

PENGGUNAAN “API” PADA INTEGRASI DATA RETAIL GUNA PENINGKATAN EFISIENSI OPERASIONAL

Eko Hardi¹

Universitas Respati Indonesia

Email : Ekohardis60@gmail.com

Abstrak

Perusahaan ritel semakin menghadapi tantangan dalam mengelola data bervolume besar yang dihasilkan dari berbagai sumber seperti sistem POS, platform e-commerce, dan perangkat lunak ERP. Pendekatan integrasi tradisional yang menggunakan pemrosesan batch dan impor/ekspor manual tidak efisien untuk operasi waktu nyata (real-time). Makalah ini mengkaji peran Antarmuka Pemrograman Aplikasi (API) dalam memungkinkan integrasi data ritel yang lancar dan real-time. Melalui analisis kualitatif dan teknis terhadap arsitektur ritel, studi ini menunjukkan bagaimana API meningkatkan aksesibilitas data, konsistensi, dan kelincahan operasional. Sebuah kerangka kerja yang diusulkan untuk integrasi berbasis API disajikan, diikuti dengan diskusi tentang tantangan implementasi dan praktik terbaik.

Kata kunci: Retail, API, Integrasi Data, POS, ERP, Waktu nyata, Transformasi Digital

Abstract

Retail enterprises increasingly face the challenge of managing large volumes of data generated from multiple sources such as POS systems, e-commerce platforms, and ERP software. Traditional integration approaches using batch processing and manual import/export are inefficient for real-time operations. This paper investigates the role of Application Programming Interfaces (APIs) in enabling seamless, real-time retail data integration. Through a qualitative and technical analysis of retail architecture, this study demonstrates how APIs improve data accessibility, consistency, and operational agility. A proposed framework for API-based integration is presented, followed by a discussion on implementation challenges and best practices.

Keywords : Retail, API, Data Integration, POS, ERP, Real-time, Digital Transformation

1. PENDAHULUAN

Lanskap ritel modern berkembang pesat berkat transformasi digital, strategi omnichannel, dan meningkatnya permintaan konsumen. Data merupakan inti dari transformasi ini. Para peritel mengandalkan data yang akurat dan tepat waktu dari berbagai sistem—Point of Sale (POS), Manajemen Gudang, Manajemen Hubungan Pelanggan (CRM), dan Perencanaan Sumber Daya Perusahaan (ERP)—untuk membuat keputusan yang tepat.

Namun, mengintegrasikan data dari berbagai sumber yang heterogen ini masih menjadi tantangan besar. Pendekatan ETL (Ekstrak,

Transformasi, Load) tradisional dan berbasis berkas tidak memadai untuk kebutuhan waktu nyata (real-time). Studi ini mengeksplorasi penggunaan Antarmuka Pemrograman Aplikasi (API) sebagai solusi untuk integrasi data ritel yang efisien, aman, dan skalabel.

2. TINJAUAN PUSTAKA

TANTANGAN DATA RETAIL

Data ritel biasanya terisolasi di berbagai platform. Menurut IBM (2021), lebih dari 70% perusahaan ritel menyebutkan integrasi data sebagai hambatan utama transformasi digital.

Dalam lanskap industri ritel yang dinamis saat ini, aliran data yang lancar di berbagai titik kontak sangat penting untuk memberikan kepuasan pelanggan yang tak tertandingi.

Bagaimana Anda mengelola bisnis yang sangat terdistribusi dan beroperasi dengan ratusan toko yang tersebar di seluruh wilayah atau bahkan global?

Tidak masalah apakah perusahaan tersebut merupakan bisnis waralaba makanan cepat saji, toko kelontong, toko obat, atau peritel besar. Mereka semua memiliki masalah data yang serupa.

Ketika data tersebar di berbagai sistem, disimpan dalam format yang berbeda, atau mengandung inkonsistensi, hal ini menghambat proses pengambilan keputusan, menghambat efisiensi operasional, dan melemahkan kinerja organisasi. Data yang terfragmentasi dan inkonsisten menghadirkan tantangan signifikan bagi manajemen yang efisien dan mendorong pertumbuhan.

Tantangan tersebut meliputi:

- Aksesibilitas Data yang Terbatas
- Data yang tersebar di berbagai sistem dan silo menyulitkan tim untuk mengakses informasi yang mereka butuhkan saat dibutuhkan.
- Inkonsistensi Data
- Tanpa solusi terpadu, inkonsistensi dalam format, definisi, dan kualitas data sering terjadi.
- Kurangnya Visibilitas Data
- Tidak adanya solusi, menyulitkan untuk memiliki pandangan holistik terhadap data organisasi, yang menyebabkan *blind spot* dan hilangnya peluang.
- Manajemen Data yang Tidak Efisien

Mengelola data yang tersebar di berbagai sistem membutuhkan upaya yang berulang-ulang sehingga mengakibatkan inefisiensi dan peningkatan biaya operasional.

Dalam skenario di mana data tersebar dan kurang koheren, para pengambil keputusan sering kali kesulitan untuk mengakses informasi yang akurat dan terkini. Kurangnya data yang andal ini menyebabkan keterlambatan dalam pengambilan keputusan karena manajer menghabiskan waktu berharga untuk merekonsiliasi sumber data yang saling bertentangan atau mencoba memverifikasi keakuratan informasi.

Lebih lanjut, inkonsistensi data dapat menyebabkan kesalahan dalam proses operasional. Misalnya, sistem manajemen inventaris dapat melaporkan tingkat stok yang saling bertentangan, yang mengakibatkan kelebihan stok atau kehabisan stok. Demikian pula, prakiraan penjualan berdasarkan data yang konsisten dapat menghasilkan perencanaan produksi yang akurat, yang menyebabkan inefisiensi dalam rantai pasokan.

Di era di mana klien selalu meningkatkan standar ekspektasi mereka, pengambilan keputusan yang buruk dan operasional yang tidak efisien dapat menyebabkan ketidakpuasan pelanggan.

PERAN API

API adalah singkatan dari **Application Programming Interface**, yang dalam Bahasa Indonesia disebut **Antarmuka Pemrograman Aplikasi**.

Definisi Umum:

API adalah sekumpulan aturan, protokol, dan alat yang memungkinkan satu aplikasi berkomunikasi dengan aplikasi lainnya. API memungkinkan sistem yang berbeda (misalnya software, database, atau layanan web) untuk saling bertukar data atau fungsi tanpa perlu mengetahui cara kerja internal satu sama lain.

Contoh Sederhana:

Bayangkan API seperti **pelayan di restoran**. Kamu (aplikasi pengguna) memesan makanan dari menu (permintaan data atau layanan), pelayan (API) menyampaikan pesanmu ke dapur (server), lalu pelayan mengantarkan makanan ke meja kamu (hasil data/laporan/layanan yang diminta).

Jenis-Jenis API:

1. **Web API** – Digunakan untuk komunikasi antar aplikasi melalui internet (contoh: REST API, SOAP API).

2. **Library API** – Menyediakan fungsi siap pakai dalam suatu pustaka kode.
3. **Operating System API** – API yang digunakan untuk berinteraksi dengan sistem operasi (contoh: Windows API).
4. **Hardware API** – Digunakan untuk berinteraksi dengan perangkat keras seperti printer atau kamera.

Contoh Penggunaan dalam Dunia Nyata:

- Aplikasi cuaca mengambil data prakiraan dari API milik BMKG.
- Aplikasi e-commerce mengambil data ongkir dari API ekspedisi (JNE, J&T, dll).
- Sistem POS retail mengirimkan data penjualan ke ERP SAP melalui API.

API menyediakan antarmuka standar untuk pertukaran data. API RESTful dan GraphQL telah muncul sebagai protokol umum untuk memfasilitasi interaksi antara layanan berbasis cloud, sistem on-premise, dan aplikasi seluler (Fielding, 2000; Microsoft, 2022).

Peran API (Antarmuka Pemrograman Aplikasi) dalam integrasi data ritel sangat kritis untuk memungkinkan komunikasi real-time, interoperabilitas, otomatisasi, dan skalabilitas di seluruh sistem seperti POS (Point of Sale), ERP, CRM, manajemen inventaris, platform e-commerce, dan lainnya. Berikut rincian peran utamanya:

1. Konektivitas Sistem yang Lancar
2. Sinkronisasi Data Real-Time
3. Integrasi Omnichannel
4. Otomatisasi Proses Bisnis
5. Pengalaman Pelanggan yang Lebih Baik
6. Skalabilitas dan Fleksibilitas
7. Analisis dan Pelaporan

KASUS PENGGUNAAN DALAM RITEL

API (Application Programming Interface) dalam konteks ritel adalah antarmuka yang memungkinkan sistem-sistem berbeda dalam ekosistem ritel (seperti POS, e-commerce, gudang, ERP, CRM, dan sistem pembayaran) untuk saling terhubung dan bertukar data secara otomatis, real-time, dan efisien.

