TUGAS CASE BASED 1

MACHINE LEARNING

Nama:

Hilman Taris Muttaqin 1301204208

Kode Dosen:

DDR

UNIVERSITAS TELKOM
TAHUN AKADEMIK 2022/2023

1. Ikhtisar kumpulan data yang dipilih

Mahasiswa dengan NIM genap mendapatkan data tentang aritmia, Membedakan antara ada dan tidak adanya aritmia jantung. Dibawah ini adalah karakteristik data yang digunakan.

Data Set Characteristics:	Multivariate	**Number of Instances:**	452	**Area:**	Life
Attribute Characteristics:	Categorical, Integer, Real	**Number of Attributes:**	279	**Date Donated**	1998-01-01
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	411976

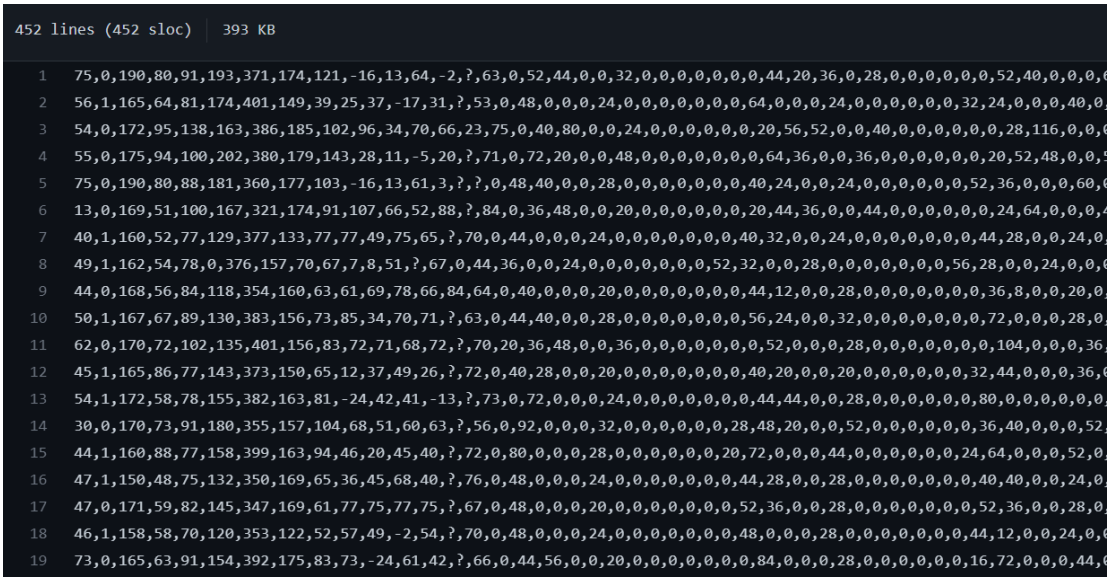
Masalah pertama saat melihat dataset ini adalah memiliki dimensi yang sangat besar, yaitu 279 dimensi data dan memiliki nilai yang bervariasi dari data float minus nol koma, hingga data yang memiliki nilai ratusan.

Saat pertama kali data dibuat menjadi dataframe, hal yang pertama kali terlihat adalah data tersebut tidak memiliki header atau nama kolom.

