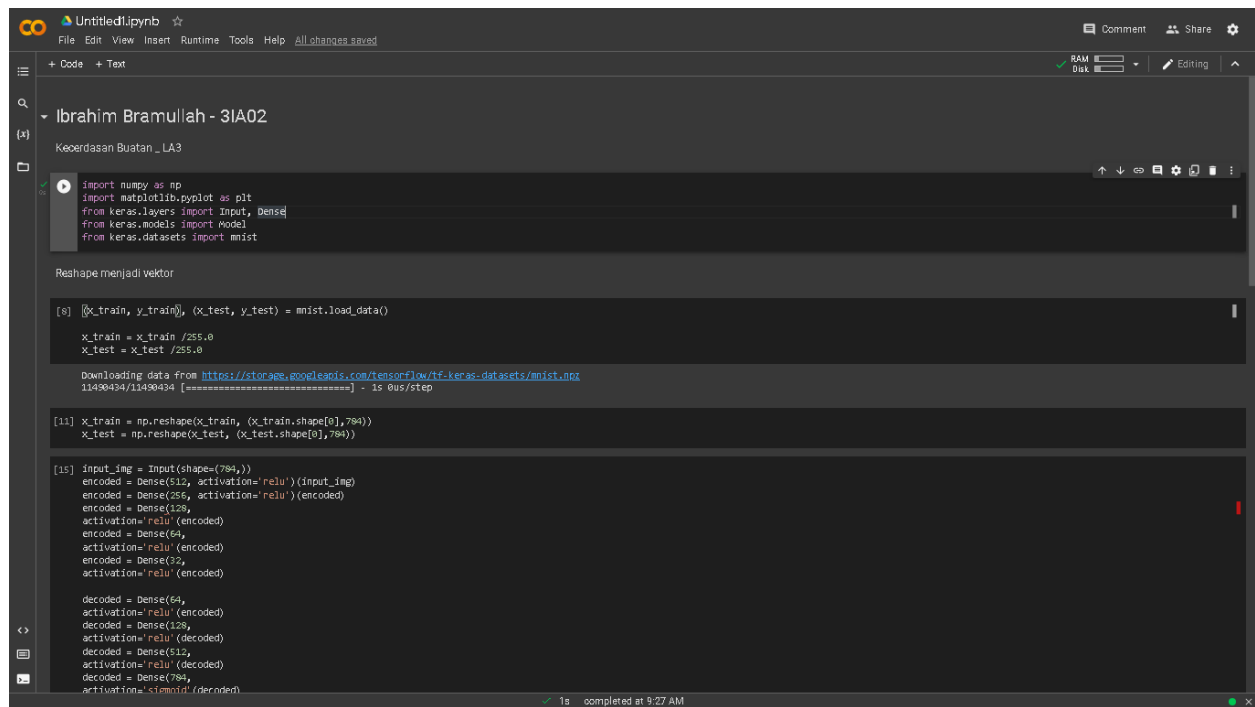


Ibrahim Bramullah

50420562 – 3IA02



The screenshot shows a Jupyter Notebook titled 'Untitled1.ipynb' with a dark theme. The notebook contains the following code cells:

```
import numpy as np
import matplotlib.pyplot as plt
from keras.layers import Input, Dense
from keras.models import Model
from keras.datasets import mnist
```

Reshape menjadi vektor

```
[8] (x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train / 255.0
x_test = x_test / 255.0
```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>
11490434/11490434 [=====] - 1s 0us/step

```
[11] x_train = np.reshape(x_train, (x_train.shape[0], 784))
    x_test = np.reshape(x_test, (x_test.shape[0], 784))
```

```
[15] input_img = Input(shape=(784,))
    encoded = Dense(512, activation='relu')(input_img)
    encoded = Dense(256, activation='relu')(encoded)
    encoded = Dense(128, activation='relu')(encoded)
    encoded = Dense(64, activation='relu')(encoded)
    encoded = Dense(32, activation='relu')(encoded)

    decoded = Dense(64, activation='relu')(encoded)
    decoded = Dense(128, activation='relu')(decoded)
    decoded = Dense(256, activation='relu')(decoded)
    decoded = Dense(512, activation='relu')(decoded)
    decoded = Dense(784, activation='sigmoid')(decoded)
```

The status bar at the bottom indicates '1s completed at 9:27 AM'.

--1

numpy > perhitungan

matplotlib > visualiasi

layer > dense

models > model

datasets > mnist

--2

load data dari dataset

Dinormalisasikan > matriksnya dari 0 1

```
Untitled Jupyter Notebook
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[11] x_train = np.reshape(x_train, (x_train.shape[0], 784))
     x_test = np.reshape(x_test, (x_test.shape[0], 784))

[15] input_img = Input(shape=(784,))
     encoded = Dense(512, activation='relu')(input_img)
     encoded = Dense(256, activation='relu')(encoded)
     encoded = Dense(128, activation='relu')(encoded)
     encoded = Dense(64, activation='relu')(encoded)
     encoded = Dense(32, activation='relu')(encoded)

     decoded = Dense(64, activation='relu')(encoded)
     decoded = Dense(128, activation='relu')(decoded)
     decoded = Dense(256, activation='relu')(decoded)
     decoded = Dense(512, activation='relu')(decoded)
     decoded = Dense(784, activation='sigmoid')(decoded)

     autoencoder = Model(input_img, decoded)

[17] autoencoder.compile(optimizer='adam', loss='mean_squared_error')
     autoencoder.fit(x_train, x_train,
                    epochs=10,
                    batch_size=100,
                    shuffle=True,
                    validation_data=(x_test, x_test))

Epoch 1/10
600/600 [=====] - 18s 28ms/step - loss: 0.0317 - val_loss: 0.0193
Epoch 2/10
600/600 [=====] - 16s 27ms/step - loss: 0.0178 - val_loss: 0.0158
Epoch 3/10
600/600 [=====] - 17s 28ms/step - loss: 0.0153 - val_loss: 0.0142
Epoch 4/10
600/600 [=====] - 16s 26ms/step - loss: 0.0137 - val_loss: 0.0131
Epoch 5/10
1s completed at 9:27 AM
```

