

# **Penerapan Fungsi Scope pada Python dengan Melacak Pergerakan: Analisis Moving Average dalam Pengambilan Keputusan**

Dearn Monica Manik, Nabila Anilda Zahrah, Abit Ahmad Oktarian,  
Allya Nurul Islami Pasha, Yohana Manik, David Bobby C. Nainggolan

Program Studi Sains Data Institut Teknologi Sumatera

Jl. Terusan Ryacudu, Way Huwi, Kec. Jatiagung, Kabupaten Lampung Selatan,  
Lampung 35365

Email:

[nabila.122450063@student.itera.ac.id](mailto:nabila.122450063@student.itera.ac.id) [dearni.122450075@student.itera.ac.id](mailto:dearni.122450075@student.itera.ac.id)  
[abit.122450042@student.itera.ac.id](mailto:abit.122450042@student.itera.ac.id) [allya.122450033@student.itera.ac.id](mailto:allya.122450033@student.itera.ac.id)  
[david.122450048@student.itera.ac.id](mailto:david.122450048@student.itera.ac.id) [yohana.122450126@student.itera.ac.id](mailto:yohana.122450126@student.itera.ac.id)

## **Pendahuluan**

Moving Average Calculator merupakan software atau perangkat lunak yang digunakan untuk menghitung rata-rata pergerakan dari sejumlah data dalam rentang waktu tertentu. Metode ini sering digunakan dalam analisis data, terutama dalam keuangan untuk mengidentifikasi tren jangka pendek dan jangka panjang dari serangkaian data.

Pada dasarnya, Moving Average Calculator mengambil serangkaian nilai data (seperti harga saham, suhu, penjualan bulanan, dll.) dan menghitung rata-rata dari nilai-nilai tersebut dalam rentang waktu tertentu. Rata-rata ini kemudian diplot sebagai garis pada grafik, yang membantu analisis dalam memahami tren pergerakan data tersebut.

Rata-rata gerakan harga dapat menghilangkan dampak dari penggulungan yang ketat dan dips yang tinggi, sehingga trend utama menjadi lebih jelas. Moving Average Calculator dapat menghitung rata-rata harga dalam jangka waktu tertentu (jangka waktu yang ditentukan) dan menghasilkan grafik yang menunjukkan rata-rata gerakan harga tersebut.

Moving Average Calculator dapat digunakan untuk menghitung berbagai jenis rata-rata gerakan harga, seperti rata-rata sederhana dan rata-rata eksponensial (EMA). Ini adalah alat yang sangat berguna untuk analisis harga dan menjadi alat yang diperlukan bagi para investor dan analis yang ingin mengikuti trend saham.

## Metode

Pada aplikasi kalkulator vektor kami menggunakan beberapa metode built-in dalam python, sebagai berikut:

1. Inisialisasi: Ketika objek Penghitung Moving Average dibuat, ukuran jendela moving average ditentukan. Data disimpan dalam sebuah list yang akan berperan sebagai jendela yang bergeser.
2. Penambahan Data: Metode `tambahkan_data()` digunakan untuk menambahkan data baru ke dalam list data. Ketika data baru ditambahkan, program memeriksa apakah jumlah data dalam list sudah melebihi ukuran jendela yang ditentukan. Jika ya, maka data pertama (yang terdapat di awal list) akan dihapus untuk menjaga agar ukuran jendela tetap.
3. Perhitungan Moving Average: Metode `hitung_moving_average()` menghitung rata-rata dari data dalam jendela saat ini. Jika jumlah data dalam jendela belum mencapai ukuran jendela yang ditentukan, maka fungsi ini akan mengembalikan `None` untuk menandakan bahwa belum cukup data untuk menghitung moving average.
4. Input dan Output: Program utama `main()` bertanggung jawab untuk menerima input dari pengguna (data baru) dan menampilkan moving average saat ini. Pengguna diminta untuk memasukkan ukuran jendela saat memulai program. Selama program berjalan, pengguna diminta untuk memasukkan data baru secara berulang sampai pengguna memasukkan 'q' untuk keluar.
5. Error Handling: Program memeriksa apakah input pengguna adalah nilai numerik atau 'q' untuk keluar. Jika input tidak valid, program akan memberikan pesan kesalahan dan meminta input kembali.

## Pembahasan

Berikut pembahasan kami mengenai program Moving Average Calculator:

### Konsep Moving Average:

Moving average (rata-rata bergerak) adalah teknik statistik yang digunakan untuk menganalisis data deret waktu dengan merata-ratakan nilai-nilai dalam suatu rentang waktu tertentu. Hal ini berguna untuk menghilangkan fluktuasi acak dan menyoroti tren atau pola yang lebih umum dalam data. Dalam kode di atas, moving average dihitung dengan menggunakan jendela geser, di mana rata-rata diambil dari sejumlah data terbaru dalam deret waktu.

### Implementasi Class PenghitungMovingAverage:

Class PenghitungMovingAverage digunakan untuk menyimpan data dan menghitung moving average. Ini memiliki atribut ukuran\_jendela untuk menentukan lebar jendela moving average dan list data untuk menyimpan data deret waktu.

