

## PEMBANGUNAN APLIKASI *VIRTUAL INVENTORY SYSTEM* (VIS) BERBASIS WEB MENGGUNAKAN *FLASK FRAMEWORK* (STUDI KASUS: PT XYZ)

Eben Haezer Yena Kristianto<sup>1</sup>, Nina Setiyawati<sup>2</sup>

<sup>1,2</sup> Universitas Kristen Satya Wacana  
672018076@student.uksw.edu

### ABSTRAK

Dalam bisnis di bidang *retail* PT. XYZ mencari cara dan terus berinovasi agar dapat mengembangkan bisnis distribusi *retail* dengan pesat. Untuk mengakomodasi dan memperlancar proses transaksi tersebut, PT XYZ memerlukan beragam sistem serta aplikasi yang dapat mendukung dan mempermudah proses transaksi tersebut. Oleh karena itu PT XYZ membangun sebuah sistem yang bernama *Virtual Inventory System* (VIS) menggunakan *Oracle form*. Beriringnya waktu, didapatkan bahwa aplikasi tersebut memiliki kekurangan seperti biaya pemeliharaan aplikasi semakin tinggi, teknologi yang digunakan sudah tertinggal, hanya dapat berjalan pada satu *platform* saja. Oleh karena itu pada penelitian ini dilakukan pengembangan ulang aplikasi *Virtual Inventory System* menggunakan *Flask Framework*. Keunggulan menggunakan *Flask Framework* dalam membangun web adalah dokumentasi yang banyak dan terstruktur, ringan ketika mengeksekusi program karena mempunyai *core* yang sederhana dan desain modular, serta mudah digunakan dan dipahami karena fiturnya yang sederhana. Aplikasi ini berbasis web agar bisa diakses di segala *platform* (*multiplatform*) sehingga lebih mempermudah *user* dalam mengakses aplikasi VIS. Penelitian ini menghasilkan modul *Pre Order Manual* (PO *Manual*) di dalam aplikasi VIS yang nantinya akan digunakan pada *Head Office* untuk mengatur *supplier*, harga barang dan mencetak faktur hasil dari transaksi barang yang sudah dilakukan.

**Keyword:** *Virtual Inventory System, Flask Framework, PostgreSQL, Python, Oracle Form, Retail.*

### 1. PENDAHULUAN

Revolusi Industri 4.0 mengakibatkan cara berpikir manusia dan cara berinteraksi manusia satu dengan lainnya berubah. Era ini mengubah pola aktivitas manusia dalam berbagai bidang, tidak hanya dalam bidang teknologi saja, namun juga bidang yang lain seperti ekonomi, sosial, dan politik [1]. Begitu juga di bidang bisnis distribusi *retail*, banyak orang berlomba-lomba dalam mencari cara dan terus berinovasi agar dapat mengembangkan bisnis distribusi *retail* dengan pesat.

PT XYZ merupakan salah satu perusahaan *retail* di Indonesia yang memiliki lebih dari 14.300 toko dan 32 cabang. Sebagai perusahaan *retail* terbesar PT XYZ dapat menghasilkan jutaan transaksi *customer* per harinya. Untuk mengakomodasi dan memperlancar proses transaksi tersebut, PT XYZ memerlukan beragam sistem serta aplikasi yang dapat mendukung dan mempermudah proses distribusi kebutuhan dari *supplier*, gudang, toko, maupun konsumen [2][3][4][5]. Oleh karena itu, PT XYZ membangun aplikasi *Virtual Inventory System* (VIS) menggunakan *Oracle form*.

Beriringnya waktu, didapatkan bahwa aplikasi tersebut memiliki kekurangan seperti biaya pemeliharaan aplikasi semakin tinggi, teknologi yang digunakan sudah tertinggal, hanya dapat berjalan pada satu *platform* saja. Oleh karena itu pada penelitian ini dilakukan pengembangan ulang aplikasi *Virtual Inventory System* menggunakan *Flask Framework*. Aplikasi VIS digunakan untuk melakukan transaksi *setting* harga, *supplier*, *Price Look up Unit* (PLU), sedangkan pada modul *Pre*

