

# Dokumentasi Tes Junior Programmer

## Fast Print Indonesia

\*Note: Perangkat saya gunakan sudah terpasang python berikut *Virtual Environment*, dokumentasi dimulai dari membuat project django.

### 1. Mengaktifkan *Virtual Environment*

```
PS D:\> cd latihan
PS D:\latihan> baru\Scripts\activate
(baru) PS D:\latihan> |
```






- Mengakses direktori dimana VE disimpan disini saya menyimpannya di direktori D:\latihan.
- Mengaktifkan VE dengan cara mengakses file activate, setelah sudah diakses maka tanda bahwa ia sudah aktif adalah terdapat nama VE sebelum direktori di terminal, disini saya menamakan VE dengan nama (baru).

### 2. Membuat project django

Setelah kita mengaktifkan VE barulah kita membuat project django dengan perintah sebagai berikut:

```
(baru) PS D:\> django-admin startproject fastprint
```

Setelah perintah tersebut berhasil dieksekusi maka di dalam direktori kita akan terdapat folder baru dengan nama sesuai dengan yang kita deklarasikan di perintah diatas, disini saya menamakan folder dengan nama “fastprint”, adapun isi dari folder project django tersebut berisi satu folder dan satu file python bernama manage.py, adapun detail isi dari folder project django dapat di lihat pada gambar berikut:

Name	Date modified	Type	Size
 <code>__init__.py</code>	21/09/2023 18:54	PY File	0 KB
 <code>asgi.py</code>	21/09/2023 18:54	PY File	1 KB
 <code>settings.py</code>	21/09/2023 18:54	PY File	4 KB
 <code>urls.py</code>	21/09/2023 18:54	PY File	1 KB
 <code>wsgi.py</code>	21/09/2023 18:54	PY File	1 KB

### 3. Membuat app untuk project django

Setelah kita membuat project django, selanjutnya kita membuat app django yaitu komponen yang lebih kecil yang digunakan untuk mengorganisasikan pengkodean aplikasi

agar lebih rapih dan sesuai dengan fungsionalitasnya masing-masing. Adapun perintah untuk membuat app di django adalah sebagai berikut:

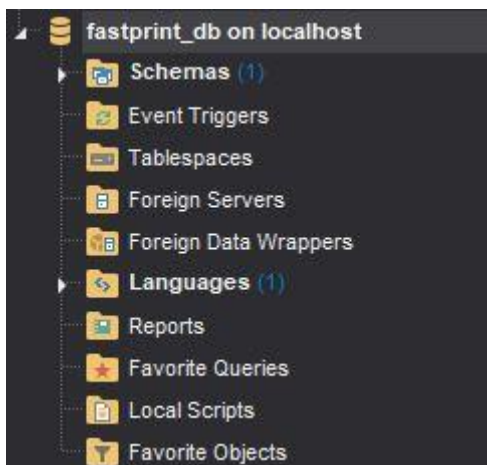
```
(baru) PS D:\> django-admin startapp fastprint_app
```

Setelah perintah berhasil dieksekusi maka akan terdapat folder app django, adapun isi dari app django dapat dilihat pada gambar berikut:

Name	Date modified	Type	Size
migrations	21/09/2023 19:13	File folder	
__init__.py	21/09/2023 19:13	PY File	0 KB
admin.py	21/09/2023 19:13	PY File	1 KB
apps.py	21/09/2023 19:13	PY File	1 KB
models.py	21/09/2023 19:13	PY File	1 KB
tests.py	21/09/2023 19:13	PY File	1 KB
views.py	21/09/2023 19:13	PY File	1 KB

#### 4. Membuat database

Disini saya membuat database dengan nama “fastprint\_db” dengan menggunakan database PostgreSQL dengan Software GUI untuk mengelola database SQL Manager for PostgreSQL, adapun tampilan dari database dapat dilihat pada gambar berikut:

