

Normalisasi Data Menggunakan Fungsi *Closure* dengan Metode *Min-Max* pada Dataset Mahasiswa

Oktavia Nurwinda Puspitasari¹, Mutiara Dian Pitaloka², Syadza Puspadari Azhar³, Kayla Amanda Sukma⁴, Muhammad Wafi Raihan⁵

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

Email: oktavia.122450041@student.itera.ac.id mutiara.122450047@student.itera.ac.id

syadza.122450072@student.itera.ac.id kayla.122450086@student.itera.ac.id

muhammad.122450144@student.itera.ac.id

1. Pendahuluan

Pada era digital saat ini pengolahan data menjadi sangat penting didalam berbagai bidang ,seperti ilmu pengetahuan, teknologi, bisnis ,dan lain-lain. Salah satu langkah penting dalam pengelolaan data adalah normalisasi, yaitu proses untuk mengubah nilai-nilai dalam dataset agar memiliki skala atau rentang nilai yang sama.

Normalisasi data dilakukan agar data dapat digunakan terlebih dahulu dan diperoleh data yang lebih akurat. Normalisasi data dilakukan agar data mahasiswa dapat dikelompokkan dengan tepat. Metode normalisasi data merupakan proses membuat beberapa variabel dengan rentang nilai yang sama, tidak ada yang terlalu besar maupun terlalu kecil sehingga analisis statistik dapat menjadi lebih mudah. Beberapa metode yang digunakan dalam melakukan normalisasi data yaitu *Min-Max normalization*, *Z-score Normalization*, dan *Decimal Scaling Normalization* [1]. Salah satu metode normalisasi yang dapat mengubah data dalam rentang nilai 0 sampai dengan 1 yaitu metode *Min-Max Normalization*. Metode ini digunakan untuk mengoptimalkan kinerja model yang digunakan [2].

2. Tujuan

- Menerapkan fungsi *Closure* dalam Python.
- Melakukan normalisasi data dengan metode *Min-Max* sebelum dilakukan uji lanjut.

3. Metode

3.1. Fungsi *Closure*

Sebuah fungsi yang dapat disimpan dalam variabel disebut dengan *Closure*. Dengan menggunakan fungsi *Closure*, kita dapat membuat fungsi di dalam fungsi (*nested function*) atau bahkan membuat fungsi yang akan mengembalikan fungsi. Fungsi anonim atau tanpa nama disebut juga sebagai *Closure*. Biasanya digunakan untuk melindungi proses yang hanya digunakan sekali atau digunakan pada blok tertentu saja [3].

3.2. Normalisasi Data

Salah satu metode persiapan data yang paling umum digunakan adalah normalisasi. Proses ini menggunakan skala umum untuk mengubah nilai kolom numerik dalam dataset dalam proses *machine learning* dan *data mining*. Dalam normalisasi data kita

mengubah data menjadi format standar atau "normal" untuk memudahkan pengolahan serta analisisnya [4].

3.3. *Min-Max Normalization*

Metode *Min-Max Normalization* adalah metode normalisasi data dengan cara melakukan transformasi linier terhadap data asli, agar nilai dari perbandingan antar data saat sebelum dan sesudah proses menjadi seimbang. Data diubah menjadi seimbang dalam rentang 0 sampai dengan 1, metode *Min-Max Normalization* merupakan salah satu metode untuk mengubah data yang kompleks dengan tidak menghilangkan isi, agar data lebih mudah diolah. Persamaan *Min-Max Normalization* yaitu [5]:

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Dengan:

X^* = Data hasil normalisasi

X = Nilai sebelum dinormalisasi

X_{max} = Nilai maksimum dari suatu parameter

X_{min} = Nilai minimum dari parameter

4. Pembahasan

4.1. Kode Fungsi Normalisasi Data dengan Metode *Min-Max*

```
import pandas as pd

def normalisasi_kolom(df, kolom):
    # Fungsi Closure untuk normalisasi
    def normalisasi(x):
        return (x - min_val) / (max_val - min_val)

    # Mendapatkan nilai minimum dan maksimum kolom
    min_val = df[kolom].min()
    max_val = df[kolom].max()

    # Menggunakan fungsi Closure untuk menerapkan normalisasi pada
    # kolom DataFrame
    df[kolom] = df[kolom].apply(normalisasi)
```

Pada fungsi ini, kami menggunakan *library* Pandas untuk membuat DataFrame. Kemudian mendefinisikan rumus fungsi *Min-Max* pada fungsi normalisasi dengan variabel x . min_val merupakan nilai minimum dari dataset, sedangkan max_val merupakan nilai maksimum dari dataset. Setelah itu, kami menggunakan fungsi *Closure* untuk menerapkan normalisasi pada DataFrame.

4.2. Data Sampel

No	Nama	Pekerjaan Orang Tua	Status Kepemilikan Rumah	Penghasilan Orang Tua
1	Atira Nabila	Nelayan	Sewa	2.500.000
2	Sawitri	Wiraswasta	Sewa	1.800.000
3	Meliana Sabet Tambunan	Petani	Milik Sendiri	2.000.000
4	Desi Novtia Siregar	Petani	Milik Sendiri	1.800.000
5	Handayani Simanjuntak	Wiraswasta	Milik Sendiri	2.500.000
6	Mardiana Pasaribu	Petani	Sewa	1.800.000
7	Yamolala Hulu	PNS	Milik Sendiri	5.000.000
8	David Ridwan Hakim	Wiraswasta	Sewa	3.000.000
9	Misael Oktavianda Harefa	Petani	Milik Sendiri	1.400.000
10	Muhammad Ali	Petani	Milik Sendiri	2.000.000

Tabel 1. Data Sampel

4.3. Data Sampel Hasil Transformasi

No	Nama	Pekerjaan Orang Tua	Status Kepemilikan Rumah	Penghasilan Orang Tua
1	Atira Nabila	1	1	2.500.000
2	Sawitri	2	1	1.800.000
3	Meliana Sabet Tambunan	1	2	2.000.000
4	Desi Novtia Siregar	1	2	1.800.000
5	Handayani Simanjuntak	2	2	2.500.000
6	Mardiana Pasaribu	1	1	1.800.000
7	Yamolala Hulu	3	2	5.000.000
8	David Ridwan Hakim	2	1	3.000.000
9	Misael Oktavianda Harefa	1	2	1.400.000

10	Muhammad Ali	1	2	2.000.000
----	--------------	---	---	-----------

Tabel 2. Data Sampel Hasil Transformasi

Sebelum melakukan normalisasi, dataset harus ditransformasi terlebih dahulu agar data bersifat numerik dengan keterangan:

- 1) Pekerjaan Orang Tua
 - a. Variabel Nelayan dan Petani = 1
 - b. Variabel Wiraswasta = 2
 - c. Variabel PNS = 3
- 2) Status Kepemilikan Rumah
 - a. Variabel Sewa = 1
 - b. Variabel Milik Sendiri = 2

4.4. Penerapannya Pada Data Sampel

```
# Membuat DataFrame
data = {
    "Nama" : ["Atira Nabila", "Sawitri", "Meliana Sabet Tambunan",
              "Desi Novria Siregar", "Handayani Simanjuntak",
              "Mardiana Pasaribu", "Yamolala Hulu", "David Ridwan
              Hakim", "Misael Oktavianda Harefa", "Muhammad Ali"],
    "Pekerjaan Orang Tua": [1, 2, 1, 1, 2, 1, 3, 2, 1, 1],
    "Status Kepemilikan Rumah": [1, 1, 2, 2, 2, 1, 2, 1, 2, 2],
    "Penghasilan Orang Tua": [2500000, 1800000, 2000000, 1800000,
                              2500000, 1800000, 5000000, 3000000,
                              1400000, 2000000]
}
df = pd.DataFrame(data)

# Normalisasi kolom "Pekerjaan Orang Tua"
normalisasi_kolom(df, "Pekerjaan Orang Tua")

# Normalisasi kolom "Status Kepemilikan Rumah"
normalisasi_kolom(df, "Status Kepemilikan Rumah")

# Normalisasi kolom "Penghasilan Orang Tua"
normalisasi_kolom(df, "Penghasilan Orang Tua")

print(df)
```

Pada tahap ini, kami membuat DataFrame berdasarkan data sampel yang telah ditransformasi dan menggabungkannya dalam variabel “data”. Setelah DataFrame dibuat, lakukan normalisasi berdasarkan fungsi yang telah dibuat sebelumnya.

