

LANGKAH-LANGKAH MEMPERSIAPKAN PROJEK STREAMLIT

A. Persiapan

1. Pastikan python sudah ada di komputer Anda

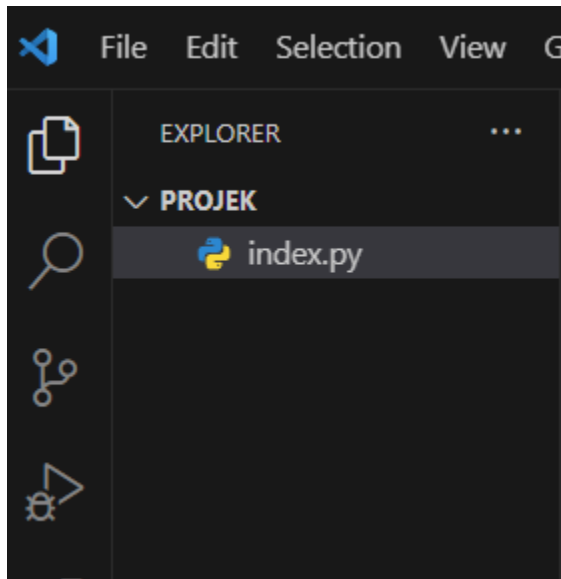
```
C:\Users\legion>python --version  
Python 3.12.7
```

2. Menginstall library yang dibutuhkan. Dilakukan di terminal atau command prompt.

```
pip install streamlit
```

B. Memastikan Streamlit berjalan

1. Pada direktori (folder) proyek yang digunakan, buat sebuah file .py.



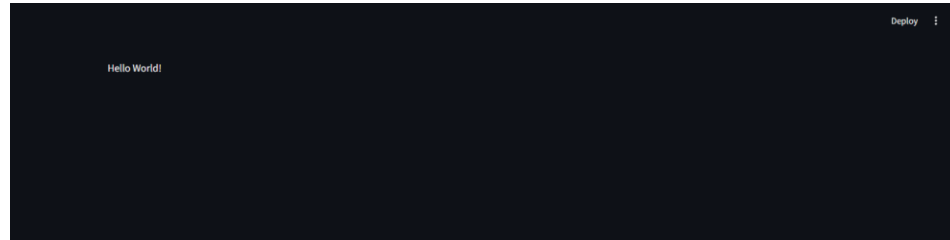
2. Buat kode starter untuk cek apakah fungsionalitas streamlit sudah benar. Disini buat contoh menampilkan “Hello World” pada tampilan yang akan kita gunakan.

```
index.py x  
index.py  
1 import streamlit as st  
2  
3 st.write("Hello World!")
```

3. Jalankan dengan menuliskan **streamlit run <nama file>** pada terminal.

```
streamlit run index.py
```

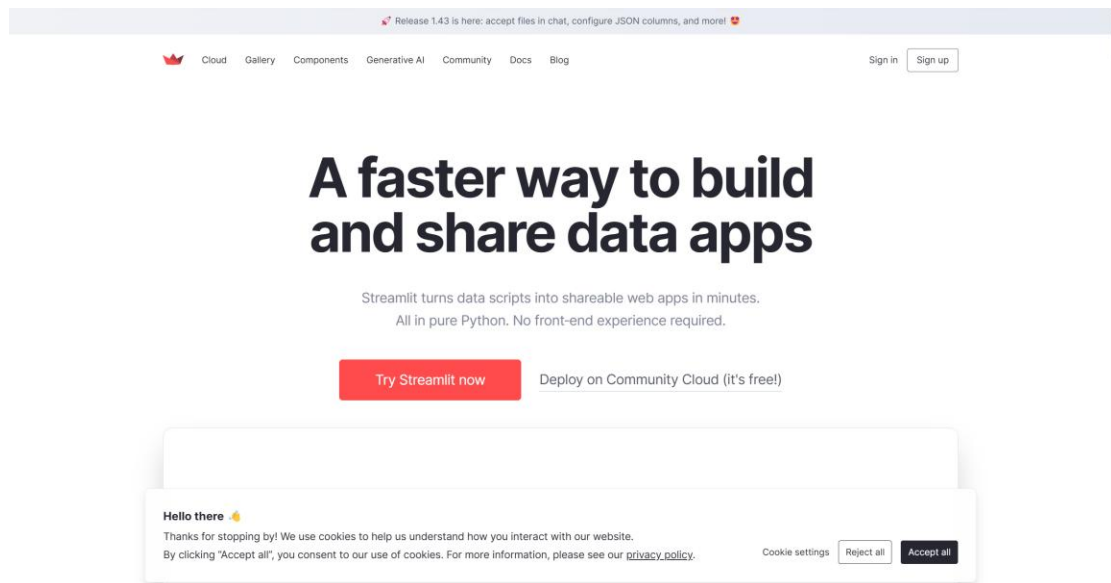
Hasil dari perintah tersebut akan menghasilkan url menuju halaman web streamlit yang dibuat



Jika sudah berhasil memunculkan tulisan “Hello World!”, maka streamlit sudah siap dikustomisasi.

