Laporan UTS Pengembangan Aplikasi Perusahaan

Dosen Pengampu:

Suryo Bramasto, S.T., M.T.

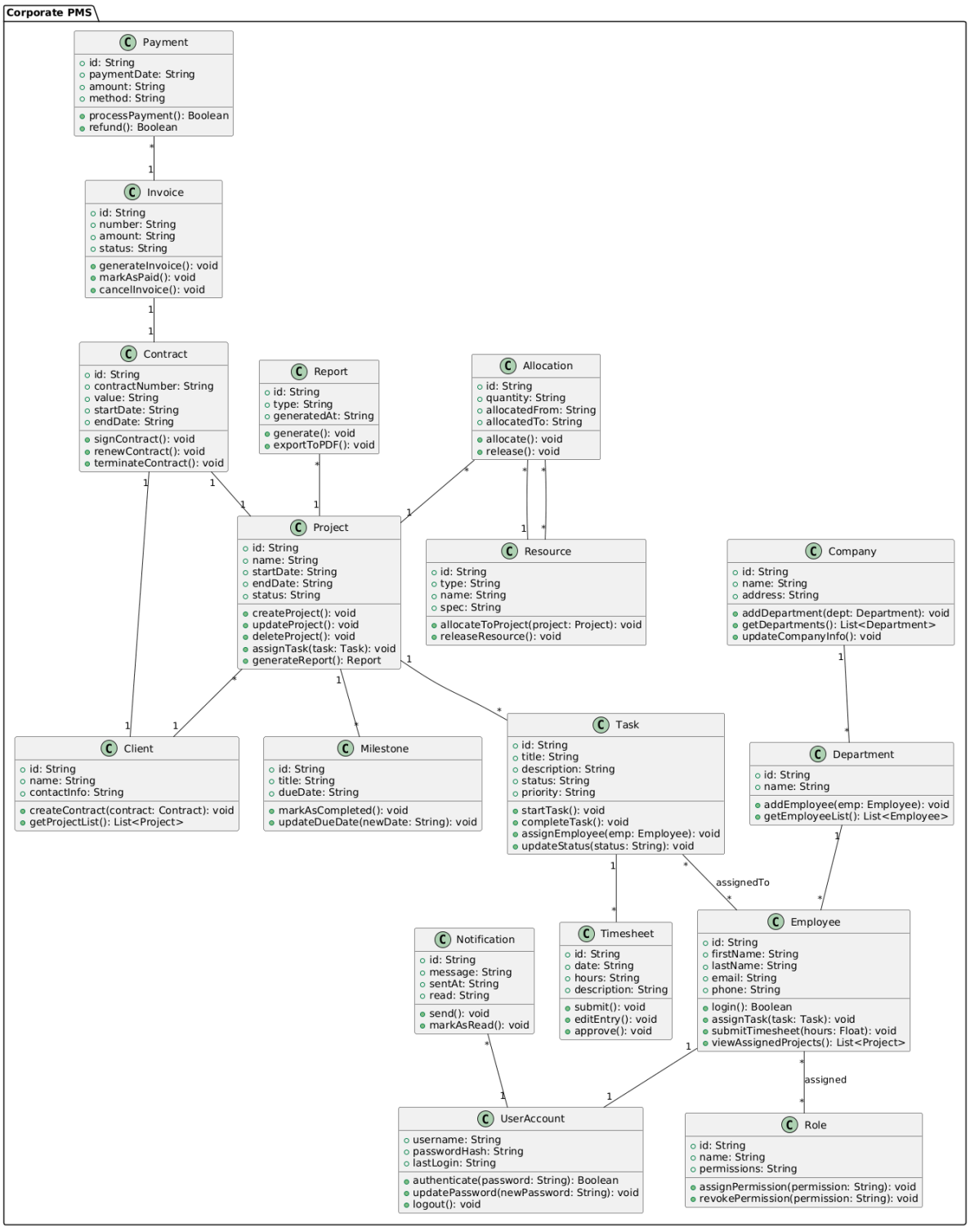
Disusun Oleh:

Jonathan Natannael Zefanya (1152200024)

Ganjil 2025/2026

1. Forward engineering – Case Corporate Project & Resource Management System

Case Tools: PlanUML



Forward engineering:

https://github.com/JonathanZefanya/UTS-PAP/tree/main/No1/generated_java

2. Reverse engineering - Case Sales Order System

Reverse Engineering:

https://github.com/JonathanZefanya/UTS-PAP/blob/main/No2/enterprise/Apps.zip

Case Tools: **Visual Paradigm Community Edition**

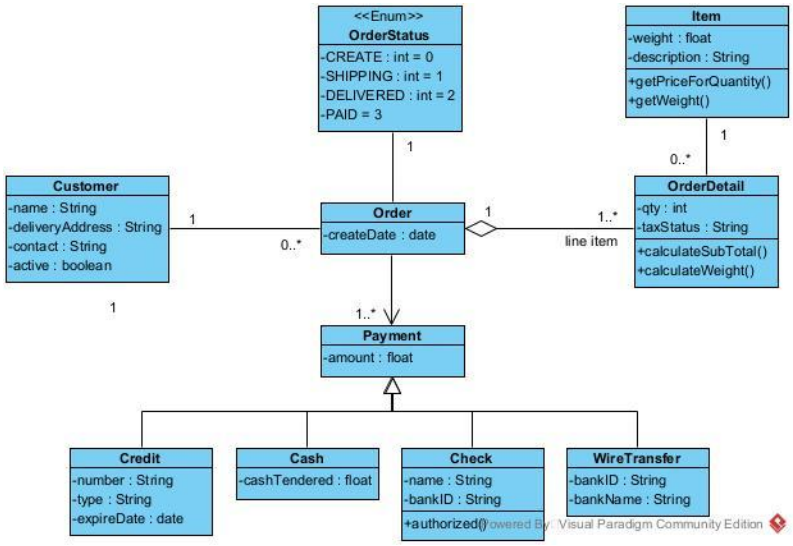


Diagram ini menjelaskan struktur dasar dari sebuah sistem pemesanan, seperti yang ditemukan di toko online atau aplikasi kasir. Semua berawal dari Customer (Pelanggan), yang memiliki data seperti nama dan alamat. Setiap Customer dapat melakukan nol atau lebih Order (Pesanan). Order adalah kelas pusat yang mencatat kapan pesanan dibuat dan menghubungkan semua bagian lain dari transaksi.

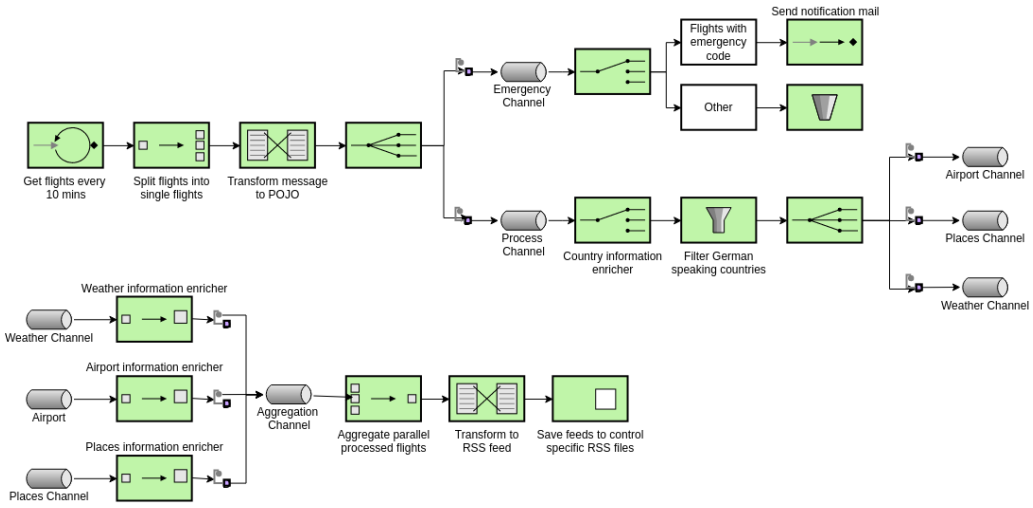
Setiap Order pasti memiliki satu atau lebih OrderDetail (Detail Pesanan). Anggap saja OrderDetail ini sebagai satu baris barang di struk belanja Anda, yang mencatat berapa banyak (qty) barang yang dibeli. Relasi ini adalah "Komposisi", yang artinya OrderDetail tidak bisa ada tanpa sebuah Order. Selanjutnya, setiap OrderDetail mengacu pada satu Item (Barang), yang merupakan produk sebenarnya dengan deskripsi dan beratnya.

Sebuah Order juga memiliki dua koneksi penting lainnya. Pertama, setiap Order memiliki satu OrderStatus (Status Pesanan) pada satu waktu, yang merupakan label status yang sudah

ditentukan (seperti CREATE, SHIPPING, DELIVERED, atau PAID). Kedua, sebuah Order harus terhubung ke satu atau lebih Payment (Pembayaran). Ini berarti satu pesanan bisa dibayar dengan beberapa cara, misalnya sebagian tunai dan sebagian dengan kartu kredit.

Terakhir, sistem pembayaran ini sangat fleksibel. Payment adalah sebuah konsep umum (kelas superclass). Ada beberapa jenis pembayaran spesifik yang merupakan turunan darinya, seperti Credit (Kartu Kredit), Cash (Tunai), Check (Cek), dan WireTransfer. Ini adalah contoh "Pewarisan" (Inheritance), yang berarti semua jenis pembayaran tersebut adalah "sebuah" Payment, tetapi masing-masing memiliki atribut uniknya sendiri (seperti Credit yang memiliki nomor kartu dan tanggal kedaluwarsa).

