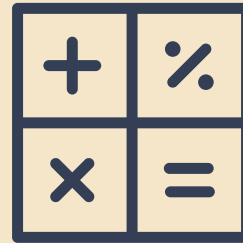


**PELUANG DAN STATISTIKA**

# **PROYEK 1A**

**➤ CHRISTOPER JONATHAN, DILIVIO C. L. TILAAAR, JUAN C. CHANDRA, PETRA W. LEKA, WESLEY HAKIM**



## PELUANG DAN STATISTIKA

# OVERVIEW

LATAR BELAKANG

TUJUAN

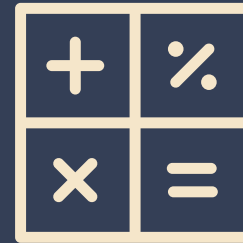
PERALATAN STATISTIKA

LANGKAH PEMROGRAMAN

KONTEN LAPORAN

KENDALA

KESIMPULAN DAN SARAN

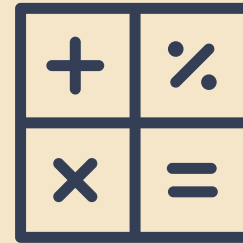


## PELUANG DAN STATISTIKA

# LATAR BELAKANG

Latar belakang dari permasalahan ini adalah sulitnya penampilan dan penafsiran data data penting yang diperlukan dalam penampilan data views, likes, dan dislikes dalam video-video youtube.

MATH1042

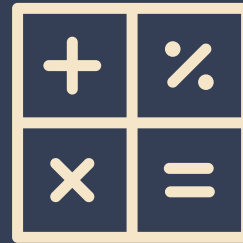


## PELUANG DAN STATISTIKA

# TUJUAN

Tujuan pengerjaan proyek ini adalah penyajian data yang deskriptif melalui diagram/ grafik

MATH1042



**PELUANG DAN STATISTIKA**

# PERALATAN STATISTIKA



**LIBRARY NUMPY**



**LIBRARY MATH**



**LIBRARY PANDAS**



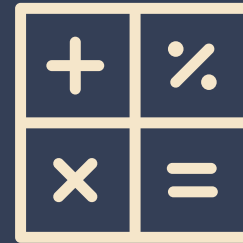
**LIBRARY SEABORN**



**LIBRARY MATPLOTLIB**



**LIBRARY SCIPY**

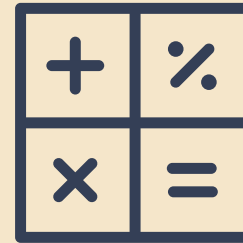


## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

### MEAN

```
1 # Menghitung rata-rata
2 df_num_parameter = df[["views", "likes", "dislikes"]]
3 rata_rata = df_num_parameter.mean()
4 print(f"Rata-rata views = {rata_rata[0]}\nRata-rata likes = {rata_rata[1]}\nRata-rata dislikes = {rata_rata[2]}")
```



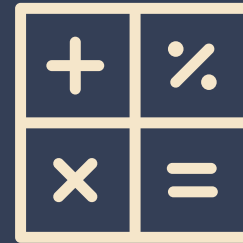
## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

## MEDIAN

```
1 # Mencari median
2 median = df_num_parameter.median()
3 print(f"Median views = {median[0]}\nMedian likes = {median[1]}\nMedian dislikes = {median[2]}")
```

MATH1042



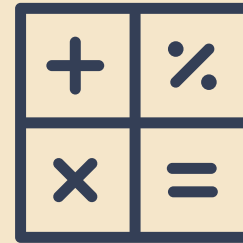
## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

## MODUS

```
1 # Menghitung modus
2 modus_views = df_num_parameter["views"].mode()
3 modus_likes = df_num_parameter["likes"].mode()
4 modus_dislikes = df_num_parameter["dislikes"].mode()
5 print(f"Modus views = {modus_views[0]}\nModus likes = {modus_likes[0]}\nModus dislikes = {modus_dislikes[0]}")
```



