class Tiket:

def \_\_init\_\_(self, penumpang, kapasitas):

self.penumpang = penumpang

self.kapasitas = kapasitas

def tambah\_penumpang(self, jumlah):

if self.penumpang + jumlah <= self.kapasitas:

self.penumpang += jumlah

print(f"{jumlah} penumpang berhasil ditambahkan. Total penumpang: {self.penumpang}.")

else:

print("Kapasitas penuh, tidak bisa menambah penumpang.")

def info\_penumpang(self):

return f"Total penumpang: {self.penumpang}/{self.kapasitas}"

class TiketKapal(Tiket):

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas)

def turunkan\_penumpang(self, jumlah):

if self.penumpang - jumlah >= 0:

self.penumpang -= jumlah

print(f"{jumlah} penumpang berhasil diturunkan dari kapal. Total penumpang: {self.penumpang}.")

else:

print("Tidak ada cukup penumpang untuk diturunkan.")

class TiketKapalDenganPembatalan(TiketKapal):

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas)

def batalkan\_tiket(self, jumlah):

if self.penumpang - jumlah >= 0:

self.penumpang -= jumlah

print(f"{jumlah} tiket berhasil dibatalkan dari kapal. Total penumpang: {self.penumpang}.")

else:

print("Tidak ada cukup penumpang untuk dibatalkan.")

class TiketPesawat(Tiket):

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas)

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang pesawat...")

super().tambah\_penumpang(jumlah)

class TiketKereta(Tiket):

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas)

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang kereta...")

super().tambah\_penumpang(jumlah)

class TiketBus(Tiket):

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas)

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang bus...")

super().tambah\_penumpang(jumlah)

class TiketFerry(Tiket):

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas)

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang ferry...")

super().tambah\_penumpang(jumlah)

# Contoh penggunaan

def main():

tiket1 = TiketKapalDenganPembatalan(penumpang=50, kapasitas=100)

tiket2 = TiketPesawat(penumpang=80, kapasitas=150)

tiket3 = TiketKereta(penumpang=40, kapasitas=80)

tiket4 = TiketBus(penumpang=30, kapasitas=50)

tiket5 = TiketFerry(penumpang=20, kapasitas=40)

# Menggunakan polymorphism

for tiket in [tiket1, tiket2, tiket3, tiket4, tiket5]:

tiket.tambah\_penumpang(10)

print(tiket.info\_penumpang())

# Melakukan operasi pada tiket kapal dengan pembatalan

tiket1.turunkan\_penumpang(10)

tiket1.batalkan\_tiket(5)

# Menjalankan program

main()

PENJELASAN

// Kelas dasar yang merepresentasikan tiket transportasi

class Tiket:

// Konstruktor untuk inisialisasi penumpang dan kapasitas

def \_\_init\_\_(self, penumpang, kapasitas):

self.penumpang = penumpang // Menyimpan jumlah penumpang saat ini

self.kapasitas = kapasitas // Menyimpan kapasitas maksimum

// Metode untuk menambah penumpang

def tambah\_penumpang(self, jumlah):

// Mengecek apakah penumpang dapat ditambahkan tanpa melebihi kapasitas

if self.penumpang + jumlah <= self.kapasitas:

self.penumpang += jumlah // Menambahkan jumlah penumpang

print(f"{jumlah} penumpang berhasil ditambahkan. Total penumpang: {self.penumpang}.")

else:

print("Kapasitas penuh, tidak bisa menambah penumpang.") // Pesan jika kapasitas penuh

// Metode untuk mendapatkan informasi jumlah penumpang

def info\_penumpang(self):

return f"Total penumpang: {self.penumpang}/{self.kapasitas}" // Mengembalikan informasi penumpang

// Kelas yang merepresentasikan tiket kapal

class TiketKapal(Tiket):

// Konstruktor untuk inisialisasi tiket kapal

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas) // Memanggil konstruktor kelas dasar

// Metode untuk menurunkan penumpang dari kapal

def turunkan\_penumpang(self, jumlah):

// Mengecek apakah ada cukup penumpang untuk diturunkan

if self.penumpang - jumlah >= 0:

self.penumpang -= jumlah // Mengurangi jumlah penumpang

print(f"{jumlah} penumpang berhasil diturunkan dari kapal. Total penumpang: {self.penumpang}.")

else:

print("Tidak ada cukup penumpang untuk diturunkan.") // Pesan jika penumpang tidak cukup

// Kelas yang menambahkan fitur pembatalan tiket untuk kapal

class TiketKapalDenganPembatalan(TiketKapal):

// Konstruktor untuk inisialisasi tiket kapal dengan pembatalan

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas) // Memanggil konstruktor kelas dasar

// Metode untuk membatalkan tiket penumpang

def batalkan\_tiket(self, jumlah):

// Mengecek apakah ada cukup penumpang untuk dibatalkan

if self.penumpang - jumlah >= 0:

self.penumpang -= jumlah // Mengurangi jumlah penumpang

print(f"{jumlah} tiket berhasil dibatalkan dari kapal. Total penumpang: {self.penumpang}.")

else:

print("Tidak ada cukup penumpang untuk dibatalkan.") // Pesan jika penumpang tidak cukup

// Kelas untuk tiket pesawat

class TiketPesawat(Tiket):

// Konstruktor untuk inisialisasi tiket pesawat

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas) // Memanggil konstruktor kelas dasar

// Metode untuk menambah penumpang, dengan logika tambahan untuk pesawat

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang pesawat...") // Pesan log untuk proses penambahan

super().tambah\_penumpang(jumlah) // Memanggil metode tambah\_penumpang dari kelas dasar

// Kelas untuk tiket kereta

class TiketKereta(Tiket):

// Konstruktor untuk inisialisasi tiket kereta

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas) // Memanggil konstruktor kelas dasar

// Metode untuk menambah penumpang, dengan logika tambahan untuk kereta

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang kereta...") // Pesan log untuk proses penambahan

super().tambah\_penumpang(jumlah) // Memanggil metode tambah\_penumpang dari kelas dasar

// Kelas untuk tiket bus

class TiketBus(Tiket):

// Konstruktor untuk inisialisasi tiket bus

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas) // Memanggil konstruktor kelas dasar

// Metode untuk menambah penumpang, dengan logika tambahan untuk bus

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang bus...") // Pesan log untuk proses penambahan

super().tambah\_penumpang(jumlah) // Memanggil metode tambah\_penumpang dari kelas dasar

// Kelas untuk tiket ferry

class TiketFerry(Tiket):

// Konstruktor untuk inisialisasi tiket ferry

def \_\_init\_\_(self, penumpang, kapasitas):

super().\_\_init\_\_(penumpang, kapasitas) // Memanggil konstruktor kelas dasar

// Metode untuk menambah penumpang, dengan logika tambahan untuk ferry

def tambah\_penumpang(self, jumlah):

print("Memproses penambahan penumpang ferry...") // Pesan log untuk proses penambahan

super().tambah\_penumpang(jumlah) // Memanggil metode tambah\_penumpang dari kelas dasar

// Fungsi utama untuk contoh penggunaan

def main():

// Membuat objek dari berbagai kelas tiket

tiket1 = TiketKapalDenganPembatalan(penumpang=50, kapasitas=100) // Objek tiket kapal dengan pembatalan

tiket2 = TiketPesawat(penumpang=80, kapasitas=150) // Objek tiket pesawat

tiket3 = TiketKereta(penumpang=40, kapasitas=80) // Objek tiket kereta

tiket4 = TiketBus(penumpang=30, kapasitas=50) // Objek tiket bus

tiket5 = TiketFerry(penumpang=20, kapasitas=40) // Objek tiket ferry

// Menggunakan polymorphism untuk menambah penumpang

for tiket in [tiket1, tiket2, tiket3, tiket4, tiket5]:

tiket.tambah\_penumpang(10) // Menambahkan 10 penumpang pada setiap tiket

print(tiket.info\_penumpang()) // Menampilkan informasi penumpang

// Melakukan operasi pada tiket kapal dengan pembatalan

tiket1.turunkan\_penumpang(10) // Menurunkan 10 penumpang dari tiket kapal

tiket1.batalkan\_tiket(5) // Membatalkan 5 tiket dari tiket kapal

// Menjalankan program

main() // Memanggil fungsi utama untuk menjalankan seluruh program

Kode ini merupakan implementasi dari sistem manajemen tiket untuk berbagai jenis transportasi menggunakan pemrograman berorientasi objek (OOP) dalam Python. Berikut adalah penjelasan dari setiap bagian kode:

**Kelas Tiket**

* **\_\_init\_\_(self, penumpang, kapasitas)**: Ini adalah konstruktor yang menginisialisasi objek Tiket. Parameter penumpang menyimpan jumlah penumpang saat ini, dan kapasitas menyimpan kapasitas maksimum yang dapat ditampung.
* **tambah\_penumpang(self, jumlah)**: Metode ini digunakan untuk menambah jumlah penumpang. Jika penumpang yang ada ditambah jumlah yang ingin ditambahkan tidak melebihi kapasitas, jumlah penumpang diperbarui, dan pesan keberhasilan ditampilkan. Jika tidak, pesan kapasitas penuh ditampilkan.
* **info\_penumpang(self)**: Metode ini mengembalikan string yang berisi informasi tentang total penumpang saat ini dibandingkan dengan kapasitas maksimum.

**Kelas TiketKapal**

* **\_\_init\_\_(self, penumpang, kapasitas)**: Kelas ini mewarisi dari kelas Tiket. Ini memanggil konstruktor kelas dasar (Tiket) untuk menginisialisasi penumpang dan kapasitas.
* **turunkan\_penumpang(self, jumlah)**: Metode ini digunakan untuk menurunkan penumpang dari kapal. Jika jumlah penumpang yang ingin diturunkan masih tersisa di kapal, jumlah penumpang akan dikurangi. Jika tidak, akan ditampilkan pesan bahwa tidak ada cukup penumpang untuk diturunkan.

**Kelas TiketKapalDenganPembatalan**

* **\_\_init\_\_(self, penumpang, kapasitas)**: Kelas ini mewarisi dari TiketKapal dan memanggil konstruktor untuk menginisialisasi penumpang dan kapasitas.
* **batalkan\_tiket(self, jumlah)**: Metode ini digunakan untuk membatalkan tiket. Jika jumlah tiket yang ingin dibatalkan tidak melebihi jumlah penumpang saat ini, maka jumlah penumpang akan dikurangi. Jika tidak, pesan kesalahan akan ditampilkan.

**Kelas TiketPesawat, TiketKereta, TiketBus, dan TiketFerry**

* Setiap kelas ini mewarisi dari kelas Tiket. Mereka memiliki konstruktor yang memanggil konstruktor kelas dasar untuk inisialisasi.
* **tambah\_penumpang(self, jumlah)**: Metode ini diubah untuk menampilkan pesan log sebelum memanggil metode tambah\_penumpang dari kelas dasar. Ini memberikan konteks tambahan tentang jenis transportasi saat menambah penumpang.

**Fungsi main**

* Fungsi ini berfungsi sebagai contoh penggunaan. Di dalamnya, beberapa objek tiket dibuat, termasuk TiketKapalDenganPembatalan, TiketPesawat, TiketKereta, TiketBus, dan TiketFerry, dengan nilai awal penumpang dan kapasitas.
* Kemudian, menggunakan polymorphism, metode tambah\_penumpang dipanggil untuk masing-masing tiket dalam loop, yang menambahkan 10 penumpang ke setiap tiket dan menampilkan informasi penumpang saat ini.
* Setelah itu, beberapa operasi dilakukan pada tiket kapal (menurunkan dan membatalkan tiket).

**Menjalankan Program**

* main() dipanggil untuk mengeksekusi seluruh logika program.

**Ringkasan**

Program ini mendemonstrasikan konsep dasar OOP seperti pewarisan (inheritance) dan polymorphism. Kode ini cukup fleksibel untuk mengelola tiket berbagai jenis transportasi, memungkinkan penambahan dan pengurangan penumpang dengan memanfaatkan metode yang berbeda untuk setiap jenis transportasi.