	0	1	2	3	4	5	6	7	8	9	...	270	271	272	273	274	275	276	277	278	279
0	75	0	190	80	91	193	371	174	121	-16	...	0.0	9.0	-0.9	0.0	0.0	0.9	2.9	23.3	49.4	8
1	56	1	165	64	81	174	401	149	39	25	...	0.0	8.5	0.0	0.0	0.0	0.2	2.1	20.4	38.8	6
2	54	0	172	95	138	163	386	185	102	96	...	0.0	9.5	-2.4	0.0	0.0	0.3	3.4	12.3	49.0	10
3	55	0	175	94	100	202	380	179	143	28	...	0.0	12.2	-2.2	0.0	0.0	0.4	2.6	34.6	61.6	1
4	75	0	190	80	88	181	360	177	103	-16	...	0.0	13.1	-3.6	0.0	0.0	-0.1	3.9	25.4	62.8	7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
447	53	1	160	70	80	199	382	154	117	-37	...	0.0	4.3	-5.0	0.0	0.0	0.7	0.6	-4.4	-0.5	1
448	37	0	190	85	100	137	361	201	73	86	...	0.0	15.6	-1.6	0.0	0.0	0.4	2.4	38.0	62.4	10
449	36	0	166	68	108	176	365	194	116	-85	...	0.0	16.3	-28.6	0.0	0.0	1.5	1.0	-44.2	-33.2	2
450	32	1	155	55	93	106	386	218	63	54	...	-0.4	12.0	-0.7	0.0	0.0	0.5	2.4	25.0	46.6	1
451	78	1	160	70	79	127	364	138	78	28	...	0.0	10.4	-1.8	0.0	0.0	0.5	1.6	21.3	32.8	1

452 rows × 280 columns

Sehingga untuk mempermudah pembacaan dan pemrosesan data, dataset akan disatukan terlebih dahulu dengan data kolomnya. Selain itu, secara sekila apabila melihat dataset yang digunakan, data tersebut mengandung banyak data kosong yang direpresentasikan dengan tanda tanya.



```
452 lines (452 sloc)  393 KB
```

```
75,0,190,80,91,193,371,174,121,-16,13,64,-2,?,63,0,52,44,0,0,32,0,0,0,0,0,0,0,44,20,36,0,28,0,0,0,0,0,52,40,0,0,0,
56,1,165,64,81,174,401,149,39,25,37,-17,31,?,53,0,48,0,0,0,24,0,0,0,0,0,0,0,64,0,0,0,24,0,0,0,0,0,0,32,24,0,0,0,40,0,
54,0,172,95,138,163,386,185,102,96,34,70,66,23,75,0,40,80,0,0,24,0,0,0,0,0,0,20,56,52,0,0,40,0,0,0,0,0,0,28,116,0,0,
55,0,175,94,100,202,380,179,143,28,11,-5,20,?,71,0,72,20,0,0,48,0,0,0,0,0,0,0,64,36,0,0,36,0,0,0,0,0,0,20,52,48,0,0,
75,0,190,80,88,181,360,177,103,-16,13,61,3,?,?,0,48,40,0,0,28,0,0,0,0,0,0,0,40,24,0,0,24,0,0,0,0,0,0,52,36,0,0,0,60,0,
13,0,169,51,100,167,321,174,91,107,66,52,88,?,84,0,36,48,0,0,20,0,0,0,0,0,0,20,44,36,0,0,44,0,0,0,0,0,0,24,64,0,0,0,4,
40,1,160,52,77,129,377,133,77,77,49,75,65,?,70,0,44,0,0,0,24,0,0,0,0,0,0,0,40,32,0,0,24,0,0,0,0,0,0,44,28,0,0,24,0,
49,1,162,54,78,0,376,157,70,67,7,8,51,?,67,0,44,36,0,0,24,0,0,0,0,0,0,0,52,32,0,0,28,0,0,0,0,0,0,56,28,0,0,24,0,0,
44,0,168,56,84,118,354,160,63,61,69,78,66,84,64,0,40,0,0,0,20,0,0,0,0,0,0,0,44,12,0,0,28,0,0,0,0,0,0,36,8,0,0,20,0,
50,1,167,67,89,130,383,156,73,85,34,70,71,?,63,0,44,40,0,0,28,0,0,0,0,0,0,0,56,24,0,0,32,0,0,0,0,0,0,72,0,0,0,28,0,
62,0,170,72,102,135,401,156,83,72,71,68,72,?,70,20,36,48,0,0,36,0,0,0,0,0,0,0,52,0,0,0,28,0,0,0,0,0,0,104,0,0,0,36,
45,1,165,86,77,143,373,150,65,12,37,49,26,?,72,0,40,28,0,0,20,0,0,0,0,0,0,0,40,20,0,0,20,0,0,0,0,0,0,32,44,0,0,0,36,0,
54,1,172,58,78,155,382,163,81,-24,42,41,-13,?,73,0,72,0,0,0,24,0,0,0,0,0,0,0,44,44,0,0,28,0,0,0,0,0,0,80,0,0,0,0,0,
30,0,170,73,91,180,355,157,104,68,51,60,63,?,56,0,92,0,0,0,32,0,0,0,0,0,0,28,48,20,0,0,52,0,0,0,0,0,0,36,40,0,0,0,52,
44,1,160,88,77,158,399,163,94,46,20,45,40,?,72,0,80,0,0,0,28,0,0,0,0,0,0,20,72,0,0,0,44,0,0,0,0,0,0,24,64,0,0,0,52,0,
47,1,150,48,75,132,350,169,65,36,45,68,40,?,76,0,48,0,0,0,24,0,0,0,0,0,0,0,44,28,0,0,28,0,0,0,0,0,0,40,40,0,0,0,24,0,
47,0,171,59,82,145,347,169,61,77,75,77,75,?,67,0,48,0,0,0,20,0,0,0,0,0,0,0,52,36,0,0,28,0,0,0,0,0,0,52,36,0,0,0,28,0,
46,1,158,58,70,120,353,122,52,57,49,-2,54,?,70,0,48,0,0,0,24,0,0,0,0,0,0,0,48,0,0,0,28,0,0,0,0,0,0,44,12,0,0,0,24,0,0,
73,0,165,63,91,154,392,175,83,73,-24,61,42,?,66,0,44,56,0,0,20,0,0,0,0,0,0,0,84,0,0,0,28,0,0,0,0,0,0,16,72,0,0,0,44,0,
```

