1. Import Fungsi

```
import numpy as np
import pandas as pd
import io
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d, CubicSpline
```

2. Membaca Data

```
data = pd.read_csv('/content/laporan_data_bersih.csv', encoding='latin-1')
x = data['KEDALAMAN (KM)'].values  # Variabel X: Kedalaman
y = data['MAGNITUDO (M)'].values    # Variabel Y: Magnitudo
```

```
#data tabel
print("=== Data Kedalaman (X) dan Magnitudo (Y) ===")
display(data[['KEDALAMAN (KM)', 'MAGNITUDO (M)']].head(10))

#data array
print("\n=== Array Kedalaman (X) ===")
print(x)
print("\n=== Array Magnitudo (Y) ===")
print(y)
```

```
=== Data Kedalaman (X) dan Magnitudo (Y) ===
```

	KEDALAMAN (KM)	MAGNITUDO (M)
0	27.0	4.47
1	31.1	3.40
2	10.0	2.31
3	77.8	2.82
4	111.5	3.25
5	26.8	3.73
6	492.6	4.68
7	18.8	1.64
8	10.0	3.58
9	24.4	3.05

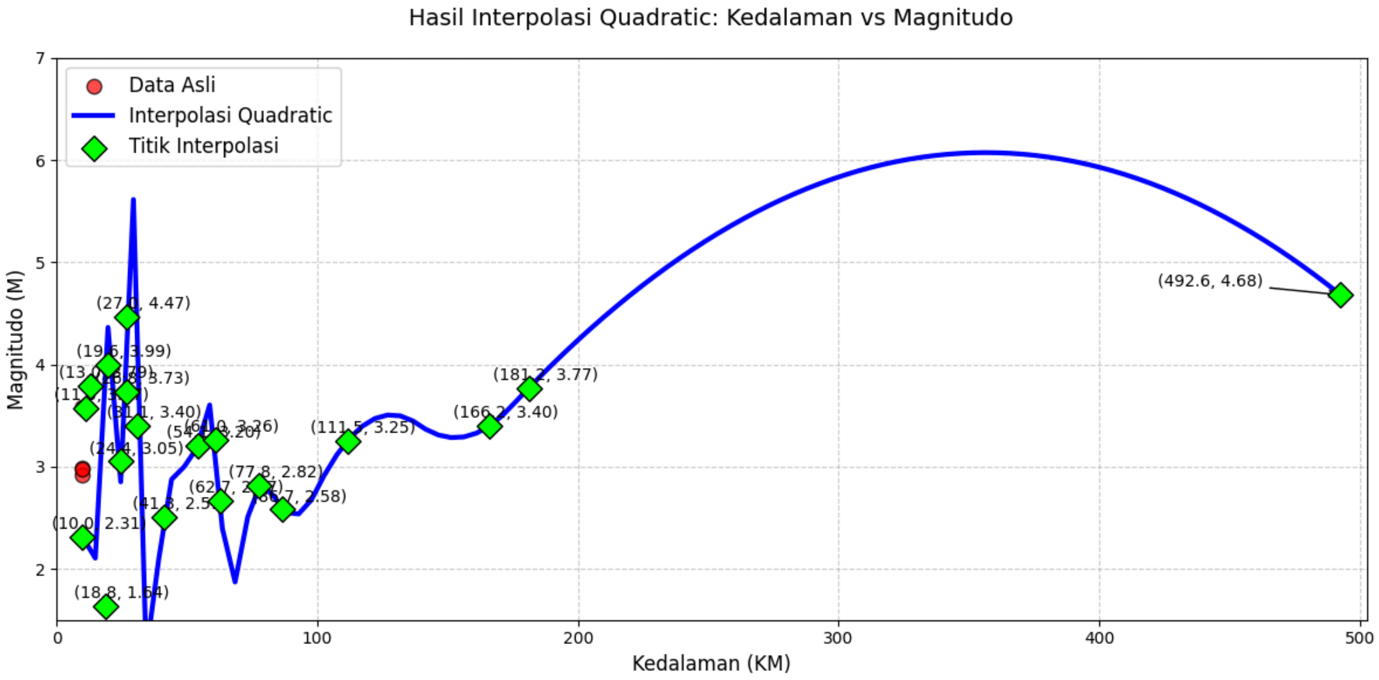
```
=== Array Kedalaman (X) ===
[ 27.   31.1  10.   77.8 111.5  26.8 492.6  18.8  10.   24.4  62.7  19.6
 166.2  41.3  11.   10.   13.   10.   54.1 181.2  10.   61.   86.7]

=== Array Magnitudo (Y) ===
[4.47 3.4  2.31 2.82 3.25 3.73 4.68 1.64 3.58 3.05 2.67 3.99 3.4  2.51
 3.57 2.99 3.79 2.92 3.2  3.77 2.98 3.26 2.58]
```

```
from collections import Counter

# Hitung frekuensi nilai x
x_counts = Counter(x)
print("Kedalaman yang duplikat:")
for depth, count in x_counts.items():
    if count > 1:
        print(f"- {depth} KM: {count} kali")
```

```
Kedalaman yang duplikat:
- 10.0 KM: 5 kali
```

3. Interpolasi Polinomial Quadratic

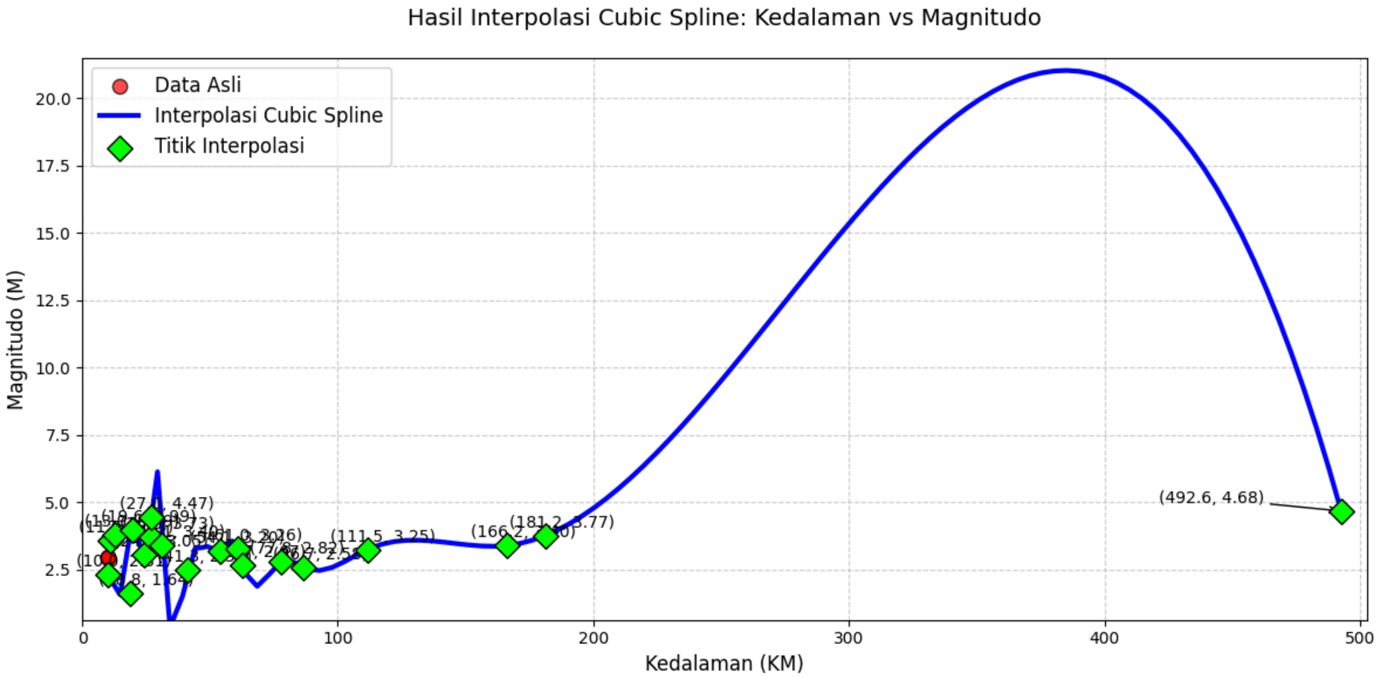
```
# ========================
# INTERPOLASI QUADRATIC
# ========================
sorted_indices = np.argsort(x)
x_sorted = x[sorted_indices]
y_sorted = y[sorted_indices]

x_unique, unique_indices = np.unique(x_sorted, return_index=True)
y_unique = y_sorted[unique_indices]

quad_interp = interp1d(x_unique, y_unique, kind='quadratic', fill_value='extrapolate')