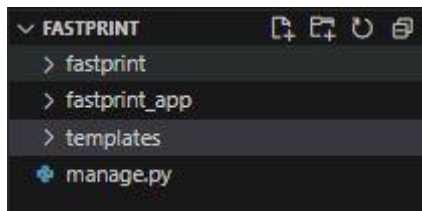


#### 5. Mengatur file settings.py di direktori project

- Mendeklarasikan direktori app di installed app agar project dapat dikenali project django, adapun source code nya dapat di lihat sebagai berikut:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'fastprint_app', #mendeklarasikan app di installed app agar app bisa dikenali oleh project
]
```

- Membuat direktori baru bernama templates, sehingga bentuk folder project django menjadi seperti berikut:



- Menginisiasi folder templates tersebut di file settings.py, source code sebagai berikut:

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')], #mendeklarasikan temp
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

- Menginisiasi database postgresSQL yang sudah di buat

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql', #menginisiasi engine postgre sql
        'NAME': 'fastprint_db', #menginisiasi database fastprint_db yang sudah dibuat
        'USER': 'postgres', #menginisiasi username database postgresql kita
        'PASSWORD': 'postgres', #menginisiasi password database postgresql kita
        'HOST': 'localhost', #menginisiasi host
        'PORT': 5432, #ini adalah port default bawaan postgre sql
    }
}
```

## 6. Membuat file urls.py di direktori fastprint\_app

Isi dari file urls.py di direktori app adalah sebagai berikut:

```
from django.urls import path #mengimport library django.urls untuk pattern url di app

app_name = 'app' #membuat app name untuk urls app agar lebih mudah dikenali
urlpatterns = [
    ]
```

## 7. Menginisiasi url app di url project

```
from django.contrib import admin
from django.urls import path, include #mengimport modul include untuk mengambil url patterns dari direktori lain
from fastprint_app import urls as dashboard #mengimport url patterns dari direktori app dan mengannti nama nya dengan alias 'dashboard'

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(dashboard)), #url '' (kosong) yang berfungsi ketika program di run maka ia adalah path url pertama yang akan dieksekusi
]
```

## 8. Membuat tabel di file models.py direktori fastprint\_app

```
from django.db import models

# Create your models here.

#Model kategori
class Kategori(models.Model):
    id_kategori = models.AutoField(primary_key=True, unique=True)
    nama_kategori = models.TextField(null=True)

class Status(models.Model):
    id_status = models.AutoField(primary_key=True, unique=True)
    nama_status = models.TextField(null=True)

class Produk(models.Model):
    id_produk = models.AutoField(primary_key=True, unique=True)
    nama_produk = models.TextField(null=True)
    harga = models.IntegerField(null=True)
    kategori = models.ForeignKey(Kategori, on_delete=models.PROTECT)
    status = models.ForeignKey(Status, on_delete=models.PROTECT)
```

Setelah mendeklarasikan model langkah selanjutnya adalah membuat perintah **py manage.py makemigrations** dan **py manage.py migrate** adapun detail dapat dilihat pada gambar berikut:



```
(baru) PS D:\fastprint> py manage.py makemigrations
Migrations for 'fastprint_app':
  fastprint_app\migrations\0001_initial.py
    - Create model Kategori
    - Create model Status
    - Create model Produk
(baru) PS D:\fastprint> py manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, fastprint_app, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying fastprint_app.0001_initial... OK
  Applying sessions.0001_initial... OK
```

## 9. Membuat views dan urls untuk mengambil data dari api

- Views.py

```
from django.shortcuts import render #modul render digunakan untuk mengembalikan render html
dengan data dictionary yang akan dikirimkan
import requests #modul request digunakan untuk membuat permintaan ke server external
import hashlib #modul ini digunakan untuk hashing string
from django.http import JsonResponse #modul jsonresponse digunakan untuk mengembalikan data
berupa json
from .models import * # Mengimport model di file models.py

def get_api(request):
    if request.method == 'GET':
        return render(request, 'index.html')
    if request.method == 'POST':
        # Data yang akan dikirim dalam permintaan POST
        username = 'tesprogrammer230923C19'
        password_md5 = hashlib.md5('bisacoding-23-09-23'.encode()).hexdigest()

        # URL API eksternal
        api_url = 'https://recruitment.fastprint.co.id/tes/api_tes_programmer'

        # Data yang akan dikirim dalam permintaan POST
        data = {
            'username': username,
            'password': password_md5
        }

        try:
            # Melakukan permintaan POST ke API eksternal
```

```

response = requests.post(api_url, data=data)

# Mengambil data header dari response
headers = response.headers

# Contoh: Cetak semua header respons
print("Headers:")
for key, value in headers.items():
    print(f"{key}: {value}")
api_data = response.json()
return JsonResponse(api_data)

except Exception as e:
    # Mengembalikan pesan kesalahan jika terjadi masalah dalam permintaan
    return JsonResponse({'error': str(e)}, status=500)
else:
    # Mengembalikan pesan kesalahan jika permintaan bukan POST
    return JsonResponse({'error': 'Permintaan harus menggunakan metode POST'},
                        status=405)

