

**Deliverable 2**  
**DESAIN BATASAN**  
**II3160 - Integrated Systems Technology**



**Disusun oleh:**  
Theresia Ivana Marella Siswahyudi - 18223126

**Program Studi Sistem Dan Teknologi Informasi**  
**Sekolah Teknik Elektro Dan Informatika**  
**Institut Teknologi Bandung**  
**2025**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>LATAR BELAKANG.....</b>	<b>3</b>
<b>PENDEFINISIAN BOUNDED CONTEXT.....</b>	<b>4</b>
A. Konsep dan Prinsip Bounded Context.....	4
B. Hubungan dengan Subdomain dan Kapabilitas Bisnis.....	4
C. Identifikasi dan Deskripsi Bounded Context.....	7
<b>PEMETAAN CONTEXT MAP.....</b>	<b>9</b>
A. Konsep Umum Context Map.....	9
B. Hubungan dan Pola Integrasi Antar Context.....	9
C. Context Map.....	11
<b>PEMILIHAN CORE CONTEXT.....</b>	<b>13</b>
A. Konsep dan Tujuan Core Context.....	13
B. Alasan Pemilihan Inventory Control sebagai Core Context.....	13
C. Ruang Lingkup dan Batasan Konteks.....	14

## LATAR BELAKANG

Dalam sistem manajemen gudang (*Warehouse Management System* / WMS), kompleksitas operasional sering kali muncul akibat banyaknya proses yang saling bergantung, mulai dari penerimaan barang (*inbound*), penyimpanan dan pengendalian stok (*inventory*), hingga pengiriman (*outbound*). Tiap proses melibatkan aktor berbeda serta data yang terus berubah secara *real-time*, seperti jumlah stok, lokasi penyimpanan, maupun status pesanan. Tanpa batasan yang jelas antara area tanggung jawab, sistem mudah menjadi monolit besar yang sulit dipelihara dan rentan inkonsistensi data. Karena itu, pendekatan Domain-Driven Design (DDD) dipilih untuk membagi domain besar ini menjadi bagian-bagian yang lebih terfokus dan terkelola.

Melalui DDD, domain bisnis diuraikan ke dalam *bounded context*, yaitu batas eksplisit tempat model dan bahasa bisnis digunakan secara konsisten. Tiap *bounded context* mewakili bagian tertentu dari sistem dengan model data, aturan bisnis, dan antarmuka yang otonom. Dengan cara ini, komunikasi antar-bagian sistem dapat diatur lewat kontrak yang jelas, seperti API atau event, tanpa harus menyatukan seluruh logika bisnis dalam satu basis kode. Pendekatan ini sejalan dengan prinsip modularitas, *loose-coupling*, serta mempermudah kolaborasi antartim pengembang.

Laporan ini berfokus pada pendefinisian *bounded context* untuk domain WMS, pembuatan *context map* guna memvisualisasikan relasi antarkonteks, serta penentuan satu konteks inti (*core context*) yang akan diimplementasikan lebih lanjut. Melalui pemetaan ini, diharapkan dapat diperoleh pemahaman menyeluruh mengenai batas tanggung jawab setiap konteks dan bagaimana kolaborasinya membentuk sistem yang utuh namun tetap fleksibel. Tahapan ini menjadi fondasi penting sebelum memasuki tahap desain dan implementasi teknis konteks inti, yaitu bagian sistem yang memberikan nilai bisnis utama.

## PENDEFINISIAN *BOUNDED CONTEXT*

### A. Konsep dan Prinsip *Bounded Context*

Dalam pendekatan Domain-Driven Design (DDD), konsep *bounded context* berfungsi untuk menetapkan batas yang jelas antara satu bagian model domain dengan bagian lainnya. Setiap *bounded context* merepresentasikan satu wilayah semantik (*semantic boundary*) di mana istilah, aturan bisnis, dan struktur data memiliki makna yang konsisten dan tidak saling bertentangan. Dengan adanya batas ini, tim pengembang dapat menghindari kesalahan komunikasi lintas modul serta menjaga agar perubahan di satu area domain tidak menimbulkan efek samping yang tidak diinginkan pada area lain.

