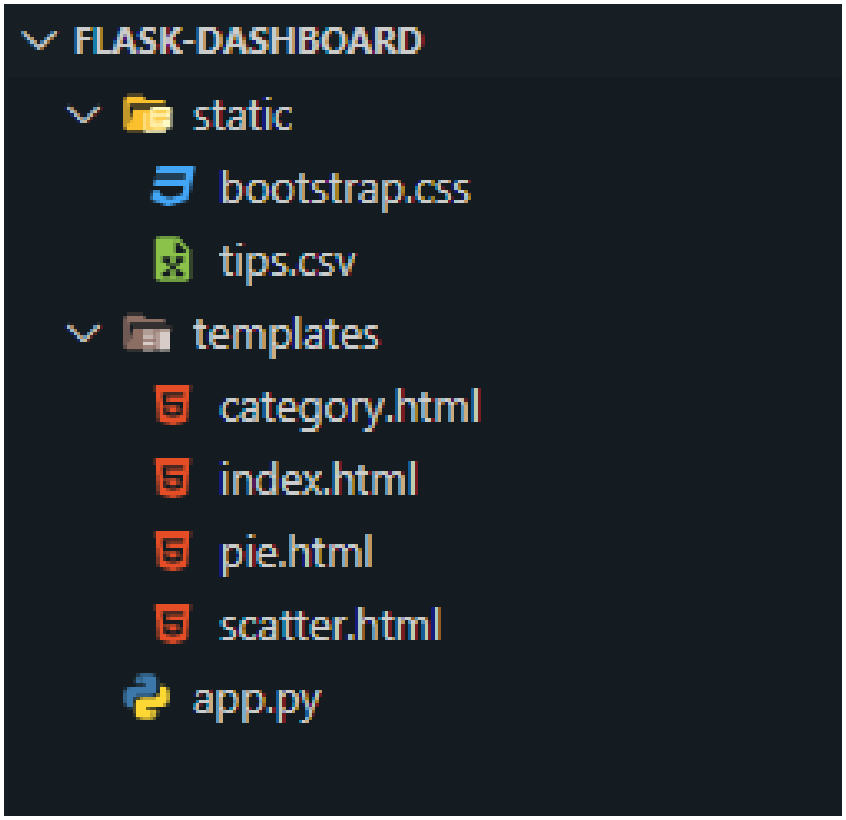
Module 01

# Flask Dashboard

Data Science Developer

Purwadhika
Startup and Coding School

# Menyiapkan folder dan files




Purwadhika
Startup and Coding School

# Flask dasar (app.py)

```python
# Flask : library utama untuk membuat API
# render_template : agar dapat memberikan respon file html
# reques : 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 file csv dan generate data frame
import pandas as pd
import json

# untuk membuat route
app = Flask(__name__)

# akses halaman menuju route '/' untuk men-test
# apakah API sudah running
@app.route('/')
def index():
    return render_template('index.html')

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

Purwadhika
Startup and Coding School

# Index.html

```html
<!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 -->
  <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>
</head>
<body>

  <h1>MPG DASHBOARD</h1>

  <!-- agar dapat menggunakan class bootstrap -->
  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
  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/4.4.1/js/bootstrap.min.js"
  integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Bootstrap 4
https://getbootstrap.com/docs/4.0/getting-started/introduction/

Cdn plotly
https://blog.heptanalytics.com/flask-plotly-dashboard/

Purwadhika
Startup and Coding School

# Index.html

```
<div class="container">
    <!-- judul dashboard -->
    <a class="text-decoration-none text-secondary " href="">
        <h1 class="text-center text-capitalize display-4 my-5">MPG Dashboard</h1>
    </a>

    <!-- membuat menu navigasi untuk berpindah 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>
```

Purwadhika
Startup and Coding School

# Running API

Running API dengan menjalankan perintah dibawah ini pada text editor

```
python app.py
```

Kemudian akses alamat dibawah menggunakan browser

```
http://localhost:5000
```

Seharusnya akan tampil seperti gambar dibawah ini

---


Purwadhika
Startup and Coding School

# Function untuk generate histogram dan box plot

```
# buat sebuah function dengan lima prameter, dimana masing - masing memiliki nilai default
def category_plot(
    cat_plot = 'histplot',
    cat_x = 'sex', cat_y = 'total_bill',
    estimator = 'count', hue = 'smoker'):

    # membuat dataframe dari tips.csv
    dfTips = pd.read_csv('./static/tips.csv')

    # jika menu yang di pilih adalah histoplot
    if cat_plot == 'histplot':
        # siapkan array kosong untuk menampung konfigurasi hist
        data = []
        # generate config histo plot dengan mengatur sumbu x dan sumbu y
        for val in dfTips[hue].unique():
            hist = go.Histogram(
                    x=dfTips[dfTips[hue] == val][cat_x], # Series
                    y=dfTips[dfTips[hue] == val][cat_y],
                    histfunc= estimator,
                    name = val
            )

            # masukkan ke dalam array
            data.append(hist)
            # tentukan title dari plot yang akan ditampilkan
            title = 'Histogram'

```

# Function untuk generate histogram dan box plot

```
    # menyiapkan config layout tempat plot akan ditampilkan
    # menentukan nama sumbu x dan sumbu y
    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
    res = {"data" : data, "layout" : layout}

    # json.dumps agan mengenerate plot dan menyimpan hasilnya pada graphJSON
    graphJSON = json.dumps(res,cls=plotly.utils.PlotlyJSONEncoder)

    return graphJSON

```

Purwadhika
Startup and Coding School

# BAGIAN 1
# category

Purwadhika
Startup and Coding School

# Route ‘ / ‘ untuk respon ‘category.html’

Route ini hanya akan diakses pada dua kondisi :

1. Saat pertama kali dashboard di akses
2. Saat kita click judul ‘MPG Dashboard’

Purwadhika
Startup and Coding School

# File : app.py

```python
@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(
        'category.html',
        plot=plot,
        focus_plot='histplot',
        focus_x='sex',
        focus_estimator='count',
        focus_hue = 'smoker',
        drop_plot = list_plot,
        drop_x = list_x,
        drop_y = list_y,
        drop_estimator = list_est,
        drop_hue = list_hue
    )
```

# File : app.py

```
# route ini akan diakses pertama kali kita mengakses halaman dashboard
# dan pada saat kita click 'MPG Dashboard' , function yang di tuju adalah 'index'
# route ini akan memberikan respon file 'category.html'
@app.route('/')
def index():
```

```
# saat pertama kali halaman dashboard diakses, maka kita tidak mengirim data apapun
# sehingga saat running function 'category_plot' tidak dapat argument didalamnya
# untuk kasus ini category_plot akan menggunakan nilai default  yg sudah dibuat
plot = category_plot()
```

Purwadhika
Startup and Coding School

# File : app.py

```
# Dropdown Menu
# kita lihat pada halaman dashboard terdapat menu dropdown
# terdapat lima menu dropdown, sehingga kita mengirimkan kelima variable dibawah ini
# kita mengirimnya dalam bentuk list agar mudah mengolahnya di halaman html menggunakan looping
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')]
```

# File : app.py

```
# akhirnya kita memberikan respon file 'category.html' beserta data yang dibutuhkan
# kemudian ini akan dirender / ditampilkan di browser
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 tampil di dropdown 'Sumbu X'
    focus_x='sex',

    # untuk sumbu Y tidak ada, nantinya menu dropdown Y akan di disable
    # karena pada histogram , sumbu y akan menunjukkan kuantitas data

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

# File : category.html

```
{% 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" >

         <!-- Dropdown Jenis plot -->


         <!-- Dropdown Sumbu X -->


         <!-- Dropdown Sumbu Y -->


         <!-- Dropdown Estimator -->


         <!-- Dropdown Hue-->

      </div>
   </form>

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

# File : category.html

```
<!-- extends menandakan bahwa file category.html akan digabung dengan index.html -->
<!-- terlihat pada halaman index.html terdapat bagian kode kosong bernama 'block' -->
<!-- bagian kosong ini akan di isi oleh category.html -->
<!-- itulah mengapa di file ini juga terdapat bagian kode yang diberi nama 'block' -->
{% extends 'index.html' %}
```

# File : category.html

```
<!-- Dropdown Jenis plot -->

<!-- Menu dropdown ini akan menampilkan list menu jenis - jenis plot yang ada -->
<!-- Disini 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>
```

Catatan : setiap menu dropdown memiliki pola yang sama yaitu menggunakan loop untuk me render / menampilkan menu dropdown.

