

Week #2

# Introduction to Streamlit

Akhsin Nurlayli



# Introduction: Streamlit Library

- History
- Installation
- Main Concepts
- Development Flow



# Apa itu Streamlit?

- **Library** yang dimiliki Python yang bersifat **open source**, release pada **Oktober 2019**
- Package seperti: Flask, Django
- Digunakan untuk membuat **web apps**
- Streamlit juga **bersifat open sharing** sehingga mudah untuk dibagikan
- Streamlit memudahkan pengguna untuk **mengubah data script** menjadi **aplikasi berbasis web yang interaktif**

# Streamlit Founder

## OUR FOUNDERS



### Adrien Treuille

Dr. Treuille has been a Zoox VP, Google X project lead, and Computer Science faculty at Carnegie Mellon. He has won numerous scientific awards, including the MIT TR35. Adrien has been featured in the documentaries *What Will the Future Be Like* by PBS/NOVA, and *Lo and Behold* by Werner Herzog.



### Thiago Teixeira

Founder of Hangouts Chat at Google, and former tech lead manager of Hangouts web clients. Former tech lead of an AI project at Google X. Thiago has a Ph.D. from Yale, and his research includes human behavior recognition in sensor networks and building a sensor-fusion middleware for the IoT.



### Amanda Kelly

Amanda led both product and operations for several stealthy Google X projects and served as Director of Launch Operations for Zoox. She has run projects across autonomous vehicles, NLP, e-commerce and mobile hardware. Amanda has an MBA from Stanford GSB and holds several AV-related patents.



# Install Streamlit

- Install Integrated Development Environment (IDE) sebelum menggunakan streamlit
- Beberapa pilihan IDE yang dapat digunakan
  - Visual Studio Code
  - PyCharm
  - Dll

<https://docs.streamlit.io/library/get-started/installation>

# Set up your virtual environment

Regardless of which package management tool you're using, we recommend running the commands on this page in a virtual environment. This ensures that the dependencies pulled in for Streamlit don't impact any other Python projects you're working on.

Below are a few tools you can use for environment management:

- [pipenv](#)
- [poetry](#)
- [venv](#)
- [virtualenv](#)
- [conda](#)

# Install Streamlit (Anaconda)

Download aplikasi <https://www.anaconda.com/products/distribution>

The screenshot shows the Anaconda website homepage. At the top, there is a navigation bar with links: Products (underlined), Pricing, Solutions, Resources, Partners, Blog, and Company. To the right of the navigation bar is a "Contact Sales" button. Below the navigation bar, the text "Individual Edition is now" is followed by a large green header "ANACONDA DISTRIBUTION". Below this, a sub-headline reads "The world's most popular open-source Python distribution platform". On the right side of the page, there is a call-to-action box with a green header "Anaconda Distribution" and a "Download" button featuring a Windows icon. Below the button, it says "For Windows" and "Python 3.9 • 64-Bit Graphical Installer • 594 MB". At the bottom of the call-to-action box, there is a link "Get Additional Installers" followed by icons for Windows, Apple, and Linux.

Individual Edition is now

# ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

## Anaconda Distribution

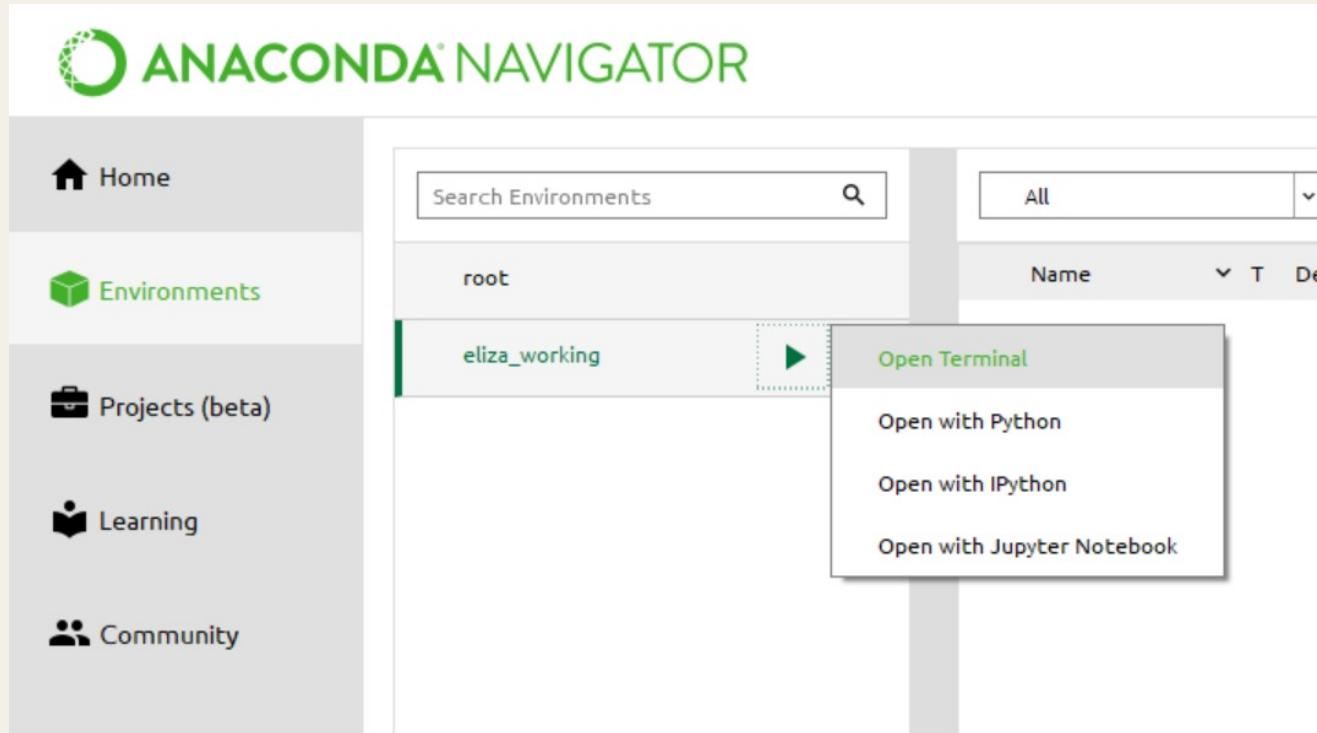
Download

For Windows  
Python 3.9 • 64-Bit Graphical Installer • 594 MB

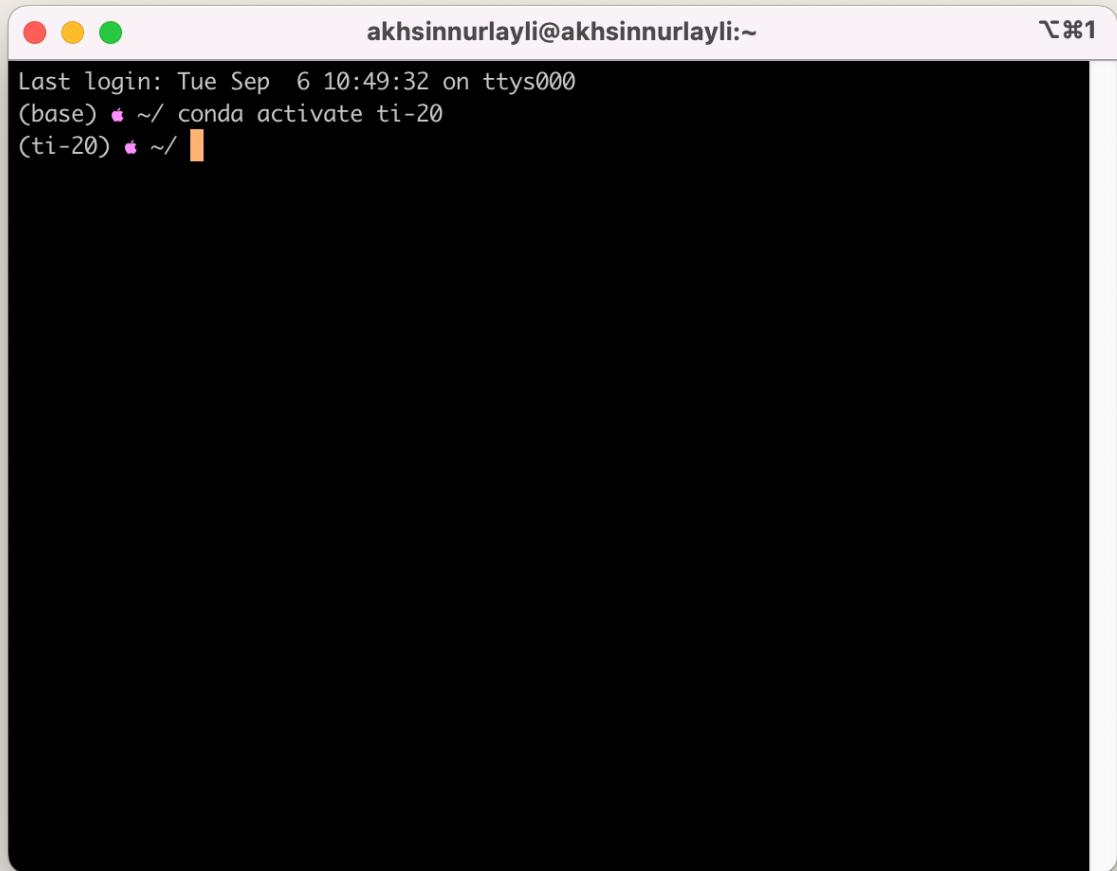
Get Additional Installers

| |

# Install Streamlit (Anaconda)



# Activate Environment



```
Last login: Tue Sep  6 10:49:32 on ttys000
(base) ✘ ~/ conda activate ti-20
(ti-20) ✘ ~/ █
```

