

Tugas 3

Berdasarkan dekomposisi, deskripsi, dan flowchart yang telah dibuat dalam tugas-tugas sebelumnya, kami telah membuat program simulasi yang menggambarkan cinema-seat booking system untuk sebuah teater bioskop CGV yang memiliki dua pilihan tipe kursi: regular dan sweetbox.

Source code dapat diakses juga di repository github dengan link : <https://github.com/fairofif/TuBesPengantarKomputasi>

Source code:

```
"""
=====
Kelompok      : 8
Anggota       : - 16520201 | Kirana Shely S
               - 16520251 | Rofif Fairuz Hawary
               - 16520261 | Khalisa Prabhasalma
               - 16520281 | Vixell
Deskripsi Program : Program ini untuk menggambarkan bagaimana sistem ticketing cinema
=====
"""

import os

# untuk menampilkan menu dengan 2 pilihan di Intro section
def intro():
    os.system("cls")
    print("Selamat datang di CGV. Pilih peran kamu!")
    print("1. Waiters\n2. Customers")

# untuk menampilkan menu waiters dengan 3 pilihan
def waitersMenu():
    os.system("cls")
    print("== Waiters Menu ==")
    print("1. Ambil order\n2. Kosongkan Beberapa Seat\n3. Close Order")

# fungsi untuk traversal kondisi seluruh kursi di studio
def printSeat():
    # Kamus:
    # Seat = array 2 dimensi(matriks) yang mencakup koordinat seat cinema
    os.system("cls")
    for j in range(5):
        print(end="          ")
        for i in range(16):
            if i == 3 or i == 11: # ini untuk memberikan celah besar di line tertentu(sebagai jalan pengunjung)S
                if j == 0:
                    print(" K ", end=" ")
                elif j == 1:
                    print(" o ", end=" ")
                elif j == 2:
                    print(" l ", end=" ")
                elif j == 3:
                    print(" o ", end=" ")
                elif j == 4:
                    print(" m ", end=" ")
            else:
                if j == 0:
                    print(" K ", end=" ")
                elif j == 1:
                    print(" o ", end=" ")
                elif j == 2:
                    print(" l ", end=" ")
                elif j == 3:
                    print(" o ", end=" ")
                elif j == 4:
                    print(" m ", end=" ")
            print()
        print(end="          ")
        for i in range(10):
            if i == 3 or i == 11: # ini untuk memberikan celah besar di line tertentu(sebagai jalan pengunjung)
                print(" ", i , end=" ")
            else:
                print(" ", i , end=" ")
        print()
```

```

print(" 10 11      12 13 14 15\n")

for j in range(10):
    print("Baris ", j, end=" ")
    for i in range(16):
        if i == 3 or i == 11:      # ini untuk memberikan celah besar di line tertentu(sebagai jalan pengunjung
)
            print(" ", seat[i][j], end=" ")
        else:
            print(" ", seat[i][j], end=" ")
    print()
print("          ===== LAYAR =====")

# fungsi untuk mengubah display sweetbox menjadi character S
def changeVisualSB(seat):
    # Kamus :
    # seat = array 2 dimensi(matriks) yang mencakup koordinat seat cinema
    for j in range(1,3):
        for i in range(16):
            seat[i][j] = "S"      # seat sweetbox akan diubah menjadi "S" logonya

# fungsi untuk mengamil jumlah orang dalam 1 pesanan
def getAudience(j_org, slot_R, slot_SB, status_slot):
    """=====
    Kamus :
    j_org = int, jumlah orang dalam satu pesanan
    slot_R = int, jumlah kursi kosong reguler
    slot_SB = int, jumlah kursi kosong sweetbox
    status_slot = boolean, sebagai penanda kursi masih muat untuk jumlah j_org atau tidak
    ====="""

    status_slot = False
    j_org = int(input("Jumlah orang: "))
    while j_org < 1 or j_org > 8:
        print("Jumlah orang tidak kurang dari 1 dan tidak lebih dari 8")
        j_org = int(input("Jumlah orang: "))
    if j_org > slot_R and j_org > slot_SB:
        print("Kursi penuh untuk ", j_org, " orang")
        status_slot = False
    else:
        status_slot = True
    return j_org, status_slot

# fungsi untuk memilih pilihan kursi customer
def typeChoice(j_org, type, slot_SB, slot_R):
    """=====
    Kamus :
    j_org = int, jumlah orang dalam satu pesanan
    type = string, berisi jenis kursi pilihan sb atau r
    slot_SB = int, jumlah kursi kosong sweetbox
    slot_R = int, jumlah kursi kosong reguler
    ====="""

    print("Pilih tipe kursimu: ")
    print("Sweet Box (sb)")
    print("Regular (r)")

    status_slot = False
    while status_slot == False:
        type = input("Pilih tipe: ")
        while type != "sb" and type != "r":
            print("Tipe kursi hanya tersedia sb dan r")
            type = input("Pilih tipe: ")
        if type == "sb":      # untuk pilihan sweetbox
            if slot_SB >= j_org:
                status_slot = True
            else:
                print("Jumlah kursi sweetbox tidak cukup untuk ", j_org, " orang")
        else:      # untuk pilihan reguler
            if slot_R >= j_org:
                status_slot = True
            else:
                print("Jumlah kursi reguler tidak cukup untuk ", j_org, " orang")

```

```

    return type, status_slot

# fungsi untuk menghitung jumlah kursi reguler yang kosong
def countSlotR(slot_R):
    # Kamus :
    # slot_R = int, jumlah kursi kosong reguler
    # seat = array 2 dimensi(matriks) yang mencakup koordinat seat cinema
    for j in range(0,10):
        for i in range(16):
            if seat[i][j] == "R":
                slot_R += 1
    print("slot R adalah ", slot_R)
    return slot_R

# fungsi untuk menghitung jumlah kursi sweetbox yang kosong
def countSlotSB(slot_SB):
    # Kamus :
    # slot_SB = slot_SB = int, jumlah kursi kosong sweetbox
    # seat = array 2 dimensi(matriks) yang mencakup koordinat seat cinema
    for j in range(1,3):
        for i in range(16):
            if seat[i][j] == "S":
                slot_SB += 1
    print("slot SB adalah", slot_SB)
    return slot_SB

# fungsi untuk memilih kursi sweetbox
def sweetBoxSeat(j_org, type, rows, column, tagged_col, tagged_row):
    """=====
    Kamus :
    j_org = j_org = int, jumlah orang dalam satu pesanan
    temp = int, sebagai wadah nilai j_org, agar nilai j_org tidak berubah saat digunakan di fungsi lain
    type = string, berisi jenis kursi pilihan sb atau r
    rows = baris pilihan customer
    column = kolom pilihan customer
    tagged_col = list, untuk mengumpulkan kolom kursi yang dibook cust
    tagged_row = list, untuk mengumpulkan baris kursi ynag dibook cust
    seat = array 2 dimensi(matriks) yang mencakup koordinat seat cinema
    ===== """

    temp = j_org    # declare temp sebagai wadah pengganti nilai j_org
    while temp > 0:
        rows = int(input("Baris kursi: "))
        column = int(input("Kolom kursi: "))
        # error handling saat input rows dan column
        while rows < 1 or rows > 2 or column < 0 or column > 15:
            print("baris sweetbox hanya tersedia dari 1-2, kolom sweetbox hanya 0-15")
            rows = int(input("Baris kursi: "))
            column = int(input("Kolom kursi: "))
        # mengganti logo seat pilihan dengan X sebagai tanda booked
        if seat[column][rows] == "S":    # jika kursi pilihan kosong
            if column % 2 == 0:    # jika sweetbox pilihan genap
                seat[column][rows] = "X"
                seat[column+1][rows] = "X"
                # menambahkan kolom dan baris pilihan ke list tagged
                # untuk digunakan saat mencetak tiket
                tagged_col = tagged_col + [column]
                tagged_col = tagged_col + [column+1]
                tagged_row = tagged_row + [rows]
                tagged_row = tagged_row + [rows]
                temp -= 2
            else:    # jika sweetbox pilihan ganjil
                seat[column][rows] = "X"
                seat[column-1][rows] = "X"
                # menambahkan kolom dan baris pilihan ke list tagged
                # untuk digunakan saat mencetak tiket
                tagged_col = tagged_col + [column-1]
                tagged_col = tagged_col + [column]
                tagged_row = tagged_row + [rows]
                tagged_row = tagged_row + [rows]
                temp -= 2
        else:
            print("Kursi sweetbox tsb sudah dibooking")