```

- **Urls.py**

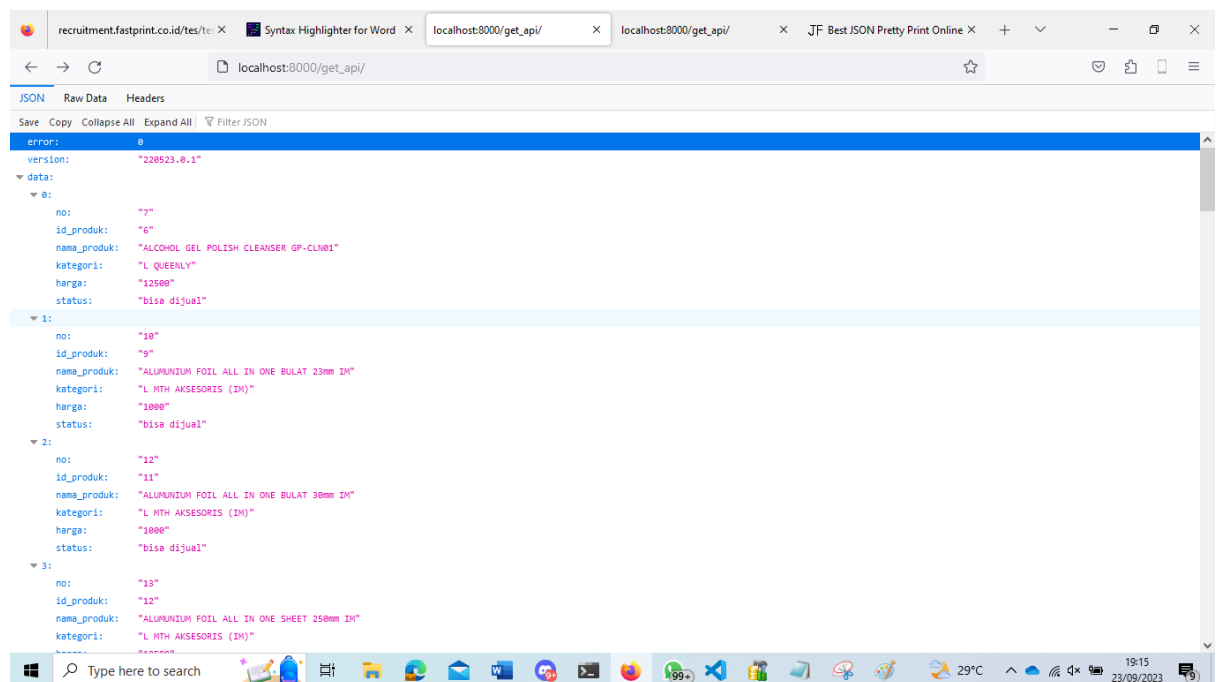
```

from django.urls import path #mengimport library django.urls untuk pattern url di app
from . import views #menginisiasi file views.py

app_name = 'app' #membuat app name untuk urls app agar lebih mudah dikenali
urlpatterns=[
    path('get_api/', views.get_api, name='get_api'), #patterns 'get_api' yang akan mengembalikan
    data dari api berupa json
]

```

- **Outputnya**



## 10. Menambahkan data produk yang sudah di ambil dari API

**\*Note:** Saya sebelumnya menambahkan data untuk kategori dan status menggunakan django rest-framework melalui Postman

- **Membuat html untuk tampilan tambah produk**

```

{% extends 'base.html' %}
{% block main %}
<div class="card-body">

