

BAB 4 PERANCANGAN DAN IMPLEMENTASI

4.1 Dasar Pemikiran

Di dalam bisnis usaha produksi bahan baku dan jasa pelayanan, pengelolaan persediaan stok inventory sangatlah penting. Inventory bertanggung jawab untuk menjaga dan mengelola segala bahan mentah, setengah jadi, dan produk jadi yang ada di perusahaan agar dapat digunakan untuk kebutuhan operasional. Barang tersebut merupakan investasi yang merupakan sumber pendapatan perusahaan. Antisipasi permasalahan stok *inventory* harus dilakukan untuk menghindari kegagalan bisnis, contohnya adalah stok kuantitas yang terlalu banyak atau sedikit di gudang akan merugikan bisnis usaha. Oleh karena itu, *monitoring* dan supervisi stok *inventory* di segala perusahaan merupakan hal penting yang tidak boleh dilupakan.

PT. Putra Ganda Jaya Pratama masih memiliki monitoring stok gudang produksi yang kurang efektif. Sistem yang berjalan masih menggunakan ms. excel untuk mengelola database stok gudang produksi dan monitoring jumlah stok gudang produksi. Solusi dari permasalahan tersebut adalah dikembangkannya Sistem Informasi Stok Gudang Produksi Berbasis Desktop dengan Bahasa Python. Sistem ini akan mempercepat proses input data yang dilakukan admin ke dalam sistem dengan adanya relasi database. Monitor stok gudang produksi juga dapat dilakukan dengan lebih mudah dengan tampilan form dan *widget* yang interaktif oleh bahasa pemrograman python. Pengembangan aplikasi menggunakan *web framework* Django yang menyediakan sarana *web hosting* di *local machine* dengan *virtual environment* agar aplikasi ini terisolasi dan independen (Sohail, 2018).

4.2 Perencanaan

Project penelitian yang akan dikembangkan adalah Sistem Informasi Stok Gudang Produksi Berbasis Desktop dengan Bahasa Python. Penelitian ini menggunakan bahasa pemrograman python dan *web framework* Django dengan database SQLite. Aplikasi ini menggunakan *virtual environment* agar aplikasi terisolasi dan independen. *Hosting* aplikasi dilakukan di local machine sehingga tidak diperlukan koneksi internet.

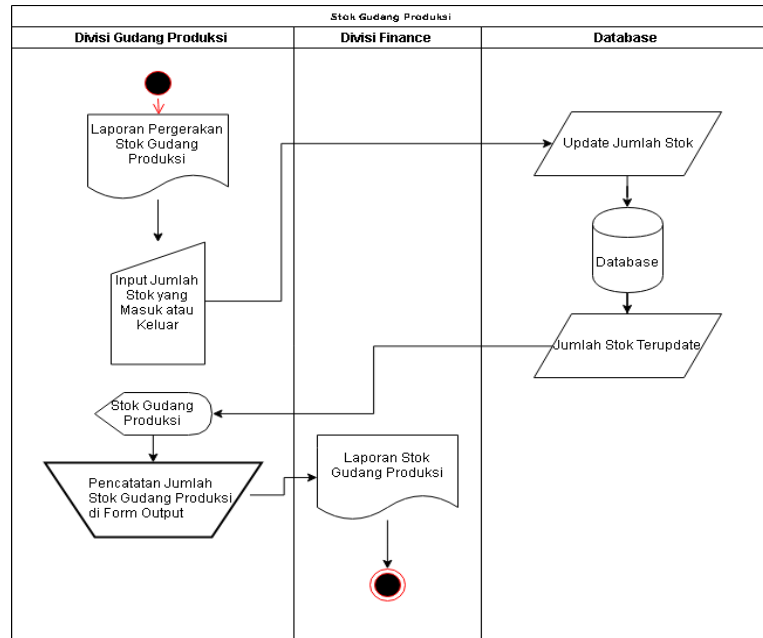
Untuk membangun project Django diperlukan instalasi Python di *local machine* yang akan digunakan. Project ini menggunakan database SQLite yang sudah termasuk ke dalam instalasi Python sehingga tidak perlu melakukan set up database lagi. Kemudian melakukan instalasi Django di direktori file yang akan digunakan, dengan menggunakan *virtual environment* terlebih dahulu. Kemudian lakukan verifikasi di command prompt untuk mengetahui apakah instalasi Python dan Django sudah berhasil.

