

Tugas Besar Milestone 2
Desain Batasan
II3160 - Integrated Systems Technology

Diampu oleh:
Daniel Wiyogo Dwiputro, S.T., M.T.



Disusun oleh :
Nakeisha Valya Shakila
18223133

PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JATINANGOR
2025

DAFTAR ISI

DAFTAR ISI.....	2
BAB I : PENDAHULUAN.....	3
1.1. Latar Belakang Masalah.....	3
1.2. Rumusan Masalah.....	3
1.3. Tujuan.....	3
BAB II : PEMBAHASAN.....	4
2.1.Context Mapping.....	4
2.2. Pemilihan Core Context.....	5
BAB III : PENUTUP.....	6
3.1. Kesimpulan.....	6
DAFTAR PUSTAKA.....	7

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Tahap ini dilaksanakan sebagai kelanjutan dari tugas sebelumnya (*Tugas 1*) yang berfokus pada identifikasi domain, subdomain, serta penyusunan *capability mapping*. Pada tahap ini, kegiatan difokuskan untuk mendefinisikan **Bounded Context**, menyusun **Context Map**, dan menentukan **Core Context** sebagai konteks inti dari sistem yang dikembangkan. Langkah ini bertujuan untuk memperjelas batas tanggung jawab setiap konteks, memetakan hubungan serta interaksi antar konteks secara sistematis, dan mengidentifikasi konteks yang memiliki nilai strategis tertinggi bagi sistem. Dengan demikian, hasil dari kegiatan ini diharapkan dapat menjadi dasar dalam perancangan arsitektur sistem yang terstruktur, terintegrasi, serta mendukung pengembangan yang lebih efektif dan berkelanjutan.

1.2. Rumusan Masalah

Dari latar belakang yang ada, terdapat beberapa rumusan masalah yang diharapkan dapat terjawab setelah membaca laporan ini, antara lain sebagai berikut:

- Bagaimana cara mendefinisikan **Bounded Context** dari subdomain yang telah diidentifikasi pada tugas sebelumnya?
- Bagaimana hubungan antar **Bounded Context** dapat divisualisasikan dalam bentuk **Context Map** yang menggambarkan interaksi dan dependensi antar konteks?
- Bagaimana menentukan **Core Context** sebagai konteks inti yang memiliki nilai strategis tertinggi bagi sistem?

1.3. Tujuan

- Mendefinisikan **Bounded Context** untuk setiap subdomain agar batas tanggung jawab sistem menjadi jelas.
- Membuat **Context Map** yang menggambarkan hubungan, kolaborasi, serta arah komunikasi antar konteks dalam sistem.
- Menentukan **Core Context** sebagai konteks utama yang memiliki nilai bisnis paling penting, guna menjadi fokus utama dalam pengembangan sistem selanjutnya.

