```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Create a mapping of Indonesian month names to their respective numbers
month_mapping = {
    'Januari': '01',
    'Februari': '02',
    'Maret': '03',
    'April': '04',
    'Mei': '05',
    'Juni': '06',
    'Juli': '07',
    'Agustus': '08',
    'September': '09',
    'Oktober': '10',
    'November': '11',
    'Desember': '12'
}
```

```
import pandas as pd

# Load the CSV file
file_path = 'data-harga-pangan-konsumen-per-konsumen.xlsx'
data = pd.read_excel(file_path)

# Display the first few rows of the dataset
data.head()
```

	No.	Kode Provinsi	Provinsi	Kode Komoditas	Komoditas	Tahun	Bulan	Harga
0	1	11	Aceh	NaN	Beras Premium	2021	Januari	12.072
1	2	12	Sumatera Utara	NaN	Beras Premium	2021	Januari	12.523
2	3	13	Sumatera Barat	NaN	Beras Premium	2021	Januari	13.096
3	4	14	Riau	NaN	Beras Premium	2021	Januari	14.605

Next steps: Generate code with data View recommended plots New interactive sheet

```
# Check for NaN or non-finite values in 'Harga'
print(data['Harga'].isnull().sum())  # Count of NaN values
print(data['Harga'].unique())  # Unique values in the column
```

```
37
[12.072 12.523 13.096 ... 26.67  24.665 23.545]
```

```
# Replace NaN values with the mean price for that commodity
data['Harga'] = data.groupby('Komoditas')['Harga'].transform(lambda x: x.fillna(x.mean()))
```

```
# Remove currency symbol and commas, then convert to integer
data['Harga'] = data['Harga']
# Replace month names with numbers using the mapping
data['Bulan'] = data['Bulan'].replace(month_mapping)

# Now combine 'Tahun' and 'Bulan' into a single datetime column
data['Tanggal'] = pd.to_datetime(data['Bulan'] + ' ' + data['Tahun'].astype(str), format='%m %Y')

# Drop the original 'Tahun' and 'Bulan' columns if not needed
data.drop(['Tahun', 'Bulan','Kode Komoditas','Kode Provinsi','No.'], axis=1, inplace=True)

# Display the cleaned data
data.head()
```

	Provinsi	Komoditas	Harga	Tanggal
0	Aceh	Beras Premium	12.072	2021-01-01
1	Sumatera Utara	Beras Premium	12.523	2021-01-01
2	Sumatera Barat	Beras Premium	13.096	2021-01-01
3	Riau	Beras Premium	14.605	2021-01-01

Next steps: Generate code with `data` View recommended plots New interactive sheet

Analisis

```
# Filter untuk komoditas yang diinginkan
selected_commodities = ['Cabai Merah Keriting', 'Cabai Rawit Merah']
filtered_data = data[data['Komoditas'].isin(selected_commodities)]

# 1. Ringkasan Data per Komoditas
average_prices = filtered_data.groupby('Komoditas')['Harga'].mean().reset_index()
highest_price = average_prices.loc[average_prices['Harga'].idxmax()]
lowest_price = average_prices.loc[average_prices['Harga'].idxmin()]

print("Harga Rata-rata per Komoditas:")
print(average_prices)
print("\nKomoditas dengan Harga Rata-rata Tertinggi:", highest_price)
print("Komoditas dengan Harga Rata-rata Terendah:", lowest_price)
```

```
Harga Rata-rata per Komoditas:
              Komoditas       Harga
0  Cabai Merah Keriting  154.933345
1     Cabai Rawit Merah  174.778504

Komoditas dengan Harga Rata-rata Tertinggi: Komoditas    Cabai Rawit Merah
Harga                174.778504
Name: 1, dtype: object
Komoditas dengan Harga Rata-rata Terendah: Komoditas    Cabai Merah Keriting
Harga                   154.933345
Name: 0, dtype: object
```

```
len(filtered_data)
```

```
1428
```

