Anggota Kelompok :
- Nadini Annisa Byant (2111522021)
- Amalia Sandi Alzahrah (2111522022)

AKUISISI DATA - A
LATIHAN WRANGLING DAN VISUALISASI DATA

CLEANING DATA

1. Mengecek *missing value* di masing-masing kolom
   Pada tahapan ini akan dilakukan pengecekan nilai kosong pada seluruh kolom yaitu Nama Kegiatan dan Waktu Penyelenggaraan, Tingkat, Prestasi, dan Tahun. Syntax dari tahapan ini dapat dilihat pada gambar tersebut :

```
# Cek missing value dalam masing-masing kolom
missing_values = df.isnull().sum()

# Tampilkan hasil
print(missing_values)
```

   Lalu akan tampil hasil dari pengecekan tersebut, yaitu sebagai berikut :

```
Nama Kegiatan dan Waktu Penyelenggaraan    0
Tingkat                                    0
Prestasi                                   0
Tahun                                      0
dtype: int64
```

   berdasarkan data diatas didapatkan bahwa tidak ada nilai kosong pada kolom Nama kegiatan dan waktu penyelenggaraan, tingkat, prestasi, dan tahun.

2. Pengecekan Tipe Data dan Merubah Tipe Data
   Pada tahapan ini dilakukan pengecekan tipe data pada masing-masing kolom dan dilakukan perubahan tipe data agar menyamakan semua format data tersebut, dapat dilihat pada gambar berikut :

```
# Cek tipe data dari masing-masing kolom
tipe_data = df.dtypes

# Tampilkan hasil
print(tipe_data)
```

```
Nama Kegiatan dan Waktu Penyelenggaraan    object
Tingkat                                    object
Prestasi                                   object
Tahun                                      object
dtype: object
```

Berdasarkan gambar tersebut, didapatkan bahwasanya seluruh kolom ini bertipe data kan object, lalu kami akan mengganti tipe data nya menjadi string dan date untuk kolom Tahun, dapat dilihat pada gambar berikut :

```
# Mengubah tipe data kolom 'Tahun' menjadi datetime
df['Tahun'] = pd.to_datetime(df['Tahun'], format='%Y')
# Tampilkan DataFrame setelah mengubah tipe data
print(df.dtypes)
```

```
Nama Kegiatan dan Waktu Penyelenggaraan            object
Tingkat                                            object
Prestasi                                           object
Tahun                                      datetime64[ns]
dtype: object
```

Pada gambar tersebut dilakukan perubahan tipe data pada kolom tahun menjadi date, dan yang lainnya itu menjadi string, namun pada format data csv object itu dapat diartikan sebagai tipe data string.

3. Menyamakan format penulisan pada kolom prestasi
   Pada tahapan ini akan dilakukan penghapusan kalimat Tingkat pada beberapa nilai prestasi, karena kami ingin melakukan pengelompokan prestasi menjadi sumatera barat, lokal, nasional, dan internasional. Syntax dapat dilihat pada gambar berikut :

```
#membersihkan Tingkat
df['Tingkat'] = df['Tingkat'].str.replace('tingkat ', '')
```

```
print(df['Tingkat'])
```

```
0     se-sumbar
1      nasional
2      nasional
3         lokal
4      nasional
        ...
68     nasional
69     nasional
70     nasional
71     nasional
72     nasional
Name: Tingkat, Length: 73, dtype: object
```

Berdasarkan gambar tersebut didapatkan hasilnya bahwa nilai pada kolom prestasi yang mengandung kalimat tingkat itu sudah hilang dan hanya berganti menjadi 1 kalimat saja yaitu nasional, lokal, internasional dan sumatera barat.

```
# Mendapatkan nilai unik dari kolom 'Tingkat'
tingkat_unik = df['Tingkat'].unique()
# Tampilkan nilai unik
print(tingkat_unik)
```

```
['se-sumbar' 'nasional' 'lokal' 'sumatera barat' 'internasional']
```