# Install Streamlit

```
akhsinnurlayli@akhsinnurlayli:~ % Last login: Tue Sep  6 10:49:32 on ttys000  
(base) % ~/conda activate ti-20  
(ti-20) % ~/pip install streamlit  
DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion at https://github.com/Homebrew/homebrew-core/issues/76621  
Requirement already satisfied: streamlit in /opt/homebrew/lib/python3.9/site-packages (1.11.0)  
Requirement already satisfied: semver in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (2.13.0)  
Requirement already satisfied: pympler>=0.9 in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (1.0.1)  
Requirement already satisfied: numpy in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (1.21.1)  
Requirement already satisfied: pillow>=6.2.0 in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (9.2.0)  
Requirement already satisfied: protobuf<4,>=3.12 in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (3.20.1)  
Requirement already satisfied: pydeck>=0.1.dev5 in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (0.7.1)  
Requirement already satisfied: pandas>=0.21.0 in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (1.4.1)  
Requirement already satisfied: importlib-metadata>=1.4 in /opt/homebrew/lib/python3.9/site-packages (from streamlit) (4.12.0)
```

# Buka Streamlit

```
streamlit hello

kages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->wid
getsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (2.21)
DEPRECATION: Configuring installation scheme with distutils config files is depr
ecated and will no longer work in the near future. If you are using a Homebrew o
r Linuxbrew Python, please see discussion at https://github.com/Homebrew/homebre
w-core/issues/76621
(ti-20) ✘ ~/ streamlit hello

Welcome to Streamlit. Check out our demo in your browser.

Local URL: http://localhost:8501
Network URL: http://10.1.161.13:8501

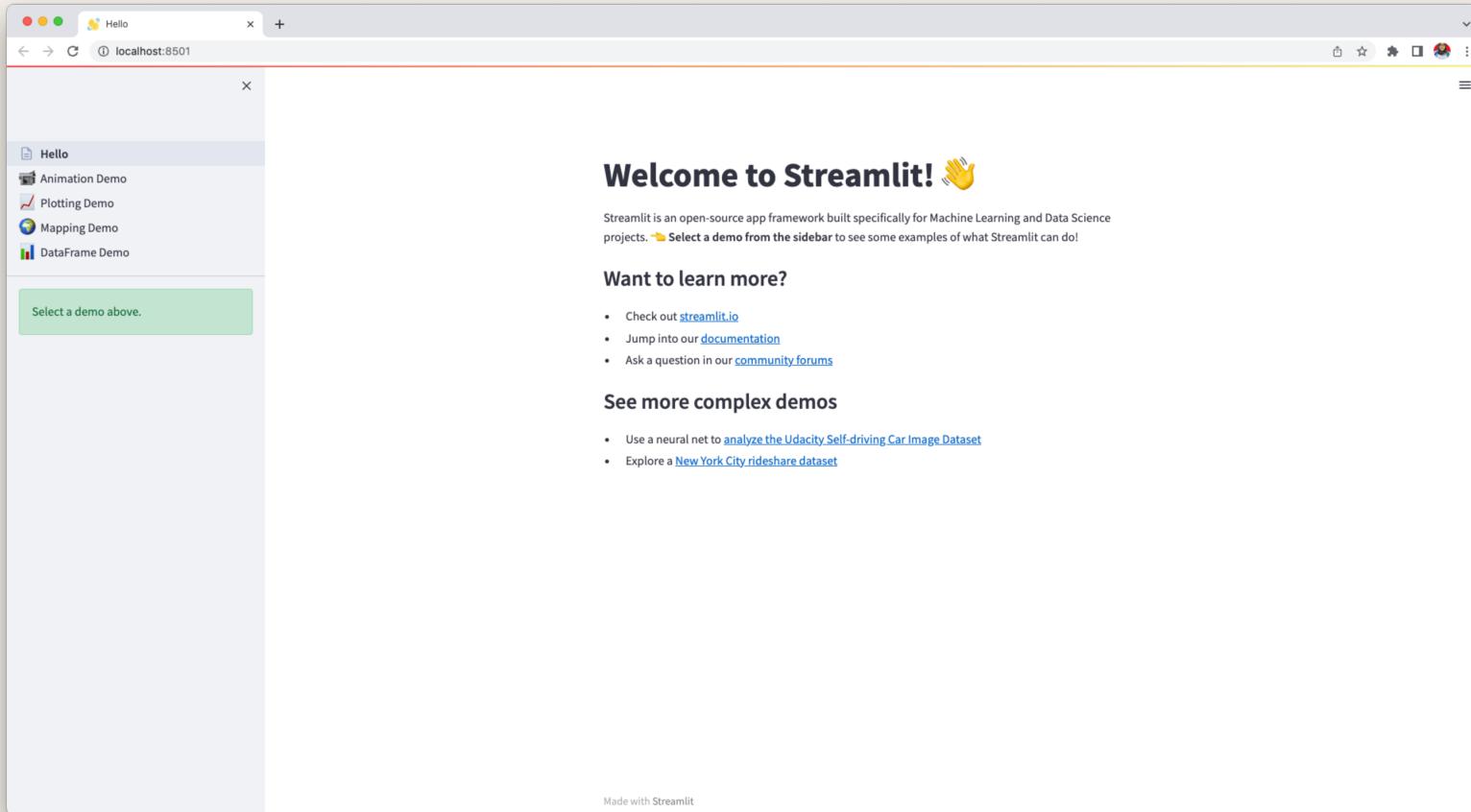
Ready to create your own Python apps super quickly?
Head over to https://docs.streamlit.io

May you create awesome apps!

For better performance, install the Watchdog module:

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