```

```

    return tagged_col, tagged_row

# fungsi untuk memilih kursi reguler satu per satu
def regulerSeat(j_org, type, rows, column, tagged_col, tagged_row):
    """=====
    Kamus :
    j_org = int, jumlah orang dalam satu pesanan
    temp = int, sebagai wadah nilai j_org, agar nilai j_org tidak berubah saat digunakan di fungsi lain
    type = string, berisi jenis kursi pilihan sb atau r
    rows = baris pilihan customer
    column = kolom pilihan customer
    tagged_col = list, untuk mengumpulkan kolom kursi yang diboook cust
    tagged_row = list, untuk mengumpulkan baris kursi ynag diboook cust
    seat = array 2 dimensi(matriks) yang mencakup koordinat seat cinema
    =====
    """

    temp = j_org      # temp sebagai wadah nilai j_org
    while temp > 0:
        rows = int(input("Baris kursi: "))
        column = int(input("Kolom kursi: "))
        # error handling saat input rows dan column
        while rows < 0 or rows > 9 or column < 0 or column > 15 or rows == 1 or rows == 2:
            print("baris reguler hanya tersedia dari 3-9 dan 0, kolom sweetbox hanya 0-15")
            rows = int(input("Baris kursi: "))
            column = int(input("Kolom kursi: "))
        # menandai seat dengan X sebagai tanda booked
        if seat[column][rows] == "R":
            seat[column][rows] = "X"
            # menambahkan kolom dan baris pilihan ke list tagged
            # untuk digunakan saat mencetak tiket
            tagged_col = tagged_col + [column]
            tagged_row = tagged_row + [rows]
            temp -= 1
        else:
            print("Kursi reguler tsb sudah diboooking")
    return tagged_col, tagged_row

# fungsi untuk melakukan tagihan dan pembayaran
def billing(j_org, type, cost):
    """=====
    Kamus:
    j_org = int, jumlah orang dalam satu pesanan
    type = string, berisi jenis kursi pilihan sb atau r
    cost = int, biaya yang perlu dibayar
    money = int, masukan uang dari customer
    =====
    """

    if type == "sb":
        if j_org%2 == 0: # jumlah orang genap
            cost = int(110000 * (j_org/2))
        else:           # jumlah orang ganjil
            cost = int(110000 * ((j_org+1)/2))
    else:
        cost = int(j_org * 50000)
    print("Total tagihan : " + str(cost))
    money = int(input("Masukkan uang : "))
    # error handling jika uang kurang
    while money < cost:
        money = int(input("Nilai uang kurang. Harap masukkan ulang nominal.\nMasukkan uang : "))
    print("Ticket telah terbayar!")
    print("Kembalian : " + str(money - cost))
    print("Grab your ticket!")
    print()

# fungsi untuk mencetak tiket
def ticket(j_org, type, tagged_col, tagged_row):
    """
    Kamus:
    j_org = int, jumlah orang dalam satu pesanan
    type = string, berisi jenis kursi pilihan sb atau r
    tagged_col = list, untuk mengumpulkan kolom kursi yang diboook cust
    tagged_row = list, untuk mengumpulkan baris kursi ynag diboook cust
    index_tag = sebagai indeks list tagged yang diincrement sebanyak +2
    """