TRANFORMATION DATA

1. Lowercase

   Pada tahapan ini akan dilakukan perubahan penulisan kalimat pada masing-masing kolom menjadi *lowercase* yaitu diawali oleh huruf kecil semua. syntax dapat dilihat pada gambar berikut :

```
df['Nama Kegiatan dan Waktu Penyelenggaraan'] = df['Nama Kegiatan dan Waktu Penyelenggaraan'].str.lower()
df['Tingkat'] = df['Tingkat'].str.lower()
df['Prestasi'] = df['Prestasi'].str.lower()
```

2. Merubah nilai kolom prestasi menjadi unik

   Pada tahapan ini kami akan melakukan perubahan kalimat, yaitu menyamakan penulisan juara 1, juara 2, dan juara 3. syntax dapat dilihat pada gambar berikut :

```
# Mendapatkan nilai unik dari kolom 'Prestasi'
prestasiawal_unik = df['Prestasi'].unique()
print(prestasiawal_unik)
```

```
['juara 2' 'pemateri' 'sem1finalis' 'juara i' 'juara ii'
 'peserta vocal group' 'juara favorite creative idea'
 'penerima seed capital' 'juara iii' 'wakil sumatera barat' 'finalis'
 'kontributor' 'peserta' 'top 10 innovation' 'poster terbaik'
 'juara iidengan tema “smart city”'
 'top 7 finalis hackathon dillo festifal 2018' 'juara 3'
 'harapan 1 lomba vokal group perwakilan dar1pengprof bpsmi'
 'finalis informatic festival and competition (invention) 2018'
 'top 20 besar' 'semifinalis national youth writing competition 2018'
 'juara 2tingkat mahasiswa se – indonesia'
 'rangking 3, sebaga1paper inovas1potensial kategor1smart city'
 'pemenang pendamping, sebaga1paper inovas1potensial kategor1ekonom1kreatif'
 'top 5 besar' 'juara harapan 3' 'finalis (best 3iitalent)' 'top 6'
 'juara 1' 'karya terbaik' '10 besar']
```

```
# Mengganti "juara ii" menjadi "juara 2" dan "juara i" menjadi "juara 1" dalam kolom 'Prestasi'
df['Prestasi'] = df['Prestasi'].str.replace(r'juara ii', 'juara 2', case=False)
df['Prestasi'] = df['Prestasi'].str.replace(r'juara i', 'juara 1', case=False)
df['Prestasi'] = df['Prestasi'].str.replace(r'juara 2i', 'juara 2', case=False)
```

Dari gambar tersebut dilakukan perubahan pada kalimat juara ii menjadi juara 2, juara i menjadi juara 1, juara 2i menjadi juara 2, sehingga data pada kolom prestasi menjadi seperti gambar dibawah ini :

```
# Mendapatkan nilai unik dari kolom 'Tingkat'
prestasi_unik = df['Prestasi'].unique()
# Tampilkan nilai unik
print(prestasi_unik)
```

```
['juara 2' 'pemateri' 'sem1finalis' 'juara 1' 'peserta vocal group'
 'juara favorite creative idea' 'penerima seed capital'
 'wakil sumatera barat' 'finalis' 'kontributor' 'peserta'
 'top 10 innovation' 'poster terbaik' 'juara 2dengan tema “smart city”'
 'top 7 finalis hackathon dillo festifal 2018' 'juara 3'
 'harapan 1 lomba vokal group perwakilan dar1pengprof bpsmi'
 'finalis informatic festival and competition (invention) 2018'
 'top 20 besar' 'semifinalis national youth writing competition 2018'
 'juara 2tingkat mahasiswa se – indonesia'
 'rangking 3, sebaga1paper inovas1potensial kategor1smart city'
 'pemenang pendamping, sebaga1paper inovas1potensial kategor1ekonom1kreatif'
 'top 5 besar' 'juara harapan 3' 'finalis (best 3iitalent)' 'top 6'
 'karya terbaik' '10 besar']
```

