

Jawaban Soal Non-Dataset:

1. Hawkins (1980) dalam Ben-Gal (2005) mendefinisikan bahwa outlier adalah hasil pengamatan yang sangat berbeda dari pengamatan yang lain, yang memunculkan kecurigaan bahwa hasil tersebut diambil dengan mekanisme yang berbeda. Secara sederhana outlier adalah data yang memiliki nilai sangat berbeda dari yang lain. Misalnya, terdapat *array* yang berisi 100 angka yang berkisar pada nilai 1-10. Kemudian, salah satu data memiliki nilai 450. Maka, data yang nilainya jauh dari yang lain itu dinamakan outlier.

Dalam analisis data, outlier dapat mempengaruhi sebaran data. Outlier menyebabkan range data membesar dan menjadi semakin luas. Outlier mengandung nilai yang berbeda dari sebaran data yang lain, sehingga ketika nilainya sangat besar atau sangat kecil, akan mempengaruhi nilai maksimum dan nilai minimum sebuah kumpulan data. Jika outlier tidak ditangani dengan baik, maka outlier dapat mempengaruhi nilai statistik dalam sebuah data seperti rata-rata.

Outlier dapat diabaikan atau dihilangkan dari data, ataupun ditangani dengan mengganti nilainya dengan nilai pengganti yang lain. Pendekatan terhadap manajemen outlier sangat bergantung kepada data yang dimiliki. Dalam setiap data, ada makna yang perlu dijaga agar dapat diuraikan menjadi informasi. Bisa saja outlier dihilangkan namun akan mempengaruhi keutuhan data. Sehingga, dua opsi untuk menghadapi outlier yaitu menghilangkan dan mengganti nilainya dengan yang lain harus dilakukan berdasarkan kebutuhan. [1]

2. Korelasi merupakan pengukur hubungan dua variabel atau lebih yang dinyatakan sebagai tingkat hubungan / derajat keeratan antar variabel. Dalam menggunakan korelasi, tidak dipersoalkan adanya ketergantungan atau dengan kata lain, variabel yang satu tidak harus berkaitan dengan variabel lainnya. [2]

Menurut Dodge, Y (2008) dalam bukunya yang berjudul "*The concise encyclopedia of statistics*." Pada korelasi terdapat koefisien korelasi yang memiliki *range* antara -1 hingga 1. Nilai "0" pada koefisien korelasi memiliki arti bahwa kedua variabel yang dihitung koefisien korelasinya tidak memiliki hubungan sama sekali / tidak berhubungan. Nilai "-1" pada koefisien korelasi memiliki arti bahwa kedua

variabel memiliki korelasi negatif sempurna (*perfect negative correlation*), sedangkan nilai “1” memiliki arti bahwa kedua variabel memiliki korelasi positif sempurna (*perfect positive correlation*). [3]

Teori statistik korelasi memiliki implikasi pada konsep teori statistik analisis regresi yang juga berguna untuk mengukur hubungan antara dua variabel. Hubungan dua variabel yang memiliki nilai koefisien korelasi mendekati 1 (satu) atau -1 (minus satu) menandakan korelasi dua variabel yang erat. Untuk mengetahui bentuk korelasi dua variabel yang erat tersebut digunakan teori statistik analisis regresi. [2]


3. Macam - macam teori dasar *machine learning*, penjelasan, dan implikasinya:

| Dasar                 | Penjelasan  | Implikasi  |
|-----------------------|---|--|
| Supervised Learning   | Adalah proses pembelajaran yang dilakukan dengan memberikan arahan kepada mesin terhadap target yang diinginkan.  | Salah satu contoh implikasi supervised learning adalah untuk memprediksi harga rumah. Dengan melakukan training pada data dengan variabel yang diperlukan, harga rumah dapat diprediksi. |
| Unsupervised Learning | Adalah proses pembelajaran yang dilakukan tanpa adanya arahan kepada mesin sehingga pola dan interpretasi harus ditemukan sendiri oleh mesin tersebut tanpa ada pembandingan maupun arahan. | Salah satu contoh implikasi unsupervised learning adalah dengan menggunakan clustering untuk menemukan segmentasi customer.  |
| Training              | Adalah sekumpulan data yang digunakan untuk membentuk pola yang menghasilkan output.  | Implikasi dari training adalah data dilatih sedemikian rupa untuk pengujian.   |
| Testing               | Adalah sekumpulan data yang digunakan untuk menguji hasil latihan   | Implikasi dari testing adalah output berupa hasil uji coba prediksi  |
| Validation            | Adalah proses membandingkan performa antara data latih dan data uji untuk melihat bagaimana performa algoritma dalam melakukan machine learning.  | Implikasi dari validasi adalah hasil evaluasi dari model algoritma. Hal tersebut untuk mengetahui bagaimana baiknya implementasi model algoritma terhadap permasalahan yang diangkat.    |

4. *Artificial intelligence* merupakan simulasi inteligensi atau pemikiran manusia dalam mesin yang diprogram untuk meniru aksi manusia. *Artificial intelligence* pertama kali dicetuskan oleh Turing (1950, 433) dengan membuat suatu percobaan yaitu *imitation game* yang dimainkan oleh 3 orang dengan berbagai pertanyaan. Lantas apa kaitannya dengan *machine learning* maupun *deep learning*? Seiring dengan kemajuan zaman, penggunaan AI semakin menonjol di berbagai mesin dan teknologi yang digunakan manusia. Sehingga *machine learning* dan *deep learning* digunakan untuk pengembangan kecerdasan buatan. Dengan *machine learning*, kecerdasan buatan dapat digunakan untuk melakukan analisis atau prediksi dari algoritma eksak, contohnya adalah pada *marketplace*, AI dapat menampilkan rekomendasi sesuai dari aktivitas yang dilakukan penggunanya. Sedangkan dengan *deep learning*, yang merupakan pengembangan dari *machine learning*, bertujuan untuk meniru cara kerja otak manusia menggunakan *neural network*. Dengan membentuk suatu jaringan tersebut, *deep learning* dapat memahami karakteristik data dan memutuskan sebuah output berdasarkan karakteristik yang dipahami. Contoh penggunaan *deep learning* adalah pada sistem keamanan yang berupa *face recognition*. Dari sini, kaitan dari AI, ML, dan DL sangatlah dalam, dikarenakan dalam pengembangan dan pengimplementasiannya, AI menggunakan ML dan DL.

5. Interpretasi data adalah proses penjelasan makna dari sebuah data kepada orang lain. Interpretasi data dilakukan dengan melihat data, melakukan pengolahan, menarik kesimpulan, kemudian menyajikan hasil dari pengolahan menjadi sebuah informasi baik dalam bentuk lisan maupun tulisan.

Interpretasi data penting dilakukan untuk dapat mengetahui makna dari sebuah data. Selain itu, interpretasi data digunakan untuk mengambil keputusan terkait dengan hal-hal yang berkaitan dengan interpretasi yang dilakukan. Dalam mengolah data, tantangan yang harus dihadapi adalah bagaimana agar hasil interpretasi sesuai dengan data. Interpretasi data harus menghasilkan sesuatu yang bernilai atau berguna. Misalnya, dari sebuah data harus dapat diketahui apa dampaknya terhadap suatu



fenomena yang terjadi dalam masyarakat. Interpretasi dapat dilakukan untuk menunjukkan keadaan sekarang, sebelumnya, maupun yang akan datang.

