

IF2150 REKAYASA PERANGKAT LUNAK
TUGAS 4
CLASS DIAGRAM

Tabung: *Tracker Budget dan Keuangan*

untuk:

Kefas Kurnia Jonathan

Dipersiapkan oleh:

11 - K01 - Dakhet

13524001 - Samuelson Dharmawan Tanuraharja

13524027 - Aufa Rienaldifaza Ahmad

13524037 - Nicholas Wise Saragih Sumbayak

13524044 - Narendra Dharma Wistara Marpaung

13524056 - Reinhard Alfonzo Hutabarat

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132

2025

Daftar Isi

Daftar Isi	2
Daftar Perubahan	3
Daftar Tabel	4
Daftar Gambar	5
1 Deskripsi Perangkat Lunak	6
2 Kebutuhan Fungsional	6
2.1 Kebutuhan Fungsional	6
3 Model Use Case	6
3.1 Identifikasi Aktor	6
3.2 Identifikasi Use Case	6
3.3 Use Case Diagram	7
3.4 Skenario Use Case	7
3.4.1 Skenario Use Case 01	7
3.4.2 Skenario Use Case XX	8
4 Diagram Kelas	8
4.1 Use Case <Nama Use Case>	8
4.1.1 Identifikasi Kelas Use Case <Nama Use Case>	8
4.1.2 Diagram Kelas Use Case <Nama Use Case>	8
4.2 Use Case <Nama Use Case>	9
4.2.1 Identifikasi Kelas Use Case <Nama Use Case>	9
4.2.2 Diagram Kelas Use Case <Nama Use Case>	9
4.3 Diagram Kelas Keseluruhan	10
5 Traceability	11

*Generate daftar isi dengan **cara otomatis**. Jika terdapat gambar dan tabel dalam dokumen, maka halaman ini **wajib** diikuti oleh halaman daftar tabel dan daftar gambar.*

Daftar Perubahan

Revisi	Deskripsi
A	<i>Deskripsikan perubahan yang dilakukan dari dokumen sebelumnya pada dokumen ini. Jika tidak terdapat perubahan, harap kosongkan tabel. Silakan tambahkan kolom baru jika dibutuhkan.</i>
B	
C	
D	
E	

Daftar Tabel

<i>Tabel X.X. <Nama Tabel></i>	<i>X</i>
<i>Tabel X.X. <Nama Tabel></i>	<i>X</i>
<i>Tabel X.X. <Nama Tabel></i>	<i>X</i>

Daftar Gambar

<i>Gambar 1. <Nama Gambar></i>	<i>X</i>
<i>Gambar 2. <Nama Gambar></i>	<i>X</i>
<i>Gambar 3. <Nama Gambar></i>	<i>X</i>

1 Deskripsi Perangkat Lunak

Perkembangan teknologi yang tak ada hentinya terus mendorong penerapannya dalam kehidupan sehari-hari manusia. Saat ini, transaksi tidak lagi bergantung pada uang tunai, melainkan semakin bergeser ke arah digital melalui layanan *online banking*, *mobile banking*, *e-wallet*, hingga pembayaran berbasis QRIS yang telah ditetapkan sebagai standar nasional. Transformasi ini memberikan kemudahan dalam melakukan pembayaran maupun pembelian secara online maupun offline, sekaligus mendorong inklusi keuangan dengan memperluas akses layanan bagi masyarakat.

Namun, di balik berbagai kemudahan tersebut, muncul tantangan baru yang tidak bisa diabaikan. Banyaknya pilihan metode pembayaran, seperti transfer bank, kartu kredit, dan e-wallet, ditambah dengan variasi jenis pembayaran seperti sekali bayar maupun berlangganan, justru menyulitkan manusia untuk mencatat pengeluaran pribadi secara menyeluruh. Akibatnya, sering terjadi kebocoran anggaran, pengeluaran yang tidak tercatat, serta lemahnya kesadaran finansial dalam mengatur pendapatan dan pengeluaran.

Untuk menjawab tantangan tersebut, perangkat lunak Tabung sebagai *expenses tracker* hadir sebagai solusi praktis yang mampu membantu individu dalam mengatur keuangannya secara lebih baik. Aplikasi ini tidak hanya berfungsi untuk mencatat transaksi, tetapi juga dapat mengelompokkan, menganalisis, serta menyajikan data keuangan dalam bentuk visual yang sederhana dan mudah dipahami. Dengan adanya fasilitas ini, pengguna dapat memiliki kontrol yang lebih baik terhadap keuangannya, mengurangi risiko pemborosan, serta mendorong pola konsumsi yang lebih bijak.

Pengguna mengharapkan sebuah sistem aplikasi manajemen keuangan yang mampu membantu dalam pencatatan, pengelolaan, dan pemantauan kondisi keuangan pribadi secara menyeluruh. Sistem ini diharapkan memiliki antarmuka yang sederhana, intuitif, dan mudah dipahami sehingga dapat digunakan oleh berbagai kalangan, termasuk pengguna awam. Sistem harus mampu menyediakan sarana untuk menambahkan transaksi (baik berupa pemasukan maupun pengeluaran) secara efisien, menyusun laporan keuangan otomatis, serta menyajikannya dalam bentuk tabel maupun visualisasi grafik.

Selain itu, sistem perlu mendukung pencatatan berbagai jenis pembayaran, baik sekali waktu maupun berlangganan, dengan fleksibilitas sesuai kebutuhan nyata pengguna. Keamanan data juga menjadi aspek penting yang harus dijaga, mengingat catatan keuangan bersifat sensitif dan pribadi. Secara keseluruhan, kebutuhan utama pengguna mencakup kemudahan penggunaan, kecepatan pencatatan transaksi, kejelasan laporan, fleksibilitas pengelolaan tipe pengeluaran, serta keamanan data yang andal. Dengan adanya perangkat lunak ini, pengguna diharapkan dapat meningkatkan disiplin finansial, mengurangi risiko kebocoran anggaran, dan mendukung pencapaian tujuan keuangan pribadi secara lebih terarah sekaligus berkontribusi pada pembangunan berkelanjutan.

2 Kebutuhan Fungsional

2.1 Kebutuhan Fungsional

Tabel 2.1. Tabel Kebutuhan Fungsional

ID KF	Kebutuhan	Penjelasan
KF01	Pendaftaran Akun	Pengguna harus dapat mendaftarkan kredensial akun (email, password) untuk membuat akun baru.
KF02	Akses Akun	Pengguna harus bisa mengakses data keuangan pribadi melalui <i>login</i> dan <i>logout</i> .
KF03	Pengaturan Kategori & Budget Rutin	Pengguna harus dapat menentukan kategori pengeluaran, alokasi budget, serta detail pengeluaran rutin.
KF04	Penentuan Batas Pengeluaran & Tabungan	Pengguna harus dapat menentukan batas pengeluaran, termasuk budget yang ingin ditabung.
KF05	Input & Rekapitulasi Transaksi	Sistem harus memungkinkan pengguna untuk memilih memasukkan data transaksi atau melihat data rekapitulasi.
KF06	Pencatatan Transaksi Pengeluaran	Sistem harus memungkinkan pengguna mencatat transaksi pengeluaran dengan nominal, kategori, deskripsi, dan waktu.
KF07	Pencatatan Transaksi Pemasukan	Sistem harus memungkinkan pengguna mencatat transaksi pemasukan dengan nominal, kategori, deskripsi, dan waktu.
KF08	Ringkasan Harian Keuangan	Sistem harus menyajikan ringkasan pengeluaran/pemasukan harian.
KF09	Dashboard Statistik & Visualisasi Keuangan	Sistem harus menampilkan dashboard statistik berupa visualisasi dalam bentuk grafik distribusi pengeluaran per kategori, pemasukan per kategori, prediksi anggaran, dan progress tabungan.
KF10	Notifikasi & Peringatan Anggaran	Sistem harus mengirimkan notifikasi harian serta peringatan kontekstual jika mendekati/melewati batas <i>budget</i> atau jatuh tempo.

