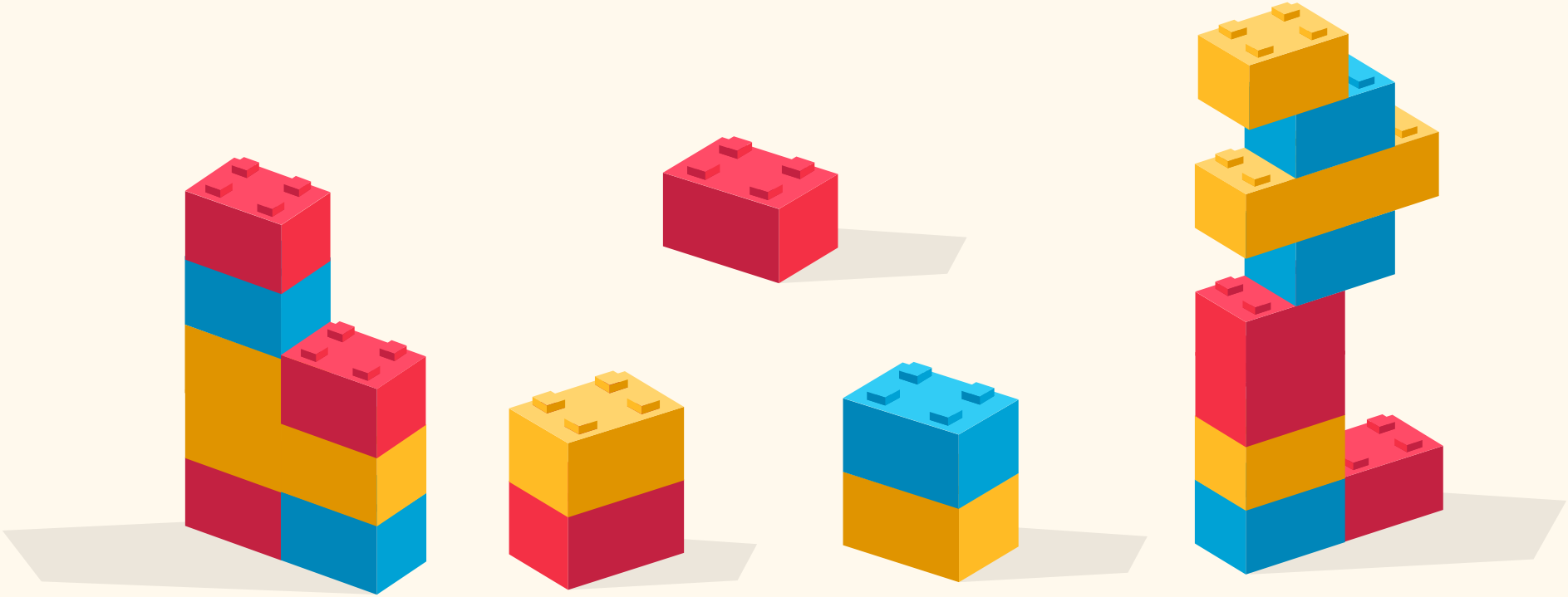
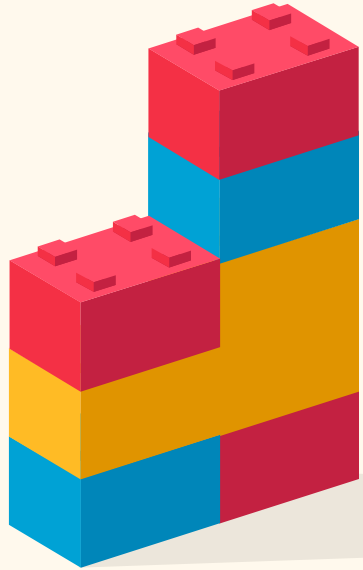
PENJELASAN PROGRAM

