AI menggunakan Deep Learning ANN Untuk Game

AFFA ALFIANDY || 50420056 || 1IA06

In [1]:

```
import pandas as pd
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
```

In [22]:

```
pilihan = ['Kertas','Gunting','Batu']
```

Membuat kumpulan data dari 1000 permainan

In [23]:

```
#Hasil akan menambahkan poin jika 2 player menang jika 0 komputer menang dan 1 jika ser
hasil = []
#Pilihan akan dimasukan ke sini
player = []
bot = []
gameset = []
game = 1
while game==1:
    angka = random.randint(1,4)
    print("Masukkan pilihan")
    print("1. Kertas")
    print("2. Gunting")
    print("3. Batu")
    inputan = random.randint(1,4)
    if inputan == 1 and pilihan[angka-1]=='Kertas':
        print("Seri")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(1)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 1 and pilihan[angka-1]=='Gunting':
        print("Anda Kalah")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(0)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 1 and pilihan[angka-1]=='Batu':
        print("Anda Menang!!")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(2)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 2 and pilihan[angka-1]=='Kertas':
        print("Anda Menang!!")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(2)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 2 and pilihan[angka-1]=='Gunting':
        print("Seri")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(1)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 2 and pilihan[angka-1]=='Batu':
        print("Anda Kalah!!")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(0)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 3 and pilihan[angka-1]=='Kertas':
        print("Anda Kalah")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(0)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 3 and pilihan[angka-1]=='Gunting':
        print("Anda Menang!!")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(2)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    elif inputan == 3 and pilihan[angka-1]=='Batu':
        print("Seri")
        print(pilihan[inputan-1]+' VS '+pilihan[angka-1])
        hasil.append(1)
        player.append(pilihan[inputan-1])
        bot.append(pilihan[angka-1])
        gameset.append(1)
    else:
        print("Pilih dengan benar!!!")
        continue

    if sum(gameset)==1000:
        break
    else:
        continue
```

```
Masukkan pilihan
1. Kertas
2. Gunting
3. Batu
Seri
Kertas VS Kertas
Masukkan pilihan
1. Kertas
2. Gunting
3. Batu
Anda Menang!!
Batu VS Gunting
Masukkan pilihan
1. Kertas
2. Gunting
3. Batu
Anda Kalah
Batu VS Kertas
Masukkan pilihan
1. Kertas
```

In [24]:

```
data = {"player":player,"bot":bot,"hasil":hasil}
```

In [25]:

```
df = pd.DataFrame(data)
```

In [26]:

```
df
```

Out[26]:

	player	bot	hasil
0	Kertas	Kertas	1
1	Batu	Gunting	2
2	Batu	Kertas	0
3	Kertas	Gunting	0
4	Kertas	Gunting	0
...	...	...	...
995	Kertas	Batu	2
996	Batu	Kertas	0
997	Batu	Batu	1
998	Gunting	Batu	0
999	Gunting	Batu	0

1000 rows × 3 columns

Merubah dari Kertas Gunting Batu menjadi 0,1,2

In [27]:

```
df['player']=df['player'].replace(['Kertas'],0)
df['player']=df['player'].replace(['Gunting'],1)
df['player']=df['player'].replace(['Batu'],2)

df['bot']=df['bot'].replace(['Kertas'],0)
df['bot']=df['bot'].replace(['Gunting'],1)
df['bot']=df['bot'].replace(['Batu'],2)
```

In [28]:

```
df
```

Out[28]:

	player	bot	hasil
0	0	0	1
1	2	1	2
2	2	0	0
3	0	1	0
4	0	1	0
...	...	...	...
995	0	2	2
996	2	0	0
997	2	2	1
998	1	2	0
999	1	2	0

