

# Implementasi RESTful API dalam Pembuatan Master Data *Planogram* Menggunakan *Framework* Flask dan Metode SDLC (Studi Kasus: PT Sumber Alfaria Trijaya, Tbk)

*RESTful API Implementation in Master Data Planogram Development using the Flask Framework (Case Study: PT Sumber Alfaria Trijaya, Tbk)*

Era Susanti<sup>1</sup>, Evangs Mailoa<sup>2</sup>

<sup>1,2</sup>Program Studi Teknik Informatika Fakultas Teknologi Informasi,  
Universitas Kristen Satya Wacana

E-mail: <sup>1</sup>672016309@student.uksw.edu, <sup>2</sup>evangs.mailoa@uksw.edu

## Abstrak

Salah satu perusahaan ritel yang sedang berkembang dan merupakan salah satu perusahaan ritel terbesar di Indonesia yaitu Alfamart yang dimiliki oleh PT. Sumber Alfaria Trijaya, Tbk. Alfamart harus memiliki strategi pemasaran yang terbaik serta meningkatkan inovasi demi kepuasan para pelanggan agar dapat bertahan dalam persaingan bisnis yang tinggi. Salah satu strategi dalam meningkatkan pemasaran yaitu penataan *display* produk yang ada pada toko yang dikenal dengan *planogram*. *Planogram* merupakan suatu konsep yang digunakan dalam perencanaan penataan dan penempatan produk sesuai dengan kategori tertentu berdasarkan kebiasaan belanja konsumen yang bertujuan meningkatkan penjualan pada ritel. Penelitian ini dilakukan untuk membuat aplikasi master data *planogram* berbasis web menggunakan *framework* Flask dengan bahasa pemrograman *python*. Metode yang digunakan pada penelitian ini adalah RESTful API yaitu implementasi dari web *service* yang bekerja melalui *link* HTTP. Penelitian ini menghasilkan sebuah aplikasi master data berbasis web yang dapat digunakan oleh *user* dalam memasukkan data-data yang dibutuhkan dalam pembuatan *planogram*.

Kata Kunci : RESTful API, *Python Flask*, *Planogram*

## Abstract

*One of developing retail company and is one of the biggest retail companies in Indonesia, namely Alfamart which is owned by PT. Sumber Alfaria Trijaya, Tbk. Alfamart must have the best marketing strategy and increase innovation for the satisfaction of customers in order to survive in high business competition. One strategy to improve marketing is the arrangement of product displays in stores known as planograms. Planogram is a concept that is used in planning the arrangement and placement of products according to certain categories based on consumer spending habits that aim to increase sales at retail. This research was conducted to create a web-based planogram master application using the Flask framework with the python programming language. The method used in this study is the RESTful API, which is the implementation of web services that work through HTTP links. This research produces a web-based master data application that can be used by users in entering data needed in making a planogram.*

Keywords: RESTful API, *Python Flask*, *Planogram*

## 1. PENDAHULUAN

Perkembangan teknologi yang terjadi pada suatu negara dapat pula mempengaruhi perkembangan di bidang lain seperti pertumbuhan ekonomi. Salah satu aspek yang mendukung perkembangan ekonomi saat ini yaitu semakin bertambahnya usaha ritel yang bermunculan. Ritel

merupakan bidang bisnis yang menjual produk dan jasa yang diberi nilai tambah guna memenuhi kebutuhan pribadi, keluarga maupun kelompok dalam jumlah eceran. Ritel menjadi mata rantai terakhir terakhir proses distribusi yang menghubungkan produsen dengan konsumen. Merupakan salah satu indikator ekonomi yang memberikan kontribusi sebesar 19-31% terhadap PDRB (Pendapatan Domestik Regional Bruto) [1].

Salah satu ritel yang sedang berkembang maju di Indonesia yaitu Alfamart. Alfamart merupakan *brand minimarket* yang menyediakan kebutuhan sehari-hari yang dimiliki oleh PT. Sumber Alfaria Trijaya, Tbk. Merupakan salah satu perusahaan ritel terbesar di Indonesia yang berdiri pada tahun 1989 dan memulai bisnis di bidang perdagangan dan distribusi seperti barang-barang konsumsi, kemudian pada tahun 1999 masuk ke sektor ritel [2]. Dalam menghadapi persaingan bisnis yang tinggi tentunya Alfamart harus memiliki strategi pemasaran yang baik dan menarik serta meningkatkan inovasi demi kepuasan para pelanggan [3]. Salah satu cara dalam meningkatkan strategi pemasaran dalam ritel yaitu melalui penataan produk yang ada di toko karena dapat meningkatkan minat beli pelanggan. Implementasi penataan dan *display* produk pada ritel yaitu *planogram*.

*Planogram* merupakan suatu konsep yang digunakan dalam perencanaan penataan dan penempatan produk sesuai dengan kategori tertentu berdasarkan kebiasaan belanja konsumen yang bertujuan meningkatkan penjualan pada ritel [4]. *Planogram* dibuat dengan tujuan memaksimalkan penjualan pada retail.

Penelitian ini dilakukan untuk membuat master data *planogram* pada aplikasi *marketing* berbasis web menggunakan *framework* Flask dengan bahasa pemrograman yang ringan dan banyak dipakai saat ini yaitu *python*.

Penelitian ini menghasilkan sebuah aplikasi master data berbasis web yang dapat digunakan oleh *user* yaitu bagian *marketing* dalam memasukkan data-data yang dibutuhkan dalam pembuatan *planogram*. Data-data yang diinputkan meliputi data jenis lokasi, data type toko, ukuran rak, kelas produk posisi rak, rak non reguler, nama regional dan regional transfer plano. Hasil penelitian ini dapat membantu memudahkan pengguna aplikasi yaitu bagian *marketing* dalam manajemen data yang dibutuhkan saat pembuatan *planogram*.