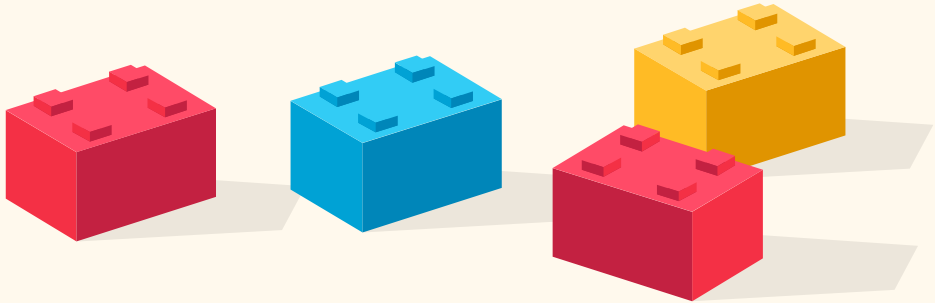
TENSORFLOW KERAS



NAMA KELOMPOK

1. MUH DENI SETIAWAN
2. ATHAUR MUTTAQIN
3. HARI NOVRIANSYAH
4. M.FIRAS RIZALDIANSYAH





projects:
ATHAUR GANTENG

NAMA KELOMPOK

01

MUH DENI SETIAWAN

20TI039

02

ATHAUR MUTTAQIN

20TI021

03

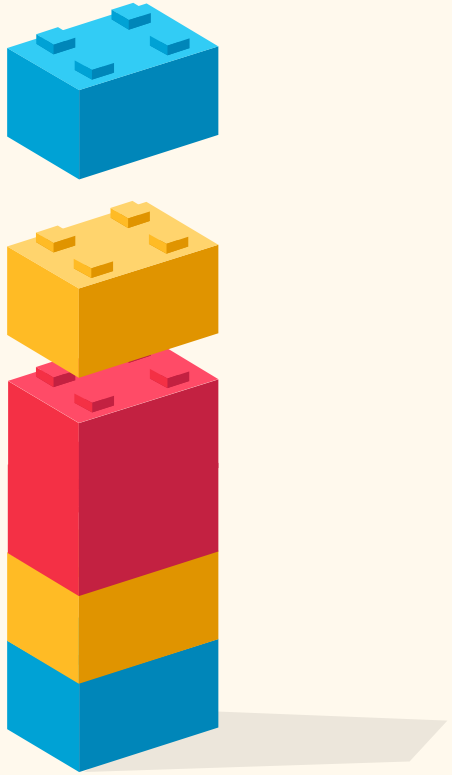
HARI NOVRIANSYAH

20TI027

04

M.FIRAS RIZALDIANSYAH

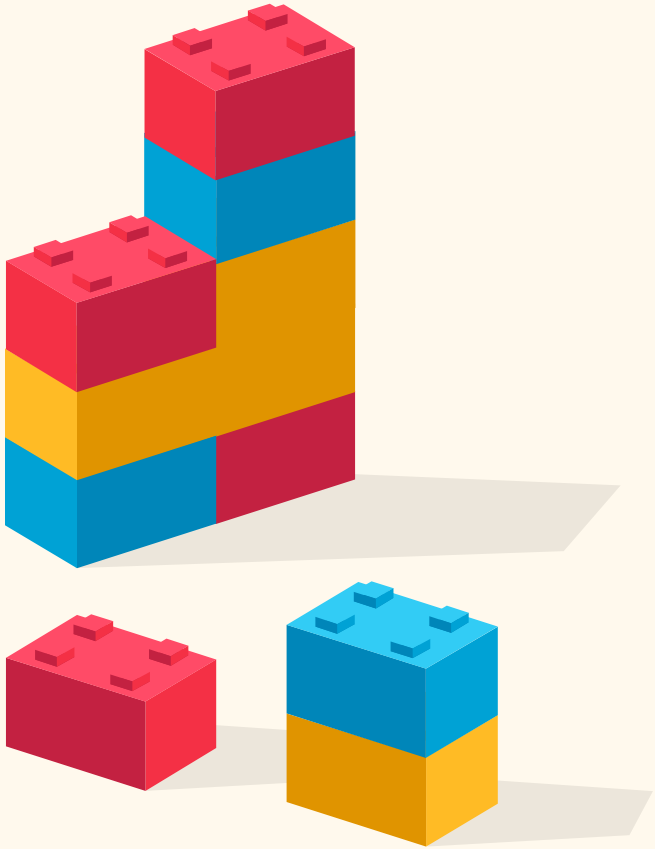
20TI036



01

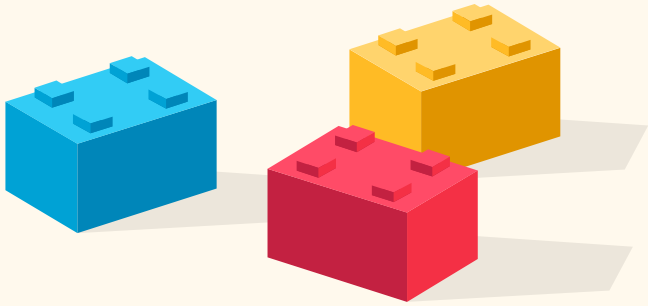
```
[ ]  import os
     os.environ['TF_CPP_MIN_LOG_LEVEL']= '2'

     import tensorflow as tf
     from tensorflow import keras
     import numpy as np
     import matplotlib.pyplot as plt
```