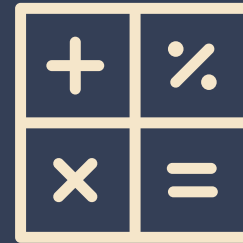


## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

## STANDAR DEVIASI

```
1 # Menghitung standar deviasi
2 std_deviasi = df_num_parameter.std()
3 print(f"Standar deviasi views = {std_deviasi[0]}\nStandar deviasi likes = {std_deviasi[1]}\n
4 Standar deviasi dislikes = {std_deviasi[2]}")
```

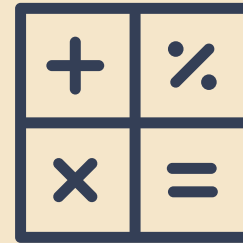


## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

### MODUS

```
1 # Menghitung kuartil ke-1 dari setiap kolom
2 q1 = df_num_parameter.quantile(q=0.25)
3
4 # Menghitung kuartil ke-2 (median) dari setiap kolom
5 q2 = df_num_parameter.quantile(q=0.5)
6
7 # Menghitung kuartil ke-3 dari setiap kolom
8 q3 = df_num_parameter.quantile(q=0.75)
9
10 print(f"Kuartil ke-1 dari views = {q1[0]}\nKuartil ke-2 dari views = {q2[0]}\nKuartil ke-3 dari views = {q3[0]}\n")
11 print(f"Kuartil ke-1 dari likes = {q1[1]}\nKuartil ke-2 dari likes = {q2[1]}\nKuartil ke-3 dari likes = {q3[1]}\n")
12 print(f"Kuartil ke-1 dari dislikes = {q1[2]}\nKuartil ke-2 dari dislikes = {q2[2]}\nKuartil ke-3 dari dislikes = {q3[2]}\n")
```

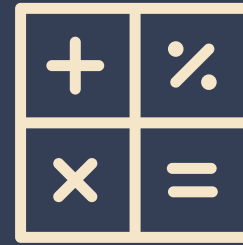


## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

## INTERKUARTIL

```
1 # Menghitung jangkauan inter kuartil
2 iqr = q3 - q1
3 print(f"Jangkauan inter kuartil dari views = {iqr[0]}\nJangkauan inter kuartil dari likes = {iqr[1]}\n
4 Jangkauan inter kuartil dari dislikes = {iqr[2]}")
```

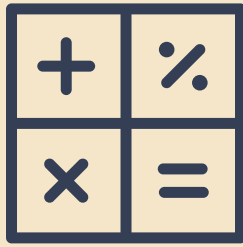


## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

### MODUS

```
1 # Mencari pencilan
2 batas_bawah = q1 - 1.5*iqr
3 batas_atas = q3 + 1.5*iqr
4
5 pencilan_bawah = df_num_parameter[(df_num_parameter < batas_bawah)]
6 pencilan_atas = df_num_parameter[(df_num_parameter > batas_atas)]
7
8 # List yang berisi data pencilan dari views
9 pencilan_bawah_views = pencilan_bawah["views"].dropna().to_list()
10 pencilan_atas_views = pencilan_atas["views"].dropna().to_list()
11
12 # List yang berisi data pencilan dari likes
13 pencilan_bawah_likes = pencilan_bawah["likes"].dropna().to_list()
14 pencilan_atas_likes = pencilan_atas["likes"].dropna().to_list()
15
16 # List yang berisi data pencilan dari dislikes
17 pencilan_bawah_dislikes = pencilan_bawah["dislikes"].dropna().to_list()
18 pencilan_atas_dislikes = pencilan_atas["dislikes"].dropna().to_list()
19
20 print(f"Jumlah data pencilan dari views adalah {len(pencilan_bawah_views)+len(pencilan_atas_views)} data.")
21 print(f"Jumlah data pencilan dari likes adalah {len(pencilan_bawah_likes)+len(pencilan_atas_likes)} data.")
22 print(f"Jumlah data pencilan dari dislikes adalah {len(pencilan_bawah_dislikes)+len(pencilan_atas_dislikes)} data.")
```