x_quad = np.linspace(min(x_unique), max(x_unique), 100)
y_quad = quad_interp(x_quad)


# ========================
# VISUALISASI INTERPOLASI QUADRATIC
# ========================
plt.figure(figsize=(12, 6))

# Plot data asli dengan batasan
plt.scatter(x, y, color='red', s=80, label='Data Asli',
            edgecolors='black', alpha=0.7, zorder=3)

# Plot hasil interpolasi
valid_mask = (x_quad >= min(x_unique)) & (x_quad <= max(x_unique))
plt.plot(x_quad[valid_mask], y_quad[valid_mask], 'b-', linewidth=3,
         label='Interpolasi Quadratic', zorder=2)

# Titik interpolasi
plt.scatter(x_unique, y_unique, color='lime', s=120,
            label='Titik Interpolasi', edgecolors='black',
            marker='D', zorder=4)

# Formatting plot
ax = plt.gca()
ax.set_xlim(min(x_unique)-10, max(x_unique)+10)
ax.set_ylim(1.5, 7.0)

# Garis kotak
for spine in ax.spines.values():
    spine.set_visible(True)

# Grid dan judul
plt.grid(True, linestyle='--', alpha=0.6)
plt.title('Hasil Interpolasi Quadratic: Kedalaman vs Magnitudo',
          fontsize=14, pad=20)
plt.xlabel('Kedalaman (KM)', fontsize=12)
plt.ylabel('Magnitudo (M)', fontsize=12)

# Anotasi
for i, (xi, yi) in enumerate(zip(x_unique, y_unique)):
    if xi > ax.get_xlim()[1]*0.8:
        xytext = (-80, 5)
    else:
        xytext = (10, 5)
    plt.annotate(f'({xi:.1f}, {yi:.2f})',
                (xi, yi),
                textcoords="offset points",
                xytext=xytext,
                ha='center',
                arrowprops=dict(arrowstyle='->', lw=1))

plt.legend(fontsize=12, loc='best')
plt.tight_layout()
plt.show()
```



4. Interpolasi Cubic Spline

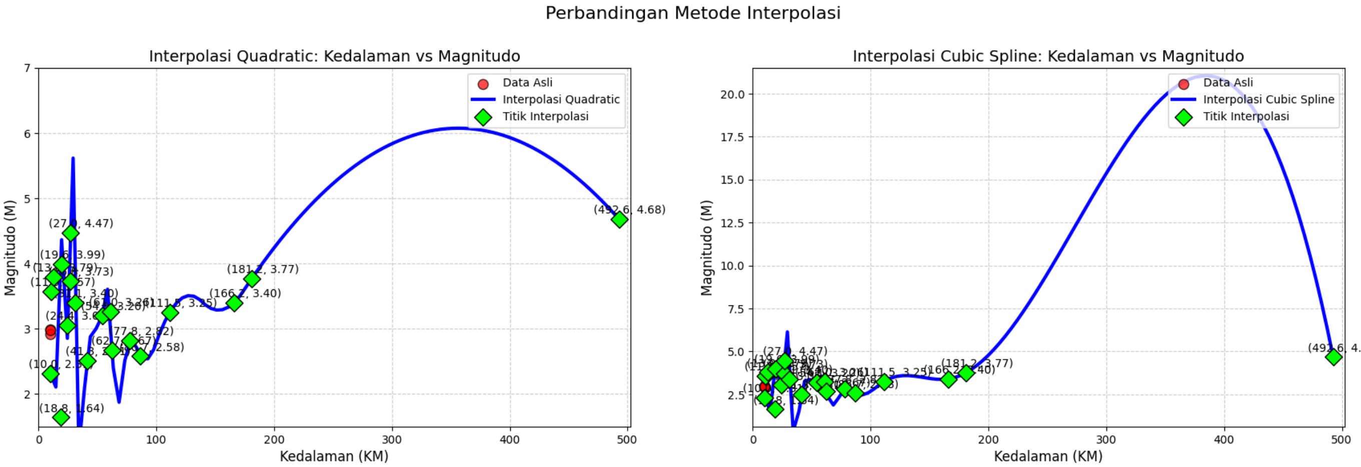
```
# ================================
# INTERPOLASI CUBIC SPLINE
# ================================
sorted_indices = np.argsort(x)
x_sorted = x[sorted_indices]
y_sorted = y[sorted_indices]