Reshape > Vektor

Encode > Dense

Compile dengan autoencoder

```
Untitled Jupyter Notebook
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
autoencoder.fit(x_train, x_train,
               epochs=10,
               batch_size=100,
               shuffle=True,
               validation_data=(x_test, x_test))

Epoch 1/10
600/600 [=====] - 18s 28ms/step - loss: 0.0317 - val_loss: 0.0193
Epoch 2/10
600/600 [=====] - 16s 27ms/step - loss: 0.0178 - val_loss: 0.0158
Epoch 3/10
600/600 [=====] - 17s 28ms/step - loss: 0.0153 - val_loss: 0.0142
Epoch 4/10
600/600 [=====] - 16s 26ms/step - loss: 0.0137 - val_loss: 0.0131
Epoch 5/10
600/600 [=====] - 17s 28ms/step - loss: 0.0127 - val_loss: 0.0123
Epoch 6/10
600/600 [=====] - 16s 27ms/step - loss: 0.0121 - val_loss: 0.0117
Epoch 7/10
600/600 [=====] - 16s 26ms/step - loss: 0.0116 - val_loss: 0.0114
Epoch 8/10
600/600 [=====] - 16s 27ms/step - loss: 0.0112 - val_loss: 0.0111
Epoch 9/10
600/600 [=====] - 16s 27ms/step - loss: 0.0109 - val_loss: 0.0113
Epoch 10/10
600/600 [=====] - 16s 27ms/step - loss: 0.0104 - val_loss: 0.0106
(keras.callbacks.History at 0x7f03483ad258)

[29] predicted= autoencoder.predict(x_test)
     313/313 [=====] - 2s 5ms/step

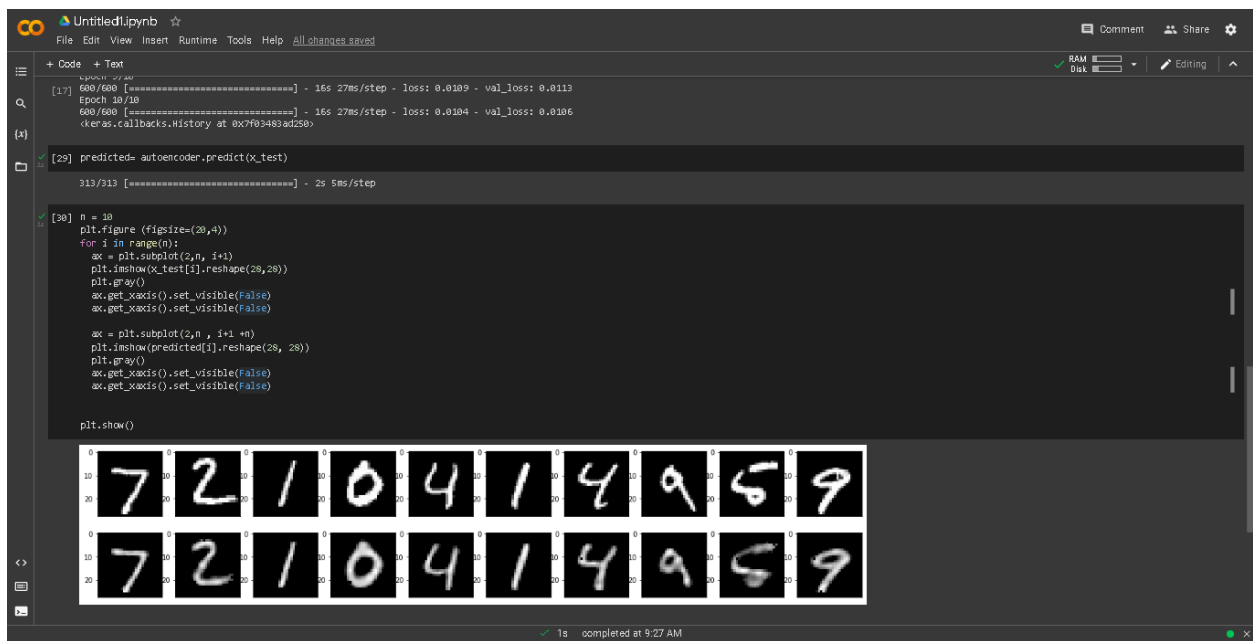
[30] n = 10
     plt.figure(figsize=(20,4))
     for i in range(n):
         ax = plt.subplot(2,n, i+1)
         plt.imshow(x_test[i].reshape(28,28))
         plt.gray()
         ax.get_xaxis().set_visible(False)
         ax.get_yaxis().set_visible(False)

         ax = plt.subplot(2,n, i+1+n)
         plt.imshow(predicted[i].reshape(28, 28))

1s completed at 9:27 AM
```

Training >= 10

Prediksi



Visualisasi