

TUGAS BESAR
“*SplitIt: Smart Expense Splitter*”



Disusun Oleh:

Edelweiss Salsabilla - 103052300038
Muthia Rezi Aisyah - 103052300114
Amalia Ananda Putri - 103052330078
Alishadena C.N.R.H - 103052300032
Annisa Azzahra Rahmah - 103052300056

PEMROGRAMAN BERORIENTASI OBJEK
PROGRAM STUDI S1 SAINS DATA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
2026

DAFTAR ISI

| | |
|---|-----------|
| PENDAHULUAN..... | 3 |
| 1.1 Latar Belakang..... | 3 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Tujuan Pengembangan..... | 3 |
| 1.5 Ruang Lingkup..... | 4 |
| RANCANGAN ARSITEKTUR DAN KONSEP PBO..... | 5 |
| 2.1. Arsitektur Aplikasi (Strategy Pattern)..... | 5 |
| 2.2. Penerapan Konsep PBO..... | 5 |
| 2.2.1. Inheritance dan Abstract Class..... | 5 |
| 2.2.2. Interface dan Polymorphism..... | 5 |
| 2.3. Prinsip PBO yang Digunakan..... | 6 |
| RANCANGAN FITUR DAN IMPLEMENTASI..... | 7 |
| 3.1. Daftar Fitur (List Features)..... | 7 |
| 3.2. Diagram Kelas (Class Diagram)..... | 8 |
| 3.3. Deskripsi Detail Kelas..... | 9 |
| IMPLEMENTASI FITUR DAN PEMBAGIAN TUGAS..... | 10 |
| HASIL APLIKASI..... | 11 |
| 5.1. Halaman Dashboard..... | 11 |
| 5.2. Halaman Buat/Lihat Group..... | 12 |
| 5.3. Halaman Detail Group + Anggota..... | 13 |
| 5.4. Halaman Tambah Transaksi Even (bagi rata)..... | 14 |
| 5.5. Halaman Tambah Transaksi Uneven (bagi rata)..... | 15 |
| 5.6. Halaman Summary..... | 17 |
| 5.7. Demo Website..... | 20 |
| PENUTUP..... | 21 |
| 6.1. Kesimpulan..... | 21 |

PENDAHULUAN

1.1 Latar Belakang

Dalam lingkungan akademik dan sosial, mahasiswa seringkali terlibat dalam kegiatan kelompok yang memerlukan pengelolaan biaya bersama (patungan). Kegiatan seperti makan bersama, membeli perlengkapan tugas, atau berlibur bersama seringkali menimbulkan kebingungan saat proses pembagian tagihan. Perhitungan manual rentan terhadap kesalahan (*human error*), tidak transparan, dan memakan waktu yang cukup lama, terutama jika metode pembagiannya kompleks (misalnya, ada yang membayar lebih dulu, ada yang tidak ikut dalam satu item atau kondisi, dan lainnya).

Kondisi tersebut dapat menimbulkan kecanggungan sosial dan potensi konflik kecil. Oleh karena itu, untuk mengatasi masalah ini, diusulkan pengembangan aplikasi berbasis Pemrograman Berorientasi Objek (PBO) yang diberi nama “SplitIt: *Smart Expense Splitter*”.

Aplikasi SplitIt dirancang untuk menyederhanakan dan mengotomatisasi seluruh proses *split bill* (patungan). Pengguna dapat membuat grup, mencatat setiap transaksi, dan membiarkan sistem menghitung secara adil dan transparan siapa yang berhutang kepada siapa hingga semua neraca keuangan impas (*settled*).

1.2 Rumusan Masalah

1. Bagaimana merancang sistem yang dapat mencatat transaksi pengeluaran dalam sebuah grup?
2. Bagaimana mengimplementasikan logika pembagian biaya yang fleksibel, mencakup pembagian rata (*even*) dan tidak rata (*uneven*)?
3. Bagaimana menerapkan konsep PBO (*inheritance, abstract class, interface*, dan lainnya) secara efektif untuk membangun arsitektur yang kokoh?
4. Bagaimana sistem dapat mengkalkulasi dan menyajikan rangkuman hutang-piutang yang sederhana dan mudah dipahami?

1.3 Tujuan Pengembangan

Tujuan dari pengembangan aplikasi SplitIt antara lain:

1. Membangun aplikasi yang mampu mengelola grup, anggota, dan mencatat transaksi keuangan.
2. Mengimplementasikan *design pattern (strategy pattern)* menggunakan *interface* untuk menangani berbagai metode pembagian biaya.
3. Menerapkan *inheritance* dan *abstract class* untuk menciptakan model data yang konsisten.
4. Menghasilkan kalkulator hutang yang dapat menyederhanakan alur pembayaran antar anggota.

1.4 Manfaat Aplikasi

1. Bagi Pengguna
Memberikan kemudahan, transparansi, dan keadilan dalam mengelola keuangan kelompok, serta mengurangi potensi konflik sosial.
2. Bagi Pengembang
Mengaplikasikan secara praktis teori dan konsep Pemrograman Berorientasi Objek (PBO) dalam sebuah proyek nyata.

1.5 Ruang Lingkup

Untuk memastikan proyek ini layak (*feasible*) dan dapat diselesaikan sesuai *timeline* mata kuliah, maka ditetapkan batasan (ruang lingkup) sebagai berikut.

1. Fokus Utama
Logika bisnis (OOP) di *backend*.
2. Yang akan Dibuat
Manajemen grup, manajemen anggota, pencatatan transaksi (dengan tipe bagi rata (*even*) dan bagi tidak rata (*uneven*)), kalkulasi rangkuman hutang, serta koneksi ke *database* (phpMyAdmin).
3. Yang Tidak akan Dibuat
Sistem *login* atau *register* (data pengguna bersifat lokal), fitur *payment gateway*, dan sinkronisasi antar perangkat.

RANCANGAN ARSITEKTUR DAN KONSEP PBO

2.1. Arsitektur Aplikasi (*Strategy Pattern*)

Aplikasi ini akan dirancang menggunakan *Strategy Design Pattern*. *Pattern* ini dipilih untuk menangani inti permasalahan, yaitu metode pembagian biaya yang berbeda.

Pada aplikasi ini, akan dipisahkan "konteks" (*transaction*) dari "strategi" (ISplitStrategy). *Transaction* akan memiliki sebuah referensi ke *interface* ISplitStrategy. Saat kalkulasi, *transaction* akan mendelegasikan pekerjaan ke strategi mana pun yang sedang digunakannya (EvenSplitStrategy atau UnevenSplitStrategy), tanpa harus tahu detail implementasinya.

2.2. Penerapan Konsep PBO

Proyek ini memenuhi syarat dari Pemrograman Berorientasi Objek (PBO) sebagai berikut.

2.2.1. *Inheritance* dan *Abstract Class*

a. Penerapan

Melalui sebuah *abstract class* bernama BaseEntity.

b. Tujuan

Kelas ini akan berfungsi sebagai *parent class* untuk semua model data utama (*User*, *Group*, dan *Transaction*).

c. Detail

BaseEntity akan memiliki atribut protected String id dan *method* konkret public String getId(). Ini memastikan bahwa setiap entitas data utama dalam sistem pasti memiliki ID yang unik sehingga memenuhi prinsip *inheritance* (pewarisan atribut dan *method*).

2.2.2. *Interface* dan *Polymorphism*

a. Penerapan melalui sebuah *interface* bernama ISplitStrategy.

b. *Interface* ini akan mendefinisikan "kontrak" atau *blueprint* untuk semua algoritma pembagian biaya.

c. Detail ISplitStrategy akan memiliki satu *method* abstrak, yaitu public Map<User, Double> calculateShares(...).

d. *Polymorphism* kelas Transaction akan memiliki atribut private ISplitStrategy strategy. Saat Transaction.executeSplit() dipanggil, ia akan memanggil strategy.calculateShares(). Perilaku yang terjadi akan berbeda-beda (polimorfik) tergantung pada objek konkret (EvenSplitStrategy atau UnevenSplitStrategy) yang di-inject ke dalam Transaction tersebut.

2.3. Prinsip PBO yang Digunakan

Prinsip Pemrograman Berorientasi Objek (PBO) yang akan digunakan dalam aplikasi ini adalah sebagai berikut.

1. *Encapsulation*

Atribut dibuat *private/protected* dan diakses lewat *getter* dan/atau *setter*.

2. *Abstraction*

Kelas BaseEntity dan ISplitStrategy memisahkan konsep dari implementasi.

3. *Inheritance*

User, Group, Transaction mewarisi BaseEntity.

4. *Polymorphism*

ISplitStrategy memungkinkan strategi pembagian bervariasi tanpa ubah kode utama.

5. *Composition*

Group memiliki daftar *User* dan *Transaction* yang akan dihapus bersamaan.

RANCANGAN FITUR DAN IMPLEMENTASI

3.1. Daftar Fitur (*List Features*)

1. F1: Manajemen Grup

Membuat grup baru dan menambahkan anggota/pengguna (*User*) ke dalamnya.

2. F2: Pencatatan Transaksi

Mencatat pengeluaran baru, mencakup siapa yang membayar, berapa totalnya, dan siapa saja yang terlibat.

3. F3: Strategi Bagi Rata (*Even Split*)

Logika untuk membagi totalAmount secara merata ke semua partisipan yang dipilih.

