

MODEL VISUALISASI LAPORAN UNTUK PENGAMBILAN KEPUTUSAN BISNIS MENGUNAKAN LARAVEL DAN CHART.JS PADA PT. XYZ

BUSINESS DECISION MAKING REPORT VISUALIZATION USING LARAVEL AND CHART.JS IN PT. XYZ

**Sultan Said Abdul Rahim¹, Jap Tji Beng^{1,2*}, Desi Arisandi¹, Rahmiyana Nurkholiza², Denis
Mathew¹, Nathannael Dennis Gunawan¹**

Fakultas Teknologi Informasi, Program Studi Sarjana Sistem Informasi, Universitas Tarumanagara¹
Laboratorium Kognisi Edukasi dan Inovasi Teknologi, Fakultas Psikologi, Universitas Tarumanagara²
t.jap@untar.ac.id*

ABSTRACT

This study addresses the challenge of interpreting raw inventory and transaction data that often fails to provide useful business insights. A visualization model was developed to transform operational data into interactive charts that support strategic decision-making at PT XYZ. The system was built using the SDLC Waterfall model, with Laravel handling data processing and Chart.js providing interactive visual output. The results show that the model effectively presents monthly revenue trends, best-selling products, inventory asset values, gross profit, and stock discrepancies from stock opname. These visualizations help business owners make more accurate decisions regarding pricing, inventory management, and purchasing. The study contributes practically by offering an effective visualization framework for SMEs, and theoretically by demonstrating the integration of Laravel and Chart.js for generating data-driven business insights.

Keywords: Data Visualization, Laravel, Chart.js, Inventory, Decision Making

ABSTRAK

Penelitian ini mengatasi keterbatasan data persediaan dan transaksi yang masih berbentuk mentah dan sulit dianalisis untuk pengambilan keputusan bisnis. Model visualisasi dikembangkan untuk mengubah data operasional menjadi grafik interaktif yang mendukung keputusan strategis pada PT XYZ. Sistem dibangun menggunakan model SDLC Waterfall dengan Laravel untuk pemrosesan data dan Chart.js untuk penyajian visual. Hasil penelitian menunjukkan bahwa model ini mampu menampilkan tren pendapatan bulanan, produk terlaris, nilai aset stok, laba kotor, serta mendeteksi selisih stok melalui stock opname. Visualisasi tersebut memberikan wawasan yang membantu pemilik bisnis dalam menetapkan harga, mengelola persediaan, dan merencanakan pembelian secara lebih akurat. Penelitian ini berkontribusi praktis dengan menyediakan kerangka visualisasi yang efektif bagi UKM, serta kontribusi teoretis melalui integrasi Laravel dan Chart.js sebagai pendekatan visualisasi berbasis data.

Kata Kunci: Visualisasi Data, Laravel, Chart.js, Inventaris, Pengambilan Keputusan

PENDAHULUAN

Pengelolaan persediaan dan analisis transaksi merupakan aktivitas penting bagi kelangsungan bisnis (Limlawan et al., 2022). Data stok dan transaksi yang akurat menjadi dasar untuk menentukan kebijakan pengadaan, penetapan harga dan promosi (Tamba, 2019). Namun, data mentah yang disajikan dalam bentuk tabel sulit memberi gambaran langsung tentang tren, anomali, dan prioritas pengambilan keputusan. Pencatatan transaksi penjualan dan pembelian menjadi lebih efisien saat dilakukan dengan perangkat lunak daripada

dilakukan secara manual (Leovin et al., 2020).

Berbagai penelitian sebelumnya telah mengembangkan sistem informasi penjualan, peramalan stok, maupun dashboard analitis, namun sebagian besar masih berfokus pada penyajian data dalam bentuk tabel atau laporan statis. Beberapa studi menekankan pentingnya visualisasi interaktif, tetapi belum banyak yang memodelkan integrasi penuh antara proses agregasi data, perhitungan analitis, dan penyajian grafik secara real-time menggunakan framework modern seperti Laravel dan Chart.js. Kesenjangan ini

menunjukkan perlunya model visualisasi yang tidak hanya menampilkan grafik, tetapi juga mengolah data operasional menjadi metrik bisnis yang relevan untuk pengambilan keputusan.

Penelitian ini bertujuan untuk membangun model visualisasi laporan yang mampu mengonversi data operasional menjadi informasi yang mudah dipahami melalui enam visual utama, yaitu tren pendapatan bulanan, produk terlaris, nilai aset stok, laba kotor per stok, laba kotor per transaksi, dan validasi stok melalui stock opname. Melalui model ini, pemilik bisnis dapat mengambil keputusan berbasis data secara lebih cepat dan akurat.

Pada studi kasus ini, penulis menggunakan data dari Perusahaan XYZ, sebuah PT yang bergerak di bidang pakaian anak. Penelitian ini memodelkan proses pengubahan data operasional menjadi visualisasi terstruktur pada studi kasus perusahaan yang membantu pemilik bisnis dalam pengambilan keputusan. Pemanfaatan visualisasi data interaktif memberikan kemudahan akses bagi pengguna dalam menganalisis kinerja penjualan dan tren produk, yang secara langsung mendukung proses pengambilan keputusan yang lebih cepat dan akurat (Limbor et al., 2024).