VISUALISASI DATA DAN ANALISIS

1. Jumlah Prestasi Per Tahun

Versi barchart

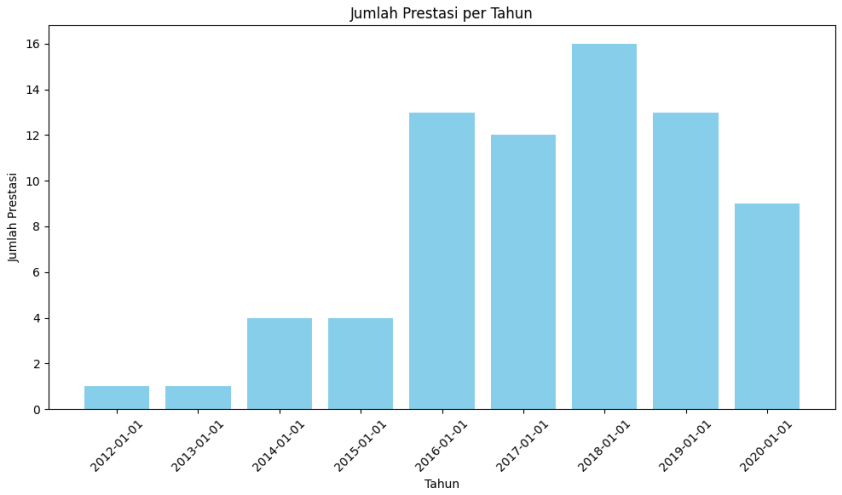
```python
# Mengonversi data ke dalam format DataFrame
data = pd.DataFrame(df)

# Menghitung jumlah prestasi per tahun
jumlah_prestasi_per_tahun = data.groupby('Tahun')['Prestasi'].count()

# Membuat plot bar untuk jumlah prestasi per tahun
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.bar(jumlah_prestasi_per_tahun.index.astype(str), jumlah_prestasi_per_tahun.values, color='skyblue')
plt.xlabel('Tahun')
plt.ylabel('Jumlah Prestasi')
plt.title('Jumlah Prestasi per Tahun')
plt.xticks(rotation=45)
plt.tight_layout()

# Menampilkan plot
plt.show()
```

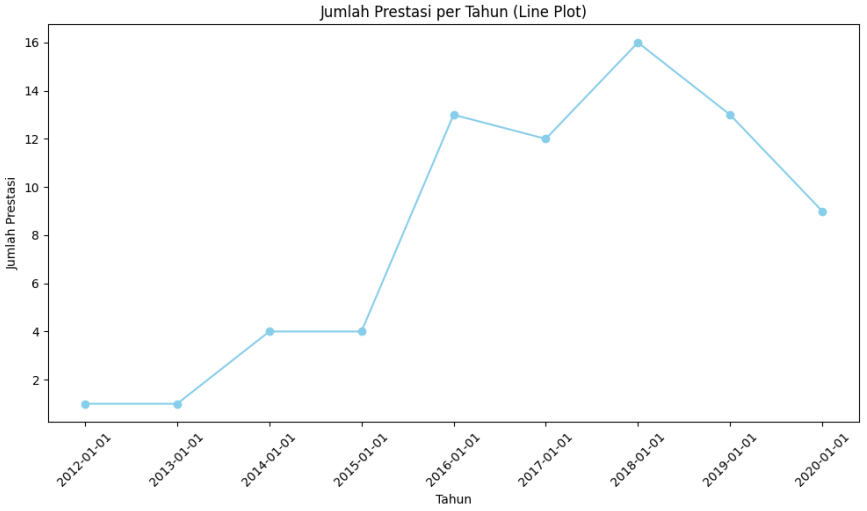


Versi line plot

```
# Menghitung jumlah prestasi per tahun
jumlah_prestasi_per_tahun = data.groupby('Tahun')['Prestasi'].count()

# Membuat line plot untuk jumlah prestasi per tahun
plt.figure(figsize=(10, 6))
plt.plot(jumlah_prestasi_per_tahun.index.astype(str), jumlah_prestasi_per_tahun.values, marker='o', linestyle='-', color='skyblue')
plt.xlabel('Tahun')
plt.ylabel('Jumlah Prestasi')
plt.title('Jumlah Prestasi per Tahun (Line Plot)')
plt.xticks(rotation=45)
plt.tight_layout()

# Menampilkan plot
plt.show()
```

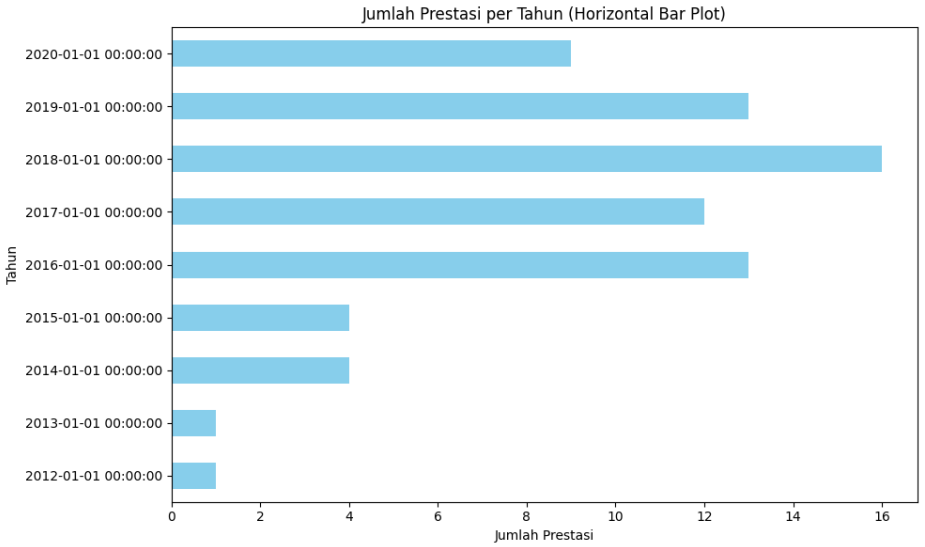


Versi horizontal bar

```
# Menghitung jumlah prestasi per tahun
jumlah_prestasi_per_tahun = data.groupby('Tahun')['Prestasi'].count()

# Membuat horizontal bar plot untuk jumlah prestasi per tahun
plt.figure(figsize=(10, 6))
jumlah_prestasi_per_tahun.plot(kind='barh', color='skyblue')
plt.xlabel('Jumlah Prestasi')
plt.ylabel('Tahun')
plt.title('Jumlah Prestasi per Tahun (Horizontal Bar Plot)')
plt.tight_layout()

# Menampilkan plot
plt.show()
```



Penjelasan:

Pada grafik tersebut dibagi menjadi 2 sumbu yaitu sumbu x dan sumbu y, sumbu x akan mempresentasikan sebagai jumlah prestasi, dan sumbu ya akan mempresentasikan sebagai tahun prestasi. Pada data tahun terdapat 9 jenis tahun yaitu 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, dan 2020. Dari grafik tersebut didapatkan hasil yaitu 2012 sebanyak 1, 2013 sebanyak 1, 2014 sebanyak 4, 2015 sebanyak 4, 2016 sebanyak 13, 2017 sebanyak 12, 2018 sebanyak 16, 2019 sebanyak 13, dan 2020 sebanyak 9.

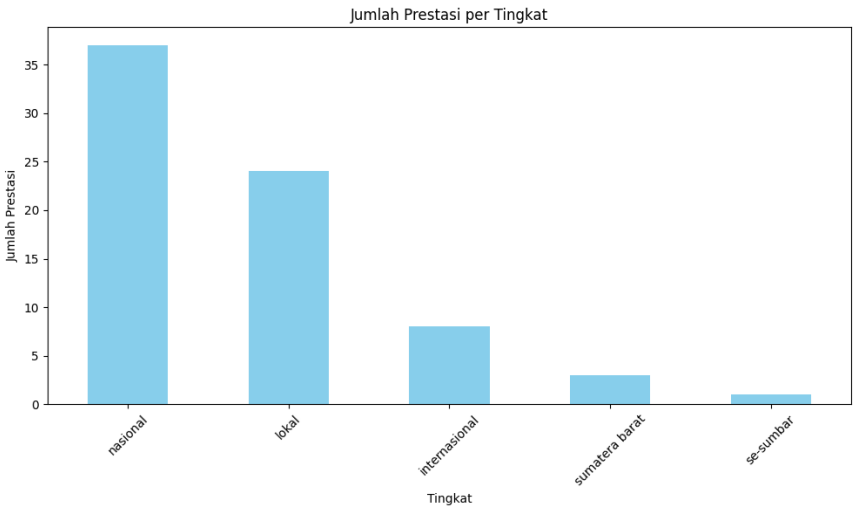
Dari angka tersebut dan dilihat dari barchart kita dapat menyimpulkan beberapa hal yaitu tahun yang paling banyak memegang prestasi yaitu adalah pada tahun 2018, dan tahun yang paling sedikit memegang prestasi yaitu adalah pada tahun 2012 dan 2013. Berdasarkan grafik line plot, juga dapat kita simpulkan bahwa kurva dari prestasi yang didapatkan terus meningkat dari tahun 2012 hingga 2018, namun terjadi sedikit penurunan prestasi pada tahun 2019 - 2020.

2. Jumlah Prestasi Per Tingkat
   versi barchart

```python
# Menghitung jumlah prestasi per tingkat
jumlah_prestasi_per_tingkat = df['Tingkat'].value_counts()

# Membuat plot bar untuk jumlah prestasi per tingkat
plt.figure(figsize=(10, 6))
jumlah_prestasi_per_tingkat.plot(kind='bar', color='skyblue')
plt.xlabel('Tingkat')
plt.ylabel('Jumlah Prestasi')
plt.title('Jumlah Prestasi per Tingkat')
plt.xticks(rotation=45)  # Mengatur label sumbu x agar mudah dibaca
plt.tight_layout()

# Menampilkan plot
plt.show()
```



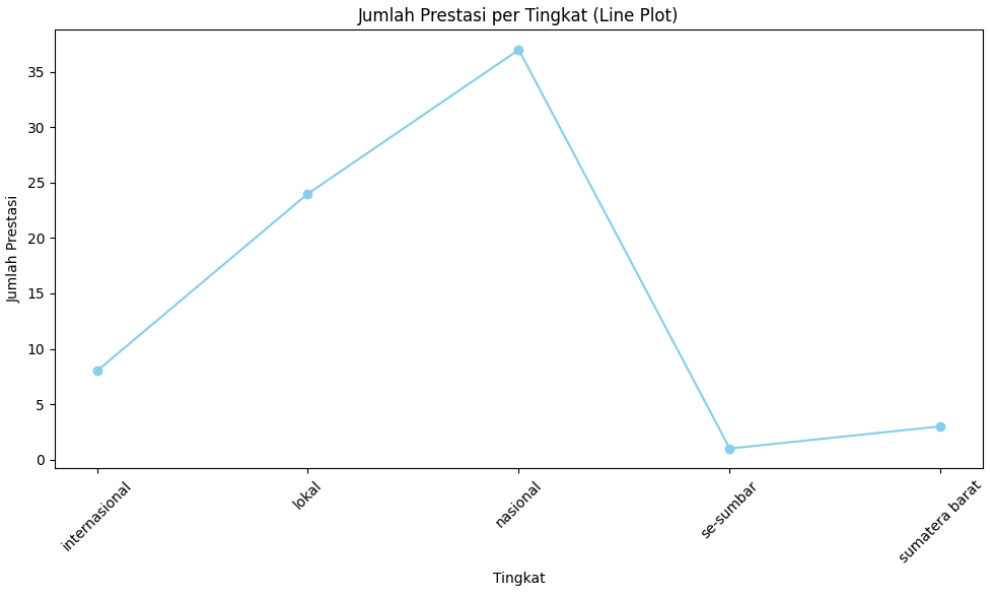
Versi lineplot

```python
# Menghitung jumlah prestasi per tingkat
jumlah_prestasi_per_tingkat = df['Tingkat'].value_counts()

# Mengurutkan indeks agar terlihat lebih baik dalam line plot
jumlah_prestasi_per_tingkat = jumlah_prestasi_per_tingkat.sort_index()

# Membuat line plot untuk jumlah prestasi per tingkat
plt.figure(figsize=(10, 6))
plt.plot(jumlah_prestasi_per_tingkat.index, jumlah_prestasi_per_tingkat.values, marker='o', linestyle='-', color='skyblue')
plt.xlabel('Tingkat')
plt.ylabel('Jumlah Prestasi')
plt.title('Jumlah Prestasi per Tingkat (Line Plot)')
plt.xticks(rotation=45)
plt.tight_layout()

# Menampilkan plot
plt.show()
```



