

IMPLEMENTASI *FLASK* PADA SISTEM PENENTUAN *MINIMAL ORDER* UNTUK TIAP ITEM BARANG DI *DISTRIBUTION CENTER* PADA PT XYZ BERBASIS WEBSITE

Graciela Fausten Novindri¹, Pratyaksa Ocsa Nugraha Saian²

^{1,2} Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Satya Wacana
672018057@student.uksw.edu

ABSTRAK

PT XYZ adalah sebuah jaringan ritel besar yang bergerak di Indonesia dengan perkembangan teknologi yang semakin pesat, PT XYZ pasti akan terus berinovasi agar tidak kalah saing dengan perusahaan lain. Dalam memaksimalkan keuntungan dan mengurangi kerugian dalam penjualan tiap barang yang ada di toko maka dibuatlah sistem minimal order untuk tiap barang yang nantinya akan dikirimkan dari *Distribution Center* ke toko-toko. Tujuan dari penelitian ini adalah membangun sebuah sistem minimal order berbasis website dengan menggunakan bahasa pemrograman Python dan memanfaatkan *Framework Flask*. Penelitian ini memakai sebuah metode *Waterfall* menurut Pressman dengan tahapan penelitian yaitu *System Analyst, Planning and Design System, Implementation, Testing*. Hasil penelitian adalah sebuah sistem yang berfungsi untuk mengatur minimal order tiap item barang yang berbasis *Website* dengan menggunakan *Framework Flask*. Sistem pada penelitian ini diuji dengan menggunakan metode *Black Box* dimana dilakukan pengujian terhadap setiap fungsi, masukan, dan keluaran yang ada dan pengujian oleh *Quality Assurance* menunjukkan hasil bahwa semua fungsi yang ada telah sesuai dengan *User Manual*. Berdasarkan dua kali proses pengujian didapatkan bahwa fungsi-fungsi yang terdapat pada sistem telah sesuai dan dapat berjalan dengan apa yang diinginkan sehingga sistem ini dapat digunakan oleh *user* untuk menentukan nilai minimal order dari sebuah barang.

Keyword : *Sistem Minimal Order, Implementasi Flask, Pembangunan Website*

1. PENDAHULUAN

Banyak inovasi baru telah diciptakan untuk memudahkan pekerjaan manusia. Salah satunya adalah bisnis retail, banyak perusahaan yang mencari cara dan terus berinovasi agar bisa cepat mengembangkan bisnis retail distribusi. Pada bisnis ritel, kepuasan pelanggan menjadi hal terpenting untuk memberikan jaminan keberlangsungan hidup dari perusahaan. Selain itu, terdapat faktor yang dapat memberikan kepuasan pelanggan yaitu ketersediaan stok atau *stock availability*.

Persediaan merupakan sekumpulan bahan yang berada di dalam bisnis untuk proses produksi, dan sebuah produk yang dipersiapkan setiap saat guna mencukupi kebutuhan dari konsumen atau pelanggan [1]. Pengendalian persediaan bertujuan untuk mempertahankan supaya jumlah dari persediaan ada pada tingkat yang optimal sehingga perusahaan tidak mengalami kerugian. Stok barang yang tidak mencukupi bisa mengakibatkan *loss of sales* atau kehilangan kesempatan menjual. Begitu juga sebaliknya persediaan barang yang terlalu banyak dapat menyebabkan *over stock*. Jika barang *over stock* dan memiliki *expired date* yang melebihi batas akan menyebabkan barang tersebut harus dimusnahkan atau dibuang dan menyebabkan kerugian bagi bisnis retail. Sehingga diperlukan perhitungan yang tepat untuk mengatur mengenai minimal order untuk tiap barang yang akan dikirimkan dari *Distribution Center* ke toko-toko yang menjadi tempat penjualan barang.

PT XYZ adalah sebuah jaringan ritel besar yang bergerak di Indonesia dengan perkembangan

teknologi yang semakin pesat. PT XYZ pasti akan terus berinovasi agar tidak kalah saing dengan perusahaan lain. Pada tahun 2020, perusahaan mengoperasikan 15.434 cabang di seluruh Indonesia, di mana 1.124 di antaranya merupakan gerai baru. Untuk mencukupi kebutuhan pasokan 15.434 cabang, hingga akhir tahun 2020, perusahaan terus didukung oleh 32 pusat distribusi yang berada di 25 provinsi di Indonesia sebagai pusat distribusi untuk jaringan penjualan perusahaan [2].