Purwadhika
Startup and Coding School

# File : category.html

```
<!-- Dropdown Jenis plot -->

<div class="col-2">
   <p class="text-center lead">Plot</p>
   <select class="form-control" name="cat_plot" onchange="form.submit()">
      <!-- pada bagian ini kita akan melakukan loop untuk semua data pada list -->
      <!-- setiap dari datanya akan tercipta satu buat tag <li> -->
      {% for drop in drop_plot %}
            <!-- bagian ini digunakan untuk menentukan menu mana yang akan tampil -->
            <!-- tag <li> yg berisi menu yang akan tampil akan diberi property selected -->
         {% if focus_plot == drop[0] %}
               return '<option value={{drop[0]}} selected>{{drop[1]}}</option>'
            <!-- lainnya tidak memiliki property selected -->
         {% else %}
               return '<option value={{drop[0]}}>{{drop[1]}}</option>'
         {% endif %}
      {% endfor %}
   </select>
</div>
```

Purwadhika
Startup and Coding School

# File : category.html

```
<!-- 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>
```

Purwadhika
Startup and Coding School

# File : category.html

```
<!-- 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>
```

Purwadhika
Startup and Coding School

# File : category.html

```
<!-- 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>
```

Purwadhika
Startup and Coding School

# File : category.html

```
<!-- 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>
```

Purwadhika
Startup and Coding School

# BAGIAN 2
# category dan index

Purwadhika
Startup and Coding School

# File: app.py

```python
# ada dua kondisi dimana kita akan melakukan request terhadap route ini
# pertama saat klik menu tab (Histogram & Box)
# kedua saat mengirim form (saat merubah salah satu menu dropdown)
@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')
```

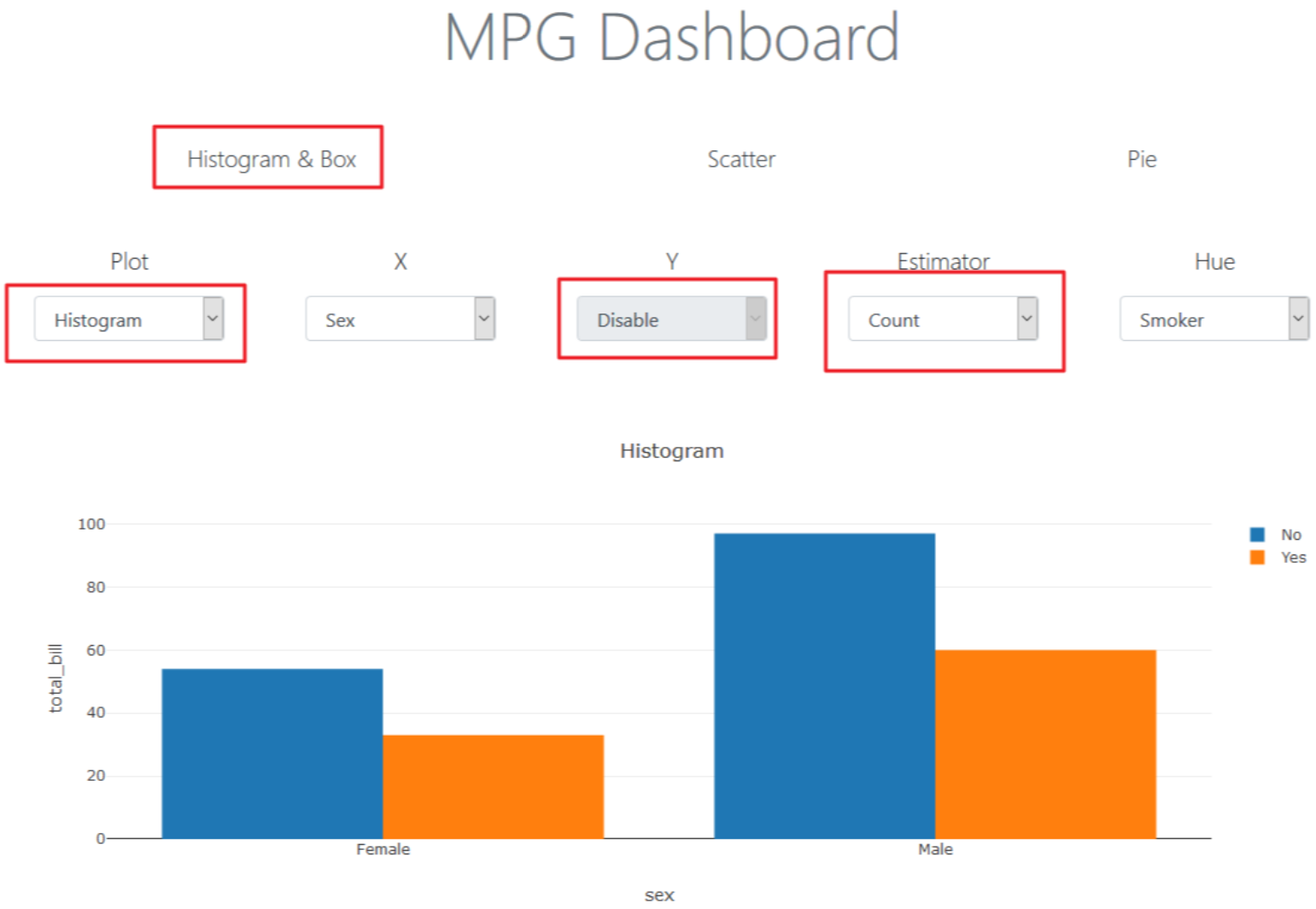
# File: app.py

```
# 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(
    'category.html',
    plot=plot,
    focus_plot=cat_plot,
    focus_x=cat_x,
    focus_y=cat_y,
    focus_estimator=estimator,
    focus_hue = hue,
    drop_plot = list_plot,
    drop_x = list_x,
    drop_y = list_y,
    drop_estimator = list_est,
    drop_hue = list_hue
)
```

