DAFTAR PUSTAKA

Adiguna, R., & Soelistio, Y. E. (2018). CNN Based Posture-Free Hand Detection. *Neural Networks*.

Alamsyah, D., & Rahmadi, M. (2018). Estimasi Kedalaman Pada Citra Dengan Conditional Random Field (CRF) dan Structured Support Vector Machine (SSVM) Menggunakan Fitur FFT. *Technology Acceptance Model , 9*, 1-6.

Alpaydin. (2009). Introduction to Machine Learning, Second Edition. *MIT Press.* London.

Alpaydin, E. (2009). Introduction to Machine Learning : Second Edition. *MIT Press.* London.

Anne, T. (2018). *MIT neuroscientists find the brain can identify omage seen for as little as 13 milliseconds*. (ViSenze) Retrieved Mei 20, Oktober, from news.mit.edu/2014/in-the-blink-of-an-eye-0116

Basha, S. S., Dubay, S. R., Pulabaigari, V., & Mukherjee, S. (2019). Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification. *Neurocomputing*.

Bishop M., C. (2016). *Pattern Recognition and Machine Learning.* Springer.

Fernandes, D. N., & Kwolek, B. (2017). *Handposture Recognition Using Convolutional Neural Network.* Peru: National University of Engineering.

Gotama, W. P. (2018). *Pengenalan konsep pembelajaran mesin dan deep learning , edisi. 1.2.*

Hanyu, Z., Jiaxin, Z., Lingling, F., & Xianghai, W. (2018). Image Classification with an RGB-channel nonsubsampled contourlet transform and a convolutional neural network. *Liaoning Normal University*.

Hindawi. (2017). A Review of Deep Learning Methods And Application Unmanned Aerial Vehicles. *Journal of Sensors*.

Lingling, F., Hanyu, Z., Jiaxin, Z., & Xianghai, W. (2018). Image retrieval using BIM and features from pretrained VGG network for indoor localization. Building and Environment. *Building and Environment*, 23–31.

Mathworks. (n.d.). *What Is a Convolutional Neural Network*. (Matlab) Retrieved Oktober 20, 2019, from https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html#howitworks

Muhammad, S. (2018). Klasifikasi Penyakit pada citra daun menggunakan convolutional neural network. Institut Pertanian Bogor.

Rahayu, D. R., Jimmy, F. R., & Dannes., V. R. (2015). Analisis Gangguan Pendengaran Pada Penyelam di Danau Tonando Desa Watumea Kecamatan Eris Kabupaten Minahasa Provinsi Sulawesi Utara 2014. *e-Biomedic, 3*.

Rahim, N. H. (2019). *Pengenalan pola ekspresi wajah secara real-time menggunakan convolutional neural network(CNN).* Makassar: Universitas Hasanuddin.

Sangul, M., Ozyildirim, B., & Avici, M. (2019). Differential Convolutional Neural Network. *Neural Network, 116*, 279-287.

Sena, S. (2018, Mei 27). *Medium*. Retrieved Oktober 10, 2019, from https://medium.com/@samuelse na/pengenalan-deep-learningpart-7-convolutional-neuralnetwork-cnn-b003b477dc94

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR.* Oxford: University of Oxford .

Snijders C., Matzat U., & Reips D. (2012). 'Big Data'. Big gaps of knowledge in the field of Internet. *International Journal of Internet Science, vol. 7*, pp. 1-5.

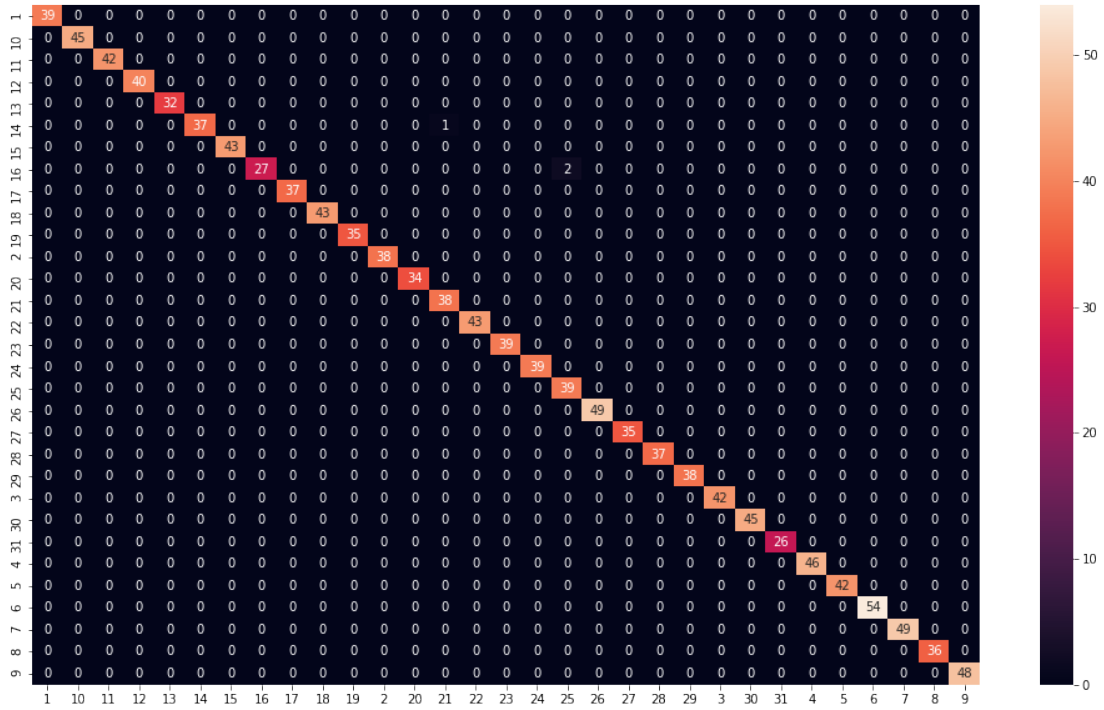
Sulfayanti, Hironori Takimoto, Hitoshi Yamauchi, & Akihiro Kanagawa. (2017). Hand Posture Classification Using Image Depth Data with Convolutional Neural Networks. *Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference 2017.* Japan.

ViSenze. (2018). *New Research from ViSenze Finds 62 Percent of Generation Z and Milenneal Consumers want visual Search Capabilities*. (ViSenze) Retrieved Oktober 20, 2019, from https://www.businesswire.com/news/home/20180829005092/en/New-Research-ViSenze-Finds-62-Percent-Generation#.W4eYrWp5Mrc.linkedin

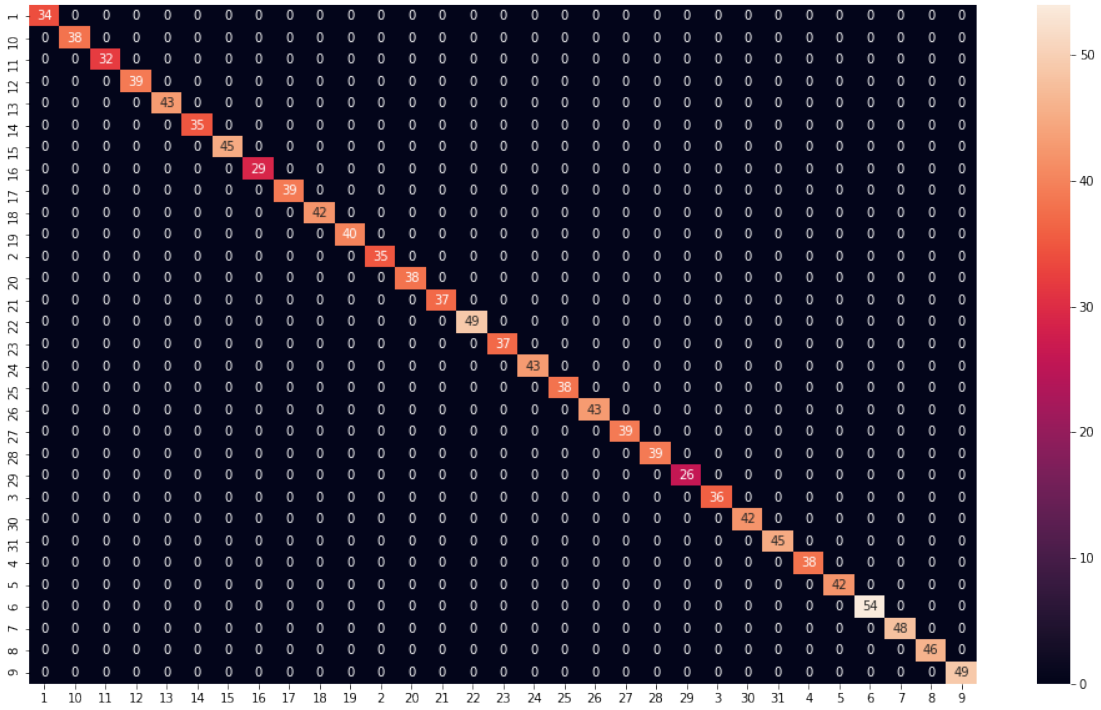
Yeung, K. (2017, Mei 17). Google Photos passes 500 million users.

