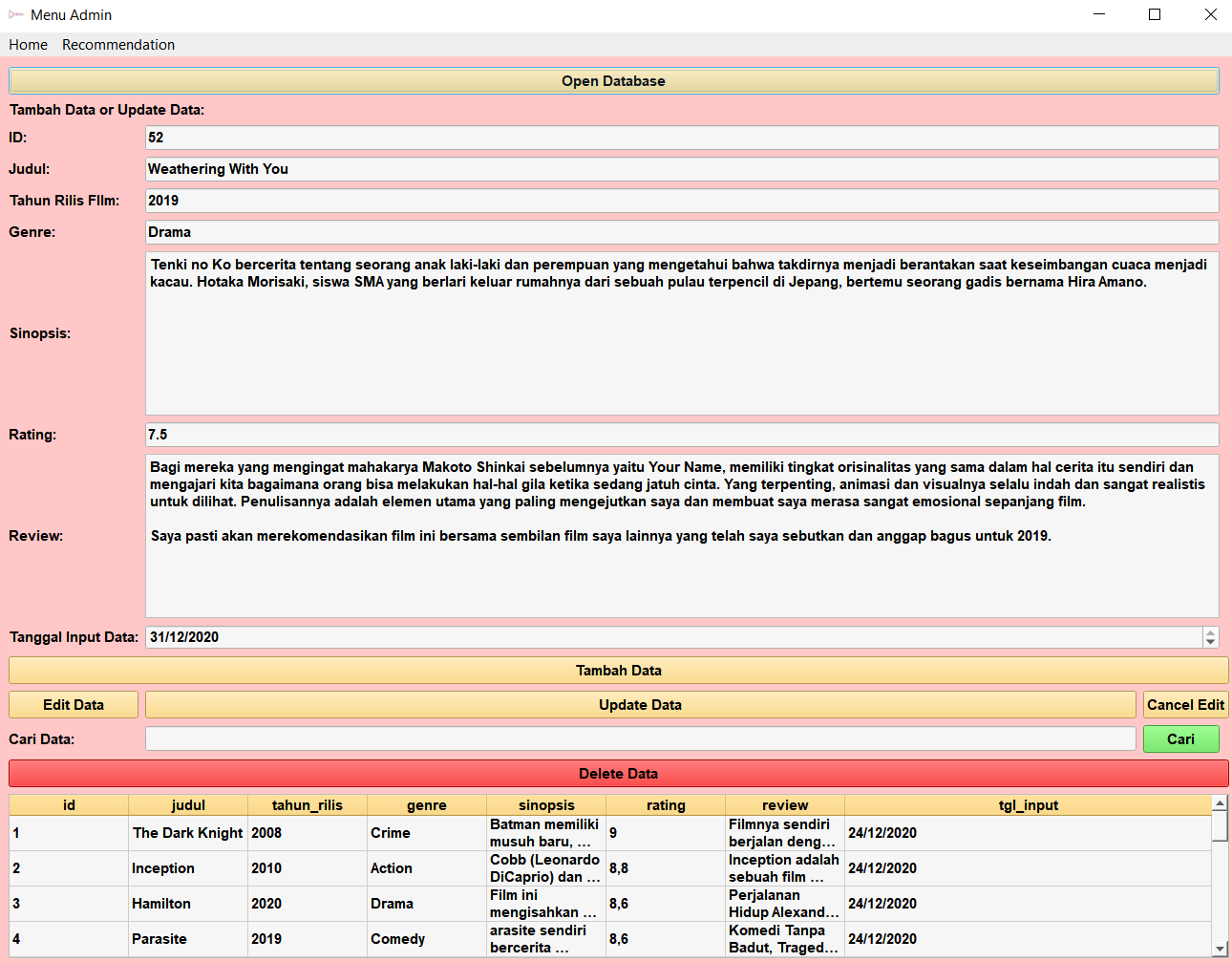


Apabila username dan password sudah cocok dan berhasil maka akan menampilkan QMassageBox seperti berikut

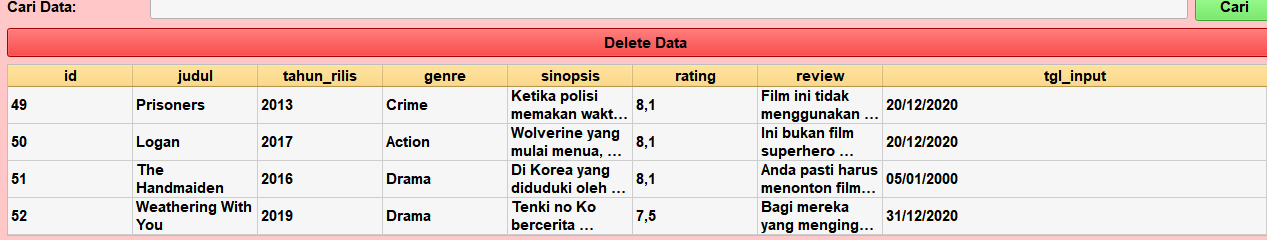


1. Halaman Admin

* Pada halaman Admin, admin dapat menambahkan data

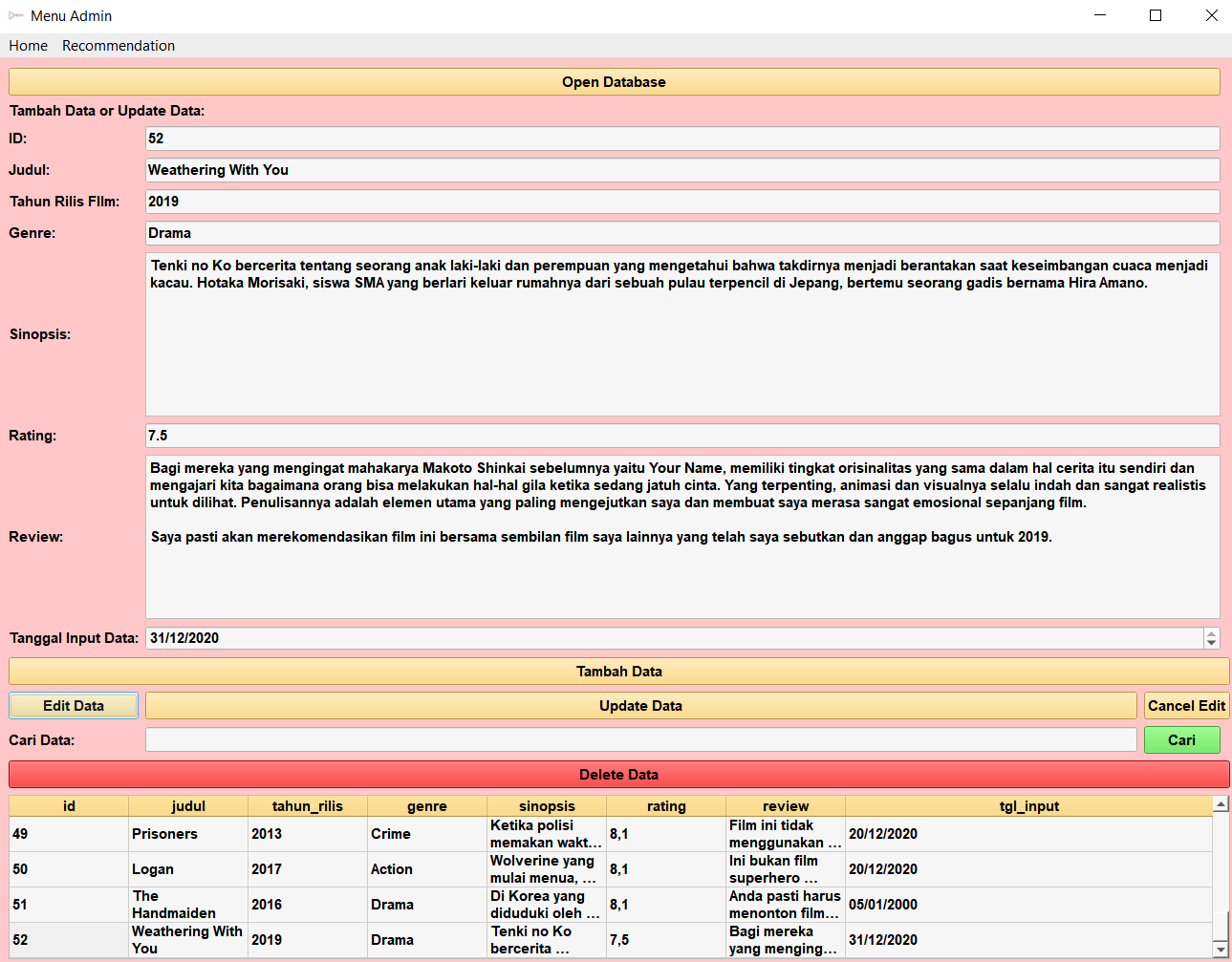


Selanjutnya menekan button Tambah data, maka data tersebut akan masuk ke dalam database dan ditampilkan ke dalam tableview

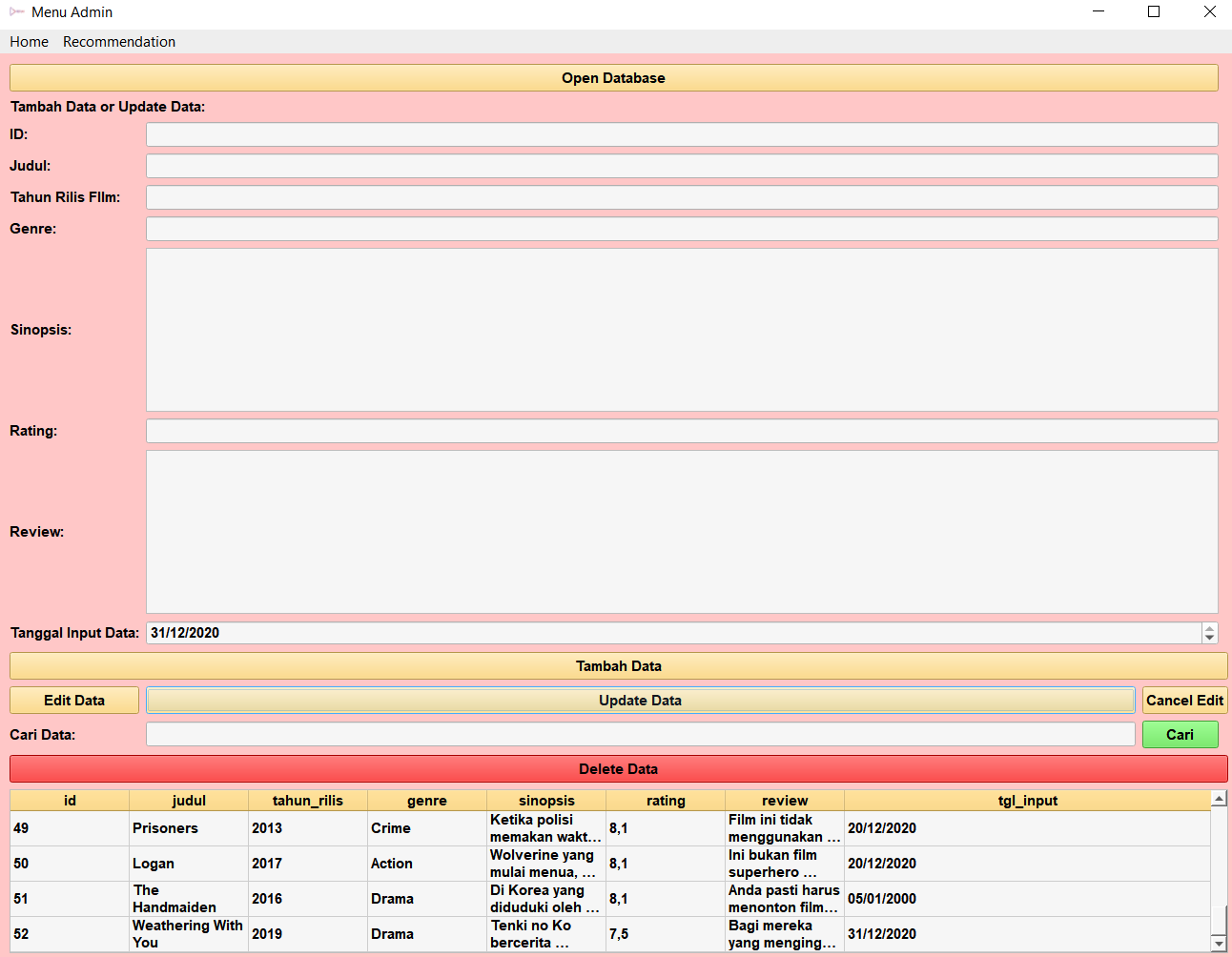


* Mengedit / Mengupdate suatu data

Pertama memilih suatu data kemudian menekan tombol Edit Data agar data tersebut ditampilkan ke widget line edit



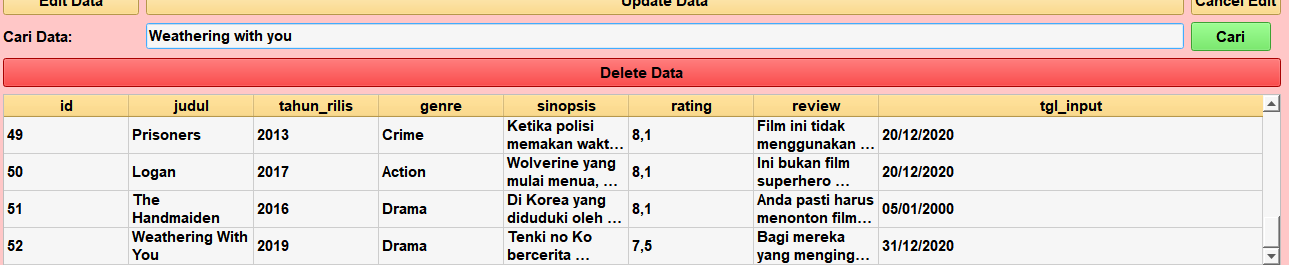
Apabila tidak jadi mengedit data tersebut, dapat menekan tombol Cancel Edit. Tetapi apabila ingin mengaupdate data tersebut segera isi data yang ingin di update kemudian jika sudah selesai dilanjutkan dengan menekan tombol Update Data, makan data tadi berhasil di update dan ditampilkan Kembali ke tableview



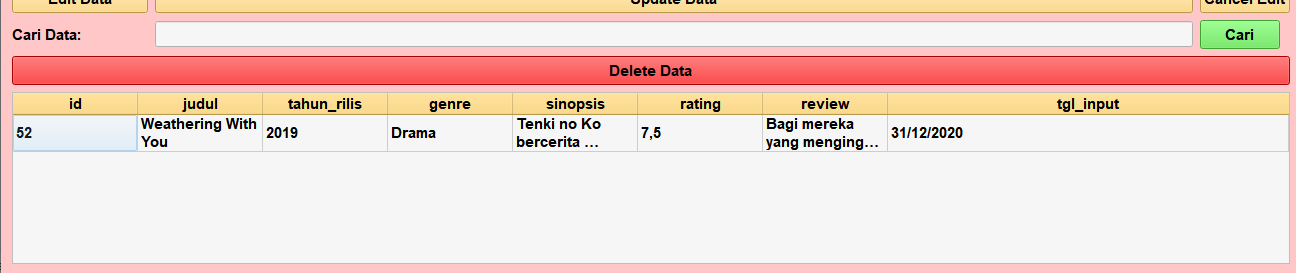
Maka widget pada line edit akan kosong kembali seoerti gambar di atas

* Mencari Data

Dengan cara memasukkan kata kunci data yang ingin dicari pad line edit seperti berikut