*Order Manual* atau yang disebut PO *Manual* sendiri berfungsi untuk mengatur harga voucher pulsa dan mencetak faktur. PO *Manual* nantinya dipakai oleh *Head Office* untuk mengatur *supplier*, harga barang dan mencetak faktur hasil dari transaksi barang yang sudah dilakukan. Aplikasi ini dibuat dengan mengimplementasikan *Flask Framework*. *Flask Framework* tergolong ke dalam *microframework* karena tidak membutuhkan *tools* atau *library* tertentu dalam penggunaannya [6]. *Flask Framework* dan *Django Framework* merupakan *framework* dari *python* sebagai pengembangan web dan aplikasi. *Flask* adalah *framework* ringan yang cocok untuk proyek kecil dan mempermudah pengembang dalam menggunakannya karena fiturnya yang sederhana. Di sisi lain, *Django* adalah *framework* yang menyediakan pengembang dengan segala yang mereka butuhkan untuk menangani proyek besar, kompleks dan sangat fungsional [7]. Oleh karena itu pada pembangunan aplikasi VIS ini penulis menggunakan *Flask Framework* yang mempunyai banyak dokumentasi sehingga lebih mempermudah dalam mencari referensi. Pada aplikasi ini juga menggunakan *bootstrap* sebagai bagian antar muka guna mempercepat dan mempermudah pengembangan web [8], dan *PostgreSQL* sebagai *database*. Dengan menggunakan sistem yang terkomputerisasi akan lebih memudahkan pengguna dalam proses pendataan barang masuk dan keluar yang sudah tersistem sehingga lebih efektif dan efisien dalam pencarian data yang diperlukan, sehingga tidak akan terjadi lagi duplikasi data, serta dapat

memper memudahkan *user* dalam melakukan transaksi di berbagai *platform* [9]. Diharapkan dengan adanya aplikasi web ini, PT XYZ dapat mengurangi biaya pemeliharaan aplikasi, mempermudah *user* dalam melakukan transaksi di berbagai *platform* dan dapat mengikuti perkembangan teknologi yang terbaru. Penelitian ini akan menghasilkan modul PO *Manual* yang nantinya akan digunakan pada *Head Office*.

## 2. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

*Flask Framework* merupakan *framework* yang mempunyai sifat *simplicity*, di mana sifat ini mempermudah pembangunan aplikasi serta menghasilkan aplikasi yang optimal sesuai dengan kebutuhan. Hal ini dapat dilihat pada penelitian-penelitian terdahulu seperti:

Dinda Fitri Ningtyas (2021) Penelitian yang berjudul “Implementasi *Flask Framework* pada Pembangunan Aplikasi *Purchasing Approval Request*” menjelaskan tentang lama waktu dalam pemberian persetujuan untuk melakukan pengajuan, karyawan yang akan mengisi *form* dengan detail mengenai barang, tujuan, dana yang diperlukan, proses pembayaran, serta daftar nama atasan penyetuju dalam suatu proses pengajuan pembelian, sehingga proses pengajuan menjadi tidak efektifitas waktu serta memperlambat karyawan baik dalam melakukan *purchasing request* maupun melakukan *approval*. Aplikasi ini menggunakan *Flask Framework* untuk membangunnya dan *PostgreSQL* sebagai *databasenya* [10].

Brian Pratama Putra (2020). Penelitian yang berjudul “Implementasi API *Master Store* Menggunakan *Flask*, *REST* dan *ORM* di PT XYZ” menjelaskan tentang mengubah arsitektur *monolithic* menjadi *microservices*, yang dimana sistem *Master Store* agar dapat digunakan secara terus menerus oleh aplikasi lain dan dapat mempermudah *developer* dalam membangun aplikasi baru menggunakan database *Master Store*. Sistem ini dibangun menggunakan *Flask Framework* dikarenakan *Flask Framework* menyediakan semua *libraries* yang dibutuhkan dan penggunaan *Flask Framework* lebih sederhana jika dibandingkan oleh *framework* lain, sehingga *Flask Framework* akan terasa lebih ringan.[11].

Ronaldo Kristoforus Ngantung (2021). Penelitian yang berjudul “Model Pengembangan Sistem Informasi Akademik Berbasis *User Centered Design* Menerapkan *Framework Flask Python*” menjelaskan tentang SMK Negeri 1 Kota Bitung yang memerlukan adanya suatu sistem informasi akademik untuk kegiatan akademik siswa maupun guru oleh karena itu penulis melakukan pembangunan aplikasi sistem informasi akademik berbasis web yang dibangun menggunakan *Flask Framework*. Aplikasi ini dibangun menggunakan *Flask Framework* dikarenakan Sebagian besar

komponen umum tidak terpasang secara bawaan di *Flask*, hal ini terjadi karena komponen dari *Flask* sudah disediakan oleh pihak ketiga yang bersifat ekstensi yang membuat komponen tersebut seakan diimplementasikan oleh *Flask* sendiri[12].

Rahardian Irsyad (2018) Penelitian yang berjudul “Penggunaan *Python Web Framework Flask* Untuk Pemula” menjelaskan tentang menggambarkan secara umum mengenai penggunaan *Python Web Framework* Bernama *Flask* untuk pemula. Dimana *Flask* sendiri memiliki fleksibilitas yang tinggi dalam penggunaannya dikarenakan memiliki *core* yang sederhana Keunggulan *Flask* lainnya yaitu dapat menangani fungsi *HTTP request* dengan mudah dan mudah untuk dipasang serta di-*deploy* untuk produksi [13].

