

```
import matplotlib.pyplot as plt
import seaborn as sns
from PIL import Image
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve, auc
import tensorflow as tf
from tensorflow.keras import layers, models
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import precision_recall_fscore_support
import cv2
from google.colab import drive

Import Dataset dan Eksplorasi Data

drive.mount('/content/drive')
data_folder = "/content/drive/My Drive/BI Dataset"
tumor_types = ["glioma", "meningioma", "pituitary"]

image_paths = []
labels = []

mounted_at = /content/drive

def load_images_from_folder(folder):
    images = []
    labels = []
    for tumor_type_idx, tumor_type in enumerate(tumor_types, start=1):
```

```

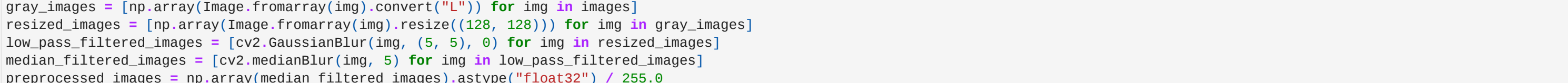
        if len(images) > 0:
            images.append(np.array(img))
            labels.append(tumor_type)
    return images, labels

images, labels = load_images_from_folder(data_folder)

# Bar Plot Kategori Tumor Berdasarkan Gambar
unique_labels, counts = np.unique(labels, return_counts=True)
plt.bar(unique_labels, counts)
plt.xlabel('Kategori Tumor')
plt.ylabel('Jumlah Gambar')
plt.title('Kategori Tumor Berdasarkan Gambar')
plt.show()

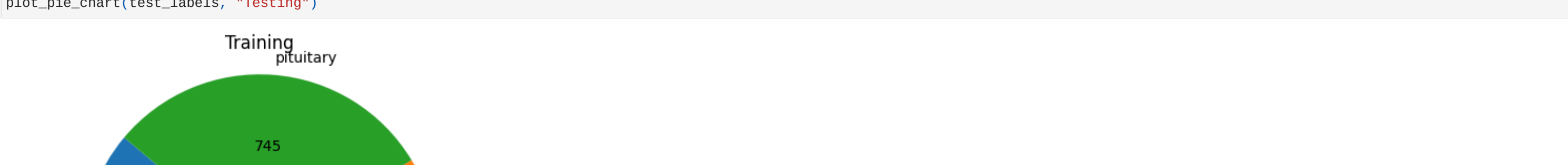
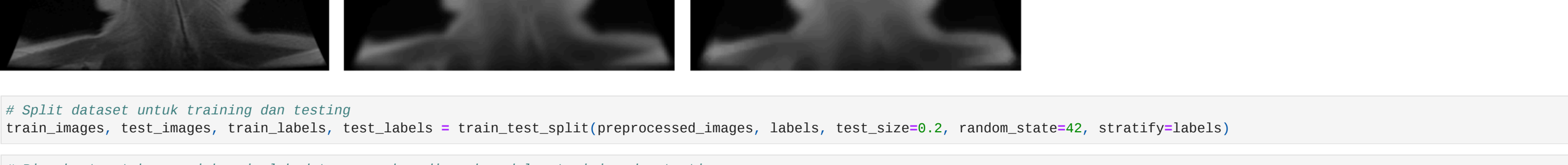
```

Kategori Tumor	Jumlah Gambar
Tumor Benign	650
Tumor Malign	1400
Tumor Tidak Pasti	950



