

**FOX ID SEBAGAI SISTEM PEMESANAN
TIKET BIOSKOP
TUGAS AKHIR MATA KULIAH ALGORITMA DAN
PEMROGRAMAN**



oleh
Mufadhal Faraz Addhifa
1202200045

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
TELKOM UNIVERSITY**

2021

KATA PENGANTAR

Puji syukur atas nikmat iman, nikmat kesehatan, dan nikmat kesempatan yang telah diberikan Allah Subhanahu wa ta'ala kepada penulis, sehingga makalah yang berjudul “FOX ID SEBAGAI SISTEM PEMESANAN TIKET BIOSKOP” ini dapat penulis selesaikan dengan baik dan tepat waktu.

Tak lupa juga, melalui kata pengantar ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penulis dalam menyusun makalah ini. Mungkin tanpa bantuan dari semua pihak yang telah membantu penulis, makalah ini tidak akan bisa selesai dengan sebaik ini.

Di dalam makalah ini, berisikan mengenai hal-hal yang berkaitan dengan sistem pemesanan tiket yang biasanya dilakukan secara langsung di bioskop-bioskop yang dituju. Selain itu makalah ini juga membahas mengenai berbagai macam jenis fitur yang dihadirkan dalam FOX ID yang membuat FOX ID lebih menarik untuk digunakan.

Meskipun makalah ini telah penulis susun dengan sebaik-baiknya, tapi tetap saja makalah ini tidak luput dari kesalahan. Semoga makalah yang telah penulis susun dengan baik ini dapat memberikan manfaat kepada pembaca sekalian dan semoga makalah ini bisa bermanfaat ke depannya. Dan tak lupa, semoga makalah ini bisa memenuhi tugas yang telah diberikan oleh Bapak Dr. Teguh Nurhadi Suharsono,ST.,MT selaku pengampu mata kuliah Algoritma dan Pemrograman, cukup sekian dan terima kasih.

Jakarta, 26 Mei 2021

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL.....	v
BAB I <u>PENDAHULUAN</u>	1
1.1 Latar Belakang	1
1.2 Identifikasi dan Rumusan Masalah	2
1.2.1 Identifikasi Masalah	2
1.2.2 Rumusan Masalah	2
1.3 Maksud dan Tujuan Penulisan	2
1.3.1 Maksud Penulisan	2
1.3.2 Tujuan Penulisan.....	2
1.4 Batasan Masalah	3
1.5 Metodologi Penulisan	3
BAB II <u>LANDASAN TEORI</u>	4
2.1 Konsep Dasar Sistem	4
2.2 Konsep Dasar Pemrograman.....	4
2.3 Flowchart	5
2.4 Python	6
2.4.1 Tipe Data Python	7
2.4.2 Variabel Python.....	7
2.4.3 Operator Python.....	8
2.4.4 Kondisi Python.....	9
2.4.5 Perulangan Python	10
2.4.6 List Python.....	11
2.4.7 Fungsi Python	13
2.4.8 Modul Python.....	14
2.4.9 Exception Python.....	15
2.4.10 Membaca Dan Menulis File	18
BAB III <u>PEMBAHASAN</u>	19

3.1 Analisis Sistem.....	19
3.1.1 Ringkasan Sistem.....	19
3.1.2 Proses Sistem	19
3.2 Flowchart	20
3.3 Source Code dan Output.....	25
3.3.1 Deklarasi Awal.....	25
3.3.2 Halaman Menu Utama	25
3.3.3 Halaman Menu Now Showing.....	27
3.3.4 Halaman Menu Movie Recommendations.....	28
3.3.5 Halaman Menu Information.....	29
3.3.6 Halaman Menu Coupon Code.....	30
3.3.7 Halaman Menu Synopsis	30
3.3.8 Halaman Menu Poster	32
3.3.9 Logika Keputusan	33
3.3.10 Halaman Menu Pembelian	34
3.3.11 Logika Mulai.....	37
3.4 Dry Run.....	38
3.5 Konfigurasi Sistem.....	38
BAB IV PENUTUP.....	39
4.1 Simpulan	39
4.2 Saran.....	39
DAFTAR PUSTAKA	40

DAFTAR GAMBAR

Gambar 2.1 Simbol flowchart 1	5
Gambar 2.2 Simbol flowchart 2	6
Gambar 3.1 Flowchart FOX ID 1.....	20
Gambar 3.2 Flowchart FOX ID 2.....	21
Gambar 3.3 Flowchart FOX ID 3.....	22
Gambar 3.4 Flowchart FOX ID 4.....	23
Gambar 3.5 Flowchart FOX ID 5.....	24
Gambar 3.6 Menu Utama	25
Gambar 3.7 Menu Now Showing.....	27
Gambar 3.8 Menu Movie Recommendations	28
Gambar 3.9 Menu Information.....	29
Gambar 3.10 Menu Coupon Code.....	30
Gambar 3.11 Menu Synopsis	30
Gambar 3.12 Menu See The Poster.....	32
Gambar 3.13 Menu Buy Ticket.....	34

DAFTAR TABEL

Tabel 3.1 Tabel Pengujian Sistem FOX ID.....	37
--	----

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi dan komunikasi saat ini begitu pesat, dan dengan perkembangan yang pesat tersebut diperlukan informasi yang cepat dan akurat, sehingga mengakibatkan persaingan yang semakin ketat. Persaingan yang ketat dan pesatnya perkembangan teknologi dan informasi yang ada membutuhkan sistem yang lebih baik, lebih cepat dan lebih dapat diandalkan untuk memecahkan masalah.

Di era modern ini, dunia perfilman berkembang pesat. Hal ini terlihat dari semakin banyaknya industri perfilman yang memproduksi film-film berkualitas tinggi. Hal ini terlihat dari kemauan masyarakat untuk menonton film-film tersebut di bioskop.

Salah satu yang menjadi perhatian penulis adalah masalah sederhana namun sering terjadi, yaitu pemesanan tiket bioskop. Banyak pelanggan yang harus mengantre cukup lama untuk mendapatkan tiket, bahkan terkadang antrean tidak berhasil mendapatkan tiket. Hal ini tentunya akan berdampak buruk, karena dengan melakukan hal lain yang lebih bermanfaat, waktu yang dihabiskan untuk mengantre akan lebih efisien.

Cara lain untuk mengatasi masalah ini adalah menghadirkan teknologi yang membuat pelanggan dapat memesan tiket untuk waktu pemutaran di bioskop tanpa harus mengantre.

Layanan ini berfungsi sebagai media informasi antara bioskop dan pelanggan, yang dapat membantu pelanggan mendapatkan waktu tayang film di bioskop dan memesan tiket film secara *online*.

Bukan hanya itu saja, penulis juga memberikan fitur tambahan lainnya seperti memberikan daftar film yang direkomendasikan, sinopsis film yang sedang tayang, poster film yang akan tayang, serta memberikan promo-promo menarik di dalamnya. Oleh karena itu penulis membuat tugas akhir ini dengan judul “FOX ID SEBAGAI SISTEM PEMESANAN TIKET BIOSKOP”.

1.2. Identifikasi dan Rumusan Masalah

1.2.1. Identifikasi Masalah

Berdasarkan latar belakang di atas, dapat diidentifikasi permasalahan yang sedang terjadi adalah:

1. Teknologi saat ini masih kurang mendukung serta belum memberikan fitur-fitur yang diharapkan oleh pelanggan.
2. Belum tersedianya teknologi informasi yang dapat memberikan sinopsis mengenai film yang sedang ditayangkan.

1.2.2. Rumusan Masalah

Dalam makalah ini penulis dapat merumuskan permasalahan-permasalahan yang sedang terjadi, antara lain:

1. Apa fitur yang dapat dihadirkan oleh FOX ID agar pelanggan mau menggunakannya?
2. Bagaimana merancang FOX ID sebagai salah satu teknologi yang dapat memudahkan dalam pembelian tiket bioskop?

1.3. Maksud dan Tujuan Penulisan

1.3.1. Maksud Penulisan

Maksud dari ditulisnya makalah ini adalah untuk memenuhi tugas akhir mata kuliah algoritma dan pemrograman pada Jurusan Sistem Informasi Fakultas Rekayasa Industri, Telkom University.

1.3.2. Tujuan Penulisan

Adapun tujuan dari penulisan makalah ini adalah:

1. Mengetahui dan menghadirkan fitur yang diinginkan oleh pelanggan ke dalam sistem FOX ID agar pelanggan mau menggunakannya.
2. Merancang FOX ID sebagai teknologi pemesanan tiket bioskop.

1.4. Batasan Masalah

Tidak semua aspek penulis bahas dalam makalah ini, penulis membuat Batasan-batasan pembahasan agar dalam pembahasan di dalam makalah ini lebih terarah dan sesuai dengan tujuan yang akan dicapai. Adapun pembatasnya antara lain:

1. Teknologi yang akan dibangun hanya meliputi jadwal penayangan film, studio penayangan, daftar film yang direkomendasikan, poster film yang sedang ditayangkan, sinopsis film, informasi film lainnya, dan menghadirkan promo dengan syarat dan ketentuan yang berlaku.
2. Pemesanan tiket dapat dilakukan tanpa mendaftar terlebih dahulu dan dimulai dengan pemilihan judul film yang akan ditonton.
3. Tidak adanya maksimal jumlah tiket dalam pemesanan.

1.5. Metodologi Penulisan

a. Studi Pustaka

Mempelajari referensi mengenai bahasa pemrograman python dan pengoperasian perangkat lunak Visual Studio Code, yang akan digunakan sebagai dasar pembuatan FOX ID.

b. Pembuatan sistem FOX ID

1. Analisis sistem:

Merupakan bagian dalam pembuatan program dengan mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang diharapkan sehingga dapat di usulkan perbaikan-perbaikannya^[1].

2. Flowchart

Penggambaran simbol grafis yang digunakan untuk menyatakan aliran algoritma secara detail dan prosedur sistem secara logika^[2].

3. Coding dan Testing

Penulisan program sesuai desain yang sudah dirancang di atas. Serta menguji program yang sudah dibuat.

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Salah satu teori mengenai sistem secara umum pertama kali yaitu, teori yang diuraikan oleh Kenneth Boulding terutama menekankan perhatian terhadap setiap bagian yang membentuk sebuah sistem.

Teori sistem mengatakan bahwa setiap unsur pembentuk organisasi harus mendapat perhatian penuh dari pemimpin organisasi secara merata baik komponen fisik maupun non-fisik. Unsur-unsur yang mewakili suatu sistem secara umum adalah memasukan (input), pengolahan (processing), dan keluaran (output)^[3].

Menurut L. James Haver, sistem merupakan prosedur yang logis dan rasional, yang dapat berguna untuk melakukan atau merancang sebuah rangkaian suatu komponen yang berkaitan satu sama lain^[4].

2.2. Konsep Dasar Pemrograman

Pemrograman merupakan sebuah kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu, di mana instruksi tersebut menggunakan bahasa yang bisa dimengerti oleh komputer. Bahasa pemrograman yang akan diubah dan diterjemahkan ke dalam bahasa yang dapat dimengerti oleh komputer menggunakan sebuah program yang diberi nama Compiler. Instruksi-instruksi dalam bahasa pemrograman ditulis terlebih dahulu ke dalam file source code untuk kemudian tugas compiler, mengkompilasi source code menjadi bentuk file yang dapat dijalankan (dieksekusi) oleh komputer^[5].






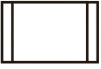
Pemrograman adalah sebuah proses yang mencakup kegiatan seperti analisis, algoritma, yang diterjemahkan ke dalam bahasa pemrograman sehingga dapat dijalankan oleh komputer.

Pengertian lainya yang secara sederhana, pemrograman adalah proses kreatif yang menginstruksikan komputer bagaimana menyelesaikan suatu tugas. Bahasa pemrograman sendiri adalah bahasa yang digunakan oleh para programmer untuk menuliskan kode-kode yang akan digunakan untuk membuat sebuah program.

Jadi bahasa pemrograman itu seperti halnya bahasa kita sehari-hari yang digunakan untuk berkomunikasi antar sesama. Namun bahasa pemrograman di sini yang menggunakan adalah sang programmer untuk berkomunikasi dengan komputer yang notabene adalah mesin. Untuk bahasa pemrograman sendiri ada banyak, seperti JavaScript, PHP, C++, Java, Python, Ruby dan lain sebagainya^[6].


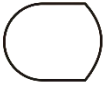




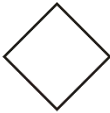
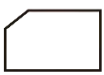


2.3. Flowchart

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program^[7]. Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart:

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure

Gambar 2.1 Simbol flowchart 1

Sumber: <https://informatikalogi.com/wp-content/uploads/2017/04/Simbol-Flowchart.jpg.webp/>

	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 2.2 Simbol flowchart 2

Sumber: <https://informatikalogi.com/wp-content/uploads/2017/04/Simbol-Flowchart.jpg.webp/>

2.4. Python

Python adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi-guna secara langsung (*interpretatif*) dengan metode orientasi objek (*Object Oriented Programming*) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan *syntax*.

Python sendiri menampilkan fitur-fitur menarik sehingga layak untuk Anda pelajari. Pertama, Python memiliki tata bahasa dan script yang sangat mudah untuk dipelajari. Python juga memiliki sistem pengelolaan data dan memori otomatis. Selain itu modul pada Python selalu diperbaharui. Ditambah lagi, Python juga memiliki banyak fasilitas pendukung. Python banyak diaplikasikan pada berbagai sistem operasi seperti Linux, Microsoft Windows, Mac OS, Android, Symbian OS, Amiga, Palm dan lain-lain^[8].

2.4.1. Tipe Data Python

Tipe data adalah suatu media atau memori pada komputer yang digunakan untuk menampung informasi. Berikut adalah tipe data dari bahasa pemrograman Python^[9]:

Boolean	: Menyatakan benar yang bernilai 1, atau salah yang bernilai 0
String	: Menyatakan karakter / kalimat bisa berupa huruf angka, dll (diapit tanda " atau ')
Integer	: Menyatakan bilangan bulat
Float	: Menyatakan bilangan desimal
List	: Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
Tuple	: Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah
Dictionary	: Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

2.4.2. Variabel Python

Variabel adalah lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Ini berarti bahwa ketika Anda membuat sebuah variabel Anda memesan beberapa ruang di memori. Variabel dapat menyimpan berbagai macam tipe data^[10].

Untuk mulai membuat variabel di Python caranya sangat mudah, Anda cukup menuliskan variabel lalu mengisinya dengan suatu nilai dengan cara menambahkan tanda sama dengan “=” diikuti dengan nilai yang ingin di masukan^[10].

Penulisan variabel Python sendiri juga memiliki aturan tertentu, yaitu^[10]:

- a. Karakter pertama harus berupa huruf atau garis bawah.
- b. Karakter selanjutnya dapat berupa huruf, garis bawah, atau angka
- c. Karakter pada nama variabel bersifat sensitif (*case-sensitif*). Artinya huruf kecil dan huruf besar dibedakan. Sebagai contoh, variabel 'namaDepan' dan 'namadepan' adalah variabel yang berbeda.

2.4.3. Operator Python

Operator adalah konstruksi yang dapat memanipulasi nilai dari operan. Bahasa pemrograman Python mendukung berbagai macam operator, di antaranya^[11]:

- a. Operator Aritmatika (*Arithmetic Operators*)
Penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), sisa bagi (%), pangkat (**), pembagian bulat (//).
- b. Operator Perbandingan (*Comparison Operators*)
Sama dengan (==), tidak sama dengan (!=), lebih besar dari (>), lebih kecil dari (<), lebih besar atau sama dengan (>=), lebih kecil atau sama dengan (<=).
- c. Operator Penugasan (*Assignment Operators*)
Sama dengan (=), tambah sama dengan (+=), kurang sama dengan (-=), kali sama dengan (*=), bagi sama dengan (/=), sisa bagi sama dengan (%=), pangkat sama dengan (**=), pembagian bulat sama dengan (//=).
- d. Operator Logika (*Logical Operators*)
- e. Operator Bitwise (*Bitwise Operators*)
- f. Operator Keanggotaan (*Membership Operators*)
- g. Operator Identitas (*Identity Operators*)

Dari semua operator di atas, masing-masing mempunyai urutan prioritas yang nantinya prioritas pertama akan dilakukan paling pertama, begitu seterusnya sampai dengan prioritas terakhir^[11].

<code>**</code>	: Aritmatika
<code>~, +, -</code>	: Bitwise
<code>*, /, %, //</code>	: Aritmatika
<code>+, -</code>	: Aritmatika
<code>>>, <<</code>	: Bitwise
<code>&</code>	: Bitwise
<code>^, </code>	: Bitwise
<code><=, <, >, >=</code>	: Perbandingan
<code><>, ==, !=</code>	: Perbandingan
<code>=, %=, /=, //=, -=, +=,</code>	: Penugasan
<code>*=, **=</code>	
<code>is, is not</code>	: Identitas
<code>in, not in</code>	: Keanggotaan
<code>not, or, and</code>	: Logika

2.4.4. Kondisi Python

a. Kondisi if

Pengambilan keputusan (kondisi if) digunakan untuk mengantisipasi kondisi yang terjadi saat jalanya program dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi.

Pada python ada beberapa statement / kondisi diantaranya adalah if, else dan elif Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar True. Jika kondisi bernilai salah False maka statement / kondisi if tidak akan di-eksekusi^[12].

b. Kondisi If Else

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil

sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai. Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else^[12].

c. Kondisi Elif

Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari “kondisi if”. Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi “else”, bedanya kondisi “elif” bisa banyak dan tidak hanya satu^[12].