LAMPIRAN

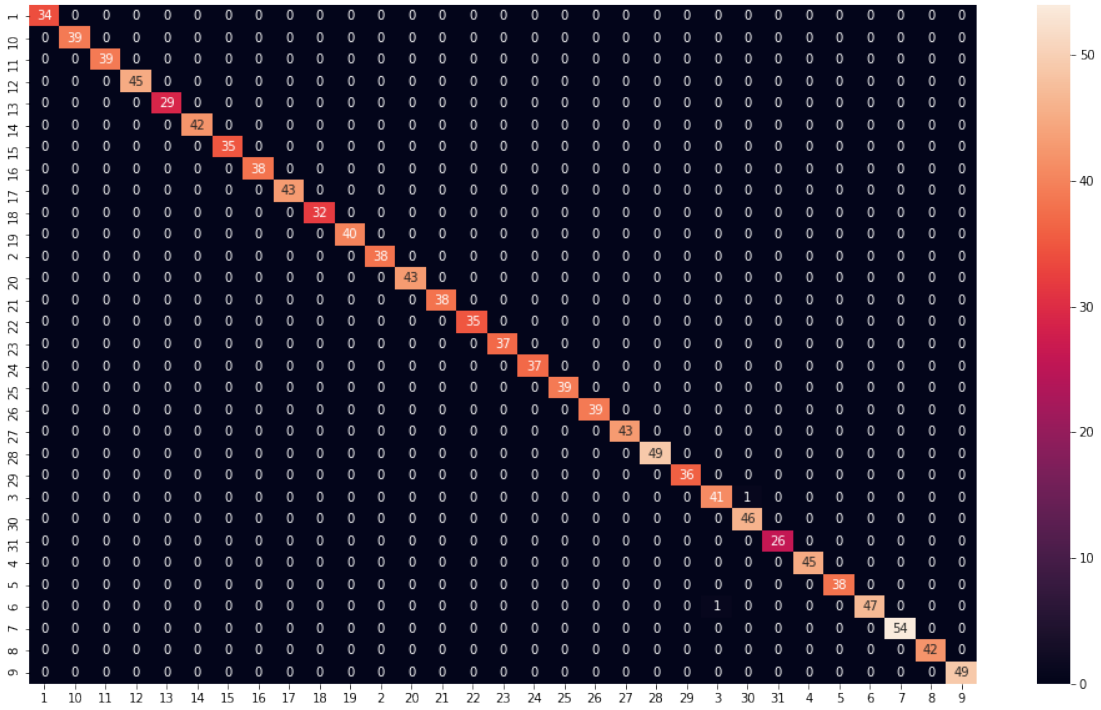
Lampiran 1 : Confusion Matrix Subjek 1



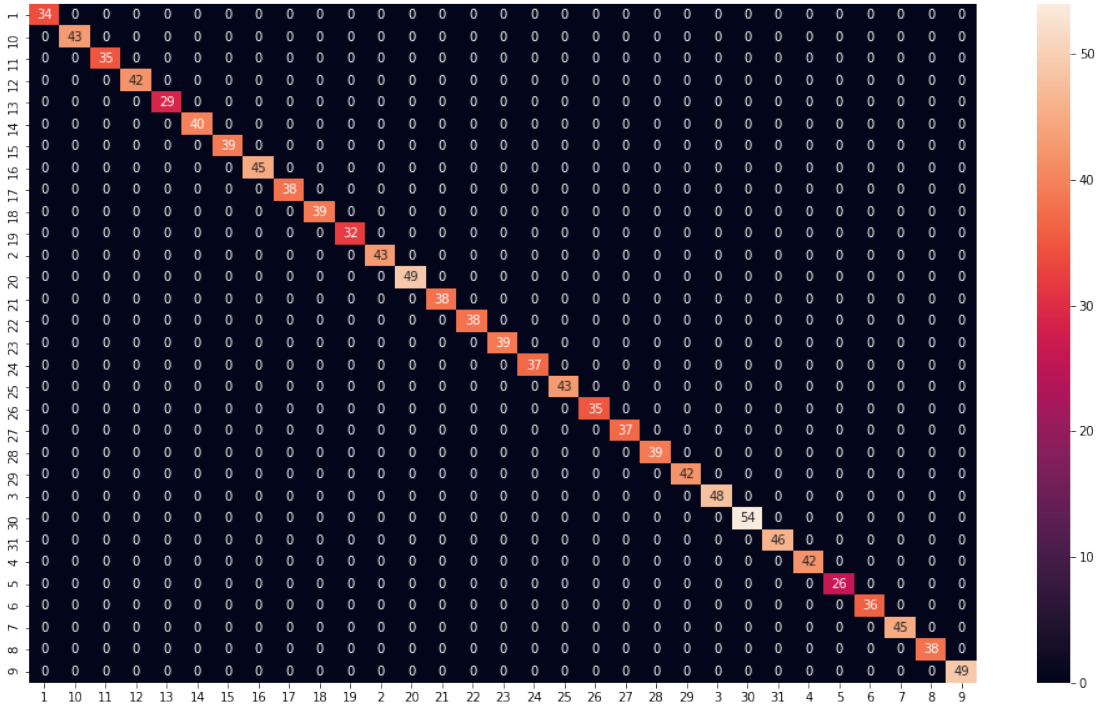
Lampiran 2 : Confusion Matrix Subjek 2



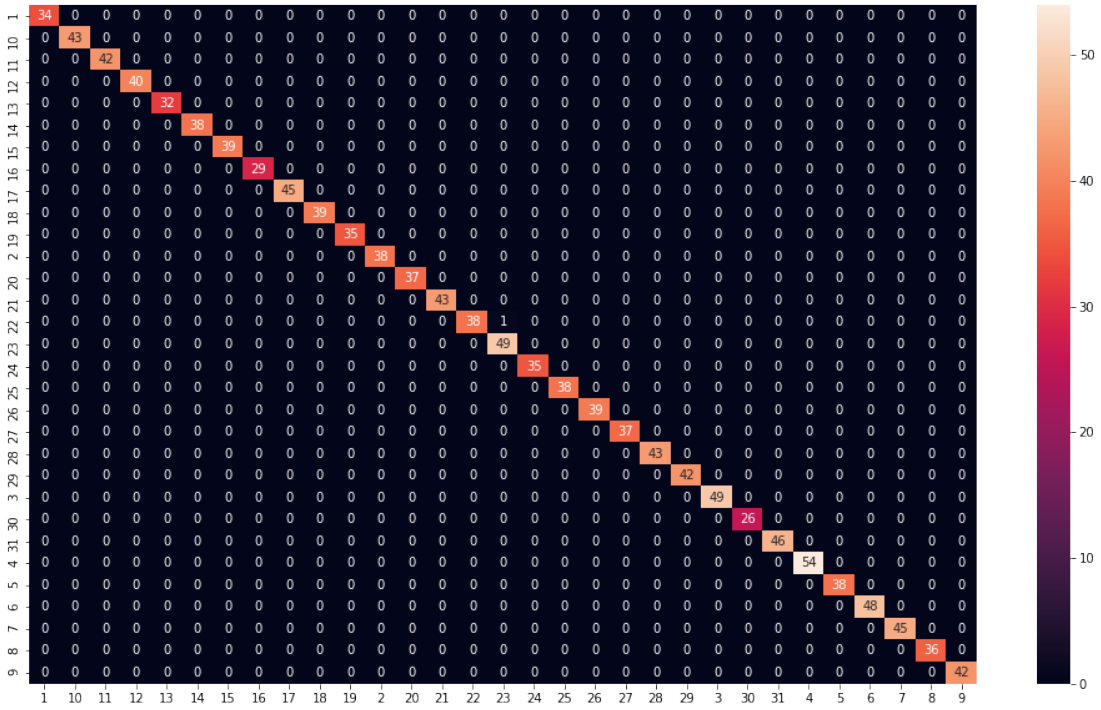
Lampiran 3 : Confusion Matrix Subjek 3



Lampiran 4 : Confusion Matrix Subjek 4



Lampiran 5 : Confusion Matrix Subjek 5



Lampiran 6 : Presisi , Recall , F1 setiap kelas Subjek 1

```
1240/1240 [==============================] - 6s 5ms/step
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        39
           1       1.00      1.00      1.00        45
           2       1.00      1.00      1.00        42
           3       1.00      1.00      1.00        40
           4       1.00      1.00      1.00        32
           5       1.00      0.97      0.99        38
           6       1.00      1.00      1.00        43
           7       1.00      0.93      0.96        29
           8       1.00      1.00      1.00        37
           9       1.00      1.00      1.00        43
          10       1.00      1.00      1.00        35
          11       1.00      1.00      1.00        38
          12       1.00      1.00      1.00        34
          13       0.97      1.00      0.99        38
          14       1.00      1.00      1.00        43
          15       1.00      1.00      1.00        39
          16       1.00      1.00      1.00        39
          17       0.95      1.00      0.97        39
          18       1.00      1.00      1.00        49
          19       1.00      1.00      1.00        35
          20       1.00      1.00      1.00        37
          21       1.00      1.00      1.00        38
          22       1.00      1.00      1.00        42
          23       1.00      1.00      1.00        45
          24       1.00      1.00      1.00        26
          25       1.00      1.00      1.00        46
          26       1.00      1.00      1.00        42
          27       1.00      1.00      1.00        54
          28       1.00      1.00      1.00        49
          29       1.00      1.00      1.00        36
          30       1.00      1.00      1.00        48
```

Lampiran 7 : Presisi , Recall , F1 setiap kelas Subjek 2

```
1240/1240 [==============================] - 6s 5ms/step
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	34
1	1.00	1.00	1.00	38
2	1.00	1.00	1.00	32
3	1.00	1.00	1.00	39
4	1.00	1.00	1.00	43
5	1.00	1.00	1.00	35
6	1.00	1.00	1.00	45
7	1.00	1.00	1.00	29
8	1.00	1.00	1.00	39
9	1.00	1.00	1.00	42
10	1.00	1.00	1.00	40
11	1.00	1.00	1.00	35
12	1.00	1.00	1.00	38
13	1.00	1.00	1.00	37
14	1.00	1.00	1.00	49
15	1.00	1.00	1.00	37
16	1.00	1.00	1.00	43
17	1.00	1.00	1.00	38
18	1.00	1.00	1.00	43
19	1.00	1.00	1.00	39
20	1.00	1.00	1.00	39
21	1.00	1.00	1.00	26
22	1.00	1.00	1.00	36
23	1.00	1.00	1.00	42
24	1.00	1.00	1.00	45
25	1.00	1.00	1.00	38
26	1.00	1.00	1.00	42
27	1.00	1.00	1.00	54
28	1.00	1.00	1.00	48
29	1.00	1.00	1.00	46
30	1.00	1.00	1.00	49

Lampiran 9 : Presisi , Recall , F1 setiap kelas Subjek 3

```
1240/1240 [==============================] - 6s 5ms/step
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	34
1	1.00	1.00	1.00	39
2	1.00	1.00	1.00	39
3	1.00	1.00	1.00	45
4	1.00	1.00	1.00	29
5	1.00	1.00	1.00	42
6	1.00	1.00	1.00	35
7	1.00	1.00	1.00	38
8	1.00	1.00	1.00	43
9	1.00	1.00	1.00	32
10	1.00	1.00	1.00	40
11	1.00	1.00	1.00	38
12	1.00	1.00	1.00	43
13	1.00	1.00	1.00	38
14	1.00	1.00	1.00	35
15	1.00	1.00	1.00	37
16	1.00	1.00	1.00	37
17	1.00	1.00	1.00	39
18	1.00	1.00	1.00	39
19	1.00	1.00	1.00	43
20	1.00	1.00	1.00	49
21	1.00	1.00	1.00	36
22	0.98	0.98	0.98	42
23	0.98	1.00	0.99	46
24	1.00	1.00	1.00	26
25	1.00	1.00	1.00	45
26	1.00	1.00	1.00	38
27	1.00	0.98	0.99	48
28	1.00	1.00	1.00	54
29	1.00	1.00	1.00	42
30	1.00	1.00	1.00	49

Lampiran 10 : Presisi , Recall , F1 setiap kelas Subjek 4

```
1240/1240 [==============================] - 6s 5ms/step
```

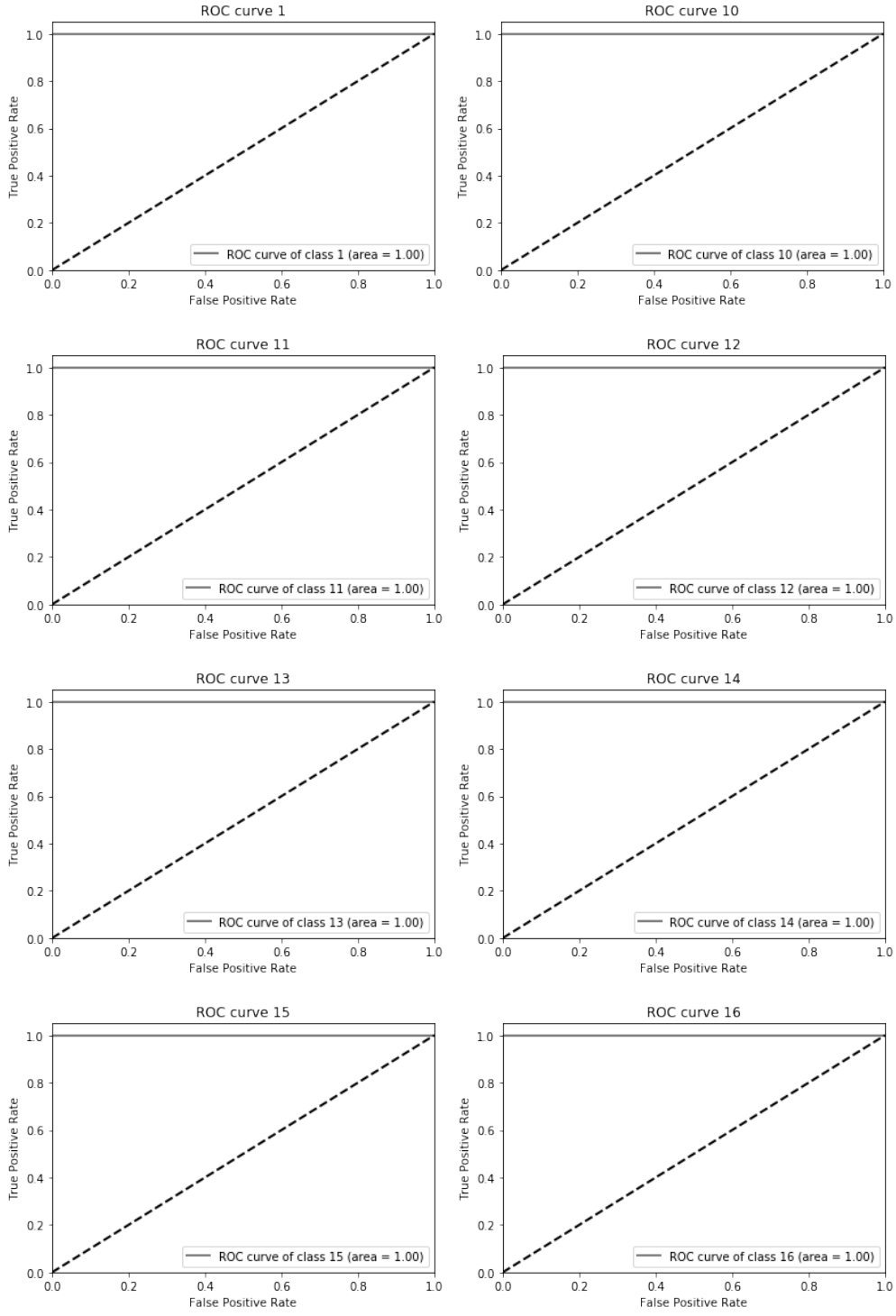
	precision	recall	f1-score	support
0	1.00	1.00	1.00	34
1	1.00	1.00	1.00	43
2	1.00	1.00	1.00	35
3	1.00	1.00	1.00	42
4	1.00	1.00	1.00	29
5	1.00	1.00	1.00	40
6	1.00	1.00	1.00	39
7	1.00	1.00	1.00	45
8	1.00	1.00	1.00	38
9	1.00	1.00	1.00	39
10	1.00	1.00	1.00	32
11	1.00	1.00	1.00	43
12	1.00	1.00	1.00	49
13	1.00	1.00	1.00	38
14	1.00	1.00	1.00	38
15	1.00	1.00	1.00	39
16	1.00	1.00	1.00	37
17	1.00	1.00	1.00	43
18	1.00	1.00	1.00	35
19	1.00	1.00	1.00	37
20	1.00	1.00	1.00	39
21	1.00	1.00	1.00	42
22	1.00	1.00	1.00	48
23	1.00	1.00	1.00	54
24	1.00	1.00	1.00	46
25	1.00	1.00	1.00	42
26	1.00	1.00	1.00	26
27	1.00	1.00	1.00	36
28	1.00	1.00	1.00	45
29	1.00	1.00	1.00	38
30	1.00	1.00	1.00	49

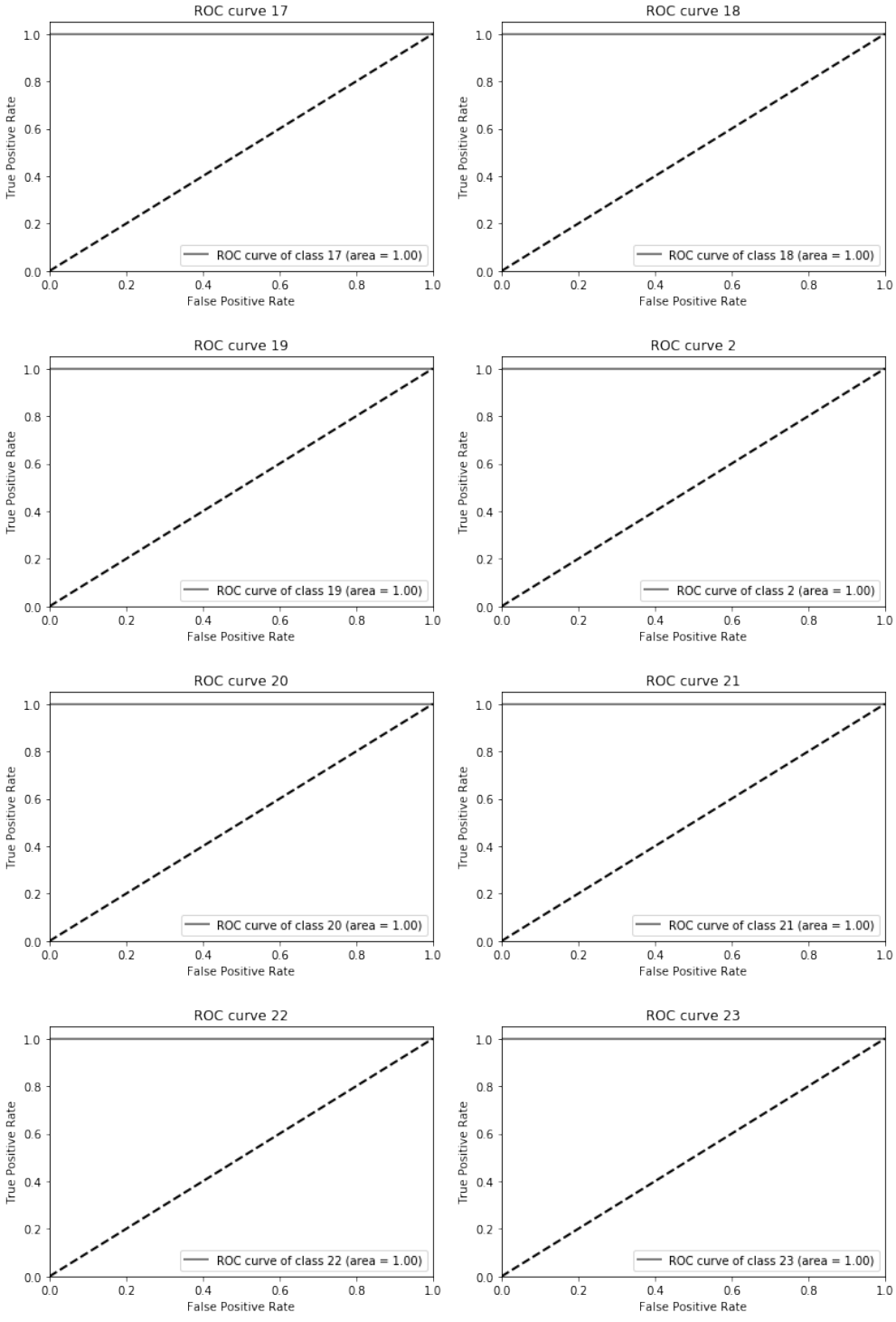
Lampiran 9 : Presisi , Recall , F1 setiap kelas Subjek 5

