



# LLM-Based Tools and Gemini API Integration for Data Scientists

+ Partnerships Hacktiv8

This Program is part of the AI Opportunity Fund: Asia Pacific, in collaboration with AVPN and supported by [Google.org](#) and the Asian Development Bank.



# **Build an LLM-powered ChatBot with Streamlit**

## **Session 5**

# Learning Session

## Introduction to AI & Generative AI with Gemini

- Pengenalan AI, Generative AI dan LLM
- AI Ethics
- Pengenalan Chatbot
- Hallucinations & Prompting Techniques
- Pengenalan Gemini
- Hands-On

## Session 2

### Implementing Generative AI with Gemini

- Teknik Prompting Lanjutan
- Konfigurasi Google Gemini
- Pemanggilan Fungsi

## Session 3

### RAG with LangChain

- Pengenalan RAG
- Vector Database
- Pengenalan LangChain
- Pengenalan Llama
- Hands-On

## Session 4

## Session 5

### Building Applications with LLM

- Pengenalan Streamlit
- Pembuatan ChatBot LLM dengan Streamlit
- Pengenalan AI Agents
- Mendeploy Aplikasi Streamlit ke Cloud

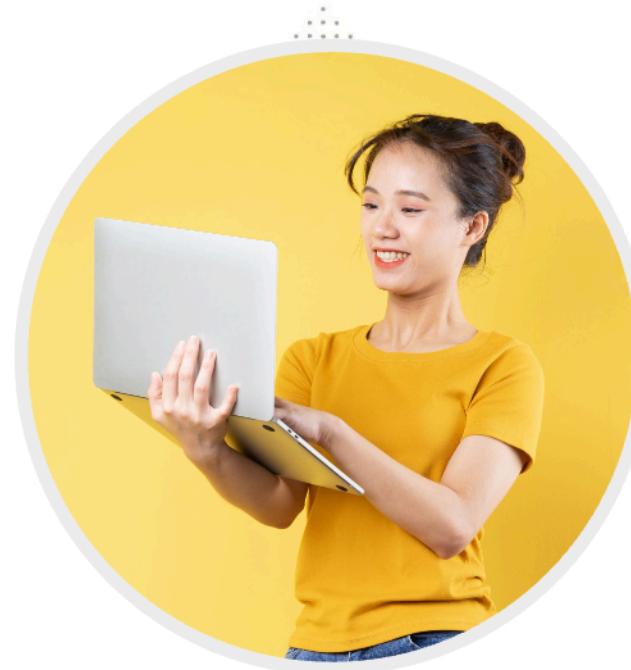
# Training Guidelines

Untuk mengikuti materi pelatihan berikut ada beberapa hal yang harus diperhatikan oleh peserta:

- Peserta diharapkan sudah memiliki akses ke Google Collab.
- Peserta sudah membuat akun pada <https://console.groq.com/>
- **Active Participation:** peserta harus terlibat aktif dengan materi pelatihan, mencatat poin-poin penting dan penjelasan yang diberikan oleh instruktur.

# List of Contents

- Pengenalan Streamlit
- Pembuatan ChatBot LLM dengan Streamlit
- Pengenalan AI Agents
- Mendeploy Aplikasi Streamlit ke Cloud



# Objectives

- Peserta memahami mengenai model Llama, jenis-jenisnya, dan cara menggunakannya.
- Peserta dapat mengimplementasikan Llama untuk membantu pekerjaan sehari-hari.
- Peserta mengenal LangChain dan berbagai kegunaannya.
- Peserta dapat mengimplementasikan LangChain untuk membantu pekerjaan sehari-hari.



# Streamlit

## Apa itu Streamlit?

Streamlit adalah framework open-source berbasis Python yang memudahkan pengembang dalam membangun aplikasi web interaktif, khususnya di bidang sains data dan machine learning. Dengan Streamlit, Anda dapat membuat antarmuka pengguna yang menarik tanpa perlu menguasai HTML, CSS, atau JavaScript.