Interpretasi data sangat berkaitan dengan *story telling* dan *decision making*. Interpretasi data bisa menjadi *story telling* kita dalam sebuah data, bisa dijelaskan alur pengumpulan data dan fungsi masing-masing atribut yang digunakan. Selain itu, bentuk *story telling* juga bisa muncul ketika sebuah baris diterjemahkan masing-masing atributnya. Misalnya, dalam sebuah tabel terdapat kolom nama, pekerjaan, dan gaji, maka bentuk *story telling* dari interpretasi data menjelaskan bahwa seseorang dengan nama A dan pekerjaan B memiliki gaji sebesar 10 juta, sedangkan nama B dan Pekerjaan C memiliki gaji sebesar 20 juta. *Story telling* menjelaskan data secara umum.

*Decision making* dalam interpretasi data berarti makna yang didapatkan akan digunakan dalam pengambilan keputusan terhadap suatu fenomena atau keadaan yang terjadi. Misalnya, dari data penjualan, jika dilakukan interpretasi, maka diketahui produk mana yang dijual dengan baik dan mana produk yang kurang laku di pasar. Maka, bisa diambil keputusan untuk melanjutkan penjualan produk-produk tertentu sesuai dengan hasil interpretasi data.

---

Koki Data

# Analisis & Prediksi Covid-19 Jakarta

---

Compfest 13 2021

Nama Anggota:

- |                            |   |                        |
|----------------------------|---|------------------------|
| ● Gentur Rizky Arganta     | - | Sistem Informasi Unair |
| ● Halim Wildan Awalurahman | - | Sistem Informasi Unair |
| ● Rifqi Hanief             | - | Sistem Informasi Unair |

## Latar Belakang

COVID-19 merupakan virus dengan intensitas penyebaran yang cepat / *superspread* [4]. Hal ini mengakibatkan infeksi pada lingkup orang yang banyak dan dapat berakibat fatal. Untuk menghadapi pandemi COVID-19, diperlukan pengambilan keputusan berdasarkan data yang terjadi di lapangan. Dengan menggunakan data, dapat dilihat bagaimana persebaran dan respons yang dilakukan untuk menghadapi pandemi.

## Jawaban Soal

### 1. Kode:

```
#Membuat dataset menjadi dataframe
df = pd.read_csv('Daily Update Data Agregat Covid-19 Jakarta.csv')
```

(Gambar 1. Memuat *dataset*)

```
#Memuat semua data pada kolom positif harian untuk dibuat tabel distribusi
dfpositifharian = df['Positif Harian']
dfpositifharian.plot(kind='hist', figsize=(10, 8), linewidth=2, color='whitesmoke', edgecolor='gray')
plt.xlabel("Positif Harian", labelpad=15)
plt.ylabel("frequency", labelpad=15)
plt.title("Distribusi Positif Harian Jakarta", y=1.012, fontsize=22)

#Mencari mean, median, dan modus dengan memasukkan data pada kolom positif harian pada fungsi dari library numpy
mean_positif_harian = round(dfpositifharian.mean(), 2)
median_positif_harian = dfpositifharian.median()
mode_positif_harian = dfpositifharian.mode().iloc[0]

#Integrasi hasil mean, median, dan modus pada tabel distribusi
measurements = [mode_positif_harian, median_positif_harian, mean_positif_harian]
names = ["modus", "median", "mean"]
colors = ['red', 'green', 'blue']
for measurement, name, color in zip(measurements, names, colors):
    plt.axvline(x=measurement, linestyle='--', linewidth=2.5, label='{0} at {1}'.format(name, measurement), c=color)
plt.legend()
plt.show()
```

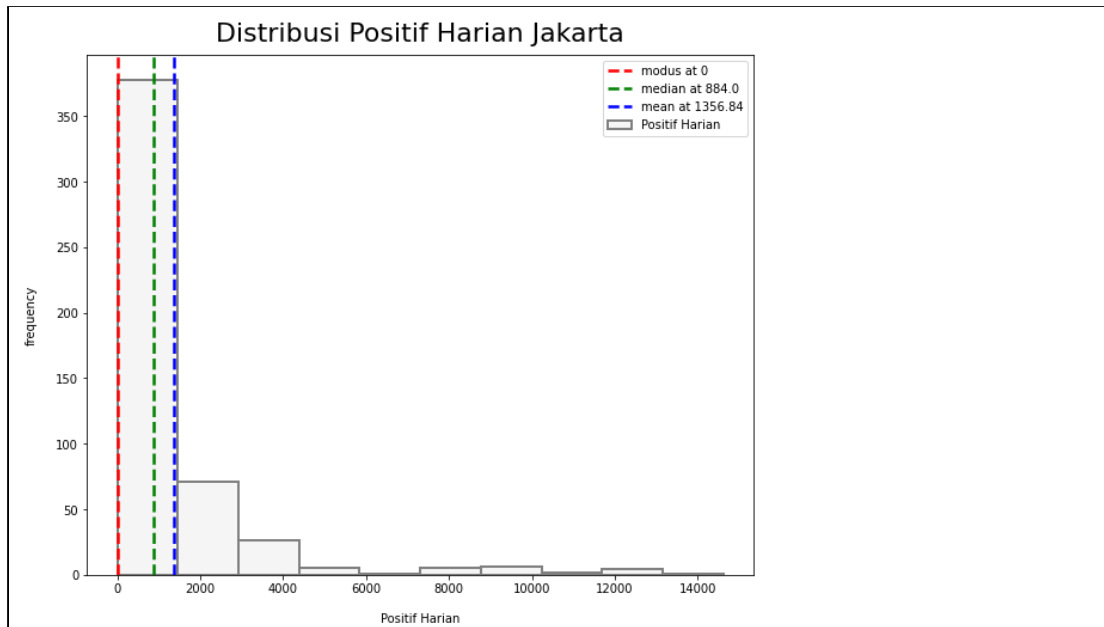
(Gambar 2. Kode mean, median, mode, dan distribusi data kolom “Positif Harian”)

### Penjelasan kode:

Pertama, dimuat *file dataset* dan dimasukkan pada variabel bernama *df*. Setelah itu, data pada kolom “Positif Harian” dimasukkan pada variabel baru bernama *df positif harian* untuk dibuat histogram dengan sumbu x yang menunjukkan distribusi positif harian Covid-19 di Jakarta dan sumbu y yang menunjukkan frekuensi distribusi positif harian Covid-19 di Jakarta. Terakhir, menghitung mean, median, dan

modus dari kolom “Positif Harian” lalu mengintegrasikannya ke dalam grafik histogram.