2.4.5. Perulangan Python

Di dalam bahasa pemrograman Python pengulangan dibagi menjadi 3 bagian, yaitu^[13]:

a. While Loop

Pengulangan While Loop di dalam bahasa pemrograman Python dieksekusi statement berkali-kali selama kondisi bernilai benar atau True.

```
count = 0
while (count < 9):
    print ("The count is: ", count)
    count = count + 1

print ("Good bye!")
```

b. For Loop

Pengulangan for pada Python memiliki kemampuan untuk mengulangi item dari urutan apapun, seperti list atau string.


```

buah = ["nanas", "apel", "jeruk"]
for makanan in buah:
    print ("Saya suka makan", makanan)

```

c. Nested Loop

Bahasa pemrograman Python memungkinkan penggunaan satu lingkaran di dalam loop lain.

```

i = 2
while(i < 100):
    j = 2
    while(j <= (i/j)):
        if not(i%j): break
        j = j + 1
    if (j > i/j) : print(i, " is prime")
    i = i + 1

print("Good bye!")

```

2.4.6. List Python

Python memiliki enam jenis urutan built-in, namun yang paling umum adalah list dan tuple. Dalam bahasa pemrograman Python, struktur data yang paling dasar adalah urutan atau lists. Setiap elemen-elemen berurutan akan diberi nomor posisi atau indeks. Indeks pertama dalam list adalah nol, indeks kedua adalah satu dan seterusnya^[14].

a. Membuat List Python

List adalah tipe data yang paling serbaguna yang tersedia dalam bahasa Python, yang dapat ditulis sebagai daftar nilai yang dipisahkan koma (item) antara tanda kurung siku. Hal penting tentang daftar adalah item dalam list tidak boleh sama jenisnya^[14].

```

list1 = ['kimia', 'fisika', 1993, 2017]
list2 = [1, 2, 3, 4, 5 ]
list3 = ["a", "b", "c", "d"]

```

b. Akses Nilai Dalam List Python

Untuk mengakses nilai dalam list python, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan nilai yang tersedia pada indeks tersebut^[14].

```
list1 = ['fisika', 'kimia', 1993, 2017]
list2 = [1, 2, 3, 4, 5, 6, 7 ]
```

```
print ("list1[0]: ", list1[0])
print ("list2[1:5]: ", list2[1:5])
```

c. Update Nilai Dalam List Python

Anda dapat memperbarui satu atau beberapa nilai di dalam list dengan memberikan potongan di sisi kiri operator penugasan, dan Anda dapat menambahkan nilai ke dalam list dengan metode `append()`^[14].

```
list = ['fisika', 'kimia', 1993, 2017]
print ("Nilai ada pada index 2 : ", list[2])
```

```
list[2] = 2001
print ("Nilai baru ada pada index 2 : ", list[2])
```

d. Hapus Nilai Dalam List Python

Untuk menghapus nilai di dalam list python, Anda dapat menggunakan salah satu pernyataan `del` jika Anda tahu persis elemen yang Anda hapus. Anda dapat menggunakan metode `remove()` jika Anda tidak tahu persis item mana yang akan dihapus^[14].

```
list = ['fisika', 'kimia', 1993, 2017]

print (list)
del list[2]
print ("Setelah dihapus nilai pada index 2 : ", list)
```

e. **Method dan Fungsi Build-in Pada List Python**

Python menyertakan fungsi built-in sebagai berikut^[14]:

<code>len(list)</code>	: Memberikan total panjang list.
<code>max(list)</code>	: Mengembalikan item dari list dengan nilai maks.
<code>min(list)</code>	: Mengembalikan item dari list dengan nilai min.
<code>list(seq)</code>	: Mengubah tuple menjadi list.

Python menyertakan methods built-in sebagai berikut^[14]:

<code>list.append(obj)</code>	: Menambahkan objek obj ke list
<code>list.count(obj)</code>	: Jumlah pengembalian berapa kali obj terjadi dalam list
<code>list.extend(seq)</code>	: Tambahkan isi seq ke list
<code>list.index(obj)</code>	: Mengembalikan indeks terendah dalam list yang muncul obj
<code>list.insert(index, obj)</code>	: Sisipkan objek obj ke dalam list di indeks offset
<code>list.pop(obj = list[-1])</code>	: Menghapus dan mengembalikan objek atau obj terakhir dari list
<code>list.remove(obj)</code>	: Removes object obj from list
<code>list.reverse()</code>	: Membalik list objek di tempat
<code>list.sort([func])</code>	: Urutkan objek list, gunakan compare func jika diberikan

2.4.7. Fungsi Python

Fungsi adalah blok kode terorganisir dan dapat digunakan kembali yang digunakan untuk melakukan sebuah tindakan. Fungsi memberikan modularitas yang lebih baik untuk aplikasi Anda dan tingkat penggunaan kode yang tinggi. Berikut adalah aturan sederhana untuk mendefinisikan fungsi dengan Python^[15]:

- a. Fungsi blok dimulai dengan kata kunci 'def' diikuti oleh nama fungsi dan tanda kurung "()".
- b. Setiap parameter masukan atau argumen harus ditempatkan di dalam tanda kurung ini. Anda juga dapat menentukan parameter di dalam tanda kurung ini.
- c. Pernyataan pertama dari sebuah fungsi dapat berupa pernyataan opsional - string dokumentasi fungsi atau docstring.
- d. Blok kode dalam setiap fungsi dimulai dengan titik dua (:) dan indentasi.
- e. Pernyataan kembali [ekspresi] keluar dari sebuah fungsi, secara opsional menyampaikan kembali ekspresi ke pemanggil. Pernyataan pengembalian tanpa argumen sama dengan return None.

```
def printme( str ):
    "This prints a passed string into this function"
    print (str)
    return
```

2.4.8. Modul Python

Modul memungkinkan Anda mengatur kode Python secara logis. Secara sederhana modul adalah file yang terdiri dari kode Python. Modul dapat mendefinisikan fungsi, kelas dan variabel. Modul juga bisa menyertakan kode yang bisa dijalankan "runable"^[16].

Ketika interpreter menemukan sebuah pernyataan import, ia mengimpor modul jika modul tersebut ada di jalur pencarian. Jalur pencarian adalah daftar direktori yang ditafsirkan juru bahasa sebelum mengimpor modul^[16].

2.4.9. Exception Python

Python menyediakan dua fitur yang sangat penting untuk menangani kesalahan tak terduga dalam program Python Anda dan menambahkan kemampuan debugging di dalamnya^[17].

- a. Exception Handling
- b. Assertions Exception adalah sebuah peristiwa, yang terjadi selama pelaksanaan program yang mengganggu aliran normal instruksi program. Secara umum, ketika skrip Python menemukan situasi yang tidak dapat diatasi, hal itu menimbulkan pengecualian. Exception adalah objek Python yang mewakili kesalahan.

Ketika skrip Python menimbulkan Exception, ia harus menangani Exception begitu saja sehingga berhenti dan berhenti^[17].

<code>Exception</code>	: Kelas dasar untuk semua pengecualian / exception
<code>StopIteration</code>	: Dibesarkan ketika metode (iterator) berikutnya dari iterator tidak mengarah ke objek apa pun.
<code>SystemExit</code>	: Dibesarkan oleh fungsi <code>sys.exit ()</code> .
<code>StandardError</code>	: Kelas dasar untuk semua pengecualian built-in kecuali <code>StopIteration</code> dan <code>SystemExit</code> .
<code>ArithmeticError</code>	: Kelas dasar untuk semua kesalahan yang terjadi untuk perhitungan numerik.
<code>OverflowError</code>	: Dibesarkan saat perhitungan melebihi batas maksimum untuk tipe numerik.
<code>FloatingPointError</code>	: Dibesarkan saat perhitungan floating point gagal.

<code>ZeroDivisonError</code>	: Dibesarkan saat pembagian atau modulo nol dilakukan untuk semua tipe numerik.
<code>AssertionError</code>	: Dibesarkan jika terjadi kegagalan pernyataan Assert.
<code>AttributeError</code>	: Dibesarkan jika terjadi kegagalan referensi atribut atau penugasan.
<code>EOFError</code>	: Dibesarkan bila tidak ada input dari fungsi <code>raw_input ()</code> atau <code>input ()</code> dan akhir file tercapai.
<code>ImportError</code>	: Dibesarkan saat sebuah pernyataan impor gagal.
<code>KeyboardInterrupt</code>	: Dibesarkan saat pengguna menyela eksekusi program, biasanya dengan menekan Ctrl + c.
<code>LookupError</code>	: Kelas dasar untuk semua kesalahan pencarian.
<code>IndexError</code>	: Dibesarkan saat sebuah indeks tidak ditemukan secara berurutan.
<code>KeyError</code>	: Dibesarkan saat kunci yang ditentukan tidak ditemukan dalam kamus.
<code>NameError</code>	: Dibesarkan saat pengenalan tidak ditemukan di namespace lokal atau global.
<code>UnboundLocalError</code>	: Dibesarkan saat mencoba mengakses variabel lokal dalam suatu fungsi atau metode namun tidak ada nilai yang ditugaskan padanya.
<code>EnvironmentError</code>	: Kelas dasar untuk semua pengecualian yang terjadi di luar lingkungan Python.

