

# IMPLEMENTASI PYTHON FRAMEWORK FLASK PADA MODUL TRANSFER OUT TOKO DI PT XYZ

David Jonathan Evan\*<sup>1)</sup>, Pratyaksa Ocsa Nugraha Saian<sup>2)</sup>

1. Universitas Kristen Satya Wacana, Indonesia
2. Universitas Kristen Satya Wacana, Indonesia

## Article Info

**Kata Kunci:** *Distribution Center; Flask; Python; Retail;*

**Keywords:** *Distribution Center; Flask; Python; Retail;*

## Article history:

Received 2 May 2023

Revised 16 May 2023

Accepted 30 May 2023

Available online 1 December 2023

## DOI :

<https://doi.org/10.29100/jipi.v8i4.4020>

\* Corresponding author.

David Jonathan Evan

E-mail address:

[672019012@student.uksw.edu](mailto:672019012@student.uksw.edu)

## ABSTRAK

Pengembangan teknologi informasi memberikan kemudahan dalam pekerjaan manusia, namun mengakibatkan teknologi lama semakin tidak relevan. Perusahaan-perusahaan berlomba untuk mengadopsi teknologi terbaru untuk meningkatkan efisiensi dan daya saing mereka. PT. XYZ, perusahaan retail di Indonesia yang memiliki lebih dari 16.492 gerai menghadapi masalah dengan teknologi lama yang mereka gunakan, yaitu Oracle Form. Oracle Form memiliki kekurangan pada biaya pemeliharaan aplikasi yang tinggi dan hanya dapat beroperasi pada satu platform saja. Karena alasan tersebut, PT. XYZ memutuskan untuk membangun modul Transfer Out Toko menggunakan Python dan framework Flask. Flask adalah micro web framework yang dirancang untuk mempercepat pengembangan aplikasi web menggunakan bahasa Python. Pengembangan modul Transfer Out Toko dilakukan menggunakan metode Waterfall. Tahapan penelitian mencakup analisis kebutuhan, perancangan, implementasi, dan pengujian oleh Quality Assurance (QA) untuk memastikan aplikasi sesuai dengan desain yang ditentukan. Setelah melalui proses pengembangan, dilakukan pengujian pada Modul Transfer Out Toko menggunakan metode unit testing untuk memastikan setiap unit program berfungsi dengan baik. Pengujian dilanjutkan dengan menyerahkan modul kepada tim Quality Assurance (QA) PT.XYZ untuk pengujian lebih lanjut. Hasil dari pengembangan aplikasi ini menunjukkan bahwa menggunakan teknologi terbaru seperti Flask dan Python dapat menghasilkan aplikasi yang lebih efisien dan dapat dioperasikan di berbagai platform.

## ABSTRACT

The development of information technology provides convenience in human work, but it also renders outdated technologies increasingly irrelevant. Companies race to adopt the latest technology to improve efficiency and competitiveness. PT. XYZ, a retail company in Indonesia that has more than 16,492 stores, faced problems with their old technology, Oracle Form. Oracle Form has disadvantages of high application maintenance costs and only being able to operate on one platform. For these reasons, PT. XYZ decided to build the Transfer Out Toko module using Python and Flask framework. Flask is a micro web framework designed to accelerate web application development using the Python language. The development of the Transfer Out Toko module was carried out using the Waterfall method. The research stages included requirements analysis, design, implementation, and Quality Assurance (QA) testing to ensure the application complied with the specified design. After going through the development process, testing was performed on the Transfer Out Toko module using unit testing to ensure that each program unit functioned properly. The testing continued by handing over the module to the PT.XYZ Quality Assurance (QA) team for further testing. The results of this application development indicate that using the latest technology like Flask and Python can produce more efficient applications that can be operated on various platforms.

## I. PENDAHULUAN

Perkembangan pesat teknologi informasi telah memudahkan pekerjaan manusia. Setiap tahun, muncul teknologi baru yang menawarkan fitur dan fungsi yang lebih kompleks dan canggih daripada teknologi serupa yang sudah ada. Teknologi lama menjadi semakin tidak relevan dan kehilangan dukungan dari perusahaan pengembang [1]. Salah satunya yaitu Oracle Form, yang juga merupakan salah satu teknologi yang digunakan oleh PT. XYZ.

PT XYZ merupakan salah satu perusahaan retail di Indonesia yang memiliki lebih dari 16.492 gerai dengan 32 gudang yang tersebar di seluruh wilayah Indonesia [2]. Untuk mengakomodasi kebutuhan barang dari setiap gerai, PT XYZ memerlukan beragam sistem serta aplikasi yang dapat mendukung dan mempermudah proses administrasi pencatatan barang keluar dari *Distribution Center* ke Toko. Oleh karena itu, PT XYZ membangun modul *Transfer Out* Toko menggunakan Oracle Form, yang merupakan bagian dari aplikasi *Distribution Center System*.

Keberadaan modul *Transfer Out* Toko memudahkan pengguna untuk melakukan *data entry* yang dibantu dengan beberapa *List Of Values* (LOV) untuk mengurangi kemungkinan *human error* dan mempermudah proses *data entry* [3]. Setelah selesai, pengguna dapat menyimpan data yang ada pada *form* untuk pencatatan dan mendapat faktur dari transaksi tersebut. Faktur ini akan digunakan untuk kepentingan proses bisnis PT XYZ selanjutnya.

Seiring berjalannya waktu, didapati bahwa aplikasi tersebut memiliki kekurangan seperti biaya pemeliharaan aplikasi yang tinggi, teknologi yang semakin terbelakang, dan hanya dapat beroperasi pada satu platform saja. Jika dibandingkan, Python dan *framework* Flask bersifat *open-source* sehingga bebas dari biaya lisensi. Selain itu, Python dan *framework* Flask dipilih juga karena ringan, cepat dan mudah untuk diimplementasikan. Adapun hasil dari aplikasi yang dibuat dengan Flask adalah aplikasi berbasis web sehingga bisa beroperasi di berbagai platform [4].

Python adalah bahasa pemrograman level tinggi, *interpreted*, berorientasi objek yang penulisan kodenya dinamis dan mudah dipahami [5]. Python dapat digunakan untuk berbagai jenis pengembangan perangkat lunak. Python hadir dengan *library* bawaan yang dapat diperluas sehingga menjadi modular dan *reuseable* [6].

Flask adalah *micro web framework* yang menggunakan bahasa Python [7]. Penggunaan *framework* ini dimaksudkan untuk mempercepat pengembangan aplikasi karena Flask sudah menyediakan *library* dan fungsionalitas dasar untuk membangun aplikasi web tanpa harus membangun kembali dari awal [8]. Selain itu, dengan mengimplementasikan Flask maka akan mengurangi penggunaan sumber daya memori karena Flask tergolong dalam *micro framework* [9]. *Framework* Flask juga sudah memiliki dokumentasi yang lengkap sehingga mempermudah *programmer* dalam memelihara aplikasi dan mencari referensi [10]. Dalam pengembangan aplikasi ini, Bootstrap dan JQuery juga digunakan sebagai UI *library* pada *frontend* aplikasi dan PostgreSQL sebagai *database*.

Pada penelitian yang berjudul “*Designing and Implementing a Database for Thesis Data Management by Using the Python Flask Framework*” oleh Suraya, dilakukan perancangan sistem pengolahan *database* yang memiliki empat tabel yaitu *lecturer table*, *student table*, *title of thesis table* dan *guidance table*. Sistem yang dibuat menghasilkan dua aplikasi yaitu aplikasi *backend* yang dibuat menggunakan *framework* Flask dan aplikasi *frontend* berbasis web. Aplikasi yang dibuat mampu melakukan proses pengolahan *database* seperti *delete*, *update*, dan *searching*. Dari penelitian Suraya, didapat kesimpulan bahwa *framework* Flask bisa menjadi solusi untuk pemrograman berbasis web menggunakan Python [11]. Penelitian yang dilakukan oleh Suraya fokus pada perancangan sistem pengolahan *database* untuk manajemen data skripsi menggunakan *framework* Flask, sementara penelitian yang akan dilakukan lebih fokus pada pengembangan aplikasi *Distribution Center System* menggunakan Python dan *framework* Flask. Dengan demikian, penelitian yang akan dilakukan memberikan kontribusi pada pengembangan teknologi di industri yang berbeda. Yaitu pada bidang retail, khususnya aplikasi *Distribution Center System*.

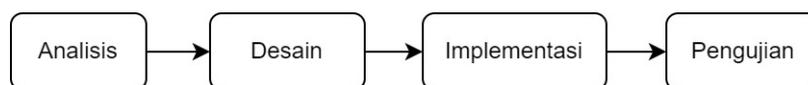
Pada penelitian yang dilakukan oleh Dinda yang berjudul “*Flask Framework Implementation in Development Purchasing Approval Request Application*” dilakukan pengembangan aplikasi *Purchasing Approval Request* (PAR) dengan menerapkan Flask. Pada penelitian ini, Flask telah memudahkan pengembangan aplikasi yang sekaligus memenuhi kebutuhan bisnis. Penggunaan *decorator* dan *library* bawaan menjadi faktor utama keberhasilan pengembangan aplikasi, seperti Jinja Template yang memudahkan komunikasi antara *backend* dan *frontend* [12]. Meskipun penelitian ini dan penelitian yang dilakukan oleh Dinda menggunakan Flask sebagai *framework* pengembangan aplikasi, penelitian ini fokus pada implementasi Flask pada aplikasi *Distribution Center System* PT. XYZ, sedangkan penelitian Dinda berfokus pada pengembangan aplikasi *Purchasing Approval Request* (PAR). Selain itu, perbedaan kedua penelitian ini terletak pada penggunaan teknologi lama yang dianggap semakin tidak relevan dan mahal dalam pemeliharannya, serta hanya dapat beroperasi pada satu platform saja. Penelitian Dinda tidak menyinggung masalah penggunaan teknologi lama, namun lebih fokus pada pengembangan aplikasi PAR menggunakan Flask sebagai solusi teknologi yang cocok untuk kebutuhan bisnis.

Penelitian lain mengenai pembangunan aplikasi pada *Distribution Center* sudah pernah dilakukan sebelumnya, seperti penelitian yang dilakukan oleh Anik dan Sigit yang berjudul “**Penerapan Knowledge Management Sistem Layanan Perbaikan Device Distribution Center (DC) Support PT. Indomarco Prismatama Cabang Bekasi**”. Pada penelitian ini, dibangun sistem informasi *Knowledge Management System DC Support* yang dibangun dengan PHP, *framework* Laravel, dan PostgreSQL. Tujuan sistem informasi ini dibangun adalah untuk menyimpan dan memanfaatkan *knowledge* yang dimiliki *DC Support* sehingga mempermudah untuk menemukan solusi dari masalah yang dihadapi [13]. Penelitian ini mempunyai perbedaan dengan penelitian Anik dan Sigit dalam hal fokus penelitian dan teknologi yang digunakan. Anik dan Sigit membangun sistem informasi *Knowledge Management System DC Support* dengan menggunakan PHP, *framework* Laravel, dan PostgreSQL yang bertujuan untuk menyimpan dan memanfaatkan *knowledge* dari *DC Support*. Sebaliknya, penelitian ini fokus pada implementasi *framework* Flask pada aplikasi *Distribution Center System* PT. XYZ yang sebelumnya menggunakan teknologi lama yang semakin tidak relevan dan mahal dalam pemeliharannya.