Hasil kode:



(Gambar 3. Grafik distribusi dan modus, median, mean data kolom “Positif Harian”)

Dari grafik histogram yang dihasilkan, diperoleh informasi bahwa:

- Modus dari data pada kolom “Positif Harian” adalah 0, median sebesar 884, dan mean sebesar 1356.84.
- Dari hasil mean jika dibulatkan dapat diartikan bahwa terdapat rata rata 1357 orang yang terinfeksi Covid-19 dari tanggal 1 Maret hingga 13 Juli atau pada saat data diunduh.
- Dari pola distribusi positif harian Jakarta dapat diartikan bahwa probabilitas jumlah positif harian paling tinggi terletak pada  $range\ 0 \leq x \leq 1461.9$ , disusul dengan  $range\ 1461.9 < x \leq 2923.8$ , lalu  $range\ 2923.8 < x \leq 4385.7$ , lalu  $range\ 4385.7 < x \leq 5847.6$  dan  $range\ 5847.6 < x \leq 7309.5$ , lalu  $range\ 7309.5 < x \leq 8771.4$ , lalu  $range\ 8771.4 < x \leq 10233.3$ , lalu dengan probabilitas yang sama pada  $range\ 10233.3 < x \leq 11695.18$ , terakhir dengan probabilitas yang sama pada  $range\ 11695.18 < x \leq 13157.1$ , lalu  $range\ 13157.1 < x \leq 14619$ . Hasil probabilitas tersebut dapat dimanfaatkan oleh Pemerintah untuk menyesuaikan

persiapan menghadapi jumlah pasien positif harian di Provinsi Jakarta untuk setidaknya tidaknya pada  $range \ 0 \leq x \leq 1461.9$  yang termasuk  $range$  dengan probabilitas paling tinggi.

2. Kode:

```
#Mengambil kolom positif harian untuk mencari nilai minimal dan maksimal
df1 = df['Positif Harian']
#Merubah nilai NaN menjadi 0
df1.fillna(0,inplace=True)
df2 = df["Tanggal"]
#mencari nilai terendah di kolom positif harian
dateMin = df1.idxmin()
testMin = df1.min()
#mencari nilai tertinggi di kolom positif harian
dateMax = df1.idxmax()
testMax = df1.max()
#Print hasil pengolahan data pminimal dan maksimal positif harian
print("nilai minimal positif harian".center(60,"="))
print(testMin)
print("nilai minimal didapat pada tanggal:", df2[dateMin])
print("")
print("nilai maksimal positif harian".center(60,"="))
print(testMax)
print("nilai maksimal didapat pada tanggal:", df2[dateMax])
```

(Gambar 4. Mengambil nilai maksimal dan minimal keseluruhan data)

```
#membuat dataframe baru yang berisi Tanggal, Positif Harian, serta month_year
dfbulan = df.iloc[:,[0,11]]
#merubah tipe data Tanggal menjadi datetime
dfbulan['Tanggal'] = pd.to_datetime(dfbulan['Tanggal'])
#menambahkan kolom month_year yang berisi bulan dan tahun
dfbulan['month_year'] = pd.to_datetime(dfbulan['Tanggal']).dt.to_period('M')
minmax = dfbulan
minmax = minmax.rename(columns={'Positif Harian': 'Positif_Harian'})
#melakukan grouping berdasar bulan dan tahun
testmax = minmax.groupby(['month_year'])
#mencari nilai minimal dan maksimal positif harian pada masing-masing bulan
minmaxbulan = testmax.agg(Nilai_Minimum=('Positif_Harian', np.min), Nilai_Maximum=('Positif_Harian', np.max))
#menampilkan nilai
print("nilai minimal dan maksimal positif harian per bulan".center(70,"="))
print(minmaxbulan)
```

(Gambar 5. Menampilkan nilai maksimal dan minimal tiap bulan)



Penjelasan kode:

Pertama, dibuat 2 buah dataframe baru yang memuat atribut 'Positif Harian' dan 'Tanggal'. Selanjutnya, membuat 2 variabel yaitu testMin dan testMax yang masing-masing bernilai minimal dan maksimal dari atribut 'Positif Harian'. Adapun dateMin dan dateMax digunakan untuk mendapatkan indeks posisi nilai minimal dan positif berada. Sehingga dapat ditemukan pada tanggal berapa nilai positif harian minimal dan maksimal. Selain itu, dilanjutkan dengan mencari nilai minimal dan maksimal pada setiap bulannya. Dibuat dataframe yang memiliki atribut 'Positif Harian', 'Tanggal', serta bulan dan tahun ('month\_year'). Setelah itu dilakukan groupby berdasarkan 'month\_year', dan dilanjutkan dengan aggregate yang berisikan nilai minimal dan maksimal positif harian.

Hasil kode dan penjelasan:

```
=====nilai minimal positif harian=====
0
nilai minimal didapat pada tanggal: 3/1/2020

=====nilai maksimal positif harian=====
14619
nilai maksimal didapat pada tanggal: 7/12/2021
```

(Gambar 6. Nilai minimum dan maksimal positif harian)

```
=====nilai minimal dan maksimal positif harian per bulan=====
      Nilai_Minimum  Nilai_Maximum
month_year
2020-03              0             98
2020-04             65            223
2020-05             55            182
2020-06             61            239
2020-07            147            585
2020-08            357           1114
2020-09            807           1505
2020-10            612           1430
2020-11            587           1579
2020-12            951           2096
2021-01           1657           3792
2021-02            373           4213
2021-03            384           2058
2021-04            393           1337
2021-05            161           1064
2021-06            519           9394
2021-07           7541          14619
```

(Gambar 7. Nilai minimum dan maksimum positif harian per bulan)

Didapatkan untuk nilai terendah pasien yang positif dalam harian adalah 0, nilai tersebut dicapai pada tanggal 1 maret 2020. Sedangkan untuk nilai tertinggi pasien yang positif dalam harian adalah 14619 jiwa, nilai tersebut dicapai pada tanggal 12 juli 2021. Adapun dihasilkan nilai terendah serta tertinggi untuk masing-masing bulan dan tahunnya.

3. Kode:

```
#Fungsi deteksi outlier
def detect_outlier(data):
    outliers = []
    threshold = 3
    mean_l = np.mean(data)
    std_l = np.std(data)

    for y in data:
        z_score = (y-mean_l)/std_l
        if np.abs(z_score) > threshold:
            outliers.append(y)

    return outliers
```

(Gambar 7. Fungsi deteksi *outlier*)

```
#Memasukkan kolom 'Positif Harian' pada fungsi deteksi outlier
outlier_datapoints = detect_outlier(df['Positif Harian'])
print('outliers found:')
print(outlier_datapoints)

#Membuat looping untuk menambahkan semua outlier yang terdeteksi pada suatu array
outliers_data = []
for i in range(len(df)):
    for j in outlier_datapoints:
        if df.iloc[i, 1] == j:
            outliers_data.append(df.iloc[i,:])

print()
print('Detail Data Outlier:')
outliers_data = pd.DataFrame(outliers_data)
print(outliers_data)
```