Pendefinisian *bounded context* membantu memecah domain besar menjadi potongan-potongan yang lebih mudah dikelola, masing-masing dengan tanggung jawab dan model data yang spesifik. Dalam konteks sistem manajemen pergudangan (*Warehouse Management System* / WMS), pembagian ini menjadi penting karena proses seperti penerimaan barang, penyimpanan, pelacakan, dan pengiriman memiliki terminologi serta alur data yang berbeda. Dengan demikian, setiap *bounded context* dapat dikembangkan secara mandiri dengan model, database, serta antarmuka yang sesuai dengan tujuan bisnisnya, namun tetap terintegrasi dalam sistem secara keseluruhan.

### B. Hubungan dengan Subdomain dan Kapabilitas Bisnis

Tahap pendefinisian *bounded context* dilakukan dengan mengacu pada hasil analisis domain dan pemetaan kapabilitas bisnis pada tahap sebelumnya. Setiap subdomain yang telah diidentifikasi, baik *core*, *supporting*, maupun *generic* dan diimplementasikan ke dalam satu atau lebih *bounded context* yang mewakili kapabilitas bisnis tertentu.

Tabel berikut menunjukkan keterkaitan antara subdomain hasil analisis dengan *bounded context* yang dikembangkan.

**Tabel 1.** Hubungan Subdomain dengan Kapabilitas Bisnis

Subdomain	<i>Bounded Context</i>	Jenis Domain	Fokus Kapabilitas Bisnis
Penerimaan Barang ( <i>Inbound Management</i> )	<i>Inbound Management Context</i>	<i>Core</i>	Mewujudkan kapabilitas <i>Inbound Process Management</i> , yaitu penerimaan barang dari supplier, pemeriksaan kualitas, dan pembaruan stok otomatis melalui modul <i>Receiving</i> , <i>Quality Check</i> , dan

			<i>Stock Update Automation.</i>
<b>Pengelolaan Stok (Inventory Management)</b>	<b><i>Inventory Control Context</i></b>	<b><i>Core</i></b>	Mengimplementasikan kapabilitas <i>Inventory Control</i> , yang mencakup <i>Real-Time Stock Tracking</i> , <i>Location Mapping</i> , dan <i>Adjustment &amp; Audit Tools</i> untuk memastikan akurasi persediaan.
<b>Pengeluaran Barang (Outbound Management)</b>	<b><i>Outbound Fulfillment Context</i></b>	<b><i>Core</i></b>	Mendukung kapabilitas <i>Outbound Process Management</i> melalui <i>Order Picking Optimization</i> , <i>Packing &amp; Labeling</i> , dan <i>Shipping Confirmation</i> agar pemenuhan pesanan efisien dan akurat.
<b>Pelacakan Barang (Tracking System)</b>	<b><i>Tracking &amp; Traceability Context</i></b>	<b><i>Supporting</i></b>	Mengimplementasikan kapabilitas <i>Audit &amp; Traceability</i> , melacak <i>Transaction Log</i> , <i>Item Trace History</i> , dan <i>Compliance Record</i> guna mendukung audit dan kepatuhan.
<b>Pelaporan dan Analitik (Performance Monitoring &amp; Reporting)</b>	<b><i>Reporting &amp; Analytics Context</i></b>	<b><i>Supporting</i></b>	Mengimplementasikan kapabilitas <i>Performance Monitoring &amp; Reporting</i> dan <i>Data-Driven Decision Support</i> untuk menampilkan KPI, tren, serta analisis

			prediktif berbasis data operasional gudang.
<b>Perencanaan dan Prediksi (<i>Forecasting &amp; Planning Engine</i>)</b>	<b><i>Reporting &amp; Analytics Context</i></b>	<i>Supporting</i>	Mendukung <i>Stock Demand Forecasting, Space Utilization Estimation, dan Resource Allocation Suggestion</i> untuk mendukung keputusan strategis gudang.
<b>Notifikasi Operasional (<i>Notification &amp; Alert System</i>)</b>	<b><i>Notification &amp; Alert Context</i></b>	<i>Supporting</i>	Mengimplementasikan kapabilitas <i>Event-Based Alerts, Push Notifications, dan Escalation Rules</i> untuk memantau kondisi kritis gudang.
<b>Otentikasi dan Otorisasi (<i>User &amp; Access Management</i>)</b>	<b><i>Identity &amp; Access Context</i></b>	<i>Generic</i>	Mendukung kapabilitas <i>Role-Based Access Control, Authentication Management, dan Session Logging</i> sebagai fondasi keamanan sistem.
<b>Integrasi Sistem Eksternal (<i>System Integration Management</i>)</b>	<b><i>External Integration Context</i></b>	<i>Generic</i>	Mewujudkan kapabilitas <i>API Management, External System Connector, dan Data Synchronization</i> untuk integrasi dengan ERP, POS, dan e-commerce.
<b>Manajemen Data dan Keamanan (<i>Data Management &amp; Security</i>)</b>	<b><i>External Integration Context (bersama Identity Context)</i></b>	<i>Generic</i>	Menyediakan <i>Centralized Data Storage, Encryption, serta Backup &amp; Recovery</i> untuk