x_unique, unique_indices = np.unique(x_sorted, return_index=True)
y_unique = y_sorted[unique_indices]

cs = CubicSpline(x_unique, y_unique)
x_spline = np.linspace(min(x_unique), max(x_unique), 100)
y_spline = cs(x_spline)


# ====================================
# VISUALISASI INTERPOLASI CUBIC SPLINE
# ====================================
plt.figure(figsize=(12, 6))

# Plot data asli dengan batasan
plt.scatter(x, y, color='red', s=80, label='Data Asli',
            edgecolors='black', alpha=0.7, zorder=3)

# Plot hasil interpolasi cubic spline
valid_mask = (x_spline >= min(x)) & (x_spline <= max(x))  # Filter data spline agar tidak keluar batas
plt.plot(x_spline[valid_mask], y_spline[valid_mask], 'b-', linewidth=3,
         label='Interpolasi Cubic Spline', zorder=2)

# Titik interpolasi dengan batasan
plt.scatter(x_unique, y_unique, color='lime', s=120,
            label='Titik Interpolasi', edgecolors='black',
            marker='D', zorder=4)

# Formatting plot dengan batas yang ketat
ax = plt.gca()
ax.set_xlim(min(x)-10, max(x)+10)  # Padding 10 unit untuk kedalaman
ax.set_ylim(min(y)-1.0, 21.5)  # Padding dan batas atas 7.0 untuk Magnitudo

# Garis kotak
```

```
for spine in ax.spines.values():
    spine.set_visible(True)

# Grid dan judul
plt.grid(True, linestyle='--', alpha=0.6)
plt.title('Hasil Interpolasi Cubic Spline: Kedalaman vs Magnitudo',
          fontsize=14, pad=20)
plt.xlabel('Kedalaman (KM)', fontsize=12)
plt.ylabel('Magnitudo (M)', fontsize=12)

# Anotasi titik penting
for i, (xi, yi) in enumerate(zip(x_unique, y_unique)):
    if xi > ax.get_xlim()[1]*0.8:
        xytext = (-80, 5)  # Anotasi ke kiri
    else:
        xytext = (10, 5)   # Anotasi ke kanan

    plt.annotate(f'({xi:.1f}, {yi:.2f})',
                (xi, yi),
                textcoords="offset points",
                xytext=xytext,
                ha='center',
                arrowprops=dict(arrowstyle='->', lw=1))

plt.legend(fontsize=12, loc='best')
plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np

# ============================================
# VISUALISASI PERBANDINGAN DUA METODE INTERPOLASI
# ============================================
plt.figure(figsize=(18, 6))

# --------------------------------------------
# Subplot 1: Interpolasi Quadratic
# --------------------------------------------
plt.subplot(1, 2, 1)  # 1 baris, 2 kolom, subplot pertama

# Plot data asli
plt.scatter(x, y, color='red', s=80, label='Data Asli',
            edgecolors='black', alpha=0.7, zorder=3)
```

```
# Plot hasil interpolasi quadratic
valid_mask_quad = (x_quad >= min(x_unique)) & (x_quad <= max(x_unique))
plt.plot(x_quad[valid_mask_quad], y_quad[valid_mask_quad], 'b-', linewidth=3,
         label='Interpolasi Quadratic', zorder=2)

# Titik interpolasi
plt.scatter(x_unique, y_unique, color='lime', s=120,
            label='Titik Interpolasi', edgecolors='black',
            marker='D', zorder=4)

# Formatting
ax1 = plt.gca()
ax1.set_xlim(min(x_unique)-10, max(x_unique)+10)
ax1.set_ylim(1.5, 7.0)
plt.grid(True, linestyle='--', alpha=0.6)
plt.title('Interpolasi Quadratic: Kedalaman vs Magnitudo', fontsize=14)
plt.xlabel('Kedalaman (KM)', fontsize=12)
plt.ylabel('Magnitudo (M)', fontsize=12)

# Anotasi
for xi, yi in zip(x_unique, y_unique):
    plt.annotate(f'({xi:.1f}, {yi:.2f})', (xi, yi),
                textcoords="offset points",
                xytext=(10, 5),
                ha='center',
                arrowprops=dict(arrowstyle='->', lw=1))

plt.legend(fontsize=10, loc='upper right')

# --------------------------------------------
# Subplot 2: Interpolasi Cubic Spline
# --------------------------------------------
plt.subplot(1, 2, 2)  # 1 baris, 2 kolom, subplot kedua

# Plot data asli
plt.scatter(x, y, color='red', s=80, label='Data Asli',
