

SESSION 8

Flask Dashboard

Create Dashboard

Outline

- Create an initial dashboard
- Create the content of Category (Histogram & Boxplot) menu
- Create the content of Scatter Plot menu
- Create the content of Pie Chart menu

SESSION 8

Flask Dashboard

Create an Initial Dashboard

INITIAL DASHBOARD (app.py)

First thing that should be created is the API. Open 'app.py' in code editor. Write the code below to create initial API using Flask module.

```
# Flask : library utama untuk membuat API
# render_template : agar dapat memberikan respon file html
# request : untuk membaca data yang diterima saat request datang
from flask import Flask, render_template, request
# plotly dan plotly.graph_objs : membuat plot
import plotly
import plotly.graph_objs as go
# pandas : untuk membaca csv dan men-generate dataframe
import pandas as pd
import json

# untuk membuat route
app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True)
```

This API file will be connected to 'index.html' page. In this page, there will be nothing happened because there is no other command connect to 'index.html' and to run this file

INITIAL DASHBOARD (index.html)

Next, fill 'index.html' page with the code below. This code linked to styling file that use bootstrap (bootstrap.css). By using this code, home page of the dashboard will be more responsive.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>DASHBOARD</title>
  <!-- agar dapat menggunakan class bootstrap secara offline -->
  <link rel="stylesheet" href="../static/bootstrap.css">

  <!-- untuk dapat memunculkan chart dari plotly -->
  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.6/d3.min.js"></script>

  <!-- Untuk script update database -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
  <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
  <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head>
<body>
```

INITIAL DASHBOARD (index.html)

Still in the same file, write the code below exactly after the first code. This code is the home page preview. As we can see, there is the dashboard title namely 'TIPS Dashboard'. Also, there are 3 menu that available. They are 'Histogram & Plot', 'Scatter', and 'Pie'. Later, this menu will be made.

```
<body>

  <div class="container">
    <!-- Judul Dashboard -->
    <a class="text-decoration-none text-secondary" href="{{url_for('index')}}">
      <h1 class="text-center text-capitalize display-4 my-5">
        TIPS Dashboard
      </h1>
    </a>
    <!-- membuat menu navigasi untuk pindah halaman / tampilan plot -->
    <ul class="nav nav-pills nav-fill">
      <li class="nav-item font-weight-bold">
        <a class="nav-link text-dark lead" href="">Histogram & Box</a>
      </li>
      <li class="nav-item font-weight-bold">
        <a class="nav-link text-dark lead" href="">Scatter</a>
      </li>
      <li class="nav-item font-weight-bold">
        <a class="nav-link text-dark lead" href="">Pie</a>
      </li>
    </ul>

    {% block content %}

    {% endblock %}
  </div>

  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
  integrity="sha384-DfxDz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
  crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
  integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
  crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
  integrity="sha384-0gVRuATP1z7JjHLku0U7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
  crossorigin="anonymous"></script>

</body>
</html>
```

INITIAL DASHBOARD PREVIEW

To show the initial dashboard, run 'app.py' in terminal or command prompt. After that, open <http://localhost:5000> in browser.

TIPS Dashboard

Histogram & Box

Scatter

Pie

SESSION 8

Flask Dashboard

Create Home and Histogram & Box Menu

HOME PAGE (app.py)

```
# category plot function
def category_plot(
    cat_plot = 'histplot',
    cat_x = 'sex', cat_y = 'total_bill',
    estimator = 'count', hue = 'smoker'):