# File: index.html (tambahkan href)

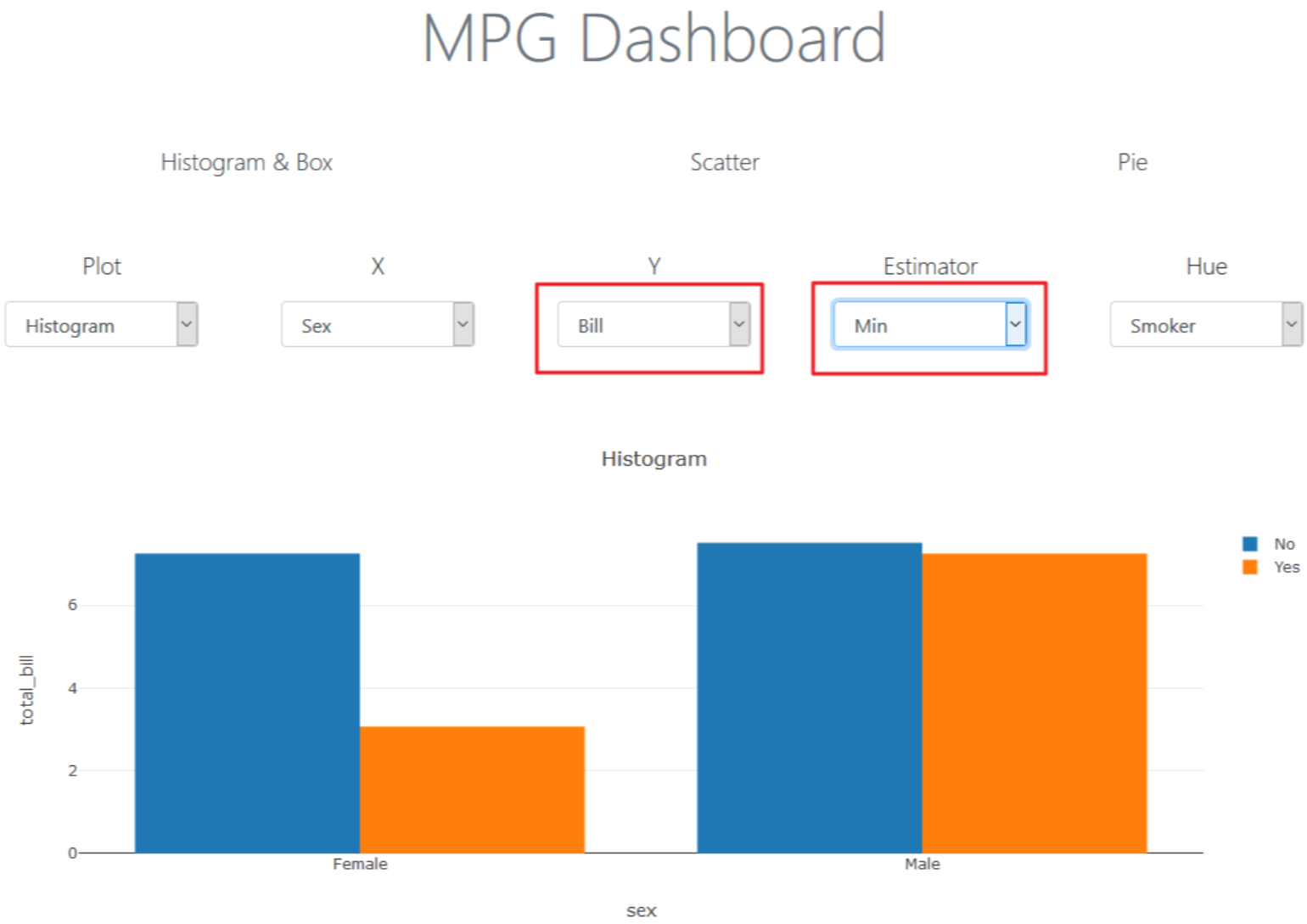
```
<!-- jika di click akan melakukan request ke route yang memiliki fungsi 'index' -->
<a class="text-decoration-none text-secondary " href="{{ url_for('index') }}">
    <h1 class="text-center text-capitalize display-4 my-5">MPG Dashboard</h1>
</a>

<ul class="nav nav-pills nav-fill">
    <!-- jika di click akan melakukan request ke route yang memiliki fungsi 'cat_fn'
      dengan membawa variable nav -->
    <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>
    <!-- jika di click akan melakukan request ke route yang memiliki fungsi 'scatt_fn' -->
    <li class="nav-item font-weight-bold ">
      <a class="nav-link text-dark lead" href="{{ url_for('scatt_fn') }}">Scatter</a>
    </li>
    <!-- jika di click akan melakukan request ke route yang memiliki fungsi 'pie_fn' -->
    <li class="nav-item font-weight-bold ">
      <a class="nav-link text-dark lead" href="{{ url_for('pie_fn') }}">Pie</a>
    </li>
  </ul>
```