4.3 Analisis

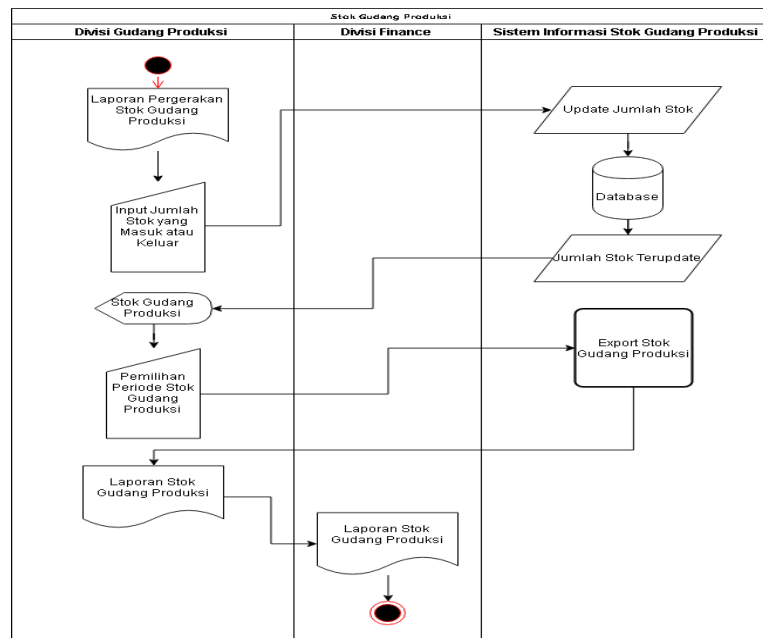
Dalam sistem berjalan pengelolaan stok gudang produksi di PT. Putra Ganda Jaya Pratama, admin mengisi seluruh pergerakan stok barang yang terjadi di database yang dikelola di ms. excel berdasarkan laporan harian yang diterima dari lapangan. Jika suatu waktu divisi finance memerlukan laporan stok gudang produksi, maka admin harus menulis jumlah total stok keluar dan total biayanya di dalam form laporan sesuai dengan yang tertera di database. Sedangkan dalam sistem yang akan

dikembangkan, admin hanya melakukan input jumlah pergerakan stok barang di aplikasi. Jika divisi *finance* memerlukan laporan stok gudang produksi, maka admin hanya melakukan print stok gudang produksi sesuai permintaan periode waktu dan menyerahkannya ke divisi *finance*.

Berikut ini adalah perancangan aplikasi stok gudang produksi pada sistem berjalan dan sistem yang akan dikembangkan.