    # generate dataframe tips.csv
    tips = pd.read_csv('./static/tips.csv')

    # jika menu yang dipilih adalah histogram
    if cat_plot == 'histplot':
        # siapkan list kosong untuk menampung konfigurasi hist
        data = []
        # generate config histogram dengan mengatur sumbu x dan sumbu y
        for val in tips[hue].unique():
            hist = go.Histogram(
                x=tips[tips[hue]==val][cat_x],
                y=tips[tips[hue]==val][cat_y],
                histfunc=estimator,
                name=val
            )
            #masukkan ke dalam array
            data.append(hist)
        #tentukan title dari plot yang akan ditampilkan
        title='Histogram'
    elif cat_plot == 'boxplot':
        data = []

        for val in tips[hue].unique():
            box = go.Box(
                x=tips[tips[hue] == val][cat_x], #series
                y=tips[tips[hue] == val][cat_y],
                name = val
            )
            data.append(box)
        title='Box'
```

Define the function for category plot. Category plot itself has two part inside. First one is histogram plot, and the second one is boxplot. So the category plot function will include this two plot. Here is the code

HOME PAGE (app.py)

Exactly after the code before, write a code to define which one is x axis and y axis. This code also still in the same function like before, and also still divided into 2 part, they are histogram and box plot. The code is showed beside.

```
title = box
# menyiapkan config layout tempat plot akan ditampilkan
# menentukan nama sumbu x dan sumbu y
if cat_plot == 'histplot':
    layout = go.Layout(
        title=title,
        xaxis=dict(title=cat_x),
        yaxis=dict(title='person'),
        # boxmode group digunakan berfungsi untuk mengelompokkan box berdasarkan hue
        boxmode = 'group'
    )
else:
    layout = go.Layout(
        title=title,
        xaxis=dict(title=cat_x),
        yaxis=dict(title=cat_y),
        # boxmode group digunakan berfungsi untuk mengelompokkan box berdasarkan hue
        boxmode = 'group'
    )
#simpan config plot dan layout pada dictionary
result = {'data': data, 'layout': layout}

#json.dumps akan mengenerate plot dan menyimpan hasilnya pada graphjson
graphJSON = json.dumps(result, cls=plotly.utils.PlotlyJSONEncoder)

return graphJSON
```

HOME PAGE (app.py)

After category plot function made, next is to make router for landing page (home page). This router actually exactly same as router's category that will be made soon. This router has function to show the plot in category plot with default plot showed is histogram, X axis is 'sex' variable, Y axis disable (if plot menu doesn't change to boxplot), estimator is 'count' variable, and default for hue is 'smoker' variable

```
@app.route('/')
def index():

    plot = category_plot()
    list_plot = [('histplot', 'Histogram'), ('boxplot', 'Box')]
    list_x = [('sex', 'Sex'), ('smoker', 'Smoker'), ('day', 'Day'), ('time', 'Time')]
    list_y = [('total_bill', 'Bill'), ('tip', 'Tip'), ('size', 'Size')]
    list_est = [('count', 'Count'), ('avg', 'Average'), ('max', 'Max'), ('min', 'Min')]
    list_hue = [('sex', 'Sex'), ('smoker', 'Smoker'), ('day', 'Day'), ('time', 'Time')]

    return render_template(
        # file yang akan menjadi response dari API
        'category.html',
        # plot yang akan ditampilkan
        plot=plot,
        # menu yang akan tampil di dropdown 'Jenis Plot'
        focus_plot='histplot',
        # menu yang akan muncul di dropdown 'sumbu X'
        focus_x='sex',

        # menu yang akan muncul di dropdown 'Estimator'
        focus_estimator='count',
        # menu yang akan tampil di dropdown 'Hue'
        focus_hue='smoker',
        # list yang akan digunakan looping untuk membuat dropdown 'Jenis Plot'
        drop_plot= list_plot,
        # list yang akan digunakan looping untuk membuat dropdown 'Sumbu X'
        drop_x= list_x,
        # list yang akan digunakan looping untuk membuat dropdown 'Sumbu Y'
        drop_y= list_y,
        # list yang akan digunakan looping untuk membuat dropdown 'Estimator'
        drop_estimator= list_est,
        # list yang akan digunakan looping untuk membuat dropdown 'Hue'
        drop_hue= list_hue)
```

HOME PAGE (app.py)

Category plot router is exactly same as home page router before. There is only one addition before write the code just like in home page router. Here is the code before adding the same code like home page router

```
@app.route('/cat_fn/<nav>')
def cat_fn(nav):

    # saat klik menu navigasi
    if nav == 'True':
        cat_plot = 'histplot'
        cat_x = 'sex'
        cat_y = 'total_bill'
        estimator = 'count'
        hue = 'smoker'