C. Buat akun Streamlit

1. Buka website Streamlit dengan link <https://streamlit.io/>



2. Sign up (jika belum mempunyai akun) atau Sign in dengan akun yang sudah ada (dapat menggunakan akun Github) hingga tampilan seperti berikut:

A faster way to build and share data apps

Streamlit turns data scripts into shareable web apps in minutes.
All in pure Python. No front-end experience required.

Sign in to Streamlit Community Cloud

Continue with Google

Continue with GitHub

OR

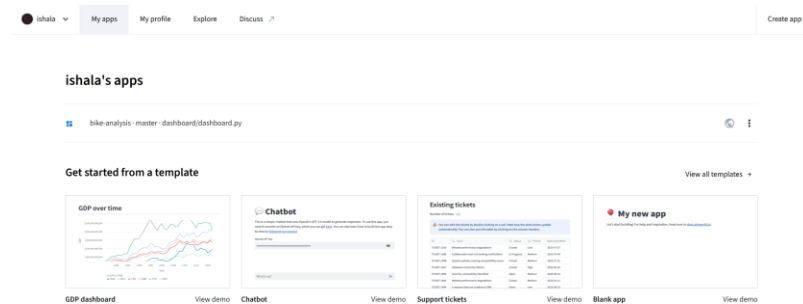
Email

Your email address

Continue

Don't have an account? [Sign up](#)

Halaman yang muncul selanjutnya contoh seperti berikut:



BASIC ELEMENTS YANG ADA DI STREAMLIT

Beberapa basic elements yang perlu diketahui dan umum digunakan yaitu sebagai berikut:

1. Write

Penggunaan write dapat menggunakan function `st.write()`, dengan contoh seperti yang ada di contoh sebelumnya.

2. Text

Penggunaan text cukup beragam, beberapa contohnya adalah sebagai berikut:

a. Markdown

```
st.markdown(
    """
    # My first app
    Halo semuanya, selamat datang di web Streamlit saya.
    """
)
```

My first app

Halo semuanya, selamat datang di web Streamlit saya.

b. Title

```
st.title("My First Project")
```

My First Project

c. Header

```
st.header('Pengembangan Dashboard')
```

Pengembangan Dashboard

d. Subheader

```
st.subheader('Pengembangan Dashboard')
```

Pengembangan Dashboard

e. Caption

```
st.caption('Copyright (c) 2023')
```

Copyright (c) 2023

f. Code

```
code = """def hello():  
    print("Hello, Streamlit!")"""  
st.code(code, language='python')
```

```
def hello():  
    print("Hello, Streamlit!")
```

g. Text

```
st.text('Selamat Datang')
```

Selamat Datang

3. Data Display

Data dapat ditampilkan di Streamlit, dengan beberapa format sebagai berikut:

a. Dataframe

```
df = pd.DataFrame({  
    'c1': [1, 2, 3, 4],  
    'c2': [10, 20, 30, 40],  
})  
  
st.dataframe(data=df, width=500, height=150)
```

	c1	c2
0	1	10
1	2	20
2	3	30
3	4	40

Note: Lebar dan tinggi dataframe dapat diatur dari parameter yang ada di fungsi dataframe.

b. Table

```
df = pd.DataFrame({  
    'c1': [1, 2, 3, 4],  
    'c2': [10, 20, 30, 40],  
})
```

```
st.table(data=df)
```

	c1	c2
0	1	10
1	2	20
2	3	30
3	4	40

BASIC WIDGETS YANG ADA DI STREAMLIT

Beberapa widgets yang dapat digunakan di dalam streamlit:

1. Text Input

```
name = st.text_input(label='Nama lengkap', value='')  
st.write('Nama: ', name)
```

Nama lengkap

Ikmal

Nama: Ikmal

2. Number Input

```
number = st.number_input(label='Umur')  
st.write('Umur: ', int(number), ' tahun')
```

Umur

1.00

Umur: 1 tahun

3. Datetime

```
date = st.date_input(label='Tanggal lahir', min_value=datetime.date(1900, 1, 1))
st.write('Tanggal lahir:', date)
```

