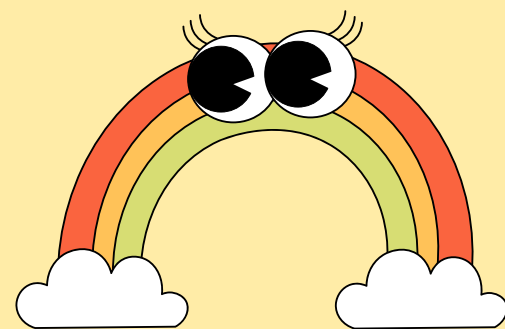


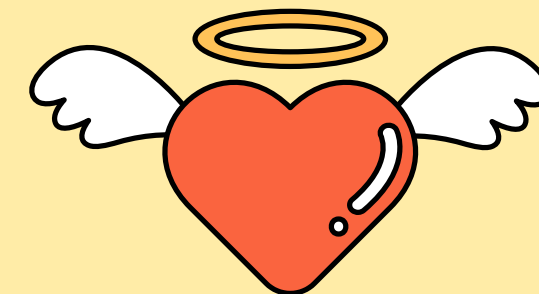
KSM ANDROID 2022/2023

 **E-WALLET** 

KOTLIN  **STUDY
CLUB**



**FINAL
PROJECT**



01. DESKRIPSI UMUM

E-Wallet merupakan sebuah program yang digunakan untuk melakukan transaksi dan pengelolaan transaksi pelanggan. Terdapat 2 fitur utama yaitu transfer antar e-wallet dan top up tapcash, serta user dapat melihat history transaksi yang dilakukan.

02. Kelas pada Program

User Class

1

- Kelas dasar yang merepresentasikan seorang pengguna dengan atribut name dan phoneNumber.
- Memiliki fungsi getUserInfo() untuk mengembalikan informasi pengguna dalam format string.

Customer Class

2

- Kelas yang mewarisi User
- Memiliki riwayat transaksi (transactionHistory) yang disimpan dalam bentuk list.

2

- Fungsi-fungsi dalam kelas ini meliputi:
1. getBalance(): Mengembalikan saldo pelanggan.
 2. addTransaction(): Menambahkan deskripsi transaksi ke riwayat.
 3. showTransactionHistory(): Menampilkan riwayat transaksi.

TransactionType Enum Class

3

- topUpTapCash(): Memungkinkan pelanggan untuk melakukan top up. Jika jumlah top up melebihi saldo, akan menampilkan pesan kesalahan.
- transfer(): Memungkinkan pelanggan untuk mentransfer sejumlah uang. Jika jumlah transfer melebihi saldo, akan menampilkan pesan kesalahan.
- runMainMenu: Fungsi untuk menjalankan menu utama

Kelas enum yang mendefinisikan nilai jenis transaksi pada program e-wallet yang tersedia, yaitu TOP_UP, TRANSFER, VIEW_HISTORY.



02. Kelas pada Program

Transaction Class

4

Kelas ini mendeklarasikan data class bernama Transaction yang menggunakan dua properti yaitu type dan description yang berguna untuk menyimpan riwayat transaksi.

Main Class

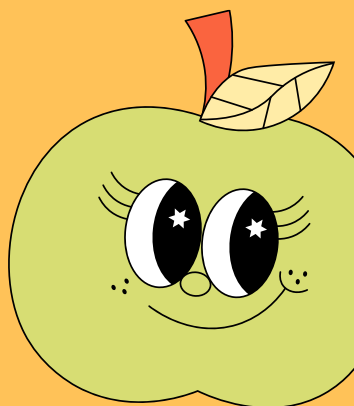
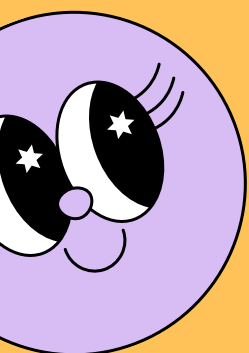
5

- Fungsi utama yang dijalankan saat program dimulai.
- Meminta pengguna untuk memasukkan nama dan nomor telepon mereka.
- Membuat objek Customer dengan saldo awal IDR 100,000.

Main Class

5

- Menampilkan menu utama yang memungkinkan pengguna untuk memilih antara top up, transfer, melihat riwayat transaksi, atau keluar dari program.
- Menu ini akan terus ditampilkan sampai pengguna memilih untuk keluar.



03. Cara Kerja Program ⚡⚡

1.

Insialisasi :

- Saat program dimulai, pengguna diminta untuk memasukkan nama dan nomor telepon mereka.
- Sebuah objek Customer dibuat dengan informasi yang diberikan oleh pengguna dan saldo awal IDR 100,000.

2.

Menu Utama :

- Setelah inisialisasi, menu utama ditampilkan kepada pengguna.
- Menu ini menawarkan empat pilihan: Top Up TapCash, Transfer, Lihat Riwayat Transaksi, dan Keluar.
- Pengguna memilih opsi dengan memasukkan angka yang sesuai.

```
6
7     val user = Customer(name, phoneNumber, initialBalance: 100000.0)
8     // Memulai loop menu utama
9     user.runMainMenu()
10 }
```

Run MainKt x MainKt x MainKt x

"C:\Users\MSI Cira\.jdk\openjdk-20.0.2\bin\java.exe" "-javaagent:C:\..."

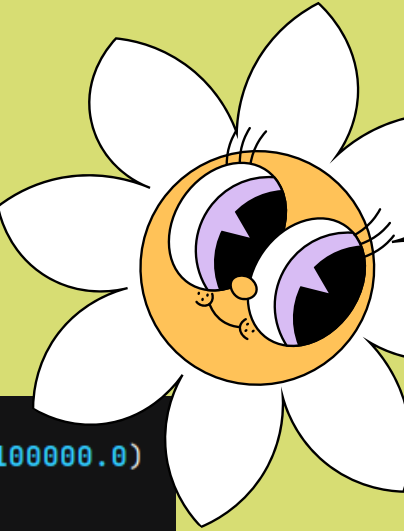
Enter your name: Cira

Enter your phone number: 951

1.

```
Menu:
Name: Cira, Phone Number: 951
Balance: IDR 100000.0
1. Top Up TapCash
2. Transfer
3. View Transaction History
4. Exit
Choose an option: 1
```

2.



03. Cara Kerja Program ⚡⚡

3.

Operasi :

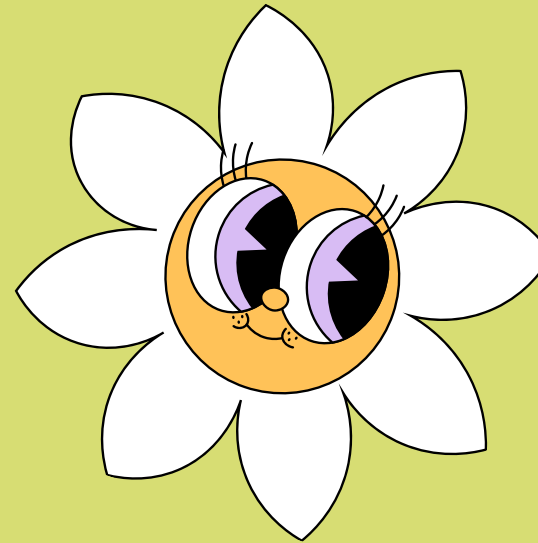
- Top Up TapCash:
 - Pengguna diminta untuk memasukkan nomor TapCash dan jumlah yang ingin ditambahkan.
 - Program memeriksa apakah saldo cukup untuk top up. Jika tidak, pesan kesalahan ditampilkan.
 - Jika saldo cukup, jumlah top up dikurangkan dari saldo dan sebuah nota ditampilkan.
 - Transaksi ini ditambahkan ke riwayat transaksi.
- Transfer :
 - Pengguna diminta untuk memasukkan nomor telepon tujuan dan jumlah yang ingin ditransfer.
 - Program memeriksa apakah saldo cukup untuk transfer. Jika tidak, pesan kesalahan ditampilkan.
 - Jika saldo cukup, jumlah transfer dikurangkan dari saldo dan sebuah nota ditampilkan.
 - Transaksi ini ditambahkan ke riwayat transaksi.



```
Choose an option: 1
Enter TapCash number: 0987654321
Enter amount to top up: 10000
-----
Top Up Successful!
Receipt:
Name: Cira, Phone Number: 951
TapCash Number: 0987654321
Top Up Amount: IDR 10000.0
Remaining Balance: IDR 90000.0
-----
```