1000 rows × 3 columns

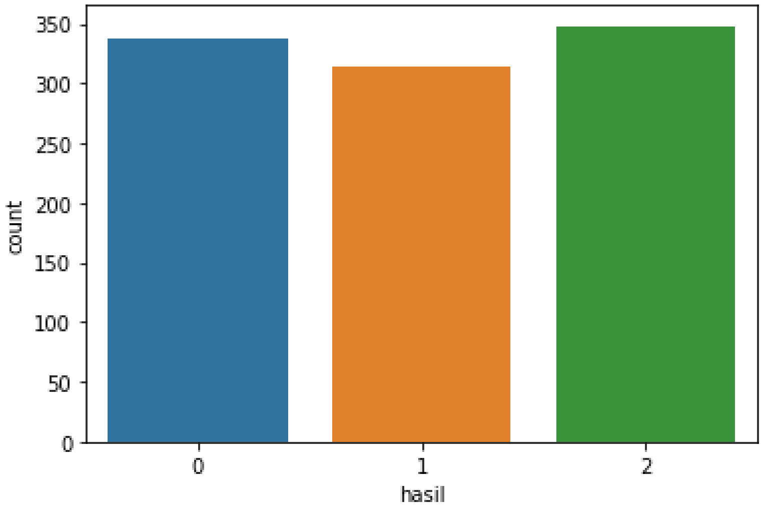
Melihat apakah penyebaran data cukup merata,0 untuk komputer menang 1 untuk seri dan 2 untuk pemain yang menang

In [59]:

```
sns.countplot(x="hasil",data=df)
```

Out[59]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e6d353bd08>
```



Melihat korelasi antar kolom pada data

In [30]:

```
df.corr()['hasil']
```

Out[30]:

```
player   -0.042275
bot       0.033639
hasil     1.000000
Name: hasil, dtype: float64
```

Memisahkan antara faktor inputan dan hasil

In [31]:

```
X = df.drop('hasil',axis=1).values
y = df['hasil'].values
```

In [32]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1
```

In [33]:

```
scaller = MinMaxScaler()
```

In [34]:

```
X_train = scaller.fit_transform(X_train)
X_test = scaller.transform(X_test)
```

Membuat model Jaringan Saraf Tiruan

In [35]:

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense,Dropout
```

In [36]:

```
X_train.shape
```

Out[36]:

```
(750, 2)
```

In [37]:

```
model = Sequential()
```

In [38]:

```
model.add(Dense(8,activation = 'tanh'))
model.add(Dense(1,activation = 'relu'))
model.compile(optimizer='adam',loss='mse')
```

In [39]:

```
model.fit(x=X_train,y=y_train,epochs=300,validation_data=(X_test,y_test))
```

```
- val_loss: 0.0564
Epoch 296/300
750/750 [==============================] - 0s 62us/sample - loss: 0.0594
- val_loss: 0.0544
Epoch 297/300
750/750 [==============================] - 0s 64us/sample - loss: 0.0577
- val_loss: 0.0530
Epoch 298/300
750/750 [==============================] - 0s 62us/sample - loss: 0.0562
- val_loss: 0.0516
Epoch 299/300
750/750 [==============================] - 0s 67us/sample - loss: 0.0546
- val_loss: 0.0502
Epoch 300/300
750/750 [==============================] - 0s 62us/sample - loss: 0.0533
- val_loss: 0.0491
```

Out[39]:

```
<tensorflow.python.keras.callbacks.History at 0x1e6cef0f988>
```

In [40]:

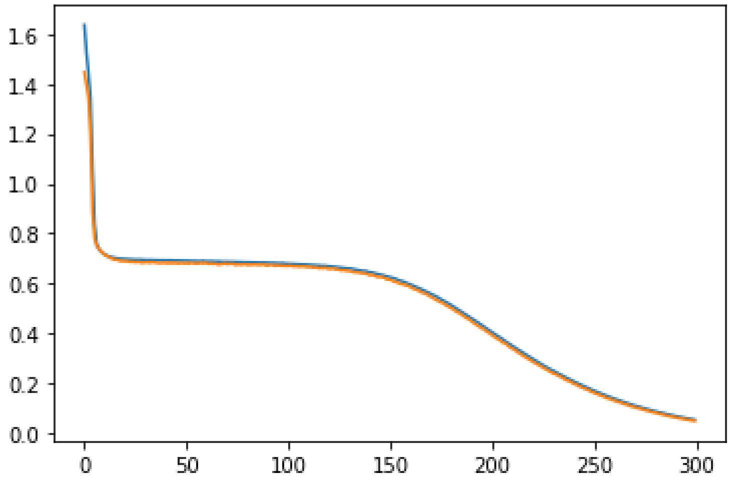
```
losses = pd.DataFrame(model.history.history)
```

In [41]:

```
plt.plot(losses)
```

Out[41]:

```
[<matplotlib.lines.Line2D at 0x1e6cf3b8f48>,
 <matplotlib.lines.Line2D at 0x1e6cf3c0e08>]
```



Melihat hasil jaringan saraf tiruan yang telah dibuat dan ditest untuk melihat nilai akhirnya, semakin kecil nilainya maka probabilitas komputer untuk menang semakin tinggi

In [50]:

```
#pilihan = ['Kertas','Gunting','Batu']
#pilihan = [0,1,2]
main = [2,1,0,2,0]
AI = [2,2,1,0,0]
testdata = {'player':main,"bot":AI}
testdata = pd.DataFrame(testdata)
testdata = scaller.transform(testdata.values.reshape(-1,2))
```

In [51]:

```
model.predict(testdata)
```

Out[51]:

```
array([[0.9824308 ],
       [0.23353866],
       [0.37170538],
       [0.228878  ],
       [0.977404  ]], dtype=float32)
```

In [57]:

```
main = []
AI1 = [0]
AI2 = [1]
AI3 = [2]

while True:
    print("Pilih")
    print("1. Kertas")
    print("2. Gunting")
    print("3. Batu")
    print("99. Selesai")
    userinput = int(input())
    userinput = userinput-1
    main.append(userinput)

    testdata1 = {'player':main,'bot':AI1}
    testdata1 = pd.DataFrame(testdata1)
    testdata1 = scaller.transform(testdata1.values.reshape(-1,2))

    testdata2 = {'player':main,'bot':AI2}
    testdata2 = pd.DataFrame(testdata2)
    testdata2 = scaller.transform(testdata2.values.reshape(-1,2))

    testdata3 = {'player':main,'bot':AI3}
    testdata3 = pd.DataFrame(testdata3)
    testdata3 = scaller.transform(testdata3.values.reshape(-1,2))

    if model.predict(testdata1) < model.predict(testdata2) and model.predict(testdata1)
        print(f" Anda : {pilihan[userinput]} vs Bot : {pilihan[0]}")
        main = []
        continue

    elif model.predict(testdata2) < model.predict(testdata1) and model.predict(testdata2
        print(f"Anda : {pilihan[userinput]} vs Bot : {pilihan[1]}")
        main = []
        continue

    elif model.predict(testdata3) < model.predict(testdata2) and model.predict(testdata3
        print(f"Anda : {pilihan[userinput]} vs Bot : {pilihan[2]}")
        main = []
        continue
    else:
        break
```

```
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
2
Anda : Gunting vs Bot : Batu
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
3
 Anda : Batu vs Bot : Kertas
```

```
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
2
Anda : Gunting vs Bot : Batu
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
1
Anda : Kertas vs Bot : Gunting
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
2
Anda : Gunting vs Bot : Batu
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
3
 Anda : Batu vs Bot : Kertas
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
2
Anda : Gunting vs Bot : Batu
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
1
Anda : Kertas vs Bot : Gunting
Pilih
1. Kertas
2. Gunting
3. Batu
99. Selesai
99
```

In [ ]: