


Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
<https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/index>
 Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
 Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
 e-ISSN: 2685-0893

Pengembangan Sistem *Reporting Purchasing* Berbasis Python dan SQL pada PT. X

DOI: <http://dx.doi.org/10.35889/jutisi.v15i3.3738>

Creative Commons License 4.0 (CC BY – NC) 

Leny Marshalina¹, Jap Tji Beng^{2*},
 Rahmiyana Nurkholiza³, Vienchenzia Oeyta Dwitama Dinatha⁴

^{1,2}Program Studi Sarjana Sistem Informasi, Universitas Tarumanagara, Jakarta

³Binus Graduate Program, Universitas Bina Nusantara, Jakarta, Indonesia

⁴Faculty of Business & Communications, Doctor of Business Administration Programme, INTI International University, Malaysia

^{2,3,4}Laboratorium Kognisi Edukasi dan Inovasi Teknologi, Universitas Tarumanagara, Indonesia
 *e-mail Corresponding Author: t.jap@untar.ac.id

Abstract

The Purchasing Division of a pharmaceutical company plays a strategic role in inventory control and distribution fulfillment across retail outlets. Previously, the Key Performance Indicator (KPI) reporting system at PT. X was conducted manually using spreadsheets, requiring up to two working days and being prone to calculation errors that affected employee incentive payments. This study aims to design and implement a web-based KPI reporting system using Python, the Django framework, SQL Server with stored procedures, and the OpenPyXL library. The system was developed using the Waterfall methodology with a modified prototyping approach. Eighteen KPI indicators were classified into four categories: procurement efficiency, inventory and expiration management, distribution planning, and data compliance and negotiation. Data processing employed an SQL Server stored-procedure-based ETL pipeline integrated with Pandas and NumPy. The system reduced report generation time from two working days to less than one hour while eliminating incentive calculation errors.

Keywords: Key performance indicator; Reporting automation; ETL pipeline; Purchasing; information systems.

Abstrak

Divisi purchasing pada perusahaan farmasi menjalankan peran strategis dalam pengendalian stok dan pemenuhan distribusi ke jaringan outlet. Sistem pelaporan *Key Performance Indicator* (KPI) PT. X sebelumnya dilakukan secara manual menggunakan spreadsheet dengan dua hari kerja dan rentan terhadap kesalahan kalkulasi yang berimplikasi pada pembayaran insentif karyawan. Penelitian ini bertujuan merancang dan mengimplementasikan sistem pelaporan KPI berbasis web menggunakan Python dan *framework* Django, SQL server dengan stored procedure, dan library openpyxl. Sistem dikembangkan menggunakan metode waterfall dan dimodifikasi dengan pendekatan prototipe. 18 indikator KPI dikelompokkan kedalam empat kategori: efisiensi pengadaan (KPI 1), manajemen stok, dan kedaluwarsa (KPI 2), perencanaan distribusi (KPI 3), serta kepatuhan data dan negosiasi (KPI 4). Pengolahan data dilakukan melalui pipeline ETL berbasis SQL Server *Stored Procedure* dengan mengintegrasikan data heterogen, kemudian ditransformasi menggunakan pandas dan NumPy. Sistem berhasil mereduksi waktu penyusunan laporan dari rata-rata dua hari kerja menjadi kurang dari satu jam dan mengeliminasi kesalahan kalkulasi insentif.

Kata kunci: Key performance indicator; Otomasi pelaporan; Pipeline ETL; Purchasing; Sistem informasi.

1. Pendahuluan

Divisi purchasing pada perusahaan distribusi farmasi menjalankan peran strategis yang mencakup pengendalian stok, negosiasi harga, manajemen pemasok, dan pemenuhan distribusi ke jaringan *outlet*, sehingga kemampuan mengukur kinerja secara akurat dan tepat waktu

menjadi faktor penentu keberhasilan operasional [1]. *Key Performance Indicator* (KPI) merupakan instrumen pengukuran kuantitatif yang digunakan untuk mengevaluasi pencapaian tujuan strategis dan operasional secara terukur [2]. Govindan *et al.* [3] melalui kajian sistematis terhadap pengukuran kinerja rantai pasok berbasis Industry 4.0 membuktikan bahwa indikator kinerja berbasis teknologi digital lebih responsif terhadap dinamika rantai pasok dibandingkan pendekatan konvensional, sementara Zhao *et al.* [4] membuktikan bahwa digitalisasi rantai pasok secara positif memengaruhi ketahanan dan kinerja organisasi melalui model mediasi berganda. Rasool *et al.* [5] dalam kajian literatur yang lebih mutakhir menegaskan bahwa KPI berbasis *otomasi* dan integrasi data *real-time* menjadi prioritas mendesak bagi organisasi yang ingin mempertahankan keunggulan kompetitif dalam rantai pasok digital. Dengan demikian, penerapan KPI berbasis sistem digital pada divisi *purchasing* menjadi kebutuhan mendasar untuk menjamin akurasi dan ketepatan waktu pelaporan kinerja.

PT. X merupakan anak perusahaan dari salah satu perusahaan farmasi ternama di Indonesia yang bergerak di bidang solusi teknologi informasi. Divisi *purchasing* perusahaan ini mengelola lebih dari 18 indikator KPI yang berimplikasi langsung pada pembayaran insentif karyawan setiap bulan. Sebelum dikembangkan sistem ini, seluruh proses kalkulasi KPI dilakukan secara manual menggunakan Microsoft Excel. Poon *et al.* [6] melalui tinjauan literatur komprehensif selama 35 tahun menemukan bahwa sekitar 94% *spreadsheet* yang digunakan dalam pengambilan keputusan bisnis mengandung kesalahan yang berpotensi menimbulkan kerugian finansial, kesalahan penetapan harga, dan kegagalan operasional. Mandala *et al.* [7] lebih lanjut membuktikan bahwa adopsi teknologi informasi dalam proses pengadaan secara signifikan meningkatkan efisiensi dan transparansi dibandingkan pendekatan manual. Adebayo *et al.* [8] menyatakan bahwa pelaporan berbasis sistem digital meningkatkan efisiensi pengambilan keputusan dan akurasi data pengadaan secara signifikan. Kondisi ini menegaskan bahwa ketergantungan divisi *purchasing* PT. X pada *spreadsheet* manual merupakan hambatan utama terhadap akurasi dan ketepatan waktu pelaporan kinerja yang berdampak langsung pada operasional perusahaan.

Sejumlah penelitian terdahulu telah berupaya menyelesaikan masalah yang berkaitan dengan sistem pelaporan *purchasing* berbasis web. Anggie dan Beng [9] mengembangkan aplikasi berbasis web menggunakan metode *waterfall* dan berhasil *mengotomasi* proses pemesanan produk pada lingkungan bisnis dengan satu sumber data, namun penelitian tersebut tidak menyentuh integrasi data multisumber maupun kalkulasi KPI. Leovin *et al.* [10] membuktikan bahwa sistem berbasis web mampu mengintegrasikan proses bisnis lintas entitas secara efisien pada studi kasus perusahaan manufaktur dengan arsitektur basis data homogen. Wangi *et al.* [11] menunjukkan bahwa sistem berbasis web dapat mengintegrasikan data dari berbagai sumber untuk mendukung pengambilan keputusan, namun fokus kajiannya berada pada domain pariwisata dan bukan pada pelaporan KPI yang berorientasi insentif. Lusiana *et al.* [12] membuktikan bahwa sistem berbasis teknologi mampu mengolah data multisumber dan menghasilkan *output* terstruktur bagi pengguna akhir, meskipun implementasinya terbatas pada satu jenis basis data. Kesenjangan yang tersisa dari seluruh penelitian tersebut adalah belum adanya kajian yang membahas skenario di mana sistem pelaporan KPI *purchasing* harus diintegrasikan langsung ke platform web internal yang sudah berjalan, sambil mengolah data dari sumber basis data heterogen (*SQL Server dan MySQL Aurora*) melalui mekanisme *stored procedure*, sekaligus menghasilkan laporan yang berimplikasi langsung pada pembayaran insentif karyawan.