Berdasarkan latar belakang yang telah diuraikan, maka identifikasi masalah dalam penelitian ini yaitu manajemen data master *planogram* sangat diperlukan untuk membantu dalam memasukkan data yang dibutuhkan dalam pengimplementasian *planogram*.

Berdasarkan latar belakang dan identifikasi masalah diatas, maka rumusan masalah pada penelitian ini adalah bagaimana mengimplementasikan metode RESTful API dalam pembuatan master data *planogram* dalam *Marketing Application System*, bagaimana penggunaan *framework* Flask dalam pembuatan master data *planogram* pada *Marketing Application System*.

Tujuan dari penelitian ini yaitu membangun aplikasi master data *planogram* berbasis web pada PT. Sumber Alfaria Trijaya, Tbk yang lebih ringan dan mempermudah dalam pengelolaan master data *planogram*. Adapun manfaat dalam penelitian ini yaitu membuat sebuah aplikasi master data berbasis web yang lebih ringan dan mudah dalam penggunaannya dan memudahkan *user* yaitu bagian *marketing* dalam manajemen data yang dibutuhkan pada saat pembuatan *planogram*.

Terdapat beberapa penelitian terdahulu yang berkaitan dengan penelitian ini. Penelitian yang dilakukan oleh Haryanto dkk yang berjudul “Pengaruh Peletakan Display Produk Berdasarkan Prinsip 5R dan Planogram terhadap Peningkatan Penjualan: Studi Kasus pada Ritel X di Lamongan” ini bertujuan untuk mengidentifikasi strategi dalam penataan barang atau produk dan meningkatkan penjualan pada Ritel X serta meningkatkan kebiasaan baik bagi karyawan dalam hal penyusunan produk. Hasil penelitian ini berdasarkan konsep 5R dan *planogram*. Pada penelitian ini data-data *planogram* yang dibuat masih menggunakan cara manual atau biasa yaitu menggunakan *pareto chart* [4].

Penelitian yang dilakukan oleh Gilang yang berjudul “Perancangan dan Implementasi Web Service sebagai Media Pertukaran Data pada Aplikasi Permainan” ini membahas tentang penerapan web *service* dalam pertukaran informasi pada platform yang berbeda antara *website* PHP dengan aplikasi android menggunakan Java. Modul yang dibuat pada penelitian ini yaitu modul *register* dan *login*, modul poin dan modul tukar pulsa. Pengujian modul pada penelitian ini menggunakan

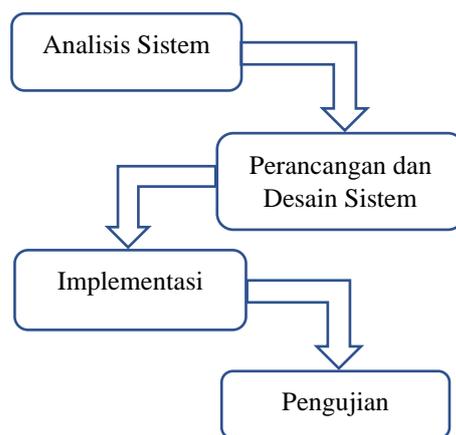
postman. Hasil dari penelitian ini yaitu web *service* dapat diimplementasikan dan dapat berjalan baik sesuai simulasi pada postman [5].

Penelitian yang dilakukan oleh Mukhammad dkk yang berjudul “Analisis dan Perancangan *Representational State Transfer (REST) Web Service* Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan *Yii Framework*” ini menghasilkan REST web *service* sistem informasi akademik pada STT Terpadu Nurul Fikri yang menyediakan layanan bagi sistem lain yang membutuhkan dan menangani pertukaran data pada *platform* yang berbeda. Penelitian ini dibuat menggunakan *Yii Framework 2.0*. Pengujian sistem pada penelitian ini dilakukan menggunakan metode *blackbox* [6].

Keterkaitan penelitian ini dengan penelitian-penelitian sebelumnya yaitu pada penelitian sebelumnya data-data *planogram* masih dibuat secara manual menggunakan *chart*, kemudian pada penelitian lain menggunakan metode web *service* sebagai layanan bagi sistem lain dan pertukaran data pada *platform* yang berbeda, sedangkan pada penelitian ini data-data *planogram* dibuat dalam sebuah aplikasi berbasis web menggunakan sebuah *framework* yaitu Flask yang berbasis bahasa python dan web *service* pada penelitian ini digunakan sebagai implementasi metode membuat data-data *planogram*.

## 2. METODE PENELITIAN

Terdapat beberapa tahapan dalam penelitian ini agar pembuatan aplikasi berbasis web sesuai dengan hasil yang diharapkan. Pada penelitian ini menggunakan metode SDLC (*System Development Life Cycle Waterfall*). Tahapan dalam penelitian ini yaitu sebagai berikut :



**Gambar 1.** Tahapan Metodologi Penelitian

Berikut ini merupakan penjelasan tahapan penelitian pada gambar 1 :

### a. Analisis Sistem

Tahap ini merupakan tahap analisis kebutuhan sistem yang digunakan dalam penelitian ini. Analisis yang dilakukan yaitu bagaimana sistem yang berjalan pada penelitian ini dan apa saja yang dapat dilakukan oleh aktor dalam penelitian ini.