# Install Streamlit (Anaconda)



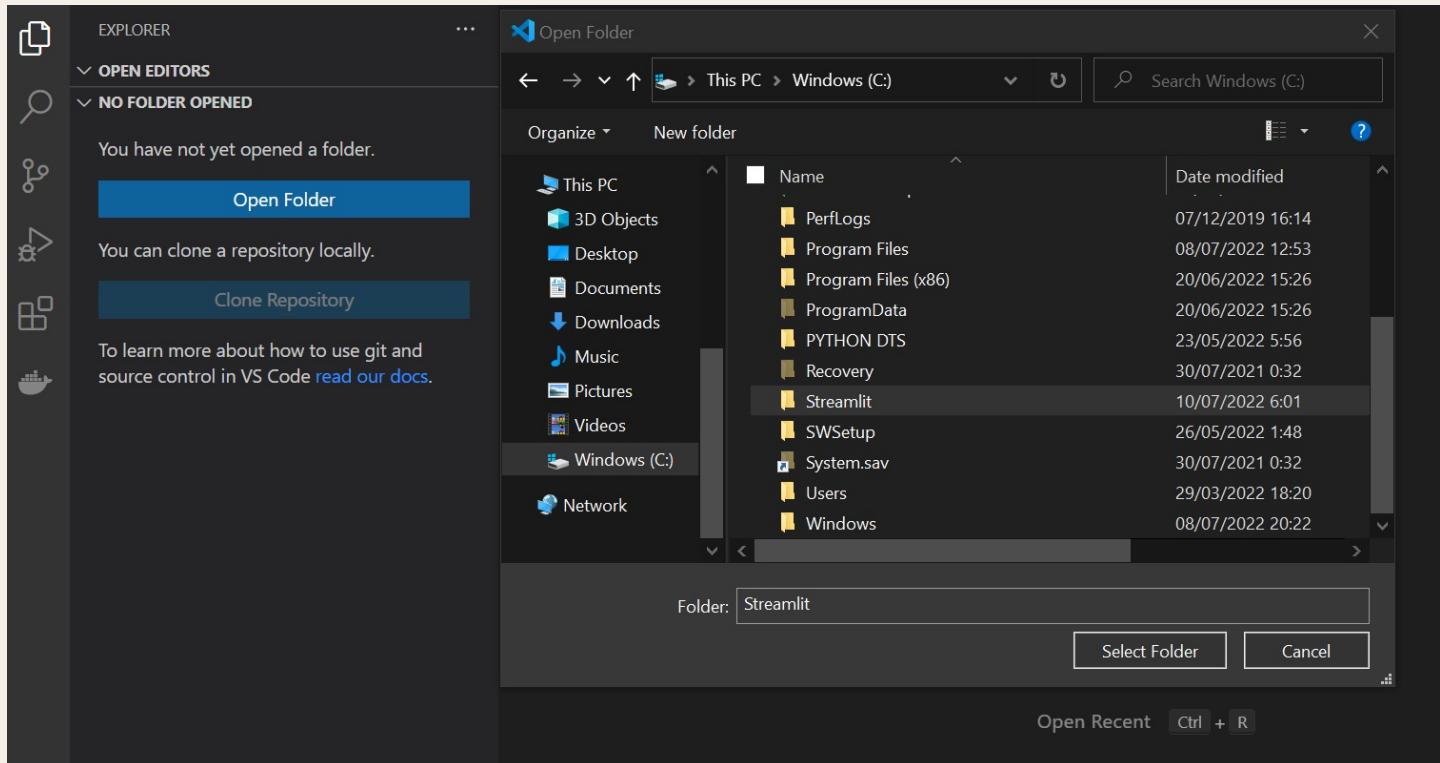


# Install Streamlit (Virtual Environment)

1. Buat folder “streamlit”
2. Buat virtual environment pada terminal
3. Install pip streamlit pada virtual environment

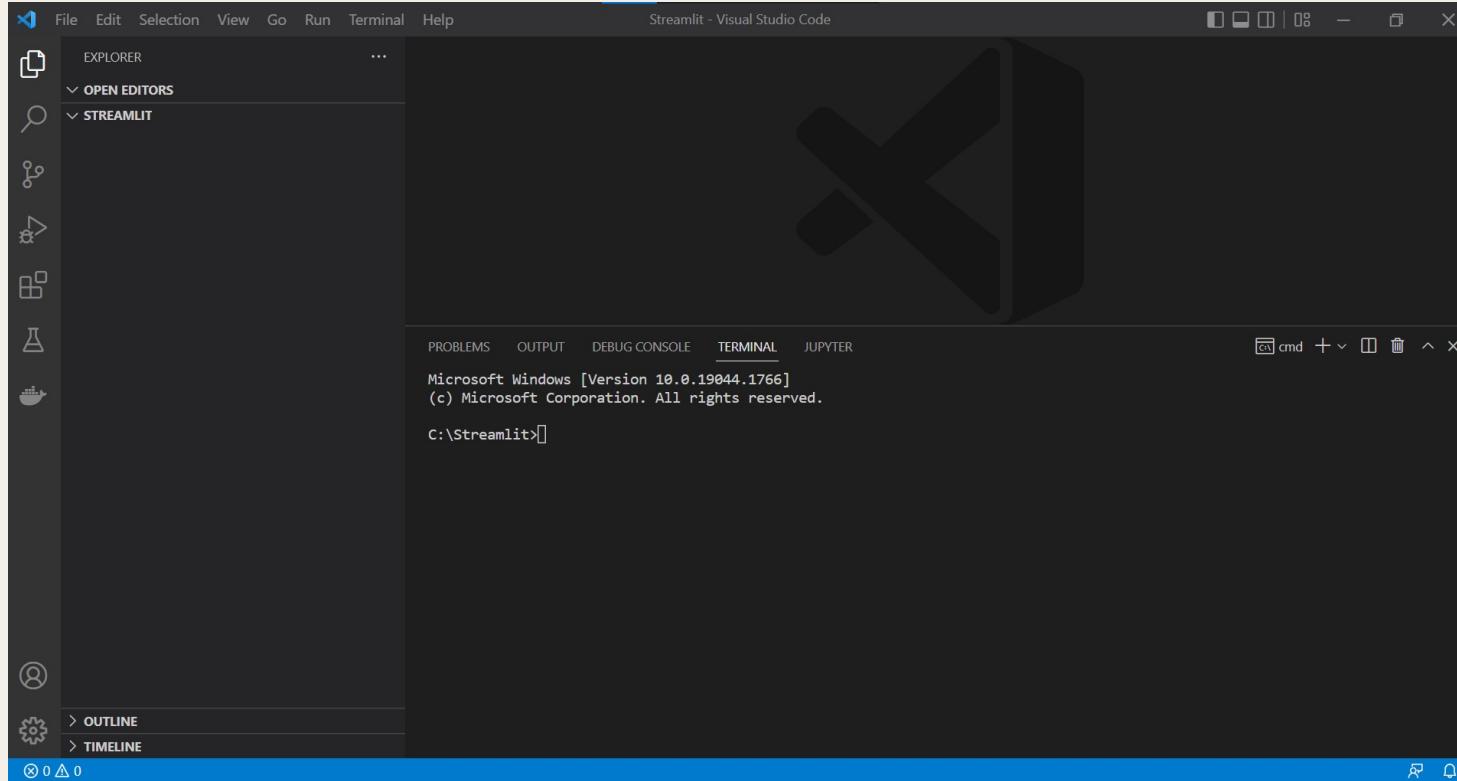
# Install Streamlit (Virtual Environment)

Buka folder yang sudah dibuat dengan Open Folder (contoh pada vs code)



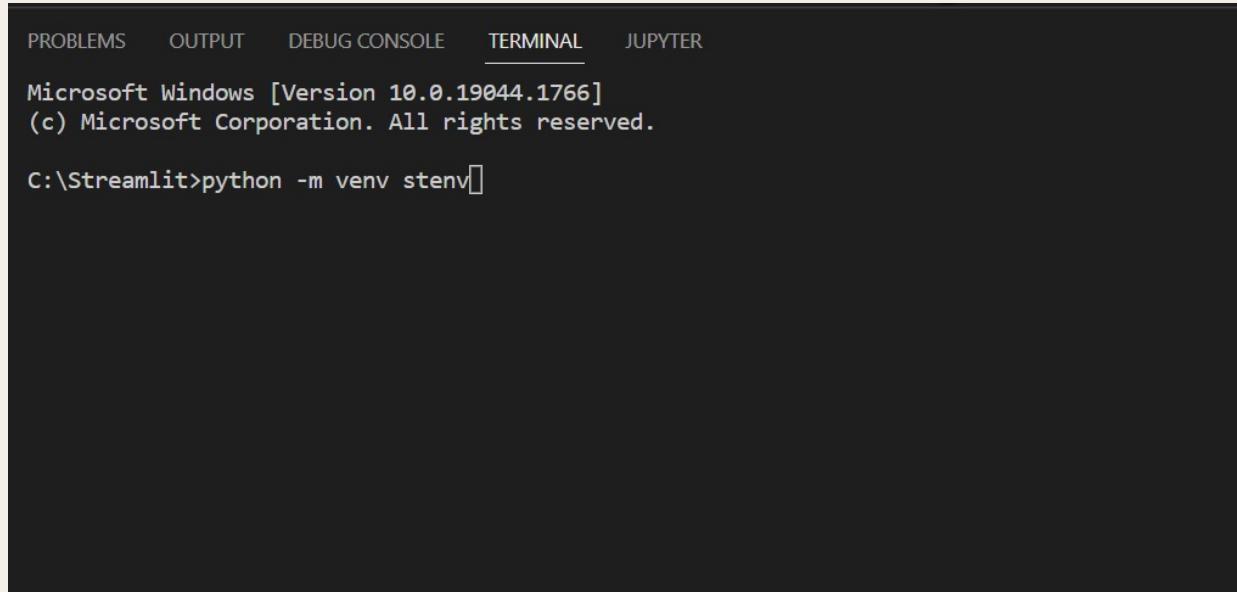
# Install Streamlit (Virtual Environment)

Aktifkan CMD pada terminal (arahkan ke direktori <nama folder yang dibuat>)



# Install Streamlit (Virtual Environment)

Buat virtual environment dengan perintah `python -m venv <nama virtual environment>`



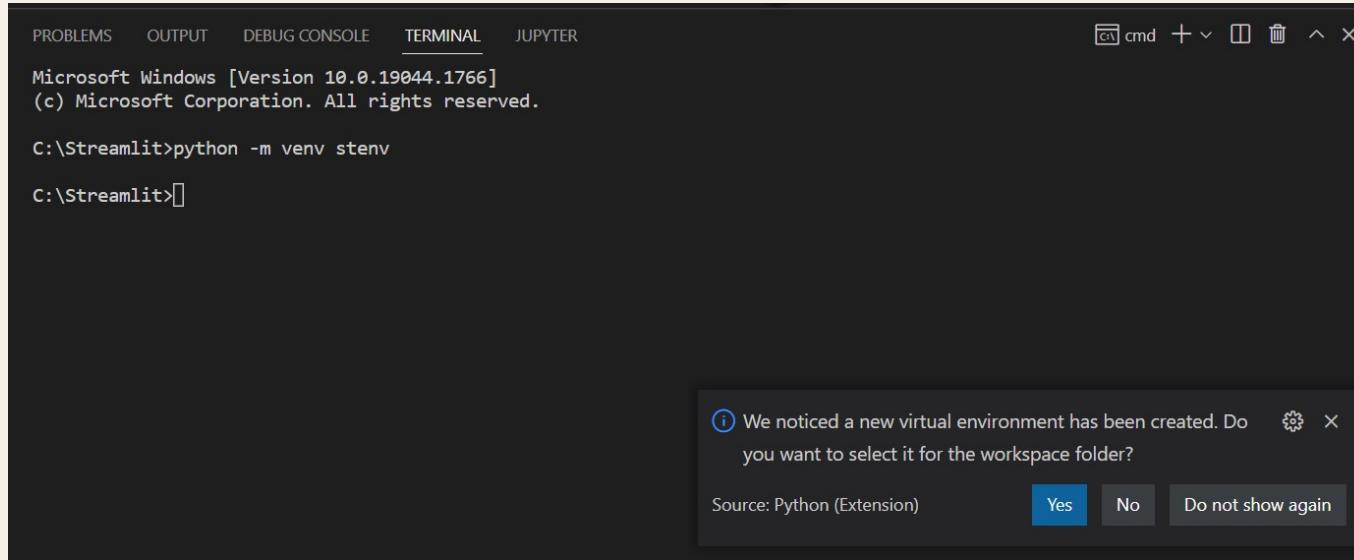
The screenshot shows a terminal window with the following interface elements:

- Top bar: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), JUPYTER
- System information: Microsoft Windows [Version 10.0.19044.1766]  
(c) Microsoft Corporation. All rights reserved.
- Terminal command: C:\Streamlit>`python -m venv stenv`

The terminal command `python -m venv stenv` is partially visible at the bottom of the terminal window.

# Install Streamlit (Virtual Environment)

Klik Yes untuk membuat virtual environment pada workspace folder



The screenshot shows a terminal window in Visual Studio Code. The terminal tab is selected, displaying the command:

```
C:\Streamlit>python -m venv stenv
```

Below the terminal, a modal dialog box is displayed, indicating that a new virtual environment has been created and asking if it should be selected as the workspace folder. The dialog includes "Yes", "No", and "Do not show again" buttons.

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

Microsoft Windows [Version 10.0.19044.1766]  
(c) Microsoft Corporation. All rights reserved.

C:\Streamlit>python -m venv stenv

C:\Streamlit>[ ]

ⓘ We noticed a new virtual environment has been created. Do you want to select it for the workspace folder?

Source: Python (Extension)    Yes    No    Do not show again

# Install Streamlit (Virtual Environment)

Aktifkan virtual environment dengan perintah activate (arahkan ke folder Scripts)