# Halaman Histogram & Box



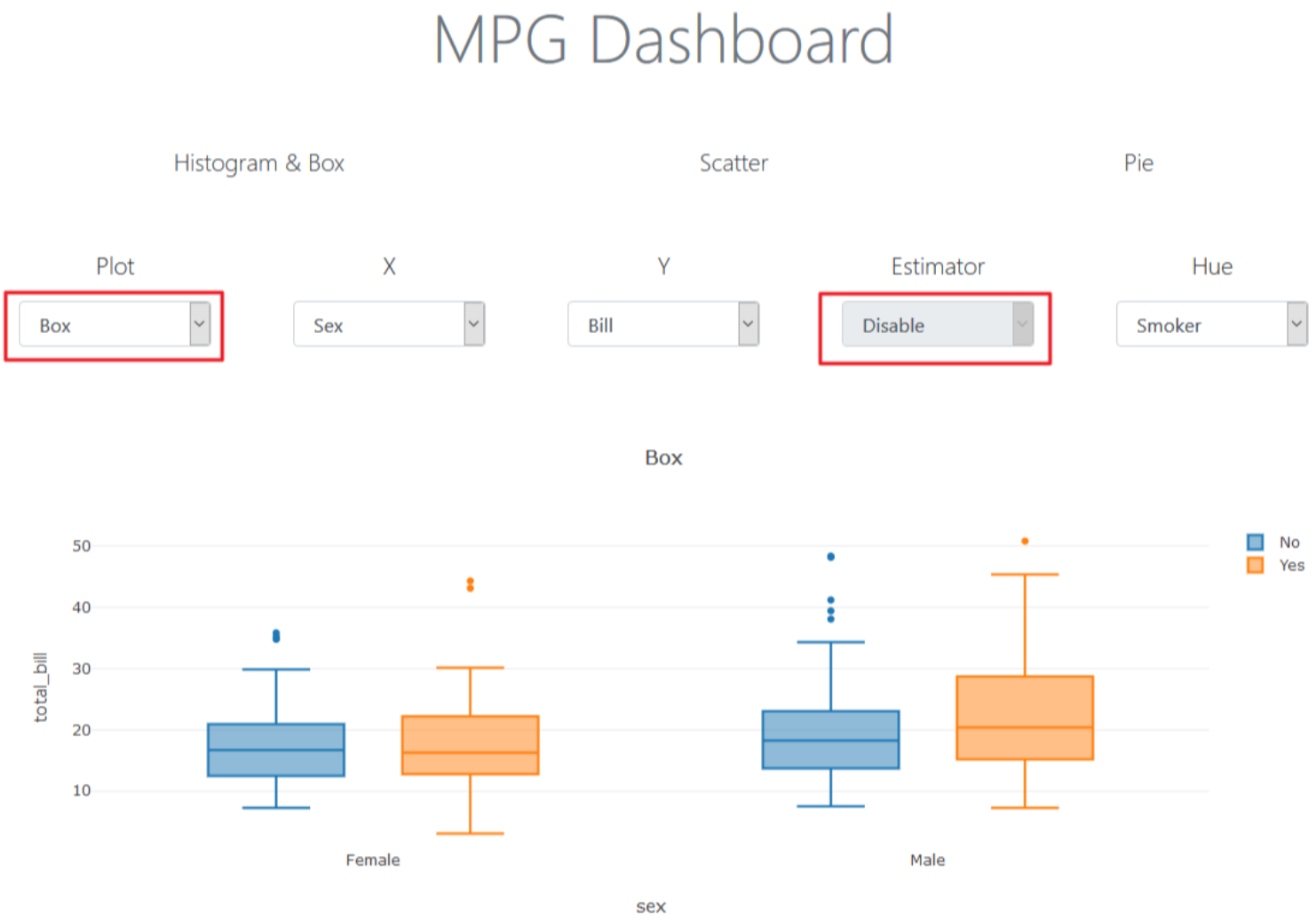

Purwadhika
Startup and Coding School

# Halaman Histogram & Box




Purwadhika
Startup and Coding School

# Halaman Histogram & Box




Purwadhika
Startup and Coding School

# BAGIAN 3
# Scatter

Purwadhika
Startup and Coding School

# File: scatter.html (template)

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

{% block content %}

   <!-- Menu dropdown -->
    <form action="{{ url_for('scatt_fn') }}" id="form">
        <div class="my-5 row d-flex justify-content-around" >

            <!-- Sumbu X -->

            <!-- Sumbu Y -->

            <!-- Hue -->

        </div>
    </form>