### b. Perancangan dan Desain Sistem

Pada tahap ini akan dilakukan perancangan sistem berdasarkan analisis kebutuhan sistem yang telah dilakukan. Kemudian sistem yang telah dirancang dan dianalisis akan diimplementasikan menggunakan *tools* yaitu bahasa *python* dengan *framework* Flask dan *database* PostgreSQL. Tahap ini juga merupakan tahap pembuatan pemodelan sistem menggunakan UML (*Unified Modeling Language*). Pembuatan model sistem menghasilkan *Use Case Diagram* dan *Activity Diagram*.

c. Implementasi

Tahap ini merupakan tahap pengimplementasian perancangan sistem yang telah dibuat. Tahap ini merupakan proses paling penting yang ada dalam metodologi penelitian ini karena inti dari pembuatan sistem ada pada tahap ini. Pembuatan sistem berdasarkan perancangan dan desain sistem yang telah dibuat pada tahap sebelumnya. Dalam implementasi penelitian ini menggunakan metode RESTful API yaitu implementasi dari web *service* yang bekerja melalui *link* HTTP [7]. Fungsi HTTP yang dapat digunakan pada REST yaitu fungsi *get*, *post*, *put* dan *delete* [7]. Pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. Sistem web *service* yang menerapkan prinsip-prinsip REST disebut RESTful. Cara kerja dari RESTful yaitu *client* mengirim sebuah permintaan melalui HTTP *Request* dan server menanggapi permintaan *client* melalui HTTP *Response*. Terdapat 2 bagian pesan untuk berkomunikasi dengan server yaitu pesan *header* dan pesan *body*. HTTP header merupakan catatan kecil setiap transaksi pada HTTP sedangkan HTTP *body* yaitu data yang akan dikirimkan [8].

d. Pengujian

Tahap ini merupakan tahap pengujian terhadap aplikasi yang telah dibuat, apakah sesuai dengan rancangan awal sistem dan memastikan semua fungsi-fungsi yang terdapat didalamnya dapat berjalan dengan baik. Pengujian merupakan tahap yang penting karena sebelum aplikasi akan digunakan oleh *user* atau pengguna harus dipastikan agar aplikasi dapat digunakan dengan baik dan tidak terdapat kesalahan-kesalahan atau *bug* didalamnya. Tahap ini juga menentukan kepuasan dari *user* ketika menggunakan aplikasi yang telah dibuat.

### 3. HASIL DAN PEMBAHASAN

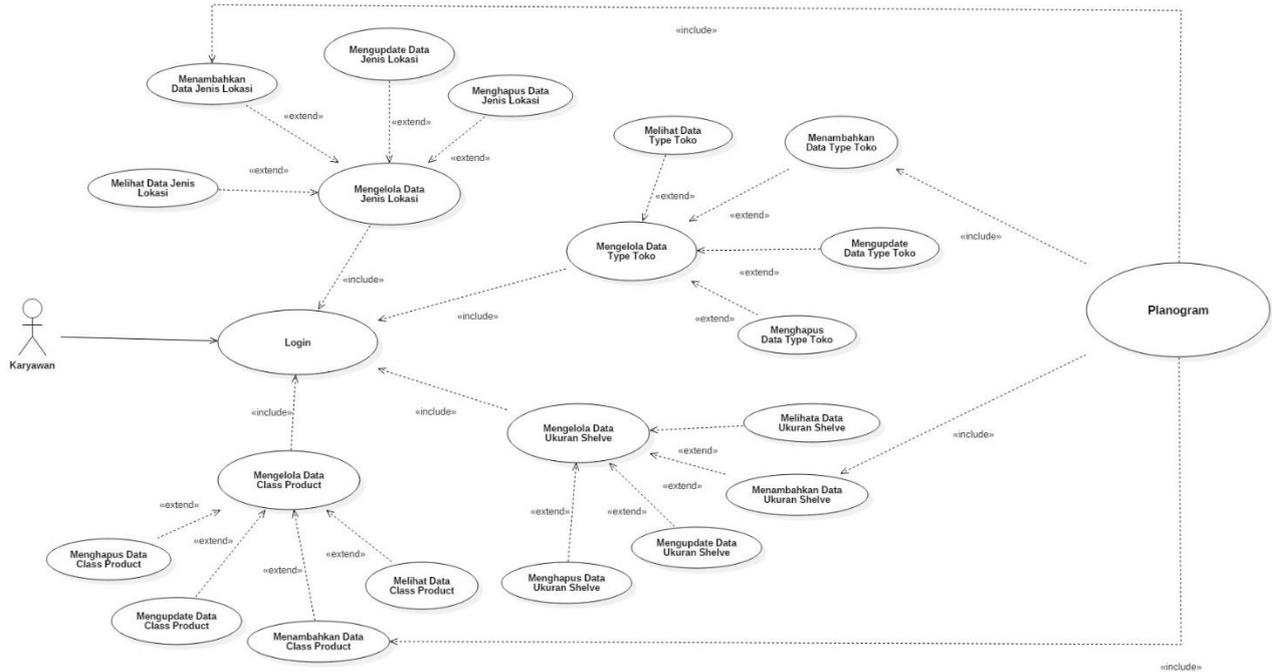
Berikut ini merupakan hasil penelitian berdasarkan perancangan sistem yang telah dibuat :