    # saat memilih value dari form
    else:
        cat_plot = request.args.get('cat_plot')
        cat_x = request.args.get('cat_x')
        cat_y = request.args.get('cat_y')
        estimator = request.args.get('estimator')
        hue = request.args.get('hue')

    # Dari boxplot ke histogram akan None
    if estimator == None:
        estimator = 'count'

    # Saat estimator == 'count', dropdown menu sumbu Y menjadi disabled dan memberikan nilai None
    if cat_y == None:
        cat_y = 'total_bill'
```

After write the code beside, copy and paste the code in home page router. The code that pasted still in the same function with the code before

FULL CODE OF CATEGORY PLOT ROUTER

```
@app.route('/cat_fn/<nav>')
def cat_fn(nav):

    # saat klik menu navigasi
    if nav == 'True':
        cat_plot = 'histplot'
        cat_x = 'sex'
        cat_y = 'total_bill'
        estimator = 'count'
        hue = 'smoker'

    # saat memilih value dari form
    else:
        cat_plot = request.args.get('cat_plot')
        cat_x = request.args.get('cat_x')
        cat_y = request.args.get('cat_y')
        estimator = request.args.get('estimator')
        hue = request.args.get('hue')

    # Dari boxplot ke histogram akan None
    if estimator == None:
        estimator = 'count'

    # Saat estimator == 'count', dropdown menu sumbu Y menjadi disabled dan memberikan nilai None
    if cat_y == None:
        cat_y = 'total_bill'

    # Dropdown menu
    list_plot = [('histplot', 'Histogram'), ('boxplot', 'Box')]
    list_x = [('sex', 'Sex'), ('smoker', 'Smoker'), ('day', 'Day'), ('time', 'Time')]
    list_y = [('total_bill', 'Bill'), ('tip', 'Tip'), ('size', 'Size')]
    list_est = [('count', 'Count'), ('avg', 'Average'), ('max', 'Max'), ('min', 'Min')]
    list_hue = [('sex', 'Sex'), ('smoker', 'Smoker'), ('day', 'Day'), ('time', 'Time')]

    plot = category_plot(cat_plot, cat_x, cat_y, estimator, hue)
    return render_template(
        # file yang akan menjadi response dari API
        'category.html',
        # plot yang akan ditampilkan
        plot=plot,
        # menu yang akan tampil di dropdown 'Jenis Plot'
        focus_plot=cat_plot,
        # menu yang akan muncul di dropdown 'sumbu X'
        focus_x=cat_x,
        focus_y=cat_y,

        # menu yang akan muncul di dropdown 'Estimator'
        focus_estimator=estimator,
        # menu yang akan tampil di dropdown 'Hue'
        focus_hue=hue,
        # list yang akan digunakan looping untuk membuat dropdown 'Jenis Plot'
        drop_plot= list_plot,
        # list yang akan digunakan looping untuk membuat dropdown 'Sumbu X'
        drop_x= list_x,
        # list yang akan digunakan looping untuk membuat dropdown 'Sumbu Y'
        drop_y= list_y,
        # list yang akan digunakan looping untuk membuat dropdown 'Estimator'
        drop_estimator= list_est,
        # list yang akan digunakan looping untuk membuat dropdown 'Hue'
        drop_hue= list_hue
    )
```

HOME PAGE (index.html)

Lets move to 'index.html' file. In this file, there is only one change. In 'href' of Histogram & Boxplot, fill with the name of category plot router. Write the code by using jinja

(`href="{{url_for('cat_fn', nav=True)}}"`)

```
<body>

  <div class="container">
    <!-- Judul Dashboard -->
    <a class="text-decoration-none text-secondary" href="{{url_for('index')}}">
      <h1 class="text-center text-capitalize display-4 my-5">
        TIPS Dashboard
      </h1>
    </a>
    <!-- membuat menu navigasi untuk pindah halaman / tampilan plot -->
    <ul class="nav nav-pills nav-fill">
      <li class="nav-item font-weight-bold">
        <a class="nav-link text-dark lead" href="{{url_for('cat_fn', nav=True)}}">Histogram & Box</a>
      </li>
      <li class="nav-item font-weight-bold">
        <a class="nav-link text-dark lead" href="">Scatter</a>
      </li>
      <li class="nav-item font-weight-bold">
        <a class="nav-link text-dark lead" href="">Pie</a>
      </li>
    </ul>