TapCash

3.



```
Choose an option: 1
Enter TapCash number: 0987654321
Enter amount to top up: 230000
Insufficient funds for top up.
```

```
Choose an option: 2
Enter destination phone number: 1234567890
Enter amount to transfer: 100000
Insufficient funds for transfer.
```

Transfer

```
Choose an option: 2
Enter destination phone number: 1234567890
Enter amount to transfer: 9000
-----
Transfer Successful!
Receipt:
Name: Cira, Phone Number: 951
Destination Number: 1234567890
Transfer Amount: IDR 9000.0
Remaining Balance: IDR 81000.0
-----
```

04. Logika Program

Kelas User

Kelas user adalah kelas dasar yang menyimpan informasi dasar tentang pengguna, yaitu nama dan nomor telepon. Kelas ini juga memiliki fungsi `getUserInfo()` yang mengembalikan informasi pengguna dalam format String.

Kelas Customer

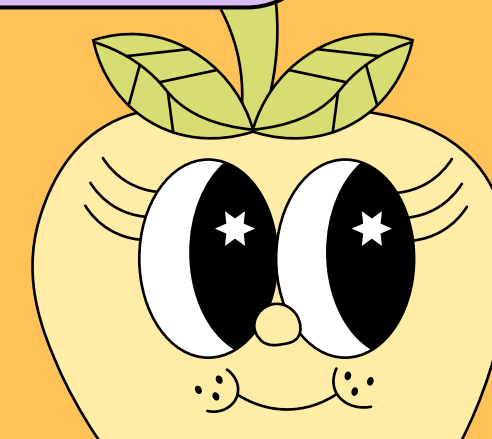
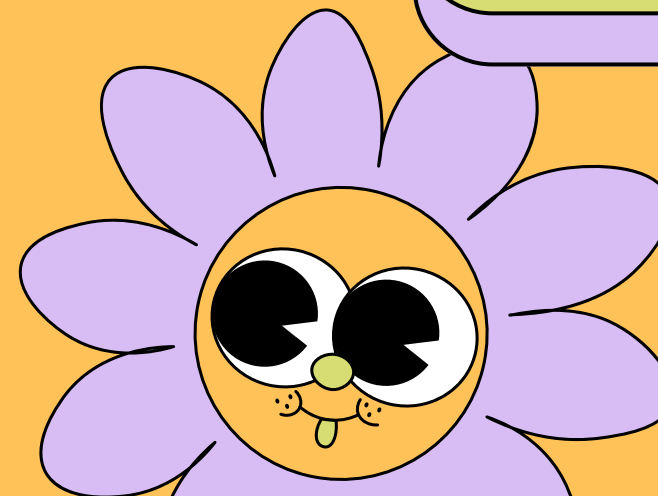
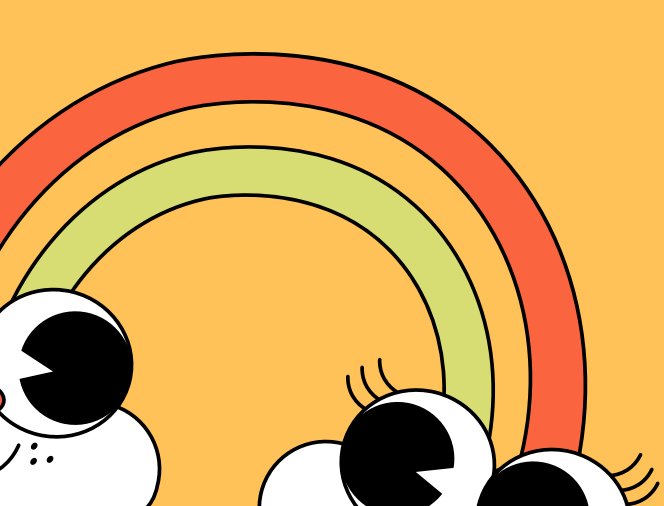
Kelas customer mewarisi kelas User, menyimpan saldo (balance) dan riwayat transaksi (transactionHistory). Logika utama pada kelas ini untuk melakukan operasi top up, transfer, dan menampilkan riwayat transaksi. Setiap kali operasi selesai, deskripsi transaksi ditambahkan ke file riwayat transaksi. Juga pada kelas ini menu interaktif dibuat agar pengguna dapat memasukkan data pengguna, setelah itu program membaca dan memanggil fungsi yang sesuai dari hasil inputan pengguna.