Tanggal lahir

2025/03/07

Tanggal lahir: 2025-03-07

4. Button

```
if st.button('Say hello'):
    st.write('Hello there')
```

Say hello

Hello there

5. Checkbox

```
agree = st.checkbox('I agree')
if agree:
    st.write('Welcome to MyApp')
```

☒ I agree

Welcome to MyApp

6. Radio Button

```
genre = st.radio(
    label="What's your favorite movie genre",
    options=('Comedy', 'Drama', 'Documentary'),
    horizontal=False
)
```

What's your favorite movie genre

☒ Comedy

☐ Drama

☐ Documentary

BASIC LAYOUTS YANG ADA DI STREAMLIT

Beberapa layouts yang sering digunakan sebagai berikut:

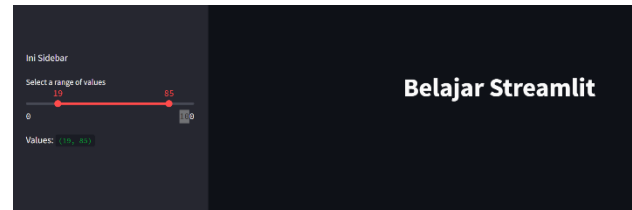
1. Sidebar

```
st.title('Belajar Streamlit')

with st.sidebar:

    st.text('Ini Sidebar')

    values = st.slider(
        label='Select a range of values',
        min_value=0, max_value=100, value=(0, 100)
    )
    st.write('Values:', values)
```



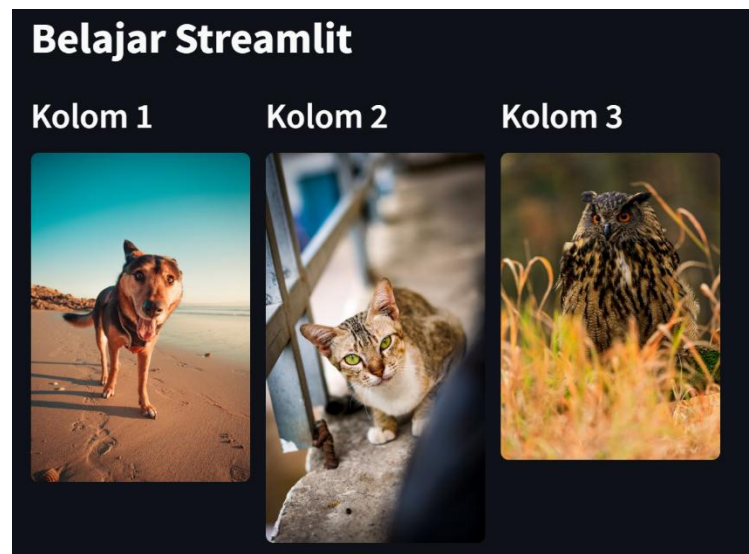
2. Columns

```
st.title('Belajar Streamlit')
col1, col2, col3 = st.columns(3)

with col1:
    st.header("Kolom 1")
    st.image("https://static.streamlit.io/examples/dog.jpg")

with col2:
    st.header("Kolom 2")
    st.image("https://static.streamlit.io/examples/cat.jpg")

with col3:
    st.header("Kolom 3")
    st.image("https://static.streamlit.io/examples/owl.jpg")
```



PROJEK HITUNG LUAS DAN KELILING BANGUN DATAR

1. Buat sebuah file .py baru atau tetap gunakan file dari contoh sebelumnya.
2. Pada file tersebut, tuliskan **import streamlit as st** pada bagian awal file.
3. Buat beberapa fungsi baik untuk hitung luas maupun hitung keliling

```
# Hitung Luas
def luas_segitiga(a, t):
    return (a * t) / 2

def luas_persegi_panjang(p, l):
    return p * l

def luas_jajar_genjang(a, t):
    return a * t
```

```
# Hitung Keliling
def keliling_segitiga(a, b, c):
    return a + b + c

def keliling_persegi_panjang(p, l):
    return 2 * (p + l)

def keliling_jajar_genjang(a, b):
    return 2 * (a + b)
```

4. Masukkan setiap fungsi ke dalam dua dictionary, hal ini bertujuan untuk nantinya dijadikan opsi

```
hitungLuas = {
    "Luas Segitiga": {
        "Fungsi": luas_segitiga,
        "Inputan": ['Alas', 'Tinggi']
    },
    "Luas Persegi Panjang": {
        "Fungsi": luas_persegi_panjang,
        "Inputan": ['Panjang', 'Lebar']
    },
    "Luas Jajar Genjang": {
        "Fungsi": luas_jajar_genjang,
        "Inputan": ['Alas', 'Tinggi']
    }
}
```

```
hitungKeliling = {
    "Keliling Segitiga": {
        "Fungsi": keliling_segitiga,
        "Inputan": ['Sisi A', 'Sisi B', 'Sisi C']
    },
    "Keliling Persegi Panjang": {
        "Fungsi": keliling_persegi_panjang,
        "Inputan": ['Panjang', 'Lebar']
    },
    "Keliling Jajar Genjang": {
        "Fungsi": keliling_jajar_genjang,
        "Inputan": ['Sisi A', 'Sisi B']
    }
}
```

5. Buat judul dan sebuah Select Box untuk memilih operasi perhitungan apa yang akan dilakukan, tepat dibawah kode-kode fungsi yang sudah dibuat

```
st.title("Aplikasi Hitung Bangun Datar")

opt = st.selectbox(
    label="Pilih operasi perhitungan",
    options=['Hitung Luas', 'Hitung Keliling']
)
```

Aplikasi Hitung Bangun Datar

Pilih operasi perhitungan

Hitung Keliling

6. Buat sebuah fungsi untuk melakukan opsi

```
def pilih_rumus(option):  
    allRumus = {}  
  
    if (option == 'Hitung Luas'):  
        allRumus = hitungLuas  
    else:  
        allRumus = hitungKeliling  
  
    return allRumus
```

7. Buat kode-kode untuk mengintegrasikan fungsi **pilih_rumus** dengan input dari Select Box menggunakan Radio Button.

```
opt = st.selectbox(  
    label="Pilih operasi perhitungan",  
    options=['Hitung Luas', 'Hitung Keliling']  
)  
  
all_rumus = pilih_rumus(opt)  
  
pilih_hitung = st.radio(  
    label='Pilih Hitung',  
    options=all_rumus.keys(),  
    horizontal=True  
)
```

Sehingga tampilan sekarang seperti ini



8. Buat kode mengenai inputan yang fleksibel dengan opsi yang sudah dipilih menggunakan Number Input.

```
inputs = [st.number_input(label, value=0.0) for label in all_rumus[pilih_hitung]["Inputan"]]
```



9. Terakhir, buat kode untuk menampilkan hasil perhitungan dengan menggunakan Button dan Markdown

```
if st.button('Hitung'):
    hasil = all_rumus[pilih_hitung]["Fungsi"](*inputs)
    st.markdown(f'<h2 style="color:green; text-align:center;">Hasil: {hasil}</h2>',
                unsafe_allow_html=True)
```

Maka contoh hasil terakhirnya adalah seperti ini

Aplikasi Hitung Bangun Datar

Pilih operasi perhitungan

Hitung Keliling

Pilih Hitung

☒ Keliling Segitiga ☐ Keliling Persegi Panjang ☐ Keliling Jajar Genjang

Sisi A

20.00 - +

Sisi B

30.00 - +

Sisi C

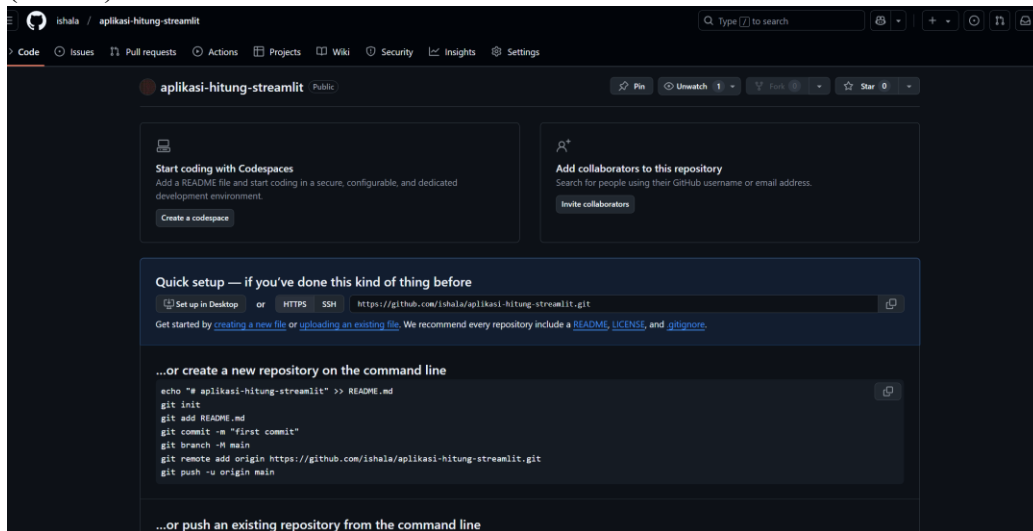
100.00 - +

Hitung

Hasil: 150.0

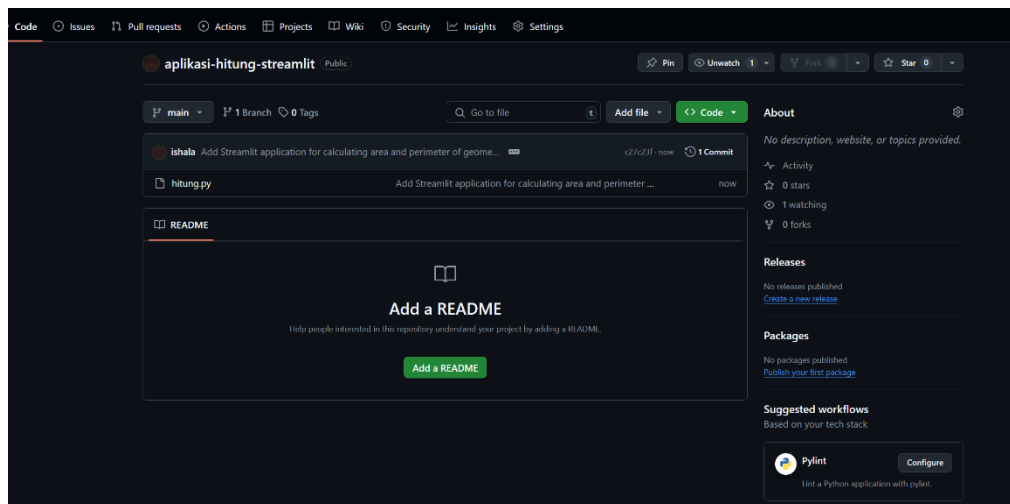
DEPLOY DENGAN STREAMLIT

1. Buat repository baru di Github dengan nama bebas, hingga tampilan seperti berikut (contoh)



2. Koneksikan dengan direktori lokal menggunakan terminal, dengan runtutan berikut:
 - Arahkan direktori terminal hingga pada root direktori proyek Anda
 - Jalankan perintah-perintah ini dengan urut
 - git init
 - git commit -m "first commit"
 - git branch -M main (sesuaikan dengan branch anda, biasanya master)
 - git remote add origin <remote repository anda>
 - git push -u origin main (sesuaikan dengan branch anda, biasanya master)

Sekarang contoh tampilan seperti ini pada repository



3. Sekarang buka halaman profil Streamlit yang sudah login dengan akun Github dengan contoh sebagai berikut, lalu pilih tombol Create App

The screenshot shows the Streamlit profile page for user 'ishala'. The navigation bar includes 'ishala', 'My apps', 'My profile', 'Explore', 'Discuss', and a 'Create app' button in the top right corner, which is highlighted with a red arrow. Below the navigation bar, the page title is 'ishala's apps'. A sub-header shows 'bike-analysis - master - dashboard/dashboard.py'. The main section is titled 'Get started from a template' with a link to 'View all templates'. It displays four templates: 'GDP dashboard' (a line chart), 'Chatbot' (a chat interface), 'Support tickets' (a table of tickets), and 'Blank app' (a simple landing page). Each template has a 'View demo' link.

4. Pilih Deploy a public app from Github

[← Back](#)

What would you like to do?



Deploy a public app from GitHub

My code is ready on a GitHub repo, and it is totally awesome.

[Deploy now](#)



Deploy a public app from a template

I want to see what kind of amazing concoctions you have for me.

[Check out templates](#)



Deploy a private app in Snowflake

I want unlimited enterprise-grade apps, with the security of Snowflake.

[Start trial →](#)

5. Pilih repository yang akan dideploy, pastikan bagian branch dan Main file path benar. File path disesuaikan dengan file kode utama.

[← Back](#)

Deploy an app

Repository [Paste GitHub URL](#)

ishala/aplikasi-hitung-streamlit

Branch

main

Main file path

hitung.py

App URL (optional)

aplikasi-hitung-app .streamlit.app

Domain is available

[Advanced settings](#)

[Deploy](#)

Anda dapat mengisi URL yang akan ditampilkan, lalu Klik Deploy
Tunggu hingga proses selesai

Hingga proses deploy selesai, dan aplikasi Anda sudah berjalan dengan Streamlit.