

APLIKASI PEMBERSIH DATA DENGAN PEMROGRAMAN PYTHON

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

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

Email:

deva.122450014@student.itera.ac.id, chevando.122450095@student.itera.ac.id,
rafly.122450140@student.itera.ac.id, novelia.122450104@student.itera.ac.id,
nabiilah.122450029@student.itera.ac.id

1. Pendahuluan

Semakin populernya bidang sains data dalam era digital, di mana peralatan teknologi yang semakin canggih, telah menciptakan lonjakan data dalam berbagai bidang. Produksi data yang melimpah, peranan ilmu sains data menjadi semakin krusial. Data, sebagai fakta yang menjelaskan objek tertentu, dapat berupa nilai kuantitatif, kualitatif, gambar, maupun teks. Tak peduli dari mana sumbernya, baik mesin, perangkat pintar, atau kuesioner, semua akan menghasilkan data mentah yang mungkin kotor (*noisy data*) bahkan data yang tidak lengkap dan tidak konsisten. Kondisi data yang demikian dapat memberikan dampak signifikan pada hasil analisis (Yusof et al., n.d., 11).

Inisiatif untuk membersihkan, merapikan, dan normalisasi data merupakan langkah awal sebelum melakukan analisis lebih lanjut. Pembersihan data (*data cleaning*) merupakan proses penting dalam manajemen dan analisis data. Ini melibatkan identifikasi, koreksi, atau penghapusan data yang tidak akurat, tidak konsisten, duplikat, salah format, atau kesalahan lainnya dari sebuah kumpulan data. Salah satu masalah umum yang diatasi selama pembersihan data adalah konsistensi dalam *menginput* data. Ini dapat mencakup variasi dalam ejaan, singkatan, unit pengukuran yang berbeda untuk variabel yang sama, kelebihan spasi, ketidaksesuaian teks, atau menangani beberapa karakter yang tidak diperlukan (Lubis, 2023, 3-4).

Berdasarkan uraian di atas, maka dibuatlah suatu program aplikasi pembersihan data yang mencakup penghapusan spasi berlebih, pengkonversian teks menjadi huruf kecil dan huruf besar, hingga menghilangkan suatu karakter non-alfanumerik, yaitu mencakup tanda baca dan spasi (Putra et al., 2022, 4), pada sebuah teks. Program aplikasi ini bertujuan untuk membuat data menjadi lebih konsisten dan akurat sehingga mempermudah dalam melakukan analisis lebih lanjut terhadap data tersebut.

2. Metode

Dalam membuat program aplikasi pembersihan data ini, digunakan beberapa fungsi sebagai berikut:

2.1 Fungsi Map

Fungsi 'map()' memetakan setiap nilai dalam array tertentu ke nilai baru dengan menerapkan fungsi lambda ke setiap nilai.

2.2 Fungsi Lambda

Fungsi 'lambda' digunakan untuk membuat dan menampilkan fungsi kustom menggunakan kumpulan nama dan formula expression yang menggunakannya.

2.3 Fungsi List

Fungsi 'list()' digunakan untuk menyimpan kumpulan nilai atau objek sama seperti array tetapi dalam fungsi ini bisa menggunakan tipe data apapun.

2.4 Fungsi Join

Fungsi 'join()' digunakan untuk menggabungkan elemen dari array satu atau beberapa dimensi menggunakan pembatas yang ditentukan.

2.5 Fungsi Split

Fungsi 'split()' bertugas memisahkan teks dengan menggunakan karakter atau string tertentu, dan menempatkan setiap fragmen ke dalam sel terpisah pada baris.

2.6 Fungsi Lower

Fungsi 'lower()' pada Python berguna untuk mengubah semua ukuran karakter dalam string yang berukuran besar menjadi kecil.

2.7 Fungsi Upper

Fungsi 'upper()' pada Python berguna untuk mengubah semua ukuran karakter dalam string yang berukuran kecil menjadi besar atau kapital.

2.8 Fungsi Filter

Fungsi 'filter()' merupakan fungsi bawaan Python yang memiliki teknik yang digunakan untuk memilih elemen tertentu dalam suatu objek seperti list, tuple, set, dan *dictionary* yang menggunakan ekspresi lambda berdasarkan kriteria tertentu atau yang ditentukan.

2.9 Fungsi Isalnum

Fungsi 'isalnum()' merupakan fungsi bawaan Python yang digunakan untuk mendeteksi apakah seluruh karakter dalam string termasuk dalam *alphanumeric* angka dan huruf.

3. Pembahasan

Pada kesempatan kali ini, kami membuat empat buah program aplikasi pembersihan data. Keempat program aplikasi tersebut masing-masing melakukan pembersihan untuk menghapus spasi berlebih, mengkonversi teks menjadi huruf kecil, mengkonversi teks menjadi huruf besar, dan menghapus karakter non-alfanumerik.

3.1 Menghapus Spasi Berlebih

3.1.1 Code dan Output

```
spasi_berlebih = lambda data: list(map(lambda x: ' '.join(x.split()), data))

# inputan data
my_data = [" ITERA ", " Friends!", "Angkatan 2022 ", "Semangat terus, ya!"]

# membersihkan data
data_bersih = spazi_berlebih(my_data)

# output
print("Data sebelum dibersihkan:")
for x in my_data:
    print(x)
print("\nData setelah dibersihkan:")
for x in data_bersih:
    print(x)
```

Gambar 1

```
Data sebelum dibersihkan:
ITERA
Friends!
Angkatan 2022
Semangat terus, ya!

Data setelah dibersihkan:
ITERA
Friends!
Angkatan 2022
Semangat terus, ya!
```

Gambar 2

3.1.2 Analisis

Program diatas mendefinisikan fungsi lambda bernama “spasi_berlebih”. Fungsi tersebut hanya menerima satu argumen, yaitu “data”, yang merupakan string sebuah list dan bertujuan untuk menghapus spasi berlebih. Di dalam fungsi tersebut, terdapat fungsi lambda “lambda x: ‘.join(x.split())” yang tugasnya menghapus spasi berlebih dari setiap string dengan “x.plit()” yang nantinya akan membagi string menjadi kata-kata dan kemudian digabungkan menjadi satu string kembali dengan satu spasi di tiap kata menggunakan “ ‘.join() ”. Fungsi lambda tersebut diterapkan untuk setiap string dalam data melalui fungsi “map()”. Kemudian, fungsi tersebut diterapkan dalam fungsi list(). Sementara itu, “data_bersih” bertugas untuk memanggil hasil pembersihan data yang dilakukan. Dengan inputan data my_data = [" ITERA ", " Friends!", "Angkatan 2022 ", "Semangat terus, ya!"], semua string dalam “my_data” akan dibagi dengan fungsi “split()”, lalu spasi berlebih di dalamnya dihapus kemudian string yang sudah bersih digabung oleh fungsi “lambda x: ‘.join(x.split())”, dan outputnya akan disimpan dalam “data_bersih” kemudian dicetak satu per satu ke dalam bentuk list menggunakan fungsi “list()” tersebut serta perulangan for loop. Output tersebut dapat dilihat pada Gambar 2.

3.2 Mengkonversi Teks Menjadi Huruf Kecil

3.2.1 Code dan Output

```
konversi_kecil = lambda data: list(map(lambda x: x.lower(), data))

# inputan data
my_data = ["ITERA", "Friends!", "Angkatan 2022", "Semangat terus, ya!"]

# membersihkan data
data_bersih = konversi_kecil(my_data)

# output
print("Data sebelum dibersihkan:")
for x in my_data:
    print(x)
print("\nData setelah dibersihkan:")
for x in data_bersih:
    print(x)
```

Gambar 3

```
Data sebelum dibersihkan:
ITERA
Friends!
Angkatan 2022
Semangat terus, ya!

Data setelah dibersihkan:
itera
friends!
angkatan 2022
semangat terus, ya!
```