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

{% endblock  %}
```

Purwadhika
Startup and Coding School

# File: scatter.html (sumbu x)

```
<!-- Sumbu X -->
<select class="col-3 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>
```

Purwadhika
Startup and Coding School

# File: scatter.html (sumbu y)

```
<!-- Sumbu Y -->
<select class="col-3 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>
```

Purwadhika
Startup and Coding School

# File: scatter.html (hue)

```
<!-- Hue -->
<select class="col-3 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>
```

Purwadhika
Startup and Coding School

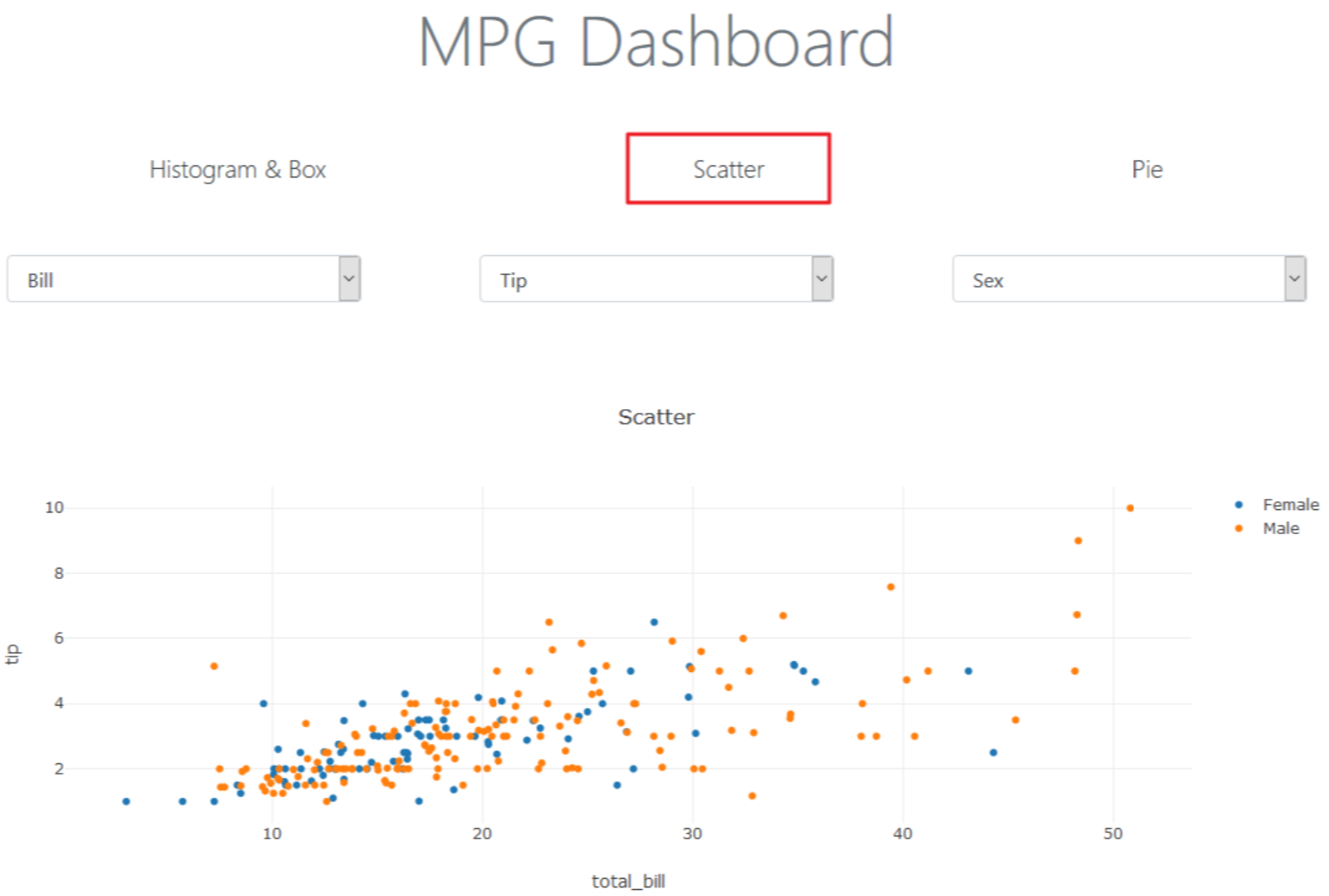
# File: app.py (scatter function)

