

A decorative graphic on the right side of the slide featuring a grid of colored squares in shades of blue, red, green, and orange, arranged in a pattern reminiscent of Tetris blocks.

DØLab

TETRIS PROGRAM

Introduction to Streamlit

Pararawendy Indarjo

#StackYourSkill

Learning Outcome



Di sesi kelas hari ini, peserta diharapkan mampu

1. Mengenali konsep dashboard beserta fungsinya
2. Mengenali library streamlit, konsep, beserta contoh 'produk' yang dihasilkan dengan streamlit
3. Mengenal elemen-elemen dasar pada streamlit seperti input elements, media, dan macam macam container atau layout
4. Plotting basic dengan Streamlit

Apa itu streamlit?



“Streamlit turns data scripts into shareable web apps in minutes”

- Build an app in a few lines of code with simple API.
- Adding a widget is the same as declaring a variable.
- Used in the world’s top data science groups
- Compatible with many python libraries

Streamlit adalah library yang memungkinkan kita membuat **data apps** menggunakan Python dengan cepat dan penulisan script yang relatif sangat mudah.

Contoh data apps:

1. Business Metrics Dashboard
2. ML model prediction
3. Educational website
4. Etc

Untuk melihat contoh data apps atau dashboard dapat dilihat [di sini](#)

#StackYourSkill

Memulai streamlit: Hello World!

Buat file baru di dalam folder project dengan nama app.py

Tulis kode berikut lalu save

Eksekusi 'streamlit run app.py' pada terminal

The screenshot shows a code editor with a file named `app.py` containing the following code:

```
1 import streamlit as st
2
3 st.write("Hello world")
4
```

Below the code editor is a terminal window with the following output:

```
conda activate streamlit_app
(base) rizkikurniawan@Rizkis-MBP streamlit_app % conda activate streamlit_app
(streamlit_app) rizkikurniawan@Rizkis-MBP streamlit_app % streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.43.23:8501

For better performance, install the Watchdog module:

$ xcode-select --install
$ pip install watchdog
```

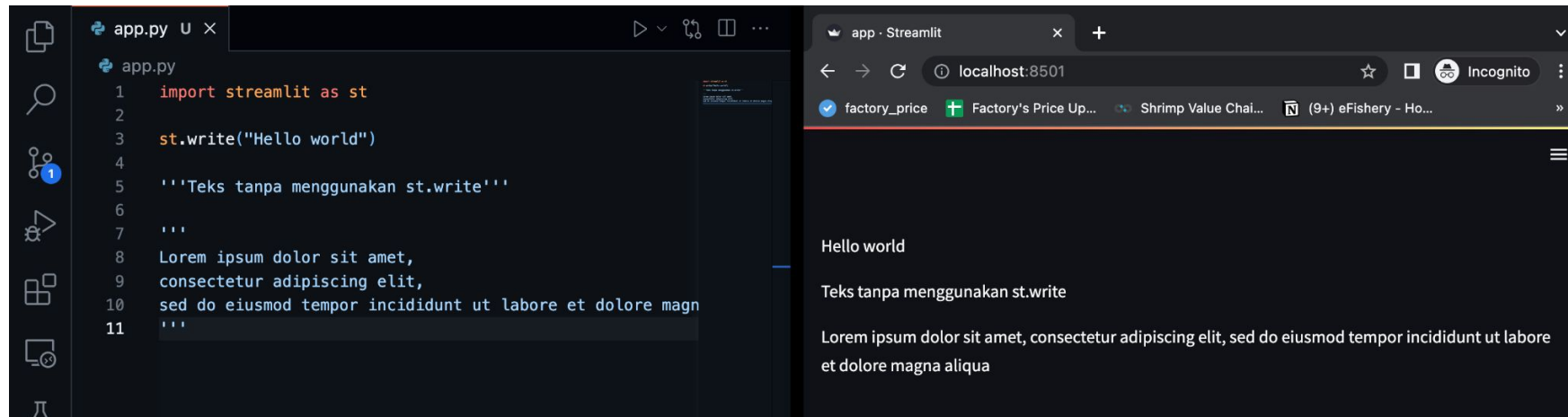
On the right side of the screenshot, a web browser is open at `localhost:8501`, displaying the text "Hello world".

Halaman web baru akan muncul. Jika tidak, masukan alamat URL pada terminal

Menampilkan teks

Kita dapat menampilkan teks pada Streamlit app melalui:

- `st.write`
- docstring: mengait teks menggunakan tripe quote



The image shows a Streamlit application running in a web browser on the right and its source code in a code editor on the left. The code editor displays a file named `app.py` with the following content:

```
1 import streamlit as st
2
3 st.write("Hello world")
4
5 '''Teks tanpa menggunakan st.write'''
6
7 '''
8 Lorem ipsum dolor sit amet,
9 consectetur adipiscing elit,
10 sed do eiusmod tempor incididunt ut labore et dolore magn
11 '''
```

The web browser on the right shows the application running at `localhost:8501`. It displays the output of the code: "Hello world", followed by "Teks tanpa menggunakan st.write", and then the Lorem ipsum text block.

Menulis teks berbagai format



```
1 import streamlit as st
2
3 # Memasukkan teks dengan metode markdown
4 st.markdown("Tulisan ini menggunakan _Markdown_")
5
6 # Menulis teks dengan style Title
7 st.title("Ini adalah judul")
8
9 # Menulis teks dengan style Header
10 st.header("Ini adalah header")
11
12 # Menulis teks dengan style Subheader
13 st.subheader("Ini adalah subheader")
14
15 # Menulis teks dengan preformatted-text/teks biasa
16 st.text("Ini adalah preformatted text")
17
18 # Menulis teks dengan style caption
19 st.caption("Ini adalah caption")
20
21 # Menulis teks kode
22 st.code("import streamlit as st")
23
24 # Menulis teks kode multibaris
25 st.code("""
26 # import library streamlit
27 import streamlit as st
28 """)
29
30 # Menulis teks Latex
31 st.latex("ax^2 + bx + c = 0")
```

Tulisan ini menggunakan *Markdown*

Ini adalah judul

Ini adalah header

Ini adalah subheader

Ini adalah preformatted text

Ini adalah caption

```
import streamlit as st
```

```
# import library streamlit
import streamlit as st
```

$$ax^2 + bx + c = 0$$

Streamlit memiliki banyak opsi dalam menampilkan teks menjadi beberapa format seperti

- Judul,
- Bab,
- Subbab,
- teks biasa
- hingga kode dan LaTeX untuk menulis formula matematika

Selain itu, user juga dapat menambahkan format **markdown** (seperti pada Colab notebook)

Menampilkan Media: Gambar & Audio

The screenshot shows a Streamlit application running in a web browser at localhost:8501. The application displays a large image of a sunset over a beach with a person walking. Below the image is a caption: "Ini adalah pemandangan matahari terbit". At the bottom, there is an audio player showing a progress bar at 0:26 / 0:26. On the left, a code editor shows the Python code used to load the image and audio files.

```
app.py > ...
1 import streamlit as st
2 from PIL import Image
3
4 image = Image.open('media/sunrise.jpeg')
5
6 st.image(
7     image,
8     caption="Ini adalah pemandangan matahari terbit"
9 )
10
11 audio_file = open('media/opening.mp3', 'rb')
12 audio_bytes = audio_file.read()
13
14 st.audio(audio_bytes, format='audio/mp3')
```

- Menampilkan **gambar** menggunakan streamlit dapat menggunakan library PIL dan `st.image`
- Untuk dapat menampilkan **audio player** yang dapat memainkan berbagai macam format audio dapat menggunakan `st.audio`

Menampilkan Media: Video

A screenshot of a Streamlit application. On the left, a code editor shows the following Python code:

```
1 import streamlit as st
2
3 video_file = open('media/dqlab.mp4', 'rb')
4 video_bytes = video_file.read()
5
6 st.video(video_bytes)
```

On the right, a web browser window displays the Streamlit app running at localhost:8501. The app shows a video player with a cartoon character of a data scientist wearing a lab coat and glasses, with a pie chart and test tubes in the background. The text "Data Scientist" is written below the character. The video player controls show a progress bar at 0:05 / 1:37. The bottom of the browser window shows the Streamlit logo and the text "Made with Streamlit".