3. Case penerapan Sistem Pemrosesan Data Penerbangan



Proses utama dimulai dengan pengambilan data penerbangan secara periodik. Sebuah komponen Polling Consumer (Konsumen Pengambil) diatur untuk "Get flights every 10 mins" (Mengambil data penerbangan setiap 10 menit). Data ini kemungkinan besar datang dalam bentuk koleksi atau batch (misalnya, file XML atau JSON yang berisi daftar banyak penerbangan).

Setelah data diterima, data tersebut langsung dikirim ke Splitter (Pemecah). Komponen ini mengambil pesan batch tersebut dan memecahnya menjadi pesan-pesan individual. Dalam konteks ini, "Split flights into single flights" (Memecah penerbangan menjadi penerbangan tunggal) berarti jika satu pesan berisi 100 data penerbangan, Splitter akan menghasilkan 100 pesan terpisah, di mana masing-masing pesan hanya berisi data untuk satu penerbangan.

Setiap pesan penerbangan tunggal kemudian melewati Transformer (Pengubah). Tujuannya adalah untuk "Transform message to POJO" (Mengubah pesan menjadi POJO). POJO adalah singkatan dari Plain Old Java Object. Ini adalah pola desain di mana data (yang mungkin awalnya dalam format XML, JSON, atau format teks lainnya) dikonversi menjadi objek Java standar. Ini sangat menyederhanakan pemrosesan di langkah-langkah selanjutnya karena logika bisnis dapat berinteraksi dengan objek yang terstruktur dengan baik, alih-alih mem-parsing teks mentah.

Setelah diubah menjadi POJO, pesan tersebut dikirim ke dua saluran (Message Channel) secara bersamaan: Emergency Channel dan Process Channel. Ini adalah awal dari Content-Based Router (Perutean Berbasis Konten).

Di Emergency Channel, sebuah Filter (Penyaring) segera memeriksa konten pesan. Jika pesan tersebut berisi "emergency code" (kode darurat), pesan tersebut akan lolos filter dan langsung memicu Service Activator (Aktivator Layanan), yaitu "Send notification mail" (Kirim email notifikasi). Ini adalah contoh bagus dari pemrosesan prioritas tinggi dan peringatan real-time. Pesan-pesan lain yang tidak memiliki kode darurat akan dibuang atau diabaikan oleh alur ini ("Other").

Sementara itu, salinan pesan yang sama dikirim ke Process Channel untuk pemrosesan normal. Di sini, pesan melewati Content Enricher (Pengaya Konten) yang menambahkan "Country information" (informasi negara) ke dalam pesan. Selanjutnya, Filter lain diterapkan untuk "Filter German speaking countries" (Menyaring negara-Bicara Jerman). Hanya pesan-pesan yang lolos filter inilah yang kemudian dikirim ke berbagai saluran tujuan seperti Airport Channel, Places Channel, dan Weather Channel untuk konsumsi oleh sistem lain.

Bagian bawah diagram menunjukkan alur yang berbeda namun saling terkait. Tiga saluran sumber data (Weather Channel, Airport, Places Channel) masing-masing memiliki komponen Content Enricher-nya sendiri. Ini menunjukkan bahwa ada proses paralel yang sedang berlangsung.

Semua data ini data penerbangan asli yang telah diproses dan data tambahan yang telah diperkaya berkumpul di Aggregation Channel. Komponen Aggregator (Pengumpul) kemudian bertugas "Aggregate parallel processed flights" (Mengumpulkan penerbangan

yang diproses paralel). Aggregator adalah pola yang cerdas; ia mengumpulkan pesan-pesan terkait (misalnya, data penerbangan ID ABC, data cuaca untuk ID ABC, data bandara untuk ID ABC) dan menunggu sampai set data lengkap terkumpul. Setelah lengkap, ia menggabungkan semua pesan ini menjadi satu pesan komprehensif tunggal.

Pesan gabungan yang komprehensif ini yang sekarang berisi semua informasi tentang penerbangan, cuaca, bandara, dan tempat kemudian dikirim ke Transformer terakhir. Kali ini, tugasnya adalah "Transform to RSS feed" (Mengubah ke format RSS). Objek data yang kaya tersebut diubah formatnya menjadi XML yang sesuai dengan standar RSS.

Akhirnya, Service Activator terakhir "Save feeds to control specific RSS files" (Simpan feed untuk mengontrol file RSS tertentu) mengambil data RSS ini dan menuliskannya ke sistem file. Hasilnya adalah file RSS yang selalu diperbarui yang dapat dikonsumsi oleh aplikasi lain, portal web, atau pengguna akhir.