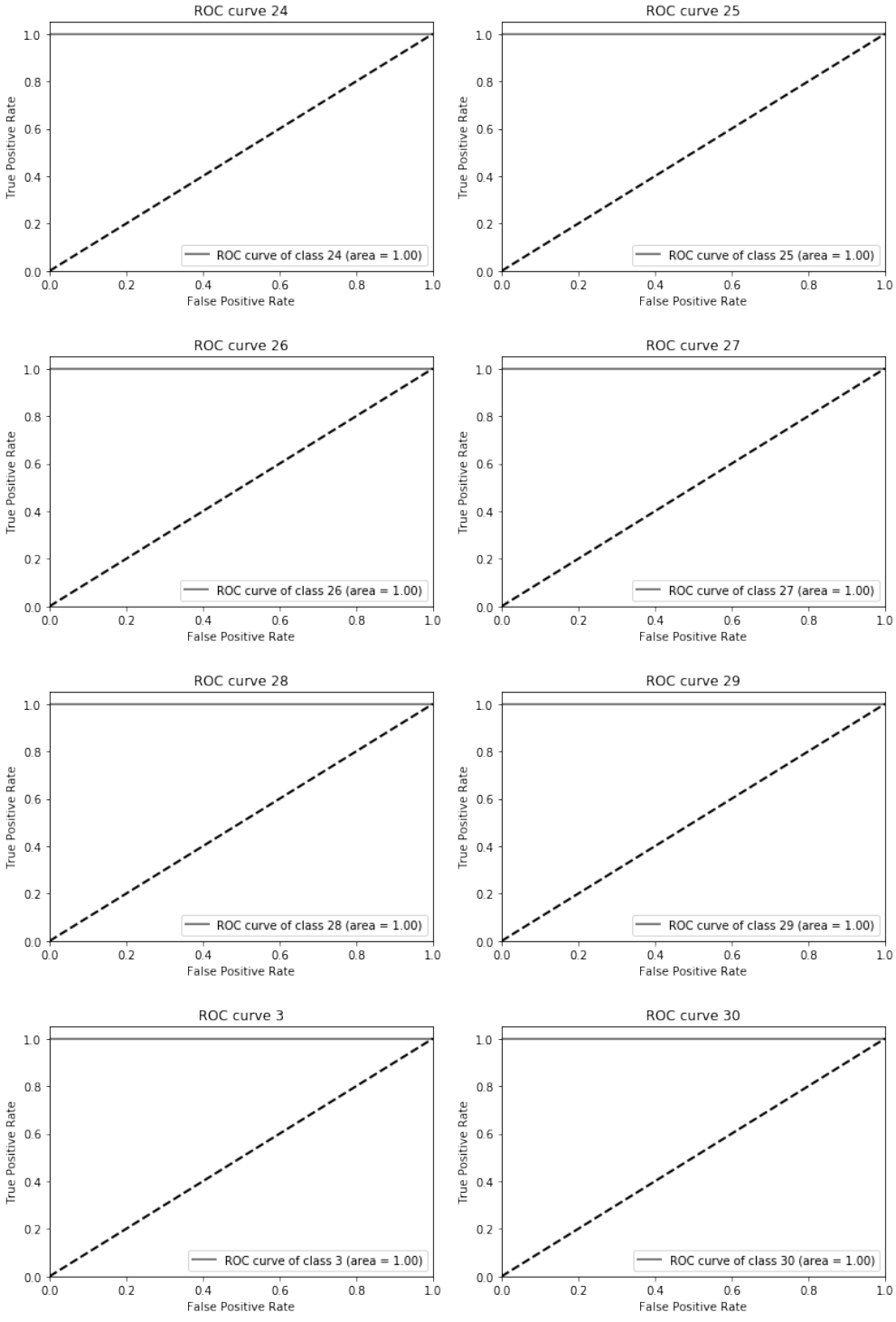
```
1240/1240 [==============================] - 6s 5ms/step
```

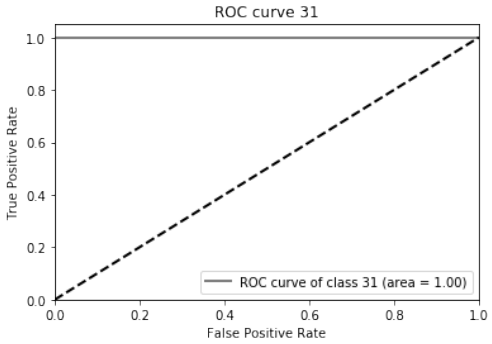
	precision	recall	f1-score	support
0	1.00	1.00	1.00	34
1	1.00	1.00	1.00	43
2	1.00	1.00	1.00	42
3	1.00	1.00	1.00	40
4	1.00	1.00	1.00	32
5	1.00	1.00	1.00	38
6	1.00	1.00	1.00	39
7	1.00	1.00	1.00	29
8	1.00	1.00	1.00	45
9	1.00	1.00	1.00	39
10	1.00	1.00	1.00	35
11	1.00	1.00	1.00	38
12	1.00	1.00	1.00	37
13	1.00	1.00	1.00	43
14	1.00	0.97	0.99	39
15	0.98	1.00	0.99	49
16	1.00	1.00	1.00	35
17	1.00	1.00	1.00	38
18	1.00	1.00	1.00	39
19	1.00	1.00	1.00	37
20	1.00	1.00	1.00	43
21	1.00	1.00	1.00	42
22	1.00	1.00	1.00	49
23	1.00	1.00	1.00	26
24	1.00	1.00	1.00	46
25	1.00	1.00	1.00	54
26	1.00	1.00	1.00	38
27	1.00	1.00	1.00	48
28	1.00	1.00	1.00	45
29	1.00	1.00	1.00	36
30	1.00	1.00	1.00	42

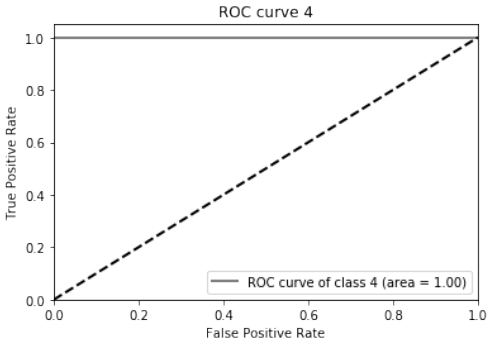
Lampiran 11 : ROC Subjek 1

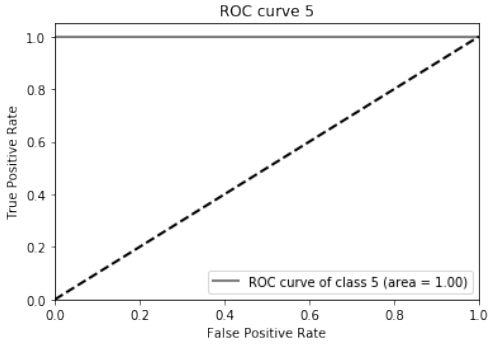


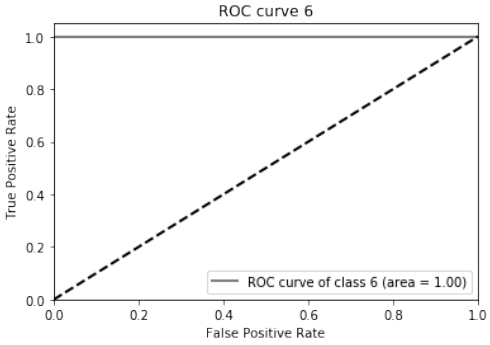

ROC curve 17
ROC curve of class 17 (area = 1.00)
ROC curve 18
ROC curve of class 18 (area = 1.00)
ROC curve 19
ROC curve of class 19 (area = 1.00)
ROC curve 2
ROC curve of class 2 (area = 1.00)
ROC curve 20
ROC curve of class 20 (area = 1.00)
ROC curve 21
ROC curve of class 21 (area = 1.00)
ROC curve 22
ROC curve of class 22 (area = 1.00)
ROC curve 23
ROC curve of class 23 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

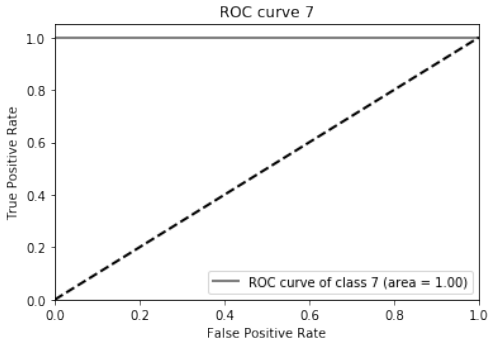

ROC curve 24
True Positive Rate
False Positive Rate
ROC curve of class 24 (area = 1.00)
ROC curve 25
True Positive Rate
False Positive Rate
ROC curve of class 25 (area = 1.00)
ROC curve 26
True Positive Rate
False Positive Rate
ROC curve of class 26 (area = 1.00)
ROC curve 27
True Positive Rate
False Positive Rate
ROC curve of class 27 (area = 1.00)
ROC curve 28
True Positive Rate
False Positive Rate
ROC curve of class 28 (area = 1.00)
ROC curve 29
True Positive Rate
False Positive Rate
ROC curve of class 29 (area = 1.00)
ROC curve 3
True Positive Rate
False Positive Rate
ROC curve of class 3 (area = 1.00)
ROC curve 30
True Positive Rate
False Positive Rate
ROC curve of class 30 (area = 1.00)

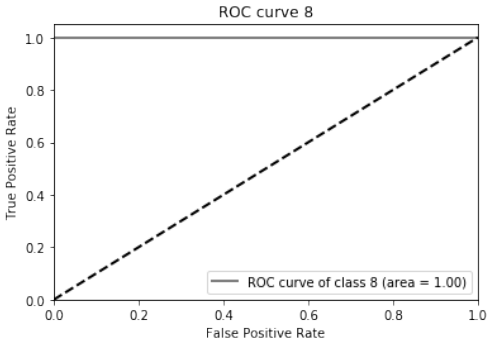

ROC curve 31
True Positive Rate
False Positive Rate
ROC curve of class 31 (area = 1.00)

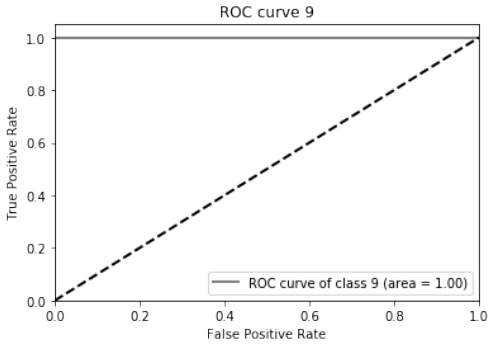

ROC curve 4
True Positive Rate
False Positive Rate
ROC curve of class 4 (area = 1.00)


ROC curve 5
True Positive Rate
False Positive Rate
ROC curve of class 5 (area = 1.00)

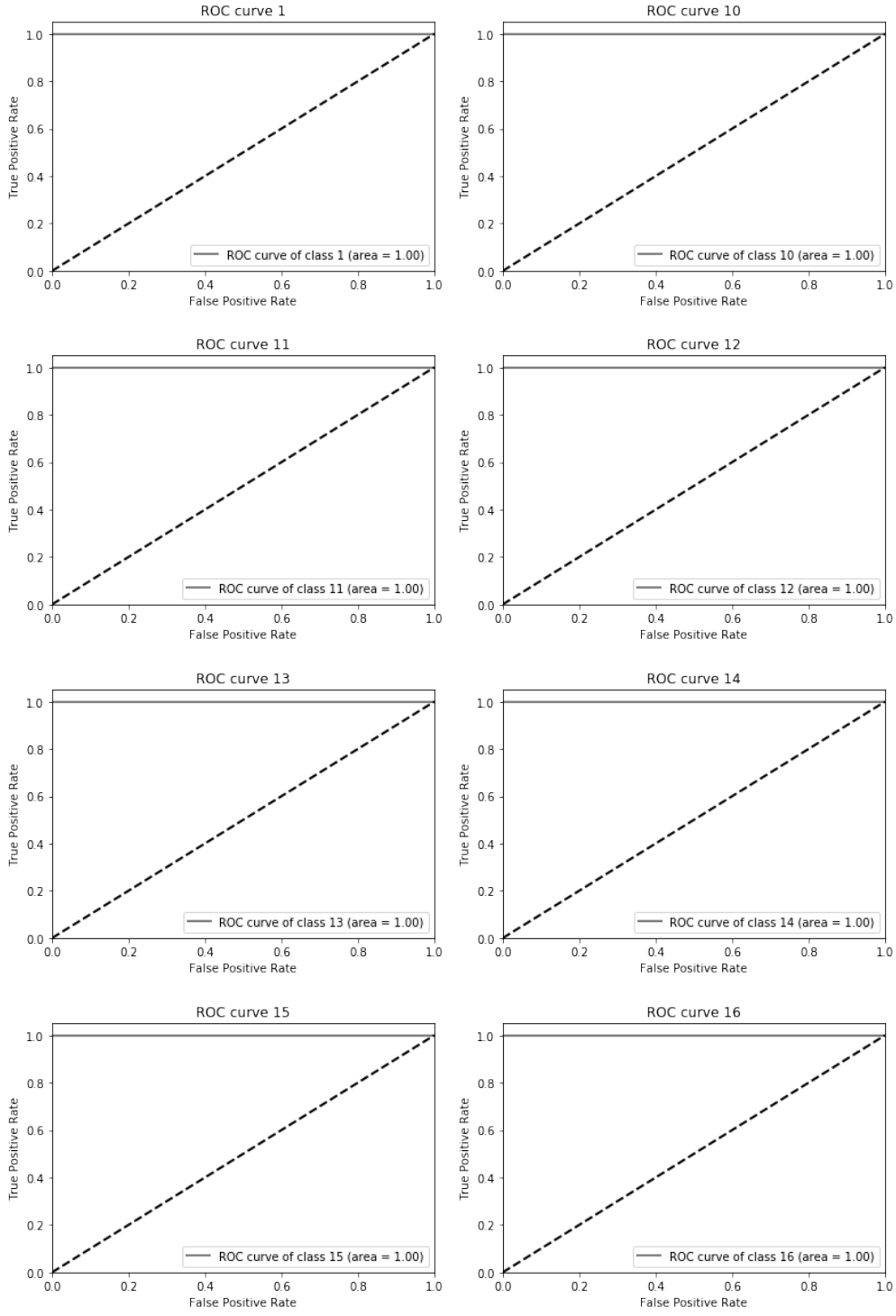

ROC curve 6
True Positive Rate
False Positive Rate
ROC curve of class 6 (area = 1.00)