Sehingga hal ini menjadi salah satu masalah yang harus diatasi dari data ini.

2. Ringkasan pra-pemrosesan data yang diusulkan

Dari masalah-masalah yang sudah dijelaskan diatas, berikut adalah ringkasan Langkah-langkah pra-pemrosesan data yang akan dilakukan.

- **Menggabungkan dataset dengan judul kolomnya.**

  Hal ini tentunya akan lebih mempermudah pra-pemrosesan data dan visualisasi data.

  Data yang semula seperti ini:

	0	1	2	3	4	5	6	7	8	9	...	270	271	272	273	274	275	276	277	278	279
0	75	0	190	80	91	193	371	174	121	-16	...	0.0	9.0	-0.9	0.0	0.0	0.9	2.9	23.3	49.4	8
1	56	1	165	64	81	174	401	149	39	25	...	0.0	8.5	0.0	0.0	0.0	0.2	2.1	20.4	38.8	6
2	54	0	172	95	138	163	386	185	102	96	...	0.0	9.5	-2.4	0.0	0.0	0.3	3.4	12.3	49.0	10
3	55	0	175	94	100	202	380	179	143	28	...	0.0	12.2	-2.2	0.0	0.0	0.4	2.6	34.6	61.6	1
4	75	0	190	80	88	181	360	177	103	-16	...	0.0	13.1	-3.6	0.0	0.0	-0.1	3.9	25.4	62.8	7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
447	53	1	160	70	80	199	382	154	117	-37	...	0.0	4.3	-5.0	0.0	0.0	0.7	0.6	-4.4	-0.5	1
448	37	0	190	85	100	137	361	201	73	86	...	0.0	15.6	-1.6	0.0	0.0	0.4	2.4	38.0	62.4	10
449	36	0	166	68	108	176	365	194	116	-85	...	0.0	16.3	-28.6	0.0	0.0	1.5	1.0	-44.2	-33.2	2
450	32	1	155	55	93	106	386	218	63	54	...	-0.4	12.0	-0.7	0.0	0.0	0.5	2.4	25.0	46.6	1
451	78	1	160	70	79	127	364	138	78	28	...	0.0	10.4	-1.8	0.0	0.0	0.5	1.6	21.3	32.8	1