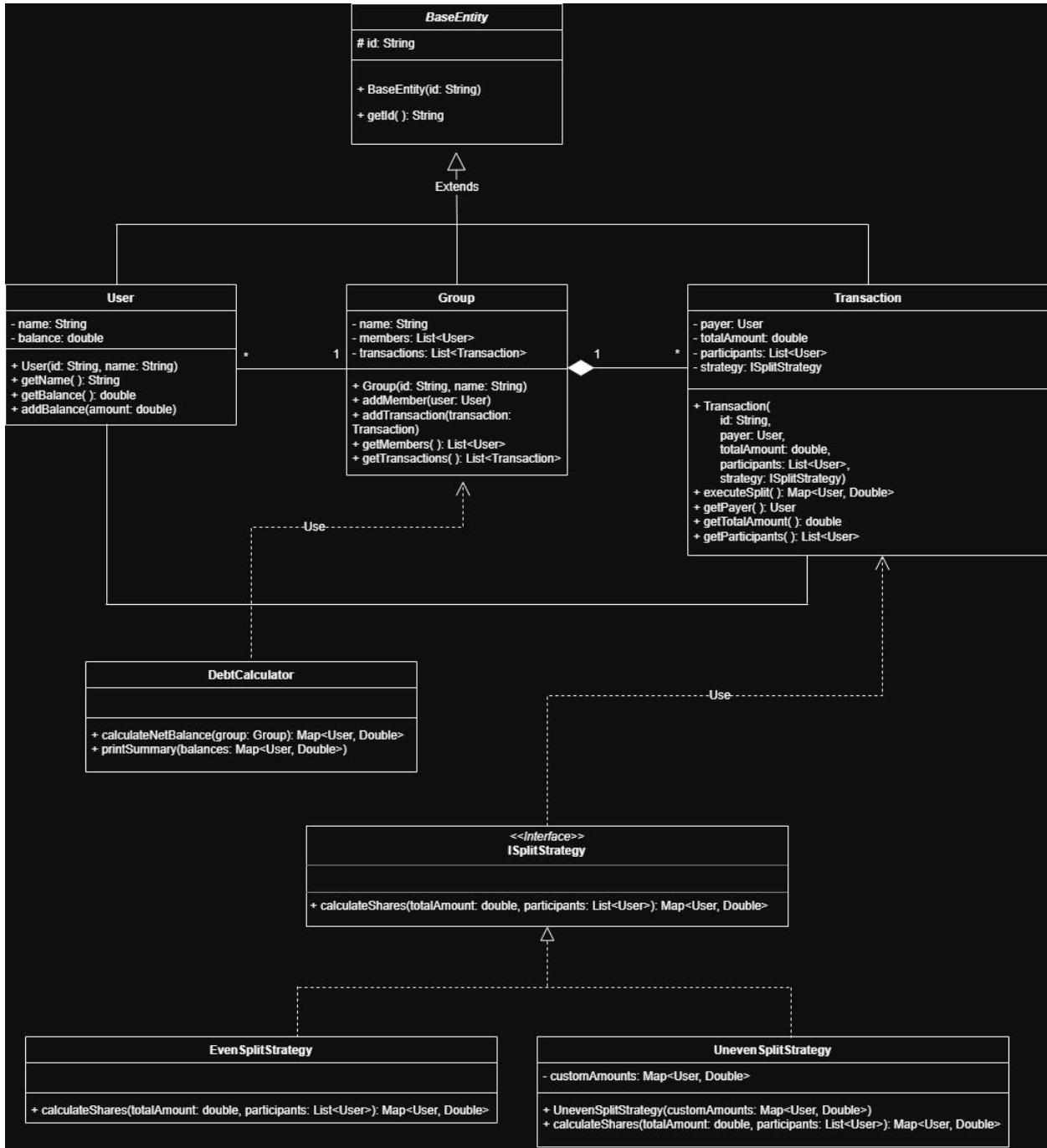
4. F4: Strategi Bagi Tidak Rata (*Uneven Split*)

Logika untuk membagi totalAmount berdasarkan nominal kustom yang diinput manual oleh pengguna untuk setiap partisipan.

5. F5: Kalkulator Hutang dan Rangkuman

Mesin yang menghitung neraca keuangan semua anggota dalam satu grup dan menyederhanakannya menjadi daftar "siapa-bayar-siapa".

3.2. Diagram Kelas (*Class Diagram*)



3.3. Deskripsi Detail Kelas

| Kelas | Deskripsi | Relasi/Konsep |
|---------------------|---|--|
| BaseEntity | Kelas abstrak berisi id dan getId() untuk konsistensi identitas tiap entitas. | <i>Abstract (Superclass)</i> |
| User | Mewakili anggota dalam grup. Menyimpan nama dan saldo hasil perhitungan <i>split</i> . | <i>Inheritance</i> dari BaseEntity |
| Group | Kumpulan anggota dan transaksi. Mengatur logika menambah anggota dan menghitung ringkasan hutang. | <i>Composition</i> (memiliki User dan Transaction) |
| Transaction | Mewakili satu pengeluaran. Menyimpan siapa yang membayar, total, dan siapa yang ikut membayar. | <i>Aggregation</i> dengan User |
| ISplitStrategy | Interface untuk mendefinisikan kontrak pembagian biaya. | <i>Interface</i> |
| EvenSplitStrategy | Implementasi pembagian rata. | Implementasi ISplitStrategy |
| UnevenSplitStrategy | Implementasi pembagian tidak rata (custom). | Implementasi ISplitStrategy |
| DebtCalculator | Utility class untuk menghitung dan menyederhanakan hutang antar anggota grup. | <i>Aggregation</i> ke Group |

IMPLEMENTASI FITUR DAN PEMBAGIAN TUGAS

1. Edelweiss Salsabila

Peran: Desain arsitektur sistem serta implementasi BaseEntity, User, dan manajemen saldo. Tanggung Jawab:

- Merancang struktur kelas utama pada aplikasi SplitBill
- Membuat class BaseEntity sebagai parent class
- Mengimplementasikan class User
- Mengelola fitur saldo (balance) pengguna

Hasil yang Sudah Dikerjakan

- Class BaseEntity berhasil dibuat untuk menyimpan atribut umum seperti id dan createdAt
- Class User berhasil diimplementasikan dengan atribut:
 - Name
 - Balance
- Method untuk:
 - Menambah saldo
 - Mengurangi saldo
 - Melihat saldo user
- Class User telah digunakan oleh class lain (Transaction dan Group)

2. Muthia Rezi Aisyah

Peran: Implementasi Transaction serta penerapan Strategy Pattern menggunakan ISplitStrategy.

Tanggung Jawab:

- Membuat class Transaction
- Menerapkan Strategy Pattern untuk metode pembagian tagihan
- Membuat interface ISplitStrategy

Hasil yang Sudah Dikerjakan:

- Class Transaction berhasil dibuat dengan atribut:
 - Payer
 - Amount
 - Participants
 - splitstrategy
- Interface ISplitStrategy berhasil dibuat sebagai kontrak pembagian
- Transaction dapat menggunakan strategy berbeda tanpa mengubah logic utama

3.

HASIL APLIKASI

5.1. Halaman Dashboard

Dashboard pada aplikasi SplitIt berfungsi sebagai halaman utama yang memberikan gambaran umum kondisi aplikasi secara cepat kepada pengguna.

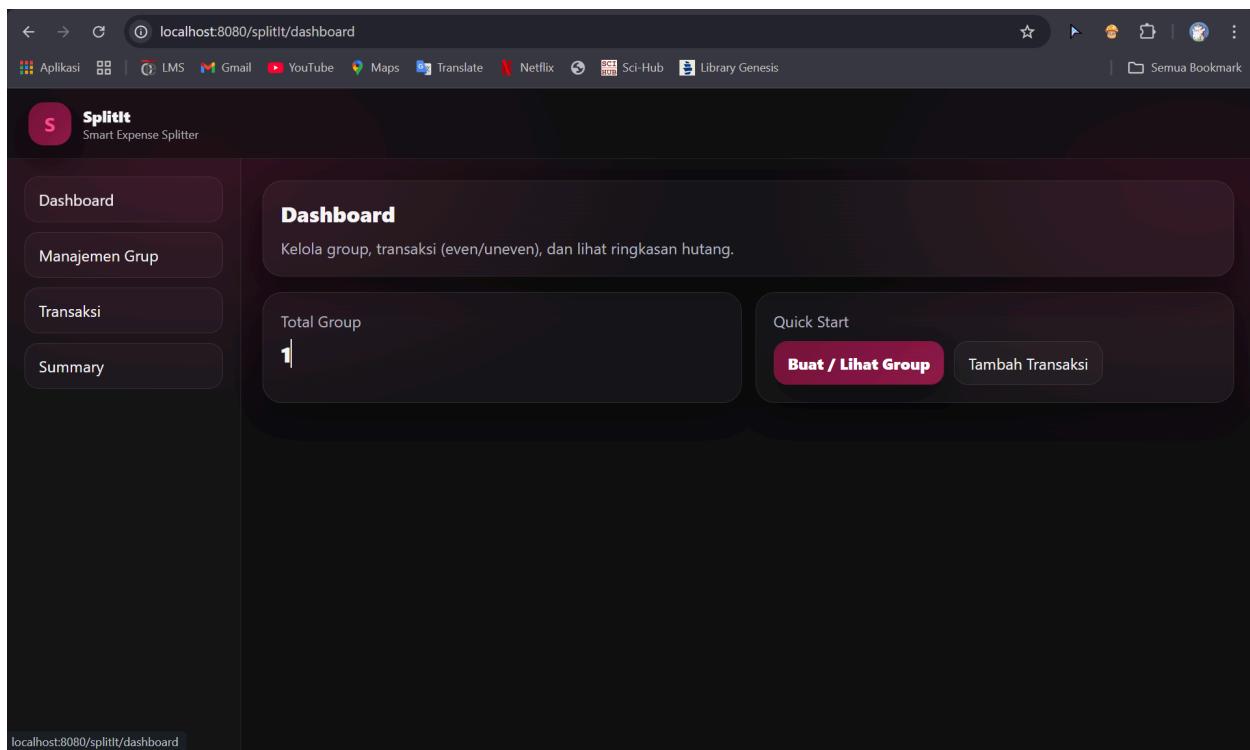
Di halaman ini, pengguna dapat:

- Melihat ringkasan data, seperti jumlah group yang sudah dibuat.
- Mengetahui status awal aplikasi, apakah sudah ada group atau belum.
- Mengakses fitur utama dengan cepat melalui tombol *Quick Start*.