```

```

"""

if type == "sb":      # type sweetbox
    index_tag = 0     # declare index untuk list tagged = 0
    for i in range((j_org+1)//2): # karena type sb maka ditambah 1 dan dibagi 2 dan dibulatkan ke bawah
        print("-----")
        print("|  ", end="")
        print("Ticket #", i+1, "          |")
        print("|  ", end="")
        print("[SweetBox]          |")
        print("|  ", end="")
        print("Seat :          |")
        print("| (Kolom,Baris) : (", tagged_col[index_tag],",", tagged_row[index_tag],")  |")
        print("-----")
        index_tag += 2 # karena type sb maka diincrement sebanyak +2
    else: # type reguler
        for i in range(j_org): # type reguler tidak perlu menggunakan index pengganti
            print("-----")
            print("|  ", end="")
            print("Ticket #", i+1, "          |")
            print("|  ", end="")
            print("[Reguler]          |")
            print("|  ", end="")
            print("Seat :          |")
            print("| (Kolom,Baris) : (", tagged_col[i],",", tagged_row[i],")  |")
            print("-----")

# fungsi untuk mengosongkan seat pilihan
def deleteSeat(seat):
    """
    Kamus:
    seat = seat = array 2 dimensi(matriks) yang mencakup koordinat seat cinema
    column = int, indeks seat kolom
    rows = int, indeks seat baris
    """
    printSeat()
    print("Pilih nomor seat yang akan dikosongkan")
    column = int(input("Kolom : "))
    # error handling saat input kolom
    while column < 0 or column > 15:
        column = int(input("Masukan salah!\nKolom : "))
    rows = int(input("Baris : "))
    # error handling saat input baris
    while rows < 0 or rows > 9:
        rows = int(input("Masukan salah!\nBaris : "))
    if rows == 1 or rows == 2: # kondisi baris yang dipilih adalah sweetbox
        if seat[column][rows] == "S":
            print("Seat tersebut kosong")
        else: # kondisi seat = "X"
            if column%2 == 0:
                seat[column][rows] = "S"
                seat[column+1][rows] = "S"
            else:
                seat[column][rows] = "S"
                seat[column-1][rows] = "S"
            print("Seat berhasil dikosongkan")
            printSeat()
    else: # untuk reguler
        if seat[column][rows] == "R":
            printSeat()
            print("Seat sudah kosong")
        else: # kondisi "X" status booked
            seat[column][rows] = "R"
            printSeat()
            print("Seat berhasil dikosongkan")

#===== M A I N =====#
seat = [["R" for j in range(10)] for i in range(16)] # membuat default seat
changeVisualSB(seat) # merubah visual sweet box menjadi char "SB"

close = False
while close == False:
    intro()