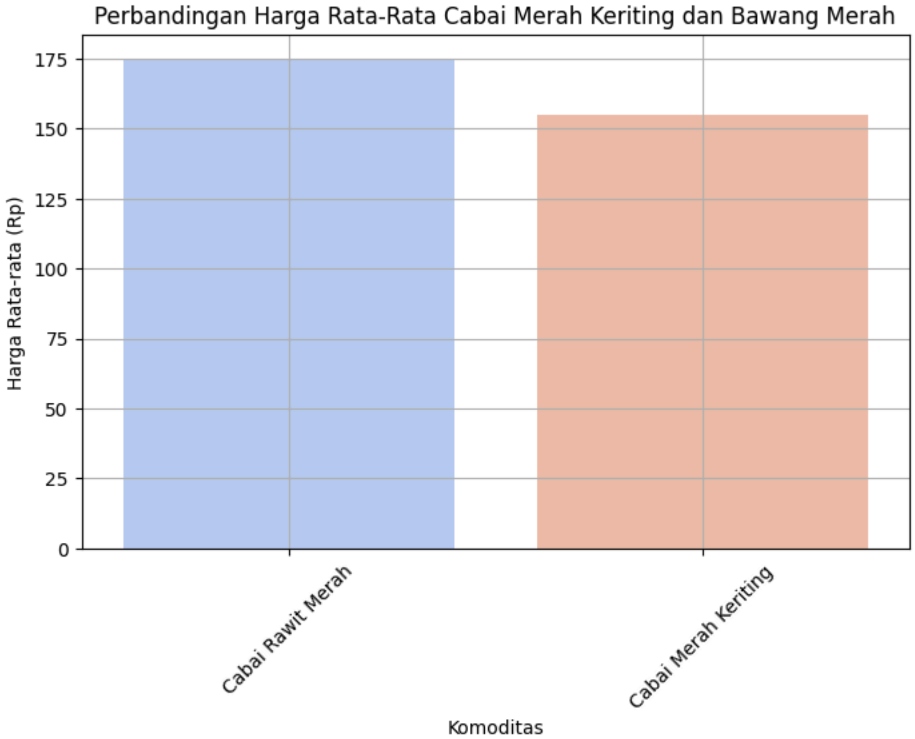
```
filtered_data.head(1000)
```

	Provinsi	Komoditas	Harga	Tanggal
3570	Aceh	Cabai Merah Keriting	42.353	2021-01-01
3571	Sumatera Utara	Cabai Merah Keriting	43.565	2021-01-01
3572	Sumatera Barat	Cabai Merah Keriting	48.343	2021-01-01
3573	Riau	Cabai Merah Keriting	54.869	2021-01-01
3574	Jambi	Cabai Merah Keriting	48.969	2021-01-01
...	...	...	...	...
4565	Kepulauan Riau	Cabai Rawit Merah	59.526	2021-09-01
4566	DKI Jakarta	Cabai Rawit Merah	32.598	2021-09-01
4567	Jawa Barat	Cabai Rawit Merah	26.015	2021-09-01
4568	Jawa Tengah	Cabai Rawit Merah	18.696	2021-09-01
4569	DI Yogyakarta	Cabai Rawit Merah	14.982	2021-09-01

