

FILTERISASI DATA MENGGUNAKAN FUNGSI CLOSURE PADA PEMROGRAMAN PYTHON

Deva Anjani Khayyuninafsyah (122450014), Nabiilah Putri Karnaia (122450029), Chevando Daffa Pramanda (122450095), Novelia Adinda (122450104), Rafly Prabu Darmawan (122450140)

Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Way Huwi, Kec. Jatiagung, Kabupaten Lampung Selatan, Lampung
35365

Email:

deva.122450014@student.itera.ac.id, nabiilah.122450029@student.itera.ac.id,
chevando.122450095@student.itera.ac.id, [novelia.122450104@student.itera.ac.id](mailto:novelias.122450104@student.itera.ac.id),
rafly.122450140@student.itera.ac.id

1. Pendahuluan

Penerapan konsep *closure* pada Python mengharuskan sebuah fungsi untuk mengakses serta memanipulasi variabel yang berada di luar cakupan (scope) dari fungsi tersebut. Penggunaan *closure* dalam konteks *filtering* kumpulan data dengan kondisi kriteria tertentu yang telah ditentukan, konsep *closure* mengharuskan pembuatan fungsi dengan penambahan *filtering* yang memiliki akses ke kumpulan data serta dapat mengembalikan subset dari kumpulan data tersebut berdasarkan kondisi yang ditentukan. Artinya, konsep *closure* mengharuskan untuk menyimpan serta mengatur algoritma *filter* ke dalam fungsi yang dapat diulang penggunaannya untuk memfilter (menyaring) kumpulan data berdasarkan kondisi yang ditentukan, misalnya kondisi pencarian yang sesuai pembahasan jurnal ini seperti nama menu pada restoran, harganya, dan kategorinya. Hal tersebut memberikan fleksibilitas dan modularitas pada sistem pencarian menu restoran, serta adanya penambahan atau mengubah kondisi pencarian tanpa mengubah algoritma *filter* yang ada. Dengan demikian, penerapan konsep *closure* pada sistem *filtering* kumpulan data untuk sebuah data menu restoran, menjadikannya solusi yang efisien serta kemudahan dalam konteks sistem pencarian menu restoran tersebut (Laksmi et al., 2020).

2. Metode

Pada kasus ini, digunakan sebuah fungsi *closure* yang memungkinkan kita untuk mengakses variabel bebas dari fungsi *outer* sesudah fungsi *outer* tersebut ditutup (*Proceedings of the Second EECE 512 Mini-Conference on Computer Security*, 2007, 10). Dalam

membuat *closure* yang menerapkan berbagai *filter* pada dataset berdasarkan kriteria tertentu ini menggunakan beberapa fungsi dan variabel, yaitu:

2.1. *def filter_menu(menu, kriteria)*

Fungsi tersebut membuat *closure* untuk memfilter data menu pada suatu restoran berdasarkan kriteria tertentu.

2.2. *def kriteria_menu()*

Fungsi ini adalah yang akan melakukan proses *filter* data menu berdasarkan kriteria.

2.3. *return [m for m in menu if kriteria(m)]*

Mengembalikan nilai dalam bentuk list berisi data yang memenuhi kriteria.

2.4. Variabel *menu*

Variabel menu ini adalah sebuah variabel berisikan data yang akan difilter.

2.5. Variabel *kriteria*

Fungsi yang menerima data dan mengembalikan *True* jika kriteria terpenuhi.

2.6. *Return*

Return digunakan untuk mengembalikan fungsi *filter* data.

3. Pembahasan

```
def filter_menu(menu, kriteria):
```

Gambar 1

Pada deret kode pertama, pendefinisianya menggunakan fungsi *filter_menu()* yang merupakan sebuah fungsi yang digunakan untuk menyaring berbagai item dari 'menu' yang memenuhi kriteria tertentu. Fungsi ini memiliki dua buah parameter, yaitu 'menu' yang merupakan deretan data berbentuk *list* yang nantinya akan disaring dan 'kriteria' yang merupakan sebuah fungsi menerima data dan mengembalikan nilai *True* jika kriterianya terpenuhi.

```
def kriteria_menu():
    """
    Fungsi ini yang akan memfilter data menu berdasarkan kriteria.

    Returns:
        Mengembalikan nilai dalam bentuk list berisi data yang memenuhi kriteria.
    """
    return [m for m in menu if kriteria(m)]

return kriteria_menu
```

Gambar 2

Pada deret kode yang kedua fungsi yang didefinisikan di dalam fungsi *filter_menu()*, yaitu sebuah fungsi bernama *kriteria_menu()*. Fungsi ini merupakan parameter yang nilai-nilainya diambil dari parameter dalam *filter_menu()* yang dalam kode ini membuat ‘menu’ dan ‘kriteria’ sebagai parameter fungsi inner yang eksplisit dengan nilai *default* yang sama dengan variabel lingkup luarnya. Kemudian akan *return* tiap elemen dalam ‘menu’ dan hanya elemen yang memenuhi ‘kriteria’ yang akan dimasukkan ke dalam list lalu dikembalikan oleh fungsi. Kemudian ditutup dengan baris kode selanjutnya yang mana fungsi *filter_menu()* akan mengembalikan fungsi *kriteria_menu()*.

```
# Dataset menu suatu restoran
menu = [{"Menu": "Nasi Goreng", "Harga": "Rp25.000", "Kategori": "Makanan Utama"},
        {"Menu": "Soto Ayam", "Harga": "Rp15.000", "Kategori": "Makanan Utama"},
        {"Menu": "Es Jeruk", "Harga": "Rp8.000", "Kategori": "Minuman"},
        {"Menu": "Es Teh", "Harga": "Rp5.000", "Kategori": "Minuman"},
        {"Menu": "Es Krim Vanila", "Harga": "Rp8.000", "Kategori": "Makanan Penutup"},
        {"Menu": "Puding Cokelat", "Harga": "Rp10.000", "Kategori": "Makanan Penutup"}]

# Filter data menu berdasarkan harga
filter_berdasarkan_harga = filter_menu(menu, lambda m: m["Harga"] == "Rp8.000")
# Filter data menu berdasarkan kategori
filter_berdasarkan_kategori = filter_menu(menu, lambda m: m["Kategori"] == "Makanan Utama")
```

Gambar 3

Selanjutnya membuat pendefinisian variabel ‘menu’ yang memiliki 3 jenis *keywords*, yaitu ‘Menu’, ‘Harga’, dan ‘Kategori’ yang menyimpan informasi terkait elemen pada menu tersebut. Lalu dilanjutkan dengan deretan kode yang menggunakan fungsi *filter_menu()* yang penyaringannya menggunakan fungsi lambda sebagai argumen kedua. Variabel *filter_berdasarkan_harga* akan memanggil fungsi *closure filter_menu()* dengan parameter menu sebagai daftar elemen menu dan fungsi lambda *lambda m: m["Harga"] == "Rp8.000"* sebagai kriteria penyaringannya. Fungsi lambda ini akan memeriksa apakah nilai dari *keywords* ‘Harga’ pada tiap menu ‘m’ sama dengan ‘Rp8.000’ yang kemudian disimpan dalam variabel *filter_berdasarkan_harga*. Dengan kata lain, fungsi *filter_menu()* disimpan dalam variabel *filter_berdasarkan_harga* akan mengambil data menu restoran yang memiliki harga Rp8.000.

Sama halnya seperti variabel *filter_berdasarkan_harga*, assignment *filter_berdasarkan_kategori = filter_menu(menu, lambda m: m["Kategori"] == "Makanan Utama")* menetapkan hasil fungsi *closure filter_menu()* pada variabel *filter_berdasarkan_kategori*. Fungsi *filter_menu()*, yang menerima parameter menu sebagai daftar elemen menu dan fungsi lambda *lambda m: m["Kategori"] == "Makanan Utama"* sebagai kriteria penyaringannya, akan dipanggil melalui variabel *filter_berdasarkan_kategori*. Fungsi lambda yang digunakan akan memeriksa apakah nilai dari *keywords* ‘Kategori’ pada tiap menu ‘m’ sama dengan ‘Makanan Utama’ yang kemudian disimpan dalam variabel

filter_berdasarkan_kategori(). Dengan demikian, *assignment* ini akan mengambil data menu restoran dengan kategori ‘Makanan Utama’.

```
# Hasil filter
hasil1 = filter_berdasarkan_harga()
hasil2 = filter_berdasarkan_kategori()
```

Gambar 4

Lalu *hasil1* dan *hasil2* merupakan variabel yang menyimpan hasil *filter*. *hasil1* merupakan variabel yang menyimpan *filter_berdasarkan_harga()*, sedangkan *hasil2* merupakan variabel yang menyimpan *filter_berdasarkan_kategori()*.

```
# Output
print(f"Menu makanan dengan harga Rp8.000: \n{hasil1}. \n")
print(f"Menu makanan utama: \n{hasil2}.")
```

Gambar 5

Dua kode terakhir merupakan *output* yang akan dicetak untuk hasil penyaringan data dengan kriteria tertentu yang digunakan pada fungsi *filter_menu()*. Pada deret kode *output* yang pertama menggunakan *fstring* untuk membuat *string format* untuk mencetak “Menu makanan dengan harga Rp8.000:” diikuti hasil penyaringan data yang sebelumnya telah disimpan pada variabel *hasil1* lalu *outputnya* akan menunjukkan bahwa terdapat dua buah menu dengan harga Rp8.000, yaitu Es Jeruk dan Es Krim Vanila. Sementara itu, deret kode *output* yang terakhir juga menggunakan *fstring* untuk membuat *string format* untuk mencetak “Menu makanan utama:” diikuti hasil penyaringan data yang sebelumnya telah disimpan pada variabel *hasil2* lalu *outputnya* akan menunjukkan bahwa terdapat dua buah menu dengan kategori ‘Makanan Utama’, yaitu Nasi Goreng dan Soto Ayam. Berikut *outputnya*.

```
Menu makanan dengan harga Rp8.000:
[{'Menu': 'Es Jeruk', 'Harga': 'Rp8.000', 'Kategori': 'Minuman'}, {'Menu': 'Es Krim Vanila', 'Harga': 'Rp8.000', 'Kategori': 'Makanan Penutup'}].

Menu makanan utama:
[{'Menu': 'Nasi Goreng', 'Harga': 'Rp25.000', 'Kategori': 'Makanan Utama'}, {'Menu': 'Soto Ayam', 'Harga': 'Rp15.000', 'Kategori': 'Makanan Utama'}].
```

Gambar 6

4. Kesimpulan

Kesimpulan yang dapat kami ambil adalah bahwa kode menggunakan fungsi *filter_menu()* untuk menyaring item dari menu berdasarkan kriteria tertentu. Fungsi ini menerima dua parameter, yaitu "menu", yang merupakan daftar item menu, dan "kriteria",

yang merupakan fungsi yang mengembalikan True jika kriterianya terpenuhi. Selain itu, kode menggunakan fungsi lambda sebagai argumen kedua untuk menyaring data berdasarkan harga dan kategori tertentu. Variabel "hasil1" dan "hasil2" menyimpan hasil penyaringan, yang kemudian dikeluarkan dalam format tertentu.

5. Daftar Pustaka

- Laksmi, N. C., Setyanto, A., & Wibowo, F. W. (2020, Juni 1). Kajian Pustaka Terstruktur Mengenai Analisis Audio Menggunakan Filterisasi Dengan Python. *Jurnal INFORMATIKA Politeknik Indonusa Surakarta*, 6.
- Proceedings of the Second EECE 512 Mini-Conference on Computer Security*. (2007). Laboratory for Education and Research in Secure Systems Engineering. [chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://ersse-dl.ece.ubc.ca/record/135/files/135.pdf](http://ersse-dl.ece.ubc.ca/record/135/files/135.pdf)