            edgecolors='black', alpha=0.7, zorder=3)

# Plot hasil interpolasi cubic spline
valid_mask_spline = (x_spline >= min(x)) & (x_spline <= max(x))
plt.plot(x_spline[valid_mask_spline], y_spline[valid_mask_spline], 'b-', linewidth=3,
         label='Interpolasi Cubic Spline', zorder=2)

# Titik interpolasi
plt.scatter(x_unique, y_unique, color='lime', s=120,
            label='Titik Interpolasi', edgecolors='black',
            marker='D', zorder=4)

# Formatting
ax2 = plt.gca()
ax2.set_xlim(min(x)-10, max(x)+10)
ax2.set_ylim(min(y)-1.0, 21.5)
plt.grid(True, linestyle='--', alpha=0.6)
plt.title('Interpolasi Cubic Spline: Kedalaman vs Magnitudo', fontsize=14)
plt.xlabel('Kedalaman (KM)', fontsize=12)
plt.ylabel('Magnitudo (M)', fontsize=12)

# Anotasi
for xi, yi in zip(x_unique, y_unique):
    plt.annotate(f'({xi:.1f}, {yi:.2f})', (xi, yi),
                textcoords="offset points",
                xytext=(10, 5),
                ha='center',
                arrowprops=dict(arrowstyle='->', lw=1))

plt.legend(fontsize=10, loc='upper right')

# --------------------------------------------
# Penyempurnaan Tampilan
# --------------------------------------------
plt.tight_layout(pad=3.0)  # Memberi jarak antar subplot
plt.suptitle('Perbandingan Metode Interpolasi', y=1.02, fontsize=16)  # Judul utama
plt.show()
```


Perbandingan Metode Interpolasi
Interpolasi Quadratic: Kedalaman vs Magnitudo
Magnitudo (M)
Kedalaman (KM)
Data Asli
Interpolasi Quadratic
Titik Interpolasi
7
6
5
4
3
2
0
100
200
300
400
500
(27.0, 4.47)
(19.6, 3.99)
(181.2, 3.77)
(166.2, 3.40)
(111.5, 3.25)
(77.8, 2.82)
(10.0, 2.3
(18.8, 1.64)
(492.6, 4.68)
Interpolasi Cubic Spline: Kedalaman vs Magnitudo
Magnitudo (M)
Kedalaman (KM)
Data Asli
Interpolasi Cubic Spline
Titik Interpolasi
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0
100
200
300
400
500
(27.0, 4.47)
(181.2, 3.77)
(111.5, 3.25)
(492.6, 4.68)