```
C:\Streamlit>stenv\Scripts\activate
```

```
C:\Streamlit>stenv\Scripts\activate
```

```
(stenv) C:\Streamlit>
```

# Install Streamlit (Virtual Environment)

Install streamlit dengan perintah pip install streamlit

```
(stenv) C:\Streamlit>pip install streamlit[]
```

The screenshot shows a terminal window with the following interface elements:

- Top bar: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (highlighted), JUPYTER.
- Right side: cmd icon, +, - (minimize/maximize), X (close).

The terminal output displays the results of the pip install command:

```
etsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (1.1.1)
Requirement already satisfied: defusedxml in c:\streamlit\stenv\lib\site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (0.7.1)
Requirement already satisfied: beautifulsoup4 in c:\streamlit\stenv\lib\site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (4.11.1)
Requirement already satisfied: fastjsonschema in c:\streamlit\stenv\lib\site-packages (from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (2.15.3)
Requirement already satisfied: pywinpty>=1.1.0 in c:\streamlit\stenv\lib\site-packages (from terminado>=0.8.3->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (2.0.6)
Requirement already satisfied: argon2-cffi-bindings in c:\streamlit\stenv\lib\site-packages (from argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (21.2.0)
Requirement already satisfied: cffi>=1.0.1 in c:\streamlit\stenv\lib\site-packages (from argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (1.15.1)
Requirement already satisfied: soupsieve>1.2 in c:\streamlit\stenv\lib\site-packages (from beautifulsoup4->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (2.3.2.post1)
Requirement already satisfied: webencodings in c:\streamlit\stenv\lib\site-packages (from bleach->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (0.5.1)
Requirement already satisfied: pycparser in c:\streamlit\stenv\lib\site-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit) (2.21)

(stenv) C:\Streamlit>[]
```

# Install Streamlit (Virtual Environment)

Test apakah streamlit sudah terinstall dengan perintah: streamlit hello

```
(stenv) C:\Streamlit>streamlit hello
```

```
(stenv) C:\Streamlit>streamlit hello

Welcome to Streamlit. Check out our demo in your browser.

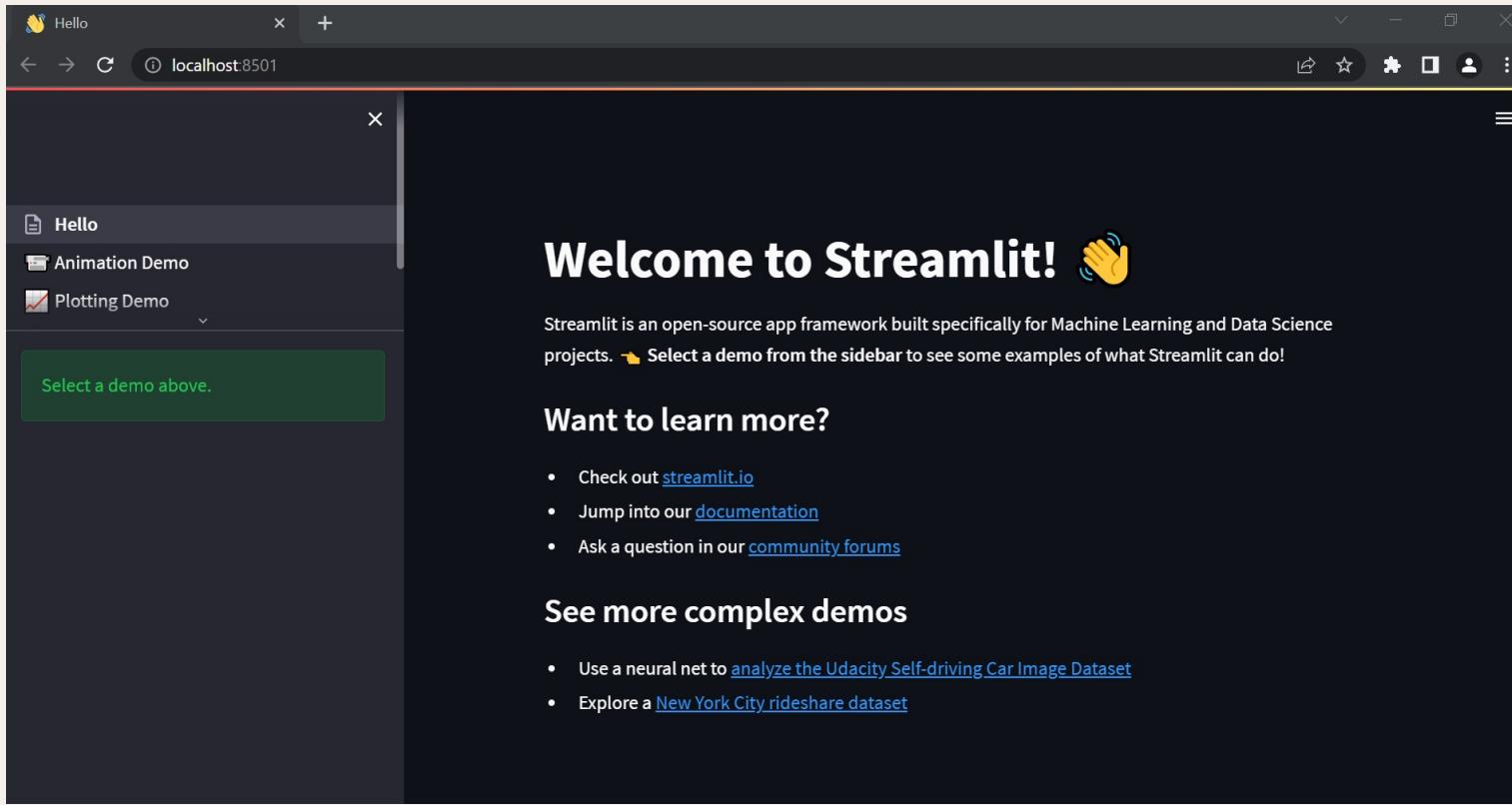
Local URL: http://localhost:8501
Network URL: http://192.168.1.3:8501

Ready to create your own Python apps super quickly?
Head over to https://docs.streamlit.io

May you create awesome apps!
```

# Install Streamlit (Virtual Environment)

Browser akan terbuka dan menampilkan halaman Welcome to Streamlit



# Konsep Streamlit (buat script pada IDE)

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





# Konsep Streamlit

- Streamlit akan berjalan pada web browser, dengan menggunakan lingkungan sendiri (local host)
- Untuk menjalankan Streamlit, ketik perintah pada terminal cmd

```
TERMINAL DEBUG CONSOLE PROBLEMS OUTPUT
(base) ✘ ~/Documents/streamlit-ti20/ conda activate ti-20
(ti-20) ✘ ~/Documents/streamlit-ti20/ streamlit run write.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8502
```

# Konsep Streamlit (Run Script)

```
>Last login: Tue Sep  6 11:26:08 on ttys000
(base) ✘ ~/ conda activate ti-20
(ti-20) ✘ ~/ cd /Users/akhsinnurlayli/Documents
(ti-20) ✘ ~/Documents/ ls
Additional Tasks Others           Zoom           tes-wit
Administration Personal Data     code           wit-2022
Lecture          Research        mooc
Luar Kampus      Software       streamlit-ti20
(ti-20) ✘ ~/Documents/ cd streamlit-ti20
(ti-20) ✘ ~/Documents/streamlit-ti20/ streamlit run write.py

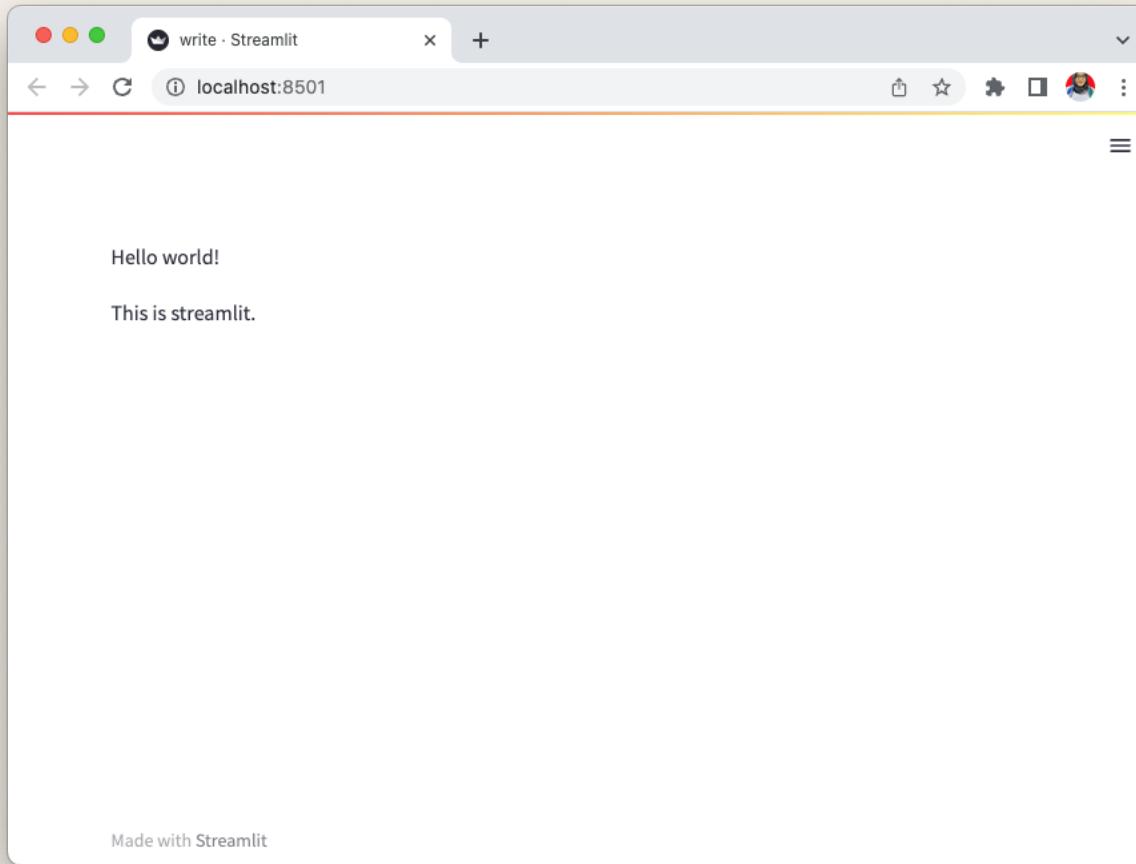
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://10.1.161.13:8501