452 rows × 280 columns

  Menjadi seperti ini:

	age	sex	height	weight	QRSduration	PRinterval	Q-Tinterval	Tinterval	Pinterval	QRS	...	chV6_QwaveAmp	chV6_Rwa
0	75	0	190	80	91	193	371	174	121	-16	...	0.0	
1	56	1	165	64	81	174	401	149	39	25	...	0.0	
2	54	0	172	95	138	163	386	185	102	96	...	0.0	
3	55	0	175	94	100	202	380	179	143	28	...	0.0	
4	75	0	190	80	88	181	360	177	103	-16	...	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
447	53	1	160	70	80	199	382	154	117	-37	...	0.0	
448	37	0	190	85	100	137	361	201	73	86	...	0.0	
449	36	0	166	68	108	176	365	194	116	-85	...	0.0	
450	32	1	155	55	93	106	386	218	63	54	...	-0.4	
451	78	1	160	70	79	127	364	138	78	28	...	0.0	

452 rows × 280 columns

- **Mengubah data kosong (?) menjadi data NaN.**

  Data kosong yang direpresentasikan menggunakan tanda tanya. Hal ini diketahui secara kasat mata ketika data divisualisasikan baik itu menggunakan grafik ataupun tidak menggunakan apa apa (raw data). Hal ini terbukti ketika pengecekan dilakukan.

```
columnContainsQuestionMark = df.columns[df.isin(['?']).any()]
columnContainsQuestionMark
```

✓ 0.7s

```
Index(['T', 'P', 'QRST', 'J', 'heartrate'], dtype='object')
```

Data kosong terdapat pada kolom T, P, QRST, J, dan heartrate. Ubah representasi data kosong menjadi numpy nan.

```
df = df.replace('?', np.nan)
```
✓ 0.7s Python

```
# check the data contains nan
def checkNan():
    for i in range(len(columnContainsQuestionMark)):
        print("column", columnContainsQuestionMark[i], "contains", df[columnContainsQuestionMark[i]].isna
```
✓ 0.6s Python

```
checkNan()
```
✓ 0.7s Python

```
column T contains 8 NaN value
column P contains 22 NaN value
column QRST contains 1 NaN value
column J contains 376 NaN value
column heartrate contains 1 NaN value
```

Proses ini bertujuan untunk mempermudah pra-pemrosesan data selanjutnya.

- **Handling data kosong menggunakan Mean Imputation.**

  Setelah data kosong dirubah menjadi data NaN, maka selanjutnya adalah proses empty data handling. Terdapat tiga cara untuk handle data tersebut, yaitu:

  1. Hapus records
  2. Mean/Mode/Median Imputation
  3. Model prediksi

  Pada studi kasus ini, cara yang digunakan adalah **Mean Imputation**. Cara tersebut dilakukan untuk menjaga jumlah data tetap sama seperti data awal yang tentunya akan berpengaruh pada performansi pembelajaran mesin.

```
imputer = SimpleImputer(missing_values = np.nan, strategy = 'mean')
imputer.fit([df['T'],df['P'],df['QRST'],df['J'],df['heartrate']])
df['T'],df['P'],df['QRST'],df['J'],df['heartrate'] = imputer.transform([df['T'],df['P'],df['QRST'],df['J'
```
✓ 0.7s Python

```
checkNan()
```
✓ 0.6s Python Python

```
column T contains 0 NaN value
column P contains 0 NaN value
column QRST contains 0 NaN value
column J contains 0 NaN value
column heartrate contains 0 NaN value
```

Setelah dilakukannya proses mean imputation, cek kembali apakah masih terdapat data kosong pada kolom tersebut. Hasilnya data sudah tidak ada yang NaN atau kosong.