Gambar 4 Flowmap Sistem Berjalan

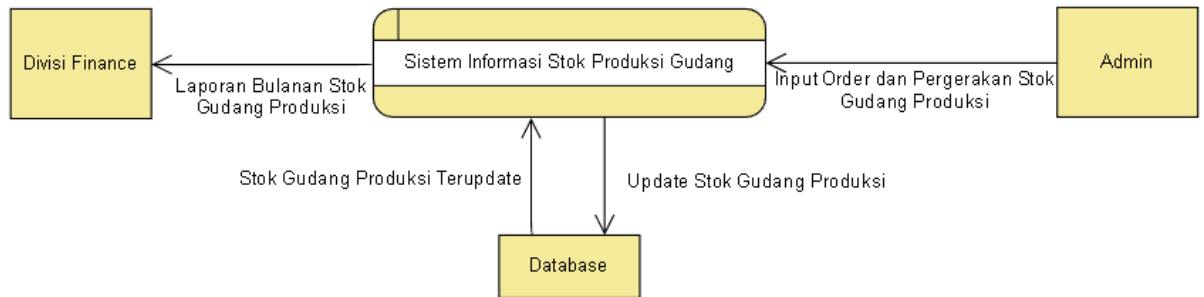


Gambar 5 Flowmap Sistem yang akan Dikembangkan

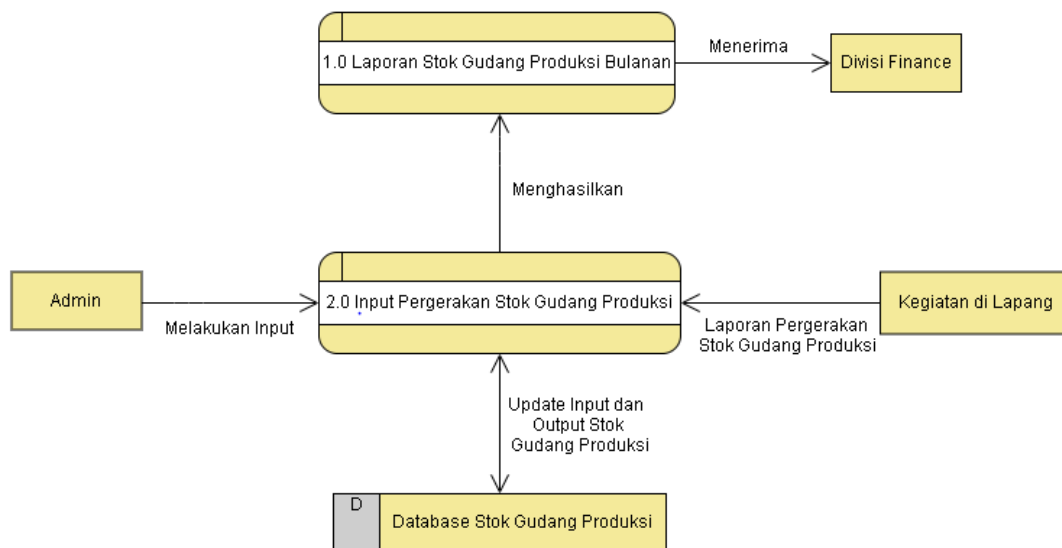
4.4 Perancangan

4.4.1 Perancangan Prosedural Data Flow Diagram

Data Flow Diagram merupakan sebuah teknik dalam menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output. Berikut ini adalah perancangan aplikasi stok gudang produksi yang menggambarkan alur pemrosesan data di dalam sistem informasi stok gudang produksi. Gambar 6 menunjukkan Context Diagram dari sistem informasi stok gudang produksi yang akan dikembangkan dan gambar 7 menunjukkan Data Flow Diagram Level 0.



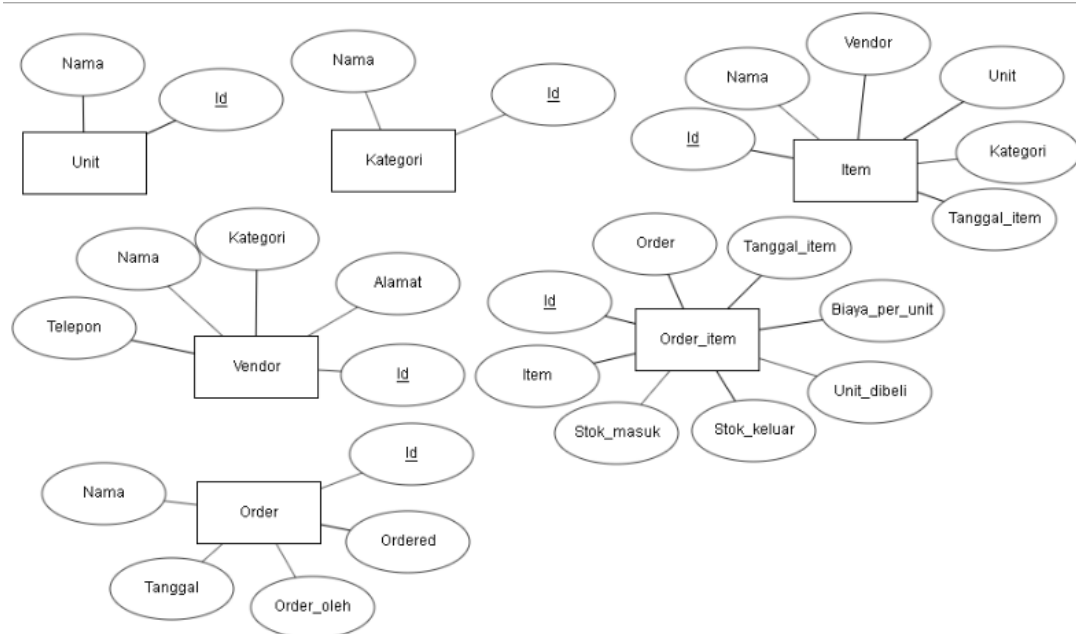
Gambar 6 Context Diagram Sistem Informasi Stok Gudang Produksi



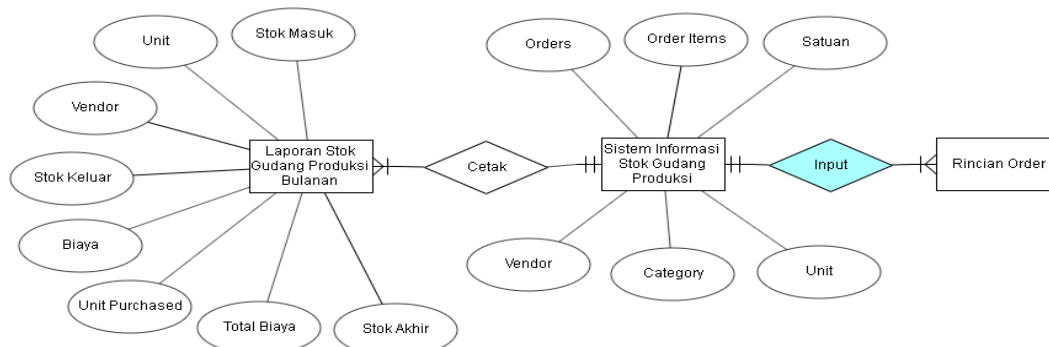
Gambar 7 Data Flow Diagram Level 0 Sistem Informasi Stok Gudang Produksi

