**PROGRESS PROJECT 2 APLIKASI MATA KULIAH PEMROGRAMAN DEKSTOP**

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**

**2020**

1. **LISTING PROGRAM INPUT DATA dan PENJELASAN PROGRAM**

* **Code**

*def Admin(self, Mwindow):*

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

*self.barAdmin = Mwindow.menuBar()*

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

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

*#menambahkan menu 'Open' ke dalam bar dan di inisialisasi ke variabel file1*

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

*#menambahkan QAction "Rumus Kecepatan" di dalam menubar open dengan inisialisasi ke variabel file1a*

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

*#memasukkan file1a ke dalam menu bar file1*

*self.file1.addAction(self.file1a)*

*self.file0.triggered.connect(self.open\_home)*

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

*window = QWidget()*

*#Membuat Grid Layout*

*grid = QGridLayout()*

*open\_database = QPushButton("Open Database")*

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

*open\_database.setStyleSheet("background-color: #ffc7c7;")*

*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)*

*add\_button = QPushButton("Tambah Data")*

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

*add\_button.setStyleSheet("background-color: #ffc7c7;")*

*add\_update = QPushButton("Update Data")*

*grid.addWidget(add\_update,12,0,1,0)*

*add\_update.setStyleSheet("background-color: #ffc7c7;")*

*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: #ffc7c7;")*

*add\_delete = QPushButton("Delete Data")*

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

*add\_delete.setStyleSheet("background-color: #ffc7c7;")*

*#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)*

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

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

*#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\_update.clicked.connect(self.updateData)*

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

*#Layout grid di jadikan layout utama*

*window.setLayout(grid)*

*#menentukan ukuran window, + title dan menampilkan*

*Mwindow.setCentralWidget(window)*

*Mwindow.setGeometry(50,50,1300,900)*

*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!')*

*#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 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)*

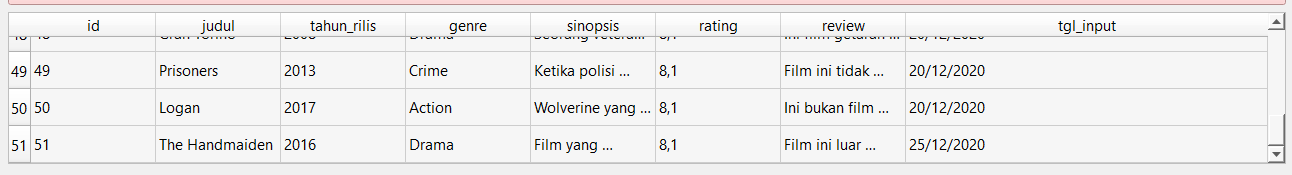
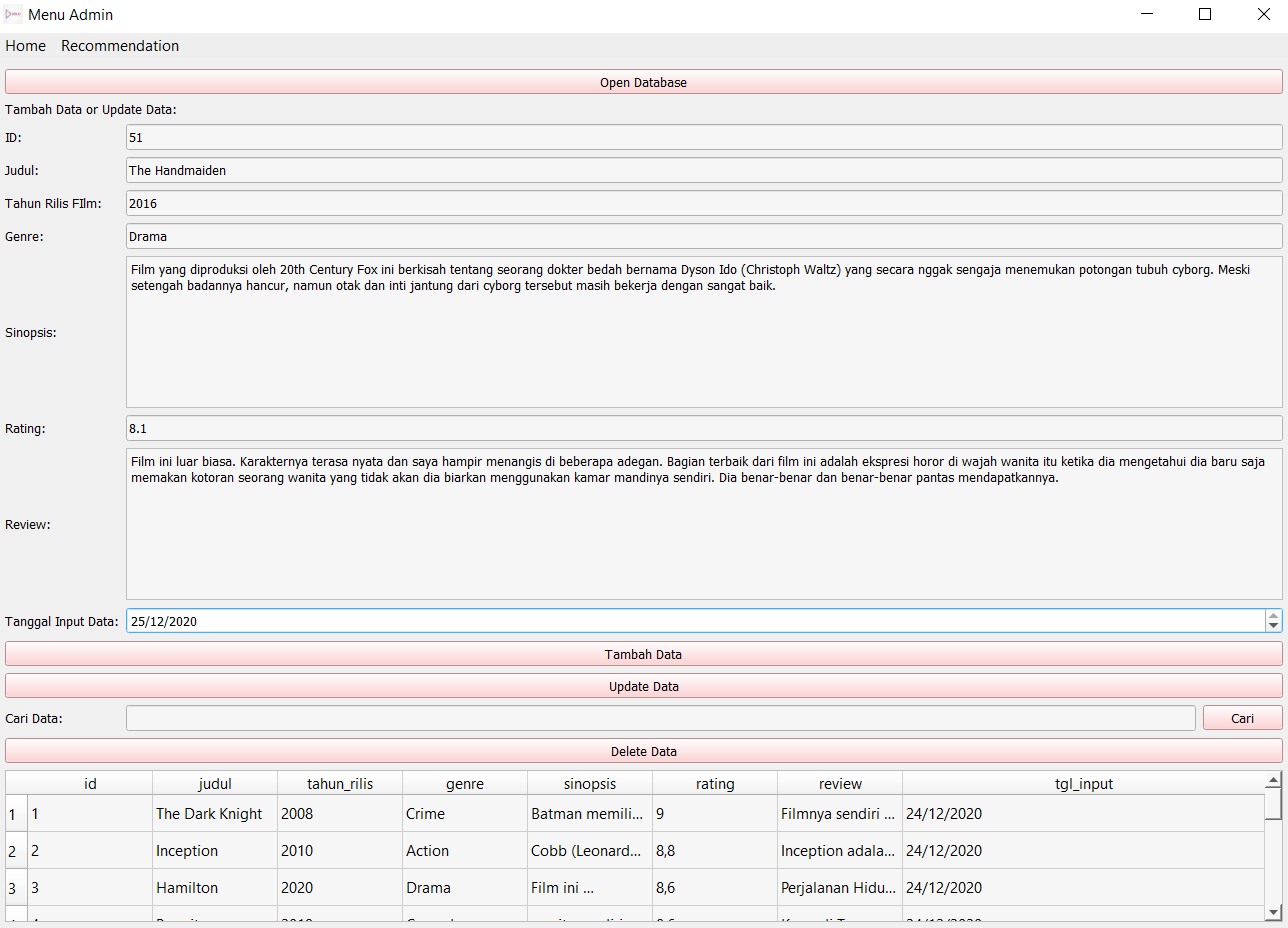
*return self.tableview*