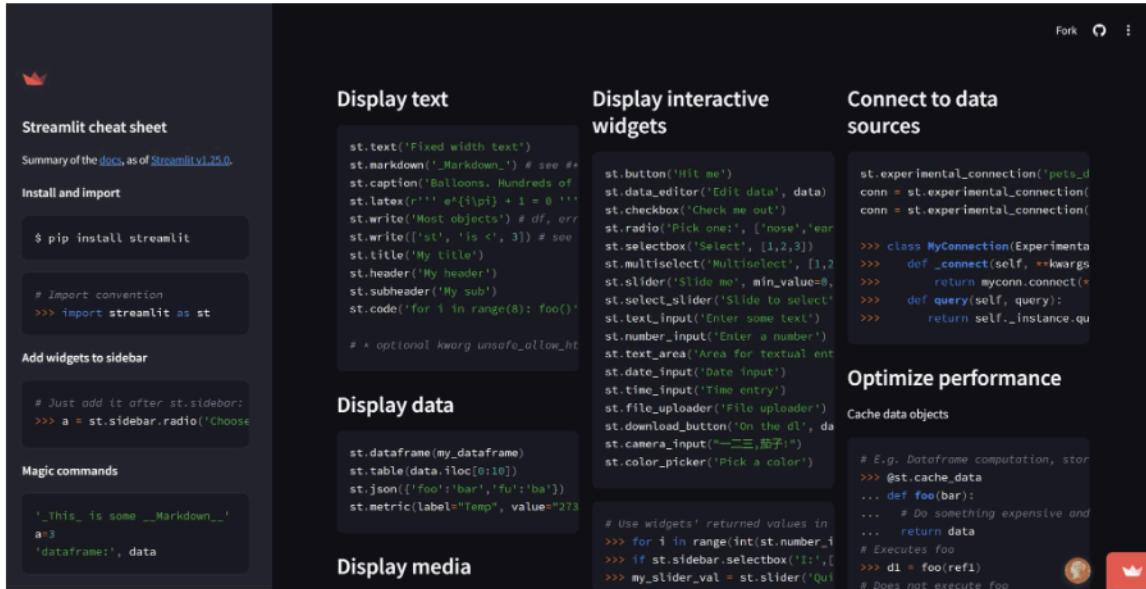
# Kenapa Streamlit?

- **Open-Source Python Library:** Streamlit merupakan library open source python sehingga perkembangannya sangat cepat
- **Interactive User Interfaces:** Kamu bisa membangun tampilan interaktif web dengan mudah tanpa memiliki kemampuan mendalam di front-end development
- **Low-Code Development:** Streamlit memungkinkan membangun website dengan minimal coding
- **Extensive Compatibility:** Terintegrasi dengan Python libraries seperti pandas, scikit-learn, bahkan langchain, sehingga mudah untuk membuat aplikasi AI atau ML

# Getting Started with Streamlit

- Buat akun streamlit di <https://share.streamlit.io/>
- Akun ini akan digunakan untuk mendeploy streamlit dengan mudah

# Getting Started with Streamlit

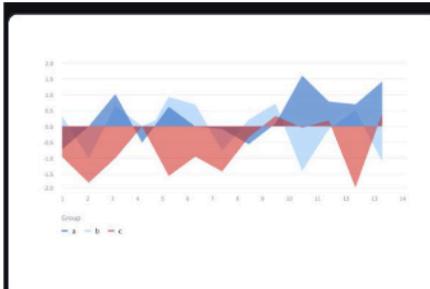


The screenshot shows a dark-themed Streamlit cheat sheet with several sections:

- Streamlit cheat sheet**: Summary of the docs, as of Streamlit v1.25.0.
- Install and import**: \$ pip install streamlit; # Import convention: >>> import streamlit as st
- Add widgets to sidebar**: # Just add it after st.sidebar: >>> a = st.sidebar.radio("Choose
- Magic commands**: !\_This\_ is some \_\_Markdown\_\_! a=3 'dataframe':, data
- Display text**: st.text('Fixed width text'); st.markdown('!Markdown!') # see #; st.caption('Balloons. Hundreds of st.latex(r''' e^{(pi)} + 1 = 0 ''') st.write('Most objects') # df, err st.write(['st', 'is <', 3]) # see st.title('My title'); st.header('My header'); st.subheader('My sub'); st.code('for i in range(8): foo()') # optional kwarg unsafe\_allow\_ho
- Display data**: st.dataframe(my\_dataframe); st.table(data.iloc[0:10]); st.json([{'foo':'bar','fu':'ba'}]); st.metric(label="Temp", value=273)
- Display media**
- Display interactive widgets**: st.button('Hit me'); st.data\_editor('Edit data', data); st.checkbox('Check me out!'); st.radio('Pick one!', ['nose','ear']); st.selectbox('Select', [1,2,3]); st.multiselect('Multiselect', [1,2]); st.slider('Slide me', min\_value=0); st.select\_slider('Slide to select'); st.text\_input('Enter some text!'); st.number\_input('Enter a number!'); st.text\_area('Area for textual ent
- Connect to data sources**: st.experimental\_connection('pets\_d conn = st.experimental\_connection( conn = st.experimental\_connection(
- Optimize performance**: # E.g. Dataframe computation, stor >>> @st.cache\_data ... def foo(bar): ... # Do something expensive and ... return data # Executes foo >>> d1 = foo(ref1) # Does not execute foo

Streamlit memiliki banyak sekali widget yang bisa kamu explore  
 (source: <https://cheat-sheet.streamlit.app/>)

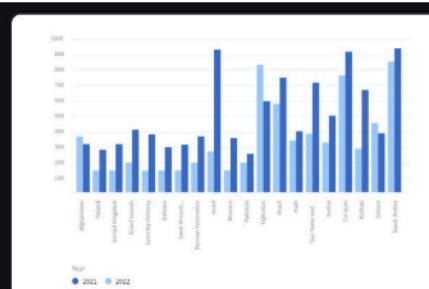
# Getting Started with Streamlit



## Simple area charts

Display an area chart.

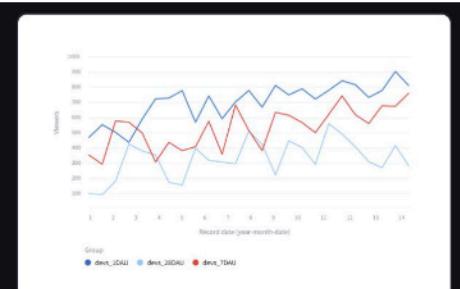
```
st.area_chart(my_data_frame)
```



## Simple bar charts

Display a bar chart.

```
st.bar_chart(my_data_frame)
```



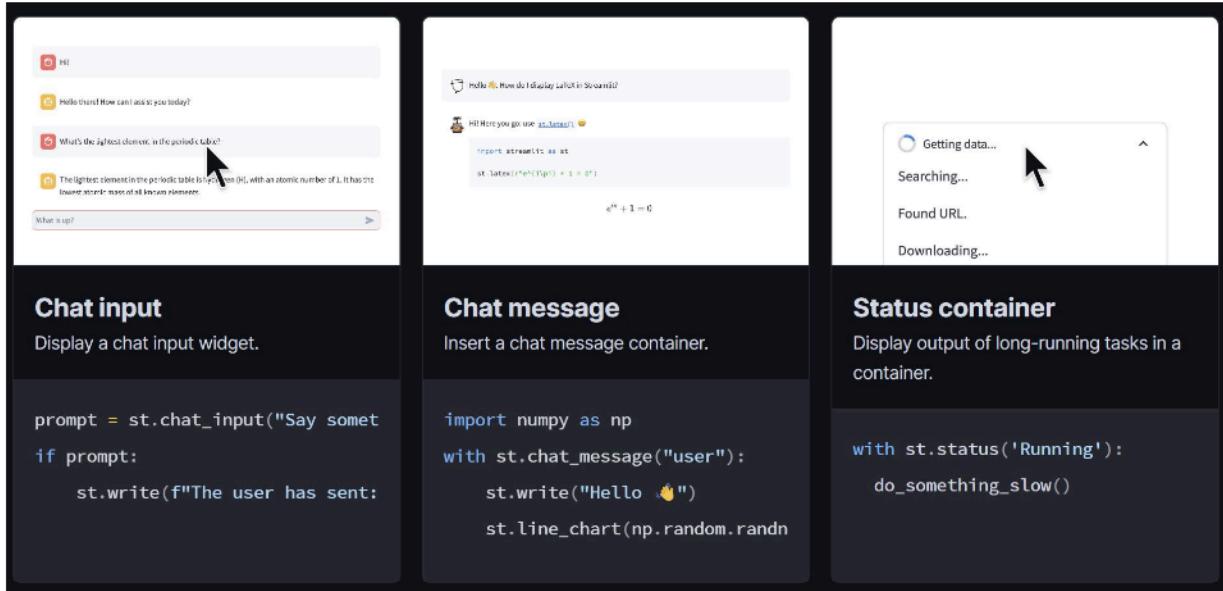
## Simple line charts

Display a line chart.

```
st.line_chart(my_data_frame)
```

Beberapa contoh widget yang dapat kamu gunakan: Chart Widget  
(source: <https://docs.streamlit.io/develop/api-reference/charts>)

# Getting Started with Streamlit



The image displays three side-by-side screenshots of Streamlit applications illustrating different types of widgets:

- Chat input:** Shows a chat interface where a user asks about the lightest element in the periodic table, and the application responds with "Hydrogen (H), with an atomic number of 1. It has the lowest atomic mass of all known elements." Below the interface, the code for creating a chat input is shown:

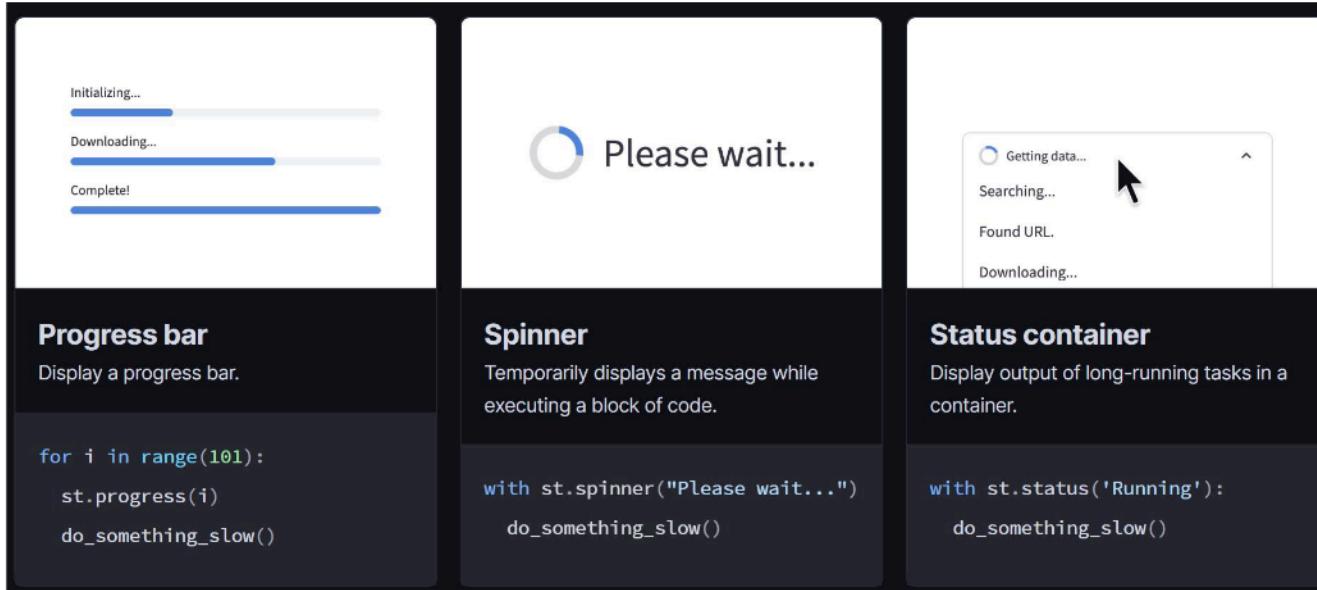
```
prompt = st.chat_input("Say something")  
if prompt:  
    st.write(f"The user has sent: {prompt}")
```
- Chat message:** Shows a chat interface where a user asks "How do I display LaTeX in Streamlit?", and the application responds with "import streamlit as st  
st.latex(r"\frac{dy}{dt} + 3 = 0")". Below the interface, the code for displaying LaTeX is shown:

```
import numpy as np  
  
with st.chat_message("user"):  
    st.write("Hello 👋")  
    st.line_chart(np.random.randn(10, 3))
```
- Status container:** Shows a progress bar indicating "Getting data..." followed by "Searching...", "Found URL.", and "Downloading...". Below the interface, the code for using a status container is shown:

```
with st.status('Running'):  
    do_something_slow()
```

Beberapa contoh widget yang dapat kamu gunakan: Chatbot Widgets  
(source: <https://docs.streamlit.io/develop/api-reference/chat>)

# Getting Started with Streamlit



Initializing...  
  
Downloading...  
  
Complete!  


### Progress bar

Display a progress bar.

```
for i in range(101):
    st.progress(i)
    do_something_slow()
```

 Please wait...

### Spinner

Temporarily displays a message while executing a block of code.

```
with st.spinner("Please wait...")
    do_something_slow()
```

 Getting data...
   
Searching...
   
Found URL...
   
Downloading...

### Status container

Display output of long-running tasks in a container.

```
with st.status('Running'):
    do_something_slow()
```

Beberapa contoh widget yang dapat kamu gunakan: Status Widgets (source: <https://docs.streamlit.io/develop/api-reference/status>)

Maju Bareng AI

## Presence - AI for Data Scientist

Buat yang belum absen,  
absen yuk! WAJIB



<https://bit.ly/absensi-data-mba>



# Streamlit for Chatbot

# GitHub Link

## Github

<https://github.com/adiptamartulandi/chatbot-streamlit-demo>



# AI Agents

# Apa itu AI Agents?

Agen Kecerdasan Buatan (AI) adalah program perangkat lunak yang dapat berinteraksi dengan lingkungannya seperti mengumpulkan data dan menggunakan data tersebut untuk melakukan tugas-tugas secara mandiri untuk mencapai tujuan yang telah ditetapkan.

AI Agents pada dasarnya adalah sekumpulan Tools (function calling) yang secara otomatis dapat langsung membuat keputusan menjalankan fungsi yang diberikan.

Misalnya, dalam Customer Center, AI Agent dapat secara otomatis mengajukan berbagai pertanyaan kepada pelanggan, mencari informasi dalam dokumen internal, dan memberikan solusi. Berdasarkan respons pelanggan, agen AI menentukan apakah dapat menyelesaikan pertanyaan tersebut sendiri atau perlu meneruskannya kepada manusia.

# Perbedaan AI Agents dengan AI Asisstants

	AI Agents	AI Assistant
Tujuan	Secara otomatis melakukan task sesuai Tools yang diberikan.	Menyarankan user melakukan tugasnya.
Kemampuan	Bisa melakukan task yang rumit seperti menghitung pengeluaran di bulan lalu.	Hanya bisa memberikan informasi, seperti cara mengecek saldo di ATM.
Interaksi	<b>Proactive;</b> goal-oriented.	<b>Reactive;</b> responds to user requests.

