**Rangkuman Pengembangan Dashboard**

**Pengenalan Dashboard**

Saat ini, dashboard banyak digunakan sebagai media untuk memonitori berbagai metrik yang dianggap penting untuk perkembangan sebuah perusahaan. Selain itu, ia juga merupakan salah satu komponen penting dalam kultur *data-driven decision making*. Umumnya, sebuah dashboard terdiri dari dari beberapa visualisasi data yang interaktif dan saling berhubungan satu sama lain.

Dashboard yang efektif merupakan kunci utama dalam kultur data-driven decision making. Berikut merupakan beberapa hal yang harus diperhatikan dalam membuat dashboard yang efektif.

* **Memahami audiens**  
  Untuk membuat dashboard yang efektif, kita harus mempertimbangkan audiens dan kebutuhannya. Oleh karena itu, pada proses pembuatan dashboard kita harus memahami siapa target user atau audiens serta goals atau kebutuhan yang ingin mereka capai dari dashboard tersebut.
* **Pertimbangkan ukuran tampilan**  
  Sejalan dengan poin pertama, kita juga harus mempertimbangkan ukuran tampilan dashboard yang sesuai dengan *device* atau perangkat yang sering digunakan oleh target user atau audiens untuk melihat dashboard yang Anda buat.
* **Manfaatkan sweet spot**  
  Anda bisa menggunakan sweet spot untuk menampilkan informasi penting dari sebuah dashboard. Tentunya Anda perlu melakukan riset terlebih dahulu untuk mengetahui *sweet spot* yang tepat untuk kebutuhan Anda.
* **Pertimbangkan load time**  
  Load time merupakan salah hal yang penting untuk kita pertimbangkan guna menjaga pengalaman user atau audiens. Oleh karena itu, kita harus memastikan dashboard yang dibuat memiliki load time yang relatif singkat.
* **Batasi komponen visual yang ditampilkan**  
  Pada proses pengembangan dashboard kita juga harus mempertimbangkan jumlah komponen visual yang akan ditampilkan. Secara umum, kita disarankan untuk hanya menampilkan dua atau tiga visualisasi data serta menghindari penggunaan komponen visual yang tidak mendukung dalam penyampaian informasi penting.

**Pengenalan Streamlit**

Streamlit merupakan salah satu*open-source web app framework* untuk bahasa pemrograman Python. Ia memungkinkan kita membuat web app yang baik dan interaktif dalam waktu yang singkat. Selain itu, ia juga kompatibel dengan berbagai library populer seperti NumPy, pandas, matplotlib, dll. Inilah yang menjadi alasan streamlit sering digunakan oleh oleh para praktisi data dan machine learning.

Seperti biasa, sebelum mulai menggunakan streamlit tentunya kita perlu menyiapkan *environment*-nya terlebih dahulu. Berikut merupakan tahapan untuk melakukannya.

1. Untuk menggunakan streamlit, kita perlu menulis kode Python dalam berkas**.py**. Oleh karena itu, Anda perlu menginstal code editor, seperti [Visual Studio Code](https://code.visualstudio.com/download), [Sublime Text](https://www.sublimetext.com/3), atau [PyCharm](https://www.jetbrains.com/pycharm/download/#section=windows) untuk mempermudah dalam menulis kode.
2. Selanjutnya, Anda perlu mengaktifkan virtual environment sebelumnya yang telah kita gunakan dalam materi latihan. Aktifkan virtual environment dengan menjalankan perintah berikut.
   1. conda activate main-ds
3. Tahap berikutnya adalah menginstal streamlit. Berikut merupakan perintah yang bisa Anda gunakan untuk menginstal streamlit.
   1. pip install streamlit

**Basic Element dalam Streamlit**

Sebagai sebuah web app framework yang andal, streamlit telah menyediakan banyak pilihan element, widget, layout serta container untuk memastikan kita dapat membuat web app atau dashboard yang menarik dan interaktif.

**Write**

Basic element pertama yang akan kita bahas ialah write. Ia merupakan elemen streamlit yang digunakan untuk menampilkan sebuah output.  Untuk menggunakan element ini, kita hanya perlu memanggil function **write()** dan diikuti inputan yang ingin ditampilkan.

**Text**

Elemen lain yang ada dalam streamlit ialah text. Sesuai dengan namanya, ia merupakan elemen yang digunakan untuk menampilkan sebuah output berupa text. Elemen ini memiliki banyak function yang bisa digunakan sesuai kebutuhan. Berikut merupakan beberapa pilihan yang tersedia saat ini.

* **markdown()**  
  Function ini digunakan untuk menampilkan output dari argument markdown.
* **title()**  
  Ini merupakan function yang digunakan untuk menampilkan teks dalam format judul.
* **header()**  
  Function ini digunakan untuk menampilkan output teks sebagai format header.
* **subheader()**  
  Ini merupakan function yang digunakan untuk menampilkan output teks sebagai format subheader.
* **caption()**  
  Ia merupakan function yang digunakan untuk menampilkan output teks dalam ukuran kecil. Function ini biasanya digunakan untuk menampilkan caption, footnotes, dll.
* **code()**  
  Pada beberapa case, kita harus menampilkan potongan kode ke dalam streamlit app (web app yang dibuat menggunakan streamlit). Untuk menjawab hal ini, streamlit telah menyediakan sebuah function bernama **code()**.
* **text()**  
  Function selanjutnya ialah **text()**. Function ini digunakan untuk menampilkan sebuah normal teks.
* **latex()**  
  Function terakhir yang dapat digunakan untuk menampilkan elemen teks ialah latex(). Sesuai namanya, function tersebut digunakan untuk menampilkan *mathematical expression* yang ditulis dalam [format LaTeX](https://katex.org/docs/supported.html).

**Data Display**

Basic element selanjutnya yang akan kita bahas ialah data display. Ia merupakan elemen yang digunakan untuk menampilkan data secara cepat dan interaktif ke dalam streamlit app yang kita buat. Elemen ini memiliki beberapa function seperti berikut.

* **dataframe()**  
  Ia merupakan function yang digunakan untuk menampilkan DataFrame sebagai sebuah tabel interaktif. Pada function ini, kita bisa mengatur ukuran dari table yang ingin ditampilkan menggunakan parameter **width** dan **height**.
* **table()**  
  Selain **dataframe()**, kita juga bisa menggunakan function **table()** untuk menampilkan data ke dalam streamlit app. Ia dapat digunakan untuk menampilkan data dalam bentuk static table.
* **metric()**  
  Ketika membuat dashboard, terkadang kita perlu menampilkan sebuah metrik tertentu. Untuk melakukan hal ini, kita bisa memanfaatkan function **metric()**.
* **json()**  
  Selain bentuk DataFrame atau tabel, terkadang kita juga perlu menampilkan data dalam format JSON. Streamlit telah menyediakan function json() untuk menampilkan data dalam format JSON.

**Chart**

Basic element terakhir yang perlu kita ketahui ialah chart. Sesuai namanya, elemen ini dapat digunakan untuk menampilkan grafik visualisasi data ke dalam streamlit app. Elemen inilah yang akan sering kita gunakan untuk membuat dashboard nantinya.

Function **pyplot()** dapat digunakan untuk menampilkan grafik visualisasi data yang dibuat menggunakan matplotlib.

**Basic Widgets dalam Streamlit**

Untuk menghasilkan sebuah dashboard yang interaktif, kita memerlukan komponen lain seperti *widget*(elemen *Graphical User Interface* yang memungkinkan pengguna untuk berinteraksi dengan aplikasi).

**Input Widget**

Kategori widget yang akan kita bahas ialah input widget. Ia merupakan kategori widget yang memungkinkan pengguna untuk memberikan input ke dalam streamlit app. Terdapat beberapa widget yang termasuk kategori ini, seperti *text input*, *number input*, *date input*, dll. Yuk, kita bahas satu per satu!

* **Text input**  
  Text input merupakan widget yang digunakan untuk memperoleh inputan berupa *single-line text*. Kita bisa menggunakan function **text\_input()** untuk membuat widget ini.
* **Text-area**  
  Text area merupakan widget yang memungkinkan pengguna untuk menginput *multi-line text*. Untuk membuat widget ini, streamlit telah menyediakan function bernama **text\_area()**.
* **Number input**  
  Ia merupakan widget yang digunakan untuk memperoleh inputan berupa angka dari pengguna. Anda dapat menggunakan function **number\_input()** untuk membuat number input widget.
* **Date input**  
  Selain inputan berupa angka dan teks, pada beberapa kasus kita juga membutuhkan input berupa tanggal dari pengguna melalui date input widget. Kita dapat membuat widget tersebut menggunakan function **date\_input()**.
* **File uploader**  
  Widget ini memungkinkan kita meminta pengguna untuk meng-*upload* sebuah berkas tertentu ke dalam web app. Kita dapat membuat file uploader widget menggunakan function  **file\_uploader()**.
* **Camera input**  
  Selain file uploader, streamlit juga menyediakan camera input widget yang dapat digunakan untuk meminta user mengambil gambar melalui *webcam* sekaligus mengunggahnya menggunakan function**camera\_input()**.