			menjaga integritas data antar konteks.
<b>Pengendalian Sumber Daya Gudang (Warehouse Resource Control)</b>	<b>Inventory Control Context</b> (beririsan dengan Outbound)	<i>Supporting</i>	Mengimplementasikan modul <i>Task Assignment</i> dan <i>Equipment Tracking</i> untuk mengoptimalkan penggunaan sumber daya gudang.

Pemetaan ini menunjukkan bahwa *bounded context* diturunkan langsung dari kebutuhan bisnis yang telah didefinisikan pada peta kapabilitas. Pada tahap sebelumnya, yaitu *deliverable 1*, setiap kapabilitas telah dipetakan ke dalam kelompok *strategic*, *supporting*, dan *operational capabilities*, beserta subdomain yang menaunginya. Struktur tersebut menjadi dasar bagi penyusunan *bounded context* pada tabel ini, sehingga hubungan antara subdomain dan kapabilitas dapat diterjemahkan secara lebih konkret ke dalam area desain yang memiliki tanggung jawab jelas. Dengan cara ini, setiap konteks berdiri sebagai implementasi konkret dari satu set kemampuan bisnis yang spesifik dan independen, sekaligus menjadi fondasi bagi perumusan arsitektur detail pada tahap desain atau *deliverable* berikutnya.

### C. Identifikasi dan Deskripsi *Bounded Context*

Bagian ini menjelaskan masing-masing *bounded context* yang telah diidentifikasi beserta tanggung jawab utamanya, batas domainnya, serta hubungannya secara umum dengan konteks lain.

**Tabel 2.** Identifikasi *Bounded Context*

<b>Nama Bounded Context</b>	<b>Deskripsi Singkat</b>	<b>Jenis Domain</b>	<b>Tanggung Jawab Utama</b>
Inbound Management	Mengelola seluruh proses penerimaan barang dari pemasok, termasuk registrasi, pemeriksaan kualitas, dan pembaruan stok otomatis setelah verifikasi.	<i>Core</i>	Mencatat data penerimaan barang, melakukan verifikasi kualitas, memperbarui stok awal, serta mengirim event <i>GoodsReceived</i> ke konteks Inventory.
Inventory Control	Pusat kendali data stok, lokasi, dan pergerakan barang di dalam gudang. Menjadi <i>single source of truth</i> bagi semua data stok.	<i>Core</i>	Menyimpan, memperbarui, dan menyediakan data stok real-time; mengelola lokasi penyimpanan, audit stok, serta mendukung kebutuhan Inbound, Outbound, dan Reporting.

Outbound Fulfillment	Mengatur proses pengeluaran barang dari pesanan pelanggan, termasuk pengambilan, pengepakan, dan konfirmasi pengiriman.	<i>Core</i>	Mengonversi pesanan menjadi aktivitas pengambilan barang, mengatur proses pengepakan dan labeling, serta memastikan pengiriman terkonfirmasi ke pelanggan.
Order Orchestration	Menjadi penghubung antara sistem eksternal dan proses outbound internal, mengelola pembentukan serta status pesanan.	<i>Supporting</i>	Mengelola data pesanan dari sistem eksternal, mengatur alur pemenuhan order, dan memicu proses pengambilan barang di konteks Outbound.
Tracking & Traceability	Melacak pergerakan barang berdasarkan SKU atau batch untuk keperluan audit dan kepatuhan.	<i>Supporting</i>	Mencatat jejak pergerakan barang, membangun histori transaksi, dan menyediakan data pelacakan untuk audit serta pelaporan.
Reporting & Analytics	Menyediakan analitik operasional, KPI, serta laporan performa dan perencanaan gudang.	<i>Supporting</i>	Mengumpulkan dan mengagregasi data dari berbagai konteks melalui event bus, menampilkan KPI dan tren performa, serta membantu pengambilan keputusan berbasis data.
Notification & Alert	Menyediakan peringatan otomatis berbasis event untuk kondisi operasional tertentu.	<i>Supporting</i>	Mengirimkan notifikasi atau peringatan kepada pengguna terkait aktivitas gudang, kondisi stok kritis, atau keterlambatan proses pengiriman.
Identity & Access	Mengelola autentikasi, otorisasi, dan audit aktivitas pengguna di seluruh konteks sistem.	<i>Generic</i>	Mengatur hak akses pengguna berdasarkan peran, mengelola proses login dan sesi, serta mencatat aktivitas untuk keperluan audit keamanan.
External Integration	Menyediakan mekanisme pertukaran data antara sistem gudang dan sistem eksternal seperti ERP, POS, dan e-commerce.	<i>Generic</i>	Menghubungkan sistem WMS dengan sistem eksternal melalui API dan sinkronisasi data, serta memastikan keamanan data dengan enkripsi dan pencadangan otomatis.