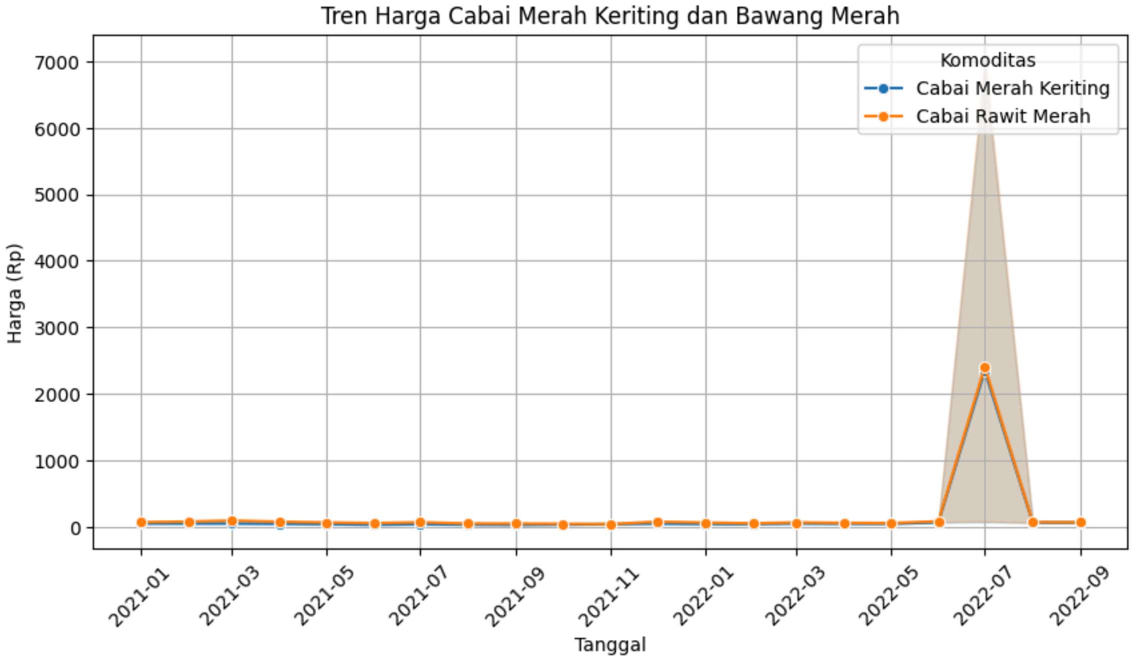
Next steps: Generate code with `filtered_data` View recommended plots New interactive sheet

```
# 2. Visualisasi Perbandingan Rata-Rata Harga Komoditas
plt.figure(figsize=(8, 5))
sns.barplot(data=average_prices.sort_values('Harga', ascending=False), x='Komoditas', y='Harga', palette='coolwarm')
plt.title('Perbandingan Harga Rata-Rata Cabai Merah Keriting dan Bawang Merah')
plt.xlabel('Komoditas')
plt.ylabel('Harga Rata-rata (Rp)')
plt.xticks(rotation=45)
plt.grid()
plt.show()

# 3. Visualisasi Tren Harga untuk Komoditas Terpilih
plt.figure(figsize=(10, 5))
sns.lineplot(data=filtered_data, x='Tanggal', y='Harga', hue='Komoditas', marker='o')
plt.title('Tren Harga Cabai Merah Keriting dan Bawang Merah')
plt.xlabel('Tanggal')
plt.ylabel('Harga (Rp)')
plt.xticks(rotation=45)
plt.grid()
plt.show()
```

```
<ipython-input-128-ccd9b348fa43>:3: FutureWarning:


Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and se


/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pas

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pas
```