Untuk memaksimalkan keuntungan dan mengurangi kerugian dalam penjualan tiap barang yang ada di toko maka diperlukannya sebuah sistem minimal order untuk menentukan nilai order untuk tiap toko yang ada.

Pada penelitian yang berjudul *Flask Framework Implementation in Development Purchasing Approval Request Application* oleh Dinda didapatkan dalam pembangunan aplikasi *Purchasing Approval Request Flask* memiliki sifat *simplicity* dimana sifat tersebut mempermudah pembangunan pada sebuah aplikasi dan dapat menghasilkan sebuah aplikasi yang optimal sesuai dengan kebutuhan, adanya *decorator* dan *library* yang telah disediakan menjadi faktor utama yang mempermudah pembangunan aplikasi [3].

Pada penelitian yang dilakukan oleh Jessica dan Sigit yang berjudul *Sistem Informasi Akuntansi Persediaan Bahan Baku Menggunakan Metode Economic Order Quantity (Studi Kasus : PT SSH)* menunjukkan hasil bahwa metode *Economic Order Quantity* dapat menangani permasalahan mengenai persediaan yang tidak diprediksi dengan baik.

Pembangunan sistem informasi akuntansi persediaan bahan baku yang dibuat dengan menggunakan *framework CodeIgniter* dapat mempermudah petugas dalam melakukan perhitungan [4].

Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya, maka peneliti membuat sebuah sistem yang digunakan untuk menentukan minimal order untuk tiap barang di *Distribution Center* dimana belum terdapat penelitian yang membahas mengenai pembangunan sistem minimal order berbasis *website* dengan memanfaatkan *Framework Flask*.

Sistem minimal order berbasis *website* ini dibangun dengan memakai *framework* Flask yang menggunakan bahasa pemrograman Python. *Framework* Flask termasuk ke dalam *micro-framework* dikarenakan tidak membutuhkan tools atau library tertentu dalam penggunaannya. Flask juga menyediakan library dan kumpulan kode program yang dapat digunakan untuk membangun sebuah *website*. Selain itu, Flask juga dapat dikembangkan dan mendukung pengembangan ekstensi di atas kerangka inti untuk menambahkan fitur aplikasi sehingga seolah-olah ekstensi merupakan bagian dari Flask itu sendiri [5].

2. TINJAUAN PUSTAKA

Distribution Center diperlukan dalam proses koordinasi pendistribusian barang yang timbul akibat ketidakseimbangan dalam proses *supply* (penawaran) dan *demand* (permintaan). Kurangnya keseimbangan antara proses penawaran dan permintaan mendorong terciptanya *inventory* (persediaan), persediaan membutuhkan ruang sebagai tempat penyimpanan sementara yang dikenal dengan istilah gudang.

2.1. Sistem

Jogiyanto menyatakan bahwa pengertian dari sistem terbagi menjadi dua pendekatan, yaitu pendekatan prosedural dan pendekatan komponen. Pengertian sistem menurut pendekatan prosedural adalah jaringan prosedur-prosedur yang saling berhubungan, disatukan untuk melakukan suatu kegiatan atau menyelesaikan suatu tujuan tertentu. Sedangkan pengertian sistem dalam pendekatan komponen adalah sekumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu [6].

2.2. Flask

Flask merupakan web *framework* yang ditulis dalam bahasa Python. *Flask* memiliki dua *external libraries* yaitu *WSGI* toolkit dan *Jinja2 template engine*. *Flask* merupakan jenis *microframework* yang tidak memerlukan library tertentu dalam penggunaannya. *Flask* dapat menggunakan ekstensi untuk menambahkan fitur dan komponen yang sudah disediakan oleh pihak ketiga dan tidak

terpasang secara standar pada *Flask* seperti *Form Validation*, *Upload Handling*, dan *Database* [7].