Berdasarkan penelitian-penelitian yang dilakukan sebelumnya, didapat kesimpulan bahwa Python dan *framework* Flask dapat membantu pengembangan aplikasi berbasis web. Sama dengan penelitian sebelumnya, penelitian kali ini akan mengimplementasikan Python dan *framework* Flask dalam proses pengembangan aplikasi. Adapun perbedaan utama antara penelitian terdahulu dengan penelitian kali ini yaitu belum ada penelitian yang membahas modul *Transfer Out* Toko pada *Distribution Center System*. Maka dari itu, akan dilakukan penelitian terkait implementasi Python dan *framework* Flask pada modul *Transfer Out* Toko yang merupakan bagian dari aplikasi *Distribution Center System*.

## II. METODE PENELITIAN

Penelitian ini menggunakan metode *Waterfall* menurut Pressman [14]. Adapun tahapan penelitian yang akan dilakukan sebagai berikut:



Gambar. 1. Metode Penelitian

Mula-mula, dilakukan analisis terhadap kebutuhan sistem yang akan digunakan dalam penelitian. Urgensi dan kebutuhan dari sistem yang akan dibuat didapat melalui wawancara kepada salah satu *Senior Programmer* di PT XYZ. Dari wawancara, didapati bahwa PT XYZ membutuhkan sistem yang dapat menggantikan Oracle Form untuk menekan biaya pemeliharaan aplikasi.

Tahapan selanjutnya adalah tahap desain. Tahapan ini ditujukan untuk mendapatkan rancangan sebuah sistem yang nantinya akan digunakan untuk memecahkan masalah yang dihadapi. Pada penelitian ini, sistem yang akan dibuat diharuskan mampu memenuhi kebutuhan dan proses bisnis yang ada pada Oracle Form.

Setelah dilakukan perancangan, maka dilakukan implementasi yang menerapkan sistem yang telah dirancang sebelumnya. Pada penelitian ini, Sistem yang telah dirancang akan dibuat pada platform *website* melalui *coding* sesuai dengan desain yang dibuat dengan bahasa Python dan *framework* Flask untuk *Backend*, Bootstrap dan JQuery untuk *frontend*, serta PostgreSQL sebagai *database*.

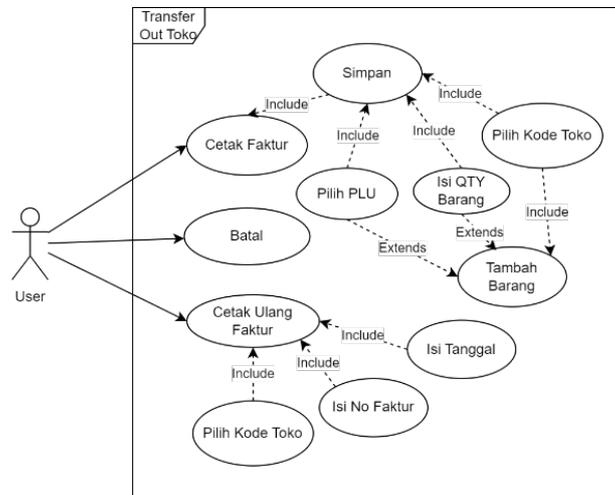
Terakhir, dilakukan pengujian aplikasi oleh *Quality Assurance* (QA) untuk mengetahui apakah aplikasi sudah sesuai dengan kebutuhan *user*. QA menganalisa *input* dan *output* dari sistem dan memastikan sudah sesuai dengan desain yang ditentukan [15] dan untuk memeriksa segala kemungkinan terjadinya kesalahan. Pengujian juga bertujuan untuk mengetahui jumlah *error* atau *bug* yang terjadi pada aplikasi yang dibangun. Dari hasil pengujian, QA akan membuat laporan. Jika masih terdapat *bug* atau *error*, aplikasi akan dikembalikan ke *programmer* untuk diperbaiki.

Dalam penelitian ini, data diperoleh dari seorang *Senior Programmer* di PT. XYZ melalui teknik wawancara tatap muka dengan menggunakan pedoman wawancara yang telah disusun sebelumnya. Pedoman wawancara berisi pertanyaan-pertanyaan terkait penggunaan teknologi lama pada sistem aplikasi yang telah digunakan sebelumnya, kendala yang dihadapi dalam penggunaan teknologi lama, serta manfaat yang diperoleh setelah beralih ke teknologi baru dengan menggunakan Python dan *framework* Flask.