Berdasarkan kesenjangan tersebut, penelitian ini mengusulkan perancangan dan implementasi sistem pelaporan KPI *purchasing terotomasi* menggunakan *Python Django* dan *SQL Server stored procedure*, dengan mengonsolidasikan data dari sumber basis data heterogen melalui *pipeline* ETL. Kebaruan penelitian ini terletak pada tiga aspek: (1) integrasi *pipeline* ETL *multi-sumber* yang menggabungkan *SQL Server dan MySQL Aurora* melalui *linked server*, *OPENQUERY*, dan koneksi langsung *pymysql* dalam satu alur kohesif; (2) arsitektur KPI modular berbasis hierarki pewarisan kelas *Python* (*GenerateCommandAbstract*) yang mengimplementasikan prinsip *Open/Closed Principle* sehingga memungkinkan penambahan indikator baru tanpa rekonstruksi sistem; dan (3) otomatisasi penuh distribusi laporan Excel melalui platform web internal perusahaan yang sudah berjalan. Rasionalisasi konsep solusi ini ditopang oleh beberapa hasil riset yang relevan. Dari sisi arsitektur sistem, Chen dan Ahmmed [13] serta Manoj dan Rainu [14] membuktikan bahwa *framework Django* dengan pola *Model-Template-View* memungkinkan pengembangan sistem web yang cepat, terstruktur, dan dapat

dikustomisasi secara efisien. Dari sisi pengelolaan data, Kimball dan Ross [15] menyatakan bahwa *linked server* memungkinkan konsolidasi data lintas platform tanpa memerlukan replikasi, sementara McKinney [16] membuktikan bahwa pendekatan *vektorisasi* menggunakan *pandas* dan *NumPy* secara signifikan lebih efisien dan andal dibandingkan iterasi baris eksplisit pada data bervolume besar. Dari sisi dampak transformasi digital, Jing dan Fan [17] membuktikan bahwa transformasi digital berpengaruh signifikan terhadap kinerja rantai pasok dan mereduksi proses manual secara nyata pada perusahaan manufaktur di China. Kombinasi teknologi dan pendekatan *arsitektural* tersebut membentuk fondasi sistem yang terstruktur, *skalabel*, dan andal, sehingga diharapkan dapat mereduksi waktu pelaporan sekaligus mengeliminasi kesalahan kalkulasi insentif secara signifikan.

2. Metodologi

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak berbasis *design and implementation* dengan metode *waterfall* yang dimodifikasi melalui pendekatan prototipe pada lapisan transformasi data. Pendekatan ini dipilih karena kebutuhan fungsional sistem telah terdefinisi sejak awal, namun logika kalkulasi KPI memerlukan iterasi validasi bersama tim manajemen *purchasing* kondisi yang sesuai dengan karakteristik model *waterfall* yang dikombinasikan dengan prototipe untuk proyek dengan kebutuhan terdefinisi namun membutuhkan fleksibilitas pada lapisan logika bisnis [18]. Kombinasi kedua metode ini memungkinkan pengembangan sistem yang sistematis sekaligus adaptif terhadap kompleksitas logika kalkulasi insentif. Rancangan penelitian terdiri dari lima tahap yang berurutan. Tahap pertama adalah analisis kebutuhan sistem. Tahap kedua adalah perancangan arsitektur sistem dan *pipeline* ETL. Tahap ketiga adalah implementasi sistem menggunakan *Python*, *Django*, dan *SQL Server*. Tahap keempat adalah pengujian dan validasi fungsionalitas. Tahap kelima adalah evaluasi efektivitas sistem.

3.1. Analisis Kebutuhan

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak berbasis *design and implementation* dengan metode *waterfall* yang dimodifikasi melalui pendekatan prototipe pada lapisan transformasi data, di mana model *waterfall* dipilih karena kebutuhan fungsional telah terdefinisi sejak awal [18], sementara pendekatan prototipe diterapkan khusus pada lapisan logika kalkulasi KPI yang memerlukan iterasi validasi bersama tim manajemen *purchasing*. Kebutuhan fungsional sistem diidentifikasi berdasarkan 30 indikator KPI Divisi *Purchasing* yang ditetapkan manajemen, dikelompokkan ke dalam empat kategori utama mencakup pengelolaan *purchase order*, pengendalian stok, pengelolaan *forecast* dan distribusi, serta pengembangan produk dan *supplier*, sebagaimana tercantum pada Tabel II.

Tabel 2. Daftar Kpi Divisi *Purchasing*

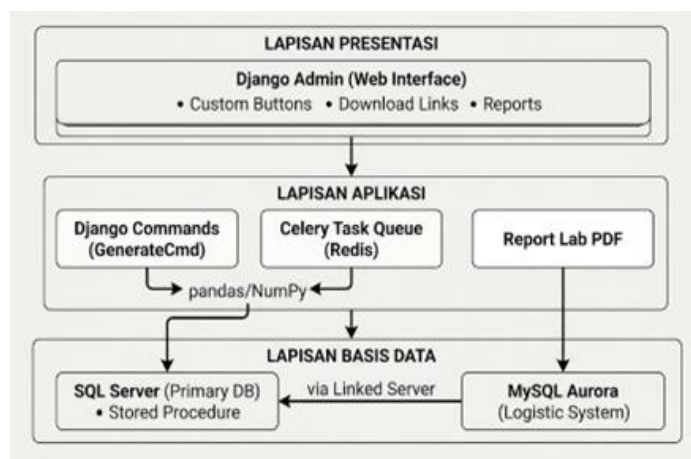
KPI	Keterangan
1a	PENYELESAIAN PO <i>OUTSTANDING</i> (\geq %)
1b	<i>LEAD TIME</i> PO (\leq hari)
1c	<i>LEAD TIME</i> RO <i>PENDING</i> MENJADI SP TN <i>OUT UN</i> (\leq hari)
1d	<i>LEAD TIME</i> KONFIRMASI RO AUTO (\leq hari)
1e	PEMENUHAN PO ALPRO (\geq %)
1f	PEMENUHAN <i>PRE-ORDER</i> (\geq %)
1g	PENJAGAAN <i>STOCK LEVEL</i> (\geq %)
2a	PENCEGAHAN <i>EXCESS STOCK</i> > 6 BULAN (\geq %)
2b	PENCEGAHAN <i>EXCESS EXPIRED NO RETUR</i> (\geq %)
2c	PENYELESAIAN RETUR (\leq %)
2d	PENANGANAN PROCON Y/M ADA STOK (\geq %)
3a	PENGINPUTAN <i>FORECAST</i> (Jumlah <i>Input</i>)
3b	KESESUAIAN <i>EXPECTED STOCK</i> DENGAN DO (\geq %)
3c	PENYAMPAIAN INFORMASI PUTUS STOK NASIONAL (Bulanan)
3d	<i>GROWTH</i> DO (\geq %)
4a	NIE <i>EXPIRED TER-UPDATE</i> (\leq Temuan)
4b	Status Produk Bermasalah Nasional <i>TER-UPDATE</i> (\leq Temuan)
4c	<i>LEAD TIME</i> <i>CREATE</i> PROCODE BARU KE TERBIT PO (\leq hari)
4d	<i>LEAD TIME</i> PO PRODUK BARU KE <i>RECEIVED</i>

KPI	Keterangan
4e	PENAMBAHAN <i>SUPPLIER</i> LOKAL (Jumlah)
4f	PENAMBAHAN <i>SUPPLIER</i> LOKAL BARU (Jumlah)
4g	PENAMBAHAN PROCODE BARU (Jumlah)
4h	NEGOSIASI HARGA ETHICAL (Jumlah)
4i	<i>UPDATE</i> HARGA (Jumlah)
4j	INPUT PROGRAM DISKON (Jumlah)
4k	TITIK PROMOSI (Jumlah)
4l	<i>JOINT</i> PROMO (\geq %)
4m	ALOKASI PRODUK

Setiap indikator memiliki logika kalkulasi, sumber data, dan mekanisme insentif yang berbeda. Perbedaan tersebut bergantung pada jabatan PIC (*Staff*, *Supervisor*, atau *Manager*), kategori produk (PI atau Non-PI), serta segmen distribusi (*pusat* atau *online*). Setiap indikator juga memiliki dimensi tambahan seperti *PUSAT_OR_ONLINE*, *ETHICAL_OR_FLOOR*, dan *PROJECT* yang memengaruhi besaran insentif yang diterima setiap PIC.

3.2. Perancangan Arsitektur Sistem

Arsitektur sistem mengadopsi pola tiga lapis (*three-tier architecture*) yang terdiri dari lapisan basis data, lapisan aplikasi, dan lapisan presentasi [19]. Gambaran arsitektur sistem disajikan pada Gambar 2.