(Gambar 8. Print *outlier*)

```
#Membuat visualisasi Outliers dengan Scatter Plot
x = outliers_data['Tanggal Jam']
y = outliers_data['Positif Harian']
plt.plot(x,y,'ro')
plt.xlabel("Tanggal", labelpad = 30)
plt.ylabel("Outliers", labelpad = 15)
plt.title("Scatter Plot Outliers Positif Harian Jakarta", y=1.012, fontsize=22)
for i_x, i_y in zip(x, y):
    plt.text(i_x, i_y, '({}, {})'.format(i_x, i_y))
plt.show()
```

(Gambar 9. Visualisasi *outlier* dengan scatter plot)

```
#Mengubah index menjadi datetimeindex
datetimeindex = df['Tanggal Jam'].str.replace('/', '-')
df['Tanggal Jam'] = pd.to_datetime(datetimeindex)
df.set_index('Tanggal Jam', inplace=True)

#Menghilangkan Outliers
z_scores = stats.zscore(df)

abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3).all(axis=1)
df = df[filtered_entries]

print(df)

#Membuat model Extreme Value Analysis
dfl = df['Positif Harian']
dfseries = pd.Series(dfl)
model = EVA(data=dfseries)
```

(Gambar 10. Membuat model EVA)

```
#Menjalankan model dan memprediksi extreme value dengan metode Markov chain Monte Carlo
model.fit_model(model='Emcee')
print(model)

model.plot_diagnostic()

plt.show()

summary = model.get_summary(
    return_period=[1, 30, 90, 180, 270, 365],
)

print(summary)
```

(Gambar 11. Menjalankan model)

#### Penjelasan kode:

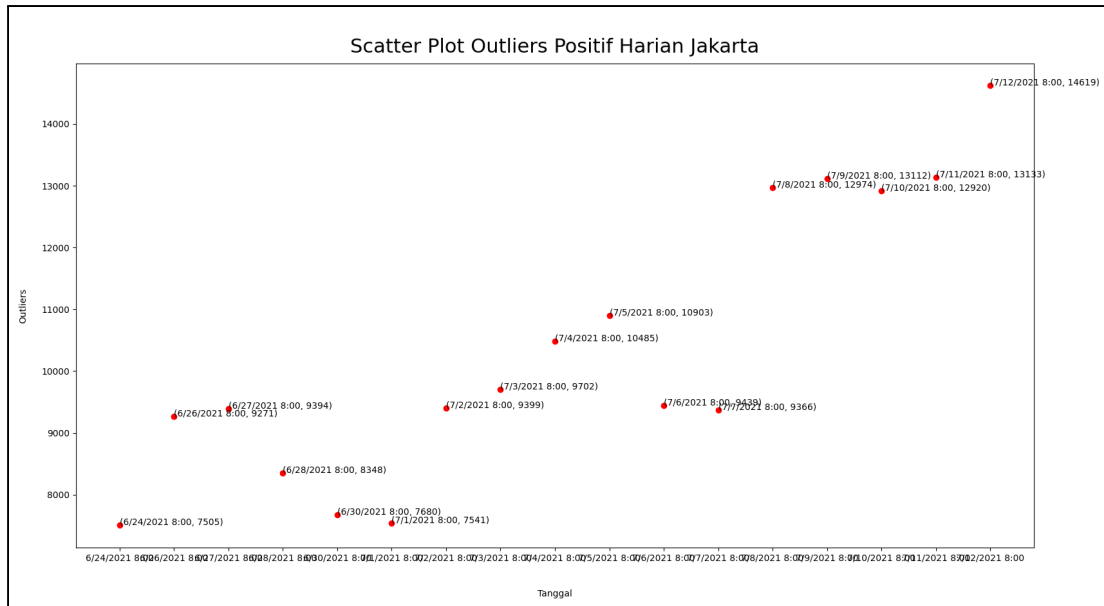
Pada kode dilakukan deteksi *outlier* dengan membuat fungsi untuk menghitung z-score dan akan menyeleksi data yang ada pada kolom “Positif Harian.” Jika data pada kolom positif harian memiliki z-score melebihi *threshold* yaitu 3, maka data tersebut akan terhitung sebagai *outlier*. Nilai z-score yang melebihi 3 menunjukkan ada deviasi yang cukup besar dari mean, oleh karena itu dianggap sebagai *outlier* atau *unusual data*. Pada kode juga dilakukan analisis dengan model *Extreme Value Analysis* (EVA) untuk memprediksi kapan kembalinya kejadian ekstrem atau tak terduga, dalam kata lain, memprediksi kapan *outlier* pada data akan muncul kembali. Memprediksi kapan terjadinya *outlier* pada data dalam konteks ini bermanfaat karena *outlier* terjadi bukan karena kesalahan atau *error* pada koleksi data, namun karena terjadinya *outbreak* atau lonjakan ekstrem penularan virus Covid-19 di Jakarta, dalam kata lain, dapat bermanfaat untuk memprediksi kapan *outbreak* akan terjadi lagi. Sebelum pembuatan model, dilakukan pembuangan terhadap data yang termasuk *outlier* agar prediksi lebih akurat dan hasil prediksi tidak terlalu “berlebihan.”

#### Hasil kode dan penjelasan:

```
outliers found:
[7505, 9271, 9394, 8348, 7680, 7541, 9399, 9702, 10485, 10903, 9439, 9366, 12974, 13112, 12920, 13133, 14619]

Detail Data Outlier:
  Tanggal Jam  Positif Harian
480 6/24/2021 8:00      7505
482 6/26/2021 8:00      9271
483 6/27/2021 8:00      9394
484 6/28/2021 8:00      8348
486 6/30/2021 8:00      7680
487 7/1/2021 8:00       7541
488 7/2/2021 8:00      9399
489 7/3/2021 8:00      9702
490 7/4/2021 8:00     10485
491 7/5/2021 8:00     10903
492 7/6/2021 8:00      9439
493 7/7/2021 8:00      9366
494 7/8/2021 8:00     12974
495 7/9/2021 8:00     13112
496 7/10/2021 8:00     12920
497 7/11/2021 8:00     13133
498 7/12/2021 8:00     14619
```

(Gambar 12. *Outlier* pada data kolom “Positif Harian”)