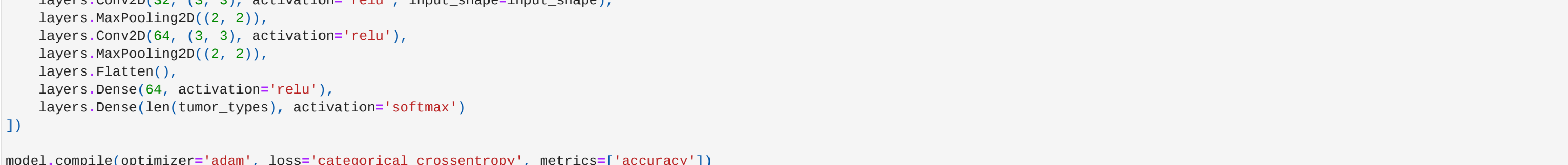
```
# Tampilkan gambar original dan gambar yang sudah di low-pass dan median filter
num_images_to_display = 3
for i in range(num_images_to_display):
    plt.figure(figsize=(10, 4))
    plt.subplot(1, 3, 1)
    plt.imshow(gray_images[i], cmap="gray")
    plt.title("Original")
    plt.axis("off")

    plt.subplot(1, 3, 2)
    plt.imshow(low_pass_filtered_images[i], cmap="gray")
    plt.title("Low-Pass Filtered")
    plt.axis("off")
```



A pie chart illustrating the distribution of cancer types. The chart is divided into three segments: a large orange segment representing colorectal cancer (1149 cases), a blue segment representing glioma (566 cases), and a green segment representing breast cancer (1149 cases). The labels and counts are placed directly on the chart segments.

Cancer Type	Count
glioma	566
colorectal cancer	1149
breast cancer	1149



```
epoch 4/10
02/62 [=====] - 51s 824ms/step - loss: 0.3987 - accuracy: 0.8318 - val_loss: 0.3740 - val_accuracy: 0.8496
epoch 5/10
02/62 [=====] - 49s 797ms/step - loss: 0.3156 - accuracy: 0.8791 - val_loss: 0.3801 - val_accuracy: 0.8537
epoch 6/10
02/62 [=====] - 49s 784ms/step - loss: 0.3011 - accuracy: 0.8821 - val_loss: 0.3633 - val_accuracy: 0.8557
epoch 7/10
02/62 [=====] - 49s 794ms/step - loss: 0.2420 - accuracy: 0.9085 - val_loss: 0.2953 - val_accuracy: 0.9065
epoch 8/10
```

```
y_pred = pd.DataFrame(test_labels_pred)

# Menghitung Confusion Matrix
cm = confusion_matrix(test_labels_encoded, test_labels_pred)
class_labels = ['glioma', 'meningioma', 'pituitary']

for i, label in enumerate(class_labels):
    TP = cm[i, i]
    FN = np.sum(cm[i, :]) - TP
```

```
FP = np.sum(cm[:, 1]) - TP
TN = np.sum(cm) - TP - FP - FN

true_positive_rate = TP / (TP + FN)
true_negative_rate = TN / (TN + FP)
false_positive_rate = FP / (FP + TN)
false_negative_rate = FN / (FN + TP)

print(f"Metrics for class '{label}':")
```

```

false Positive Rate: 0.04620048620048623
false Negative Rate: 0.010752688172043012

# Plot heatmap confusion matrix
plt.figure(figsize=(8, 6))
sns.set(font_scale=1.2)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False, annot_kws={"size": 12}, xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')

```

	glioblastoma	astrocytoma	oligodendroglioma
glioblastoma	100	25	21
astrocytoma	29	252	6

	giloma		meningioma Predicted		pituitary
	precision	recall	f1-score	support	
giloma	0.78	0.74	0.76	142	
meningioma	0.81	0.88	0.90	287	
pituitary	0.90	0.99	0.94	106	

	0.78	0.86	187
meningioma	0.91	0.88	287
pituitary	0.90	0.99	186
accuracy		0.88	615
macro avg	0.86	0.87	615
weighted avg	0.88	0.88	615

```

!jupyter nbconvert --to html " /CNN_20Mar" --output-dir=" /"

```

```
weighted avg      0.66      0.66      0.66      0.15
```

```
!jupyter nbconvert --to html "/CNN_20Mar" --output-dir="/"
```

```
[NbConvertApp] Converting notebook ./CNN_20Mar.ipynb to html
[NbConvertApp] WARNING | Alternative text is missing on 7 image(s).
[NbConvertApp] Writing 1029624 bytes to CNN_20Mar.html
```

In []: