

Tugas Personal ke-1

Week 2

Nama: Frans Sebastian

NIM: 2502121162

Soal Teori

1. (LO1; 10%) Jelaskan apa itu Domain Driven Design (DDD) dan mengapa ditujukan untuk aplikasi yang kompleks? Apa yang dimaksud aplikasi yang kompleks? Berikan contoh!

Jawab:

Domain Driven Design (DDD) adalah pendekatan pengembangan perangkat lunak yang berfokus pada model domain yang kompleks. DDD dirancang untuk mengatasi kompleksitas dalam pengembangan perangkat lunak dengan menyambungkan implementasi ke model yang berkembang. Ini mencakup penekanan pada domain utama dan logika domain, pembentukan desain yang kompleks berdasarkan model domain, dan kolaborasi kreatif antara ahli teknis dan domain untuk memperbaiki secara iteratif model konseptual yang menangani masalah domain tertentu. Aplikasi yang kompleks adalah sistem yang memiliki banyak aturan bisnis yang kompleks, perubahan yang cepat, dan kebutuhan untuk fleksibilitas dan skalabilitas.

Contoh aplikasi kompleks adalah:

- Sistem manajemen proyek yang harus menangani berbagai aspek seperti penjadwalan, pengelolaan sumber daya, dan pelacakan kemajuan.
 - Sistem perbankan yang melibatkan banyak proses bisnis, seperti manajemen akun, transfer dana, dan analisis risiko.
2. (LO1; 10%) Jelaskan apa yang dimaksud dengan Domain, Subdomain dan jenis – jenis Subdomain. Berikan juga contohnya.

Jawab:

Domain: Domain adalah area atau bidang yang mencakup sejumlah besar konsep, aturan, dan proses bisnis tertentu. Contohnya adalah domain perbankan atau domain e-commerce.

Subdomain: Subdomain adalah bagian dari domain yang lebih spesifik. Subdomain dapat dibagi lagi menjadi beberapa jenis, seperti Core Domain, Supporting Domain, dan Generic Domain.

- **Core Domain:** Merupakan inti dari bisnis yang memberikan nilai utama. Contohnya adalah sistem pembayaran dalam domain e-commerce.
- **Supporting Domain:** Merupakan domain yang mendukung operasi dari core domain. Contohnya adalah manajemen pengguna dalam domain e-commerce.
- **Generic Domain:** Merupakan domain yang bersifat umum dan dapat digunakan di berbagai domain. Contohnya adalah manajemen autentikasi pengguna.

3. (LO1; 10%) Jelaskan yang dimaksud dengan Bounded Context dan Ubiquitous Language. Apa perbedaan Bounded Context dan Subdomain? Jelaskan pendapat anda.

Jawab:

- **Bounded Context:** Bounded Context adalah batasan yang jelas dan spesifik terhadap model domain dalam suatu aplikasi. Setiap Bounded Context memiliki batasan yang terdefinisi dengan baik, termasuk terminologi, aturan bisnis, dan model domainnya sendiri.
- **Ubiquitous Language:** Ubiquitous Language adalah bahasa yang digunakan secara konsisten dan umum oleh semua pemangku kepentingan, termasuk pengembang dan ahli domain. Bahasa ini mencerminkan model domain dan digunakan untuk berkomunikasi antara tim pengembang dan pemangku kepentingan.

Perbedaan antara Bounded Context dan Subdomain terletak pada tingkat abstraksi dan cakupan. Bounded Context adalah batasan yang spesifik terhadap model domain dalam suatu aplikasi, sementara Subdomain adalah bagian dari domain yang lebih spesifik. Bounded Context dapat mencakup satu atau beberapa Subdomain.

4. (LO1; 10%) Jelaskan, apakah Bounded Context dapat saling berkomunikasi? Bagaimana implementasinya?

Jawab:

Bounded Context dapat berkomunikasi satu sama lain melalui integrasi antar-bounded context. Implementasinya dapat dilakukan dengan menggunakan berbagai metode, seperti REST API, Event-Driven Architecture (EDA), Messaging Systems, atau Remote Procedure Calls (RPC). Pendekatan ini memungkinkan Bounded Context yang terpisah untuk berkomunikasi dan bertukar informasi, memungkinkan kesatuan yang lebih besar dalam aplikasi yang kompleks serta memungkinkan komunikasi antara konteks yang berbeda.

5. (LO2; 10%) Jelaskan apa itu Layered Architecture. Jelaskan pula perbedaan Application Logic dan Domain Logic.

Jawab:

Layered Architecture adalah pendekatan dalam desain perangkat lunak di mana sistem dibagi menjadi beberapa lapisan (layers), di mana setiap lapisan memiliki tanggung jawabnya sendiri. Biasanya, lapisan-lapisan tersebut mencakup Presentation Layer, Application Layer, Domain Layer, dan Infrastructure Layer.