Pada pengujian aplikasi oleh tim QA, dilakukan analisis kualitatif terhadap program lama berbasis Oracle Form yang digunakan sebelumnya oleh PT. XYZ. Analisis kualitatif dilakukan dengan cara mengidentifikasi fitur dan kebutuhan bisnis yang ada pada teknologi lama tersebut menjadi sebuah *test scenario*. *Test scenario* tersebut dicatat dan dikumpulkan dalam bentuk tabel. Selanjutnya, tim QA melakukan pengujian terhadap aplikasi baru yang telah dikembangkan dengan membandingkan fitur dan kebutuhan bisnis yang ada pada Oracle Form dengan aplikasi yang baru. Tim QA melakukan interpretasi terhadap hasil pengujian untuk menentukan apakah aplikasi yang baru

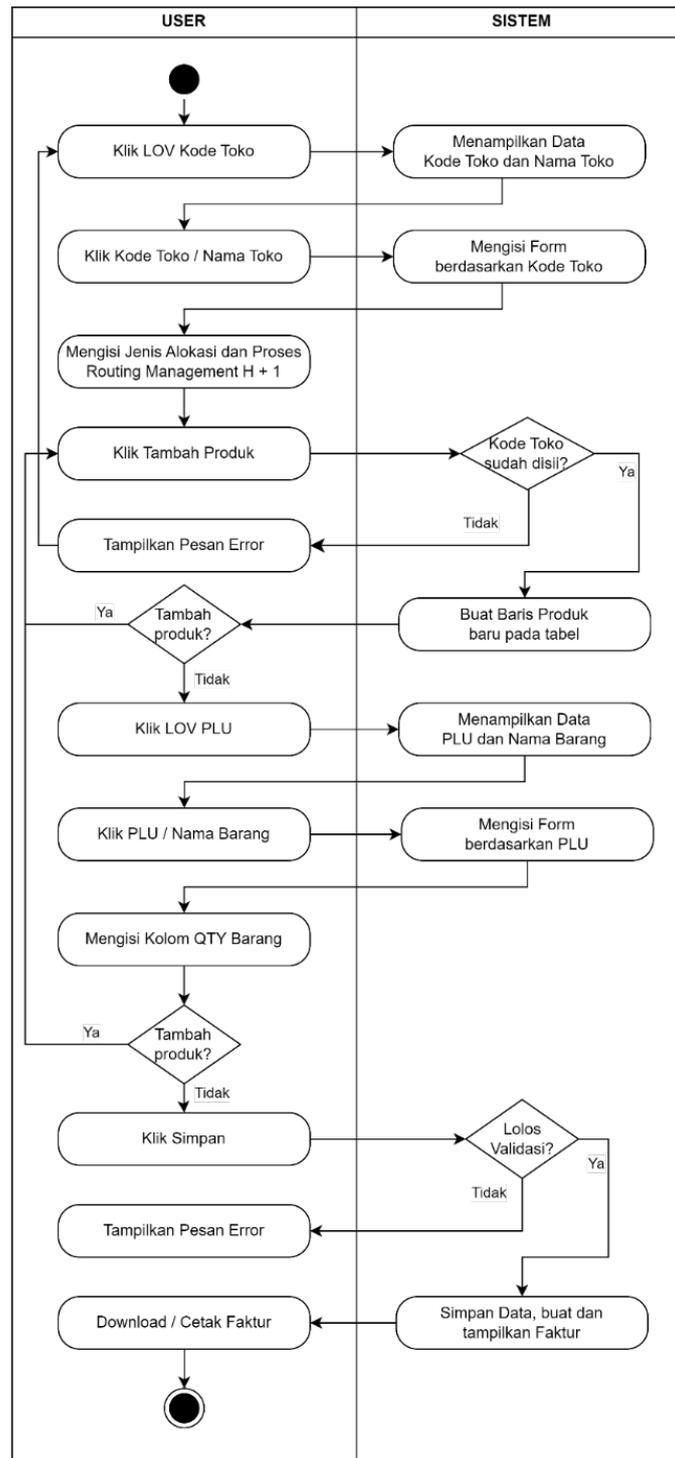
sudah memenuhi kriteria yang dimiliki aplikasi pendahulunya. Hal ini dilakukan dengan cara mengevaluasi fitur dan kode program aplikasi sesuai dengan *System Requirement* pada modul *Transfer Out Toko*.

Berikut adalah bagan UML yang menggambarkan perancangan sistem dalam modul *Transfer Out Toko* dengan menggunakan *use case diagram* dan *activity diagram*.



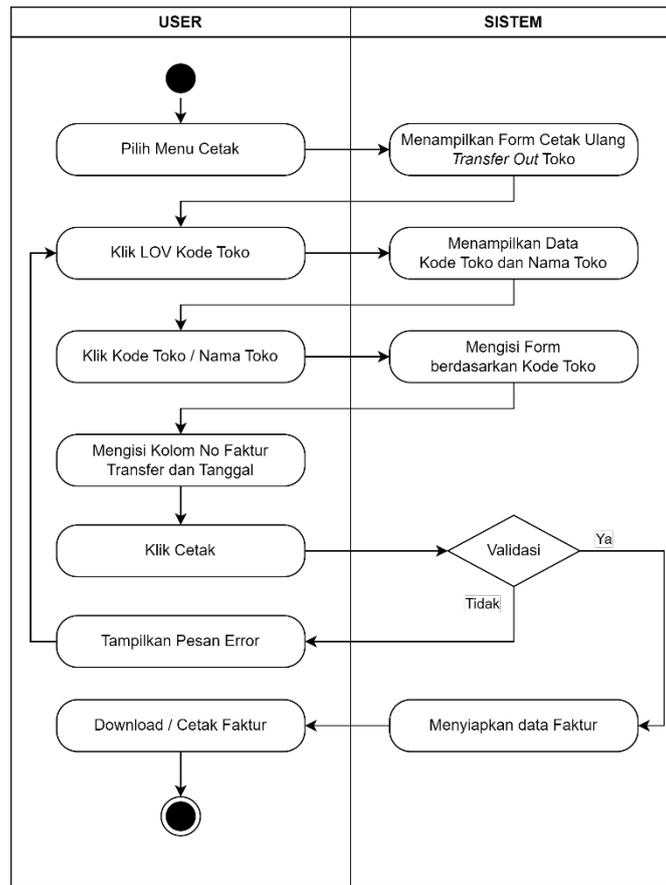
Gambar. 2. Use Case Diagram

Gambar 2 adalah *use case diagram* untuk modul *Transfer Out Toko* yang menjelaskan akses *user* pada aplikasi. *User* dapat membuat transaksi *Transfer Out* dari *distribution center* ke toko cabang yang menghasilkan sebuah faktur. Selain itu, *user* juga dapat mencetak faktur sesuai dengan kode toko, tanggal, dan nomor faktur transaksi yang sudah dilakukan. *User* juga dapat melakukan aksi batal, yaitu menghapus seluruh isi form seperti sedia kala.



Gambar. 3. Activity Diagram Simpan Data

Gambar 3 menunjukkan alur aktivitas dari modul *Transfer Out* Toko ketika user membuat transaksi baru. Langkah pertama yang harus dilakukan oleh *user* adalah memilih kode toko dari toko yang dituju. Kemudian, *user* dapat menambah jumlah barang yang akan dibeli. Untuk setiap barang, *user* perlu mengisi PLU dari barang terkait dan mengatur jumlah barang yang dibutuhkan. Ketika mengeklik tombol simpan, aplikasi akan melakukan proses validasi seperti kode toko, PLU barang dan jumlah barang harus diisi. Jika tidak lolos validasi, proses akan gagal dan user harus memeriksa kembali agar semua kolom terisi dengan benar. Jika semua terisi dengan benar, aplikasi akan melakukan pencatatan pada sistem dan menampilkan modal yang berisi notifikasi berhasil dan nomor faktur transaksi yang dibuat. Selanjutnya, aplikasi menampilkan faktur dari transaksi terkait dalam bentuk file PDF yang dapat diunduh oleh *user*.



Gambar. 4. Activity Diagram Cetak Ulang

Gambar 4 menunjukkan alur aktivitas dari modul Transfer Out Toko ketika *user* mencetak ulang faktur yang pernah dibuat. Pertama, *user* memilih kode toko dari toko yang dituju melalui *List of Values* (LOV). Setelah itu, *user* mengisi nomor faktur dan tanggal faktur yang ingin dicetak ulang. Setelah seluruh kolom terisi, *user* dapat menekan tombol cetak. Sistem akan melakukan validasi, bahwa jika faktur dengan kriteria pada kolom tidak ditemukan, maka akan muncul pesan error dan *user* dapat mengisi ulang kolom pada *form*. Jika faktur ditemukan pada sistem, sistem akan mengembalikan faktur tersebut kepada *user* dalam bentuk *file* PDF yang bisa diunduh.

### III. HASIL DAN PEMBAHASAN

Berikut adalah implementasi Python dan *framework* Flask yang digunakan untuk membangun modul ini. Untuk menjalankan aplikasi yang menggunakan *framework* Flask, ditambahkan kode seperti yang tercantum pada Kode Program 1 di file *WSGI.py*, dengan menentukan *host* dan *port* yang diinginkan.

TABEL 1  
 KODE PROGRAM WSGI.PY

```

from application import create_app

app = create_app()

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5001, debug=True)
    
```

Konfigurasi aplikasi berbasis *framework* Flask dapat dilakukan dengan menggunakan Kode Program 2. Pada cuplikan Kode Program 2, dilakukan inisialisasi aplikasi Flask, pengelolaan konfigurasi, dan pendaftaran *blueprint*.

TABEL 2  
KODE PROGRAM INIT .PY

```
from importlib import import_module
from flask import Flask
from flask_session import Session
from flask_cors import CORS
# ...
from controller import tf_out_toko_bp
# ...

def create_app():
    app = Flask(__name__, instance_relative_config=True)
    CORS(app)
    app.config.from_object("config.Config")
    # ...
    app.config["KEY"] = "VALUE"
    # ...
    app.register_blueprint(tf_out_toko_bp)
    # ...
    return app
```

Konfigurasi *Blueprint* dari *framework* Flask terlampir pada Kode Program 3, dimana dilakukan inisialisasi *Blueprint* beserta *route* URL-nya. Aplikasi membuat *get request* ke API untuk mengambil data yang akan ditampilkan oleh LOV Toko. Hasil dari API akan dikembalikan dalam bentuk JSON untuk ditampilkan pada *Frontend*.

TABEL 3  
CONTOH CONTROLLER APLIKASI

```
from flask import request, render_template, request, Blueprint, session
from config import Config
import requests
import json

tf_out_toko_bp = Blueprint(
    "TransferOutTokoController",
    __name__,
    url_prefix="PREFIX-URL-DISINI",
)

# ...

@tf_out_toko_bp.route("/URL-LOV ", methods=["GET"])
def get_lov():
    kode = request.headers["kode"]

    query_string = request.get_json().get("query_string")

    response = requests.get(
        url=Config.BACKEND_URL(session["kode"]) + API_URL + "lov",
        headers=kode,
        data=json.dumps(data),
    )

    datas = response.json()
    return datas

# ...
```

Berikut merupakan implementasi Python dan *framework* Flask pada salah satu *controller* milik API untuk menampilkan salah satu LOV yang ada pada aplikasi. API akan mengambil data dari *database* melalui *Data Access Object* (DAO). Hasil kembalian dari DAO akan dikembalikan lagi sebagai *response* dalam bentuk JSON.

