



# Implementasi Neural Network pada

## Masalah Klasifikasi Varian COVID-19

### Kelompok 1

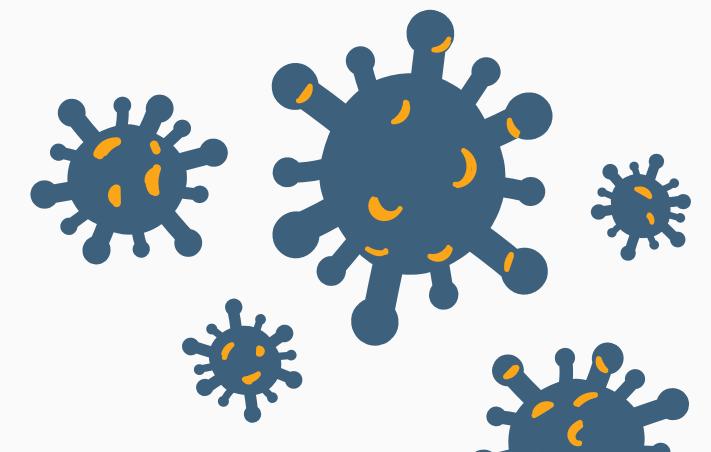
- |                              |            |
|------------------------------|------------|
| 1. Aqila Zahra Vanisa        | 2006568582 |
| 2. Herardita Cahyaning Wulan | 2006568632 |
| 3. Siti Khamiilaah Sayid     | 2006568922 |





# Pendahuluan

COVID-19 merupakan penyakit menular yang disebabkan oleh jenis coronavirus. Seiring berjalannya waktu, muncul beberapa varian COVID-19 di sejumlah negara karena mutasi genetik pada virus SARS-Cov-2 itu sendiri. Di setiap negara memungkinkan mempunyai gejala yang berbeda-beda, oleh karena itu kami akan mengklasifikasikan varian COVID-19 berdasarkan asal negaranya yang ditinjau dari beberapa gejala yang muncul.



# Ide Dasar



Eksplorasi Data



Pre-processing



Train-Test Data



Membuat Model



Hyperparameter  
Tuning



Evaluasi Model

# Eksplorasi Data

```
[ ] data = pd.read_csv("Cleaned-Data.csv")  
data
```

	Fever	Tiredness	Dry-Cough	Difficulty-in-Breathing	Sore-Throat	None_Sympton	Pains	Nasal-Congestion	Runny-Nose	Diarrhea	...	Gender_Male	Gender_Transgender	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe	Contact_Dont-Know	Contact_No	Contact_Yes	Country	
0	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	0	0	China	
1	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	0	1	0	China
2	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	1	0	0	China
3	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	0	0	0	1	0	China
4	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	0	0	0	1	0	China
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
316795	0	0	0	0	0	1	0	0	0	...	0	1	0	0	0	1	0	1	0	0	Other	
316796	0	0	0	0	0	1	0	0	0	0	...	0	1	0	0	0	1	1	0	0	Other	
316797	0	0	0	0	0	1	0	0	0	0	...	0	1	0	0	1	0	0	0	1	Other	
316798	0	0	0	0	0	1	0	0	0	0	...	0	1	0	0	1	0	0	0	1	Other	
316799	0	0	0	0	0	1	0	0	0	0	...	0	1	0	0	0	1	0	1	0	Other	

316800 rows x 27 columns



## Informasi Umum

- Ukuran Data : 316800 baris x 27 kolom
- Jumlah Fitur : 26
- Jumlah Kelas : 10



# Eksplorasi Data



## Fitur-fitur

'Fever', 'Tiredness', 'Dry-Cough', 'Difficulty-in-Breathing','Sore-Throat',  
'None\_Sympton', 'Pains', 'Nasal-Congestion','Runny-Nose', 'Diarrhea',  
'None\_Experiencing', 'Age\_0-9', 'Age\_10-19', 'Age\_20-24', 'Age\_25-59',  
'Age\_60+', 'Gender\_Female', 'Gender\_Male', 'Gender\_Transgender',  
'Severity\_Mild', 'Severity\_Moderate', 'Severity\_None', 'Severity\_Severe',  
'Contact\_Dont-Know', 'Contact\_No', 'Contact\_Yes'



## Kelas

'China', 'Italy', 'Iran', 'Republic of Korean', 'France', 'Spain',  
'Germany', 'UAE', 'Other-EUR', 'Other'



#

# Pre-processing

1

Grouping fitur

2

Memilih fitur

3

Mengubah jenis data

4

Menyimpan data ke array

# # Pre-processing : Grouping fitur yang mirip

'Severity\_Mild', 'Severity\_Moderate', 'Severity\_None', 'Severity\_Severe' → 'Condition'

```
severity_columns = data.filter(like='Severity_').columns

data['Severity_None'].replace({1:'None',0:'No'},inplace =True)
data['Severity_Mild'].replace({1:'Mild',0:'No'},inplace =True)
data['Severity_Moderate'].replace({1:'Moderate',0:'No'},inplace =True)
data['Severity_Severe'].replace({1:'Severe',0:'No'},inplace =True)

data['Condition'] = data[severity_columns].values.tolist()

def removing(list1):
    list1 = set(list1)
    list1.discard("No")
    a = ''.join(list1)
    return a

data['Condition'] = data['Condition'].apply(remove)
```

- \*Severity : Keparahan,
- None = Tidak parah
  - Mild = Ringan
  - Moderate = Sedang
  - Severe = Parah

'Age\_0-9', 'Age\_10-19', 'Age\_20-24', 'Age\_25-59', 'Age\_60+' → 'Age'

```
age_columns = data.filter(like='Age_').columns

data['Age_0-9'].replace({1:'0-9',0:'No'},inplace =True)
data['Age_10-19'].replace({1:'10-19',0:'No'},inplace =True)
data['Age_20-24'].replace({1:'20-24',0:'No'},inplace =True)
data['Age_25-59'].replace({1:'25-59',0:'No'},inplace =True)
data['Age_60+'].replace({1:'60+',0:'No'},inplace =True)

data['Age'] = data[age_columns].values.tolist()

def removing(list2):
    list2 = set(list2)
    list2.discard("No")
    b = ''.join(list2)
    return b

data['Age'] = data['Age'].apply(remove)
```



# # Pre-processing : Grouping fitur yang mirip

'Gender\_Female', 'Gender\_Male', 'Gender\_Transgender' → 'Gender'

```
gender_columns = data.filter(like='Gender_').columns

data['Gender_Female'].replace({1:'Female',0:'No'},inplace =True)
data['Gender_Male'].replace({1:'Male',0:'No'},inplace =True)
data['Gender_Transgender'].replace({1:'Transgender',0:'No'},inplace =True)

data['Gender'] = data[gender_columns].values.tolist()

def removing(list3):
    list3 = set(list3)
    list3.discard("No")
    c = ''.join(list3)
    return c

data['Gender'] = data['Gender'].apply(remove)
```

'Contact\_Dont-Know', 'Contact\_No','Contact\_Yes' → 'Contact'

