

INTEGRASI XML DAN PYTHON PADA PENGEMBANGAN FRAMEWORK ODOO SISTEM GUDANG

Anisya ¹⁾, Indra Warman ²⁾, Dede Wira Trise Putra ³⁾, Eva Yulianti ⁴⁾, Hendra Cahyo Setiawan ⁵⁾

^{1,2,3,4,5)} Institut Teknologi Padang

Email : anisya@itp.ac.id ¹⁾, indrawmn@itp.ac.id ²⁾, dedewiratriseputra@itp.ac.id ³⁾, evayulianti@itp.ac.id ⁴⁾, hendrasetyawan1945@gmail.com ⁵⁾

ABSTRAK

Sebuah perusahaan yang bergerak di bidang produksi membutuhkan sebuah sistem yang mampu handle kegiatan operasional, akurasi pencatatan inventaris, manajemen persediaan, serta pemantauan barang digudang. Pemantauan barang digudang ini sangat penting dalam kegiatan operasional perusahaan. Setiap barang yang keluar dan masuk harus dicatat secara komprehensif karena hal ini berkaitan dengan informasi tersedianya stok yang real-time. Salah satu sistem yang mampu melakukan kegiatan tersebut yakni *Enterprise Resource Planning* (ERP) yang dikemas dalam aplikasi Odoo 15. ini merupakan platform *open source* yang populer untuk mengelola berbagai aspek bisnis, termasuk pembelian, penjualan, inventaris, proyek, dan pembiayaan. Pada aplikasi ini perlu penyesuaian dengan kebutuhan operasional perusahaan. Untuk melakukan penyesuaian ini, dilakukan pengembangan aplikasi Odoo 15 dengan integrasi kompleks antara XML, Python, dan PostgreSQL. Proses integrasi ini dilakukan untuk mengembangkan *interface system* dan penyesuaian dengan proses transaksi produksi khususnya pengelolaan data gudang (*warehouse*) pada aplikasi Odoo 15. Hasil dari penelitian ini berupa sistem ERP berbasis Odoo 15 dengan mengembangkan modul gudang yang dapat terintegrasi dengan modul lainnya, untuk mendukung pengelolaan stok dan memberikan fleksibilitas tinggi dengan fitur penyesuaian yang lengkap, termasuk pengaturan lokasi, jenis, jumlah, dan kondisi barang. serta menghasilkan laporan stok yang komprehensif, memberikan informasi detail tentang ketersediaan stok.

Kata Kunci : XML, Python, PostgreSQL, Odoo, ERP

ABSTRACT

A company operating in the production sector needs a system that is capable of handling operational activities, accurate inventory recording, inventory management, and monitoring goods in the warehouse. Supervision of goods in the warehouse is very important in the company's operational activities. Every incoming and outgoing item must be recorded comprehensively because this is related to real-time stock availability information. One system that is capable of carrying out these activities is *Enterprise Resource Planning* (ERP) which is packaged in the Odoo 15 application. This is a popular open source platform for managing various aspects of business, including purchasing, sales, inventory, projects and financing. This implementation needs to be adjusted to the company's operational needs. To carry out this customization, an Odoo 15 application was developed with complex integration between XML, Python, and PostgreSQL. This integration process was carried out to develop the system interface and adapt it to the production transaction process, especially warehouse data management in the Odoo 15 application. The result of this research is an Odoo 15 based ERP system by developing a warehouse module that can be integrated with other modules, to support stock management and provide high flexibility with complete customization features, including setting location, type, quantity and condition of goods. And also generate comprehensive stock reports, providing detailed information on availability of stock,

.Keywords: XML, Python, PostgreSQL, Odoo, ERP

1. PENDAHULUAN

Rekayasa perangkat lunak Odoo sebagai bagian dari sistem Enterprise Resource Planning (ERP) adalah proses pengembangan sistem informasi manajemen bisnis (SIMB) yang menggunakan platform sumber terbuka Odoo (Katuu, 2020). Sistem ERP dapat membantu perusahaan meningkatkan efisiensi dan efektivitas bisnis mereka melalui integrasi dan otomatisasi proses bisnis (Mustafri et al., 2016). Odoo memiliki lebih dari 4 juta pengguna aktif dan memfasilitasi kolaborasi antar departemen perusahaan. Implementasi sistem ERP seperti Odoo dapat membantu perusahaan memperoleh keunggulan kompetitif dan memastikan kesuksesan bisnis jangka Panjang (Katuu, 2020).