a. Analisis Sistem

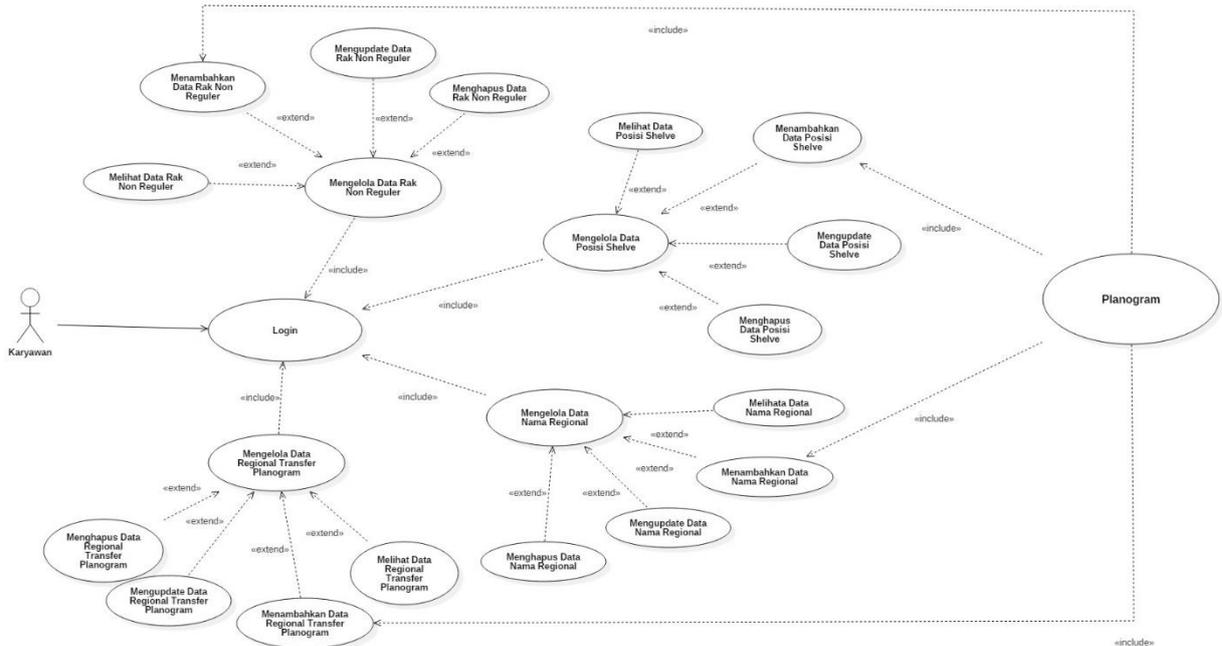
Sistem yang akan berjalan pada penelitian ini yaitu pengguna dapat melakukan fungsi CRUD (*Create, Read, Update, Delete*) dalam pengolahan data pada aplikasi ini. Fungsi tersebut dapat dilakukan pada modul-modul yang tersedia untuk pembuatan *planogram* yaitu modul Jenis Lokasi, *Type* Toko, Ukuran *Shelve*, *Class Product*, Rak Non Reguler, Nama Regional, Regional Transfer Plano dan Planogram. Pengguna dapat melakukan *method get, post, put* dan *delete* pada masing-masing modul tersebut. Aktor yaitu pihak yang terlibat dalam sistem aplikasi [9] pada penelitian ini adalah user, dimana dalam hal ini yang bertindak sebagai user yaitu karyawan.

b. Perancangan dan Desain Sistem

Berikut ini merupakan hasil pembuatan desain pada penelitian ini.



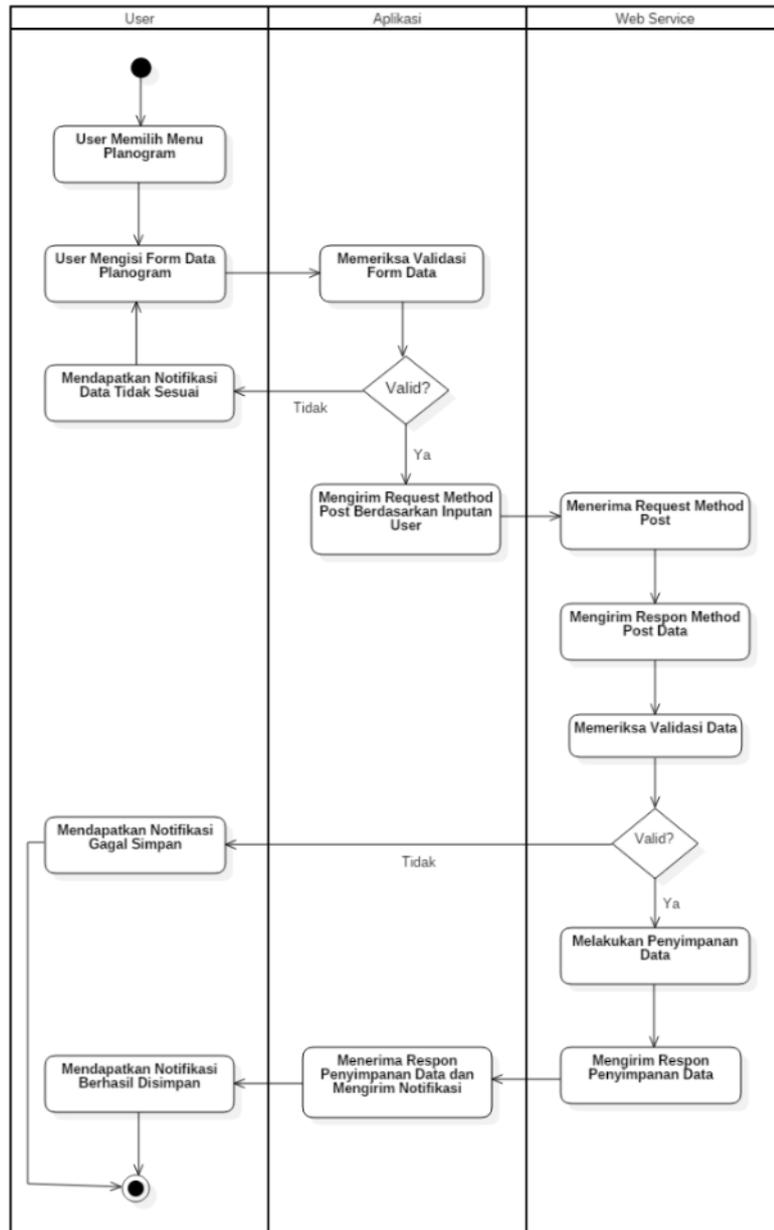
Gambar 2. Use Case Diagram Data Jenis Lokasi, Type Toko, Ukuran Shelve, Class Product



Gambar 3. Use Case Diagram Data Rak Non Reguler, Posisi Shelve, Nama Regional, Regional Transfer Plano