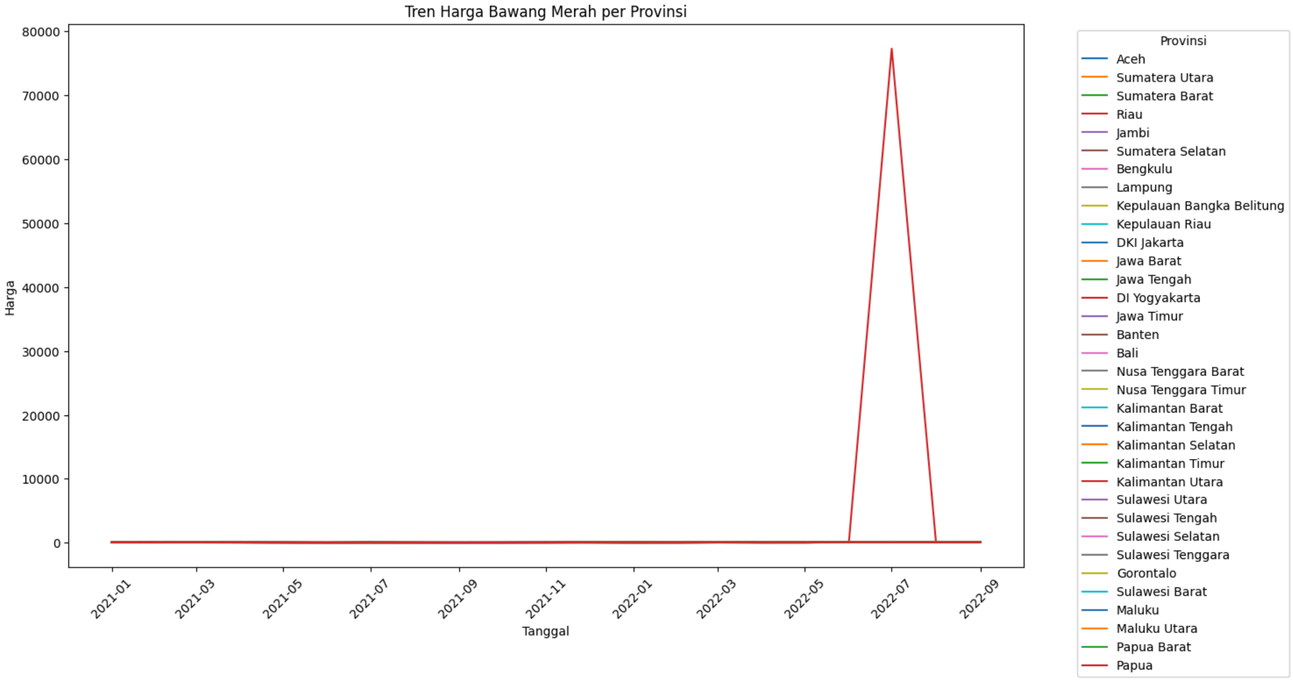


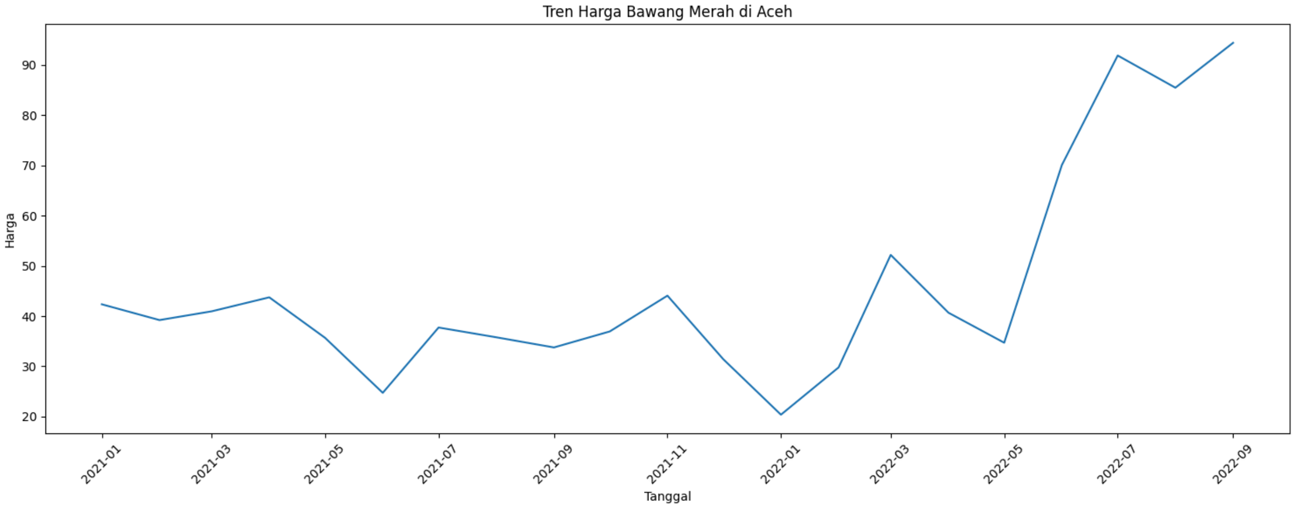
```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pas

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pas
```