Perusahaan yang jadi studi kasus di penelitian ini merupakan UD Agung, perusahaan pengolahan batu koral di Tulungagung, Jawa Timur yang menghadapi beberapa masalah operasional. Salah satunya adalah kesulitan dalam pencatatan inventaris/stok batu koral yang tidak selaras antara gudang dan buku manual. Perusahaan juga kekurangan sistem manajemen gudang yang efektif. Untuk mengatasi masalah ini, mereka memutuskan untuk mengimplementasikan sistem Enterprise Resource Planning (ERP) dengan modul gudang Odoo yang memerlukan beberapa modifikasi. Ini termasuk penyesuaian fitur dan alur kerja sesuai dengan proses bisnis perusahaan serta integrasi kompleks antara XML dan Python. Dengan mengatasi tantangan ini, UD Agung berharap meningkatkan efisiensi operasional, akurasi pencatatan inventaris, manajemen persediaan, dan pemantauan gudang. Implementasi sistem ERP yang terintegrasi diharapkan dapat membantu mereka mengoptimalkan pengelolaan bisnis mereka dalam industri pengolahan batu koral.

Odoo adalah platform pengembangan aplikasi terbuka yang populer untuk mengelola berbagai aspek bisnis, termasuk pembelian, penjualan, inventaris, proyek, dan pembiayaan (Mustafri et al., 2016). Odoo Framework memiliki modul-modul terintegrasi yang dapat

disesuaikan sesuai kebutuhan perusahaan, dengan antarmuka yang user-friendly dan dukungan aktif dari komunitas pengembang (Mardhiah et al., 2022). Penggunaan Python dan XML dalam pengembangan Odoo memberikan fleksibilitas dan kemudahan dalam mengadaptasi sistem ini sesuai dengan kebutuhan bisnis, serta meminimalkan biaya dan waktu implementasi ERP (Widodo and Sarah, 2022). Ini membuat Odoo menjadi pilihan yang menarik untuk banyak perusahaan.

Framework Odoo ERP juga merupakan sistem alur kerja bisnis sumber terbuka yang populer dan dibangun menggunakan bahasa pemrograman Python dan basis data PostgreSQL (Shereef, 2019). Dalam struktur Odoo, databasenya menggunakan PSQL, pengembangan perangkat dilakukan dengan bahasa Python, dan tampilan dibuat melalui berkas XML. Template HTML digunakan untuk membentuk tampilan dalam bagian situs web (Shereef, 2019). Perbedaan utama antara pengembangan modul gudang dan sistem gudang konvensional adalah bahwa modul gudang memastikan efisiensi dalam proses gudang, mengoptimalkan penggunaan waktu dan sumber daya, serta meningkatkan akurasi data. Sebagai bagian dari sistem ERP, modul gudang dapat terintegrasi dengan modul lainnya, mendukung pertumbuhan bisnis, dan memberikan fleksibilitas tinggi dengan fitur penyesuaian yang lengkap, termasuk pengaturan lokasi, jenis, jumlah, dan kondisi barang. Selain itu, sistem ini memantau dan mengontrol stok, serta menghasilkan laporan stok yang komprehensif, memberikan informasi detail tentang ketersediaan, pemesanan, penerimaan, dan lainnya.

Penelitian ini menggunakan metodologi *Waterfall* untuk mendukung mengintegrasikan teknologi XML, Python dan Postgresql pada modul gudang berbasis Odoo. Tujuannya adalah mengembangkan aplikasi odoo untuk bisa menampilkan halaman stok untuk usaha UD. Agung.

2. METODE

Metodologi *Waterfall* mengikuti pendekatan linear yang terstruktur, dengan

tahapan yang jelas dari analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan (Pricillia and Zulfachmi, 2021). Tahapan yang dilakukan berupa :

2.1. Tahap Perencanaan

Tahap perencanaan menyangkut studi tentang kebutuhan pengguna, kelayakan baik secara teknik maupun secara teknologi. Model *waterfall* dilakukan perencanaan tentang sistem yang akan dibangun.

2.2 Tahap analisis

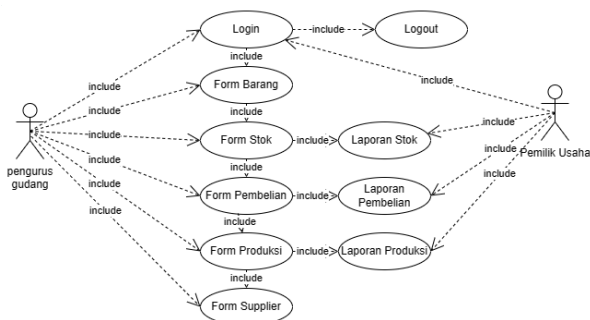
Merupakan proses pendalaman dengan masalah yang terjadi di UD Agung. Disini penulis menemukan bahwasanya sering terjadi kekeliruan dalam penghitungan sisa stok yang ada digudang karena pencatatan data Gudang yang masih manual sehingga ada beberapa berkas yang mengalami kehilangan.

2.3 Tahap perancangan

Menyangkut perancangan sistem dimana akan memberikan rencana solusi dari masalah yang muncul pada tahap analisis.

a. Use Case Diagram

Proses perancangan sistem menggunakan *Unified Modeling Language* (UML), yang merupakan bahasa pemodelan visual yang digunakan untuk merancang proses bisnis dan sejenisnya, serta untuk menganalisis, mendesain, dan mengimplementasikan sistem berbasis perangkat lunak (Atieh et al., 2016). *Diagram Use Case* digunakan untuk menggambarkan bagaimana sistem memenuhi kebutuhan pengguna. Di dalamnya terdapat fungsi-fungsi yang diberikan oleh sistem yang akan dikembangkan dan pengguna yang akan menggunakan sistem (Kuswoyo, 2023).

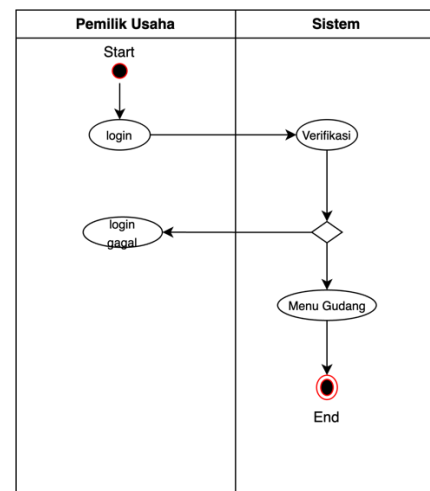


Gambar 1. Use Case Diagram

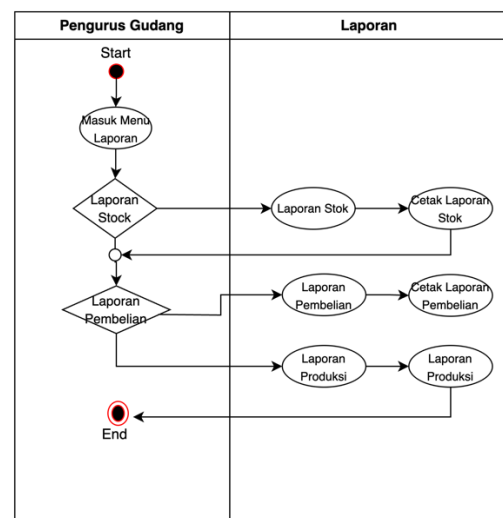
Pada gambar 1, terdapat 2 level user yakni Pengurus Gudang dan Pemilik Usaha. Level user yang mampu mengelola data di sistem ini yakni Pengurus Gudang, sedangkan Pemilik Usaha hanya mampu melihat informasi dari data yang ada di sistem.

b. Activity Diagram

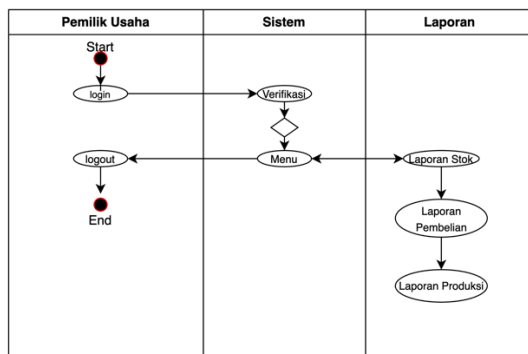
Dari *use case* tersebut, akan dibuat activity diagram yang menunjukkan langkah-langkah untuk mencapai tujuan dari setiap case pada use case (Purwanto et al., 2021).