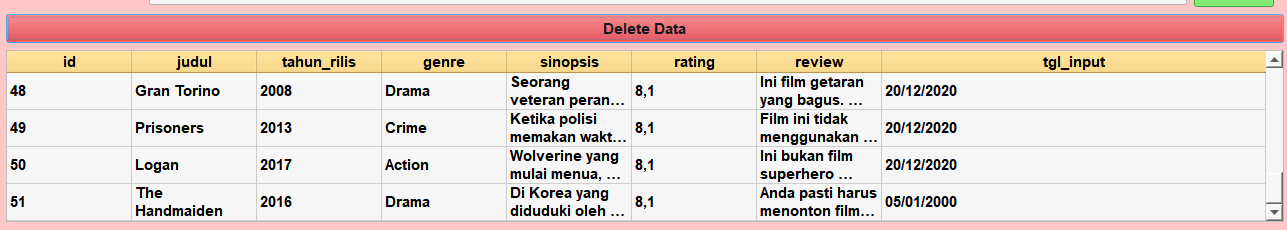
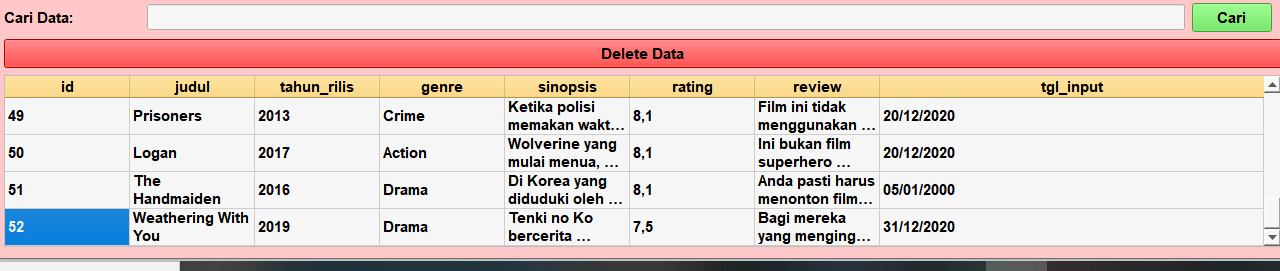


Kemudian menekan button Cari, maka akan menampilkan data sesuai dengan hasil inputan pada line edit seperti berikut

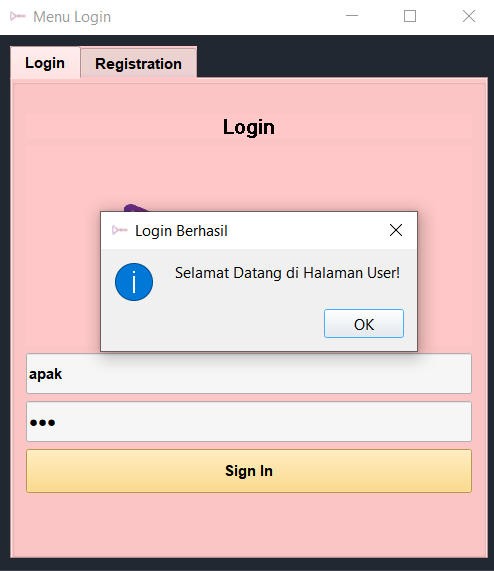


* Menghapus Data

Dengan cara memilih id data yang ingin dihapus, kemudian menekan tombol Delete Data, maka data yang dipilih akan terhapus, baik pada database ataupun tableview



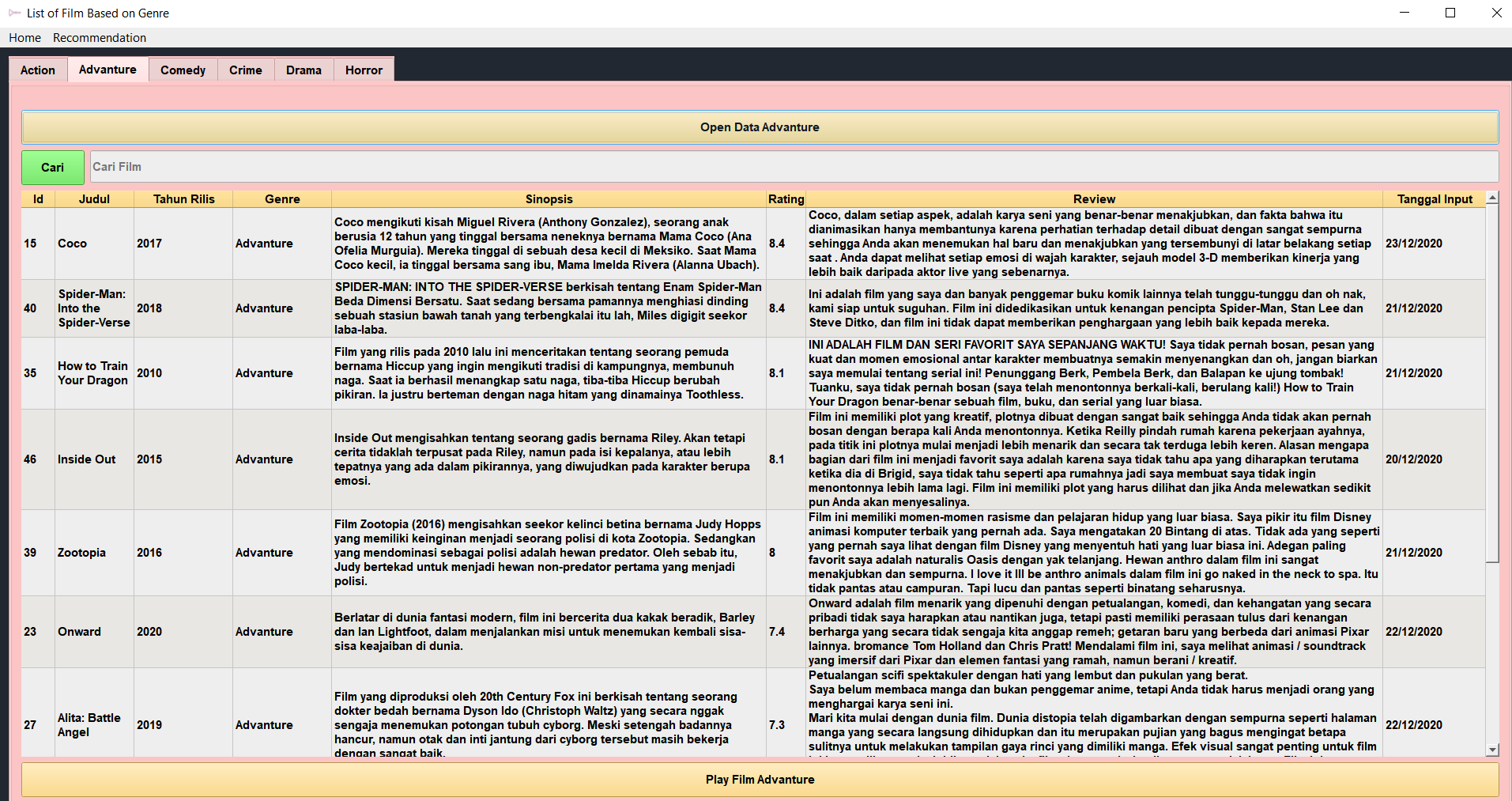
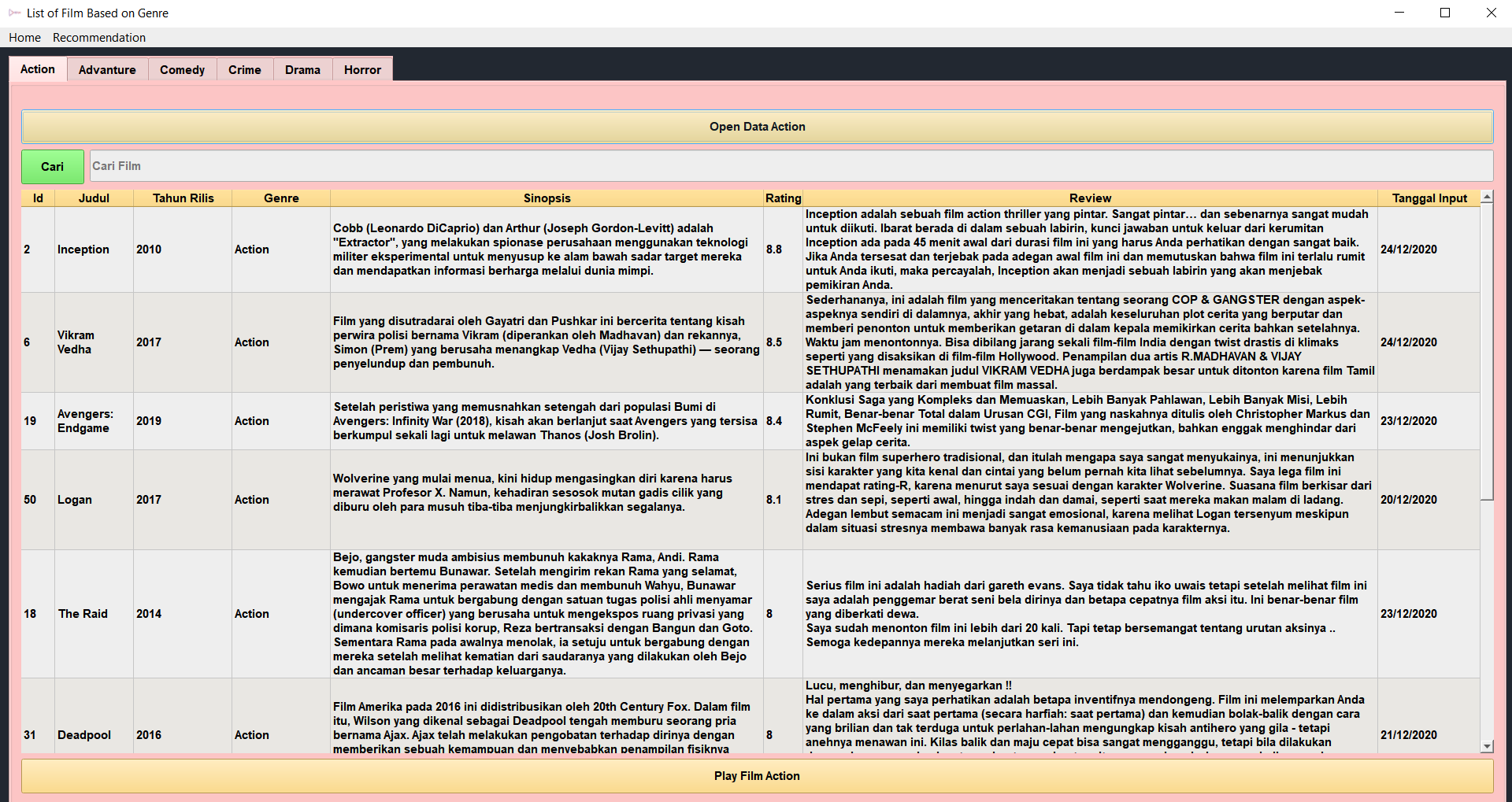
1. Halaman User

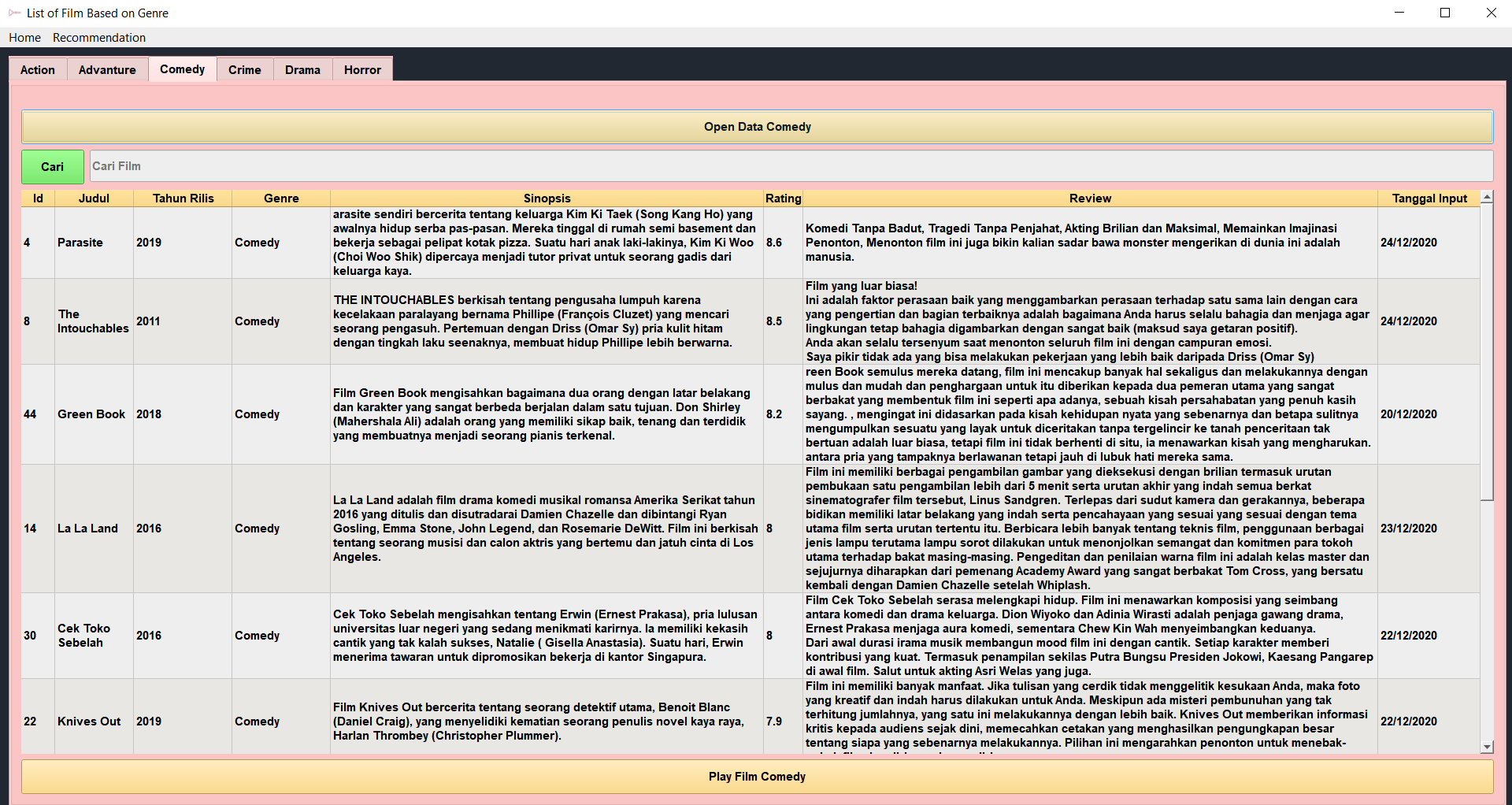


Halaman awal user merupakan instruksi yang diberikan oleh admin untuk users, dan terdapat 2 menu bar, yaitu menu Home untuk menutup aplikasi dan menu Recommendation untuk membuka halaman rekomendasi