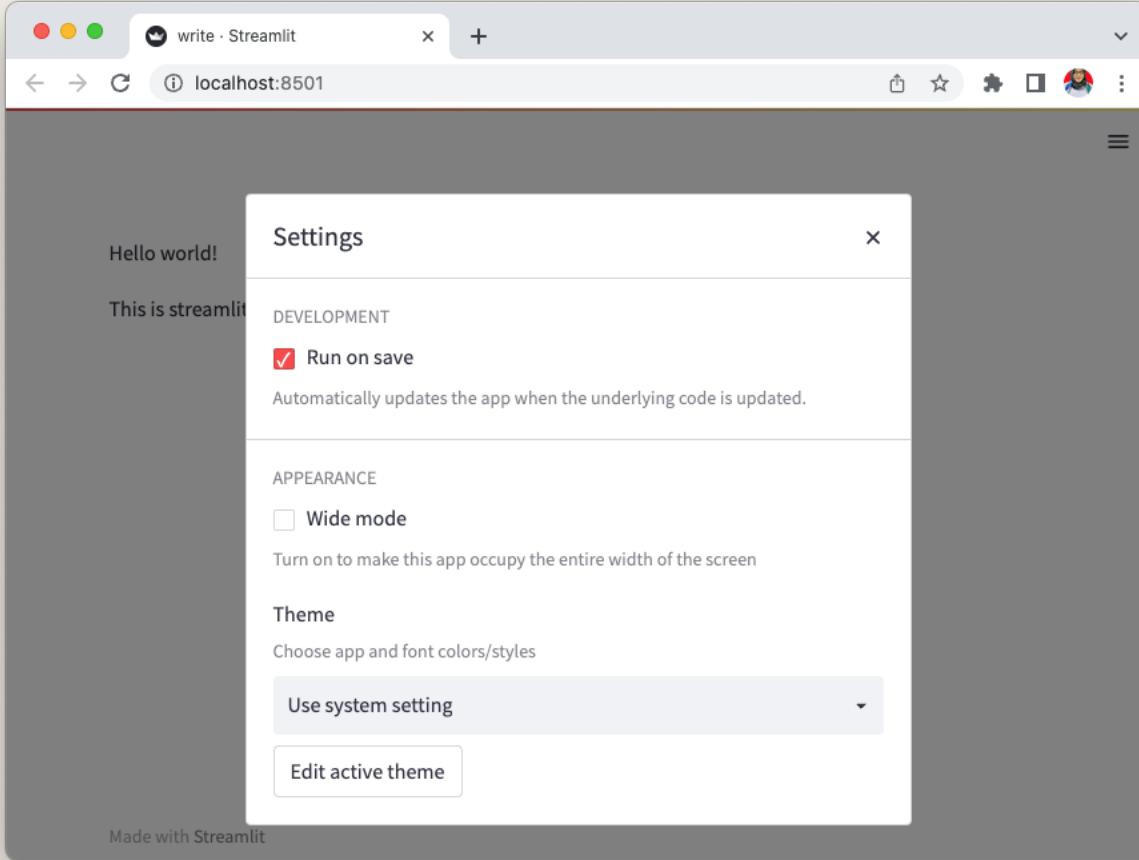
For better performance, install the Watchdog module:

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

# Konsep Streamlit (Run Script)



# Konsep Streamlit (Run Script)



Ketika **Run on save** aktif maka otomatis dilakukan perubahan tampilan ketika save file



# Konsep Streamlit (Run Script)

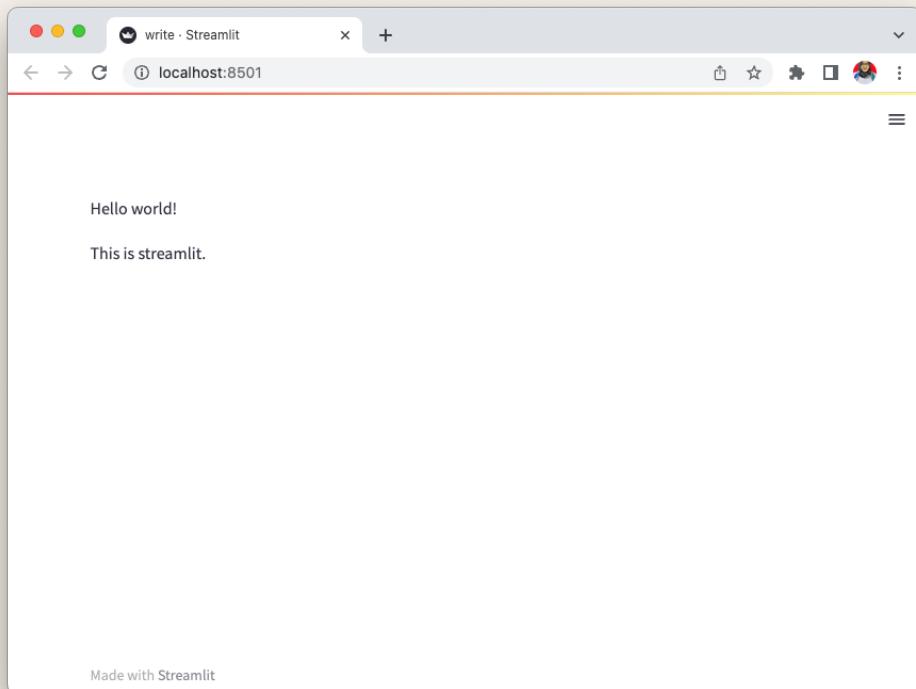
Ketika Streamlit di eksekusi (**run**):

- Streamlit akan membuat (create) server local
- Aplikasi yang dibuat akan tampil di tab baru browser (default)
- Tab browser tersebut merupakan *canvas*, dimana kita dapat membuat chart, text, widget, table, dll

# Alur Membangun (*Development Flow*) Streamlit

Pada Streamlit menulis code pada IDE, dan run hasil code pada browser

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





# Alur Membangun (*Development Flow*) Streamlit

st.write() dapat digunakan untuk menampilkan :

- Text
- Data
- Chart/Figure (matplotlib, altair)
- dll



# Alur Data (*Data Flow*) Streamlit

- Streamlit akan membaca dan menjalankan ulang seluruh *source code* ketika:
- Source code diubah atau dimodifikasi
- Ketika ada interaksi widget dari user, contoh: *slider*, memasukan *input text*, *klik button*



## Alur Data (*Data Flow*) Streamlit

- Bagaimana dengan komputasi yang berat?
- Streamlit menyediakan `@st.cache` decorator, untuk mengatasi hal tersebut, yang akan dibahas pada modul selanjutnya

# Development: Streamlit Library

- Text Elements
- Display Style
- Media Elements
- Multipage Apps

# **Text Elements:**

## **Display Text**



# Text Elements

- st.markdown()
- st.title()
- st.header()
- st.subheader()
- st.caption()
- st.text()

# st.markdown ():

Streamlite support format penulisan dengan markdown, membuat format text dari plain-text editor

Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	<b>bold text</b>
Italic	<i>italicized text</i>
Blockquote	>blockquote
Ordered List	1. First item 2. Second item 3. Third item
Unordered List	- First item - Second item - Third item
Code	`code`
Horizontal Rule	---

<https://www.markdownguide.org/basic-syntax/>  
<https://www.markdownguide.org/cheat-sheet/>

# st.title ():

Menampilkan text dengan format title

```
st.title('This is a title')
```



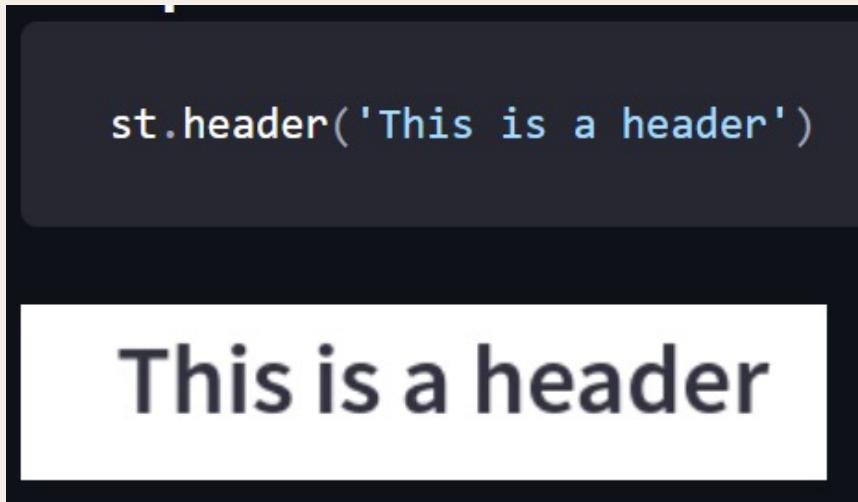
The image displays a screenshot of a Streamlit application. At the top, there is a dark gray header bar with a small white minus sign icon on the left. Below the header is a white rectangular content area. Inside this content area, the text "This is a title" is displayed in a large, bold, dark font.