- **Melihat kualitas data.**

  Jumlah data yang sangat banyak sangat rentan terhadap kesalahan pengukuran dan outliers seperti anomali objek. Untuk mengukur kualitas data, dapat menggunakan variansi dan covariansi.

```
    df.var()
17] ✓ 0.7s

age                271.149947
sex                  0.247959
height            1381.634181
weight             275.254729
QRSduration        236.064596
                       ...
chV6_PwaveAmp        0.120778
chV6_TwaveAmp        2.033624
chV6_QRSA          182.355902
chV6_QRSTA         342.025335
class               19.422503
Length: 280, dtype: float64
```

```
df.cov()
```

0.1s Python

	age	sex	height	weight	QRSduration	PRinterval	Q-Tinterval	Tinterval	Pinterval
age	271.149947	-0.484121	-66.995688	104.238835	-1.020034	30.384666	107.580274	15.052857	42.423631
sex	-0.484121	0.247959	-2.307816	-2.049708	-2.579087	-1.044370	1.197820	-3.277893	-1.042354
height	-66.995688	-2.307816	1381.634181	-46.225011	-3.614701	22.669677	-294.493741	-50.875110	27.863911
weight	104.238835	-2.049708	-46.225011	275.254729	25.514707	89.146886	65.722809	88.669442	51.704344
QRSduration	-1.020034	-2.579087	-3.614701	25.514707	236.064596	15.041010	112.171811	217.587956	19.714322
...	...	...	...	...	...	...	...	...	...
chV6_PwaveAmp	-0.216757	0.002459	0.874146	-0.266832	-0.350258	2.232765	-0.405750	0.608960	2.226764
chV6_TwaveAmp	-6.375520	0.046643	-0.449001	-3.423992	-4.867839	3.910994	-1.851220	-9.392874	0.591253
chV6_QRSA	4.012061	0.213081	-45.361018	13.954301	26.930022	-16.768074	115.503650	62.651220	-5.707433
chV6_QRSTA	-60.823721	0.429192	-63.404933	-15.550640	-23.524975	17.455537	93.218440	-9.509126	2.045573
class	-6.704109	-0.390803	1.089035	-6.591576	21.930597	-19.753341	4.164590	15.330936	-13.886388

280 rows × 280 columns

- **Drop kolom yang memiliki satu nilai.**

  Sebelum melakukan scaling, terdapat kolom yang hanya terdapat 1 nilai. Hal tersebut akan memberatkan proses perhitungan kedepannya dan solusinya adalah drop kolom tersebut. Kenapa? karena data dengan 1 nilai tidak memiliki korelasi apapun terhadap record yang ada. Contohnya adalah kolom 'chDI_SPwave'. Proses ini disebut dimentionality reduction.

```
singleValueColumn = checkUnuniqueColumn()
singleValueColumn
```

✓ 0.1s

```
['chDI_SPwave',
 'chAVL_SPwave',
 'chAVL_RRwaveExists',
 'chAVF_RPwaveExists',
 'chV4_RPwaveExists',
 'chV4_DD_RPwaveExists',
 'chV5_SPwave',
 'chV5_RRwaveExists',
 'chV5_RPwaveExists',
 'chV5_RTwaveExists',
 'chV6_SPwave',
 'chV6_DD_RPwaveExists',
 'chV6_RTwaveExists',
 'chDI_SPwaveAmp',
 'chAVL_SPwaveAmp',
 'chV5_SPwaveAmp',
 'chV6_SPwaveAmp']
```

Cek beberapa kolom tersebut apakah benar memiliki 1 nilai saja.

```
450    0
451    0
Name: chV5_SPwave, Length: 452, dtype: int64,
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
       ...
447    0.0
448    0.0
449    0.0
450    0.0
451    0.0
Name: chV6_SPwaveAmp, Length: 452, dtype: float64,
0      0
...
448    0
449    0
450    0
451    0
Name: chV5_RPwaveExists, Length: 452, dtype: int64)
```

Setelah dipastikan, maka drop kolom tersebut. Karena proses ini, dimensi dataset berkurang 17 kolom.

- **Scaling data.**

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Untuk scaling data, menggunakan min max scaling. Metode normalisasi min max mengubah kumpulan data menjadi skala mulai dari 0 hingga 1. Proses ini dilakukan untuk memperkecil angka dalam proses perhitungan, proses ini berpengaruh pada kecepatan pembelajaran mesin karena angka yang diproses bukan angka yang besar.

	age	sex	height	weight	QRSduration	PRinterval	Q-Tinterval	Tinterval	Pinterval	QRS	...	chV5_QRSTA	ch
0	0.903614	0.0	0.125926	0.435294	0.270677	0.368321	0.501805	0.241758	0.590244	0.457478	...	0.660574	
1	0.674699	1.0	0.088889	0.341176	0.195489	0.332061	0.610108	0.150183	0.190244	0.577713	...	0.558747	
2	0.650602	0.0	0.099259	0.523529	0.624060	0.311069	0.555957	0.282051	0.497561	0.785924	...	0.583812	
3	0.662651	0.0	0.103704	0.517647	0.338346	0.385496	0.534296	0.260073	0.697561	0.586510	...	0.687206	
4	0.903614	0.0	0.125926	0.435294	0.248120	0.345420	0.462094	0.252747	0.502439	0.457478	...	0.587467	
...	...	...	...	...	...	...	...	...	...	...	...	...	
447	0.638554	1.0	0.081481	0.376471	0.187970	0.379771	0.541516	0.168498	0.570732	0.395894	...	0.328982	
448	0.445783	0.0	0.125926	0.464706	0.338346	0.261450	0.465704	0.340659	0.356098	0.756598	...	0.762924	
449	0.433735	0.0	0.090370	0.364706	0.398496	0.335878	0.480144	0.315018	0.565854	0.255132	...	0.000000	
450	0.385542	1.0	0.074074	0.288235	0.285714	0.202290	0.555957	0.402930	0.307317	0.662757	...	0.650653	
451	0.939759	1.0	0.081481	0.376471	0.180451	0.242366	0.476534	0.109890	0.380488	0.586510	...	0.535248	

452 rows × 262 columns

Gambar diatas adalah hasil dari min max scaling pada dataset.

3. Menerapkan algoritma yang dipilih

Algoritma yang digunakan adalah ANN (Artificial Neural Network) / JST (Jaringan Syaraf Tiruan). JST (Jaringan Syaraf Tiruan) merupakan kecerdasan buatan yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibangun dengan prinsip dasar perambatan sinyal sehingga dapat mengenali pola pelatihan. Perceptron merupakan analogi dari Neuron pada otak manusia (sel syaraf buatan).

Gambaran sederhana perceptron
Input function, summirizing function, and actication function.

Input function:
Menerima variable inputan dan mengalikannya dengan bobot (W).

Activation function:
Mengolah data inputan menggunakan model matematika yang tepat.

Satu perceptron dapat digunakan untuk klasifikasi, prediksi dan kategorisasi.
Pada studi kasus ini, menggunakan library tensorflow untuk melakukan train.

```python
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(x_train.shape[1],)),
    tf.keras.layers.Dense(32, activation='leaky_relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(17, activation='softmax')
])
```