Sistem dikembangkan dengan kerangka kerja *Laravel* pada sisi server, yang menghasilkan data teragregasi. Dan didukung dengan *Chart.js* yang merupakan *framework* yang berfungsi untuk menyajikan data informasi dalam format visual grafik. *Chart.js* terbukti lebih efektif dan mudah dipahami oleh pengguna dibandingkan penyajian data dalam bentuk tabel mentah (Toppany et al., 2024). Fokus laporan meliputi tren penjualan, performa produk, nilai aset stok, analisis laba kotor baik per stok maupun per transaksi, dan validasi stok melalui *stock opname*. Penelitian ini menjelaskan metodologi perancangan sistem, teknik pengambilan dan perhitungan data sesuai implementasi pada *controller*, serta interpretasi hasil

visualisasi untuk pengambilan keputusan operasional dan strategis.

METODE

Metodologi Pengembangan Sistem

Dalam membuat sistem sebagai perantara visualisasi data yang bisa mendukung pengambilan keputusan, sistem ini mengadopsi model *Software Development Life Cycle (SDLC) Waterfall*. SDLC didefinisikan sebagai alur yang terstruktur dan sistematis untuk pengembangan perangkat lunak, mencakup fase-fase berurutan mulai dari *requirement Analysis* (perencanaan), *System Design*, Pengembangan, pengujian, hingga implementasi dan pemeliharaan sistem (Vebrianto et al., 2025).

Waterfall dipercaya dapat memudahkan Tahapan pengembangan sistem (Abdurrahman & Masripah, 2017). Pemilihan model ini didasarkan pada kebutuhan bisnis yang sudah terdefinisi jelas dari awal dan meminimalisir perubahan di tengah proses, sehingga memungkinkan alur kerja yang sekuensial dan terstruktur. Tahapan yang dilalui meliputi:

1. *Requirement* (Analisis Kebutuhan): Pengumpulan kebutuhan sistem melalui wawancara dan observasi proses bisnis.
2. *System Design* (Desain Sistem): Perancangan basis data (Konseptual, Logikal, Fisikal) dan pemodelan alur sistem (UML seperti *Use Case*, *Activity*, dan *Sequence*).
3. *Pengembangan*: Pengkodean aplikasi *fullstack* menggunakan *framework Laravel* dan *database MySQL*.
4. *Testing* (Pengujian): Pengujian fungsionalitas fitur dan logika bisnis (*Black-box testing*).
5. *Deployment* dan *Maintenance*: Penempatan sistem di lingkungan produksi lokal dan pemeliharaan berkala.

Perancangan Basis Data

Sebagai fondasi utama yang menjamin konsistensi data operasional

sebelum divisualisasikan, basis data dirancang menggunakan tiga tahap. Proses perancangan basis data dilakukan melalui tiga tahapan: pertama, desain konseptual, yang berfokus pada pemodelan data secara abstrak tanpa mempertimbangkan aspek fisik penyimpanan; kedua, desain logikal, yang memetakan model data ke dalam struktur yang spesifik (seperti tabel dan relasi) namun masih independen dari sistem manajemen basis data (DBMS) tertentu; dan ketiga, desain fisikal, yang merupakan implementasi detail bagaimana data disimpan dan dihubungkan dalam DBMS yang sebenarnya (Umar et al., 2019). Perancangan ini menghasilkan sebuah *Entity Relationship Diagram* (ERD) yang memetakan hubungan antar entitas bisnis. Struktur ERD utama sistem ini terdiri dari 14 entitas yang saling berelasi, meliputi:

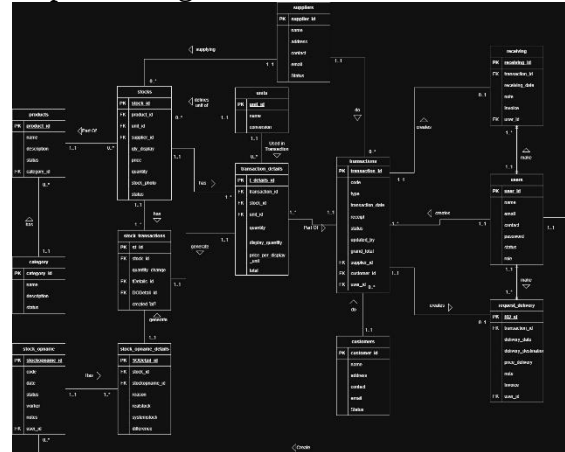
- **Data Master:** *users*, *categories*, *products*, *suppliers*, dan *customers*.
- **Data Transaksi:** *transactions*, *transaction_details*, *receiving*, dan *request_delivery*.
- **Data Stok & Akurasi:** *stocks*, *stock_transactions*, *stock_opname*, dan *stock_opname_details*.

Inti dari data yang divisualisasikan adalah entitas *stocks*, *transactions*, dan *transaction_details*. Hubungan penting dalam skema ini mencakup:

1. **Stok dan Produk:** Entitas *products* memiliki banyak *stocks* (satu produk bisa memiliki beberapa stok dengan harga atau pemasok berbeda) dan *stocks* memiliki hubungan satu-ke-banyak dengan *stock_transactions* (mencatat riwayat perubahan kuantitas).
2. **Transaksi:** Entitas *transactions* (yang digeneralisasi menjadi *supplier_transaction* dan *customer_transaction* pada desain logikal) berelasi satu-ke-banyak dengan *transaction_details*, yang kemudian berelasi dengan *stocks*.
3. **Laba Kotor:** Perhitungan laba kotor (*Gross profit*) yang divisualisasikan bersumber dari perkalian data harga beli (*stocks.price*) dan kuantitas transaksi

yang dicatat di *transaction_details*, yang kemudian diselisihkan dengan *total_ammount* dari *transactions*.