BAB II

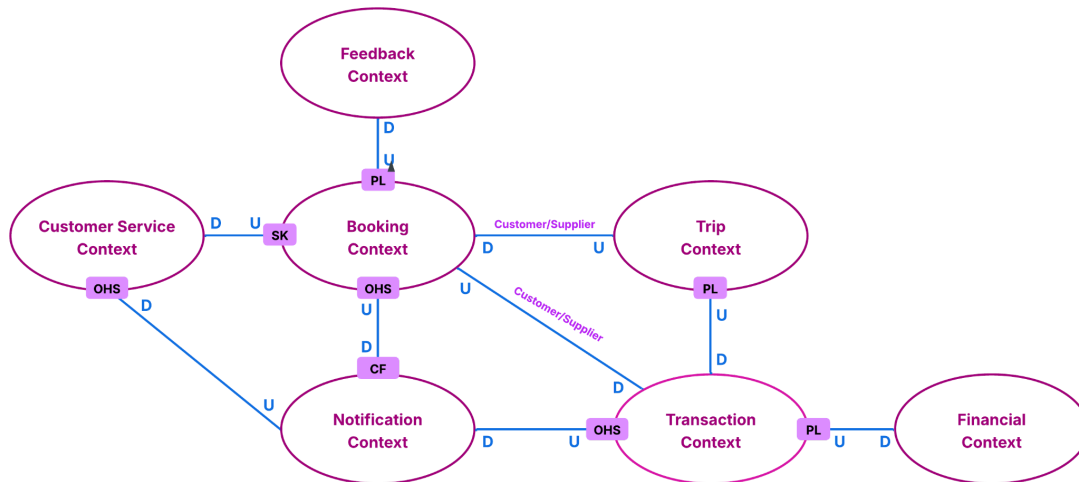
PEMBAHASAN

2.1.Context Mapping

Setiap subdomain dalam core domain *Manajemen Booking Open Trip* memiliki *bounded context* yang membatasi ruang lingkup tanggung jawab, data, serta model yang digunakan untuk memastikan setiap bagian sistem memiliki fokus yang jelas dan dapat saling berinteraksi tanpa tumpang tindih fungsi. Berikut penjelasan masing-masing bounded context:

Subdomain	Bounded Context	Deskripsi
Trip Management	Trip Context	Mengelola informasi dan aktivitas terkait perjalanan wisata seperti itinerary, destinasi, jadwal, kapasitas, serta penugasan guide untuk setiap trip yang ditawarkan.
Booking System	Booking Context	Menangani proses pemesanan peserta open trip, mulai dari registrasi, pembatalan, <i>refund</i> , hingga konfirmasi status perjalanan secara real-time.
Finance and Payment	Transaction Context	Menyediakan layanan pendukung internal yang menghubungkan berbagai komponen sistem, termasuk pengelolaan permintaan bantuan dan dukungan pelanggan.
	Financial Report Context	Menghasilkan laporan dan ringkasan keuangan berdasarkan data transaksi dan hasil trip, guna mendukung analisis profitabilitas dan evaluasi bisnis.
Support System	Feedback Context	Mengelola pengumpulan ulasan dan rating dari peserta setelah perjalanan selesai untuk menilai kualitas layanan dan kepuasan pelanggan.
	Customer Service Context	Menangani pertanyaan, keluhan, serta permintaan bantuan peserta terkait proses booking atau pelaksanaan trip melalui layanan pelanggan.
Notification System	Notification Context	Mengatur proses pengiriman notifikasi otomatis kepada peserta, seperti konfirmasi pemesanan, status pembayaran, serta pengingat keberangkatan.

Berdasarkan hasil identifikasi bounded context, Context Mapping memetakan hubungan komunikasi dan pertukaran data antar konteks dalam domain *Manajemen Booking Open Trip* sehingga interaksi antarsubdomain berjalan terkoordinasi, konsisten, serta mendukung pengembangan sistem yang terstruktur dan modular.



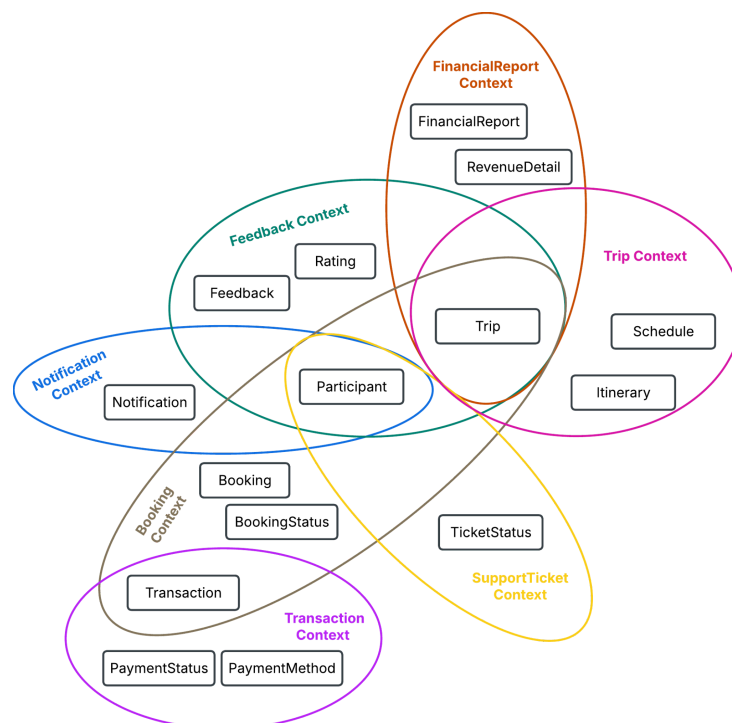
Gambar 2.1 Context Mapping Berdasarkan DDD

