

**LAPORAN RESPONSI
STRUKTUR DATA PRAKTIK
PERTEMUAN KE-14
(SISTEM PEMBELIAN TIKET KONSER)**

Dosen Pengampu : Sri Wulandari, S.Kom., M.Cs.



REGITA CAHYA ARRAHMA

5220411359

**PROGRAM STUDI INFORMATIKA FAKULTAS SAINS DAN
TEKNOLOGI UNIVERSITAS TEKNOLOGI YOGYAKARTA
2023**

A. Penjelasan Program

Sistem Pembelian Tiket Konser merupakan program sistem tiket yang diimplementasikan menggunakan Binary Search Tree, Queue, dan Graphical user interface (GUI) dengan bahasa pemrograman Python.

Terdapat beberapa class pada program ini, yaitu:

1. Class Customer
Berfungsi untuk mewakili pelanggan dengan email, nama, dan alamat mereka.
2. Class PembelianTiket:
Merupakan pembelian tiket dengan detail seperti pelanggan, hari konser, jenis tiket, jumlah tiket, dan harga tiket.
3. Class Transaksi:
Merupakan transaksi dengan metode pembayaran.
4. Class BSTNode:
Berfungsi untuk mewakili simpul dalam pohon pencarian biner (BST) yang digunakan untuk menyimpan pembelian tiket. Setiap node memiliki kunci (subtotal) dan nilai (PembelianTiket).
5. Class BST:
Berfungsi untuk menerapkan pohon pencarian biner (BST) untuk menyimpan pembelian tiket berdasarkan subtotalnya.
6. Class QueueNode:
Merupakan node dalam antrian yang digunakan untuk menyimpan pembayaran.
7. Class Queue:
Berfungsi untuk menerapkan antrean untuk menyimpan data pembayaran sesuai urutan penerimaannya.
8. Class SistemTiket:
Berfungsi untuk merepresentasikan sistem ticketing dan berisi metode untuk membeli tiket, melakukan pembayaran, mengecek ketersediaan tiket, dan menampilkan menu utama. Itu menggunakan kelas lain seperti Pelanggan, PembelianTiket, Transaksi, BST, dan Antrian untuk mengelola data dan operasi yang terkait dengan tiket dan transaksi.

Graphical user interface (GUI) digunakan untuk menampilkan menu utama dan beberapa pemberitahuan dari sistem ticketing ini. Untuk menggunakan program sistem pembelian tiket konser ini, kita dapat memanggil metode `main_menu()` untuk memulai dan menjalankan program. Dengan hal tersebut, akan menampilkan menu di mana customer dapat memilih opsi untuk membeli tiket, melakukan pembayaran, memeriksa ketersediaan tiket, atau keluar dari program.

Link video : <https://youtu.be/UhpnCBShO4U>

Link kode program :

<https://colab.research.google.com/drive/17WvDZkkgYTtLvWdN9MtlBBjPYNwgEp5?usp=sharing>

B. Kode Program

```
import os
import datetime
now=datetime.datetime.now()
hari = now.strftime("%A")
import tkinter as tk
from tkinter import messagebox
from PIL import Image, ImageTk

class Customer:
    def __init__(self, email, nama, alamat):
        self.email = email
        self.nama = nama
        self.alamat = alamat

class PembelianTiket:
    def __init__(self, customer, hari, jenis_tiket, jumlah_tiket,
harga_tiket):
        self.customer = customer
        self.hari = hari
        self.jenis_tiket = jenis_tiket
        self.jumlah_tiket = jumlah_tiket
        self.harga_tiket = harga_tiket

    def subtotal(self):
        return self.jumlah_tiket * self.harga_tiket

class Transaksi:
    def __init__(self):
        self.metode_pembayaran = ""

    def set_metode_bayar(self, method):
        self.metode_pembayaran = method

class BSTNode:
    def __init__(self, key, value):
        self.key = key
        self.value = value
        self.left = None
        self.right = None

class BST:
    def __init__(self):
        self.root = None

    def insert(self, key, value):
        new_node = BSTNode(key, value)
```

```

    if self.root is None:
        self.root = new_node
        return True

    temp = self.root
    while temp:
        if key == temp.key:
            return False
        elif key < temp.key:
            if temp.left is None:
                temp.left = new_node
                return True
            temp = temp.left
        else:
            if temp.right is None:
                temp.right = new_node
                return True
            temp = temp.right

def search(self, key):
    temp = self.root
    while temp:
        if key == temp.key:
            return temp.value
        elif key < temp.key:
            temp = temp.left
        else:
            temp = temp.right
    return None

def find_min(self):
    if self.root is None:
        return None
    temp_node = self.root
    while temp_node.left is not None:
        temp_node = temp_node.left
    return temp_node

def delete_min(self):
    if self.root is None:
        return None
    if self.root.left is None:
        min_node = self.root
        self.root = self.root.right
        return min_node
    parent_node = self.root
    temp_node = self.root.left
    while temp_node.left is not None:

```

```

        parent_node = temp_node
        temp_node = temp_node.left
        parent_node.left = temp_node.right
        return temp_node

class QueueNode:
    def __init__(self, data):
        self.data = data
        self.next = None

class Queue:
    def __init__(self):
        self.front = None
        self.rear = None

    def is_empty(self):
        return self.front is None

    def enqueue(self, data):
        new_node = QueueNode(data)
        if self.is_empty():
            self.front = self.rear = new_node
        else:
            self.rear.next = new_node
            self.rear = new_node

    def dequeue(self):
        if self.is_empty():
            return None
        else:
            removed_node = self.front
            self.front = self.front.next
            if self.front is None:
                self.rear = None
            return removed_node.data

class SistemTiket():
    def __init__(self):
        self.bst = None
        self.ticket_data = None
        self.Transaksi = Transaksi()
        self.jenis_tiket()
        self.customer_data = {}

    def jenis_tiket(self):
        self.bst = BST()
        self.ticket_data = {
            "Sabtu, 14 Desember 2024 (6:30 PM)": {

```

```

        "CAT 4 Seating": {"capacity": 1000, "price": 150000},
        "CAT 3 Seating": {"capacity": 500, "price": 250000},
        "CAT 2 Seating": {"capacity": 300, "price": 500000},
        "CAT 1 Seating": {"capacity": 200, "price": 1000000}
    },
    "Minggu, 15 Desember 2024 (6:30 PM)": {
        "CAT 4 Seating": {"capacity": 800, "price": 100000},
        "CAT 3 Seating": {"capacity": 400, "price": 200000},
        "CAT 2 Seating": {"capacity": 200, "price": 400000},
        "CAT 1 Seating": {"capacity": 100, "price": 800000}
    }
}

def main_menu(self):
    root = tk.Tk()

    gui_width = 400
    gui_height = 600
    root.geometry(f"{gui_width}x{gui_height}")
    root.resizable(False, False)
    root.title("PEMBELIAN TIKET KONSER")

    bg_image = Image.open("konser.png")
    bg_image = bg_image.resize((gui_width, gui_height),
Image.ANTIALIAS)
    bg_photo = ImageTk.PhotoImage(bg_image)
    bg_label = tk.Label(root, image=bg_photo)
    bg_label.place(x=0, y=0, relwidth=1, relheight=1)

    def on_purchase_ticket():
        root.destroy()
        self.beli_tiket()

    def on_payment():
        root.destroy()
        self.bayar()

    def on_cek_kuota():
        root.destroy()
        self.cek_tiket()

    def on_exit():
        root.destroy()
        messagebox.showinfo("Info", "Terima kasih telah menggunakan
layanan kami!")

    purchase_ticket_button = tk.Button(root, text="Pembelian Tiket",
bg="black", fg="white", command=on_purchase_ticket)

```

```

        purchase_ticket_button.pack()
        purchase_ticket_button.place(x=155, y=270)

        payment_button = tk.Button(root, text="Pembayaran Tiket",
        bg="black", fg="white", command=on_payment)
        payment_button.pack()
        payment_button.place(x=150, y=305)

        exit_button = tk.Button(root, text="Cek Kuota Tiket", bg="black",
        fg="white", command=on_cek_kuota)
        exit_button.pack()
        exit_button.place(x=156, y=340)

        exit_button = tk.Button(root, text="Exit Program", bg="black",
        fg="white", command=on_exit)
        exit_button.pack()
        exit_button.place(x=163, y=375)
        root.mainloop()

def beli_tiket(self):
    os.system("cls")
    print("-".center(60, "-"))
    print(" BELI TIKET ".center(60, "="))
    print("-".center(60, "-"))
    email = input("Masukkan alamat email : ")
    nama = input("Masukkan nama : ")
    alamat = input("Masukkan alamat : ")
    customer = Customer(email, nama, alamat)
    print("-".center(60, "-"))
    print(" Jadwal Konser ".center(60, "="))
    print("-".center(60, "-"))
    for i, hari in enumerate(self.ticket_data.keys()):
        print(f"{i+1}. {hari}")
    print("-".center(60, "-"))
    pilihan_hari = int(input("Masukkan pilihan : "))
    if pilihan_hari == 1:
        print("-".center(60, "-"))
        list_hari = list(self.ticket_data.keys())
        pilih_hari = list_hari[pilihan_hari - 1]
        print(" Jenis Tiket ".center(60, "-"))
        print("-".center(60, "-"))
        tiket_list = list(self.ticket_data[pilih_hari].keys())
        for i, tiket in enumerate(tiket_list):
            print(f"{i+1}. {tiket}")
        print("-".center(60, "-"))
        pilihan_tiket = int(input("Masukkan pilihan : "))
        print("-".center(60, "-"))
        pilih_tiket = tiket_list[pilihan_tiket - 1]

```

```

        jumlah_tiket = int(input("Masukkan jumlah tiket yang ingin
dibeli : "))
        harga_tiket =
self.ticket_data[pilih_hari][pilih_tiket]["price"]
        subtotal = jumlah_tiket * harga_tiket
        tiket_beli = PembelianTiket(customer, pilih_hari, pilih_tiket,
jumlah_tiket, harga_tiket)
        print("-".center(60, "-"))
        messagebox.showinfo("Info","Tiket berhasil ditambahkan!")
        print("Subtotal: Rp", subtotal)
        print("-".center(60, "-"))
        self.bst.insert(subtotal, tiket_beli)

    elif pilihan_hari == 2:
        print("-".center(60, "-"))
        list_hari = list(self.ticket_data.keys())
        pilih_hari = list_hari[pilihan_hari - 1]
        print(" Jenis Tiket ".center(60, "-"))
        print("-".center(60, "-"))
        tiket_list = list(self.ticket_data[pilih_hari].keys())
        for i, tiket in enumerate(tiket_list):
            print(f"{i+1}. {tiket}")
        print("-".center(60, "-"))
        pilihan_tiket = int(input("Masukkan pilihan : "))
        print("-".center(60, "-"))
        pilih_tiket = tiket_list[pilihan_tiket - 1]
        jumlah_tiket = int(input("Masukkan jumlah tiket yang ingin
dibeli : "))
        harga_tiket =
self.ticket_data[pilih_hari][pilih_tiket]["price"]
        subtotal = jumlah_tiket * harga_tiket
        tiket_beli = PembelianTiket(customer, pilih_hari, pilih_tiket,
jumlah_tiket, harga_tiket)
        print("-".center(60, "-"))
        messagebox.showinfo("Info","Tiket berhasil ditambahkan!")
        print("Subtotal: Rp", subtotal)
        print("-".center(60, "-"))
        self.bst.insert(subtotal, tiket_beli)

    else:
        print("-".center(60, "-"))
        messagebox.showinfo("Info","Pilihan tidak valid. Silakan coba
lagi!")
        print("-".center(60, "-"))
        os.system('pause')
        self.main_menu()

def cek_tiket(self):

```



```

os.system("cls")
print("-".center(50, "-"))
print("TIKET TERSEDIA ".center(50, "="))
print("-".center(50, "-"))
print("Jadwal Konser ".center(50, "="))
print("-".center(50, "-"))
for i, hari in enumerate(self.ticket_data.keys()):
    print(f"{i+1}. {hari}")
print("-".center(50, "-"))
pilihan_hari = int(input("Masukkan pilihan : "))
print("-".center(50, "-"))
if pilihan_hari == 1:
    list_hari = list(self.ticket_data.keys())
    pilih_hari = list_hari[pilihan_hari - 1]

    print("Daftar Tiket Tersedia ".center(50, "-"))
    print("-".center(50, "-"))
    tiket_list = list(self.ticket_data[pilih_hari].keys())
    for tiket in tiket_list:
        kapasitas = self.ticket_data[pilih_hari][tiket]["capacity"]
        print(f"{tiket}: {kapasitas} tiket tersedia")
    print("-".center(50, "-"))
elif pilihan_hari == 2:
    list_hari = list(self.ticket_data.keys())
    pilih_hari = list_hari[pilihan_hari - 1]

    print("Daftar Tiket Tersedia ".center(50, "-"))
    print("-".center(50, "-"))
    tiket_list = list(self.ticket_data[pilih_hari].keys())
    for tiket in tiket_list:
        kapasitas = self.ticket_data[pilih_hari][tiket]["capacity"]
        print(f"{tiket}: {kapasitas} tiket tersedia")
    print("-".center(50, "-"))
else:
    print("-".center(50, "-"))
    messagebox.showinfo("Info", "Pilihan tidak valid. Silakan coba lagi!")
    print("-".center(50, "-"))
    self.cek_tiket()
os.system('pause')
self.main_menu()

def metode_bayar(self):
    os.system("cls")
    print("-".center(70, "-"))
    print("Metode Pembayaran ".center(70, "="))
    print("-".center(70, "-"))
    print("1. Transfer Bank")