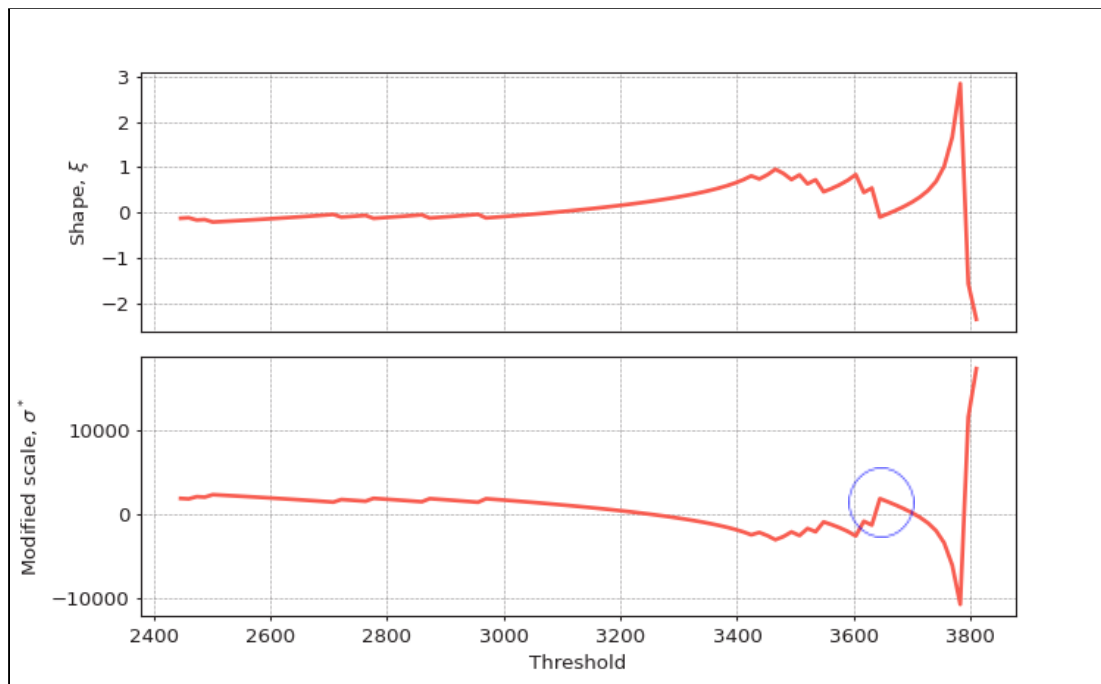
(Gambar 13. Scatter plot pada data kolom “Positif Harian”)

Dari usaha pencarian *outlier* pada kode, ditemukan sebanyak 17 *outlier* yaitu:

| Tanggal      | Outlier |
|--------------|---------|
| 25 Juni 2021 | 7505    |
| 26 Juni 2021 | 9271    |
| 27 Juni 2021 | 9394    |
| 28 Juni 2021 | 8348    |
| 30 Juni 2021 | 7680    |
| 1 Juli 2021  | 7541    |
| 2 Juli 2021  | 9399    |
| 3 Juli 2021  | 9702    |
| 4 Juli 2021  | 10485   |
| 5 Juli 2021  | 10903   |
| 6 Juli 2021  | 9439    |
| 7 Juli 2021  | 9366    |
| 8 Juli 2021  | 12974   |
| 9 Juli 2021  | 13112   |

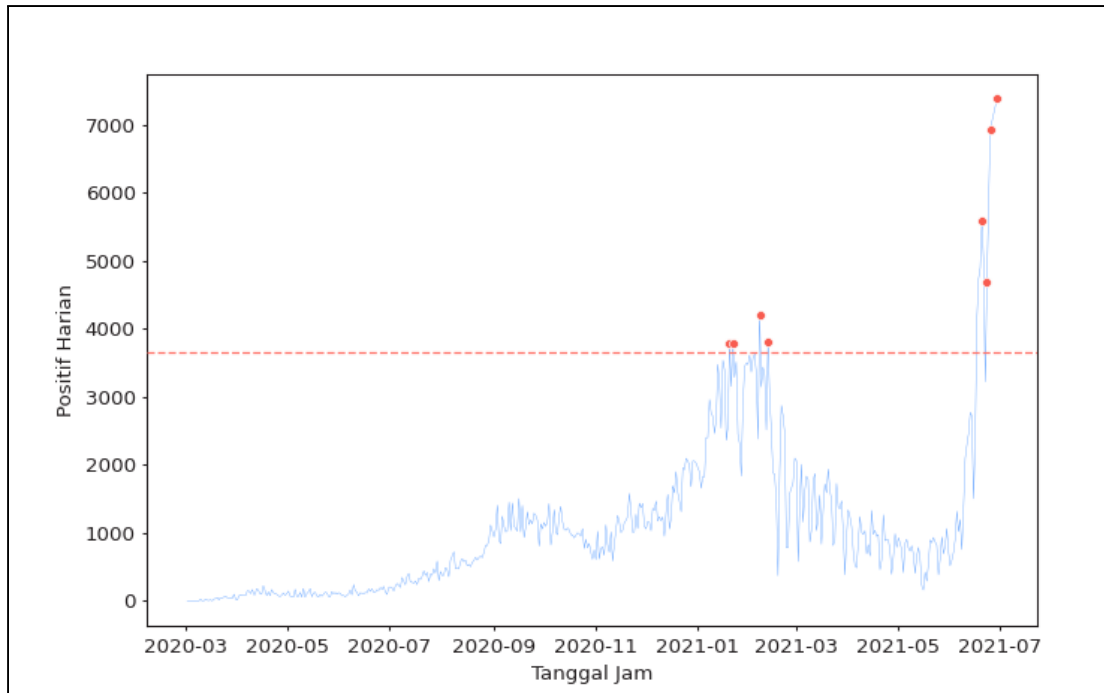
|              |       |
|--------------|-------|
| 10 Juli 2021 | 12920 |
| 11 Juli 2021 | 13133 |
| 12 Juli 2021 | 14619 |

Data *outlier* tersebut terjadi bukan karena kesalahan pada pengambilan data namun menandakan bahwa pada tanggal munculnya *outlier* tersebut terjadi *outbreak* atau penularan ekstrem virus Covid-19 di Jakarta. Dapat dilakukan usaha untuk memprediksi kapan kemunculan *outlier* tersebut dengan *Extreme Value Analysis* (EVA). Prediksi *outlier* dapat berguna untuk menyiapkan kemungkinan *outbreak* di masa yang akan datang.



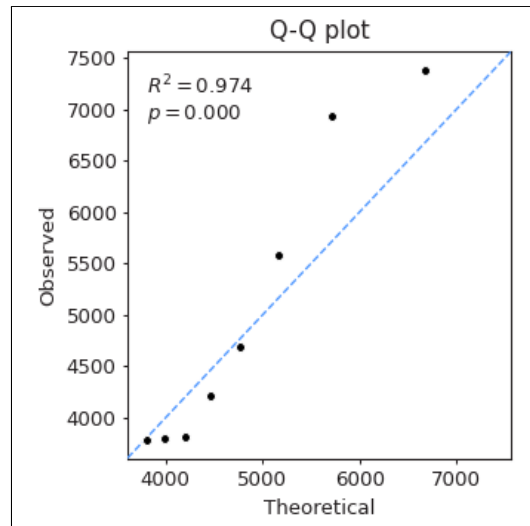
(Gambar 14. Grafik *Generalized Pareto Distribution* (GPD))

Pendekatan yang digunakan untuk *Extreme Value Analysis* adalah *Peak-Over-Threshold* (POT) dimana variabel yang melebihi dari *threshold* yang sudah ditentukan akan dikategorikan sebagai *extreme value*. Penentuan *threshold* tersebut diambil dari perhitungan *Generalized Pareto Distribution* (GPD) dan dicari titik mana pada grafik yang menunjukkan instabilitas, dalam kasus kali ini terletak pada *threshold* 3644 yang ditunjukkan pada lingkaran biru di gambar 14.



(Gambar 15. Grafik *extreme value* yang dihasilkan dari perhitungan)

Pada gambar 15, terlihat terdapat 8 *extreme value* yang melebihi *threshold* yang sudah ditentukan yaitu, 3644. Dari data tersebut dapat diterjemahkan bahwa pernah terjadi *outbreak* atau kenaikan ekstrem penularan virus Covid-19 di Jakarta sebanyak 8 kali yaitu pada tanggal 20 Januari 2021, 22 Januari 2021, 7 Februari 2021, 12 Februari 2021, 20 Juni 2021, 23 Juni 2021, 25 Juni 2021, dan 29 Juni 2021.



(Gambar 16. Grafik kesesuaian model)

Grafik kesesuaian model didapatkan koefisien determinasi sebesar 0.974 yang mendekati 1. Hal tersebut menandakan bahwa model prediksi dengan jenis data cukup sesuai.

| return period | return value |
|---------------|--------------|
| 30.0          | 10809.174890 |
| 90.0          | 12323.581186 |
| 180.0         | 13279.065178 |
| 270.0         | 13837.987483 |
| 365.0         | 14253.562878 |

(Gambar 17. Hasil prediksi model *Extreme Value Analysis*)

Prediksi *return period* dan *return value* dari model *Extreme Value Analysis* (EVA) menggunakan pendekatan *Peak-Over-Threshold* (POT) jika dibulatkan adalah:

- Kemungkinan *outbreak* sebesar 10809 kasus setiap 30 hari / 1 bulan sekali.
- Kemungkinan *outbreak* sebesar 12323 kasus setiap 90 hari / 3 bulan sekali.
- Kemungkinan *outbreak* sebesar 13279 kasus setiap 180 hari / 6 bulan sekali.
- Kemungkinan *outbreak* sebesar 13837 kasus setiap 270 hari / 9 bulan sekali.
- Kemungkinan *outbreak* sebesar 14253 kasus setiap 365 hari / 1 tahun sekali.



#### 4. Kode:

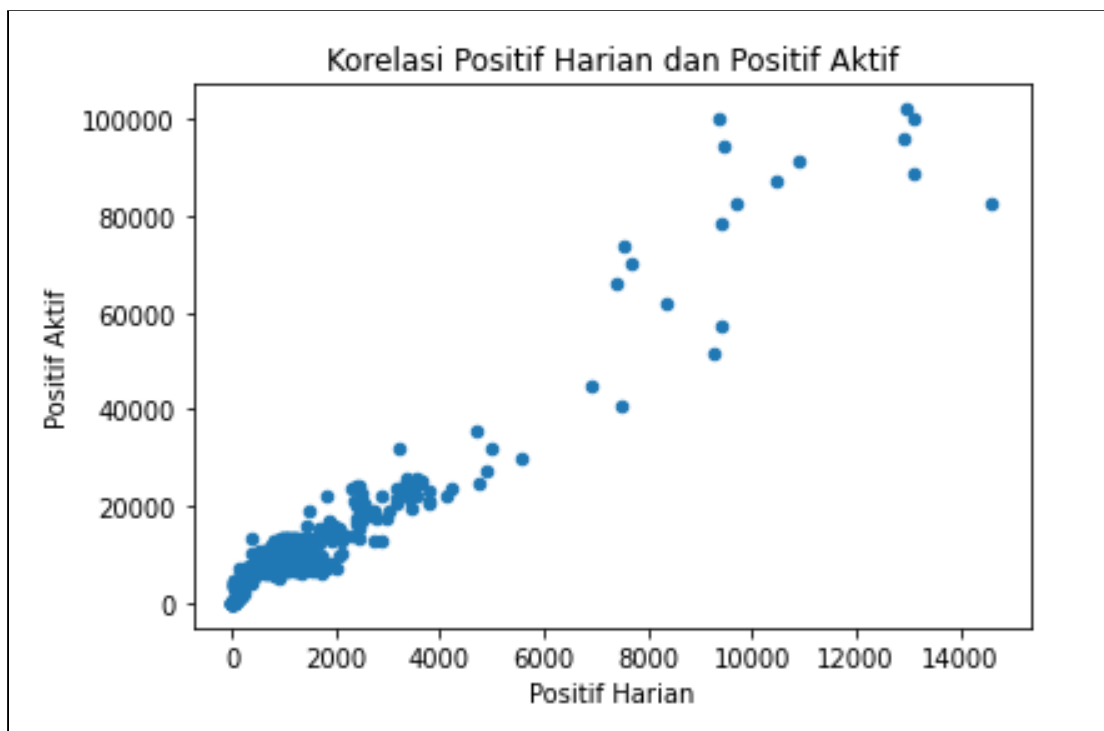
```
df.plot.scatter(x='Positif Harian', y='Positif Aktif', title= "Korelasi Positif Harian dan Positif Aktif");  
plt.show(block=True);
```

(Gambar 18. Menampilkan hubungan kedua variabel dengan plot)

#### Penjelasan kode:

Dua buah variabel yang diambil adalah variabel 'Positif Harian' dan 'Positif Aktif'. Kami menggunakan analisis korelasi dengan bantuan scatter plot. Hasil dari scatter plot seperti berikut.

#### Hasil kode dan penjelasan:



(Gambar 19. Scatter plot hubungan dua variabel)

Melihat hasil scatter plot diatas, terbentuk suatu hubungan korelasi yang positif. Dari korelasi tersebut, dapat disimpulkan bahwa peningkatan positif dalam hitungan harian akan mempengaruhi peningkatan pasien yang bersifat positif aktif.

## Hasil Analisis Tambahan

### Problem Statement

1. Bagaimana kebutuhan positif aktif terhadap rumah sakit
2. Bagaimana kebutuhan pasien yang bergejala terhadap rumah sakit

### Hypothesis

1. 50% positif aktif sudah mendapatkan perawatan di rumah sakit
2. 50% orang yang bergejala covid-19 sudah mendapatkan perawatan di rumah sakit

### Exploratory Data Analysis

1. Mengetahui jenis data

```
[ ] print(df.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 499 entries, 0 to 498
Data columns (total 17 columns):
 #   Column                                Non-Null Count  Dtype  
---  --
 0   Tanggal                              499 non-null    object  
 1   Jam                                  480 non-null    float64  
 2   Tanggal Jam                          499 non-null    object  
 3   Total Pasien                        499 non-null    int64  
 4   Sembuh                              499 non-null    int64  
 5   Meninggal                           499 non-null    int64  
 6   Self Isolation                      499 non-null    int64  
 7   Masih Perawatan                     499 non-null    int64  
 8   Belum Diketahui (masih verifikasi)  0 non-null      float64  
 9   Menunggu Hasil                     6 non-null      float64  
10  Tenaga Kesehatan Terinfeksi          6 non-null      float64  
11  Positif Harian                       499 non-null    int64  
12  Positif Aktif                       499 non-null    int64  
13  Sembuh Harian                       499 non-null    int64  
14  Tanpa Gejala                        359 non-null    float64  
15  Bergejala                           359 non-null    float64  
16  Belum Ada Data                      359 non-null    float64  
dtypes: float64(7), int64(8), object(2)
memory usage: 66.4+ KB
None
```

(Gambar 20. Tipe data pada *dataset*)