## PEMETAAN *CONTEXT MAP*

### A. Konsep Umum *Context Map*

*Context Map* merupakan artefak visual dan konseptual dalam *Domain-Driven Design* (DDD) yang menggambarkan bagaimana setiap *bounded context* berinteraksi satu sama lain. Dalam sistem yang kompleks seperti *Warehouse Management System* (WMS), setiap konteks memiliki tanggung jawab dan bahasa domainnya sendiri, namun harus tetap dapat berkomunikasi secara konsisten.

*Context Map* ini menunjukkan:

- 1) Arah hubungan antar konteks (upstream dan downstream).
- 2) Jenis kontrak komunikasi (misalnya API, event, atau message bus).
- 3) Pola integrasi yang digunakan untuk menjaga stabilitas antar model domain.

Tujuan utamanya adalah memastikan koordinasi yang jelas antar tim pengembang serta menghindari kebingungan semantik atau duplikasi data lintas konteks.

### B. Hubungan dan Pola Integrasi Antar *Context*

Hubungan antar konteks dalam sistem WMS dibangun berdasarkan kebutuhan data, urutan proses, serta tanggung jawab domain. Tabel berikut menjelaskan hubungan utama antar konteks dan pola integrasi yang digunakan:

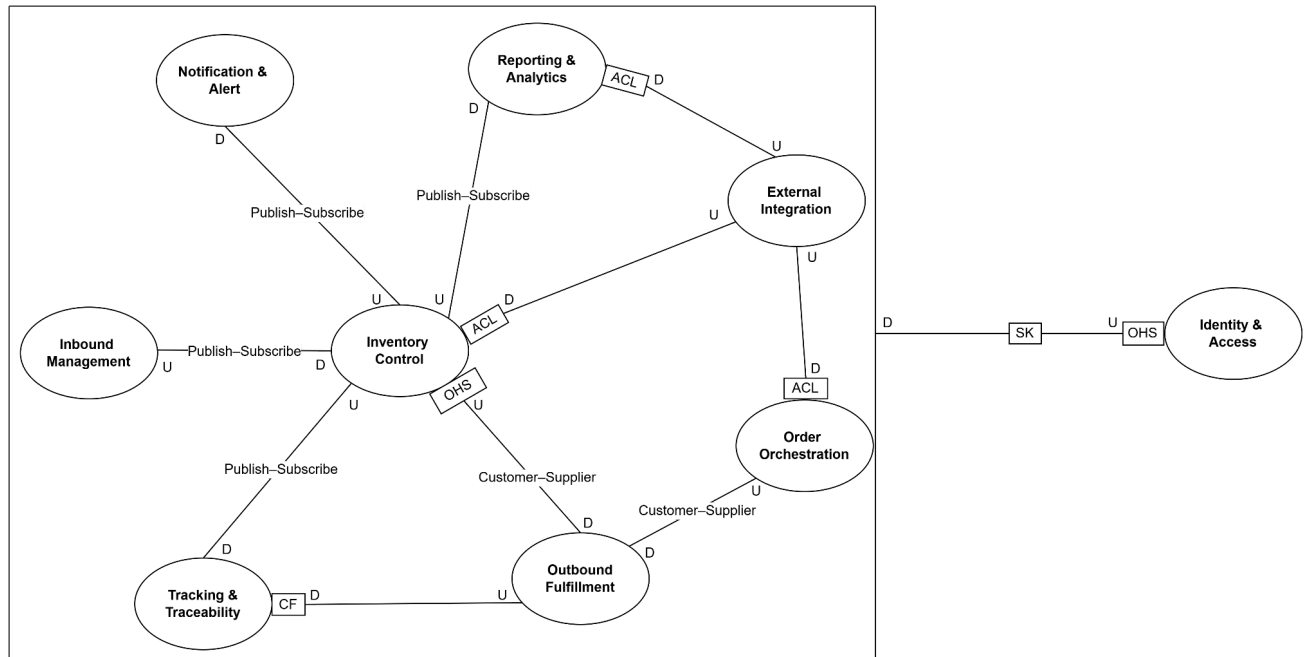
**Tabel 3.** Hubungan dan Pola Antar *Context*