<code>IOError</code>	: Dibesarkan saat operasi input / output gagal, seperti pernyataan cetak atau fungsi <code>open ()</code> saat mencoba membuka file yang tidak ada.
<code>OSError</code>	: Dibangkitkan untuk kesalahan terkait sistem operasi.
<code>SyntaxError</code>	: Dibesarkan saat ada kesalahan dengan sintaks Python.
<code>IndentationError</code>	: Dibesarkan saat indentasi tidak ditentukan dengan benar.
<code>SystemError</code>	: Dibesarkan saat penafsir menemukan masalah internal, namun bila kesalahan ini ditemui juru bahasa Python tidak keluar.
<code>SystemExit</code>	: Dibesarkan saat juru bahasa Python berhenti dengan menggunakan fungsi <code>sys.exit ()</code> . Jika tidak ditangani dalam kode, menyebabkan penafsir untuk keluar.
<code>TypeError</code>	: Dibesarkan saat operasi atau fungsi dicoba yang tidak valid untuk tipe data yang ditentukan.
<code>ValueError</code>	: Dibesarkan ketika fungsi bawaan untuk tipe data memiliki jenis argumen yang valid, namun argumen tersebut memiliki nilai yang tidak valid yang ditentukan.
<code>RuntimeError</code>	: Dibesarkan saat kesalahan yang dihasilkan tidak termasuk dalam kategori apa pun.
<code>NotImplementedError</code>	: Dibesarkan ketika metode abstrak yang perlu diimplementasikan di kelas warisan sebenarnya tidak dilaksanakan.

2.4.10. Membaca Dan Menulis File

Python sudah menyediakan fungsi `open()` untuk membaca dan menulis file. Fungsi ini memiliki dua parameter, yaitu nama file dan mode. Objek file adalah variabel objek yang menampung isi file. Kita bisa melakukan pemrosesan file berkatnya.

Nama file bisa kita isi langsung apabila file-nya terletak dalam satu direktori dengan skrip python. Namun, apabila terletak di direktori yang berbeda, maka kita harus memberikan alamat path file-nya. Ada beberapa mode yang tersedia^[18]:

- “r” : hanya membaca file saja.
- “w” : akses untuk menulis file, jika file sudah ada, maka file akan di replace dan diganti dengan yang baru ditulis.
- “a” : digunakan untuk *append* atau menambah data ke file, artinya jika sudah ada data dalam file, maka akan ditambahkan dan tidak di-replace.
- “r+” : digunakan untuk membaca sekaligus menulis data ke file.

BAB III

PEMBAHASAN

3.1. Analisis Sistem

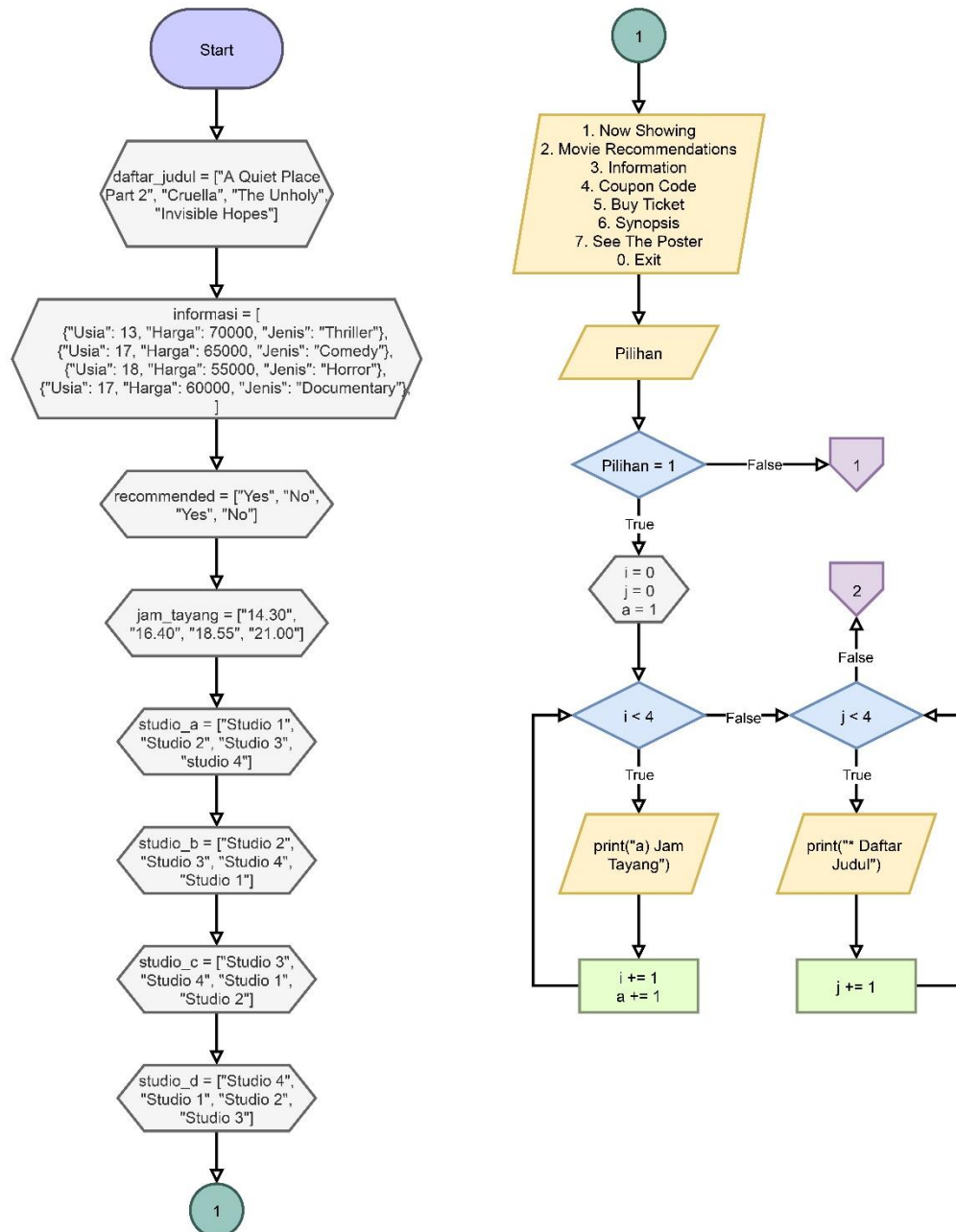
3.1.1. Ringkasan Sistem

FOX ID adalah sebuah aplikasi sederhana yang dirancang sebagai layanan hiburan terdepan di Indonesia yang memberikan pengalaman baru dalam pembelian tiket film yang sedang ditayangkan di bioskop. Dengan FOX ID, pengguna dapat merasakan manfaat dari fitur yang disediakan seperti informasi mengenai film terkini, film yang direkomendasikan oleh FOX ID, sinopsis film yang ditayangkan, poster film, serta pemesanan tiket bioskop dengan mudah, cepat, dan aman.