Gambar 2 dan 3 merupakan *Use Case Diagram* yang ada pada penelitian ini. Pada diagram tersebut karyawan berperan sebagai aktor yang akan menjalankan aplikasi. Pada gambar tersebut karyawan dapat menjalankan operasi pada beberapa modul. Sebelum menjalankan modul-modul tersebut karyawan harus melakukan *login* terlebih dahulu. Fungsi utama yang dapat dijalankan oleh karyawan yaitu pada *Planogram*, namun sebelum menjalankan fungsi pada *planogram* karyawan harus melakukan operasi pada modul-modul data master. Modul-modul data master yang dapat dijalankan oleh karyawan diantaranya yaitu mengelola data jenis lokasi, mengelola data *type* toko, mengelola data ukuran *shelve*, mengelola data *class product*, mengelola data rak non reguler, mengelola data posisi

*shelve*, mengelola data nama regional dan mengelola data regional transfer plano. Dalam modul data master tersebut karyawan dapat melakukan fungsi pada masing-masing modul data master seperti melihat, menambahkan, mengedit dan menghapus data. Kemudian setelah melakukan operasi pada modul data master karyawan dapat menjalankan modul *planogram* dimana data *planogram* ini didapat dari modul-modul data master.



Gambar 3. Activity Diagram

Gambar 3 merupakan *Activity Diagram* pada penelitian ini yaitu pada menu *Planogram* proses *insert* data. Alur dari diagram ini yaitu langkah awal user memilih menu *Planogram* kemudian untuk menginsert data langkah selanjutnya *user* mengisi *form* data yang diperlukan untuk menginsert data. Setelah *user* mengisi *form*, aplikasi akan memeriksa validasi inputan *form* dari *user* jika input data valid maka aplikasi akan mengirim *request method post* berdasarkan inputan *user* tersebut, jika input data tidak valid maka *user* akan mendapatkan notifikasi jika data yang diisi tersebut tidak sesuai dan

*user* harus mengisi ulang *form* tersebut sampai data dinyatakan valid. Setelah aplikasi mengirim *request method post* maka *web service* akan menerima *request* tersebut dan mengirim respon *method post*, kemudian *web service* akan memeriksa validasi data yang akan diinsert tersebut jika data tidak valid maka *user* akan mendapatkan notifikasi gagal menyimpan data, jika data valid maka *web service* akan melakukan penyimpanan data dan mengirim respon penyimpanan data. Setelah *web service* mengirim respon, aplikasi akan menerima respon data dan mengirim notifikasi, setelah itu *user* akan mendapatkan notifikasi data berhasil simpan.

c. Implementasi

Berdasarkan hasil perancangan sistem yang telah dibuat dan diimplementasikan menggunakan bahasa *python* dan *framework flask*, hasil penelitian ini yaitu aplikasi master data *planogram* berbasis web. Sistem ini dibangun menggunakan bahasa pemrograman *python* dengan *framework flask* dan *database postgresql*, sistem ini juga menggunakan metode *restful API*. Berikut ini merupakan hasil dari penelitian ini.

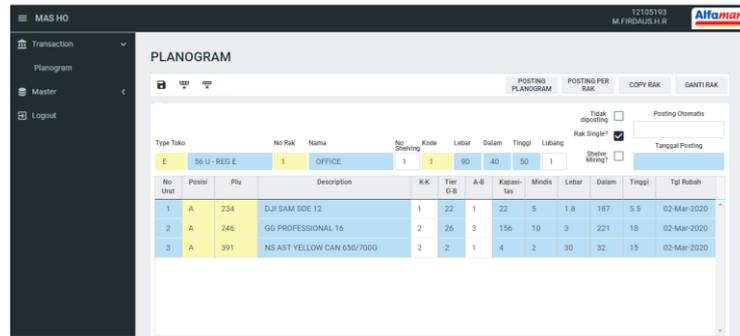
Gambar 4 Halaman Login

Gambar 4 merupakan halaman *login*. Pada saat *login*, *user* harus memasukkan absen atau *user* dan *password*. Proses *login* ini yaitu ketika *user* memasukkan absen dan tekan enter maka jika data tersebut sesuai dengan *database*, pada kolom nama dan *store* secara otomatis terisi data sesuai inputan *user* tersebut, kemudian *user* memasukkan *password* dan klik tombol *login*. Jika data tersebut tidak sesuai dengan *database* maka akan muncul *error handling* dan *user* harus memasukkan kembali data yang benar. Jika *login* berhasil maka *user* akan masuk ke halaman utama web.

Jenis	Nama Lokasi	Tgl Rubah	Action
A	REGULER A	06-Feb-2020	✓
B	REG B	15-Feb-2020	✓
E	REG E	16-Feb-2020	✓
F	ASD	15-Feb-2020	✓
H	REG	10-Feb-2020	✓
I	REGULER I	20-Feb-2020	✓
J	ASD	15-Jan-2020	✓

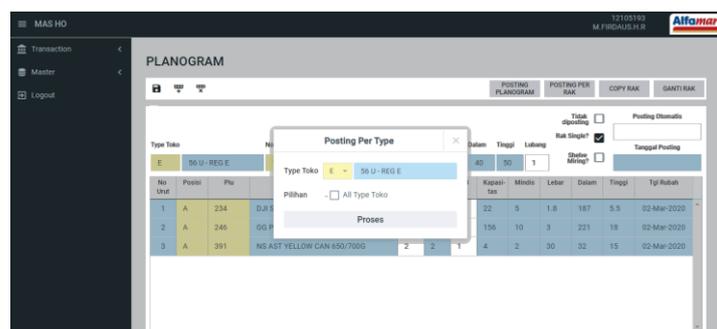
Gambar 5 Halaman Menu Jenis Lokasi

Gambar 5 merupakan halaman menu master jenis lokasi. Dalam menu ini terdapat jenis lokasi serta nama lokasi. Pada menu ini user dapat menjalankan fungsi lihat data, tambah data, edit data dan hapus data. Pada master jenis lokasi ini user akan menginputkan jenis lokasi dan nama lokasi yang akan digunakan untuk *planogram*.