```

```

<h5 class="text-center mb-5">{% if not edit %} Tambah {% else %} Ubah {% endif %} Data
Produk</h5>
<form method="post" class="text-center">
    {% csrf token %}
    <div class="form-row">
        <div class="col-4">
            <div class="form-group">
                <label for="id">Masukan ID Produk</label>
                <input type="number" name="id" class="form-control">
            </div>
        </div>
        <div class="col-4">
            <div class="form-group">
                <label for="id">Masukan Nama Produk</label>
                <input type="text" name="nama" class="form-control" required>
            </div>
        </div>
        <div class="col-4">
            <div class="form-group">
                <label for="id">Masukan Harga Produk</label>
                <input type="number" name="harga" class="form-control">
            </div>
        </div>
    </div>
    <div class="form-row justify-content-center">
        <div class="col-4">
            <div class="form-group">
                <label for="id">Pilih Kategori Produk</label>
                <select name="kategori" class="form-control">
                    <option value="" {% if not edit %} selected {% endif %}>---PILIH--
-</option>
                    {% for x in kategori %}
                        <option value="{{x.id_kategori}}">{{x.nama_kategori}}</option>
                    {% endfor %}
                </select>
            </div>
        </div>
        <div class="col-4">
            <div class="form-group">
                <label for="id">Pilih Status Produk</label>
                <select name="status" class="form-control">
                    <option value="" {% if not edit %} selected {% endif %}>---PILIH--
-</option>
                    {% for x in status %}
                        <option value="{{x.id_status}}">{{x.nama_status}}</option>
                    {% endfor %}
                </select>
            </div>
        </div>
    </div>
    <a type="button" class="btn btn-danger" href="{% url 'app:admin_produk'
%}">Cancel</a>
    <button type="submit" class="btn btn-success">Create</button>
</form>
</div>

{% endblock %}

```

- Views.py

```

def tambah(request):
    if request.method == 'GET':
        kategori = Kategori.objects.all() # Mengambil data pada model kategori
        status = Status.objects.all() # Mengambil data pada model status

        # Menyimpannya kedalam variabel dictionary bernama x dan mengimnya ke template
        x = {
            'kategori' : kategori,
            'status' : status,
        }
        return render(request, 'create.html', x)

    if request.method == 'POST':

```

```

id = request.POST.get('id')
nama = request.POST.get('nama')
harga = int(request.POST.get('harga'))
kategori = request.POST.get('kategori')
dt_kat = Kategori.objects.get(id_kategori = kategori)
status = request.POST.get('status')
dt_status = Status.objects.get(id_status = status)
try:
    produk = Produk(
        id_produk = id,
        nama_produk = nama,
        harga = harga,
        kategori = dt_kat,
        status = dt_status
    )
    produk.save()
except Exception as f:
    print(f)
return redirect('app:admin_produk')

```

- Urls.py

```
path('tambah-produk/', views.tambah, name='tambah_produk'),
```

- Tampilan

**Tambah Data Produk**

Masukan ID Produk

Masukan Nama Produk

Masukan Harga Produk

Pilih Kategori Produk  
---PILIH---

Pilih Status Produk  
---PILIH---

Cancel
Create

## 11. Menampilkan semua data yang berstatus “bisa dijual”

- Source code html

```

{% extends 'base.html' %} <!-- mengekstend base.html -->
{% load custom_tags %}
{% block main %}
    <div class="card-body">
        <h4>Daftar Produk</h4>
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">ID Produk</th>
                    <th scope="col">Nama Produk</th>
                    <th scope="col">Harga</th>
                    <th scope="col">Kategori</th>
                    <th scope="col">Status</th>
                    <th scope="col">Aksi</th>
                </tr>
            </thead>
            <tbody>
                {% for x in data %}
                <tr>
                    <td>{{ x.id_produk }}</td>
                    <td>{{ x.nama_produk }}</td>
                    <td>{{ x.harga|currency }}</td>
                    <td>{{ x.kategori.nama_kategori }}</td>
                    <td>{{ x.status.nama_status }}</td>
                    <td>
                        <a type="button" class="btn btn-warning" href="">Update</a>
                        <a type="button" class="btn btn-danger" href="">Delete</a>
                    </td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
{% endblock %}

```



```

</tbody>
</table>
<div class="row">
    <a type="button" class="btn btn-success" href="{% url 'app:tambah produk' %}"
style="margin-right: 5px;">Create</a>
    <form action="{% url 'app:get_api' %}" method="post">
        {% csrf_token %}
        <button type="submit" class="btn btn-primary">Get API Data</button>
    </form>
</div>
</div>
{% endblock %}

```

- Source code Views.py

```

def produk(request):
    if request.method == 'GET':
        data = Produk.objects.filter(status_id_nama_status = 'bisa dijual') # Menampilkan data dari tabel produk dengan filter nama status
        x = {
            'data': data,
        }
        return render(request, 'dashboard.html', x)

