# Penerapan Teknik Closure dalam Pembuatan Plot Dinamis Menggunakan Python

**Tessa Kania Sagala[1], Renisha Putri Giani[2], Rian Bintang Wijaya[3], Gymnastiar Al Khoarizmy[4], Virdio Samuel Saragih[5]**

Jurusan Sains Data, Fakultas Sains, Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Email: tessa.122450040@student.itera.ac.id, renisha.122450079@student.itera.ac.id, rian.122450094@student.itera.ac.id, gymnastiar.122450096@student.itera.ac.id, virdio.122450124@student.itera.ac.id

## 1. Pendahuluan

Plotting data merupakan proses memvisualisasikan informasi untuk menginterpretasi sebuah data, seperti dalam bentuk diagram, grafik, atau peta. Teknik plotting data merupakan salah satu teknik yang paling banyak digunakan dalam analisis data, tujuannya adalah untuk mengungkap pola, anomali dan hubungan antara variabel dalam data. Teknik teknik plotting data antara lain adalah grafik garis, diagram batang, diagram lingkaran, scatter plot, dan histogram. Dengan memahami teknik teknik tersebut, kita dapat mengubah kumpulan data menjadi narasi visual.

Closure adalah konsep dalam pemrograman fungsional di Python. Dengan closure memungkinkan untuk deklarasi fungsi yang berada di fungsi *(nested function)*. Closure sering digunakan untuk membuat fungsi yang beroperasi dengan data yang tidak berada dalam cangkupan global. Di dalam closure sendiri terdapat konsep dasar yang meliputi fungsi dalam fungsi, fungsi yang dikembalikan dan lingkup leksikal. Closure dapat berguna untuk penggunaan data yang lebih fleksibel, pembuatan fungsi yang dinamis dan pemeliharaan state. Dengan menggunakan closure pembuatan kode akan lebih dinamis dan fleksibel dalam python.

Dalam artikel ini, kami akan membahas penerapan plot dinamis dengan menggunakan teknik closure pada python yang dimulai dengan memahami konsep plotting data dan teknik closure setelah itu dilanjut dengan contoh implementasi dengan menggunakan matplotlib dan pembahasannya.

## 2. Metode

Dalam pengerjaan Penerapan Teknik Closure dalam Pembuatan Plot Dinamis Menggunakan Python, kami menggunakan beberapa fungsi yang esensial untuk mencapai tujuan kami:

### 2.1. Fungsi *matplotlib.pyplot*

Fungsi dari *matplotlib.pyplot* digunakan untuk mengakses fungsi plot dari library matplotlib. Fungsi ini memungkinkan kami untuk membuat plot dinamis dengan -berbagai jenis plot, seperti plot garis (*line*) dan plot titik (*scatter*), sesuai dengan

kebutuhan kami. Dengan fungsi ini, kami dapat memvisualisasikan data dengan mudah dan fleksibel.

2.2. Fungsi *pandas*

Pandas adalah *library* yang digunakan untuk manipulasi dan analisis data. Pandas menyediakan struktur data seperti Data Frame yang memungkinkan pengolahan dan penyajian data yang efisien. Fungsi pandas digunakan dalam laporan ini untuk mengolah data yang digunakan menjadi sebuah struktur data yang disebut *data frame*. Dengan menggunakan library pandas, kami dapat dengan mudah memanipulasi, membersihkan, dan menganalisis data secara efisien. Data frame ini kemudian dapat digunakan sebagai input untuk pembuatan plot dinamis.

2.3. Fungsi *sklearn*

Scikit-Learn adalah perpustakaan machine learning open-source yang dibangun di atas perpustakaan NumPy dan SciPy. Ini menyediakan alat yang kuat dan sederhana untuk membangun dan menerapkan berbagai algoritma machine learning seperti regresi, klasifikasi, klastering, pengoptimalan, dan banyak lagi. Fungsi dari sklearn (*Scikit-learn*) digunakan dalam laporan ini untuk mengakses dataset Iris yang tersedia dalam library tersebut. Dataset Iris ini merupakan dataset yang umum digunakan dalam pengujian algoritma pembelajaran mesin dan klasifikasi. Dengan menggunakan dataset ini, kami dapat mendemonstrasikan penerapan teknik closure dalam pembuatan plot dinamis dengan data dunia nyata.

2.4. Fungsi *dynamic_plotter()*

Fungsi *dynamic_plotter()* merupakan inti dari penerapan teknik closure dalam pembuatan plot dinamis. Konsep closure digunakan di sini dengan adanya variabel inner dan outer. Fungsi inner, yaitu fungsi *plot()*, akan menerima argumen tambahan berupa jenis plot (plot_type) dan label legenda (legend) untuk plot yang akan dibuat. Dengan menggunakan teknik closure, penulis dapat membuat fungsi plot yang dapat disesuaikan dengan berbagai macam jenis plot dan memberikan keterangan legenda sesuai dengan kebutuhan.

