



TONE TIX



ANGGOTA :

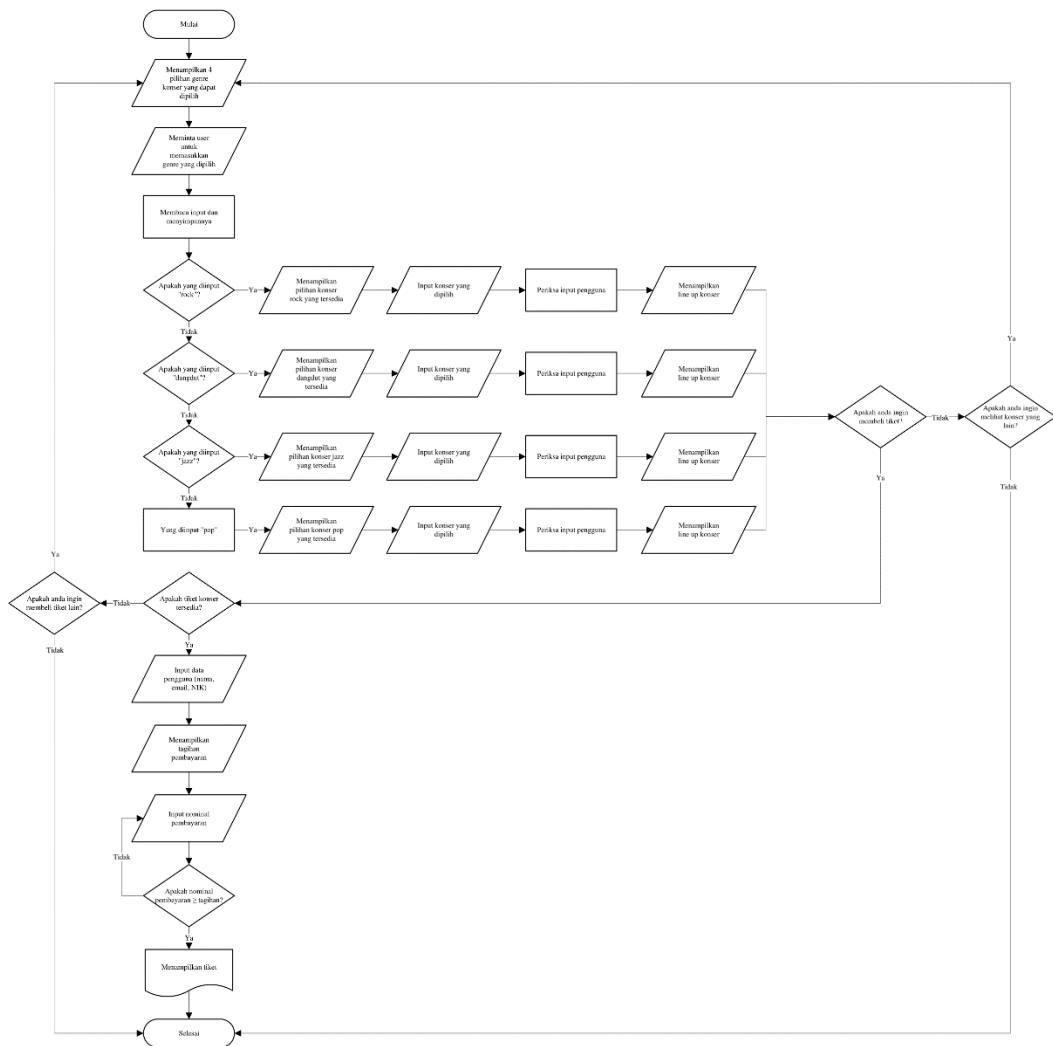
- ▶ Sania Amanati IO322115
- ▶ Taris Zulfania IO322121
- ▶ Yehuda Andelta IO322130
- ▶ Zaki Faiz IO322132

BAB I**Deskripsi Masalah**

Pada zaman ini dimana manusia mulai menaruh perhatian pada hal-hal yang dapat menyenangkan diri mereka, terdapat banyak sekali hal yang dapat dilakukan untuk memuaskan hal tersebut, salah satunya dengan menonton konser, hal ini dapat menyenangkan bagi individu itu sendiri dan juga bentuk support mereka kepada seniman yang melakukan konser tersebut. Namun kerap kali masyarakat kurang informasi tentang konser-konser apa saja yang akan diadakan mendatang, bagaimana cara membeli tiket untuk melihat konser tersebut, dan kurang baiknya sistem penjualan tiket yang akhirnya menurunkan tingkat kepuasan dan kesempatan yang hilang untuk menikmati hiburan yang diinginkan. Ada juga beberapa masyarakat yang pertama kali ingin melihat atau menonton konser secara langsung, bingung ingin melihat atau menonton konser apa yang ingin mereka lihat. Karena kurangnya pengetahuan tentang genre musik dan artis yang disukai, dan ketidak tahuhan akan adanya konser tertentu. Kekurangan informasi dapat menghambat kemampuan mereka untuk merencanakan konser. Dengan program ini masyarakat mendapat informasi mengenai konser yang akan datang dengan mudah, pengguna juga dapat memilih dan membeli konser yang mereka inginkan. Dengan adanya program ini diharapkan masyarakat dapat akan lebih mudah dalam memilih konser yang diinginkan, meningkatkan kepuasan, dan pengalaman mereka dalam mengeksplorasi hiburan konser.

BAB II

Flowchart dan Penjelasannya



Flowchart ini dimulai dengan kotak start yang menunjukkan awal dari proses. Langkah pertama adalah program menampilkan 4 pilihan genre yang dapat dipilih oleh pengguna yaitu Jazz, Rock, Pop, Dangdut. Selanjutnya pengguna diminta untuk menginput genre yang dipilih dengan menginputkan angka 1 untuk Jazz, 2 untuk Rock, 3 untuk Pop, dan 4 untuk Dangdut. Alur akan memeriksa pilihan pengguna dan melanjutkan ke langkah berikutnya. Jika pengguna menginputkan angka 1 maka program akan mengambil data konser yang tersedia dari file csv dan menampilkan daftar konser Jazz yang tersedia kepada pengguna. Hal ini juga berlaku apabila pengguna menginputkan angka 2 maka program akan menampilkan daftar konser Rock yang tersedia dari file csv dan menampilkan daftar konser Rock yang tersedia kepada pengguna, jika menginput angka 3 maka menampilkan daftar konser Pop yang tersedia dari file csv dan menampilkan daftar konser Pop yang tersedia kepada pengguna, ataupun menginput angka 4 maka menampilkan daftar konser Dangdut yang tersedia dari file csv dan menampilkan daftar konser Dangdut yang tersedia kepada pengguna. Apabila pengguna ingin keluar dari program dapat menginput angka 5.

Alur akan memeriksa pilihan pengguna dan melanjutkan ke langkah berikutnya. Langkah berikutnya, pengguna menginput nomer konser yang dipilih maka program akan menampilkan line up dan kategori tiket konser. Selanjutnya program akan menampilkan pilihan, apakah pengguna akan membeli tiket atau tidak. Jika pengguna memilih “tidak” membeli tiket maka program akan menampilkan pertanyaan apakah pengguna ingin memilih konser yang lain, jika “ya” program akan kembali ke proses menampilkan 4 genre namun jika memilih tidak maka program selesai. Jika pengguna memilih “ya” pada pertanyaan apakah pengguna ingin membeli tiket maka program akan memberi pilihan kategori tiket konser VIP atau reguler. Setelah pengguna memilih kategori tiket konser, pengguna diminta untuk mengisi identitas diri meliputi nama, NIK, dan email. Pengguna juga diminta untuk menginput jumlah tiket yang ingin dibeli dan nomor rekening pembayaran. Setelah itu program akan menampilkan nominal yang harus dibayar. Pengguna diminta untuk memasukan nominal pembayaran sesuai dengan total harga tiket yang dipesan. Jika nominal yang dimasukan kurang maka program akan menampilkan pemberitahuan bahwa nominal yang dimasukan kurang dan diminta untuk menginput ulang nominal yang benar. Jika nominal

yang dimasukan melebihi total pembayaran maka proses pemesanan tiket akan dilanjutkan dan terdapat pemberitahuan bahwa kelebihan nominal akan dikembalikan ke rekening yang diinputkan. Apabila nominal yang diinputkan sudah sesuai maka program akan berlanjut. Program akan berlanjut menampilkan tiket konser yang dipesan yang berupa E-ticket.

Bab III

Kode Pemrograman

```
● ● ●  
Nama Konser,Tempat,Tanggal,Line-up 1,Line-up 2,Line-up 3,Harga Tiket VIP,Harga Tiket Reguler  
Konser Dangdut Fever Jakarta,Gelora Bung Karno,Jakarta,10 Agustus 2023,Rita Sugiarto,Inul  
Daratista,Cita Citata,500000,200000  
Konser Dangdut Fest Surabaya,Gelora Delta Sidoarjo,Surabaya,15 Agustus 2023,Rhoma Irama,Elvy  
Sukaesih,Siti Badriah,450000,180000  
Konser Dangdut Sensation Bandung,Gedung Sate,Bandung,20 September 2023,Nella Kharisma,Via  
Vallen,Soimah,400000,160000  
Konser Dangdut Mania Medan,Stadion Teladan,Medan,27 September 2023,Lesti,Rizki Ridho,Zaskia  
Gotik,350000,140000  
Konser Dangdut Rhythm Bali,Pantai Kuta,Bali,13 Oktober 2023,Mansyur S,Cici Paramida,Fitri  
Carlina,300000,120000  
Konser Dangdut Megah Yogyakarta,Stadion Maguwoharjo,Yogyakarta,21 Oktober 2023,Danang,Ikke Nurjanah,Iis  
Dahlia,400000,160000  
Konser Dangdut Rocker Makassar,Lapangan Karebosi,Makassar,5 November 2023,Inul Daratista,Cita  
Citata,Zaskia Gotik,350000,140000  
Konser Dangdut Joss Palembang,Stadion Gelora Sriwijaya,Palembang,3 Desember 2023,Rita Sugiarto,Elvy  
Sukaesih,Siti Badriah,450000,180000  
Konser Dangdut Fiesta Solo,Stadion Manahan,Solo,17 Desember 2023,Via Vallen,Nella Kharisma,Fitri  
Carlina,400000,160000  
Konser Dangdut Harmoni Jakarta,Istora Senayan,Jakarta,25 Desember 2023,Rhoma Irama,Rizki Ridho,Iis  
Dahlia,500000,200000
```

```
● ● ●  
Nama Konser,Tempat,Tanggal,Lineup-1,Lineup-2,Lineup-3,Harga Tiket VIP,Harga Tiket Reguler  
Konser Jazz Night at Jakarta,Jakarta Convention Center,Jakarta,15 Agustus 2023,John Smith,Jane  
Doe,David Johnson,750000,350000  
Konser Jazz in the Garden,Taman Bunga Bandung,Bandung,23 Agustus 2023,Amy Williams,Robert Davis,Sarah  
Thompson,500000,250000  
Konser Bali Jazz Festival,Bali Nusa Dua Convention Center,Bali,30 Agustus 2023,Michael Wilson,Laura  
Anderson,Chris Parker,800000,400000  
Konser Jazz by the Beach,Pantai Sanur,Bali,02 September 2023,Emily Clark,Thomas Roberts,Natalie  
Lee,400000,200000  
Konser Jazz in the Old Town,Kota Tua Jakarta,Jakarta,10 September 2023,Matthew Brown,Olivia  
Garcia,Andrew Taylor,600000,300000  
Konser Jazz on the Hill,Bukit Darmo Golf Surabaya,Surabaya,18 September 2023,Sophia Martinez,James  
Wilson,Lily Davis,450000,200000  
Konser Jazz by the Lake,Danau Toba,Samosir,05 Oktober 2023,William Turner,Victoria White,Mark  
Adams,700000,350000  
Konser Jazz Harmony Festival,Gedung Kesenian Yogyakarta,Yogyakarta,13 Oktober 2023,Ava  
Green,Christopher Moore,Grace Hall,550000,250000  
Konser Jazz at the Pier,Pantai Losari Makassar,Makasar,11 November 2023,Daniel Mitchell,Isabella  
Lewis,Joseph Walker,400000,200000  
Konser Jazz in the City,Jogja National Museum,Yogyakarta,02 Desember 2023,Emily Harris,Nathan  
King,Samantha Carter,500000,250000
```



Nama Konser,Tempat,Tanggal,Line-up 1,Line-up 2,Line-up 3,Harga Tiket VIP,Harga Tiket Reguler
 Konser Pop Extravaganza,Gelora Bung Karno,Jakarta,15 Juli 2023,Raisa,Isyana Sarasvati,Afgan,500000,250000
 Konser Pop Hits Plaza Senayan,Plaza Senayan,Jakarta,25 Agustus 2023,Judika,Rossa,Andmesh Kamaleng,450000,200000
 Konser Pop Dance Fever,ICE BSD,Tangerang,5 September 2023,Agnez Mo,Tulus,Nidji,400000,180000
 Konser Pop Legends,Stadium Utama,Bandung,10 Oktober 2023,Sheila on 7,Padi Reborn,HIVI!,500000,250000
 Konser Pop Ballad Serenade,Venue A,Surabaya,20 November 2023,Yura Yunita,Maudy Ayunda,Rizky Febian,350000,150000
 Konser Pop Indie Showcase,The Kasablanka,Jakarta,5 Desember 2023,Payung Teduh,Efek Rumah Kaca,Kunto Aji,400000,180000
 Konser Pop Melayu Sensasi,Colosseum,Jakarta,10 Januari 2024,Anji,Armada,Wali Band,450000,200000
 Konser Pop Dangdut Party,Diamond Club,Bandung,15 Februari 2024,Via Vallen,Nella Kharisma,Siti Badriah,350000,150000
 Konser Pop R&B Soulful,The Foundry,Surabaya,20 Maret 2024,RAN,Maliq & D'Essentials,Teddy Adhitya,400000,180000
 Konser Pop Acoustic Serenity,Rooftop Garden,Bali,5 April 2024,Rizky Febian,Tulus,Maudy Ayunda,350000,150000



Nama Konser,Tempat,Tanggal,Line-up 1,Line-up 2,Line-up 3,Harga Tiket VIP,Harga Tiket Reguler
 Konser Rocking Jakarta,Stadion Utama Gelora Bung Karno,Jakarta,1 Agustus 2023,Andra and The Backbone,Navicula,The SIGIT,500000,250000
 Konser Rock Fest Surabaya,Gelora Delta Sidoarjo,Surabaya,19 Agustus 2023,Endank Soekamti,Padi Reborn,Rocket Rockers,450000,200000
 Konser Rock Revolution Bandung,Gedung Sate,Bandung,11 September 2023,Superman Is Dead,Slank,Netral,400000,180000
 Konser Rock On Medan,Stadion Teladan,Medan,21 September 2023,Gigi,Jamrud,Fourtwny,350000,150000
 Konser Rock the Island Bali,Pantai Kuta,Bali,30 September 2023,Sheila on 7,Burgerkill,Koil,400000,180000
 Konser Rock Legacy Yogyakarta,Stadion Maguwoharjo,Yogyakarta,5 Oktober 2023,J-Rocks,Seringai,Efek Rumah Kaca,450000,200000
 Konser Rock Anthem Makassar,Lapangan Karebosi,Makassar,17 Oktober 2023,NTRL,Endank Soekamti,Tipe-X,400000,180000
 Konser Rock Invasion Palembang,Stadion Gelora Sriwijaya,Palembang,27 Oktober 2023,Fourtwny,Payung Teduh,The Changcuters,350000,150000
 Konser Rock Fest Solo,Stadion Manahan,Solo,10 November 2023,Ran,Naif,The Adams,450000,200000
 Konser Rock Symphony Jakarta,Istora Senayan,Jakarta,15 Desember 2023,Andra and The Backbone,Kotak,The SIGIT,500000,250000

```
import csv
import random
import sys
from datetime import datetime

# Program ToneTix
def welcome_message():
    print("=====SELAMAT DATANG DI TONETIX!")
    print("      Kami adalah platform penjualan tiket konser terbaik untukmu!")
    print("=====")
    print("Tonetix menyediakan tiket konser untuk berbagai genre musik populer,")
    print("dengan kualitas suara dan pengalaman konser yang tak terlupakan.")
    print("Dengan Tonetix, Anda dapat menjelajahi dan membeli tiket untuk")
    print("konser-konser terbaik dari musisi dan band favorit Anda.")
    print("=====")
    print("Untuk membeli tiket, silakan pilih nomor genre musik yang Anda minati.")
    print("Kami akan membantu Anda mendapatkan pengalaman konser yang luar biasa!")
    print("=====")
    print()

def show_genre_menu():
    print("Pilih nomor genre musik yang Anda minati:")
    print("1. Jazz")
    print("2. Rock")
    print("3. Pop")
    print("4. Dangdut")
    print("5. Keluar")
    print("\n=====")

def get_genre_choice():
    choice = input("\nMasukkan nomor genre musik yang Anda pilih: ")
    return choice
```

- `import csv`: Mengimpor modul `csv` yang digunakan untuk membaca dan menulis file CSV.
- `import random`: Mengimpor modul `random` yang digunakan untuk menghasilkan angka acak.
- `import sys`: Mengimpor modul `sys` yang memberikan akses ke beberapa variabel dan fungsi yang digunakan untuk memanipulasi interpreter Python.
- `from datetime import datetime`: Mengimpor kelas `datetime` dari modul `datetime` yang digunakan untuk memanipulasi tanggal dan waktu.
- `def welcome_message():`: Mendefinisikan fungsi `welcome_message()` yang mencetak pesan sambutan untuk program ToneTix.
- `print():` Mencetak baris kosong.
- `def show_genre_menu():`: Mendefinisikan fungsi `show_genre_menu()` yang mencetak menu genre musik yang tersedia.
- `def get_genre_choice():`: Mendefinisikan fungsi `get_genre_choice()` yang meminta pengguna memasukkan nomor genre musik yang dipilih.
- `choice = input("\nMasukkan nomor genre musik yang Anda pilih: ")`: Mengambil input dari pengguna dalam bentuk nomor genre musik yang dipilih.
- `return choice`: Mengembalikan nilai `choice` dari fungsi `get_genre_choice()`.

```

def show_jazz_concerts():
    with open('konser_jazz.csv', 'r') as file:
        reader = csv.reader(file)
        next(reader)
        print()
        print("Daftar Konser Jazz yang Tersedia:")
        print("====")
        count = 1
        concerts = [] # List untuk menyimpan data konser
        for row in reader:
            concerts.append(row) # Menambahkan data konser ke list
            print("Nomor Konser : ", count)
            print("Nama Konser : ", row[0])
            print("Venue : ", row[1])
            print("Lokasi : ", row[2])
            print("Tanggal : ", row[3])
            print("Harga : - VIP : Rp. ",row[7])
            print("          - Reguler : Rp. ",row[8])
            print("====")
            count += 1
    return concerts

def choose_jazz_concert(concerts):
    choice = input("\nMasukkan nomor konser yang Anda pilih: ")
    try:
        choice = int(choice)
        if 1 <= choice <= len(concerts):
            chosen_concert = concerts[choice - 1]
            print()
            print("Anda memilih konser:", chosen_concert[0])
            print("Line-up:")
            print("- Line-up 1 :", chosen_concert[4])
            print("- Line-up 2 :", chosen_concert[5])
            print("- Line-up 3 :", chosen_concert[6])
            print("Harga Tiket:")
            print("- VIP : Rp.", chosen_concert[7])
            print("- Reguler : Rp.", chosen_concert[8])
            print()

            while True:
                buy_ticket = input("Apakah Anda ingin membeli tiket? (ya/tidak): ")
                if buy_ticket.lower() == "ya":
                    while True:
                        ticket_type = input("Pilih jenis tiket (VIP/Reguler): ")
                        if ticket_type.lower() == "vip":
                            buy_vip_ticket(chosen_concert)
                            break
                        elif ticket_type.lower() == "reguler":
                            buy_reguler_ticket(chosen_concert)
                            break
                        else:
                            print("Jenis tiket yang Anda pilih tidak valid.")
                    break
                elif buy_ticket.lower() == "tidak":
                    view_other_conerts = ""
                    while view_other_conerts.lower() not in ["ya", "tidak"]:
                        view_other_conerts = input("Apakah Anda ingin melihat konser yang lain? (ya/tidak): ")
                        if view_other_conerts.lower() not in ["ya", "tidak"]:
                            print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
                    if view_other_conerts.lower() == "ya":
                        print("Kembali ke menu awal :)")
                    else:
                        print("Terima kasih telah menggunakan ToneTix!")
                        sys.exit()
                    break
                else:
                    print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
    except ValueError:
        print("Input yang Anda masukkan bukan nomor konser yang valid.")
    print("Input yang Anda masukkan bukan nomor konser yang valid.")

```

`def show_jazz_concerts():` Mendefinisikan fungsi `show_jazz_concerts()` yang bertujuan untuk menampilkan daftar konser Jazz yang tersedia.

- `with open('konser_jazz.csv', 'r') as file:` Membuka file CSV 'konser_jazz.csv' dalam mode baca ('r') menggunakan pernyataan `with`. File akan ditutup secara otomatis setelah keluar dari blok `with`.
- `reader = csv.reader(file)`: Membaca file CSV menggunakan objek `reader` dari modul `csv`.
- `next(reader)`: Membaca baris pertama file CSV yang berisi judul kolom dan melewatkannya.
- `count = 1`: Menginisialisasi variabel `count` sebagai penanda nomor konser.
- `concerts = []`: Membuat list kosong `concerts` yang akan digunakan untuk menyimpan data konser.
- `for row in reader:`: Melakukan iterasi pada setiap baris dalam file CSV.
 - o `concerts.append(row)`: Menambahkan data konser dalam bentuk baris ke dalam list `concerts`.
 - o Mencetak informasi tentang konser, seperti nomor konser, nama konser, venue, lokasi, tanggal, dan harga.
- `return concerts`: Mengembalikan list `concerts` yang berisi data konser.

`def choose_jazz_concert(concerts):` Mendefinisikan fungsi `choose_jazz_concert()` yang meminta pengguna memilih konser Jazz dan melakukan tindakan berdasarkan pilihan pengguna.

- `choice = input("\nMasukkan nomor konser yang Anda pilih: ")`: Mengambil input dari pengguna berupa nomor konser yang dipilih.
- `try:` Memulai blok percobaan untuk menangani exception.
 - o `choice = int(choice)`: Mengubah input pengguna menjadi tipe data integer.
 - o `if 1 <= choice <= len(concerts):`: Memeriksa apakah nomor konser yang dipilih valid.
 - `chosen_concert = concerts[choice - 1]`: Mengambil data konser yang dipilih dari list `concerts`.
 - Mencetak informasi tentang konser yang dipilih, seperti nama konser, line-up, dan harga tiket.
 - Meminta pengguna apakah ingin membeli tiket dan melakukan tindakan berdasarkan jawaban pengguna.

`def show_rock_concerts():` Mendefinisikan fungsi `show_rock_concerts()` yang bertujuan untuk menampilkan daftar konser Rock yang tersedia.