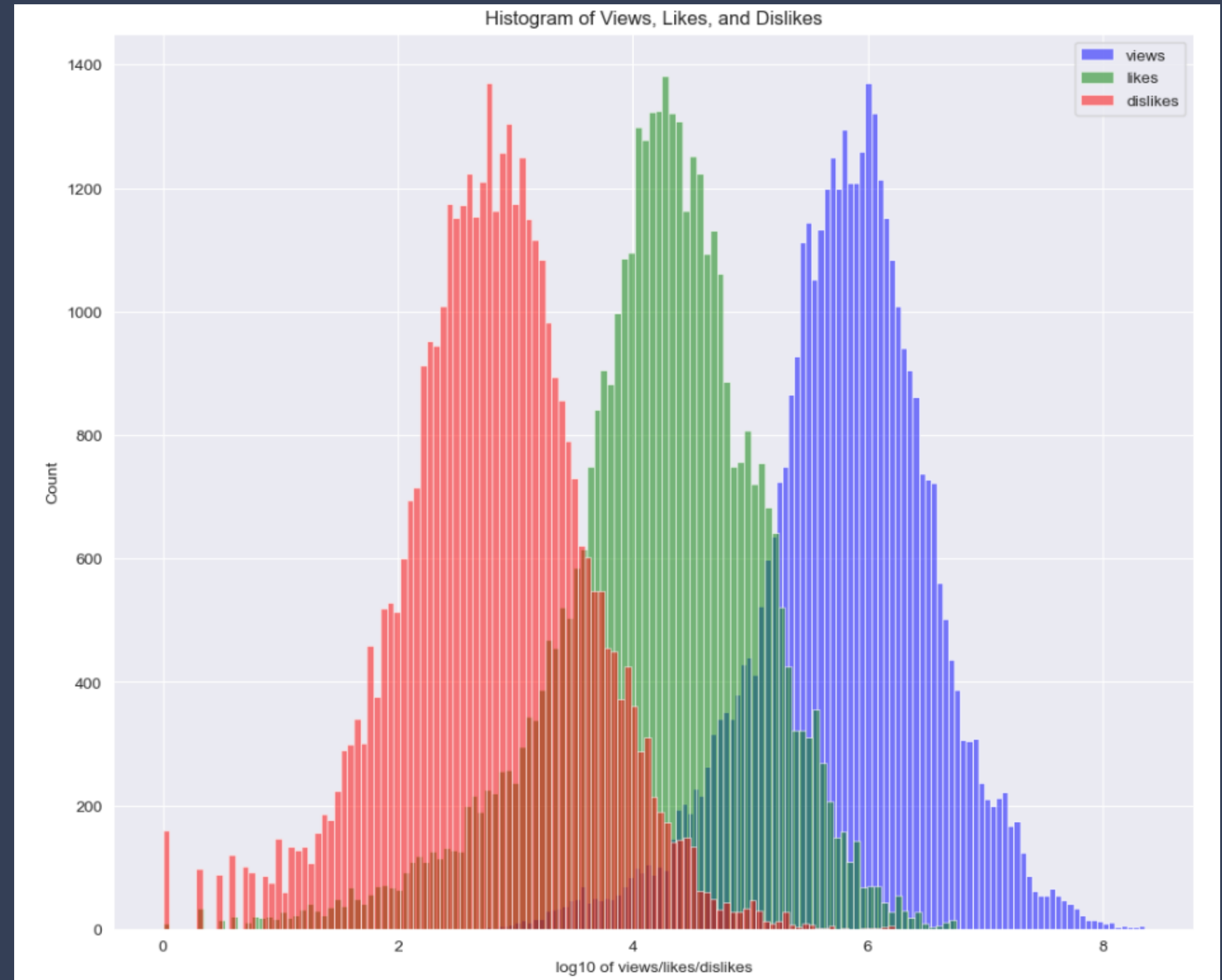


PELUANG DAN STATISTIKA

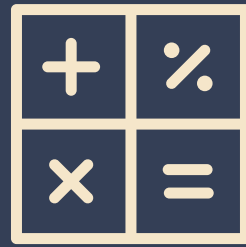
# LANGKAH PEMROGRAMAN

## HISTOGRAM

MATH1042



#Views #likes #dislikes

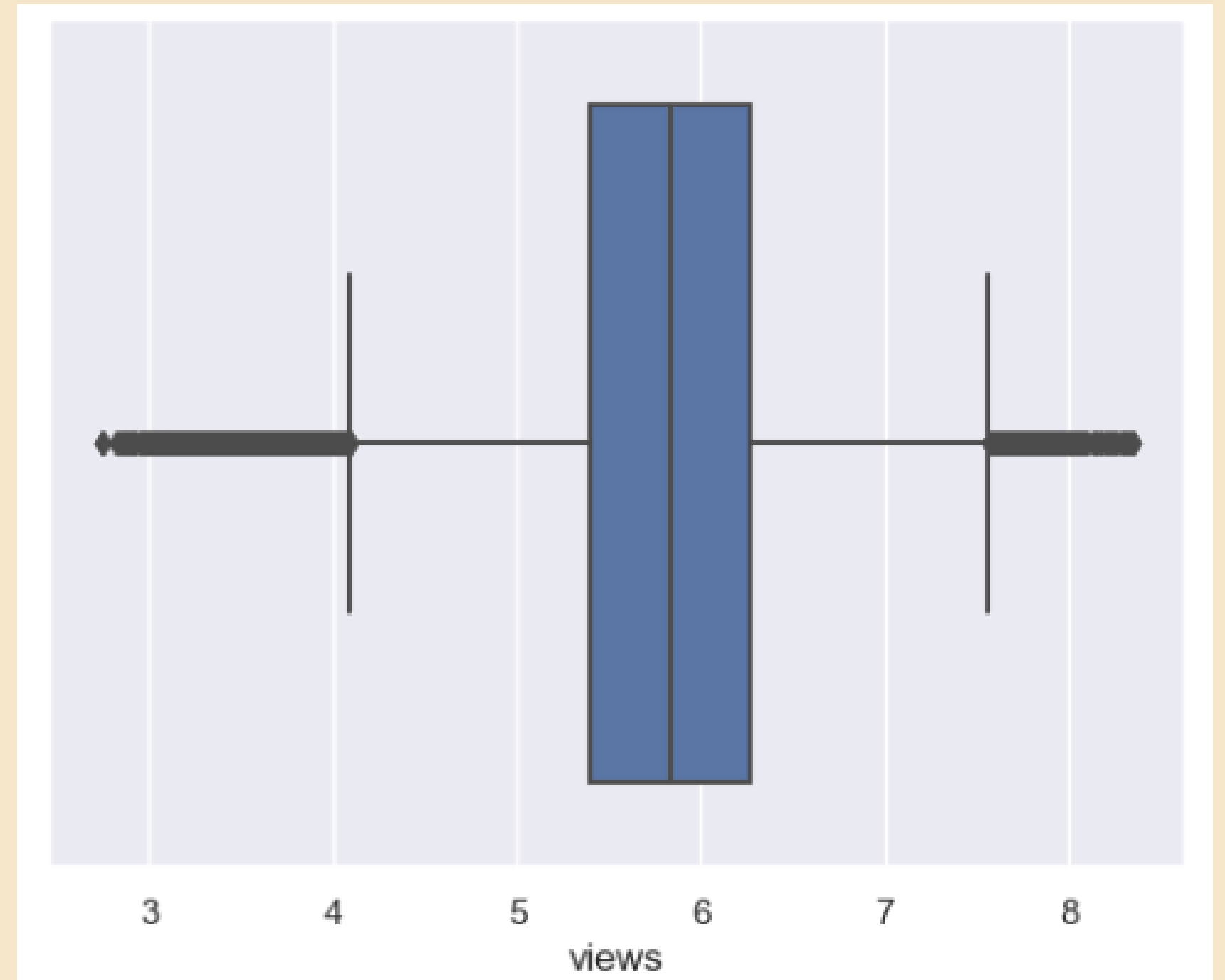


PELUANG DAN STATISTIKA

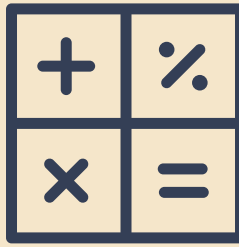
# LANGKAH PEMROGRAMAN

BOXPLOT #VIEWS

MATH1042



#Views #likes #dislikes

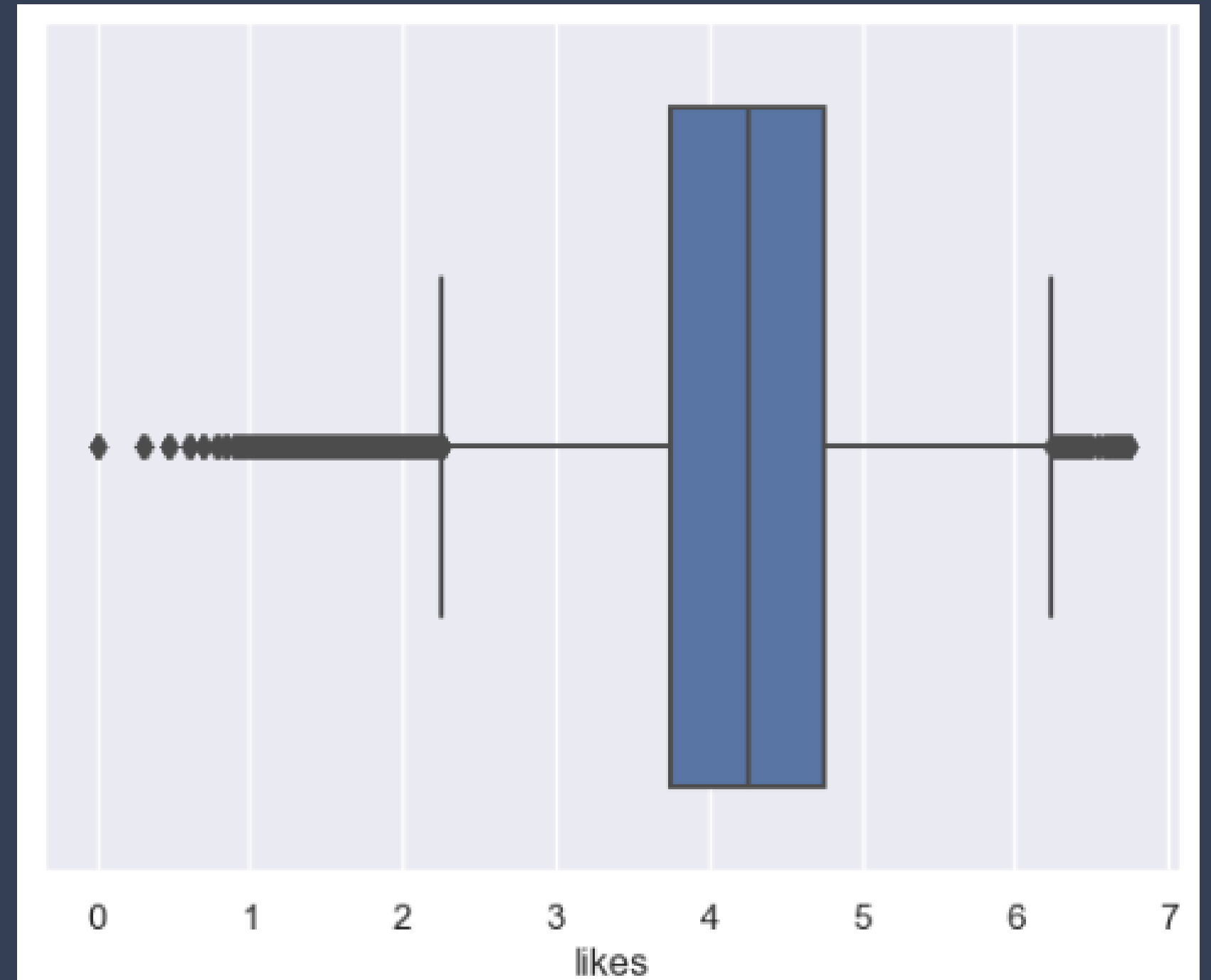


PELUANG DAN STATISTIKA

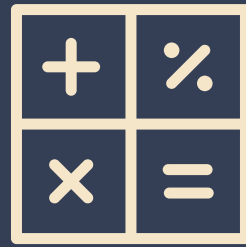
# LANGKAH PEMROGRAMAN

BOXPLOT#LIKES

MATH1042



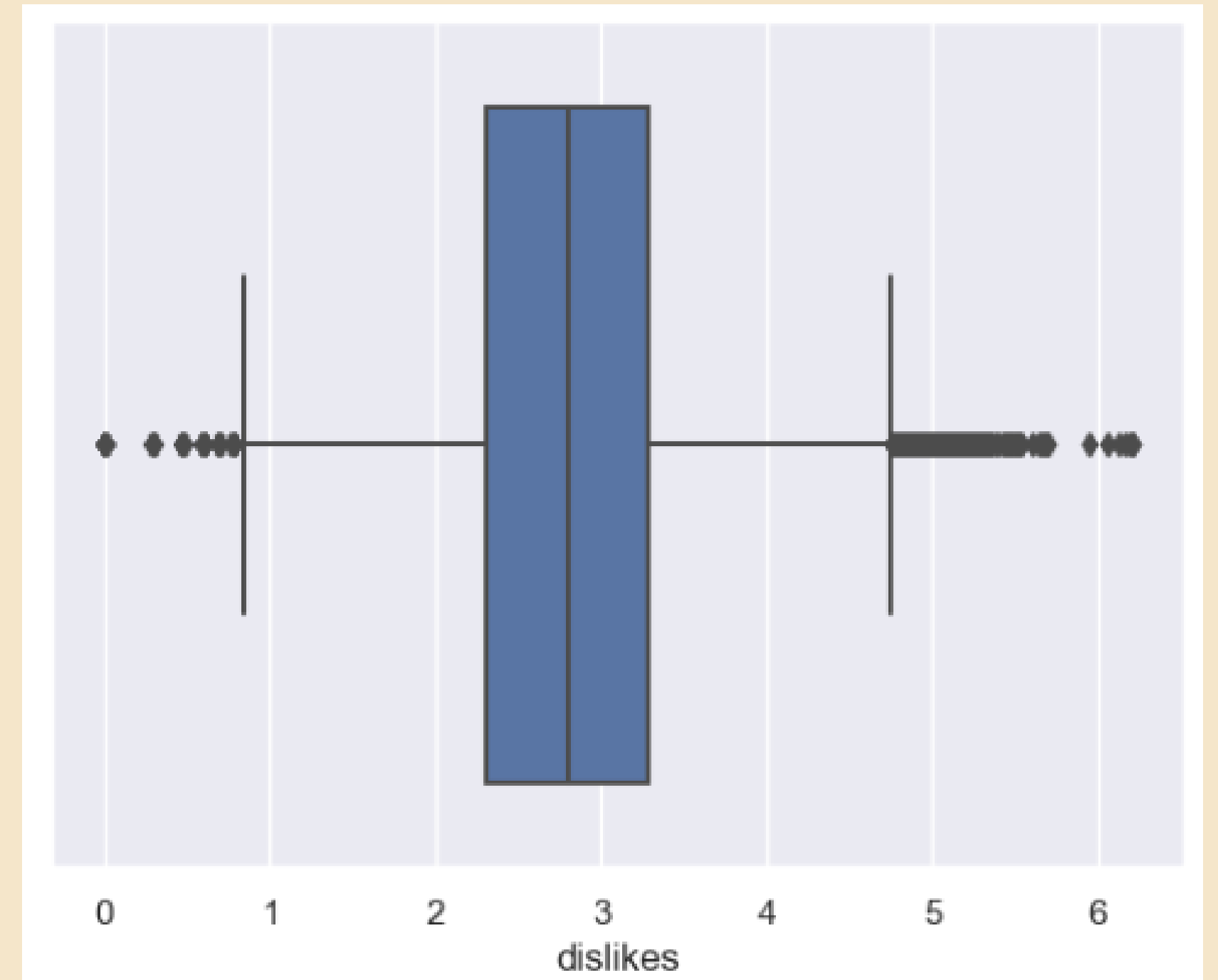
#Views #likes #dislikes



PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