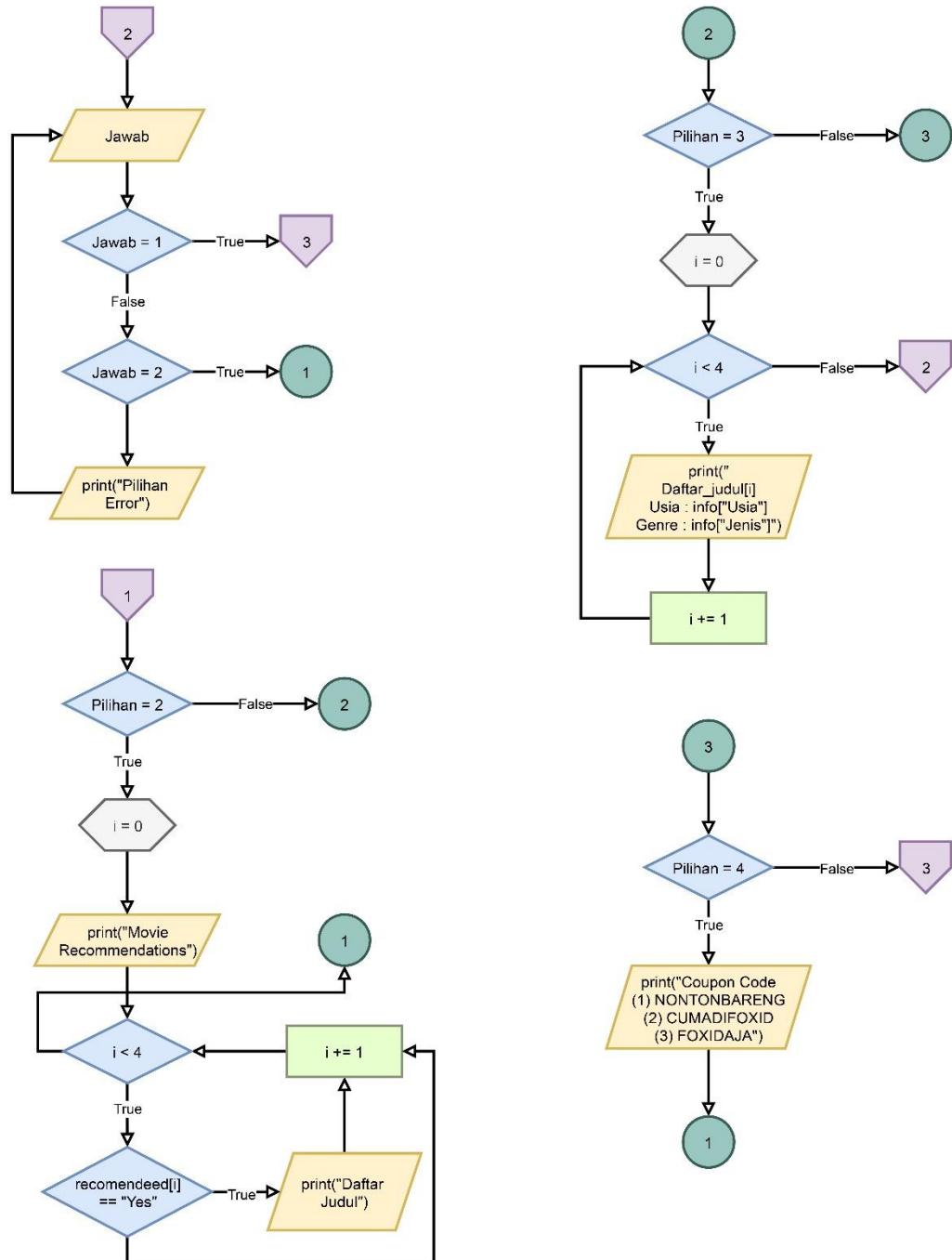
3.1.2. Proses Sistem

FOX ID telah mendata informasi film yang ditayangkan di bioskop-bioskop yang mendukung dari sumber-sumber yang terpercaya. FOX ID menawarkan berbagai macam kode promo yang dapat digunakan dengan syarat dan ketentuan yang berlaku. Selain dapat mempercepat pembelian tiket bioskop, FOX ID mengklaim menjadi penyedia layanan pemesanan tiket dengan fitur paling baik dari yang lainnya.

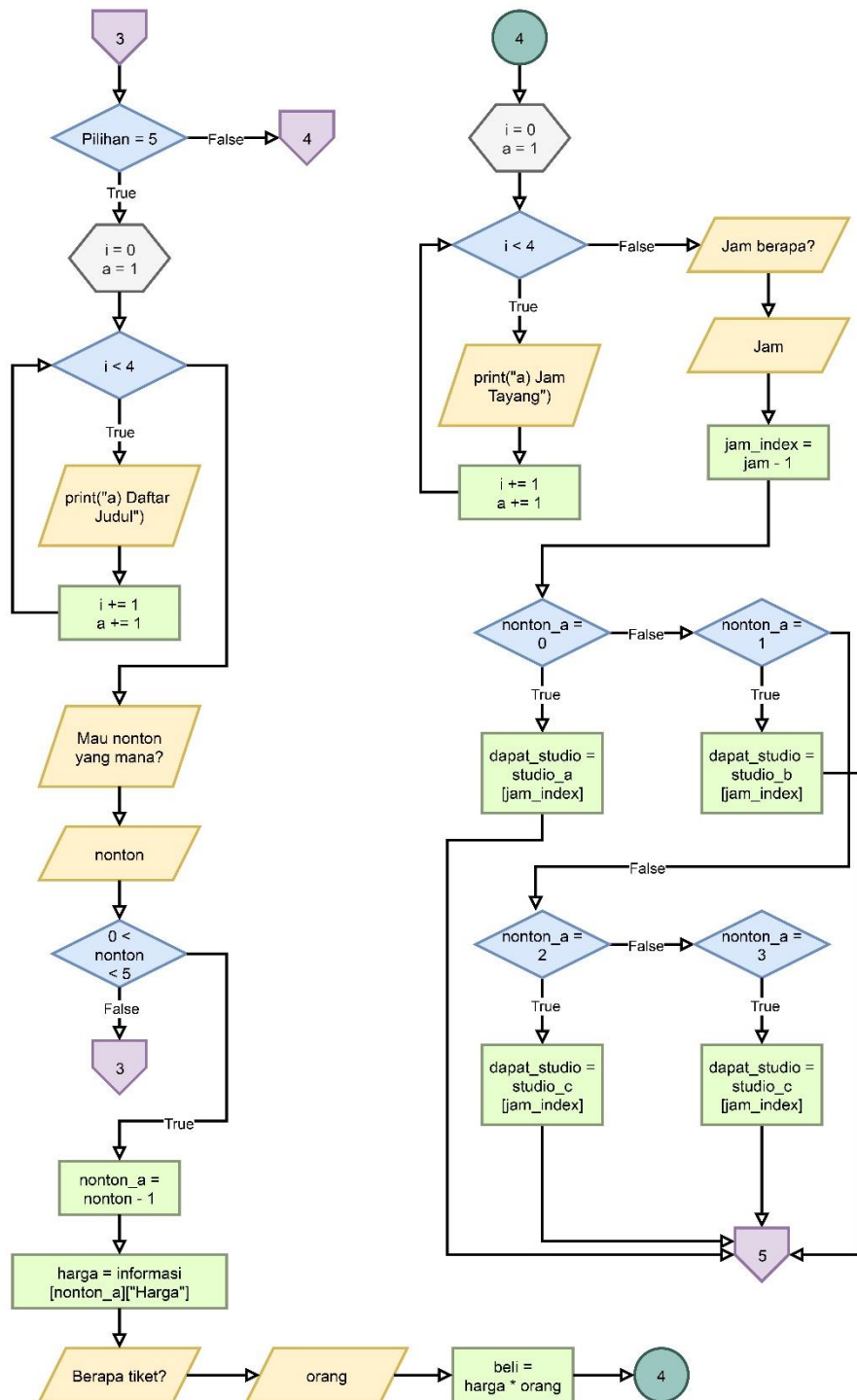
3.2. Flowchart



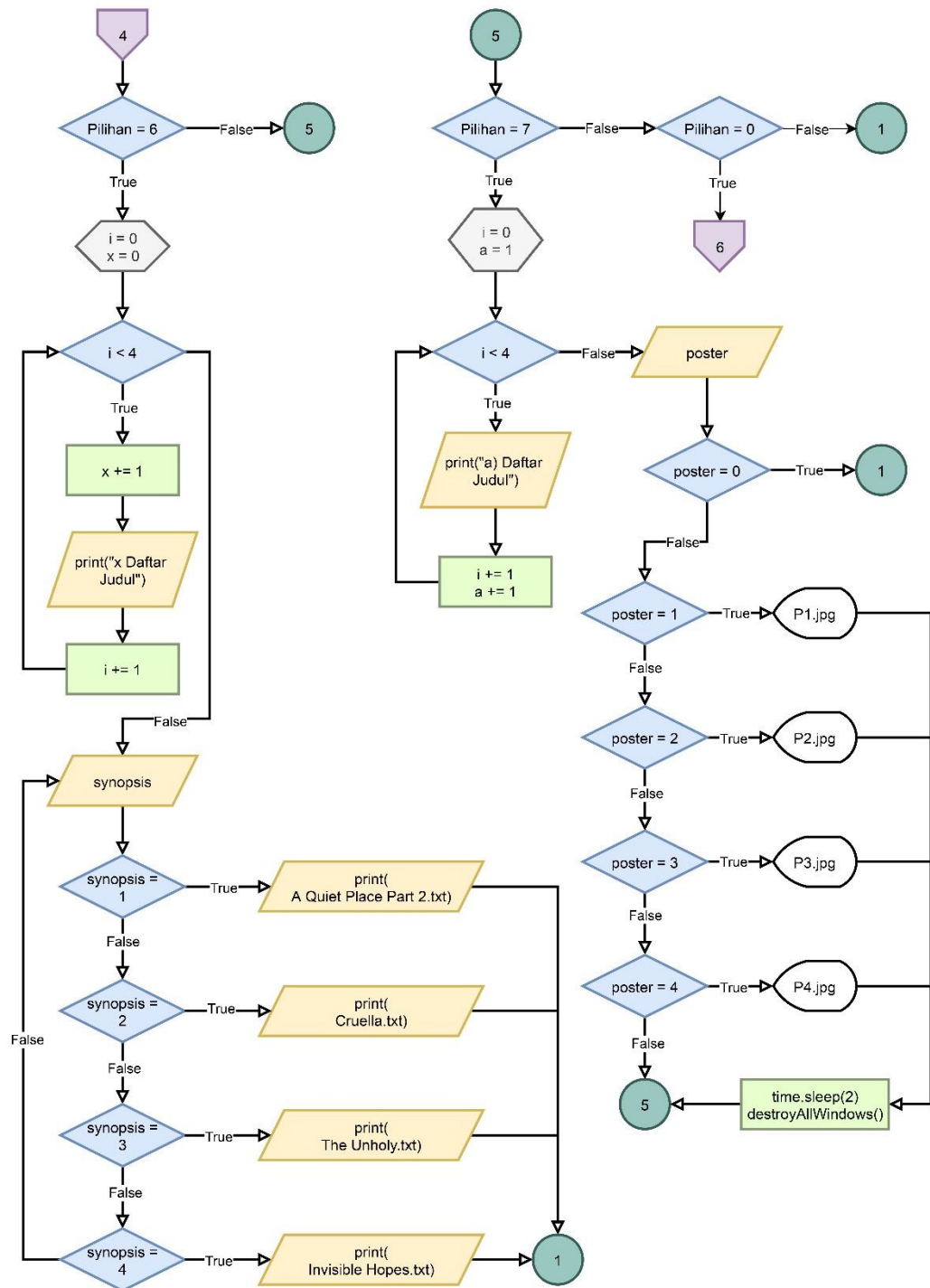
Gambar 3.1 Flowchart FOX ID 1



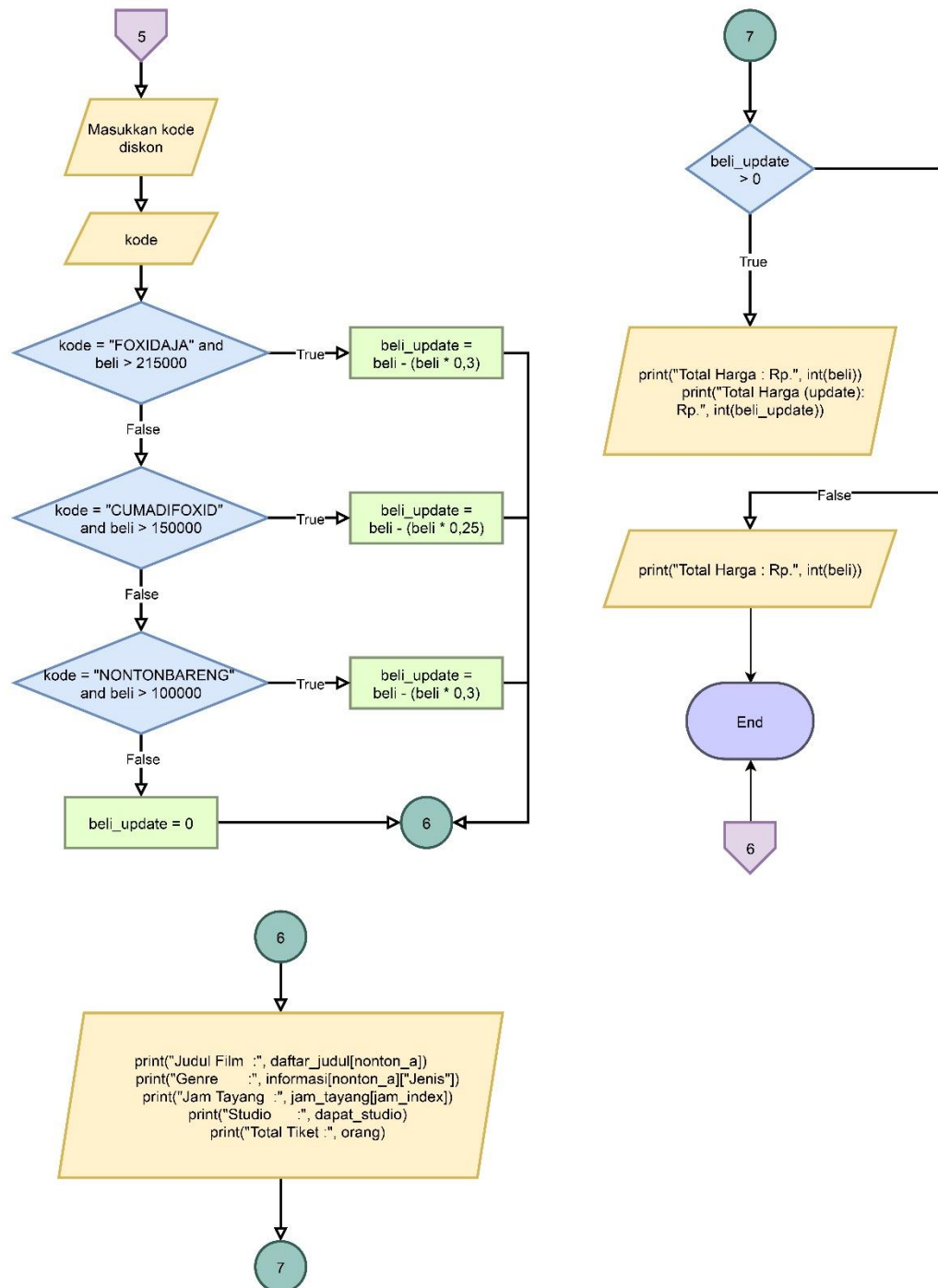
Gambar 3.2 Flowchart FOX ID 2



Gambar 3.3 Flowchart FOX ID 3



Gambar 3.4 Flowchart FOX ID 4



Gambar 3.5 Flowchart FOX ID 5

3.3.1. Deklarasi Awal

```
import time
```

```
import time
import cv2

daftar_judul = ["A Quiet Place Part 2", "Cruella",
"The Unholy", "Invisible Hopes"]

informasi = [
    {"Usia": 13, "Harga": 70000, "Jenis": "Thriller"},
    {"Usia": 17, "Harga": 65000, "Jenis": "Comedy"},
    {"Usia": 18, "Harga": 55000, "Jenis": "Horror"},
    {"Usia": 17, "Harga": 60000, "Jenis": "Documentary"},
]

recommended = ["Yes", "No", "Yes", "No"]

jam_tayang = ["14.30", "16.40", "18.55", "21.00"]

# film A Quiet Place Part 2
studio_a = ["Studio 1", "Studio 2", "Studio 3", "Studio 4"]

# film Cruella
studio_b = ["Studio 2", "Studio 3", "Studio 4", "Studio 1"]

# film The Unholy
studio_c = ["Studio 3", "Studio 4", "Studio 1", "Studio 2"]

# film Invisible Hopes
studio_d = ["Studio 4", "Studio 1", "Studio 2", "Studio 3"]
```

[illegible]

Gambar 3.6 Menu Utama

Halaman menu utama memberikan daftar menu yang dapat dipilih oleh pengguna, terdapat 8 menu yang dapat dipilih oleh pengguna. Scriptnya adalah:

```
# Menu utama yang dapat dipilih oleh pengguna
def menu_utama():
    try:
        print("\n◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆")
        print("1. NOW SHOWING")
        print("2. MOVIE RECOMMENDATIONS")
        print("3. INFORMATION")
        print("4. COUPON CODE")
        print("5. BUY TICKET")
        print("6. SYNOPSIS")
        print("7. SEE THE POSTER")
        print("0. EXIT")
        pilihan = int(input("PILIH : "))

        if pilihan == 1:
            now_showing()
        elif pilihan == 2:
            rec()
        elif pilihan == 3:
            informasi_film()
        elif pilihan == 4:
            code()
        elif pilihan == 5:
            pembelian()
        elif pilihan == 6:
            synopsis()
        elif pilihan == 7:
            movie_poster()
        elif pilihan == 0:
            exit()
        else:
            print("Pilihan Error")
            menu_utama()

    except ValueError:
        print("Pilihan Error")
        menu_utama()
```


3.3.3. Halaman Menu Now Showing

```
1) 14.30
2) 16.40
3) 18.55
4) 21.00

* A Quiet Place Part 2
* Cruella
* The Unholy
* Invisible Hopes
=====
Beli Tiket Sekarang?
1. Ya, 2. Tidak
= █
```

Gambar 3.7 Menu Now Showing

Halaman menu now showing menampilkan daftar judul film yang sedang ditayangkan di bioskop serta memberikan jam tayang film di bioskop tersebut. Scriptnya adalah:

```
# Menampilkan jam tayang dan judul film yang tersedia
def now_showing():
    i = 0
    j = 0
    a = 1

    print()
    while i < len(jam_tayang):
        print(str(a) + ")", jam_tayang[i])
        a += 1
        i += 1
    print()

    while j < len(daftar_judul):
        print("*", daftar_judul[j])
        j += 1

    print("=====")
    milih_beli()
```