## Contoh Use Case



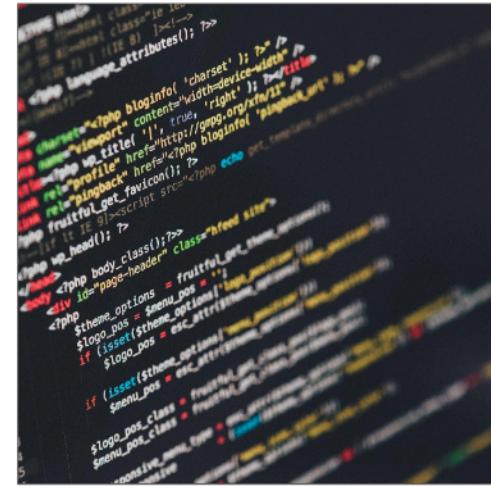
## Data Agent

Data Agent mampu menganalisis data untuk mendapatkan jawaban dari pertanyaan user, seperti total customer minggu lalu, product terjual bulan lalu, dan lainnya



## **Customer Service Agent**

Agent ini mampu menyelesaikan keluhan pelanggan dan langsung mengambil action dari keluhan tersebut



## Code Agents

Code Agent membantu developer dalam menyusun code nya menjadi lebih rapih, membantu memperbaiki bugs, dan lain lain.

Maju Bareng AI

## Presence - AI for Data Scientist

Buat yang belum absen,  
absen yuk! WAJIB



<https://bit.ly/absensi-data-mba>



# Exercises

*Hands-on*

## Latihan

Berikut adalah link google colab yang dapat anda gunakan sebagai referensi untuk latihan:

### Google Colab

[https://colab.research.google.com/drive/1zVJJghDYUGijQ9ADg--l8iTheZMMlb9O#scrollTo=xNHSJfzoP\\_ls](https://colab.research.google.com/drive/1zVJJghDYUGijQ9ADg--l8iTheZMMlb9O#scrollTo=xNHSJfzoP_ls)

Jangan lupa untuk duplicate  
Notebooknya ke Drive kamu ya!



# Quiz

*LLM-Based Tools and Gemini API Integration for Data Scientists*

## Quiz Session 5



<https://bit.ly/quizdata5>

Due Date:  
**Hari ini, 23.59 WIB**

Ingat, absensi juga dihitung  
dari pengisian Quiz.

# Certificate from Hacktiv8

Phase	Task	Due Date
Session 1	Presence	Session 1
Session 2	Presence & Quiz 1	Session 2
Session 3	Presence & Quiz 2	Session 3
Session 4	Presence & Quiz 3	Session 4
Session 5	Presence & Quiz 4	Session 5
Final Project	<b>Form Submission:</b> Final Project	H+2 Session 5

## ✨ Special Reward! ✨

Di akhir program, akan ada merchandise yang dibagikan untuk peserta yang beruntung yang telah menyelesaikan seluruh rangkaian program\* 🎁

\*Satu pemenang untuk satu training topics.





# Final Project

## *LLM-Based Tools and Gemini API Integration for Data Scientists*

# Final Project

Buatlah sebuah chatbot berbasis AI dengan use case dan konfigurasi parameter yang sesuai dengan kreativitas masing-masing. Chatbot harus menggunakan model AI (misalnya NLP/LLM) untuk memproses bahasa alami dan memberikan respon yang relevan kepada pengguna.

- **Contoh use case:** customer service bot, education bot, travel assistant, personal productivity assistant, dll.
- **Contoh parameter kreatif:** gaya bahasa (formal/santai), domain pengetahuan tertentu (misalnya kesehatan, edukasi, hobi), integrasi API eksternal, atau fitur tambahan seperti memory dan rekomendasi.

### Deliverables:

- URL repositori GitHub
- Screenshoots User Interface

### Contoh:



Chatbot untuk asistensi proses pembelajaran



Chatbot untuk analisis gizi pada makanan

*LLM-Based Tools and Gemini API Integration for Data Scientists*

# Final Project

Due Date:  
**H+2 Session 5 | 23.59 WIB**



<https://bit.ly/itdatafinal>

# Thank You

**Hacktiv8**

[www.hacktiv8.com](http://www.hacktiv8.com)

(021) 8067 5787  
[halo@hacktiv8.com](mailto:halo@hacktiv8.com)