- **Application Logic:** Merupakan logika yang menangani alur kontrol dan koordinasi proses aplikasi. Ini berfungsi sebagai penghubung antara Presentation Layer dan Domain Layer.
- **Domain Logic:** Merupakan logika yang mencerminkan aturan bisnis dan pemrosesan domain yang spesifik. Ini berada di dalam Domain Layer dan bertanggung jawab atas pemodelan domain bisnis serta menjamin konsistensi dan validitas data.

Perbedaan utama antara keduanya adalah bahwa Application Logic berfokus pada bagaimana aplikasi berinteraksi dengan pengguna dan sistem lain, sementara Domain Logic berfokus pada aturan dan proses bisnis yang ada dalam domain.

Referensi: <https://github.com/heynickc/awesome-ddd>

Soal Kasus

Pilihlah satu dari tema aplikasi, yaitu e-commerce (contoh Zalora, Bhinneka, dsb), transportasi online (contoh Gojek, Grab, dsb) dan tutorial online (contoh Udemy, Ruang Guru, dsb). Sebutkan nama aplikasi yang anda pilih dan lakukan analisa sebagai berikut : Saya memilih Gojek sebagai aplikasi yang akan saya analisa.

- a. (LO1; 15%) Jelaskan kompleksitas dalam aplikasi yang anda pilih.

Jawab:

Gojek adalah aplikasi super app yang menyediakan berbagai layanan, termasuk transportasi, pengiriman makanan, belanja, pembayaran, dan layanan lainnya. Kompleksitas dalam aplikasi ini dapat dijelaskan sebagai berikut:

- **Multi-Layanan:** Gojek menyediakan berbagai layanan, termasuk transportasi, pemesanan makanan, pembayaran, dan lainnya. Setiap layanan memiliki kompleksitasnya sendiri, seperti sistem pencocokan pengemudi dan penumpang, sistem pembayaran, dan sistem manajemen inventori di bawah satu platform, yang memerlukan pengelolaan infrastruktur yang kompleks dan integrasi yang kuat antara layanan-layanan tersebut.
- **Logistik dan Rute:** Layanan transportasi dan pengiriman memerlukan manajemen logistik yang kompleks, termasuk pemetaan rute, estimasi waktu tiba, dan penjadwalan yang efisien.
- **Keamanan dan Privasi:** Gojek mengumpulkan data sensitif pengguna seperti lokasi, informasi pembayaran, dan data pribadi lainnya, yang memerlukan sistem keamanan yang kuat untuk melindungi informasi tersebut.
- **Ketersediaan dan Kinerja:** Sebagai aplikasi yang sangat bergantung pada layanan real-time, Gojek harus menjaga ketersediaan sistem dan kinerja aplikasi agar tetap stabil bahkan saat menghadapi lonjakan penggunaan yang besar.
- **Skala operasi:** Gojek beroperasi di beberapa negara dan memiliki jutaan pengguna dan mitra driver. Hal ini meningkatkan kompleksitas dalam hal skalabilitas dan manajemen data.

- b. (LO1; 15%) Jelaskan subdomain dan jenis subdomain dalam aplikasi tersebut.

Jawab:

Subdomain dalam Gojek mencakup berbagai layanan yang disediakan oleh aplikasi tersebut. Beberapa contoh subdomain beserta jenisnya di Gojek:

1. Transportasi Online (GoRide, GoCar):

- **Pemesanan Transportasi (Core Subdomain):** Mengelola proses pemesanan transportasi seperti mobil, motor, dan sepeda.
- **Pemetaan dan Rute (Core Subdomain):** Memiliki informasi tentang pemetaan, rute, estimasi waktu tiba, dan navigasi.
- **Pembayaran Transportasi (Supporting Subdomain):** Mengelola proses pembayaran untuk layanan transportasi yang digunakan oleh pengguna.
- **Laporan Transaksi Transportasi (Supporting Subdomain):** Memproses dan menghasilkan laporan transaksi yang terkait dengan layanan transportasi.

2. Pengiriman Makanan (GoFood):

- **Pemesanan Makanan (Core Subdomain):** Mengelola proses pemesanan makanan dari restoran-restoran yang bekerja sama.
- **Manajemen Restoran (Core Subdomain):** Menyediakan alat dan fitur untuk restoran untuk mengelola pesanan, menu, dan inventaris.
- **Pembayaran Makanan (Supporting Subdomain):** Mengelola proses pembayaran untuk layanan pengiriman makanan.
- **Laporan Transaksi Makanan (Supporting Subdomain):** Memproses dan menghasilkan laporan transaksi yang terkait dengan layanan pengiriman makanan.