## 3. Pembahasan

Pada artikel ini, kami menggunakan dataset yang terdiri dari data sepal length, sepal width, petal length, dan petal width dengan satuan cm. Untuk memvisualisasikan data pada dataset tersebut, kami menggabungkan beberapa fungsi, termasuk plot line dan plot scatter, serta memanfaatkan dua library utama, yaitu Pandas dan Scikit-learn. Library Pandas digunakan untuk membuat DataFrame kosong dan mengisi kolom-kolom DataFrame dengan nilai-nilai yang diimpor dari library Scikit-learn.

3.1. Pembuatan Fungsi dynamic_plotter() Menggunakan Konsep Closure

```
import matplotlib.pyplot as plt
```

```
def dynamic_plotter():
    def plot(data, plot_type='line', legend=None):
        if plot_type == 'line':
            plt.plot(data, label=legend)
        elif plot_type == 'scatter':
            plt.scatter(range(len(data)), data, label=legend)
        plt.title('Plot Dinamis')
        plt.xlabel('X Axis')
        plt.ylabel('Y Axis')
        plt.grid(True)
        plt.legend(loc='upper right')
        plt.show()
    return plot

plot = dynamic_plotter() # Contoh pemakaian
```

Pada bagian ini, kami mengimplementasikan fungsi dynamic_plotter() yang menggunakan konsep closure. Pertama, kami mengimpor package matplotlib untuk menggunakan fungsi plot(). Selanjutnya, kami mendefinisikan fungsi dynamic_plotter(), yang menghasilkan fungsi plot() dengan bantuan closure. Fungsi plot() ini menerima argumen seperti data yang ingin diplot, jenis plot yang diinginkan (default: line), dan label legenda (opsional). Fungsi ini memungkinkan kita untuk membuat plot dinamis dengan mudah sesuai dengan kebutuhan, baik itu plot garis (line) maupun plot titik (scatter).

3.2. Mengimport Dataset Iris

```
import pandas as pd
from sklearn import datasets

iris = datasets.load_iris()
df = pd.DataFrame(data = iris['data'], columns = iris['feature_names'])
df['Iris type'] = iris['target']
df['Iris name'] = df['Iris type'].apply(lambda x: 'setosa' if x == 0 else ('versicolor' if x == 1 else 'virginica'))
```

Pada bagian ini, kami mengimpor library Pandas dan library Sklearn untuk memanfaatkan dataset Iris yang tersedia dalam library Sklearn. Kami menggunakan dataset ini untuk membuat sebuah DataFrame kosong dan mengisi kolom DataFrame dengan nilai-nilai dari dataset Iris. Data ini akan digunakan sebagai input untuk pembuatan plot dinamis.

3.3. Mendefinisikan Variabel Data Sekaligus Menjelaskan Judul Plot yang Akan Dibuat

```
sepalLength = df[df["Iris name"]=="setosa"]["sepal length (cm)"].tolist()
sepalWidth = df[df["Iris name"]=="setosa"]["sepal width (cm)"]
petalLength = df[df["Iris name"]=="setosa"]["petal length (cm)"]
petalWidth = df[df["Iris name"]=="setosa"]["petal width (cm)"]
```

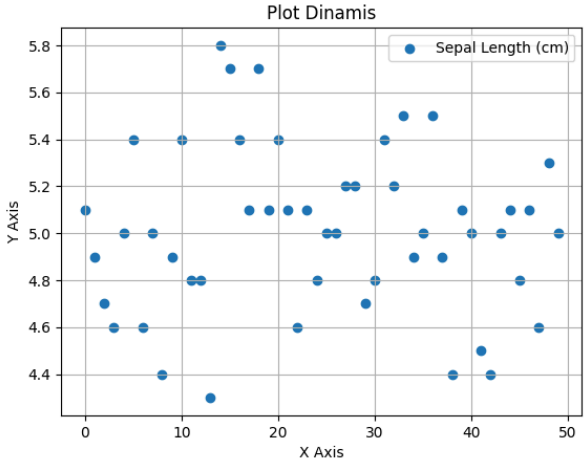
Dalam bagian ini, kami mendefinisikan variabel-variabel yang akan digunakan untuk menyimpan data sepal length, sepal width, petal length, dan petal width dari dataset Iris. Kami menggunakan library Pandas untuk memanipulasi data, dengan cara memilih subset data yang sesuai dengan jenis bunga iris tertentu (dalam contoh ini, jenis setosa) dan mengkonversinya menjadi list untuk kemudian digunakan dalam pembuatan plot dinamis.