- `with open('konser_rock.csv', 'r') as file:` Membuka file CSV 'konser_rock.csv' dalam mode baca ('r') menggunakan pernyataan `with`. File akan ditutup secara otomatis setelah keluar dari blok `with`.
- Fungsi ini memiliki struktur yang mirip dengan fungsi `show_jazz_concerts()`, namun menampilkan daftar konser Rock yang tersedia.
- `return concerts`: Mengembalikan list `concerts` yang berisi data konser.

```

def show_rock_concerts():
    with open('konser_rock.csv', 'r') as file:
        reader = csv.reader(file)
        next(reader)
        print("Daftar Konser Rock yang Tersedia:")
        print("====")
        count = 1
        concerts = [] # List untuk menyimpan data konser
        for row in reader:
            concerts.append(row) # Menambahkan data konser ke list
            print("Nomor Konser : ", count)
            print("Nama Konser : ", row[0])
            print("Venue : ", row[1])
            print("Lokasi : ", row[2])
            print("Tanggal : ", row[3])
            print("Harga - VIP : Rp. ",row[7])
            print(" - Reguler : Rp. ",row[8])
            print("====")
            count += 1
    return concerts

def choose_rock_concert(concerts):
    choice = input("\nMasukkan nomor konser yang Anda pilih: ")
    try:
        choice = int(choice)
        if 1 <= choice <= len(concerts):
            chosen_concert = concerts[choice - 1]
            print()
            print("Anda memilih konser: ", chosen_concert[0])
            print("Line-up:")
            print("- Line-up 1 : ", chosen_concert[4])
            print("- Line-up 2 : ", chosen_concert[5])
            print("- Line-up 3 : ", chosen_concert[6])
            print("Harga Tiket:")
            print("- VIP : Rp.", chosen_concert[7])
            print("- Reguler : Rp.", chosen_concert[8])
            print()

            while True:
                buy_ticket = input("Apakah Anda ingin membeli tiket? (ya/tidak): ")
                if buy_ticket.lower() == "ya":
                    while True:
                        ticket_type = input("Pilih jenis tiket (VIP/Reguler): ")
                        if ticket_type.lower() == "vip":
                            buy_vip_ticket(chosen_concert)
                            break
                        elif ticket_type.lower() == "reguler":
                            buy_reguler_ticket(chosen_concert)
                            break
                        else:
                            print("Jenis tiket yang Anda pilih tidak valid.")
                    break
                elif buy_ticket.lower() == "tidak":
                    view_other_concerts = ""
                    while view_other_concerts.lower() not in ["ya", "tidak"]:
                        view_other_concerts = input("Apakah Anda ingin melihat konser yang lain? (ya/tidak): ")
                    if view_other_concerts.lower() not in ["ya", "tidak"]:
                        print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
                    if view_other_concerts.lower() == "ya":
                        print("Kembali ke menu awal :")
                    else:
                        print("Terima kasih telah menggunakan ToneTix!")
                        sys.exit()
                    break
                else:
                    print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
    except ValueError:
        print("Input yang Anda masukkan bukan nomor konser yang valid.")

```

`def show_rock_concerts():` Mendefinisikan fungsi `show_rock_concerts()` yang mirip dengan `show_pop_concerts()`, tetapi berhubungan dengan konser Rock.

- Fungsinya adalah membuka file CSV 'konser_rock.csv', menampilkan daftar konser Rock yang tersedia, dan menyimpan data konser dalam list `concerts`.

`def choose_rock_concert(concerts):` Mendefinisikan fungsi `choose_rock_concert()` yang mirip dengan `choose_pop_concert()`, tetapi berhubungan dengan konser Rock.

- Fungsinya adalah meminta pengguna memilih konser Rock, menampilkan informasi tentang konser yang dipilih, dan memberikan opsi untuk membeli tiket.
- Fungsi ini menggunakan parameter `concerts` yang merupakan list yang berisi data konser Rock.

```
def show_pop_concerts():
    with open('konser_pop.csv', 'r') as file:
        reader = csv.reader(file)
        next(reader)
        print("Daftar Konser Pop yang Tersedia:")
        print("=====")
        count = 1
        concerts = [] # List untuk menyimpan data konser
        for row in reader:
            concerts.append(row) # Menambahkan data konser ke list
            print("Nomor Konser :", count)
            print("Nama Konser :", row[0])
            print("Venue      :", row[1])
            print("Lokasi     :", row[2])
            print("Tanggal    :", row[3])
            print("Harga       : - VIP      : Rp. ",row[7])
            print("           - Reguler : Rp. ",row[8])
            print("=====")
            count += 1
    return concerts

def choose_pop_concert(concerts):
    choice = input("\nMasukkan nomor konser yang Anda pilih: ")
    try:
        choice = int(choice)
        if 1 <= choice <= len(concerts):
            chosen_concert = concerts[choice - 1]
            print()
            print("Anda memilih konser:", chosen_concert[0])
            print("Line-up:")
            print("- Line-up 1 :", chosen_concert[4])
            print("- Line-up 2 :", chosen_concert[5])
            print("- Line-up 3 :", chosen_concert[6])
            print("Harga Tiket:")
            print("- VIP      : Rp.", chosen_concert[7])
            print("- Reguler : Rp.", chosen_concert[8])
            print()
    except ValueError:
        print("Pilihan tidak valid. Silakan masukkan angka yang benar.")
```

```
while True:
    buy_ticket = input("Apakah Anda ingin membeli tiket? (ya/tidak): ")
    if buy_ticket.lower() == "ya":
        while True:
            ticket_type = input("Pilih jenis tiket (VIP/Reguler): ")
            if ticket_type.lower() == "vip":
                buy_vip_ticket(chosen_concert)
                break
            elif ticket_type.lower() == "reguler":
                buy_reguler_ticket(chosen_concert)
                break
            else:
                print("Jenis tiket yang Anda pilih tidak valid.")
        break
    elif buy_ticket.lower() == "tidak":
        view_other_concerts = ""
        while view_other_concerts.lower() not in ["ya", "tidak"]:
            view_other_concerts = input("Apakah Anda ingin melihat konser yang lain?
(ya/tidak): ")
            if view_other_concerts.lower() not in ["ya", "tidak"]:
                print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
            if view_other_concerts.lower() == "ya":
                print("Kembali ke menu awal :)")
            else:
                print("Terima kasih telah menggunakan ToneTix!")
                sys.exit()
        break
    else:
        print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
else:
    print("Nomor konser yang Anda pilih tidak valid.")
except ValueError:
    print("Input yang Anda masukkan bukan nomor konser yang valid.")
```

def show_pop_concerts(): Mendefinisikan fungsi `show_pop_concerts()` yang mirip dengan `show_rock_concerts()`, tetapi berhubungan dengan konser Pop.

- Fungsinya adalah membuka file CSV 'konser_pop.csv', menampilkan daftar konser Pop yang tersedia, dan menyimpan data konser dalam list `concerts`.

def choose_pop_concert(concerts): Mendefinisikan fungsi `choose_pop_concert()` yang mirip dengan `choose_rock_concert()`, tetapi berhubungan dengan konser Pop.s

- Fungsinya adalah meminta pengguna memilih konser Pop, menampilkan informasi tentang konser yang dipilih, dan memberikan opsi untuk membeli tiket.

Fungsi ini menggunakan parameter `concerts` yang merupakan list yang berisi data konser Pop.

```

def show_dangdut_concerts():
    with open('konser_dangdut.csv', 'r') as file:
        reader = csv.reader(file)
        next(reader)
        print("Daftar Konser Dangdut yang Tersedia:")
        print("====")
        count = 1
        concerts = [] # List untuk menyimpan data konser
        for row in reader:
            concerts.append(row) # Menambahkan data konser ke list
            print("Nomor Konser : ", count)
            print("Nama Konser : ", row[0])
            print("Venue : ", row[1])
            print("Lokasi : ", row[2])
            print("Tanggal : ", row[3])
            print("Harga - VIP : Rp. ", row[7])
            print(" - Reguler : Rp. ", row[8])
            print("====")
            count += 1
    return concerts

def choose_dangdut_concert(concerts):
    choice = input("\nMasukkan nomor konser yang Anda pilih: ")
    try:
        choice = int(choice)
        if 1 <= choice <= len(concerts):
            chosen_concert = concerts[choice - 1]
            print()
            print("Anda memilih konser: ", chosen_concert[0])
            print("Line-up:")
            print("- Line-up 1 : ", chosen_concert[4])
            print("- Line-up 2 : ", chosen_concert[5])
            print("- Line-up 3 : ", chosen_concert[6])
            print("Harga Tiket:")
            print("- VIP : Rp.", chosen_concert[7])
            print("- Reguler : Rp.", chosen_concert[8])
            print()

            while True:
                buy_ticket = input("Apakah Anda ingin membeli tiket? (ya/tidak): ")
                if buy_ticket.lower() == "ya":
                    while True:
                        ticket_type = input("Pilih jenis tiket (VIP/Reguler): ")
                        if ticket_type.lower() == "vip":
                            buy_vip_ticket(chosen_concert)
                            break
                        elif ticket_type.lower() == "reguler":
                            buy_reguler_ticket(chosen_concert)
                            break
                        else:
                            print("Jenis tiket yang Anda pilih tidak valid.")
                    break
                elif buy_ticket.lower() == "tidak":
                    view_other_concerts = ""
                    while view_other_concerts.lower() not in ["ya", "tidak"]:
                        view_other_concerts = input("Apakah Anda ingin melihat konser yang lain? (ya/tidak): ")
                        if view_other_concerts.lower() not in ["ya", "tidak"]:
                            print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
                    if view_other_concerts.lower() == "ya":
                        print("Kembali ke menu awal :")
                    else:
                        print("Terima kasih telah menggunakan ToneTix!")
                        sys.exit()
                    break
                else:
                    print("Jawaban tidak valid. Silakan masukkan 'ya' atau 'tidak'.")
            else:
                print("Nomor konser yang Anda pilih tidak valid.")
        except ValueError:
            print("Input yang Anda masukkan bukan nomor konser yang valid.")
    
```

`def show_dangdut_concerts():` Mendefinisikan fungsi `show_dangdut_concerts()` yang mirip dengan `show_rock_concerts()` dan `show_pop_concerts()`, tetapi berhubungan dengan konser Dangdut.

- Fungsinya adalah membuka file CSV 'konser_dangdut.csv', menampilkan daftar konser Dangdut yang tersedia, dan menyimpan data konser dalam list `concerts`.

`def choose_dangdut_concert(concerts):` Mendefinisikan fungsi `choose_dangdut_concert()` yang mirip dengan `choose_rock_concert()` dan `choose_pop_concert()`, tetapi berhubungan dengan konser Dangdut.

- Fungsinya adalah meminta pengguna memilih konser Dangdut, menampilkan informasi tentang konser yang dipilih, dan memberikan opsi untuk membeli tiket.

Fungsi ini menggunakan parameter `concerts` yang merupakan list yang berisi data konser Dangdut.

```
def get_ticket_prices(genre, concert_number):
    filename = f'konser_{genre.lower()}.csv'
    with open(filename, 'r') as file:
        reader = csv.reader(file)
        next(reader)
        for index, row in enumerate(reader):
            if index + 1 == concert_number:
                return int(row[7]), int(row[8])
    return None, None

def buy_vip_ticket(concert):
    nama = input("Masukkan Nama : ")
    nik = input("Masukkan NIK : ")
    email = ""
    while "@" not in email:
        email = input("Masukkan Email : ")
        if "@" not in email:
            print("Email yang Anda masukkan tidak valid. Mohon masukkan email dengan benar.")
    jumlah_tiket = int(input("Masukkan jumlah tiket yang ingin Anda beli : "))
    no_rekening = int(input("Masukkan No. Rekening : "))
```

`def get_ticket_prices(genre, concert_number):` Mendefinisikan fungsi `get_ticket_prices()` yang digunakan untuk mendapatkan harga tiket berdasarkan genre konser dan nomor konser yang dipilih.

- Fungsi ini menerima dua argumen, yaitu `genre` (jenis konser) dan `concert_number` (nomor konser).
- Fungsi akan membaca file CSV yang sesuai dengan genre konser yang dipilih, mencari baris yang sesuai dengan nomor konser yang dipilih, dan mengembalikan harga tiket VIP dan Reguler dari baris tersebut.
- Jika tidak ditemukan baris yang sesuai, fungsi akan mengembalikan `None, None` untuk harga tiket.

`def buy_vip_ticket(concert):` Mendefinisikan fungsi `buy_vip_ticket()` yang digunakan untuk membeli tiket VIP.

- Fungsi ini akan meminta pengguna memasukkan nama, NIK, email, jumlah tiket yang ingin dibeli, dan nomor rekening.