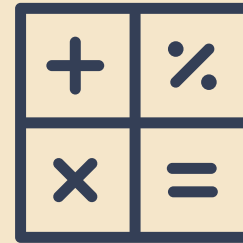
BOXPLOT #DISLIKE



MATH1042

#Views #likes #dislikes



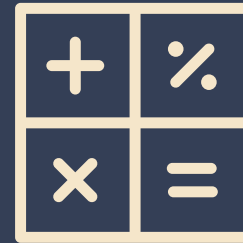


## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

### MEAN VIEWS PERCATEGORY ID

```
1 df_category_id = df[['category_id', 'views']]
2 category_id = df_category_id.groupby('category_id')
3 category_id_mean = category_id.mean()
4 category_id_mean
```

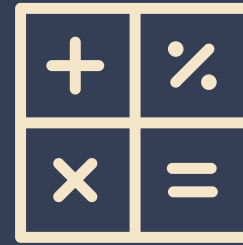


## PELUANG DAN STATISTIKA

# LANGKAH PEMROGRAMAN

### PLOTTING PERSEBARAN DATA

```
1 from matplotlib import pyplot as plt
2 import seaborn as sns
3
4 category_id_list = df['category_id'].to_list()
5 id_list = list(set(category_id_list))
6
7 fig, axs = plt.subplots(len(id_list), 2, figsize=(12, 5*len(id_list)), constrained_layout=True)
8 axs = axs.flatten()
9
10 for i in range(len(id_list)):
11     data = category_id.get_group(id_list[i])
12
13     sns.set_theme()
14     sns.set_style("white")
15
16     sns.histplot(data=data, x=np.log10(data['views']), color='pink', ax=axs[2*i])
17     axs[2*i].set_title(f"Histogram untuk Category id = {id_list[i]}")
18
19     sns.boxplot(data=data, x=np.log10(data['views']), color='pink', ax=axs[2*i+1])
20     axs[2*i+1].set_title(f"Box plot untuk Category id = {id_list[i]}")