Context Mapping dalam **Domain-Driven Design (DDD)** merupakan teknik untuk memetakan hubungan antar *bounded context* dalam suatu sistem yang kompleks. Setiap *bounded context* memiliki model dan tanggung jawab yang berbeda, sehingga diperlukan context map untuk menggambarkan bagaimana konteks-konteks tersebut saling berinteraksi, berbagi data, dan saling bergantung. **Trip Context** menyediakan informasi penting seperti jadwal, kapasitas, dan detail trip yang digunakan oleh **Booking Context**. Setelah peserta melakukan pemesanan, Booking Context mengirimkan event seperti *BookingConfirmed* yang dipakai oleh **Feedback Context**, **Notification Context**, dan juga **Customer Service Context** untuk memeriksa data booking saat menangani pertanyaan atau keluhan peserta. Pembayaran diproses oleh **Transaction Context**, lalu hasilnya dikirim ke **Financial Report Context** untuk dibuatkan laporan keuangan. Melalui context mapping, dapat diidentifikasi arah komunikasi seperti *upstream (U)* dan *downstream (D)*, serta jenis hubungan seperti pada tabel dibawah ini :

Upstream	Downstream	Jenis Hubungan	Penjelasan
Trip Context	Booking Context	CS (Customer/ Supplier)	Booking Context menjadi <i>customer</i> yang bergantung pada data trip, jadwal, dan kapasitas yang disediakan oleh Trip Context.
	Transaction Context	PL (Published Language)	Trip Context menyediakan data referensi (tripId dan schedule) yang digunakan Transaction Context untuk validasi laporan biaya per trip.
Booking Context	Transaction Context	CS (Customer/ Supplier)	Transaction Context memproses pembayaran berdasarkan permintaan yang dikirimkan oleh Booking Context.
	Feedback Context	PL (Published Language)	Booking Context mempublikasikan event <i>TripCompleted</i> dan <i>ParticipantInfo</i> yang digunakan oleh Feedback Context untuk pengumpulan ulasan.
	Customer Service	SK (Shared Kernel)	Kedua context berbagi model domain <i>Participant</i> dan sebagian data booking untuk kebutuhan validasi

	Context		tiket bantuan pelanggan.
Booking Context	Notification Context	OHS (Open Host Service)	Booking Context membuka layanan event publik seperti <i>BookingConfirmed</i> dan <i>BookingCancelled</i> yang dikonsumsi Notification Context.
		CF (Conformist)	Notification Context menyesuaikan struktur pesan dan format event sesuai model data Booking agar pesan dapat dikirim tanpa duplikasi transformasi.
Transaction Context	Financial Report Context	PL (Published Language)	Transaction Context menerbitkan data transaksi yang digunakan Financial Report Context untuk menghitung pendapatan dan biaya.
	Notification Context	OHS (Open Host Service)	Transaction Context membuka endpoint event <i>PaymentSuccess</i> dan <i>PaymentFailed</i> agar Notification Context dapat mengirimkan notifikasi otomatis.
Customer Service Context	Notification Context	OHS (Open Host Service)	Customer Service Context memanfaatkan layanan notifikasi terbuka untuk mengirimkan update tiket atau pesan bantuan ke peserta.

Pendekatan ini membantu dalam memahami peran dan tanggung jawab setiap bagian sistem, menjaga keselarasan hubungan antar bagian, serta memastikan proses pengembangan berlangsung secara teratur, efisien, dan mudah dikembangkan di masa mendatang.