4.4.2 Perancangan Database

Entity Relation Diagram adalah diagram yang menggambarkan hubungan antara obyek data. Model ER-Diagram menjadi salah satu pemodelan data konseptual yang paling sering digunakan dalam proses pengembangan basis data bertipe relasional. Berikut ini adalah perancangan aplikasi stok gudang produksi yang menggambarkan entitas yang saling berhubungan di dalam sistem informasi stok gudang produksi. Gambar 8 menunjukkan entitas yang berhubungan di database pada sistem informasi stok gudang produksi yang akan dikembangkan dan gambar 9 menunjukkan ERD pemrosesan data untuk menghasilkan laporan bulanan untuk divisi *finance*.



Gambar 8 Entitas yang Berada di Database Sistem Informasi Stok Gudang Produksi



Gambar 9 Entity Relation Diagram Sistem Informasi Stok Gudang Produksi

Tabel 2 Spesifikasi Tabel Unit

Tabel	Tipe	Keterangan
Id	Int (5)	PK
Nama	Varchar (45)	Not Null

Tabel 3 Spesifikasi Tabel Item

Tabel	Tipe	Keterangan
Id	Int (5)	PK
Nama	Varchar (45)	Not Null
Vendor	Varchar (45)	FK
Unit	Varchar (45)	FK
Kategori	Varchar (45)	FK
Tanggal_item	Date	Not Null

Tabel 4 Spesifikasi Tabel Kategori

Tabel	Tipe	Keterangan
Id	Int (5)	PK
Nama	Varchar (45)	Not Null

Tabel 5 Spesifikasi Tabel Vendor

Tabel	Tipe	Keterangan
Id	Int (5)	PK
Nama	Varchar (45)	Not Null
Kategori	Varchar (45)	FK
Alamat	Varchar (45)	Not Null
Telepon	Varchar (16)	Not Null

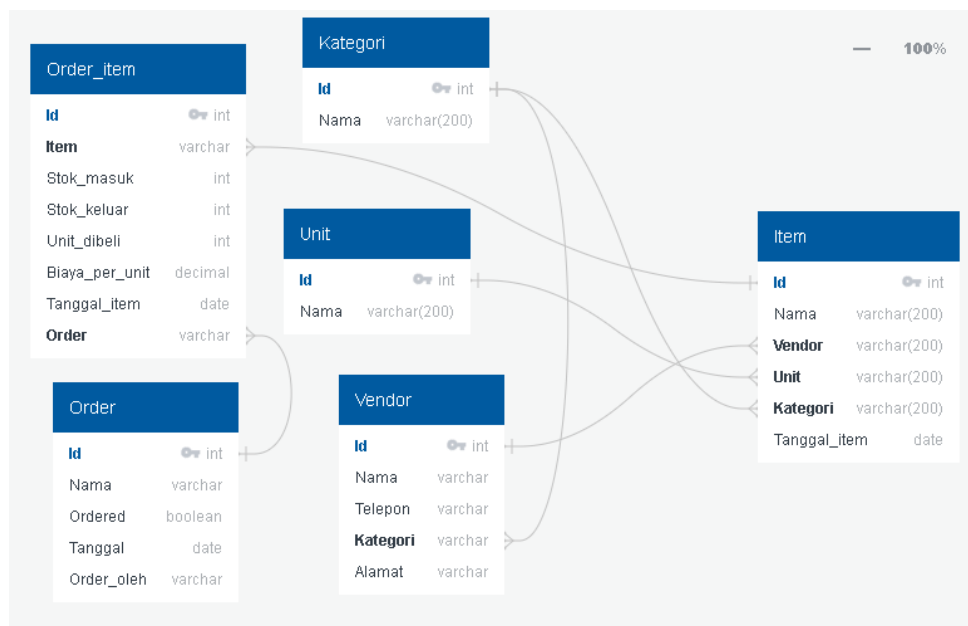
Tabel 6 Spesifikasi Tabel Order_item

Tabel	Tipe	Keterangan
Id	Int (5)	PK
Order	Varchar (45)	FK
Tanggal_item	Date	Not Null
Biaya_per_unit	Decimal	Not Null
Unit_dibeli	Integer	Not Null
Stok_masuk	Integer	Not Null
Stok_keluar	Integer	Not Null
Item	Varchar (45)	FK