21
22 plt.show()
23
```

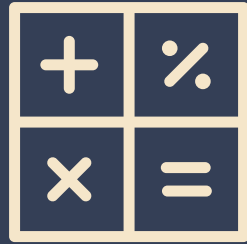


## PELUANG DAN STATISTIKA

# KONTEN LAPORAN

### ➤ HIGHLIGHT

- Lebih banyak orang yang menonton tanpa berinteraksi dengan videonya
- Orang-orang lebih cenderung berinteraksi saat dia menyukai video tersebut dibanding ketika tidak menyukai
- Data paling tersebar secara berurut adalah: views, likes, dislikes
- Pada kategori 10, 29, 1 terlihat memiliki rata-rata penonton yang lebih banyak sehingga membuat video kategori ini lebih mungkin ditonton



# KONTEN LAPORAN

## ➤ 2A. SEBERAPA TERPENCIL?

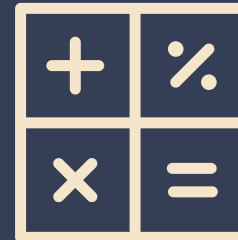
sangat terpencil dengan peluang sebesar  
0.0009906139329849675

## ➤ 2B. BATAS LIKE

memiliki jumlah like sebesar  
781575.5689778763

## ➤ 2C. IDENTIFIKASI VIDEO BAIK

Video yang memiliki like lebih dari 2B, dan memiliki rasio like dislike sebesar 75:1

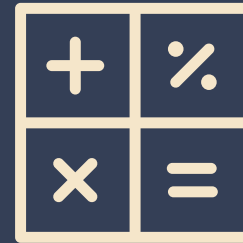


## PELUANG DAN STATISTIKA

# KONTEN LAPORAN

views	
category_id	
10	6.210307e+06
29	3.167400e+06
1	3.112240e+06
20	2.647433e+06
24	2.073128e+06
17	2.069401e+06
22	1.529018e+06
23	1.483936e+06

28	1.382127e+06
2	1.373286e+06
26	9.849565e+05
43	9.035273e+05
19	8.546196e+05
15	8.347004e+05
27	7.187950e+05
25	6.008645e+05