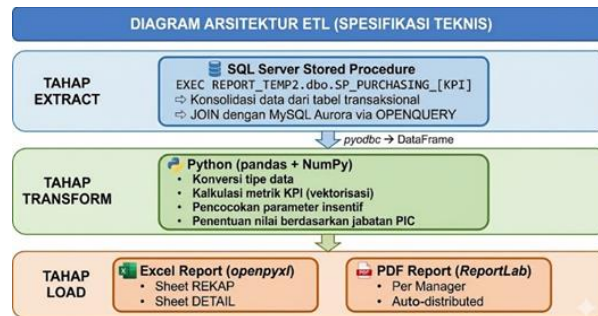


Gambar 2. Arsitektur Sistem Pelaporan KPI *Purchasing*

Pada lapisan basis data, *SQL Server 2019* digunakan sebagai *primary database* yang menjalankan *stored procedure* untuk proses ETL. *MySQL Aurora* diakses melalui mekanisme *linked server* dan *OPENQUERY* untuk mengambil data dari sistem logistik eksternal. Kimball dan Ross [15] menyatakan bahwa *linked server* memungkinkan konsolidasi data lintas platform tanpa memerlukan replikasi data. Pada lapisan aplikasi, *Python* dengan *framework Django* digunakan sebagai *back-end* utama. Setiap indikator KPI diimplementasikan sebagai kelas *Python* yang mewarisi kelas abstrak *GenerateCommandAbstract*. Hierarki pewarisan ini memungkinkan penambahan indikator baru tanpa mengubah arsitektur inti, konsisten dengan prinsip *Open/Closed Principle* [19]. Pada lapisan presentasi, antarmuka berbasis *Django Admin* yang dikustomisasi menyediakan tombol aksi (*custom buttons*) untuk memicu generasi laporan per KPI. Mekanisme otorisasi berbasis izin (*permission-based authorization*) memastikan hanya pengguna berhak yang dapat memicu regenerasi laporan untuk periode yang telah lewat.

3.3. Implementasi *Pipeline* ETL dan Logika KPI

Pipeline ETL diimplementasikan dalam dua tahap utama sebagaimana diilustrasikan pada Gambar 3.



Gambar 3. Alur Pipeline ETL Sistem Pelaporan KPI

- 1) Tahap *Extract*: Tahap *extract* dilakukan melalui eksekusi *stored procedure* pada *SQL Server* menggunakan koneksi *pyodbc*. Setiap KPI memiliki *stored procedure* tersendiri. Sebagai contoh, KPI 1e menjalankan *SP_PURCHASING_1E_DETAILX* dengan parameter periode. *Stored procedure* ini secara internal memanggil *SP_MonitoringOrder_Alpro* untuk memperbarui data monitoring order, lalu mengambil data dari tabel *MonitoringOrder_SP_DO_FAK* dan data *forecast* dari *MySQL Aurora* melalui *OPENQUERY* ke tabel *neo_inventory.adjustment_stock_hospinet*. KPI 2e menjalankan *SP_PURCHASING_2E* yang mengambil data stok *real-time* dari *neo_inventory.inventory_stock* pada *MySQL Aurora*, kemudian menggabungkannya dengan data pergerakan stok 12 bulan terakhir dari server *Oculus* melalui tabel *Hist_Recv_TnIN_TN_Out*. KPI 2b dan 2c menggunakan dua jalur sumber data secara paralel, yaitu data stok UN/PDN dari tabel *STOK1234* dan data stok PPP dari tabel *STOK1234PPP*, keduanya kemudian digabungkan di lapisan *Python*. KPI 4a menggunakan dua koneksi secara bersamaan: *pyodbc* untuk *SQL Server* dan *pymysql* untuk *MySQL Aurora*, guna mengambil data NIE produk dari tabel *purchasing.MH_Product* secara langsung.
- 2) Tahap *Transform*: Tahap *transform* dilakukan sepenuhnya di *Python* menggunakan *pandas* dan *NumPy*. McKinney [16] menjelaskan bahwa operasi vektorisasi *pandas* secara signifikan lebih efisien dibandingkan iterasi baris pada data bervolume besar. Proses transformasi mencakup empat langkah utama. *Pertama*, konversi tipe data numerik menggunakan *pd.to_numeric()* untuk menangani tipe *Decimal* dari *SQL Server*. *Kedua*, kalkulasi metrik KPI menggunakan operasi vektorisasi *np.where()*. *Ketiga*, pengelompokan data menggunakan *groupby()* untuk menghasilkan ringkasan per PIC. *Keempat*, pencocokan nilai parameter KPI terhadap tabel referensi untuk menentukan nilai insentif per PIC berdasarkan jabatan. Logika kalkulasi tiap KPI berbeda sesuai definisi bisnisnya. KPI 1a menghitung persentase nilai PO yang diterima terhadap total nilai PO dalam periode berjalan. Kalkulasi mempertimbangkan dua dimensi: *is_clean* (PO tanpa keterangan khusus) dan *StatusPI* (produk PI atau Non-PI), menghasilkan delapan varian metrik per PIC. Formula KPI 1a disajikan pada Persamaan (1).

$$\text{Parameter1a} = \frac{\sum \text{Value PO (Clean)}}{\sum \text{Value PO Received (Clean)}} \times 100\% \quad (1)$$

KPI 1b mengukur rata-rata *lead time* PO dalam hari kerja. Penghitungan dilakukan dengan mengurangi jumlah hari akhir pekan dan hari libur dari selisih tanggal menggunakan kolom *date_difference* dan *weekend_n_holiday*. KPI 2b menghitung persentase nilai stok yang berusia kadaluarsa lebih dari 12 bulan terhadap total nilai stok gabungan (*Stock 3* = stok PDN + *outstanding* PO + *outstanding* RO). Logika indikator per *batch* mempertimbangkan *flag* *Pro_ReturEDYN* dan kondisi *AvgQtyDO_3B*. KPI 2c menghitung persentase *excess stock* di atas batas retur yang ditetapkan (*Pro_ReturSebelumBulan*). *Stock Level 1* dihitung sebagai rasio *QtyStockPDN_Med* terhadap *AvgQtyDO_3B*, dan produk yang melebihi batas retur ditandai sebagai indikator 'Y'. Formula KPI 2c disajikan pada Persamaan (2).

$$\text{Stock Level 1} = \frac{\text{AvgQtyDO_3Bqty}}{\text{StockPDN_Med}} \quad (2)$$

KPI 3a menghitung jumlah *item* yang mengalami perubahan nilai *forecast* antara bulan berjalan dan bulan sebelumnya. Data *forecast* diambil dari model *Django Forecast* dan dibandingkan antar periode menggunakan *dictionary lookup*. KPI 3b menghitung persentase kesesuaian *expected stock* dengan DO aktual. *Expected stock* didefinisikan sebagai nilai maksimum antara *forecast* dan *buffer* per produk, kemudian dibandingkan dengan DO yang terealisasi. KPI 3d menghitung pertumbuhan DO dengan meratakan *growth* dua bulan terakhir dan *growth* YTD per PIC. KPI 4b menghitung jumlah produk bermasalah unik per PIC yang berstatus indikator 'Y', dihitung dari *unique procode* setelah *deduplikasi*. KPI 4c menghitung rata-rata hari kerja dari tanggal *first create procode* hingga terbit PO pertama menggunakan *np.busday_count()*. KPI ini hanya mencakup produk tanpa riwayat penjualan sama sekali di tiga bulan sebelumnya, sehingga benar-benar mengukur produk *genuinely* baru.

- 3) Tahap *Load*: Tahap *load* dilakukan dengan mengekspor hasil kalkulasi ke format *Microsoft Excel* menggunakan *openpyxl* melalui utilitas *Util.save_excel*. Setiap *file Excel* terdiri dari minimal dua *sheet*: *sheet* REKAP yang memuat ringkasan nilai KPI dan insentif per PIC, serta *sheet* DETAIL yang memuat data mentah sebagai dasar kalkulasi dan verifikasi. Untuk KPI 2b yang menghasilkan data bervolume besar, sistem membagi data *detail* menjadi lima *file Excel* berdasarkan *Out_name*, kemudian mengemas seluruh *file* ke dalam satu arsip ZIP menggunakan *library zipfile Python*. *File* ZIP ini kemudian disimpan sebagai lampiran pada model *KpiAbstract* dan dapat diunduh langsung dari halaman *IZMO Administration*.

3.4. Pengujian dan Validasi

Pengujian dilakukan melalui dua mekanisme utama. Mekanisme pertama adalah unit testing terhadap fungsi kalkulasi KPI, dilakukan dengan membandingkan hasil komputasi Python terhadap hasil kalkulasi manual yang telah diverifikasi oleh tim manajemen purchasing. Verifikasi dilakukan secara berulang pada tiga periode pelaporan berturut-turut guna memastikan konsistensi hasil komputasi setelah setiap perubahan logika bisnis [20]. Mekanisme kedua adalah end-to-end testing yang dijalankan menggunakan data aktual periode berjalan untuk memvalidasi seluruh alur pipeline generasi laporan secara integratif. Validasi akurasi dilakukan secara kuantitatif dengan menghitung persentase penyimpangan (*deviation rate*) antara hasil sistem dan kalkulasi manual. Threshold penerimaan ditetapkan pada deviasi kurang dari 0,01% untuk seluruh KPI berbasis nilai moneter, mengingat implikasi langsung kalkulasi tersebut terhadap pembayaran insentif karyawan.