Gambar 2. Activity diagram Login Pengurus Gudang



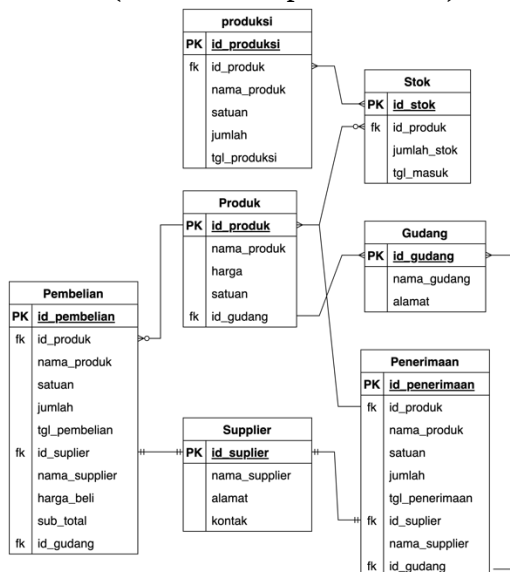
Gambar 3 Activity diagram Menu Laporan Pengurus Gudang



Gambar 4. Activity diagram Menu Pemilik Usaha

c. Class Diagram

Suatu teknik pemodelan yang digunakan untuk merepresentasikan objek-objek dalam suatu sistem dan hubungan antara objek-objek tersebut (Veza and Ropianto, 2017).



Gambar 5. Class Diagram

Pada gambar 5, penulis menampilkan *class diagram* untuk semua kebutuhan sistem, namun untuk pembahasan nanti, penulis hanya membahas untuk halaman stok.

2.4 Tahap implementasi, adalah tahapan dimana sistem diimplementasikan pada situasi nyata dengan pemilihan perangkat keras dan penyusunan desain (coding) (Mustafri et al., 2016). Untuk implementasi yaitu dengan memberitahu user, melatih user, dan menjelaskan cara menggunakan sistem

2.5 Tahap pengujian, tahap untuk menguji sistem sudah berjalan sesuai rencana yang sudah disepakati sebelumnya, termasuk pengujian masing-masing menu apa masih ada error atau tidak. Tujuan dari pengujian ini adalah untuk meminimalisir cacat desain web sehingga sistem yang dikembangkan benar-benar dapat berjalan dengan sebaik mungkin (Chandra and Poerbo, 2019).

3. HASIL DAN PEMBAHASAN

Dari beberapa tahapan yang telah dilakukan diatas, dilanjutkan dengan proses pembangunan sistem, dimana hasil yang telah diperoleh berupa sistem informasi pengelolaan data Gudang berbasis odoo.

3.1 Konfigurasi Pada Framework Odoo 15

Terdapat beberapa langkah yang perlu dilakukan untuk mempersiapkan dan mengatur sistem sesuai dengan kebutuhan. Konfigurasi ini melibatkan pengaturan modul, pengguna, tampilan antarmuka, aturan bisnis, dan lain-lain (Kerbache, 2002). Dengan melakukan konfigurasi yang tepat, pengguna dapat mengoptimalkan fungsionalitas dan kinerja sistem Odoo 15 sesuai dengan kebutuhan bisnis mereka.