Tabel 7 Spesifikasi Tabel Order

Tabel	Tipe	Keterangan
Id	Int (5)	PK
Nama	Varchar (45)	Not Null
Tanggal	Date	Not Null
Order_oleh	Varchar (45)	FK
Ordered	Boolean	Not Null

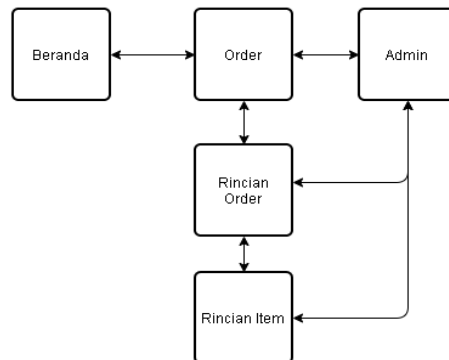
Tabel 2,3,4,5,6, dan 7 menunjukkan spesifikasi tabel yang berada di dalam database sistem informasi stok gudang produksi. Gambar 10 menunjukkan relasi database sistem informasi stok gudang produksi.



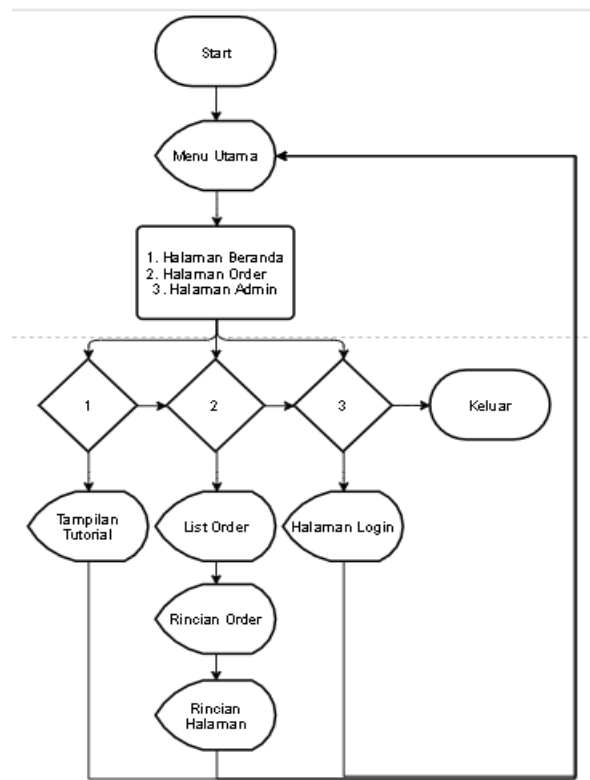
Gambar 10 Relasi Database Inventory di Sistem Informasi Stok Gudang Produksi

