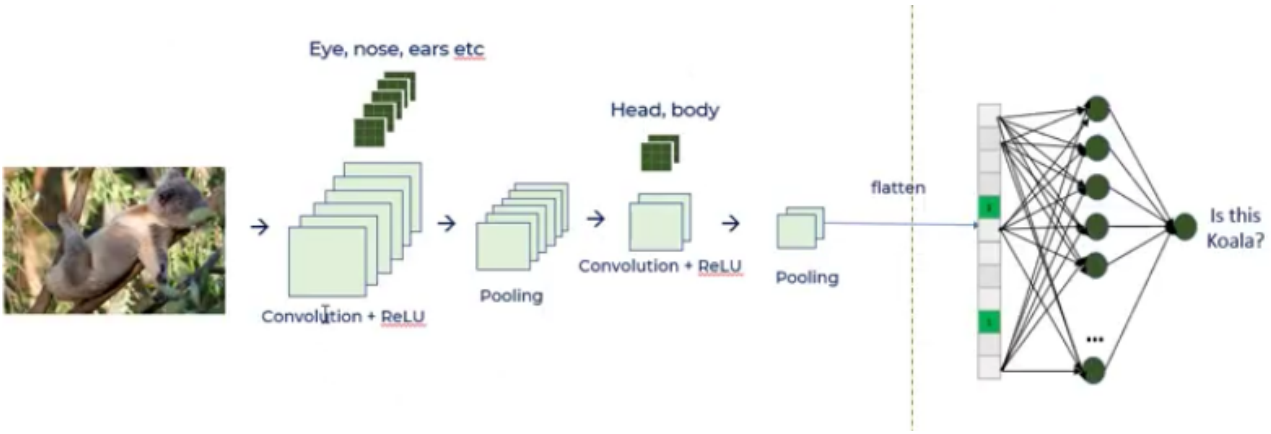
Nama : Andik Wijanarko
NIM: 33220006
Tugas : Eksplorasi Hyperparameter CNN dan Neural Network

A. CNN
Data : Cifar10
Data Type : Image
Size : 32 x 32 Pixels
Channel size: 3 (RGB)
`Jumlah data : 50.000 images`
Jumlah Class : 10
`['pesawat','mobil','burung','kucing','rusa','anjing','katak','kuda','kapal', 'truk']`

Skema CNN



Berapa banyaknya convolution layar yang optimal? **Jawab : 2**
Berapa ukuran filter yang optimal untuk setiap convolution layar? **Jawab : 3 x 3**
Berapa banyaknya filter yang optimal untuk setiap convolution layar? **Jawab :**
Filter Convolution 1 : 32
Filter Convolution 2 : 64
Berapa banyaknya hidden unit yang optimal pada bagian fully connected network? **Jawab : 64**

Dari semua pilihan yang disedikan oleh Keras Optimizer, mana yang menghasilkan kinerja paling baik (pada nilai parameter default) ? **Jawab : SGD**

Dari semua pilihan yang disedaikan oleh Keras (Probabilistic) Losses, mana yang menghasilkan kinerja paling baik?
Jawab: `sparse_categorical_crossentropy`

Eksplorasi 1
Jumlah Convolution layer : 1
Jumlah Maxpooling : 1
ukuran filter : 32 / convolutional layer
filter size : 3 x 3
fungsi aktivasi : Relu
banyaknya hidden unit fully connected network : 64

hasil explorasi 1

```
[20] cnn.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

```
cnn.fit(X_train,y_train,epochs=5)

Epoch 1/5
1563/1563 [==============================] - 8s 5ms/step - loss: 1.5446 - accuracy: 0.4447
Epoch 2/5
1563/1563 [==============================] - 7s 5ms/step - loss: 1.2429 - accuracy: 0.5621
Epoch 3/5
1563/1563 [==============================] - 7s 5ms/step - loss: 1.1341 - accuracy: 0.6024
Epoch 4/5
1563/1563 [==============================] - 7s 5ms/step - loss: 1.0609 - accuracy: 0.6315
Epoch 5/5
1563/1563 [==============================] - 7s 5ms/step - loss: 1.0021 - accuracy: 0.6516
<keras.callbacks.History at 0x7fcb1024d750>
```

eksplorasi 2
Jumlah Convolution layer : 2
Jumlah Maxpooling : 2
ukuran filter : 32 / convolutional layer
filter size : 3 x 3
fungsi aktivasi : relu
banyaknya hidden unit fully connected network : 64

hasil explorasi 2

```
cnn.fit(X_train,y_train,epochs=5)