Fungsi ini menggunakan parameter `concert` yang berisi informasi tentang konser yang dipilih.

```
harga_tiket = int(concert[7])
total_harga = harga_tiket*jumlah_tiket

tanggal_waktu = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
with open('data.csv', 'a', newline='') as file:
    writer = csv.writer(file)
    writer.writerow([tanggal_waktu,"Nama :", nama,"NIK :", nik,"email :", email,"Jumlah Tiket :",
jumlah_tiket,"Nomor Rekening :", no_rekening])
```

harga_tiket = int(concert[7]): Mengambil harga tiket VIP dari `concert` (informasi konser) yang diberikan dan menyimpannya dalam variabel `harga_tiket`. Angka 7 menunjukkan indeks kolom harga tiket VIP dalam data konser.

total_harga = harga_tiket * jumlah_tiket: Mengalikan harga tiket VIP dengan jumlah tiket yang ingin dibeli untuk mendapatkan total harga tiket, dan menyimpannya dalam variabel `total_harga`.

tanggal_waktu = datetime.now().strftime("% Y-% m-% d % H:% M:% S"): Mengambil tanggal dan waktu saat ini menggunakan modul `datetime` dan menyimpannya dalam variabel `tanggal_waktu`. Format tanggal dan waktu yang digunakan adalah "YYYY-MM-DD HH:MM:SS".

with open('data.csv', 'a', newline="") as file: Membuka file CSV dengan nama "data.csv" dalam mode "a" (append) untuk menambahkan data baru. Jika file tidak ada, maka akan dibuat.

writer = csv.writer(file): Membuat objek `writer` untuk menulis data ke file CSV yang terbuka.

writer.writerow([tanggal_waktu, "Nama :", nama, "NIK :", nik, "email :", email, "Jumlah Tiket :", jumlah_tiket,

"Nomor Rekening :", no_rekening]): Menulis baris data baru ke file CSV. Data tersebut terdiri dari `tanggal_waktu`, `nama`, `NIK`, `email`, `Jumlah Tiket`, dan `Nomor Rekening` yang dimasukkan oleh pengguna. Baris data ditulis dalam bentuk list, dengan setiap elemen list mewakili nilai dari kolom-kolom data yang ada.

```

print("Terima kasih, tiket VIP Anda telah berhasil dibeli.")
print("Total harga yang harus dibayar: Rp.", total_harga)
while True:
    payment = int(input("Masukkan nominal pembayaran: "))
    if payment == total_harga:
        print("Pembayaran cukup. Tiket Anda telah diterbitkan.")
        print("===== TIKET KONSER =====")
        kode_unik = random.randint(1000000000, 9999999999)
        print("Nama      :", nama)
        print("NIK       :", nik)
        print("Email     :", email)
        print("Jenis Tiket : ", "VIP")
        print("Jumlah Tiket : ", jumlah_tiket)
        print("Kode Tiket  : ", kode_unik)
        print("=====")
        print("Silahkan tunjukkan E-Ticket ini di loket pembelian tiket")
        print("          Untuk mendapatkan tiket Anda")
        print("          Terima kasih. Selamat menikmati konser!")
        sys.exit()
    elif payment > total_harga:
        print("Pembayaran cukup. Tiket Anda telah diterbitkan.")
        print("===== TIKET KONSER =====")
        kode_unik = random.randint(1000000000, 9999999999)
        print("Nama      :", nama)
        print("NIK       :", nik)
        print("Email     :", email)
        print("Jenis Tiket : ", "Reguler")
        print("Jumlah Tiket : ", jumlah_tiket)
        print("Kode Tiket  : ", kode_unik)
        print("Uang kembalian akan dikirim ke Nomor Rekening Anda")
        print("=====")
        print("Silahkan tunjukkan E-Ticket ini di loket pembelian tiket")
        print("          Untuk mendapatkan tiket Anda")
        print("          Terima kasih. Selamat menikmati konser!")
    else:
        print("Pembayaran kurang. Mohon masukkan nominal pembayaran yang cukup.")
        sys.exit()

```

- `print("Terima kasih, tiket VIP Anda telah berhasil dibeli.")`: Menampilkan pesan konfirmasi bahwa pembelian tiket VIP berhasil.
- `print("Total harga yang harus dibayar: Rp.", total_harga)`: Menampilkan total harga yang harus dibayar oleh pengguna.
- `while True:`: Memulai perulangan tak terbatas untuk meminta input nominal pembayaran dari pengguna.
- `payment = int(input("Masukkan nominal pembayaran: "))`: Mengambil input nominal pembayaran dari pengguna dan menyimpannya dalam variabel `payment`. Input ini diharapkan dalam bentuk integer.
- `if payment == total_harga:`: Memeriksa apakah nominal pembayaran yang dimasukkan pengguna sama dengan total harga yang harus dibayar.
- `print("Pembayaran cukup. Tiket Anda telah diterbitkan.")`: Menampilkan pesan bahwa pembayaran sudah cukup untuk membeli tiket.
- `kode_unik = random.randint(1000000000, 9999999999)`: Menghasilkan sebuah kode unik secara acak antara 1000000000 dan 9999999999, dan menyimpannya dalam variabel `kode_unik`.
- Menampilkan informasi tiket yang telah diterbitkan, termasuk nama, NIK, email, jenis tiket (VIP atau Reguler), jumlah tiket, dan kode tiket yang dihasilkan.
- `sys.exit()`: Menghentikan program secara langsung setelah penerbitan tiket.

- **elif payment > total_harga:**: Memeriksa apakah nominal pembayaran yang dimasukkan pengguna lebih besar dari total harga yang harus dibayar.
- Menampilkan informasi tiket reguler yang telah diterbitkan, termasuk nama, NIK, email, jenis tiket (Reguler), jumlah tiket, dan kode tiket yang dihasilkan. Pesan tambahan juga ditampilkan bahwa uang kembalian akan dikirim ke nomor rekening pengguna.
- **else::** Bagian ini akan dijalankan jika nominal pembayaran yang dimasukkan pengguna kurang dari total harga yang harus dibayar.
- **print("Pembayaran kurang. Mohon masukkan nominal pembayaran yang cukup."):** Menampilkan pesan bahwa nominal pembayaran yang dimasukkan pengguna kurang.
- **sys.exit():** Menghentikan program secara langsung setelah pesan kesalahan ditampilkan.

```
def buy_reguler_ticket(concert):
    nama = input("Masukkan Nama : ")
    nik = input("Masukkan NIK : ")
    email = input("Masukkan Email : ")
    email = ""
    while "@" not in email:
        email = input("Masukkan Email : ")
        if "@" not in email:
            print("Email yang Anda masukkan tidak valid. Mohon masukkan email dengan benar.")
    jumlah_tiket = int(input("Masukkan jumlah tiket yang ingin Anda beli : "))
    no_rekening = int(input("Masukkan No. Rekening : "))

    harga_tiket = int(concert[8])
    total_harga = harga_tiket * jumlah_tiket

    print("Terima kasih, tiket Reguler Anda telah berhasil dibeli.")
    print("Total harga yang harus dibayar: Rp.", total_harga)
    while True:
        payment = int(input("Masukkan nominal pembayaran: "))
        if payment == total_harga:
            print("Pembayaran cukup. Tiket Anda telah diterbitkan.")
            print("===== TIKET KONSER =====")
            kode_unik = random.randint(1000000000, 9999999999)
            print("Nama : ", nama)
            print("NIK : ", nik)
            print("Email : ", email)
            print("Jenis Tiket : ", "Reguler")
            print("Jumlah Tiket : ", jumlah_tiket)
            print("Kode Tiket : ", kode_unik)
            print("=====")
            print("Silahkan tunjukkan E-Ticket ini di loket pembelian tiket")
            print("Untuk mendapatkan tiket Anda")
            print("Terima kasih. Selamat menikmati konser!")
            sys.exit()
```

nama = input("Masukkan Nama : "): Mengambil input nama dari pengguna dan menyimpannya dalam variabel **nama**.

- **nik = input("Masukkan NIK : ")**: Mengambil input NIK dari pengguna dan menyimpannya dalam variabel **nik**.
- **email = ""**: Mendefinisikan variabel **email** sebagai string kosong.
- **while "@" not in email::** Memulai perulangan hingga pengguna memasukkan email yang valid (mengandung karakter '@').
- **email = input("Masukkan Email : ")**: Mengambil input email dari pengguna dan menyimpannya dalam variabel **email**.

- if "@" not in email:: Memeriksa apakah email yang dimasukkan pengguna tidak valid (tidak mengandung karakter '@').
- print("Email yang Anda masukkan tidak valid. Mohon masukkan email dengan benar."): Menampilkan pesan kesalahan bahwa email yang dimasukkan pengguna tidak valid.
- jumlah_tiket = int(input("Masukkan jumlah tiket yang ingin Anda beli : ")): Mengambil input jumlah tiket yang ingin dibeli dari pengguna dan menyimpannya dalam variabel **jumlah_tiket**. Input ini diharapkan dalam bentuk integer.
- no_rekening = int(input("Masukkan No. Rekening : ")): Mengambil input nomor rekening dari pengguna dan menyimpannya dalam variabel **no_rekening**. Input ini diharapkan dalam bentuk integer.
- harga_tiket = int(concert[8]): Mengambil harga tiket Reguler dari data konser yang dipilih dan menyimpannya dalam variabel **harga_tiket**.
- total_harga = harga_tiket * jumlah_tiket: Menghitung total harga yang harus dibayar dengan mengalikan harga tiket Reguler dengan jumlah tiket yang dibeli, dan menyimpan hasilnya dalam variabel **total_harga**.
- print("Terima kasih, tiket Reguler Anda telah berhasil dibeli."): Menampilkan pesan konfirmasi bahwa pembelian tiket Reguler berhasil.
- print("Total harga yang harus dibayar: Rp.", total_harga): Menampilkan total harga yang harus dibayar oleh pengguna.
- while True:: Memulai perulangan tak terbatas untuk meminta input nominal pembayaran dari pengguna.
- payment = int(input("Masukkan nominal pembayaran: ")): Mengambil input nominal pembayaran dari pengguna dan menyimpannya dalam variabel **payment**. Input ini diharapkan dalam bentuk integer.
- if payment == total_harga:: Memeriksa apakah nominal pembayaran yang dimasukkan pengguna sama dengan total harga yang harus dibayar.
- Menampilkan informasi tiket yang telah diterbitkan, termasuk nama, NIK, email, jenis tiket (Reguler), jumlah tiket, dan kode tiket yang dihasilkan.
- sys.exit(): Menghentikan program secara langsung setelah penerbitan tiket

```

        sys.exit()
    elif payment > total_harga:
        print("Pembayaran cukup. Tiket Anda telah diterbitkan.")
        print("===== TIKET KONSER =====")
        kode_unik = random.randint(1000000000, 9999999999)
        print("Nama : ", nama)
        print("NIK : ", nik)
        print("Email : ", email)
        print("Jenis Tiket : ", "Reguler")
        print("Jumlah Tiket : ", jumlah_tiket)
        print("Kode Tiket : ", kode_unik)
        print("Uang kembalian akan dikirim ke Nomor Rekening Anda")
        print("=====")
        print("Silahkan tunjukkan E-Ticket ini di loket pembelian tiket")
        print("Untuk mendapatkan tiket Anda")
        print("Terima kasih. Selamat menikmati konser!")
        sys.exit()
    else:
        print("Pembayaran kurang. Mohon masukkan nominal pembayaran yang cukup.")

```

- `print("Pembayaran cukup. Tiket Anda telah diterbitkan.")`: Menampilkan pesan bahwa pembayaran yang dimasukkan oleh pengguna cukup.
- `print("===== TIKET KONSER =====")`: Menampilkan judul "TIKET KONSER" sebagai pembatas sebelum informasi tiket ditampilkan.
- `kode_unik = random.randint(1000000000, 9999999999)`: Menghasilkan nomor kode unik secara acak antara 10 digit dan menyimpannya dalam variabel `kode_unik`.
- Menampilkan informasi tiket yang telah diterbitkan, termasuk nama, NIK, email, jenis tiket (Reguler), jumlah tiket, kode tiket, dan informasi bahwa uang kembalian akan dikirimkan ke nomor rekening pengguna.
- `sys.exit()`: Menghentikan program secara langsung setelah penerbitan tiket.

`print("Pembayaran kurang. Mohon masukkan nominal pembayaran yang cukup.")`: Menampilkan pesan bahwa pembayaran yang dimasukkan oleh pengguna kurang dari total harga yang harus dibayar. Pesan ini meminta pengguna untuk memasukkan nominal pembayaran yang cukup.

```

# Program utama
if __name__ == "__main__":
    welcome_message()

    while True:
        show_genre_menu()
        genre = get_genre_choice()

        if genre == "1":
            concerts = show_jazz_concerts()
            choose_jazz_concert(concerts)
        elif genre == "2":
            concerts = show_rock_concerts()
            choose_rock_concert(concerts)
        elif genre == "3":
            concerts = show_pop_concerts()
            choose_pop_concert(concerts)
        elif genre == "4":
            concerts = show_dangdut_concerts()
            choose_dangdut_concert(concerts)
        elif genre == "5":
            print("Keluar dari program...")
            break # Keluar dari loop dan program berakhir
        else:
            print("Genre musik yang Anda pilih tidak valid.")

```

- `welcome_message()`: Memanggil fungsi `welcome_message()` untuk menampilkan pesan sambutan kepada pengguna.
- Memulai loop `while True` yang akan terus berjalan hingga kondisi di dalamnya tidak terpenuhi (dalam hal ini, hanya saat pengguna memilih keluar dari program).
- `show_genre_menu()`: Memanggil fungsi `show_genre_menu()` untuk menampilkan menu genre musik kepada pengguna.
- `genre = get_genre_choice()`: Memanggil fungsi `get_genre_choice()` untuk mendapatkan pilihan genre musik yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `genre`.
- Menggunakan struktur `if-elif-else` untuk memeriksa pilihan genre musik yang dimasukkan oleh pengguna.
 - o Jika pengguna memilih genre "1" (jazz), maka fungsi `show_jazz_concerts()` akan dipanggil untuk menampilkan daftar konser jazz, dan fungsi `choose_jazz_concert()` akan dipanggil untuk memilih konser jazz.
 - o Jika pengguna memilih genre "2" (rock), maka fungsi `show_rock_concerts()` akan dipanggil untuk menampilkan daftar konser rock, dan fungsi `choose_rock_concert()` akan dipanggil untuk memilih konser rock.
 - o Jika pengguna memilih genre "3" (pop), maka fungsi `show_pop_concerts()` akan dipanggil untuk menampilkan daftar konser pop, dan fungsi `choose_pop_concert()` akan dipanggil untuk memilih konser pop.
 - o Jika pengguna memilih genre "4" (dangdut), maka fungsi `show_dangdut_concerts()` akan dipanggil untuk menampilkan daftar konser dangdut, dan fungsi `choose_dangdut_concert()` akan dipanggil untuk memilih konser dangdut.
 - o Jika pengguna memilih genre "5", maka program akan mencetak pesan "Keluar dari program..." dan menggunakan pernyataan `break` untuk keluar dari loop `while True` dan mengakhiri program.
 - o Jika pengguna memilih genre lain yang tidak valid, maka program akan mencetak pesan "Genre musik yang Anda pilih tidak valid."

**KELOMPOK
23**

TONETIX

PROJECT TEAM PROGRAM KOMPUTER
2023

```
import tkinter as tk
from PIL import ImageTk, Image
import csv
from tkinter import messagebox
import random
import string
import datetime

def welcome_message():
    new_window = tk.Toplevel(window)
    new_window.title("Tonetix")
    new_window.geometry("600x400")

    welcome_label = tk.Label(new_window, text="SELAMAT DATANG DI TONETIX!", font=("Arial", 18, "bold"))
    welcome_label.pack()

    description_label = tk.Label(new_window, text="Kami adalah platform penjualan tiket konser terbaik untukmu!", font=("Arial", 14), fg="red")
    description_label.pack()

    additional_label = tk.Label(new_window, text="Tonetix menyediakan tiket konser untuk berbagai genre musik populer,\n dengan kualitas suara dan pengalaman konser yang tak terlupakan.\n Dengan Tonetix,\n Anda dapat menjelajahi dan membeli tiket untuk konser-konser terbaik\n dari musisi dan band favorit\n Anda.", font=("Arial", 10))
    additional_label.pack()

    genre_label = tk.Label(new_window, text="Pilih nomor genre musik yang Anda minati:", font=("Arial", 12, "bold"))
    genre_label.pack()

    genre_options = [
        "Jazz",
        "Rock",
        "Pop",
        "Dangdut"
    ]

    for i, option in enumerate(genre_options, start=1):
        genre_button = tk.Button(new_window, text=f"{i}. {option}", width=12, bg="grey", fg="white",
        command=lambda opt=option: genre_selected(opt, new_window))
        genre_button.pack(pady=3)

    exit_button = tk.Button(new_window, text="Keluar", width=12, bg="red", fg="white", command=keluar)
    exit_button.pack()

    additional_label2 = tk.Label(new_window, text="\nUntuk membeli tiket, silakan pilih nomor genre\nmusik yang Anda minati.\nKami akan membantu Anda mendapatkan pengalaman konser yang luar biasa!", font=("Arial", 10))
    additional_label2.pack()

def genre_selected(genre, window):
    if genre == "Jazz":
        show_jazz_concerts(window)
    elif genre == "Rock":
        show_rock_concerts(window)
    elif genre == "Pop":
        show_pop_concerts(window)
    elif genre == "Dangdut":
        show_dangdut_concerts(window)
    elif genre == "Keluar":
        keluar()

def show_jazz_concerts(window):
    new_window = tk.Toplevel(window)
    new_window.title("Daftar Konser Jazz")
    new_window.geometry("600x400")
```

```

new_window.geometry("600x400")

canvas = tk.Canvas(new_window)
scrollbar = tk.Scrollbar(new_window, orient="vertical", command=canvas.yview)
concert_frame = tk.Frame(canvas)

concert_frame.bind(
    "<Configure>",
    lambda e: canvas.configure(
        scrollregion=canvas.bbox("all")
    )
)

canvas.create_window((0, 0), window=concert_frame, anchor="nw")
canvas.configure(yscrollcommand=scrollbar.set)

canvas.pack(side="left", fill="both", expand=True)
scrollbar.pack(side="right", fill="y")

with open('konser_jazz.csv', 'r') as file:
    reader = csv.reader(file)
    next(reader)
    print()
    count = 1
    concerts = []
    for row in reader:
        concerts.append(row)
        concert_label = tk.Label(concert_frame, text=f"\nNomor Konser: {count}\nNama Konser: {row[0]}\nVenue: {row[1]}\nLokasi: {row[2]}\nTanggal: {row[3]}\nHarga:\n - VIP: Rp. {row[7]}\n - Regular: Rp. {row[8]}", font=("Arial", 12, "bold"), justify="center")
        concert_label.pack()
        concert_button = tk.Button(concert_frame, text="Pilih", width=12, bg="grey", fg="white", font=("Arial", 10), command=lambda info=row: buy_ticket(info))
        concert_button.pack()
        count += 1

    back_button = tk.Button(new_window, text="Kembali", width=12, bg="red", fg="white", command=lambda: back_to_welcome(new_window))
    back_button.pack()

def show_rock_concerts(window):
    new_window = tk.Toplevel(window)
    new_window.title("Daftar Konser Rock")
    new_window.geometry("600x400")

    canvas = tk.Canvas(new_window)
    scrollbar = tk.Scrollbar(new_window, orient="vertical", command=canvas.yview)
    concert_frame = tk.Frame(canvas)

    concert_frame.bind(
        "<Configure>",
        lambda e: canvas.configure(
            scrollregion=canvas.bbox("all")
        )
    )

    canvas.create_window((0, 0), window=concert_frame, anchor="nw")
    canvas.configure(yscrollcommand=scrollbar.set)

    canvas.pack(side="left", fill="both", expand=True)
    scrollbar.pack(side="right", fill="y")

    with open('konser_rock.csv', 'r') as file:
        reader = csv.reader(file)
        next(reader)
        print()
        count = 1
        concerts = []
        for row in reader:
            concerts.append(row)
            concert_label = tk.Label(concert_frame, text=f"\nNomor Konser: {count}\nNama Konser: {row[0]}\nVenue: {row[1]}\nLokasi: {row[2]}\nTanggal: {row[3]}\nHarga:\n - VIP: Rp. {row[7]}\n - Regular: Rp. {row[8]}", font=("Arial", 12, "bold"), justify="center")
            concert_label.pack()

```

```

concert_button = tk.Button(concert_frame, text="Pilih", width=12, bg="grey",
                           fg="white", font=("Arial", 10), command=lambda info=row: buy_ticket(info))
concert_button.pack()
count += 1

back_button = tk.Button(new_window, text="Kembali", width=12, bg="red", fg="white",
                        command=lambda: back_to_welcome(new_window))
back_button.pack()

def show_pop_concerts(window):
    new_window = tk.Toplevel(window)
    new_window.title("Daftar Konser Pop")
    new_window.geometry("600x400")

    canvas = tk.Canvas(new_window)
    scrollbar = tk.Scrollbar(new_window, orient="vertical", command=canvas.yview)
    concert_frame = tk.Frame(canvas)

    concert_frame.bind(
        "<Configure>",
        lambda e: canvas.configure(
            scrollregion=canvas.bbox("all")
        )
    )

    canvas.create_window((0, 0), window=concert_frame, anchor="nw")
    canvas.configure(yscrollcommand=scrollbar.set)

    canvas.pack(side="left", fill="both", expand=True)
    scrollbar.pack(side="right", fill="y")

    with open('konser_pop.csv', 'r') as file:
        reader = csv.reader(file)
        next(reader)
        print()
        count = 1
        concerts = []
        for row in reader:
            concerts.append(row)
            concert_label = tk.Label(concert_frame, text=f"\nNomor Konser: {count}\nNama Konser: {row[0]}\nVenue: {row[1]}\nLokasi: {row[2]}\nTanggal: {row[3]}\nHarga:\n - VIP: Rp. {row[7]}\n - Regular: Rp. {row[8]}", font=("Arial", 12, "bold"), justify="center")
            concert_label.pack()
            concert_button = tk.Button(concert_frame, text="Pilih", width=12, bg="grey", fg="white",
                                       font=("Arial", 10), command=lambda info=row: buy_ticket(info))
            concert_button.pack()
            count += 1

    back_button = tk.Button(new_window, text="Kembali", width=12, bg="red", fg="white",
                           command=lambda: back_to_welcome(new_window))
    back_button.pack()

def show_dangdut_concerts(window):
    new_window = tk.Toplevel(window)
    new_window.title("Daftar Konser Dangdut")
    new_window.geometry("600x400")

    canvas = tk.Canvas(new_window)
    scrollbar = tk.Scrollbar(new_window, orient="vertical", command=canvas.yview)
    concert_frame = tk.Frame(canvas)

    concert_frame.bind(
        "<Configure>",
        lambda e: canvas.configure(
            scrollregion=canvas.bbox("all")
        )
    )

    canvas.create_window((0, 0), window=concert_frame, anchor="nw")
    canvas.configure(yscrollcommand=scrollbar.set)

    canvas.pack(side="left", fill="both", expand=True)
    scrollbar.pack(side="right", fill="y")

```

```

scrollbar.pack(side=RIGHT, fill=Y)

with open('konser_dangdut.csv', 'r') as file:
    reader = csv.reader(file)
    next(reader)
    print()
    count = 1
    concerts = []
    for row in reader:
        concerts.append(row)
        concert_label = tk.Label(concert_frame, text=f"\nNomor Konser: {count}\nNama Konser: {row[0]}\nVenue: {row[1]}\nLokasi: {row[2]}\nTanggal: {row[3]}\nHarga:\n - VIP: Rp. {row[7]}\n - Regular: Rp. {row[8]}", font=("Arial", 12, "bold"), justify="center")
        concert_label.pack()
        concert_button = tk.Button(concert_frame, text="Pilih", width=12, bg="grey", fg="white", font=("Arial", 10), command=lambda info=row: buy_ticket(info))
        concert_button.pack()
        count += 1

    back_button = tk.Button(new_window, text="Kembali", width=12, bg="red", fg="white", command=lambda: back_to_welcome(new_window))
    back_button.pack() #baru ini yang diedit

def buy_ticket(concert_info):
    line_up_window = tk.Toplevel()
    line_up_window.title("Line-up Konser")
    line_up_window.geometry("600x400")

    label_font = ("Arial", 16, "bold")

    label_lineup1 = tk.Label(line_up_window, text=f"\nLine-up 1:\n{concert_info[4]}", font=label_font, justify="center")
    label_lineup1.pack(pady=10)
    label_lineup2 = tk.Label(line_up_window, text=f"\nLine-up 2:\n{concert_info[5]}", font=label_font, justify="center")
    label_lineup2.pack(pady=10)
    label_lineup3 = tk.Label(line_up_window, text=f"\nLine-up 3:\n{concert_info[6]}", font=label_font, justify="center")
    label_lineup3.pack(pady=10)

    # Membuat frame untuk tombol
    button_frame = tk.Frame(line_up_window)
    button_frame.pack(pady=25)

    # Membuat tombol untuk membeli tiket
    buy_button = tk.Button(button_frame, text="Beli Tiket", width=12, font=("Arial", 10), command=lambda: buy_ticket_confirmation(concert_info))
    buy_button.grid(row=0, column=0, padx=(0, 10))

    back_button = tk.Button(button_frame, text="Kembali", width=12, font=("Arial", 10), bg="red", fg="white", command=lambda: back_to_welcome(window))
    back_button.grid(row=0, column=1, padx=(10, 0))

def buy_ticket_confirmation(concert_info):
    # Menampilkan konfirmasi pembelian tiket
    confirmation_window = tk.Toplevel()
    confirmation_window.title("Konfirmasi Pembelian Tiket")
    confirmation_window.geometry("600x100")

    confirmation_label = tk.Label(confirmation_window, text="Apakah Anda ingin membeli tiket?", font=("Arial", 12))
    confirmation_label.pack(pady=20)

    # Membuat frame untuk tombol
    button_frame = tk.Frame(confirmation_window)
    button_frame.pack()

def show_ticket_prices():
    ticket_prices_window = tk.Toplevel()
    ticket_prices_window.title("Harga Tiket")
    ticket_prices_window.geometry("600x300")