Menampilkan **video** player dengan `st.video`

Widget



- Widget adalah elemen interaktif yang memungkinkan users untuk berinteraksi dan mengontrol konten dari sebuah Streamlit App
- Widget sangat berguna untuk membuat dynamic dan responsive user interfaces
- Streamlit menyediakan banyak built-in widgets yang dapat kita gunakan dengan mudah
- Di kelas ini, kita akan mencoba menggunakan widget-widget berikut
 - Input text
 - Button
 - Checkbox
 - Radio
 - Multiselect
 - Slider

Input widget: beragam jenis button



The screenshot displays a Streamlit application interface with the following components:

- Python Code (Left Panel):**

```
1 import streamlit as st
2
3 # output: True/False
4 tombol = st.button("Tekan tombol ini")
5 st.write(tombol)
6
7 # output: True/False
8 agree = st.checkbox("Apakah kamu setuju?")
9 st.write(agree)
10
11 genre = st.radio(
12     "Pilih genre musik favoritmu",
13     ("Pop", "Rock", "Metal", "Indie")
14 )
15 st.write(genre)
16
17 kota = st.selectbox(
18     "Pilih kota tujuan",
19     ("Bandung", "Jakarta", "Surabaya", "Padang")
20 )
21 st.write(kota)
22
23 multi_kota = st.multiselect(
24     "Pilih kota tujuan",
25     ("Bandung", "Jakarta", "Surabaya", "Padang")
26 )
27 st.write(multi_kota)
```
- Application Interface (Right Panel):**
 - A button labeled "Tekan tombol ini".
 - A checkbox labeled "Apakah kamu setuju?" with a green "False" label.
 - A radio button group for "Pilih genre musik favoritmu" with "Pop" selected.
 - A selectbox for "Pilih kota tujuan" with "Surabaya" selected.
 - A multiselect box for "Pilih kota tujuan" with "Surabaya" and "Padang" selected.
 - A list output showing `0 : "Surabaya"` and `1 : "Padang"`.

- `st.button` awalnya akan bernilai False lalu akan bernilai True jika diklik
- `st.checkbox` akan bernilai True jika dicentang dan False untuk sebaliknya
- `st.radio` akan memiliki nilai sesuai dengan opsi yang dipilih
- `st.selectbox` akan memiliki nilai sesuai dengan opsi yang dipilih
- `st.multiselect` akan mengembalikan *list* berisi label yang dipilih

Input widget: Slider dan Text

The image shows a Streamlit application in two parts: the code editor on the left and the rendered web interface on the right. The code editor shows a Python script using Streamlit widgets. The rendered interface shows the corresponding UI elements: a slider for parameter X, a select_slider for shirt size, a number_input for parameter Y, a text_input for name, and a text_area for comments.

```
1 import streamlit as st
2
3 x = st.slider(
4     "Masukkan nilai parameter X",
5     min_value=0,
6     max_value=100,
7     step=1,
8     value=10 # nilai default
9 )
10 st.write(x)
11
12 size = st.select_slider(
13     "Masukkan ukuran baju",
14     ("XS", "S", "M", "L", "XL", "XXL"),
15     value="L" # nilai default
16 )
17 st.write(size)
18
19 y = st.number_input(
20     "Masukkan nilai parameter Y",
21     min_value=0,
22     max_value=10,
23     step=1,
24     value=3 # nilai default
25 )
26 st.write(y)
27
28 nama = st.text_input("Masukkan nama kamu")
29 st.write(nama)
30
31 komentar = st.text_area("Tulis komentar di sini")
32 st.write(komentar)
```

The rendered interface shows the following UI elements:

- A slider for "Masukkan nilai parameter X" with a range from 0 to 100 and a default value of 10.
- A select_slider for "Masukkan ukuran baju" with options ("XS", "S", "M", "L", "XL", "XXL") and a default value of "L".
- A number_input for "Masukkan nilai parameter Y" with a range from 0 to 10 and a default value of 3.
- A text_input for "Masukkan nama kamu" with the value "Rizki".
- A text_area for "Tulis komentar di sini" with the value "Sangat baik".