### 2.2. Kajian Teori

#### a. Python

*Python* adalah bahasa pemrograman yang bersifat *open source*, interpretatif multiguna dan didukung dengan fungsionalitas pustaka standar yang besar, *Python* dirancang untuk dapat memberikan kemudahan bagi *programmer* baik dari segi efisiensi waktu, kemudahan dalam pengembangan, serta kompatibilitas dengan sistem.

Sintaks *Python* yang sederhana dan mudah dipelajari dan karenanya mengurangi biaya pemeliharaan program. *Python* mendukung modul dan paket, yang mendorong modularitas program [14].

*Python* merupakan salah satu bahasa pemrograman tingkat tinggi (*High Level Language*) dan bisa digunakan untuk membuat aplikasi *standalone* (berdiri sendiri) dan pemrograman *script* (*Scripting programming*) [15].

#### b. Flask Framework

*Flask* sebuah web *framework* yang ditulis dengan bahasa *Python* dan tergolong sebagai jenis *microframework*. *Flask Framework* berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan *Flask Framework* dan bahasa pemrograman *Python*, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur perilaku suatu web dengan lebih mudah. *Flask Framework* dirancang dan didesain untuk memulai dengan cepat dan mudah (berfungsi untuk meminimalisir waktu proses), dengan kemampuan ini dapat meningkatkan kemampuan aplikasi yang kompleks.

Tergolong ke dalam *microframework* menjadikan sebagian besar fungsi dan komponen umum seperti validasi *form*, *database*, dan sebagainya tidak terpasang secara default di dalam *Flask Framework*. Ini dimaksudkan agar hanya komponen yang dibutuhkan saja yang terpasang, sedangkan komponen yang tidak dibutuhkan tidak perlu dipasang agar tidak membebani performa *framework* [13].

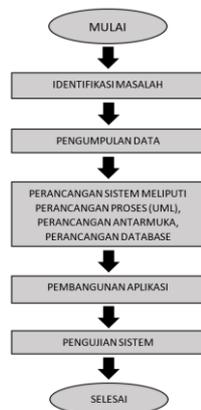
c. PostgreSQL

PostgreSQL merupakan sebuah Object Relational Database Management System (ORDBMS) yang dikembangkan oleh Berkeley Computer Science Department. Sistem yang ditawarkan PostgreSQL diharapkan sanggup dan dapat mencukupi untuk kebutuhan proses aplikasi data masa depan. PostgreSQL juga menawarkan tambahan-tambahan yang cukup signifikan yaitu class, inheritance, type, dan function. Beberapa keistimewaan lain yang tidak dimiliki Database Management System yang lain berupa constraint, triggers, rule, dan transaction integrity. Melihat adanya feature (keistimewaan) tersebut maka para pemakai dapat dengan mudah mengimplementasikan dan menyampaikan sistem ini [16].

3. METODE PENELITIAN

3.1 Metode Pengembangan Sistem

Proses pengembangan aplikasi Virtual Inventory System sebagai perantara antara client yang menggunakan Flask Framework dengan database perusahaan yang menggunakan database PostgreSQL dilakukan dengan alur penelitian yang dipakai ada lima tahapan, yaitu: (1) Identifikasi Masalah, (2) Pengumpulan Data, (3) Perancangan Sistem menggunakan Unified Modelling Language (UML), Perancangan Antarmuka, Perancangan Database, (4) Pembangunan Aplikasi, (5) Pengujian sistem.



Gambar 1. Bagan metodologi penelitian

Penjelasan mengenai tahapan pembuatan modul PO Manual dalam aplikasi VIS pada Gambar 1 adalah sebagai berikut:

a. Identifikasi Masalah

Pada tahap identifikasi masalah dilakukan dengan observasi dan konsultasi mengenai kebutuhan sistem yang akan dibuat. Dari hasil observasi dan konsultasi, didapatkan informasi yaitu ingin memperbarui aplikasi dari oracle form yang masih menggunakan teknologi lama ke aplikasi yang berbasis web, dikarenakan biaya pemeliharaan oracle form yang cukup besar [17] bila dibandingkan dengan pemeliharaan web. Selain itu

oracle form hanya dapat berjalan pada satu personal computer (PC) saja dan tampilan antarmuka dari oracle form masih bersifat statis sedangkan untuk web dapat berjalan di berbagai platform (multiplatform). Oleh karena itu tampilan antarmuka web sudah bersifat dinamis karena harus menyesuaikan platform yang ada. Dengan adanya aplikasi berbasis web ini diharapkan dapat mengurangi biaya pemeliharaan aplikasi dan mempermudah user dalam melakukan transaksi di berbagai platform.

b. Pengumpulan Data

Pada tahap pengumpulan data, didapatkan dari dokumentasi aplikasi yang lama. Data yang dikumpulkan dapat dijadikan acuan untuk pembuatan database pada aplikasi ini. Data yang dibutuhkan adalah:

- Data vendor yaitu data semua vendor pulsa yang bekerjasama dengan PT XYZ.
- Data barang yaitu data yang berisi identitas barang yang akan dibeli oleh perusahaan.
- Data branch yaitu data yang berisi identitas dari branch tersebut.