Dashboard juga menjadi pusat navigasi, karena dari sini pengguna bisa langsung:

- Masuk ke Manajemen Grup untuk membuat atau mengatur group.
- Menuju halaman Transaksi untuk menambahkan transaksi baru.
- Melihat Summary untuk mengetahui hasil perhitungan hutang dan piutang.

Dengan adanya dashboard, pengguna tidak perlu langsung masuk ke halaman teknis, tapi bisa memahami kondisi aplikasi dan menentukan langkah selanjutnya dengan lebih mudah.



5.2. Halaman Buat/Lihat Group

Fungsi utama halaman ini adalah sebagai pusat pengelolaan grup dalam aplikasi SplitIt. Di halaman ini, pengguna bisa membuat grup baru dan melihat daftar grup yang sudah ada. Penjelasan fungsional:

Halaman *Manajemen Grup* berfungsi untuk:

1. Membuat grup baru
 - User mengisi ID Group dan Nama Group.
 - Saat tombol Simpan *Group* ditekan, data group disimpan ke database.
 - Group ini nantinya menjadi wadah transaksi dan anggota.
2. Melihat daftar grup yang sudah dibuat
 - Menampilkan tabel berisi ID Group dan Nama Group.
 - Setiap group punya aksi:
 - Detail + Anggota → masuk ke halaman detail group.
 - Ringkasan → melihat ringkasan transaksi dan hutang/piutang grup tersebut.

Konsep OOP yang dipakai:

- Group sebagai objek (model).
- Servlet sebagai penghubung antara UI dan database.
- DataSplit sebagai layer akses data (DAO).

The screenshot shows a web browser window with the URL `localhost:8080/splitt/groups`. The page has a dark theme. On the left, there's a sidebar with buttons for 'Dashboard', 'Manajemen Grup' (which is currently selected), 'Transaksi', and a 'Home' link. The main content area has two sections: 'Buat Group Baru' (Create New Group) and 'Daftar Group' (List Group).

In the 'Buat Group Baru' section, there are input fields for 'ID Group' (containing 'G2') and 'Nama Group' (with a placeholder 'Contoh: Sushi Tei'). A red 'Simpan Group' button is at the bottom right.

In the 'Daftar Group' section, there's a table with three columns: 'ID', 'NAMA', and 'AKSI'. It contains one row with data: 'G1', 'Sushi', and two buttons: 'Detail + Anggota' and 'Ringkasan'.

The browser's address bar shows the URL `localhost:8080/splitt/groups`, and the title bar says 'SplitIt Manajemen Grup'.

5.3. Halaman Detail Group + Anggota

Fungsi utama halaman ini adalah untuk mengelola anggota (User) di dalam sebuah group tertentu. Penjelasan fungsional:

Halaman *Detail Group* digunakan untuk:

1. Menampilkan informasi group yang dipilih
 - Nama group
 - ID group
 - Konteks ini penting supaya semua aksi (tambah user / transaksi) jelas masuk ke group mana.
2. Menambahkan anggota ke group
 - User mengisi:
 - ID User
 - Nama User
 - Data user disimpan ke tabel `users`, lalu dihubungkan ke group lewat `group_members`.
3. Menampilkan daftar anggota group
 - Menampilkan semua user yang tergabung dalam group.
 - Data ini dipakai lagi di halaman transaksi sebagai:
 - Payer
 - Participants

Kenapa halaman ini penting?

- Transaksi tidak bisa dibuat tanpa group dan anggota.
- Semua perhitungan split (Even / Uneven) bergantung pada anggota group ini.

The screenshot shows a web browser window with the URL `localhost:8080/splitit/group-detail?id=G1`. The page is titled "SplitIt Detail Group". On the left, there's a sidebar with buttons for "Kembali", "Tambah Transaksi", and "Home". The main content area is titled "Group: Sushi G1" and contains the instruction "Tambah anggota/pengguna (User) ke dalam group.". Below this is a form titled "Tambah Anggota" with fields for "ID User" and "Nama User". A "Tambah" button is located to the right of the form. At the bottom, there's a table titled "Anggota" with columns "ID" and "NAMA", listing users U1 (Muti), U2 (Alish), and U3 (Edel). A vertical scrollbar is visible on the right side of the table.

5.4. Halaman Tambah Transaksi Even (bagi rata)

Halaman Tambah Transaksi (Even Split) digunakan untuk mencatat transaksi pengeluaran dalam suatu group dengan metode pembagian bagi rata. Pada mode ini, total pengeluaran akan dibagi secara sama besar kepada seluruh peserta yang terlibat dalam transaksi.