- `st.slider` dan `st.number_input` memiliki fungsi yang hampir sama yaitu meminta input berupa angka yang terdapat pada rentang tertentu
- `st.select_slider` digunakan untuk meminta inputan berupa data ordinal seperti ukuran baju, indeks nilai, dsb
- `st.text_input` meminta inputan berupa teks singkat
- sedangkan `st.text_area` meminta inputan berupa teks panjang

Input widget: Date Input dan Color Picker

The screenshot shows a Streamlit application in two parts. On the left is a code editor with the following Python code:

```
1 import streamlit as st
2
3 tanggal_lahir = st.date_input(
4     "Masukkan tanggal lahir kamu"
5 )
6 st.write(tanggal_lahir)
7
8 awal_meeting = st.time_input(
9     "Masukkan waktu meeting"
10 )
11 st.write(awal_meeting)
12
13 warna = st.color_picker(
14     "Masukkan warna yang disukai"
15 )
16 st.write(warna)
17
```

On the right is the rendered web application in a browser at localhost:8501. It displays three input widgets: a date input with the value '2023/05/20', a time input with the value '16:26', and a color picker with a red square and the hex code '#7f3048'. The bottom of the browser window shows 'Made with Streamlit'.

- `st.date_input` dan `st.time_input` adalah input widget dalam streamlit yang digunakan untuk meminta inputan berupa tanggal dan waktu
- `st.color_picker` membantu user dalam memilih kode warna hexadecimal

App Layout



- Kita dapat mengatur layout dari Streamlit app sesuai kebutuhan
- Umumnya kustomisasi layout bertujuan untuk meningkatkan readibilitas dan estetika app kita
- Streamlit menyediakan beragam pilihan pengaturan layout, seperti:
 - Column
 - Sidebar
 - Expander
 - Container
 - Tab

Columns



A screenshot of a Streamlit application. On the left, a code editor shows two Python code blocks. The first block creates three equal-width columns and titles them 'Kolom 1', 'Kolom 2', and 'Kolom 3'. The second block creates two columns with widths of 1/4 and 3/4, and titles them 'Kolom 1' and 'Kolom 2'. Red arrows point from these code blocks to the corresponding output in the browser. The browser window on the right shows the rendered application with three columns of equal width, each titled 'Kolom 1', 'Kolom 2', and 'Kolom 3'. Below them, two columns are shown with widths of 1/4 and 3/4, titled 'Kolom 1' and 'Kolom 2'. The Streamlit logo and 'Made with Streamlit' text are visible at the bottom.

- `st.columns` merupakan container yang tersusun secara horizontal dan berfungsi untuk mengelompokkan object
- Untuk memasukkan `n` buah kolom sama lebar, cukup masukkan nilai `n` pada `st.columns(n)`
- Untuk memasukkan `n` buah kolom dengan lebar variatif, masukkan parameter berupa list sepanjang `n` yang berisi nilai lebar dari kolom secara berurutan
- Kolom bisa memuat kolom lagi namun maksimal 1 (one level nesting)

Sidebar

The image shows a Streamlit application in two parts. On the left is a code editor window titled 'app.py' showing the following Python code:

```
1 import streamlit as st
2
3 with st.sidebar:
4     st.write("Ini adalah sidebar")
5     nama = st.text_input("Masukkan nama kamu")
```

A red box highlights the code inside the `with st.sidebar:` block. Below the code, there is a text box with the following text:

Sidebar dapat memuat berbagai macam object.

Untuk memasukkan object ke dalam sidebar user dapat menulis script di dalam area ini

On the right is a browser window titled 'app - Streamlit' showing the rendered application at 'localhost:8501'. The application has a dark theme and a sidebar on the left. The sidebar contains the text 'Ini adalah sidebar' and a text input field with the placeholder 'Masukkan nama kamu'. The main area of the application is currently empty.