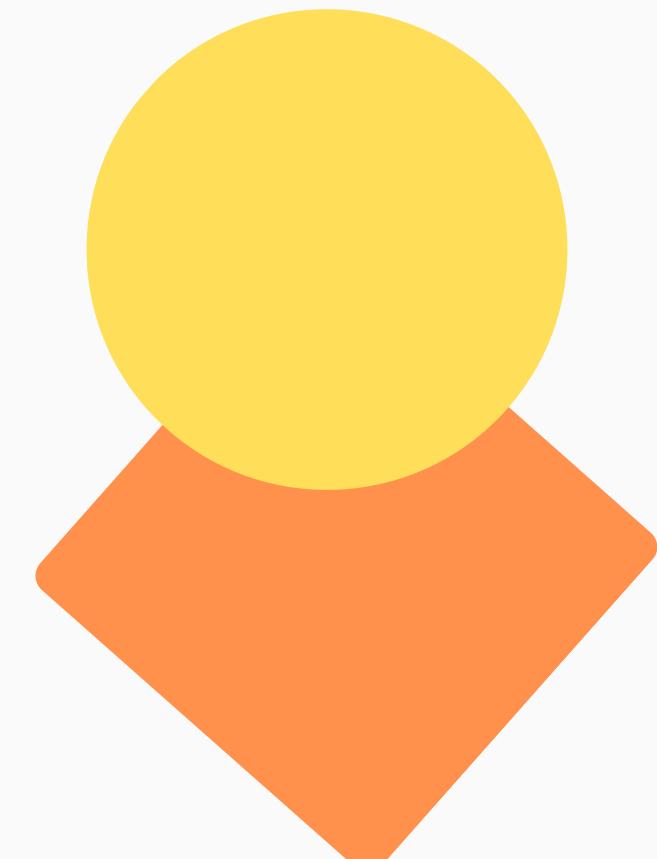
```
contact_columns = data.filter(like='Contact_').columns

data['Contact_Dont-Know'].replace({1:'Dont-Know',0:'No'},inplace =True)
data['Contact_No'].replace({1:'Not',0:'No'},inplace =True)
data['Contact_Yes'].replace({1:'Yes',0:'No'},inplace =True)

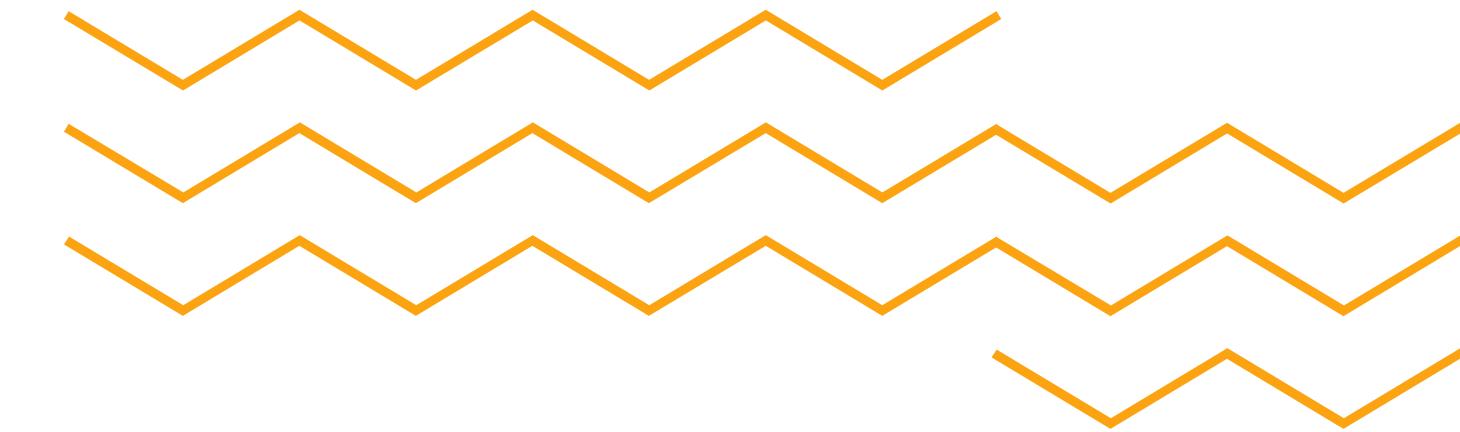
data['Contact'] = data[severity_columns].values.tolist()

def removing(list4):
    list4 = set(list4)
    list4.discard("No")
    d = ''.join(list4)
    return d

data['Contact'] = data['Contact'].apply(remove)
```



# # Pre-processing : Fitur yang dipilih



	Fever	Tiredness	Dry-Cough	Difficulty-in-Breathing	Sore-Throat	None_Sympton	Pains	Nasal-Congestion	Runny-Nose	Diarrhea	None_Experiencing	Age	Condition	Gender	Contact	Country
0	1	1	1	1	1	0	1	1	1		0	0-9	Mild	Male	Mild	China
1	1	1	1	1	1	0	1	1	1		0	0-9	Mild	Male	Mild	China
2	1	1	1	1	1	0	1	1	1		0	0-9	Mild	Male	Mild	China
3	1	1	1	1	1	0	1	1	1		0	0-9	Moderate	Male	Moderate	China
4	1	1	1	1	1	0	1	1	1		0	0-9	Moderate	Male	Moderate	China
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
316795	0	0	0	0	0	1	0	0	0	0	1	60+	Severe	Transgender	Severe	Other
316796	0	0	0	0	0	1	0	0	0	0	1	60+	Severe	Transgender	Severe	Other
316797	0	0	0	0	0	1	0	0	0	0	1	60+	None	Transgender	None	Other
316798	0	0	0	0	0	1	0	0	0	0	1	60+	None	Transgender	None	Other
316799	0	0	0	0	0	1	0	0	0	0	1	60+	None	Transgender	None	Other

316800 rows × 16 columns



Fitur-fitur

'Fever','Tiredness, 'Dry Cough', 'Difficulty-in-Breathing','Sore-Throat', 'None\_Sympton', 'Pains', 'Nasal-Congestion', 'Runny-Nose','Diarrhea', 'None Experiencing', 'Age', 'Condition', 'Gender', 'Contact','Country'

