Perbandingan Metode Imputasi pada Data Hilang: Analisis Penggunaan Mean, Median, dan Modus dengan Pendekatan Pemrograman Berbasis Fungsi

Elok Fiola[1] ,Vira Putri Maharani[2] , Hermawan Manurung[3], Diana Syafithri[4], Jihan Putri Yani[5], Rani Puspita Sari[6]

Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera,
Jalan Terusan Ryacudu, Way Huwi, Kec. Jati Agung, Kab. Lampung Selatan, Lampung

Pendahuluan

Dalam analisis data, keberadaan nilai yang kosong atau hilang seringkali menjadi tantangan yang harus diatasi karena dapat menjadi faktor yang mempengaruhi hasil analisis (Little & Rubin, 2019). Oleh karena itu, penting untuk menangani nilai-nilai yang hilang dengan baik. Untuk mengatasi hal tersebut, berbagai metode imputasi telah dikembangkan untuk mengisi nilai-nilai yang hilang tersebut. Metode konvensioanl menjadi salah satu teknik umum yang digunakan meliputi penggunaan nilai rerata (mean), median, atau modus dari data yang tersedia. Namun, sebelum menerapkan teknik-teknik ini proses preprocessing menjadi langkah awal yang mencakup identifikasi dan pengukuran lokasi *missing values* dalam data. Setelah itu, data yang telah dipersiapkan akan diuji menggunakan metode imputasi untuk mengevaluasi akurasinya.

Pentingnya memahami dan menggunakan metode imputasi telah meningkat seiring dengan kompleksitas data yang semakin tinggi. Artikel ini bertujuan untuk mengevaluasi dan membandingkan kinerja metode imputasi yang berbeda, dengan fokus pada penggunaan mean, median, dan modus, dalam menangani nilai-nilai yang hilang dalam data. Diharapkan dengan memahami perbandingan kinerja metode imputasi yang berbeda, dapat lebih baik dalam menggunakan teknik-teknik imputasi untuk meningkatkan keakuratan dan keandalan hasil analisis data.

Metode

Metode yang digunakan dalam analisis ini mengadaptasi pendekatan Pemrograman Berbasis Fungsi (PBF) yang menggabungkan pendekatan kualitatif melalui studi pustaka dan pendekatan kuantitatif melalui metode imputasi untuk menangani data hilang (*Missing data*). Digunakan metode konvensional, seperti mean, median, dan modus yang kemudian diukur menggunakan *Root Mean Square Error* (*RMSE)*.

Imputasi

Imputasi adalah suatu teknik yang digunakan untuk mengisikan nilai yang kosong atau hilang dalam data. Dalam menangani nilai-nilai yang hilang, terdapat beberapa metode yang dapat diterapkan seperti menghapus kolom yang mengandung data yang hilang, mengisi data yang hilang dengan nilai rerata, median, atau modus dari data tersebut (Somasundaram & Nedunchezhian, 2011).

Mean

Mean adalah nilai tunggal yang mewakili pusat distribusi data. Ini adalah hasil pembagian total nilai dengan jumlah pengamatan. Dalam distribusi normal, mean adalah nilai tengah distribusi frekuensi. Matematisnya, mean dari satu data tunggal adalah jumlah semua data dibagi dengan jumlah kejadian (Husnul, Prasetya, Sadewa, Ajimat, & Purnomo, 2020).

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad (1)$$

dengan x_i merupakan setiap nilai dalam dataset dan n merupakan jumlah total observasi dalam dataset.

Median

Median merupakan nilai tengah dari data yang sudah diurutkan dari nilai terkecil ke terbesar ataupun sebaliknya. Menurut (Boediono, 2015), median adalah nilai yang kecil keterletak di tengah untuk data yang berjumlah ganjil, atau merupakan rata-rata dari dua nilai yang besar.tengah jika jumlah data adalah genap.

Rumus meghitung median dari data ganjil:

$$Med = x_{\frac{n+1}{2}} \qquad (2)$$

Rumus meghitung median dari data genap:

$$Med = \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} \qquad (3)$$

dengan x_n merupakan elemen data ke-*n* dataset.