- Sidebar adalah elemen dalam streamlit yang berfungsi untuk menambahkan layout berupa sidebar interaktif yang *collapsible* dan memuat object

Expander



```
app.py > ...
1 import streamlit as st
2 from lorem_text import lorem
3
4 with st.expander("See Lorem Ipsum"):
5     text = lorem.paragraphs(3)
6
7     st.write(text)
```

See Lorem Ipsum

See Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nihil quas veritatis laboriosam quia qui atque nemo possimus deleniti nisi porro, officiis aliquid fuga, consequatur minima ad corporis, similique magnam expedita doloremque quod quae fuga est voluptatum rem voluptatem? Blanditiis unde consectetur voluptatibus sequi accusamus iste reiciendis amet quia in, cupiditate facilis culpa a modi, eaque voluptate tempora, ex sed a fugit, fuga quisquam nemo aperiam architecto quas placeat itaque accusamus eaque quia repellat. Maxime explicabo blanditiis eligendi dolorum odio expedita velit, ratione officiis laborum, nisi sed harum ut quisquam non, quae animi culpa sed quia blanditiis commodi? Cumque dignissimos assumenda tenetur pariatur enim modi, vero ipsa quasi dolor, recusandae odio eaque rem illo dolor deleniti incidunt molestiae voluptate dolorum provident? Unde necessitatibus nobis cum voluptate, nam voluptatibus illo alias labore rerum optio qui officia blanditiis asperiores, tempore nostrum ab, iste possimus at quasi qui sapiente iusto quidem consequatur a minus, error consequatur id vel facilis quasi hic amet praesentium? Reprehenderit iusto doloremque fugiat, quo nesciunt ullam expedita iusto debitis soluta, vitae voluptatibus voluptatem dolores quod dolor a suscipit earum numquam, beatae iure placeat blanditiis temporibus culpa?

- st.expander adalah elemen kontainer yang collapsible (dapat dibuka/tutup)
- karena st.expander adalah salah satu jenis kontainer maka st.expander dapat dimasukkan beberapa element
- Tidak dapat membuat expander di dalam expander

Container



Lorem ipsum dolor sit amet

Consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Diam quis enim lobortis scelerisque fermentum dui faucibus in. Pharetra magna ac placerat vestibulum lectus mauris ultrices.



Sem integer vitae justo eget. In egestas erat imperdiet sed euismod nisi porta lorem mollis. Eu feugiat pretium nibh ipsum consequat nisi vel pretium. Elit ut aliquam purus sit amet. Aliquet nibh praesent tristique magna sit. Dapibus ultrices in iaculis nunc.