```

```

pilihan_intro = int(input("Pilih Peran (1 atau 2): "))
# eror handling jika masukan tidak sesuai pilihan
while pilihan_intro != 1 and pilihan_intro != 2:
    pilihan_intro = int(input("Masukan salah!\nPilih Peran (1 atau 2): "))

if pilihan_intro == 2: # sebagai customer
    selesai_order = False
    while selesai_order == False: # looping hingga customer stop order
        #=====
=====
        # semua variable yang bersifat akan digunakan terus menerus dan berubah2 nilainya
        # di declare terlebih dahulu ke nilai yang salah sehingga dapat terus digunakan ke dalam fungsi fungsi
        # yang telah ada, karena akan terjadi looping yang akan menggunakan fungsi secara terus menerus
        j_org = 0
        type = "empty"
        slot_R = 0
        slot_SB = 0
        status_slot = False
        rows = -1
        column = -1
        cost = 0
        tagged_col = []
        tagged_row = []
        #=====
=====

        printSeat()
        slot_R = countSlotR(slot_R)
        slot_SB = countSlotSB(slot_SB)
        j_org, status_slot = getAudience(j_org, slot_R, slot_SB, status_slot)
        type, status_slot = typeChoice(j_org,type, slot_SB, slot_R)

        if status_slot == True:
            if type == "sb":
                tagged_col, tagged_row = sweetBoxSeat(j_org,type, rows, column, tagged_col, tagged_row)
                printSeat()
                billing(j_org,type,cost)
                ticket(j_org,type,tagged_col,tagged_row)
            else: # type = r
                tagged_col, tagged_row = regulerSeat(j_org, type, rows, column, tagged_col, tagged_row)
                printSeat()
                billing(j_org,type,cost)
                ticket(j_org,type,tagged_col,tagged_row)

        pick = input("Ingin menambah pesanan? y/t:")
        # error handling
        while pick != "t" and pick != "y":
            pick = input("Ulangi masukan: ")
        if pick == "t":
            selesai_order = True

else: # sebagai waiters

    # sebelum masuk menu harus log in terlebih dahulu
    id = str(input("ID : "))
    password = str(input("Password : "))
    while id != "waiters" or password != "inipassword":
        print("ID atau password salah.")
        id = str(input("ID : "))
        password = str(input("Password : "))

    waitersMenu()
    pilihan_waiters = int(input("Masukan pilihan menu : "))
    # error handling
    while pilihan_waiters < 1 or pilihan_waiters > 3:
        pilihan_waiters = int(input("Masukan salah!\nMasukan pilihan menu : "))

    if pilihan_waiters == 2: # memilih kosongkan seat
        stop_delete = False
        while stop_delete == False:
            deleteSeat(seat)
            pick = input("Ingin kosongkan seat lain? (y/t) : ")
            # error handling
            while pick != "y" and pick != "t":

```

```

        pick = input("Masukan salah!\nIngin kosongkan seat lain? (y/t) : ")
        if pick == "t":
            stop_delete = True
    elif pilihan_waiters == 1: # ambil orderan
        selesai_order = False
        while selesai_order == False: # looping hingga stop order
            #=====
=====
            # semua variable yang bersifat akan digunakan terus menerus dan berubah2 nilainya
            # di declare terlebih dahulu ke nilai yang salah sehingga dapat terus digunakan ke dalam fungsi fungsi

            # yang telah ada, karena akan terjadi looping yang akan menggunakan fungsi secara terus menerus
            j_org = 0
            type = "empty"
            slot_R = 0
            slot_SB = 0
            status_slot = False
            rows = -1
            column = -1
            cost = 0
            tagged_col = []
            tagged_row = []
            #=====
=====

            printSeat()
            slot_R = countSlotR(slot_R)
            slot_SB = countSlotSB(slot_SB)
            j_org, status_slot = getAudience(j_org, slot_R, slot_SB, status_slot)
            type, status_slot = typeChoice(j_org,type, slot_SB, slot_R)

            if status_slot == True:
                if type == "sb":
                    tagged_col, tagged_row = sweetBoxSeat(j_org,type, rows, column, tagged_col, tagged_row)
                    printSeat()
                    billing(j_org,type,cost)
                    ticket(j_org,type,tagged_col,tagged_row)
                else: # type = r
                    tagged_col, tagged_row = regularSeat(j_org, type, rows, column, tagged_col, tagged_row)
                    printSeat()
                    billing(j_org,type,cost)
                    ticket(j_org,type,tagged_col,tagged_row)

            pick = input("Ingin menambah pesanan? y/t:")
            # error handling
            while pick != "t" and pick != "y":
                pick = input("Ulangi masukan: ")
            if pick == "t":
                selesai_order = True

        else: # pilihan 3, close order cinema
            close = True

```