## 2. Mencari missing value dalam data

```
[ ] print(df.isna().sum())
```

|                                    |     |
|------------------------------------|-----|
| Tanggal                            | 0   |
| Jam                                | 19  |
| Tanggal Jam                        | 0   |
| Total Pasien                       | 0   |
| Sembuh                             | 0   |
| Meninggal                          | 0   |
| Self Isolation                     | 0   |
| Masih Perawatan                    | 0   |
| Belum Diketahui (masih verifikasi) | 499 |
| Menunggu Hasil                     | 493 |
| Tenaga Kesehatan Terinfeksi        | 493 |
| Positif Harian                     | 0   |
| Positif Aktif                      | 0   |
| Sembuh Harian                      | 0   |
| Tanpa Gejala                       | 140 |
| Bergejala                          | 140 |
| Belum Ada Data                     | 140 |
| dtype: int64                       |     |

(Gambar 21. Jumlah *missing value* pada masing masing kolom)

## 3. Mengetahui deskripsi statistik data

```
print(df.describe())
```

|       | Jam        | Total Pasien  | Sembuh        | Meninggal   | Self Isolation | Masih Perawatan |
|-------|------------|---------------|---------------|-------------|----------------|-----------------|
| count | 499.000000 | 499.000000    | 499.000000    | 499.000000  | 499.000000     | 499.000000      |
| mean  | 8.056112   | 176716.869739 | 161918.719439 | 3130.250501 | 7840.384770    | 3827.51503      |
| std   | 2.460299   | 177294.318794 | 166476.819553 | 2715.741135 | 10745.130805   | 4405.32151      |
| min   | 0.000000   | 0.000000      | 0.000000      | 0.000000    | 0.000000       | 0.000000        |
| 25%   | 8.000000   | 11931.500000  | 7243.000000   | 649.000000  | 2997.500000    | 1864.000000     |
| 50%   | 8.000000   | 109411.000000 | 98806.000000  | 2331.000000 | 5004.000000    | 2827.000000     |
| 75%   | 8.000000   | 353595.000000 | 340992.500000 | 5903.000000 | 8868.000000    | 4362.500000     |
| max   | 18.000000  | 677061.000000 | 584912.000000 | 9462.000000 | 73239.000000   | 30418.000000    |

[8 rows x 15 columns]

(Gambar 22 & 23. Deskripsi *dataset* berupa jumlah data, mean, standar deviasi, minimum, kuartil bawah, kuartil tengah, kuartil atas, dan maksimum)

#### 4. Mengetahui korelasi kolom tertentu



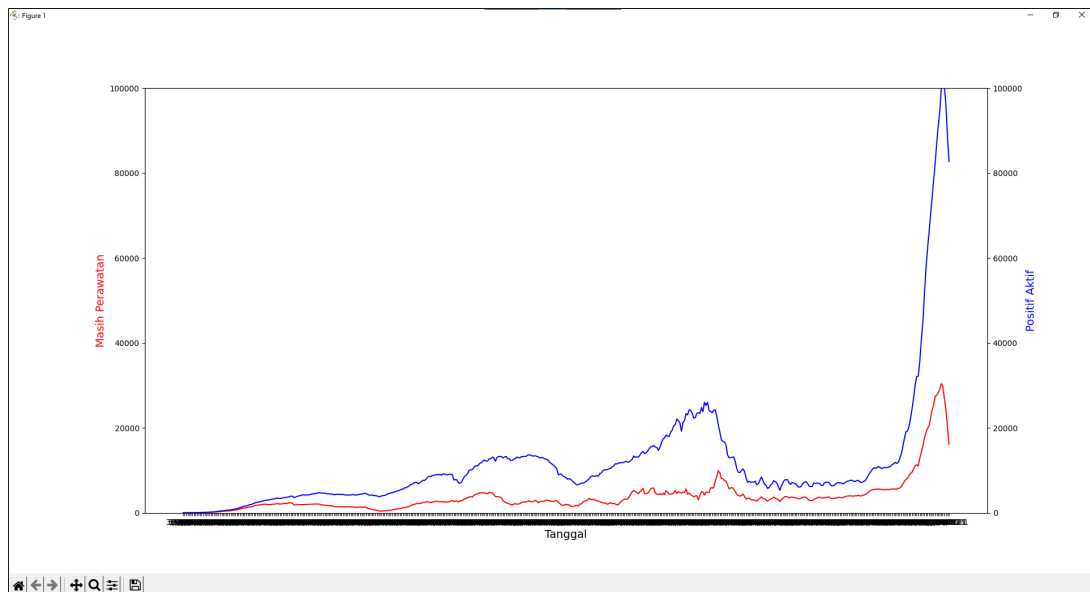
(Gambar 24. Heatmap hubungan korelasi antar dua variabel pada *dataset*)

#### Initial Findings

1. Korelasi positif antara positif aktif dan masih perawatan sebesar 0.95 mendekati 1 yang berarti bahwa kedua variabel memiliki korelasi. Artinya, ketika ada kenaikan pada positif aktif, akan terjadi kenaikan pada masih perawatan dan sebaliknya. Sehingga keduanya saling berpengaruh.
2. Korelasi positif antara bergejala dan masih perawatan sebesar 0.83 mendekati 1 yang berarti bahwa kedua variabel memiliki korelasi. Artinya, ketika ada kenaikan pada positif aktif, akan terjadi kenaikan pada masih perawatan dan sebaliknya. Sehingga keduanya saling berpengaruh.

## Deep Dive Analysis

### 1. Positif aktif terhadap masih perawatan



(Gambar 25. Grafik linear hubungan positif aktif dengan masih perawatan)

Ditemukan bahwa angka positif aktif yang mendapatkan perawatan memiliki kurva yang sama dengan positif aktif. Hal ini membuktikan korelasi positif yang terjadi antara keduanya.

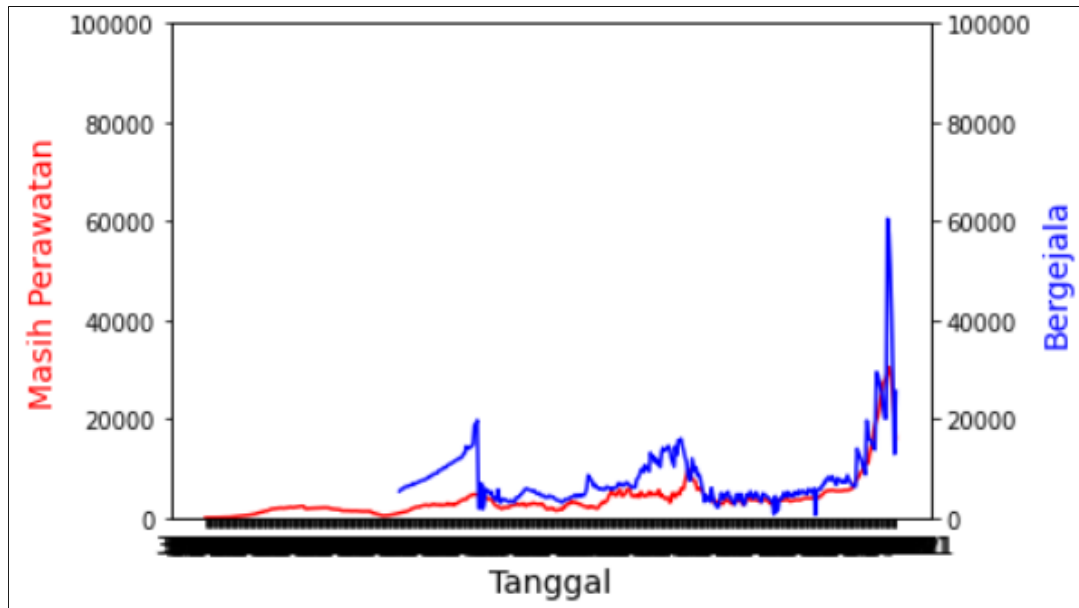
```
persentase_rawat = baru.loc[498, 'Masih Perawatan']/baru.loc[498, 'Positif Aktif']  
print('Persentase positif aktif terakhir yang mendapat perawatan: ', persentase_rawat)
```

```
Persentase positif aktif terakhir yang mendapat perawatan: 0.19478273489182096
```