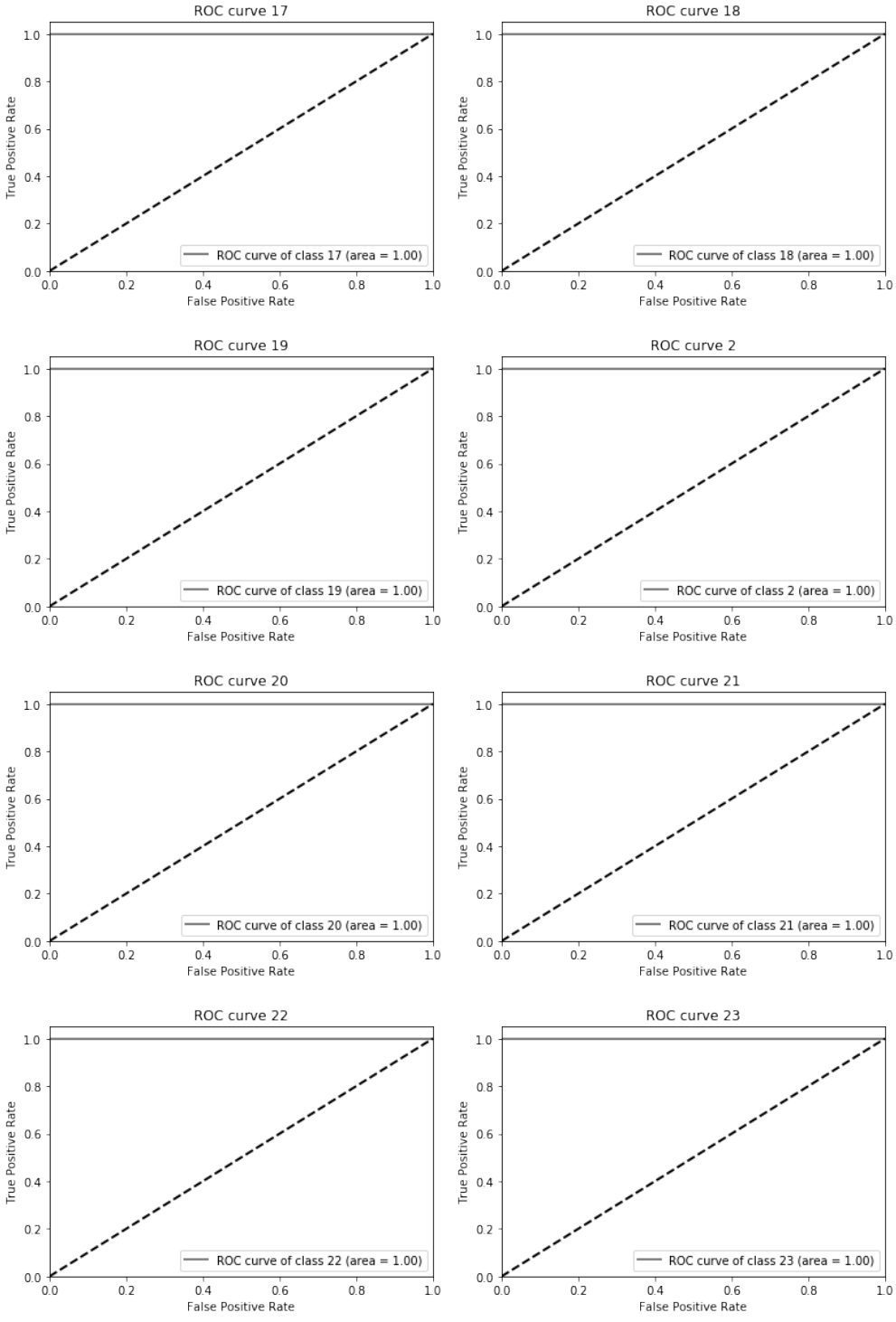

ROC curve 7
True Positive Rate
False Positive Rate
ROC curve of class 7 (area = 1.00)

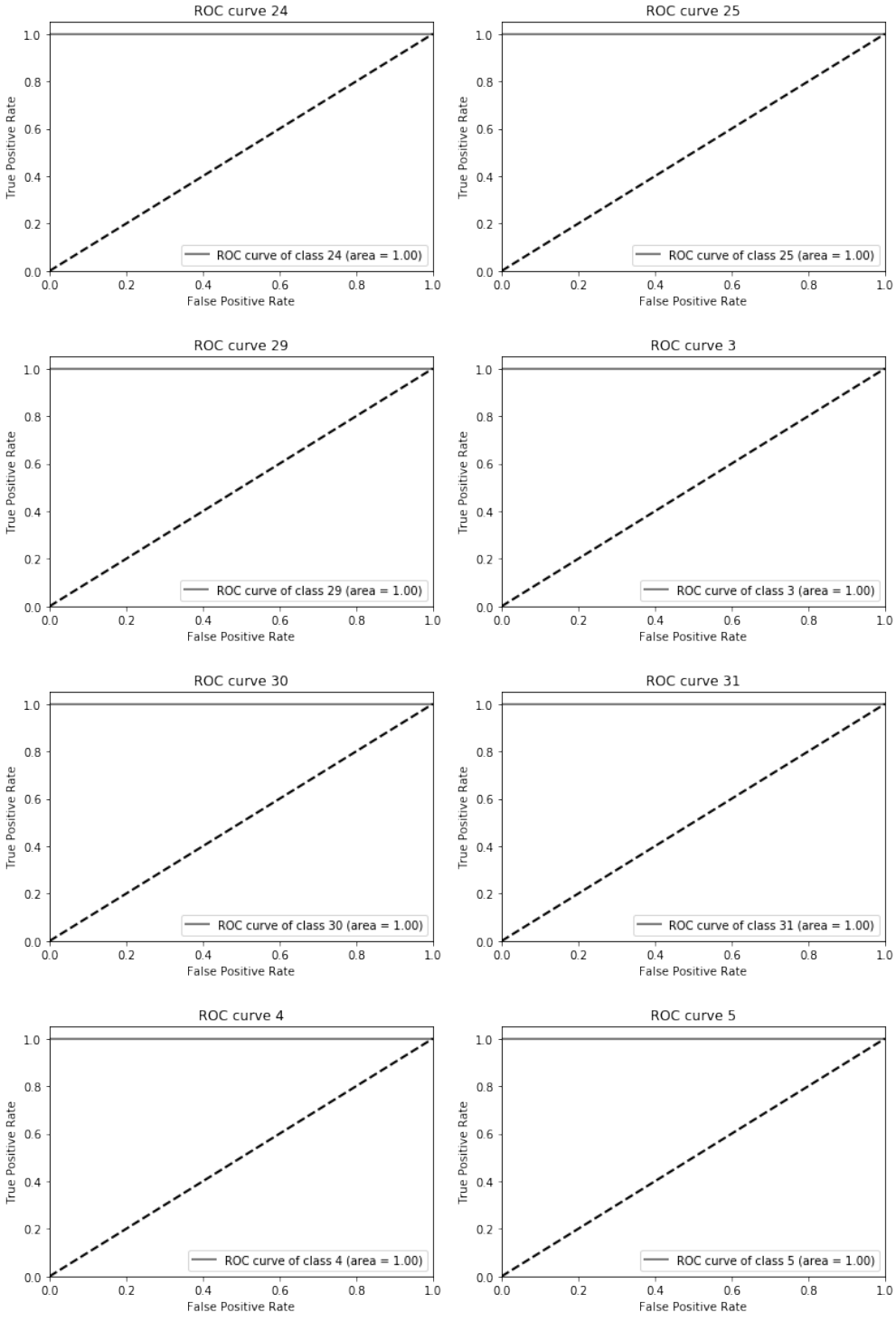

ROC curve 8
True Positive Rate
False Positive Rate
ROC curve of class 8 (area = 1.00)

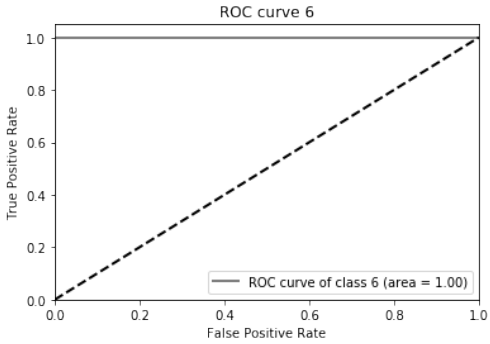

ROC curve 9
True Positive Rate
False Positive Rate
ROC curve of class 9 (area = 1.00)

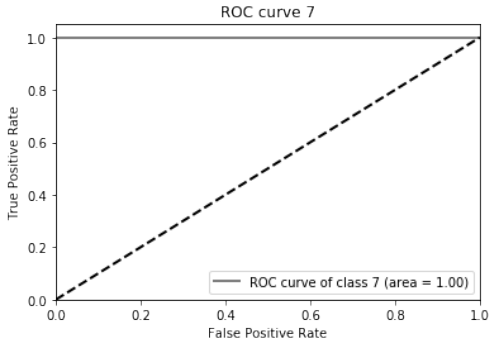
Lampiran 12 : ROC Subjek 2

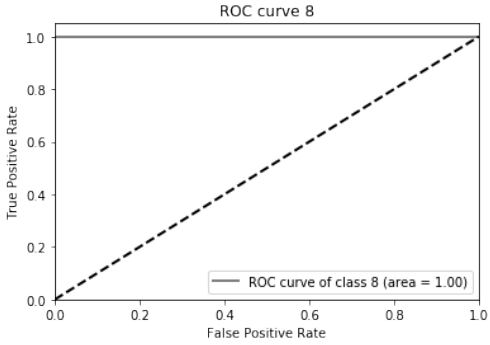


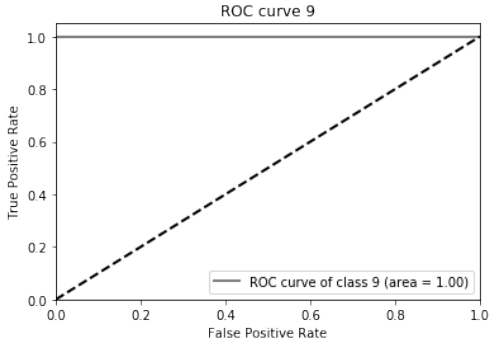

ROC curve 17
ROC curve of class 17 (area = 1.00)
ROC curve 18
ROC curve of class 18 (area = 1.00)
ROC curve 19
ROC curve of class 19 (area = 1.00)
ROC curve 2
ROC curve of class 2 (area = 1.00)
ROC curve 20
ROC curve of class 20 (area = 1.00)
ROC curve 21
ROC curve of class 21 (area = 1.00)
ROC curve 22
ROC curve of class 22 (area = 1.00)
ROC curve 23
ROC curve of class 23 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

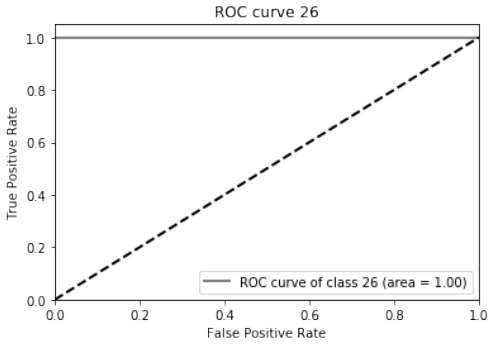

ROC curve 24
ROC curve of class 24 (area = 1.00)
ROC curve 25
ROC curve of class 25 (area = 1.00)
ROC curve 29
ROC curve of class 29 (area = 1.00)
ROC curve 3
ROC curve of class 3 (area = 1.00)
ROC curve 30
ROC curve of class 30 (area = 1.00)
ROC curve 31
ROC curve of class 31 (area = 1.00)
ROC curve 4
ROC curve of class 4 (area = 1.00)
ROC curve 5
ROC curve of class 5 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

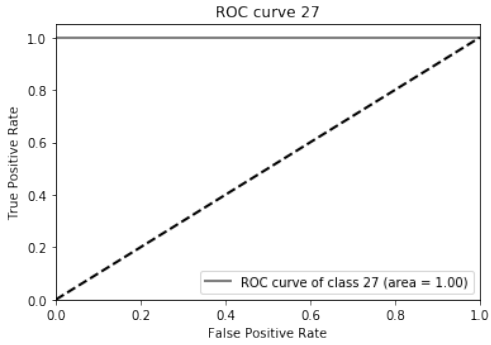

ROC curve 6
True Positive Rate
False Positive Rate
ROC curve of class 6 (area = 1.00)

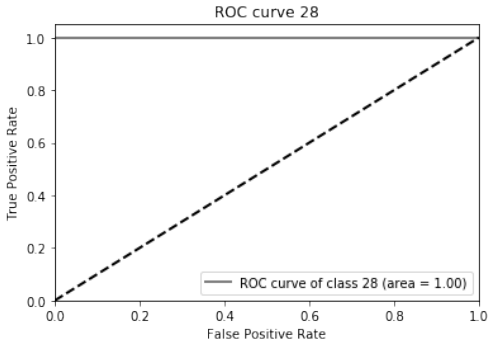

ROC curve 7
True Positive Rate
False Positive Rate
ROC curve of class 7 (area = 1.00)