<https://docs.streamlit.io/library/api-reference/text/st.title>

# st.header ():

Menampilkan text dengan format header

```
st.header('This is a header')
```



The image shows a dark-themed Streamlit application window. Inside, there is a single line of Python code: `st.header('This is a header')`. Below the code, the resulting output is displayed in a large, bold, dark font on a white background. The text reads "This is a header". The overall aesthetic is clean and modern.

<https://docs.streamlit.io/library/api-reference/text/st.header>

## st.subheader ():

Menampilkan text dengan format subheader

```
st.subheader('This is a subheader')
```

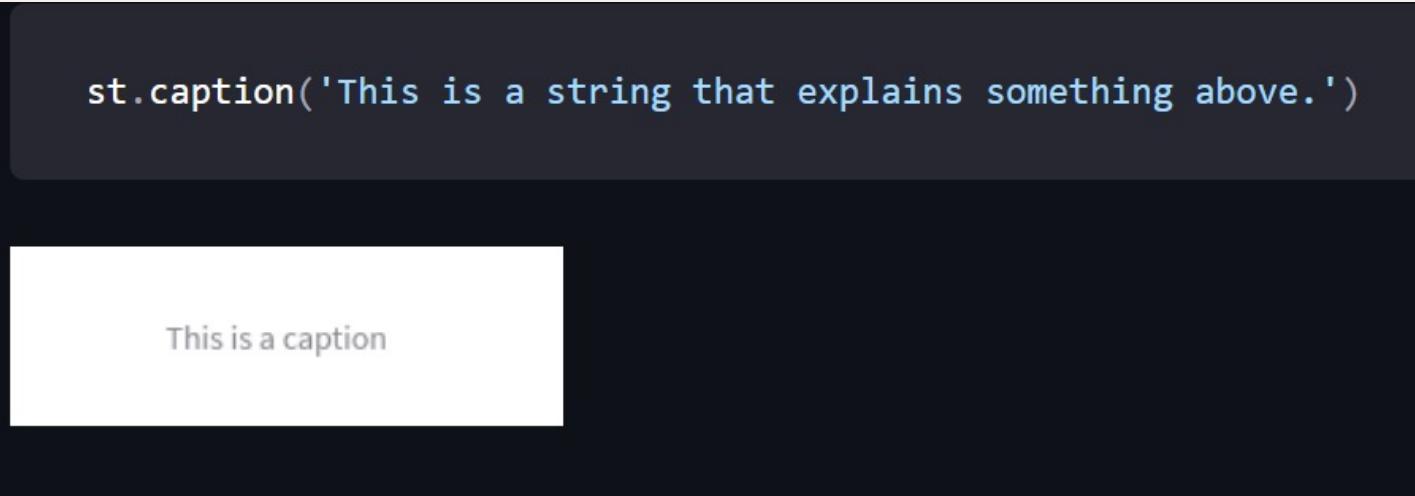
This is a subheader

<https://docs.streamlit.io/library/api-reference/text/st.subheader>

## st.caption ():

Menampilkan text dengan format font ukuran kecil (small font), biasa digunakan untuk catatan kaki, caption, dan penjelasan tambahan

```
st.caption('This is a string that explains something above.')
```



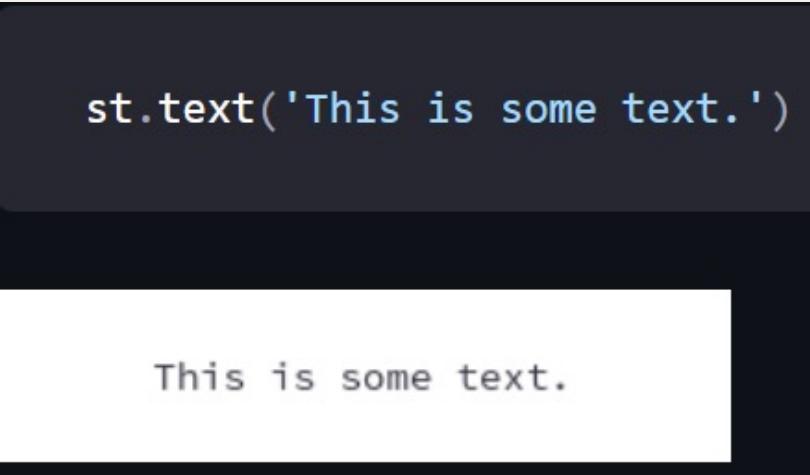
This is a caption

<https://docs.streamlit.io/library/api-reference/text/st.caption>

## st.text ():

Menampilkan text dengan format fix width dan preformatted text

```
st.text('This is some text.')
```



This is some text.

<https://docs.streamlit.io/library/api-reference/text/st.text>

# Text Elements

```
display_text.py
1 import streamlit as st
2
3 st.title('Ini adalah tulisan TITLE st.title() :smile: :smile:')
4 st.header('Ini adalah HEADER st.header() :icecream: 🍦')
5 st.subheader('Ini adalah Subheader st.subheader() 🎉')
6 st.caption('ini adalah TULISAN -CAPTION- st.caption()')
7 st.text('Ini adalah tulisan dari text DENGAN FORMAT st.text()')
8 st.write('Ini adalah tulisan dari write DENGAN FORMAT st.write()')
```

The screenshot shows a Streamlit application running at localhost:8504. The browser window has two tabs: 'write · Streamlit' and 'display\_text · Streamlit'. The 'display\_text · Streamlit' tab is active, displaying the following content:

**Ini adalah tulisan TITLE st.title()** 😊 😊

**Ini adalah HEADER st.header()** 🍦 🍦

**Ini adalah Subheader st.subheader()** 🎉

ini adalah TULISAN -CAPTION- st.caption()

Ini adalah tulisan dari text DENGAN FORMAT st.text()

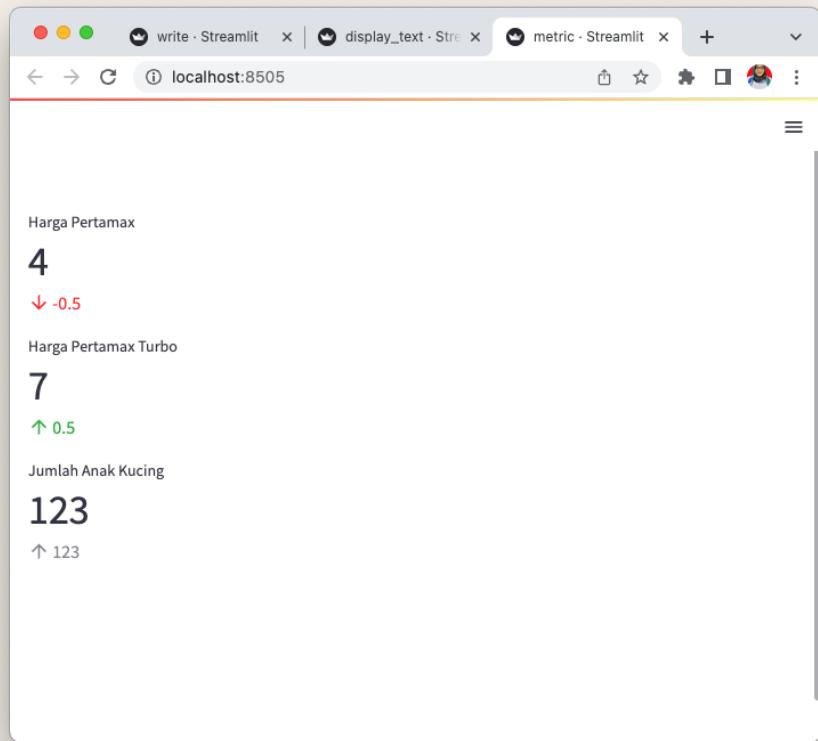
Ini adalah tulisan dari write DENGAN FORMAT st.write()

# **Text Elements: Metric & Columns**

# st.metric ():

Menampilkan text dengan format *metric* dalam huruf besar, tebal, dan indikator perubahan

```
metric.py
1 import streamlit as st
2
3 st.metric(label="Harga Pertamax", value=4, delta=-0.5)
4 st.metric(label="Harga Pertamax Turbo", value=7, delta=0.5)
5 st.metric(label="Jumlah Anak Kucing", value=123, delta=123, delta_color="off")
```