#### **Metode tambahkan\_data():**

Metode ini digunakan untuk menambahkan data baru ke dalam jendela moving average. Jika jumlah data dalam list data melebihi ukuran\_jendela, maka data pertama akan dihapus untuk menjaga ukuran jendela tetap konstan.

#### **Metode hitung\_moving\_average():**

Metode ini menghitung moving average dari data dalam jendela saat ini. Jika jumlah data dalam jendela belum mencapai ukuran\_jendela, maka fungsi ini akan mengembalikan None, menandakan bahwa belum cukup data untuk menghitung moving average.

#### **Logika Utama Program:**

Program utama main() bertanggung jawab untuk menerima input dari pengguna, memprosesnya dengan menggunakan class PenghitungMovingAverage, dan menampilkan hasil moving average kepada pengguna. Selama berjalan, program akan terus meminta input data baru dan menampilkan moving average sesuai dengan data yang telah dimasukkan.

#### **Error Handling:**

Program memeriksa apakah input pengguna merupakan nilai numerik atau 'q' untuk keluar. Jika input tidak valid, program memberikan pesan kesalahan yang sesuai dan meminta input kembali.

#### **Pengimplementasian program:**

Pada program ini, pengguna diminta untuk memasukkan ukuran jendela moving average dan kemudian memasukkan data secara berulang hingga memilih untuk keluar dengan memasukkan 'q'. Setiap kali data dimasukkan, program akan menghitung dan menampilkan moving average terbaru dari data yang dimasukkan.

#### **Kelas PenghitungMovingAverage:**

```
class PenghitungMovingAverage:
    def __init__(self, ukuran_jendela):
        self.ukuran_jendela = ukuran_jendela
        self.data = []

    def tambahkan_data(self, data_baru):
        self.data.append(data_baru)
        if len(self.data) > self.ukuran_jendela:
            self.data.pop(0)

    def hitung_moving_average(self):
        if len(self.data) < self.ukuran_jendela:
            return None # Belum cukup data
        return sum(self.data) / self.ukuran_jendela
```

Dalam contoh program Python yang ditunjukkan sebelumnya, penggunaan metode Scope saat membuat kelas MovingAverage Counter sangat jelas.

Metode cakupan memungkinkan Anda mengatur data dan operasi terkait dalam kelas-kelas ini, memungkinkan Anda mengakses dan mengelola data secara terstruktur dan terorganisir.

Bagian program yang menggunakan metode cakupan:

- ❖ Metode `_init_`: Ini konstruktor untuk kelas MovingAverageCounter.
- ❖ Di sini kita menginisialisasi atribut `window_size` dan data dari setiap objek yang dibuat dari kelas ini.
- ❖ Metode `Add_data`: Metode ini bertanggung jawab untuk menambahkan data baru ke daftar data.
- ❖ Jika jumlah data melebihi ukuran jendela yang ditentukan, metode ini membuang data terlama untuk menjaga ukuran jendela tetap konstan.
- ❖ Metode `Hitung_rata-rata_bergerak`: Metode ini menghitung rata-rata pergerakan dari data yang disimpan dalam daftar data.
- ❖ Metode ini mengembalikan `Tidak Ada` jika data tidak cukup untuk mengisi ukuran jendela.

### Penggunaan kelas dalam fungsi `main()`:

```
def main():
    ukuran_jendela = int(input("Masukkan ukuran jendela untuk perhitungan moving average: "))
    penghitung_moving_avg = PenghitungMovingAverage(ukuran_jendela)

    while True:
        data_baru = input("Masukkan data baru (atau 'q' untuk keluar): ")
        if data_baru.lower() == 'q':
            break
        try:
            data_baru = float(data_baru)
            penghitung_moving_avg.tambahkan_data(data_baru)
            moving_avg = penghitung_moving_avg.hitung_moving_average()
            if moving_avg is not None:
                print(f"Moving Average: {moving_avg}")
            else:
                print("Belum cukup data.")
        except ValueError:
            print("Input tidak valid! Harap masukkan nilai numerik atau 'q' untuk keluar.")
```

Dalam fungsi `main()`, kita membuat objek kelas MovingAverageCounter dan menggunakan metodenya untuk menambahkan data baru dan menghitung rata-rata pergerakan. Ini menunjukkan bagaimana metode kelas ini digunakan dalam fungsi `main()` untuk mengelola penghitungan rata-rata bergerak.

### Kesimpulan

- implementasi sederhana dari perhitungan moving average dengan menggunakan jendela geser. Dalam program yang dibuat kita memiliki sebuah class PenghitungMovingAverage yang memungkinkan pengguna untuk menentukan ukuran jendela moving average dan

menambahkan data secara interaktif. Program kemudian menghitung dan menampilkan moving average dari data yang dimasukkan.

- Dengan menggunakan konsep moving average, program ini dapat digunakan untuk menganalisis data deret waktu dan mengidentifikasi tren atau pola yang mendasarinya. Pembahasan kode memberikan pemahaman yang jelas tentang cara kerja program, termasuk konsep moving average, implementasi class, logika utama program, dan penanganan kesalahan.