**Button Widgets**

Oke, kategori widget selanjutnya yang akan kita bahas ialah button widget. Ia merupakan kategori widget yang terdiri dari button, checkbox, radio button, dll.

* **Button**  
  Button merupakan widget untuk menampilkan tombol interaktif. Tombol tersebut dapat digunakan pengguna untuk melakukan aksi tertentu. Untuk membuat widget ini, kita bisa menggunakan function **button()**.
* **Checkbox**  
  Checkbox merupakan widget yang digunakan untuk menampilkan sebuah *checklist* untuk pengguna. Kita bisa menggunakan function **checkbox()** untuk membuat dan menampilkan checklist dalam streamlit app.
* **Radio button**  
  Ia merupakan widget yang memungkinkan pengguna untuk memilih satu dari beberapa pilihan yang ada. Untuk membuat widget ini, kita bisa menggunakan function **radio()**.
* **Select Box**  
  Select box merupakan widget yang memungkinkan pengguna untuk memilih salah satu dari beberapa pilihan yang ada. Ia merupakan opsi alternatif dari radio button. Streamlit telah menyediakan function **selectbox()** untuk membuat select box widget.
* **Multiselect**  
  Ia merupakan widget yang digunakan agar user dapat memilih lebih dari satu pilihan dari sekumpulan pilihan yang ada. Untuk mempermudah kita dalam membuat multiselect widget, streamlit telah menyediakan function bernama **multiselect()**.
* **Slider**  
  Slider merupakan widget yang memungkinkan pengguna untuk untuk memilih nilai (atau range nilai) dari sebuah range nilai yang telah ditentukan. Streamlit telah menyediakan function **slider()** untuk membuat slider widget.

**Basic Layouts dalam Streamlit**

Untuk membuat dashboard yang rapi, kita perlu belajar cara mengatur layout atau tampilan dari sebuah dashboard. Sebagai sebuah web app framework, streamlit telah menyediakan berbagai pilihan layout yang dapat digunakan untuk mengatur tampilan web app (atau dashboard) yang akan dibuat.

**Sidebar**

Elemen layout pertama yang akan kita bahas ialah sidebar. Ia merupakan area yang berada di samping konten utama. Pada streamlit, posisi sidebar secara default berada di bagian kiri dari konten utama dan dapat memuat berbagai hal mulai dari gambar, teks, hingga widget.

Untuk menambahkan sebuah elemen atau widget ke dalam sidebar, kita bisa menggunakan notasi **with** yang diikuti sebuah object bernama **sidebar** yang telah disediakan oleh streamlit.

**Columns**

Columns merupakan elemen layout yang memungkinkan kita untuk mengatur tampilan pada konten utama ke dalam beberapa kolom sesuai kebutuhan.

Untuk membuat column dalam streamlit app, kita harus membuat object dari setiap kolom terlebih dahulu. Hal ini dapat dilakukan dengan memanfaatkan function **columns()**. Selanjutnya, kita hanya perlu menambahkan sebuah elemen atau widget ke dalam column tersebut menggunakan notasi **with**.

**Tabs**

Ia merupakan elemen layout yang memungkinkan kita untuk menambahkan beberapa tabs ke dalam konten utama. Hal ini tentunya akan sangat membantu kita dalam menghasilkan dashboard yang interaktif.

Sama halnya dengan columns yang sebelumnya kita bahas, untuk membuat tabs, kita harus membuat object dari setiap tab menggunakan function **tabs()** yang diikuti dengan list dari nama dari setiap tab.

**Container**

Container merupakan elemen layout dalam streamlit yang memungkinkan kita untuk membuat sebuah wadah untuk menampung suatu atau beberapa elemen dengan ukuran yang tetap. Container ini dapat kita gunakan untuk menghasilkan dashboard yang rapi.

Untuk membuat container, kita bisa menggunakan notasi **with** dan diikuti function **container()**.

**Expander**

Elemen layout selanjutnya yang tidak kalah penting ialah expander. Anda dapat menganggap elemen layout ini sebagai sebuah container yang dapat diperluas atau diciutkan secara interaktif.

Untuk membuat elemen layout ini, kita bisa menggunakan notasi **with** dan diikuti dengan function **expander()**.