3.4. Membuat plot dinamis untuk sepal length

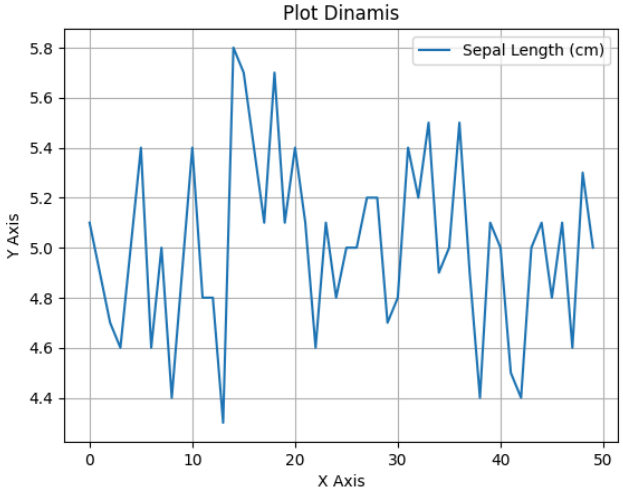
```
plot(sepalLength, plot_type='scatter', legend='Sepal Length (cm)')
plot(sepalLength, plot_type='line', legend='Sepal Length (cm)')

plot(sepalWidth, plot_type='scatter', legend='Sepal Width (cm)')
plot(sepalWidth, plot_type='line', legend='Sepal Width (cm)')

plot(petalLength, plot_type='scatter', legend='Petal Length (cm)')
plot(petalLength, plot_type='line', legend='Petal Length (cm)')

plot(petalWidth, plot_type='scatter', legend='Petal Width (cm)')
plot(petalWidth, plot_type='line', legend='Petal Width (cm)')
```

Berikut adalah kode yang digunakan untuk membuat plot dinamis untuk sepal length. Kami menggunakan fungsi dynamic_plotter() yang telah didefinisikan sebelumnya untuk membuat plot garis (line) dan plot titik (scatter) untuk data sepal length. Hasil plot ini akan memperlihatkan distribusi sepal length, petal length, dan petal width dari jenis bunga iris setosa secara dinamis.

Contoh hasil plot sebagai berikut:




Gambar 1. Scatter Plot Sepal Length




Gambar 2. Line Plot Sepal Length

## 4. Kesimpulan

Berdasarkan pembahasan yang telah dijelaskan, konsep teknik closure telah dimanfaatkan secara efektif dalam pembuatan plot dinamis menggunakan bahasa pemrograman Python. Fungsi-fungsi yang dikembangkan, seperti dynamic_plotter(), memungkinkan kami untuk membuat plot dengan berbagai jenis, seperti plot garis (line) dan plot titik (scatter), serta memberikan fleksibilitas dalam menambahkan label legenda sesuai

kebutuhan. Penggunaan teknik closure memainkan peran penting dalam hal ini dengan memungkinkan fungsi plot() untuk memiliki akses ke variabel-variabel luar, seperti data yang akan diplot dan jenis plot yang diinginkan, sehingga menciptakan fleksibilitas yang diperlukan dalam pembuatan plot dinamis.

Studi kasus pada distribusi fitur 'sepal length (cm)' dari dataset Iris juga menggambarkan kegunaan dan keampuhan dari pendekatan ini. Plot dinamis dari sepal length memperlihatkan variasi data secara visual, yang dapat memberikan wawasan yang berharga tentang distribusi data tersebut. Dengan menggunakan konsep closure, kami dapat dengan mudah menyesuaikan jenis plot dan menambahkan label legenda, sehingga meningkatkan kejelasan dan informativitas plot.

Penerapan teknik closure dalam pembuatan plot dinamis ini memperlihatkan potensi besar dari pendekatan pemrograman berbasis fungsi dalam analisis data. Melalui kode yang modular, efisien, dan mudah dipertahankan, kami berhasil menghasilkan solusi yang memungkinkan analisis yang lebih mendalam dan pemahaman yang lebih baik terhadap data. Dengan demikian, konsep teknik closure memberikan kontribusi yang signifikan dalam memungkinkan ekstraksi wawasan yang berharga dari data melalui pemrograman berbasis fungsi.

## 5. Referensi


Hermanto, K., Salim, D., Wu, B., Salim, O. R., & Gunadi, R. B. (2023, Juli). Penggunaan Python Untuk Menganalisis Pola Penyebaran Covid-19 Di Masa Pandemi. *Journal of Student Development Information System (JoSDIS)*, *3*, 64.

Saragih, R. R. (2018, Desember 24). *PEMROGRAMAN DAN BAHASA PEMROGAMAN*.

Syaferi, F. (2023, Oktober 18). *Pengenalan ke Machine Learning dengan Scikit-Learn*. Universitas Mahakarya Asia. https://blog.unmaha.ac.id/pengenalan-ke-machine-learning-dengan-scikit-learn