TABEL 4  
CONTOH CONTROLLER API

```
from flask import request, Blueprint, jsonify, request
from dao import TransferOutToko

tf_out_toko_bp = Blueprint(
    "TransferOutTokoController",
    __name__,
    url_prefix="PREFIX-URL-DISINI",
)

#...

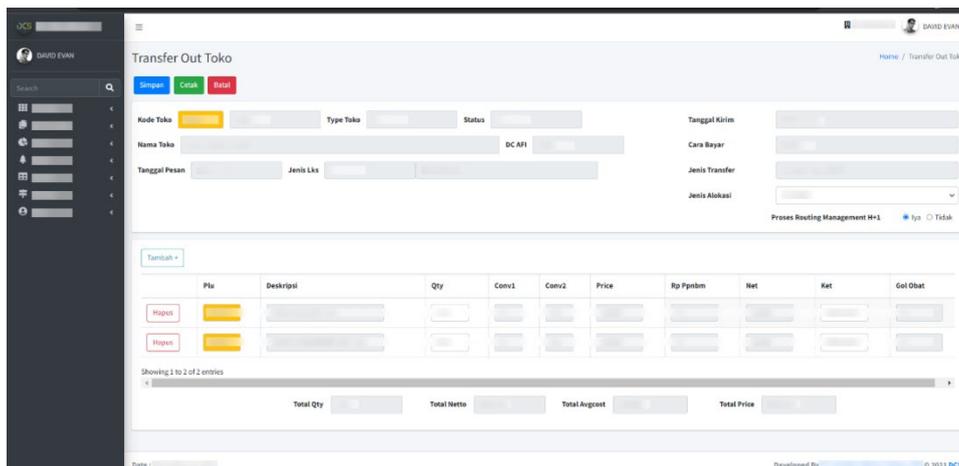
@tf_out_toko_bp.route("/URL-LOV ", methods=["GET"])
def get_lov():
    kode = request.headers["kode"]

    query_string = request.get_json().get("query_string")

    dao = TransferOutToko(kode)
    result = dao.get_lov(query_string)
    return jsonify(result)

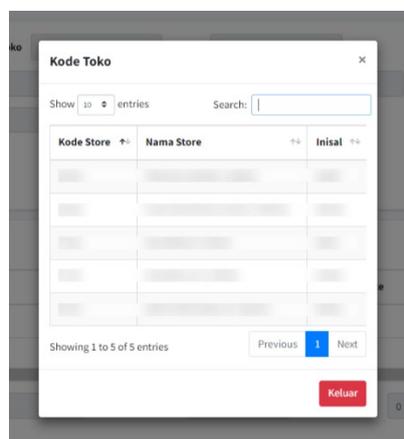
# ...
```

Berikut adalah tampilan halaman utama yang digunakan pada penelitian ini.



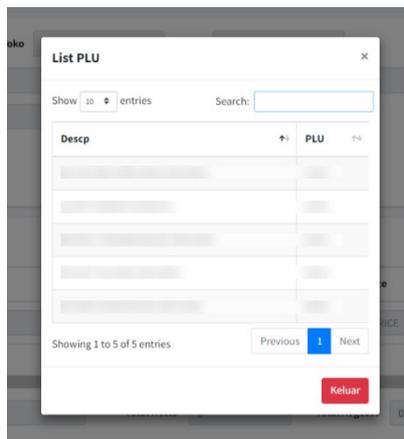
Gambar. 5. Halaman *Transfer Out Toko*

Gambar 5 adalah tampilan awal dari modul *Transfer Out Toko* yang bertugas untuk melakukan pencatatan transaksi keluarnya barang dari *Distribusi Center* (DC) ke toko beserta pencetakan fakturnya. *User* dapat menambah jumlah barang yang diinginkan dengan menekan tombol tambah setelah memasukkan kode toko. Setelah selesai, *user* dapat melakukan pencatatan dengan menekan tombol simpan.



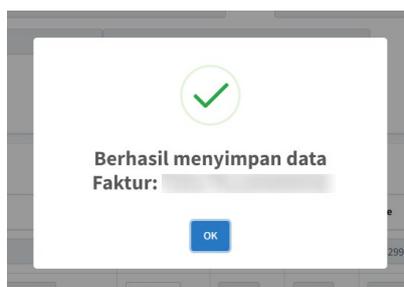
Gambar. 6. LOV Kode Toko

Gambar 6 merupakan tampilan dari *popup List of Values (LOV) Kode Toko* yang akan membantu pengguna memasukkan informasi tentang toko yang dituju. Disini pengguna dapat memilih dengan cara mengeklik baris yang ada pada LOV atau mencarinya terlebih dulu melalui *searchbar* berdasarkan kode atau nama toko.



Gambar. 7. LOV PLU

Adapun Gambar 7 menunjukkan tampilan dari *popup LOV PLU* yang akan membantu pengguna mengisi informasi produk berdasarkan PLU atau nama produk yang hendak dikirimkan.