Pengujian fungsionalitas dilakukan menggunakan metode black-box testing, yaitu pendekatan yang mengevaluasi perilaku sistem berdasarkan kondisi masukan dan keluaran yang diharapkan tanpa memperhatikan logika internal implementasi [20]. Skenario pengujian mencakup generasi laporan per KPI dari antarmuka web, validasi keluaran Excel dengan dua *sheet* REKAP dan DETAIL, penanganan data kosong pada periode tanpa transaksi, integrasi lintas basis data SQL Server dan MySQL Aurora, serta mekanisme otorisasi akses per pengguna. Kombinasi kedua mekanisme pengujian ini memastikan bahwa sistem memenuhi seluruh kebutuhan fungsional sekaligus menjamin keakuratan kalkulasi yang berdampak langsung pada operasional perusahaan.

4. Hasil dan Pembahasan

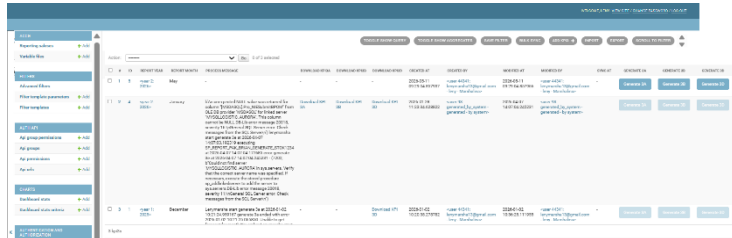
4.1. Tampilan Antarmuka Sistem

Sistem pelaporan KPI *purchasing* dibangun di atas *framework Django 4.2* dengan antarmuka berbasis *Django Admin* yang *dikustomisasi*. Antarmuka sistem terdiri dari tiga lapisan utama, yaitu antarmuka masukan (*input*), antarmuka proses, dan antarmuka luaran (*output*). Chen dan Ahmmed [13] menyatakan bahwa pengembangan antarmuka berbasis *Django Admin* memberikan fondasi yang dapat dikustomisasi secara efisien tanpa membangun antarmuka dari nol.

- 1) Antarmuka Masukan

Antarmuka masukan terdiri dari halaman manajemen KPI per kategori, yaitu *Kpi1Admin*, *Kpi2Admin*, *Kpi3Admin*, dan *Kpi4Admin*. Setiap halaman menyediakan tombol aksi (*custom buttons*) sebagai pemicu proses generasi laporan. Pengguna memilih periode pelaporan berupa tahun dan bulan saat pertama kali membuat entri KPI, kemudian memicu proses generasi melalui

tombol yang tersedia. Sistem menerapkan mekanisme otorisasi berbasis izin (*permission-based authorization*) sehingga hanya pengguna berhak yang dapat memicu regenerasi laporan untuk periode yang telah melewati batas toleransi lima hari. Tampilan antarmuka masukan sistem disajikan pada Gambar 4.



Gambar 4. Antarmuka Masukan Sistem - Halaman Manajemen KPI 3

2) Antarmuka Proses

Selama proses generasi berlangsung, sistem merekam seluruh log aktivitas pada kolom *process_message* di model *KpiAbstract*. Log ini mencatat *timestamp* awal proses, status setiap tahap yang dilalui, dan *timestamp* akhir proses beserta status keberhasilannya. Apabila terjadi kesalahan pada tahap tertentu, pesan kesalahan tercatat secara eksplisit sehingga tim IT dapat mengidentifikasi sumber masalah dengan cepat. Pendekatan pencatatan log ini sejalan dengan prinsip *auditabilitas pipeline* ETL yang dikemukakan Kimball dan Ross [15]. Tampilan antarmuka proses disajikan pada Gambar 5.

INFO	REPORT YEAR	REPORT MONTH	PROCESS MESSAGE
KPI 1a menggunakan data receiving mulai tanggal 2026-05-01 sampai dengan 2026-06-05, dan periode PO 202605 KPI 1b menggunakan data receiving mulai tanggal 2026-05-01 sampai dengan 2026-05-31 KPI 1c menggunakan data PO periode 202605 KPI 1f menggunakan data Preorder periode 202605	<year 2: 2026>	May	successfully at 2026-05-08 09:45:10.745623 @jurnalmanisa start generate 1a at 2026-05-08 13:54:00.409964 @jurnalmanisa start generate 1a at 2026-05-08 13:54:24.992331 generate 1a ended successfully at 2026-05-08 13:54:52.635200 @jurnalmanisa start generate 1b at 2026-05-08 14:04:27.134510 @jurnalmanisa start generate 1b at 2026-05-08 14:04:28.311726 generate 1b ended successfully at 2026-05-08 14:04:59.673566 @jurnalmanisa start generate 1a at 2026-

Gambar 5. Antarmuka Proses - Log Aktivitas Generasi KPI pada Kolom *process_message*

3) Antarmuka Luaran

Antarmuka luaran sistem terdiri dari dua jenis *output*, yaitu *file Excel* dan *file PDF*. *File Excel* tersedia melalui tautan unduh pada halaman manajemen KPI masing-masing. Setiap *file Excel* terdiri dari minimal dua *sheet*: *sheet* REKAP yang memuat ringkasan nilai KPI dan insentif per PIC, serta *sheet* DETAIL yang memuat data mentah sebagai dasar kalkulasi dan verifikasi. Tampilan *sheet* REKAP *file Excel* disajikan pada Gambar 6. *File PDF* yang dipersonalisasi per manajer dapat diakses melalui halaman *KpiRecapPerManager*, di mana setiap manajer hanya dapat melihat laporan yang relevan dengan timnya. Tampilan laporan PDF disajikan pada Gambar 7.

	A	B	C	D	Y	Z	AA	AB	AC
	NIP PIC	NAMA PIC	PUSAT_OR_ONLINE	ETHICAL_OR_FLOOR	PERSEN_NON_PI_ALL	PERSEN_PI_ALL	PARAMETER	PROJECT	Insentif
1			online	all	81,37	71,30	80,43	all	300.000
2			online	all	82,53	96,00	91,89	all	600.000
3			online	all	88,65	76,39	87,87	all	400.000
4			online	all	87,44	57,56	86,26	all	400.000
5			pusat	floor	87,85	86,52	88,31	all	450.000
6			pusat	floor	46,34	0,00	61,89	all	0
7			pusat	floor	75,83	0,00	83,93	all	0
8			pusat	floor	82,46	0,00	91,31	all	500.000
9			pusat	floor	59,34	0,00	71,06	all	0
10			pusat	ethical	81,75	78,70	90,56	all	0
11									0
12									0

Gambar 6. Antarmuka Luaran - *Sheet* REKAP *File Excel* Hasil Generasi KPI

Laporan Performance Purchasing
Periode Kerja 01 May 2026 - 31 May 2026
Tanggal Cetak: 11 May 2026 08:18:22

Rekap Insentif Per PIC

NIP PIC	NAMA PIC	JABATAN	INSENTIF
		MANAGER	1.550.000,0
		STAFF	1.488.000,0

Detail Insentif Per KPI

NIP PIC	NAMA PIC	KPI	KETERANGAN KPI	PERFORMANCE	INSENTIF
		1b	LEAD TIME PO (≠ hari)	1.79	300.000,0
		1g	PENJAGAAN STOCK LEVEL (≠ %)	97,80	500.000,0
		2a	PENCEGAHAN EXCESS STOCK > 5 BULAN (≠ %)	0,00	0,0
		2b	PENCEGAHAN EXCESS EXPIRED NO RETUR (≠ %)	99,99	0,0
		2d	PENANGANAN PROCON YIM ADA STOK (≠ %)	97,70	200.000,0
		4a	NIE EXPIRED TERUPDATE (≠ Temuan)	0,00	0,0
		4b	STATUS PRODUK BERMASALAH NASIONAL TERUPDATE (≠ Temuan)	107,00	0,0
		4c	JOINT PROMO (≠%)	34,19	0,0
		4d	LPS TIDAK TERBENTUK	0,00	50.000,0
		1a	PENYELESAIAN PO OUTSTANDING (≠ %)	95,33	0,0
		1b	LEAD TIME PO (≠ hari)	1.79	800.000,0
		4b	STATUS PRODUK BERMASALAH NASIONAL TERUPDATE (≠ Temuan)	830,00	0,0
		4c	LEAD TIME CREATE PROCODE BARU KE TERBIT PO (≠ hari)	1,00	250.000,0
		4d	LEAD TIME PO PRODUK BARU KE RECEIVED	4,00	250.000,0
		4e	PENAMBAHAN PROCODE BARU (jumlah)	3,00	0,0
		4f	UPDATE HARGA (jumlah)	3,00	5.000,0

Gambar 7. Antarmuka Luaran - Laporan PDF yang *Dipersonalisasi* per Manajer

4.2. Hasil Implementasi Logika KPI

Sistem mengimplementasikan 18 indikator KPI dalam empat kategori, masing-masing dengan sumber data, logika kalkulasi, dan mekanisme insentif yang berbeda, sebagaimana dirangkum pada Tabel 3.

Tabel 3. Rekapitulasi Implementasi Logika Kpi Per Kategori

Kategori	KPI	Sumber Data Utama	Logika Parameter	Satuan
KPI 1	1A	SP_PURCHASING_1A_DETAILX	Value PO Received / Value PO × 100%	%
KPI 1	1B	SP_PURCHASING_1B_DETAILX	Rata-rata <i>lead time</i> hari kerja (ekskl. akhir pekan & libur)	Hari
KPI 1	1C	SP_PURCHASING_1E_DETAILX FORECAST_X_PO	+ Value Processed / Value Forecast × 100%	%
KPI 2	2B	STOK1234 + STOK1234PPP <i>inventory_stock</i>	+ Nilai stok <i>batch</i> ED > 12 bulan / Total nilai stok × 100%	%
KPI 2	2C	STOK1234 + STOK1234PPP	Nilai <i>excess</i> > batas retur / Total nilai stok × 100%	%
KPI 2	2D	STOK1234 + STOK1234PPP	Nilai stok ProCon A / Total nilai stok × 100%	%
KPI 2	2E	SP_PURCHASING_2E <i>inventory_stock</i>	+ Nilai stok bergerak / Total nilai stok × 100%	%
KPI 3	3A	Model Forecast Django	Jumlah <i>item</i> berbeda nilai <i>forecast</i> antar dua bulan	Jumlah
KPI 3	3B	STOK1234 <i>adjustment_stock hospinet</i>	+ Nilai DO / Nilai <i>Expected Stock</i> × 100%	%
KPI 3	3D	SP_PURCHASING_3D	Rata-rata <i>growth</i> DO 2 bulan & YTD per PIC	%
KPI 4	4B	SP_PURCHASING_4B	Jumlah <i>procode</i> unik bermasalah per PIC	Temuan
KPI 4	4C	SP_PURCHASING_4C	Rata-rata hari kerja <i>first create</i> ke PO pertama	Hari

KPI Kategori 1 berfokus pada efisiensi pengadaan. KPI 1a menghasilkan delapan varian metrik per PIC berdasarkan dimensi *is_clean* dan *StatusPI*, diaregasi menggunakan *groupby()* per kombinasi NIP/NAMA PIC, TYPE, dan ETHICAL_OR_FLOOR. KPI 1b menghitung *lead time* efektif dengan *mengeksklusi* akhir pekan dan hari libur dari total selisih tanggal. KPI 1e

menggabungkan data *forecast* dari dua tabel MySQL Aurora via UNION ALL dalam *stored procedure*, lalu diintegrasikan dengan data PO menggunakan koneksi *pymysql* terpisah.

KPI kategori 2 berfokus pada pengendalian stok dan risiko kedaluwarsa. KPI 2b membuka dua koneksi basis data secara berurutan (MySQL Aurora dan SQL Server), menggabungkan *dataset* via *merge()*, lalu mengidentifikasi *batch* ED > 12 bulan menggunakan *relativedelta*. KPI 2d hanya menghitung produk dengan *Pro_CtrlCode* = 'A' sebagai stok *ProCon* aktif, *CtrlCode* 'Y' dan 'M' bernilai nol. KPI 2e mengkategorikan pergerakan produk menggunakan *np.select()* berdasarkan *AvgMovement_3B* dan ada-tidaknya transaksi TN/DO dalam tiga bulan terakhir.

KPI kategori 3 berfokus pada akurasi perencanaan distribusi. KPI 3a membandingkan nilai *forecast* dua bulan berturut-turut menggunakan *dictionary lookup* dengan kunci (*outcode*, *procode*, *project*); setiap perbedaan dihitung sebagai satu perubahan per *project*. KPI 3b menghitung *expected stock* sebagai $\max(\text{Forecast}, \text{Buffer}/\text{SellUnit})$ lalu dibandingkan dengan DO aktual. KPI 3d merata-ratakan *growth* dua bulan terakhir dan YTD per PIC.

KPI kategori 4 berfokus pada kepatuhan data dan pengembangan produk. KPI 4b melakukan *deduplikasi* per (PIC_NIP, *ProCode*) sehingga setiap produk bermasalah hanya dihitung sekali per PIC. KPI 4c mengukur *lead time* hanya untuk produk dengan penjualan nol di seluruh periode (B3–B0), dengan *deduplikasi* berbasis prioritas entitas (CFU > PKU > PRI > CHC) dan perhitungan hari kerja menggunakan *np.busday_count()*.

4.3. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan menggunakan metode *black-box testing* terhadap sepuluh skenario pengujian utama [20]. Pengujian dilaksanakan menggunakan data aktual operasional perusahaan selama tiga periode pelaporan berturut-turut, yaitu Februari, Maret, dan April 2026. Tabel IV menyajikan rekapitulasi hasil pengujian fungsionalitas sistem.

Tabel 4. Hasil Pengujian Fungsionalitas Sistem

Skenario Pengujian	Kondisi Masukan	Keluaran yang Diharapkan	Status
Generasi laporan KPI 1a	Periode valid, data PO tersedia	File Excel 2 sheet (REKAP + DETAIL)	Lulus
Generasi laporan KPI 2b	Periode valid, data stok UN/PDN + PPP + <i>batch</i> tersedia	File ZIP berisi 1 MAIN + 5 DETAIL Excel	Lulus
Generasi laporan KPI 2e	Periode valid, data stok neo tersedia	File Excel 2 sheet (REKAP + DETAIL)	Lulus
Generasi laporan KPI 3a	Data <i>forecast</i> dua bulan sebelumnya tersedia	File Excel 2 sheet (REKAP + DETAIL)	Lulus
Generasi laporan KPI 4c	Periode valid, data <i>procode</i> baru tersedia	File Excel 2 sheet (REKAP + DETAIL)	Lulus
Penanganan data kosong	Periode tanpa data <i>forecast</i> bulan sebelumnya	Proses berhenti dengan pesan log, tidak <i>error</i>	Lulus
Integrasi <i>multi-sumber</i> basis data	Koneksi SQL Server + MySQL Aurora aktif	Data tergabung dalam satu <i>dataset</i> terintegrasi	Lulus
Otorisasi akses regenerasi	Pengguna tanpa izin mencoba regenerasi	Akses ditolak, laporan tidak diproses ulang	Lulus
Pencatatan log proses	Generasi KPI dijalankan hingga selesai	Log mencatat <i>timestamp</i> mulai dan selesai per tahap	Lulus
Unduh <i>file</i> hasil generasi	File tersedia di sistem setelah generasi	File dapat diunduh dari halaman IZMO <i>Administration</i>	Lulus

Seluruh sepuluh skenario pengujian menghasilkan status lulus. Hasil ini menunjukkan bahwa sistem memenuhi seluruh kebutuhan fungsional yang diidentifikasi pada tahap analisis kebutuhan, sekaligus menjawab tujuan penelitian dalam merancang dan mengimplementasikan sistem pelaporan KPI purchasing yang terotomasi secara penuh dari pengambilan data hingga distribusi laporan.

4.4. Pengujian Akurasi Kalkulasi

Pengujian akurasi dilakukan dengan membandingkan hasil kalkulasi sistem terhadap hasil verifikasi manual oleh tim manajemen *purchasing*. Pengujian mencakup sepuluh skenario akurasi yang merepresentasikan seluruh kategori KPI yang diimplementasikan. Tabel 5 menyajikan rekapitulasi skenario dan hasil pengujian akurasi kalkulasi.