```

```

ticket_prices_window.geometry("600x300")

vip_price_label = tk.Label(ticket_prices_window, text="\nHarga Tiket VIP:\n Rp
{}".format(concert_info[7]), font=("Arial", 12))
vip_price_label.pack(pady=10)

regular_price_label = tk.Label(ticket_prices_window, text="Harga Tiket Reguler:\n Rp
{}".format(concert_info[8]), font=("Arial", 12))
regular_price_label.pack(pady=10)

ticket_selection_label = tk.Label(ticket_prices_window, text="Pilih tiket yang ingin Anda
beli:", font=("Arial", 12, "bold"))
ticket_selection_label.pack(pady=10)

# Membuat frame untuk tombol tiket
ticket_button_frame = tk.Frame(ticket_prices_window)
ticket_button_frame.pack()

vip_ticket_button = tk.Button(ticket_button_frame, text="Tiket VIP", width=12,
command=purchase_vip_ticket)
vip_ticket_button.pack(side=tk.LEFT, padx=10)

# Tombol untuk membeli tiket reguler
regular_ticket_button = tk.Button(ticket_button_frame, text="Tiket Reguler", width=12,
command=purchase_regular_ticket)
regular_ticket_button.pack(side=tk.LEFT, padx=10)

def purchase_vip_ticket():
def cancel_purchase():
    purchase_window.destroy()

purchase_window = tk.Toplevel()
purchase_window.title("Pembelian Tiket VIP")
purchase_window.geometry("600x400")

nama_label = tk.Label(purchase_window, text="Nama:")
nama_label.pack()
nama_entry = tk.Entry(purchase_window, width=30)
nama_entry.pack()

nik_label = tk.Label(purchase_window, text="NIK:")
nik_label.pack()
nik_entry = tk.Entry(purchase_window, width=30)
nik_entry.pack()

email_label = tk.Label(purchase_window, text="Email:")
email_label.pack()
email_entry = tk.Entry(purchase_window, width=30)
email_entry.pack()

nomor_rekening_label = tk.Label(purchase_window, text="Nomor Rekening:")
nomor_rekening_label.pack()
nomor_rekening_entry = tk.Entry(purchase_window, width=30)
nomor_rekening_entry.pack()

jumlah_tiket_label = tk.Label(purchase_window, text="Jumlah Tiket:")
jumlah_tiket_label.pack()
jumlah_tiket_entry = tk.Entry(purchase_window, width=30)
jumlah_tiket_entry.pack()

def generate_kode_unik(length=9):
    characters = string.ascii_letters + string.digits
    kode_unik = ''
    for _ in range(length):
        if len(kode_unik) < length - 1:
            kode_unik += random.choice(characters)
        else:
            kode_unik += random.choice(string.digits)
    return kode_unik

def lihat_tiket():
    tiket_window = tk.Toplevel()
    tiket_window.title("Tiket")
    tiket_window.geometry("600x350")
    tiket_window.configure(bg="skyblue")

```

```

    "bold"))
    tiket_label = tk.Label(tiket_window, text="Tiket Anda", bg="skyblue", font=("Arial", 16,
    tiket_label.pack(pady=10)

    nama = nama_entry.get()
    nik = nik_entry.get()
    email = email_entry.get()
    jumlah_tiket = int(jumlah_tiket_entry.get())
    kode_unik = generate_kode_unik()

    nama_label = tk.Label(tiket_window, text="Nama : {}".format(nama), bg="skyblue", font=
    ("Arial", 12))
    nama_label.pack(pady=5, anchor="w")

    nik_label = tk.Label(tiket_window, text="NIK : {}".format(nik), bg="skyblue", font=
    ("Arial", 12))
    nik_label.pack(pady=5, anchor="w")

    email_label = tk.Label(tiket_window, text="Email : {}".format(email), bg="skyblue", font=
    ("Arial", 12))
    email_label.pack(pady=5, anchor="w")

    jumlah_tiket_label = tk.Label(tiket_window, text="Jumlah Tiket : {}".format(jumlah_tiket),
    bg="skyblue", font=("Arial", 12))
    jumlah_tiket_label.pack(pady=5, anchor="w")

    kode_unik_label = tk.Label(tiket_window, text="Kode Tiket : {}".format(kode_unik),
    bg="skyblue", font=("Arial", 12, "bold"))
    kode_unik_label.pack(pady=5, anchor="w")

    printout_label = tk.Label(tiket_window, text="Silahkan tunjukkan E-Ticket ini di loket
    pembelian tiket untuk mendapatkan tiket Anda. \nTerima kasih. Selamat menikmati konser!", bg="skyblue",
    font=("Arial", 10, "italic"))
    printout_label.pack(pady=10)

    exit_button = tk.Button(tiket_window, text="Exit", width=12, bg="red", fg="white",
    command=keluar)
    exit_button.pack(pady=10)

def submit_purchase():
    nama = nama_entry.get()
    nik = nik_entry.get()
    email = email_entry.get()
    nomor_rekening = nomor_rekening_entry.get()
    jumlah_tiket = int(jumlah_tiket_entry.get())
    tanggal_waktu = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    with open('pembelian_tiket.csv', 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([tanggal_waktu, nama, nik, email, nomor_rekening, jumlah_tiket,"VIP"])

    harga_tiket = int(concert_info[8])
    total_harga = harga_tiket * jumlah_tiket

    total_harga_label = tk.Label(purchase_window, text="Total Harga: Rp
    {}".format(total_harga), font=("Arial", 12, "bold"))
    total_harga_label.pack()

    nominal_pembayaran_label = tk.Label(purchase_window, text="Nominal Pembayaran : (Rp.) ")
    nominal_pembayaran_label.pack()

    nominal_pembayaran_entry = tk.Entry(purchase_window)
    nominal_pembayaran_entry.pack()

def submit_payment():
    nominal_pembayaran = int(nominal_pembayaran_entry.get())

    if nominal_pembayaran == total_harga:
        payment_success_window = tk.Toplevel()
        payment_success_window.title("Pembayaran Berhasil")
        payment_success_window.geometry("400x200")

```

```
retry_label = tk.Label(retry_payment_window, text="\nPembayaran belum selesai.\nMohon masukkan nominal pembayaran yang sesuai.",font=("Arial", 12, "bold"))
retry_label.pack()

submit_button = tk.Button(purchase_window, text="Submit Pembayaran", width=20,
command=submit_payment)
submit_button.pack(pady=10)

button_frame = tk.Frame(purchase_window)
button_frame.pack(pady=10)

submit_button = tk.Button(button_frame, text="Beli", width=12, command=submit_purchase)
submit_button.pack(side=tk.LEFT, padx=5)

cancel_button = tk.Button(button_frame, text="Cancel",bg="red", fg="white", width=12,
command=cancel_purchase)
cancel_button.pack(side=tk.LEFT, padx=5)

def purchase_regular_ticket():
def cancel_purchase():
    purchase_window.destroy()

purchase_window = tk.Toplevel()
purchase_window.title("Pembelian Tiket VIP")
purchase_window.geometry("600x400")

nama_label = tk.Label(purchase_window, text="Nama:")
nama_label.pack()
nama_entry = tk.Entry(purchase_window, width=30)
nama_entry.pack()

nik_label = tk.Label(purchase_window, text="NIK:")
nik_label.pack()
nik_entry = tk.Entry(purchase_window, width=30)
nik_entry.pack()

email_label = tk.Label(purchase_window, text="Email:")
email_label.pack()
email_entry = tk.Entry(purchase_window, width=30)
email_entry.pack()

nomor_rekening_label = tk.Label(purchase_window, text="Nomor Rekening:")
nomor_rekening_label.pack()
nomor_rekening_entry = tk.Entry(purchase_window, width=30)
nomor_rekening_entry.pack()

jumlah_tiket_label = tk.Label(purchase_window, text="Jumlah Tiket:")
jumlah_tiket_label.pack()
jumlah_tiket_entry = tk.Entry(purchase_window, width=30)
jumlah_tiket_entry.pack()

def generate_kode_unik(length=9):
    characters = string.ascii_letters + string.digits
    kode_unik = ''
    for _ in range(length):
        if len(kode_unik) < length - 1:
            kode_unik += random.choice(characters)
        else:
            kode_unik += random.choice(string.digits)
    return kode_unik

def lihat_tiket():
    tiket_window = tk.Toplevel()
    tiket_window.title("Tiket")
    tiket_window.geometry("600x350")
    tiket_window.configure(bg="skyblue")

    tiket_label = tk.Label(tiket_window, text="Tiket Anda", bg="skyblue", font=("Arial", 16,
"bold"))
    tiket_label.pack(pady=10)
```

```
nama = nama_entry.get()
nik = nik_entry.get()
email = email_entry.get()
jumlah_tiket = int(jumlah_tiket_entry.get())
kode_unik = generate_kode_unik()

nama_label = tk.Label(tiket_window, text="Nama : {}".format(nama), bg="skyblue", font=("Arial", 12))
nama_label.pack(pady=5, anchor="w")

nik_label = tk.Label(tiket_window, text="NIK : {}".format(nik), bg="skyblue", font=("Arial", 12))
nik_label.pack(pady=5, anchor="w")

email_label = tk.Label(tiket_window, text="Email : {}".format(email), bg="skyblue", font=("Arial", 12))
email_label.pack(pady=5, anchor="w")

jumlah_tiket_label = tk.Label(tiket_window, text="Jumlah Tiket : {}".format(jumlah_tiket), bg="skyblue", font=("Arial", 12))
jumlah_tiket_label.pack(pady=5, anchor="w")

kode_unik_label = tk.Label(tiket_window, text="Kode Tiket : {}".format(kode_unik), bg="skyblue", font=("Arial", 12, "bold"))
kode_unik_label.pack(pady=5, anchor="w")

printout_label = tk.Label(tiket_window, text="Silahkan tunjukkan E-Ticket ini di loket pembelian tiket untuk mendapatkan tiket Anda. \nTerima kasih. Selamat menikmati konser!", bg="skyblue", font=("Arial", 10, "italic"))
printout_label.pack(pady=10)

exit_button = tk.Button(tiket_window, text="Exit", width=12, bg="red", fg="white", command=keluar)
exit_button.pack(pady=10)

def submit_purchase():
    nama = nama_entry.get()
    nik = nik_entry.get()
    email = email_entry.get()
    nomor_rekening = nomor_rekening_entry.get()
    jumlah_tiket = int(jumlah_tiket_entry.get())
    tanggal_waktu = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    with open('pembelian_tiket.csv', 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([tanggal_waktu, nama, nik, email, nomor_rekening, jumlah_tiket, "Reguler"])

    harga_tiket = int(concert_info[8])
    total_harga = harga_tiket * jumlah_tiket

    total_harga_label = tk.Label(purchase_window, text="Total Harga: Rp {}".format(total_harga), font=("Arial", 12, "bold"))
    total_harga_label.pack()

    nominal_pembayaran_label = tk.Label(purchase_window, text="Nominal Pembayaran : (Rp. )")
    nominal_pembayaran_label.pack()

    nominal_pembayaran_entry = tk.Entry(purchase_window)
    nominal_pembayaran_entry.pack()

def submit_payment():
    nominal_pembayaran = int(nominal_pembayaran_entry.get())

    if nominal_pembayaran == total_harga:
        payment_success_window = tk.Toplevel()
        payment_success_window.title("Pembayaran Berhasil")
        payment_success_window.geometry("400x200")

        success_label = tk.Label(payment_success_window, text="\nPembayaran telah diselesaikan.", font=("Arial", 12, "bold"))
        success_label.pack()
```

```

    success_label = tk.Label(payment_success_window, text="\nPembayaran telah
diselesaikan.\nKembalian: Rp {}, \nKembalian akan dikirim ke Nomor Rekening
Anda".format(kembalian),font=("Arial", 12, "bold"))
    success_label.pack()

    lihat_tiket_button = tk.Button(payment_success_window, text="Lihat Tiket",
width=12, command=lihat_tiket)
        lihat_tiket_button.pack(side="bottom", anchor="s", pady=10)
    else:
        retry_payment_window = tk.Toplevel()
        retry_payment_window.title("Pembayaran Belum Selesai")
        retry_payment_window.geometry("400x200")

        retry_label = tk.Label(retry_payment_window, text="\nPembayaran belum selesai.\n
Mohon masukkan nominal pembayaran yang sesuai.",font=("Arial", 12, "bold"))
        retry_label.pack()

        submit_button = tk.Button(purchase_window, text="Submit Pembayaran", width=20,
command=submit_payment)
        submit_button.pack(pady=10)

        button_frame = tk.Frame(purchase_window)
        button_frame.pack(pady=10)

        submit_button = tk.Button(button_frame, text="Beli", width=12, command=submit_purchase)
        submit_button.pack(side=tk.LEFT, padx=5)

        cancel_button = tk.Button(button_frame, text="Cancel",bg="red", fg="white", width=12,
command=cancel_purchase)
        cancel_button.pack(side=tk.LEFT, padx=5)

        # Membuat tombol untuk konfirmasi pembelian tiket
        confirm_button = tk.Button(button_frame, text="Ya", width=12, command=show_ticket_prices)
        confirm_button.pack(side=tk.LEFT, padx=30)

        # Membuat tombol untuk membatalkan pembelian tiket
        cancel_button = tk.Button(button_frame, text="Tidak", width=12,
command=confirmation_window.destroy)
        cancel_button.pack(side=tk.LEFT, padx=10)

def keluar():
    window.destroy()

def back_to_welcome(window):
    welcome_message()

# Membuat jendela utama
window = tk.Tk()
window.title("ToneTix")
window.geometry("600x400")

# Menambahkan gambar
image = Image.open("TampilanAwal.png")
image = image.resize((597, 350), Image.ANTIALIAS)
photo = ImageTk.PhotoImage(image)
label = tk.Label(window, image=photo)
label.grid(row=0, column=0, columnspan=2)

# Menambahkan tombol
masuk_button = tk.Button(window, text="Start", bg="grey", fg="white", command=welcome_message,
width=12, height=1)
masuk_button.grid(row=1, column=1)

keluar_button = tk.Button(window, text="Exit", bg="grey", fg="white", command=keluar, width=12,
height=1)
keluar_button.grid(row=1, column=0, pady=10)

# Menjalankan aplikasi
window.mainloop()