4.4.3 Perancangan Navigasi

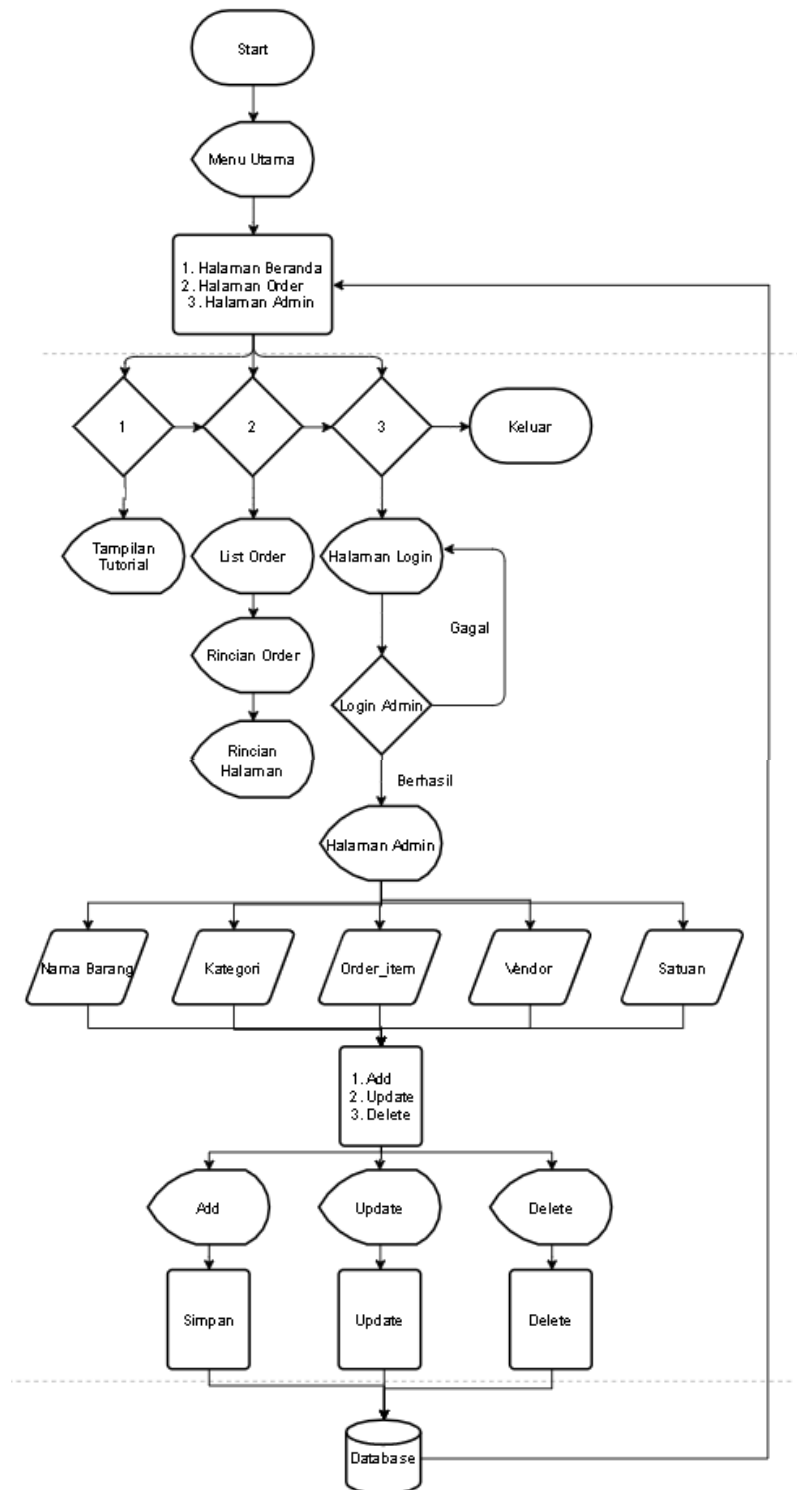
Flowchart navigasi adalah representasi alur dari proses navigasi yang dapat dilakukan di dalam sistem, dimana setiap langkah proses direpresentasikan oleh simbol dan memiliki deskripsi singkat mengenai proses tersebut. Flowchart *front-end* merepresentasikan alur proses yang dapat dilakukan oleh pengguna di dalam sistem, sedangkan flowchart *back-end* merepresentasikan alur proses yang dapat dilakukan oleh admin di dalam sistem. Gambar 11 menunjukkan flowchart navigasi, gambar 12 menunjukkan flowchart *front-end*, dan gambar 13 menunjukkan flowchart *back-end*.



Gambar 11 Flowchart Navigasi Sistem Informasi Stok Gudang Produksi



Gambar 12 Flowchart Front-End Sistem Informasi Stok Gudang Produksi

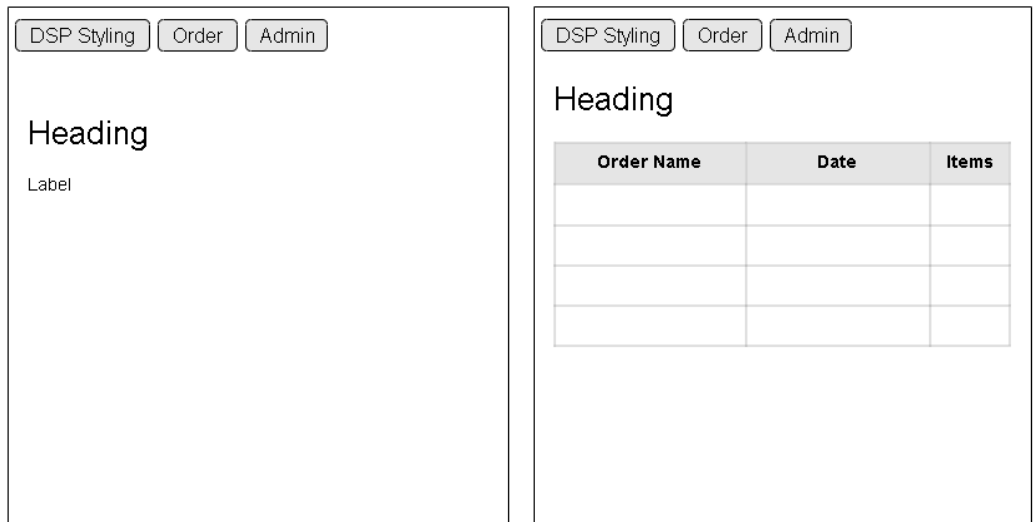


Gambar 13 Flowchart Back-End Sistem Informasi Stok Gudang Produksi

4.4.4 Perancangan Mockup Antarmuka Pengguna

Mockup Antarmuka Pengguna adalah rancangan prototipe dari model design sepenuhnya yang berguna untuk memberikan evaluasi dan evaluasi dari fungsionalitas sebuah sistem sehingga dapat dilakukan testing. Gambar 14,15,16,17, dan 18 menunjukkan lima perancangan halaman utama yang akan dikembangkan di dalam