Pengguna dapat menentukan:

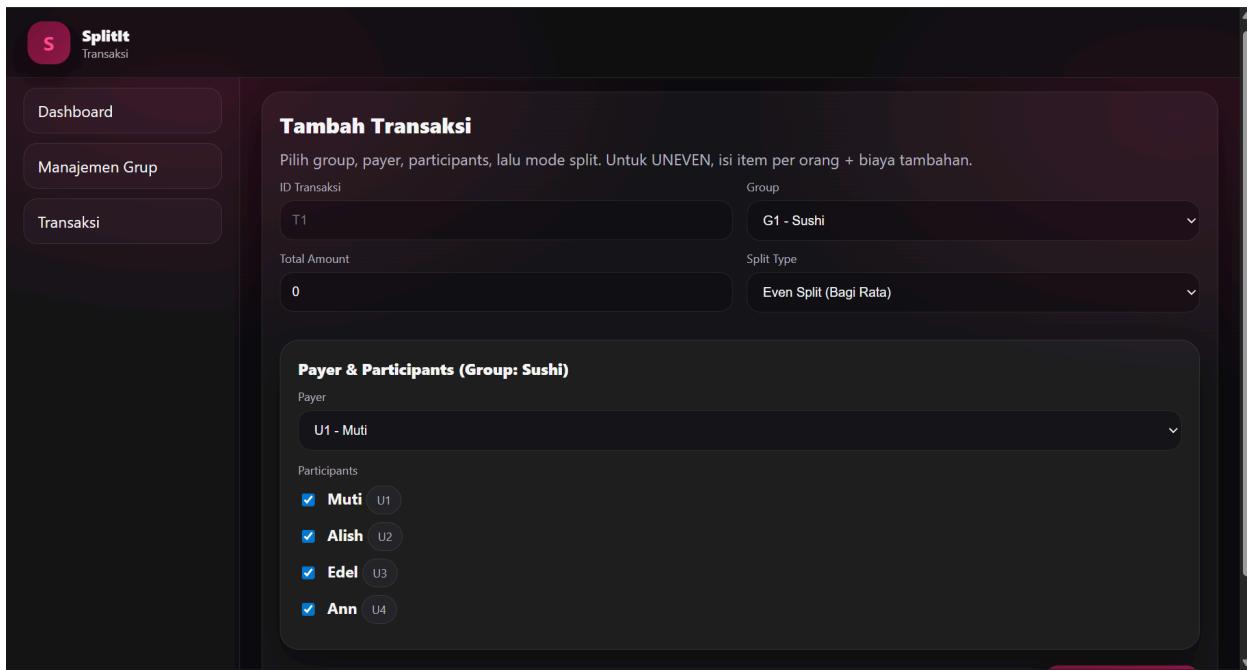
- ID transaksi sebagai identitas unik
- memilih group tempat transaksi dicatat
- memasukkan total nominal pengeluaran.

Selanjutnya, pengguna memilih:

- payer, yaitu anggota yang membayar transaksi terlebih dahulu,
- dan menentukan participants, yaitu anggota group yang ikut menanggung biaya tersebut.

Dalam metode *Even Split*, sistem secara otomatis menghitung jumlah yang harus dibayar oleh setiap participant dengan cara membagi total amount secara merata. Hasil perhitungan ini kemudian akan digunakan untuk menentukan posisi utang dan piutang masing-masing anggota pada halaman summary.

Halaman ini mempermudah pencatatan transaksi sederhana tanpa perlu memasukkan detail item per orang, sehingga cocok digunakan untuk pengeluaran bersama seperti makan bareng, iuran, atau biaya kolektif lainnya.



5.5. Halaman Tambah Transaksi Uneven (bagi rata)

Halaman Tambah Transaksi (Uneven) digunakan untuk mencatat transaksi dengan pembagian biaya tidak merata, yaitu setiap peserta dapat memiliki total pembayaran yang berbeda sesuai item yang dikonsumsi dan biaya tambahan yang dikenakan.

Pada halaman ini, pengguna terlebih dahulu mengisi ID Transaksi dan memilih Group yang aktif. Setelah itu, pengguna menentukan payer, yaitu orang yang membayar transaksi secara keseluruhan, serta memilih participants, yaitu anggota grup yang ikut terlibat dalam transaksi tersebut.

Mode Uneven Split (Per Item + Biaya Tambahan) memungkinkan pengguna memasukkan:

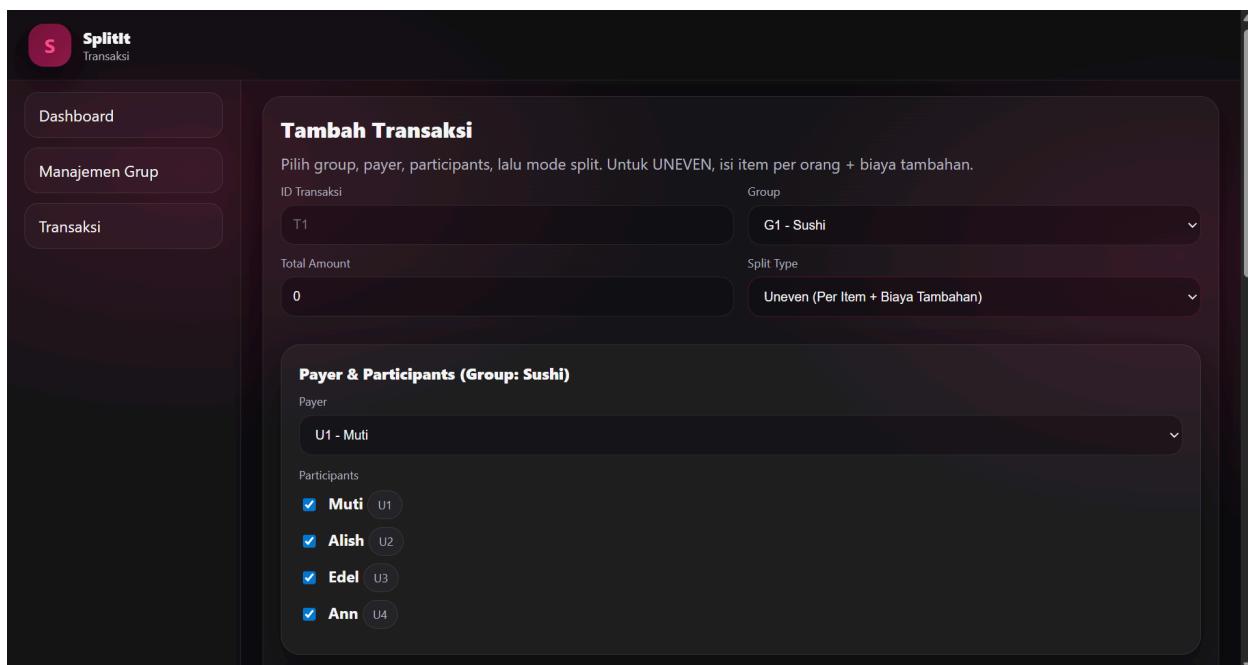
- Item per orang, yaitu daftar item yang dikonsumsi oleh masing-masing peserta beserta nominal harganya.
- Biaya tambahan, seperti pajak, service, atau admin, yang bisa berupa persentase maupun nominal tetap.

Sistem kemudian akan:

1. Menghitung subtotal item untuk setiap peserta berdasarkan item yang dimasukkan.
2. Menghitung total biaya tambahan dari pajak/service/admin.
3. Membagikan biaya tambahan ke setiap peserta sesuai aturan sistem (misalnya proporsional terhadap subtotal item).
4. Menampilkan preview pembagian, yang berisi:
 - Subtotal item per user
 - Share biaya tambahan per user
 - Total akhir yang harus dibayar masing-masing user

Preview ini bersifat real-time, sehingga setiap perubahan pada item atau biaya tambahan langsung memperbarui hasil perhitungan.

Setelah hasil pembagian dirasa sesuai, pengguna dapat menyimpan transaksi, dan data akan diproses oleh backend untuk disimpan ke database serta digunakan dalam perhitungan hutang dan piutang di halaman summary.



Uneven Builder

Isi item per orang, lalu biaya tambahan (pajak/service/admin bisa dimasukin di sini). Sistem akan bagi biaya sesuai masukan.

Item per Orang

Bisa tambah sebanyak yang diperlukan.

+ Tambah Item

| Nama | Item | Nominal |
|-----------|-----------|---------|
| U1 - Muti | Nama Item | 0 |
| | | Hapus |

Biaya Tambahan

+ Tambah Biaya

| Keterangan | Tipe | Nilai |
|-------------------------|---------|-------|
| Pajak / Service / Admin | Percent | 0 |
| | | Hapus |

Preview Pembagian (UNEVEN)

Subtotal: 0 Fee: 0 Total Akhir: 0

| USER | SUBTOTAL ITEM | SHARE BIAYA | TOTAL AKHIR |
|------|---------------|-------------|-------------|
| U1 | 0 | 0 | 0 |
| U2 | 0 | 0 | 0 |
| U3 | 0 | 0 | 0 |
| U4 | 0 | 0 | 0 |

Simpan Transaksi

5.6. Halaman Summary

Halaman Summary digunakan untuk menampilkan ringkasan akhir pembagian biaya dalam satu grup.

Di halaman ini, sistem menggabungkan seluruh transaksi yang pernah dicatat (baik Even Split maupun Uneven Split) lalu menghitung posisi keuangan masing-masing anggota grup.

Summary membantu pengguna:

- Mengetahui siapa yang punya piutang (dibayar orang lain)
 - Mengetahui siapa yang punya utang (harus membayar)
 - Melihat detail kontribusi setiap transaksi secara transparan
- a. Net Balance (Hutang / Piutang)

Bagian Net Balance menunjukkan akumulasi saldo bersih tiap anggota grup dari seluruh transaksi.

Cara kerja konsepnya:

- Nilai positif (Piutang) → anggota tersebut membayar lebih besar dibanding kewajibannya
- Nilai negatif (Utang) → anggota tersebut masih harus membayar ke grup

Contoh dari tampilan:

- Muti (U1) – 619.255,00 (Piutang)
Artinya Muti sering menjadi payer dan total uang yang ia keluarkan lebih besar dari yang seharusnya ia tanggung sendiri.
- Alish, Edel, Ann – nilai negatif (Utang)
Artinya mereka memiliki kewajiban membayar ke anggota yang memiliki piutang.