Dalam industri ritel, API digunakan untuk mengintegrasikan berbagai sistem operasional — mulai dari transaksi di kasir (POS), manajemen stok, hingga pelaporan keuangan dan pengalaman pelanggan — guna menciptakan arsitektur digital yang saling terhubung dan responsif terhadap kebutuhan pasar.

Contoh Penggunaan API di Retail:

1. **Integrasi POS dan ERP:** API mengirimkan data transaksi penjualan harian dari kasir ke sistem ERP seperti SAP untuk mencatat revenue dan stok.
2. **Integrasi Marketplace dan Sistem Toko:** API menghubungkan sistem inventori toko fisik dengan e-commerce seperti Tokopedia atau Shopee.
3. **Integrasi Sistem Loyalty:** Saat pelanggan berbelanja, sistem POS memanggil API ke sistem loyalty untuk memberikan atau menukar poin.
4. **Real-Time Stock Update:** Saat barang terjual di satu cabang, API memperbarui stok pusat secara otomatis.

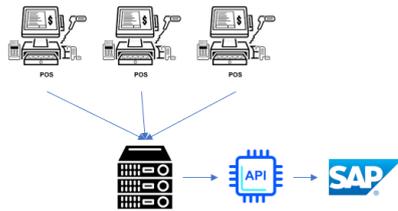
Peritel seperti Walmart dan Amazon menggunakan API untuk mengintegrasikan aplikasi pihak ketiga, sistem inventaris, gateway pembayaran, dan layanan logistik. SAP, Oracle Retail, dan Microsoft Dynamics menawarkan konektor API untuk meningkatkan integrasi ERP dengan platform POS dan e-commerce.

3. METODOLOGI

Studi ini menggunakan pendekatan studi kasus penelitian lapangan dengan membuat simulasi. Studi kasus menguji prototipe pada sebuah bisnis ritel supermarket. Penelitian ini merupakan penelitian kualitatif yang didukung oleh prototipe teknis. Arsitektur ritel disimulasikan dengan komponen-komponen berikut:

- Aplikasi POS berbasis database MS SQL dan operating system Microsoft
- Backend berbasis ERP
- Middleware menggunakan API RESTful

Simulasi kasus menggunakan 3(tiga) mesin kasir dan 1 server induk dengan model prototipe arsitektur pada gambar-1.



Gambar-1 Prototipe Arsitektur API Retail

Simulasi digunakan untuk menguji keberhasilan integrasi menggunakan framework API. Keberhasilan diukur berdasarkan tingkat ketepatan dan konsistensi data sinkronisasi antara POS, Middleware API dan ERP

HASIL DAN PEMBAHASAN

Dalam simulasi , dibentuk integrasi yang terdiri dari empat lapisan , yakni :

1. Lapisan Sumber Data (POS, Online Store)
2. Lapisan API Gateway/Middleware
3. Lapisan Logik Bisnis (Transformasi data , Validasi)
4. Lapisan ERP/Basis Data

Komponen simulasi data integrasi retail menggunakan API sebanyak 4 (empat) lapisan, sebagaimana tergambar di gambar.2



Gambar-2 Lapisan Integrasi

Detail proses adalah sebagai berikut :

- Lapisan 1
Merupakan lapisan awal sumber *data touch point*. Secara umum pengumpulan data awal dalam retail baik transaksi di toko atau melalui *e-commerce* diawali di titik ini. Data yang di himpun berupa data penjualan dan inventori. Format data heterogen tergantung pada masing2 jenis *Point Of Sales*
- Lapisan 2
Merupakan lapisan mekanisme penghubung data menggunakan API.
Data API yang di pertukarkan antara lain :