- Install Python 3 dan Dependensinya
- Instalasi dependensi web
- Konfigurasi Folder Models
- Konfigurasi Folder Views
- Konfigurasi Folder Security

3.2 Koneksi

Untuk bisa sistem berjalan sebelumnya dibuatkan script untuk penghubung *database* PostgreSQL dengan aplikasi odoo.

```
./odoo-bin--addons-
path=/home/hendra/Dokumen/odoo-
15.0/addons,/home/hendra/Dokumen/odoo-
15.0/myaddos,/home/hendra/Dokumen/gudang/m
yaddonsku -d mygudang --
db_port 5432 --xmlrpc-port=8000 --limit-
memory-hard 0 --limit-time-real=10000 -u
ud_agung
```

Dari script diatas, disebutkan nama database mygudang.

Column	Type	Collation	Nullable	Default	Storage	Compression	Stats target	Description
id	integer		not null	nextval('u_stok_id_seq'::regclass)	plain			
name	character varying		not null		extended			Nama Barang
kode	character varying		not null		extended			Kode Barang
stokbahan	integer		not null		plain			Stok Bahan
bahanbertambah	integer				plain			Bahan Bertambah
bahanberkurang	integer				plain			Bahan Berkurang
stokbarang	integer		not null		plain			Stok Barang
barangbertambah	integer				plain			Barang Bertambah
barangberkurang	integer				plain			Barang Berkurang
satuan	character varying				extended			Satuan
create_uid	integer				plain			Created by
create_date	timestamp without time zone				plain			Created on
write_uid	integer				plain			Last Updated by
write_date	timestamp without time zone				plain			Last Updated on
hargabeli	integer		not null		plain			Harga Beli
hargajual	integer		not null		plain			Harga Jual

Indexes:
"u_stok_pkey" PRIMARY KEY, btree (id)

Foreign-key constraints:
"u_stok_create_uid_fkey" FOREIGN KEY (create_uid) REFERENCES res_users(id) ON DELETE SET NULL
"u_stok_write_uid_fkey" FOREIGN KEY (write_uid) REFERENCES res_users(id) ON DELETE SET NULL

Referenced by:
TABLE "u_detailproduksi" CONSTRAINT "u_detailproduksi_barang_id_fkey" FOREIGN KEY (barang_id) REFERENCES u_stok(id) ON DELETE RESTRICT
TABLE "u_penjualan_barang_rel" CONSTRAINT "u_penjualan_barang_rel_barang_id_fkey" FOREIGN KEY (barang_id) REFERENCES u_stok(id) ON DELETE CASCADE
TABLE "u_penjualan_detail" CONSTRAINT "u_penjualan_detail_barang_fkey" FOREIGN KEY (barang) REFERENCES u_stok(id) ON DELETE RESTRICT
TABLE "u_purchase_barang_rel" CONSTRAINT "u_purchase_barang_rel_barang_id_fkey" FOREIGN KEY (barang_id) REFERENCES u_stok(id) ON DELETE CASCADE
TABLE "u_purchase_detail" CONSTRAINT "u_purchase_detail_barang_fkey" FOREIGN KEY (barang) REFERENCES u_stok(id) ON DELETE RESTRICT

Access method: heap

Gambar 6. Tabel u_stok Dalam Postgresql

Tabel di atas, yang dibentuk dari model "u.stok" dalam dalam berkas model/stok.py. Tabel ini berisi informasi penting tentang barang dan transaksi terkait. Setiap entri dalam tabel memiliki identifikasi unik yang disimpan dalam kolom "id". Nama dan kode barang disimpan dalam kolom "name" dan "kode" secara berturut-turut, sementara stok bahan mentah dan barang jadi

diwakili oleh kolom-kolom "stokbahan" dan "stokbarang", masing-masing dengan perubahan melalui kolom "bahanbertambah", "bahanberkurang", "barangbertambah", dan "barangberkurang". Kolom "satuan" berisi informasi tentang ukuran barang. Dibawah ini, halaman form (tampilan pengisian) untuk model "stok".