Main Class

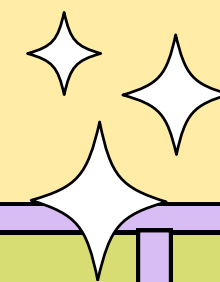
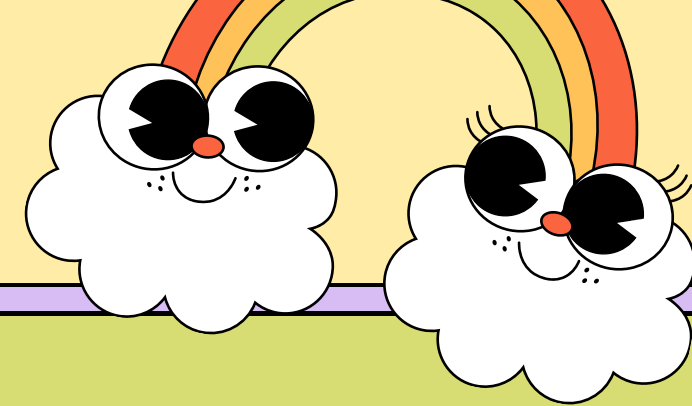
Kelas main adalah inti dari sebuah program, kelas ini mengatur interaksi dengan pengguna yang meminta data nama & nomor telepon serta menampilkan menu utama. Kelas ini menggunakan perulangan do-while untuk terus menampilkan menu sampai pengguna memilih untuk keluar.

Pemeriksaan Saldo

Pemeriksaan saldo dilakukan sebelum kegiatan top up atau transfer, program akan memeriksa apakah saldo cukup untuk melakukan transaksi, hal ini mencegah saldo menjadi negatif.



05. Materi



1

Variabel & Tipe Data:

- Penggunaan variabel seperti name, phoneNumber, dan balance.
- Tipe data dasar seperti String, Double, dan Int

2

Kelas & Objek:

- Terdapat 5 kelas dalam program e-wallet, di antaranya adalah kelas user, customer, transaction, transactionType dan kelas main.
- Pembuatan objek dari kelas customer yang digunakan untuk memanggil fungsi yang ada dalam kelas ini seperti: `val user = Customer(name, phoneNumber, 100000.0)`

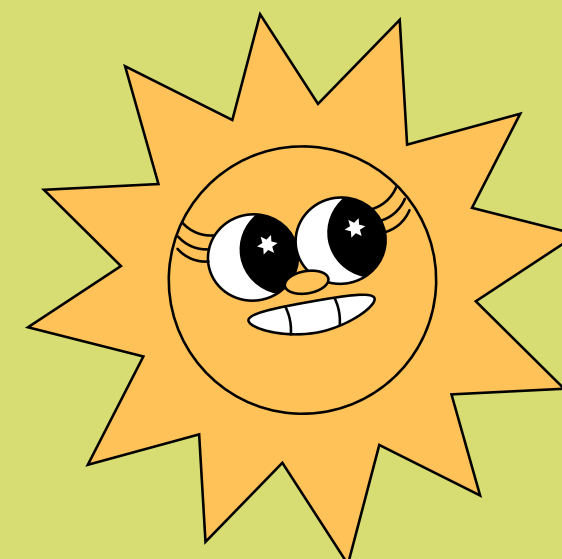
3

Konversi Tipe Data :

Salah satu penggunaan materi ini pada program sederhana e-wallet adalah mengkonversi nilai enum ke string. Ketika menampilkan nilai enum TransactionType dalam metode `addTransaction()`, nilai enum dikonversi ke string dengan menggunakan metode `toString()`.

```
phoneNumber: String, initialBalance: Double)
```

```
val user = Customer(name,  
    phoneNumber, initialBalance: 100000.0)
```



05. Materi



4

Pewarisan (Inheritance):

Kelas customer mewarisi User dengan mendeklarasikan kelas customer dengan memiliki 3 parameter yaitu nama, no telepon dan saldo awal yang ditunjukkan dengan sintaks : class Customer(name: String, phoneNumber: String, initialBalance: Double) : User(name, phoneNumber) { ... }

```
class Customer(name: String, phoneNumber: String,  
               initialBalance: Double) :  
    User(name, phoneNumber) {
```