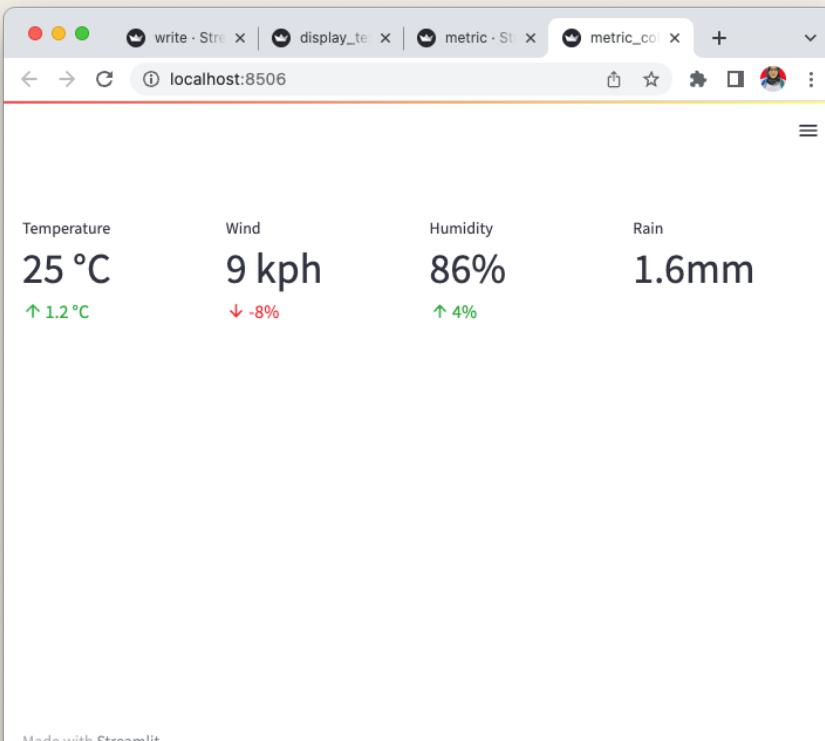


<https://docs.streamlit.io/library/api-reference/data/st.metric>

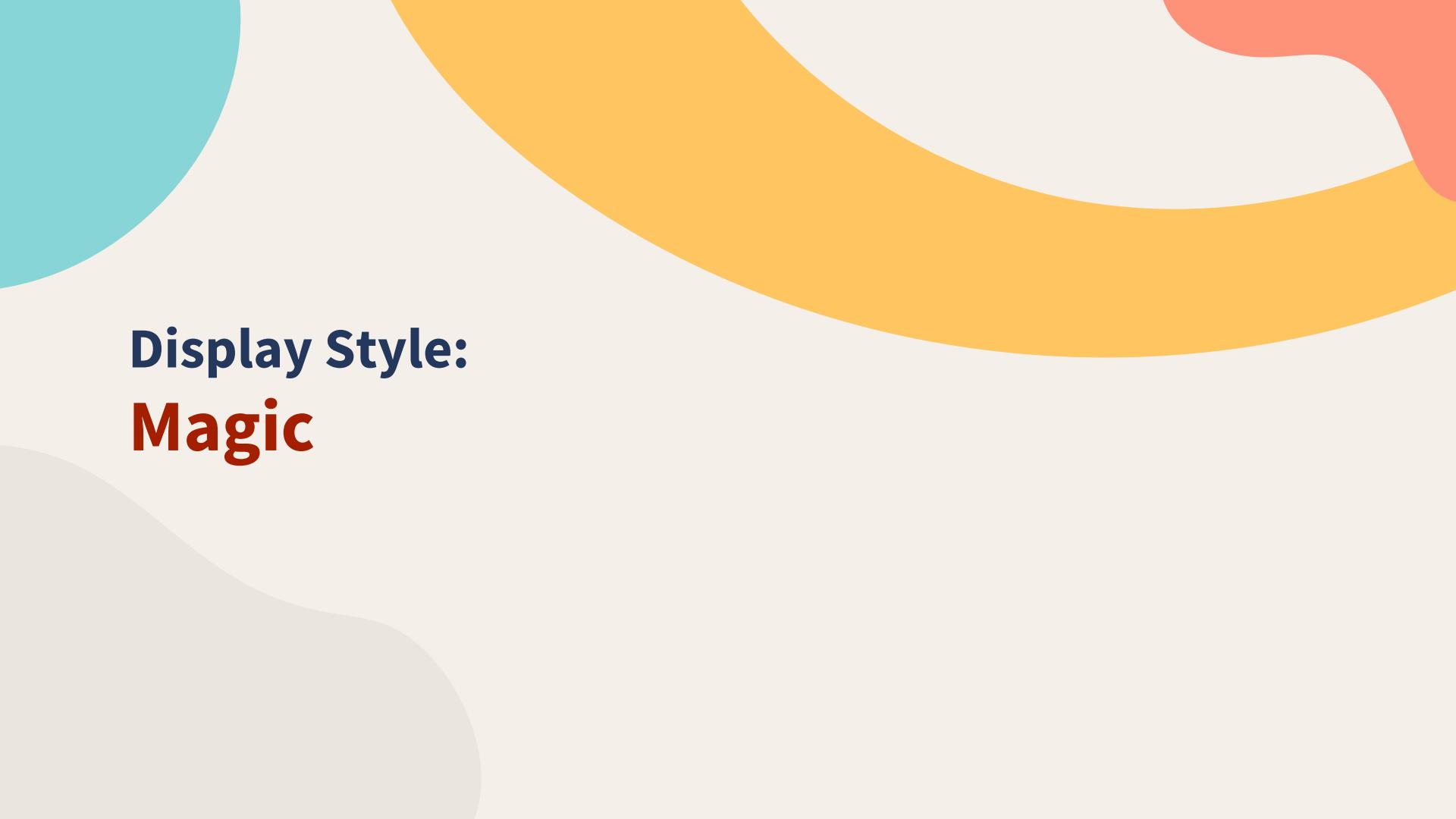
# st.columns ():

st.metric() juga dapat dikombinasikan dengan st.columns()

```
metric_column.py > ...
1  import streamlit as st
2
3  col1, col2, col3, col4 = st.columns(4)
4  col1.metric("Temperature", "25 °C", "1.2 °C")
5  col2.metric("Wind", "9 kph", "-8%")
6  col3.metric("Humidity", "86%", "4%")
7  col4.metric("Rain", "1.6mm")
```



<https://docs.streamlit.io/library/api-reference/data/st.metric>



**Display Style:**  
**Magic**



# Display Style Streamlit

Ada beberapa cara untuk menampilkan (*display*) data pada Streamlit, contohnya dengan menggunakan [magic](#).

# Display Style Streamlit

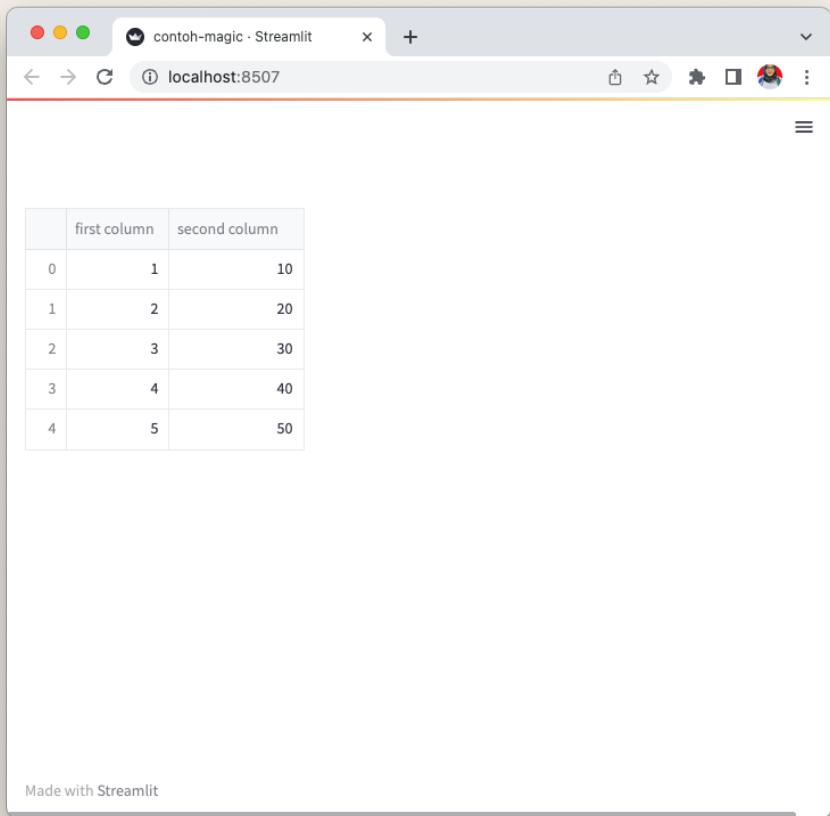
Perhatikan penggunaan magic pada baris code dibawah ini

```
⚡ contoh-magic.py > ...
1  import streamlit as st
2  import pandas as pd
3
4  df = pd.DataFrame({
5      'first column': [1,2,3,4,5],
6      'second column': [10,20,30,40,50]
7  })
8
9  df
```

# Display Style Streamlit

Tanpa perlu menuliskan perintah st.write dapat langsung menampilkan *data frame*

```
contoh-magic.py > ...
1  import streamlit as st
2  import pandas as pd
3
4  df = pd.DataFrame({
5      'first column': [1,2,3,4,5],
6      'second column': [10,20,30,40,50]
7  })
8
9  df
```



A screenshot of a web browser window titled "contoh-magic · Streamlit" at "localhost:8507". The page displays a data frame with two columns: "first column" and "second column". The data is as follows:

	first column	second column
0	1	10
1	2	20
2	3	30
3	4	40
4	5	50

At the bottom of the page, a small footer says "Made with Streamlit".