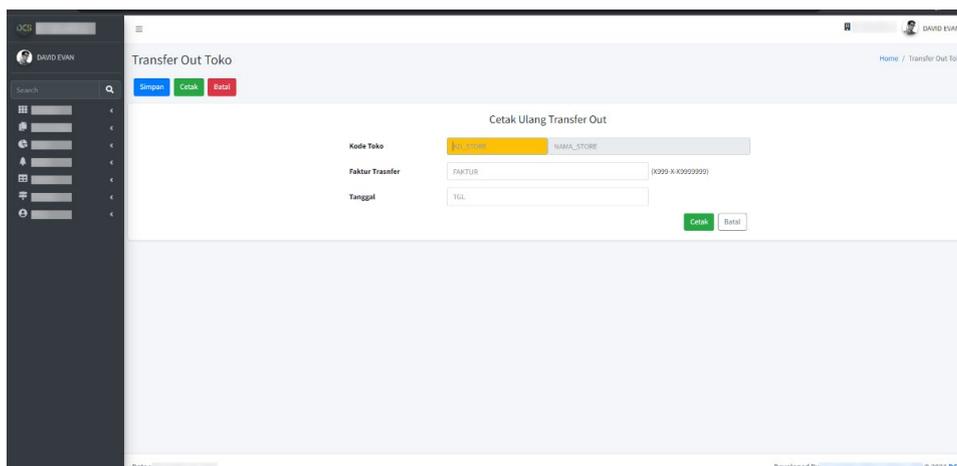
Gambar. 8. Notifikasi berhasil simpan

Gambar 8 menunjukkan tampilan *popup* yang akan muncul ketika sistem berhasil melakukan pencatatan. *Popup* juga akan menampilkan nomor faktur dari transaksi yang baru saja dibuat. Setelah tombol ok ditekan, aplikasi akan menyediakan faktur dalam bentuk PDF yang langsung bisa diunduh.



Gambar. 9. Faktur PDF

Gambar 9 menunjukkan tampilan faktur yang didapat setelah sistem berhasil melakukan pencatatan transaksi. Faktur yang dihasilkan adalah *file* dalam bentuk PDF yang bisa dicetak untuk kepentingan proses bisnis selanjutnya.



Gambar. 10. Tampilan Cetak Ulang

Gambar 10 menunjukkan tampilan *form* yang digunakan untuk mencetak ulang faktur. *Form* ini akan muncul setelah pengguna menekan tombol cetak yang terletak di samping tombol simpan.

Selanjutnya dilakukan pengujian sistem pada aplikasi modul *Transfer Out* Toko menggunakan metode unit *testing*. Metode pengujian ini dilakukan dengan menguji setiap unit program secara terpisah untuk memastikan setiap unit tersebut berfungsi dengan baik.

Dalam pengujian unit *testing*, dilakukan pengujian terhadap fungsi-fungsi yang terdapat dalam modul *Transfer Out* Toko. Setiap fungsi diuji dengan memberikan *input* yang bervariasi dan memeriksa *output* yang dihasilkan. Namun, dari hasil pengujian tersebut, ditemukan beberapa kesalahan yang perlu diperbaiki, yaitu: data yang diambil melalui LOV tidak lengkap, tombol tambah tidak menambah jumlah barang, hasil perhitungan total untuk faktur tidak sesuai, dan fungsi validasi untuk kolom QTY tidak aktif. Dengan melakukan pengujian unit *testing*, kesalahan-kesalahan pada setiap unit program dapat diidentifikasi dan diperbaiki secara lebih efisien dan efektif, sehingga dapat meningkatkan kualitas dari modul *Transfer Out* Toko secara keseluruhan.

Setelah pengujian mandiri dan perbaikan dilakukan, modul diserahkan kepada tim *Quality Assurance* (QA) PT.XYZ untuk pengujian lebih lanjut. Hasil pengujian dari QA dijabarkan sebagai berikut:

TABEL 5  
 HASIL PENGUJIAN QA

No	Uraian	Output	Status
1	Jika DC yang dikehendaki tidak aktif akan menampilkan <i>alert</i> "tidak bisa <i>transfer out</i> "	Sudah sesuai <i>System Requirement</i>	OK
2	Jika stok awal belum dilakukan maka akan menampilkan <i>alert</i> "proses <i>stock</i> awal belum dilakukan"	Sudah sesuai <i>System Requirement</i>	OK
3	User memilih toko dengan LOV	Sudah sesuai <i>System Requirement</i>	OK
4	Setelah memilih toko, <i>user</i> diarahkan mengisi PLU dengan LOV	Sudah sesuai <i>System Requirement</i>	OK
5	Setelah user memilih PLU berdasarkan LOV, maka terdapat validasi	Sudah sesuai <i>System Requirement</i>	OK
6	Saat klik <i>save</i> data tersimpan pada tabel yang benar pada <i>database</i>	Sudah sesuai <i>System Requirement</i>	OK
7	Mencetak Barcode PLU dan Faktur Transaksi yang tersimpan	Sudah sesuai <i>System Requirement</i>	OK

Tabel 5 menampilkan hasil pengujian sistem yang dilakukan oleh tim *Quality Assurance* (QA) PT. XYZ dengan menggunakan metode *whitebox testing*. Proses QA ini dilakukan untuk memastikan bahwa aplikasi yang dibangun telah memenuhi kriteria yang diharapkan oleh pengguna. Proses QA dimulai dengan menelusuri alur aplikasi berbasis Oracle Form untuk modul *Transfer Out* Toko. Setelah itu, dibuat *test scenario* berdasarkan *input*, *output*, dan perilaku dari aplikasi Oracle Form untuk diujikan pada aplikasi web.

Selanjutnya dilakukan pengujian untuk memastikan bahwa modul *Transfer Out* Toko berfungsi dengan baik sesuai dengan versi Oracle Form. Pengujian dilakukan dengan memasukkan modul ke dalam sebuah *test environment* dan memastikan bahwa aplikasi berjalan dengan baik dan menghasilkan *output* yang sesuai dengan harapan. Jika masih terdapat *bug* atau *error*, aplikasi akan dikembalikan ke *programmer* untuk diperbaiki. Dari hasil pengujian pada Tabel 1 didapat hasil bahwa modul *Transfer Out* Toko telah memenuhi kriteria yang diharapkan oleh pengguna.