Modus

Modus adalah nilai dalam data yang memiliki kemunculan paling banyak atau frekuensi tertinggi. Keberadaan modus dalam data tidak selalu terjamin. Jika data memiliki modus, kemungkinan ada lebih dari satu nilai modus atau bahkan tidak ada nilai tunggal yang menjadi modus (Marliana, 2019).

Scope

Variabel *Scope* merupakan variabel merujuk pada tempat dimana variabel tersebut dideklarasikan dan dimana variabel tersebut berlaku. *Scope* variabel dapat dibagi menjadi dua jenis, yaitu *Global* dan *Local* (Satria, 2024)

Closure

Python *closure* memungkinkan fungsi mengakses variabel di lingkungan luar. *Closure* digunakan ketika fungsi-fungsi saling berhubungan dan mereferensikan nilai-nilai dalam lingkup yang terkait. Ini cocok saat kompleksitas tidak terlalu tinggi, namun disarankan untuk beralih ke penggunaan *class* ketika kompleksitasnya meningkat (Satria, 2024).

Root Mean Square Error (RMSE)

Root Mean Square Error (*RMSE*) adalah ukuran kesalahan prediksi, di mana semakin kecil nilai *RMSE* (mendekati 0), maka prediksi akan lebih akurat. *RMSE* dihitung dengan persamaan berikut.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y_i})^2} \qquad (4)$$

dengan *n* merupakan jumlah total observasi, $dan\ \ (y_i - \hat{y}_i)^2$ merupakan selisih kuadrat antara nilai actual dan nilai prediksi untuk setiap observasi. (Suprayogi, Trimaijon, & Mahyudin, 2017).

Pembahasan

Proses *preprocessing*

Proses *preprocessing* merupakan langkah pertama dalam melakukan imputasi terhadap *missing values*. Pada tahap ini, dilakukan identifikasi terhadap data yang tidak lengkap. Kemudian dilakukan pengukuran untuk menentukan lokasi *missing values*, dan kemudian dilakukan imputasi menggunakan metode konvensional.

Tabel 1. Dataset Daftar Nilai Akhir UTS PBF RC

Daftar Nilai				
27	36	18	9	44
39	59	57	22	38
73	45	32	61	52
29	25	34	49	42
NaN	47	30	27	75
47	83	51	81	34
43	80	40	41	42
43	53	55	47	53
37	31	59	35	46
8	48	38	37	37

Dataset yang digunakan merupakan DNA UTS PBF RC 2024 yang berjumlah 50 baris. Terdapat *missing data* pada data ke-21.

Metode Konvensional

Pada langkah ini, data yang telah melalui proses *preprocessing* akan langsung diuji untuk imputasinya menggunakan nilai rata-rata (mean), median, dan modus dari seluruh data yang tersedia, kemudian diukur menggunakan *RMSE*. Berikut adalah hasilnya. Berikut ini merupakan nilai statistika deskriiptif dari dataset.

Tabel 2. Statistika deskriptif data

Statistika Deskriptif		
Mean	**Median**	**Modus**
44	42	47

Nilai-nilai tersebut akan digunakan untuk mengisi *missing data* pada dataset.

Tabel 3. Hasil data imputer menggunakan mean, median, dan modus

Hasil Data Imputer		
Mean	*Median*	*Modus*
27	27	27
36	36	36
18	18	18
9	9	9
44	44	44
39	39	39
:	:	:
42	42	42
44	**42**	**47**
47	47	47
30	30	30
:	:	:
37	37	37

Data pada Tabel 3, menunjukkan bahwa penggunaan nilai mean, median, dan modus menghasilkan hasil yang serupa, dengan sedikit variasi di antara metode tersebut. Terdapat kesesuaian nilai yang besar antara ketiga metode, namun terdapat beberapa perbedaan kecil dalam nilai imputer antara metode median dan modus.

Analisa Perbandingan Metode

Tabel 3. Hasil *RMSE* masing-masing metode

Metode	RMSE
Mean	0.00
Median	0.28284
Modus	0.42426

Berdasarkan tabel, diketahui bahwa penggunaan mean untuk mengimputasi data menghasilkan *RMSE* yang paling rendah dibandingkan metode lainnya, yaitu sebesar 0.00. Nilai tersebut menunjukkan interpretasi yang sangat akurat terhadap kelayakan metode.