Import library berfungsi untuk mengambil/memasukan modul seperti : tensorflow, numpy dan matplotib, agar bisa di gunakan

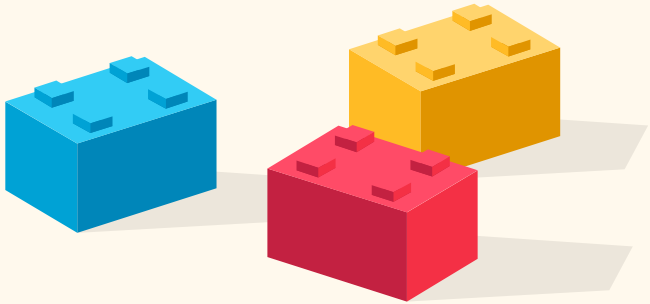
02



```
[ ]  mnist = keras.datasets.mnist

     (x_train, y_train), (x_test, y_test) = mnist.load_data()
     print(x_train.shape, y_train.shape)
```

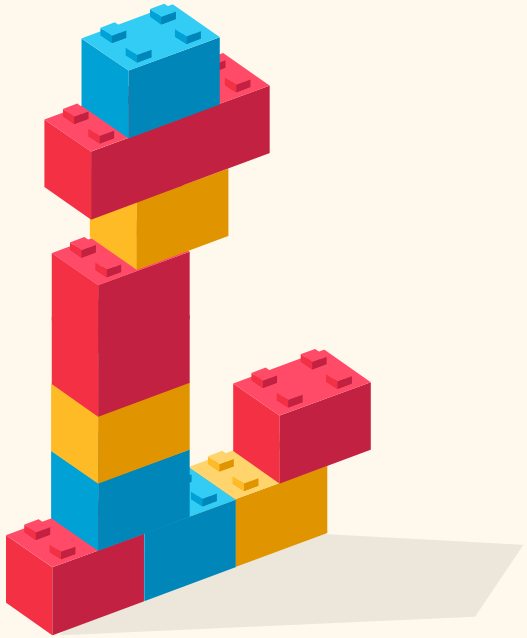
Melakukan load dataset dari keras serta melakukan training dan testing terhadap data yang berasal dari dataset Mnist.



```
[ ]  model = keras.models.Sequential([
         keras.layers.Flatten(input_shape=(28,28)),
         keras.layers.Dense(128, activation='relu'),
         keras.layers.Dense(10),
     ])
     print(model.summary())
```

03

Kode di atas berfungsi untuk penentuan keras model, dimana keras model dapat di definisikan sebagai urutan lapisan dimana kita membuat Model sequential untuk menambahkan lapisan satu persatu, yang dimana di sini terdiri dari 3 lapisan, yang di antaranya menggunakan kelas Flatten dan Dense, kelas Flatten menggunakan input shape 28,28, pada kelas Dense ada dua yaitu pada lapisan dua dan tiga, pada lapisan kedua menggunakan 128 neuron dan fungsi aktifasi relu sedangkan pada lapisan ketiga tidak menggunakan fungsi aktifasi relu .

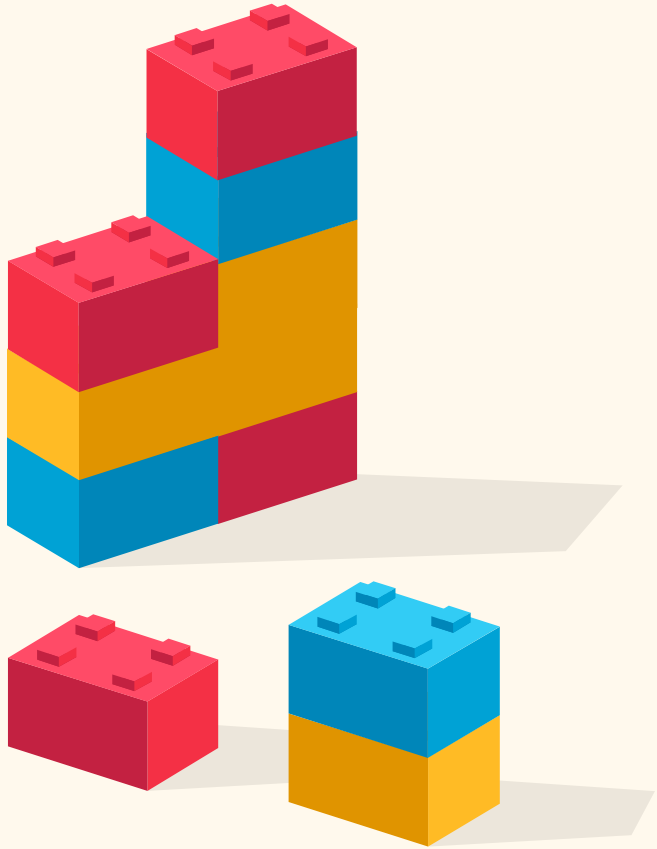


04

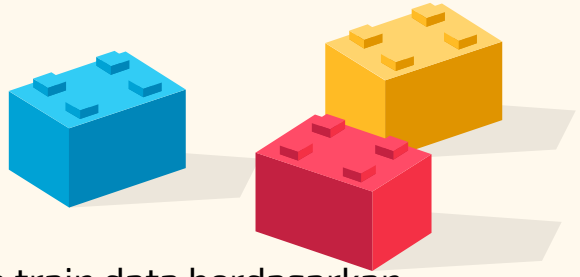
```
[ ]  loss = keras.losses.SparseCategoricalCrossentropy(from_logits=True)
     optim = keras.optimizers.Adam(lr=0.001)
     metrics = ["accuracy"]

     model.compile(loss=loss, optimizer=optim, metrics=metrics)
```

Setelah memodelkan keras, selanjutnya kita mengkompile keras model dengan menggunakan koment model.compile, dimana dari loss ini menggunakan SparseCategoricalCrossentropy , dengan optimizer adam, karena kita menggunakan klasifikasi tentu kita menggunakan accuracy, dimana kita simpan nilai accuracy nya pada metrics



05



```
[22] batch_size = 64
     epochs = 10

     model.fit(x_train, y_train, batch_size=batch_size,
               epochs=epochs, shuffle=True, verbose=2)
```

```
Epoch 1/10
938/938 - 3s - loss: 0.2970 - accuracy: 0.9161 - 3s/epoch - 3ms/step
Epoch 2/10
938/938 - 2s - loss: 0.1370 - accuracy: 0.9606 - 2s/epoch - 2ms/step
Epoch 3/10
938/938 - 2s - loss: 0.0951 - accuracy: 0.9722 - 2s/epoch - 2ms/step
Epoch 4/10
938/938 - 2s - loss: 0.0734 - accuracy: 0.9779 - 2s/epoch - 2ms/step
Epoch 5/10
938/938 - 2s - loss: 0.0576 - accuracy: 0.9830 - 2s/epoch - 2ms/step
Epoch 6/10
938/938 - 2s - loss: 0.0457 - accuracy: 0.9864 - 2s/epoch - 2ms/step
Epoch 7/10
938/938 - 3s - loss: 0.0367 - accuracy: 0.9893 - 3s/epoch - 3ms/step
Epoch 8/10
938/938 - 2s - loss: 0.0305 - accuracy: 0.9905 - 2s/epoch - 2ms/step
Epoch 9/10
938/938 - 2s - loss: 0.0260 - accuracy: 0.9923 - 2s/epoch - 2ms/step
Epoch 10/10
938/938 - 2s - loss: 0.0212 - accuracy: 0.9936 - 2s/epoch - 2ms/step
<keras.callbacks.History at 0x7f6d63c2e070>
```

Selanjutnya kita train data berdasarkan dataset dengan menggunakan perintah model.fit, dengan epochs = 5 ,batch size = 64, dan verbose = 2.

Kemudian Setelah dataset nya di jalankan atau di tarin, selanjutnya kita cari accuracynya dengan menggunakan mode.evaluate dengan variable x dan y, accuracy yang di dapatkan adalah 0.9766 Dengan loss sebanyak 0.0751.

Dan terakhir melakukan perdiksi.