- **POST /Penjualan:** Kirim transaksi data penjualan dari POS ke ERP
- **GET /Master Data:** Akses master data dari ERP
- **PUT /Status Pembayaran:** Update Konfirmasi Pembayaran

Dalam simulasi ini API data penjualan di tulis dalam format JSON.

```
{
  "order_id": "POS123456",
  "date": "2025-07-30",
  "items": [
    {"sku": "A123", "qty": 2, "price": 100}
  ],
  "payment": {"method": "cash", "amount": 200}
}
```

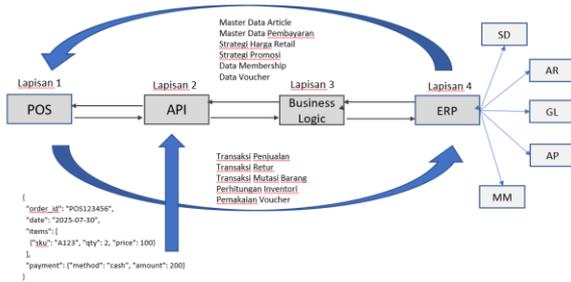
- Lapisan 3
Lapisan ini akan mengolah data sesuai dengan kebutuhan bisnis. Algoritma akan disiapkan unt mengolah data API dari format JSON ke *Staging Table* sebelum data masuk ke ERP di lapisan 4 (Empat).
Proses algoritma dapat meliputi :
 - Konsolidasi Data
 - Validasi Data
 - Agregasi Data
 - Simulasi Data

Tahap ini merupakan tahap krusial, dimana hasil data akan menjadi sumber data induk di lapisan 4(empat) , yakni berupa ERP system.

- Lapisan 4
Lapisan terakhir, yakni sebagai muara data dari toko dan terkumpul di sistem ERP. Sampai tahap ini data dianggap sudah valid dan sudah ter agregasi untuk kebutuhan pelaporan.

PEMODELAN API RETAIL

Sebagaimana dibahas bahwa simulasi penggunaan API pada industri retail terdiri dari 4 (empat) tahap. Masing-masing tahap mempunyai siklus proses yang saling terkait satu dengan lainnya. Pemodelan API di Retail disimulasikan pada Gambar-3.



Gambar-3 Pemodelan API

Pemodelan API retail yang di uji-coba adalah menggunakan pendekatan 2 arah

1. Outbound Proses
2. Inbound Proses

Inbound merupakan aliran data dari system ERP di back-end system menuju ke front-end POS. Data tersebut akan dipakai sebagai pendukung operasional system POS yang ada di setiap kasir di toko. Inbound data antara lain terdiri dari :

- Master data article barang
- Master data Jenis Pembayaran
- Master Data Vendor
- Master Data Kartu Membership
- Master Data Voucher

Outbound merupakan aliran data dari system front-end POS menuju ERP backend. Data ini dikirim dari semua mesin kasir yang ada pada masing-masing toko. Data tersebut antara lain :

- Transaksi Penjualan
- Transaksi Retur Penjualan
- Transaksi Transfer Antar Toko
- Transaksi Penerimaan Barang
- Transaksi Pencacatan Voucher

PROTOTYPE API - OUTBOUND

Prototipe ini memberikan ilustrasi jenis data outbound yang dikirimkan dari ERP sistem ke POS sistem.

Bentukan detail yang dikirimkan adalah :

Master Data	SAP Table/Source	Deskripsi
Material (Product)	MARA, MAKT, MWKE	Info produk, nama, kategori
Harga	KONV, VKPMA	Harga jual, diskon, promosi
Pelanggan	KNA1, KKNV	Info pelanggan (jika relevan)
Stok	MBEW, MSKU	Informasi stok di lokasi
Unit of Measure	MARM	Konversi satuan produk
Tax Data	T007A, TAX0K	Informasi pajak produk

Data ini terdiri dari Material produk, harga, pelanggan, stock, UoM dan tax data.

Bentukan JSON file adalah :

```

{
  "material_id": "MAT001",
  "material_name": "Sabun Mandi 100gr",
  "material_group": "FG001",
  "material_type": "FERT",
  "uom": "PCS",
  "description": {
    "en": "Bath Soap 100gr",
    "id": "Sabun Mandi 100gr"
  },
  "price": {
    "currency": "IDR",
    "standard_price": 3500,
    "sales_price": 5000,
    "discount": 0.1
  },
  "tax_code": "PPN10",
  "availability": {
    "plant": "1001",
    "stock": 150,
    "stock_unit": "PCS"
  }
}
    
```

Data outbound akan dikirimkan dari ERP ke POS melalui *program job scheduling* yang akan dijalankan di tengah malam mulai jam 00.00 sd 03.00, menunggu saat tidak ada *pick season* di jalur server dan komunikasi data.