4.5.Data Sampel Hasil Normalisasi Data

No	Nama	Pekerjaan Orang Tua	Status Kepemilikan Rumah	Penghasilan Orang Tua
1	Atira Nabila	0	0	0,3056
2	Sawitri	0,5	0	0,1111
3	Meliana Sabet Tambunan	0	1	0,1667
4	Desi Novtia Siregar	0	1	0,1111
5	Handayani Simanjuntak	0,5	1	0,3056
6	Mardiana Pasaribu	0	0	0,1111
7	Yamolala Hulu	1	1	1
8	David Ridwan Hakim	0,5	0	0,4444
9	Misael Oktavianda Harefa	0	1	0
10	Muhammad Ali	0	1	0,1667

Tabel 3. Data Hasil Normalisasi

Tabel di atas menunjukkan hasil dari data sampel yang telah dilakukan normalisasi data berdasarkan fungsi yang telah kami buat pada Python. Setelah data sampel dinormalisasi, maka data sudah bisa digunakan dalam proses selanjutnya atau uji lanjut.

5. Kesimpulan

Pengelolaan data menjadi semakin penting dalam berbagai industri, seperti bisnis, ilmu pengetahuan, dan teknologi di era digital yang semakin berkembang. Salah satu langkah penting dalam pengelolaan data adalah normalisasi data. Proses ini memungkinkan perubahan nilai kumpulan data agar memiliki skala atau rentang nilai yang sama, sehingga data dapat digunakan dengan lebih akurat. Normalisasi data dimaksudkan untuk mempermudah analisis statistik dan pemodelan data. Ini memungkinkan data dikelompokkan dengan lebih tepat dan efektif, yang memastikan bahwa setiap variabel memiliki pengaruh yang seimbang pada analisis. *Min-Max Normalization*, *Z-score Normalization*, dan *Decimal Scaling Normalization* adalah beberapa metode normalisasi yang umum digunakan. Dalam penelitian kali ini, kami menggunakan metode *Min-Max Normalization* menggunakan fungsi *Closure* dalam Python

untuk melakukan normalisasi pada beberapa kolom dalam sebuah DataFrame. Hasil dari data sampel yang telah dinormalisasi adalah data dengan rentang nilai antara 0 dan 1, dan siap untuk digunakan dalam proses analisis atau uji lanjut. Dengan demikian, normalisasi data adalah langkah penting dalam persiapan data untuk analisis lebih lanjut dan pemahaman tentang berbagai metode.

6. Daftar Pustaka

- [1] D. A. Nasution, H. H. Khotimah and N. Chamidah, "PERBANDINGAN NORMALISASI DATA UNTUK KLASIFIKASI WINE MENGGUNAKAN ALGORITMA K-NN," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 4, no. 1, pp. 78-82, 2019.
- [2] I. Permana and F. N. Salisah, "The Effect of Data Normalization on the Performance of the Classification Results of the Backpropagation Algorithm," *Ijrse*, vol. 2, no. 1, pp. 67-72, 2022.
- [3] N. Agung, "Golang Fungsi *Closure*," 2009. [Online]. Available: <https://dasarpemrogramangolang.novalagung.com/A-fungsi-Closure.html>. [Accessed 11 April 2024].
- [4] Z. Efendy, "e-Journal UIN Suska," *NORMALISASI DALAM DESAIN DATABASE*, vol. 4, no. 1, p. 10, Juni 2008.
- [5] V. R. Patel and R. G. Mehta, *Data Warehouse dan Data Mining Modul Standar untuk Digunakan Dalam Perkuliahan di Universitas Mercu Buana*, 2007.