```python
def scatter_plot(cat_x, cat_y , hue):

    dfTips = pd.read_csv('./static/tips.csv')

    data = []

    for val in dfTips[hue].unique():
        scatt = go.Scatter(
            x = dfTips[dfTips[hue] == val][cat_x],
            y = dfTips[dfTips[hue] == val][cat_y],
            mode = 'markers',
            name = val
        )

        data.append(scatt)

    layout = go.Layout(
        title='Scatter',
        title_x=0.5,
        xaxis=dict(title=cat_x),
        yaxis=dict(title=cat_y)
    )

    res = {"data" : data, "layout" : layout}

    graphJSON = json.dumps(res,cls=plotly.utils.PlotlyJSONEncoder)

    return graphJSON
```

# File: app.py (scatter route)

```python
@app.route('/scatt_fn')
def scatt_fn():
    cat_x = request.args.get('cat_x')
    cat_y = request.args.get('cat_y')
    hue = request.args.get('hue')

    if cat_x == None and cat_y == None and hue == None:
        cat_x = 'total_bill'
        cat_y = 'tip'
        hue = 'sex'

    # Dropdown Menu
    list_x = [('total_bill', 'Bill'),('tip', 'Tip'),('size', 'Size')]
    list_y = [('total_bill', 'Bill'),('tip', 'Tip'),('size', 'Size')]
    list_hue = [('sex', 'Sex'), ('smoker ', 'Smoker'), ('day', 'Daytime'), ('time', 'Time')]

    plot = scatter_plot(cat_x, cat_y, hue)

    return render_template(
        'scatter.html',
        plot=plot,
        focus_x=cat_x,
        focus_y=cat_y,
        focus_hue = hue,
        drop_x = list_x,
        drop_y = list_y,
        drop_hue = list_hue
    )
```

# Halaman Scatter




Purwadhika
Startup and Coding School

# BAGIAN 4
# Pie

Purwadhika
Startup and Coding School

# File: pie.html

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

{% block content %}

   <form action="{{ url_for('pie_fn') }}" id="form">
      <div class="my-5 row d-flex justify-content-center" >
         <!-- Menu dropdown hue -->
         <select class="col-3 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>

   <!-- Menampilkan chart / plot -->
   <div class="chart" id="plot">
      <script>
         var graphs = {{plot | safe}};
         Plotly.plot('plot',graphs,{});
      </script>
   </div>

{% endblock  %}
```

Purwadhika
Startup and Coding School

# File: app.py (function pie)

```python
def pie_plot(hue = 'sex'):

    dfTips = pd.read_csv('./static/tips.csv')

    vcounts = dfTips[hue].value_counts()

    labels = []
    values = []

    for item in vcounts.iteritems():
        labels.append(item[0])
        values.append(item[1])

    data = [
        go.Pie(
            labels=labels,
            values=values
        )
    ]

    layout = go.Layout(title='Pie', title_x = 0.48)

    res = {"data" : data, "layout" : layout}

    graphJSON = json.dumps(res,cls=plotly.utils.PlotlyJSONEncoder)

    return graphJSON
```

Purwadhika
Startup and Coding School

# File: app.py

```python
# Pie Plot
def pie_plot(hue = 'sex'):
    # membuat dataframe dari csv
    dfTips = pd.read_csv('./static/tips.csv')

    # menentukan berdasarkan data tertentu (hue)
    vcounts = dfTips[hue].value_counts()

    # menentukan presentasi dari setiap data pada hue
    labels = []
    values = []

    # memasukkan nilai ke masing - masing list
    for item in vcounts.iteritems():
        labels.append(item[0])
        values.append(item[1])

    # generate object pie
    data = [
        go.Pie(
            labels=labels,
            values=values
        )
    ]
```