Upstream Context	Downstream Context	Jenis Hubungan	Pola Kolaborasi / Integrasi	Penjelasan Singkat
Inbound Management	Inventory Control	<i>Data Flow</i>	<i>Publish–Subscribe (Event-Driven)</i>	Saat barang diterima dan diverifikasi, <i>Inbound</i> mempublikasikan event <i>GoodsReceived</i> yang di-subscribe oleh <i>Inventory</i> untuk memperbarui stok secara otomatis.
Inventory Control	Outbound Fulfillment	<i>Data Query</i>	<i>Open Host Service (OHS)</i>	<i>Inventory</i> menyediakan API publik yang stabil agar <i>Outbound</i> dapat mengecek ketersediaan barang tanpa mengetahui detail internal <i>Inventory</i> .

<b>Order Orchestration</b>	<b>Outbound Fulfillment</b>	<i>Command Trigger</i>	<i>Customer–Supplier</i>	<i>Order Orchestration</i> bertindak sebagai <i>customer</i> yang memicu proses pengambilan barang di <i>Outbound</i> , dengan kontrak komunikasi yang disepakati bersama.
<b>Inventory Control</b>	<b>Tracking &amp; Traceability</b>	<i>Event Notification</i>	<i>Publish–Subscribe</i>	<i>Inventory</i> mempublikasikan event <i>StockChanged</i> setiap kali terjadi perubahan stok, dan <i>Tracking</i> menggunakan event ini untuk memperbarui histori pergerakan barang.
<b>Outbound Fulfillment</b>	<b>Tracking &amp; Traceability</b>	<i>Data Feed</i>	<i>Conformist</i>	<i>Tracking</i> mengikuti model data pengiriman yang dimiliki oleh <i>Outbound</i> , karena format data pengiriman sudah menjadi standar sistem.
<b>Inventory Control</b>	<b>Reporting &amp; Analytics</b>	<i>Event Stream</i>	<i>Publish–Subscribe</i>	<i>Reporting</i> berlangganan event dari <i>Inventory</i> , <i>Inbound</i> , dan <i>Outbound</i> untuk melakukan agregasi data performa gudang dan pembaruan KPI Dashboard.
<b>Inventory Control</b>	<b>Notification &amp; Alert</b>	<i>Event Stream</i>	<i>Publish–Subscribe</i>	<i>Notification</i> menerima event seperti <i>LowStockAlert</i> atau <i>ShipmentDelayed</i> untuk dikirim ke pengguna terkait.
<b>Identity &amp; Access</b>	Semua Context	<i>Shared Service</i>	<i>Open Host Service (OHS)</i>	Semua konteks lain menggunakan layanan autentikasi dan otorisasi dari <i>Identity &amp; Access</i> melalui API bersama.
<b>External Integration</b>	<b>Order Orchestration, Reporting, Inventory Control</b>	<i>Data Sync</i>	<i>Anti-Corruption Layer (ACL)</i>	Menerjemahkan format data eksternal (ERP, POS, e-commerce) ke model internal sistem gudang untuk menjaga konsistensi semantik antar sistem.

### C. Context Map

Secara konseptual, peta konteks sistem WMS dapat digambarkan sebagai berikut.



Gambar 1. Context Map Sistem WMS

Diagram *context map* di atas menunjukkan bahwa *Inventory Control Context* berperan sebagai inti komunikasi (*core context*) yang menghubungkan seluruh konteks operasional dalam sistem manajemen gudang. Konteks ini berfungsi sebagai sumber kebenaran utama (*single source of truth*) untuk seluruh data stok, lokasi penyimpanan, serta pergerakan barang, sehingga setiap interaksi antar konteks lainnya secara langsung atau tidak langsung bergantung pada informasi yang dihasilkan oleh *Inventory*.