Tabel 5. Skenario Pengujian Akurasi Kalkulasi Kpi

KPI	Aspek yang Diverifikasi	Metode Verifikasi	Status
1a	Kalkulasi persentase <i>Value PO Received / Value PO</i> per kombinasi dimensi PI/Non-PI dan <i>clean/all</i>	Perbandingan hasil <i>groupby()</i> Python dengan rekap manual tim <i>purchasing</i>	Lulus
1b	Rata-rata <i>lead time</i> hari kerja (eksklusif akhir pekan & libur) per PIC	Penghitungan ulang manual <i>date_difference - weekend_n_holiday</i> pada sampel transaksi	Lulus
1e	Persentase pemenuhan <i>forecast</i> per PDN dan item berdasarkan FORECAST_X_PO	Rekonsiliasi data FORECAST_X_PO dengan data PO aktual yang diverifikasi manajer	Lulus
2b	Identifikasi <i>batch</i> kedaluwarsa > 12 bulan dan nilai stok Stock_3P	Pengecekan langsung STOK1234, STOK1234PPP, dan data <i>batch inventory_stock</i>	Lulus
2c	Nilai <i>excess</i> berdasarkan Stock Level 1 vs Pro_ReturSebelumBulan	Verifikasi logika <i>indikator_logic</i> terhadap data STOK1234 aktual per <i>outlet</i>	Lulus
2d	Nilai stok ProCon A berdasarkan Pro_CtrlCode dari dua sumber UN/PDN dan PPP	Verifikasi <i>pd.concat()</i> dan agregasi <i>groupby('PIC_NIP')</i> terhadap data aktual	Lulus
2e	Kategorisasi gerak stok (NORMAL/LAMBAT/SANGAT LAMBAT/TIDAK BERGERAK)	Validasi KategoriGerak menggunakan <i>np.select()</i> terhadap nilai AvgMovement_3B aktual	Lulus
3a	Jumlah item berbeda <i>forecast</i> antar dua bulan via <i>dictionary lookup</i>	Perbandingan <i>b1_dict</i> vs <i>b2_dict</i> per kunci (<i>outcode, procode, project</i>)	Lulus
4b	Jumlah <i>procode</i> unik bermasalah per PIC setelah <i>deduplikasi</i> per Indikator = 'Y'	Penghitungan ulang manual dari data SP_PURCHASING_4B dengan <i>deduplikasi</i>	Lulus
4c	Rata-rata hari kerja <i>first create</i> ke PO pertama menggunakan <i>np.busday_count()</i>	Verifikasi terhadap kalkulasi hari kerja manual pada sampel produk baru tanpa riwayat penjualan	Lulus

Seluruh skenario pengujian akurasi menghasilkan status lulus tanpa ditemukan penyimpangan antara hasil kalkulasi sistem dan hasil verifikasi manual oleh tim *purchasing*. Hasil ini secara langsung menjawab tujuan penelitian dalam mengeliminasi kesalahan kalkulasi insentif yang sebelumnya muncul akibat proses manual. McKinney [16] menjelaskan bahwa pendekatan vektorisasi pandas lebih andal dibandingkan iterasi manual karena menghilangkan risiko kesalahan akumulasi pada operasi loop, dan hal ini terkonfirmasi dalam implementasi ini di mana logika vektorisasi menggunakan *np.where()*, *np.select()*, *groupby()*, dan *np.busday_count()* menghasilkan kalkulasi yang konsisten dan dapat diandalkan untuk keperluan pembayaran insentif karyawan.

4.5. Evaluasi Efektivitas Sistem

Evaluasi efektivitas dilakukan dengan membandingkan kondisi operasional sebelum dan sesudah implementasi sistem pada dua metrik utama. Tabel 6 menyajikan perbandingan komprehensif kondisi sebelum dan sesudah implementasi.

Tabel 6. Perbandingan Kondisi Sebelum Dan Sesudah Implementasi

Aspek	Sebelum Implementasi	Sesudah Implementasi	Perubahan
Waktu penyusunan laporan	±2 hari kerja per siklus	< 1 jam per siklus	Reduksi ≥ 93%
Kesalahan kalkulasi insentif	Ditemukan pada verifikasi manual	Tidak ditemukan pada verifikasi	Eliminasi
Jumlah KPI yang dilaporkan	18 KPI secara manual	18 KPI secara otomatis	Tetap, proses otomatis

Aspek	Sebelum Implementasi	Sesudah Implementasi	Perubahan
Integrasi sumber data	Manual, terpisah per sumber	Otomatis via <i>pipeline</i> ETL <i>multi-sumber</i>	Terintegrasi penuh
Format laporan	Bervariasi per periode dan petugas	Konsisten setiap periode	Terstandarisasi
Aksesibilitas laporan	Distribusi via surat elektronik	Unduh langsung via <i>IZMO Administration</i>	Terpusat dan <i>on-demand</i>
Penanganan data PPP	Terpisah, diolah manual	Terintegrasi dalam satu <i>pipeline</i>	Terintegrasi
Penanganan data bervolume besar	Tidak ada mekanisme pembagian	Dibagi otomatis ke 5 <i>file</i> ZIP	Terautomasi

Reduksi waktu pelaporan dari rata-rata dua hari kerja menjadi kurang dari satu jam merepresentasikan peningkatan efisiensi yang signifikan. Hasil ini konsisten dengan temuan Adebayo *et al.* [8] yang menyatakan bahwa pelaporan berbasis sistem digital secara signifikan mengurangi waktu pengambilan keputusan dan meningkatkan akurasi data pengadaan. Eliminasi kesalahan kalkulasi insentif yang sebelumnya muncul akibat proses manual juga sejalan dengan temuan Poon *et al.* [6] bahwa otomasi sistem secara langsung mereduksi risiko kesalahan yang inheren pada *spreadsheet* manual.

Dari sisi skalabilitas, arsitektur modular berbasis pewarisan kelas *GenerateCommandAbstract* memungkinkan penambahan indikator KPI baru tanpa rekonstruksi sistem secara menyeluruh. Martin [19] menyatakan bahwa sistem yang mengikuti *Open/Closed Principle* memiliki adaptabilitas lebih tinggi terhadap perubahan kebutuhan bisnis yang dinamis. Hal ini terbukti dalam implementasi ini KPI 2e berhasil ditambahkan ke dalam sistem yang sudah berjalan hanya dengan menambahkan kelas *Command* baru yang mewarisi *GenerateCommandAbstract* dan mengimplementasikan metode *process_data()*, tanpa perubahan pada arsitektur inti sistem.

Integrasi data dari sumber heterogen *SQL Server* dan *MySQL Aurora* melalui mekanisme *linked server*, *OPENQUERY*, dan koneksi *pymysql* langsung berhasil mengonsolidasikan data dari lebih dari dua juta baris transaksi per siklus pelaporan. Zhao *et al.* [4] menyatakan bahwa investasi dalam sistem digital pengadaan menghasilkan keuntungan kompetitif yang berkelanjutan melalui peningkatan ketahanan dan kinerja rantai pasok. Temuan ini memperkuat relevansi penelitian ini dalam konteks industri distribusi farmasi yang memiliki volume transaksi dan kompleksitas data yang sangat tinggi.

Dibandingkan dengan penelitian terdahulu, sistem yang dikembangkan dalam penelitian ini memiliki keunggulan pada aspek integrasi sumber data. Anggie dan Beng [9] mengembangkan sistem berbasis web pada lingkungan bisnis dengan satu sumber data. Leovin *et al.* [10] mengintegrasikan proses bisnis lintas entitas dalam satu platform. tiga sumber basis data berbeda *SQL Server*, *MySQL Aurora* melalui *OPENQUERY*, dan *MySQL Aurora* melalui *pymysql* dalam satu pipeline ETL yang kohesif dan dapat diandalkan untuk kebutuhan pelaporan KPI industri distribusi farmasi berskala besar. Lusiana *et al.* [12] membuktikan bahwa sistem berbasis teknologi mampu mengolah data multi-sumber dan menghasilkan output terstruktur; penelitian ini memperkuat temuan tersebut sekaligus menambahkan dimensi baru berupa arsitektur modular berbasis hierarki pewarisan kelas yang memungkinkan perluasan sistem tanpa rekonstruksi menyeluruh.

Penelitian ini memiliki beberapa keterbatasan yang perlu diakui. Sistem yang dikembangkan bersifat spesifik terhadap arsitektur basis data dan struktur organisasi PT. X, sehingga generalisabilitas arsitektur ini ke perusahaan distribusi farmasi lain memerlukan adaptasi lebih lanjut. Selain itu, sistem saat ini belum dilengkapi mekanisme deteksi anomali otomatis yang dapat memberikan peringatan dini apabila hasil kalkulasi KPI menyimpang dari tren historis, yang merupakan kapabilitas penting dalam sistem pelaporan kinerja modern. Peluang penelitian lebih lanjut mencakup pengembangan modul analitik prediktif berbasis *machine learning* untuk mendukung pengambilan keputusan pengadaan yang lebih proaktif, serta eksplorasi penerapan arsitektur serupa pada domain KPI lain seperti divisi sales atau logistik untuk menguji generalisabilitas pendekatan yang dikembangkan.