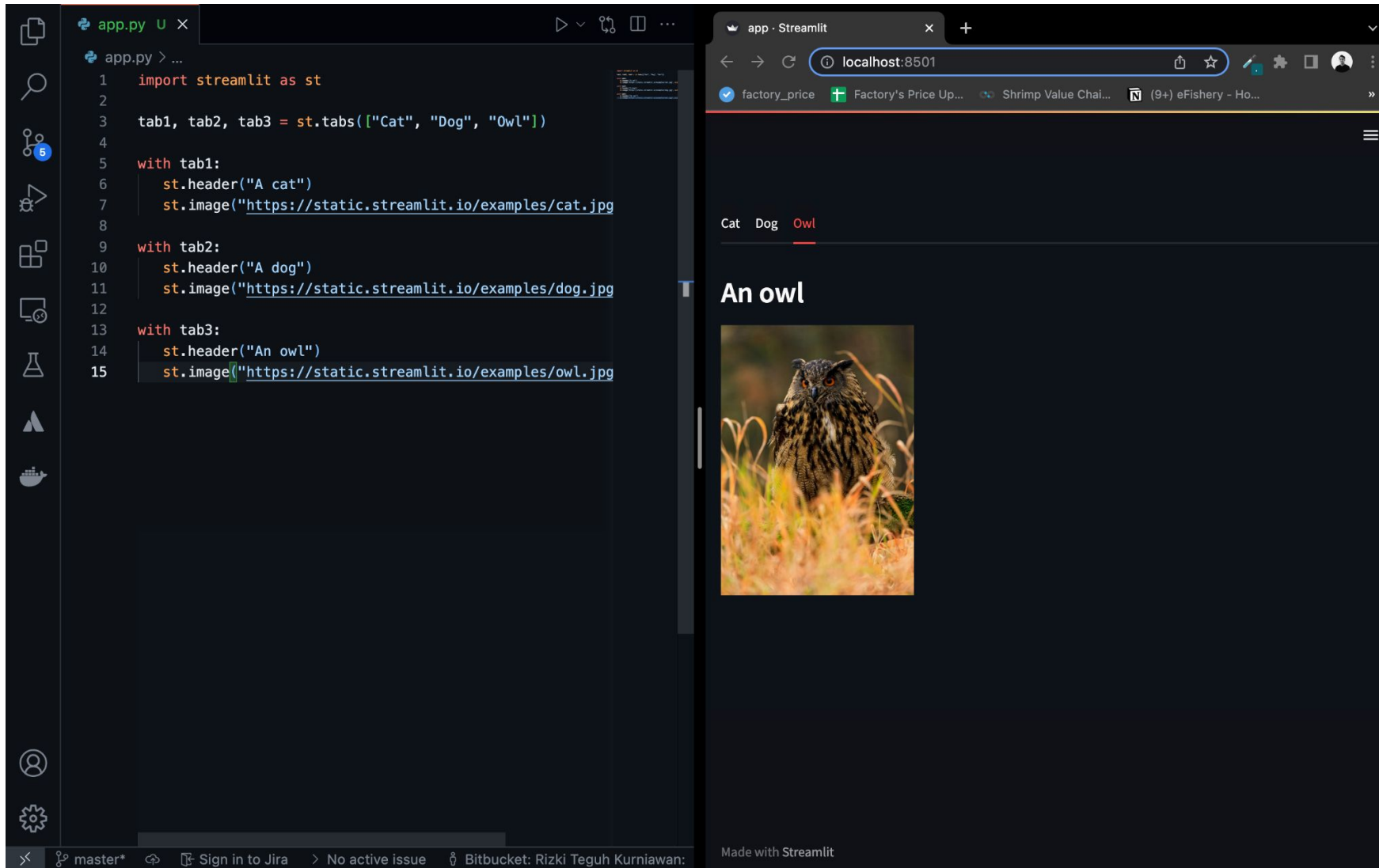
Enim eu turpis egestas pretium aenean pharetra. Nunc sed blandit libero volutpat sed cras ornare arcu. Etiam erat velit scelerisque in. Purus semper eget dui at tellus at urna condimentum mattis. Sapien faucibus et vestibulum at blandit sed lectus vestibulum mattis. Odio ut sem nulla pharetra diam sit amet.

```
app.py U x
app.py
1 import streamlit as st
2 from lorem_text import lorem
3
4 with st.container():
5     st.write('ini di dalam kontainer')
6
7 st.write('ini di luar kontainer')
```

st.container adalah element yang berguna untuk mengelola penempatan dan pengelompokkan elemen.

untuk dapat memasukkan elemen ke dalam kontainer cukup masukkan elemen di dalam notasi with dari kontainer

Tab



- st.tabs adalah kontainer berbentuk tab. memudahkan user untuk bernavigasi di antara group konten yang saling berhubungan.
- Untuk membuat n buah tab, berikan parameter pada st.tabs suatu list sepanjang n yang berisi judul untuk masing-masing tab

Plotting in Streamlit



- Tentu saja, kita dapat menampilkan visualisasi data di dalam Streamlit App
- Streamlit menyediakan basic plotting options berikut
 - Bar chart
 - Line chart
 - Scatter plot
- Bagaimana jika saya membutuhkan visualisasi lainnya?
 - Gunakan library data visualization! (materi kelas selanjutnya)