# Test:

Buatlah *dataframe* sederhana untuk stok barang mini market



# **Media Elements:** **Image, Audio, & Video**

# Media Element: st.image()

```
# Media Elements st.image
# import modul Image dari Python Imaging Library

from PIL import Image
image = Image.open('C:\Streamlit\photo\pantai.jpg')
st.image(image, caption='Hari yang cerah di Pantai yang indah')
```



Hari yang cerah di Pantai yang indah

<https://docs.streamlit.io/library/api-reference/media/st.image>

<https://unsplash.com/>

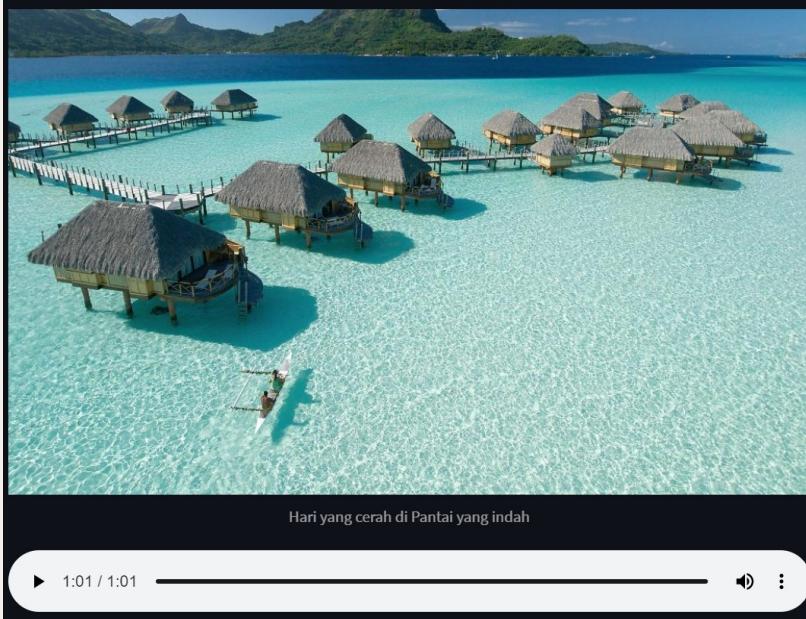
<https://pixabay.com/>

# Media Element: st.audio()

```
# Media Elements st.audio

audio_file = open('C:\Streamlit\sound\sound.mp3', 'rb')
audio_bytes = audio_file.read()

st.audio(audio_bytes, format='audio/mp3')
```



<https://docs.streamlit.io/library/api-reference/media/st.image>

<https://unsplash.com/>

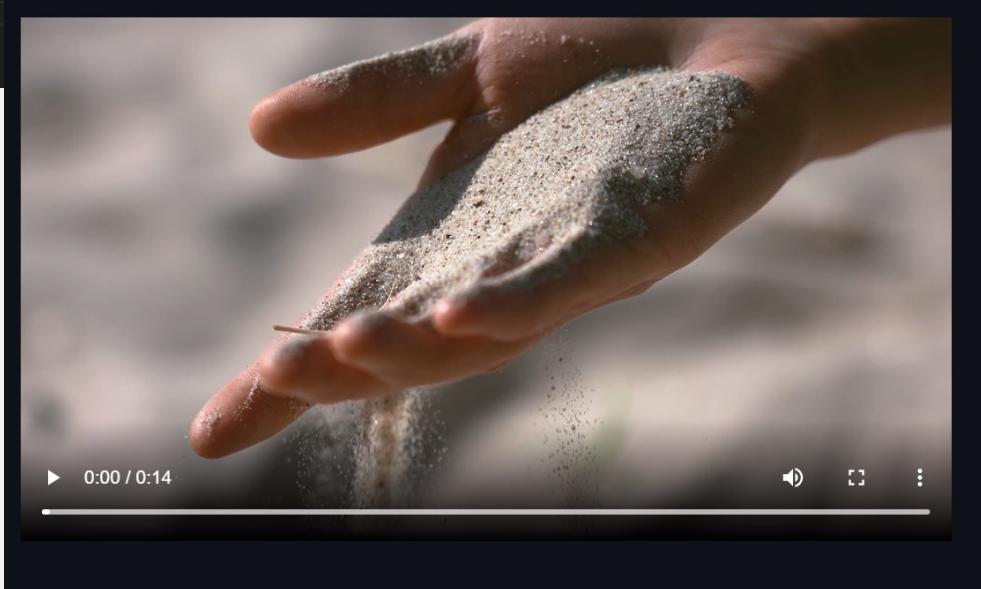
<https://pixabay.com/music/>

# Media Element: st.video()

```
# Media Elements st.video

video_file = open('C:\\Streamlit\\clip\\sand.mp4', 'rb')
video_bytes = video_file.read()

st.video(video_bytes)
```



<https://docs.streamlit.io/library/api-reference/media/st.video>  
<https://pixabay.com/videos/>

# Multipage: Native Multipage Apps

# Multipages Apps dengan function

```
import streamlit as st

def main_page():
    st.markdown("# Main page 🎉")
    st.sidebar.markdown("# Main page 🎉")

def page2():
    st.markdown("# Page 2 🌟")
    st.sidebar.markdown("# Page 2 🌟")

def page3():
    st.markdown("# Page 3 🎉")
    st.sidebar.markdown("# Page 3 🎉")

page_names_to_funcs = {
    "Main Page": main_page,
    "Page 2": page2,
    "Page 3": page3,
}

selected_page = st.sidebar.selectbox("Select a page", page_names_to_funcs.keys())
page_names_to_funcs[selected_page]()
```

<https://blog.streamlit.io/introducing-multipage-apps/>



# Streamlit Native Multipage Apps

- Native multipages pada streamlit adalah sebuah fitur baru yang hadir pada streamlit versi 1.10.0 pada 2 Juni 2022
- Dengan fitur ini kita dapat membuat lebih dari satu halaman dengan mudah, tanpa perlu membuat link antara satu halaman dengan halaman lain

# Streamlit Native Multipage Apps

Cara membuat multipages pada streamlit dengan struktur seperti dibawah ini

```
Home.py # This is the file you run with "streamlit run"  
└── pages/  
    ├── About.py # This is a page  
    ├── 2_Page_two.py # This is another page  
    └── 3_😊_three.py # So is this
```

Halaman\_utama.py  
pages  
 halaman\_2.py  
 halaman\_3.py  
 ...  
 halaman\_10.py

```
my_app  
├── streamlit_app.py      <-- Your main script  
└── pages  
    ├── page_2.py          <-- New page 2!  
    └── page_3.py          <-- New page 3!
```



# Streamlit Native Multipage Apps

- Ketentuan Streamlit Native Multipages:  
Hanya file .py di dalam directory pages/ yang akan ditampilkan, seluruh directory dan subdirectory akan diabaikan

<https://docs.streamlit.io/library/get-started/multipage-apps>



# Streamlit Native Multipage Apps

Nama valid yang digunakan pada pages:

- Number, terdiri dari angka
- Separator, bisa \_ atau -
- Label, terdiri dari huruf
- Extension, harus selalu .py

<https://docs.streamlit.io/library/get-started/multipage-apps>



# Streamlit Native Multipage Apps

Aturan urutan file pada pages:

- File dengan angka (number) akan muncul sebelum file dengan angka
- File diurutkan berdasarkan angka jika ada, diikuti dengan huruf (title)
- Ketika file diurutkan, streamlit memperlakukan angka (number) sebagai angka bukan string, artinya 03 sama dengan 3

<https://docs.streamlit.io/library/get-started/multipage-apps>

# Streamlit Native Multipage Apps

Aturan urutan file pada pages:

## Examples:

### Filename

### Rendered label

1 - first\_page.py

first page

12\_monkeys.py

monkeys

123.py

123

123\_hello\_dear\_world.py

hello dear world

\_12\_monkeys.py

12 monkeys

<https://docs.streamlit.io/library/get-started/multipage-apps>

# Streamlit Native Multipage Apps

Nama file pada pages support emoji:



## Tip

Emojis can be used to make your page names more fun! For example, a file named `🏡_Home.py` will create a page titled "🏡 Home" in the sidebar.

You can also add emojis! 🎉 Try renaming the script files to:

- `01_🎈_main_page.py`
- `pages/02_❄️_page2.py`
- `pages/03_🎉_page3.py`

<https://docs.streamlit.io/library/get-started/multipage-apps>



# Streamlit Native Multipage Apps

Link emoji:

<https://share.streamlit.io/streamlit/emoji-shortcodes>

<https://www.webfx.com/tools/emoji-cheat-sheet/>

<https://emojipedia.org/>

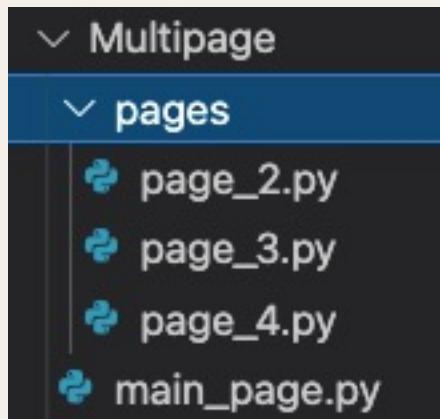
<https://github.com/ikatyang/emoji-cheat-sheet/>

<https://docs.streamlit.io/library/get-started/multipage-apps>

# Streamlit Native Multipage Apps

Test Mencoba membuat multipage:

Struktur Folder



<https://docs.streamlit.io/library/get-started/multipage-apps>

# Test Mencoba membuat multipage Apps

## Main Page:

```
🐍 main_page.py ×  
stenv > App > 🐍 main_page.py  
1 import streamlit as st  
2  
3 st.markdown("# Main Page :smile:")  
4 st.sidebar.markdown("Halaman Utama :woman-gesturing-ok:")  
5  
6 st.write("Haloo ini halaman ke-1")
```

## Page 2:

```
🐍 page_2.py ×  
stenv > App > pages > 🐍 page_2.py  
1 import streamlit as st  
2  
3 st.markdown("# Page 2 :yum:")  
4 st.sidebar.markdown("Halaman kedua :woman-raising-hand:")  
5  
6 st.write("Haiii ini halaman ke-2")
```

## Page 3:

```
🐍 page_3.py ×  
stenv > App > pages > 🐍 page_3.py  
1 import streamlit as st  
2  
3 st.markdown("# Page 3 :satisfied:")  
4 st.sidebar.markdown("Halaman ketiga :rocket:")  
5  
6 st.write("Dan ini halaman ke-3 hai haii...")  
7
```

<https://docs.streamlit.io/library/get-started/multipage-apps>