

Latihan UAS

Nama : Steven Chowina

NIM : 2702295373

Kelas : LE01

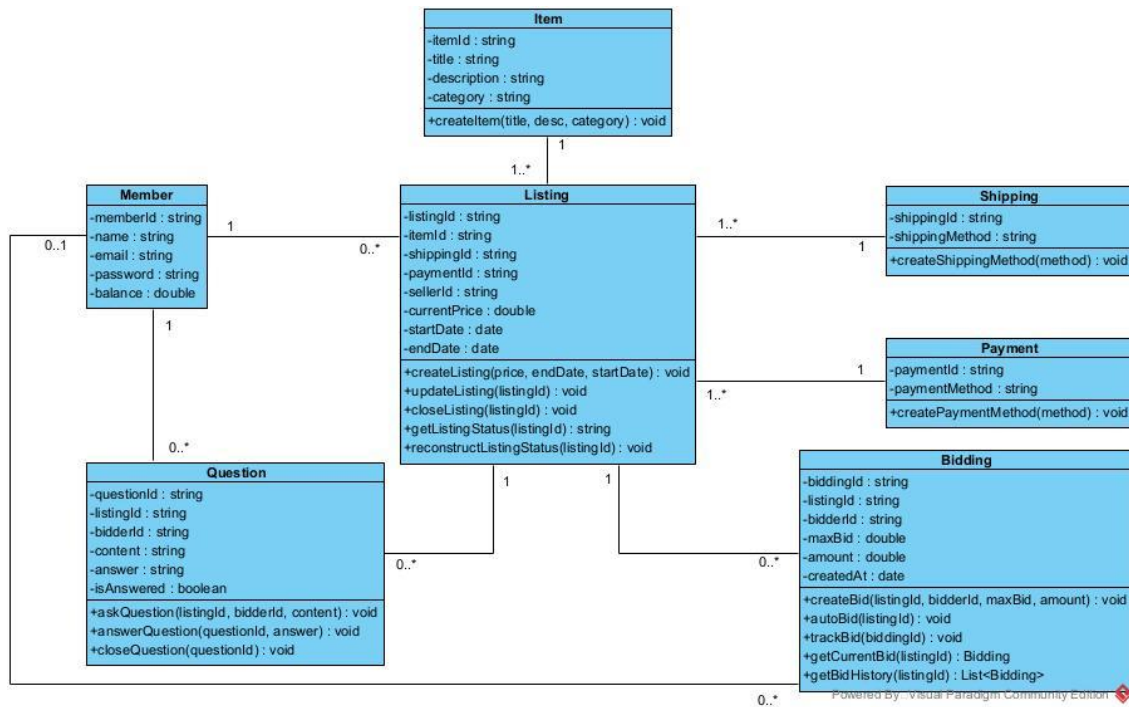
Tema Aplikasi: E-Bidder – Listing Management System untuk Lelang Online

Requirement Aplikasi:

1. Pengguna sistem adalah member yang bisa menjual atau membeli barang. Setiap member memiliki akun dan dapat:
 - Membuat dan mengelola listing (daftar barang yang akan dilelang)
 - Menawar pada listing milik pengguna lain
2. Listing adalah inti sistem dan mencakup:
 - Barang yang dijual beserta informasi (judul, deskripsi, kategori)
 - Setidaknya satu metode pembayaran dan satu metode pengiriman
 - Informasi auction (tanggal mulai, tanggal selesai)
 - Hubungan ke seller dan informasi penawaran saat ini
3. Penawaran (bidding):
 - Setiap member dapat mengajukan tawaran maksimum (max bid)
 - Sistem akan secara otomatis menawar naik sesuai kebutuhan (auto-bid)
4. Setiap member dapat mengajukan pertanyaan pada sebuah listing, dan seller dapat memberikan jawaban serta menutup pertanyaan tersebut.
5. Semua aktivitas penawaran dicatat agar bisa dianalisis ulang atau diaudit.
6. Diperlukan fitur untuk memungkinkan sistem merekonstruksi status sebuah listing berdasarkan histori penawaran.