c. Perancangan Sistem

Berdasarkan hasil identifikasi pada tahap sebelumnya tahap ini akan dilakukan perancangan sistem. Perancangan sistem menggunakan UML. Pada penelitian ini digambarkan dengan use case diagram. Use case diagram menggambarkan interaksi antara aktor dengan sistem, dibuat sesuai dengan aktivitas bisnis yang ditetapkan pada analisis sistem yang sedang berlangsung [18], serta activity diagram. Activity diagram menggambarkan alur kerja dari suatu sistem atau proses bisnis atau menu yang ada pada perangkat lunak [19]. Perancangan antarmuka, perancangan database dengan menggunakan PostgreSQL. Tahap terakhir penggabungan semua rancangan sistem menjadi satu kesatuan di dalam sebuah script yang nanti akan dieksekusi di dalam sebuah modul.

d. Pembangunan Aplikasi

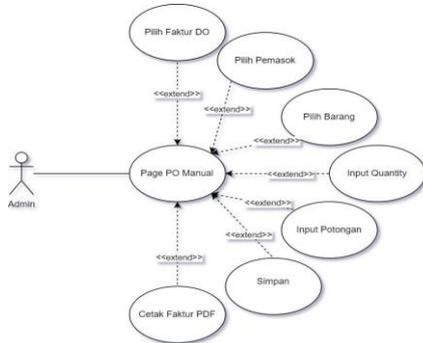
Pada tahap pembangunan aplikasi hasil rancangan yang sudah dibuat diimplementasikan ke dalam bahasa pemrograman yaitu proses coding yang menggunakan Flask Framework Python sebagai kerangka kerja aplikasi, PostgreSQL sebagai database, dan Bootstrap untuk mengimplementasikan tampilan antarmuka.

e. Pengujian Sistem

Pengujian dilakukan oleh Quality Assurance (QA) untuk mengetahui apakah sudah sesuai dengan hasil rancangan yang dibuat atau belum. QA melakukan pengamatan terhadap input dan output sistem apakah sudah sesuai dengan perancangan yang sudah dibuat atau belum [20] dan untuk memeriksa segala kemungkinan terjadinya kesalahan. Pengujian juga dimaksudkan untuk mengetahui seberapa banyak error bug yang terjadi dalam aplikasi yang dibangun. Dari hasil pengujian tersebut dapat dibuat laporan, jika masih terdapat

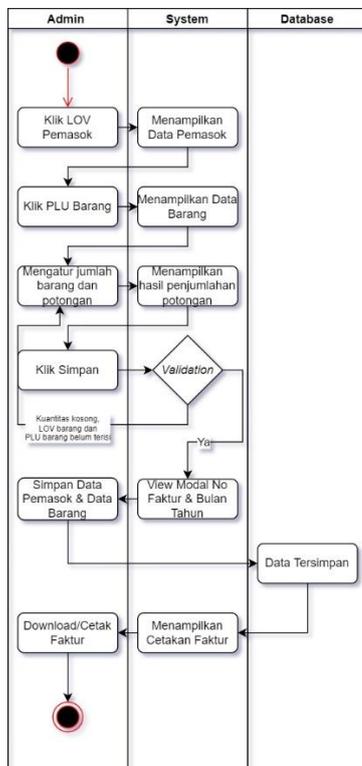
error atau bug maka aplikasi dikembalikan ke developer untuk diperbaiki, tetapi jika aplikasi sudah sesuai dengan perancangan maka akan dilanjutkan ke proses berikutnya.

Berikut adalah perancangan sistem menggunakan UML yang digambarkan dengan use case diagram dan activity diagram pada modul PO Manual.



Gambar 2. Use case diagram halaman PO Manual

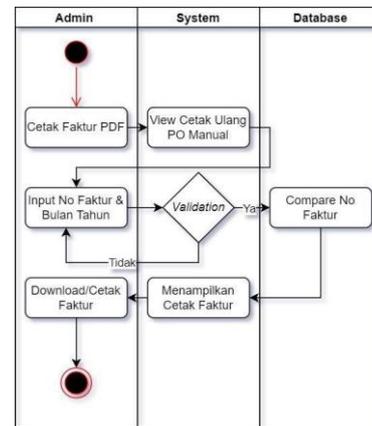
Gambar 2 merupakan use case diagram dari modul PO Manual, admin mempunyai akses untuk mengendalikan aplikasi tersebut. Admin dapat membuat transaksi pre order manual untuk pemesanan barang ke vendor terkait. Selain itu admin juga dapat mencetak faktur sesuai dengan nomor faktur transaksi yang sudah dilakukan dan admin bebas untuk melakukan aktivitas apa saja di dalam aplikasi VIS karena memiliki hak akses.