4. **Stock opname:** Entitas *stock_opname* dan *stock_opname_details* berfungsi memvalidasi akurasi stok sistem, dan perbedaan (*difference*) yang dicatat menjadi data penting untuk visualisasi perbandingan stok.



Gambar 1. Logical ERD

Sumber: Dokumentasi pribadi

Arsitektur sistem

Sistem mengikuti pola *Model-View-Controller*. *Laravel*, dengan fondasi MVC-nya, mempermudah pemahaman dan pengelolaan struktur kode. Karakteristik ini menjadi faktor kunci yang mengakselerasi pengembangan aplikasi web, menghasilkan sistem yang lebih cepat diselesaikan (Gede Erick Waisnawa et al., 2024).

Alur kerja dari *Laravel* dimulai ketika data operasional seperti transaksi, detail transaksi, stok, dan hasil *stockopname* tersimpan di basis data relasional. Kedua, *ReportController* (*Controller* yang digunakan untuk fitur Pelaporan) di server mengeksekusi *query* agregasi untuk menghitung metrik yang diperlukan. Ketiga, hasil agregasi dikemas menjadi struktur data sederhana (label dan series) dan dikirim ke *view*. Keempat, *view* menggunakan *Chart.js* untuk merender grafik interaktif berdasarkan data tersebut.

Pemrosesan dilakukan di sisi server agar perhitungan berat terjadi di database, memanfaatkan fungsi agregasi dan pengelompokan untuk menjaga konsistensi

dan performa. Filter periode tanggal dan parameter lain dikirim oleh pengguna melalui antarmuka dan diteruskan ke *controller*.

Teknik pengambilan dan perhitungan data laporan

Dalam menyediakan informasi untuk pengambilan keputusan bisnis, visualisasi laporan memegang peran krusial. Proses ini memerlukan teknik pengambilan data yang spesifik dan logika perhitungan yang akurat, yang diimplementasikan secara langsung pada *controller* aplikasi.

Tren Bulanan Pendapatan

Pembuatan tren pendapatan bulanan sangat berdampak pada keputusan bisnis. Seperti yang dikatakan pada sebuah penelitian bahwa Ekspektasi pendapatan sangat berpengaruh pada pengambilan keputusan dalam berwirausaha (Hidayah & Fitriyah, 2023). Dalam menganalisis tren bulanan pendapatan, data utama bersumber dari tabel transaksi. Kriteria yang diterapkan secara ketat adalah hanya mengambil transaksi yang bertipe penjualan dan memiliki status selesai. Secara teknik, data diambil per bulan untuk tahun yang dipilih oleh pengguna. Prosesnya menggunakan fungsi agregasi yang bertugas menjumlahkan kolom *grand_total* untuk setiap bulan. Logika perhitungan di *controller* melibatkan ekstraksi bulan dari tanggal transaksi, dilanjutkan dengan penghitungan jumlah total pendapatan per bulan. Hasil akhir yang disajikan berupa label bulan mulai dari Januari hingga Desember, beserta nilai total pendapatan yang bersesuaian untuk tahun yang dianalisis. Laporan ini disajikan menggunakan *line chart* (grafik garis), di mana sumbu horizontal mewakili label bulan dan sumbu vertikal berisi total pendapatan.

Tren Stok Terlaris Berdasarkan Tanggal

Visualisasi tren stok menjadi elemen penting dalam pengambilan

keputusan. Dengan visualisasi data tersebut *stakeholders* perusahaan dapat melihat dengan cepat pola/tren dari produk mereka (Susi et al., 2023). Analisis tren stok terlaris membutuhkan data yang bersumber dari tabel detail transaksi, yang berelasi dengan tabel stok dan produk. Data yang digunakan dibatasi oleh kriteria transaksi bertipe penjualan, berstatus selesai, dan berada dalam rentang tanggal yang dipilih oleh pengguna.

Secara teknik, agregasi dilakukan dengan menjumlahkan kuantitas pada kolom *quantity* apabila basis analisis volume yang dipilih, atau menjumlahkan nilai pada kolom total jika basis *revenue* yang digunakan. Hasil ini kemudian dikelompokkan per nama stok (gabungan produk dan *supplier*), diurutkan secara menurun, dan dibatasi hanya pada sepuluh teratas. Pada *controller*, perhitungan yang diterapkan adalah fungsi *SUM (quantity)* untuk setiap stok ketika basis kuantitas dipilih. Sebaliknya, jika basis *revenue* yang dipilih, perhitungan yang digunakan adalah *SUM (total)* untuk setiap stok. Data keluaran terdiri dari label nama stok dan nilai total yang sesuai dengan basis analisis. Penyajiannya menggunakan *bar chart* (grafik batang) horizontal atau vertikal untuk menampilkan peringkat 10 produk teratas menurut kuantitas atau pendapatan dalam rentang tanggal.