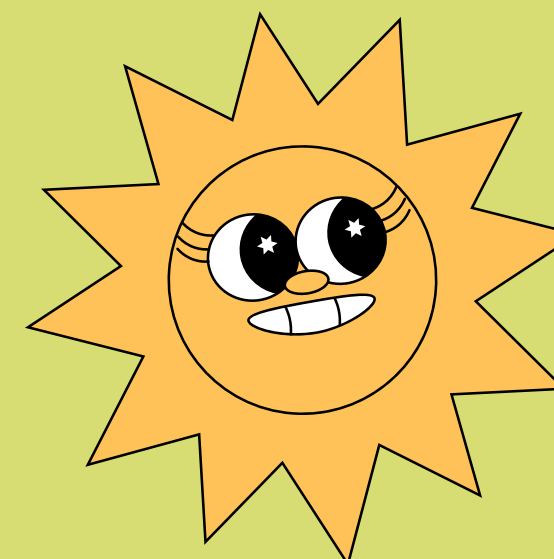
5

Enkapsulasi:

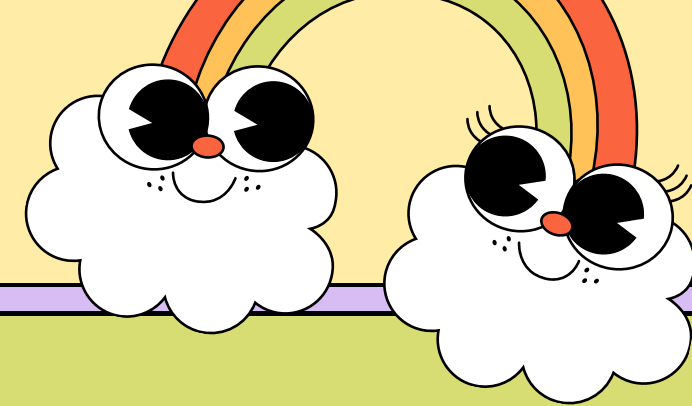
Properti balance & transactionHistory dan fungsi addTransaction diatur sebagai private untuk membatasi akses langsung dari luar kelas.

```
private var balance: Double = ini  
private val transactionHistory =
```

Sedangkan fungsi getBalance(), showTransactionHistory(), topUpTapCash(), transfer(), dan runMainMenu() di kelas Customer diatur sebagai public agar dapat digunakan oleh kelas lain.



05. Materi



6

Fungsi & Metode:

- Definisi fungsi seperti getUserInfo() di kelas User.
- Definisi metode dalam kelas Customer seperti topUpTapCash(), transfer(), dan lainnya.

7

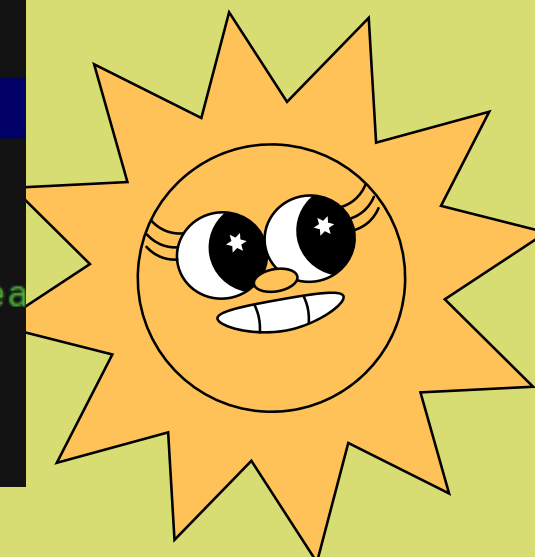
Control Flow

1. **Percabangan:** Menggunakan if dalam fungsi topUpTapCash() dan transfer() yang memeriksa kondisi tertentu, seperti saldo yang cukup untuk melakukan transaksi.
2. **Pemilihan:** Menggunakan when dalam fungsi runMainMenu yang akan memeriksa nilai dari choice yang dipilih oleh pengguna.
3. **Perulangan:** Menggunakan do-while untuk menampilkan menu berulang kali sampai pengguna memilih untuk keluar.

```
if (amount > balance) { // jika saldo tidak me
    println("Insufficient funds for top up.")
    return
}
```

```
do {
    println("\nMenu:")
    println(getUserInfo())
    println("Balance: IDR ${getBalance()}")
    println("1. Top Up TapCash")
    println("2. Transfer")
    println("3. View Transaction History")
    println("4. Exit")
    print("Choose an option: ")
    choice = readLine()?.toIntOrNull() ?: 4

    when (choice) {
        1 -> topUpTapCash()
        2 -> transfer()
        3 -> showTransactionHistory()
        4 -> exitProcess(status: 0)
        else -> println("Invalid option, please try again")
    }
} while (choice != 4)
```



05. Materi



8

String Templates & Multiline Strings:

Pada program ini digunakan `$variableName` untuk menyisipkan nilai variabel ke dalam string dan menggunakan string multiline (dengan `"""`) untuk format output yang lebih rapi.

9

Input/Output:

- Menggunakan `print()` dan `println()` untuk menampilkan informasi ke pengguna.
- Menggunakan `readLine()` untuk menerima input dari pengguna.

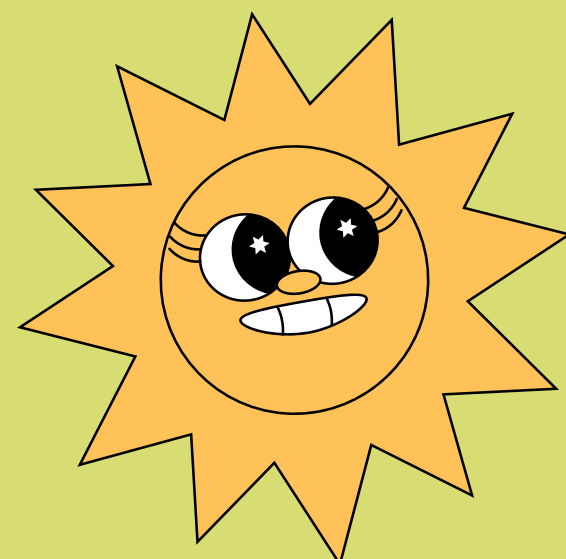
10

Nullable types & Accessing Nullable Types

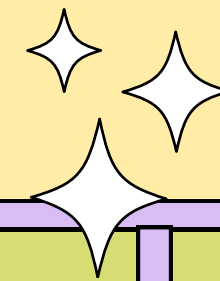
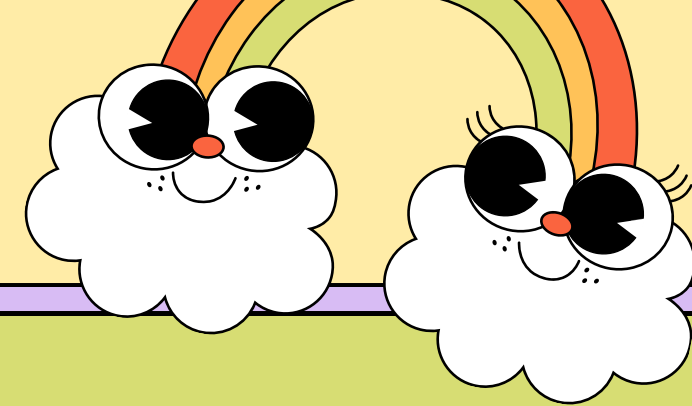
Beberapa variabel seperti `name` dan `phoneNumber` didefinisikan sebagai nullable (`String?`), yang berarti mereka dapat memiliki nilai `null`. Juga terdapat penggunaan non-null assertion (`!!`) untuk mengakses nilai dari variabel nullable.

```
val receipt = """
-----
Top Up Successfu
Receipt:
${getUserInfo()}
TapCash Number:
Top Up Amount: 1
Remaining Balanc
-----
""".trimIndent()
```

```
val tapCashNumber = readLine() ?!
```



05. Materi



11

Data Class

- Menggunakan dua properti yaitu type dan description.
- Properti type adalah jenis tipe data enum dari kelas TransactionType yang mewakili jenis nilai transaksi seperti TOP_UP, TRANSFER, dan VIEW_HISTORY.
- Properti description adalah jenis tipe data string yang mewakili deskripsi transaksi untuk menyimpan riwayat transaksi.