```

```

        print("2. Kartu Kredit")
        print("3. Go Pay")
        print("-".center(70, "-"))
        pilihan_metode = int(input("Masukkan pilihan : "))
        print("-".center(70, "-"))
        if pilihan_metode == 1:
            self.Transaksi.set_metode_bayar("Transfer Bank")
        elif pilihan_metode == 2:
            self.Transaksi.set_metode_bayar("Kartu Kredit")
        elif pilihan_metode == 3:
            self.Transaksi.set_metode_bayar("Go Pay")
        else:
            print("-".center(70, "-"))
            messagebox.showinfo("Info", "Pilihan tidak valid. Silakan coba
lagi!")
            print("-".center(70, "-"))
            os.system("pause")

    def cetak_tiket(self, tiket_beli):
        print("-".center(70, "-"))
        print("TIKET TREASURE CONCERT IN YOGKYAKARTA ".center(70, "="))
        print("-".center(70, "-"))
        print("Hari/Tanggal Pembelian :", hari, ",", now)
        print("Nama :", tiket_beli.customer.nama)
        print("Alamat :", tiket_beli.customer.alamat)
        print("Email :", tiket_beli.customer.email)
        print("Hari Pertunjukan :", tiket_beli.hari)
        print("Jenis Tiket :", tiket_beli.jenis_tiket)
        print("Jumlah Tiket :", tiket_beli.jumlah_tiket)
        print("Harga Tiket : Rp.", tiket_beli.harga_tiket)
        print("Subtotal : Rp.", tiket_beli.subtotal())
        print("Metode Pembayaran :", self.Transaksi.metode_pembayaran)
        print("-".center(70, "-"))
        messagebox.showinfo("Info", "Enjoy The Concert TEUME!")
        print("-".center(70, "-"))
        os.system('pause')
        self.main_menu()

    def bayar(self):
        os.system("cls")
        print("-".center(60, "-"))
        print("TAGIHAN PEMBAYARAN ".center(60, "="))
        print("-".center(60, "-"))
        if self.bst.root is None:
            messagebox.showinfo("Info", "Tidak ada tiket untuk dibayar!")
            print("-".center(60, "-"))
            os.system("pause")

```

```

        self.main_menu()
        return
    subtotal = self.bst.find_min().key
    tiket_beli = self.bst.find_min().value

    print("-".center(60, "-"))
    print(" Rincian Pembelian Tiket ".center(60, "-"))
    print("-".center(60, "-"))
    print("Nama            :", tiket_beli.customer.nama)
    print("Alamat           :", tiket_beli.customer.alamat)
    print("Email            :", tiket_beli.customer.email)
    print("Hari Pertunjukan :", tiket_beli.hari)
    print("Jenis Tiket       :", tiket_beli.jenis_tiket)
    print("Jumlah Tiket      :", tiket_beli.jumlah_tiket)
    print("Harga Tiket       : Rp.", tiket_beli.harga_tiket)
    print("Subtotal         : Rp.", subtotal)
    print("-".center(60, "-"))

    pilihan_bayar = input("Apakah Anda ingin melakukan pembayaran?
(y/n): ")
    print("-".center(60, "-"))
    if pilihan_bayar.lower() == "y":
        self.metode_bayar()
        self.tiket_data[tiket_beli.hari][tiket_beli.jenis_tiket]["capa
city"] -= tiket_beli.jumlah_tiket
        messagebox.showinfo("Info", "Pembayaran Berhasil!")
        print("-".center(70, "-"))
        self.bst.delete_min()
        print("\n")
        self.cetak_tiket(tiket_beli)
    elif pilihan_bayar.lower() == "n":
        messagebox.showinfo("Info", "Pembayaran dibatalkan!")
        self.main_menu()
    else:
        print("-".center(60, "-"))
        messagebox.showinfo("Info", "Input tidak valid. Pembayaran
dibatalkan!")
        print("-".center(60, "-"))
        self.main_menu()

if __name__ == "__main__":
    sistem_tiket = SistemTiket()
    sistem_tiket.main_menu()