```

Bab IV

Hasil Running Python

```
● ● ●  
=====SELAMAT DATANG DI TONETIX!  
Kami adalah platform penjualan tiket konser terbaik untukmu!  
=====Tonetix menyediakan tiket konser untuk berbagai genre musik populer,  
dengan kualitas suara dan pengalaman konser yang tak terlupakan.  
Dengan Tonetix, Anda dapat menjelajahi dan membeli tiket untuk  
konser-konser terbaik dari musisi dan band favorit Anda.  
=====Untuk membeli tiket, silakan pilih nomor genre musik yang Anda minati.  
Kami akan membantu Anda mendapatkan pengalaman konser yang luar biasa!  
=====
```

Hasil setelah program di running akan muncul “SELAMAT DATANG DI TONETIX” dan penjelasan secara singkat tentang ToneTix.

```
Pilih nomor genre musik yang Anda minati:  
1. Jazz  
2. Rock  
3. Pop  
4. Dangdut  
5. Keluar  
=====
```

Setelah itu pengguna diminta untuk memilih genre yang diinginkan.

```

Masukkan nomor genre musik yang Anda pilih: 3
Daftar Konser Pop yang Tersedia:
=====
Nomor Konser : 1
Nama Konser : Konser Pop Extravaganza
Venue : Gelora Bung Karno
Lokasi : Jakarta
Tanggal : 15 Juli 2023
Harga : - VIP : Rp. 500000
        - Reguler : Rp. 250000
=====
Nomor Konser : 2
Nama Konser : Konser Pop Hits Plaza Senayan
Venue : Plaza Senayan
Lokasi : Jakarta
Tanggal : 25 Agustus 2023
Harga : - VIP : Rp. 450000
        - Reguler : Rp. 200000
=====
Nomor Konser : 3
Nama Konser : Konser Pop Dance Fever
Venue : ICE BSD
Lokasi : Tangerang
Tanggal : 5 September 2023
Harga : - VIP : Rp. 400000
        - Reguler : Rp. 180000
=====
Nomor Konser : 4
Nama Konser : Konser Pop Legends
Venue : Stadium Utama
Lokasi : Bandung
Tanggal : 10 Oktober 2023
Harga : - VIP : Rp. 500000
        - Reguler : Rp. 250000
=====
Nomor Konser : 5
Nama Konser : Konser Pop Ballad Serenade
Venue : Vanue A
Lokasi : Surabaya
Tanggal : 20 November 2023
Harga : - VIP : Rp. 350000
        - Reguler : Rp. 150000
=====
Nomor Konser : 6
Nama Konser : Konser Pop Indie Showcase
Venue : The Kasablanka
Lokasi : Jakarta
Tanggal : 5 Desember 2023
Harga : - VIP : Rp. 400000
        - Reguler : Rp. 180000
=====
Nomor Konser : 7
Nama Konser : Konser Pop Melayu Sensasi
Venue : Colosseum
Lokasi : Jakarta
Tanggal : 10 Januari 2024
Harga : - VIP : Rp. 450000
        - Reguler : Rp. 200000
=====
Nomor Konser : 8
Nama Konser : Konser Pop Dangdut Party
Venue : Diamond Club
Lokasi : Bandung
Tanggal : 15 Februari 2024
Harga : - VIP : Rp. 350000
        - Reguler : Rp. 150000
=====
Nomor Konser : 9
Nama Konser : Konser Pop R&B Soulful
Venue : The Foundry
Lokasi : Surabaya
Tanggal : 20 Maret 2024
Harga : - VIP : Rp. 400000
        - Reguler : Rp. 180000
=====
Nomor Konser : 10
Nama Konser : Konser Pop Acoustic Serenity
Venue : Rooftop Garden
Lokasi : Bali
Tanggal : 5 April 2024
Harga : - VIP : Rp. 350000
        - Reguler : Rp. 150000
=====
```

Setelah memilih genre yang diinginkan maka program akan memunculkan 10 rekomendasi konser yang akan dapat dilihat mendatang.

Masukkan nomor konser yang Anda pilih: 7

Anda memilih konser: Konser Pop Melayu Sensasi

Line-up:

- Line-up 1 : Anji
- Line-up 2 : Armada
- Line-up 3 : Wali Band

Harga Tiket:

- VIP : Rp. 450000
- Reguler : Rp. 200000

Setelah pengguna memilih nomer konser yang diinginkan maka program akan memunculkan konser yang anda pilih yang berisi line up dan harga tiket VIP dan reguler.

Apakah Anda ingin membeli tiket? (ya/tidak): ya

Pilih jenis tiket (VIP/Reguler): vip

Masukkan Nama : kay

Masukkan NIK : 3332154687

Masukkan Email : kay@gmali.com

Masukkan jumlah tiket yang ingin Anda beli : 5

Masukkan No. Rekening : 45621378

Terima kasih, tiket VIP Anda telah berhasil dibeli.

Total harga yang harus dibayar: Rp. 2250000

Masukkan nominal pembayaran: 2250000

Pembayaran cukup. Tiket Anda telah diterbitkan.

Program menanyakan apakah pengguna ingin membeli tiket dan kategori tiket apa yang akan dipilih. Setelah menjawab pertanyaan tersebut dengan valid maka pengguna diminta untuk memasukan Nama, NIK, Email, Jumlah tiket yang dibeli, dan No rekening. Setelah memasukan semuanya dengan valid maka program akan memunculkan total harga yang harus dibayar. Setelah itu pengguna memasukan nominal sesuai harga tiket dan tiket pengguna telah terbit.

===== TIKET KONSER =====

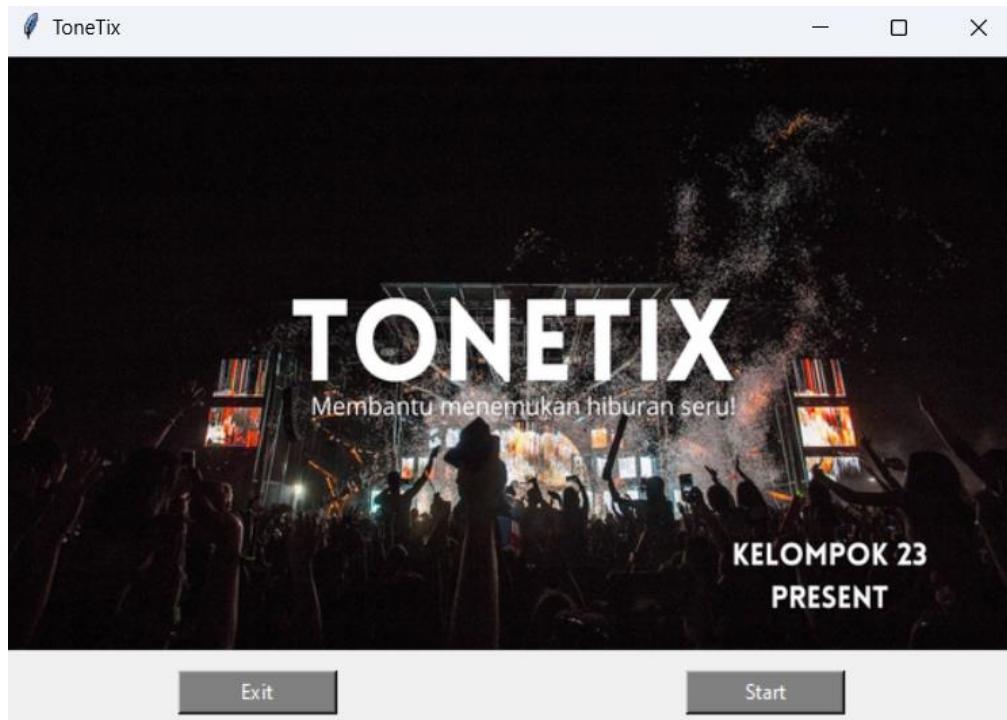
Nama : kay
NIK : 3332154687
Email : kay@gmali.com
Jenis Tiket : VIP
Jumlah Tiket : 5
Kode Tiket : 7274664132

=====
Silaikan tunjukkan E-Ticket ini di loket pembelian tiket

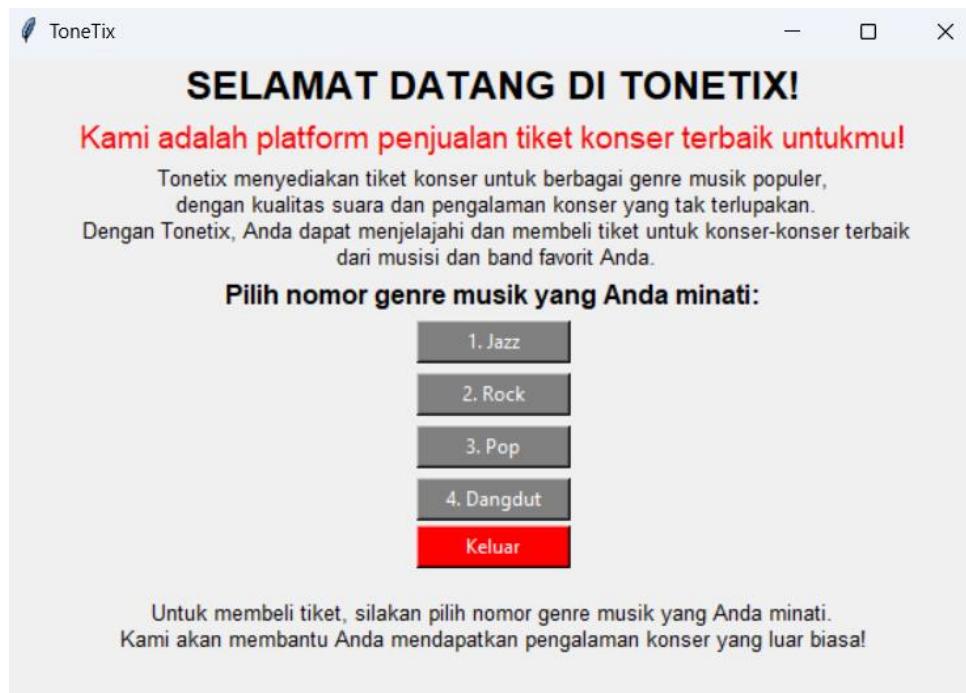
Untuk mendapatkan tiket Anda

Terima kasih. Selamat menikmati konser!