2.3. Website

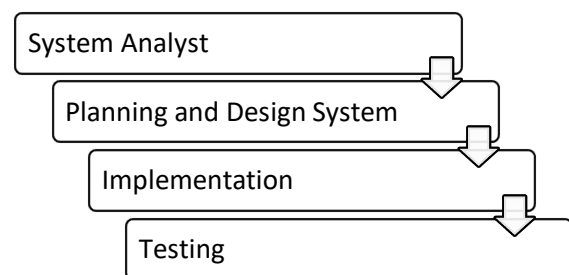
Pengertian Website ialah kumpulan dari beberapa halaman web dimana informasi berupa teks, gambar, suara, dan lain-lain disajikan dalam bentuk *hypertext* dan dapat diakses dengan menggunakan perangkat lunak yaitu *browser*. Informasi di sebuah website pada umumnya ditulis dengan format HTML. Situs web adalah alat online yang menghubungkan dokumen secara lokal dan jarak jauh.

2.4. Python

Python merupakan bahasa pemrograman tingkat tinggi yang sering digunakan oleh *developer* dalam pembuatan atau pengembangan aplikasi saat ini. Python mendukung pemrograman berorientasi objek serta dirancang agar mudah dipelajari dan digunakan. [8]

3. METODE PENELITIAN

Dalam penelitian ini memanfaatkan metode *Waterfall* menurut Pressman [9]. Tahapan penelitian ini dilakukan sebagai berikut:



Gambar 1. Tahap Metodologi Penelitian

Penjelasan dari tahap metodologi penelitian yang digunakan oleh peneliti, sebagai berikut:

3.1. System Analyst

System Analyst atau analisis sistem ini adalah tahap awal pengembangan sistem dimana dilakukan analisis kebutuhan sistem yang akan digunakan dalam penelitian. Pada tahap ini dilakukan wawancara kepada *Programmer Analyst* di PT XYZ mengenai kebutuhan dari sistem yang akan dibuat.

3.2. Planning and Desain System

Pada tahap perancangan memiliki tujuan untuk membuat desain sebuah sistem yang nantinya akan digunakan untuk memecahkan masalah yang dihadapi dengan menggunakan pemilihan sistem yang terbaik

3.3. Implementation

Implementasi adalah tahapan yang menerapkan sistem yang telah dirancang dan dibuat pada sebuah *platform* yang akan digunakan, dalam penelitian ini yaitu *website*. Implementasi dilakukan

dengan cara melakukan *coding* sesuai dengan desain yang dibuat dengan menggunakan bahasa Python dan *Framework* Flask

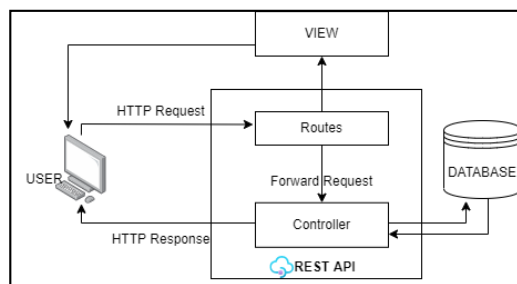
3.4. Testing

Pengujian atau testing merupakan tahap akhir yang merupakan tahapan untuk menguji aplikasi yang sudah dibuat, apakah sistem yang dikerjakan sudah sesuai dengan yang dirancang semula dan juga memastikan semua fungsi yang ada dalam sistem dapat berfungsi dan bekerja dengan baik.[10]

4. HASIL DAN PEMBAHASAN

4.1. Arsitektur Sistem

Sistem yang berjalan dalam penelitian ini memiliki fungsi *Create, Read, Update, Delete* dalam pengolahan data. Fungsi-fungsi itu digunakan dalam setiap modul yang digunakan dalam mendukung penentuan nilai minimal order untuk setiap barang yang akan dikirimkan dari *Distribution Center* ke tiap toko-toko yang ada. Sistem ini dibangun juga dengan menggunakan REST API yang digunakan untuk melakukan interaksi dengan menggunakan protokol HTTP untuk pertukaran data. Berikut merupakan arsitektur pada sistem aplikasi:



Gambar 2. Arsitektur Sistem

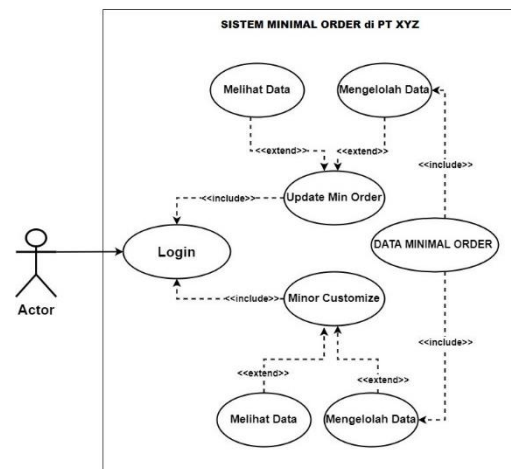
Pada gambar arsitektur sistem menjelaskan bahwa user harus mengakses URL dimana URL tersebut berfungsi sebagai penghubung menuju ke *view* dan *controller*. *Controller* tersebut akan menjadi penghubung untuk dapat mengakses ke dalam database dan nantinya data dari database tersebut akan ditampilkan ke *view* melalui *controller*.

Permintaan *request* yang dilakukan oleh user agar dapat dipahami oleh sebuah sistem biasanya menggunakan POST, GET, PUT, DELETE. Dalam penerimaan data yang diminta adalah *response* berupa JSON yang nantinya akan diolah sesuai dengan kebutuhan.

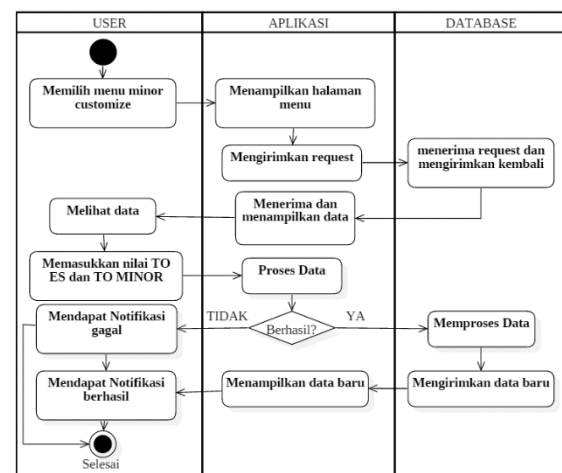
4.2. Perancangan dan Desain Sistem

Pada gambar 3 dapat dilihat *Use Case Diagram* yang ada dalam penelitian ini. Pada gambar tersebut terdapat sebuah *user* yaitu karyawan yang berperan sebagai seorang *actor* yang menjalankan aplikasi ini. *User* tersebut mampu menjalankan beberapa fungsi modul yang ada diaplikasi namun sebelum menjalankan fungsi-fungsi tersebut, *user* harus

melakukan *login* terlebih dahulu. Fungsi utama pada penelitian ini adalah menentukan minimal order untuk tiap item barang, yang ditunjukkan pada *use case Update Minimal Order*.



Gambar 3. Use Case Diagram Minimal Order



Gambar 4. Activity Diagram Minimal Order

Pada Gambar 4 menunjukkan proses minimal order yang dilakukan oleh *user*, dimana pertama *user* harus memilih menu *Minor Customize*. Pada menu ini akan menampilkan data berupa tabel yang berisi toko-toko yang terdapat dalam 1 DC. *User* tinggal memilih mana saja toko yang ingin diatur minimal ordernya lalu mengisi TO_Minor (nilai minimal order) dan TO_ES (nilai lama ketahanan stok barang) yang terdapat pada tabel. Jika proses yang dilakukan berhasil maka akan menyimpan data ke database dan jika gagal akan menampilkan notifikasi.

Pelaksanaan pemrosesan data dilakukan dengan menggunakan perhitungan dari rumus internal milik perusahaan untuk menghasilkan nilai akhir jumlah barang yang dapat dipesan. Proses ini diimplementasikan kedalam bahasa pemrograman Python sesuai dengan rumus perhitungan yang ada.