Gambar 6 Halaman Utama Planogram

Gambar 6 merupakan halaman utama *planogram*. Pada menu *planogram* ini user dapat menjalankan fungsi lihat data, tambah data, edit data dan hapus data. Jika user ingin melihat data *planogram* maka user harus memasukkan data tipe toko, nomor rak dan nomor shelve pada *form* yang ada dalam menu ini. Jika data yang telah dimasukkan tersebut sesuai dengan *database* maka data *planogram* akan muncul sesuai dengan tipe toko, nomor rak dan nomor shelve tersebut. Jika user ingin menambahkan data *planogram* maka user harus menambahkan data master karena data-data *planogram* ini diambil dari data master.



Gambar 7 Halaman Posting Planogram

Gambar 7 merupakan halaman *posting planogram* yang terdapat pada menu *planogram*. Fungsi dari *posting planogram* ini yaitu menyalin data dari tabel *head\_temporary* dan *det\_temporary* ke tabel *head\_t* dan *det\_t*. Jadi sebelum *planogram* diposting, data-data *planogram* disimpan pada tabel *temporary*. User dapat memilih data tertentu yang ingin diposting yaitu dengan memasukkan satu tipe toko, tetapi jika user ingin *posting* semua data *planogram* maka pilih *all type* toko. Data *planogram* yang diposting berdasarkan tipe toko yang telah diinput oleh user.

**Kode Program 1** Controller Fungsi Get Data pada Modul Jenis Lokasi

```

1 class Jenis_Lokasi(Resource):
2     def __init__(self):
3         self.skema_jenislokasi = SkemaJenisLokasi()
4         self.skema_jenisLokasis = SkemaJenisLokasi(many=True)
5         self.skema_detailokasis = SkemaDetailCabang(many=True)
6         self.skema_aksesuser = SkemaAksesUser()
7
8     def get(self):
9         jenis_lokasi = Lokasi.query.filter_by(kode_cabang='A000')
10        hasil = skema_jenisLokasis.dump(jenis_lokasi)
11        return jsonify(hasil)
    
```

Kode program 1 merupakan fungsi untuk menampilkan data pada modul Jenis Lokasi. Pada saat menampilkan data pada modul ini menggunakan skema dan menggunakan *dump* untuk menampilkan skema tersebut. Setelah proses *dump*, hasil tersebut diubah ke bentuk *json* dengan *code*

*jsonify()*. Hasil implementasi Restfull API dapat dilihat pada fungsi *get* pada gambar tersebut. Fungsi tersebut nantinya akan dilempar ke tampilan web melalui *request* HTTP.

**Kode Program 2** *Controller* Fungsi *Get* Data Posisi pada Modul Planogram

```

1 class getPss(Resource):
2     def get(self):
3         listpss = []
4         pss = PssShelve.query.all()
5         for a in pss:
6             l = {}
7             l['id'] = a.kd_pss
8             l['value'] = a.kd_pss
9             listpss.append(l)
10        return jsonify(listpss)

```

Kode program 2 berfungsi untuk menampilkan data posisi pada modul Planogram. Dalam fungsi ini terdapat *list* kosong untuk menampung data posisi yang akan ditampilkan. Kemudian setelah dilakukan *query* data maka data tersebut ditambahkan pada *list* dan hasilnya diubah ke bentuk *json*. Hasil implementasi Restfull API dapat dilihat pada fungsi *get* pada gambar tersebut. Fungsi tersebut nantinya akan dilempar ke tampilan web melalui *request* HTTP.

d. Pengujian

Pengujian sistem pada aplikasi ini dilakukan menggunakan metode *black box testing* dan *user acceptance testing*.

1. *Black box Testing*

*Black box testing* merupakan metode pengujian program yang lebih menekankan pada fungsional program yang diuji dan pemeriksaan proses *input* dan data *output* pada program [10]. Berikut ini adalah hasil pengujian yang telah dilakukan dengan metode *black box* pada penelitian ini.

**Tabel 2.** Pengujian pada Halaman *Login*

No.	Fungsi	Inputan	Output yang Diharapkan	Output	Hasil
1	<i>Login</i>	Username dan Password tidak diisi	Muncul pesan kesalahan	Muncul pesan “Username dan Password harus diisi!”	Valid
2	<i>Login</i>	Username diisi data yang tidak sesuai	Muncul pesan kesalahan	Muncul pesan “Data tidak valid!”	Valid
3	<i>Login</i>	Password diisi data yang tidak sesuai	Muncul pesan kesalahan	Muncul pesan “Username atau Password salah!”	Valid
4	<i>Login</i>	Username dan Password diisi data yang sesuai	Berhasil <i>login</i>	Berhasil <i>login</i> dan masuk ke halaman utama	Valid

**Tabel 3.** Pengujian pada Halaman Master Data Jenis Lokasi

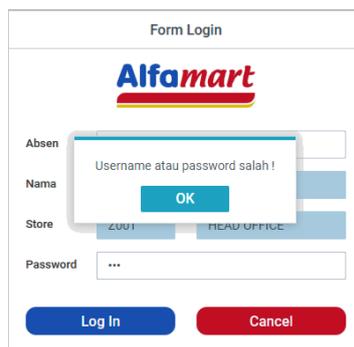
No.	Fungsi	Inputan	Output yang Diharapkan	Output	Hasil
1	<i>Insert</i>	Salah satu maupun semua <i>field</i> pada <i>form</i> tidak diisi	Muncul pesan kesalahan	Muncul pesan “Data Kosong!”	Valid
2	<i>Insert</i>	Mengisi <i>form</i> dengan data jenis lokasi yang sudah ada dalam <i>database</i>	Muncul pesan duplikat data	Muncul pesan “Data Jenis Lokasi sudah ada!”	Valid