    {% block content %}

    {% endblock %}
  </div>
```

HOME PAGE (category.html)

In the beginning of this page ('category.html'), the first thing that should be made is create an extends and block content. This code is to connect from 'index.html' file and continue the code start from the place where the code was written.

```
</h1>
</a>
<!-- membuat menu navigasi untuk pindah halaman / tampilan plot -->
<ul class="nav nav-pills nav-fill">
  <li class="nav-item font-weight-bold">
    <a class="nav-link text-dark lead" href="{{url_for('cat_fn', nav=True)}}">Histogram & Box</a>
  </li>
  <li class="nav-item font-weight-bold">
    <a class="nav-link text-dark lead" href="{{url_for('scatt_fn')}}">Scatter</a>
  </li>
  <li class="nav-item font-weight-bold">
    <a class="nav-link text-dark lead" href="{{url_for('pie_fn')}}">Pie</a>
  </li>
</ul>

{% block content %}

{% endblock %}
</div>
```

From 'index.html'

```
{% extends 'index.html' %}

{% block content %}
```

From 'category.html'

HOME PAGE (category.html)

This page will be divided into some parts, depends on how many drop down menu that created. The first one is to create drop down menu for the plot type. Here is the html code.

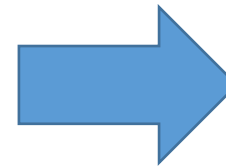
```
{% extends 'index.html' %}

{% block content %}

<!-- Menu Dropdown -->
<form action="{{url_for('cat_fn', nav=False)}}" id="form">
  <div class="my-5 row d-flex justify-content-around">
    <!-- Dropdwon Jenis Plot -->

    <!-- Menu dropdown ini akan menampilkan list menu jenis-jenis plot yang ada -->
    <!-- Di sini kita akan melakukan looping terhadap list yang dikirim dari API -->
    <!-- list tersebut adalah drop_plot -->

    <div class="col-2">
      <p class="text-center lead">Plot</p>
      <select class="form-control" name="cat_plot" onchange="form.submit()">
        {% for drop in drop_plot %}
          {% if focus_plot == drop[0] %}
            return '<option value={{drop[0]}} selected>{{drop[1]}}</option>'
          {% else %}
            return '<option value={{drop[0]}}>{{drop[1]}}</option>'
          {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
</form>
```



Plot

Histogram

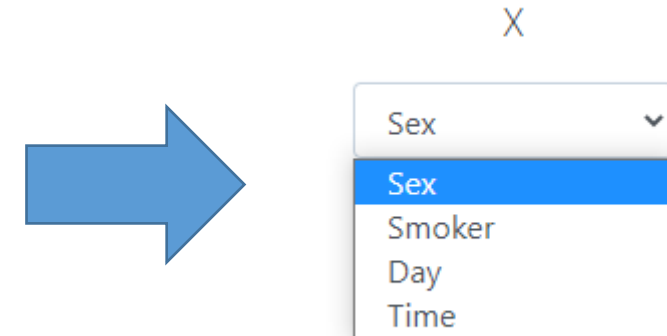
Histogram

Box

HOME PAGE (category.html)

The next code is to create X axis drop down menu. In this part, value of 'drop_x' from 'app.py' will be looped. Every element of 'drop_x' variable in API file will be showed in X axis drop down menu. Below the code and output

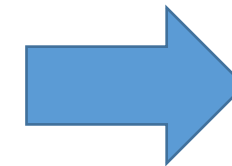
```
<!-- Dropdwon Sumbu X -->
<div class="col-2">
  <p class="text-center lead">X</p>
  <select class="form-control" name="cat_x" onchange="form.submit()">
    {% for drop in drop_x %}
      {% if focus_x == drop[0] %}
        return '<option value={{drop[0]}} selected>{{drop[1]}}</option>'
      {% else %}
        return '<option value={{drop[0]}}>{{drop[1]}}</option>'
      {% endif %}
    {% endfor %}
  </select>
</div>
```



HOME PAGE (category.html)

Not like X axis drop down menu, Y axis drop down menu initial form is disabled. This is because initial plot from plot drop down menu is 'Histogram' that doesn't need Y axis. If the plot changed into 'Box', Y axis drop down menu will active and show the value of 'drop_y' from 'app.py' files

```
<!-- Dropdwon Sumbu Y -->
<div class="col-2">
  <p class="text-center lead">Y</p>
  {% if focus_estimator == 'count' and focus_plot == 'histplot' %}
    <select class="form-control" name="cat_y" disabled>
      <option>Disable</option>
    </select>
  {% else %}
    <select class="form-control" name="cat_y" onchange="form.submit()">
      {% for drop in drop_y %}
        {% if focus_y == drop[0] %}
          return '<option value={{drop[0]}} selected>{{drop[1]}}</option>'
        {% else %}
          return '<option value={{drop[0]}}>{{drop[1]}}</option>'
        {% endif %}
      {% endfor %}
    </select>
  {% endif %}
</div>
```



Y

Disable ▾

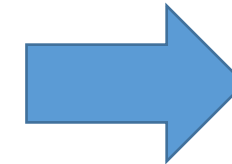
Y

Bill ▾
Bill
Tip
Size

HOME PAGE (category.html)

Next is Estimator drop down menu. This menu just like Y axis drop down menu. If Y axis will disabled if plot chosen is Histogram, Estimator menu will disabled if plot chosen is Box plot and will show value 'drop_estimator' value if the plot is Histogram. Below the code.

```
<!-- Dropdwon Estimator -->
<div class="col-2">
  <p class="text-center lead">Estimator</p>
  {% if focus_plot == 'boxplot' %}
    <select class="form-control" name="estimator" disabled onchange="form.submit()">
      <option value='count' selected>Disable</option>
    </select>
  {% else %}
    <select class="form-control" name="estimator" onchange="form.submit()">
      {% for drop in drop_estimator %}
        {% if focus_estimator == drop[0] %}
          return '<option value={{drop[0]}} selected>{{drop[1]}}</option>'
        {% else %}
          return '<option value={{drop[0]}}>{{drop[1]}}</option>'
        {% endif %}
      {% endfor %}
    </select>
  {% endif %}
</div>
```



Estimator

Count ▾
Count
Average
Max
Min

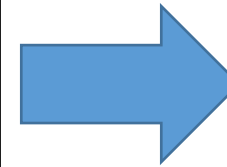
Estimator

Disable ▾

HOME PAGE (category.html)

Last drop down menu is Hue. This menu will active every time both when plot value is Histogram or even Box plot. Options that showed in this menu are from the value in 'drop_hue' variable in 'app.py' file. Here is the code for Hue drop down menu

```
<!-- Dropdwon Hue -->
<div class="col-2">
  <p class="text-center lead">Hue</p>
  <select class="form-control" name="hue" onchange="form.submit()">
    {% for drop in drop_hue %}
      {% if focus_hue == drop[0] %}
        return '<option value={{drop[0]}} selected>{{drop[1]}}</option>'
      {% else %}
        return '<option value={{drop[0]}}>{{drop[1]}}</option>'
      {% endif %}
    {% endfor %}
  </select>
</div>
</form>
```



Hue

Smoker

▼

Sex

Smoker

Day

Time

HOME PAGE (category.html)

Last of this Category page is to show the graph and endblock content. Below the code of this part

```
<!-- Chart / Plot -->
<div class="chart" id="plot">
    <script>
        var graphs = {{plot | safe}};
        Plotly.plot('plot',graphs,{});
    </script>
</div>
{% endblock content %}
```

DASHBOARD CATEGORY PREVIEW

To show the dashboard, run 'app.py' in terminal or command prompt. After that, open <http://localhost:5000> in browser.

