PYTHON DEVELOPMENT ON WINDOWS

- Anaconda: https://www.anaconda.com/download
  - Instalasi: klik kanan anaconda.exe > Run as Administrator
- Visual Studio Code: https://code.visualstudio.com/
- Github Desktop: https://desktop.github.com/download/
- Modul Python (buka cmd)
  - Plotly: pip install plotly
  - Dash: pip install dash

MA2213
VISUALISASI DALAM SAINS

"Pengenalan dashboard interaktif dan framework Dash"

Rifky Fauzi
Prodi Matematika
Institut Teknologi Sumatera

PLOTLY: STRUKTUR DAN CONTOH

```
import plotly.express as px
import pandas as pd

# dataframe dari tabel excel/database
df =

# Membuat scatter plot
fig = px.scatter(
    df,
    x= ‘nama_kolom_sumbu_x’,
    y='nama_kolom_sumbu_x’,
    color= warna,
    title= judul,
    labels= nama sumbu,
    size_max= ukuran
)

# Pengaturan layout
fig.update_layout(
    width= lebar,
    height= tinggi,
    showlegend=True,
    template='plotly_white'
)

# menampilkan plot
fig.show()
```

```
import plotly.express as px
import pandas as pd

# Sample data
df = pd.DataFrame({
    'x': [1, 2, 3, 4, 5],
    'y': [10, 11, 8, 15, 13],
    'category': ['A', 'B', 'A', 'B', 'A']
})

# Membuat scatter plot
fig = px.scatter(
    df,
    x='x',
    y='y',
    color='category',
    title='Sample Scatter Plot',
    labels={'x': 'X Axis', 'y': 'Y Axis'},
    size_max=60
)

# Pengaturan layout
fig.update_layout(
    width=800,
    height=600,
    showlegend=True,
    template='plotly_white'
)

# menampilkan plot
fig.show()
```

Secara default Plotly akan menampilkan plot melalui web browser

Apabila aplikasi dijalankan:

python app.py

Hasilnya akan muncul pada localhost

http://127.0.0.1:56324/

DASH: STRUKTUR DAN CONTOH

```
from dash import Dash, dcc, html
from dash.dependencies import Input, Output
import plotly.express as px
import pandas as pd

# InisialisasiDash app
app = Dash(__name__)

# data
df =

# Membuat scatter plot
fig = px.scatter(
    df,
    x= ‘nama_kolom_sumbu_x’,
    y='nama_kolom_sumbu_x’,
    color= warna,
    title= judul,
    labels= nama sumbu,
    size_max= ukuran
)

# Pengaturan plot
fig.update_layout(
    width= lebar,
    height= tinggi,
    showlegend=True,
    template='plotly_white'
)


# Deklarasi layout dashboard
app.layout = html.Div([
    # Title
    html.H1(“Judul H1", style={'textAlign': 'center'}),

    # Graph component
    dcc.Graph(id='scatter-plot', figure=fig),

    # Perintah lain terkait tampilan
])

# Callback untuk update tampilan berdasar input user
@app.callback(
    #terkait tampilan
    Output(‘tulis diharapkan berubah pada tampilan dari input user ‘),
    [Input(‘tuliskan input’)]

)
def suatu_fungsi_terkait_tampilan(argumen):

    return

# Jalankan aplikasi
if __name__ == '__main__':
    app.run_server(debug=True)
```

Jalankan aplikasi

python app.py

Hasilnya akan muncul pada localhost

*http://127.0.0.1:**8050**/*

```
from dash import Dash, dcc, html
from dash.dependencies import Input,
Output
import plotly.express as px
import pandas as pd

# Inisialisasi Dash app
app = Dash(__name__)

# data
df = pd.DataFrame({
    "study_hours": [2, 4, 6, 8, 10],
    "grades": [60, 75, 85, 90, 95],
    "class": ["A", "B", "A", "B", "A"]
})

# Membuat scatter plot
fig = px.scatter(
    df,
    x='study_hours',
    y='grades',
    color='class',
    title='Study Hours vs Grades',
    labels={'study_hours': 'Study Hours',
'grades': 'Grades'},
    size_max=60
)

# Pengaturan plot
fig.update_layout(
    width=800,
    height=600,
    showlegend=True,
    template='plotly_white'
)
```

```
# Deklarasi layout dashboard
app.layout = html.Div([
    # Title
    html.H1("Judul H1", style={'textAlign': 'center'}),

    # Graph component
    dcc.Graph(id='scatter-plot', figure=fig),

    # Perintah lain terkait tampilan
    html.Label("Select Class:"),
    dcc.Dropdown(
        id='class-dropdown',
        options=[
            {'label': 'All', 'value': 'All'},
            {'label': 'Class A', 'value': 'A'},
            {'label': 'Class B', 'value': 'B'}
        ],
        value='All',
        style={'width': '50%'}
    )
])

# Callback untuk update tampilan berdasar input user
@app.callback(
    #terkait tampilan
    Output('scatter-plot', 'figure'),
    [Input('class-dropdown', 'value')]
)
def suatu_fungsi_terkait_tampilan(selected_class):
    # Filter data based on selected class
    if selected_class == 'All':
        filtered_df = df
    else:
        filtered_df = df[df['class'] == selected_class]

    # Create updated scatter plot
    updated_fig = px.scatter(
        filtered_df,
        x='study_hours',
        y='grades',
        color='class',
        title=f'Study Hours vs Grades (Class: {selected_class})',
        labels={'study_hours': 'Study Hours', 'grades': 'Grades'},
        size_max=60
    )

    # Update layout
    updated_fig.update_layout(
        width=800,
        height=600,
        showlegend=True,
        template='plotly_white'
    )

    return updated_fig

# Jalankan aplikasi
if __name__ == '__main__':
    app.run_server(debug=True)
```

LATIHAN: MEMBUAT DASHBOARD VISUALISASI DATA

- Siapkan data dalam Google Spreadsheet
- Running Program dalam PC/Laptop
- Upload ke Github

TUGAS: BUAT DASHBOARD DARI DATA TUGAS BESAR

- Siapkan data dalam Google Spreadsheet (data tugas besar)
- Running Program dalam PC/Laptop
- Upload ke Github