

Platoo

A Smart Platform to Reduce Food Waste and Promote Sustainable Consumption



Rayhan Hidayatul Fikri
18223022
18223022@std.stei.itb.ac.id



Princessfa Azzahra Alvin
18223044
18223044@std.stei.itb.ac.id

Proposed by

Bat&Bat

DAFTAR ISI

DAFTAR ISI	2
A. Deskripsi Sistem	3
B. Bounded Context	4
C. Dekomposisi Subdomain dan Sub-subdomain	5

BAB 1

Pendahuluan

1. Latar Belakang

Food waste masih menjadi permasalahan serius dalam ekosistem pangan global, termasuk di sektor ritel dan kuliner. Berbagai laporan menunjukkan bahwa sebagian besar makanan terbuang bukan karena tidak layak konsumsi, melainkan akibat ketidakseimbangan antara produksi, distribusi, dan permintaan pasar. Makanan yang mendekati masa kedaluwarsa atau mengalami kelebihan stok seringkali berakhir sebagai limbah meskipun masih aman untuk dikonsumsi. Kondisi ini tidak hanya menimbulkan kerugian ekonomi bagi penyedia, tetapi juga berdampak pada lingkungan melalui pemborosan sumber daya, peningkatan emisi karbon, dan penumpukan sampah organik. Di sisi lain, masih terdapat kelompok masyarakat yang memiliki keterbatasan akses terhadap pangan dengan harga terjangkau. Kesenjangan inilah yang mendorong perlunya solusi sistemik berbasis teknologi.



Gambar 1. Logo Platoo

Platoo hadir sebagai *platform* digital yang dirancang untuk menjawab permasalahan *food waste* melalui distribusi dan pemanfaatan makanan tidak terjual. *Platform* ini menghubungkan restoran, minimarket, dan supermarket dengan konsumen dalam satu ekosistem terintegrasi. Penyedia dapat mempublikasikan makanan yang masih layak konsumsi namun berisiko tidak terjual, seperti produk mendekati masa kedaluwarsa atau stok berlebih. Setiap produk ditampilkan dalam katalog dengan informasi yang jelas dan terstruktur, mencakup nama, foto, harga, kualitas, jumlah, serta lokasi dan waktu pengambilan. Pendekatan berbasis data ini memungkinkan proses distribusi dilakukan secara lebih efisien dan transparan. Dengan demikian, Platoo berperan sebagai penghubung antara surplus pangan dan kebutuhan nyata di masyarakat.

Bagi pihak penyedia, Platoo memberikan fleksibilitas dalam mengelola dan mendistribusikan makanan surplus. Penyedia dapat menetapkan harga yang lebih rendah dari harga normal atau mendonasikan produk secara gratis sebagai bentuk kontribusi sosial. Opsi ini membantu mengurangi kerugian akibat stok berlebih sekaligus memaksimalkan nilai dari makanan yang sebelumnya berpotensi terbuang. Selain manfaat ekonomi, keterlibatan dalam Platoo juga berkontribusi

pada peningkatan citra positif penyedia sebagai entitas yang peduli terhadap isu lingkungan dan keberlanjutan. Seluruh proses pengelolaan produk dilakukan melalui sistem digital sehingga lebih mudah dipantau dan dikendalikan. Hal ini mendorong praktik bisnis yang lebih bertanggung jawab dan berkelanjutan.

Dari sisi konsumen, Platoo menyediakan akses terhadap katalog makanan yang dapat ditelusuri berdasarkan kategori, lokasi, maupun kondisi produk. Konsumen dapat memperoleh makanan dengan harga lebih terjangkau atau menerima donasi dengan hanya menanggung biaya pengiriman maupun mengambil langsung di lokasi. Transparansi informasi pada setiap produk memastikan konsumen memahami alasan di balik harga khusus atau skema distribusi gratis, misalnya karena mendekati masa kadaluarsa. Kejelasan ini membangun kepercayaan antara konsumen dan penyedia dalam proses transaksi. Melalui pemanfaatan makanan surplus, konsumen turut berperan aktif dalam mengurangi food waste. Secara keseluruhan, Platoo mendukung terciptanya ekosistem pangan yang lebih efisien, inklusif, dan berorientasi pada keberlanjutan lingkungan serta solidaritas sosial.

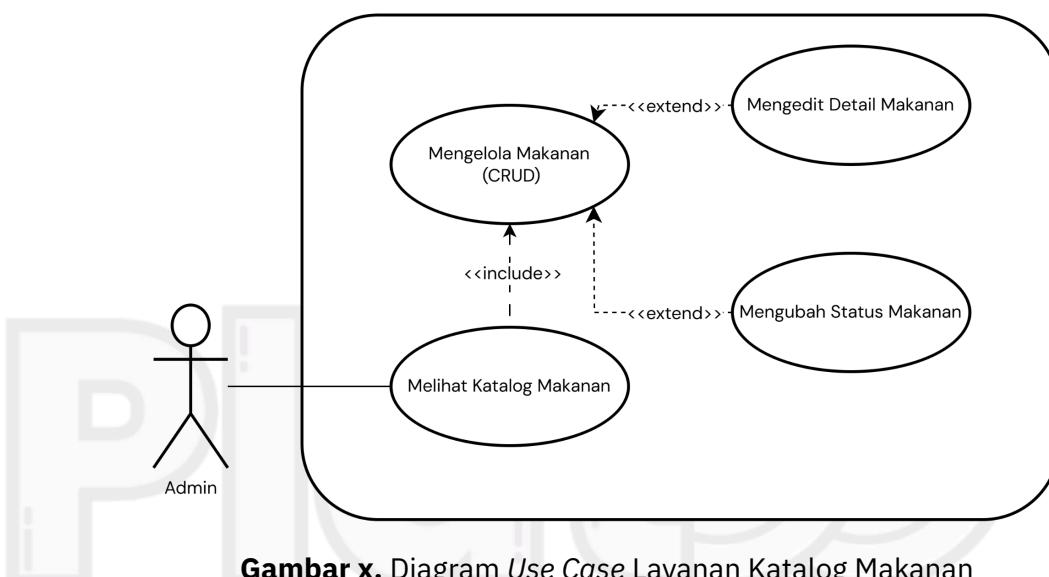
2. Deskripsi Sistem

2.1. Layanan Katalog Makanan

Core domain katalog makanan dalam aplikasi / *platform* digital Platoo menjadi salah satu pilar utama yang berfungsi sebagai etalase virtual bagi setiap restoran. Layanan ini dirancang untuk memberikan kemudahan serta akses penuh kepada pihak restoran dalam mengelola seluruh produk yang mereka tawarkan. Pengelolaan produk yang dimaksud meliputi penambahan menu baru, pembaruan informasi, hingga pengaturan ketersediaan makanan. Dengan *interface* yang intuitif, pihak restoran dapat dengan mudah menyajikan daftar makanan yang akurat dan menarik bagi calon pembeli. Keberadaan layanan ini krusial untuk operasional harian penjual dan menjadi sumber data utama bagi layanan lainnya dalam sistem Platoo.

Use case utama dari layanan katalog makanan adalah penyediaan data produk yang konsisten dan terpusat. Ketika seorang pembeli menjelajahi menu atau saat layanan lain dalam sistem membutuhkan detail item, maka sudah menjadi tugas dari layanan ini untuk memberi respons dengan informasi yang akurat. Permintaan tersebut akan mengambil data fundamental seperti nama makanan, harga, foto, stok, serta status ketersediaan. Layanan katalog makanan tidak menangani logika transaksi atau keranjang belanja, melainkan hanya bertindak sebagai satu-satunya sumber kebenaran untuk semua atribut produk. Layanan ini akan menghasilkan data terstruktur yang menjamin konsistensi informasi di seluruh *platform* digital Platoo.

Use case berikutnya adalah manajemen siklus hidup produk oleh penjual yang akan dieksekusi melalui *microservice* spesifik. Ketika pihak restoran mengubah detail menu, maka *microservice* ‘edit makanan’ akan menangani validasi dan pembaruan data seperti harga atau stok. Sementara itu, jika pihak restoran perlu mengubah ketersediaan produk, maka layanan ubah status makanan akan menyediakan *endpoint* untuk mengubah status item menjadi aktif atau nonaktif. Proses ini memastikan bahwa setiap perubahan yang dilakukan oleh pihak restoran segera tercermin pada seluruh sistem Platoo. Dengan memisahkan fungsi *read* dan *write*, integritas data katalog akan tetap terjaga dan pihak restoran tetap memiliki kendali penuh secara *real-time* terhadap etalase mereka.



Gambar x. Diagram Use Case Layanan Katalog Makanan

Dengan begitu, layanan katalog makanan berperan sebagai pondasi data yang menjaga integritas dan akurasi seluruh produk dalam sistem Platoo. Layanan ini tidak terlibat dalam logika transaksi atau interaksi pengguna, melainkan fokus sepenuhnya pada penyediaan data produk yang valid dan ter-update. Pemecahan fungsionalitasnya menjadi *microservice* ‘edit makanan’ dan ‘ubah status makanan’ memungkinkan manajemen data yang fleksibel dan responsif, sekaligus mengisolasi proses pembaruan yang kompleks dari perubahan status yang cepat. Layanan katalog makanan menjadi komponen kunci untuk memastikan informasi yang diterima pembeli dan layanan lainnya dapat diandalkan dan menopang seluruh alur operasional sistem Platoo.

2.2. Layanan Voucher

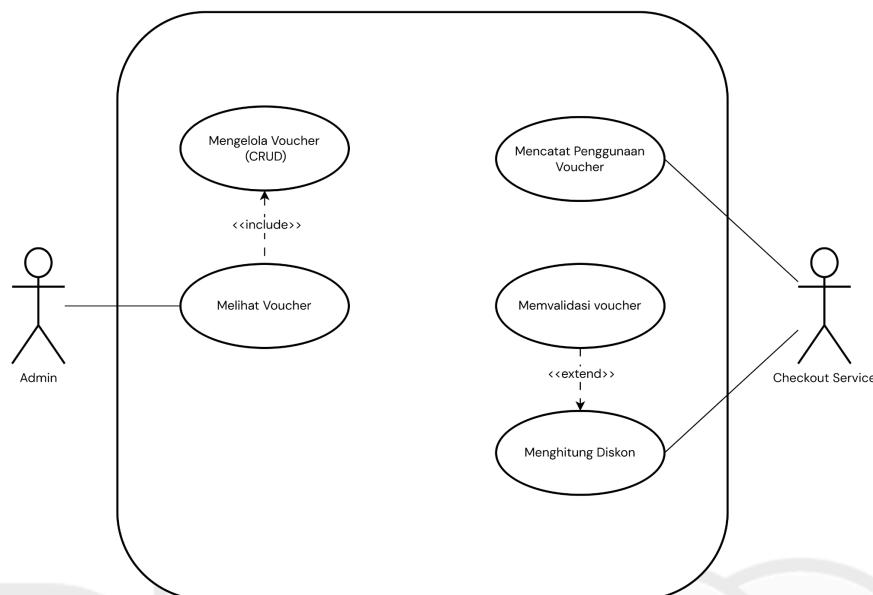
Core domain atau domain inti dari Voucher Service dalam sistem Plato adalah **pengelolaan** dan **penerapan aturan promosi** secara **konsisten** terhadap transaksi. Layanan ini bertanggung jawab untuk menentukan apakah suatu insentif dapat diterapkan pada konteks transaksi tertentu dan berapa besar manfaat yang diberikan. Fokus utama voucher bukan pada data produk atau transaksi, melainkan pada aturan (*rules*) dan keputusan (*decision making*) terkait promosi. Dengan demikian, Voucher Service berfungsi sebagai *policy engine* yang mengevaluasi kelayakan promosi berdasarkan kondisi yang telah ditetapkan. Core domain ini berfokus pada konsistensi logika, ketepatan evaluasi, serta keterlacakkan penggunaan voucher. Seluruh nilai bisnis voucher muncul dari kemampuan layanan ini dalam membuat keputusan diskon yang benar dan dapat dipertanggungjawabkan.

Sistem Voucher mengelola data dan logika yang berkaitan dengan kode voucher, periode berlaku, batas penggunaan, serta syarat penggunaan lainnya. Setiap voucher memiliki seperangkat aturan yang menentukan kapan dan bagaimana voucher tersebut dapat digunakan. Aturan ini dapat mencakup batas minimum transaksi, pembatasan jenis produk, atau pembatasan berdasarkan identitas pengguna. Sistem bertugas menyimpan dan menegakkan aturan tersebut secara konsisten pada setiap permintaan. Dengan pendekatan ini, logika promosi tidak tersebar di berbagai bagian sistem. Seluruh perubahan aturan diskon cukup dilakukan pada sistem Voucher tanpa mempengaruhi komponen lain.