Hubungan antar konteks dibangun dengan prinsip *upstream-downstream*, di mana konteks upstream bertugas menyediakan data atau layanan (*provider*), sedangkan konteks downstream bertindak sebagai konsumen (*consumer*) yang menggunakan atau menyesuaikan data tersebut. Pola integrasi yang digunakan disesuaikan dengan kebutuhan masing-masing interaksi, antara lain:

- *Publish-Subscribe (Event-Driven)* digunakan secara luas antara *Inventory Control* dengan konteks lain seperti *Reporting & Analytics*, *Tracking & Traceability*, dan *Notification & Alert*. Pola ini memungkinkan komunikasi *asynchronous*, di mana setiap perubahan stok, penerimaan barang, atau pergerakan item akan memicu event yang otomatis dikonsumsi oleh konteks *downstream* tanpa mengganggu proses utama.
- *Open Host Service (OHS)* diterapkan pada *Inventory Control* dan *Identity & Access* sebagai penyedia layanan publik yang dapat diakses oleh konteks lain. OHS memastikan kontrak API dan format data (*Published Language*) yang konsisten sehingga integrasi antar sistem dapat dilakukan dengan cara yang stabil dan terstandarisasi.
- *Anti-Corruption Layer (ACL)* diterapkan pada *External Integration* untuk mengisolasi model domain internal dari sistem eksternal seperti ERP, POS, atau *e-commerce*. Dengan cara ini, format

data eksternal diterjemahkan sebelum digunakan oleh *Inventory Control* maupun *Reporting & Analytics*, menjaga agar domain internal tetap bersih dari pengaruh model luar.

- *Customer–Supplier Relationship* menggambarkan koordinasi antara *Order Orchestration* (sebagai customer) dan *Outbound Fulfillment* (sebagai supplier), di mana pemicu proses pengiriman berasal dari permintaan pesanan yang dikontrol oleh *Order Context*.
- *Conformist Pattern* digunakan antara *Outbound Fulfillment* dan *Tracking & Traceability*, karena model data pengiriman yang digunakan oleh Outbound telah menjadi standar yang diikuti *Tracking* tanpa translasi tambahan.

Dari keseluruhan peta konteks, dapat dilihat bahwa arsitektur komunikasi sistem ini bersifat modular namun terintegrasi, setiap *bounded context* memiliki batas tanggung jawabnya sendiri, namun terhubung melalui mekanisme event, API, atau ACL yang menjaga kohesi domain. Pendekatan ini memungkinkan sistem untuk berkembang secara *scalable* dan *loosely coupled*, di mana perubahan pada satu konteks (misalnya penambahan fitur *Inventory* atau pelaporan) tidak akan berdampak langsung terhadap konteks lainnya.

Pada konteks *Inventory Control*, penerapan pola *Open Host Service* (OHS) tidak hanya berfungsi sebagai mekanisme integrasi antarkonteks, tetapi juga menjadi dasar pembentukan kontrak layanan yang bersifat stabil dan dapat diakses oleh modul lain. Kontrak OHS ini nantinya diwujudkan secara langsung melalui endpoint pada lapisan backend, khususnya jalur */ohs/...*, yang digunakan oleh *Inbound*, *Outbound*, *Order*, maupun konteks lain untuk berinteraksi dengan sumber data stok secara konsisten. Implementasi teknis dari kontrak tersebut akan terlihat jelas pada *deliverable 4*, ketika antarmuka layanan ini direalisasikan melalui API *Inventory Control*. Dengan demikian, pola integrasi yang tergambar pada *context map* memiliki kaitan langsung dengan tahapan implementasi berikutnya, terutama dalam pemenuhan kebutuhan akses data *real-time* terhadap *Inventory Control* sebagai sumber kebenaran utama.

## PEMILIHAN *CORE CONTEXT*

### A. Konsep dan Tujuan *Core Context*

Pada pendekatan *Domain-Driven Design* (DDD), *core context* merupakan area domain yang paling penting secara strategis karena mengandung logika bisnis utama dan memberikan nilai kompetitif tertinggi bagi sistem. Penentuan konteks inti dilakukan agar fokus desain dan implementasi diarahkan pada bagian domain yang menjadi pusat aktivitas dan sumber data utama.