Gambar 4

3.2.2 Analisis

Program di atas mendefinisikan fungsi lambda bernama “konversi_kecil” yang hanya menerima satu argumen, yaitu “data”. Fungsi tersebut yang digunakan untuk melakukan pengkonversian setiap string dalam “data” menjadi huruf kecil. Dalam fungsi tersebut, diterapkan fungsi lambda “lambda x: x.lower()” untuk mengubah setiap string tersebut menjadi huruf kecil. Fungsi lambda tersebut diterapkan dalam fungsi map() untuk menerapkan fungsi lambda “lambda x: x.lower()” ke setiap string di dalam “data” agar diubah menjadi huruf kecil. Fungsi-fungsi tersebut diterapkan pada fungsi list(). Sementara itu, “data_bersih” bertugas untuk memanggil hasil pembersihan data yang dilakukan. Dengan inputan data my_data = ["ITERA", "Friends!", "Angkatan 2022", "Semangat terus, ya!"], semua string dalam “my_data” akan diubah menjadi huruf kecil dengan fungsi lambda “x.lower()” dan outputnya akan disimpan dalam “data_bersih” untuk kemudian dicetak satu per satu ke dalam bentuk list oleh fungsi “list()” tersebut menggunakan perulangan for loop. Output tersebut dapat dilihat pada Gambar 4.

3.3 Mengkonversi Teks Menjadi Huruf Besar

3.3.1 Code dan Ouput

```
konversi_besar = lambda data: list(map(lambda x: x.upper(), data))

# inputan data
my_data = ["ITERA", "Friends!", "Angkatan 2022", "Semangat terus, ya!"]

# membersihkan data
data_bersih = konversi_besar(my_data)

# output
print("Data sebelum dibersihkan:")
for x in my_data:
    print(x)
print("\nData setelah dibersihkan:")
for x in data_bersih:
    print(x)
```

Gambar 5

```
Data sebelum dibersihkan:
ITERA
Friends!
Angkatan 2022
Semangat terus, ya!

Data setelah dibersihkan:
ITERA
FRIENDS!
ANGKATAN 2022
SEMANGAT TERUS, YA!
```

Gambar 6

3.3.2 Analisis

Program di atas mendefinisikan fungsi lambda bernama “konversi_besar” yang hanya menerima satu argumen, yaitu “data”. Fungsi tersebut digunakan untuk melakukan pengkonversian masing-masing string dalam “data” menjadi huruf besar. Dalam fungsi tersebut, diterapkan fungsi lambda “lambda x: x.upper()” untuk mengubah setiap string tersebut menjadi huruf besar. Fungsi lambda tersebut diterapkan dalam fungsi map() untuk menerapkan fungsi lambda “lambda x: x.upper()” ke masing-masing string tersebut agar diubah menjadi huruf besar. Fungsi-fungsi tersebut diterapkan pada fungsi list(). Sementara itu, “data_bersih” bertugas untuk memanggil hasil pembersihan data yang dilakukan. Dengan inputan data my_data = ["ITERA", " Friends!", "Angkatan 2022", "Semangat terus, ya!"], semua string dalam “my_data” akan diubah menjadi huruf besar dengan fungsi lambda “x.upper()” dan outputnya akan disimpan dalam “data_bersih” untuk kemudian dicetak satu per satu ke dalam bentuk list oleh fungsi “list()” tersebut menggunakan perulangan for loop. Output tersebut dapat dilihat pada Gambar 6.

3.4 Menghilangkan Karakter Non-Alfanumerik

3.4.1 Code

```
del_nonalfanumerik = lambda data: list(map(lambda x: ''.join(filter(str.isalnum, x)), data))

# inputan data
my_data = [" ITERA ", " Friends!", "Angkatan 2022 ", "Semangat terus, ya!"]

# membersihkan data
data_bersih = del_nonalfanumerik(my_data)

# output
print("Data sebelum dibersihkan:")
for x in my_data:
    print(x)
print("\nData setelah dibersihkan:")
for x in data_bersih:
    print(x)
```

Gambar 7

```
Data sebelum dibersihkan:
 ITERA 
 Friends!
Angkatan 2022 
Semangat terus, ya!

Data setelah dibersihkan:
ITERA
Friends
Angkatan2022
Semangatterusya
```

Gambar 8

3.4.2 Analisis

Program di atas mendefinisikan fungsi lambda bernama “del_nonalfanumerik” untuk menghilangkan karakter non-alfanumerik pada setiap string dalam argumen “data”. Dalam fungsi lambda tersebut diterapkan fungsi lambda “lambda x: ‘.join(filter(str.isalnum,x))” yang mana fungsi filter(str.isalnum, x) akan melakukan penyaringan terhadap karakter non-alfanumerik dari setiap string dalam “data” atau x, selanjutnya karakter yang sudah disaring tersebut akan disatukan oleh fungsi “ ‘.join() ”

menjadi string baru dalam bentuk alfanumerik. Fungsi lambda tersebut diterapkan pada fungsi map() untuk menerapkan fungsi-fungsi tersebut ke setiap string dalam “data”. Fungsi-fungsi tersebut diterapkan pada fungsi list(). Sementara itu, “data_bersih” bertugas untuk memanggil hasil pembersihan data yang dilakukan. Dengan inputan my_data = ["ITERA ", " Friends!", "Angkatan 2022 ", "Semangat terus, ya!"], seluruh string dalam list “my_data” akan difilter karakter non-alfanumeriknya oleh “str.isalnum” dalam fungsi filter(), lalu digabungkan kembali menjadi string yang sudah bersih dari karakter non-alfanumerik oleh fungsi join(), outputnya disimpan dalam “data_bersih” berupa string baru dengan karakter alfanumerik untuk kemudian dicetak satu per satu ke dalam bentuk list oleh fungsi “list()” tersebut menggunakan perulangan for loop. Output tersebut dapat dilihat pada Gambar 8.

4. Kesimpulan

Berdasarkan pembahasan yang dijelaskan dalam jurnal ini mengenai aplikasi pembersih data menggunakan pemrograman Python, dapat disimpulkan bahwa dengan menggunakan kombinasi fungsi ‘map’ dan fungsi ‘lambda’ serta fungsi bawaan lainnya dalam Python, sangat berguna untuk melakukan transformasi data dan melakukan pembersihan data dengan cepat dan efisien. Kemampuan yang memungkinkan otomatisasi proses transformasi data, seperti menghilangkan spasi tambahan, mengkonversi teks ke dalam format tertentu, serta mengubah atau membersihkan data dari karakter non-alfanumerik, tersedia dalam fitur-fitur pada aplikasi pembersih data. Secara keseluruhan, kombinasi yang ada dalam aplikasi ini membantu meningkatkan kualitas dan konsistensi data serta mempercepat proses analisis dan pemrosesan data.

5. Referensi

- Lubis, S. A. H. (2023, Desember 22). *MENINGKATKAN KUALITAS BIG DATA MELALUI REKAYASA DAN PEMBERSIHAN DATA YANG EFEKTIF*, 3-4. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.researchgate.net/profile/Syaiful-Lubis-2/publication/376720216_MENINGKATKAN_KUALITAS_BIG_DATA_MELALUI_REKAYASA_DAN_PEMBERSIHAN_DATA_YANG_EFEKTIF/links/658502ad2468df72d3c3b465/MENINGKATKAN-KUA
- Putra, T. I. Z. M., Suprpto, & Bukhori, A. F. (2022, November 11). Jurnal Ilmu Siber dan Teknologi Digital (JISTED). *Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia*, 1, 4. <https://penerbitgoodwood.com/index.php/jisted/article/view/1509>
- Yusof, R., Mohd Yunus, S. Y., Rahman, N. A., Hashim, N., & Fadzillah, N. A. A. (n.d.). APB Rembau e-Bulletin. *PEMBERSIHAN DATA DALAM DUNIA DATA SAINS*, 11. <https://ir.uitm.edu.my/id/eprint/77576/1/77576.pdf>