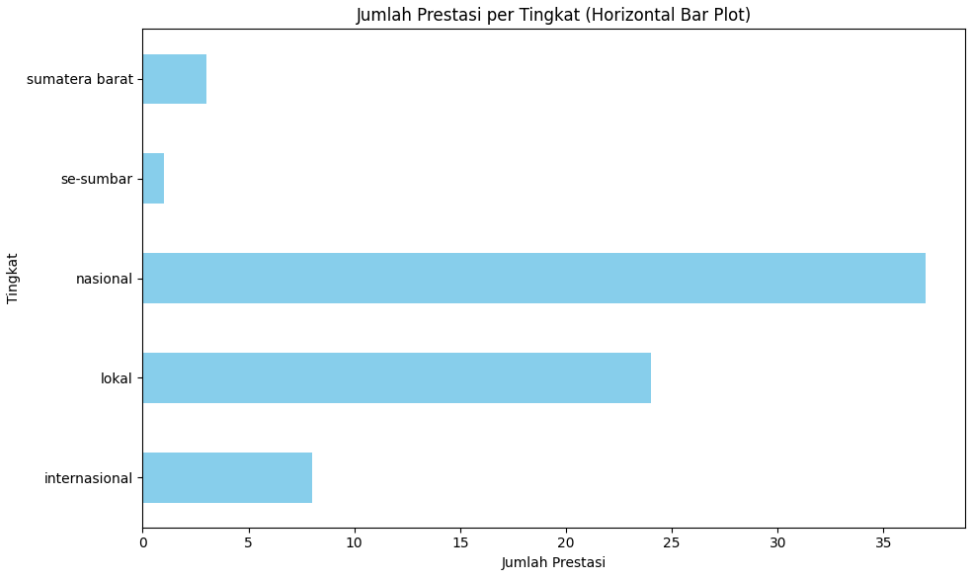
Versi horizontal bar

```python
# Menghitung jumlah prestasi per tingkat
jumlah_prestasi_per_tingkat = df['Tingkat'].value_counts()

# Mengurutkan indeks agar terlihat lebih baik dalam horizontal bar plot
jumlah_prestasi_per_tingkat = jumlah_prestasi_per_tingkat.sort_index()

# Membuat horizontal bar plot untuk jumlah prestasi per tingkat
plt.figure(figsize=(10, 6))
jumlah_prestasi_per_tingkat.plot(kind='barh', color='skyblue')
plt.ylabel('Tingkat')
plt.xlabel('Jumlah Prestasi')
plt.title('Jumlah Prestasi per Tingkat (Horizontal Bar Plot)')
plt.tight_layout()

# Menampilkan plot
plt.show()
```



Penjelasan

Dengan melihat visualisasi ini, kita dapat dengan mudah membandingkan jumlah prestasi antara berbagai tingkat. Misalnya, apakah lebih banyak prestasi Nasional daripada Regional atau Lokal. Dari berbagai data yang ada disana, dapat kita lihat bahwa prestasi tingkat nasional berjumlah lebih dari 30, lokal lebih dari 20, se sumatera barat yang jika dijumlahkan sekitar 5, dan internasional tidak sampai 10.
Visualisasi ini membantu dalam pemahaman tingkat prestasi yang dicapai dalam berbagai kegiatan atau kompetisi. Ini memungkinkan kita untuk melihat distribusi prestasi dalam berbagai tingkat kompetisi.