Kode Pemrograman

Pemrograman ini dibuat menggunakan variabel *scope* dan *closure* dalam bahasa Python. Kode Python Data Imputer lengkap dapat di akses melalui GitHub.

```
import csv
import math
from collections import Counter
from sklearn.metrics import mean_squared_error
```

Gambar 1 . Import modul Python

Kode tersebut mengimpor beberapa modul Python penting: csv untuk mengolah file CSV, math untuk operasi matematika, *Counter* dari *collections* untuk menghitung frekuensi kemunculan elemen, serta *mean_squared_error* dari *sklearn.metrics* untuk menghitung error kuadrat rata-rata.

```
def data_imputer(strategy='mean'):
    def impute(data):
        if strategy == 'mean':
            mean_val = sum(x for x in data if x is not None) / len([x for x in data if x is not None])
            return [x if x is not None else math.ceil(mean_val) for x in data]
        elif strategy == 'median':
            sorted_data = sorted(x for x in data if x is not None)
            median_val = sorted_data[len(sorted_data) // 2]
            return [x if x is not None else math.ceil(median_val) for x in data]
        elif strategy == 'mode':
            mode_val = Counter(x for x in data if x is not None).most_common(1)[0][0]
            return [x if x is not None else math.ceil(mode_val) for x in data]
        else:
            raise ValueError("Unsupported imputation strategy")
    return impute
```

Gambar 2 . Fungsi dalam fungsi

Kode ini menciptakan fungsi *data_imputer()* menggunakan *closure*. Fungsi ini menerima argumen '*strategy*' dengan nilai *default* 'mean'. Di dalamnya, ada fungsi *impute()* yang memproses data berdasarkan strategi yang diberikan. Jika strateginya 'mean', maka

nilai rata-rata dari data yang tidak *None* akan dihitung, dan nilai *None* akan diganti dengan rata-rata tersebut. Untuk strategi 'median', data diurutkan dan nilai tengahnya diambil sebagai median. Sedangkan untuk 'mode', nilai yang paling sering muncul dalam data dipilih sebagai modus. Fungsi *impute()* mengembalikan data yang telah diimputasi sesuai strategi. Jika strateginya tidak didukung, *ValueError* akan muncul. Terakhir, fungsi *data_imputer*() mengembalikan fungsi *impute*() yang diatur sesuai strategi.

```
def read_csv_file(filename):
    data = []
    with open(filename, 'r') as file:
        csv_reader = csv.reader(file)
        for row in csv_reader:
            data.extend(int(cell) if cell.strip().isdigit() else None for cell in row if cell.strip())
    return data
```

Gambar 3 . Fungsi untuk membaca file CSV

Fungsi *read_csv_file()* membaca file CSV dan mengembalikan data yang telah dibaca. Setiap baris dari file dibaca dan nilai-nilai yang merupakan digit disimpan sebagai integer dalam list data, sedangkan yang bukan digit diabaikan.

```
# Contoh penggunaan closure
# Imputasi menggunakan mean
impute_mean = data_imputer('mean')
imputed_data_mean = impute_mean(data)

# Imputasi menggunakan median
impute_median = data_imputer('median')
imputed_data_median = impute_median(data)

# Imputasi menggunakan modus
impute_mode = data_imputer('mode')
imputed_data_mode = impute_mode(data)
```