3 Model Use Case

3.1 Identifikasi Aktor

Tabel 3.1.1. Deskripsi Aktor Perangkat Lunak

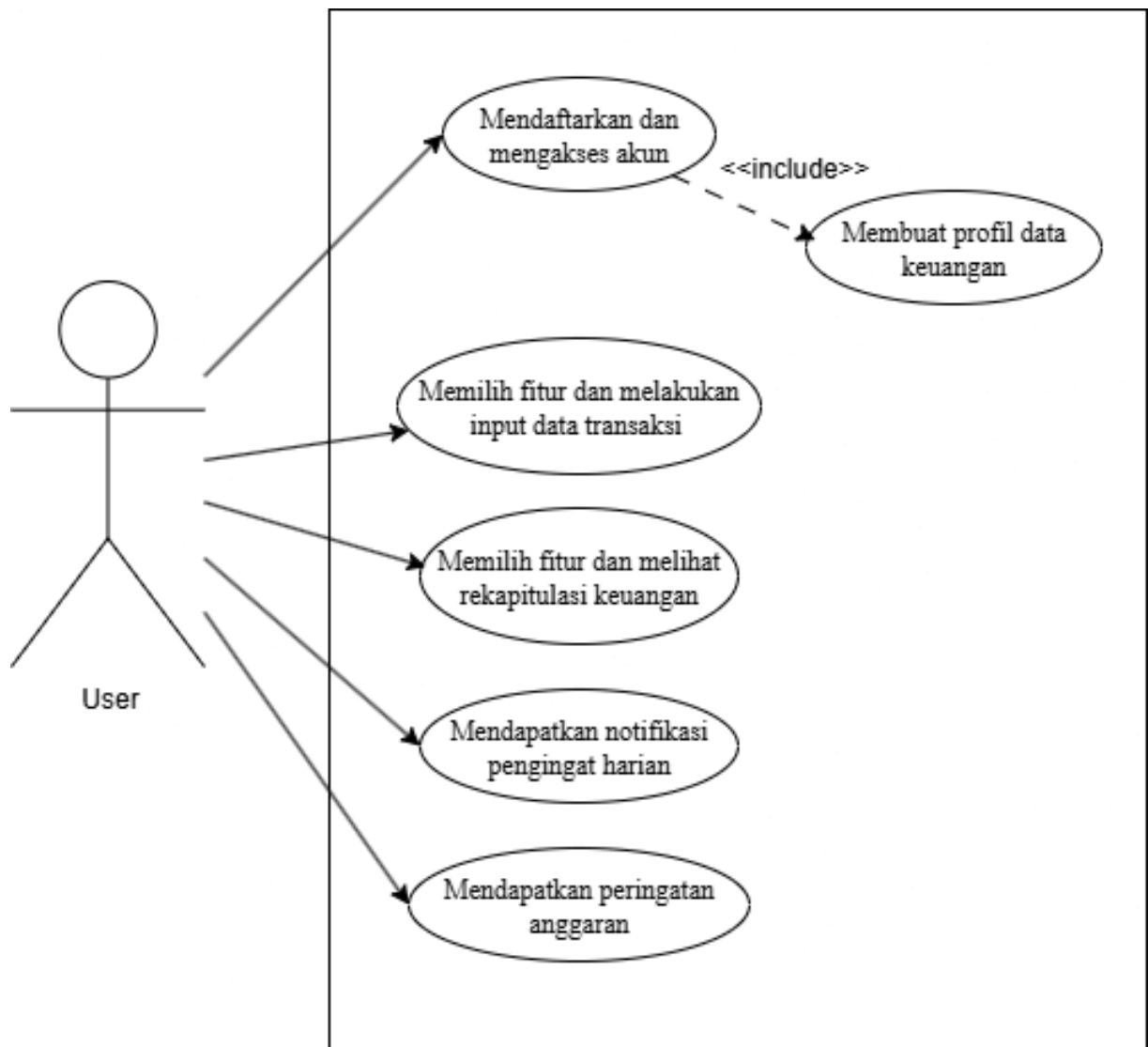
ID Aktor	Nama	Deskripsi
A01	User	User dapat membuat akun untuk membuat profil keuangan yang terdapat pemasukan, pengeluaran, dan <i>budget</i> tabungan, mencatat transaksi harian baik pemasukan dan pengeluaran, serta melihat statistik keuangan berdasarkan riwayat transaksi.

3.2 Identifikasi Use Case

Tabel 3.2.1. Identifikasi Use Case

ID UC	Use Case	Deskripsi	ID KF
UC01	Mendaftarkan dan mengakses akun	Pengguna dapat membuat akun baru dengan kredensial email dan <i>password</i> tertentu dan masuk ke akun yang sudah dibuat,	KF01, KF02
UC02	Membuat profil data keuangan	Pengguna memasukkan detail mengenai pemasukan, pengeluaran dan target yang ingin dicapai tiap bulan untuk tabungan.	KF03, KF04
UC03	Memilih fitur dan melakukan input data transaksi	Ketika aplikasi baru dijalankan, pengguna dapat memilih fitur input data transaksi dan melakukan input nominal pemasukkan/pengeluaran beserta dengan keterangannya tambahannya.	KF05, KF06, KF07
UC04	Memilih fitur dan melihat rekapitulasi keuangan	Pengguna mampu memilih fitur rekapitulasi dan melihat ringkasan transaksi harian beserta rekapitulasi keuangan dalam bentuk dashboard statistik dan visualisasi grafis	KF05, KF08, KF09
UC05	Mendapatkan notifikasi pengingat harian	Pengguna akan mendapatkan notifikasi pengingat harian pada waktu yang ditentukan untuk melakukan rekapitulasi transaksi harian apabila ada yang belum direkap.	KF10
UC06	Mendapatkan peringatan anggaran	Pengguna akan mendapatkan notifikasi peringatan apabila pengguna melakukan pengeluaran di atas <i>budget</i> atau ada tagihan yang akan jatuh tempo	KF10

3.3 Use Case Diagram



Gambar 3.3.1 Diagram Use Case

3.4 Skenario Use Case

3.4.1 Skenario Use Case 01

Nama Use Case: Mendaftarkan dan mengakses akun

Skenario:

Tabel 3.4.1. Skenario Use Case 01: Mendaftarkan dan Mengakses Akun

Aksi Aktor (User)	Reaksi P/L
<i>Skenario Normal</i>	
1. User membuka aplikasi untuk pertama kalinya.	
	2. Memunculkan UI untuk “Daftar Akun”/”Masuk Akun”.
3. User memilih “Daftar Akun”.	
	4. Meminta input <i>email</i> dan <i>password</i> .
4. User memasukkan data <i>email</i> dan <i>password</i> .	
	5. Menyimpan data User di <i>database</i> .
	6. Menunjukkan informasi bahwa pendaftaran akun telah berhasil.
	7. Memunculkan UI untuk “Daftar Akun”/”Masuk Akun”.
8. User memilih “Masuk Akun”.	
	9. Meminta input <i>email</i> dan <i>password</i> .
9. User memasukkan data <i>email</i> dan <i>password</i> .	
	10. Melakukan validasi <i>input user</i> dengan <i>database</i> .
	11. Menunjukkan proses Masuk Akun berhasil.
<i>Skenario Alternatif: User mengganti password</i>	
	10. Melakukan validasi <i>input user</i> dengan <i>database</i> .
	11. Hasil validasi menunjukkan <i>input user</i> salah, P/L menunjukkan opsi untuk “Ganti Password”.
12. User memilih “Ganti Password”.	
	13. Menunjukkan UI untuk penggantian password.
	14. Meminta <i>input email</i> .
15. Memasukkan data <i>email</i> yang sudah terdaftar.	
	15. Mengirimkan kode ke <i>email</i> user.
16. Memasukkan kode yang sesuai dengan yang tertera di <i>email</i> .	
	17. Melakukan validasi <i>input</i> kode.
	18. Apabila validasi benar, meminta <i>input password</i> baru.
19. Memasukkan <i>password</i> baru.	
	19. Melakukan validasi dengan <i>password</i> yang terdapat pada <i>database</i> .
	20. Apabila berbeda, gunakan <i>password</i> baru. Sebaliknya, berikan pesan bahwa <i>input</i> sama dengan <i>password</i> lama.
	21. Memperbarui <i>database user</i> dengan <i>password</i> yang baru.

3.4.2 Skenario Use Case 02

Nama Use Case: Membuat profil data keuangan

Skenario:

Tabel 3.4.2. Skenario Use Case 02: Membuat profil data keuangan

Aksi Aktor (Mahasiswa)	Reaksi P/L
<i>Skenario Normal</i>	
1. User membuka aplikasi dan memilih menu “Profil Keuangan”, lalu menekan “Masukkan Data”.	
	2. Menampilkan layar Form Profil Keuangan dengan tiga seksi: Pemasukan Bulanan, Pengeluaran Bulanan, dan Target Tabungan Bulanan.
3. User memasukkan nominal, kategori, dan keterangan pemasukan dalam satuan IDR.	
	4. Memvalidasi <i>input</i> masukan user, nominal > 0.
5. User menambahkan memasukkan nominal, kategori, dan keterangan pengeluaran untuk pengeluaran yang bersifat langganan.	
	6. Memvalidasi <i>input</i> pengeluaran langganan user, nominal > 0.
7. User memasukkan nominal target pengeluaran bulanan	
	7. Memvalidasi <i>input</i> target pengeluaran langganan user, nominal > 0.
8. User menekan tombol “Simpan”.	
	8. Menyimpan data pemasukan, pengeluaran langganan, dan target pengeluaran di <i>database</i> .

3.4.3 Skenario Use Case 03

Nama Use Case: Memilih fitur dan melakukan input data transaksi

Skenario:

Tabel 3.4.3. Skenario Use Case 03: Memilih fitur dan melakukan input data transaksi

Aksi Aktor (Mahasiswa)	Reaksi P/L
<i>Skenario Normal</i>	
1. User membuka aplikasi dan memilih menu “Masukkan Data Transaksi”	
	2. Menampilkan layar Form Transaksi dengan isian Waktu Transaksi, Jenis Transaksi (Pemasukan/Pengeluaran), Nominal, Kategori, Akun Keuangan (E-Wallet, Bank, dan lain-lain), Deskripsi Tambahan
3. User memasukkan isian, waktu transaksi, jenis transaksi, nominal, kategori, akun keuangan yang digunakan, dan deskripsi tambahan	
	4. Memvalidasi <i>input</i> user dengan nominal > 0
5. User menekan tombol “Simpan”	
	6. Menyimpan <i>input</i> user di <i>database</i>
	7. Menyesuaikan saldo yang sudah ada berdasarkan input user (Pemasukan/Pengeluaran)

3.4.4 Skenario Use Case 04

Nama Use Case: Memilih fitur dan melihat rekapitulasi keuangan

Skenario:

Tabel 3.4.4. Skenario Use Case 04: Memilih fitur dan melihat rekapitulasi keuangan

Aksi Aktor (Mahasiswa)	Reaksi P/L
<i>Skenario Normal</i>	
1. User membuka aplikasi dan memilih menu “Lihat Rekapitulasi Transaksi”	
	2. Menampilkan layar pilihan visualisasi untuk rekapitulasi transaksi, yaitu dalam bentuk tabel/list laporan keuangan atau diagram, seperti diagram lingkaran (<i>pie chart</i>)
3. User memilih pilihan visualisasi dalam bentuk tabel/list	
	4. Menampilkan saldo setiap akun keuangan beserta tabel berisikan semua pengeluaran yang dapat di filter berdasarkan masing-masing kolom
	5. Menampilkan pilihan filter berdasarkan kategori, akun keuangan, dan waktu transaksi
6. User memilih pilihan kategori untuk filter	
	7. Menampilkan tabel berdasarkan filter yang dipilih user
<i>Skenario Alternatif: Visualisasi dalam bentuk diagram lingkaran</i>	
3. User memilih pilihan visualisasi dalam bentuk diagram lingkaran	
	4. Menampilkan saldo setiap akun keuangan beserta diagram lingkaran berisikan semua pengeluaran berdasarkan kategori pengeluaran
	5. Menampilkan pilihan filter berdasarkan kategori, akun keuangan, dan waktu transaksi
6. User memilih pilihan kategori untuk filter	
	7. Menampilkan diagram lingkaran berdasarkan filter yang dipilih user

3.4.5 Skenario Use Case 05

Nama Use Case: Mendapatkan notifikasi pengingat harian

Skenario:

Tabel 3.4.5. Skenario Use Case 05: Mendapatkan notifikasi pengingat harian

Aksi Aktor (Mahasiswa)	Reaksi P/L
<i>Skenario Normal</i>	
1. User membuka aplikasi dan mengaktifkan fitur notifikasi harian	
	2. Sistem menampilkan pengaturan notifikasi harian
3. User memilih waktu pengingat	
	4. Sistem menyimpan jadwal notifikasi harian ke dalam database
5. User memilih kategori pengingat harian	
	6. Sistem menampilkan konfirmasi pengaturan berhasil disimpan

7. User mendapatkan notifikasi harian sesuai dengan jadwal notifikasi yang ditentukan	
	8. Sistem memunculkan notifikasi di layar desktop berisi pesan sesuai user
Skenario Alternatif: User tidak memilih waktu pengingat harian	
	2. Sistem menampilkan pengaturan notifikasi harian
3. User tidak memilih waktu pengingat	
	4. Sistem menyimpan jadwal notifikasi harian secara <i>default</i> (Jam 20.00) ke <i>database</i>
5. User memilih kategori pengingat harian	
	6. Sistem menampilkan konfirmasi pengaturan berhasil disimpan
7. User mendapatkan notifikasi harian sesuai dengan jadwal <i>default</i>	
	8. Sistem memunculkan notifikasi di layar desktop berisi pesan sesuai <i>user</i>

3.4.6 Skenario Use Case 06

Nama Use Case: Mendapatkan peringatan anggaran

Skenario:

Tabel 3.4.5. Skenario Use Case 06 Mendapatkan peringatan anggaran

Aksi Aktor (Mahasiswa)	Reaksi P/L
<i>Skenario Normal</i>	
1. User membuka aplikasi dan mengaktifkan fitur pengingat.	
	2. Sistem menampilkan pengaturan peringatan
3. User memilih batas nominal pengeluaran untuk pengingat dikirimkan	
4. User memilih jarak dari tanggal jatuh tempo untuk pengingat pengeluaran periodik dikirimkan	
	5. Sistem menyimpan informasi batasan pengeluaran dan jadwal untuk mengirim peringatan ke dalam database
	6. Sistem menampilkan konfirmasi pengaturan berhasil disimpan
7. User mendapatkan peringatan sesuai dengan batasan dan jadwal yang ditentukan	
	8. Sistem mengirimkan pengingat di layar desktop berisi pesan sesuai user
Skenario Alternatif: User tidak memasukkan batas nominal pengeluaran	
	2. Sistem menampilkan pengaturan peringatan
3. User tidak memasukkan batas nominal pengeluaran akan dikirimkan	
4. User memilih jarak dari tanggal jatuh tempo untuk pengingat pengeluaran periodik dikirimkan	
	5. Sistem menyimpan batasan pengeluaran secara <i>default</i> yaitu 80% dari alokasi pengeluaran dan jadwal untuk ditentukan untuk mengirim peringatan ke dalam database

Skenario Alternatif: User tidak memilih waktu pengingat untuk pembayaran jatuh tempo	
	2. Sistem menampilkan pengaturan peringatan
3. User memilih batas nominal pengeluaran untuk pengingat dikirimkan	
4. User tidak memilih jarak dari tanggal jatuh tempo untuk pengingat pengeluaran periodik dikirimkan	
	5. Sistem menyimpan informasi batasan pengeluaran yang ditentukan dan jadwal secara <i>default</i> yaitu 1 dan 3 hari sebelum jatuh tempo untuk mengirim peringatan ke dalam database
	6. Sistem menampilkan konfirmasi pengaturan berhasil disimpan
7. User mendapatkan peringatan sesuai dengan batasan dan jadwal yang ditentukan	
	8. Sistem mengirimkan pengingat di layar desktop berisi pesan sesuai user

4 Diagram Kelas

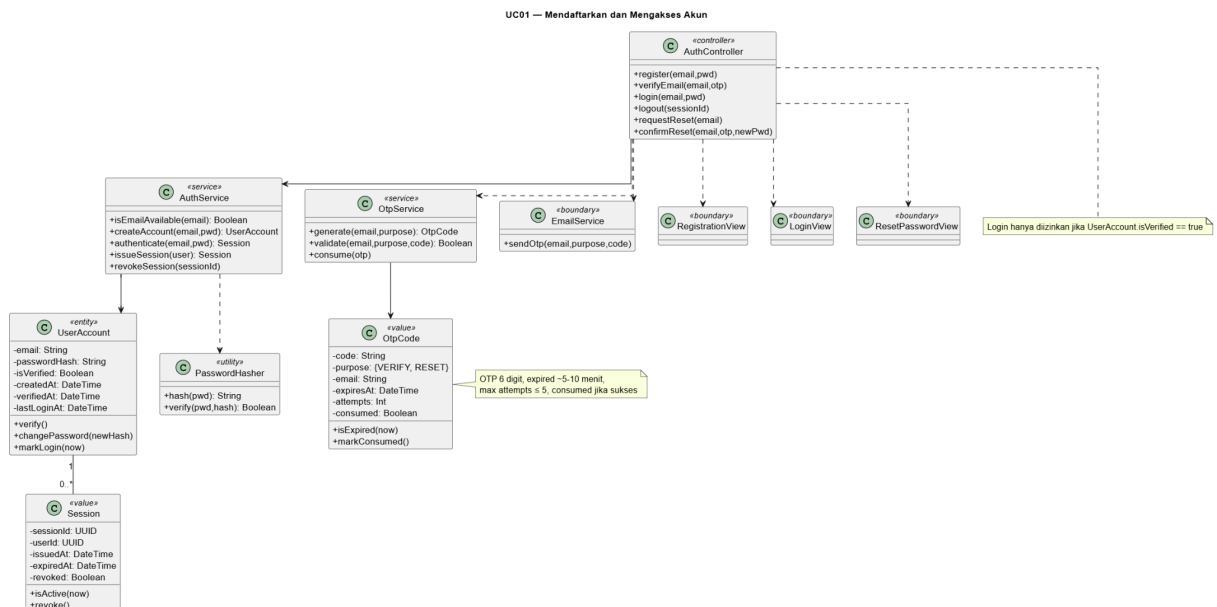
4.1 Use Case Mendaftarkan Dan Mengakses Akun

4.1.1 Identifikasi Kelas Use Case Mendaftarkan Dan Mengakses Akun

Tabel 4.1.1 Identifikasi Kelas UC01

ID Kelas	Nama Kelas	Deskripsi Kelas	ID Use Case
C-01	UserAccount	Menyimpan kredensial dan status akun (email, passwordHash, isVerified, createdAt).	UC01
C-02	AuthController	Mengorkestrasi pendaftaran, login, logout, reset password.	UC01
C-03	AuthService	Memuat logika autentikasi (verifikasi kredensial, hashing & verifikasi password, pembuatan token sesi).	UC01
C-04	PasswordHasher	Proteksi data User menggunakan algoritma <i>hashing</i> & verifikasi password.	UC01
C-05	Session	Representasi sesi login (sessionId/token, issuedAt, expiredAt, revoked).	UC01
C-06	EmailService	Mengirimkan email verifikasi/kode reset password	UC01
C-07	OtpService	Generate, simpan, dan validasi OTP.	UC01
C-08	OtpCode	Kode OTP (value, purpose=VERIFY/RESET, expiresAt, attempts, consumed).	UC01
C-09	RegistrationView	Tampilan input “Daftar Akun”.	UC01
C-10	LoginView	Tampilan input “Masuk Akun”.	UC01
C-11	ResetPasswordView	Tampilan input email, kode OTP, dan password baru.	UC01

4.1.2 Diagram Kelas Use Case Mendaftarkan Dan Mengakses Akun



Gambar 4.1 Diagram Kelas Use Case Mendaftarkan Dan Mengakses Akun

Tabel 4.2. Komponen Kelas UC01

ID Kelas	Nama Kelas	Atribut	Metode/Operasi
C-01	UserAccount	userId: UUID; email: String; passwordHash: String; isVerified: Boolean; createdAt: DateTime; verifiedAt?: DateTime; lastLoginAt?: DateTime	verify(); markLogin(now); changePassword(newHash)
C-02	AuthController	-	register(email, pwd); verifyEmail(email, otp); login(email, pwd); logout(sessionId); requestReset(email); confirmReset(email, otp, newPwd)
C-03	AuthService	passwordHasher: PasswordHasher	isEmailAvailable(email): Boolean; createAccount(email, pwd): UserAccount; authenticate(email, pwd): Session; issueSession(user): Session; revokeSession(sessionId)
C-04	PasswordHasher	-	hash(pwd): String; verify(pwd, hash): Boolean
C-05	Session	sessionId: UUID; userId: UUID; issuedAt: DateTime; expiredAt: DateTime; revoked: Boolean	isActive(now): Boolean; revoke()
C-06	EmailService	-	sendOtp(email, purpose, code)
C-07	OtpService	-	generate(email, purpose): OtpCode; validate(email, purpose, code): Boolean; consume(otp): void
C-08	OtpCode	code: String; purpose: Enum {VERIFY, RESET}; email: String; expiresAt: DateTime; attempts: Int; consumed: Boolean	isExpired(now): Boolean; canAttempt(): Boolean; markConsumed()
C-09	RegistrationView	-	captureEmailPassword(); showSuccess(); showErrorMessage()
C-10	LoginView	-	captureEmailPassword(); showLoginSuccess(); showLoginErrorMessage()

C-11	ResetPasswordView	-	captureEmail(); captureOtpAndNewPassword(); showResetSuccess(); showError(msg)
------	-------------------	---	---

4.2 Use Case Membuat Profil Data Keuangan

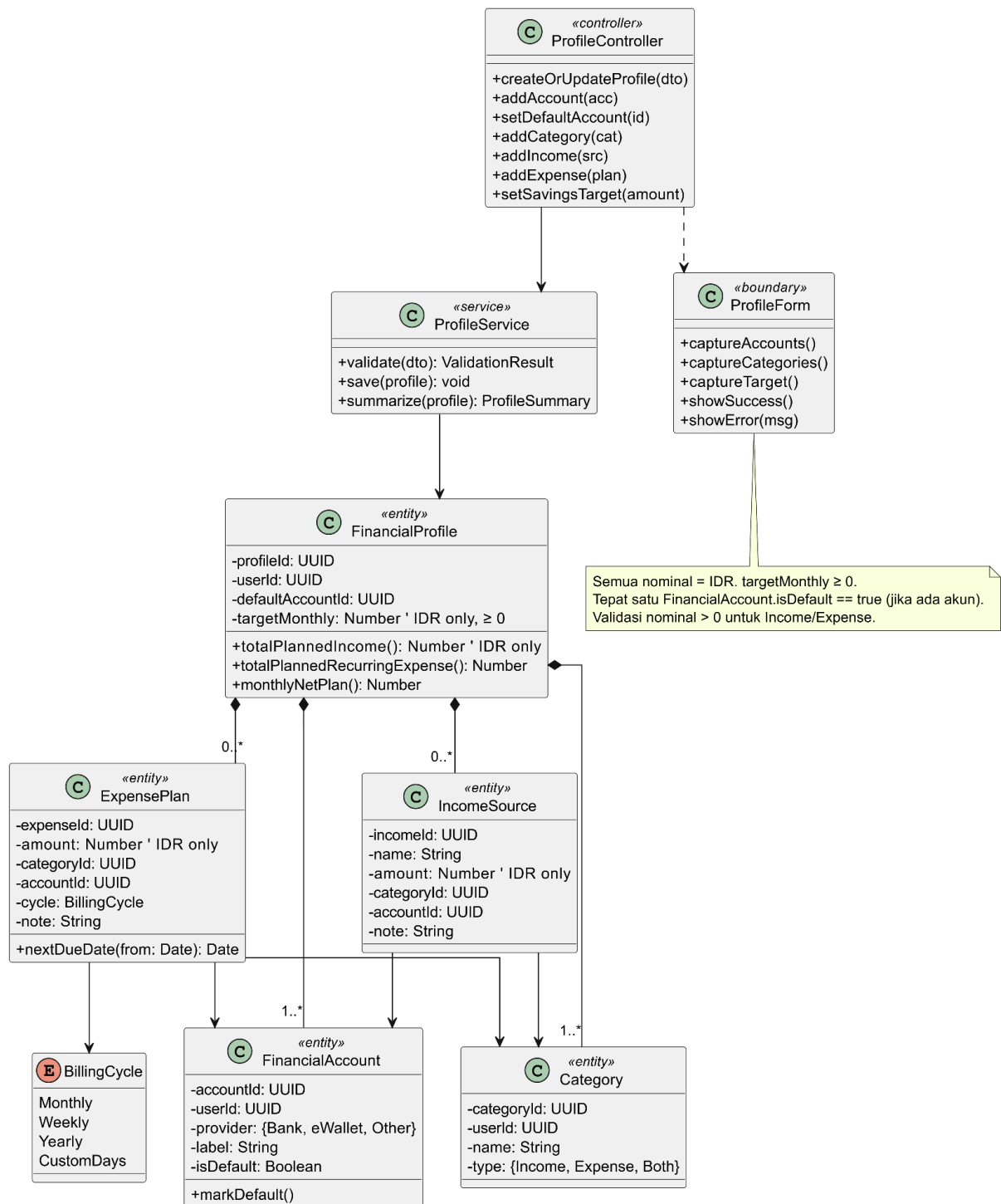
4.2.1 Identifikasi Kelas Use Case Membuat Profil Data Keuangan

Tabel 4.3. Identifikasi Kelas UC02

ID Kelas	Nama Kelas	Deskripsi Kelas	ID Use Case
C-12	FinancialProfile	Profil keuangan per user (jumlah pemasukan bulanan, batas pengeluaran, target tabungan).	UC02
C-13	FinancialAccount	Sumber dana <i>e-wallet</i> User (myBCA, GoPay, dll).	UC02
C-14	Category	Kategori pengeluaran <i>custom</i> per User (misal: Makanan, Transport, Gaji, Freelance).	UC02
C-15	IncomeSource	Sumber pemasukan bulanan (nama/kategori, nominal, keterangan).	UC02
C-16	ExpensePlan	Rencana pengeluaran berulang/berlangganan (kategori, nominal, periode).	UC02
C-17	BillingCycle	Keterangan siklus tagihan (bulanan, mingguan, tahunan, atau <i>custom</i>).	UC02
C-18	ProfileController	Pembuatan/penyuntingan profil.	UC02
C-19	ProfileService	Validasi & penyimpanan profil; perhitungan ringkasan.	UC02
C-20	ProfileForm	Input Profil Keuangan (pemasukan, pengeluaran langganan, dan target menabung).	UC02

4.2.2 Diagram Kelas Use Case Membuat Profil Data Keuangan

UC02 — Membuat Profil Data Keuangan



Gambar 4.2 Diagram Kelas Use Case Membuat Profil Data Keuangan

Tabel 4.3. Komponen Kelas UC02

ID Kelas	Nama Kelas	Atribut	Metode/Operasi
C-12	FinancialProfile	profileId: UUID; userId: UUID; target: SavingsTarget; defaultAccountId?: UUID; createdAt, updatedAt	totalPlannedIncome(): Rupiah; totalPlannedRecurringExpense(): Rupiah; monthlyNetPlan(): Rupiah
C-13	FinancialAccount	accountId: UUID; userId: UUID; name: String; provider: Enum {Bank, eWallet, Other}; label: String (mis. "myBCA"); isDefault: Boolean	markDefault(); rename(label)
C-14	Category	categoryId: UUID; userId: UUID; name: String; type: Enum {Income, Expense, Both}	-
C-15	IncomeSource	incomeId: UUID; profileId: UUID; name: String; amount: Rupiah; categoryId?: UUID; accountId?: UUID; note?: String	-
C-16	ExpensePlan	expenseId: UUID; profileId: UUID; categoryId?: UUID; amount: Rupiah; cycle: BillingCycle; accountId?: UUID; note?: String	nextDueDate(from: Date): Date
C-17	BillingCycle	Enum {Monthly, Weekly, Yearly, CustomDays}	-
C-18	ProfileController	-	createOrUpdateProfile(dto); addAccount(acc); setDefaultAccount(id); addCategory(cat); addIncome(src); addExpense(plan); setSavingsTarget(amount)
C-19	ProfileService	-	validate(dto): ValidationResult; save(profile): void; summarize(profile): ProfileSummary
C-20	ProfileForm	-	captureAccounts(); captureCategories(); captureTarget(); showSuccess(); showError(msg)