Barchart



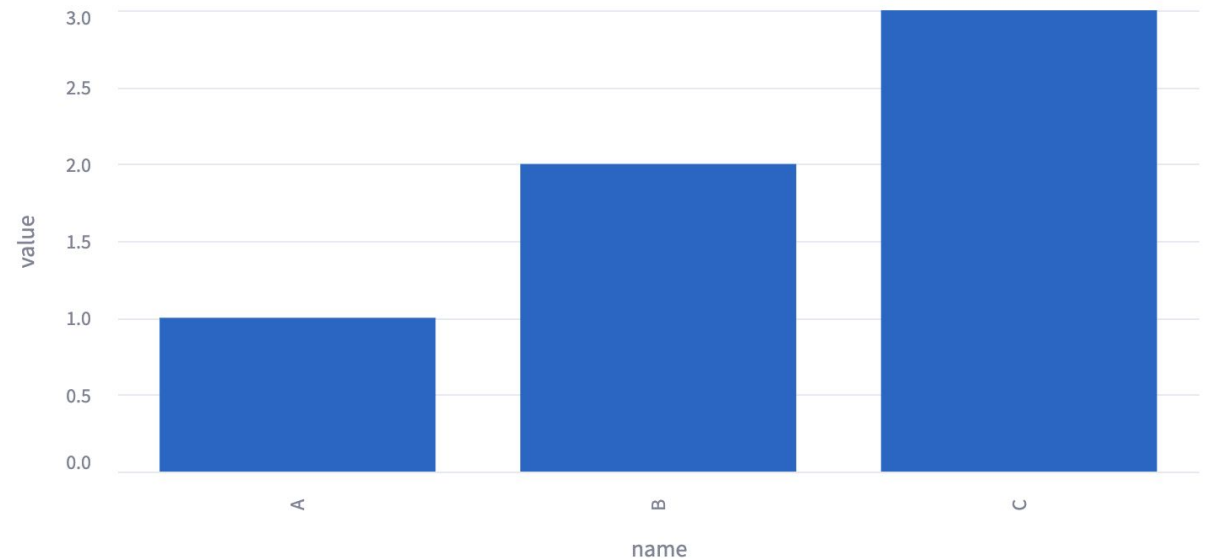
```
st.bar_chart(data, x, y)
```

- data: pandas dataframe
- x: column name as x axis
 - Best practice: kategorik
- y: column name as y axis

Contoh di samping

```
df = pd.DataFrame({  
    "name": ["A","B","C"],  
    "value": [1,2,3]  
})
```

```
st.bar_chart(data=df, x="name", y="value")
```



