

UNGUIDED 07

INHERITANCE DAN OVERRIDING

Waktu Pengerjaan: **2 jam 30 menit**

Petunjuk Pengumpulan Jawaban

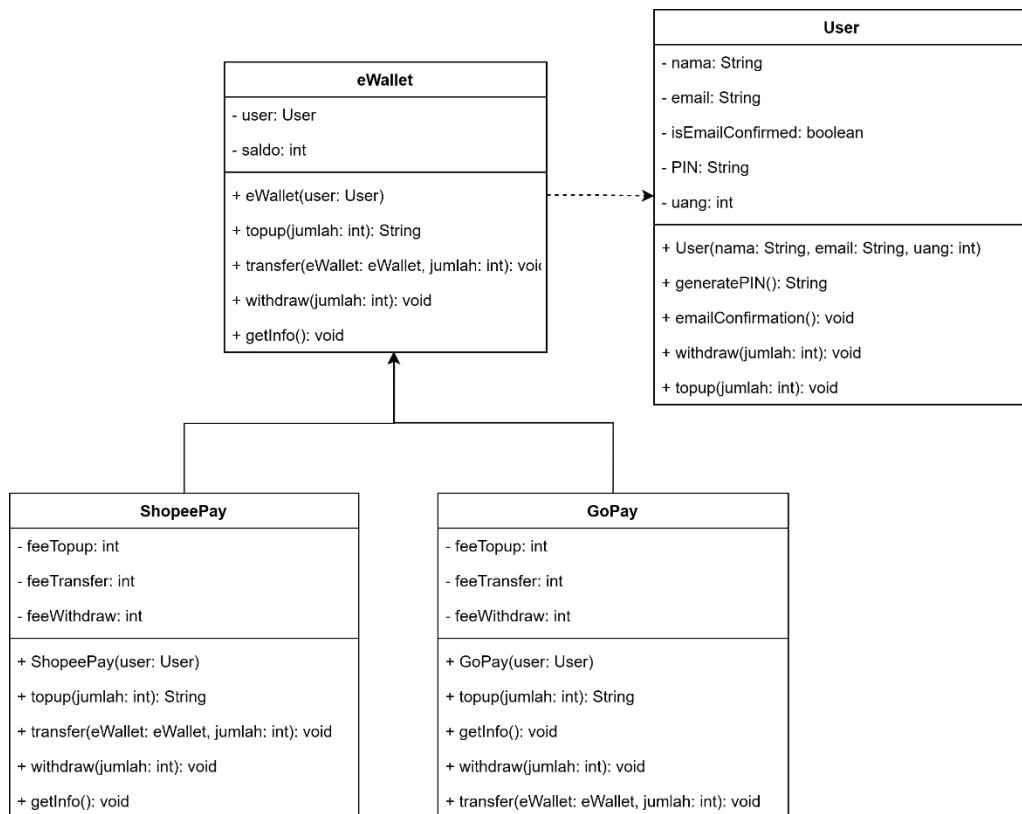
Buatlah folder bernama **UG7A_NIM** sebagai tempat untuk menyimpan jawaban kamu. Selanjutnya, buatlah project sesuai dengan ketentuan soal. Simpan project tersebut ke dalam folder yang baru saja kamu buat. Upload/push folder **UG7A_NIM** berisi project jawaban kamu ke repository UG_RPLBO_NIM_GRUP yang sudah ada di Github kamu.

Soal (100 Poin)

Buatlah sebuah project Maven dengan ketentuan:

- **GroupId:** com.ug7.ewallet
- **Nama Project/ArtifactId:** **NIM**-UG7-EWALLET --> Pastikan ini menjadi folder yang berisi jawaban project kamu! Hint: kamu dapat membuat folder **NIM**-UG7-EWALLET terlebih dahulu, sebelum membuat project.
- **Archetype:** maven-archetype-quickstart

Seiring dengan perkembangan teknologi yang semakin pesat, masyarakat dituntut untuk mengikuti tren-tren yang ada saat ini. Salah satunya adalah melalui teknologi *e-wallet*. Apa itu *e-wallet*? Secara garis besar, *e-wallet* adalah dompet digital yang berfungsi sebagai alat transaksi terintegrasi. Sebagai mahasiswa informatika yang “AMPUH”, kamu diminta untuk membuat program *e-wallet* sederhana dengan menerapkan *Class Diagram* seperti berikut:



Petunjuk Pengerjaan Soal

Class User merupakan **dependency**, dimana *Class User* merupakan tipe dari salah satu atribut yang ada di *Class eWallet*. Selain itu, *Class ShopeePay* dan *Class GoPay* merupakan anak dari *Class eWallet*. Berikut ini adalah **ketentuan untuk Class ShopeePay dan Class GoPay:**

x	ShopeePay	GoPay
feeTopup	= 500	= 1000
feeTransfer	= 0	= 500
feeWithdraw	= 1000	= 2500
topup(jumlah)	Bebas	Jumlah minimal Rp10.000,-
transfer(eWallet, jumlah)	1. Harus konfirmasi email dulu (manfaatkan atribut isEmailConfirmed yang ada di Class User) 2. Tidak perlu PIN 3. Proses transfer ada delay 3 detik	1. Tidak perlu konfirmasi email 2. Butuh PIN, validasi delay 3 detik (PIN diperoleh dari Class User, tinggal pakai) 3. Proses transfer ada delay 3 detik
getInfo()	[ShopeePay e-Wallet] Nama Email Uang cash	[GoPay e-Wallet] Nama Email Uang cash

	Saldo	Saldo
--	-------	-------

Keterangan Atribut

Nama Atribut	Kegunaan
isEmailConfirmed	Atribut ini merupakan status dari email apakah sudah terkonfirmasi (bernilai true) atau belum (bernilai false). Awalnya nilainya tidak terdefinisi yang berarti bernilai false.
PIN	PIN merupakan password yang digenerate dengan method generatePIN() yang sudah tersedia di soal (Class User). Gunakan PIN ini untuk transaksi transfer pada e-Wallet GoPay.
feeTopup	Atribut ini berguna untuk menyimpan jumlah fee yang ditanggung ketika melakukan top up, sehingga uang cash yang diperlukan untuk top up = jumlah top up + feeTopup
feeTransfer	Atribut ini berguna untuk menyimpan jumlah fee yang ditanggung ketika melakukan transfer, sehingga saldo yang diperlukan untuk transfer = jumlah transfer + feeTransfer
feeWithdraw	Atribut ini berguna untuk menyimpan jumlah fee yang ditanggung ketika melakukan withdraw, sehingga saldo yang diperlukan untuk withdraw = jumlah withdraw + feeWithdraw

Keterangan Method

Nama Method	Kegunaan
emailConfirmation()	<i>Method</i> ini berguna untuk mengonfirmasi email user. Singkatnya, method ini mengubah nilai dari atribut isEmailConfirmed menjadi true.
topup(jumlah)	<p><i>Method</i> ini berguna untuk melakukan topup saldo e-wallet, uang cash user akan diubah menjadi saldo e-wallet dengan tanggungan fee tertentu</p> <p>Ketika user melakukan top up, uang cashnya akan berkurang sebanyak jumlah topup (parameter jumlah) + feeTopup dan saldonya akan bertambah sebanyak jumlah topup</p> <p>Misal: feeTopup = 1.000; uang cash = 50.000; saldo = 0 Topup sebanyak 30.000 Maka, kondisi sekarang menjadi: uang cash = 29.000; saldo = 30.000</p> <p>Perhatikan juga ketentuan untuk Class ShopeePay dan Class GoPay di atas</p>
transfer(eWallet, jumlah)	<p><i>Method</i> ini berguna untuk melakukan transfer, saldo e-wallet akan dikirimkan ke akun e-Wallet user yang lain (parameter eWallet) dengan tanggungan fee tertentu</p> <p>Ketika user melakukan transfer, saldonya sendiri akan berkurang sebanyak jumlah transfer (parameter jumlah) + feeTransfer dan</p>

	<p>saldo penerimanya akan bertambah sebanyak jumlah transfer (parameter jumlah).</p> <p>Misal: feeTransfer = 2.000; saldo si A = 20.000; saldo si B = 5.000 Si A transfer sebanyak 10.000 ke si B Maka, kondisi sekarang menjadi: saldo si A = 8.000; saldo si B = 15.000</p> <p>Perhatikan juga ketentuan untuk Class ShopeePay dan Class GoPay di atas</p>
withdraw(jumlah)	<p><i>Method</i> ini kebalikan dari <i>method</i> topup, <i>method</i> ini berguna untuk melakukan penarikan, saldo e-wallet user sebanyak parameter jumlah akan diubah menjadi uang cash dengan tanggungan fee tertentu</p> <p>Ketika user melakukan <i>withdraw</i>, saldonya akan berkurang sebanyak jumlah <i>withdraw</i> (parameter jumlah) + feeWithdraw dan uang cashnya akan bertambah sebanyak jumlah <i>withdraw</i> (parameter jumlah).</p> <p>Misal: feeWithdraw = 2.500; uang cash = 0; saldo = 50.000 Withdraw sebanyak 30.000 Maka, kondisi sekarang menjadi: uang cash = 30.000; saldo = 17.500</p>
getInfo()	<p><i>Method</i> ini akan menampilkan informasi akun user (nama, email, status email (jika belum terkonfirmasi tidak usah ditampilkan), PIN, uang cash) dan saldo e-walletnya. Pada line pertama akan ditampilkan jenis e-Wallet yang digunakan.</p> <p>Output yang diharapkan dari pemanggilan <i>method</i> ini:</p> <p>[ShopeePay e-Wallet] Nama: Bantolo Setiadi [PIN: BanBan480] Email: bantolo.setiadi@ti.ukdw.ac.id [Confirmed] Uang cash: Rp50.000,00 Saldo e-wallet: Rp0,00</p>

Konsep program *e-Wallet* ini **dibuat semirip mungkin dengan *e-Wallet* yang ada pada dunia nyata**. Atribut uang yang ada pada *Class User* adalah representasi dari uang *cash* yang dimiliki oleh *user*, nantinya uang tersebut akan diubah menjadi saldo *e-Wallet* dengan cara *topup* yang akan dikenakan *fee* sesuai dengan kebijakan masing-masing *e-Wallet*. Setelah uang *cash* berhasil diubah menjadi saldo, saldo tersebut dapat ditransferkan baik ke sesama jenis *e-Wallet* ataupun yang berbeda yang akan dikenakan *fee* sesuai dengan kebijakan masing-masing *e-Wallet*. Selain itu, saldo juga dapat ditarik kembali menjadi uang cash dengan melakukan *withdraw* yang akan dikenakan *fee* sesuai dengan kebijakan masing-masing *e-Wallet*.

Class User sudah disediakan dalam arsip soal, jadi yang harus kamu lakukan adalah **membuat 3 Class yang lain (silakan cek Class Diagram)** yang menerapkan konsep *inheritance* dan *overriding*!

Kamu dibebaskan untuk menambahkan *method* lain, **HANYA METHOD!** Kamu tidak diperbolehkan membuat *class* lain atau atribut tambahan. Hal ini bertujuan untuk mempermudah kamu dalam mengerjakan unguided ini. Silakan buat program sekreatif kamu dengan algorithamu sendiri!

Dalam arsip soal juga sudah disediakan *Class Main* untuk dapat menguji program yang kamu buat. **Jangan lupa ganti nama, email, dan NIM ya!** Di dalam *Class Main* juga terdapat *method static* untuk menyajikan angka dalam format rupiah dan untuk delay! Selain itu, kamu juga dapat mencocokkannya dengan *file JAR* yang juga tersedia di dalam arsip soal!

Petunjuk Penilaian

Setiap bagian di bawah ini memiliki nilai 10 poin, sehingga jika kamu dapat mengerjakan keseluruhan program yang mengeluarkan output sama seperti contoh di file JAR (**dengan logika yang benar tentunya**), kamu akan mendapatkan 100 poin!

1. ===== AWAL =====
2. ===== TOP UP #1 =====
3. ===== TOP UP #2 =====
4. ===== TOP UP #3 =====
5. ===== TRANSFER #1 =====
6. ===== KONFIRMASI EMAIL =====
7. ===== TRANSFER #2 =====
8. ===== TRANSER #3 =====
9. ===== WITHDRAW #1 =====
10. ===== AKHIR =====

Selamat mengerjakan! Soal di atas mudah untuk dikerjakan, apalagi jika tidak dikerjakan