ROC curve 8
True Positive Rate
False Positive Rate
ROC curve of class 8 (area = 1.00)

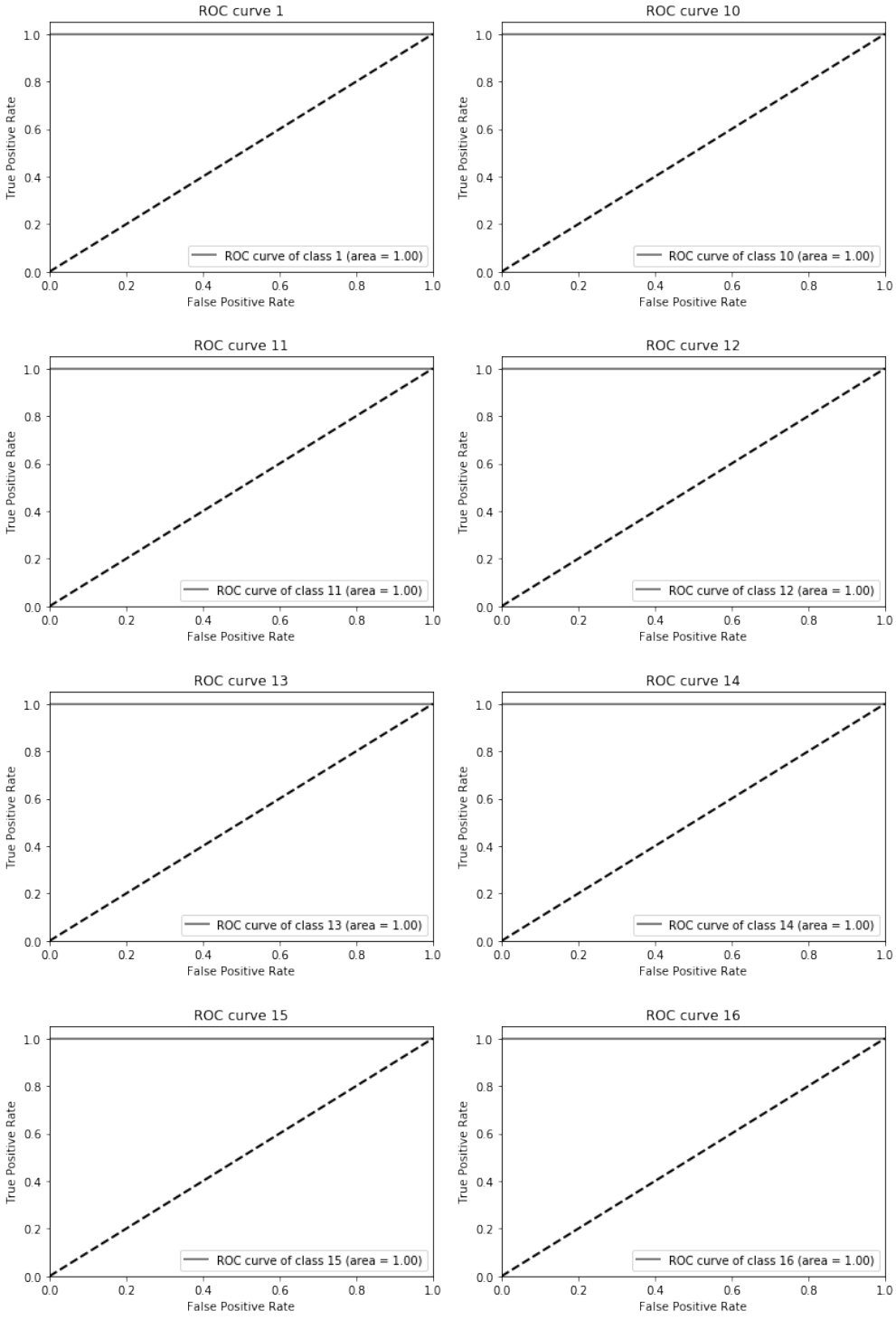

ROC curve 9
True Positive Rate
False Positive Rate
ROC curve of class 9 (area = 1.00)

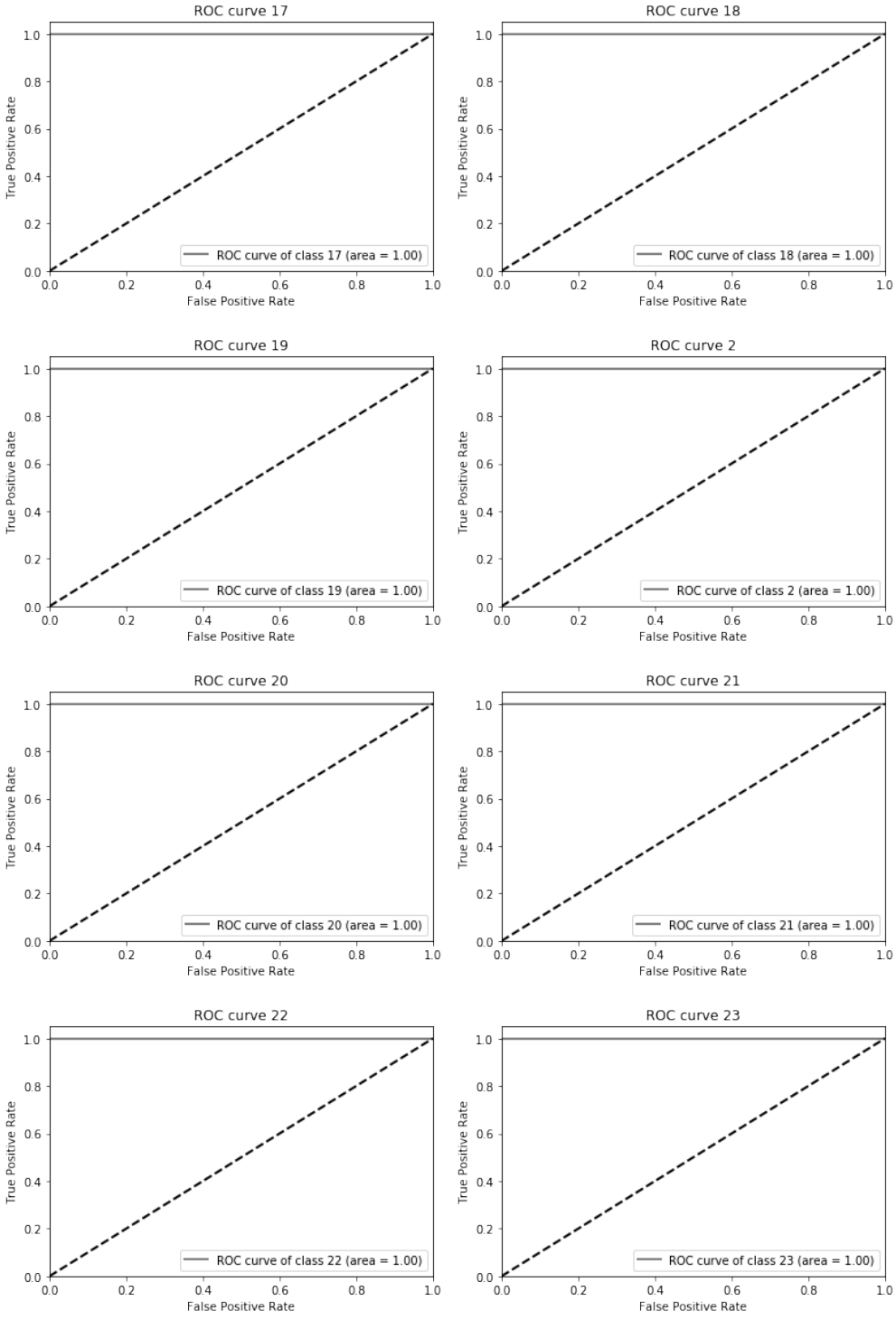

ROC curve 26
True Positive Rate
False Positive Rate
ROC curve of class 26 (area = 1.00)

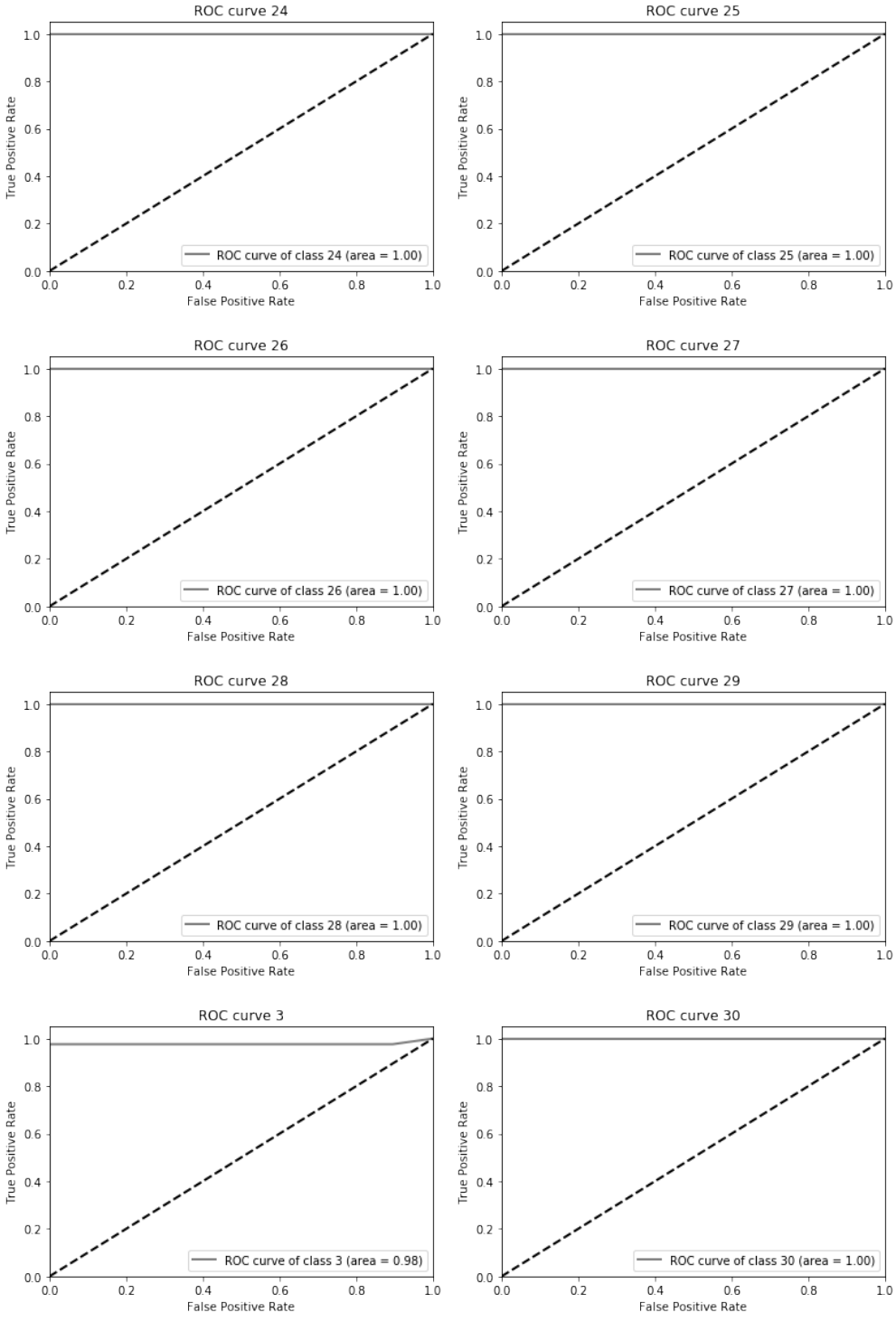

ROC curve 27
True Positive Rate
False Positive Rate
ROC curve of class 27 (area = 1.00)

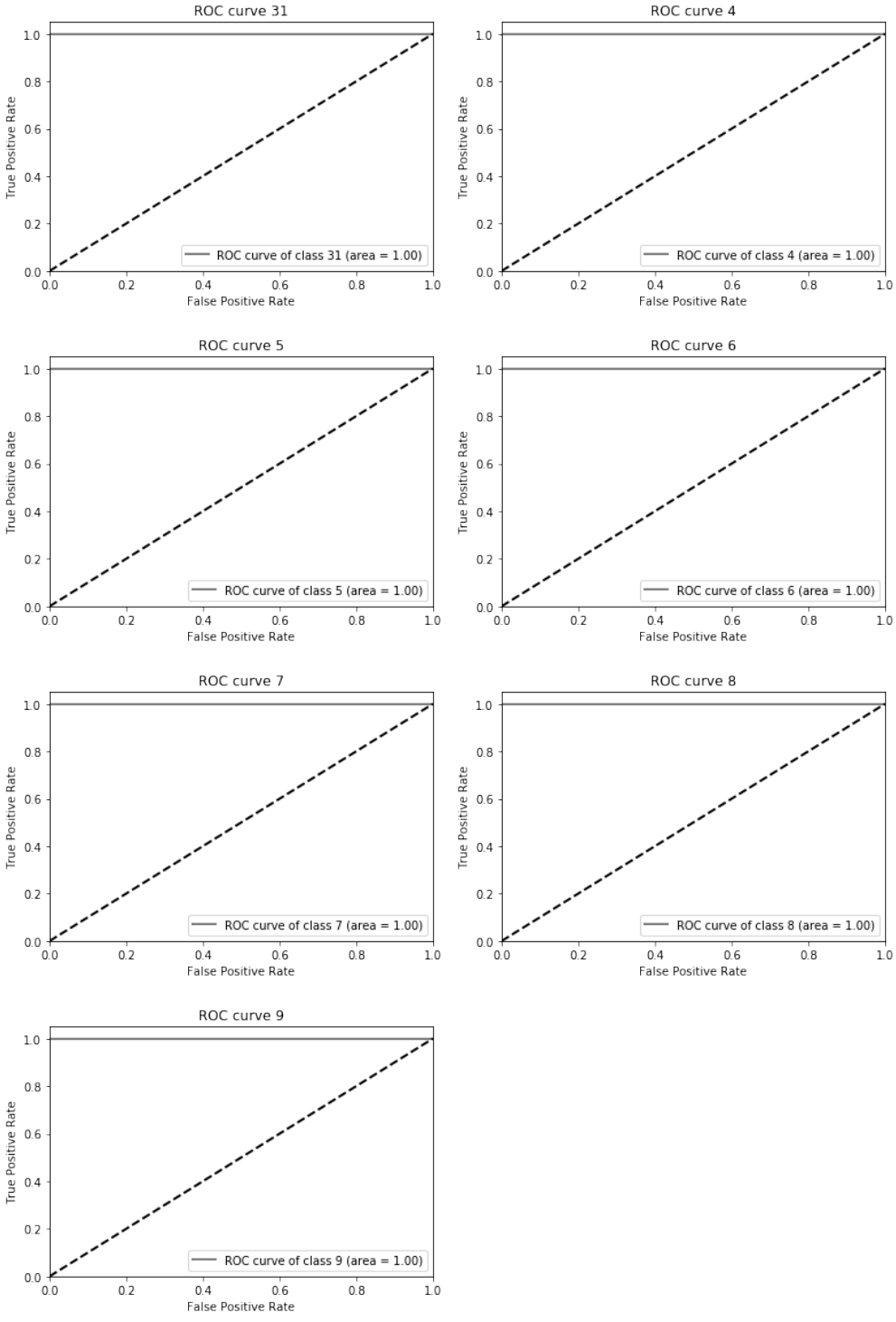

ROC curve 28
True Positive Rate
False Positive Rate
ROC curve of class 28 (area = 1.00)