# # Pre-processing : Mengubah jenis data

Data fitur 'Condition', 'Age', 'Gender', dan 'Country' menjadi numerik

```
[ ] from sklearn import preprocessing

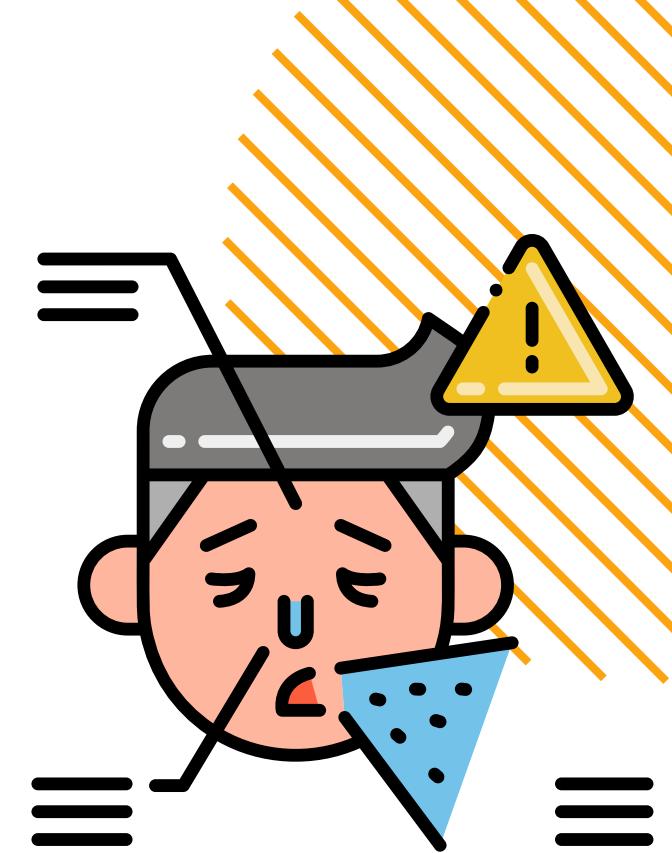
le_cond = preprocessing.LabelEncoder()
le_cond.fit(['None', 'Mild', 'Moderate', 'Severe'])
new_data['Condition'] = le_cond.fit_transform(new_data['Condition'])

le_country = preprocessing.LabelEncoder()
le_country.fit(['China', 'Italy', 'Iran', 'Republic of Korean', 'France',
                'Spain', 'Germany', 'UAE', 'Other-EUR', 'Other'])
new_data['Country'] = le_country.fit_transform(new_data['Country'])

le_age = preprocessing.LabelEncoder()
le_age.fit(['0-9', '10-19', '20-24', '25-59', '60+'])
new_data['Age'] = le_age.fit_transform(new_data['Age'])

le_gender = preprocessing.LabelEncoder()
le_gender.fit(['Female', 'Male', 'Transgender'])
new_data['Gender'] = le_gender.fit_transform(new_data['Gender'])

le_contact = preprocessing.LabelEncoder()
le_contact.fit(['Dont-Know', 'No', 'Yes'])
new_data['Contact'] = le_contact.fit_transform(new_data['Contact'])
```



	0 = China	6 = Other -EUR
1	= France	7 = Republic of Korean
2	= German	8 = Spain
3	= Iran	9 = UAE
4	= Italy	
5	= Other	