3.3.4. Halaman Menu Movie Recommendations

```
=====
==Movie Recommendations==
=====
A Quiet Place Part 2
The Unholy
=====
```

Gambar 3.8 Menu Movie Recommendations

Halaman menu movie recommendations memberikan daftar judul film terbaik dan sedang tayang di bioskop tersebut. Scriptnya adalah:

```
# Menampilkan daftar film yang direkomendasikan
def rec():
    i = 0

    print()
    print("=====")
    print("==Movie Recommendations==")
    print("=====")

    while i < len(recommended):
        if recommended[i] == "Yes":
            print(daftar_judul[i])
            i += 1

    print("=====")
    menu_utama()
```

3.3.5. Halaman Menu Information

```
A Quiet Place Part 2
Usia 13+
Genre : Thriller
=====
Cruella
Usia 17+
Genre : Comedy
=====
The Unholy
Usia 18+
Genre : Horror
=====
Invisible Hopes
Usia 17+
Genre : Documentary
=====
Beli Tiket Sekarang?
1. Ya, 2. Tidak
= █
```

Gambar 3.9 Menu Information

Halaman menu information, halaman ini memberikan informasi seputar film yang sedang tayang di bioskop. Informasi yang dapat dilihat oleh pengguna antara lain judul film, aturan usia penonton, serta jenis film tersebut. Scriptnya adalah:

```
# Menampilkan informasi seputar film
def informasi_film():
    i = 0
    info = len(daftar_judul)

    print()
    while i < len(daftar_judul):
        for info in informasi:
            print(
                daftar_judul[i],
                "\nUsia",
                str(info["Usia"]) + " + \nGenre :",
                str(info["Jenis"]),
                "\n=====")
            i += 1

    milih_beli()
```

3.3.6. Halaman Menu Coupon Code

```
=====Coupon Code=====
(1) NONTONBARENG
(2) CUMADIFOXID
(3) FOXIDAJA
```

Gambar 3.10 Menu Coupon Code

Halaman menu coupon code memberikan informasi mengenai kode-kode yang dapat digunakan oleh pengguna ketika sedang melakukan pemesanan tiket. Syarat dan ketentuan berlaku ketika menggunakan kode-kode tersebut. Scriptnya adalah:

```
# Menampilkan daftar kode kupon yang dapat digunakan
def code():
    print("=====Coupon Code=====")
    print("(1) NONTONBARENG")
    print("(2) CUMADIFOXID")
    print("(3) FOXIDAJA")
    menu_utama()
```

3.3.7. Halaman Menu Synopsis

```
=====
1 A Quiet Place Part 2
2 Cruella
3 The Unholy
4 Invisible Hopes

Synopsis : 3
-----The Unholy-----
Alice, seorang gadis muda tuna rungu yang secara ajaib
dapat mendengar, berbicara, dan menyembuhkan orang sakit
setelah ia mengaku mendapat kunjungan dari Bunda Maria.
Begitu kabar tersebut tersiar dan orang-orang dari wilayah
dekat maupun jauh berbondong-bondong datang untuk menyaksikan
mukjizat-mukjizat yang dibuatnya, seorang jurnalis bereputasi
buruk yang berharap dapat memulihkan kariernya, mendatangi
kota kecil di New England untuk menyelidiki. Namun, ketika
peristiwa-peristiwa kelam mulai meneror di sekitarnya, ia
mulai mempertanyakan apakah fenomena-fenomena yang terjadi
adalah benar berasal dari Bunda Maria, atau malah dari sesuatu
yang jahat.
```

Gambar 3.11 Menu Synopsis

Halaman menu synopsis memberikan sinopsis mengenai film yang sedang tayang di bioskop tersebut. Pengguna dapat memilih ingin membaca sinopsis film yang mereka inginkan yang sedang ditayangkan. Scriptnya adalah:

```
# Menampilkan sinopsis dari film yang dipilih
def synopsis():
    print()
    print("=====")
    i = 0

    while i < len(daftar_judul):
        print(i + 1, daftar_judul[i])
        i += 1

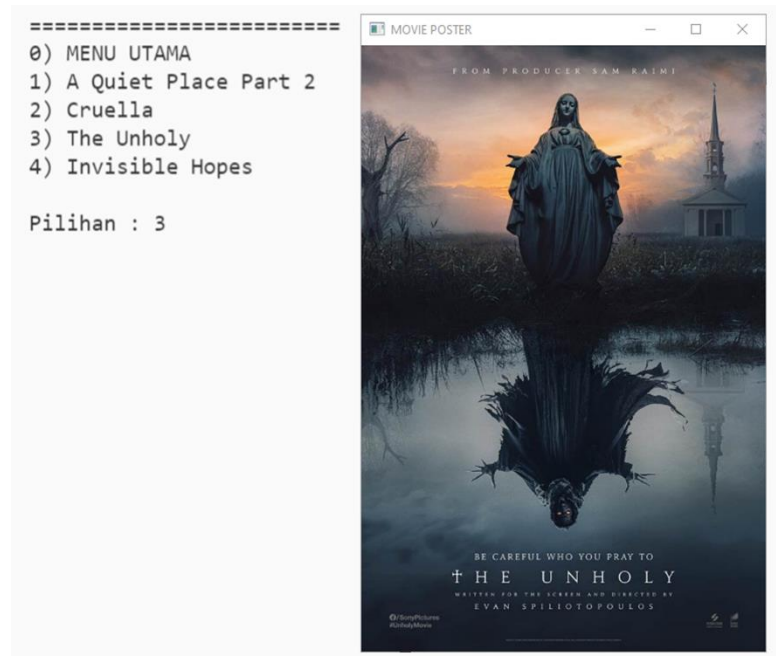
    try:
        print()
        sy = int(input("Synopsis : "))

        if sy == 1:
            with open("D:\Tubes\Quiet Pt 2.txt", "r") as docs:
                print(docs.read())
        elif sy == 2:
            with open("D:\Tubes\Cruella.txt", "r") as docs:
                print(docs.read())
        elif sy == 3:
            with open("D:\Tubes\The Unholy.txt", "r") as docs:
                print(docs.read())
        elif sy == 4:
            with open("D:\Tubes\Hopes.txt", "r") as docs:
                print(docs.read())
        else:
            print("Pilihan Error")
            synopsis()

    except ValueError:
        print("Pilihan Error")
        synopsis()

menu_utama()
```

3.3.8. Halaman Menu Poster



Gambar 3.12 Menu See The Poster

Halaman menu poster memberikan pilihan kepada pengguna ingin melihat poster film apa yang mereka inginkan. Setelah memasukkan pilihan, poster dari film yang dipilih oleh pengguna akan muncul, dan menghilang dalam 2 detik. Scriptnya adalah:

```
# Menampilkan poster film yang dipilih
def movie_poster():
    i = 0
    a = 1
    window_name = "MOVIE POSTER"

    print()
    print("=====")
    print("0) MENU UTAMA")

    while i < len(daftar_judul):
        print(str(a) + ")", daftar_judul[i])
        i += 1
        a += 1
```

```

try:
    print()
    poster = int(input("Pilihan : "))

    if poster == 0:
        menu_utama()
    elif poster == 1:
        old_img = cv2.imread("D:\Tubes\P1.jpg")
    elif poster == 2:
        old_img = cv2.imread("D:\Tubes\P2.jpg")
    elif poster == 3:
        old_img = cv2.imread("D:\Tubes\P3.jpg")
    elif poster == 4:
        old_img = cv2.imread("D:\Tubes\P4.jpg")
    else:
        print("Pilihan Error")
        movie_poster()

except ValueError:
    print("Pilihan Error")
    movie_poster()

img = cv2.resize(old_img, (400, 600))
cv2.imshow(window_name, img)
cv2.setWindowProperty(window_name, cv2.WND_PROP_T
OPMOST, 1)
cv2.waitKey(1)
time.sleep(2)
cv2.destroyAllWindows()
movie_poster()

```

3.3.9. Logika Keputusan

```

# Logika apakah akan melanjutkan pembelian tiket atau
tidak
def milih_beli():
    try:
        print("Beli Tiket Sekarang?")
        print("1. Ya, 2. Tidak")
        jawab = int(input("= "))

        if jawab == 1:
            pembelian()
        elif jawab == 2:
            menu_utama()
        else:
            print("Pilihan Error")
            milih_beli()

    except ValueError:
        print("Pilihan Error")
        milih_beli()