Lampiran 13 : ROC Subjek 3

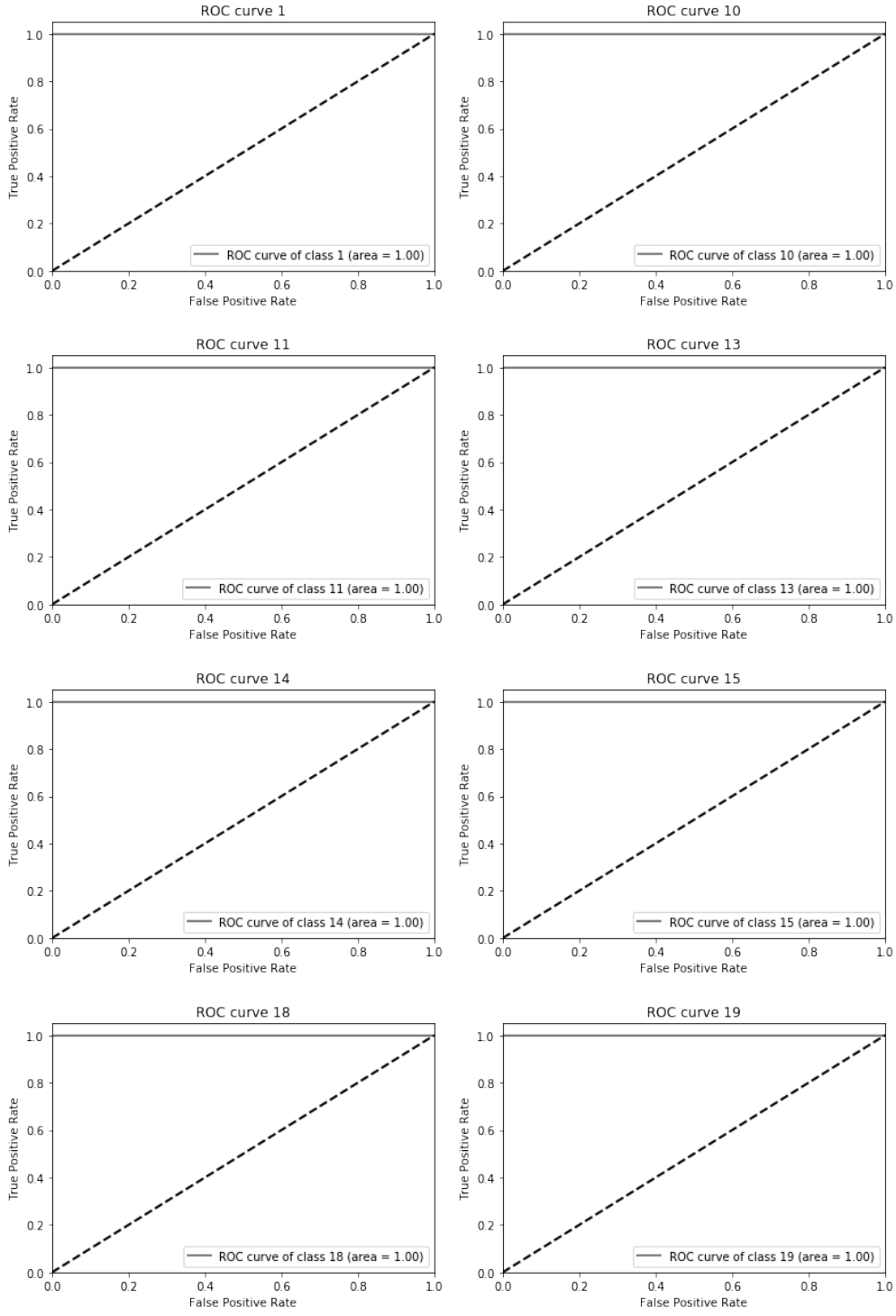


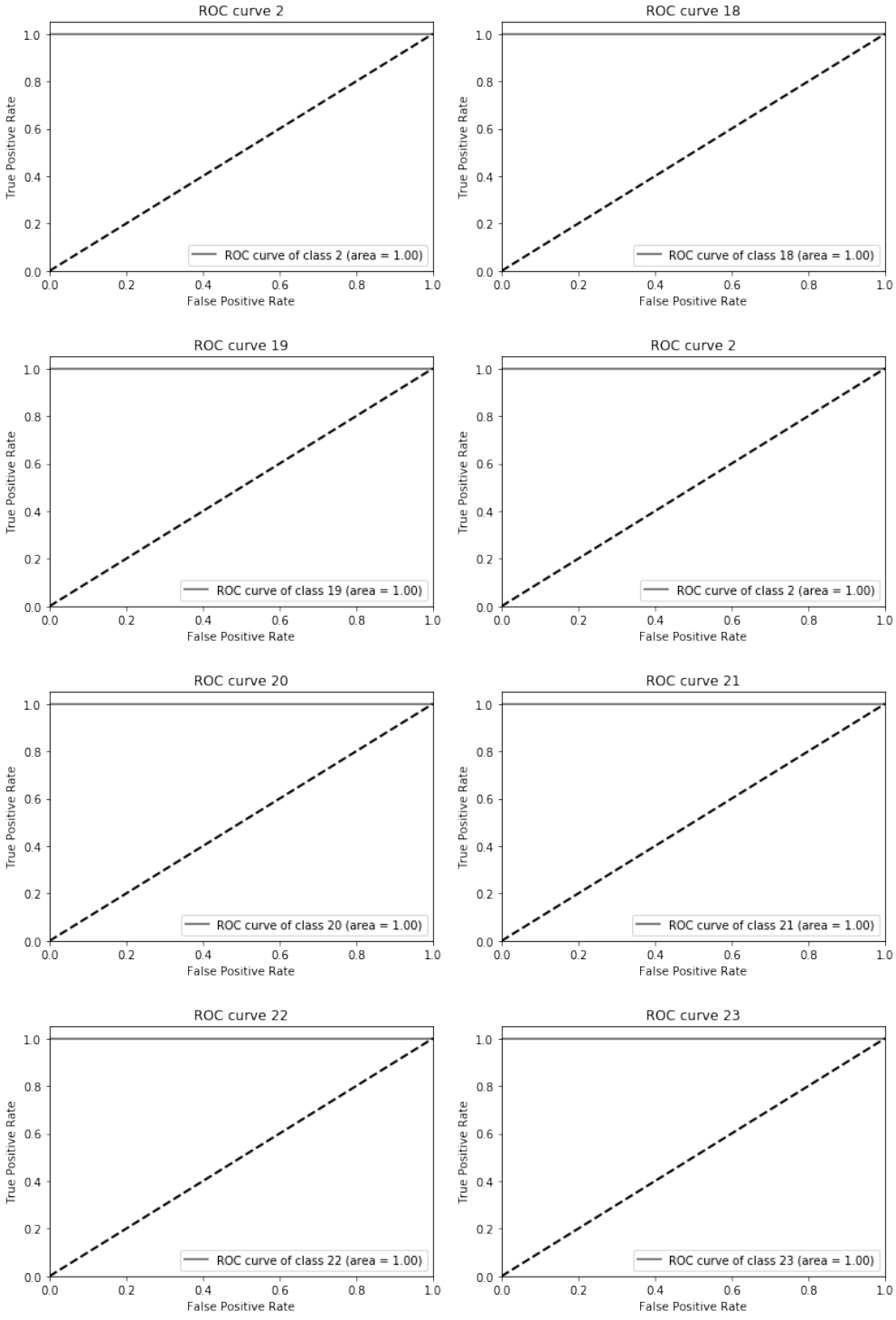

ROC curve 17
ROC curve of class 17 (area = 1.00)
ROC curve 18
ROC curve of class 18 (area = 1.00)
ROC curve 19
ROC curve of class 19 (area = 1.00)
ROC curve 2
ROC curve of class 2 (area = 1.00)
ROC curve 20
ROC curve of class 20 (area = 1.00)
ROC curve 21
ROC curve of class 21 (area = 1.00)
ROC curve 22
ROC curve of class 22 (area = 1.00)
ROC curve 23
ROC curve of class 23 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

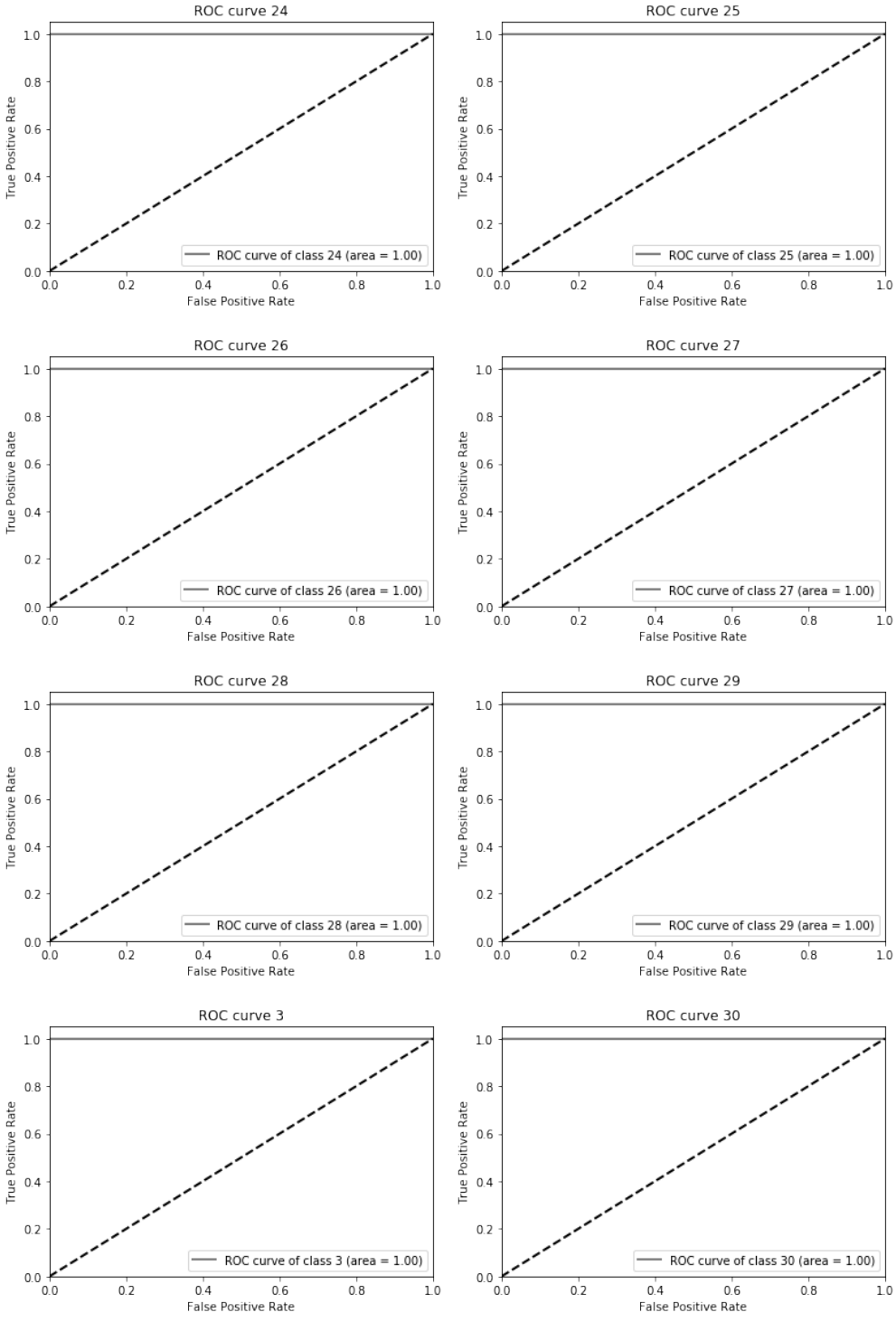

ROC curve 24
ROC curve of class 24 (area = 1.00)
ROC curve 25
ROC curve of class 25 (area = 1.00)
ROC curve 26
ROC curve of class 26 (area = 1.00)
ROC curve 27
ROC curve of class 27 (area = 1.00)
ROC curve 28
ROC curve of class 28 (area = 1.00)
ROC curve 29
ROC curve of class 29 (area = 1.00)
ROC curve 3
ROC curve of class 3 (area = 0.98)
ROC curve 30
ROC curve of class 30 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