Tren Stok dengan Nilai Aset Tertinggi

Tren yang diidentifikasi berdasarkan nilai aset bertujuan mengidentifikasi tren stok dengan nilai aset tertinggi, dengan bersumber pada tabel stok. Kriteria yang diterapkan hanya pada stok yang berstatus aktif. Secara teknik, perhitungan nilai aset stok untuk setiap item dilakukan dengan mengalikan jumlah yang ditampilkan di sistem (*qty display*) dengan harga per unit (*price*) pada stok tersebut. Logika perhitungan di *controller* secara spesifik menghitung nilai aset stok (*stock value*) sebagai hasil kali (*qty_display* x *price*) untuk setiap stok aktif, lalu mengambil 10 stok tertinggi berdasarkan nilai aset

tersebut. Penyajian laporan menggunakan *bar chart* yang menampilkan nama stok dan nilai asetnya.

Laba Kotor per Stok

Laba kotor (*Gross profit*) merupakan indikator keuntungan yang didapatkan perusahaan dari penjualan, yang dihitung sebelum adanya pengurangan terhadap seluruh biaya operasional dan beban lain yang ditanggung oleh perusahaan (Mukhlisotul et al., 2018). Untuk menghitung laba kotor per stok, diperlukan *join* antara tabel detail transaksi dengan tabel stok, produk, dan pemasok. Kriteria transaksi yang diambil adalah penjualan, berstatus selesai, dan berada dalam rentang tanggal yang difilter. Secara teknik, untuk setiap stok unik *controller* menghitung beberapa agregat, yaitu total kuantitas terjual (menggunakan satuan *display*), total pendapatan dari penjualan stok tersebut, total biaya modal barang yang terjual, serta laba kotor sebagai selisih antara pendapatan dan biaya modal. Logika perhitungan di *controller* secara rinci menghitung $SUM(display_quantity)$ sebagai jumlah terjual, $SUM(total)$ sebagai *revenue*, $(SUM(display_quantity) \times stocks.price)$ sebagai *cost of goods*, dan $SUM(total - display_quantity \times stocks.price)$ sebagai *gross profit* per stok. Data kemudian dikelompokkan per stok dan diurutkan menurut laba kotor menurun. Laporan ini disajikan menggunakan *bar chart* atau tabel berperingkat yang menampilkan kolom nama stok, kuantitas terjual, *revenue*, *cost of goods*, dan *gross profit*.

Laba Kotor per Transaksi

Perhitungan laba kotor per transaksi bersumber dari tabel transaksi, tabel detail transaksi, dan tabel stok. Kriterianya sama dengan laba kotor per stok, yaitu transaksi penjualan, berstatus selesai, dan berada dalam rentang tanggal tertentu. Secara teknik, untuk setiap transaksi *controller* menjumlahkan total biaya modal dari item-item di dalamnya dengan mengalikan harga

stok dengan *display quantity per item*. Kemudian, laba kotor per transaksi dihitung sebagai *grand total* transaksi dikurangi total biaya modal tersebut. Logika perhitungan di *controller* untuk setiap transaksi adalah menghitung $SUM(stocks.price \times transaction_details.display_quantity)$ sebagai *total cost of goods*, setelah itu *gross profit* per transaksi diperoleh dari $transactions.grand_total$ dikurangi *total cost of goods*. Laporan ini disajikan sebagai tabel dengan kolom tanggal, partner (pelanggan atau supplier), *grand total*, *total cost*, dan *gross profit*. Untuk visualisasi, dapat ditampilkan juga sebagai *bar chart* perankingan transaksi berdasarkan *gross profit*.

Stock Opname

Pada laporan *stock opname* menggunakan data dari tabel *stock_opname* dan *stock_opname_details*, berelasi dengan tabel stok, produk, dan pemasok. Kriteria yang diambil adalah *stock opname* yang berstatus *approved* dalam bulan dan tahun yang dipilih, dengan fokus hanya pada catatan *stock opname* terbaru untuk setiap stok dalam periode tersebut.

Secara teknik, *controller* melakukan *join* antar tabel dan menggunakan *subquery* untuk memilih *stock_opname_id* terbesar per stok pada periode yang sama, sehingga hanya catatan *stockopname* terakhir per stok di bulan yang dipilih yang diambil. Dari catatan tersebut, nilai stok sistem, stok real, dan selisihnya diambil. Untuk ringkasan penyesuaian kuantitas, fungsi terpisah mengakumulasi perubahan positif dan negatif pada transaksi penyesuaian, mengelompokkan hasilnya per kode *stock opname*. Logika perhitungan di *controller* mengambil *fields system_stock*, *real_stock*, dan *difference* untuk stok yang termasuk dalam *stockopname* terakhir bulan itu. Untuk ringkasan penyesuaian, $SUM(quantity_change)$ dihitung terpisah untuk perubahan positif dan negatif menggunakan kondisi *CASE*, lalu dikelompokkan per kode *opname*. Laporan

ini disajikan menggunakan *clustered bar chart* yang menampilkan untuk setiap item dua batang bersebelahan, yakni satu batang untuk stok sistem dan satu batang untuk stok fisik.

Keterangan filter tanggal dan penyajian data

Semua laporan yang bersifat periodik menerima parameter rentang tanggal atau tahun melalui antarmuka. *Controller* menggunakan parameter ini untuk menyaring data menggunakan kondisi *whereBetween* pada tanggal transaksi atau *whereMonth* dan *whereYear* pada *stockopname*. Hasil pengolahan dikembalikan dalam struktur label dan series yang siap dirender oleh *Chart.js*.