Use case utama dari Voucher Service adalah validasi voucher dalam konteks transaksi. Ketika sistem menerima permintaan validasi, layanan ini mengevaluasi kode voucher terhadap aturan yang berlaku menggunakan data kontekstual yang diberikan, seperti subtotal transaksi, identitas pengguna, dan atribut item. Voucher Service tidak melakukan perhitungan harga dasar, melainkan hanya menentukan nilai diskon atau status penolakan. Hasil evaluasi dikembalikan dalam bentuk keputusan yang eksplisit, misalnya valid atau tidak valid, beserta alasan dan besaran diskon. Dengan pendekatan ini, seluruh keputusan promosi dipusatkan dalam satu layanan. Hal ini mencegah duplikasi logika promosi di berbagai bagian sistem.

Use case lanjutan dari Voucher Service adalah pencatatan dan pengendalian penggunaan voucher. Setelah suatu voucher dinyatakan valid dan digunakan dalam pembentukan order, layanan ini bertanggung jawab untuk mencatat penggunaan tersebut sebagai redemption. Proses ini memastikan bahwa kuota voucher berkurang secara tepat dan tidak dapat digunakan kembali secara tidak sah. Endpoint yang menangani proses ini harus bersifat idempotent untuk mencegah duplikasi akibat retry atau

kegagalan jaringan. Voucher Service juga memastikan bahwa voucher yang telah kedaluwarsa atau melebihi kuota tidak dapat digunakan kembali. Dengan mekanisme ini, integritas aturan promosi tetap terjaga sepanjang siklus transaksi.

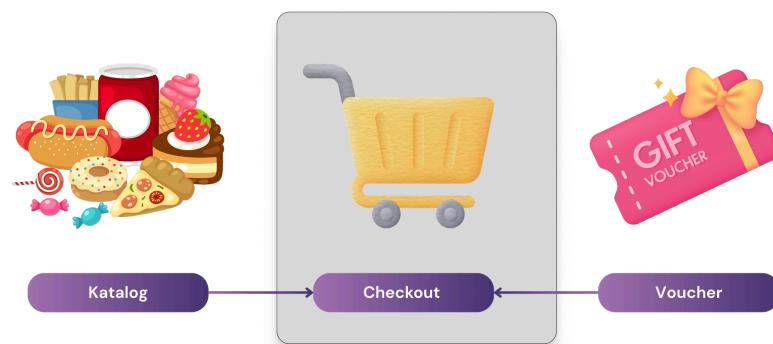


Gambar x. Diagram Use Case Voucher Service

Secara keseluruhan, core domain Voucher Service terletak pada penerjemahan aturan promosi menjadi keputusan sistem yang deterministik dan konsisten. Layanan ini tidak bergantung pada penyimpanan data katalog atau detail transaksi secara penuh, melainkan pada evaluasi konteks yang diberikan. Pemisahan Voucher Service sebagai microservice tersendiri memungkinkan pengembangan, pengujian, dan perubahan aturan promosi dilakukan secara independen. Selain itu, pendekatan ini mengurangi risiko kesalahan logika diskon yang dapat berdampak langsung pada nilai transaksi. Dengan desain tersebut, Voucher Service menjadi komponen strategis yang mengendalikan kebijakan promosi dalam sistem Platoo. Core domain ini memastikan bahwa setiap insentif yang diberikan sistem selalu sesuai dengan aturan bisnis yang telah ditetapkan.

2.3. Layanan Checkout

Checkout Service berperan sebagai layanan orkestrasi yang mengoordinasikan beberapa microservice untuk membentuk proses transaksi yang utuh. Layanan ini tidak memiliki kepemilikan penuh atas data katalog maupun aturan promosi, melainkan menggabungkan respons dari layanan lain untuk menghasilkan keputusan transaksi.



Gambar X. Ilustrasi Checkout

Pada tahap awal checkout, layanan ini menerima permintaan dari pengguna yang berisi item yang dipilih dan kode voucher opsional. Checkout Service kemudian memanggil Catalog Service untuk mengambil detail item dan harga dasar yang diperlukan dalam perhitungan transaksi. Data tersebut digunakan untuk menghitung subtotal dan memvalidasi ketersediaan item. Dengan pendekatan ini, tanggung jawab pengelolaan data produk tetap terisolasi pada Catalog Service.

Setelah semua hal tersebut berjalan, selanjutnya Checkout Service memanggil Voucher Service untuk melakukan validasi voucher dan perhitungan diskon berdasarkan konteks transaksi yang telah dibentuk. Voucher Service mengembalikan keputusan berupa status validitas voucher serta besaran diskon atau alasan penolakan. Checkout Service menggabungkan hasil tersebut dengan subtotal untuk menghasilkan harga akhir yang konsisten dan siap digunakan pada proses pembentukan order. Dengan menjadikan Checkout sebagai service terpisah, sistem menghindari ketergantungan langsung antara Catalog Service dan Voucher Service. Pendekatan ini memastikan pemisahan tanggung jawab yang jelas sekaligus memusatkan logika integrasi pada satu layanan. Checkout Service dengan demikian berfungsi sebagai penghubung utama antar microservice dalam alur transaksi.

Indikator dari terwujudnya arsitektur terdistribusi ini adalah ketika terdapat seorang pelanggan yang berinteraksi dengan aplikasi Platoo. Saat pelanggan membuka aplikasi untuk melihat penawaran, terdapat API gateway yang bertindak sebagai fasilitator utama yang meminta data dari

beberapa microservices. API gateway tersebut akan menghubungi layanan manajemen katalog makanan untuk mendapatkan daftar makanan yang tersedia beserta dengan harga dan stoknya. Pada saat yang bersamaan, API juga akan menghubungi layanan manajemen voucher untuk mengambil daftar promo aktif yang relevan. Informasi dari kedua layanan ini akan disajikan sebagai satu penawaran utuh kepada pelanggan.

BAB 2

Bounded Context

Pendekatan bounded context membantu untuk memahami domain bisnis dan menerjemahkannya karena setiap bounded context yang didefinisikan memiliki aturannya tersendiri. Selain itu, dalam aplikasi Plato, bounded context diperlukan untuk memecah domain besar menjadi bagian-bagian yang lebih kecil dan terkelola. Setiap bounded context memiliki batasan yang jelas dan model domain yang spesifik untuk konteks tersebut. Dua bounded context utama yang menjadi fokus pengembangan adalah katalog dan voucher, yang masing-masing menangani aspek bisnis yang berbeda namun saling melengkapi dalam ekosistem aplikasi Plato.

1. **Bounded Context Layanan Katalog Makanan**

Domain katalog makanan pada sistem Plato dipecah ke dalam beberapa bounded context yang terdefinisi dengan jelas untuk mengelola data produk secara sistematis dan memisahkan tanggung jawab yang berbeda. Pemisahan ini bertujuan untuk memastikan bahwa setiap aspek pengelolaan produk ditangani dalam lingkup yang terisolasi dan konsisten. Setiap bounded context memiliki peran, model data, dan aturan bisnisnya sendiri, namun tetap berinteraksi untuk membentuk layanan katalog yang utuh. Pendekatan ini memungkinkan setiap bagian untuk berkembang secara independen dan menjaga kejelasan makna dari setiap entitas data di dalamnya.

Bounded context yang pertama adalah **Catalog Management Context**, yang berfokus pada pengelolaan siklus hidup produk dari sisi administratif oleh penjual. Dalam konteks ini, ‘item makanan’ dimaknai sebagai sebuah entitas konfigurasi yang memiliki atribut statis seperti nama, deskripsi, harga dasar, dan foto. Aturan bisnis utamanya mencakup proses pembuatan item baru, penghapusan, serta pembaruan detail produk yang ditangani oleh *microservice* ‘edit makanan’. Model data yang dikelola adalah *FoodItem* dan *ItemAttribute*, yang menyimpan semua konfigurasi produk. Konteks ini tidak bertanggung jawab atas perubahan status ketersediaan secara *real-time* dan tidak terlibat dalam penyajian data kepada pembeli.

Bounded context berikutnya adalah **Catalog Availability Context**, yang bertanggung jawab atas status operasional produk yang bersifat dinamis dan sering berubah. Pada konteks ini, ‘item makanan’ memiliki status ketersediaan, seperti ‘aktif’ atau ‘nonaktif’. Aturan bisnis utama yang berlaku adalah memvalidasi dan mengeksekusi perubahan status yang ditangani oleh *microservice* ‘ubah status makanan’. Model data yang digunakan bersifat operasional, seperti *AvailabilityStatusUpdate*, yang hanya berisi ID item dan status barunya. Konteks ini

berinteraksi dengan *dashboard* penjual untuk menerima perintah perubahan status dan memastikan perubahan tersebut segera berlaku. Batasannya adalah konteks ini tidak mengubah atribut detail produk seperti harga atau nama.

Bounded context yang terakhir adalah **Catalog Presentation Context**, yang berfungsi sebagai *read-only* untuk menyajikan data produk kepada konsumen. Dalam konteks ini, "item makanan" berperan sebagai informasi produk yang siap ditampilkan, yang merupakan **gabungan data** dari konteks **Catalog Management** dan **Catalog Availability**. Aturan bisnisnya adalah menerapkan filter kemudian hanya menyajikan item yang berstatus 'aktif' kepada pembeli serta memastikan data yang ditampilkan selalu konsisten. Model data utamanya adalah *MenuItem*, sebuah DTO (Data Transfer Object) yang dioptimalkan untuk tampilan. Konteks ini berinteraksi dengan *frontend* pembeli dan layanan lain untuk menyediakan data katalog yang mereka butuhkan. Batasan dari konteks ini adalah konteks ini sama sekali tidak melakukan operasi tulis atau perubahan data apa pun.

2. Bounded Context Layanan Voucher

Untuk mendukung pengelolaan voucher secara terstruktur dan menjaga pemisahan tanggung jawab dalam sistem, domain voucher pada Plato dibagi ke dalam beberapa *bounded context* yang saling berhubungan. Pemisahan ini dilakukan untuk memastikan bahwa setiap aspek pengelolaan voucher, mulai dari konfigurasi, evaluasi penggunaan, hingga pencatatan pemakaian, ditangani dalam ruang lingkup yang jelas dan konsisten. Setiap *bounded context* memiliki peran dan tanggung jawab yang berbeda namun saling melengkapi dalam keseluruhan alur transaksi. Dengan pendekatan ini, perubahan pada satu bagian sistem voucher tidak akan berdampak langsung pada bagian lain yang tidak terkait. Selain itu, pemisahan bounded context membantu menjaga kejelasan makna istilah, aturan bisnis, serta kepemilikan data pada masing-masing bagian. Bagian berikut akan menjelaskan masing-masing bounded context tersebut secara lebih rinci.

Voucher Management Context merupakan bounded context yang menangani pengelolaan siklus hidup voucher dari sisi administratif. Dalam konteks ini, voucher dimaknai sebagai entitas promosi yang memiliki atribut konfigurasi seperti kode, status aktif/nonaktif, periode berlaku, kuota penggunaan, dan aturan dasar. Aturan bisnis pada konteks ini mencakup pembuatan, perubahan, dan penghapusan voucher, serta pengaturan parameter promosi sebelum voucher digunakan dalam transaksi. Model data utama yang dikelola adalah *Voucher* dan *VoucherRule*, yang menyimpan konfigurasi statis promosi. Konteks ini berinteraksi dengan sistem autentifikasi untuk memastikan hanya aktor berwenang yang dapat mengelola voucher. Batasan konteks ini adalah tidak melakukan validasi voucher terhadap transaksi dan tidak mencatat penggunaan voucher.

Bounded context yang kedua adalah **Voucher Validation Context**, yang berfokus pada evaluasi kelayakan voucher dalam konteks transaksi yang sedang

berlangsung. Dalam konteks ini, voucher dimaknai sebagai aturan yang harus diuji terhadap data transaksi seperti subtotal, atribut item, dan identitas pengguna. Aturan bisnis mencakup pemeriksaan status voucher, masa berlaku, ketersediaan kuota, serta kesesuaian terhadap syarat penggunaan. Model data yang digunakan bersifat operasional dan mencakup ValidationRequest dan ValidationResult, bukan data transaksi lengkap. Konteks ini berinteraksi dengan layanan checkout atau orchestrator untuk menerima konteks transaksi dan mengembalikan keputusan valid atau tidak valid beserta nilai diskon. Batasan dari konteks ini adalah tidak melakukan perubahan data permanen dan tidak mencatat penggunaan voucher.

Setelah proses validasi berhasil, bounded context yang terakhir adalah **Voucher Redemption Context**, yang bertanggung jawab atas pencatatan dan pengendalian penggunaan voucher setelah transaksi dikonfirmasi. Dalam konteks ini, istilah redemption dimaknai sebagai penggunaan voucher yang telah disetujui dan dikaitkan dengan transaksi tertentu. Aturan bisnis mencakup pengurangan kuota, pencegahan penggunaan ganda, serta penanganan kondisi idempotency untuk menghindari pencatatan ganda akibat retry. Model data utama pada konteks ini adalah Redemption, yang menyimpan referensi voucher, transaksi, waktu penggunaan, dan status. Konteks ini berinteraksi dengan layanan checkout atau order untuk menerima sinyal bahwa transaksi berhasil. Batasan konteks ini adalah tidak mengevaluasi kelayakan voucher dan tidak memodifikasi konfigurasi voucher.

Bab 3

Dekomposisi Subdomain dan Sub-subdomain

Berdasarkan identifikasi **core domain** sebelumnya, selanjutnya *core domain* tersebut akan didekomposisi ke dalam beberapa subdomain yang memiliki peran dan tingkat kepentingan yang berbeda. Dekomposisi subdomain dilakukan untuk memisahkan area masalah berdasarkan fokus tanggung jawab dan kompleksitas aturan bisnis yang ditangani. Setiap subdomain kemudian diklasifikasikan ke dalam kategori **core, supporting, atau generic** sesuai dengan kontribusinya terhadap nilai utama sistem.

1. Dekomposisi Layanan Katalog Makanan

Sesuai dengan *core domain* yang telah diidentifikasi sebelumnya, layanan katalog makanan didekomposisi ke dalam beberapa subdomain. Pemecahan ini bertujuan untuk memisahkan area tanggung jawab yang berbeda, mulai dari definisi data produk, kontrol ketersediaan, hingga cara data tersebut disajikan kepada pengguna. Setiap subdomain diklasifikasikan sebagai **core, supporting, atau generic** untuk merefleksikan kontribusinya terhadap nilai bisnis utama layanan. Dengan struktur ini, kompleksitas sistem dapat dikelola secara efektif, memastikan setiap bagian memiliki fokus yang jelas dan dapat dikembangkan secara independen.

Subdomain *product information* dan *product availability* dikategorikan sebagai core subdomain karena keduanya merupakan jantung dari layanan katalog. *Product information* bertanggung jawab atas data-data fundamental produk seperti nama dan harga, yang di dalamnya terdapat sub-subdomain *item definition* sebagai tempat logika *microservice* ‘edit makanan’ beroperasi. Sementara itu, terdapat subdomain *product availability* yang fokus pada data dinamis, dengan sub-subdomain *stock tracking* dan *status control* yang menjadi domain untuk *microservice* ‘ubah status makanan’. Kedua subdomain memastikan integritas dan akurasi data produk untuk seluruh transaksi di sistem Platoo.

Subdomain lainnya berperan sebagai pendukung dan pelengkap bagi *core subdomain*. *Catalog presentation* diklasifikasikan sebagai *supporting subdomain* karena fungsinya adalah menyajikan data produk yang sudah valid kepada pengguna akhir tanpa mengubah data itu sendiri. Di sisi lain, terdapat *digital asset handling* yang dikategorikan sebagai *generic subdomain* karena menyediakan fungsi umum untuk penyimpanan dan pengiriman gambar, yang tidak spesifik untuk domain katalog. Pemisahan ini memastikan bahwa logika bisnis inti terkait data produk tidak tercampur dengan logika presentasi atau teknis pengelolaan file. Dibawah ini

merupakan diagram hasil dekomposisi subdomain dan sub-subdomain yang telah didefinisikan sebelumnya.