Gambar 7. Form Stok

Script interface supplier terdapat pada gudang/ud_agung/views/stok.xml

```
<record id="view_stok_form" model="ir.ui.view">
  <field name="name">u.stok.form</field>
  <field name="model">u.stok</field>
  <field name="arch" type="xml">
    <form string="stok">
      <sheet>
        <group name="barang" string="Data Barang">
          <field name="kode"/>
```

```
</field name="name"/>
</field name="satuan"/>
</group>
</group>
        <group name="stok bahan" string="Stok Bahan">
          <field name="stokbahan"/>
          <field name="hargabeli"/>
        </group>
        <group name="stok barang" string="Stok Barang">
          <field name="stokbarang"/>
          <field name="hargajual"/>
        </group>
```

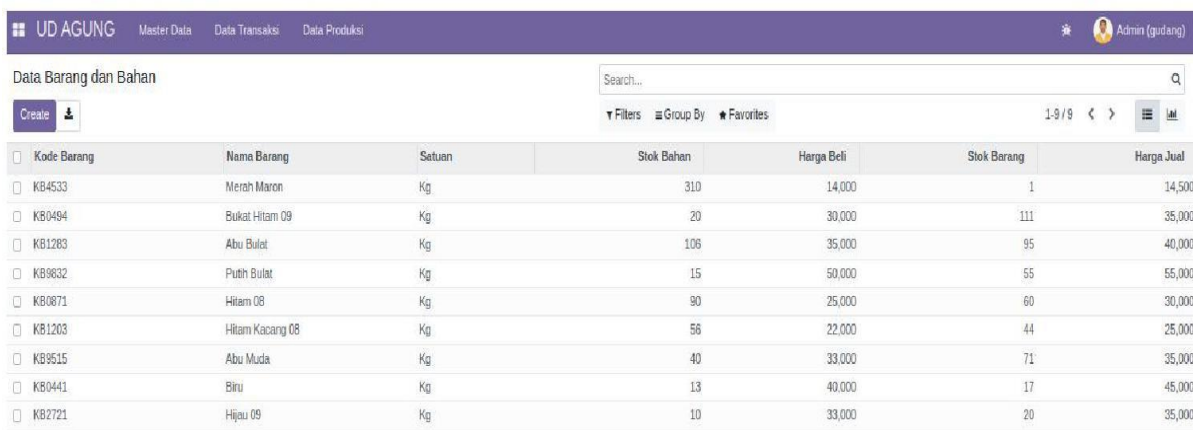
```
</group>
</sheet>
</form>
</field>
</record>
```

Pada XML di atas, terdapat script `view_stok_form` yang menghubungkan model `u.stok` dengan tampilan `u.stok.form` menggunakan model `ir.ui.view` yang berada dalam `addons/website/models/ir_ui_view.py`. Dalam Framework odoo terdapat `<field name="model">u.stok</field>` fitur ini disediakan oleh odoo untuk memudahkan dalam membuat fitur CRUD dibandingkan dengan native sehingga memudahkan dalam mendevelop. Model ini berfungsi untuk mendefinisikan tampilan data dalam format XML, termasuk tata letak, elemen yang ditampilkan, dan atribut lainnya. Ini memungkinkan pengguna untuk membuat, mengedit, dan menghapus data. Model `u.stok` melalui `<field name= model >u.stok</field>`. Di tampilan XML, terdapat tampilan formulir (form view) yang

menampilkan informasi objek `u.stok`. Di dalamnya, terdapat beberapa grup yang berisi field-field terkait dengan model `u.stok`. Di dalam Data Barang, field kode, name, dan satuan menampilkan info tentang barang. Grup Stok Bahan menampilkan field terkait stok bahan, termasuk **stokbahan**, **bahanbertambah**, **bahanberkurang**, dan **hargabeli**. Grup Stok Barang menampilkan field terkait stok barang, termasuk **stokbarang**, **barangbertambah**, **barangberkurang**, dan **hargajual**. Dengan terhubungan model `u.stok` dan tampilan `u.stok.form`, saat membuka tampilan form untuk objek `u.stok`, yang nantinya ketika diinputkan maka akan menyimpan data dalam *database* serta menampilkan informasi dari field-field ini akan ditampilkan sesuai data dalam model.

4.6.2.3 Halaman Interface View Stok

Pada halaman view (tampilan daftar) dibawah ini untuk model "Stok" dalam Odoo.