HASIL DAN PEMBAHASAN

Tren Pendapatan Bulanan



Gambar 2. Chart Tren Pendapatan Bulanan

Sumber: Dokumentasi pribadi

Grafik ini dibuat dari data bulanan yang dihasilkan *controller* melalui agregasi jumlah *grand total* transaksi penjualan per bulan. Sumbu horizontal menampilkan bulan Januari sampai Desember, sedangkan sumbu vertikal menampilkan total pendapatan per bulan. Visual ini membantu melihat pola musiman dan perubahan nilai pendapatan sepanjang tahun.

Dengan adanya grafik ini, para pengambil keputusan (*Stakeholders*) bisa mengambil keputusan bisnis. Jika terlihat penurunan pendapatan selama beberapa bulan berturut-turut, langkah evaluatif yang bisa diambil antara lain meninjau kebijakan harga, meluncurkan promosi bertarget, atau

melakukan analisis penyebab penurunan produk per produk. Jika pendapatan meningkat tanpa kenaikan volume transaksi, hal ini mengindikasikan kenaikan nilai transaksi rata-rata sehingga strategi upselling atau peningkatan harga mungkin efektif. Sebaliknya, jika volume transaksi naik tetapi pendapatan stagnan, fokus bisa diarahkan pada peningkatan nilai per transaksi melalui bundling atau promosi nilai tambah.

Visualisasi tren pendapatan bulanan tidak hanya menunjukkan pola fluktuasi penjualan, tetapi juga membantu perusahaan mengidentifikasi periode dengan performa terbaik maupun terendah. Pola kenaikan pendapatan dapat menunjukkan keberhasilan strategi harga, promosi, atau peningkatan permintaan musiman. Sebaliknya, penurunan pendapatan pada bulan-bulan tertentu dapat menjadi indikator awal adanya permasalahan seperti penurunan permintaan, kurangnya stok pada produk kunci, atau efektivitas promosi yang menurun. Pemahaman terhadap pola musiman ini memungkinkan perusahaan menyusun proyeksi penjualan yang lebih akurat, mengalokasikan sumber daya dengan lebih efisien, serta merencanakan pengadaan stok dan insentif promosi pada waktu yang tepat.

Tren stok terlaris



Gambar 3. Chart Tren Stok Terlaris

Sumber: Dokumentasi pribadi

Data berasal dari detail transaksi yang diaggregasi per stok. Tergantung parameter analisis, *controller* menghitung total kuantitas terjual atau total *revenue* per stok, lalu mengambil 10 teratas. Visual menyorot produk yang paling sering dibeli

atau paling banyak menyumbang pendapatan.

Produk yang menempati peringkat atas berdasarkan kuantitas harus diprioritaskan untuk manajemen stok sehingga tidak terjadi kekosongan. Produk yang tinggi berdasarkan *revenue* meskipun tidak selalu memiliki volume besar menunjukkan produk bernilai tinggi yang layak mendapat perhatian pemasaran dan pengelolaan margin. Untuk produk yang berkinerja buruk secara konsisten, keputusan meliputi pemberian diskon, *bundling*, reposisi produk, atau penghentian penjualan.

Interpretasi tren stok terlaris memberikan pemahaman langsung mengenai perilaku konsumen dan kontribusi tiap produk terhadap pendapatan. Produk dengan volume penjualan tinggi perlu mendapatkan prioritas dalam perencanaan persediaan agar tidak terjadi kekurangan stok yang dapat menyebabkan potensi kehilangan penjualan. Produk yang berada di peringkat atas berdampak besar pada stabilitas operasional, sehingga pergerakan permintaannya harus terus dipantau. Sementara itu, produk dengan performa rendah dapat dianalisis lebih lanjut untuk menentukan strategi optimal, seperti promosi, *bundling*, reposisi, atau bahkan penghentian penjualan. Informasi ini membantu perusahaan mengoptimalkan modal kerja dengan fokus pada produk yang memberikan kontribusi terbesar.

Tren stok aset tertinggi



Gambar 4. Tren Stok Aset Tertinggi

Sumber: Dokumentasi pribadi

Controller menghitung nilai aset tiap stok sebagai hasil perkalian *qty_display* dengan *price*. Nilai ini mencerminkan jumlah modal yang terikat pada setiap item stok. *Chart* menampilkan sepuluh item dengan nilai modal tertinggi. Jika sebagian besar modal persediaan terkonsentrasi pada beberapa item, perusahaan menghadapi risiko likuiditas apabila permintaan untuk item tersebut menurun. Kebijakan yang dapat dipertimbangkan termasuk diversifikasi stok, negosiasi ulang harga pembelian, dan pengelolaan pembelian bertahap untuk mengurangi modal yang terlalu besar tertahan pada satu atau beberapa SKU.

Nilai aset tertinggi pada stok mencerminkan besarnya modal perusahaan yang terikat dalam persediaan. Produk dengan nilai aset tinggi namun tingkat perputaran rendah berpotensi menimbulkan risiko likuiditas karena modal tidak kembali secara optimal. Sebaliknya, jika produk bernilai aset tinggi juga termasuk kategori fast-moving, hal ini menandakan bahwa alokasi modal sudah tepat. Dengan memahami konsentrasi modal pada beberapa item tertentu, perusahaan dapat menentukan apakah diperlukan diversifikasi pemasok, negosiasi ulang harga pembelian, atau strategi pembelian bertahap untuk mengurangi penumpukan modal. Visualisasi ini menjadi dasar bagi manajemen untuk menjaga efisiensi pengelolaan inventory dan meminimalisir risiko dead stock.