Line chart



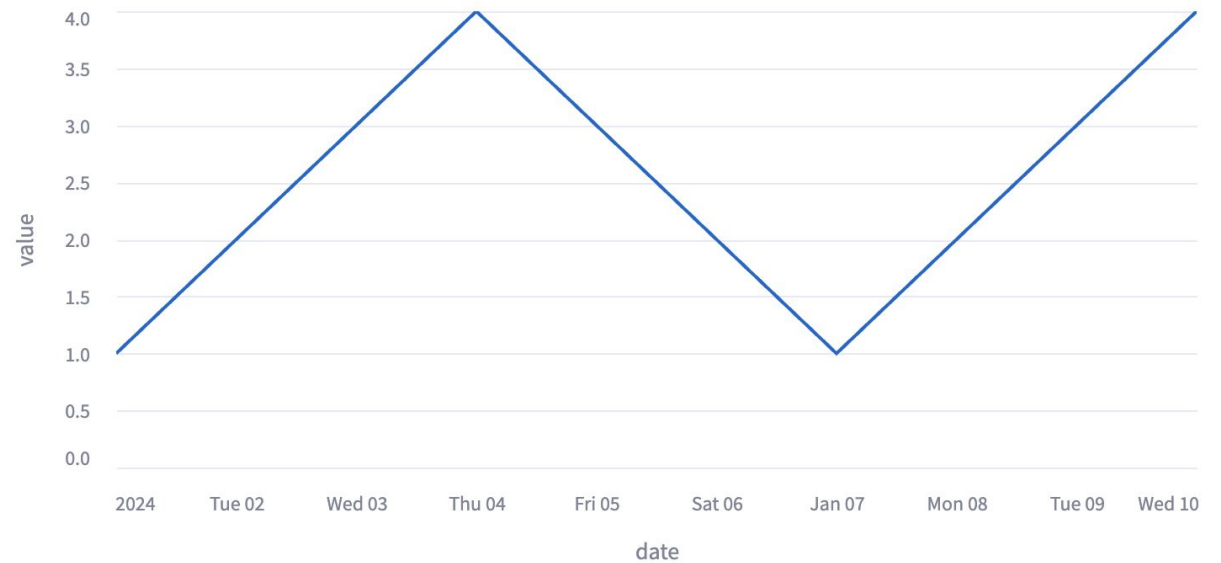
```
st.line_chart(data, x, y)
```

- data: pandas dataframe
- x: column name as x axis
 - Best practice: temporal features
- y: column name as y axis

Contoh di samping

```
df = pd.DataFrame({  
    "date": pd.date_range(start='2024-01-01', end='2024-01-10',  
        freq='D'),  
    "value": [1,2,3,4,3,2,1,2,3,4]  
})
```

```
st.line_chart(data=df, x="date", y="value")
```



Scatter Plot



```
st.scatter_chart(data, x, y)
```

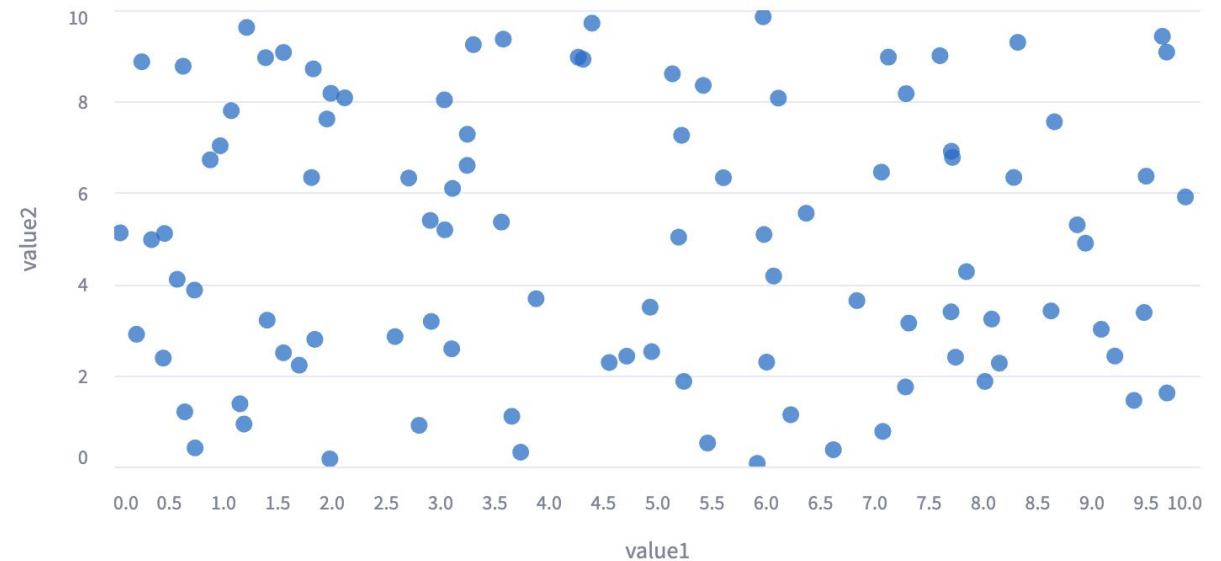
- data: pandas dataframe
- x: column name as x axis
 - Bertipe numerik
- y: column name as y axis
 - Bertipe numerik

Contoh di samping

```
np.random.seed(42)
random_numbers = np.random.uniform(0, 10, 200)
```

```
df = pd.DataFrame({
    "value1": random_numbers[:100],
    "value2": random_numbers[100:]
})
```

```
st.scatter_chart(data=df, x="value1", y="value2")
```



DΦLab

AYO #STACKYOURSKILL SEKARANG

dan Persiapkan Diri Menjadi Praktisi Data!