✓ 0.2s

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

✓ 0.7s

```python
model.fit(x_train, y_train, epochs=350, batch_size=20)
```

✓ 24.1s

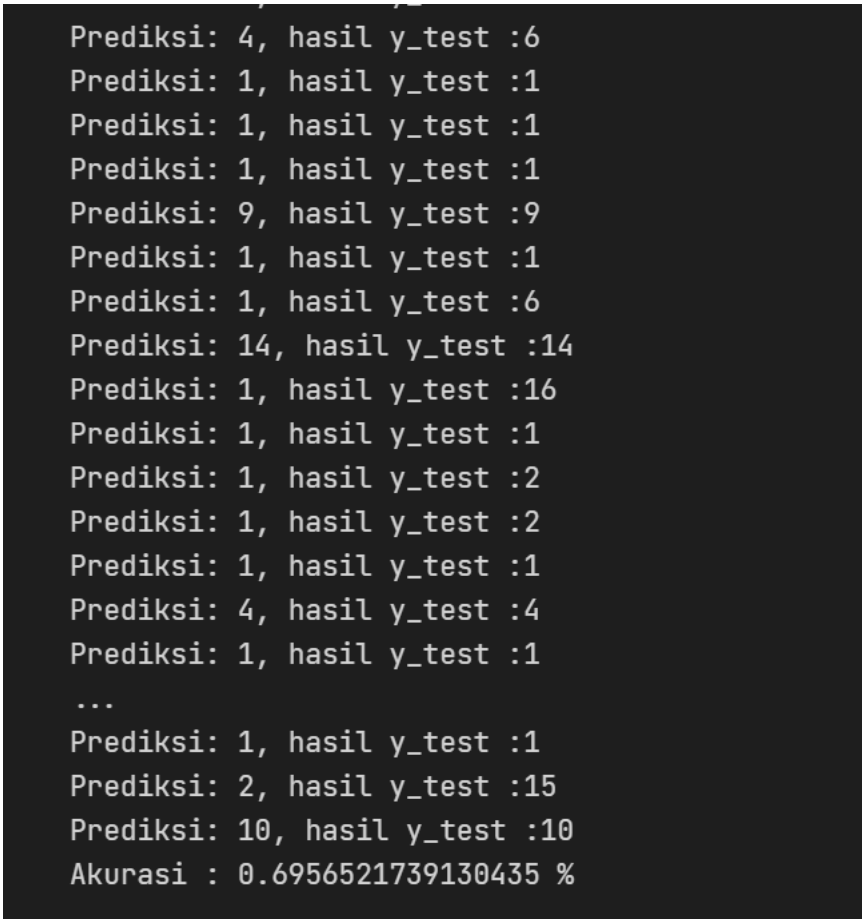
Model dilakukan sebanyak epochs 350 kali yang menghasilkan akurasi yang cukup tinggi pada data train.

```
Epoch 5/350
21/21 [==============================] - 0s 2ms/step - loss: 1.5756 - accuracy: 0.5542
Epoch 6/350
21/21 [==============================] - 0s 2ms/step - loss: 1.5811 - accuracy: 0.5567
Epoch 7/350
21/21 [==============================] - 0s 2ms/step - loss: 1.4943 - accuracy: 0.5567
Epoch 8/350
21/21 [==============================] - 0s 2ms/step - loss: 1.4672 - accuracy: 0.5665
Epoch 9/350
21/21 [==============================] - 0s 2ms/step - loss: 1.4303 - accuracy: 0.5640
Epoch 10/350
21/21 [==============================] - 0s 4ms/step - loss: 1.4120 - accuracy: 0.5862
Epoch 11/350
21/21 [==============================] - 0s 5ms/step - loss: 1.3555 - accuracy: 0.6133
Epoch 12/350
21/21 [==============================] - 0s 5ms/step - loss: 1.3255 - accuracy: 0.5985
Epoch 13/350
...
Epoch 349/350
21/21 [==============================] - 0s 3ms/step - loss: 0.1755 - accuracy: 0.9458
Epoch 350/350
21/21 [==============================] - 0s 2ms/step - loss: 0.1599 - accuracy: 0.9704

<keras.callbacks.History at 0x269f8aec070>
```

4. Evaluasi hasil

Kesimpulan diambil ketika data test digunakan pada model yang telah dilatih. Hasilnya adalah sebagai berikut.



Memiliki akurasi sekitar 69% pada data test. Hasil akurasi pada data test ini memiliki nilai yang lumayan baik sehingga dimulai dari pra-pemrosesan data sudah dilakukan dengan baik, data dilatih dan menghasilkan akurasi data test sekitar 60-70%.

5. Presentasi video

https://youtu.be/UOLSp6DpuyY

https://github.com/Hilmantm/tugas-machine-learning-case-based-1