Laba kotor per stok

Laba Kotor Penjualan (01/Nov/2025 - 30/Nov/2025)					
STOK (PRODUK/SUPPLIER)	QTY TERJUAL	SATUAN	PENDAPATAN (JUAL)	HARGA PEROK (BELI)	LABA KOTOR
Oblong Anak Hige (Odi)	70	lusin	Rp 4.100.000	Rp 2.450.000	Rp 1.650.000
Oblong Anak Hige (sugiyanto)	5	lusin	Rp 1.250.000	Rp 800.000	Rp 450.000
TOTAL LABA KOTOR:					Rp 2.100.000

Gambar 5. Laba Kotor Per Stok

Sumber: Dokumentasi pribadi

Controller menghasilkan agregat per stok yang mencakup jumlah unit terjual (*display quantity*), *revenue* dari stok tersebut, total *cost of goods* yang dihitung sebagai jumlah *display quantity* dikali

harga stok, dan *gross profit* yang merupakan selisih antara *revenue* dan *cost of goods*. Hasil dikelompokkan per stok dan diurutkan berdasarkan *gross profit* menurun.

Chart ini membantu mengidentifikasi stok yang menghasilkan laba terbesar sehingga pemilik dapat memprioritaskan stok tersebut untuk *replenishment*, promosi, dan strategi margin. Juga membantu menemukan produk yang laku namun berkontribusi sedikit terhadap laba sehingga perlu evaluasi harga atau biaya.

Analisis laba kotor per stok memberikan gambaran mengenai profitabilitas nyata dari setiap produk, bukan hanya performa penjualannya. Produk dengan volume tinggi tetapi laba rendah membutuhkan evaluasi terkait struktur biaya, strategi harga, atau efisiensi pemasok. Sebaliknya, produk dengan margin besar layak diprioritaskan sebagai produk unggulan untuk meningkatkan total profit. Informasi ini juga membantu perusahaan mengidentifikasi peluang *upselling*, menentukan promosi yang tepat, serta menyesuaikan strategi harga agar tetap kompetitif tanpa mengurangi margin keuntungan. Dengan demikian, visualisasi laba kotor per stok membantu perusahaan menyeimbangkan antara volume penjualan dan tingkat profitabilitas.

Laba kotor per transaksi

TANGGAL	ID TRANSAKSI	PARTNER	GRAND TOTAL	TOTAL BIAYA MODAL	LABA KOTOR
15 Nov 2025	23	diabate musta	1.277.000	800.000	477.000
07 Nov 2025	22	Sultan Said Abdul Rahim	2.600.000	700.000	1.900.000
06 Nov 2025	11	Mr. Baraji	2.650.000	1.750.000	900.000

Gambar 6. Laba Kotor Per Transaksi

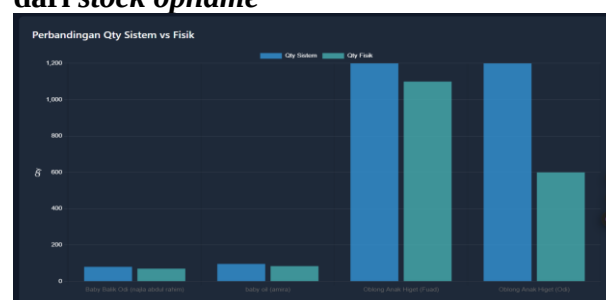
Sumber: Dokumentasi pribadi

Untuk setiap transaksi, *controller* menghitung total biaya modal item-item yang terlibat sebagai jumlah harga stok kali *display quantity*. Laba kotor per transaksi adalah *grand total* transaksi dikurangi total biaya modal tersebut. Visual ini dapat memperlihatkan transaksi paling menguntungkan dan transaksi yang kurang menguntungkan. Analisis transaksi

memungkinkan identifikasi pola seperti transaksi yang mengandung diskon berlebihan, kombinasi produk berdampak rendah margin, atau transaksi yang menunjukkan margin tinggi sehingga bisa direplikasi. Keputusan dapat berupa penyesuaian kebijakan diskon, *review* kombinasi produk dalam satu *invoice*, atau perubahan kebijakan penjualan khusus.

Laba kotor per transaksi memberikan wawasan mengenai efektivitas penjualan pada tingkat operasional. Transaksi dengan margin tinggi dapat menjadi indikator keberhasilan kombinasi produk, strategi harga, atau kebijakan diskon yang tepat sasaran. Sementara itu, transaksi bernilai rendah atau bahkan hampir tanpa margin menunjukkan potensi masalah seperti pemberian diskon berlebihan, penjualan produk berbiaya tinggi tanpa kompensasi harga, atau keputusan pelanggan yang tidak selaras dengan target keuntungan perusahaan. Dengan menganalisis pola ini, perusahaan dapat memperbaiki aturan diskon, mengidentifikasi kebiasaan pembelian yang merugikan, serta merancang paket penjualan yang meningkatkan margin tanpa mengurangi daya tarik bagi pelanggan.