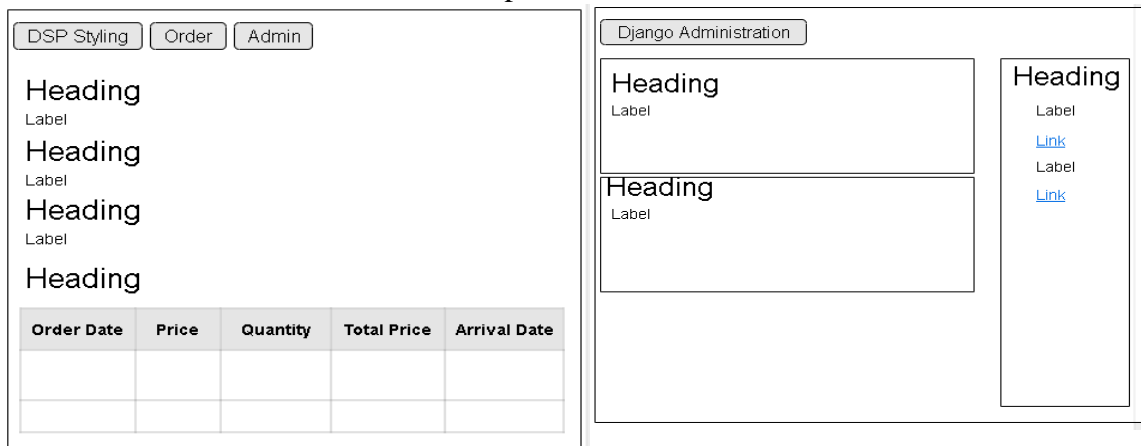
sistem informasi stok gudang produksi.



Gambar 14 dan 15 Mockup Halaman Beranda dan Halaman List Order



Gambar 16 Mockup Halaman Rincian Order



Gambar 17 dan 18 Mockup Halaman Rincian Item dan Halaman Admin

4.5 Implementasi

Implementasi dari penelitian Sistem Informasi Stok Gudang Produksi Berbasis Desktop dengan Bahasa Python yang akan dikembangkan dilakukan dengan cara memahami bagaimana laporan stok gudang produksi diproduksi secara manual dan juga komputerisasi. Hal ini dilakukan dengan cara wawancara ke narasumber berupa admin gudang produksi dan divisi *finance*. Wawancara dilakukan agar diketahui kebutuhan admin gudang produksi sebagai klien sehingga pengembangan software dapat berjalan dengan lancar dan aplikasi ini dapat digunakan sebagai solusi dari permasalahan yang ada. Gambar 19, 20, 21, 22, 23, dan 24 menunjukkan pembuatan tabel database inventory di Sistem Informasi Stok Gudang

Table: **inventory_vendor**

	id	name	telepon	alamat
	Filter	Filter	Filter	Filter
1	2	PT. MAKMUR ...	0213509955	Jl. Musi No. 2...
2	3	PT. WARNA I...	0215604765	Green Villa Bl...

Database Structure Browse Data Edit Pragas Execute SQL

Table: **inventory_order**

	id	name	created	ordered	order_date	on
	Filter	Filter	Filter	Filter	Filter	Filter
1	5	kucing	2019-09-17 1...	0	2019-09-17	2
2	6	Isis	2019-09-22 0...	0	2019-09-22	1

New Record Delete Record

Gambar 19 dan 20 Tabel Vendor dan Order di Database Inventory

Table: **inventory_unit**

	id	name
	Filter	Filter
1	2	Pcs
2	3	Roll
3	4	Set
4	5	Klg
5	6	Tube
6	7	Lbr
7	8	Lembar
8	9	Liter
9	10	Kg

Database Structure Browse Data Edit Pragas Execute SQL

Table: **inventory_order_items**

	id	units_purchased	cost	date_arrived
	Filter	Filter	Filter	Filter
1	6	20	14000	2019-09-17
2	7	140	2000	2019-09-17
3	8	10	15000	2019-09-17
4	9	20	12000	2019-09-17
5	10	0	NULL	2019-09-17
6	11	15	NULL	2019-09-22
7	12	200	NULL	2019-09-22