(Gambar 26. Hasil persentase positif aktif yang mendapat perawatan)

Setelah ditelusuri, dari record terakhir pada tanggal 12 Juli 2021, ditemukan bahwa hanya 19% positif aktif yang mendapatkan perawatan di rumah sakit.

## 2. Gejala terhadap masih perawatan



(Gambar 27. Grafik linear hubungan masih perawatan dengan bergejala)

Ditemukan bahwa kurva pasien yang bergejala dengan masih perawatan tidak memiliki awalan yang sama, hal tersebut kemungkinan diakibatkan karena tidak adanya data pasien yang memiliki gejala. Walaupun begitu, kurva bergejala setiap harinya semakin mendekati kurva masih perawatan bahkan pada akhirnya melebihi masih perawatan. Sehingga terdapat korelasi antara kedua variabel tersebut.

```
persentase_rawat_bergejala= (baru.loc[498, 'Masih Perawatan']/baru.loc[498, 'Bergejala']) * 100  
print('Jumlah pasien bergejala terakhir yang masih perawatan =',persentase_rawat_bergejala,'%')
```

```
Jumlah pasien bergejala terakhir yang masih perawatan = 62.921436105793646 %
```

(Gambar 28. Hasil persentase bergejala yang masih dalam perawatan)

Adapun dari rekaman data terakhir, ternyata terdapat sekitar 62% pasien yang bergejala masih dalam perawatan di rumah sakit.

## Conclusion and Recommendations

Dari hasil temuan dan analisis yang didapatkan, ternyata hanya terdapat 19%, pasien positif aktif yang masih dalam perawatan, karena angka tersebut berada di bawah 50%, dapat diartikan bahwa hipotesis 1 tidak terpenuhi. Namun untuk pasien yang memiliki gejala penyakit, terdapat 62% yang masih dalam perawatan, karena angka tersebut lebih dari 50% dapat diartikan bahwa hipotesis 2 terpenuhi. Melihat kondisi persentase pasien positif aktif yang hanya 19% masih mendapatkan perawatan, kami menyarankan untuk pemerintah untuk memberikan perawatan kepada pasien aktif melalui fasilitas lain seperti asrama haji atau tempat yang memungkinkan lainnya, serta memberikan fasilitas penuh bagi pasien yang melakukan isolasi mandiri.

## Alasan Penggunaan Teknik EDA

Penggunaan teknik EDA yang digunakan adalah dengan mencari korelasi antar variabel. Alasan dari penggunaan teknik EDA tersebut adalah untuk memastikan serta mencari tahu bagaimana perubahan nilai pada suatu variabel akan mempengaruhi variabel lain. Sehingga dapat memudahkan dalam melakukan analisis lebih lanjut berdasarkan variabel yang memiliki nilai korelasi tinggi.

## Kesimpulan

1. Dari soal dataset ditemukan bahwa:
  - a. Mean dari dataset adalah sebesar 1357 saat *dataset* diunduh.
  - b. Median dari dataset adalah sebesar 884 saat *dataset* diunduh.
  - c. Modus dari dataset adalah 0 saat *dataset* diunduh.
  - d. Distribusi paling banyak ada pada *range*  $0 \leq x \leq 1461.9$ .
  - e. Nilai maksimal positif harian adalah sebesar 14619 saat *dataset* diunduh.
  - f. Nilai minimal positif harian adalah sebesar 0.
  - g. Nilai maksimal tiap bulan, terhitung dari Maret 2020 adalah, 98, 223, 182, 239, 585, 1114, 1505, 1430, 1579, 2096, 3792, 4213, 2058, 1337, 1064, 9394, 14619.

- h. Nilai minimum tiap bulan, terhitung dari Maret 2020 adalah, 0, 65, 55, 61, 147, 357, 807, 612, 587, 951, 1657, 373, 384, 393, 161, 519, 7541.
- i. Outlier ditemukan pada 17 titik yang bernilai 7505, 9271, 9394, 8348, 7680, 7541, 9399, 9702, 10485, 10903, 9439, 9366, 12974, 13112, 12920, 13133, dan 14619.
- j. Berdasarkan prediksi menggunakan model *Extreme Value Analysis* (EVA). Ditemukan bahwa ada kemungkinan *outbreak* sebesar 10809 kasus setiap 1 bulan sekali, 12323 kasus setiap 3 bulan sekali, 13279 kasus tiap 6 bulan sekali, 13837 kasus tiap 9 bulan sekali, dan 14253 kasus tiap 1 tahun sekali.
- k. Terdapat korelasi antara dua variabel yang dipilih yaitu positif aktif dan positif harian.

2. Dari analisis tambahan ditemukan bahwa:

Ternyata hanya terdapat 19%, pasien positif aktif yang masih dalam perawatan, karena angka tersebut berada di bawah 50%, dapat diartikan bahwa hipotesis 1 tidak terpenuhi. Namun untuk pasien yang memiliki gejala penyakit, terdapat 62% yang masih dalam perawatan, karena angka tersebut lebih dari 50% dapat diartikan bahwa hipotesis 2 terpenuhi. Melihat kondisi persentase pasien positif aktif yang hanya 19% masih mendapatkan perawatan, kami menyarankan untuk pemerintah untuk memberikan perawatan kepada pasien aktif melalui fasilitas lain seperti asrama haji atau tempat yang memungkinkan lainnya, serta memberikan fasilitas penuh bagi pasien yang melakukan isolasi mandiri.





## Daftar Pustaka

- [1] Ben-Gal, I. (2005). Outlier detection. In *Data mining and knowledge discovery handbook* (pp. 131-146). Springer, Boston, MA.
  
- [2] Kurniawan, R. (2016). *Analisis regresi*. Prenada Media.
  
- [3] Dodge, Y. (2008). *The concise encyclopedia of statistics*. Springer Science & Business Media.
  
- [4] Wong, F., & Collins, J. J. (2020). Evidence that coronavirus superspreading is fat-tailed. *Proceedings of the National Academy of Sciences*, 117(47), 29416-29418.