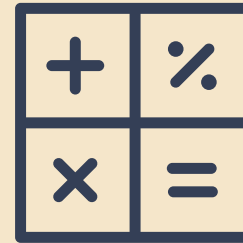


## PELUANG DAN STATISTIKA

# KONTEN LAPORAN

## 2A. SEBERAPA TERPENCIL

```
1 from scipy.stats import norm
2 import math
3 mu, sigma = -10.06, 4.02
4
5 v = (10**8 - rata_rata[0])/std_deviasi[0]
6 #print(rata_rata[0],std_deviasi[0], v)
7 print((1-norm.cdf(v, 0, 1))*100,)
8 count = (df['views'] > 10**8).sum()
9 c2 = ( df['views']>= 0 ).sum()
10 # Print the result
11 print(f'Number of videos with views above 100 million: {count} dan total videos {c2} probability {count/c2}')#0.000006 = 0
```

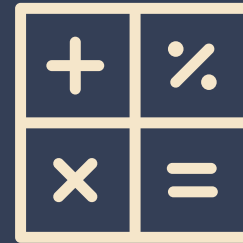


## PELUANG DAN STATISTIKA

# KONTEN LAPORAN

## 2B. BATAS LIKE

```
1  
2 print(f'Untuk bisa video like top 0.1% dibutuhkan like sebanyak {norm.ppf(0.999,0,1)*std_deviasi[1]+rata_rata[1]}')  
3 '''tes = (781575.5689778763-rata_rata[1])/std_deviasi[1]  
4 print(norm.cdf(tes,0,1))'''|
```



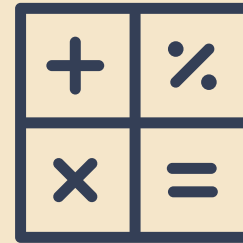
## PELUANG DAN STATISTIKA

# KONTEN LAPORAN

## 2C. IDENTIFIKASI VIDEO BAIK

```
1 #syarat kualitas baik video favorit sekaligus perbandingan antara likes/dislikes = 0.7
2 syt = ( df['likes'] >= 781575.5689778763 ).sum()
3 print(syt)
4 df['like_dislike_ratio'] = df.apply(lambda row: row['likes'] / row['dislikes'] if row['likes'] >= 781575.5689778763
5                                     else np.nan, axis=1)
6 # jumlah data yang perbandingan likes dan dislikes-nya kurang dari 0.75
7 jumlah_kurang_dari = len(df.loc[df['like_dislike_ratio'] < 75])
8 # jumlah data yang perbandingan likes dan dislikes-nya lebih dari 0.75
9 jumlah_lebih_dari = len(df.loc[df['like_dislike_ratio'] > 75])
10 print(jumlah_kurang_dari, jumlah_lebih_dari)
```

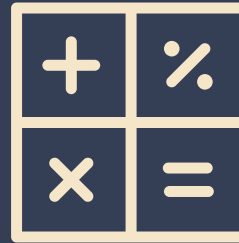




## PELUANG DAN STATISTIKA

# KENDALA

- Mengatur jadwal untuk memiliki waktu berkumpul dan berdiskusi bersama.
- Data yang terlalu besar, sehingga histogram tidak dapat menampilkan data dengan baik



## PELUANG DAN STATISTIKA

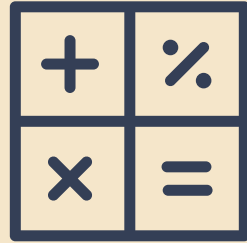
# KESIMPULAN DAN SARAN

### ➤ KESIMPULAN

saat kita membuat video berada dalam kategori 10, 29, dan 1 video kita menjadi lebih mungkin ditonton

### ➤ SARAN

Buatlah video dengan kategori 10, 29, dan 1



**PELUANG DAN STATISTIKA**

---

**THANK YOU**

**➤ CHRISTOPER JONATHAN, DILIVIO C. L. TILAAAR, JUAN C. CHANDRA, PETRA W. LEKA, WESLEY HAKIM**