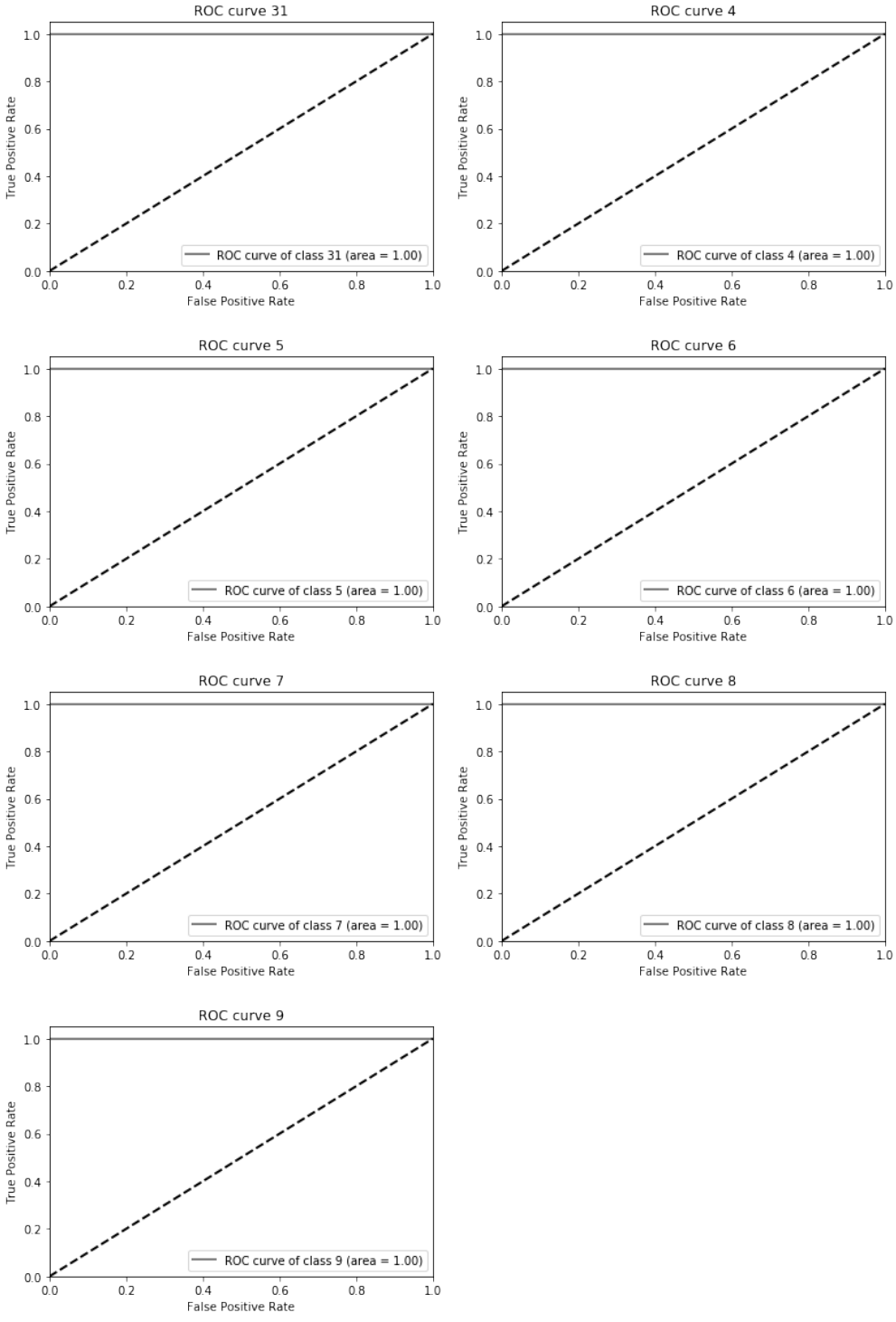

ROC curve 31
True Positive Rate
False Positive Rate
ROC curve of class 31 (area = 1.00)
ROC curve 4
True Positive Rate
False Positive Rate
ROC curve of class 4 (area = 1.00)
ROC curve 5
True Positive Rate
False Positive Rate
ROC curve of class 5 (area = 1.00)
ROC curve 6
True Positive Rate
False Positive Rate
ROC curve of class 6 (area = 1.00)
ROC curve 7
True Positive Rate
False Positive Rate
ROC curve of class 7 (area = 1.00)
ROC curve 8
True Positive Rate
False Positive Rate
ROC curve of class 8 (area = 1.00)
ROC curve 9
True Positive Rate
False Positive Rate
ROC curve of class 9 (area = 1.00)

Lampiran 14 : ROC Subjek 4

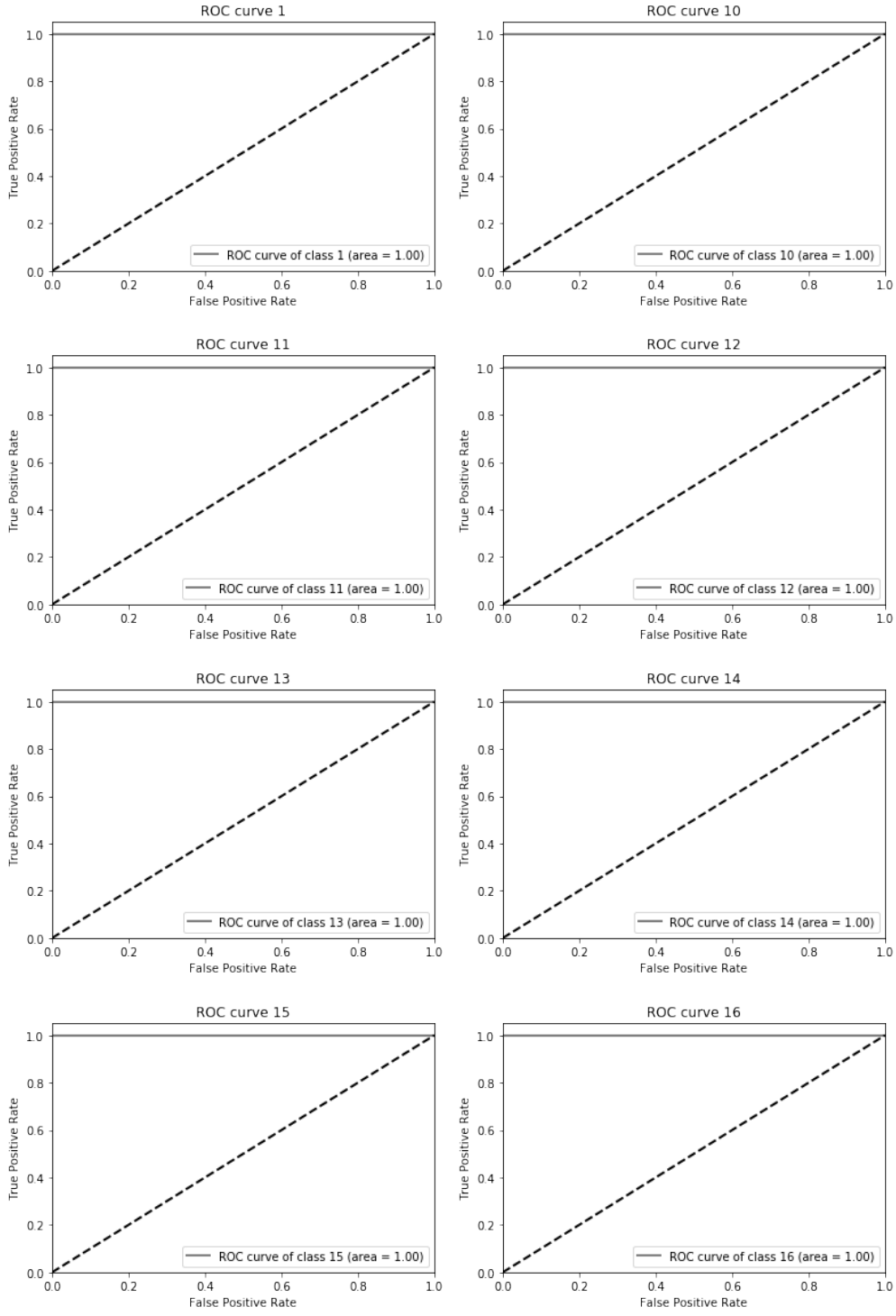


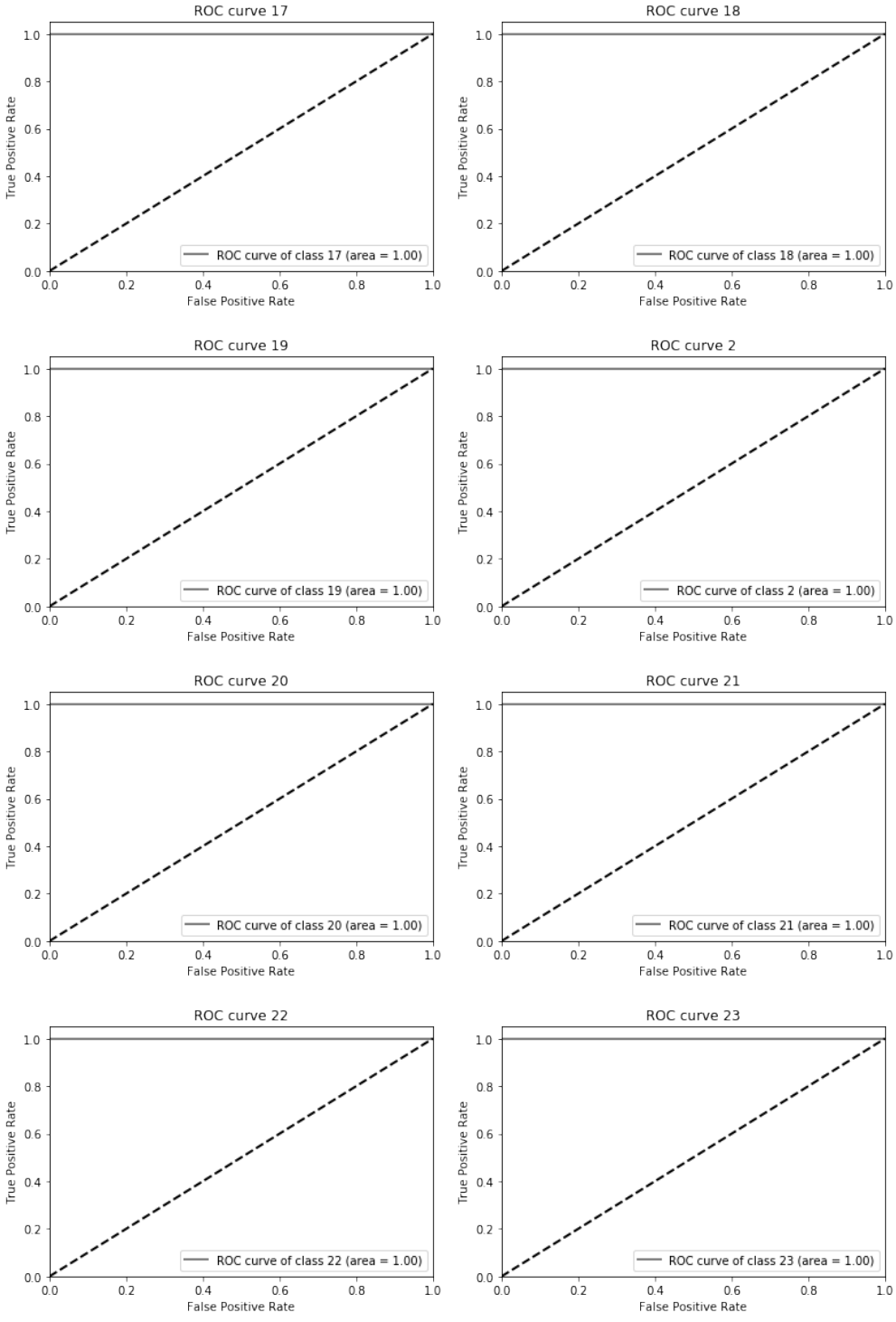

ROC curve 2
True Positive Rate
False Positive Rate
ROC curve of class 2 (area = 1.00)
ROC curve 18
True Positive Rate
False Positive Rate
ROC curve of class 18 (area = 1.00)
ROC curve 19
True Positive Rate
False Positive Rate
ROC curve of class 19 (area = 1.00)
ROC curve 2
True Positive Rate
False Positive Rate
ROC curve of class 2 (area = 1.00)
ROC curve 20
True Positive Rate
False Positive Rate
ROC curve of class 20 (area = 1.00)
ROC curve 21
True Positive Rate
False Positive Rate
ROC curve of class 21 (area = 1.00)
ROC curve 22
True Positive Rate
False Positive Rate
ROC curve of class 22 (area = 1.00)
ROC curve 23
True Positive Rate
False Positive Rate
ROC curve of class 23 (area = 1.00)

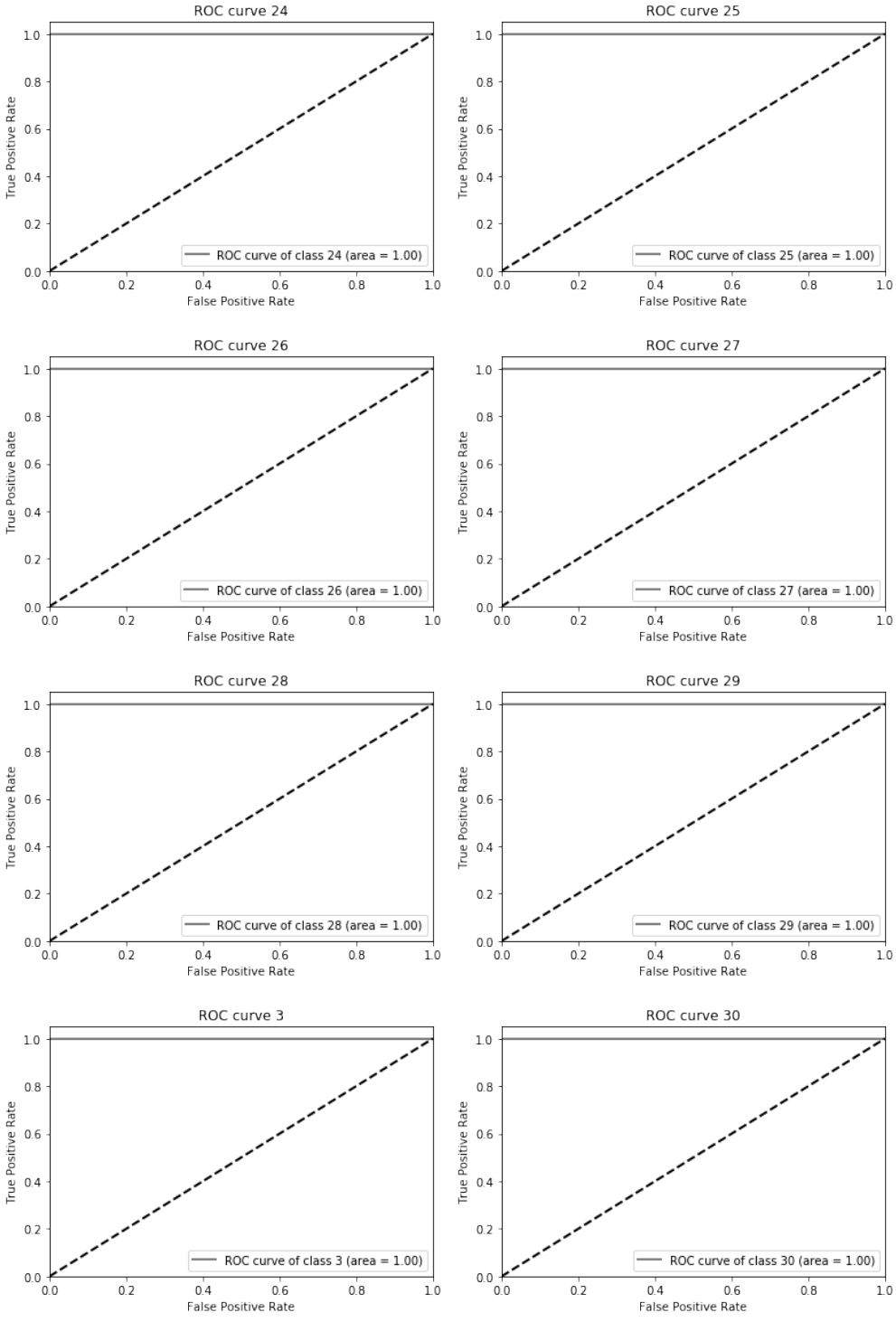

ROC curve 24
ROC curve of class 24 (area = 1.00)
ROC curve 25
ROC curve of class 25 (area = 1.00)
ROC curve 26
ROC curve of class 26 (area = 1.00)
ROC curve 27
ROC curve of class 27 (area = 1.00)
ROC curve 28
ROC curve of class 28 (area = 1.00)
ROC curve 29
ROC curve of class 29 (area = 1.00)
ROC curve 3
ROC curve of class 3 (area = 1.00)
ROC curve 30
ROC curve of class 30 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