3	Insert	Form diisi data yang sesuai dengan ketentuan	Berhasil menambahkan data	Muncul pesan “Data berhasil diinsert!” dan data berhasil ditambahkan	Valid
---	--------	--	---------------------------	--	-------

**Tabel 4.** Pengujian pada Halaman Planogram

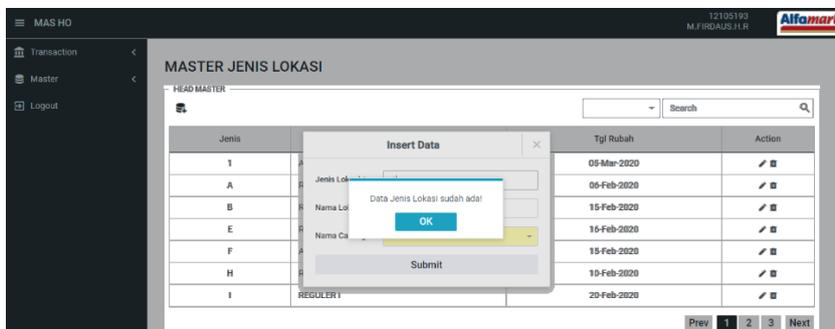
No.	Fungsi	Inputan	Output yang Diharapkan	Output	Hasil
1	Lihat Data	Salah satu <i>field</i> tidak diisi	Muncul pesan kesalahan	Muncul keterangan “Tidak ada data”	Valid
2	Lihat Data	Data yang diisi tidak sesuai dengan <i>database</i>	Muncul pesan kesalahan	Muncul keterangan “Tidak ada data”	Valid
4	Insert	Data diisi sesuai dengan ketentuan	Berhasil menambahkan data	Muncul pesan “Data berhasil disimpan!” dan data berhasil ditambahkan	Valid

Berikut ini merupakan tampilan pengujian yang dilakukan menggunakan metode *Black Box*.



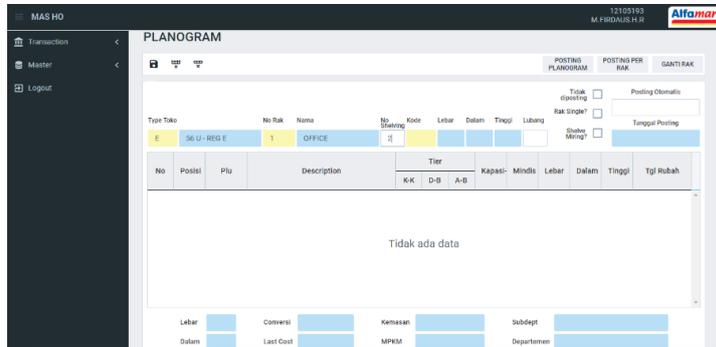
**Gambar 9** Tampilan Pengujian *Login*

Gambar 9 merupakan tampilan pengujian pada halaman *login* ketika data *password* diisi dengan data yang tidak sesuai. *Output* dari pengujian tersebut yaitu muncul pesan seperti pada gambar diatas dan *output* sesuai dengan yang diharapkan.



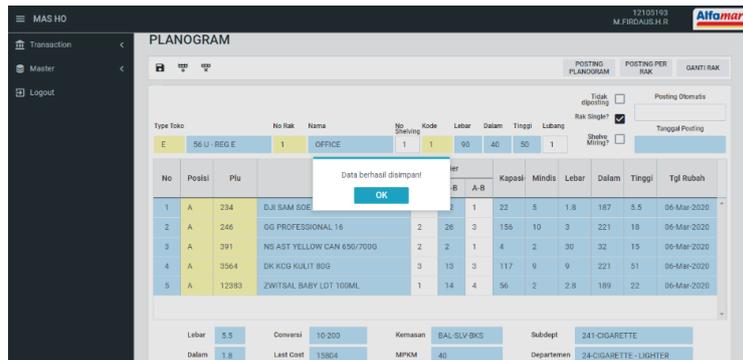
**Gambar 10** Tampilan Pengujian Master Jenis Lokasi

Gambar 10 merupakan tampilan pengujian pada halaman data master jenis lokasi pada fungsi *insert* data ketika data jenis lokasi diisi dengan data yang sudah ada dalam *database*. *Output* dari pengujian tersebut yaitu muncul pesan seperti pada gambar diatas dan *output* sesuai dengan yang diharapkan.



Gambar 11 Tampilan Pengujian *Planogram*

Gambar 11 merupakan tampilan pengujian pada halaman *planogram* pada fungsi lihat data ketika data yang diisi tidak sesuai dengan *database*. *Output* dari pengujian tersebut yaitu muncul keterangan seperti pada gambar diatas dan *output* sesuai dengan yang diharapkan.



Gambar 12 Tampilan Pengujian *Planogram*

Gambar 12 merupakan tampilan pengujian pada halaman *planogram* pada fungsi *insert* data ketika data yang diisi sesuai dengan *database*. *Output* dari pengujian tersebut yaitu muncul data sesuai dengan inputan tersebut dan *output* sesuai dengan yang diharapkan.

## 2. User Acceptacne Testing (UAT)

*User acceptacne testing* merupakan pengujian yang melibatkan user untuk mengetahui apakah sistem yang telah dibuat sudah sesuai dengan kebutuhan maupun berfungsi sesuai dengan yang diharapkan. User yang terlibat dalam pengujian UAT ini yaitu 2 *Senior Programmer Analyst*, 3 *Quality Assurance* dan 15 karyawan bagian marketing. Hasil pengujian dapat dilihat pada tabel 5.