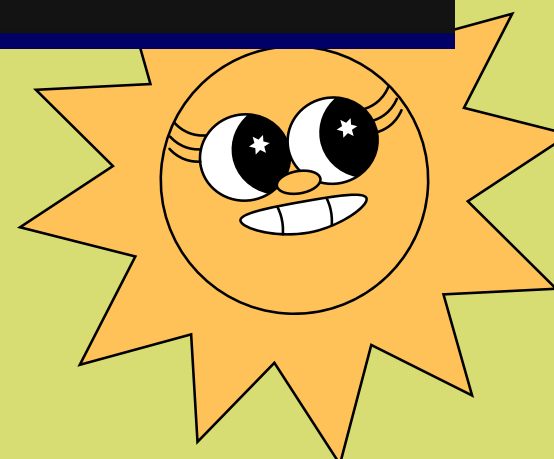
```
// Data class untuk merepresentasikan transaksi.  
data class Transaction(val type:  
                        TransactionType,  
                        val description: String)
```

12

Exception Handling

Pada metode topUpTapCash() dan transfer(), terdapat blok try-catch yang digunakan untuk menangani kesalahan inputan dari pengguna. Blok try berisi kode yang berpotensi menimbulkan kesalahan, sedangkan blok catch berisi kode yang akan dieksekusi jika terjadi kesalahan.

```
fun topUpTapCash() {  
    try {  
        print("Enter TapCash number: ") // memasukkan nomor kartu tapcash  
        val tapCashNumber = readLine() ?: throw Exception("TapCash number required.")  
        print("Enter amount to top up: ") // memasukkan jumlah yang ingin di isi  
        val amount = readLine()?.toDoubleOrNull() ?: throw Exception("Invalid amount.")  
  
        if (amount > balance) { // jika saldo tidak mencukupi maka transaksi gagal  
            println("Insufficient funds for top up.")  
            return  
        }  
    }  
}
```



05. Materi



13

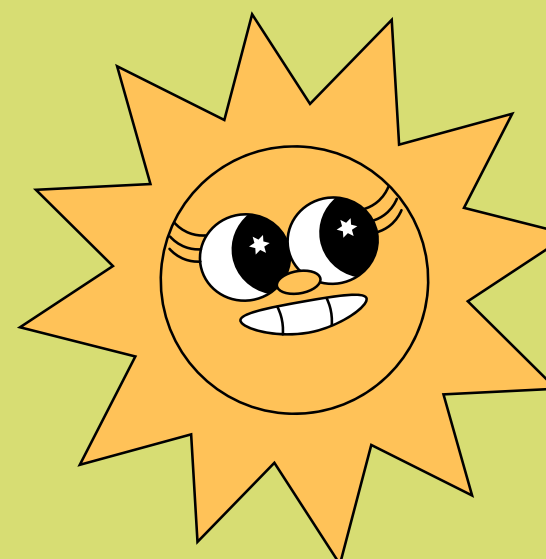
Polymorphism

Polymorphism pada program e-wallet diterapkan pada metode `getUserInfo()`. Contoh pengimplementasian dari metode `getUserInfo()` :

- Metode `getUserInfo()` didefinisikan pada kelas `User` dan diwarisi oleh kelas `Customer` yang mengembalikan informasi pengguna berupa nama dan nomor telepon.
- Pada kelas `User`, metode `getUserInfo()` mengembalikan informasi pengguna secara sederhana, namun, pada kelas `Customer`, metode `getUserInfo()` dapat mengembalikan informasi pengguna yang lebih lengkap, misalnya termasuk saldo saat ini.

```
fun getUserInfo(): String =
```

```
val receipt = ""  
-----  
Top Up Successful!  
Receipt:  
${getUserInfo()}  
TapCash Number: $tapCashNumber  
Top Up Amount: IDR $amount  
Remaining Balance: IDR $balance  
-----  
"".trimIndent()
```



05. Materi



14

Collection

Pada program ini menggunakan `mutableListOf` untuk menyimpan riwayat transaksi.