PROTOTYPE API - INBOUND

Prototipe ini memberikan ilustrasi data inbound yang dikirimkan dari POS ke ERP.

Bentukan detail data yang dikirim : Sales Order/Billing, Pembayaran pelanggan, Update Stok, Mutasi stock antar toko.

Bentukan JSON file :

```

json
{
  "transaction_id": "TRX202507301001",
  "store_id": "STORE001",
  "pos_terminal": "POS02",
  "cashier_id": "CASH1001",
  "transaction_datetime": "2025-07-30T14:25:00+07:00",
  "items": [
    {
      "material_id": "MAT001",
      "quantity": 2,
      "uom": "PCS",
      "unit_price": 5000,
      "discount": 0
    },
    {
      "material_id": "MAT002",
      "quantity": 1,
      "uom": "PCS",
      "unit_price": 15000,
      "discount": 1500
    }
  ]
}
    
```

Data inbound secara umum pada industri retail akan dikirim 1 (satu) kali sehari pada malam hari ketika toko sudah tutup. Prosesnya akan di trigger melalui *program background job* yang dijalankan mulai toko tutup 22.00 – 00.00

Struktur lengkap model simulasi JSON untuk pengiriman transaksi penjualan dari POS ke ERP adalah :

```
{
  "transaction_id": "TRX202507301001",
  "store_id": "STORE001",
  "pos_terminal": "POS02",
  "cashier_id": "CASH1001",
  "transaction_date": "2025-07-30",
  "transaction_time": "14:25:00",
  "currency": "IDR",
  "items": [
    {
      "material_id": "MAT001",
      "material_name": "Sabun Mandi 100gr",
      "quantity": 2,
      "unit": "PCS",
      "unit_price": 5000,
      "discount": 0,
      "tax_code": "PPN10"
    },
    {
      "material_id": "MAT002",
      "material_name": "Shampoo 200ml",
      "quantity": 1,
      "unit": "PCS",
      "unit_price": 15000,
      "discount": 1500,
      "tax_code": "PPN10"
    }
  ],
  "totals": {
    "subtotal": 25000,
    "tax": 2500,
    "total": 27500
  },
  "payment": {
    "payment_method": "QRIS",
    "amount_paid": 27500,
    "reference_number": "QR123456789"
  },
  "customer": {
    "customer_id": "CUST001",
    "loyalty_points_used": 50,
    "customer_name": "Budi Santoso"
  }
}
```

}

PROSES AKHIR KONSOLIDASI DATA

Rangkaian akhir dari proses simulasi API adalah melakukan konsolidasi data di tingkat *staging server*. Pada tingkatan ini seluruh data yang telah masuk akan divalidasi, dikonsolidasi dan di agregasi.

- Validasi : menguji keakuratan data antara POS, Middleware (API) dan ERP. Ketiga sumber data harus sama baik dari struktur *data header* maupun *data detail*
- Konsolidasi : Seluruh data took akan di *merging* dalam satu *staging table* , setelah diberikan tanda pengenal asal data dan waktu perekaman data.
- Agregasi. Merupakan proses mengumpulkan data yang sejenis menjadi satu record data dan di agregasi. Tujuannya adalah untuk mendapatkan kelengkapan data di level ikhtisar. Simulasi pemodelan agregasi dapat dilihat pada Table-1.

Detail Level					
Struk #	Kode	Nama	Jumlah	Harga	Total
001	A-1	Aqua	10	500	5.000
002	B-1	Roti	5	1.000	5.000
003	A-1	Aqua	5	500	2.500
004	B-1	Roti	5	1.000	5.000
		Total			17.500
Agregasi Level					
	A-1	Aqua	15	500	7.500
	B-1	Roti	10	1.000	10.000
		Total			17.500

Tabel-1 Simulasi Pemodelan Agregasi

KEUNGGULAN DAN KELEMAHAN

Berdasarkan penelitian ini dapat diketahui keunggulan dan kelemahan penggunaan data integrasi retail menggunakan API.