Hasil pengembangan modul *Transfer Out* Toko menunjukkan bahwa penggunaan teknologi terbaru seperti Python dan *framework* Flask memberikan manfaat dalam mengurangi biaya pemeliharaan aplikasi dibandingkan teknologi lama berbasis Oracle Form. Aplikasi yang dikembangkan menggunakan teknologi terbaru juga dapat dioperasikan di berbagai platform. Temuan dari pengujian tim QA menunjukkan bahwa modul *Transfer Out* Toko menggunakan Python dan *framework* Flask mampu memenuhi kriteria yang diharapkan oleh pengguna. Dengan adanya pengurangan biaya pemeliharaan aplikasi, PT. XYZ dapat mengalokasikan anggaran untuk pengembangan aplikasi lain atau kebutuhan bisnis lainnya yang lebih produktif.

Dalam penelitian sebelumnya, telah terbukti bahwa *framework* Flask dapat menjadi solusi untuk aplikasi berbasis web menggunakan Python, seperti pada penelitian yang dilakukan oleh Suraya dan Dinda. Namun, penelitian yang dilakukan ini lebih fokus pada pengembangan aplikasi *Distribution Center System* serta mengatasi masalah biaya pemeliharaan aplikasi dan kemampuan aplikasi untuk beroperasi di berbagai platform.

#### IV. KESIMPULAN

Dalam penelitian ini, dilakukan pengembangan modul *Transfer Out* Toko pada aplikasi *Distribution Center System* dengan mengadopsi teknologi Python dan *framework* Flask. Pengujian dilakukan untuk memastikan bahwa modul tersebut berfungsi dengan baik sesuai dengan versi Oracle Form dan memenuhi kriteria yang diharapkan oleh pengguna. Hasil pengembangan modul ini menunjukkan bahwa penggunaan teknologi terbaru seperti Python dan *framework* Flask memberikan manfaat dalam mengurangi biaya pemeliharaan aplikasi dibandingkan teknologi lama berbasis Oracle Form. Aplikasi yang dikembangkan menggunakan teknologi terbaru juga dapat dioperasikan di berbagai platform. Dengan adanya pengurangan biaya pemeliharaan aplikasi, PT. XYZ dapat lebih fleksibel dalam mengalokasikan anggaran perusahaan. Adapun beberapa saran yang dapat diterapkan kedepannya untuk modul *Transfer Out* Toko ini seperti melakukan evaluasi dan pemeliharaan aplikasi secara berkala baik dari segi fitur, tampilan, atau hal lain yang dinilai perlu untuk menunjang kebutuhan bisnis.

#### DAFTAR PUSTAKA

- [1] S. S. Kidi and W. A. Madya, "Teknologi dan Aktivitas Dalam Kehidupan Manusia," 2018.
- [2] PT Sumber Alfaria Trijaya Tbk, "Laporan Tahunan 2021 PT Sumber Alfaria Trijaya Tbk," *alfamart.co.id*, 2021.
- [3] TessyBadriyah, "BAB 4 MEMBUAT LIST OF VALUE (LOV) DAN NON-BASE TABLE FIELDS."
- [4] B. P. Putra and Y. A. Susetyo, "IMPLEMENTASI API MASTER STORE MENGGUNAKAN FLASK, REST DAN ORM DI PT XYZ," *Jurnal Sistem Informasi (SISTEMASI)*, pp. 543–556, 2020.
- [5] Python Software Foundation, "What is Python? Executive Summary," *Python.org*, 2019.
- [6] Kalyani Adawadkar, "Python Programming-Applications and Future," *International Journal of Advance Engineering and Research Development (IAERD)*, pp. 1–4, Apr. 2017.
- [7] Nidhi Chauhan, Ayushi Verma, Mandeep Singh, Aashwaath Parasher, and Gaurav Budhiraja, "Implementation of Database Using Python Flask Framework," *International Journal Of Engineering And Computer Science (IJECS)*, vol. 8, no. 2, pp. 24894–24899, 2019.
- [8] Devndra Ghimire, "Comparative study on Python web frameworks: Flask and Django," 2020.
- [9] Kunal Relan, *Building REST APIs with Flask*. CA: Apress, 2019.
- [10] E. Haezer, Y. Kristianto, and N. Setiyawati, "PEMBANGUNAN APLIKASI VIRTUAL INVENTORY SYSTEM (VIS) BERBASIS WEB MENGGUNAKAN FLASK FRAMEWORK (STUDI KASUS: PT XYZ)," 2021.
- [11] M. Sholeh, "Designing and Implementing a Database for Thesis Data Management by Using the Python Flask Framework," *International Journal of Engineering, Science & Information Technology (IJESTY)*, 2022, doi: 10.52088/ijesty.v1i1.197.
- [12] D. F. Ningtyas and N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request Flask Framework Implementation in Development Purchasing Approval Request Application," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.
- [13] A. Sri Wahyuningsih and S. Junianto, "PENERAPAN KNOWLEDGE MANAGEMENT SISTEM LAYANAN PERBAIKAN DEVICE DISTRIBUTION CENTER (DC) SUPPORT PT.INDOMARCO PRISMATAMA CABANG BEKASI," *Jurnal Informatika SIMANTIK*, vol. 5, no. 1, 2020, [Online]. Available: [www.jurnal.stmki cikarang.ac.id](http://www.jurnal.stmki cikarang.ac.id)
- [14] Roger S. Pressman, *Software Engineering: A Practitioner's Approach*. 2015.
- [15] Roy Mubarak, "Implementasi Metode White Box Testing Pada Proses Quality Assurance Perangkat Lunak Berbasis Web Dan Mobile Collection System," *Jurnal ESIT (E-Bisnis, Sistem Informasi, Teknologi Informasi)*, 2021.