Gambar X. Diagram Dekomposisi Domain Layanan Katalog Makanan

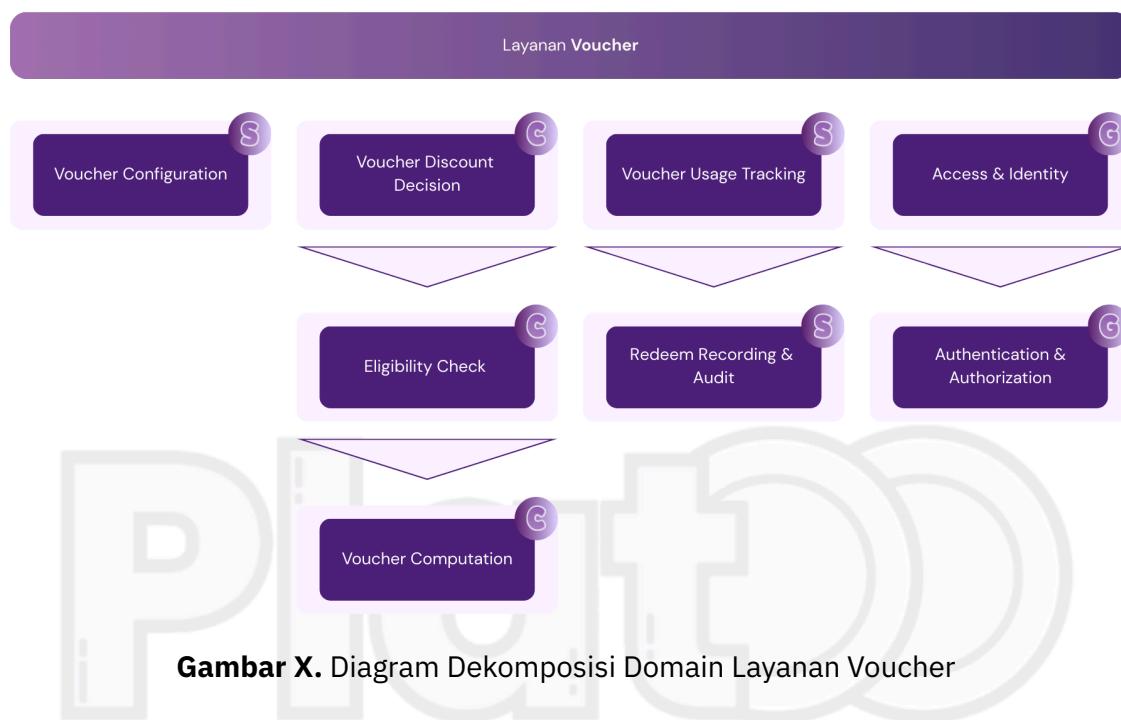
2. Dekomposisi Layanan Voucher

Berdasarkan identifikasi core domain sebelumnya, yaitu pengambilan keputusan diskon voucher dalam konteks transaksi, layanan voucher didekomposisi ke dalam beberapa subdomain dengan tingkat kepentingan yang berbeda. Dekomposisi ini bertujuan untuk memisahkan area tanggung jawab berdasarkan fokus logika bisnis yang ditangani oleh sistem. Setiap subdomain diklasifikasikan sebagai core, supporting, atau generic sesuai dengan kontribusinya terhadap nilai utama layanan voucher. Dengan pendekatan ini, kompleksitas sistem dapat dikelola secara lebih terstruktur tanpa mencampuradukkan tanggung jawab antar bagian. Dekomposisi dilakukan secara sederhana agar tetap sesuai dengan skala dan kebutuhan sistem.

Subdomain Voucher Discount Decision dikategorikan sebagai core subdomain karena menjadi pusat nilai utama layanan voucher. Subdomain ini bertanggung jawab dalam menentukan apakah suatu voucher dapat digunakan serta menghitung besaran diskon yang dihasilkan. Di dalamnya terdapat fungsi Eligibility Check dan Voucher Computation yang mendukung proses evaluasi kelayakan dan perhitungan diskon secara terintegrasi. Keputusan yang dihasilkan oleh subdomain ini secara langsung memengaruhi hasil transaksi pengguna. Oleh karena itu, subdomain ini menjadi fokus utama dalam keseluruhan layanan voucher.

Subdomain lain berperan sebagai pendukung dan pelengkap bagi core subdomain. Voucher Configuration dan Voucher Usage Tracking diklasifikasikan sebagai supporting subdomain, karena masing-masing menangani konfigurasi

voucher serta pencatatan dan pengendalian penggunaan voucher setelah transaksi berhasil. Sementara itu, Access & Identity dikategorikan sebagai generic subdomain karena menyediakan fungsi autentikasi dan otorisasi yang bersifat umum dan dapat digunakan lintas domain. Pemisahan ini memastikan bahwa perubahan pada konfigurasi, pencatatan penggunaan, atau mekanisme akses tidak berdampak langsung pada logika inti pengambilan keputusan diskon. Dekomposisi tersebut digambarkan pada diagram berikut.



Gambar X. Diagram Dekomposisi Domain Layanan Voucher

Bab 4

Diagram dan Antarmuka