Pertanyaan:

1. Buatlah **class diagram** sederhana untuk menggambarkan struktur aplikasi ini! Sertakan atribut dan metode utama.



2. Rancang dan jelaskan konsep-konsep **Tactical Pattern** dari Domain Driven Design dalam rancangan sistem Aplikasi E-Bidder:

1. Entity, Value Object (jika ada), dan Domain Service (jika ada)

a. Entities:

Entity adalah object yg memiliki identitas dan lifecycle

- Member : memiliki identitas yaitu memberId dan juga dapat melakukan bidding dan juga listing
- Listing : Objek utama sistem yg mengatur sistem dari pevelangan dan juga memiliki identitas dan lifecycle nya sendiri
- Bidding : Objek yang memiliki identitas sendiri yg menunjukkan penawaran spesifik
- Item : Objek yang dilelang memiliki identitasnya sendiri
- Shipping : Shipping merepresentasikan cara pengiriman
- Payment : Payment merepresentasikan cara pembayaran
- Question : Entitas sebagai perantara tanya jawab antara seller dan juga buyer sehingga memiliki identitas dan lifecycle nya sendiri

b. Value Object:

Value Object mendeskripsikan hal yg dimiliki oleh sebuah entity sehingga VO tidak mempunyai identitasnya sendiri karena tidak memiliki makna apapun.

- Balance (money) : Uang yg dimiliki member. Tidak mempunyai ID khusus.
- Start Date & End Date : hanya merepresentasikan rentang waktu lelang. Tidak unik, tapi penting untuk durasi validitas.

c. **Domain Service:**

Domain Service adalah behavior yang exist di suatu permasalahan domain (mewakili business logic).

- **AutoBid** : Menyediakan logika untuk menaikkan bidding (penawaran) secara otomatis sampai batas maxBid
- **TrackBid** : logika untuk menganalisa semua aktivitas bidding sehingga bisa dianalisis ulang atau diaudit
- **ReconstructListingStatus** : Logika untuk merekonstruksi status sebuah listing berdasarkan bidding history

2. Aggregate

Aggregate adalah Kumpulan entity dan value object yg dikelola sebagai satu unit kelompok, aggregate juga harus menjaga domain invariants (aturan bisnis) tetap valid.

a. **Root:**

- **Listing** : sebagai inti dari semua operasi terhadap sistem E-bidder ini. Sehingga semua perubahan pada Bidding ataupun Question harus melewati listing.

b. **Child:**

- **Bidding** : ditambahkan lewat method createBid() melalui Listing
- **Question** : ditambahkan atau dijawab lewat Listing

c. **Invariant:**

- Tidak boleh ada bid yang lebih rendah dari current bid.
- Tidak boleh ada question yang dijawab dua kali.
- Listing hanya bisa di-close setelah endDate.

3. Factory

Factory digunakan untuk membuat sebuah entitas kompleks yg mempunyai banyak depedensi

- a. **ListingFactory** untuk membuat listing baru (createListing()) dengan semua depedensi nya item, payment, shipping, dll
- b. **BiddingFactory** untuk membuat sebuah bid baru
- c. **QuestionFactory** untuk membuat sebuah pertanyaan baru

4. Repository

Repository digunakan untuk menjaga mekanisme storing/fetching data dari aggregate ke database sehingga menjaga aggregate agar tetap bersih dari technical database dan lebih focus ke bisnis.

- a. **ListingRepository** bertugas untuk menyimpan dan mengambil data Listing dari database
- b. **BiddingRepository** bertugas untuk menyimpan dan mengambil data Bidding dari database

- c. QuestionRepository bertugas untuk menyimpan dan mengambil data Question dari database

5. Domain Event

Digunakan untuk mencatat peristiwa penting yang terjadi didalam domain.

- BidPlacedEvent : peristiwa domain ketika ada penawaran baru dari buyer
- ListingClosedEvent : peristiwa domain ketika sebuah listing ditutup
- QuestionClosedEvent : peristiwa domain ketika seller telah menjawab question dan menutup nya

6. Event Sourcing

Digunakan untuk merekonstruksi state sistem dari history event

- Setiap kali BidPlacedEvent maka untuk tahu status dari sebuah listing saat ini (buyer dgn bid tertinggi) maka kita cukup melihat dari history event bid itu.