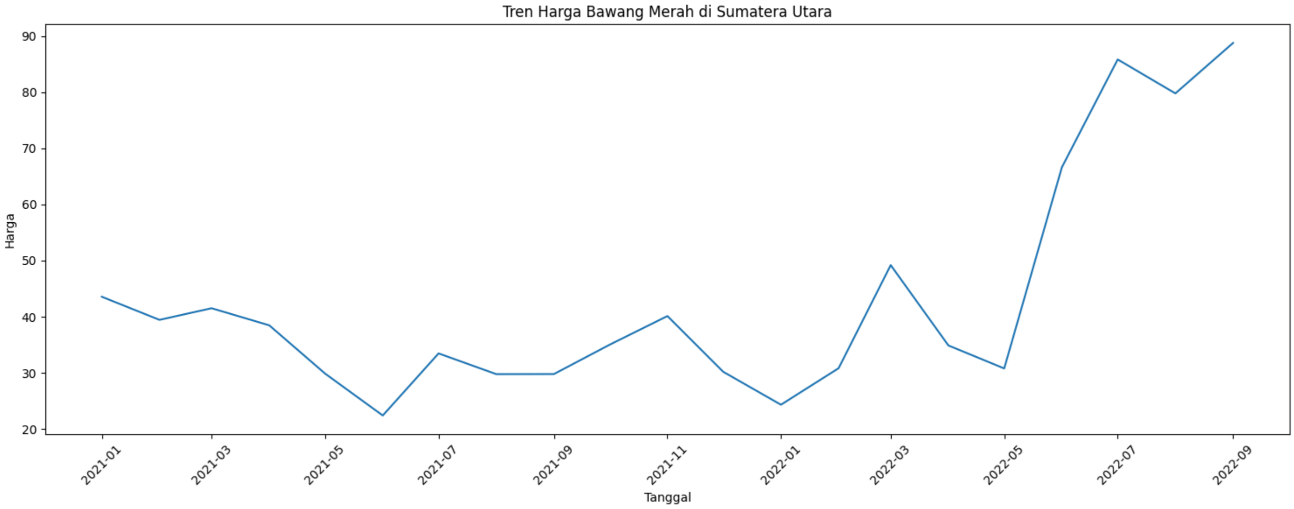


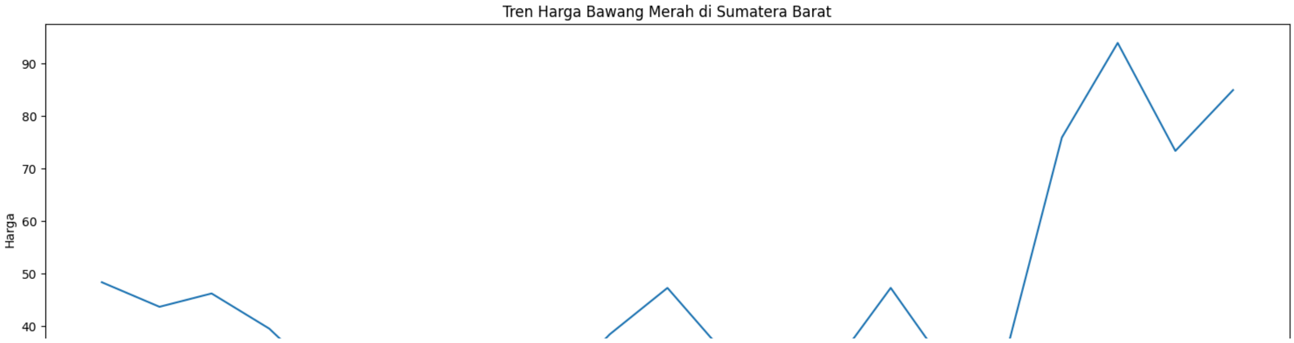
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

filtered_data = filtered_data.copy()

# Konversi kolom Tanggal ke tipe datetime
filtered_data['Tanggal'] = pd.to_datetime(filtered_data['Tanggal'])

# Cek dan tangani duplikat
duplicates = filtered_data.duplicated(subset=['Tanggal', 'Provinsi'], keep=False)
if duplicates.any():
    print(f"Ditemukan {duplicates.sum()} entri duplikat.")
    filtered_data = filtered_data.drop_duplicates(subset=['Tanggal', 'Provinsi'])

# Membuat daftar unik provinsi
provinces = filtered_data['Provinsi'].unique()

# Plotting untuk masing-masing provinsi
plt.figure(figsize=(15, 8))
for province in provinces:
    province_data = filtered_data[filtered_data['Provinsi'] == province]
    plt.plot(province_data['Tanggal'], province_data['Harga'], label=province)

plt.title('Tren Harga Bawang Merah per Provinsi')
plt.xlabel('Tanggal')
plt.ylabel('Harga')
plt.xticks(rotation=45)
plt.legend(title='Provinsi', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()

# Visualisasi terpisah untuk setiap provinsi
for province in provinces:
    plt.figure(figsize=(15, 6))
    province_data = filtered_data[filtered_data['Provinsi'] == province]
    sns.lineplot(x='Tanggal', y='Harga', data=province_data)
    plt.title(f'Tren Harga Bawang Merah di {province}')
    plt.xlabel('Tanggal')
    plt.ylabel('Harga')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

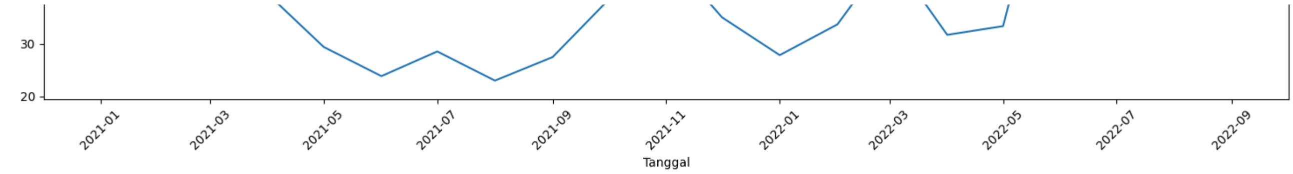
```
Ditemukan 1428 entri duplikat.
```



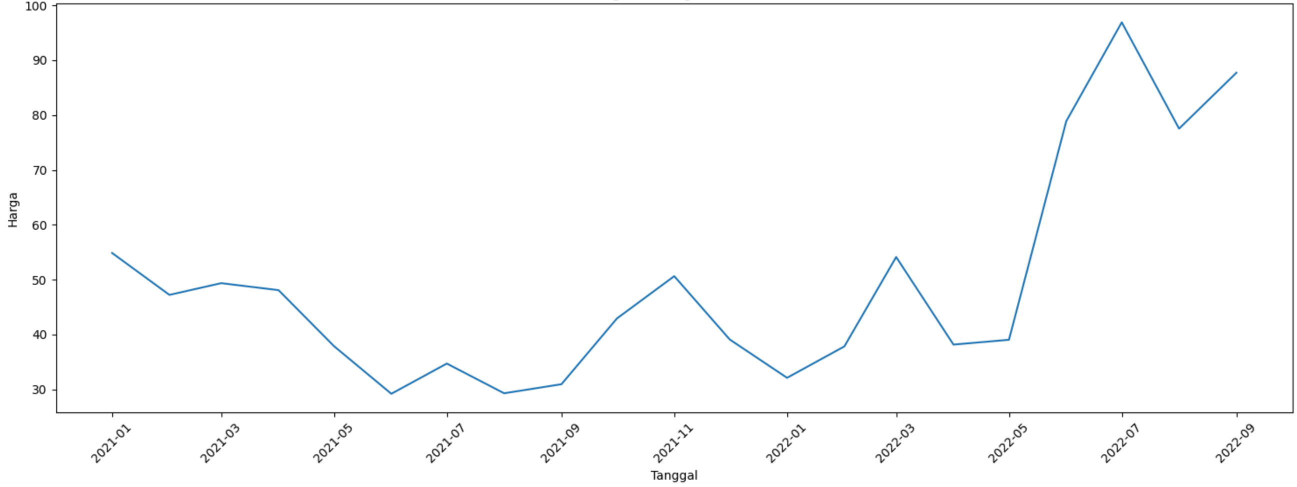




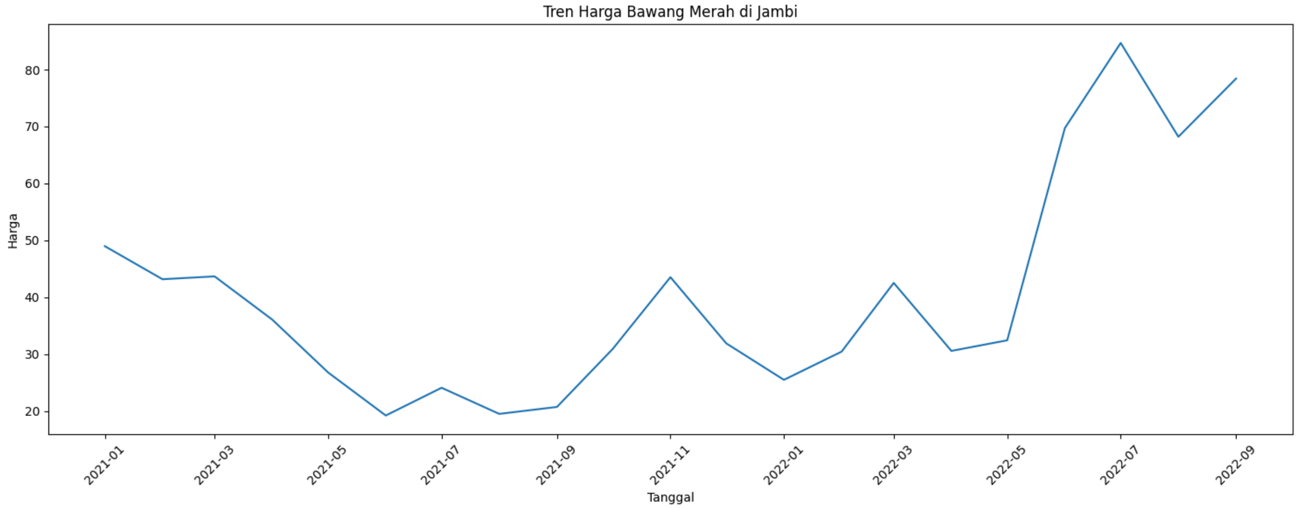




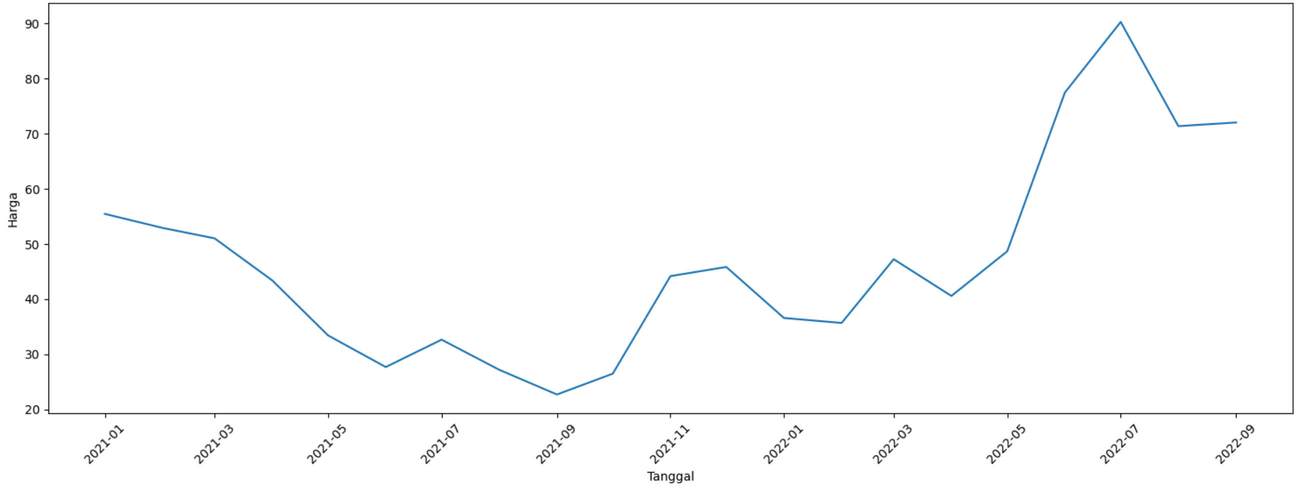
Tren Harga Bawang Merah di Riau



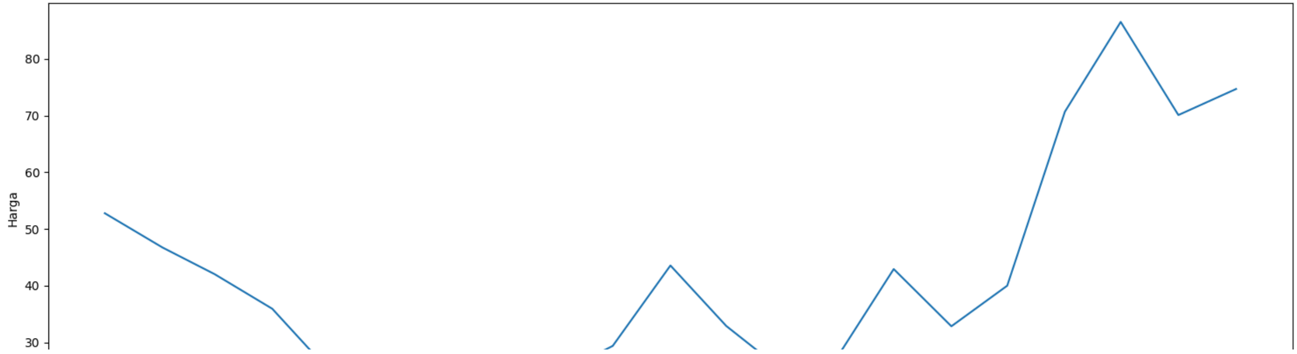
Tren Harga Bawang Merah di Jambi



Tren Harga Bawang Merah di Sumatera Selatan



Tren Harga Bawang Merah di Bengkulu


Harga