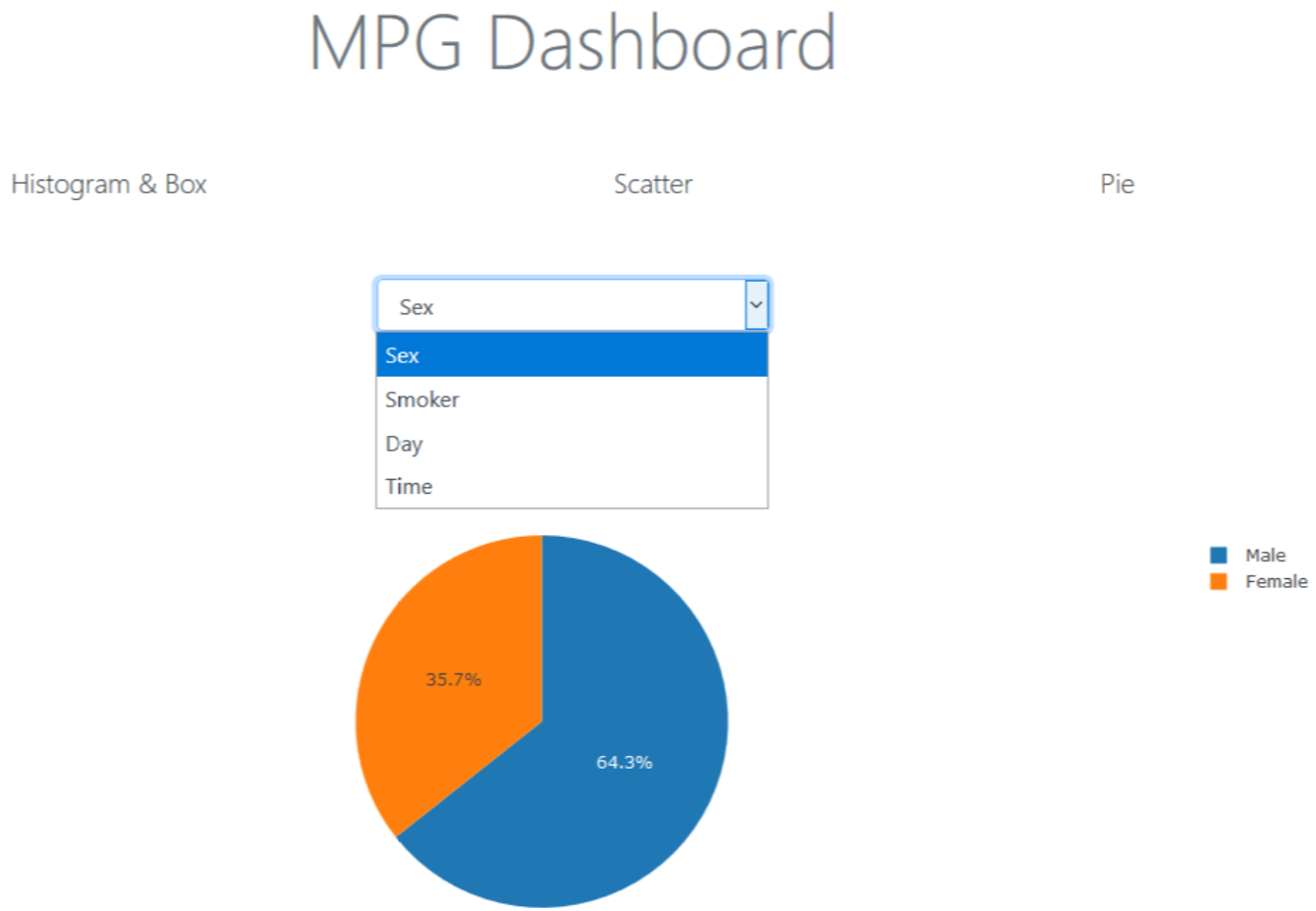
Purwadhika
Startup and Coding School

# File: app.py (route pie)

```python
@app.route('/pie_fn')
def pie_fn():
    hue = request.args.get('hue')

    # saat baru pertama akses halaman ini maka tidak ada data hue yangg terkirim
    if hue == None:
        hue = 'sex'

    # Dropdown Menu
    list_hue = [('sex', 'Sex'),('smoker', 'Smoker'),('day', 'Day'),('time', 'Time')]

    plot = pie_plot(hue)
    return render_template('pie.html', plot=plot, focus_hue=hue, drop_hue = list_hue)
```

Purwadhika
Startup and Coding School

# Halaman Pie




Purwadhika
Startup and Coding School