Gambar 3. Diagram aktivitas PO Manual

Gambar 3 merupakan penggambaran aktivitas dari modul PO Manual yang akan dipakai untuk

melakukan transaksi pengaturan harga voucher pulsa. Hal yang harus dilakukan oleh admin pertama kali adalah memilih pemasok sebagai tujuan transaksi, setelah pemasok sudah dipilih kemudian admin dapat memilih barang yang akan dibeli. Setelah PLU barang sudah terpilih dapat digunakan untuk mengatur jumlah barang dan potongan yang dibutuhkan. Ketika klik simpan List of Value (LOV) pemasok, PLU barang dan jumlah barang harus diisi jika tidak diisi maka tidak mendapatkan nomor faktur atau proses gagal dan admin harus memeriksa kembali agar semua terisi. Jika semua sudah terisi dengan benar ketika diklik simpan maka akan muncul modal yang menampilkan nomor faktur dan bulan tahun ketika transaksi dibuat. Nomor faktur bersifat increment yaitu bertambah satu angka ketika proses simpan berhasil. Klik simpan yang ada di modal maka data akan tersimpan di dalam database dan akan muncul halaman baru di browser yang menampilkan faktur. Admin dapat mengunduh dan mencetaknya sebagai arsip.



Gambar 4. Diagram aktivitas cetak faktur pdf

#### 4. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan modul PO Manual berbasis web yang dibangun menggunakan Flask Framework yang memungkinkan dapat berjalan di berbagai platform. Pembangunan modul ini dikerjakan secara offline dengan menggunakan database dummy dan server perusahaan yang dapat diakses dengan Virtual Private Network (VPN) yang sudah disediakan. Selanjutnya akan dibahas mengenai interface dan fungsi pada modul.

##### 4.1. Halaman PO Manual



Gambar 5. Halaman login

Pada Gambar 6 dijelaskan bagi admin yang akan melakukan transaksi harus memasukkan NIK dan PIN terlebih dahulu. Jika berhasil masuk admin akan diarahkan ke halaman *dashboard* dan dapat melakukan beberapa aktivitas dalam aplikasi VIS. Berikut adalah kode program dari halaman *login*.

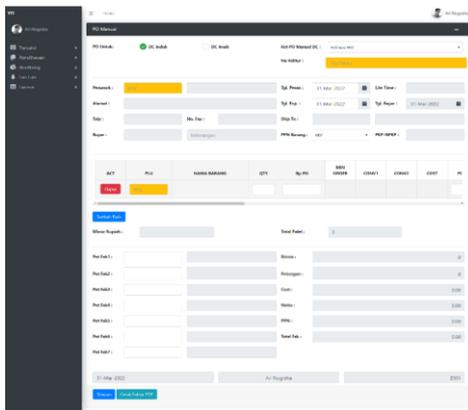
```

1  var login = () => {
2      $.ajax
3      ({
4          type: "POST",
5          data: {
6              username: $("#nik").val(),
7              password: $("#pin").val()
8          },
9          url: "/signin",
10         success: (e) => {
11             console.log(e.status)
12             if (e.status) {
13
14                 window.location.href="./"
15             } else {
16
17                 $("#error").html(e.msg);
18             }
19         },
20         error: function (hr, status, err)
21
22             alert(hr.responseText);
23     });
24 }
25

```

Kode Program 1. Kode untuk menampilkan bagian halaman *login*

Kode Program 1 merupakan kode dari halaman *login* yang mempunyai fungsi pengecekan terhadap *username* dan *password*. Jika *username* dan *password* cocok program akan eksekusi halaman *dashboard*, tetapi jika tidak cocok program akan menampilkan pesan *error*.



Gambar 6. Halaman PO *Manual*

Pada Gambar 7 menunjukkan halaman PO *Manual* dalam aplikasi VIS, di halaman inilah bagian *Head Office* (HO) melakukan pengaturan harga pulsa dan atribut lainnya. Tampilan ini juga dibangun menggunakan *Bootstrap*. Pada halaman ini terdapat *text box* warna kuning yang disebut *List of Value* (LOV) yang akan menampilkan daftar faktur, pemasok dan PLU seperti yang terlihat pada Gambar 8 ketika diklik.

Terdapat *text box* warna putih pada halaman PO *Manual* berfungsi sebagai *input text* atau ketik manual. *Text box* abu berfungsi menampilkan data saja tidak bisa diketik. *Radio button* berfungsi untuk memilih *item*. *Datetimepicker* berfungsi untuk menampilkan tanggal. *Select option* untuk memilih *item* dengan berbagai pilihan. *Button* hapus berfungsi untuk menghapus baris dalam tabel, *button* tambah baris berfungsi untuk menambah baris baru dan *button* simpan berfungsi untuk menyimpan data ke dalam *database*, dan *button* cetak faktur PDF berfungsi untuk mencetak faktur. Modul ini dibuat menggunakan *Flask Framework*, *Flask Framework* diterapkan pada setiap LOV karena di dalam LOV tersebut terjadi proses ambil data dari *controller* ke *Data Access Object* (DAO) dengan menggunakan *Asynchronous JavaScript and XML* (AJAX). Kode Program 2 adalah kode untuk membangun modul PO *Manual*.

```

1  <div class="container-fluid" ng-
2  controller="POManualController" ng-app="app">
3  <!-- FORM DEFAULT -->
4  <form name="frm" id="frm">
5  <div class="card card-dark">
6  <div class="card-header">
7  <h3 class="card-title">PO
8  Manual</h3>
9
10 <div class="card-tools">
11 <button type="button"
12 class="btn btn-tool" data-card-
13 widget="collapse"><i
14 class="fas fa-
15 minus"></i></button>
16 <button type="button"
17 class="btn btn-tool" data-card-
18 widget="remove"><i
19 class="fas fa-
20 remove"></i></button>
21 </div>
22 </div>