Epoch 1/5
1563/1563 [==============================] - 9s 5ms/step - loss: 1.5039 - accuracy: 0.4602
Epoch 2/5
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1937 - accuracy: 0.5804
Epoch 3/5
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0770 - accuracy: 0.6229
Epoch 4/5
1563/1563 [==============================] - 8s 5ms/step - loss: 0.9965 - accuracy: 0.6524
Epoch 5/5
1563/1563 [==============================] - 8s 5ms/step - loss: 0.9324 - accuracy: 0.6750
<keras.callbacks.History at 0x7fcb10171ad0>
```

eksplorasi 3
Jumlah Convolution layer : 2
Jumlah Maxpooling : 2
ukuran filter : 64 / convolutional layer
filter size : 3 x 3
fungsi aktivasi : relu
banyaknya hidden unit fully connected network : 64

```
cnn.fit(X_train,y_train,epochs=5)
```

```
Epoch 1/5
1563/1563 [==============================] - 12s 7ms/step - loss: 1.4649 - accuracy: 0.4730
Epoch 2/5
1563/1563 [==============================] - 12s 7ms/step - loss: 1.1239 - accuracy: 0.6044
Epoch 3/5
1563/1563 [==============================] - 12s 7ms/step - loss: 0.9965 - accuracy: 0.6521
Epoch 4/5
1563/1563 [==============================] - 12s 7ms/step - loss: 0.9155 - accuracy: 0.6796
Epoch 5/5
1563/1563 [==============================] - 12s 8ms/step - loss: 0.8537 - accuracy: 0.6995
<keras.callbacks.History at 0x7fcb100da110>
```

eksplorasi 4
Jumlah Convolution layer : 2
Jumlah Maxpooling : 2
ukuran filter 1 : 32 / convolutional layer
ukuran filter 2 : 64 / convolutional layer
filter size : 3 x 3
fungsi aktivasi : relu

```
[29] cnn.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

```
cnn.fit(X_train,y_train,epochs=5)
```

```
Epoch 1/5
1563/1563 [==============================] - 10s 6ms/step - loss: 1.4723 - accuracy: 0.4722
Epoch 2/5
1563/1563 [==============================] - 10s 6ms/step - loss: 1.1192 - accuracy: 0.6104
Epoch 3/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.9808 - accuracy: 0.6570
Epoch 4/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.8934 - accuracy: 0.6890
Epoch 5/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.8198 - accuracy: 0.7161
<keras.callbacks.History at 0x7fca96e2e190>
```

+ Kode + Teks

banyaknya hidden unit fully connected network : 64

eksplorasi 5
Jumlah Convolution layer : 3
Jumlah Maxpooling : 3
ukuran filter 1 : 32 / convolutional layer
ukuran filter 2 : 32 / convolutional layer
ukuran filter 3 : 64 / convolutional layer
filter size : 3 x 3
fungsi aktivasi : relu
banyaknya hidden unit fully connected network : 64

```
cnn.fit(X_train,y_train,epochs=5)
```

```
Epoch 1/5
1563/1563 [==============================] - 10s 6ms/step - loss: 1.6153 - accuracy: 0.4077
Epoch 2/5
1563/1563 [==============================] - 9s 6ms/step - loss: 1.2745 - accuracy: 0.5467
Epoch 3/5
1563/1563 [==============================] - 9s 6ms/step - loss: 1.1325 - accuracy: 0.5995
Epoch 4/5
1563/1563 [==============================] - 9s 6ms/step - loss: 1.0358 - accuracy: 0.6361
Epoch 5/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.9603 - accuracy: 0.6646
<keras.callbacks.History at 0x7fcb100d1590>
```

eksplorasi 6
Jumlah Convolution layer : 2
Jumlah Maxpooling : 2
ukuran filter 1 : 32 / convolutional layer
ukuran filter 2 : 64 / convolutional layer
filter size : 3 x 3
fungsi aktivasi : relu
Optimizer : SGD

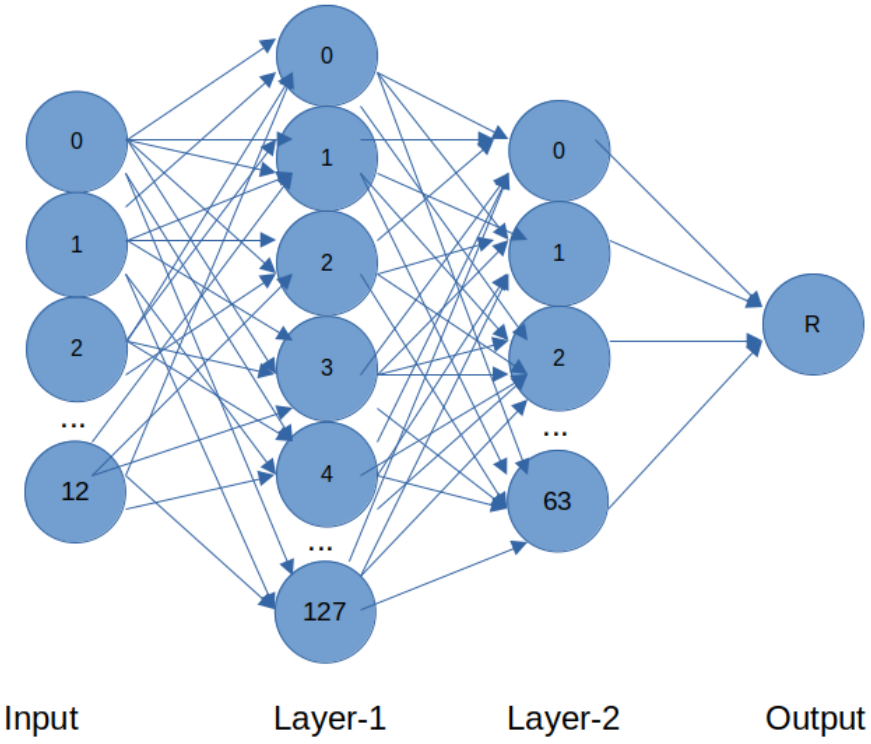
```
cnn.compile(optimizer='SGD',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

```
cnn.fit(X_train,y_train,epochs=5)
```

```
Epoch 1/5
1563/1563 [==============================] - 10s 6ms/step - loss: 0.7840 - accuracy: 0.7305
Epoch 2/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.7032 - accuracy: 0.7572
Epoch 3/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.6784 - accuracy: 0.7660
Epoch 4/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.6615 - accuracy: 0.7721
Epoch 5/5
1563/1563 [==============================] - 9s 6ms/step - loss: 0.6470 - accuracy: 0.7782
<keras.callbacks.History at 0x7fca9689f950>
```

B. Fully Connected NN
Data : Boston Housing Price
Data Type : Tabel numerik (float)
Jumlah Kolom : 13
Jumlah baris : 506

skema NN



Hasil Ekplorasi
Berapa banyaknya hidden layar yang optimal? **Jawab: 2**
Berapa banykanya hidden unit yang optimal di setiap hidden layar? **Jawab: 128 dan 64**
Apa activation function di setiap layaer sehingga hasilnya optimal? **Jawab : Relu**
Dari semua pilihan optimer, apa optimizer yang hasilnya optimal? **Jawab : Adam**
Dari semua pilihan loss function, apa yang hasilnya optimal?
Jawab = Mean Absolute Error (MAE)

Detail Eksplorasi

Eksplorasi 1

```
[136] def HousePricePredictionModel():
        model=Sequential()
        model.add(Dense(128,activation='relu',input_shape=(train_x[0].shape)))
        model.add(Dense(64,activation='relu'))
        model.add(Dense(32,activation='relu'))
        model.add(Dense(16,activation='relu'))
        model.add(Dense(8,activation='relu'))
        model.add(Dense(4,activation='relu'))
        model.add(Dense(1))
        model.compile(optimizer='Adam',loss='mse', metrics=['mae'])
        return model
```

```
Epoch 100/100
404/404 [==============================] - 2s 4ms/step - loss: 16.1819 - mae: 2.9084 - val_loss: 29.8165 - val_mae: 3.5819
<keras.callbacks.History at 0x7f52f5b98250>
```

Eksplorasi 2

```
def HousePricePredictionModel():
        model=Sequential()
        model.add(Dense(128,activation='relu',input_shape=(train_x[0].shape)))
        model.add(Dense(64,activation='relu'))
        model.add(Dense(32,activation='relu'))
        model.add(Dense(16,activation='relu'))
        model.add(Dense(8,activation='relu'))
        model.add(Dense(1))
        model.compile(optimizer='Adam',loss='mse', metrics=['mae'])
        return model
```

```
Epoch 99/100
404/404 [==============================] - 1s 3ms/step - loss: 14.5748 - mae: 2.8035 - val_loss: 37.0248 - val_mae: 4.0469
Epoch 100/100
404/404 [==============================] - 1s 3ms/step - loss: 15.2515 - mae: 2.8645 - val_loss: 25.8637 - val_mae: 3.4836
```

Eksplorasi 3