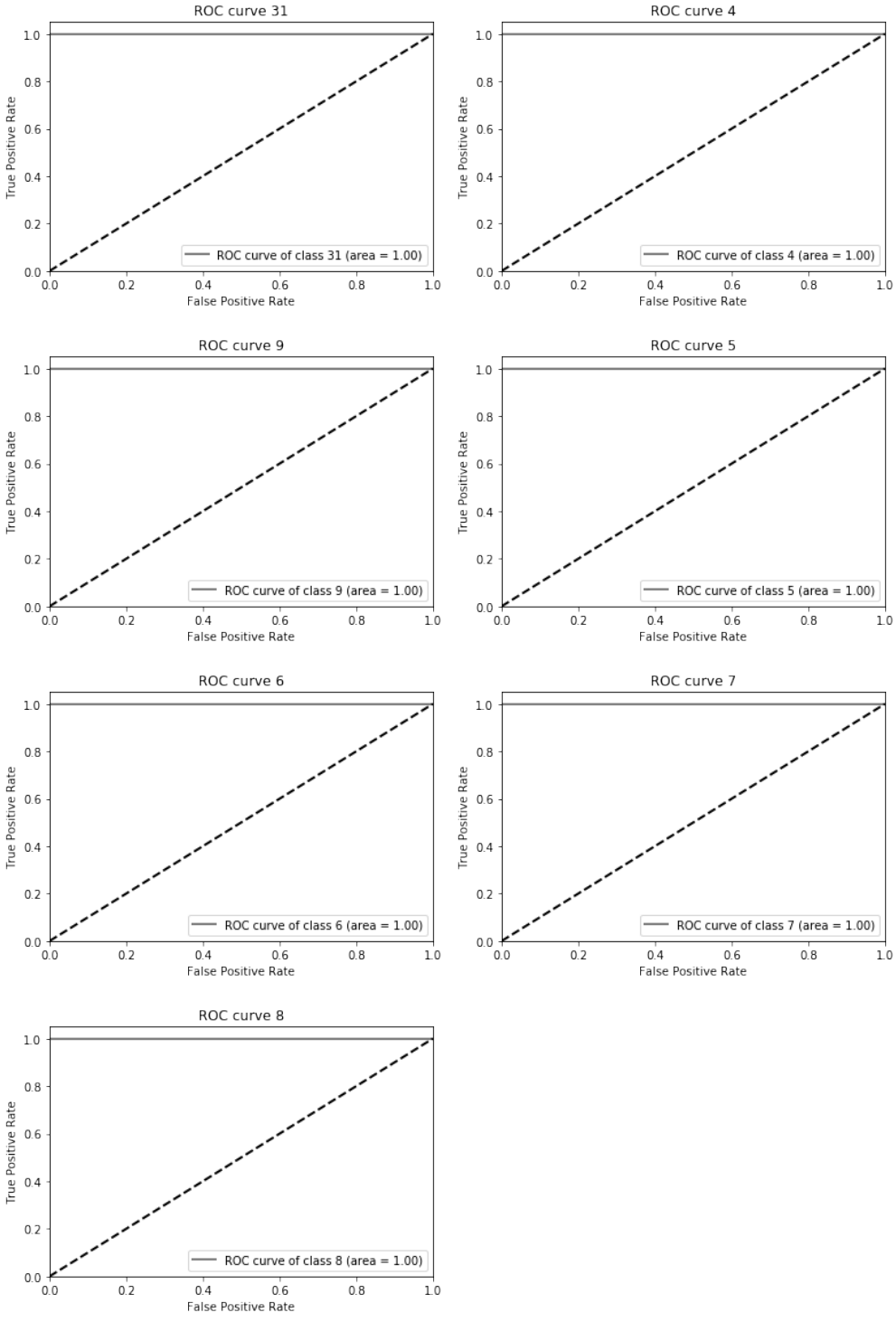

ROC curve 31
True Positive Rate
False Positive Rate
ROC curve of class 31 (area = 1.00)
ROC curve 4
True Positive Rate
False Positive Rate
ROC curve of class 4 (area = 1.00)
ROC curve 5
True Positive Rate
False Positive Rate
ROC curve of class 5 (area = 1.00)
ROC curve 6
True Positive Rate
False Positive Rate
ROC curve of class 6 (area = 1.00)
ROC curve 7
True Positive Rate
False Positive Rate
ROC curve of class 7 (area = 1.00)
ROC curve 8
True Positive Rate
False Positive Rate
ROC curve of class 8 (area = 1.00)
ROC curve 9
True Positive Rate
False Positive Rate
ROC curve of class 9 (area = 1.00)

Lampiran 15 : ROC Subjek 5




ROC curve 17
ROC curve of class 17 (area = 1.00)
ROC curve 18
ROC curve of class 18 (area = 1.00)
ROC curve 19
ROC curve of class 19 (area = 1.00)
ROC curve 2
ROC curve of class 2 (area = 1.00)
ROC curve 20
ROC curve of class 20 (area = 1.00)
ROC curve 21
ROC curve of class 21 (area = 1.00)
ROC curve 22
ROC curve of class 22 (area = 1.00)
ROC curve 23
ROC curve of class 23 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve 24
ROC curve of class 24 (area = 1.00)
ROC curve 25
ROC curve of class 25 (area = 1.00)
ROC curve 26
ROC curve of class 26 (area = 1.00)
ROC curve 27
ROC curve of class 27 (area = 1.00)
ROC curve 28
ROC curve of class 28 (area = 1.00)
ROC curve 29
ROC curve of class 29 (area = 1.00)
ROC curve 3
ROC curve of class 3 (area = 1.00)
ROC curve 30
ROC curve of class 30 (area = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve 31
True Positive Rate
False Positive Rate
ROC curve of class 31 (area = 1.00)
ROC curve 4
True Positive Rate
False Positive Rate
ROC curve of class 4 (area = 1.00)
ROC curve 9
True Positive Rate
False Positive Rate
ROC curve of class 9 (area = 1.00)
ROC curve 5
True Positive Rate
False Positive Rate
ROC curve of class 5 (area = 1.00)
ROC curve 6
True Positive Rate
False Positive Rate
ROC curve of class 6 (area = 1.00)
ROC curve 7
True Positive Rate
False Positive Rate
ROC curve of class 7 (area = 1.00)
ROC curve 8
True Positive Rate
False Positive Rate
ROC curve of class 8 (area = 1.00)

Lampiran 16 : Source Code

```
import numpy as np
import pickle
import cv2
import keras
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.convolutional import Convolution2D, MaxPooling2D, ZeroPadding2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import adam
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import matplotlib
import pandas as pd
from skimage.color import rgb2gray
from skimage.transform import rescale, resize, downscale_local_mean
from skimage.feature import local_binary_pattern
from sklearn.model_selection import train_test_split
from skimage import data, color, feature
from skimage.feature import hog
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Input, Activation, Dense, Conv2D, MaxPooling2D, ZeroPadding2D
, Flatten
from keras.optimizers import Adam
from keras.layers.normalization import BatchNormalization
from keras.utils.np_utils import to_categorical
from keras.callbacks import TensorBoard
from sklearn.model_selection import train_test_split
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
INIT_LR = 1e-3
BS = 32
default_image_size = tuple((224, 224))
image_size = 0
directory_root = '/content/drive/My Drive/subjek3/subjek3'
width=224
height=224
depth=3
```

```
def convert_image_to_array(image_directory):

    try:
        image = cv2.imread(image_directory)

        if image is not None :

            image = cv2.resize(image, default_image_size)

            return img_to_array(image)
        else :
            return np.array([])

    except Exception as e:
        print(f"Error : {e}")

        return None
```

```
image_list, label_list = [], []

try:
    print("[INFO] Loading images ...")
    root_dir = listdir(directory_root)
    for directory in root_dir :
        # remove .DS_Store from list
        if directory == ".DS_Store" :
            root_dir.remove(directory)

    for hand_folder in root_dir :
        hand_posture_folder_list = listdir(f"{directory_root}/{hand_folder}")
        print(directory_root)
        for posture_folder in hand_posture_folder_list :
            # remove .DS_Store from list
            if posture_folder == ".DS_Store" :
                hand_posture_folder_list.remove(posture_folder)

        for hand_posture_folder in hand_posture_folder_list:
            print(f"[INFO] Processing {hand_posture_folder} ...")
            hand_posture_image_list = listdir(f"{directory_root}/{hand_folder}/{hand_p
osture_folder}")

            for single_hand_posture_image in hand_posture_image_list :
                if single_hand_posture_image == ".DS_Store" :
                    hand_posture_image_list.remove(single_hand_posture_image)

            for image in hand_posture_image_list[:30]:

                image_directory = f"{directory_root}/{hand_folder}/{hand_posture_folde
r}/{image}"
                if image_directory.endswith(".jpeg") == True or image_directory.endswi
th(".JPEG") == True:

                    image_list.append(convert_image_to_array(image_directory))

                    label_list.append(hand_posture_folder)
    print("[INFO] Image loading completed")
except Exception as e:
    print(f"Error : {e}")
```

```
image_size = len(image_list)
image_size
```

```
label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
n_classes = len(label_binarizer.classes_)
```

```
print(label_binarizer.classes_)
print(n_classes)
```

```
np_image_list = np.array(image_list, dtype=np.float16) / 225.0
```

```
x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels, test_
size=0.2, random_state = 42)
```

```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=0.2,horizontal_flip=True,
    fill_mode="nearest")
```

```
inputShape = (height, width, depth)
chanDim = -1
if K.image_data_format() == "channels_first":
    inputShape = (depth, height, width)
    chanDim = 1
model = Sequential()
model.add(Conv2D(input_shape=(224,224,3),filters=64,kernel_size=(3,3),padding="same",
activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```

```
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Flatten())
model.add(Dense(units=4096,activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Dense(units=4096,activation="relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Dense(units=n_classes, activation="softmax"))
model.summary()
```

```
opt = adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
# distribution
model.compile(loss="categorical_crossentropy", optimizer=opt,metrics=["accuracy"])
# train the network
print("[INFO] training network...")
```

```
history = model.fit_generator(
    aug.flow(x_train, y_train, batch_size=BS),
    validation_data=(x_test, y_test),
    steps_per_epoch=len(x_train)/BS,
    epochs=EPOCHS, verbose=1
    )
```

```
val_acc = history.history['val_acc']
acc = history.history['acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
#Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
plt.title('Training and Validation accurarcy')
plt.legend()
```

```
plt.figure()
#Train and validation loss
plt.plot(epochs, val_acc, 'b', label='Validation Accuracy')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Validation Accuracy and Validation loss')
plt.legend()
plt.show()
```

```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

```
from sklearn.metrics import confusion_matrix
import sklearn.metrics as metrics
predictions = model.predict(x_test)
predictions
y_pred = (predictions > 0.5)
```

```
matrix = metrics.confusion_matrix(y_test.argmax(axis=1), y_pred.argmax(axis=1))
matrix
```

```
from sklearn.metrics import classification_report
pred = model.predict(x_test, batch_size=32, verbose=1)
predicted = np.argmax(pred, axis=1)
report = classification_report(np.argmax(y_test, axis=1), predicted)
print(report)
```

```
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt
vizualization = pd.DataFrame(matrix, index = label_binarizer.classes_,
                  columns = label_binarizer.classes_)
plt.figure(figsize = (17,10))
sn.heatmap(vizualization, annot=True)
```

```
for lay in model.layers:
    print(lay.name)
    print(lay.get_weights())
```

```
y_pred = model.predict_proba(x_test)
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, y_pred)
```

```
import numpy as np
from scipy import interp
import matplotlib.pyplot as plt
from itertools import cycle
from sklearn.metrics import roc_curve, auc

n_classes = len(label_binarizer.classes_)
y_score = model.predict(x_test)
# y_score = pred_ada
disease_class = label_binarizer.classes_
```

```
# Plot linewidth.
lw = 2

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

colors = cycle(['grey'])
for i, color in zip(range(n_classes), colors):
    plt.figure(i)
    plt.plot(fpr[i], tpr[i], color=color, lw=lw,
             label='ROC curve of class {0} (area = {1:0.2f})'
             ''.format(disease_class[i], roc_auc[i]))

    plt.plot([0, 1], [0, 1], 'k--', lw=lw)
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('ROC curve {0}'.format(disease_class[i]))
    plt.legend(loc="lower right")
    plt.show()
```