```

3.3.10. Halaman Menu Pembelian

```
1) A Quiet Place Part 2
2) Cruella
3) The Unholy
4) Invisible Hopes

Mau nonton yang mana? 3
Berapa tiket? 5
1) 14.30
2) 16.40
3) 18.55
4) 21.00
Jam berapa? 4
Masukkan Kode Diskon : FOXIDAJA

=====
===informasi pembelian===
=====
Judul Film   : The Unholy
Genre       : Horror
Jam Tayang   : 21.00
Studio      : Studio 2
Total Tiket  : 5
Total Harga  : Rp. 275000
Total Harga (update): Rp. 192500
PS D:\Tubes>
```

Gambar 3.13 Menu Buy Ticket

Halaman menu pembelian memberikan pilihan yang dapat dipilih oleh pengguna seperti judul film yang hendak ditonton, jumlah tiket, jam tayang, serta pengguna dapat memasukkan kode diskon ataupun tidak. Rincian pembelian akan ditampilkan diantaranya judul film, jenis film, jam tayang, studio penayangan, jumlah tiket yang dibeli, serta total harga yang harus dibayarkan oleh pengguna. Scripnya adalah:

```
# Menu pembelian tiket bioskop
def pembelian():
    i = 0
    a = 1
    print()
```



```

while i < len(daftar_judul):
    print(str(a) + " ", daftar_judul[i])
    i += 1
    a += 1

try:
    print()
    nonton = int(input("Mau nonton yang mana? "))

    if 0 < nonton < 5:
        nonton_a = nonton - 1
        harga = informasi[nonton_a]["Harga"]
        orang = int(input("Berapa tiket? "))
        beli = orang * harga
    else:
        print("Pilihan Error")
        pembelian()

except ValueError:
    print("Pilihan Error")
    pembelian()

# Memilih jam tayang dan studio
def studio():
    i = 0
    a = 1

    while i < len(jam_tayang):
        print(str(a) + " ", jam_tayang[i])
        i += 1
        a += 1

    try:
        pilih_jam = int(input("Jam berapa? "))
        jam_index = pilih_jam - 1
        if nonton_a == 0:
            dapat_studio = studio_a[jam_index]
        elif nonton_a == 1:
            dapat_studio = studio_b[jam_index]
        elif nonton_a == 2:
            dapat_studio = studio_c[jam_index]
        elif nonton_a == 3:
            dapat_studio = studio_d[jam_index]
        else:
            print("Pilihan Error")
            studio()

    except ValueError:
        print("Pilihan Error")
        studio()

kode_a = str(input("Masukkan Kode Diskon : "))
kode = kode_a.upper()

```

```

if kode == "FOXIDAJA" and beli > 215000:
    beli_update = beli - (beli * 0.3)
elif kode == "CUMADIFOXID" and beli > 150000:
    beli_update = beli - (beli * 0.25)
elif kode == "NONTONBARENG" and beli > 100000:
    beli_update = beli - (beli * 0.2)
else:
    beli_update = 0

print("\n=====")
print("===informasi pembelian===")
print("=====")
print("Judul Film   :", daftar_judul[nonton_a])
print("Genre       :", informasi[nonton_a]["Jenis"])
print("Jam Tayang   :", jam_tayang[jam_index])
print("Studio       :", dapat_studio)
print("Total Tiket  :", orang)

if beli_update > 0:
    print("Total Harga : Rp.", int(beli))
    print("Total Harga (update): Rp.", int(beli_update))
else:
    print("Total Harga : Rp.", int(beli))

time.sleep(5)
exit()

studio()

```

3.3.11. Logika Mulai

```

print("=====")
print("          FOX ID")
print("=====")
menu_utama()

```

3.4. Dry Run

Berikut ini merupakan tabel pengujian program utama dari sistem FOX ID:

Tabel 3.1 Tabel Pengujian Sistem FOX ID

No.	Judul Film	Banyak Tiket	Jam Tayang	Kode Kupon
1	A Quiet Place Part 2	4	18.55	NONTONBARENG
2	Invisible Hopes	2	14.30	-
3	The Unholy	3	14.30	CUMADIFOXID
4	Cruella	5	21.00	FOXIDAJA
5	The Unholy	1	21.00	-

No.	Studio	Total Harga	Total Harga (Update)
1	Studio 3	Rp 280.000	Rp 224.000
2	Studio 4	Rp 120.000	Rp 120.000
3	Studio 3	Rp 165.000	Rp 123.750
4	Studio 1	Rp 325.000	Rp 227.500
5	Studio 2	Rp 55.000	Rp 55.000

3.5. Konfigurasi Sistem

Hardware : HP Pavilion x360 Convertible Intel® Core™ i5-6200U

Operating System : Microsoft Windows 10 Pro 64-bit (10,0 Build 19042)

Software : Visual Studio Code, Microsoft Word, Microsoft Excel, Draw.io

BAB IV

PENUTUP

4.1. Simpulan

Berdasarkan hasil dari analisa dan pengujian program yang dilakukan, diperoleh kesimpulan sebagai berikut:

1. FOX ID dapat menghadirkan fitur-fitur yang membuat FOX ID dapat menjadi salah satu pilihan terbaik dalam pilihan program pemesanan tiket bioskop.
2. Program yang dijalankan dalam FOX ID berjalan sebagaimana mestinya dan tidak ditemukan error sehingga pengguna akan lebih nyaman dalam menggunakannya.

4.2. Saran

Beberapa saran yang dapat penulis berikan untuk dapat mengembangkan program FOX ID menjadi lebih baik adalah:

1. FOX ID sebaiknya dapat dibuat dengan mengimplementasikan database untuk mempermudah dalam pengolahan data.
2. Sebaiknya FOX ID dapat digunakan secara online agar dapat diimplementasikan di seluruh bioskop di Indonesia.

DAFTAR PUSTAKA

- [1]. Artikel DosenPendidikan (14 Mei 2021). Diakses pada 28 Mei 2021, dari website DosenPendidikan: <https://www.dosenpendidikan.co.id/analisis-sistem/>
- [2]. Artikel Codepolitan. (2021). Diakses pada 28 Mei 2021, dari website CodePolitan: <https://www.codepolitan.com/mengenal-sekilas-jenis-jenis-flowchart-pada-pemrograman-589be38b9417f/>
- [3]. Konsep Sistem. (14 Februari 2018). Diakses pada 28 Mei 2021, dari website Program Studi Teknologi Pendidikan Universitas Pendidikan Indonesia: <http://kurtek.upi.edu/2018/02/14/konsep-sistem/>
- [4]. Artikel GuruPendidikan (16 Maret 2021). Diakses pada 28 Mei 2021, dari website Gurupendidikan: <https://www.gurupendidikan.co.id/elemen-sistem/>
- [5]. Garuda Cyber Indonesia. (16 April 2018). Garuda Cyber Indonesia. Diakses pada 28 Mei 2021, dari website Garuda Cyber Indonesia: <https://garudacyber.co.id/artikel/587-konsep-dasar-pemrograman/>
- [6]. Ngopot. (16 Juli 2020). Pengertian Pemrograman. Diakses pada 28 Mei 2021, dari website Ngopot: <https://www.ngopot.com/2020/07/pengertian-pemrograman.html/>
- [7]. Informatikalogi. (10 April 2017). Pengertian Flowchart Dan Jenis – Jenisnya. Diakses pada 28 Mei 2021, dari website Informatikalogi: <https://informatikalogi.com/pengertian-flowchart-dan-jenis-jenisnya/>
- [8]. Informasi. (2019). Bahasa Pemrograman Python: Pengertian, Sejarah, Kelebihan dan Kekurangannya. Diakses pada 28 Mei 2021, dari website Baktikominfo: https://www.baktikominfo.id/en/informasi/pengetahuan/bahasa_pemrograman_python_pengertian_sejarah_kelebihan_dan_kekurangannya-954/
- [9]. Belajarpython. (2021). Belajarpython – Tipe Data Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/tipe-data-python/>

- [10]. Belajarpython. (2021). Belajarpython – Variabel Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/variabel-python/>
- [11]. Belajarpython. (2021). Belajarpython – Operator Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/operator-python/>
- [12]. Belajarpython. (2021). Belajarpython – Kondisi Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/kondisi-if-else-python/>
- [13]. Belajarpython. (2021). Belajarpython – Perulangan. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/loop-python/>
- [14]. Belajarpython. (2021). Belajarpython – List Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/list-python/>
- [15]. Belajarpython. (2021). Belajarpython – Fungsi Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/fungsi-python/>
- [16]. Belajarpython. (2021). Belajarpython – Modul Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/modul-python/>
- [17]. Belajarpython. (2021). Belajarpython – Exception Python. Diakses pada 29 Mei 2021, dari website Belajarpython.com: <https://belajarpython.com/tutorial/exception-python/>
- [18]. Muhandian, A. (9 Agustus 2017). Belajar Python: Cara Membaca dan Menulis File di Python. Diakses pada 29 Mei 2021, dari website Petanikode.com: <https://www.petanikode.com/python-file/>