4.3. Implementasi Sistem

Berikut merupakan implementasi dari flask yang digunakan di dalam sistem ini. Dalam menjalankan aplikasi yang menggunakan Framework *Flask* ini, di file *main.py* harus ditambahkan kode seperti pada contoh Kode Program 1, dengan memasukkan *host* dan *port* yang diinginkan.

```
1. from flask import Flask
2.
3. app = Flask(__name__)
4. if __name__ == '__main__':
5.     app.run(host='0.0.0.0', port=7007, debug=True)
```

Kode Program 1. Menjalankan Flask

Cara implementasi Framework *Flask* pada sistem dapat dilakukan dengan menggunakan Kode Program 2 dimana dilakukan pemanggilan *Flask* dengan cara *import*. Fungsi *route* *Flask* merupakan sebuah *decorator* yang mana nantinya URL tersebutlah yang akan dipanggil oleh sistem. *Flash* berfungsi untuk menampilkan pesan.

```
1. from flask import render_template, request, g, session, redirect, jsonify,
2. flash
3. @app.route('/minorCustomize', methods=['GET', 'POST'])
4.
5. def minorCustomize():
6.     uResponse = requests.get(url + '/userAccess', data={'user': g.user, 'modul':
7.         'kode_modul'}).json()
8.     if not g.user:
9.         flash('session timeout!')
10.    return redirect(url_for('login'))
11. else:
12.    return render_template('minorCustomize.html', akseshapus=uResponse
13.    ['hapus'], aksesproses=uResponse['proses'])
```

Kode Program 1. Import Flask

Penerapan *Flask-RESTful* dalam penelitian ini dilakukan dengan menggunakan cara seperti pada Kode Program 3, dimana *resource* didapat dengan melakukan *import* pada controller yang berisi *class*, sebagai berikut:

```
1. from flask import Flask
2. from flask_restful import Api
3.
4. app = Flask(__name__)
5. api = Api(app)
6.
7. from controller.minorCustomizeController import
8. minorCustomize
9.
10. api.add_resource(minorCustomize, '/transMinorCustomize/')
11.
12. if __name__ == '__main__':
13.     app.run(host='0.0.0.0', port=7007, debug=True)
```

Kode Program 2. Penerapan Flask Restful

Dalam melakukan pertukaran data antara server dan klien digunakan *jQuery* yang merupakan *library* dari *JavaScript* dengan *Asynchronous Javascript and XML (AJAX)* seperti pada contoh kode program 4.

```
1. $.ajax({
2.     type: 'POST',
3.     url: '/transMinorCustomize/',
4.     data: dataMinorCustomize,
5.     contentType: "application/json",
6.     traditional: true,
7.     dataType: 'json',
8.     success: function (data) {
9.         SUCCESS
10.    },
11.    error: function (error) {
12.        ERROR
13.    }
14. });
```

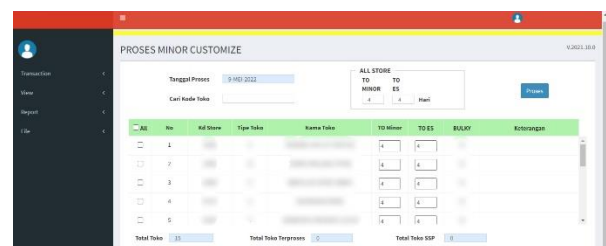
Kode Program 3. Penggunaan AJAX

Resource merupakan salah satu fungsi yang disediakan oleh *Flask-RESTful* yang akan memberikan akses ke beberapa *method* HTTP seperti pada Kode Program 5. Pada server dalam menerima data yang dikirimkan klien dengan menggunakan salah satu fungsi *Flask* yaitu *Request*.

```
1. from flask_restful import Resource
2.
3. class minorCustomize(Resource):
4.     def __init__(self):
5.         KONEKSI KE DATABASE
6.     def get(self):
7.         try:
8.             data = request.get_json()
9.             data2 = request.args.get('data2')
10.            return jsonify({'data': list_Data})
11.        except Exception as e:
12.            message = 'Proses Minor Customize Gagal..!'
13.            print(e)
14.            return message
15.    def put(self):
16.        try:
17.            PROSES
18.        except Exception as e:
19.            message = 'Proses Minor Customize Gagal..!'
20.            print(e)
21.            return message
22.    def post(self):
23.        try:
24.            PROSES
25.        except Exception as e:
26.            message = 'Proses Minor Customize Gagal..!'
27.            print(e)
28.            return message
```