Keunggulan :

No	Keunggulan	Penjelasan
1	Real-time Data Exchange	Data bisa dikirim dan diterima langsung saat terjadi transaksi (misalnya penjualan di POS langsung tercatat di SAP).
2	Fleksibel dan Terbuka	API dapat diakses oleh berbagai platform (web, mobile, cloud, desktop).

No	Keunggulan	Penjelasan
3	Modular dan Scalable	Sistem bisa dikembangkan atau diintegrasikan secara bertahap sesuai kebutuhan.
4	Mudah Diotomatisasi dan Dijadwalkan	Mudah mengatur proses integrasi berkala atau otomatis (misalnya tiap jam atau harian).
5	Dukungan Standar Modern (JSON/REST)	Komunikasi antar sistem lebih mudah dipahami dan dibangun karena menggunakan protokol terbuka seperti HTTP, JSON, XML.
6	Audit & Monitoring Mudah	Aktivitas API bisa dilacak dan dimonitor dengan tools modern (API Gateway, log server).
7	Integrasi Antar Vendor Lebih Mudah	Sistem dari vendor berbeda bisa dihubungkan tanpa harus terlalu tergantung pada format atau protokol lama seperti IDoc.

Kelemahan :

No	Kelemahan	Penjelasan
1	Ketergantungan pada Konektivitas	API membutuhkan koneksi jaringan stabil, terutama untuk real-time. Jika offline, data tidak bisa terkirim.
2	Kompleksitas Keamanan	Harus mengelola otentikasi, otorisasi, dan enkripsi agar data tidak bocor.
3	Perubahan API bisa berdampak besar	Jika ada perubahan di struktur API (misalnya field baru, field dihapus), bisa menyebabkan error di sisi konsumen API.
4	Pemeliharaan dan Versi	API perlu didokumentasikan dan diberi versi (versioning) agar integrasi tetap kompatibel.
5	Beban Infrastruktur Tambahan	Untuk volume besar, API butuh server, load balancer, dan monitoring tools agar tetap responsif.
6	Error Handling Harus Dirancang Baik	Jika tidak ada log & mekanisme retry, kegagalan pengiriman data bisa hilang tanpa jejak.
7	Konsistensi Data Bisa Terganggu	Jika tidak dirancang dengan baik, bisa terjadi duplikasi, data hilang, atau inkonsistensi antar sistem.

KESIMPULAN DAN SARAN

KESIMPULAN

Berdasarkan simulasi dari prototipe yang telah dilakukan selama penelitian, dapat disimpulkan :

- Integration retail berbasis API merupakan salah satu alternative pilihan model pengiriman data dalam dunia retail
- Solusi ini dapat dipergunakan dan diterapkan bagi industri retail offline (store based) maupun online dan ecommerce
- Dengan diterapkan simulasi model ini maka tingkat akurasi dan efisiensi bisnis retail dapat meningkat. Disisi lain proses pengambilan keputusan menjadi lebih cepat, dikarenakan ketersediaan data tepat waktu.

SARAN

Mengingat penelitian ini bersifat simulasi dengan model uji coba sampel data sangat sedikit, dengan model arsitektur retail yang sangat sederhana, maka hasilnya tidak dapat menjadi representasi dari dunia retail secara menyeluruh. Tetap diperlukan penelitian lebih jauh, secara comprehensive dengan prototipe arsitektur yang lebih kompleks, dan sampel data yang lebih banya, sehingga akan dapat memberikan gambaran lebih komprehensif dan akurat.

DAFTAR PUSTAKA

1. Leonard Richardson, Mike Amundsen, dan Sam Ruby, RESTful WEB API, O'Reily Media, ISBN-1449358068
2. Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation, University of California, Irvine).
3. IBM. (2021). *The State of Data Integration in Retail*.
4. Microsoft. (2022). *Designing RESTful APIs*.
5. SAP SE. (2023). *SAP API Hub Documentation*.

7. Oracle. (2023). *Oracle Retail Integration Cloud Services*.
8. <https://mia-platform.eu/blog/retail-data-integration/>
9. <https://developer.mozilla.org/en-US/docs/Glossary/API>