```

C. Hasil Output

Halaman awal adalah Menu utama. Customer dapat memilih menu-menu tersebut. Customer dapat melihat kuota tiket terlebih dahulu pada menu ke-3



Program akan menampilkan ketersediaan kuota tiket sebelum adanya aktivitas pembelian tiket.

```
----- TIKET TERSEDIA -----  
----- Jadwal Konser -----  
1. Sabtu, 14 Desember 2024 (6:30 PM)  
2. Minggu, 15 Desember 2024 (6:30 PM)  
-----  
Masukkan pilihan : 1  
----- Daftar Tiket Tersedia -----  
CAT 4 Seating: 1000 tiket tersedia  
CAT 3 Seating: 500 tiket tersedia  
CAT 2 Seating: 300 tiket tersedia  
CAT 1 Seating: 200 tiket tersedia  
-----  
Press any key to continue . . .
```

Setelah itu customer bisa kembali ke menu utama.

Ketika customer memilih menu Pembayaran tiket maka akan muncul tampilan seperti dibawah ini.

```
----- BELI TIKET -----  
-----  
Masukkan alamat email : rgtama14@gmail.com  
Masukkan nama : Regita Cahya Arrahma  
Masukkan alamat : Pamarican, Ciamis
```

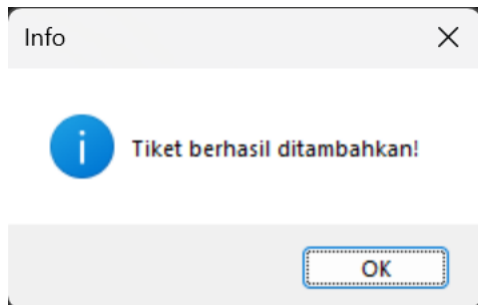
Customer diminta untuk mengisi data diri untuk syarat pembelian tiket. Setelah mengisi data tersebut customer akan diminta untuk memilih jadwal konser.

```
----- Jadwal Konser -----  
-----  
1. Sabtu, 14 Desember 2024 (6:30 PM)  
2. Minggu, 15 Desember 2024 (6:30 PM)  
-----  
Masukkan pilihan : 1
```

Setelah memilih jadwal konser, customer diminta untuk memilih jenis tiket dan jumlah yang tiket yang ingin dibeli.

```
----- Jenis Tiket -----  
-----  
1. CAT 4 Seating  
2. CAT 3 Seating  
3. CAT 2 Seating  
4. CAT 1 Seating  
-----  
Masukkan pilihan : 1  
-----  
Masukkan jumlah tiket yang ingin dibeli : 5  
-----
```

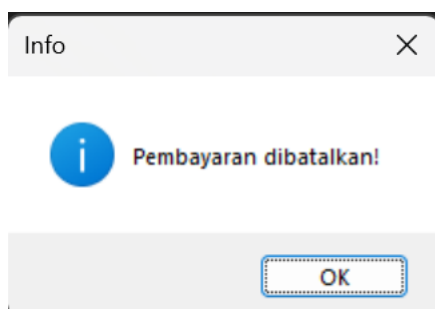
Setelah memilih jenis tiket maka akan muncul notifikasi:



```
-----  
Subtotal: Rp 750000  
-----  
Press any key to continue . . . █
```