Hasil tiket yang telah terbit. Pengguna diminta menunjukan E-Ticket di loket pembelian tiket untuk mendapatkan tiket anda.



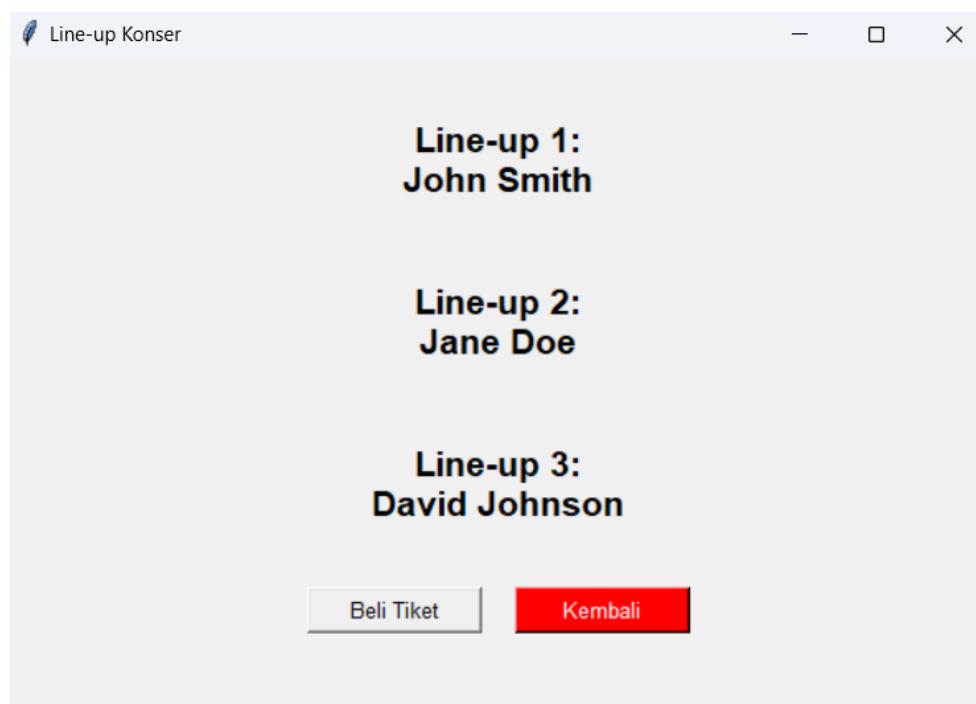
GUI di running maka akan muncul cover depan yang memiliki pilihan Exit dan Start.



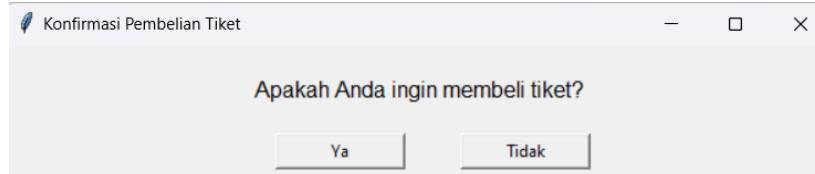
Jika memilih start maka akan muncul pilihan genre. Pengguna diminta untuk memilih genre yang diinginkan.



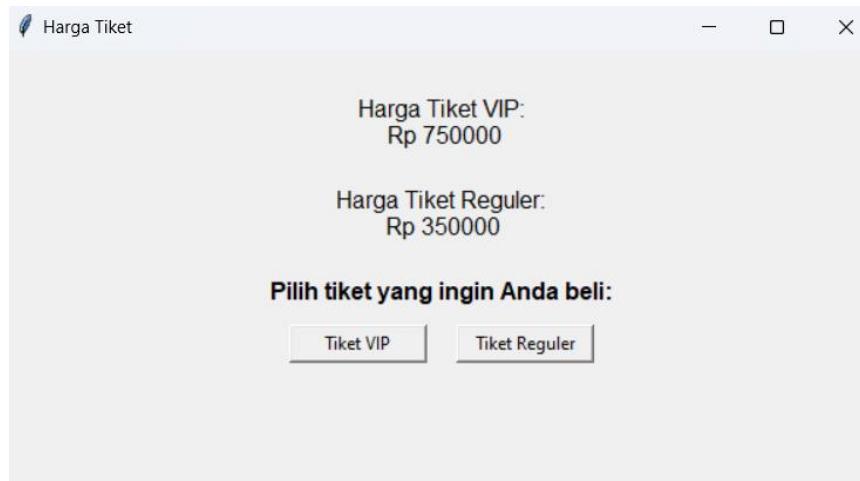
Setelah memilih genre yang dipilih maka program akan memunculkan 10 rekomendasi konser yang dapat dilihat mendatang.



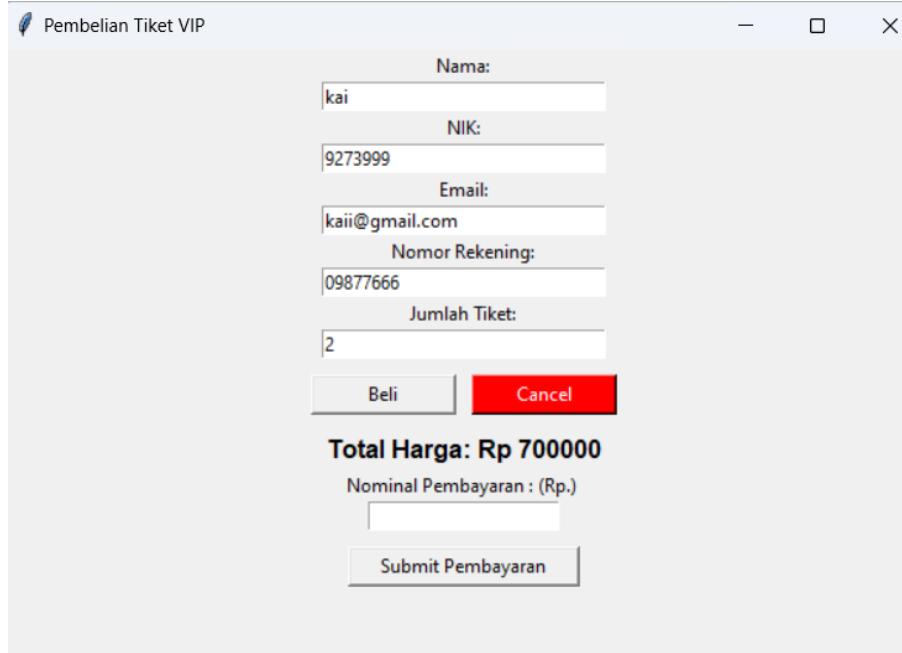
Setelah memilih genre yang diinginkan maka akan menampilkan line up.



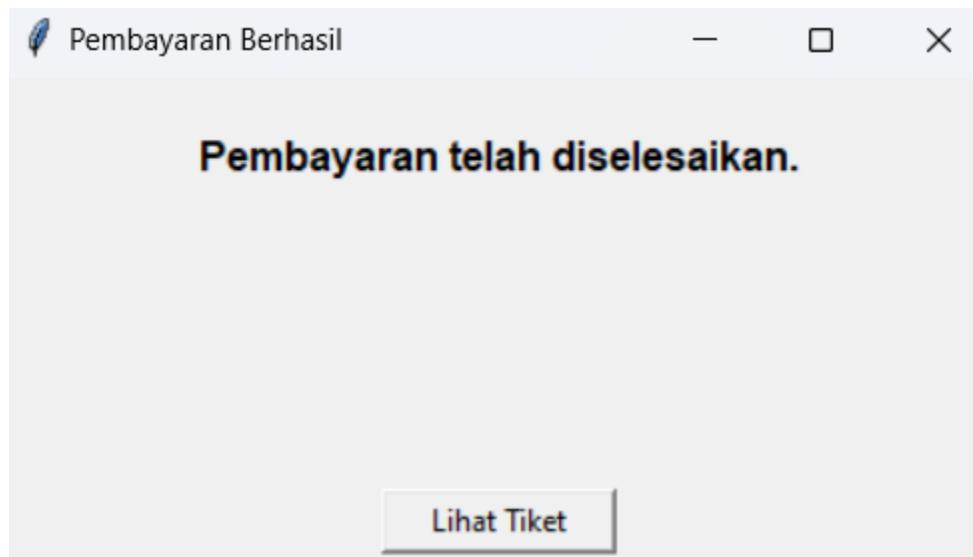
Jika memilih beli tiket maka akan ditanya apakah anda akan membeli tiket.



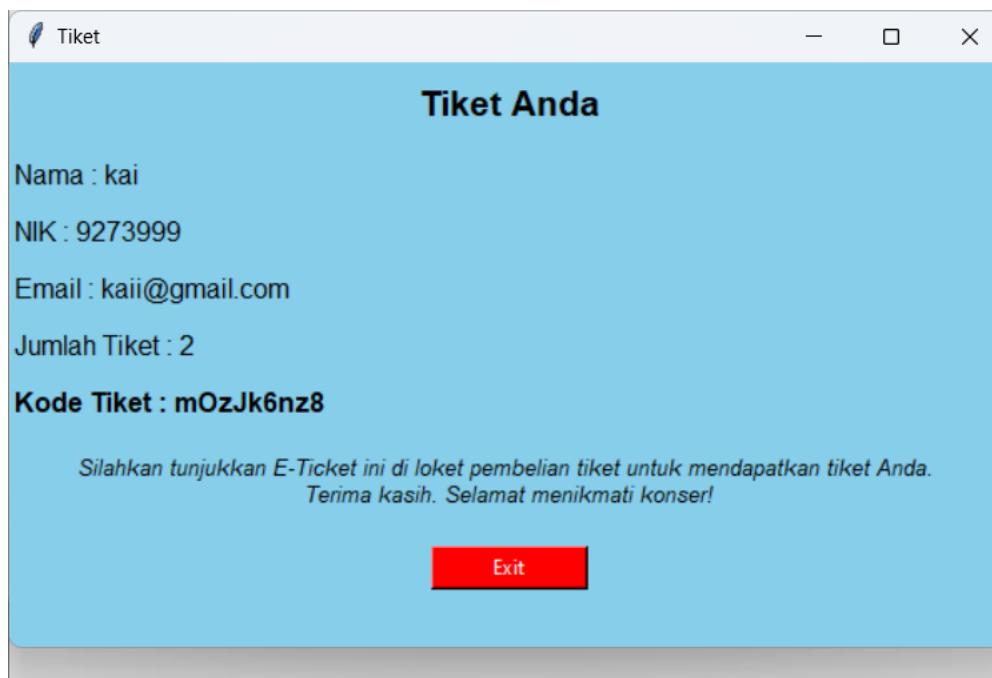
Jika pada jendela sebelumnya memilih ya maka akan ditampilkan pilihan kategori tiket yang dapat dibeli.



Setelah pengguna memilih kategori tiket maka akan diminta untuk mengisi Nama, NIK, Email, No rekening, dan jumlah tiket yang ingin dibeli. jika memilih beli maka pengguna diminta memasukan total pembayaran yang harus dibayarkan.



Setelah pengguna menyelesaikan pembayaran maka akan menampilkan pembayaran telah selesai.



Program akan menampilkan bahwa tiket berhasil dibeli dan ada penjelasan agar pengguna dapat mendapatkan tiket.