3. Pembayaran dan Keuangan (Gopay):

- **Manajemen Pembayaran (Core Subdomain):** Mengelola proses pembayaran untuk layanan-layanan yang ditawarkan oleh Gojek.

- **Integrasi dengan Lembaga Keuangan (Core Subdomain):** Mengelola integrasi dengan lembaga keuangan untuk menangani transaksi keuangan.
- **Laporan Keuangan (Supporting Subdomain):** Memproses dan menghasilkan laporan keuangan untuk kebutuhan analisis dan pelaporan.
- **Penanganan Aduan dan Keluhan (Supporting Subdomain):** Mengelola proses penanganan aduan dan keluhan terkait dengan pembayaran dan keuangan.

4. Manajemen Pengguna (User Profile):

- **Manajemen Akun Pengguna (Core Subdomain):** Mengelola informasi pengguna, otentikasi, dan otorisasi akses.
- **Manajemen Data Pribadi (Core Subdomain):** Menyediakan fasilitas untuk mengelola data pribadi pengguna.
- **Manajemen Riwayat Transaksi Pengguna (Supporting Subdomain):** Mengelola riwayat transaksi pengguna untuk keperluan audit dan historisasi.
- **Pusat Bantuan dan Dukungan Pengguna (Supporting Subdomain):** Menyediakan layanan dukungan dan bantuan kepada pengguna.

5. GoMart:

- **Manajemen Produk (Core Subdomain):** Mengelola katalog produk yang tersedia di GoMart, termasuk informasi harga, stok, dan deskripsi produk.
- **Pemesanan Barang (Core Subdomain):** Mengelola proses pemesanan barang oleh pengguna, termasuk pemilihan produk, pembayaran, dan pengiriman.
- **Logistik Pengiriman (Core Subdomain):** Menyediakan sistem untuk mengatur pengiriman barang dari pusat distribusi ke alamat pengguna dengan efisien.
- **Pembayaran Barang (Supporting Subdomain):** Mengelola proses pembayaran untuk layanan GoMart.
- **Laporan Transaksi Barang (Supporting Subdomain):** Memproses dan menghasilkan laporan transaksi yang terkait dengan layanan GoMart.

6. GoSend:

- **Pemesanan Layanan Kurir (Core Subdomain):** Mengelola proses pemesanan layanan kurir untuk pengiriman barang.
- **Pembayaran Layanan Kurir (Supporting Subdomain):** Mengelola proses pembayaran untuk layanan GoSend.
- **Manajemen Kurir dan Pengiriman (Core Subdomain):** Mengelola informasi tentang kurir, rute pengiriman, dan status pengiriman.
- **Laporan Transaksi Kurir (Supporting Subdomain):** Memproses dan menghasilkan laporan transaksi yang terkait dengan layanan GoSend.

7. GoBox:

- **Pemesanan Layanan Pindahan (Core Subdomain):** Mengelola proses pemesanan layanan pindahan barang menggunakan truk GoBox.
- **Pembayaran Layanan Pindahan (Supporting Subdomain):** Mengelola proses pembayaran untuk layanan GoBox.
- **Manajemen Truk dan Pengiriman (Core Subdomain):** Mengelola informasi tentang truk, rute pengiriman, dan status pengiriman untuk layanan GoBox.
- **Laporan Transaksi Pindahan (Supporting Subdomain):** Memproses dan menghasilkan laporan transaksi yang terkait dengan layanan GoBox.

8. GoPay Pinjam:

- **Peminjaman Dana (Core Subdomain):** Mengelola proses peminjaman dana oleh pengguna melalui layanan GoPay Pinjam.
- **Pembayaran Cicilan (Supporting Subdomain):** Mengelola proses pembayaran cicilan pinjaman yang telah diberikan.
- **Manajemen Riwayat Pembayaran (Core Subdomain):** Mengelola riwayat pembayaran dan jadwal pembayaran cicilan.

- **Laporan Transaksi Pinjaman (Supporting Subdomain):** Memproses dan menghasilkan laporan transaksi yang terkait dengan layanan GoPay Pinjam.

Dengan pembagian subdomain seperti di atas, Gojek dapat mengelola dan mengorganisir berbagai aspek bisnisnya dengan lebih terstruktur dan efisien. Setiap subdomain memiliki peran dan tanggung jawabnya sendiri dalam menyediakan layanan kepada pengguna akhir dan memastikan operasi aplikasi berjalan lancar.

- c. (LO1; 20%) Jelaskan bounded context yang terdapat dalam aplikasi dan ubiquitous language yang terdapat dalam bounded context tersebut. Jelaskan pula integrasi bounded context yang digunakan. Gambarkan desain bounded context yang anda buat.