```

- Source code urls.py

```

path('', views.produk, name='admin_produk'),

```

- Tampilan website

ID Produk	Nama Produk	Harga	Kategori	Status	Aksi
6	ALCOHOL GEL POLISH CLEANSER GP-CLN01	Rp12.500,00	L QUEENLY	bisa dijual	Update Delete
9	ALUMUNIUM FOIL ALL IN ONE BULAT 23mm IM	Rp1.000,00	L MTH AKSESORIS (IM)	bisa dijual	Update Delete
11	ALUMUNIUM FOIL ALL IN ONE BULAT 30mm IM	Rp1.000,00	L MTH AKSESORIS (IM)	bisa dijual	Update Delete
15	ALUMUNIUM FOIL HDPE/PE BULAT 23mm IM	Rp12.500,00	L MTH AKSESORIS (IM)	bisa dijual	Update Delete
17	ALUMUNIUM FOIL HDPE/PE BULAT 30mm IM	Rp1.000,00	L MTH AKSESORIS (IM)	bisa dijual	Update Delete

## 12. Mengubah data

Disini saya akan membuat data dummy yang nanti nya digunakan untuk update data dan delete data.

- Source code html

\*file html yang digunakan update data sama dengan file html create data

```

{% extends 'base.html' %}
{% block main %}
    <div class="card-body">

```

```

        <h5 class="text-center mb-5">{% if not edit %} Tambah {% else %} Ubah {% endif %}
Data Produk</h5>
        <form method="post" class="text-center"
        {% if not edit %} action="{% url 'app:tambah_produk' %}" {% else %} action="{% url
'app:ubah_produk' data.id_produk %}" {% endif %}>
            {% csrf_token %}
            <div class="form-row">
                <div class="col-4">
                    <div class="form-group">
                        <label for="id">Masukan ID Produk</label>
                        <input type="number" name="id" class="form-control" {% if edit %}
disabled {% endif %} value="{{data.id_produk}}">
                    </div>
                </div>
                <div class="col-4">
                    <div class="form-group">
                        <label for="id">Masukan Nama Produk</label>
                        <input type="text" name="nama" class="form-control" required
value="{{data.nama_produk}}">
                    </div>
                </div>
                <div class="col-4">
                    <div class="form-group">
                        <label for="id">Masukan Harga Produk</label>
                        <input type="number" name="harga" class="form-control"
value="{{data.harga}}">
                    </div>
                </div>
            </div>
            <div class="form-row justify-content-center">
                <div class="col-4">
                    <div class="form-group">
                        <label for="id">Pilih Kategori Produk</label>
                        <select name="kategori" class="form-control">
                            {% if not edit %}
                                <option value="" {% if not edit %} selected {% endif %}>---
PILIH---</option>
                                {% else %}
                                    <option
value="{{data.kategori.id_kategori}}">{{data.kategori.nama_kategori}}</option>
                                {% endif %}
                                {% for x in kategori %}
                                    <option
value="{{x.id_kategori}}">{{x.nama_kategori}}</option>
                                {% endfor %}
                            </select>
                        </div>
                    </div>
                    <div class="col-4">
                        <div class="form-group">
                            <label for="id">Pilih Status Produk</label>
                            <select name="status" class="form-control">
                                {% if not edit %}
                                    <option value="" {% if not edit %} selected {% endif %}>---
PILIH---</option>
                                {% else %}
                                    <option
value="{{data.status.id_status}}">{{data.status.nama_status}}</option>
                                {% endif %}
                                {% for x in status %}
                                    <option value="{{x.id_status}}">{{x.nama_status}}</option>
                                {% endfor %}
                            </select>
                        </div>
                    </div>
                </div>
            </div>
            <a type="button" class="btn btn-danger" href="{% url 'app:admin_produk'
%}">Cancel</a>
            <button type="submit" class="btn btn-success">{% if not edit %} Create {% else %}
Update {% endif %}</button>
        </form>
    </div>
{% endblock %}