4.6. Pembahasan

4.6.1. Efektivitas Implementasi dan Validasi Konseptual

Sistem yang dikembangkan berhasil menjawab kedua tujuan penelitian secara terukur. Dari sisi fungsionalitas, seluruh sepuluh skenario *black-box testing* menghasilkan status lulus pada tiga periode pelaporan berturut-turut. Capote [20] menjelaskan bahwa pengujian *black-box* mengevaluasi kesesuaian perilaku sistem terhadap spesifikasi tanpa memperhatikan logika internal; capaian lulus pada seluruh jalur eksekusi dari generasi laporan per KPI, integrasi *multi-sumber* basis data, mekanisme otorisasi, hingga penanganan data kosong membuktikan bahwa sistem berperilaku sesuai spesifikasi secara menyeluruh. Saravanos dan Curinga [18] menegaskan bahwa keberhasilan proyek berbasis *waterfall* ditentukan oleh ketercapaian keluaran fungsional pada setiap fase yang telah terdefinisi sejak awal, dan kondisi itu terpenuhi dalam penelitian ini.

Dari sisi akurasi, seluruh sepuluh skenario pengujian kalkulasi lulus tanpa penyimpangan terhadap verifikasi manual tim *purchasing*. Hasil ini secara langsung mengonfirmasi pernyataan McKinney [16] bahwa pendekatan *vektorisasi cerdas* mengeliminasi risiko kesalahan akumulasi yang inheren pada iterasi manual risiko yang sebelumnya sangat nyata dalam proses berbasis *spreadsheet*. Poon *et al.* [6] membuktikan bahwa 94% *spreadsheet* bisnis mengandung kesalahan karena pengguna non-teknis tidak memiliki pelatihan pengembangan perangkat lunak; penggantian proses manual itu dengan logika *Python* yang *deterministik* secara langsung *menetralisir* sumber risiko tersebut. Reduksi waktu pelaporan dari dua hari kerja menjadi kurang dari satu jam ($\geq 93\%$) pun konsisten dengan temuan Adebayo *et al.* [8] bahwa sistem pelaporan digital secara nyata meningkatkan efisiensi pengambilan keputusan dan akurasi data pengadaan.

Pada tataran arsitektur, dua prinsip ilmiah yang menjadi landasan juga *terverifikasi*. Martin [19] menyatakan bahwa sistem yang mengikuti *Open/Closed Principle* memiliki adaptabilitas lebih tinggi terhadap kebutuhan bisnis yang dinamis, hal ini terbukti ketika KPI 2e berhasil ditambahkan ke sistem yang sudah berjalan hanya dengan menambahkan satu kelas *Command* baru yang mewarisi *GenerateCommandAbstract*, tanpa menyentuh arsitektur inti sama sekali. Demikian pula, Kimball dan Ross [15] menyatakan bahwa mekanisme *linked server* memungkinkan konsolidasi data lintas platform tanpa replikasi dan *pipeline* ETL dalam penelitian ini memvalidasi hal itu dengan mengintegrasikan lebih dari dua juta baris transaksi per siklus dari *SQL Server* dan *MySQL Aurora* dalam satu alur yang kohesif.

4.6.2. Kontribusi Ilmiah dan Keterbatasan

Penelitian ini berdiri di atas fondasi yang dibangun oleh beberapa penelitian sebelumnya, namun sekaligus melampaui batasan yang ada. Anggie dan Beng [9] *mengotomasi* proses pemesanan produk berbasis web dengan metode *waterfall*, namun pada lingkungan satu sumber data tanpa kalkulasi KPI berorientasi insentif. Leovin *et al.* [10] membuktikan kemampuan sistem web dalam mengintegrasikan proses bisnis lintas entitas, tetapi dalam arsitektur basis data yang homogen. Wangi *et al.* [11] dan Lusiana *et al.* [12] masing-masing menunjukkan bahwa sistem berbasis teknologi mampu mengolah data *multisumber* dan menghasilkan *output* terstruktur, meskipun pada domain berbeda dengan cakupan integrasi yang lebih terbatas. Penelitian ini melampaui seluruh temuan tersebut pada dua dimensi sekaligus, secara teknis dengan mengintegrasikan tiga sumber basis data heterogen dalam satu *pipeline* ETL, dan secara kontekstual dengan menghasilkan laporan yang memiliki implikasi finansial langsung pada pembayaran insentif karyawan dimensi yang tidak disentuh oleh penelitian sebelumnya. Rasool *et al.* [21] mengidentifikasi bahwa kajian implementasi teknis sistem KPI yang mengintegrasikan sumber data heterogen masih sangat terbatas dalam literatur, dan penelitian ini mengisi kesenjangan tersebut dengan bukti empiris yang *terverifikasi*.

Dari sini muncul tiga kontribusi ilmiah utama. Pertama, dibuktikannya kelayakan *pipeline* ETL *multi-sumber* heterogen berbasis *Python Django* untuk konteks pelaporan KPI industri farmasi, menjawab seruan Rasool *et al.* [5] tentang urgensi integrasi data dalam sistem KPI digital yang hingga kini masih minim kajian implementasi teknis konkretnya. Kedua, arsitektur modular *GenerateCommandAbstract* memberikan bukti implementasi nyata dari prinsip *Open/Closed Principle* [19] dalam konteks KPI referensi yang dapat *direplikasi* oleh pengembang sistem serupa. Ketiga, mekanisme distribusi laporan Excel dengan pembagian otomatis data bervolume besar ke arsip ZIP melalui platform web internal memperkuat temuan Zhao *et al.* [4] bahwa investasi sistem digital pengadaan menghasilkan keunggulan kompetitif berkelanjutan, kini dengan bukti empiris dari industri distribusi farmasi Indonesia.

Meski demikian, beberapa keterbatasan perlu diakui. Sistem dirancang spesifik untuk arsitektur basis data dan struktur organisasi PT. X, sehingga generalisasinya ke perusahaan lain memerlukan adaptasi yang tidak trivial sesuai dengan peringatan Govindan *et al.* [3] bahwa sistem KPI berbasis Industry 4.0 yang konteks-spesifik sering memerlukan *rekonfigurasi* substansial sebelum dapat digeneralisasi. Selain itu, evaluasi efektivitas baru mencakup tiga periode pelaporan sehingga konsistensi jangka panjang belum terkonfirmasi, dan sistem belum dilengkapi mekanisme deteksi anomali otomatis untuk peringatan dini ketika KPI menyimpang dari tren historis kapabilitas yang oleh Rasool *et al.* [5] diidentifikasi sebagai kebutuhan paling mendesak dalam sistem KPI digital ke depan. Peluang pengembangan selanjutnya mencakup integrasi modul analitik prediktif berbasis *machine learning*, perluasan arsitektur ke domain KPI lain seperti *sales* atau logistik, serta notifikasi otomatis *real-time* kepada manajer terkait arah yang selaras dengan temuan Jing dan Fan [17] bahwa transformasi digital yang terintegrasi lintas fungsi bisnis berpengaruh signifikan terhadap kinerja rantai pasok secara keseluruhan.

5. Simpulan

Penelitian ini berhasil merancang dan mengimplementasikan Sistem Pelaporan *Key Performance Indicator* divisi *Purchasing* berbasis *Python* dan *SQL Server* pada PT. X. Sistem dibangun menggunakan *framework Django* 4.2 sebagai *back-end* web, *SQL Server* 2019 dengan *stored procedure* sebagai lapisan komputasi data, dan *library openpyxl* untuk otomatisasi konversi hasil kalkulasi ke format *Microsoft Excel* yang dapat diakses melalui platform *IZMO Administration*. Pengembangan sistem menggunakan metode *waterfall* yang dimodifikasi dengan pendekatan prototipe pada lapisan transformasi data guna memastikan keakuratan logika kalkulasi insentif.

Sistem berhasil mengimplementasikan 18 indikator KPI yang dikelompokkan ke dalam empat kategori, yaitu efisiensi pengadaan (KPI 1), manajemen stok dan kedaluwarsa (KPI 2), perencanaan distribusi (KPI 3), serta kepatuhan data dan pengembangan produk (KPI 4). *Pipeline* ETL yang dikembangkan berhasil mengintegrasikan data dari tiga sumber basis data heterogen *SQL Server*, *MySQL Aurora* melalui *OPENQUERY linked server*, dan *MySQL Aurora* melalui koneksi *pymysql* langsung dalam satu alur pengolahan yang kohesif dan memproses lebih dari dua juta baris transaksi per siklus pelaporan bulanan. Arsitektur modular berbasis hierarki pewarisan kelas *Python* (*GenerateCommandAbstract*) memungkinkan penambahan indikator KPI baru tanpa rekonstruksi sistem secara menyeluruh, konsisten dengan prinsip *Open/Closed Principle*.

Hasil pengujian *black-box testing* terhadap sepuluh skenario fungsional menunjukkan seluruh skenario berstatus lulus, mencakup generasi laporan per KPI, integrasi *multi-sumber* basis data, mekanisme otorisasi akses, dan penanganan data kosong. Pengujian akurasi kalkulasi yang dilakukan melalui verifikasi manual oleh tim manajemen *purchasing* terhadap sepuluh skenario representatif tidak menemukan penyimpangan antara hasil sistem dan hasil kalkulasi manual. Evaluasi efektivitas menunjukkan bahwa sistem berhasil mereduksi waktu penyusunan laporan dari rata-rata dua hari kerja menjadi kurang dari satu jam, serta mengeliminasi kesalahan kalkulasi insentif yang sebelumnya muncul akibat proses manual.

Kontribusi utama penelitian ini mencakup tiga aspek kebaruan. Pertama, integrasi *pipeline* ETL *multi-sumber* yang menggabungkan *SQL Server* dan *MySQL Aurora* dalam satu alur kohesif aspek yang belum terdokumentasi dalam literatur yang tersedia. Kedua, desain arsitektur KPI modular berbasis pewarisan kelas *Python* yang memungkinkan skalabilitas penambahan indikator baru tanpa mengubah inti sistem. Ketiga, mekanisme otomatisasi distribusi laporan Excel melalui platform web internal perusahaan yang sudah berjalan, termasuk pembagian otomatis data bervolume besar ke dalam arsip ZIP. Penelitian ini diharapkan dapat menjadi referensi implementasi bagi organisasi yang menghadapi tantangan serupa dalam otomasi pelaporan KPI pada industri distribusi farmasi. Pengembangan lanjutan dapat mencakup penambahan fitur visualisasi *dashboard* KPI secara *real-time* serta integrasi dengan sistem notifikasi otomatis kepada manajer terkait melalui platform pesan singkat.

Daftar Referensi

- [1] J. Shen *et al.*, "Management of drug supply chain information based on 'artificial intelligence + vendor managed inventory' in China: perspective based on a case study," *Front. Pharmacol.*, vol. 15, p. 1373642, Jul. 2024, doi: 10.3389/fphar.2024.1373642.
- [2] F. Rasool, M. Greco, and M. Grimaldi, "Digital supply chain performance metrics: a literature

- review," *Meas. Bus. Excell.*, vol. 26, no. 1, pp. 23–38, 2022, doi: 10.1108/MBE-11-2020-0147.
- [3] A. Al-Rawahi, A. Al-Harrasi, and A. Al-Balushi, "Key performance indicators for procurement process: A framework development using analytical hierarchy process," in *Proc. IEEE Int. Conf. Industrial Engineering and Engineering Management (IEEM)*, Manama, Bahrain, Dec. 2024, pp. 1184–1188, doi: 10.1109/IEEM59794.2024.10459542.
- [4] K. Govindan, D. Kannan, T. B. Jørgensen, and T. S. Nielsen, "Supply Chain 4.0 performance measurement: A systematic literature review, framework development, and empirical evidence," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 164, p. 102725, Aug. 2022, doi: 10.1016/j.tre.2022.102725.
- [5] N. Zhao, J. Hong, and K. H. Lau, "Impact of supply chain digitalization on supply chain resilience and performance: A multi-mediation model," *Int. J. Prod. Econ.*, vol. 259, p. 108817, May 2023, doi: 10.1016/j.ijpe.2023.108817.
- [6] F. Rasool, M. Greco, and S. Strazzullo, "Understanding the future KPI needs for digital supply chain," *Supply Chain Forum An Int. J.*, vol. 25, no. 4, pp. 392–408, 2024, doi: 10.1080/16258312.2023.2253524.
- [7] V. I. Adebayo, P. O. Paul, and N. L. Eyo-Udo, "The role of data analysis and reporting in modern procurement: Enhancing decision-making and supplier management," *GSC Adv. Res. Rev.*, vol. 20, no. 1, pp. 88–97, 2024, doi: 10.30574/gscarr.2024.20.1.0246.
- [8] P.-L. Poon, M. F. Lau, Y. T. Yu, and S.-F. Tang, "Spreadsheet quality assurance: a literature review," *Front. Comput. Sci.*, vol. 18, no. 2, p. 182203, Jan. 2024, doi: 10.1007/s11704-023-2384-6.
- [9] A. Y. Anggie, J. T. Beng, and Wasino, "Perancangan aplikasi berbasis web untuk pemesanan produk eksterior dan interior pada bengkel las Krisna," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 11, no. 1, pp. 1–7, Jun. 2023, doi: 10.24912/jiksi.v11i1.24087.
- [10] A. Leovin, J. Tji Beng, and E. Dewayani, "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*, IOP Publishing, Dec. 2020, p. 12156. doi: 10.1088/1757-899X/1007/1/012156.
- [11] V. H. Wangi, J. Tji Beng, and Wasino, "Start to end: Recommended travel routes based on tourist preference," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, Jul. 2020, p. 12163. doi: 10.1088/1757-899X/852/1/012163.
- [12] K. Manoj and N. Rainu, "Role of Python in Rapid Web Application Development Using Django," in *Proc. Int. Conf. Innovative Computing and Communication (ICICC 2024)*, Singapore: Springer Nature Singapore, 2024, pp. 551–560, doi: 10.2139/ssrn.4751833.
- [13] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. Indianapolis, IN: Wiley, 2013.
- [14] W. Chen and S. Ahmmed, "Analysis of Python web development applications based on the Django framework," in *Proc. SPIE 13181, Third International Conference on Electronic Information Engineering, Big Data, and Computer Technology (EIBDCT 2024)*, Wuhan, China: SPIE, 2024, Art. no. 131816C, doi: 10.1117/12.3031411.
- [15] W. McKinney, *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter*, 3rd ed. Sebastopol, CA: O'Reilly Media, 2022.
- [16] F. Lusiana, J. Tji Beng, and Wasino, "Grouping of tourism objects using geotagged photo with hierarchical clustering method in Bantul and Sleman," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, Jul. 2020, p. 12166. doi: 10.1088/1757-899X/852/1/012166.
- [17] H. Jing and Y. Fan, "Digital transformation, supply chain integration and supply chain performance: Evidence from Chinese manufacturing listed firms," *SAGE Open*, vol. 14, no. 3, pp. 1–15, Sep. 2024, doi: 10.1177/21582440241281616.
- [18] J. He, M. Fan, and Y. Fan, "Digital transformation and supply chain efficiency improvement: An empirical study from A-share listed companies in China," *PLoS One*, vol. 19, no. 4, p. e0302133, Apr. 2024, doi: 10.1371/journal.pone.0302133.
- [19] K. L. Lee, C. X. Teong, H. M. Alzoubi, M. T. Alshurideh, M. El Khatib, and S. M. Al-Gharaibeh, "Digital supply chain transformation: The role of smart technologies on operational performance in manufacturing industry," *SAGE Open*, vol. 14, no. 1, 2024, doi: 10.1177/18479790241234986.
- [20] N. O. Mandala, I. R. Ayoyi, and S. K. Too, "The impact of information technology adoption

- on efficiency and transparency in public procurement processes in Kenya,” *Eur. Sci. J.*, vol. 20, no. 13, p. 167, 2024, doi: 10.19044/esj.2024.v20n13p167.
- [21] R. C. Martin, *Clean Architecture: A Craftsman’s Guide to Software Structure and Design*. Upper Saddle River, NJ: Prentice Hall, 2018.
- [22] A. Saravanos and M. X. Curinga, “Simulating the Software Development Lifecycle: The Waterfall Model,” *Appl. Syst. Innov.*, vol. 6, no. 6, p. 108, Nov. 2023, doi: 10.3390/asi6060108.
- [23] I. K. Kirpitsas and T. P. Pachidis, “Evolution towards Hybrid Software Development Methods and Information Systems Audit Challenges,” *Software*, vol. 1, no. 3, pp. 316–363, Aug. 2022, doi: 10.3390/software1030015.
- [24] T. D. Capote, “A Comparative Study of Black Box and White Box Testing Techniques in Modern Software Development,” *Front. Eng. Technol.*, vol. 5, no. 1, pp. 1–7, 2023, [Online]. Available: https://iaeme.com/Home/article_id/FET_05_01_001