Kode Program 4. Resource Flask-Restful

Pada penelitian ini *Flask* berguna dalam pembangunan *back-end* dimana diimplementasikan untuk komunikasi antara sistem dan *user*. Berikut adalah tampilan halaman utama yang digunakan pada penelitian ini



Gambar 5. Halaman Update Minor Customize

Halaman *update minor customize* ini merupakan fungsi utama dalam penelitian ini,

halaman ini berfungsi untuk mengolah data yang akan menentukan minimal order suatu barang yang nantinya barang tersebut akan dikirimkan oleh DC ke toko sesuai dengan hasil pengolahan data yang ada dalam modul ini.

Data yang diolah dalam modul ini adalah data penjualan untuk tiap toko. Halaman ini akan mengakses data yang terdapat dalam database dan akan ditampilkan ke dalam sebuah tabel, nantinya *user* dapat memasukkan perubahan jumlah TO Minor dan TO ES yang baru pada *field input* yang tersedia. TO Minor merupakan jumlah stok barangnya dan TO ES merupakan ketahanan stok di toko mampu bertahan. *User* dapat memilih toko mana saja yang akan diproses dengan cara memberi *checkbox* pada toko yang diinginkan.

4.4. Testing

Adapun pelaksanaan *testing* pada sistem dilakukan dengan menggunakan teknik pengujian *Black Box* menurut Sukanto untuk mengetahui apakah fungsi, masukan dan keluaran telah sesuai. Berikut adalah rincian rencana pengujian dalam Tabel 1:

Tabel 1. Rincian Pengujian

| No | Komponen Sistem yang diuji | Jenis Pengujian |
|----|----------------------------|------------------|
| 1 | Button Proses | <i>Black Box</i> |
| 2 | Pencarian Kode toko | <i>Black Box</i> |
| 3 | Tanggal Proses | <i>Black Box</i> |
| 4 | Form All Store | <i>Black Box</i> |
| 5 | Pemrosesan Data | <i>Black Box</i> |
| 6 | Checked dan Unchecked | <i>Black Box</i> |

Setelah dilakukan pengujian *Black Box* oleh peneliti, selanjutnya dilakukan pengujian oleh *Quality Assurance* (QA) di PT XYZ. Pengujian yang dilakukan oleh QA yaitu dengan memastikan semua fungsi yang terdapat pada sistem ini bekerja dengan baik dan sesuai dengan apa yang tertulis pada *User Manual*. Berikut adalah hasil dari pengujian yang dilakukan oleh QA :

Tabel 2. Hasil Pengujian QA

| No | Uraian | Output | Status |
|----|---|---|--------|
| 1 | Button Proses | Dapat ditekan untuk memproses data | OK |
| 2 | Cari Kode Toko | Dapat menampilkan data sesuai dengan apa yang di masukkan | OK |
| 3 | Tanggal Proses | Tanggal hari ini | OK |
| 4 | All Store bisa untuk input buat mengubah data, tidak disable. | Mengubah data secara keseluruhan | OK |
| 5 | Input TO Minor dan TO ES jika gagal harusnya ada message di keterangan. | Memunculkan pesan di keterangan bahwa proses gagal | OK |

| No | Uraian | Output | Status |
|----|--|--|--------|
| 6 | Input TO Minor dan TO ES jika berhasil harusnya ada notifikasi | Proses update data berhasil dilakukan dan muncul notifikasi | OK |
| 7 | Kondisi cek pada tabel apakah data sudah siap dan jika sudah memproses data pada bulan yang sama | Muncul notifikasi jika minor belum siap dan jika sudah dilakukan proses pada bulan yang sama | OK |
| 8 | Ceklis All Store dan UnChecklist | Dapat dilakukan checklist di All store dan semua store terchecklist | OK |

Pada Tabel 2 yang menunjukkan hasil dari pengujian QA, apabila status 'OK' berarti hasil *output* pengujian telah sesuai dengan apa yang terdapat pada *User Manual*, sedangkan jika berstatus 'NOK' menunjukkan masih terdapat kesalahan dan harus dilakukan perbaikan terhadap fungsi tersebut.