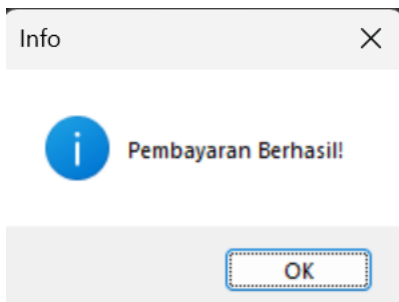
Untuk melakukan pembayaran tiket customer dapat kembali ke menu utama. Lalu customer dapat memilih menu pembayaran tiket. Pada menu ini customer akan melihat rincian pembelian tiket dan diminta untuk mengkonfirmasi pembayaran tersebut. Jika customer menolak melakukan pembayaran maka pembayaran akan dibatalkan.

```
-----  
===== TAGIHAN PEMBAYARAN =====  
-----  
----- Rincian Pembelian Tiket -----  
-----  
Nama       : Regita Cahya Arrahma  
Alamat     : Pamarican, Ciamis  
Email      : rgtama14@gmail.com  
Hari Pertunjukan : Sabtu, 14 Desember 2024 (6:30 PM)  
Jenis Tiket : CAT 4 Seating  
Jumlah Tiket : 5  
Harga Tiket : Rp 150000  
Subtotal   : Rp 750000  
-----  
Apakah Anda ingin melakukan pembayaran? (y/n): n  
-----
```

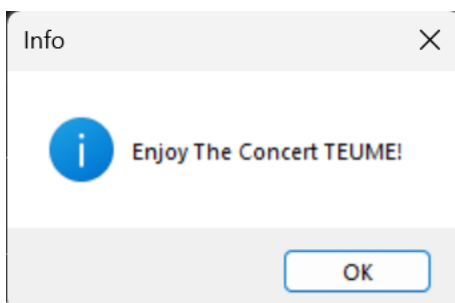


Jika customer menyetujui melakukan pembayaran, maka customer akan diminta untuk memilih metode pembayaran. Dan setelah memilih metode pembayaran, maka tiket akan dicetak.

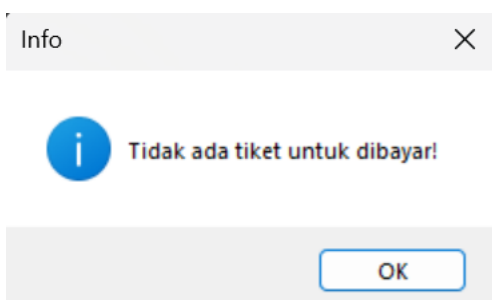
```
-----
===== Metode Pembayaran =====
-----
1. Transfer Bank
2. Kartu Kredit
3. Go Pay
-----
Masukkan pilihan : 1
-----
```



```
-----
===== TIKET TREASURE CONCERT IN YOGYAKARTA =====
-----
Hari/Tanggal Pembelian : Sunday , 2023-06-18 20:15:52.797592
Nama                   : Regita Cahya Arrahma
Alamat                 : Pamarican, Ciamis
Email                  : rgtama14@gmail.com
Hari Pertunjukan       : Sabtu, 14 Desember 2024 (6:30 PM)
Jenis Tiket            : CAT 4 Seating
Jumlah Tiket           : 5
Harga Tiket            : Rp. 150000
Subtotal               : Rp. 750000
Metode Pembayaran      : Transfer Bank
-----
```



Jika customer memilih menu pembayaran tiket tetapi sebelumnya belum melakukan pembelian tiket, maka akan muncul pemberitahuan seperti dibawah ini.



```
-----  
===== TAGIHAN PEMBAYARAN =====  
-----  
Press any key to continue . . .
```

Untuk melihat kuota tiket yang tersedia, customer dapat memilih menu Cek Kuota Tiket pada menu utama. Pada transaksi diawal customer memilih jadwal pertama dan jenis tiket CAT 4 Seating. Dapat dilihat bahwa kuota tiketnya berkurang setelah customer melakukan pembayaran.

```
-----  
===== TIKET TERSEDIA =====  
-----  
===== Jadwal Konser =====  
-----  
1. Sabtu, 14 Desember 2024 (6:30 PM)  
2. Minggu, 15 Desember 2024 (6:30 PM)  
-----  
Masukkan pilihan : 1  
-----  
----- Daftar Tiket Tersedia -----  
-----  
CAT 4 Seating: 995 tiket tersedia  
CAT 3 Seating: 500 tiket tersedia  
CAT 2 Seating: 300 tiket tersedia  
CAT 1 Seating: 200 tiket tersedia  
-----  
Press any key to continue . . .
```

Untuk keluar program, customer dapat memilih menu keluar dan program akan berhenti.