```

- Source Code function untuk rubah data

```
def ubah(request, id):
    if request.method == 'GET':
        kategori = Kategori.objects.all() # Mengambil data pada model kategori
        status = Status.objects.all() # Mengambil data pada model status
        data = Produk.objects.get(id_produk = id) # Mengambil data yang id produk nya sama
        dengan id yang dikirimkan

        # Menyimpangkannya kedalam variabel dictionary bernama x dan mengimnya ke template
        x = {
            'kategori' : kategori,
            'status' : status,
            'data' : data,
            'edit' : 'true',
        }
        return render(request, 'create.html', x)

    if request.method == 'POST':
        # Mengambil semua name yang ada di inputan html
        # id = request.POST.get('id')
        nama = request.POST.get('nama')
        harga = int(request.POST.get('harga'))

        # Mengambil data dari model kategori
        kategori = request.POST.get('kategori')
        dt_kat = Kategori.objects.get(id_kategori = kategori)

        # Mengambil data dari model status
        status = request.POST.get('status')
        dt_status = Status.objects.get(id_status = status)

        # Menggunakan try and catch untuk menyimpan data
        try:
            produk = Produk.objects.get(id_produk = id)
            produk.nama_produk = nama
            produk.harga = harga
            produk.kategori = dt_kat
            produk.status = dt_status

            produk.save()
        except Exception as f:
            print(f) # Mencetak kesalahan di terminal

        return redirect('app:admin_produk')
```

- Path url

```
path('ubah-produk/<str:id>/', views.ubah, name='ubah_produk'), # Path url untuk merubah data dengan parameter id
```

- Tampilan

- Tampilan sebelum diubah

**\*Note:** untuk form ubah data saya buat disable field inputan id\_produk

Ubah Data Produk

Masukan ID Produk

Masukan Nama Produk

Masukan Harga Produk

100

Dummy Data

1000

Pilih Kategori Produk

Pilih Status Produk

S MTH STEPEL (IM)

bisa dijual

Cancel

Update

- Tampilan setelah diubah

### 13. Hapus Data

Saya menggunakan sweetlert sebagai alert confirm untuk menghapus data. Adapun source code html dan js berikut:

- Html

```
<button class="btn btn-danger text-white" data-url="{% url 'app:hapus_produk' x.id_produk %}"
        data-back="{% url 'app:admin_produk' %}"
        onclick="delete_data(this,
        '{{x.nama_produk}}')">Delete</button>
```

- Javascript

```
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
<script>
function delete_data(e, categoryName) {
    Swal.fire({
        title: 'Hapus Produk',
        text: `Apakah Kamu Ingin menghapus "${categoryName}"?`,
        icon: 'warning',
        showCancelButton: true,
        confirmButtonText: 'Yes',
        cancelButtonText: 'No',
        buttonsStyling: false,
        customClass: {
            confirmButton: 'btn btn-danger',
            cancelButton: 'btn btn-primary',
        },
    }).then(function (result) {
        if (result.isConfirmed) {
            $.ajax({
                url: $(e).attr('data-url'),
                data: {},
                dataType: "json",
                timeout: 10000,
                success: function(response){
                    if(response.message == 'error'){
                        Swal.fire({
                            text: data.error_message,
                            icon: "error",
                            buttonsStyling: false,
                            confirmButtonText: "Ok, mengerti!",
                            customClass: { confirmButton: "btn btn-primary" }
                        });
                    }else{
                        Swal.fire({
                            text: "Data anda berhasil dihapus!",
                            icon: "success",
                            buttonsStyling: false,
                            confirmButtonText: "Ok, mengerti!",
                            customClass: { confirmButton: "btn btn-primary" }
                        }).then(function (result) {
                            if (result.isConfirmed) {
                                window.location.reload(true);
                            }
                        });
                    }
                },
            },
            error: function (response) {
                console.log(response);
                Swal.fire('Arsip data gagal dilakukan!', '', 'error');
                setTimeout(function(){
                    window.location.href = $(e).attr('data-back');
                }, 2000);
            }
        )
    });
}
```

```
});
</script>
```