Setelah pengujian oleh QA terhadap fungsi-fungsi pada sistem ini dan semua fungsi telah sesuai dengan apa yang tertulis dalam *User Manual*, maka sistem ini sudah dapat digunakan oleh *user* dalam membantu untuk menentukan nilai *minimal order* item barang.

5. KESIMPULAN DAN SARAN

Sistem penentuan minimal order untuk tiap item barang di *distribution center* berbasis website ini dibangun untuk memproses data dalam menentukan nilai minimal order untuk setiap barang. Pemanfaatan *Framework* Flask dengan menggunakan bahasa Python mempermudah karena lebih ringan dalam pengimplementasiannya, sehingga peneliti dapat menggunakan *Framework* ini sesuai dengan kebutuhan dengan menambahkan *library* yang disediakan oleh pihak ketiga. Beberapa fungsi Flask yang digunakan dalam sistem ini seperti pengembalian data, penerimaan data, pengaturan route. Berdasarkan pada pengujian sistem dengan menggunakan metode *Black Box* diperoleh hasil bahwa semua fungsi yang terdapat pada sistem ini telah sesuai sehingga sistem ini dapat digunakan oleh *user*.

Berdasarkan pada penelitian yang telah dilakukan oleh peneliti, berikut adalah saran yang dapat diberikan oleh peneliti untuk pengembangan sistem ini yaitu penggunaan bahasa Python dapat lebih diperluas dengan menggunakan validasi-validasi yang disediakan oleh Python dan juga penggunaan *error handling* dapat lebih dikhususkan agar lebih mudah terbaca oleh *user* karena notifikasi gagal masih sangat umum sehingga *user* tidak tahu apa yang menyebabkan kegagalan proses.

DAFTAR PUSTAKA

- [1] L. Fitriyaningrum, M. Arifin, and M. E. Sony, "Rancang Bangun Aplikasi Perencanaan Issn 2338-137X," *Jsika*, vol. 5, no. 8, pp. 1–8, 2016.
- [2] P. S. A. Trijaya Tbk, "Laporan Tahunan 2020 PT Sumber Alfaria Trijaya Tbk," pp. 1–220, 2020.
- [3] D. F. Ningtyas and N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request Flask Framework Implementation in Development Purchasing Approval Request Application," *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.
- [4] S. D. Ramdan, "Sistem Informasi Akuntansi Persediaan Bahan Baku Menggunakan Metode Economic Order Quantity (Studi Kasus : Pt Ssh)," vol. 1, no. 1, pp. 1–9, 2021.
- [5] M. Singh, A. Verma, A. Parasher, N. Chauhan, and G. Budhiraja, "Implementation of Database Using Python Flask Framework," *International Journal of Engineering and Computer Science*, vol. 8, no. 12, pp. 24890–24893, 2019, doi: 10.18535/ijecs/v8i12.4399.
- [6] H. Kasman, "Perancangan Sistem Informasi Pendataan Dan Transaksi Downline District Pulsa Pekanbaru Berbasis Web," *J. Ilmu Komput. dan Bisnis*, vol. 9, no. 1, pp. 1879–1906, 2018, doi: 10.47927/jikb.v9i1.122.
- [7] F. A. Aslam, H. N. Mohammed, J. Musab, and M. Munir, "International Journal of Advanced Research in Computer Science Available Online at www.ijarcs.info Efficient Way Of Web Development Using Python And Flask," vol. 6, no. 2, pp. 54–57, 2015.
- [8] K. Adawadkar, "Python Programming-Applications and Future," *Int. J. Adv. Eng. Res. Dev.*, vol. 4, no. 04, pp. 1–4, 2017, doi: 10.21090/ijaerd.it032.
- [9] M. Muslih *et al.*, "Implementasi Metode Waterfall Dalam Pembangunan Sistem Informasi Klinik Tiara Bunda Berbasis Web Service," *J. Rekayasa Teknol. Nusa Putra*, vol. 5, no. 2, pp. 20–25, 2019.
- [10] S. Masripah and L. Ramayanti, "Pengujian Black Box Pada Sistem Informasi Penerimaan Siswa Baru Berbasis Web," *fInformation Syst. Educ. Prof.*, vol. 4, no. 1, pp. 1–12, 2019.