```

Kode Program 2. Kode untuk menampilkan halaman PO *Manual*

Kode Program 2 merupakan kode dari halaman PO *Manual* yang mempunyai fungsi untuk menampilkan halaman PO *Manual* itu sendiri.



Gambar 7. LOV pemasok

Gambar 8 menampilkan daftar data pemasok yang bekerjasama dengan perusahaan XYZ. Di dalam modal ini admin dapat mencari pemasok dengan memasukkan kode pemasok atau nama pemasok yang sudah terdaftar di *database*. Setiap pemasok mempunyai kode yang berbeda-beda yang menjadi *primary key* agar dapat membedakan pemasok satu dengan lainnya.

```

1 $http({
2     method: 'GET',
3     headers: {
4         'Content-Type':
5         'application/x-www-form-
6         urlencoded;charset=utf-8'
7     },
8     url:
9     '/pomannual/lovSupp?param=' +
10    angular.element("#cariSupp").val()
11    }).then(function
12    successCallback(response) {
13
14        if(!response.data){
15
16        Swal.fire('Error','Kode Supplier tidak ada di
17        File Supplier !!','error');
18        }else{
19            $rootScope.dataSupp =
20            response.data;
21            $scope.loading =
22            'Jumlah Data : ' + response.data.length;;
23        }
24    }, function
25    errorCallback(error) {
26        $scope.error = error;
27        $scope.loading = error;
28    });
29
30

```

Kode Program 3. Kode untuk menampilkan daftar pemasok

Kode Program 3 kode dari halaman PO *Manual* yang berfungsi untuk menampilkan daftar pemasok. Jika pemasok yang dicari tidak terdapat dalam daftar maka akan muncul pesan *error*.



Gambar 8. Cetakn faktur PO *Manual*

Gambar 9 merupakan cetakan faktur PO *Manual*, faktur dicetak jika sudah melakukan *pre order* dengan cara dilihat dari nomor faktur, jika nomor faktur tidak ada di *database* maka akan muncul *error* dan jika nomor faktur ada di *database* maka faktur akan dicetak. Data transaksi seperti nama pemasok, nama barang, potongan, dan bonus tercantum dalam faktur. Faktur dapat diunduh oleh admin sebagai bukti transaksi berupa *hard file*. Berikut adalah Kode Program dari cetakan faktur.

```

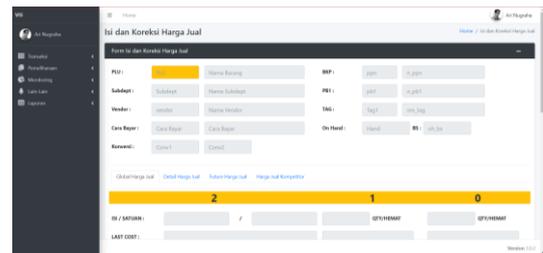
1 <div id="print" style="display: none;" >
2   <table id="tblHead" style="border:
3   none;">
4     <tr style='border: none;'>
5       <td style="text-align: left;
6       margin-right: 50px; border: none;">
7         <span style='font-
8         size:10pt'><b>{{{{ head1(dataPrint)
9         }}}}</b></span><br>
10        <span style='font-
11        size:12pt'><b>FORMULIR PESANAN PEMBELIAN /
12        MANUAL</b></span>
13      </td>
14      <td style="text-align: left;
15      margin-left: 100px; border: none;">
16        <br><span style='font-
17        size:8pt'>{{{{ xcetak(var) }}}}</span>
18      </td>
19    </tr>
20  </table>

```

Kode Program 4. Kode untuk menampilkan cetakan faktur

Kode Program 4 kode dari halaman PO *Manual* yang berfungsi untuk menampilkan cetakan faktur. Cetakan faktur berbentuk PDF yang dapat langsung diunduh atau dicetak.