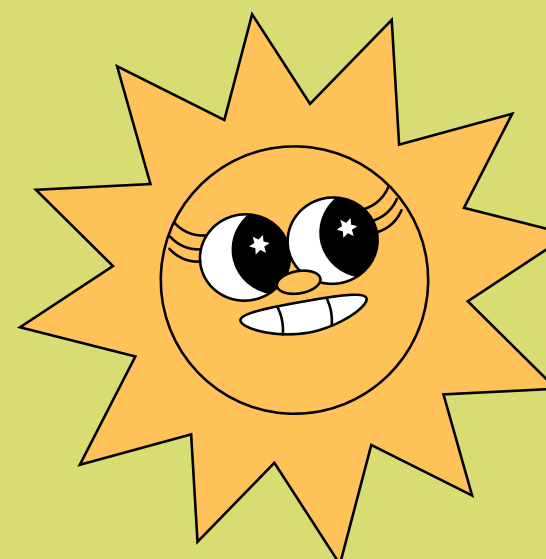
15

Enumeration

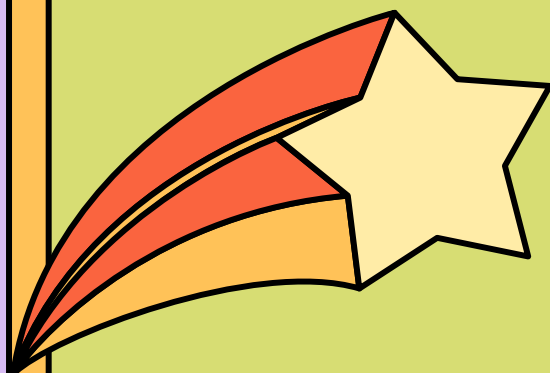
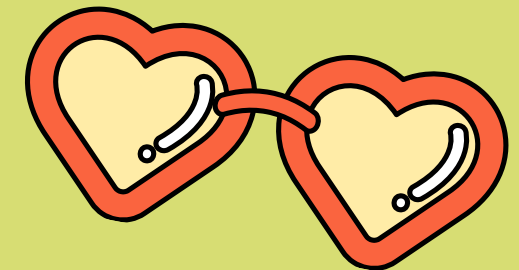
Enumeration `TransactionType` digunakan untuk menyimpan jenis transaksi untuk setiap transaksi dalam riwayat transaksi pengguna.

```
private val transactionHistory =  
    mutableListOf<Transaction>()
```

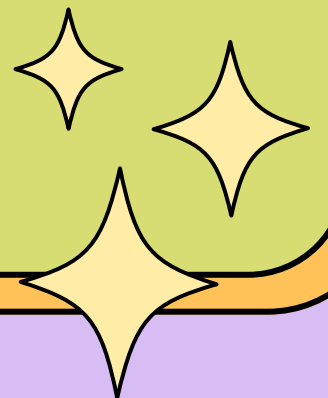
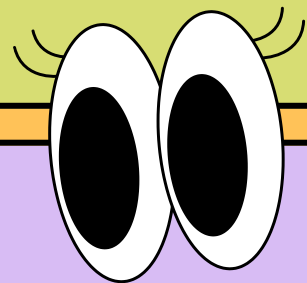
```
// Enum yang mendefinisikan jenis transaksi yang tersedia.  
enum class TransactionType {  
    TOP_UP, TRANSFER, VIEW_HISTORY  
}
```

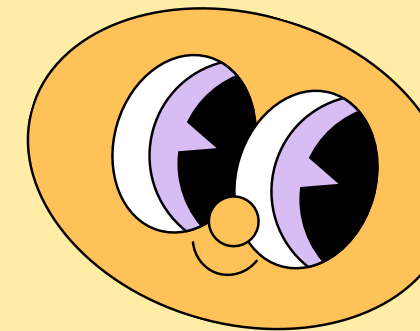
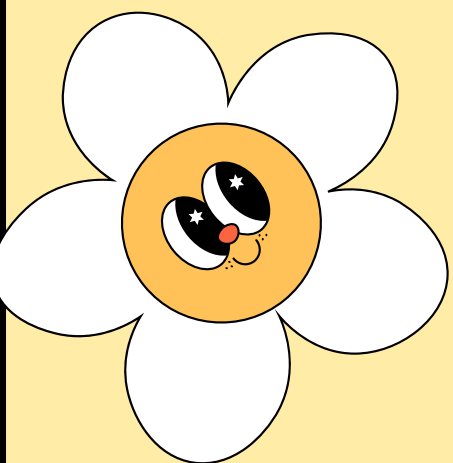


06. Kendala



Kendala saat mengerjakan project adalah sempat terjadi error dalam membuat file riwayat transaksi dan saat di awal pengerjaan untuk final project, lumayan bingung untuk menentukan apa yang ingin dikembangkan dalam program.





**TERIMA
KASIH**