Gambar 4 . Membuat *closure*

Kode tersebut menggunakan konsep *closure* untuk membuat fungsi-fungsi imputasi data dengan strategi yang berbeda, seperti mean, median, dan modus. Selanjutnya, contoh penggunaan *closure* ini ditunjukkan dengan membuat tiga fungsi imputasi berbeda: mean, median, dan modus. Setiap fungsi imputasi kemudian dipanggil dengan data yang akan diimputasi untuk menghasilkan data yang telah diimputasi sesuai dengan strategi yang dipilih.

```
# Proses pengolahan data setelah imputasi
start_index = next((i for i, x in enumerate(data) if x is not None), None)
imputed_length_mean = len(imputed_data_mean[start_index:])
imputed_length_median = len(imputed_data_median[start_index:])
imputed_length_mode = len(imputed_data_mode[start_index:])

# Hitung RMSE untuk setiap metode imputasi
rmse_mean = math.sqrt(mean_squared_error(imputed_data_mean, imputed_data_mean))
rmse_median = math.sqrt(mean_squared_error(imputed_data_mean, imputed_data_median))
rmse_mode = math.sqrt(mean_squared_error(imputed_data_mean, imputed_data_mode))
```

Gambar 5 . Pengolahan data

Setelah proses imputasi data, langkah berikutnya adalah melakukan pengolahan data. Pertama, kode mencari indeks pertama dari data yang tidak kosong untuk memulai pengolahan. Selanjutnya, panjang data yang telah diimputasi dengan strategi mean, median, dan modus dihitung. Setelah itu, dilakukan perhitungan *Root Mean Square Error (RMSE)* untuk masing-masing metode imputasi.

```
# Pilih hasil imputasi dengan RMSE terkecil sebagai hasil terbaik
hasil_imputasi_terbaik = ''
if rmse_mean <= rmse_median and rmse_mean <= rmse_mode:
    hasil_imputasi_terbaik = 'Mean'
    print("Panjang data:", imputed_length_mean)
    print("Hasil imputasi menggunakan mean:", imputed_data_mean[start_index:])
    print(f"Metode imputasi data terbaik menggunakan mean dengan RMSE: {rmse_mean}")
elif rmse_median <= rmse_mean and rmse_median <= rmse_mode:
    hasil_imputasi_terbaik = 'Median'
    print("Panjang data:", imputed_length_median)
    print("Hasil imputasi menggunakan median:", imputed_data_median[start_index:])
    print(f"Metode imputasi data terbaik menggunakan median dengan RMSE: {rmse_median}")
else:
    hasil_imputasi_terbaik = 'Modus'
    print("Panjang data:", imputed_length_mode)
    print("Hasil imputasi menggunakan modus:", imputed_data_mode[start_index:])
    print(f"Metode imputasi data terbaik menggunakan modus dengan RMSE: {rmse_mode}")
```

Gambar 6. Memilih hasil imputasi data terbaik

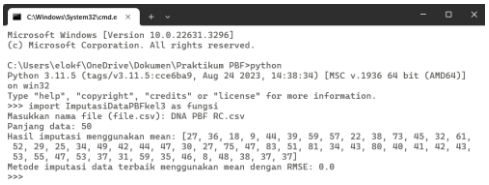
Kode tersebut bertujuan untuk memilih hasil imputasi terbaik berdasarkan nilai *Root Mean Square Error (RMSE)* terkecil. Jika *RMSE* menggunakan metode imputasi 'mean' lebih kecil dari *RMSE* metode imputasi 'median' dan 'mode', maka hasil imputasi terbaik dipilih sebagai 'mean'. Jika *RMSE* metode imputasi 'median' lebih kecil dari 'mean' dan 'mode', hasil imputasi terbaik adalah 'median'. Jika tidak, hasil imputasi terbaik adalah 'modus'. Setelah itu, panjang data hasil imputasi, hasil imputasi, dan nilai *RMSE* untuk metode terbaik akan ditampilkan.

```
Masukkan nama file (file.csv): DNA PBF RC.csv
Panjang data: 50
Hasil imputasi menggunakan mean: [27, 36, 18, 9, 44, 39, 59, 57,
22, 38, 73, 45, 32, 61, 52, 29, 25, 34, 49, 42, 44, 47, 30, 27,
75, 47, 83, 51, 81, 34, 43, 80, 40, 41, 42, 43, 53, 55, 47, 53,
37, 31, 59, 35, 46, 8, 48, 38, 37, 37]
Metode imputasi data terbaik menggunakan mean dengan RMSE: 0.0
```

Gambar 7 . Output yang dihasilkan menggunakan dataset

Output tersebut menunjukkan hasil dari proses imputasi data menggunakan metode mean. Terdapat 50 data yang telah diimputasi, dan nilai-nilai tersebut ditampilkan dalam bentuk list. Selain itu, hasil evaluasi menunjukkan bahwa metode imputasi terbaik adalah menggunakan mean, dengan nilai *Root Mean Square Error (RMSE)* sebesar 0.00, yang menunjukkan tingkat akurasi yang sangat tinggi.

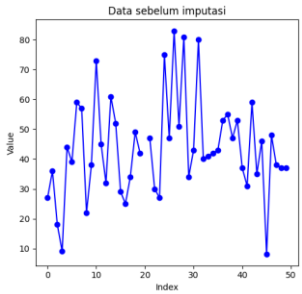
Menjalankan Program di Terminal



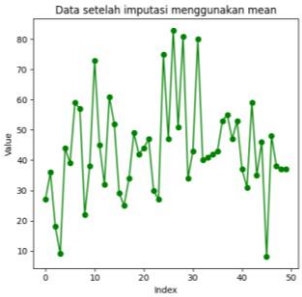
Gambar 8 . Menjalankan program pada *Command Prompt*

Kode Python dijalankan di *Command Prompt* setelah menginstall paket *scikit-learn* dengan perintah *pip install scikit-learn*. File ImputasiDataPBFkel3 diimpor sebagai fungsi. Pengguna diminta memasukkan nama file CSV. Setelah memproses data, hasil imputasi menggunakan metode mean ditampilkan. Evaluasi menunjukkan metode mean memiliki *RMSE* terendah, yaitu 0.0, menunjukkan akurasi yang tinggi.

Visualisasi Data



Gambar 9 . Visualisasi sebelum imputasi data



Gambar 10 . Visualisasi sesudah imputasi data menggunakan nilai mean

Berdasarkan Gambar 10, terlihat bahwa seluruh titik (*plot*) dihubungkan dengan garis, sedangkan pada Gambar 9, terdapat *plot* yang tidak dihubungkan dengan garis, yang mengindikasikan adanya data yang hilang. Kekosongan garis hubung terletak disekitar indeks ke-21 dalam dataset.

Simpulan

Kesimpulan artikel menekankan pentingnya penanganan nilai-nilai yang hilang dalam analisis data karena dapat berdampak signifikan pada hasil akhir. Metode imputasi seperti mean, median, dan modus menjadi pilihan umum untuk mengatasi masalah ini. Evaluasi menggunakan *Root Mean Square Error (RMSE)* menunjukkan bahwa metode mean memberikan hasil terbaik dengan *RMSE* terendah, menunjukkan akurasi tinggi dalam menangani data yang hilang. Artikel juga mencatat efisiensi akses ke variabel di lingkungan luar yang dimungkinkan oleh Python *closure*. Dengan menerapkan metode imputasi yang sesuai, diharapkan analisis data dapat dilakukan dengan lebih akurat dan andal, meningkatkan kualitas hasil analisis secara keseluruhan.

Referensi

Boediono, D. (2015). *Statistika dan Probabilitas.* Bandung: ROSDA.

Husnul, N. R., Prasetya, E. R., Sadewa, P., Ajimat, & Purnomo, L. I. (2020). *Statistika Deskriptif.* Pamulang: Unpam Press.

Little, R., & Rubin, D. (2019). *Statistical Analysis with Missing Data, Third Edition.* John Willey & Sons.

Marliana, R. R. (2019, Januari). Probabilitas dan Statistika. *Sekolah Tinggi Manajemen Informatika & Komputer*, (pp. 1-162). Sumedang.

Satria, A. (2024, Februari). *Modul 4 Praktikum Pemrograman Berbasis Fungsi*.

Somasundaram, R., & Nedunchezhian, R. (2011). Evaluation of three Simple Imputation Methods for. *International Journal of Computer Applications (0975 – 8887)*, 21, 14-19.

Suprayogi, I., Trimaijon, & Mahyudin. (2017). Model Prediksi Liku Kalibrasi Menggunakan Pendekatan Jaringan Saraf Tiruan (JST). *Media Neliti*, 1-18.