Dalam sistem manajemen gudang ini, *core context* dipilih dengan mempertimbangkan tiga aspek utama: nilai bisnis, ketergantungan konteks lain, dan kompleksitas domain. Dengan demikian, konteks yang terpilih akan menjadi pondasi bagi integrasi dan stabilitas keseluruhan sistem.

### B. Alasan Pemilihan *Inventory Control* sebagai *Core Context*

Berdasarkan hasil analisis pada *context map*, konteks *Inventory Control* menempati posisi sentral dalam aliran informasi dan komunikasi antar domain. Kriteria pemilihan ini dapat dijelaskan melalui beberapa pertimbangan berikut:

1) Nilai Strategis Tinggi:

*Inventory Control* menjadi sumber data utama bagi seluruh aktivitas gudang, mulai dari penerimaan barang (*inbound*), pengeluaran (*outbound*), hingga analisis performa dan notifikasi operasional. Integritas dan akurasi data stok dari konteks ini sangat mempengaruhi kinerja konteks lainnya.

2) Ketergantungan Lintas Konteks:

Hampir semua konteks lain bergantung pada data atau *event* yang dihasilkan oleh *Inventory Control*. Misalnya, *Reporting & Analytics* memanfaatkan *event* perubahan stok untuk pembaruan KPI, sementara *Outbound Fulfillment* mengakses API untuk memvalidasi ketersediaan barang.

3) Kompleksitas dan Stabilitas Model Tinggi:

*Inventory Control* menangani aturan bisnis yang kompleks, seperti audit stok, reservasi, serta sinkronisasi lokasi penyimpanan barang. Model ini bersifat stabil dan dapat digunakan sebagai acuan bagi konteks lain tanpa sering berubah.

4) Potensi Modularisasi:

*Inventory Control* memiliki struktur domain yang dapat diisolasi dan dikembangkan sebagai layanan mandiri (*microservice core*), menjadikannya pondasi ideal bagi tahap implementasi berikutnya.

Selain itu, pengembangan pada tahap teknis awal, yakni *deliverable 3* (Desain Taktis) dan *deliverable 4* (Implementasi Awal), akan memusatkan perhatian pada konteks *Inventory Control* sebagai fondasi sistem. Sementara itu, konteks lain seperti *Inbound Management*, *Outbound Fulfillment*, *Reporting & Analytics*, serta *Notification & Alert* berada pada tahap perancangan konseptual dan integrasi berbasis event, sehingga penyempurnaan dan implementasinya dapat dilakukan secara bertahap setelah kerangka utama pada *core context* ini terbentuk dengan stabil.

### C. Ruang Lingkup dan Batasan Konteks

Sebagai konteks inti, *Inventory Control* memiliki batas tanggung jawab yang jelas. Ruang lingkupnya meliputi seluruh pengelolaan stok, audit, dan pemetaan lokasi penyimpanan barang di dalam gudang. Namun, konteks ini tidak mencakup aspek seperti pemrosesan pesanan (*Order Orchestration*), notifikasi operasional, ataupun pelaporan performa, karena fungsi-fungsi tersebut telah ditangani oleh konteks lain.

**Tabel 4.** Ruang Lingkup & Batasan Konteks

<b>Termasuk Dalam <i>Boundary</i></b>	<b>Di Luar <i>Boundary</i></b>
Pengelolaan stok dan lokasi penyimpanan	Pembuatan dan pengelolaan pesanan pelanggan
Perubahan dan audit stok	Analisis KPI dan pembuatan laporan
Reservasi stok dan pengaturan kapasitas ruang	Pengiriman notifikasi dan <i>alert</i>
Penerbitan event stok untuk konteks lain	Integrasi data eksternal (ERP, POS, dll.)

Dengan batasan tersebut, *Inventory Control* berperan sebagai pusat data dan logika stok yang dapat digunakan oleh konteks lain melalui *event* dan API yang telah distandarisasi, tanpa perlu saling bergantung secara langsung. Penegasan lingkup ini juga memastikan bahwa implementasi teknis pada tahap berikutnya dapat difokuskan pada pembentukan layanan inti, sementara pengembangan konteks lain dapat berkembang secara bertahap berdasarkan kebutuhan integrasi dan pemrosesan *event*.