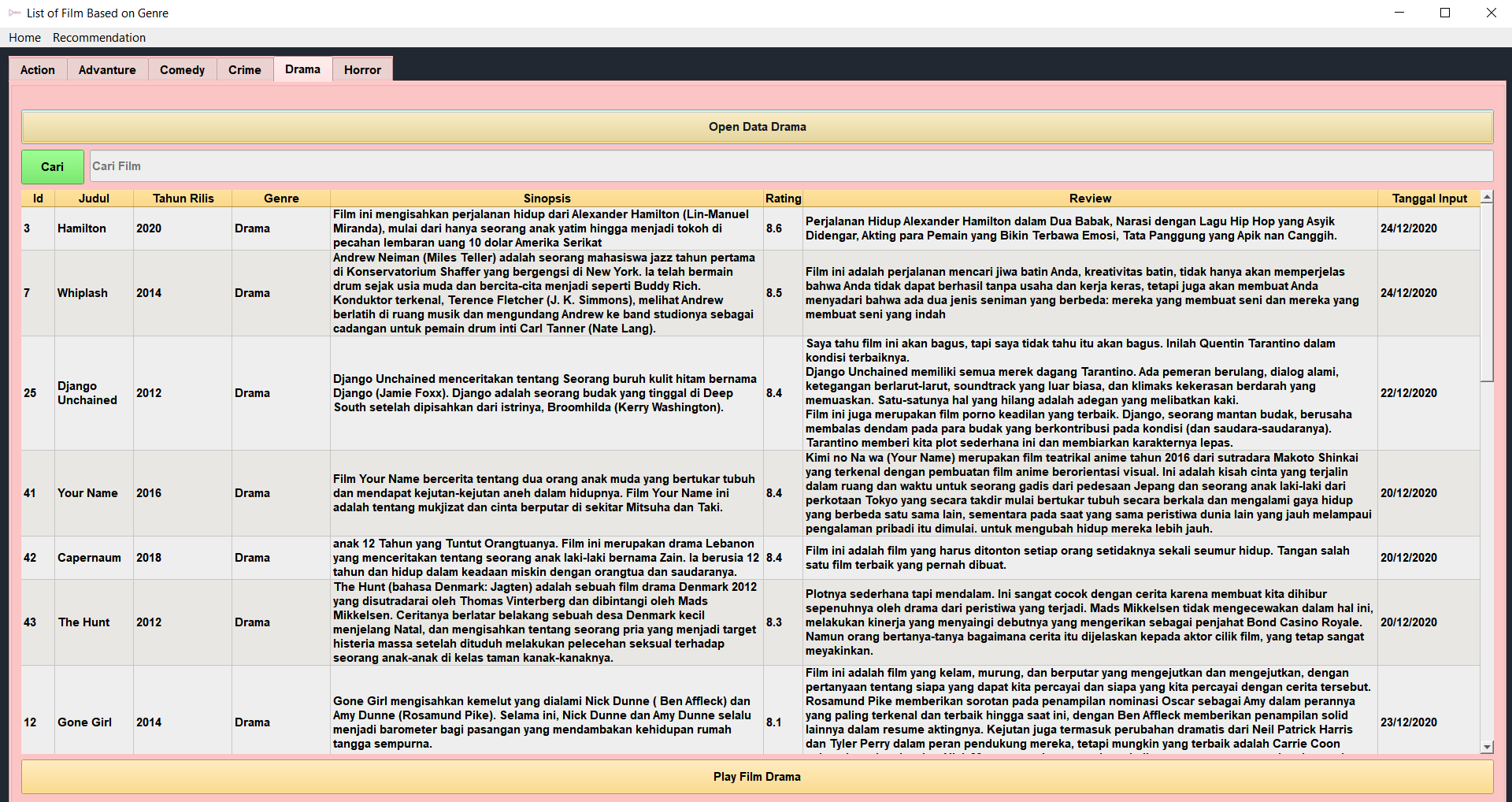


Pada tampilan halaman rekomendasi, terdapat beberapa pilihan genre, yang masing – masing genre menampilkan data berdasarkan nilai rating tertinggi ke rating terendah.



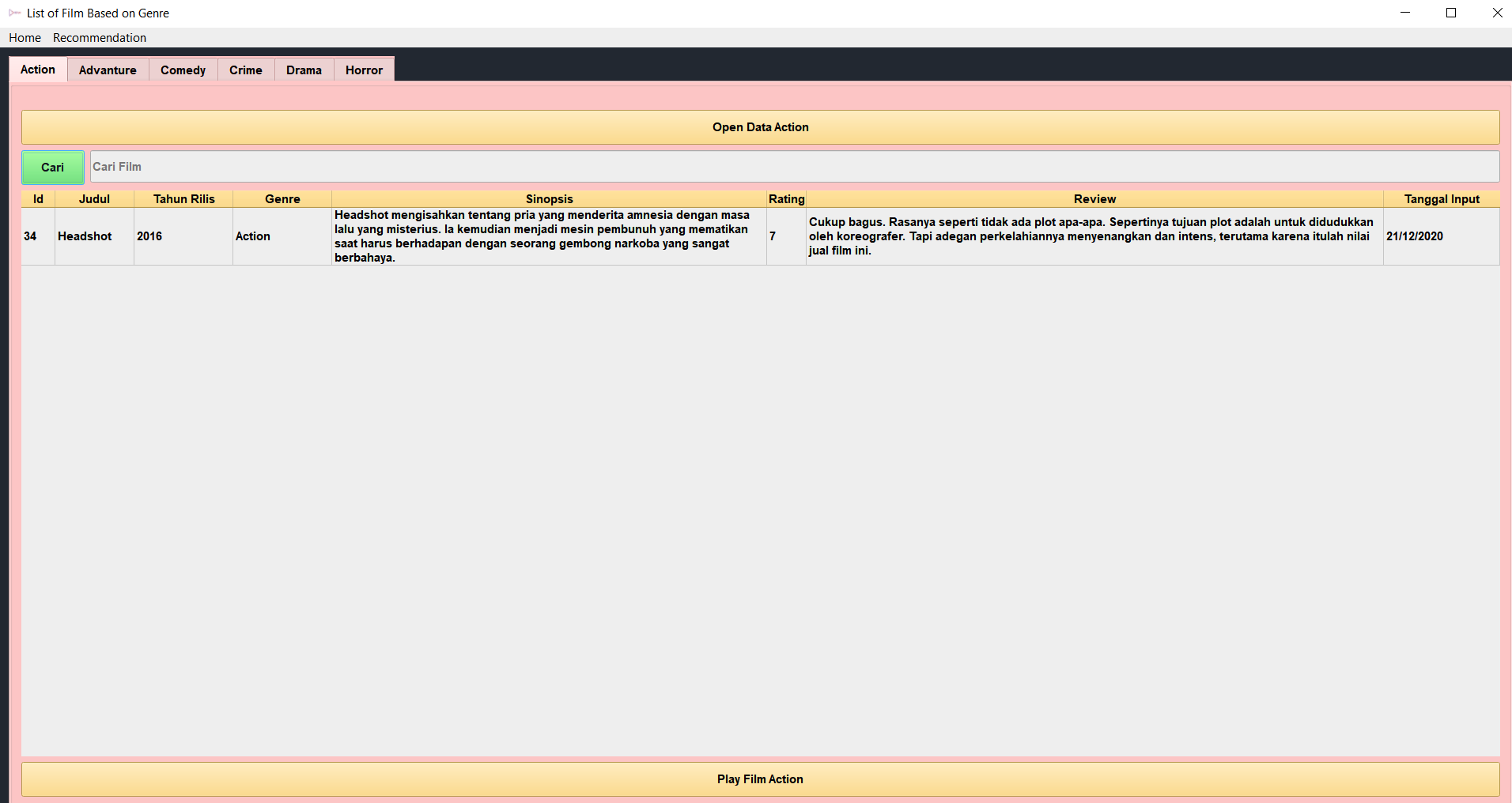
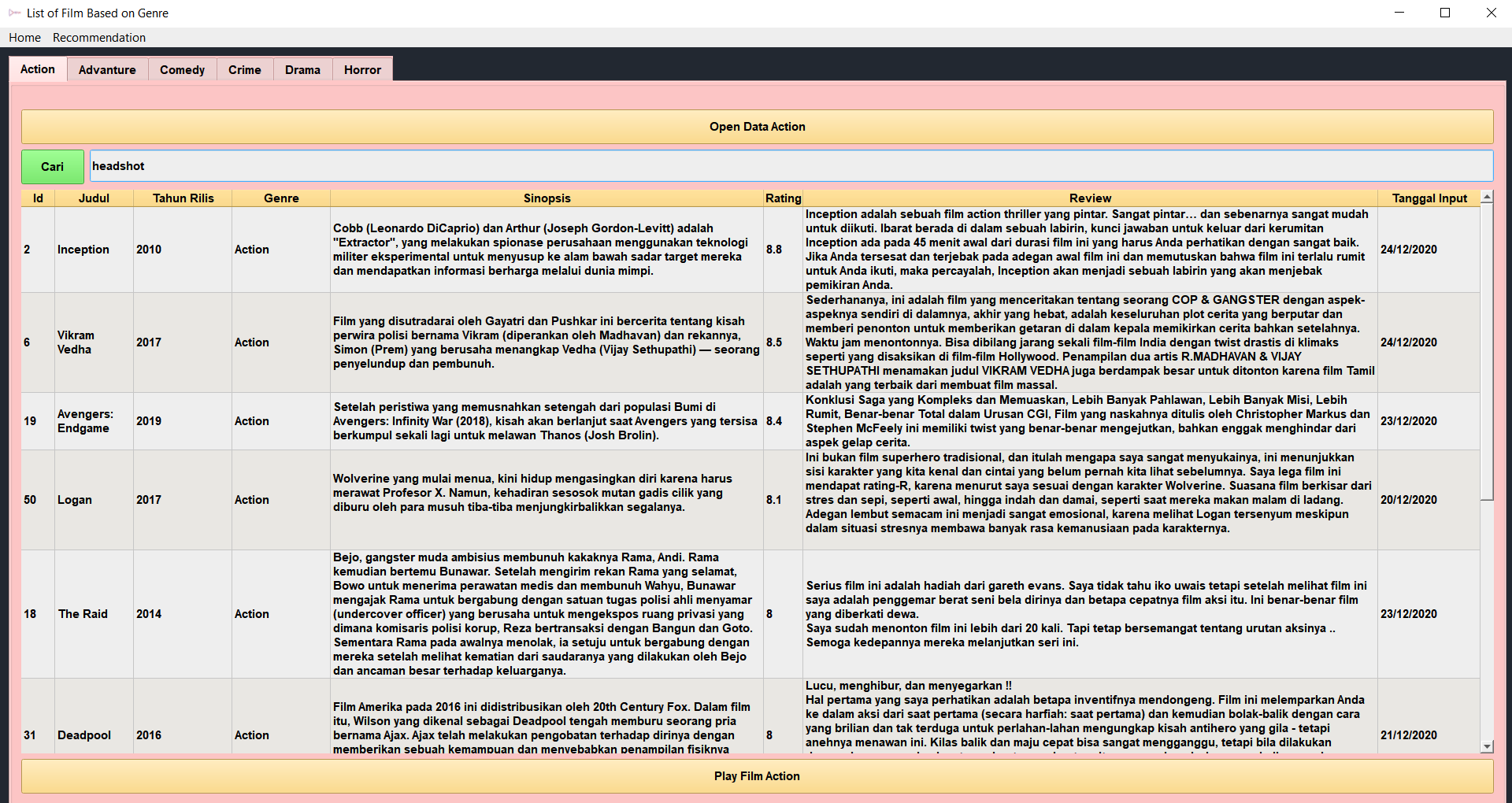






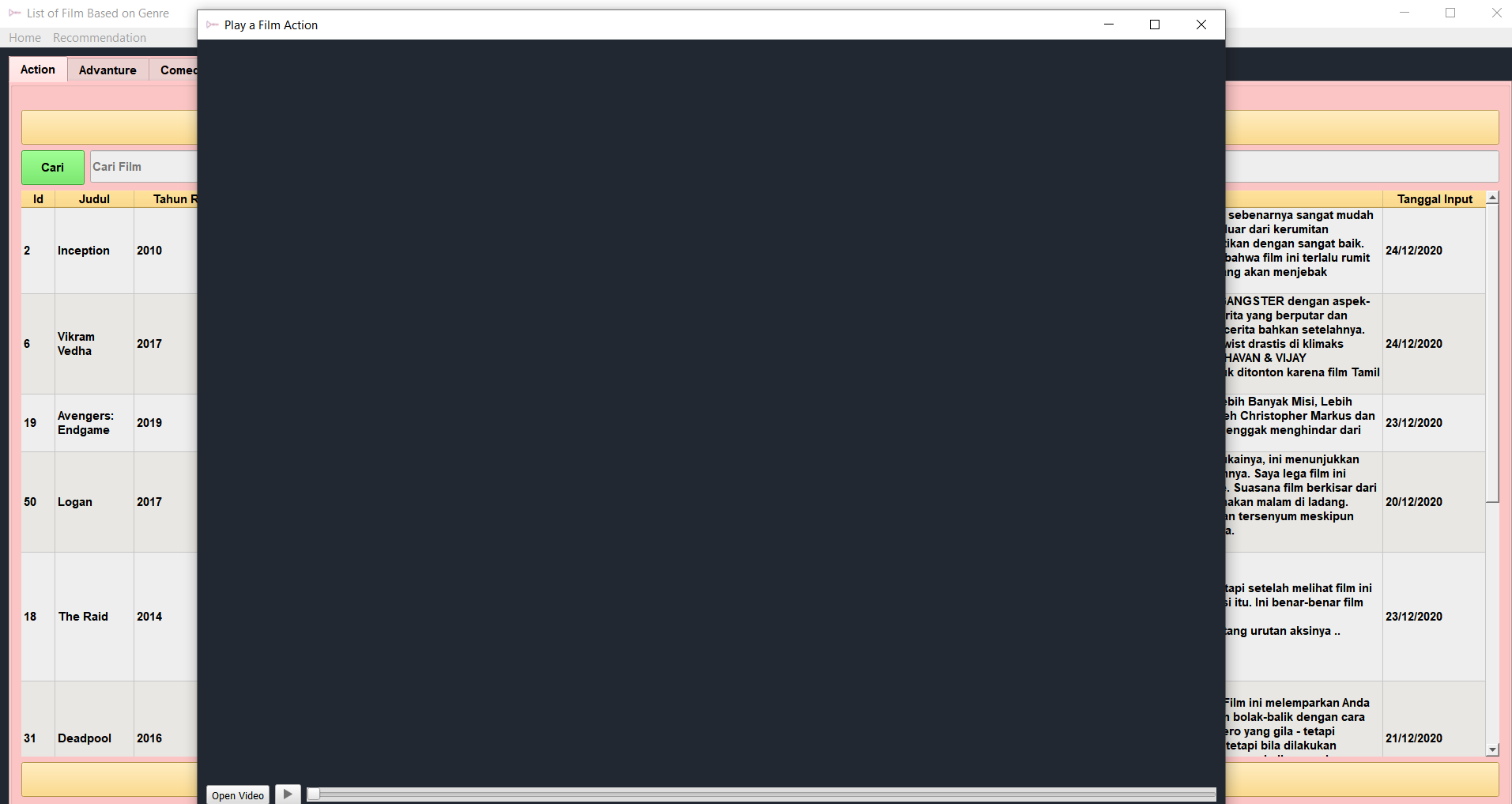
* Mencari Data Film

Pada masing – masing data film tersbut user dapat mencari film yang diingikan dengan cara memaukkan kata kunci pada line edit dan kemudian menekan tombol Cari, kata kunci tersebut akan dicocokkan ke database admin, apabila ada data film yang sesuai, maka akan ditampilkan ke dalam table widget

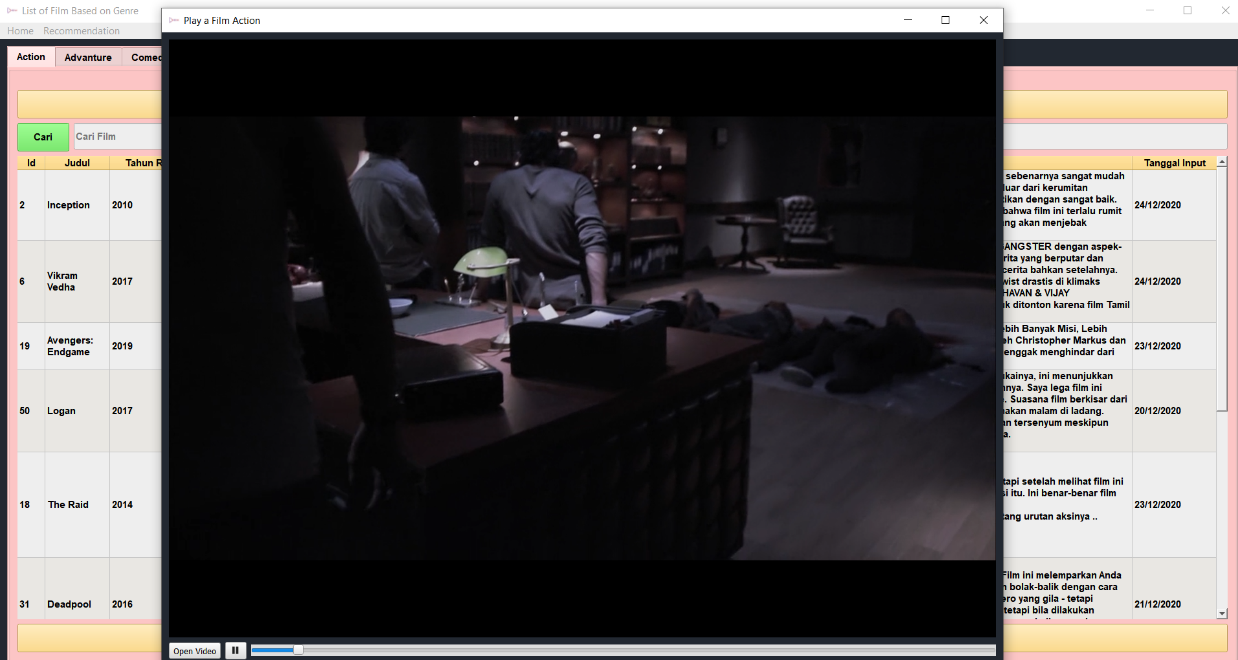


* Menonton Film yang Tersedia

User dapat menonton film yang sudah disediakan oleh admin, dengan cara menekan button Play Film pada masing – masing genre kemudian memilih film yang ingin ditonton



Untuk menjalankan film, tekan tombol icon play, maka film akan berjalan sesuai deurasi film



1. **EVALUASI OLEH REKAN**

OlehMuhamad Zulvan Fahmi dengan NIM 190411100143

Aplikasi replay adalah aplikasi ringkasan review dan pemutar film, data yang disajikan cukup lengkap, sehingga membuat user dapat menentukan saran film yang ingin diputar. Ketika di jalankan aplikasi tidak mengalami error. Sebelum menggunakan aplikasi pengguna diharuskan login terlebih dahulu, hal ini membuat keamanan aplikasi terjamin. Setelah login akan masuk halaman home berisi pesan peringatan untuk pengguna, secara tampilan aplikasi sudah sangat menarik, warna background, warna huruf dan logo sangat serasi, penataan letak widget-nya juga sangat rapi, sedikit kekurangan pada halaman home, menurut saya harus ada tombol untuk menuju halaman list film, sebagai pengguna awal saya agak bingung mencari tombolnya yang ternyata menggunakan menu di kiri atas.