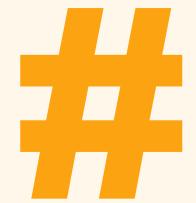


# # Pre-processing : Data numerik dalam array

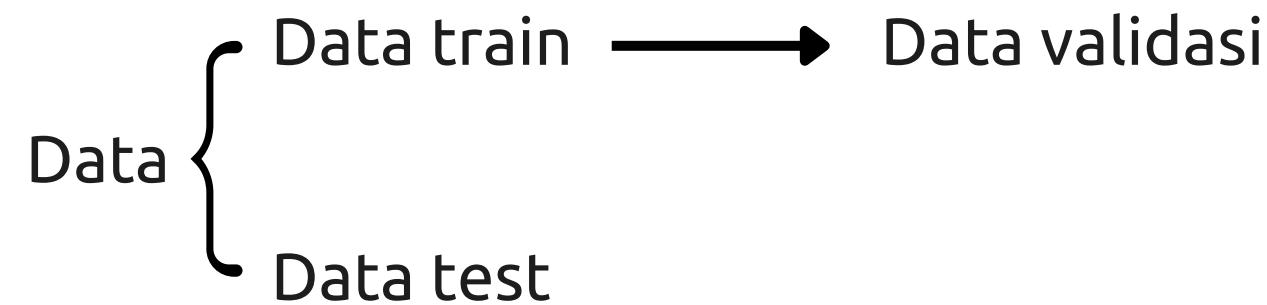
```
X = np.asarray(new_data[['Fever', 'Tiredness', 'Dry-Cough', 'Difficulty-in-Breathing',  
    'Sore-Throat', 'None_Sympton', 'Pains', 'Nasal-Congestion',  
    'Runny-Nose', 'Diarrhea', 'None_Experiencing', 'Age', 'Condition', 'Gender', 'Contact']]).values  
print('X : ',X)  
  
Y = np.asarray(new_data[['Country']]).values  
print('\nY : ',Y)
```

```
X : [[1 1 1 ... 0 1 0]           Y : [[0]  
    [1 1 1 ... 0 1 0]           [0]  
    [1 1 1 ... 0 1 0]           [0]  
    ...  
    [0 0 0 ... 2 2 2]           ...  
    [0 0 0 ... 2 2 2]           [5]  
    [0 0 0 ... 2 2 2]]          [5]  
                                [5]]
```





# Membagi data

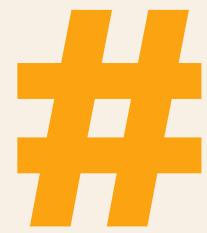


```
[ ] 1 from sklearn.model_selection import train_test_split  
2 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20)
```

```
[ ] 1 print("X_train: ", X_train.shape)  
2 print("Y_train: ", Y_train.shape)  
3 print("X_test: ", X_test.shape)  
4 print("Y_test: ", Y_test.shape)
```

```
X_train: (253440, 15)  
Y_train: (253440, 1)  
X_test: (63360, 15)  
Y_test: (63360, 1)
```

```
[ ] 1 X_valid, X_trainX = X_train[:150000]/255.0, X_train[150000:]/255.0  
2 y_valid, y_trainX = Y_train[:150000], Y_train[150000:]
```



# Membuat model

```
[ ] def build_model(hp):
    model = keras.models.Sequential()

    hp_layers = hp.Choice('layer', [3,10])
    hp_neurons = hp.Int('neuron', min_value = 100, max_value = 500, step = 50)

    for layers in range(hp_layers):
        model.add(keras.layers.Dense(hp_neurons, activation = 'relu'))
    model.add(keras.layers.Dense(10, activation = 'softmax'))

    optimizer = tf.keras.optimizers.Adam()
    model.compile(loss = 'sparse_categorical_crossentropy',
                  optimizer = optimizer,
                  metrics = ['accuracy'])
    return model
```





# Membuat Hyperparameter Tuning

```
[ ] tuner = kt.Hyperband(build_model,  
                         objective = 'val_accuracy',  
                         max_epochs = 10,  
                         directory = 'my_di',  
                         project_name = 'my_tuner')
```

```
[ ] stop_early = tf.keras.callbacks.EarlyStopping(  
    monitor = 'val_loss',  
    patience = 5  
)
```

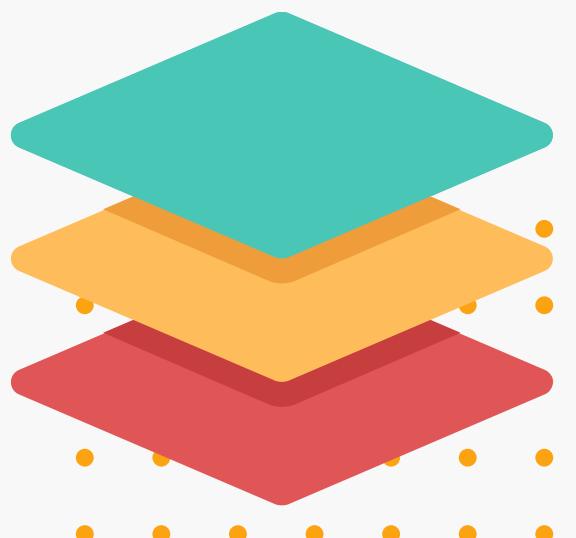
```
[ ] #proses pencarian hyperparameter  
tuner.search(X_trainX, y_trainX,  
              epochs = 50,  
              validation_data = [X_valid, y_valid],  
              callbacks = [stop_early])
```

```
[ ] best_hps = tuner.get_best_hyperparameters()[0]
```

Mencari hyperparameter terbaik

```
print("best number of layers: ", best_hps.get("layer"))  
print("best number of neuron: ", best_hps.get("neuron"))  
print("Best number of epochs: ", best_hps.get("tuner/epochs"))
```

```
best number of layers: 10  
best number of neuron: 300  
Best number of epochs: 2
```





# Memprediksi dengan Model

```
[ ] model = tuner.hypermodel.build(best_hps)

[ ] model.fit(X_trainX, y_trainX,
              epochs = 2,
              validation_data = [X_valid, y_valid])

Epoch 1/2
3233/3233 [=====] - 56s 17ms/step - loss: 2.3028 - accuracy: 0.0996 - val_loss: 2.3027 - val_accuracy: 0.1009
Epoch 2/2
3233/3233 [=====] - 60s 19ms/step - loss: 2.3027 - accuracy: 0.0996 - val_loss: 2.3028 - val_accuracy: 0.0989
<keras.callbacks.History at 0x7f792d02acd0>
```

```
[ ] #memprediksi dengan model
model.evaluate(X_test, Y_test)

1980/1980 [=====] - 7s 4ms/step - loss: 2.3027 - accuracy: 0.1010
[2.3027138710021973, 0.10101009905338287]
```

```
[ ] y_proba = model.predict(X_test)
y_proba

y_p = np.array([np.argmax(i) for i in y_proba])
print(y_p)
```

```
[6 6 6 ... 6 6 6]
```



# Mengapa hasil akurasi kecil ?

Find and Replace

Find  Replace

Find what:  No Format Set Format...

Within: Sheet  Match case  
Search: By Rows  Match entire cell contents  
Look in: Formulas Options <<

Find All Find Next Close

Book	Sheet	Name	Cell	Value
Cleaned-Data.csv	Cleaned-Data		\$A\$2	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,China
Cleaned-Data.csv	Cleaned-Data		\$A\$31682	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,Italy
Cleaned-Data.csv	Cleaned-Data		\$A\$63362	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,Iran
Cleaned-Data.csv	Cleaned-Data		\$A\$95042	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,Republic of Korean
Cleaned-Data.csv	Cleaned-Data		\$A\$126722	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,France
Cleaned-Data.csv	Cleaned-Data		\$A\$158402	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,Spain
Cleaned-Data.csv	Cleaned-Data		\$A\$190082	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,Germany
Cleaned-Data.csv	Cleaned-Data		\$A\$221762	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1, UAE
Cleaned-Data.csv	Cleaned-Data		\$A\$253442	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,Other-EUR
Cleaned-Data.csv	Cleaned-Data		\$A\$285122	1,1,1,1,0,1,1,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,Other

10 cell(s) found

Terlihat bahwa data tidak satu-satu

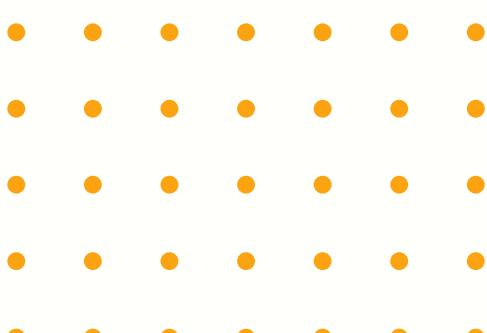
Terdapat data yang sama inputnya di semua fitur-fitur tetapi menghasilkan output (country) yang berbeda-beda.



# Kesimpulan

Model yang kami buat menghasilkan akurasi yang sangat kecil dikarenakan dataset yang kami pilih tidak satu-satu, artinya terdapat input yang sama persis pada semua fitur tetapi menghasilkan output (country) yang berbeda.

Hal ini mungkin saja terjadi karena gejala-gejala varian COVID-19 bisa saja sama walaupun variannya berbeda.





# Referensi

---

[1] Hungund, Bilal. COVID-19 Symptoms Checker. Kaggle, 2020

<https://www.kaggle.com/datasets/iamhungundji/covid19-symptoms-checker/metadata>

[2] Asisten Lab Matematika. MODUL 9 SAINS DATA. FMIPA UI, 2022