Bagian ini penting karena:

- Memberikan gambaran akhir keuangan grup
- Menjadi dasar penyelesaian pembayaran antar anggota

b. Detail Split per Transaksi

Bagian ini menampilkan rincian pembagian biaya untuk setiap transaksi, sehingga pengguna bisa melihat bagaimana angka Net Balance terbentuk.

- Transaksi T2 – Even Split

Total: 500.000,00

Payer: Muti (U1)

Karena menggunakan Even Split:

- Total biaya dibagi sama rata ke semua participant
- Jumlah peserta = 4 orang
- Masing-masing membayar: 125.000,00

Ini cocok untuk transaksi sederhana seperti: patungan makan, bensin, atau tiket dengan porsi yang sama.

- Transaksi T1 – Uneven Split

Total: 321.540,00

Payer: Muti (U1)

Karena menggunakan Uneven Split:

- Setiap orang membayar sesuai item yang dikonsumsi
- Ditambah biaya tambahan seperti pajak / service
- Pembagian tidak sama, tergantung input item dan nominal

Contoh:

- Muti membayar item tertentu + share biaya tambahan → 77.285,00
- Alish membayar item berbeda → 98.385,00
- Dan seterusnya

Transaksi ini mencerminkan kondisi dunia nyata, misalnya: makan bareng tapi pesan menu berbeda-beda.

The screenshot shows the SplitIt mobile application interface. At the top left is the logo 'SplitIt' with the word 'Summary' below it. On the left sidebar, there are three buttons: 'Home', 'Manajemen Grup' (Group Management), and 'Tambah Transaksi' (Add Transaction). The main content area has a title 'Ringkasan Biaya - Sushi' and a note 'Positif = piutang, negatif = utang.' Below this is a section titled 'Net Balance (Hutang/Piutang)' with a table:

| USER | BALANCE | STATUS |
|------------|-------------|---------|
| Muti (U1) | 619.255,00 | Piutang |
| Alish (U2) | -223.385,00 | Utang |
| Edel (U3) | -194.385,00 | Utang |
| Ann (U4) | -201.485,00 | Utang |

This screenshot shows the 'Detail Split per Transaksi' section of the SplitIt app. It displays two transaction details:

Transaksi: T2 • Total: 500.000,00
Payer: Muti (U1)

| PARTICIPANT | SHARE (HARUS BAYAR) |
|-------------|---------------------|
| Muti (U1) | 125.000,00 |
| Alish (U2) | 125.000,00 |
| Edel (U3) | 125.000,00 |
| Ann (U4) | 125.000,00 |

Transaksi: T1 • Total: 321.540,00
Payer: Muti (U1)

| PARTICIPANT | SHARE (HARUS BAYAR) |
|-------------|---------------------|
| Muti (U1) | 77.285,00 |
| Alish (U2) | 98.385,00 |

5.7. Demo Website

Berikut adalah link video presentasi kelompok kami :

<https://youtu.be/mzvX5mVkewo?si=bNHi0VFRoBHSMe>

5.8. Pembagian tugas

Link pembagian tugas detailnya dapat diakses disini :

https://drive.google.com/file/d/1LWaXrvxxazb3ZWfp0rBbxzf_KaEMK7Us/view?usp=sharing

PENUTUP

6.1. Kesimpulan

Aplikasi *SplitIt: Smart Expense Splitter* berhasil dikembangkan sebagai solusi untuk membantu pengelolaan dan pembagian pengeluaran dalam suatu kelompok secara adil dan transparan. Aplikasi ini mampu mencatat transaksi, mengelola grup dan anggota, serta menghitung pembagian biaya baik secara rata (*even*) maupun tidak rata (*uneven*), sehingga dapat mengurangi kesalahan perhitungan manual dan potensi konflik antar anggota kelompok.

Dari sisi implementasi teknis, aplikasi SplitIt telah menerapkan konsep Pemrograman Berorientasi Objek (PBO) dengan baik, meliputi *encapsulation*, *abstraction*, *inheritance*, dan *polymorphism*. Penerapan *Strategy Design Pattern* melalui *interface ISplitStrategy* memungkinkan sistem menangani berbagai metode pembagian biaya secara fleksibel tanpa mengubah struktur utama program. Penggunaan *abstract class BaseEntity* juga membantu menjaga konsistensi dan kerapihan struktur data dalam aplikasi.

Secara keseluruhan, proyek ini tidak hanya memenuhi kebutuhan fungsional sebagai aplikasi pembagi tagihan, tetapi juga berhasil menjadi sarana penerapan konsep PBO dalam pengembangan perangkat lunak nyata. Aplikasi SplitIt menunjukkan bahwa perancangan arsitektur yang baik dan modular dapat menghasilkan sistem yang mudah dikembangkan dan dipelihara di masa mendatang.