Tabel 5. Hasil Pengujian dengan *User Acceptance Testing*

No	Pertanyaan	Jawaban					Jumlah	Indeks
		Ax5	Ax4	Ax3	Ax2	Ax1		
1	Apakah aplikasi yang telah dibuat dapat digunakan dengan	14	6				94	94%

	mudah dan ringan oleh user?				
2.	Apakah aplikasi yang telah dibuat sudah sesuai dengan kebutuhan yang telah ditentukan?	15	5	95	95%
3.	Apakah aplikasi yang telah dibuat dapat membantu memudahkan pekerjaan karyawan yang bersangkutan?	14	6	94	94%
4.	Apakah aplikasi yang telah dibuat dapat mempercepat pekerjaan pengguna dalam mengelola data planogram?	13	7	93	93%
	Total	56	24	376	94%

Berikut ini merupakan penjelasan pada tabel 5. Pada kolom jawaban Ax5 merupakan sangat setuju, Ax4 merupakan setuju, Ax3 merupakan cukup, Ax2 merupakan tidak setuju dan Ax1 merupakan sangat tidak setuju. Kolom jumlah merupakan total perhitungan nilai jawaban. Kolom indeks merupakan presentase jumlah / total *sample user* \* 100.

Berdasarkan Tabel 5 dapat dihitung nilai rata-rata pada jawaban pertama adalah 18,8 maka persentasenya adalah 94%, nilai rata-rata pada jawaban kedua adalah 19 maka persentasenya adalah 95%, nilai rata-rata pada jawaban ketiga adalah 18,8 maka persentasenya adalah 94% dan nilai rata-rata pada jawaban keempat adalah 18,6 maka persentasenya adalah 93%. Berdasarkan perhitungan *User Acceptance Test*, indeks yang didapatkan untuk pengujian *User Acceptance Test* adalah sebesar 94%. Berdasarkan analisa pengujian *User Acceptance Test* yang telah dilakukan didapatkan bahwa sistem Master Data *Planogram* berbasis web ini dapat membantu menyelesaikan permasalahan yang ada, fungsi-fungsi yang terdapat di dalam sistem tersebut juga sudah bekerja dengan baik, Master Data *Planogram* berbasis web ini juga memiliki kemudahan dalam penggunaannya.

#### 4. KESIMPULAN DAN SARAN

Hasil penelitian ini yaitu aplikasi master data *planogram* berbasis web. Metode yang digunakan pada penelitian ini adalah RESTful API yaitu implementasi dari web *service* yang bekerja melalui *link* HTTP. Bahasa yang digunakan pada aplikasi ini merupakan bahasa yang sedang banyak dipakai pada saat ini yaitu *python* dengan menggunakan *framework* Flask. Aplikasi ini bertujuan untuk memudahkan *user* untuk memasukkan data-data master yang akan dipakai pada saat pembuatan *planogram*.

Saran dari penulis untuk pengembangan aplikasi ini yaitu:

- Penggunaan *python* Flask bisa dikembangkan lebih dalam dan dapat ditambahkan *JSON Web Token* untuk melindungi API agar tidak mudah diakses oleh orang yang tidak berkepentingan
- Mungkin kedepannya aplikasi ini dapat dikembangkan berbasis *mobile* dan dibuat lebih dinamis serta efisien dengan versi *mobile*.

DAFTAR PUSTAKA

- [1] D. Hikmawati and C. Nuryakin, “Keberadaan Ritel Modern dan Dampaknya terhadap Pasar Tradisional di DKI Jakarta,” *J. Ekon. dan Pembang. Indones.*, vol. 17, no. 2, p. 195, 2017, doi: 10.21002/jepi.v17i2.768.
- [2] “Identitas Perusahaan Alfamart,” *Alfamartku*. [Online]. Available: <http://corporate.alfamartku.com/identitas-perusahaan>. [Accessed: 08-Oct-2019].
- [3] S. S. Widodo, “Pengaruh Discount, Display Produk dan Lokasi Toko terhadap Keputusan Pembelian di Toko D’Sport Kediri,” *Simki-Economic*, vol. 01, no. 01, 2017.
- [4] R. Haryanto, T. Octavia, and R. Sugianto, “Pengaruh Peletakan Display Produk Berdasarkan Prinsip 5R dan Planogram terhadap Peningkatan Penjualan : Studi Kasus pada Ritel X di Lamongan,” *J. Titra*, vol. 6, no. 2, pp. 385–392, 2018.
- [5] G. A. Priyatna Zaman, “Perancangan Dan Implementasi Web Service sebagai Media Pertukaran Data pada Aplikasi Permainan,” *J. Inform.*, vol. 11, no. 2, pp. 22–30, 2017.
- [6] M. Agus, A. Teknik, S. T. T. Terpadu, and N. Fikri, “Analisis dan Perancangan Representational State Transfer (REST) Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan Yii Framework,” *J. Teknol. Terpadu*, vol. 2, no. 2, 2016.
- [7] F. N. Rofiq and A. Susanto, “Implementasi RESTful Web Service untuk Sistem Penghitungan Suara Secara Cepat pada Pilkada,” *EKSPLORA Inform.*, vol. 6, no. 2, pp. 159–168, 2017.
- [8] A. Rahmatulloh, H. Sulastrri, and R. Nugroho, “Keamanan RESTful Web Service Menggunakan JSON Web Token ( JWT ) HMAC SHA-512,” *JNTETI*, vol. 7, no. 2, p. 132, 2018.
- [9] F. Esti, M. Rizal, E. Hari, D. Rosal, M. Setiadi, and C. Atika, “Implementasi E-arsip Untuk Penyimpanan Dokumen Digital Pada PT BPD Jateng (Bank Jateng),” *Tecno.COM*, vol. 18, no. 4, pp. 299–311, 2019.
- [10] M. Komarudin, “Pengujian perangkat Lunak metode Black box berbasis partitions pada aplikasi sistem informasi di sekolah,” *J. Mikrotik*, vol. 06, no. 3, pp. 02–16, 2016, doi: <http://dx.doi.org/10.1016/j.brat.2012.09.004>.