4.3 Use Case Memilih Fitur dan Melakukan Input Data Transaksi

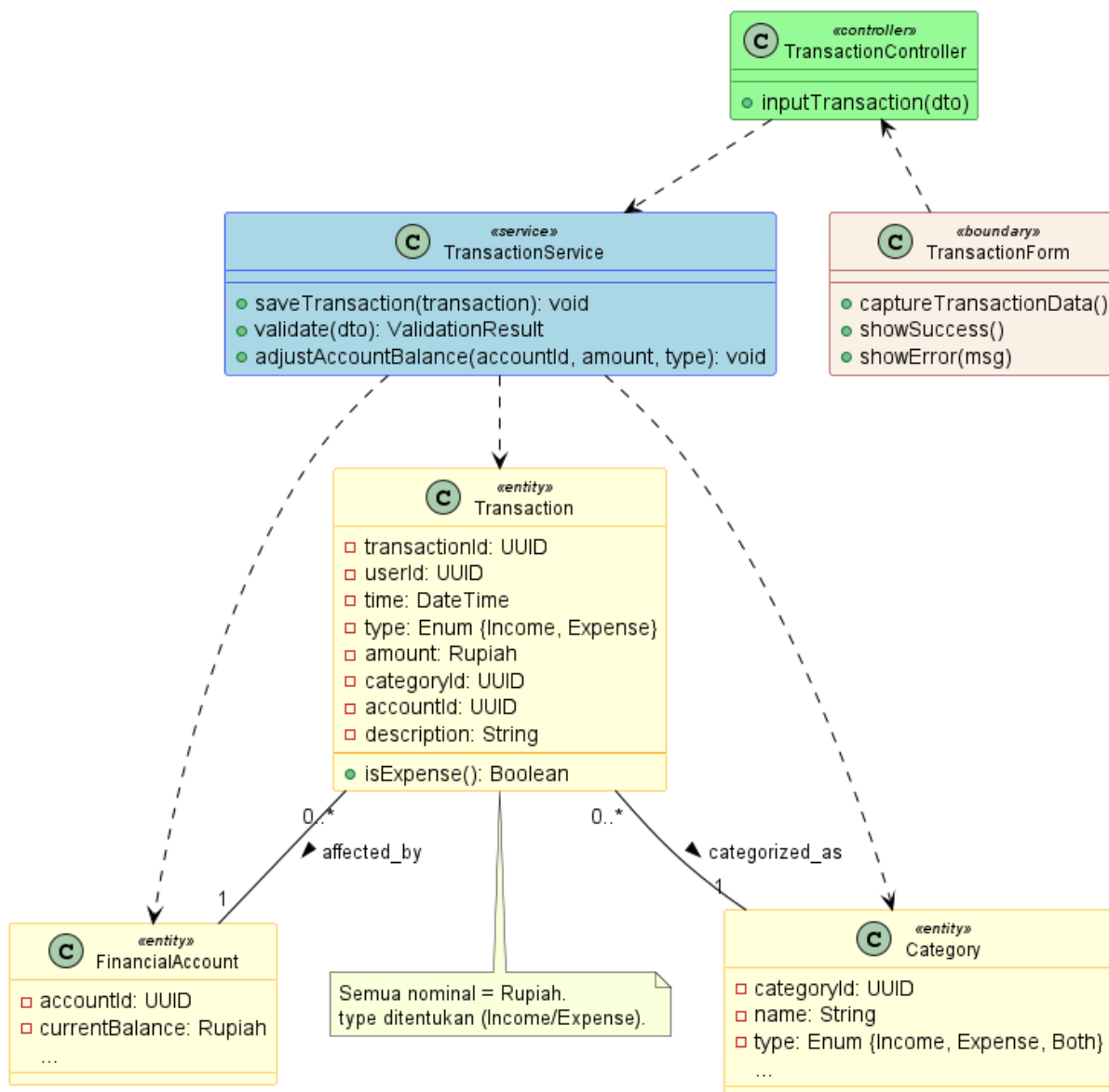
4.3.1 Identifikasi Kelas Use Case Memilih Fitur dan Melakukan Input Data Transaksi

Tabel 4.5. Identifikasi Kelas UC03

ID Kelas	Nama Kelas	Deskripsi Kelas	ID Use Case
C-21	Transaction	Menyimpan detail transaksi (waktu, jenis, nominal, kategori, akun, deskripsi).	UC03
C-22	TransactionController	Mengatur proses input dan penyimpanan transaksi.	UC03
C-23	TransactionService	Memuat logika bisnis untuk validasi, penyimpanan, dan penyesuaian saldo transaksi.	UC03
C-24	TransactionForm	Tampilan input untuk merekam data transaksi baru.	UC03
C-13	FinancialAccount	Sumber dana yang terpengaruh oleh transaksi (Bank, eWallet, dll).	UC02, UC03
C-14	Category	Kategori transaksi (Pemasukan atau Pengeluaran).	UC02, UC03

4.3.2 Diagram Kelas Use Case Memilih Fitur dan Melakukan Input Data Transaksi

UC03 - Memilih Fitur dan Melakukan Input Data Transaksi



Gambar 4.3 Diagram Kelas Use Case Memilih Fitur dan Melakukan Input Data Transaksi

Tabel 4.6. Komponen Kelas UC03

ID Kelas	Nama Kelas	Atribut	Metode/Operasi
C-21	Transaction	transactionId: UUID; userId: UUID; time: DateTime; type: Enum {Income, Expense}; amount: Rupiah; categoryId?: UUID; accountId: UUID; description?: String;	isExpense(): Boolean; updateBalance();
C-22	TransactionController		inputTransaction(dto);
C-23	TransactionService		saveTransaction(transaction); validate(dto);

			ValidationResult; adjustAccountBalance(a ccountId, amount, type): void;
C-24	TransactionForm		captureTransactionData(); showSuccess(); showError(msg);
C-13	FinancialAccount	accountId: UUID; userId: UUID; provider: Enum {Bank, eWallet, Other}; label: String; isDefault: Boolean; currentBalance: Rupiah	markDefault(); rename(label); updateBalance(amount, type)
C-14	Category	categoryId: UUID; userId: UUID; name: String; type: Enum {Income, Expense, Both}	

4.4 Use Case Memilih Fitur dan Melihat Rekapitulasi Keuangan

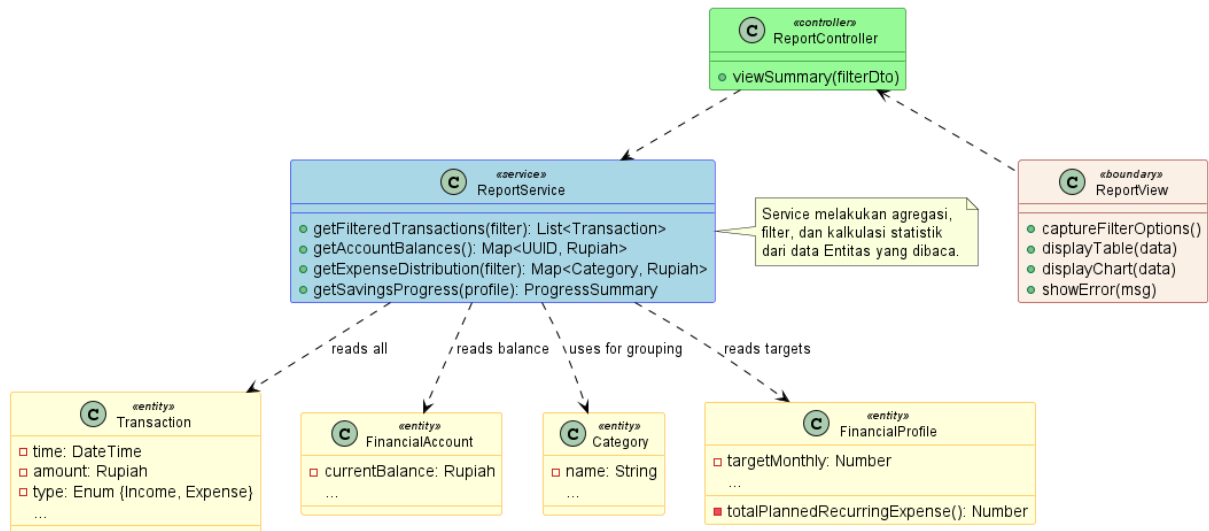
4.4.1 Identifikasi Kelas Use Case Memilih Fitur dan Melihat Rekapitulasi Keuangan

Tabel 4.7. Identifikasi Kelas UC04

ID Kelas	Nama Kelas	Deskripsi Kelas	ID Use Case
C-25	ReportController	Mengatur permintaan laporan dan statistik keuangan.	UC04
C-26	ReportService	Memuat logika untuk mengambil, memfilter, dan mengagregasi data transaksi menjadi laporan/statistik.	UC04
C-27	ReportView	Tampilan untuk menampilkan rekapitulasi (tabel/grafik) dan filter.	UC04
C-21	Transaction	Data sumber untuk rekapitulasi dan visualisasi.	UC04
C-12	FinancialProfile	Menyediakan target anggaran/tabungan untuk perbandingan.	UC02, UC04
C-13	FinancialAccount	Menyediakan saldo dan label akun untuk laporan.	UC02, UC04
C-14	Category	Menyediakan label kategori untuk filter dan visualisasi.	UC02, UC04

4.4.2 Diagram Kelas Use Case Memilih Fitur dan Melihat Rekapitulasi Keuangan

UC04 - Memilih Fitur dan Melihat Rekapitulasi Keuangan



Gambar 4.4 Diagram Kelas Use Case Memilih Fitur dan Melihat Rekapitulasi Keuangan

Tabel 4.8. Komponen Kelas UC04

ID Kelas	Nama Kelas	Atribut	Metode/Operasi
C-25	ReportController		viewSummary(filterDto);
C-26	ReportService		getFilteredTransactions(filter): List<Transaction>; getAccountBalances(): Map<UUID, Rupiah>; getExpenseDistribution(filter): Map<Category, Rupiah>; getIncomeDistribution(filter): Map<Category, Rupiah>; getSavingsProgress(profile): ProgressSummary;
C-27	ReportView		captureFilterOptions(); displayTable(data); displayChart(data);
C-21	Transaction	transactionId: UUID; userId: UUID; time: DateTime; type: Enum {Income, Expense}; amount: Rupiah; categoryId?: UUID; accountId: UUID; description?: String;	
C-12	FinancialProfile	profileId: UUID; userId: UUID; targetMonthly: Rupiah; totalPlannedRecurringExpense()	

C-13	FinancialAccount	accountId: UUID; userId: UUID; currentBalance: Rupiah	
C-14	Category	categoryId: UUID; userId: UUID; name: String; type: Enum {Income, Expense, Both}	

4.5 Use Case Mendapatkan Notifikasi Pengingat Harian

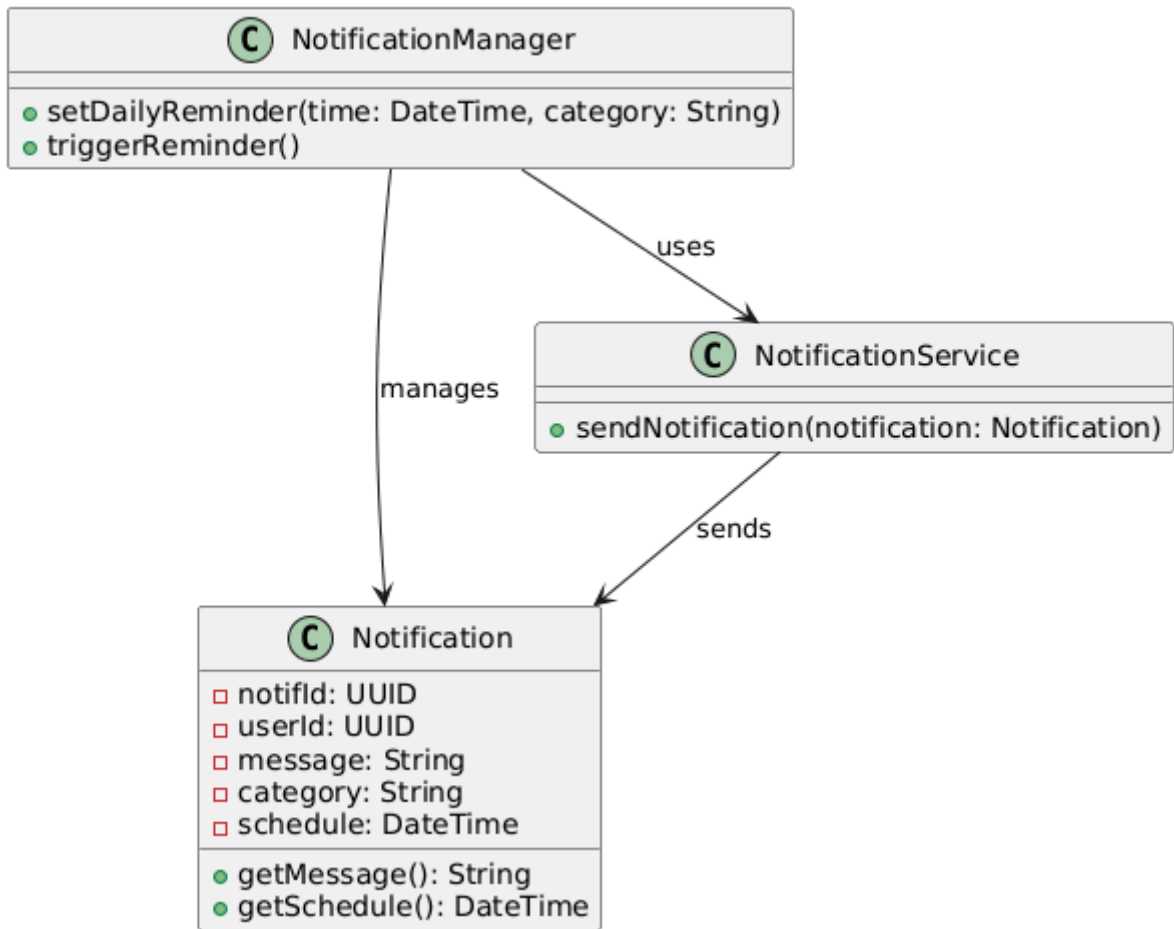
4.5.1 Identifikasi Kelas Use Case Mendapatkan Notifikasi Pengingat Harian

Tabel 4.9. Identifikasi Kelas UC05

ID Kelas	Nama Kelas	Deskripsi Kelas	ID Use Case
C-28	Notification	Informasi inti dari setiap notifikasi yang akan dikirimkan seperti ID dari notifikasi, pesan dari notifikasi, kategori, serta jadwal untuk dikirimkan.	UC05
C-29	NotificationManager	Memuat metode untuk menentukan waktu dan tanggal pengiriman notifikasi dan mengirimkan notifikasi tersebut.	UC05
C-30	NotificationService	Memuat metode untuk mengirimkan notifikasi.	UC05

4.5.2 Diagram Kelas Use Case Mendapatkan Notifikasi Pengingat Harian

Class Diagram UC05 - Notification



Gambar 4.5 Diagram Kelas Use Case Mendapatkan Notifikasi Pengingat Harian

Tabel 4.10. Komponen Kelas UC05

ID Kelas	Nama Kelas	Atribut	Metode/Operasi
C-28	Notification	notifId: UUID; userId: UUID; message: String; schedule: DateTime; category: String	schedule(time); send()
C-29	NotificationManager	-	setDailyReminder(time, category); triggerReminder()
C-30	NotificationService	-	saveSetting(userId, notif); sendNotification(notif)

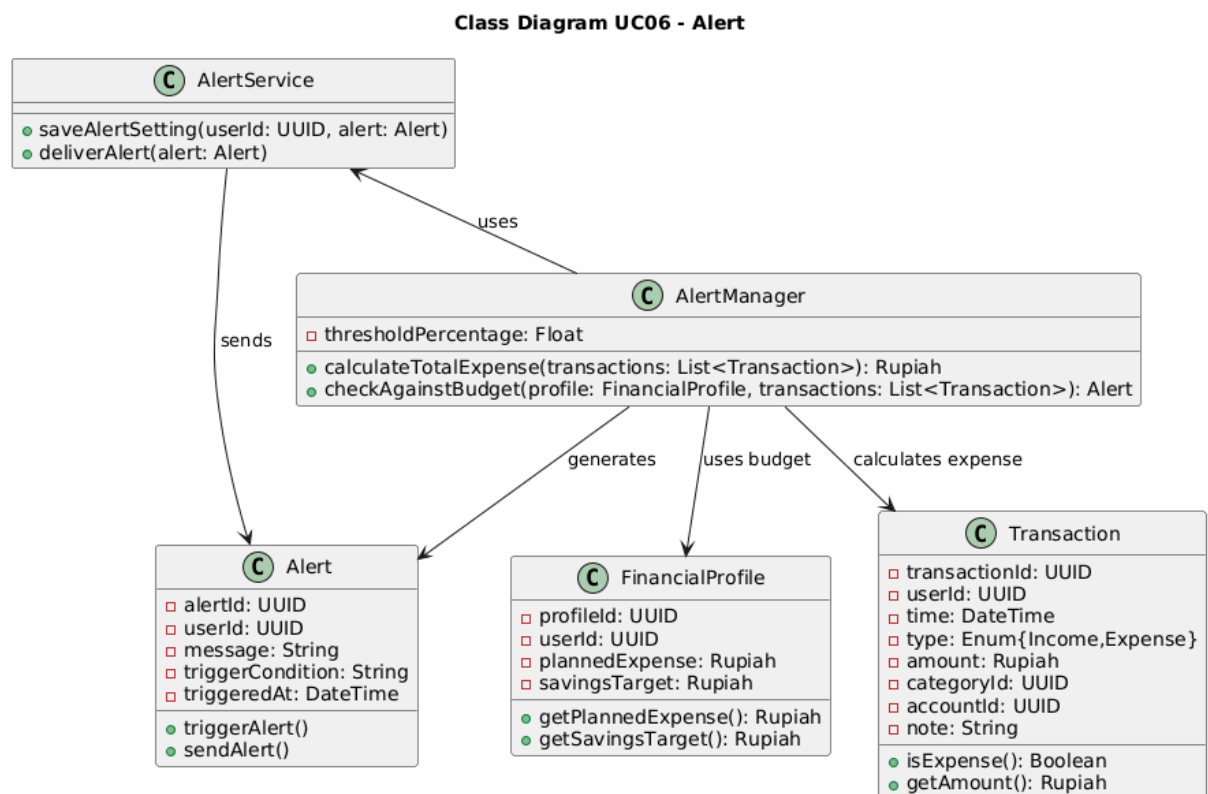
4.6 Use Case Mendapatkan Peringatan Anggaran

4.6.1 Identifikasi Kelas Use Case Mendapatkan Peringatan Anggaran

Tabel 4.11. Identifikasi Kelas UC06

ID Kelas	Nama Kelas	Deskripsi Kelas	ID Use Case
C-12	FinancialProfile	Profil keuangan user (pemasukan, pengeluaran rutin, target tabungan, batas pengeluaran). Jadi sumber data budget.	UC02
C-21	Transaction	Data transaksi aktual (waktu, nominal, kategori, akun, deskripsi).	UC03
C-31	Alert	Representasi peringatan anggaran (pesan, kondisi, jadwal).	UC06
C-32	AlertManager	Menghitung total pengeluaran dari transaksi, membandingkan dengan budget dari FinancialProfile, memicu alert jika melewati ambang batas.	UC06
C-33	AlertService	Menyimpan konfigurasi alert & mengirim peringatan ke user.	UC06

4.6.2 Diagram Kelas Use Case Mendapatkan Peringatan Anggaran



Gambar 4.6 Diagram Kelas Use Case Mendapatkan Peringatan Anggaran

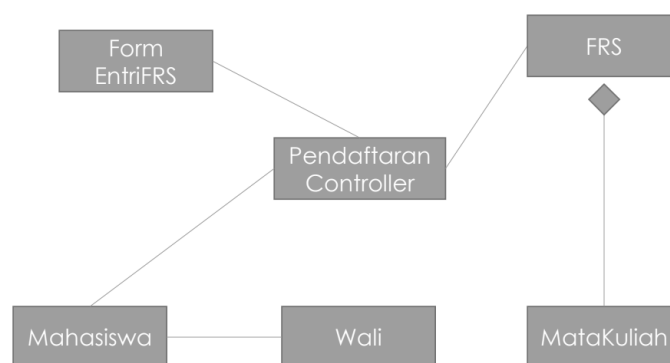
Tabel 4.12. Komponen Kelas UC06

ID Kelas	Nama Kelas	Atribut	Metode/Operasi
C-12	FinancialProfile	profileId: UUID; userId: UUID; plannedExpense: Rupiah; savingsTarget: Rupiah	getPlannedExpense(); getSavingsTarget()
C-21	Transaction	transactionId: UUID; userId: UUID; time: DateTime; type: Enum{Income,Expense}; amount: Rupiah; categoryId?: UUID	isExpense(): Boolean; getAmount(): Rupiah
C-31	Alert	alertId: UUID; userId: UUID; message: String; triggerCondition: String; triggeredAt: DateTime	triggerAlert(); sendAlert()
C-32	AlertManager	thresholdPercentage: Float	calculateTotalExpense(transactions): Rupiah; checkAgainstBudget(profile, transactions): Alert
C-33	AlertService	-	saveAlertSetting(userId, alert); deliverAlert(alert)

Untuk penomoran diagram kelas keseluruhan, silahkan edit suffix dari numbering menjadi angka lanjutan dari jumlah usecase kalian. Apabila use case ada 5 maka suffix-nya diganti menjadi 6.

4.7 Diagram Kelas Keseluruhan

Gambarkan gabungan diagram kelas yang telah dibuat per usecase di sini. Penggambaran diagram kelas cukup dituliskan nama kelasnya saja. Daftar attribute dan method dapat dituliskan dalam tabel terpisah. Tuliskan attribute dan method seperlunya saja.



Gambar 7. Diagram Kelas Keseluruhan

Tabel 4.2. <Nama Tabel>

ID Kelas	Nama Kelas	Atribut	Metode/Operasi
C-01	Class-1		

<i>C-02</i>	<i>Class-2</i>		
<i>C-02</i>	<i>Class-3</i>		

5 Traceability

Cocokkan setiap kebutuhan fungsional, use case, dengan diagram kelas yang mendukung atau mengimplementasikan kebutuhan tersebut.

Tabel 5.1 <Nama Tabel>

Kode Kelas	Kode Use Case	Kode Kebutuhan Fungsional
<i>C-01</i>	<i>UC01</i>	<i>KF01, KF02</i>
<i>C-02</i>	<i>UC02</i>	<i>KF03, KF04</i>