Gambar 2.2 Integration Context Mapping

Integration Context Map merupakan representasi visual yang menunjukkan bagaimana setiap *bounded context* atau sistem besar saling berinteraksi dan bertukar data. Fokus utamanya adalah **hubungan integrasi antar sistem**, bukan hanya batas domain. Setiap konteks dapat berisi beberapa *entity* untuk menggambarkan data utama yang dimiliki atau digunakan dalam interaksi tersebut. Dengan menampilkan entitas di dalamnya, diagram ini membantu memahami **alur data dan dependensi antar konteks** secara lebih konkret. Berbeda dengan **Context Map pada Domain-Driven Design (DDD)** yang umumnya hanya menyoroti batas domain dan jenis hubungan tanpa menampilkan detail isi konteks, *Integration Context Map* menambahkan elemen entitas internal agar hubungan yang digambarkan lebih operasional dan berbasis data. Pada sistem ini, alur data utama dimulai dari **Trip Context** menyediakan informasi seperti jadwal dan kapasitas yang dipakai **Booking Context** untuk memproses pemesanan. Setelah booking selesai, **Booking Context** mengirimkan event yang dimanfaatkan **Feedback Context**, **Notification Context**, dan **Customer Service Context** untuk memberi update atau menangani kebutuhan peserta. Proses pembayaran diolah oleh **Transaction Context** dan hasilnya diteruskan ke **Financial Report Context** untuk penyusunan laporan keuangan. (harus tetap 3 kalimat)

2.2. Pemilihan Core Context

Dalam proses pembuatan *context map*, langkah penting setelah mengidentifikasi seluruh *bounded context* adalah menentukan *core context*, yaitu konteks yang memiliki nilai strategis tertinggi bagi bisnis dan menjadi pusat dari aliran proses utama dalam sistem. *Core context* dipilih karena memiliki peran paling besar dalam mendukung tujuan bisnis dan secara langsung memengaruhi pengalaman pengguna. Pada sistem pemesanan open trip, **Booking Context** dipilih sebagai core context karena menjadi pusat dari seluruh aktivitas pemesanan serta berfungsi sebagai penghubung utama antara pelanggan dan sistem internal. Peran ini menjadikannya elemen penting dalam menjaga kelancaran operasional dan pengalaman pengguna. Pada tahap berikutnya, Milestone 3 akan berfokus pada perancangan domain Booking secara lebih rinci, sedangkan Milestone 4 akan mulai mengintegrasikan Trip Context, Booking Context, dan Transaction Context sebagai dasar alur pemesanan dan pembayaran.

BAB III

PENUTUP

3.1. Kesimpulan

Context mapping berperan penting dalam menggambarkan bagaimana setiap bounded context saling berinteraksi dan berbagi tanggung jawab dalam suatu sistem. Melalui pemetaan ini, hubungan dan alur data antar konteks dapat dipahami dengan jelas, sehingga memudahkan integrasi dan pengelolaan sistem secara keseluruhan. Dari hasil context mapping tersebut, dapat diidentifikasi **core context** yang menjadi pusat aktivitas bisnis dan memiliki nilai strategis tertinggi. Dalam sistem pemesanan open trip, **Sistem Booking dan Registrasi Peserta** ditetapkan sebagai core context karena menjadi penghubung utama antara pelanggan dan sistem internal lain, serta berperan langsung dalam proses pemesanan dan konfirmasi perjalanan. Dengan demikian, context mapping membantu menentukan prioritas desain dan pengembangan sistem melalui penentuan konteks inti yang paling berpengaruh terhadap tujuan bisnis.

DAFTAR PUSTAKA

- Context map.* (2023, July 6). *Domain-driven Design: A Practitioner's Guide.*
<https://ddd-practitioners.com/home/glossary/context-map/>
- GeeksforGeeks.* (2025, July 12). *DomainDriven Design (DDD).* *GeeksforGeeks.*
<https://www.geeksforgeeks.org/system-design/domain-driven-design-ddd/>
- Alokmishra, V. a. P. B.* (2025, May 24). *DDD Context Mapping by example: Policy Management.* *Alok Mishra.*
<https://alok-mishra.com/2021/06/29/ddd-context-mapping-by-example-policy-management/>
- Wojtyna, M.* (2024, April 12). *Context mapping.* *Medium.*
https://medium.com/@mike_7149/context-mapping-4b4909cf195a
- M.* (2022, March 2). *Domain Driven Design | Deep dive into context mapping.* *Medium.*
<https://medium.com/geekculture/domain-driven-design-deep-dive-into-context-mapping-f9cd8acddd7e>