Perbandingan stok sistem vs stok aktual dari *stock opname*



Gambar 7. Perbandingan Stok Aktual dan Stok Sistem

Sumber: Dokumentasi pribadi

Controller memilih catatan *stockopname* terakhir untuk setiap stok pada bulan dan tahun yang dipilih dengan memastikan status *stockopname* adalah *approved*. Untuk setiap stok diambil nilai *system_stock*, *real_stock*, dan selisihnya. *Chart* menampilkan dua batang berdampingan yaitu satu batang stok yang

tercatat di sistem dan satu batang stok hasil penghitungan fisik.

Perbedaan negatif di mana stok real lebih kecil daripada stok sistem menandakan potensi kehilangan, kesalahan input, atau prosedur gudang yang lemah; langkah korektif termasuk audit gudang, pemeriksaan alur penerimaan dan pengeluaran, atau penguatan kontrol inventaris. Jika stok real lebih besar daripada stok sistem, hal ini menandakan kemungkinan transaksi masuk yang tidak tercatat atau kesalahan pencatatan; tindakan yang sesuai adalah rekonsiliasi dan perbaikan data historis. *Controller* juga menyediakan ringkasan penyesuaian kuantitas per kode *stock opname* dengan mengelompokkan perubahan positif dan negatif, yang dapat divisualisasikan sebagai *chart* terpisah untuk melihat kapan dan seberapa besar penyesuaian dilakukan.

Perbandingan antara stok sistem dan stok aktual memberikan gambaran mengenai tingkat akurasi dan integritas pengelolaan persediaan. Selisih negatif menunjukkan potensi kehilangan barang (*loss*), kesalahan input, pencatatan yang tidak konsisten, atau kelemahan dalam alur penerimaan dan pengeluaran barang. Selisih positif menunjukkan adanya barang yang masuk namun tidak tercatat atau kesalahan pencatatan historis. Kondisi ini dapat menimbulkan keputusan pembelian yang tidak tepat, menurunkan akurasi laporan keuangan, dan memengaruhi kepercayaan terhadap sistem. Dengan memahami pola ketidaksesuaian ini, perusahaan dapat memperbaiki kontrol internal, memperketat standar operasional gudang, dan meningkatkan frekuensi audit untuk mengurangi risiko kesalahan dan kerugian.

Sinergi antar *chart* dan rekomendasi operasional

Integrasi dari keenam visualisasi data ini memungkinkan pemilik bisnis untuk melakukan analisis menyeluruh yang mendukung pengambilan keputusan strategis. Analisis ini bersifat iteratif dan

terstruktur, memungkinkan penelusuran dari masalah tingkat makro ke detail operasional.

Alur Analisis Praktis Terpadu

Proses analisis dimulai dengan pemeriksaan tren pendapatan. Apabila terdeteksi adanya penurunan, langkah selanjutnya adalah menelusuri penyebabnya dengan memeriksa apakah terjadi penurunan volume penjualan pada produk tertentu melalui visualisasi tren stok terlaris. Setelah mengidentifikasi produk bermasalah atau produk potensial, pemilik bisnis perlu memverifikasi apakah produk yang secara konsisten menyumbang laba terbesar masih memiliki stok yang memadai dengan merujuk pada *chart* stok aset.

Sebelum mengambil keputusan restok besar, sangat penting untuk memeriksa adanya selisih stok (*inconsistencies*) melalui *chart stockopname*, dan jika selisih signifikan ditemukan, audit mendalam harus segera dilakukan. Berdasarkan kombinasi informasi komprehensif ini mulai dari tren pendapatan, volume penjualan, nilai aset stok, hingga akurasi inventori keputusan strategis mengenai penetapan harga, peluncuran promosi, atau pembelian kembali (*restock*) dapat diambil dengan dasar data yang kuat.

Rekomendasi Operasional Spesifik

Berdasarkan temuan dari analisis terpadu tersebut, beberapa rekomendasi operasional spesifik dapat dirumuskan. Pertama, untuk memastikan kelancaran operasional dan meminimalisir peluang kehilangan penjualan, sistem dapat dikonfigurasi untuk menetapkan aturan restock otomatis bagi produk yang menempati volume penjualan tertinggi dan yang berkontribusi signifikan terhadap *revenue* total. Kedua, sangat penting untuk memprioritaskan audit inventori secara mendalam dan segera bagi *Stock Keeping Unit* (SKU) yang menunjukkan ketidaksesuaian stok (*inconsistent*

inventory) pada *chart stockopname*. Ketiga, evaluasi margin produk harus dilakukan secara periodik dengan merujuk pada *chart* laba kotor per stok, dan strategi penetapan harga harus disesuaikan berdasarkan analisis margin ini untuk mengoptimalkan profitabilitas. Terakhir, *chart* laba kotor per transaksi dapat dimanfaatkan sebagai alat ukur efektivitas, di mana analisis per transaksi digunakan untuk menilai efektivitas promosi dan kebijakan diskon yang mungkin telah diterapkan.

Batasan Penelitian

Penelitian ini memiliki beberapa keterbatasan. Pertama, sistem diuji menggunakan data dari satu perusahaan sehingga generalisasinya masih terbatas untuk industri lain. Kedua, model visualisasi hanya berfokus pada data transaksi dan stok tanpa memasukkan variabel eksternal seperti tren pasar atau perilaku pelanggan. Ketiga, penelitian ini tidak membahas performa sistem dalam skala data yang lebih besar sehingga perlu diuji lebih lanjut pada lingkungan operasional dengan beban tinggi.