New Record

Gambar 21 dan 22 Tabel Unit dan Order_item di Database Inventory

Table: inventory_item			Table: inventory_category		
	id	name		id	name
	Filter	Filter		Filter	Filter
1	6	Lampu LED Fr...	1	4	Gudang Atas
2	7	Cat Pajero Bla...	2	5	Cat
3	8	Amplas Nikke...	3	6	Bahan Penolong
4	9	OH 38	4	7	RIM
5	10	Resin 220 SHCP	5	8	FRP
6	11	kucing			

Gambar 23 dan 24 Tabel Item dan Category di Database Inventory

Web framework Django menggunakan lima aspek penting dalam pengembangan web, yaitu url, model, admin, view, dan template. Url berisi link yang menghubungkan model yang dibuat dengan view yang akan ditampilkan di template. Model adalah segala entitas berbentuk class yang ada di sistem informasi dan seluruh aspeknya berhubungan langsung dalam pembuatan tabel database. Admin adalah segala konfigurasi yang mengatur fitur apa saja yang dapat dilakukan untuk manipulasi seluruh aspek lainnya. View adalah pengaturan tampilan bagaimana sebuah model ditampilkan di template. Template adalah file html yang digunakan untuk menampilkan interface kepada user. Gambar 25, 26, 27, 28, dan 29 menunjukkan sebagian coding dari lima aspek tersebut.

```
from django.conf.urls import url

from inventory import views

app_name = "inventory"
urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^orders/$', views.OrdersView.as_view(), name='orders'),
    url(r'^orders/(?P<pk>\d+)/$', views.OrderView.as_view(), name='order'),
    url(r'^items/(?P<pk>\d+)/$', views.ItemView.as_view(), name='item'),
]
```

Gambar 25 File urls.py dalam Sistem Informasi Stok Gudang Produksi

```
from __future__ import unicode_literals

from django.contrib.auth.models import User
from django.urls import reverse
from django.db import models
import datetime
from django.db.models import Sum

class Category(models.Model):
    name = models.CharField(max_length=45)

    def __str__(self):
        return self.name

    class Meta:
        verbose_name_plural = "kategori"
        ordering = ['name']
```

Gambar 26 File models.py yang Menunjukkan Class Category

```

from django.contrib import admin
from inventory.models import Item, Category, Unit, Vendor, Order, OrderItem
import csv
from django.http import HttpResponse

class OrderItemAdmin(admin.ModelAdmin):

    date_hierarchy = 'date_arrived'
    fields = ('stok_masuk', 'stok_keluar', 'units_purchased', 'cost', 'date_arrived')

    list_display = ('name', 'order_date', 'date_arrived', 'total_price', )
    list_filter = ('item_name', 'order_order_date', 'date_arrived', )
admin.site.register(OrderItem, OrderItemAdmin)

for model in (Category, Unit, Vendor):
    admin.site.register(model)

```

Gambar 27 File admin.py yang Menunjukkan Konfigurasi Rincian Order

```

from django.shortcuts import get_object_or_404, render
from django.http import HttpResponseRedirect, HttpResponse
from django.urls import reverse
from django.views import generic
from inventory.models import Item, Category, Unit, Vendor, Order, OrderItem
from django.db.models import Sum
from inventory.models import Order, Item

def index(request):
    return render(request, 'inventory/index.html')

```

Gambar 28 File views.py yang Menunjukkan Tampilan Halaman Beranda

```

{% extends "base_view.html" %}
{% block title %} meliza-lab : orders {% endblock %}
{% block content %}
{% if order_list %}
<p class="lead">Orders:</p>
<table class="table table-striped table-condensed">
  <thead>
    <th>Order name</th>
    <th>Date</th>
    <th>Items</th>
  </thead>
  <tbody>
    {% for order in order_list %}
      <tr>
        <td><a href="{% url 'inventory:order' order.id %}">{{ order.name }}</a></td>
        <td>{{ order.order_date }}</td>
        <td>{{ order.item_count }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>

```

Gambar 29 File orders.html sebagai Template untuk Halaman List Order

```

Command Prompt - python manage.py runserver
Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User.LAPTOP-LJ8HT70G>cd inventory
C:\Users\User.LAPTOP-LJ8HT70G\inventory>venv\Scripts\activate
(venv) C:\Users\User.LAPTOP-LJ8HT70G\inventory>cd mysite
(venv) C:\Users\User.LAPTOP-LJ8HT70G\inventory\mysite>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
September 29, 2019 - 21:57:28
Django version 2.0.13, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

Gambar 30 Aktivasi Sistem Informasi dengan *Virtual Environment*