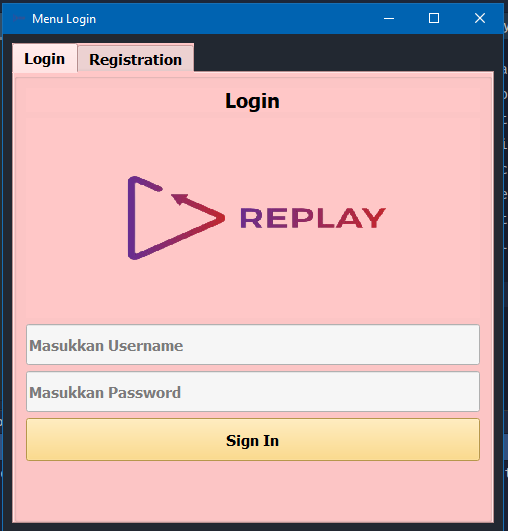
Halaman list film memperlihatkan review film dan beberapa film memiliki video yang dapat diputar. Tampilan halaman sangat menarik dan tertata rapi. Data yang disajikan untuk review film sudah lengkap, juga terdapat fitur pencarian yang mempercepat mencari judul film. Saran saya sebaiknya begitu masuk halaman list film, secara otomatis terhubung dengan database, sehingga review film bisa langsung ditampilkan tanpa perlu klik tombol open data. Untuk bagian kolom tanggal, ada sedikit bagian yang terpotong jadi tidak menampilkan semua tulisan tanggal. Bagian table view menurut saya harusnya tidak bisa diedit, apabila kita double-click pada data di tableview, kita bisa mengedit data, menurut saya hal ini tidak perlu. Saran saya menambah fitur mengurutkan rating sehingga kita dapat mengetahui dengan cepat film dengan rating bagus dan jelek.

Halaman admin berguna untuk menambah film baru ke dalam database, form yang ditampilkan sudah lengkap, fitur yang ada juga sudah lengkap seperti tambah data, update data, delete data dan cari data, tetapi sama seperti sebelumnya perlu klik tombol open database untuk terhubung ke database. Tampilan sudah cukup bagus tapi mungkin tableview bisa ditaruh di tab lain karena terlalu kecil. Saran ditambah message box saat berhasil menambah film.

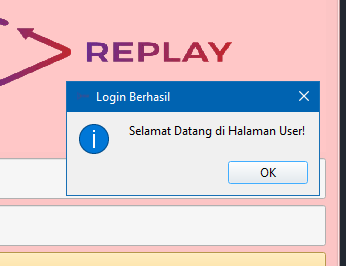
Secara keseluruhan aplikasi sudah memenuhi fungsi dan kegunaannya. Halaman home dan list film memiliki desain yang bagus dan serasi, semua tombol menggunakan warna dan ada hover yang menambah bagus tampilan aplikasi. Data yang disajikan sangat lengkap. Serta, respon aplikasi dalam memutar film juga cukup cepat tanpa ada lag.

1. **KOMENTAR EVALUATOR**

OlehMuhamad Zulvan Fahmi dengan NIM 190411100143

****

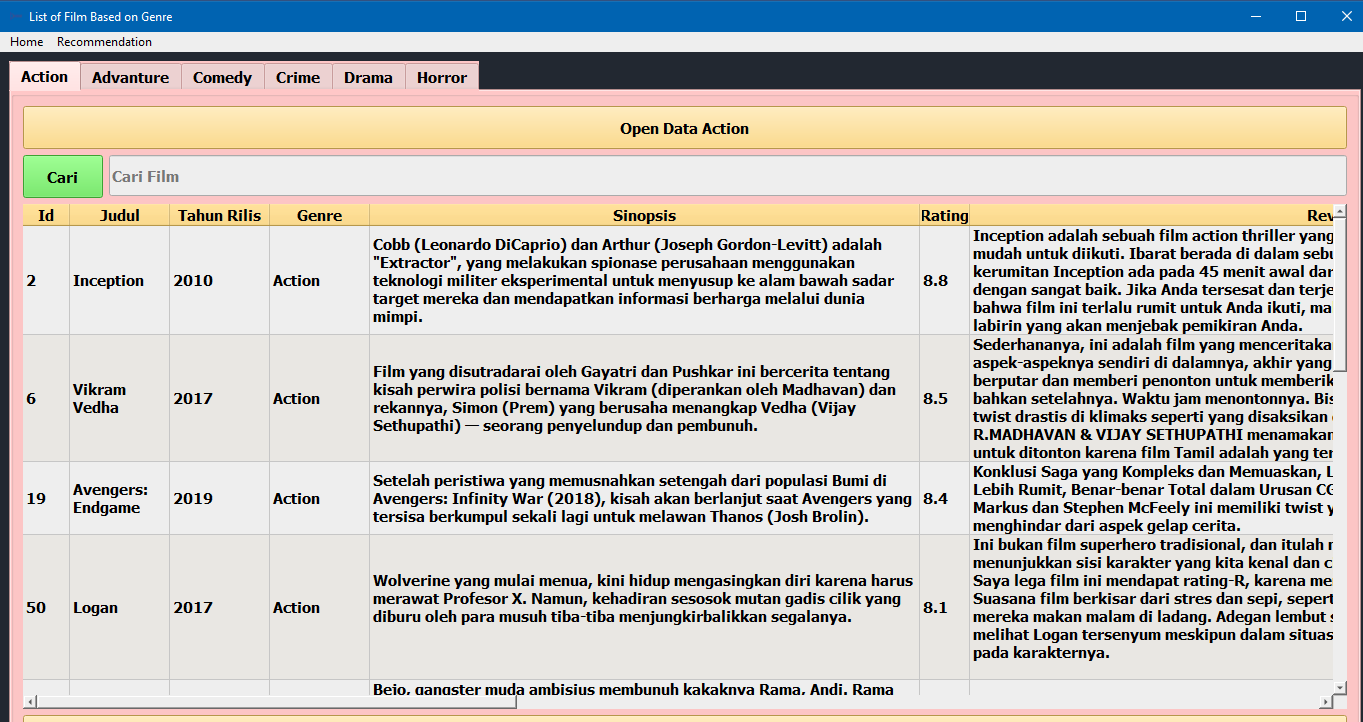
Aplikasi memiliki fitur login yang memperkuat keamanan, tampilan halaman terlihat serasi

****

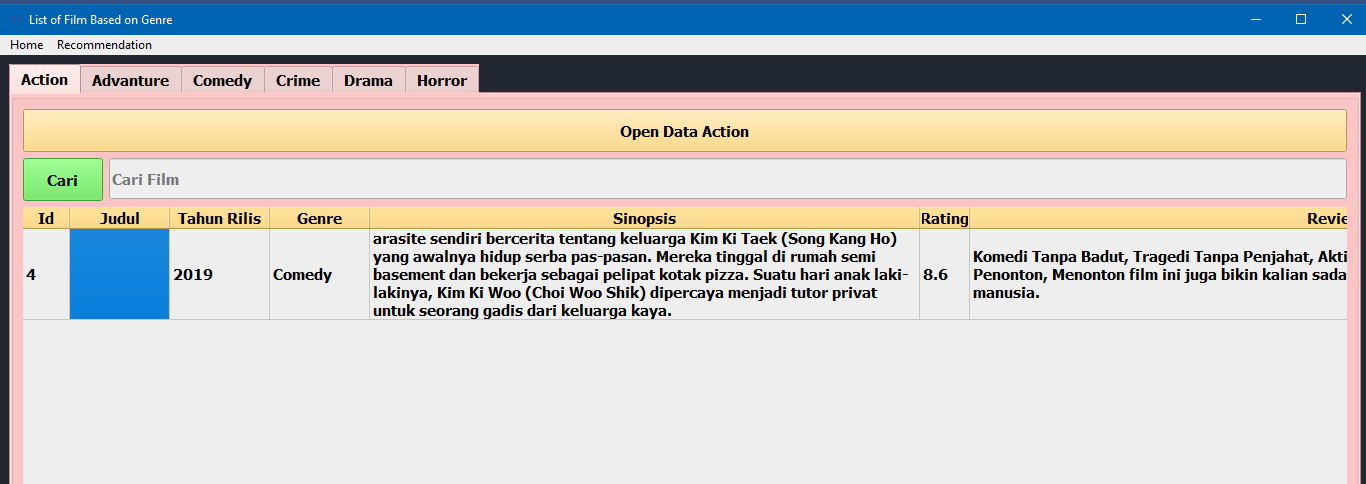
Terdapat banyak message box yang membuat adanya interaksi antara aplikasi dan pengguna

****

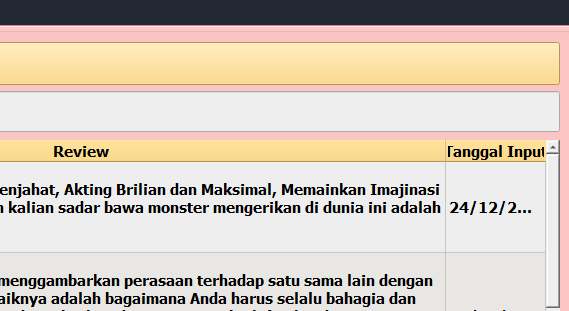
Tampilan home, pemiliha warna dan tata letak sangat bagus, saran menambah tombol ke list daftar film

****

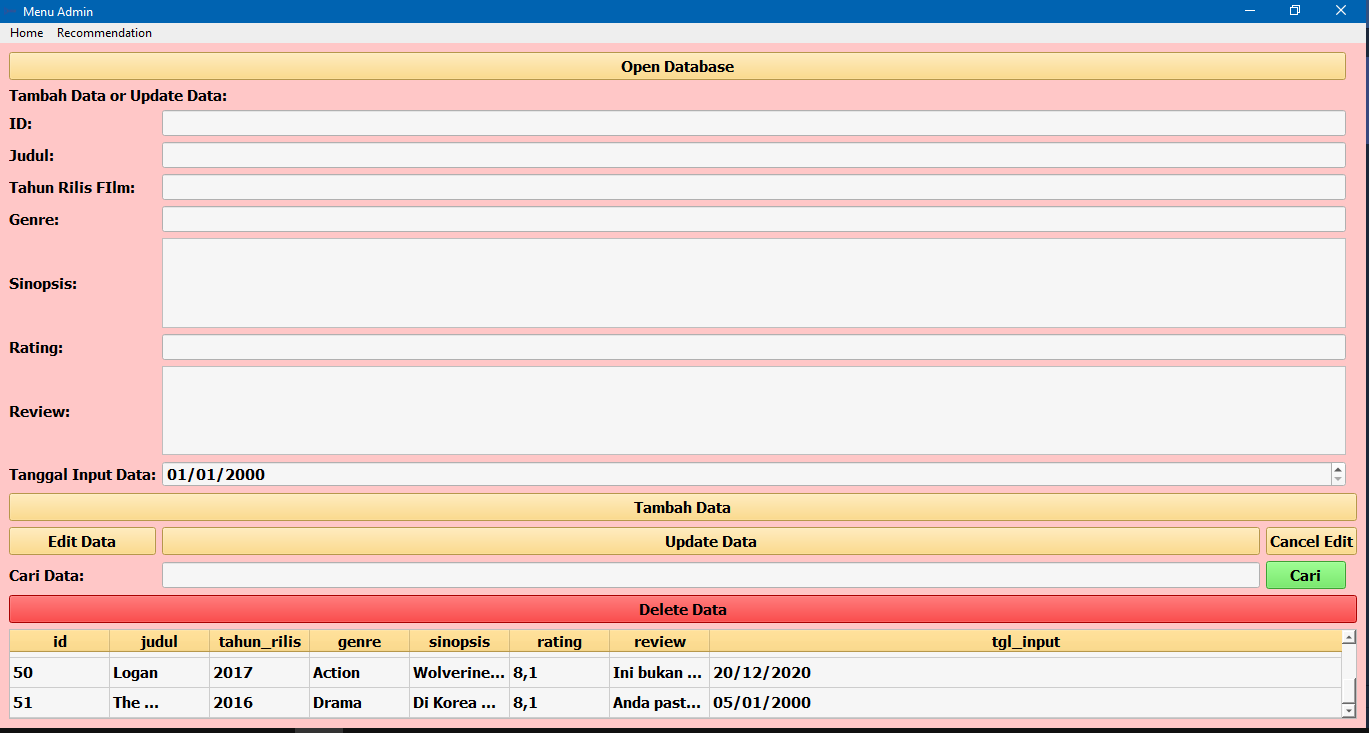
Tampilan list film menyajikan data yang lengkap berdasarkan genre

****

Isi dari data seharusnya tidak dapat diedit

****

Tulisan tanggal input terpotong, jadi tidak dapat melihat tanggal dengan lengkap



Halaman admin sudah lengkap untuk form dan fiturnya, hanya saja table view terlalu kecil

* **Saran**

1. Memperbaiki beberapa bug
2. Menambah tombol di halaman home, tombol untuk menuju ke halaman list
3. Menambah video dan data film baru
4. Saat masuk halaman list film dan halaman admin langsung terhubung pada database, tanpa perlu klik tombol open data
5. Menambah fitur urutkan berdasarkan rating
6. Menambah message box saat berhasil menambah film pada halaman admin
7. **LISTING CODE PROGRAM**
8. Login.py

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

from PyQt5 import \*

from PyQt5.QtCore import \*

from PyQt5.QtMultimedia import \*

from PyQt5.QtMultimediaWidgets import \*

import sqlite3

import sqlite3 as mdb

import playFilmAction, playFilmAdvanture, playFilmComedy, playFilmCrime, playFilmDrama

class Login(QWidget):

##################################### -LOGIN- #####################################

#membuat fungsi dengan nama Loginwindow

def Loginwindow(self, Mwindow):

#membuat QWidget dengan nama variabel window

window = QWidget()

grid = QGridLayout()

tab = QTabWidget(self)

grid.addWidget(tab, 0,0)

group1 = QGroupBox(self)

group2 = QGroupBox(self)

#membuat sebuah form layout dengan nama variabel form dan form2

form = QFormLayout()

form2 = QFormLayout()

group1.setLayout(form)

group2.setLayout(form2)

tab.addTab(group1,"Login")

tab.addTab(group2,"Registration")

tab.setStyleSheet("background-color: #ffc7c7; font:bold 15px roboto;")

label = QLabel("Registration",self)

label.setAlignment(QtCore.Qt.AlignCenter)

form2.addRow(label)

label.setStyleSheet("font:bold 20px;")

gambar = QPixmap("replay\_ok.png")

gambar\_ok = gambar.scaled(350, 200)

label0 = QLabel(self)

label0.setPixmap(gambar\_ok)

label0.setAlignment(QtCore.Qt.AlignCenter)

label0.setStyleSheet("height: 10px;")

form2.addRow(label0)

self.usernamereg = QLineEdit()

self.usernamereg.setPlaceholderText("Masukkan Username")

form2.addRow(self.usernamereg)

self.usernamereg.setStyleSheet("background-color: #f6f6f6; height: 35px; font:bold 15px;")

self.passwordreg = QLineEdit()

self.passwordreg.setEchoMode(QLineEdit.Password)

self.passwordreg.setPlaceholderText("Masukkan Password")

form2.addRow(self.passwordreg)

self.passwordreg.setStyleSheet("background-color: #f6f6f6; height: 35px; font:bold 15px;")

self.passwordreg2 = QLineEdit()

self.passwordreg2.setEchoMode(QLineEdit.Password)

self.passwordreg2.setPlaceholderText("Konfirmasi Password")

form2.addRow(self.passwordreg2)

self.passwordreg2.setStyleSheet("background-color: #f6f6f6; height: 35px; font:bold 15px;")

signup = QPushButton("Sign Up")

form2.addRow(signup)

signup.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

#Ketika button di klik akan memanggi fungsi masing - masing

signup.clicked.connect(self.regis\_user)

label = QLabel("Login",self)

label.setAlignment(QtCore.Qt.AlignCenter)

form.addRow(label)

label.setStyleSheet("font:bold 20px;")

gambar = QPixmap("replay\_ok.png")

gambar\_ok = gambar.scaled(350, 200)

label0 = QLabel(self)

label0.setPixmap(gambar\_ok)

label0.setAlignment(QtCore.Qt.AlignCenter)

label0.setStyleSheet("height: 10px;")

form.addRow(label0)

self.usernameln = QLineEdit()

self.usernameln.setPlaceholderText("Masukkan Username")

form.addRow(self.usernameln)

self.usernameln.setStyleSheet("background-color: #f6f6f6; height: 35px; font:bold 15px;")

self.passwordln = QLineEdit()

self.passwordln.setEchoMode(QLineEdit.Password)

self.passwordln.setPlaceholderText("Masukkan Password")

form.addRow(self.passwordln)

self.passwordln.setStyleSheet("background-color: #f6f6f6; height: 35px; font:bold 15px;")

signin = QPushButton("Sign In")

form.addRow(signin)

signin.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

signin.clicked.connect(self.cek\_user)

#membuat layout grid menjadi layout kedua yang berada di dalam main window

window.setLayout(grid)

#menentukan ukuran Main window, + title dan menampilkan

Mwindow.setCentralWidget(window)

Mwindow.setStyleSheet("background-color: #222831;")

Mwindow.setGeometry(700,250,500,300)

Mwindow.setWindowTitle("Menu Login")

def OpenDatabase(self):

#Mendeklarasikan database

db = QtSql.QSqlDatabase.addDatabase('QSQLITE')

#Membuat nama database

db.setDatabaseName('test.db')

#Mengecek Database Apakah sudah terkoneksi atau belum

if db.open():

print('Berhasil membuka Database')

else:

print('Gagal membuka Database!')

def cek\_user(self):

self.OpenDatabase()

notif = QMessageBox()

usernamelog = str(self.usernameln.text())

pass1log = str(self.passwordln.text())

#Mendefinisikan Query

query = QtSql.QSqlQuery()

connection = sqlite3.connect("test.db")

cek = connection.execute("SELECT username,password FROM users WHERE username = ? AND password = ? ",(usernamelog,pass1log))

if usernamelog == "admin" and pass1log == "123":

notif.information(self, "Login Berhasil", "Selamat Datang di Halaman Admin! ")

self.open\_admin()

elif (len(cek.fetchall()) > 0):

notif.information(self, "Login Berhasil", "Selamat Datang di Halaman User! ")

self.open\_user()

else:

notif.warning(self, "Login Gagal", "Username Dan Password Anda Tidak Cocok!", notif.Ok)

def createTabelUsers(self):

query = QtSql.QSqlQuery()

query.exec\_("create table users(" "username varchar(20) primary key, "

"password varchar(20))")

print ("True")

def regis\_user(self):

self.OpenDatabase()

self.createTabelUsers()

notif = QMessageBox()

username = str(self.usernamereg.text())

pass1 = str(self.passwordreg.text())

pass2 = str(self.passwordreg2.text())

#Mendefinisikan Query

query = QtSql.QSqlQuery()

if pass1 != pass2:

notif.warning(self, "Konfirmasi Password Salah", "Tolong Masukkan password dengan benar! ", notif.Ok)

else:

#Menjalankan Perintah Sql

query.prepare("INSERT INTO users VALUES ('" + username + "', '" + pass1 + "')")

query.exec\_()

notif.information(self, "Selamat!", "Anda Berhasil Membuat Akun!")

self.usernamereg.setText("")

self.passwordreg.setText("")

self.passwordreg2.setText("")

#membuat fungsi open\_user

def open\_user(self):

#memanggil/menampilkan aplikasi window\_go

self.window\_go(Mwindow)

#membuat fungsi open\_admin

def open\_admin(self):

self.Admin(Mwindow)

##################################### -END OF LOGIN- #####################################

##################################### -USER- #####################################

def window\_go(self, Mwindow):

#membuat QWidget dengan nama variabel window

window = QWidget()

#menginisialisai sebuah menu bar di dalam main window sengan nama variabel baruser

self.baruser = Mwindow.menuBar()

self.baruser.setStyleSheet("background-color: #eeeeee;")

self.file0user = self.baruser.addMenu("Home")

#menambahkan menu 'Recommendation' ke dalam bar dan di inisialisasi ke variabel file1user

self.file1user = self.baruser.addMenu("Recommendation")

#menambahkan QAction "Close Application" di dalam menubar home dengan inisialisasi ke variabel file0user

self.file0auser = QAction("Close Application")

#menambahkan QAction "List of Reccomendation" di dalam menubar Recommendation dengan inisialisasi ke variabel file1auser

self.file1auser = QAction("List of Recommendation")

#memasukkan file0auser ke dalam menu bar file0user

self.file0user.addAction(self.file0auser)

#memasukkan file1user ke dalam menu bar file1auser

self.file1user.addAction(self.file1auser)

self.file0auser.triggered.connect(self.open\_home)

self.file1auser.triggered.connect(self.open\_recom)

#membuat sebuah Vbox layout dengan nama variabel vbox

vbox = QVBoxLayout()

gambar = QPixmap("replay\_ok.png")

gambar\_ok = gambar.scaled(500, 300)

label0 = QLabel(self)

label0.setPixmap(gambar\_ok)

label0.setAlignment(QtCore.Qt.AlignCenter)

label0.setStyleSheet("height: 10px;")

vbox.addWidget(label0)

#membuat sebuah label dengan dengan nama variabel label

label = QLabel("Intructions For The Users! \n - Tidak semua list film yang ada di table terdapat videonya "

"\n - Film yang disediakan berdsarkan ID kemudian diikuti judul dan tahun rilis film"

"\n - Untuk menonton film pilih dulu film dengan menekan tombol open video pada layout play film"

"\n - Dimohon untuk pause video sebelum menutup layout play film",window)

#mengatur posisi label menjadi align center

label.setAlignment(QtCore.Qt.AlignCenter)

#mengatur label dengan warna, font bold dan berukuran 25px

label.setStyleSheet("color: #0a043c; font: bold 25px;")

#memasukkan label ke dalam layout vbox

vbox.addWidget(label)

#membuat layout vbox menjadi layout kedua yang berada di dalam main window

window.setLayout(vbox)

window.setStyleSheet("background-color: #ffc7c7; font:bold 15px roboto; border: 2px solid black; ")

#menentukan ukuran main window, + title dan menampilkan

Mwindow.setCentralWidget(window)

Mwindow.setGeometry(350,250,1300,300)

Mwindow.setWindowTitle("Menu Home")

#membuat fungsi open\_home

def open\_home(self):

#Menutup Main Window

Mwindow.close()

#membuat fungsi open\_recom

def open\_recom(self):

#memanggil/menampilkan aplikasi LayoutRecommendation

self.LayoutRecommendation(Mwindow)

##################################### -END OF USER- #####################################

##################################### -ADMIN- #####################################

def Admin(self, Mwindow):

#menginisialisai sebuah menu bar di dalam main window sengan nama variabel barAdmin

self.barAdmin = Mwindow.menuBar()

self.barAdmin.setStyleSheet("background-color: #eeeeee;")

self.file0 = self.barAdmin.addMenu("Home")

self.file0a = QAction("Close Application")

self.file1 = self.barAdmin.addMenu("Recommendation")

self.file1a = QAction("List of Recommendation")

self.file1.addAction(self.file1a)

self.file0.addAction(self.file0a)

self.file0a.triggered.connect(self.open\_home)

self.file1a.triggered.connect(self.open\_recom)

window = QWidget()

#Membuat Grid Layout

grid = QGridLayout()

window.setStyleSheet("background-color: #ffc7c7; font:bold 15px roboto;")

open\_database = QPushButton("Open Database")

grid.addWidget(open\_database,1,0,1,3)

open\_database.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 20px;")

add\_data = QLabel("Tambah Data or Update Data:")

grid.addWidget(add\_data,2,0,1,0)

add\_id = QLabel("ID:")

grid.addWidget(add\_id,3,0)

self.Id = QLineEdit(self)

grid.addWidget(self.Id,3,1,1,2)

self.Id.setStyleSheet("background-color: #f6f6f6;")

add\_judul = QLabel("Judul:")

grid.addWidget(add\_judul,4,0)

self.Judul = QLineEdit(self)

grid.addWidget(self.Judul,4,1,1,2)

self.Judul.setStyleSheet("background-color: #f6f6f6;")

tahun\_rilis = QLabel("Tahun Rilis FIlm:")

grid.addWidget(tahun\_rilis,5,0)

self.Tahun = QLineEdit(self)

grid.addWidget(self.Tahun,5,1,1,2)

self.Tahun.setStyleSheet("background-color: #f6f6f6;")

add\_genre = QLabel("Genre:")

grid.addWidget(add\_genre,6,0)

self.Genre = QLineEdit(self)

grid.addWidget(self.Genre,6,1,1,2)

self.Genre.setStyleSheet("background-color: #f6f6f6;")

add\_sinopsis = QLabel("Sinopsis:")

grid.addWidget(add\_sinopsis,7,0)

self.Sinopsis = QTextEdit(self)

grid.addWidget(self.Sinopsis,7,1,1,2)

self.Sinopsis.setStyleSheet("background-color: #f6f6f6;")

add\_rating = QLabel("Rating:")

grid.addWidget(add\_rating,8,0)

self.Rating = QLineEdit(self)

grid.addWidget(self.Rating,8,1,1,2)

self.Rating.setStyleSheet("background-color: #f6f6f6;")

add\_review = QLabel("Review:")

grid.addWidget(add\_review,9,0)

self.Review = QTextEdit(self)

grid.addWidget(self.Review,9,1,1,2)

self.Review.setStyleSheet("background-color: #f6f6f6;")

add\_tgl = QLabel("Tanggal Input Data:")

grid.addWidget(add\_tgl,10,0)

self.tgl\_input = QDateEdit(self)

self.tgl\_input.setMaximumDate(QtCore.QDate(2020,12,31))

grid.addWidget(self.tgl\_input,10,1,1,2)

self.tgl\_input.setStyleSheet("background-color: #f6f6f6;")

add\_button = QPushButton("Tambah Data")

grid.addWidget(add\_button,11,0,1,0)

add\_button.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 20px;")

add\_edit = QPushButton("Edit Data")

grid.addWidget(add\_edit,12,0)

add\_edit.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 20px;")

add\_update = QPushButton("Update Data")

grid.addWidget(add\_update,12,1,1,1)

add\_update.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 20px;")

add\_cancel = QPushButton("Cancel Edit")

grid.addWidget(add\_cancel,12,2,1,2)

add\_cancel.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 20px;")

search = QLabel("Cari Data:")

grid.addWidget(search,13,0)

self.Cari = QLineEdit(self)

grid.addWidget(self.Cari,13,1)

self.Cari.setStyleSheet("background-color: #f6f6f6;")

search\_button = QPushButton("Cari")

grid.addWidget(search\_button,13,2)

search\_button.setStyleSheet("background-color: #54e346; font:bold 15px; height: 20px;")

add\_delete = QPushButton("Delete Data")

grid.addWidget(add\_delete,14,0,1,0)

add\_delete.setStyleSheet("background-color: #ec0101; font:bold 15px; height: 20px;")

#Membuat widget table view yang diberi nama "Data" dan akan dimasukkan ke dalam layout Grid

self.tableview = QTableView(self)

self.tableview.setObjectName("Data")

self.tableview.horizontalHeader().setStretchLastSection(True)

self.tableview.setWordWrap(True)

self.tableview.verticalHeader().setVisible(False)

grid.addWidget(self.tableview,15,0,1,0)

self.tableview.setStyleSheet("background-color: #f6f6f6;")

self.tableview.horizontalHeader().setStyleSheet("background-color: #ffd369; font:bold 15px;")

#Ketika button di klik akan memanggi fungsi masing - masing

open\_database.clicked.connect(self.OpenDatabaseAdmin)

add\_button.clicked.connect(self.tambahData)

search\_button.clicked.connect(self.filterData)

add\_edit.clicked.connect(self.editData)

add\_update.clicked.connect(self.updateData)

add\_cancel.clicked.connect(self.cancelData)

add\_delete.clicked.connect(self.hapusData)

#Layout grid di jadikan layout utama

window.setLayout(grid)

#menentukan ukuran main window, + title dan menampilkan

Mwindow.setCentralWidget(window)

Mwindow.setStyleSheet("background-color: #222831;")

Mwindow.setGeometry(250,40,1300,980)

Mwindow.setWindowTitle("Menu Admin")

def OpenDatabaseAdmin(self):

#Mendeklarasikan database

self.db = QtSql.QSqlDatabase.addDatabase('QSQLITE')

#Membuat nama database

self.db.setDatabaseName('test.db')

#Mengecek Database Apakah sudah terkoneksi atau belum

if self.db.open():

print('Berhasil membuka Database')

self.createTabel()

self.tampilDataListfilm()

else:

print('Gagal membuka Database!')

def createTabel(self):

query = QtSql.QSqlQuery()

query.exec\_("create table listfilm(" "id integer primary key AUTOINCREMENT, "

"judul varchar(20), tahun\_rilis int(10), genre varchar(20), sinopsis varchar(200), rating int(10), review varchar(400), tgl\_input varchar(15))")

print ("True")

#Fungsi untuk menampilankan data di dalam table

def tampilDataListfilm(self):

#Membuat Model

model = QSqlQueryModel()

#Mendefinisikan sql

sql = "SELECT \* FROM listfilm"

#Mengeksekusi Model Query

model.setQuery(sql)

#Mengeset Data Model Ke table view

self.tableview.setModel(model)

#self.tableview.setWindowTitle(title)

return self.tableview

#Fungsi tambah Data ke dalam tabel mahasiswa

def tambahData(self):

#Mengambil Text inputan

self.OpenDatabaseAdmin()

Id = str(self.Id.text())

Judul = str(self.Judul.text())

Tahun = str(self.Tahun.text())

Genre = str(self.Genre.text())

Sinopsis = str(self.Sinopsis.toPlainText())

Review = str(self.Review.toPlainText())

Rating = str(self.Rating.text())

Tgl\_Input = str(self.tgl\_input.text())

query = QtSql.QSqlQuery()

#Menjalankan Perintah Sql

query.prepare("INSERT INTO listfilm VALUES ('" + Id + "','" + Judul + "', '" + Tahun + "', '" + Genre + "', '" + Sinopsis + "', '" + Rating + "', '" + Review + "', '" + Tgl\_Input + "')")

#Mengecek apakah query berjalan dengan baik

if query.exec\_():

self.Id.setText("")

self.Judul.setText("")

self.Tahun.setText("")

self.Genre.setText("")

self.Sinopsis.setText("")

self.Review.setText("")

self.Rating.setText("")

#Menampilkan Data

self.tampilDataListfilm()

else:

#Apabila error akan menampilkan errornya ke dalam terminal

print("Insert Error: ", query.lastError().text())

#Fungsi Filter Data

def filterData(self):

#Membuat Model

model = QSqlQueryModel()

#Mengambil Inputan filter

filter\_search = str(self.Cari.text())

sql = "SELECT \* FROM listfilm WHERE id LIKE '%"+str(filter\_search)+"%' OR judul LIKE '%"+str(filter\_search)+"%' OR tahun\_rilis LIKE '%"+str(filter\_search)+"%' OR genre LIKE '%"+str(filter\_search)+"%' OR tgl\_input LIKE '%"+str(filter\_search)+"%'"

self.Cari.setText("")

#Mengeksekusi Model Query

model.setQuery(sql)

#Mengeset Data Model Ke table view

self.tableview.setModel(model)

#mengembalikan nilai table view

return self.tableview

def hapusData(self):

# Mengambil Data Yang Di klik

select = self.tableview.selectedIndexes()[0]

Id = str(self.tableview.model().data(select))

Judul = str(self.tableview.model().data(select))

Tahun = str(self.tableview.model().data(select))

Genre = str(self.tableview.model().data(select))

Sinopsis = str(self.tableview.model().data(select))

Review = str(self.tableview.model().data(select))

Rating = str(self.tableview.model().data(select))

Tgl\_Input = str(self.tableview.model().data(select))

# Mendefinisikan Query

query = QSqlQuery()

# Menjalankan Perintah Query

query.prepare("DELETE from listfilm WHERE (id = '" + Id + "'OR judul = '" + Judul + "'OR tahun\_rilis = '" + Tahun + "'OR genre = '" + Genre + "'OR sinopsis = '" + Sinopsis + "'OR rating = '" + Rating + "'OR review = '" + Review + "'OR tgl\_input = '" + Tgl\_Input + "')")

query.exec\_()

if query.exec\_():

print("Hapus Data Success")

self.tampilDataListfilm()

else:

print("del\_records Error: ", query.lastError().text())

def editData(self):

model = QSqlQueryModel()

connection = sqlite3.connect("test.db")

# Mengambil Data Yang Di klik

select = self.tableview.selectedIndexes()[0]

Id = str(self.tableview.model().data(select))

# Mendefinisikan Query

# Menjalankan Perintah Query

# Eksekusi Perintah SQL

waw = connection.execute("SELECT \* FROM listfilm WHERE id = '%s' " % (Id))

# Mengambil Satu Data

data = waw.fetchall()

self.Id.setText(str(data[0][0]))

self.Judul.setText(str(data[0][1]))

self.Tahun.setText(str(data[0][2]))

self.Genre.setText(str(data[0][3]))

self.Sinopsis.setText(str(data[0][4]))

self.Rating.setText(str(data[0][5]))

self.Review.setText(str(data[0][6]))

def updateData(self):

# Mengambil Data Yang Di input

Id = str(self.Id.text())

Judul = str(self.Judul.text())

Tahun = str(self.Tahun.text())

Genre = str(self.Genre.text())

Sinopsis = str(self.Sinopsis.toPlainText())

Review = str(self.Review.toPlainText())

Rating = str(self.Rating.text())

Tgl\_Input = str(self.tgl\_input.text())

#Mendefinisikan Query

query = QtSql.QSqlQuery()

#Menjalankan Perintah Sql

query.prepare("""UPDATE listfilm SET judul=:Judul, tahun\_rilis=:Tahun, genre=:Genre, sinopsis=:Sinopsis, rating=:Rating, review=:Review, tgl\_input=:Tgl\_Input WHERE id =:Id""")

query.bindValue(":Id", Id)

query.bindValue(":Judul", Judul)

query.bindValue(":Tahun", Tahun)

query.bindValue(":Genre", Genre)

query.bindValue(":Sinopsis", Sinopsis)

query.bindValue(":Review", Review)

query.bindValue(":Rating", Rating)

query.bindValue(":Tgl\_Input", Tgl\_Input)

if query.exec\_():

self.Id.setText("")

self.Judul.setText("")

self.Tahun.setText("")

self.Genre.setText("")

self.Sinopsis.setText("")

self.Review.setText("")

self.Rating.setText("")

#Menampilkan Data

self.tampilDataListfilm()

else:

#Apabila error akan menampilkan errornya ke dalam terminal

print("Insert Error: ", query.lastError().text())

def cancelData(self):

self.Id.setText("")

self.Judul.setText("")

self.Tahun.setText("")

self.Genre.setText("")

self.Sinopsis.setText("")

self.Review.setText("")

self.Rating.setText("")

##################################### -END OF ADMIN- #####################################

##################################### -LIST OF FILM BASED ON GENRE- #####################################

def LayoutRecommendation(self, Mwindow):

windowx = QWidget(self)

gridx = QGridLayout()

tabx = QTabWidget(self)

tabx.setStyleSheet("background-color: #ffc7c7; font:bold 15px roboto;")

gridx.addWidget(tabx, 0,0)

groupAct = QGroupBox(self)

groupAdv = QGroupBox(self)

groupCom = QGroupBox(self)

groupCri = QGroupBox(self)

groupDra = QGroupBox(self)

groupHor = QGroupBox(self)

#membuat sebuah grid layout dengan nama variabel grid

form = QFormLayout()

formAdvanture = QFormLayout()

formComedy = QFormLayout()

formCrime = QFormLayout()

formDrama = QFormLayout()

formHorror = QFormLayout()

groupAct.setLayout(form)

groupAdv.setLayout(formAdvanture)

groupCom.setLayout(formComedy)

groupCri.setLayout(formCrime)

groupDra.setLayout(formDrama)

groupHor.setLayout(formHorror)

tabx.addTab(groupAct,"Action")

tabx.addTab(groupAdv,"Advanture")

tabx.addTab(groupCom,"Comedy")

tabx.addTab(groupCri,"Crime")

tabx.addTab(groupDra,"Drama")

tabx.addTab(groupHor,"Horror")

##################################### - ACTION - #####################################

btn = QPushButton("Open Data Action")

btn.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

#membuat sebuah main window dengan nama variabel Mwindow

#membuat QWidget dengan nama variabel window

Mwindow.setStyleSheet("background-color: #222831;")

self.add\_cari = QLineEdit(self)

self.add\_cari.setStyleSheet("background-color: #eeeeee; font:bold 15px; height: 35px;")

self.add\_cari.setPlaceholderText("Cari Film")

btn\_cari = QPushButton("Cari")

btn\_cari.setStyleSheet("background-color: #54e346; font:bold 15px; height: 35px;")

self.tableWidget = QTableWidget()

self.tableWidget.setAlternatingRowColors(True)

self.tableWidget.setColumnCount(8)

self.tableWidget.horizontalHeader().setCascadingSectionResizes(False)

self.tableWidget.horizontalHeader().setSortIndicatorShown(False)

self.tableWidget.horizontalHeader().setStretchLastSection(True)

self.tableWidget.verticalHeader().setVisible(False)

self.tableWidget.verticalHeader().setCascadingSectionResizes(False)

self.tableWidget.verticalHeader().setStretchLastSection(False)

self.tableWidget.verticalHeader().setSectionResizeMode(QHeaderView.Stretch)

self.tableWidget.setHorizontalHeaderLabels(("Id", "Judul", "Tahun Rilis", "Genre", "Sinopsis","Rating","Review","Tanggal Input"))

self.tableWidget.setColumnWidth(0,10)

self.tableWidget.setColumnWidth(1,100)

self.tableWidget.setColumnWidth(4,550)

self.tableWidget.setColumnWidth(5,50)

self.tableWidget.setColumnWidth(6,730)

self.tableWidget.resizeRowsToContents()

self.tableWidget.verticalScrollBar().maximum()

self.tableWidget.verticalScrollBar().setValue(0)

self.tableWidget.setStyleSheet("background-color: #eeeeee; border:2px #ffd369;")

self.tableWidget.horizontalHeader().setStyleSheet("background-color: #ffd369; font:bold 15px;")

self.tableWidget.verticalHeader().setSectionResizeMode(QtWidgets.QHeaderView.ResizeToContents)

self.tableWidget.setWordWrap(True)

btnPlayAction = QPushButton("Play Film Action")

btnPlayAction.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

form.addRow(btn)

form.addRow(btn\_cari,self.add\_cari)

form.addRow(self.tableWidget)

form.addRow(btnPlayAction)

btn\_cari.clicked.connect(self.filterDataAction)

btn.clicked.connect(self.tampilDataAction)

btnPlayAction.clicked.connect(self.PlayFilmAction)

##################################### - END OF ACTION - #####################################

##################################### - ADVANTURE - #####################################

btnAdvanture = QPushButton("Open Data Advanture")

btnAdvanture.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

#membuat sebuah main window dengan nama variabel Mwindow

self.add\_cariAdvanture = QLineEdit(self)

self.add\_cariAdvanture.setStyleSheet("background-color: #eeeeee; font:bold 15px; height: 35px;")

self.add\_cariAdvanture.setPlaceholderText("Cari Film")

btn\_cariAdvanture = QPushButton("Cari")

btn\_cariAdvanture.setStyleSheet("background-color: #54e346; font:bold 15px; height: 35px;")

self.tableWidgetAdvanture = QTableWidget()

self.tableWidgetAdvanture.setAlternatingRowColors(True)

self.tableWidgetAdvanture.setColumnCount(8)

self.tableWidgetAdvanture.horizontalHeader().setCascadingSectionResizes(False)

self.tableWidgetAdvanture.horizontalHeader().setSortIndicatorShown(False)

self.tableWidgetAdvanture.horizontalHeader().setStretchLastSection(True)

self.tableWidgetAdvanture.verticalHeader().setVisible(False)

self.tableWidgetAdvanture.verticalHeader().setCascadingSectionResizes(False)

self.tableWidgetAdvanture.verticalHeader().setStretchLastSection(False)

self.tableWidgetAdvanture.verticalHeader().setSectionResizeMode(QHeaderView.Stretch)

self.tableWidgetAdvanture.setHorizontalHeaderLabels(("Id", "Judul", "Tahun Rilis", "Genre", "Sinopsis","Rating","Review","Tanggal Input"))

self.tableWidgetAdvanture.setColumnWidth(0,10)

self.tableWidgetAdvanture.setColumnWidth(1,100)

self.tableWidgetAdvanture.setColumnWidth(4,550)

self.tableWidgetAdvanture.setColumnWidth(5,50)

self.tableWidgetAdvanture.setColumnWidth(6,730)

self.tableWidgetAdvanture.resizeRowsToContents()

self.tableWidgetAdvanture.verticalScrollBar().maximum()

self.tableWidgetAdvanture.verticalScrollBar().setValue(0)

self.tableWidgetAdvanture.setStyleSheet("background-color: #eeeeee; border:2px #ffd369;")

self.tableWidgetAdvanture.horizontalHeader().setStyleSheet("background-color: #ffd369; font:bold 15px;")

self.tableWidgetAdvanture.verticalHeader().setSectionResizeMode(QtWidgets.QHeaderView.ResizeToContents)

self.tableWidgetAdvanture.setWordWrap(True)

btnPlayAdvanture = QPushButton("Play Film Advanture")

btnPlayAdvanture.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

formAdvanture.addRow(btnAdvanture)

formAdvanture.addRow(btn\_cariAdvanture,self.add\_cariAdvanture)

formAdvanture.addRow(self.tableWidgetAdvanture)

formAdvanture.addRow(btnPlayAdvanture)

btn\_cariAdvanture.clicked.connect(self.filterDataAdvanture)

btnAdvanture.clicked.connect(self.tampilDataAdvanture)

btnPlayAdvanture.clicked.connect(self.PlayFilmAdvanture)

##################################### - END OF ADVANTURE - #####################################

##################################### - COMEDY - #####################################

btnComedy = QPushButton("Open Data Comedy")

btnComedy.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

#membuat sebuah main window dengan nama variabel Mwindow

self.add\_cariComedy = QLineEdit(self)

self.add\_cariComedy.setStyleSheet("background-color: #eeeeee; font:bold 15px; height: 35px;")

self.add\_cariComedy.setPlaceholderText("Cari Film")

btn\_cariComedy = QPushButton("Cari")

btn\_cariComedy.setStyleSheet("background-color: #54e346; font:bold 15px; height: 35px;")

self.tableWidgetComedy = QTableWidget()

self.tableWidgetComedy.setAlternatingRowColors(True)

self.tableWidgetComedy.setColumnCount(8)

self.tableWidgetComedy.horizontalHeader().setCascadingSectionResizes(False)

self.tableWidgetComedy.horizontalHeader().setSortIndicatorShown(False)

self.tableWidgetComedy.horizontalHeader().setStretchLastSection(True)

self.tableWidgetComedy.verticalHeader().setVisible(False)

self.tableWidgetComedy.verticalHeader().setCascadingSectionResizes(False)

self.tableWidgetComedy.verticalHeader().setStretchLastSection(False)

self.tableWidgetComedy.verticalHeader().setSectionResizeMode(QHeaderView.Stretch)

self.tableWidgetComedy.setHorizontalHeaderLabels(("Id", "Judul", "Tahun Rilis", "Genre", "Sinopsis","Rating","Review","Tanggal Input"))

self.tableWidgetComedy.setColumnWidth(0,10)

self.tableWidgetComedy.setColumnWidth(1,100)

self.tableWidgetComedy.setColumnWidth(4,550)

self.tableWidgetComedy.setColumnWidth(5,50)

self.tableWidgetComedy.setColumnWidth(6,730)

self.tableWidgetComedy.resizeRowsToContents()

self.tableWidgetComedy.verticalScrollBar().maximum()

self.tableWidgetComedy.verticalScrollBar().setValue(0)

self.tableWidgetComedy.setStyleSheet("background-color: #eeeeee; border:2px #ffd369;")

self.tableWidgetComedy.horizontalHeader().setStyleSheet("background-color: #ffd369; font:bold 15px;")

self.tableWidgetComedy.verticalHeader().setSectionResizeMode(QtWidgets.QHeaderView.ResizeToContents)

self.tableWidgetComedy.setWordWrap(True)

btnPlayComedy = QPushButton("Play Film Comedy")

btnPlayComedy.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

formComedy.addRow(btnComedy)

formComedy.addRow(btn\_cariComedy,self.add\_cariComedy)

formComedy.addRow(self.tableWidgetComedy)

formComedy.addRow(btnPlayComedy)

btn\_cariComedy.clicked.connect(self.filterDataComedy)

btnComedy.clicked.connect(self.tampilDataComedy)

btnPlayComedy.clicked.connect(self.PlayFilmComedy)

##################################### - END OF COMEDY - #####################################

##################################### - CRIME - #####################################

btnCrime = QPushButton("Open Data Crime")

btnCrime.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

#membuat sebuah main window dengan nama variabel Mwindow

self.add\_cariCrime = QLineEdit(self)

self.add\_cariCrime.setStyleSheet("background-color: #eeeeee; font:bold 15px; height: 35px;")

self.add\_cariCrime.setPlaceholderText("Cari Film")

btn\_cariCrime = QPushButton("Cari")

btn\_cariCrime.setStyleSheet("background-color: #54e346; font:bold 15px; height: 35px;")

self.tableWidgetCrime = QTableWidget()

self.tableWidgetCrime.setAlternatingRowColors(True)

self.tableWidgetCrime.setColumnCount(8)

self.tableWidgetCrime.horizontalHeader().setCascadingSectionResizes(False)

self.tableWidgetCrime.horizontalHeader().setSortIndicatorShown(False)

self.tableWidgetCrime.horizontalHeader().setStretchLastSection(True)

self.tableWidgetCrime.verticalHeader().setVisible(False)

self.tableWidgetCrime.verticalHeader().setCascadingSectionResizes(False)

self.tableWidgetCrime.verticalHeader().setStretchLastSection(False)

self.tableWidgetCrime.verticalHeader().setSectionResizeMode(QHeaderView.Stretch)

self.tableWidgetCrime.setHorizontalHeaderLabels(("Id", "Judul", "Tahun Rilis", "Genre", "Sinopsis","Rating","Review","Tanggal Input"))

self.tableWidgetCrime.setColumnWidth(0,10)

self.tableWidgetCrime.setColumnWidth(1,100)

self.tableWidgetCrime.setColumnWidth(4,550)

self.tableWidgetCrime.setColumnWidth(5,50)

self.tableWidgetCrime.setColumnWidth(6,730)

self.tableWidgetCrime.resizeRowsToContents()

self.tableWidgetCrime.verticalScrollBar().maximum()

self.tableWidgetCrime.verticalScrollBar().setValue(0)

self.tableWidgetCrime.setStyleSheet("background-color: #eeeeee; border:2px #ffd369;")

self.tableWidgetCrime.horizontalHeader().setStyleSheet("background-color: #ffd369; font:bold 15px;")

self.tableWidgetCrime.verticalHeader().setSectionResizeMode(QtWidgets.QHeaderView.ResizeToContents)

self.tableWidgetCrime.setWordWrap(True)

btnPlayCrime = QPushButton("Play Film Crime")

btnPlayCrime.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

formCrime.addRow(btnCrime)

formCrime.addRow(btn\_cariCrime,self.add\_cariCrime)

formCrime.addRow(self.tableWidgetCrime)

formCrime.addRow(btnPlayCrime)

btn\_cariCrime.clicked.connect(self.filterDataCrime)

btnCrime.clicked.connect(self.tampilDataCrime)

btnPlayCrime.clicked.connect(self.playFilmCrime)

##################################### - END OF CRIME - #####################################

##################################### - DRAMA - #####################################

btnDrama = QPushButton("Open Data Drama")

btnDrama.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

#membuat sebuah main window dengan nama variabel Mwindow

self.add\_cariDrama = QLineEdit(self)

self.add\_cariDrama.setStyleSheet("background-color: #eeeeee; font:bold 15px; height: 35px;")

self.add\_cariDrama.setPlaceholderText("Cari Film")

btn\_cariDrama = QPushButton("Cari")

btn\_cariDrama.setStyleSheet("background-color: #54e346; font:bold 15px; height: 35px;")

self.tableWidgetDrama = QTableWidget()

self.tableWidgetDrama.setAlternatingRowColors(True)

self.tableWidgetDrama.setColumnCount(8)

self.tableWidgetDrama.horizontalHeader().setCascadingSectionResizes(False)

self.tableWidgetDrama.horizontalHeader().setSortIndicatorShown(False)

self.tableWidgetDrama.horizontalHeader().setStretchLastSection(True)

self.tableWidgetDrama.verticalHeader().setVisible(False)

self.tableWidgetDrama.verticalHeader().setCascadingSectionResizes(False)

self.tableWidgetDrama.verticalHeader().setStretchLastSection(False)

self.tableWidgetDrama.verticalHeader().setSectionResizeMode(QHeaderView.Stretch)

self.tableWidgetDrama.setHorizontalHeaderLabels(("Id", "Judul", "Tahun Rilis", "Genre", "Sinopsis","Rating","Review","Tanggal Input"))

self.tableWidgetDrama.setColumnWidth(0,10)

self.tableWidgetDrama.setColumnWidth(1,100)

self.tableWidgetDrama.setColumnWidth(4,550)

self.tableWidgetDrama.setColumnWidth(5,50)

self.tableWidgetDrama.setColumnWidth(6,730)

self.tableWidgetDrama.resizeRowsToContents()

self.tableWidgetDrama.verticalScrollBar().maximum()

self.tableWidgetDrama.verticalScrollBar().setValue(0)

self.tableWidgetDrama.setStyleSheet("background-color: #eeeeee; border:2px #ffd369;")

self.tableWidgetDrama.horizontalHeader().setStyleSheet("background-color: #ffd369; font:bold 15px;")

self.tableWidgetDrama.verticalHeader().setSectionResizeMode(QtWidgets.QHeaderView.ResizeToContents)

self.tableWidgetDrama.setWordWrap(True)

btnPlayDrama = QPushButton("Play Film Drama")

btnPlayDrama.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

formDrama.addRow(btnDrama)

formDrama.addRow(btn\_cariDrama,self.add\_cariDrama)

formDrama.addRow(self.tableWidgetDrama)

formDrama.addRow(btnPlayDrama)

btn\_cariDrama.clicked.connect(self.filterDataDrama)

btnDrama.clicked.connect(self.tampilDataDrama)

btnPlayDrama.clicked.connect(self.playFilmDrama)

btnHorror = QPushButton("Open Data Horror")

btnHorror.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

#membuat sebuah main window dengan nama variabel Mwindow

self.add\_cariHorror = QLineEdit(self)

self.add\_cariHorror.setStyleSheet("background-color: #eeeeee; font:bold 15px; height: 35px;")

self.add\_cariHorror.setPlaceholderText("Cari Film")

btn\_cariHorror = QPushButton("Cari")

btn\_cariHorror.setStyleSheet("background-color: #54e346; font:bold 15px; height: 35px;")

self.tableWidgetHorror = QTableWidget()

self.tableWidgetHorror.setAlternatingRowColors(True)

self.tableWidgetHorror.setColumnCount(8)

self.tableWidgetHorror.horizontalHeader().setCascadingSectionResizes(False)

self.tableWidgetHorror.horizontalHeader().setSortIndicatorShown(False)

self.tableWidgetHorror.horizontalHeader().setStretchLastSection(True)

self.tableWidgetHorror.verticalHeader().setVisible(False)

self.tableWidgetHorror.verticalHeader().setCascadingSectionResizes(False)

self.tableWidgetHorror.verticalHeader().setStretchLastSection(False)

self.tableWidgetHorror.verticalHeader().setSectionResizeMode(QHeaderView.Stretch)

self.tableWidgetHorror.setHorizontalHeaderLabels(("Id", "Judul", "Tahun Rilis", "Genre", "Sinopsis","Rating","Review","Tanggal Input"))

self.tableWidgetHorror.setColumnWidth(0,10)

self.tableWidgetHorror.setColumnWidth(1,100)

self.tableWidgetHorror.setColumnWidth(4,550)

self.tableWidgetHorror.setColumnWidth(5,50)

self.tableWidgetHorror.setColumnWidth(6,730)

self.tableWidgetHorror.resizeRowsToContents()

self.tableWidgetHorror.verticalScrollBar().maximum()

self.tableWidgetHorror.verticalScrollBar().setValue(0)

self.tableWidgetHorror.setStyleSheet("background-color: #eeeeee; border:2px #ffd369;")

self.tableWidgetHorror.horizontalHeader().setStyleSheet("background-color: #ffd369; font:bold 15px;")

self.tableWidgetHorror.verticalHeader().setSectionResizeMode(QtWidgets.QHeaderView.ResizeToContents)

self.tableWidgetHorror.setWordWrap(True)

btnPlayHorror = QPushButton("Play Film Horror")

btnPlayHorror.setStyleSheet("background-color: #ffd369; font:bold 15px; height: 35px;")

formHorror.addRow(btnHorror)

formHorror.addRow(btn\_cariHorror,self.add\_cariHorror)

formHorror.addRow(self.tableWidgetHorror)

formHorror.addRow(btnPlayHorror)

btn\_cariHorror.clicked.connect(self.filterDataHorror)

btnHorror.clicked.connect(self.tampilDataHorror)

btnPlayHorror.clicked.connect(self.PlayFilmHorror)

#Layout grid di jadikan layout utama

windowx.setLayout(gridx)

#menentukan ukuran window, + title dan menampilkan

#membuat QWidget dengan nama variabel window

Mwindow.setStyleSheet("background-color: #222831;")

Mwindow.setCentralWidget(windowx)

Mwindow.setGeometry(0,35,2200,1000)

Mwindow.setWindowTitle("List of Film Based on Genre")

##################################### - END OF DRAMA - #####################################

##################################### -END LIST OF FILM BASED ON GENRE- #####################################

##################################### -LIST OF ACTION- #####################################

#Fungsi untuk menampilankan data di dalam table

def tampilDataAction(self):

self.connection = sqlite3.connect("test.db")

query = "SELECT \* FROM listfilm WHERE genre LIKE 'Action' ORDER BY rating DESC"

result = self.connection.execute(query)

self.tableWidget.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidget.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidget.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

#Fungsi Filter Data

def filterDataAction(self):

self.connection = sqlite3.connect("test.db")

filter\_search = str(self.add\_cari.text())

query = "SELECT \* FROM listfilm WHERE id LIKE '%"+str(filter\_search)+"%' OR judul LIKE '%"+str(filter\_search)+"%' OR tahun\_rilis LIKE '%"+str(filter\_search)+"%' OR genre LIKE '%"+str(filter\_search)+"%'"

result = self.connection.execute(query)

self.tableWidget.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidget.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidget.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

self.add\_cari.setText("")

##################################### -END LIST OF ACTION- #####################################

##################################### -LIST OF ADVANTURE- #####################################

#Fungsi untuk menampilankan data di dalam table

def tampilDataAdvanture(self):

self.connection = sqlite3.connect("test.db")

query = "SELECT \* FROM listfilm WHERE genre LIKE 'Advanture' ORDER BY rating DESC"

result = self.connection.execute(query)

self.tableWidgetAdvanture.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetAdvanture.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetAdvanture.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

#Fungsi Filter Data

def filterDataAdvanture(self):

self.connection = sqlite3.connect("test.db")

filter\_search = str(self.add\_cariAdvanture.text())

query = "SELECT \* FROM listfilm WHERE id LIKE '%"+str(filter\_search)+"%' OR judul LIKE '%"+str(filter\_search)+"%' OR tahun\_rilis LIKE '%"+str(filter\_search)+"%' OR genre LIKE '%"+str(filter\_search)+"%'"

result = self.connection.execute(query)

self.tableWidgetAdvanture.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetAdvanture.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetAdvanture.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

self.add\_cariAdvanture.setText("")

##################################### -END LIST OF ADVANTURE- #####################################

##################################### -LIST OF COMEDY- #####################################

#Fungsi untuk menampilankan data di dalam table

def tampilDataComedy(self):

self.connection = sqlite3.connect("test.db")

query = "SELECT \* FROM listfilm WHERE genre LIKE 'Comedy' ORDER BY rating DESC"

result = self.connection.execute(query)

self.tableWidgetComedy.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetComedy.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetComedy.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

#Fungsi Filter Data

def filterDataComedy(self):

self.connection = sqlite3.connect("test.db")

filter\_search = str(self.add\_cariComedy.text())

query = "SELECT \* FROM listfilm WHERE id LIKE '%"+str(filter\_search)+"%' OR judul LIKE '%"+str(filter\_search)+"%' OR tahun\_rilis LIKE '%"+str(filter\_search)+"%' OR genre LIKE '%"+str(filter\_search)+"%'"

result = self.connection.execute(query)

self.tableWidgetComedy.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetComedy.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetComedy.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

self.add\_cariComedy.setText("")

##################################### -END LIST OF COMEDY- #####################################

##################################### -LIST OF CRIME- #####################################

#Fungsi untuk menampilankan data di dalam table

def tampilDataCrime(self):

self.connection = sqlite3.connect("test.db")

query = "SELECT \* FROM listfilm WHERE genre LIKE 'Crime' ORDER BY rating DESC"

result = self.connection.execute(query)

self.tableWidgetCrime.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetCrime.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetCrime.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

#Fungsi Filter Data

def filterDataCrime(self):

self.connection = sqlite3.connect("test.db")

filter\_search = str(self.add\_cariCrime.text())

query = "SELECT \* FROM listfilm WHERE id LIKE '%"+str(filter\_search)+"%' OR judul LIKE '%"+str(filter\_search)+"%' OR tahun\_rilis LIKE '%"+str(filter\_search)+"%' OR genre LIKE '%"+str(filter\_search)+"%'"

result = self.connection.execute(query)

self.tableWidgetCrime.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetCrime.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetCrime.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

self.add\_cariCrime.setText("")

##################################### -END LIST OF CRIME- #####################################

##################################### -LIST OF DRAMA- #####################################

#Fungsi untuk menampilankan data di dalam table

def tampilDataDrama(self):

self.connection = sqlite3.connect("test.db")

query = "SELECT \* FROM listfilm WHERE genre LIKE 'Drama' ORDER BY rating DESC"

result = self.connection.execute(query)

self.tableWidgetDrama.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetDrama.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetDrama.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

#Fungsi Filter Data

def filterDataDrama(self):

self.connection = sqlite3.connect("test.db")

filter\_search = str(self.add\_cariDrama.text())

query = "SELECT \* FROM listfilm WHERE id LIKE '%"+str(filter\_search)+"%' OR judul LIKE '%"+str(filter\_search)+"%' OR tahun\_rilis LIKE '%"+str(filter\_search)+"%' OR genre LIKE '%"+str(filter\_search)+"%'"

result = self.connection.execute(query)

self.tableWidgetDrama.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetDrama.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetDrama.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

self.add\_cariDrama.setText("")

##################################### -END LIST OF DRAMA- #####################################

##################################### -LIST OF DRAMA- #####################################

#Fungsi untuk menampilankan data di dalam table

def tampilDataHorror(self):

self.connection = sqlite3.connect("test.db")

query = "SELECT \* FROM listfilm WHERE genre LIKE 'Horror' ORDER BY rating DESC"

result = self.connection.execute(query)

self.tableWidgetHorror.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetHorror.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetHorror.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

#Fungsi Filter Data

def filterDataHorror(self):

self.connection = sqlite3.connect("test.db")

filter\_search = str(self.add\_cariHorror.text())

query = "SELECT \* FROM listfilm WHERE id LIKE '%"+str(filter\_search)+"%' OR judul LIKE '%"+str(filter\_search)+"%' OR tahun\_rilis LIKE '%"+str(filter\_search)+"%' OR genre LIKE '%"+str(filter\_search)+"%'"

result = self.connection.execute(query)

self.tableWidgetHorror.setRowCount(0)

for row\_number, row\_data in enumerate(result):

self.tableWidgetHorror.insertRow(row\_number)

for column\_number, data in enumerate(row\_data):

self.tableWidgetHorror.setItem(row\_number, column\_number,QTableWidgetItem(str(data)))

self.connection.close()

self.add\_cariHorror.setText("")

##################################### -END LIST OF DRAMA- #####################################

##################################### -EXPORT NEW WINDOW FILM- #####################################

def PlayFilmAction(self):

playFilmAction.layoutFilmAction()

def PlayFilmAdvanture(self):

playFilmAdvanture.layoutFilmAdvanture()

def PlayFilmComedy(self):

playFilmComedy.layoutFilmComedy()

def playFilmCrime(self):

playFilmCrime.layoutFilmCrime()

def playFilmDrama(self):

playFilmDrama.layoutFilmDrama()

def PlayFilmHorror(self):

playFilmDrama.layoutFilmDrama()

##################################### -END OF EXPORT NEW WINDOW FILM- #####################################

if \_\_name\_\_ == '\_\_main\_\_':

#Inisisalisai pyqt

app = QApplication(sys.argv)

#membuat style pada app dengan style fusion

app.setStyle("fusion")

#menambahkan icon pada tampilan window dengan nama file icon 'replay\_ok.png'

app.setWindowIcon(QIcon('replay\_ok.png'))

#membuat sebuah main window dengan nama variabel Mwindow

window2 = QWidget()

Mwindow = QMainWindow()

ui = Login()

ui.Loginwindow(Mwindow)

#menampilkan isi mainwiondow dan membuat sistem exit

Mwindow.show()

sys.exit(app.exec\_())

1. playFilmAction.py

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

from PyQt5 import \*

from PyQt5.QtCore import \*

from PyQt5.QtMultimedia import \*

from PyQt5.QtMultimediaWidgets import \*

app = QApplication(sys.argv)

#membuat style pada app dengan style fusion

app.setStyle("fusion")

#menambahkan icon pada tampilan window dengan nama file icon 'math.png'

app.setWindowIcon(QIcon('replay.jpg'))

#membuat sebuah main window dengan nama variabel Mwindow

windowx = QWidget()

self = windowx

Mwindow = QMainWindow()

def layoutFilmAction():

windowx = QWidget(self)

#Membuat media player

self.mediaPlayer = QMediaPlayer(None, QMediaPlayer.VideoSurface)

#Membuat videowidget

videowidget = QVideoWidget()

#Menampilkan videowidget ke dalam media player

self.mediaPlayer.setVideoOutput(videowidget)

#Membuat open button

openBtn = QPushButton('Open Video')

openBtn.setStyleSheet("background-color: #dddddd;")

openBtn.clicked.connect(Open\_file)

#Membuat play button

self.playBtn = QPushButton()

self.playBtn.setStyleSheet("background-color: #dddddd;")

self.playBtn.setEnabled(False)

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

self.playBtn.clicked.connect(Play\_video)

#Membuat slider

self.slider = QSlider(Qt.Horizontal)

self.slider.setStyleSheet("background-color: #dddddd;")

self.slider.setRange(0,0)

self.slider.sliderMoved.connect(Set\_position)

#Membuat hbox layout

hbox = QHBoxLayout()

#Memasukkan beberapa widget ke dalam hbox

hbox.addWidget(openBtn)

hbox.addWidget(self.playBtn)

hbox.addWidget(self.slider)

#Membuat vbox layout

vbox = QVBoxLayout()

vbox.addWidget(videowidget)

vbox.addLayout(hbox)

#Membuat media player signals

self.mediaPlayer.stateChanged.connect(State\_Change)

self.mediaPlayer.positionChanged.connect(Position\_Change)

self.mediaPlayer.durationChanged.connect(Duration\_Change)

#Layout grid di jadikan layout utama

windowx.setLayout(vbox)

#membuat QWidget dengan nama variabel window

Mwindow.setStyleSheet("background-color: #222831;")

#menentukan ukuran window, + title dan menampilkan

Mwindow.setCentralWidget(windowx)

Mwindow.setGeometry(250,50,1300,980)

Mwindow.setWindowTitle("Play a Film Action")

Mwindow.show()

#sys.exit(app.exec\_())

def Open\_file():

filename, \_ = QFileDialog.getOpenFileName(self, "Open Video","Action/01.mp4")

if filename != '':

self.mediaPlayer.setMedia(QMediaContent(QUrl.fromLocalFile(filename)))

self.playBtn.setEnabled(True)

def Play\_video():

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.mediaPlayer.pause()

else:

self.mediaPlayer.play()

def State\_Change( state):

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPause))

else:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

def Position\_Change(position):

self.slider.setValue(position)

def Duration\_Change( duration):

self.slider.setRange(0, duration)

def Set\_position(position):

self.mediaPlayer.setPosition(position)

if \_\_name\_\_ == '\_\_main\_\_':

layoutFilmAction()

1. playFilmAdvanture.py

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

from PyQt5 import \*

from PyQt5.QtCore import \*

from PyQt5.QtMultimedia import \*

from PyQt5.QtMultimediaWidgets import \*

app = QApplication(sys.argv)

#membuat style pada app dengan style fusion

app.setStyle("fusion")

#menambahkan icon pada tampilan window dengan nama file icon 'math.png'

app.setWindowIcon(QIcon('replay.jpg'))

#membuat sebuah main window dengan nama variabel Mwindow

windowx = QWidget()

self = windowx

Mwindow = QMainWindow()

def layoutFilmAdvanture():

windowx = QWidget(self)

#Membuat media player

self.mediaPlayer = QMediaPlayer(None, QMediaPlayer.VideoSurface)

#Membuat videowidget

videowidget = QVideoWidget()

#Menampilkan videowidget ke dalam media player

self.mediaPlayer.setVideoOutput(videowidget)

#Membuat open button

openBtn = QPushButton('Open Video')

openBtn.setStyleSheet("background-color: #dddddd;")

openBtn.clicked.connect(Open\_file)

#Membuat play button

self.playBtn = QPushButton()

self.playBtn.setStyleSheet("background-color: #dddddd;")

self.playBtn.setEnabled(False)

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

self.playBtn.clicked.connect(Play\_video)

#Membuat slider

self.slider = QSlider(Qt.Horizontal)

self.slider.setStyleSheet("background-color: #dddddd;")

self.slider.setRange(0,0)

self.slider.sliderMoved.connect(Set\_position)

#Membuat hbox layout

hbox = QHBoxLayout()

#Memasukkan beberapa widget ke dalam hbox

hbox.addWidget(openBtn)

hbox.addWidget(self.playBtn)

hbox.addWidget(self.slider)

#Membuat vbox layout

vbox = QVBoxLayout()

vbox.addWidget(videowidget)

vbox.addLayout(hbox)

#media player signals

self.mediaPlayer.stateChanged.connect(State\_Change)

self.mediaPlayer.positionChanged.connect(Position\_Change)

self.mediaPlayer.durationChanged.connect(Duration\_Change)

#Layout grid di jadikan layout utama

windowx.setLayout(vbox)

#membuat QWidget dengan nama variabel window

Mwindow.setStyleSheet("background-color: #222831;")

#menentukan ukuran window, + title dan menampilkan

Mwindow.setCentralWidget(windowx)

Mwindow.setGeometry(250,50,1300,980)

Mwindow.setWindowTitle("Play a Film Advanture")

Mwindow.show()

#sys.exit(app.exec\_())

def Open\_file():

filename, \_ = QFileDialog.getOpenFileName(self, "Open Video","Advanture/01.mp4")

if filename != '':

self.mediaPlayer.setMedia(QMediaContent(QUrl.fromLocalFile(filename)))

self.playBtn.setEnabled(True)

def Play\_video():

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.mediaPlayer.pause()

else:

self.mediaPlayer.play()

def State\_Change( state):

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPause))

else:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

def Position\_Change(position):

self.slider.setValue(position)

def Duration\_Change( duration):

self.slider.setRange(0, duration)

def Set\_position(position):

self.mediaPlayer.setPosition(position)

if \_\_name\_\_ == '\_\_main\_\_':

layoutFilmAdvanture()

1. playFilmComedy.py

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

from PyQt5 import \*

from PyQt5.QtCore import \*

from PyQt5.QtMultimedia import \*

from PyQt5.QtMultimediaWidgets import \*

app = QApplication(sys.argv)

#membuat style pada app dengan style fusion

app.setStyle("fusion")

#menambahkan icon pada tampilan window dengan nama file icon 'math.png'

app.setWindowIcon(QIcon('replay.jpg'))

#membuat sebuah main window dengan nama variabel Mwindow

windowx = QWidget()

self = windowx

Mwindow = QMainWindow()

def layoutFilmComedy():

windowx = QWidget(self)

#Membuat media player

self.mediaPlayer = QMediaPlayer(None, QMediaPlayer.VideoSurface)

#Membuat videowidget

videowidget = QVideoWidget()

#Menampilkan videowidget ke dalam media player

self.mediaPlayer.setVideoOutput(videowidget)

#Membuat open button

openBtn = QPushButton('Open Video')

openBtn.setStyleSheet("background-color: #dddddd;")

openBtn.clicked.connect(Open\_file)

#Membuat play button

self.playBtn = QPushButton()

self.playBtn.setStyleSheet("background-color: #dddddd;")

self.playBtn.setEnabled(False)

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

self.playBtn.clicked.connect(Play\_video)

#Membuat slider

self.slider = QSlider(Qt.Horizontal)

self.slider.setStyleSheet("background-color: #dddddd;")

self.slider.setRange(0,0)

self.slider.sliderMoved.connect(Set\_position)

#Membuat hbox layout

hbox = QHBoxLayout()

#Memasukkan beberapa widget ke dalam hbox

hbox.addWidget(openBtn)

hbox.addWidget(self.playBtn)

hbox.addWidget(self.slider)

#Membuat vbox layout

vbox = QVBoxLayout()

vbox.addWidget(videowidget)

vbox.addLayout(hbox)

#media player signals

self.mediaPlayer.stateChanged.connect(State\_Change)

self.mediaPlayer.positionChanged.connect(Position\_Change)

self.mediaPlayer.durationChanged.connect(Duration\_Change)

#Layout grid di jadikan layout utama

windowx.setLayout(vbox)

#membuat QWidget dengan nama variabel window

Mwindow.setStyleSheet("background-color: #222831;")

#menentukan ukuran window, + title dan menampilkan

Mwindow.setCentralWidget(windowx)

Mwindow.setGeometry(250,50,1300,980)

Mwindow.setWindowTitle("Play a Film Comedy")

Mwindow.show()

#sys.exit(app.exec\_())

def Open\_file():

filename, \_ = QFileDialog.getOpenFileName(self, "Open Video","Comedy/01.mp4")

if filename != '':

self.mediaPlayer.setMedia(QMediaContent(QUrl.fromLocalFile(filename)))

self.playBtn.setEnabled(True)

def Play\_video():

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.mediaPlayer.pause()

else:

self.mediaPlayer.play()

def State\_Change( state):

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPause))

else:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

def Position\_Change(position):

self.slider.setValue(position)

def Duration\_Change( duration):

self.slider.setRange(0, duration)

def Set\_position(position):

self.mediaPlayer.setPosition(position)

if \_\_name\_\_ == '\_\_main\_\_':

layoutFilmComedy()

1. playFIlmCrime.py

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

from PyQt5 import \*

from PyQt5.QtCore import \*

from PyQt5.QtMultimedia import \*

from PyQt5.QtMultimediaWidgets import \*

app = QApplication(sys.argv)

#membuat style pada app dengan style fusion

app.setStyle("fusion")

#menambahkan icon pada tampilan window dengan nama file icon 'math.png'

app.setWindowIcon(QIcon('replay.jpg'))

#membuat sebuah main window dengan nama variabel Mwindow

windowx = QWidget()

self = windowx

Mwindow = QMainWindow()

def layoutFilmCrime():

windowx = QWidget(self)

#Membuat media player

self.mediaPlayer = QMediaPlayer(None, QMediaPlayer.VideoSurface)

#Membuat videowidget

videowidget = QVideoWidget()

#Menampilkan videowidget ke dalam media player

self.mediaPlayer.setVideoOutput(videowidget)

#Membuat open button

openBtn = QPushButton('Open Video')

openBtn.setStyleSheet("background-color: #dddddd;")

openBtn.clicked.connect(Open\_file)

#Membuat play button

self.playBtn = QPushButton()

self.playBtn.setStyleSheet("background-color: #dddddd;")

self.playBtn.setEnabled(False)

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

self.playBtn.clicked.connect(Play\_video)

#Membuat slider

self.slider = QSlider(Qt.Horizontal)

self.slider.setStyleSheet("background-color: #dddddd;")

self.slider.setRange(0,0)

self.slider.sliderMoved.connect(Set\_position)

#Membuat hbox layout

hbox = QHBoxLayout()

#Memasukkan beberapa widget ke dalam hbox

hbox.addWidget(openBtn)

hbox.addWidget(self.playBtn)

hbox.addWidget(self.slider)

#Membuat vbox layout

vbox = QVBoxLayout()

vbox.addWidget(videowidget)

vbox.addLayout(hbox)

#media player signals

self.mediaPlayer.stateChanged.connect(State\_Change)

self.mediaPlayer.positionChanged.connect(Position\_Change)

self.mediaPlayer.durationChanged.connect(Duration\_Change)

#Layout grid di jadikan layout utama

windowx.setLayout(vbox)

#membuat QWidget dengan nama variabel window

Mwindow.setStyleSheet("background-color: #222831;")

#menentukan ukuran window, + title dan menampilkan

Mwindow.setCentralWidget(windowx)

Mwindow.setGeometry(250,50,1300,980)

Mwindow.setWindowTitle("Play a Film Crime")

Mwindow.show()

#sys.exit(app.exec\_())

def Open\_file():

filename, \_ = QFileDialog.getOpenFileName(self, "Open Video","Crime/01.mp4")

if filename != '':

self.mediaPlayer.setMedia(QMediaContent(QUrl.fromLocalFile(filename)))

self.playBtn.setEnabled(True)

def Play\_video():

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.mediaPlayer.pause()

else:

self.mediaPlayer.play()

def State\_Change( state):

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPause))

else:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

def Position\_Change(position):

self.slider.setValue(position)

def Duration\_Change( duration):

self.slider.setRange(0, duration)

def Set\_position(position):

self.mediaPlayer.setPosition(position)

if \_\_name\_\_ == '\_\_main\_\_':

layoutFilmCrime()

1. playFilmDrama.py

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

from PyQt5 import \*

from PyQt5.QtCore import \*

from PyQt5.QtMultimedia import \*

from PyQt5.QtMultimediaWidgets import \*

app = QApplication(sys.argv)

#membuat style pada app dengan style fusion

app.setStyle("fusion")

#menambahkan icon pada tampilan window dengan nama file icon 'math.png'

app.setWindowIcon(QIcon('replay.jpg'))

#membuat sebuah main window dengan nama variabel Mwindow

windowx = QWidget()

self = windowx

Mwindow = QMainWindow()

def layoutFilmDrama():

windowx = QWidget(self)

#Membuat media player

self.mediaPlayer = QMediaPlayer(None, QMediaPlayer.VideoSurface)

#Membuat videowidget

videowidget = QVideoWidget()

#Menampilkan videowidget ke dalam media player

self.mediaPlayer.setVideoOutput(videowidget)

#Membuat open button

openBtn = QPushButton('Open Video')

openBtn.setStyleSheet("background-color: #dddddd;")

openBtn.clicked.connect(Open\_file)

#Membuat play button

self.playBtn = QPushButton()

self.playBtn.setStyleSheet("background-color: #dddddd;")

self.playBtn.setEnabled(False)

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

self.playBtn.clicked.connect(Play\_video)

#Membuat slider

self.slider = QSlider(Qt.Horizontal)

self.slider.setStyleSheet("background-color: #dddddd;")

self.slider.setRange(0,0)

self.slider.sliderMoved.connect(Set\_position)

#Membuat hbox layout

hbox = QHBoxLayout()

#Memasukkan beberapa widget ke dalam hbox

hbox.addWidget(openBtn)

hbox.addWidget(self.playBtn)

hbox.addWidget(self.slider)

#Membuat vbox layout

vbox = QVBoxLayout()

vbox.addWidget(videowidget)

vbox.addLayout(hbox)

#media player signals

self.mediaPlayer.stateChanged.connect(State\_Change)

self.mediaPlayer.positionChanged.connect(Position\_Change)

self.mediaPlayer.durationChanged.connect(Duration\_Change)

#Layout grid di jadikan layout utama

windowx.setLayout(vbox)

#membuat QWidget dengan nama variabel window

Mwindow.setStyleSheet("background-color: #222831;")

#menentukan ukuran window, + title dan menampilkan

Mwindow.setCentralWidget(windowx)

Mwindow.setGeometry(250,50,1300,980)

Mwindow.setWindowTitle("Play a Film Drama")

Mwindow.show()

#sys.exit(app.exec\_())

def Open\_file():

filename, \_ = QFileDialog.getOpenFileName(self, "Open Video","Drama/01.mp4")

if filename != '':

self.mediaPlayer.setMedia(QMediaContent(QUrl.fromLocalFile(filename)))

self.playBtn.setEnabled(True)

def Play\_video():

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.mediaPlayer.pause()

else:

self.mediaPlayer.play()

def State\_Change( state):

if self.mediaPlayer.state() == QMediaPlayer.PlayingState:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPause))

else:

self.playBtn.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

def Position\_Change(position):

self.slider.setValue(position)

def Duration\_Change( duration):

self.slider.setRange(0, duration)

def Set\_position(position):

self.mediaPlayer.setPosition(position)

if \_\_name\_\_ == '\_\_main\_\_':

layoutFilmDrama()