* **Penjelasan**

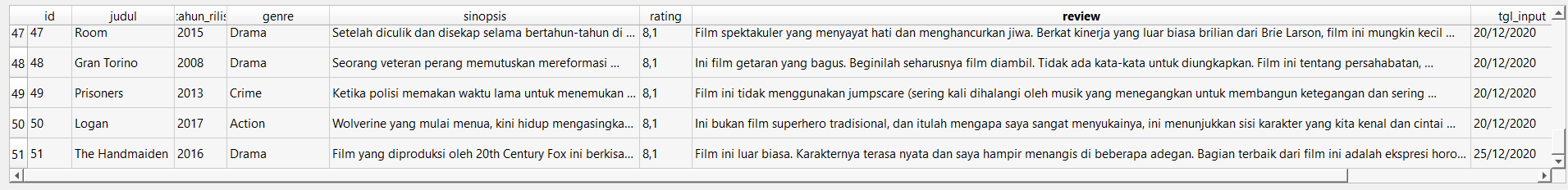
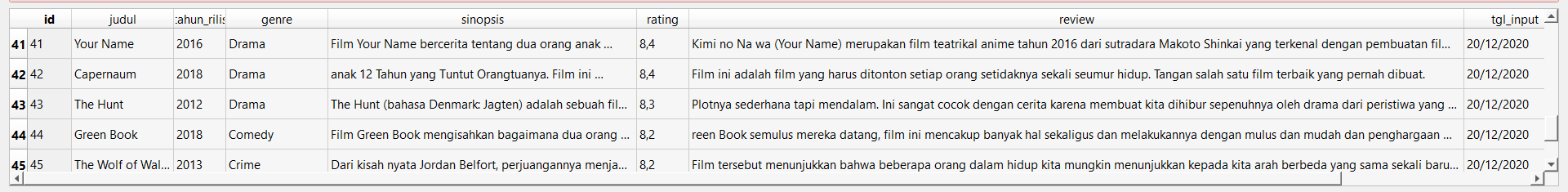
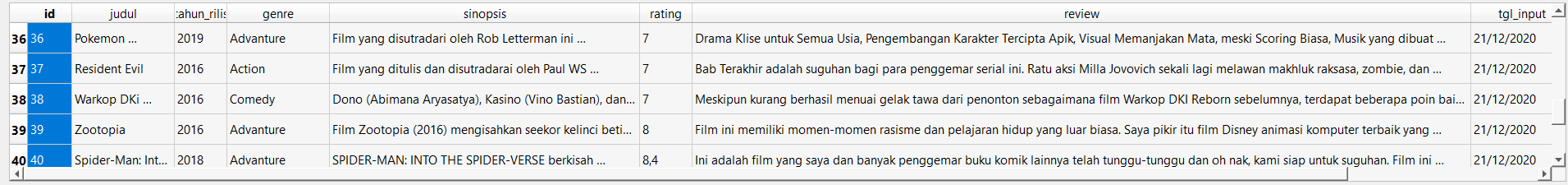
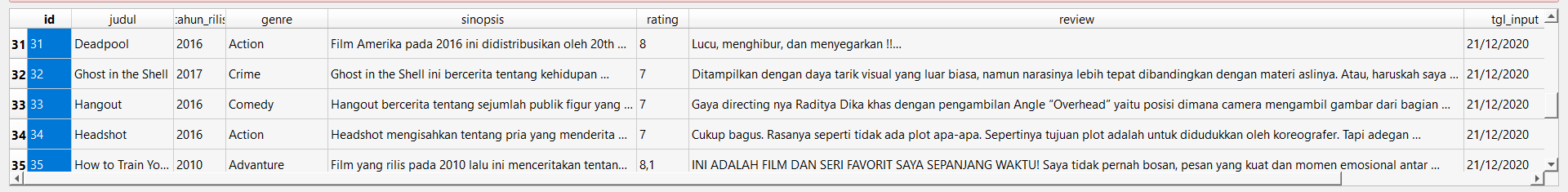
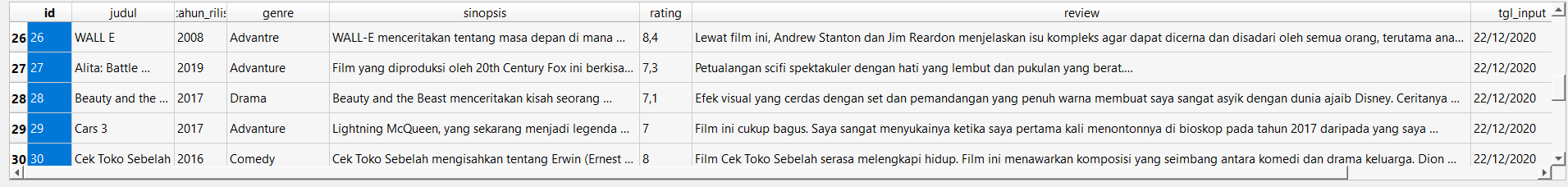
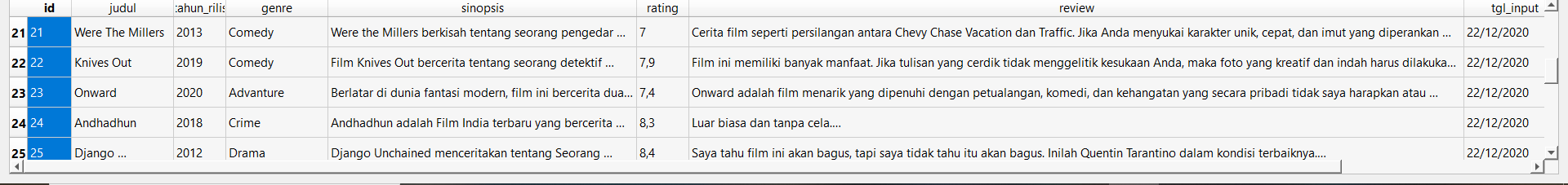
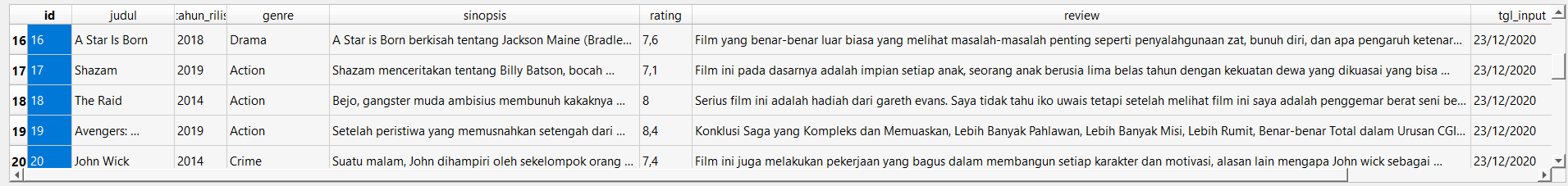
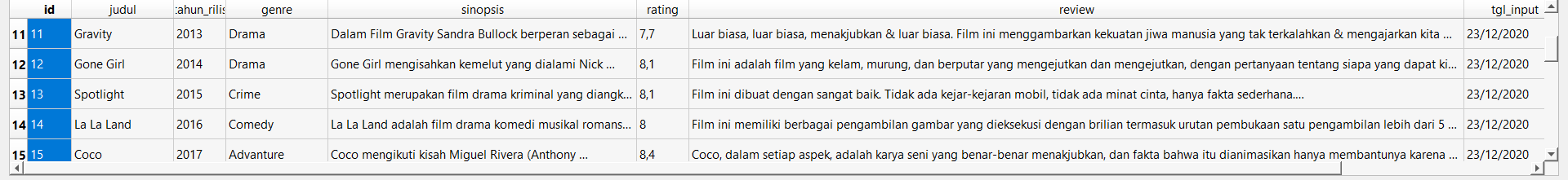
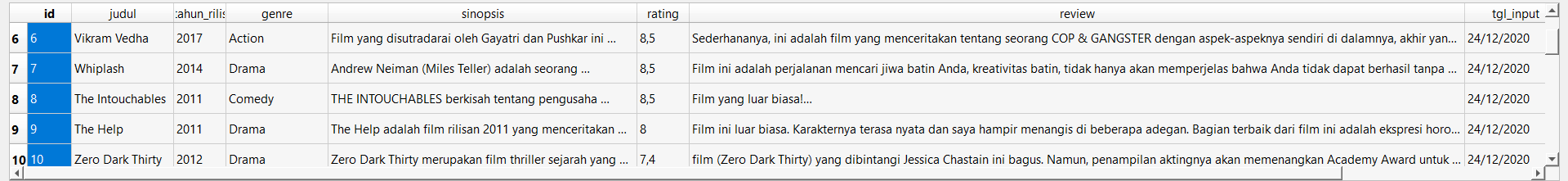
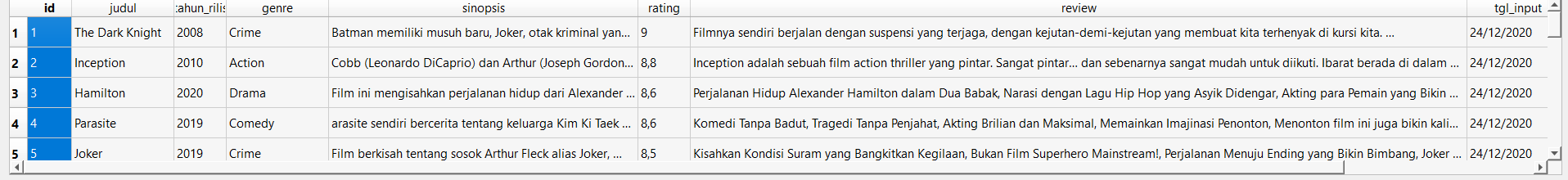
Membuat beberapa label, line edit, text edit, date edit, table view dan button. Kemudian widget – widget tersebut dimasukkan ke dalam form layout yang sudah di buat. Lalu membuat sebuah signal ketika button ditekan maka akan memberikan signal kepada fungsi / slot yang telah dibuat, seperti pada saat button tambah data ditekan maka akan menghubungkan ke fungsi tambahData. Selanjutnya adalah membuat fungsi OpenDatabaseAdmin untuk menghubungkan ke dalam database yang telah dibuat, dan fungsi tambahData untuk menambahkan data ke dalam database, dan fungsi tampilDataListFilm untuk menampilkan data – data yang ada di film ke widget table view.

Ketika button tambahData di klik maka akan menjalankan fungsi tambahData yaitu dengan cara menerima semua inputan user dan dimasukkan ke dalam varbale baru yang telah dibuat. Kemudian menjalankan perintah query untuk memasukkan data – data yang diinputkan ke dalam table listfilm. Jika berhasil makan akan menampilkan data baru yang telah diinputkan dengan menjalankan fungsi tampilDataListFilm, namun jika gagal maka akan menampilkan pesan error di dalam terminal.

* **Capture**



1. **HASIL INPUT DATA**

****