Kontribusi praktis penelitian ini adalah menyediakan model visualisasi terstruktur yang dapat membantu pemilik usaha dalam mengambil keputusan berbasis data secara cepat. Kontribusi teoretisnya terletak pada penerapan kombinasi *Laravel* dan *Chart.js* sebagai pendekatan terintegrasi dalam visualisasi data operasional, yang dapat diadopsi dan dikembangkan pada penelitian selanjutnya.

KESIMPULAN

Implementasi model visualisasi menggunakan *Laravel* untuk perhitungan server-side dan *Chart.js* untuk rendering klien berhasil menyediakan enam jenis visual utama yang diperlukan oleh pemilik usaha: tren bulanan pendapatan, tren produk terlaris, daftar stok nilai aset tertinggi, laba kotor per stok, laba kotor per transaksi, dan perbandingan stok sistem dengan stok aktual dari *stock opname*.

Setiap visual dihasilkan berdasarkan perhitungan dan query yang terimplementasi di *controller* sehingga hasil yang ditampilkan konsisten dengan data operasional. Visualisasi menyediakan pijakan bagi pengambilan keputusan yang bersifat operasional maupun strategis, misalnya penjadwalan *restock*, optimasi harga, evaluasi promosi, dan tindakan perbaikan operasional gudang. Secara ringkas, model ini mengubah data mentah menjadi wacana tindakan yang dapat dijalankan, sehingga pemilik bisnis dapat merespons kondisi nyata dengan cepat dan berbasis bukti.

DAFTAR PUSTAKA

- Abdurrahman, A., & Masripah, S. (2017). Metode Waterfall Untuk Sistem Informasi Penjualan. *Information System For Educators and Professionals*, 2(1), 95–104.
- Hidayah, N. L., & Fitriyah, H. (2023). Pengaruh E-Commerce, Modal Usaha, Ekspektasi Pendapatan, dan Penggunaan Sistem Informasi Akuntansi Terhadap Pengambilan Keputusan Berwirausaha dengan Self-Efficacy Sebagai Variabel Moderasi. *Equilibrium: Jurnal Ekonomi-Manajemen-Akuntansi*, 19(2), 153-164.
- Jannah, M. (2018). Analisis pengaruh biaya produksi dan tingkat penjualan terhadap laba kotor. *Banque Syar'i: Jurnal Ilmiah Perbankan Syariah*, 4(1), 87-112. <https://doi.org/10.24127/exclusive.v4i1.8389>
- Leovin, A., Beng, J. T., & Dewayani, E. (2020, December). Business to business e-commerce sales system using web-based quotation: A case study on company X. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1007, No. 1, Article 012156). IOP Publishing. <https://doi.org/10.1088/1757-899X/1007/1/012156>

- Limbor, E. G., Putri, T. A., Qadriah, S. A. R., Arisandi, D., & Beng, J. T. (2024). Perancangan Dashboard Penjualan Pakaian di Toko Cleo dengan Power BI. *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, 13(3), 2061-2069. [10.35889/jutisi.v13i3.2383](https://doi.org/10.35889/jutisi.v13i3.2383)
- Limlawan, V., Wiwattanakornwong, K., & Suantubtim, S. (2022). Importance of inventory management knowledge for new entrepreneurs: A demonstration through student performance in a business game. In *Proceedings of the RSU International Research Conference 2022* (pp. 2023–2209).
- Susilawati Sugiana, N. S., & Musty, B. (2023). Analisis Data Sistem Informasi Monitoring Marketing; Tools Pengambilan Keputusan Strategic. *Jutisi: Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*. Doi: [10.35889/jutisi.v13i3.2383](https://doi.org/10.35889/jutisi.v13i3.2383)
- Tamba, M. (2019). Sistem Pendukung Keputusan Prediksi Stok Barang Menggunakan Metode Moving Average Berbasis Client Server Pada PT Union. *Jurnal Times*, 8(1), 1-18. <https://doi.org/10.51351/jtm.8.1.2019598>
- Toppary, M. P., Kurnia, D., & Adji, R. P. H. (2024). Integrasi Framework Bootstrap Dan Chart. Js Untuk Visualisasi Data Sensor pada Sistem Hidroponik Berbasis Internet Of Things (IoT). *JMeka: Jurnal Rekayasa Mekatronika*, 1(2), 11-21.
- Umar, R., Hadi, A., Widiandana, P., Anwar, F., Jundullah, M., Ikrom, A., & Dahlan Jl Jl Soepomo, A. (2019). Perancangan database point of sales apotek dengan menerapkan model data relasional. *Query: Journal of Information Systems*, 5341, 33-41. <http://dx.doi.org/10.58836/query.v3i2.6394>
- Vebrianto, D., Beng, J. T., & Arisandi, D. (2025). Perancangan Sistem Informasi Penjualan Pada Toko Gorden XYZ Menggunakan Metode Waterfall. *Jutisi : Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 14(2), 1112. <https://doi.org/10.35889/jutisi.v14i2.2901>
- Waisnawa, I. G. E., Nata, G. N. M., & Dewi, K. H. S. (2024, June). Sistem Informasi Inventory Barang Pada Bali Jacosi berbasis Website Menggunakan Framework Laravel. In *Seminar Hasil Penelitian Informatika dan Komputer (SPINTER)| Institut Teknologi dan Bisnis STIKOM Bali* (pp. 406-411).