```
[142] def HousePricePredictionModel():
        model=Sequential()
        model.add(Dense(128,activation='relu',input_shape=(train_x[0].shape)))
        model.add(Dense(64,activation='relu'))
        model.add(Dense(32,activation='relu'))
        model.add(Dense(16,activation='relu'))
        model.add(Dense(1))
        model.compile(optimizer='Adam',loss='mse', metrics=['mae'])
        return model
```

```
Epoch 95/100
404/404 [==============================] - 2s 4ms/step - loss: 15.5132 - mae: 2.8745 - val_loss: 30.8432 - val_mae: 3.7887
Epoch 96/100
404/404 [==============================] - 1s 4ms/step - loss: 15.4672 - mae: 2.8593 - val_loss: 29.3681 - val_mae: 3.6561
Epoch 97/100
404/404 [==============================] - 1s 3ms/step - loss: 13.7625 - mae: 2.7412 - val_loss: 28.2854 - val_mae: 3.5518
Epoch 98/100
404/404 [==============================] - 1s 3ms/step - loss: 15.8283 - mae: 2.9117 - val_loss: 29.8397 - val_mae: 3.6351
Epoch 99/100
404/404 [==============================] - 1s 3ms/step - loss: 16.1588 - mae: 2.8430 - val_loss: 26.5212 - val_mae: 3.4705
Epoch 100/100
404/404 [==============================] - 1s 3ms/step - loss: 16.0148 - mae: 2.8513 - val_loss: 33.0665 - val_mae: 4.0061
<keras.callbacks.History at 0x7f52f5841310>
```

Eksplorasi 4

```
[145] def HousePricePredictionModel():
        model=Sequential()
        model.add(Dense(128,activation='relu',input_shape=(train_x[0].shape)))
        model.add(Dense(64,activation='relu'))
        model.add(Dense(32,activation='relu'))
        model.add(Dense(1))
        model.compile(optimizer='Adam',loss='mse', metrics=['mae'])
        return model
```

```
Epoch 95/100
404/404 [==============================] - 1s 3ms/step - loss: 14.4224 - mae: 2.7463 - val_loss: 29.2971 - val_mae: 3.6206
Epoch 96/100
404/404 [==============================] - 1s 3ms/step - loss: 15.5237 - mae: 2.8958 - val_loss: 28.7757 - val_mae: 3.5866
Epoch 97/100
404/404 [==============================] - 1s 3ms/step - loss: 15.8912 - mae: 2.8684 - val_loss: 32.9450 - val_mae: 3.8845
Epoch 98/100
404/404 [==============================] - 1s 4ms/step - loss: 17.3785 - mae: 2.8855 - val_loss: 31.2902 - val_mae: 3.7443
Epoch 99/100
404/404 [==============================] - 1s 4ms/step - loss: 16.5671 - mae: 2.9186 - val_loss: 28.8252 - val_mae: 3.6248
Epoch 100/100
404/404 [==============================] - 1s 4ms/step - loss: 14.8916 - mae: 2.7747 - val_loss: 28.6249 - val_mae: 3.7039
<keras.callbacks.History at 0x7f52f5597910>
```

Eksplorasi 5

```
[151] def HousePricePredictionModel():
        model=Sequential()
        model.add(Dense(128,activation='relu',input_shape=(train_x[0].shape)))
        model.add(Dense(64,activation='relu'))
        model.add(Dense(1))
        model.compile(optimizer='Adam',loss='mse', metrics=['mae'])
        return model
```

```
Epoch 95/100
404/404 [==============================] - 1s 3ms/step - loss: 15.1599 - mae: 2.7534 - val_loss: 35.3469 - val_mae: 4.1609
Epoch 96/100
404/404 [==============================] - 1s 3ms/step - loss: 14.7243 - mae: 2.8228 - val_loss: 33.5135 - val_mae: 4.0289
Epoch 97/100
404/404 [==============================] - 1s 3ms/step - loss: 15.1145 - mae: 2.8924 - val_loss: 29.2708 - val_mae: 3.6973
Epoch 98/100
404/404 [==============================] - 1s 3ms/step - loss: 15.5098 - mae: 2.8844 - val_loss: 29.8191 - val_mae: 3.8377
Epoch 99/100
404/404 [==============================] - 1s 3ms/step - loss: 15.0095 - mae: 2.7845 - val_loss: 32.4882 - val_mae: 3.8248
Epoch 100/100
404/404 [==============================] - 1s 3ms/step - loss: 15.8758 - mae: 2.8643 - val_loss: 27.6126 - val_mae: 3.5228
<keras.callbacks.History at 0x7f52f5447790>
```