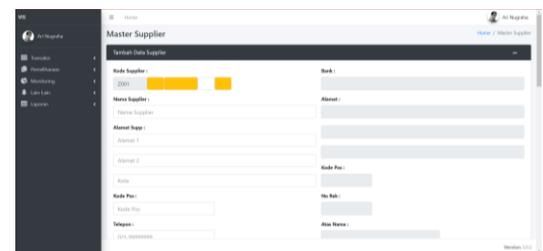
#### 4.2. Halaman Harga Jual



Gambar 9. Halaman harga jual

Gambar 10 merupakan tampilan untuk mengatur harga jual yang dilakukan oleh HO. Ketika PLU diklik akan muncul daftar yang berisi nama barang berupa voucher pulsa dari setiap vendor yang sudah bekerjasama dengan perusahaan.

#### 4.3. Halaman Tambah Data *Supplier*



Gambar 10. Halaman tambah data *supplier*

#### 4.4. Pengujian Sistem

Pengujian dilakukan oleh *Quality Assurance* (QA). QA adalah metode manajemen yang didefinisikan sebagai semua tindakan terencana dan sistematis yang diperlukan untuk memberikan keyakinan yang memadai pada suatu produk, layanan atau hasil akan memenuhi persyaratan terhadap kualitas yang diberikan dan layak untuk

digunakan [21]. Pengujian dilakukan baik logika internal maupun eksternal untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari aplikasi tersebut sesuai dengan hasil yang diinginkan serta mengetahui seberapa banyak *error*, *bug* yang terjadi dalam aplikasi ini. Pengujian ini dengan tujuan untuk mengetahui apakah sistem sudah berfungsi dengan baik atau tidak. Berdasarkan pengujian dari *Quality Assurance* didapatkan laporan seperti pada Tabel 1.

Tabel 1. Pengujian *Quality Assurance*

No	Uraian	Tipe Test (+/-)	Hasil yang diinginkan	Status
1	Form belum berfungsi belum dapat membaca <i>user</i> sesuai dengan <i>user</i> login	+	Sudah sesuai	OK
2	<i>tgl_exp</i> ( <i>tgl_tiba</i> ) harus lebih besar dari tanggal pesan jika <i>tgl_exp</i> < dari tanggal pesan , terdapat alert	+	Sudah sesuai	OK
3	Jika <i>user</i> menginput plu yang sama dalam sebuah PO maka terdapat alert "plu sudah diinputkan !"	+	Sudah sesuai	OK
4.	Jika <i>user</i> sudah klik simpan seharusnya mereload halaman / mereset data yang sudah di input dalam form	+	Sudah sesuai	OK
5.	tanggal bayar harus > dari tanggal <i>exp</i>	+	Sudah sesuai	OK
6.	untuk no fax belum terdapat ditampilkan header cetakan PO	+	Sudah sesuai	OK

Tabel 1 merupakan hasil pengujian modul PO *Manual* dari QA. Kolom uraian berisi tentang fungsional dari program yang diuji. Terdapat dua tipe *test*, (+) jika program sudah berfungsi sesuai alur, jika (-) masih terdapat *bug* atau *error* dari program. Pada kolom hasil yang diinginkan berisi tentang alur program, jika program terdapat *bug* maka diberi catatan tentang alur program yang seharusnya, tetapi jika program sudah sesuai maka ditulis sudah sesuai. Jika program sudah sesuai statusnya OK, jika belum statusnya NOK.

Dari Tabel 1 didapatkan bahwa modul PO *Manual* sudah sesuai dengan *input* dan *output* yang diinginkan, selanjutnya dari QA akan dilanjutkan ke pengujian berikutnya yaitu *User Acceptance Testing* (UAT). UAT adalah pengujian terakhir dan paling penting dari empat fase umum pengujian perangkat lunak. Pada tahap ini, pengujian sistem dilakukan untuk menentukan apakah sistem memenuhi kebutuhan pengguna dan dapat mendukung semua skenario bisnis dan pengguna. UAT dilakukan oleh *client* dan *end-user* [22].

**5. KESIMPULAN DAN SARAN**

Dari hasil Pembangunan Aplikasi *Virtual Inventory System* (VIS) Berbasis Web Menggunakan *Flask Framework* pada penelitian ini,

maka dapat menarik kesimpulan yaitu, penelitian ini menghasilkan modul PO *Manual* yang dibangun menggunakan *Flask Framework*. Modul PO *Manual* dapat membantu admin untuk mengatur harga voucher pulsa dan mencetak faktur barang. Keuntungan menggunakan web adalah bisa diakses di segala *platform* (*multiplatform*) sehingga lebih mempermudah *user* dalam mengakses aplikasi VIS ini. Pada tahap pengujian sistem menunjukkan bahwa hasil pembangunan modul PO *Manual* sudah sesuai dengan kebutuhan PT. XZY. Saran terhadap aplikasi VIS ini adalah belum adanya fitur pelacakan *user* yang *login* sehingga masih kesusahan untuk melacak siapa saja yang telah login di aplikasi tersebut. Oleh karena itu pada penelitian berikutnya dapat diterapkan dan aplikasi VIS dapat dijadikan sebagai pedoman untuk mengembangkan suatu aplikasi yang baru secara sistematis yang dapat disusun menjadi sebuah sistem yang kompleks.

