📱 Sesi 3

Dasar Streamlit

Hari 2 - 120 menit

Membangun Aplikasi Web Interaktif

🎯 Objektif Sesi

Setelah sesi ini, Anda dapat:

- ✅ Setup aplikasi Streamlit
- ✅ Membuat layout dengan columns & containers
- ✅ Memakai widgets interaktif
- ✅ Menampilkan metrics & kartu KPI
- ✅ Menampilkan data table
- ✅ Menambahkan tombol & aksi
- ✅ Mengelola input pengguna

🚀 Apa itu Streamlit?

Streamlit = framework Python untuk data apps

- 🐍 **Pure Python:** Tanpa HTML/CSS/JS
- ⚡ **Cepat:** Dari prototipe ke produksi dalam hitungan jam
- 🎨 **Indah:** UI modern bawaan
- 🔄 **Reaktif:** Auto-refresh saat kode berubah
- 📱 **Responsif:** Nyaman di mobile
- 🆓 **Hosting gratis:** Streamlit Cloud

Tagline: "Cara tercepat membangun data apps"

💡 Mengapa Streamlit?

Web Tradisional vs Streamlit

Tradisional (Flask/Django):

```
@app.route('/data')
def show_data():
    return render_template('data.html', data=df)
```

- Template HTML
- Styling CSS
- JavaScript untuk interaktivitas

  = **Berjam-jam/Hari**

💡 Mengapa Streamlit?

Streamlit:

```
import streamlit as st
st.dataframe(df)
```

= Hitungan detik!

🏗️ Struktur App

```
import streamlit as st

# 1. Konfigurasi halaman (harus paling atas!)
st.set_page_config(title="Aplikasi Saya", layout="wide")

# 2. Load data
@st.cache_data
def load_data():
    return pd.read_parquet('data.parquet')

df = load_data()

# 3. Bangun UI
st.title("Dashboard Saya")
st.dataframe(df)
```

1️⃣ Elemen Teks

```
# Judul
st.title("🎯 Judul Utama")
st.header("Header")
st.subheader("Subheader")

# Teks
st.text("Teks biasa")
st.markdown("**Bold** dan *italic*")
st.caption("Teks kecil/caption")

# Kode
st.code("print('Hello')", language='python')
```

2 Menampilkan Data

```
# DataFrame
st.dataframe(df)                          # Interaktif
st.dataframe(df, height=400)              # Tinggi tetap
st.dataframe(df, use_container_width=True)  # Lebar penuh

# Tabel statis
st.table(df.head(10))

# Metrics
st.metric("Total Paket", "16,430")
st.metric("Total Pagu", "2.1 T", delta="+5%")
```

3 Layout: Kolom

```
# Membuat kolom
col1, col2, col3 = st.columns(3)

# Menggunakan kolom
with col1:
    st.metric("Metrik 1", "100")

with col2:
    st.metric("Metrik 2", "200")

with col3:
    st.metric("Metrik 3", "300")

# Atau inline
col1.metric("Metrik 1", "100")
col2.metric("Metrik 2", "200")
```

🎨 Lebar Kolom

```
# Lebar sama
col1, col2 = st.columns(2)

# Rasio kustom
col1, col2 = st.columns([2, 1])  # Rasio 2:1
col1, col2, col3 = st.columns([3, 1, 1])  # 3:1:1
```

4️⃣ Layout: Kontainer

```
# Container - mengelompokkan elemen
with st.container():
    st.write("Di dalam container")
    st.dataframe(df)

# Expander - bagian yang bisa dibuka/tutup
with st.expander("📖 Lihat Detail"):
    st.write("Awalnya tersembunyi")
    st.dataframe(df)
```

5 Sidebar

```
# Tambah ke sidebar
st.sidebar.title("Filter")
st.sidebar.selectbox("Pilih:", options)

# Atau dengan context
with st.sidebar:
    st.title("Filter")
    option = st.selectbox("Pilih:", options)
```

Kasus penggunaan: Filter, pengaturan, navigasi

6 Widgets: Input

```
# Input teks
name = st.text_input("Masukkan nama:", placeholder="Nama Anda")

# Input angka
age = st.number_input("Masukkan umur:", min_value=0, max_value=120)

# Text area
comment = st.text_area("Komentar:", height=100)

# Input tanggal
date = st.date_input("Pilih tanggal:")
```

7️⃣ Widgets: Seleksi

```
# Selectbox (dropdown)
option = st.selectbox(
    "Pilih metode:",
    ["Semua", "Tender", "E-Purchasing"]
)

# Multiselect
options = st.multiselect(
    "Pilih lebih dari satu:",
    ["Opsi 1", "Opsi 2", "Opsi 3"],
    default=["Opsi 1"]  # Pre-selected
)
```

8 Widgets: Slider & Range

```
# Slider
value = st.slider("Pilih nilai:", 0, 100, 50)

# Range slider
range_val = st.slider(
    "Pilih rentang:",
    min_value=0.0,
    max_value=100.0,
    value=(25.0, 75.0)  # Rentang default
)
```

9️⃣ Widgets: Boolean

```
# Checkbox
agree = st.checkbox("Saya setuju")

if agree:
    st.write("Terima kasih!")

# Toggle (sama dengan checkbox)
enabled = st.checkbox("Aktifkan fitur", value=True)

# Radio buttons
choice = st.radio(
    "Pilih satu:",
    ["Opsi A", "Opsi B", "Opsi C"]
)
```

🔟 Tombol

```
# Tombol sederhana
if st.button("Klik saya"):
    st.write("Tombol diklik!")

# Tombol dengan tipe
if st.button("Tombol utama", type="primary"):
    st.success("Tombol utama diklik!")

# Tombol nonaktif
if st.button("Nonaktif", disabled=True):
    pass  # Tidak akan dieksekusi
```

📥 Tombol Unduh

```
# Download CSV
csv = df.to_csv(index=False).encode('utf-8')

st.download_button(
    label="Unduh CSV",
    data=csv,
    file_name="data.csv",
    mime="text/csv"
)
```

💬 Pesan & Peringatan

```
# Sukses
st.success("✅ Operasi berhasil!")

# Info
st.info("ℹ️ Informasi tambahan")

# Warning
st.warning("⚠️ Pesan peringatan")

# Error
st.error("❌ Terjadi error!")
```

⏳ Progress & Status

```
# Progress bar
progress = st.progress(0)
for i in range(100):
    progress.progress(i + 1)

# Spinner (memuat)
with st.spinner("Memuat..."):
    time.sleep(2)
    st.success("Selesai!")

# Status
st.status("Memproses...", state="running")
```

🎯 Contoh Lengkap

```
import streamlit as st
import pandas as pd

st.title("📊 RUP Dashboard")

# Filter di sidebar
st.sidebar.header("Filter")
metode = st.sidebar.selectbox("Metode:", ["Semua", "Tender"])
pagu_range = st.sidebar.slider("Pagu (M):", 0, 100, (0, 100))

# Terapkan filter
filtered_df = df[
    (df['pagu'] >= pagu_range[0] * 1e6) &
    (df['pagu'] <= pagu_range[1] * 1e6)
]

# Metrics
col1, col2 = st.columns(2)
col1.metric("Total Paket", len(filtered_df))
col2.metric("Total Pagu", f"{filtered_df['pagu'].sum()/1e9:.2f} M")

# Data
st.dataframe(filtered_df)
```

🔄 App Rerun

Streamlit **menjalankan ulang seluruh script** ketika:

- Pengguna berinteraksi dengan widget
- Kode berubah (mode dev)
- Refresh manual

Penting: Gunakan `@st.cache_data` agar data tidak terus di-load!

```
@st.cache_data
def load_data():
    return pd.read_parquet('data.parquet')  # Hanya jalan sekali!
```

💾 Caching

```
# Cache data (immutable)
@st.cache_data
def load_data():
    return pd.read_parquet('data.parquet')

# Cache resources (koneksi, model)
@st.cache_resource
def init_connection():
    return duckdb.connect(':memory:')

# Clear cache
st.cache_data.clear()
```

🎨 Styling

```
# Custom CSS
st.markdown("""
    <style>
    .main-header {
        font-size: 3rem;
        color: #1f77b4;
    }
    </style>
    <h1 class="main-header">Judul Kustom</h1>
""", unsafe_allow_html=True)

# Teks berwarna
st.markdown(":blue[Teks biru]")
st.markdown(":red[Teks merah]")
```

🎯 Tips Praktis

✅ **Letakkan filter di sidebar** untuk layout bersih
✅ **Gunakan kolom** untuk metrics & perbandingan
✅ **Pakai expander** untuk detail opsional
✅ **Tambahkan progress bar** untuk operasi lama
✅ **Cache operasi mahal** dengan @st.cache_data
✅ **Gunakan label deskriptif** untuk widgets

🚀 Menjalankan App

```
# Jalankan app
streamlit run app.py

# Otomatis buka browser di http://localhost:8501

# Opsi
streamlit run app.py --server.port 8502
streamlit run app.py --server.headless true
```

Mode dev: Auto-refresh saat file disimpan!

🎓 Latihan Praktik

Bangun dashboard sederhana (60 menit):

1. ✅ Konfigurasi halaman & judul
2. ✅ Sidebar dengan filter (metode, jenis, rentang pagu)
3. ✅ 4 metrik (total paket, total pagu, rata-rata pagu, total satker)
4. ✅ Tabel data dengan formatting
5. ✅ Tombol unduh untuk CSV
6. ✅ Expander dengan statistik

File: `app_part1.py`

📚 Ringkasan Inti

✅ Streamlit = web app Python murni
✅ Tanpa HTML/CSS/JS
✅ Reaktif: auto-rerun saat interaksi
✅ Layout: columns, containers, expanders, sidebar
✅ Widgets: input, select, slider, checkbox, button
✅ Caching: @st.cache_data untuk performa
✅ Run: `streamlit run app.py`

🔗 Referensi

- **Streamlit Docs:** https://docs.streamlit.io/
- **API Reference:** https://docs.streamlit.io/library/api-reference
- **Gallery:** https://streamlit.io/gallery
- **Cheat Sheet:** https://docs.streamlit.io/library/cheatsheet
- **Components:** https://streamlit.io/components

☕ Waktunya Istirahat!

Selanjutnya: Sesi 4 - Dashboard Lengkap

Tambah chart, fitur lanjutan & deploy!