- Function untuk hapus data

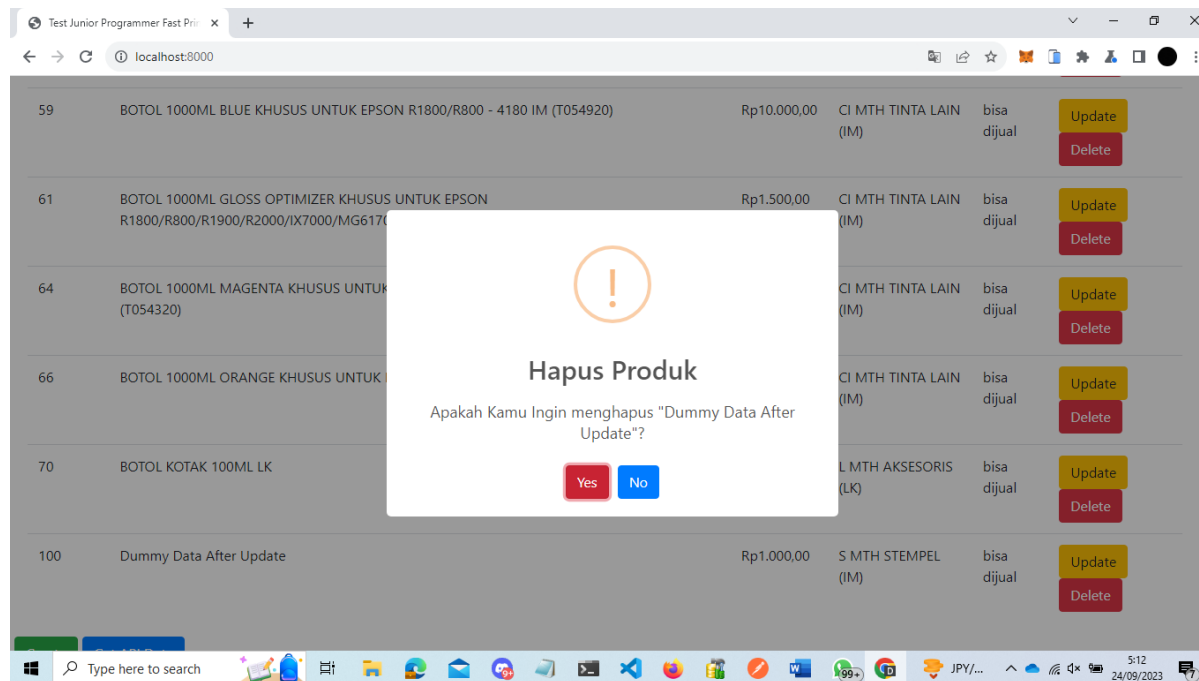
```
def delete(request, id):
    try:
        data = Produk.objects.get(id_produk = id) # Mengambil data berdasarkan parameter url yg dikirimkan
        data.delete() # Menghapus data
        return JsonResponse({'success': True}) # Mengembalikan respon ke alert True
    except Exception as f:
        error_message = "Terjadi kesalahan karena " + str(f) # Digunakan untuk memberikan pesan jika terjadi kesalahan saat menghapus data
        return JsonResponse({'success': False, 'error_message': error_message})
```

- Path url untuk menghapus data

```
path('hapus-produk/<str:id>/', views.delete, name='hapus_produk'), # Path url untuk menghapus data dengan parameter id
```

- Tampilan

- Alert konfirmasi untuk menghapus



- Tampilan setelah data dihapus

Test Junior Programmer Fast Pri x +

localhost:8000

			SPAREPART (LK)	dijual	<a href="#">Update</a> <a href="#">Delete</a>
59	BOTOL 1000ML BLUE KHUSUS UNTUK EPSON R1800/R800 - 4180 IM (T054920)	Rp10.000,00	CI MTH TINTA LAIN (IM)	bisa dijual	<a href="#">Update</a> <a href="#">Delete</a>
61	BOTOL 1000ML GLOSS OPTIMIZER KHUSUS UNTUK EPSON R1800/R800/R1900/R2000/IX7000/MG6170 - 4100 IM (T054020)	Rp1.500,00	CI MTH TINTA LAIN (IM)	bisa dijual	<a href="#">Update</a> <a href="#">Delete</a>
64	BOTOL 1000ML MAGENTA KHUSUS UNTUK EPSON R1800/R800/R1900/R2000 - 4140 IM (T054320)	Rp1.000,00	CI MTH TINTA LAIN (IM)	bisa dijual	<a href="#">Update</a> <a href="#">Delete</a>
66	BOTOL 1000ML ORANGE KHUSUS UNTUK EPSON R1900/R2000 IM - 4190 (T087920)	Rp1.500,00	CI MTH TINTA LAIN (IM)	bisa dijual	<a href="#">Update</a> <a href="#">Delete</a>
70	BOTOL KOTAK 100ML LK	Rp1.000,00	L MTH AKSESORIS (LK)	bisa dijual	<a href="#">Update</a> <a href="#">Delete</a>

[Create](#) [Get API Data](#)

Type here to search

18°C 5:14 24/09/2023

Sekian Dokumentasi saya untuk test junior programmer di PT Fastprint Indonesia, kurang lebihnya mohon maaf jika ada yang kurang jelas maupun kesalahan. Terimakasih atas perhatiannya.

Regards,

Gema Putranto Sudibyo