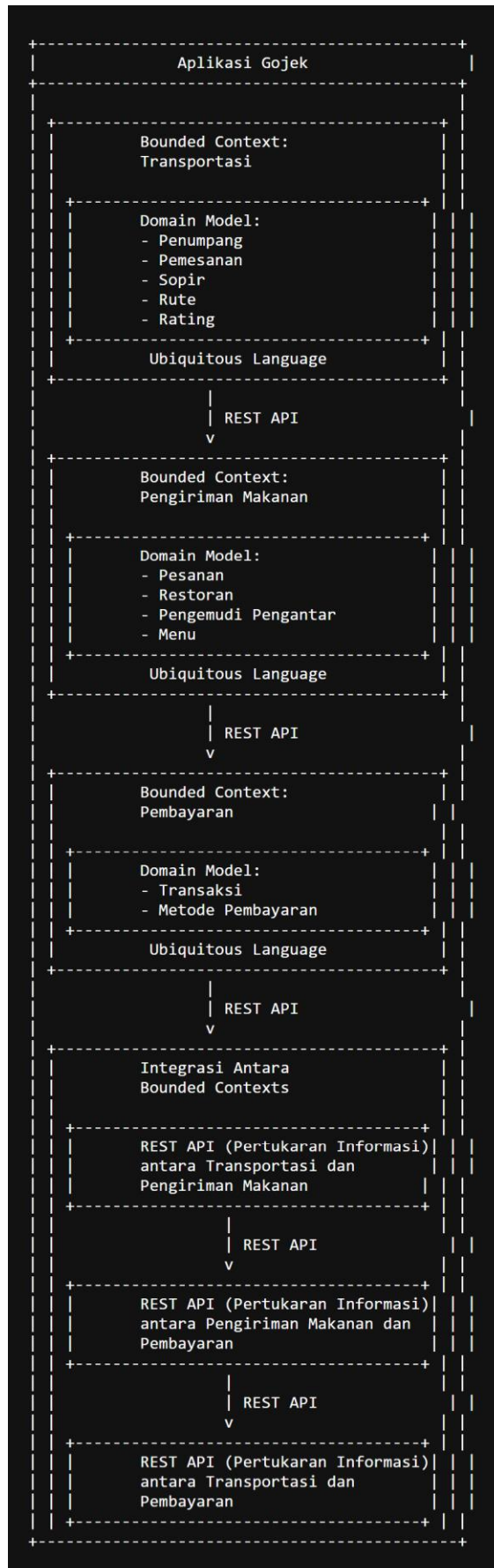
Jawab:

Bounded contexts dalam Gojek mencerminkan pembagian fungsional antara berbagai layanan yang disediakan. Misalnya, ada bounded context yang terpisah untuk layanan transportasi, pengiriman makanan, pembayaran, dan sebagainya. Ubiquitous language yang digunakan dalam masing-masing bounded context mencakup terminologi dan bahasa yang spesifik untuk domain tersebut. Contohnya:

Bounded Context Transportasi: Ubiquitous language di sini mungkin termasuk kata-kata seperti "penumpang", "pemesanan", "sopir", "rute", dan "rating".

Bounded Context Pengiriman Makanan: Ubiquitous language di sini mungkin termasuk kata-kata seperti "pesanan", "restoran", "pengemudi pengantar", dan "menu".

Integrasi antara bounded context dilakukan melalui antarmuka atau API yang memungkinkan pertukaran informasi antara layanan-layanan tersebut. Sebagai contoh, sistem pembayaran dalam pengiriman makanan harus terhubung dengan sistem pembayaran utama Gojek untuk memproses transaksi. Berikut adalah contoh desain bounded context yang mungkin digunakan dalam Gojek:



Dalam diagram ini, integrasi antara ketiga bounded contexts dilakukan melalui REST API, yang memungkinkan pertukaran informasi antara mereka. Panah yang ditunjukkan antara konteks menunjukkan arah pertukaran informasi.

1. Pertama, ada integrasi antara "Transportasi" dan "Pengiriman Makanan", yang memungkinkan pertukaran informasi terkait dengan pengiriman makanan kepada pelanggan.
2. Selanjutnya, terdapat integrasi antara "Pengiriman Makanan" dan "Pembayaran", yang memungkinkan pembayaran untuk pesanan makanan yang telah dikirim.
3. Terakhir, ada integrasi antara "Transportasi" dan "Pembayaran", yang memungkinkan pembayaran untuk layanan transportasi yang telah digunakan oleh pelanggan.

Dengan demikian, gambar tersebut memberikan gambaran yang lebih lengkap tentang bagaimana ketiga bounded contexts berinteraksi satu sama lain dan bagaimana informasi dialirkan melalui sistem aplikasi Gojek.