**DAFTAR PUSTAKA**

[1] R. Industri, D. A. N. Tantangan, and P. Sosial, "Revolusi Industri 4.0 Dan Tantangan Perubahan Sosial," *IPTEK J. Proc. Ser.*, vol. 0, no. 5, pp. 22–27, 2018, doi: 10.12962/j23546026.y2018i5.4417.

[2] A. O. Pranoto and E. Sedyono, "Perancangan Sistem Informasi Inventaris Barang Berbasis Web," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 2, pp. 357–372, 2021, doi: 10.28932/jutisi.v7i2.3597.

[3] R. Anjarsari, "Sistem informasi inventaris barang di kantor arsip dan perpustakaan daerah surakarta," *Sist. Inf. Inventar. Barang Di Kant. Arsip Dan Perpust. Drh. Surakarta*, p. 60, 2009.

[4] A. A. A. Syukri, "Information System of Inventory Goods Web-Based on the," *Indones. J. Inf. Syst. Sist.*, vol. 1, no. April 2016, pp. 31–38, 2016.

[5] R. A. Pribachtiar, "Perancangan Sistem Informasi Inventory Barang (E-Gudang) Pada Cv Jaya Water Solusindo," vol. 5, no. 3, pp. 54–63, 2021.

[6] N. Chauhan, M. Singh, A. Verma, A. Parasher, and G. Budhiraja, "Implementation of database using python flask framework," *Int. J. Eng. Comput. Sci.*, vol. 8, no. 12, 2019, doi: 10.18535/ijecs/v8i12.4390.

[7] "Flask Vs Django : Apa Itu ? Ini Reviewnya - DosenIT.com." <https://dosenit.com/python/flask-vs-django> (accessed Jul. 05, 2022).

[8] H. Alatas, "Responsive Web Design dengan PHP dan Bootstrap," *LokoMedia*, vol. 1, no. 3, pp. 1–231, 2013.

[9] Ramadani, Ruslan, and E. S. Rahman, "Pengembangan Aplikasi Monitoring Stok Barang Berbasis Web Di Gudang Pt . Pjb Services Unit Pltd Suppa Pinrang Untuk

- Meningkatkan Kinerja Perusahaan.”
- [10] D. F. Ningtyas and N. Setiyawati, “Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request,” *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.
- [11] B. P. Putra and Y. A. Susetyo, “Implementasi Api Master Store Menggunakan Flask, Rest Dan Orm Di Pt Xyz,” *Sistemasi*, vol. 9, no. 3, p. 543, 2020, doi: 10.32520/stmsi.v9i3.899.
- [12] R. K. Ngantung and M. A. I. Pakereng, “Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python,” *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 1052, 2021, doi: 10.30865/mib.v5i3.3054.
- [13] R. Irsyad, “Penggunaan Python Web Framework Flask Untuk Pemula,” 2018, doi: 10.31219/osf.io/t7u5r.
- [14] “What is Python? Executive Summary | Python.org.”  
<https://www.python.org/doc/essays/blurb/> (accessed Nov. 04, 2021).
- [15] R. R. Saragih, “Pemrograman dan bahasa pemrograman,” no. December, 2018.
- [16] R. Yuliardi, “Panduan Administrasi Database PostgreSQL,” p. 129, [Online]. Available: [https://docplayer.info/29971702-Panduan-administrasi-database-postgresql-versi-1-4-buku-1.html#tab\\_1\\_1\\_2](https://docplayer.info/29971702-Panduan-administrasi-database-postgresql-versi-1-4-buku-1.html#tab_1_1_2).
- [17] Oracle, “Oracle Technology Global Price List Software Investment Guide,” pp. 1–13, 2012, [Online]. Available: <http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>.
- [18] M. Eka Purnama Rijaludin, W. Witanti, and A. Id Hadiana, “Sistem Informasi Administrasi Persuratan Terintegrasi Jurusan Informatika Dan Fakultas Mipa Universitas Jenderal Achmad Yani,” *J. Mnemon.*, vol. 1, no. 1, pp. 25–31, 2019, doi: 10.36040/mnemonic.v1i1.16.
- [19] W. Aprianti and U. Maliha, “Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati,” vol. 2, no. 2013, pp. 21–28, 2016.
- [20] R. Mubarak, “Implementasi Metode White Box Testing Pada Proses Quality Assurance Perangkat Lunak Berbasis Web Dan Mobile Collection System,” *J. Teknol. Inf. ESIT*, vol. XV, no. 10, pp. 57–63, 2020.
- [21] a Storey, R. Briggs, H. Jones, and R. Russell, “Chapter 4 \*: QUALITY ASSURANCE,” no. January, pp. 38–43, 2000.
- [22] “User Acceptance Test – School of Information Systems.”  
<https://sis.binus.ac.id/2020/10/28/user-acceptance-test/> (accessed Jun. 05, 2022).