Kode Barang	Nama Barang	Satuan	Stok Bahan	Harga Beli	Stok Barang	Harga Jual
KB4533	Merah Maron	Kg	310	14,000	1	14,500
KB0494	Bulat Hitam 09	Kg	20	30,000	111	35,000
KB1283	Abu Bulat	Kg	106	35,000	95	40,000
KB9832	Putih Bulat	Kg	15	50,000	55	55,000
KB0871	Hitam 08	Kg	90	25,000	60	30,000
KB1203	Hitam Kacang 08	Kg	56	22,000	44	25,000
KB9515	Abu Muda	Kg	40	33,000	71	35,000
KB0441	Biru	Kg	13	40,000	17	45,000
KB2721	Hijau 09	Kg	10	33,000	20	35,000

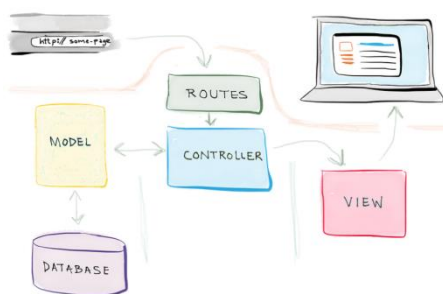
Gambar 8. View Stok

Script interface supplier terdapat pada `gudang/ud_agung/views/stok.xml`

```
<record id="view_stok_tree" model="ir.ui.view">
<field name="name">u.stok.tree</field>
<field name="model">u.stok</field>
<field name="arch" type="xml">
<tree string="stok">
<field name="kode"/>
<field name="name"/>
<field name="satuan"/>
```

```
<field name="stokbahan"/>
<field name="bahanbertambah"/>
<field name="bahanberkurang"/>
<field name="hargabeli"/>
<field name="stokbarang"/>
<field name="barangbertambah"/>
<field name="barangberkurang"/>
<field name="hargajual"/>
</tree>
</field>
</record>
```

Pada XML di atas, terdapat record dengan ID "view_stok_tree" yang menghubungkan model u.stok dengan view u.stok.tree. Record ini menggunakan model ir.ui.view. Di dalam *record* tersebut, menentukan bahwa view ini terkait dengan model u.stok dengan menggunakan `<field name="model"> u.stok </field>`. Pada tampilan XML, terdapat tree view yang menampilkan tampilan tabel untuk u.stok. Di dalam tree view tersebut, terdapat beberapa *field* yang ditampilkan, yaitu: kode, name, satuan, stokbahan, bahanbertambah, bahanberkurang, hargabeli, stokbarang, barangbertambah, barangberkurang, dan hargajual. Dengan keterkaitan antara model u.stok dan view u.stok.tree, membuka tampilan tree untuk objek u.stok, informasi dari field-field tersebut akan ditampilkan sesuai dengan data yang tersimpan dalam model. Hal ini memungkinkan untuk melihat dan melakukan pencarian pada informasi stok bahan dan stok barang melalui tampilan tree yang telah didefinisikan dalam file XML.



Gambar 9. MVC dalam Framework Odoo ERP (Shereef, 2019)

Bagian-bagian MVC dalam framework Odoo ERP:

1. *Model*: Merujuk pada tabel-tabel yang dibuat dalam basis data PostgreSQL. Model ini mencerminkan struktur data yang digunakan oleh sistem untuk menyimpan informasi seperti entitas bisnis, riwayat transaksi, dan lain-lain.
2. *View*: Meliputi berkas XML dan template HTML yang mendefinisikan tampilan antarmuka pengguna. View ini menangani bagaimana data dari model ditampilkan kepada pengguna dalam bentuk yang informatif dan estetik.

3. *Controller*: Terdiri dari kelas-kelas Python dan fungsi-fungsi yang mengatur alur logika bisnis. *Controller* menghubungkan antara model dan view, mengendalikan interaksi pengguna, dan memproses permintaan pengguna.

Ketika pengguna berinteraksi dengan antarmuka untuk menjalankan operasi tertentu, *Controller* yang sesuai akan diaktifkan. Pada titik ini, fungsi Python yang relevan akan dieksekusi. Fungsi ini dapat berkomunikasi dengan model-model data (yang mewakili tabel-tabel dalam basis data PostgreSQL) untuk mengambil data yang diperlukan atau langsung mengirimkan data ke tampilan. Tampilan akan menampilkan data dalam bentuk yang dapat dimengerti oleh manusia, dengan gaya yang diterapkan melalui berkas XML atau HTML. Dengan pendekatan ini, framework Odoo berhasil mengadopsi dan menerapkan konsep *Model-View-Controller* (MVC) dalam arsitekturnya. *Controller* bertugas mengambil data dari model sesuai permintaan dan meneruskannya ke tampilan. Tampilan ini akan menampilkan data yang telah diambil, menghadirkannya kepada pengguna melalui formulir-formulir yang sesuai.

4. PENUTUP

4.1. Kesimpulan

Dengan mengintegrasikan teknologi XML, Python, dan PostgreSQL dalam framework Odoo 15 serta menggunakan metode *waterfall* untuk pengembangan, dapat membantu dalam penyelarasan proses bisnis pada UD Agung, dengan pembuatan model menggunakan Python untuk berinteraksi dengan tabel database PostgreSQL serta proses CRUD (*Create, Read Update, and Delete*) dari database PostgreSQL, untuk menampilkan data menggunakan XML

4.2. Saran

Untuk pengembangan teknologi XML, Python, dan PostgreSQL yang lebih lanjut, berikut adalah beberapa saran yang dapat dipertimbangkan:

1. Implementasi *Database Transaksional Multi-User*, untuk menjaga integritas data

dan mendukung akses simultan oleh banyak pengguna. akurat,serta agar digunakan oleh banyak pengguna dalam transaksi yang berbeda-beda.

2. Perlu pengembangan antarmuka terkait penggunaan teknologi XML

5. DAFTAR PUSTAKA

- Atieh, A.M., Kaylani, H., Al-abdallat, Y., Qaderi, A., Ghoul, L., Jaradat, L., Hdairis, I., 2016. Performance Improvement of Inventory Management System Processes by an Automated Warehouse Management System. *Procedia CIRP* 41, 568–572. <https://doi.org/10.1016/j.procir.2015.12.122>
- Chandra, B., Poerbo, H., 2019. SISTEM INFORMASI AKUNTANSI BERBASIS WEB PADA PERUSAHAAN JASA TENAGA KERJA 2.
- Katuu, S., 2020. Enterprise Resource Planning: Past, Present, and Future. *New Review of Information Networking* 25, 37–46. <https://doi.org/10.1080/13614576.2020.1742770>
- Kerbache, L., 2002. Enterprise resource planning (ERP): The dynamics of operations management. *Interfaces* 32, 104–105.
- Kuswoyo, D., 2023. Analisis Perancangan Sistem Pada Sistem Informasi Penjualan PT. Bangsawan Cyberindo. *JOMMIT* 7, 036–041. <https://doi.org/10.46961/jommit.v7i1.753>
- Mardhiah, *Nurul, Rahayu, G., Oktavia, D., 2022. Design and Implementation of Odoo ERP Application as a Solution to Re-quirement of ULVA-Q MSME Business Information Systems Pangandaran. *REJHH* 5, 404–410. <https://doi.org/10.24815/jr.v5i2.27705>
- Mustafri, A., Witarsyah, D., Witjaksono, W., 2016. PENGEMBANGAN ODOO MODUL WAREHOUSE PADA GUDANG PT. TARUMATEX MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT.
- Pricillia, T., Zulfachmi, 2021. Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD). *bangkitindonesia* 10, 6–12. <https://doi.org/10.52771/bangkitindonesia.v10i1.153>
- Purwanto, H., Hananto, A.Z., Maulana, F., Pratama, G., 2021. PENERAPAN ENTERPRISE RESOURCE PLANNING (ERP) MODUL SALES UNTUK PENINGKATAN PENJUALAN LITTLE INK'S BANDUNG 7.
- Shereef, N., 2019. What is Open Source Odoo ERP? *New Review of Information Networking* 25, 37–46.
- Veza, O., Ropianto, M., 2017. Perancangan Sistem Informasi Inventory Data Barang Pada PT. Andalas Berlian Motors 2.
- Widodo, S., Sarah, S., 2022. PLANNING AND IMPLEMENTATION OF ODOO ERP HUMAN RESOURCE APPLICATION MODULE USING ACCELERATED SAP (ASAP) METHOD IN HEAVY EQUIPMENT RENTAL COMPANY 1.