Proyek Akhir : Klasifikasi Gambar

- Nama: Farrel Rasyad
- Email: farrelrasyad.frr@gmail.com (mailto:farrelrasyad.frr@gmail.com)
- Id Dicoding: farrel_rasyad_eypa

In [1]:
```
# from google.colab import drive
# drive.mount('/content/drive')
```

In [2]:
```
# local_zip = '/content/drive/MyDrive/rockpaperscissors.zip'
# zip_ref = zipfile.ZipFile(local_zip, 'r')
# zip_ref.extractall('/content/drive/MyDrive/extractedData')
# zip_ref.close()
```

In [3]:
```
# base_dir = '/content/drive/MyDrive/extractedData/rockpaperscissors/rps-cv-im
```

Import semua library yang dibutuhkan

In [4]:
```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import zipfile, os
from google.colab import files
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Import dataset

In [5]:
```
!wget --no-check-certificate \
  https://github.com/dicodingacademy/assets/releases/download/release/rockpape
  -O /tmp/rockpaperscissors.zip #import dari fungsi wget agar dataset tidak ad
```

```
--2023-12-05 02:13:26--  https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip (https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip)
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-b65867166957?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20231205%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20231205T021326Z&X-Amz-Expires=300&X-Amz-Signature=10c2a7b1865c4ce1b1f6f887d718a8acef07e32d0e223e1f2702b7e424c4c5ad&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=391417272&response-content-disposition=attachment%3B%20filename%3Drockpaperscissors.zip&response-content-type=application%2Foctet-stream (https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-b65867166957?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20231205%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20231205T021326Z&X-Amz-Expires=300&X-Amz-Signature=10c2a7b1865c4ce1b1f6f887d718a8acef07e32d0e223e1f2702b7e424c4c5ad&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=391417272&response-content-disposition=attachment%3B%20filename%3Drockpaperscissors.zip&response-content-type=application%2Foctet-stream) [following]
--2023-12-05 02:13:26--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-b65867166957?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20231205%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20231205T021326Z&X-Amz-Expires=300&X-Amz-Signature=10c2a7b1865c4ce1b1f6f887d718a8acef07e32d0e223e1f2702b7e424c4c5ad&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=391417272&response-content-disposition=attachment%3B%20filename%3Drockpaperscissors.zip&response-content-type=application%2Foctet-stream (https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-b65867166957?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20231205%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20231205T021326Z&X-Amz-Expires=300&X-Amz-Signature=10c2a7b1865c4ce1b1f6f887d718a8acef07e32d0e223e1f2702b7e424c4c5ad&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=391417272&response-content-disposition=attachment%3B%20filename%3Drockpaperscissors.zip&response-content-type=application%2Foctet-stream)
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 322873683 (308M) [application/octet-stream]
Saving to: ‘/tmp/rockpaperscissors.zip’

/tmp/rockpaperscis 100%[===================>] 307.92M   185MB/s    in 1.7s

2023-12-05 02:13:28 (185 MB/s) - ‘/tmp/rockpaperscissors.zip’ saved [322873683/322873683]
```

Unzip dan Assign dataset base directory

In [6]:
```
local_zip = '/tmp/rockpaperscissors.zip' #select zipped file
zip_ref = zipfile.ZipFile(local_zip, 'r') #read zipped file
zip_ref.extractall('/tmp') #extract to location
zip_ref.close() #close zipped file

base_dir = '/tmp/rockpaperscissors/rps-cv-images' #directory yang memuat file
```

In [7]:
```
os.listdir(base_dir)
```

Out[7]:
```
['scissors', 'paper', 'README_rpc-cv-images.txt', 'rock']
```

Image Data Generator

In [8]:
```
train_datagen = ImageDataGenerator(
                    rescale=1./255, # rescaling factor. If None or 0, no resca
                    rotation_range=20, # Int. Degree range for random rotation
                    horizontal_flip=True, # Boolean. Randomly flip inputs hori
                    vertical_flip=True, # Boolean. Randomly flip inputs vertic
                    shear_range = 0.2, # Float. Shear Intensity (Shear angle i
                    fill_mode = 'nearest', # Points outside the boundaries of
                    validation_split=0.4,) # split data menjadi 40% validation


# test_datagen = ImageDataGenerator( #no actual need for this i guess because
#                     rescale=1./255,
#                     validation_split=0.4,)

# source for ImageDataGenerator: https://www.tensorflow.org/api_docs/python/tf
```

Pembagian Dataset Train dan Test

In [9]:
```
train_generator = train_datagen.flow_from_directory(
        base_dir, # folder/size yang digunakan
        target_size=(200, 300),  # ubah resolusi gambar menjadi 200 x 300 (dic
        batch_size=4,
        class_mode='categorical', #categorical karena pilihan output lebih dar
        subset="training") # karena di split dari train_datagen menggunakan va

validation_generator = train_datagen.flow_from_directory(
        base_dir,
        target_size=(200, 300),
        batch_size=4,
        class_mode='categorical',
        subset="validation") # karena di split dari train_datagen menggunakan

# source for flow_from_directory: https://www.tensorflow.org/api_docs/python/t
```

```
Found 1314 images belonging to 3 classes.
Found 874 images belonging to 3 classes.
```

Sequential Model

In [10]:
```
# ini dari contoh
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(200, 300
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(256, (3,3), activation='relu'), #254 was
    # tf.keras.layers.MaxPooling2D(2,2),
    # tf.keras.layers.Conv2D(512, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    # tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax') #diganti 1->3 , sigmoid->so
])

# contoh penggunaan dan penjelasan dari softmax https://youtu.be/7HPwo4wnJeA?t=
```

In [11]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 198, 298, 32)      896

 max_pooling2d (MaxPooling2  (None, 99, 149, 32)       0
 D)

 conv2d_1 (Conv2D)           (None, 97, 147, 64)       18496

 max_pooling2d_1 (MaxPoolin  (None, 48, 73, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 46, 71, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 23, 35, 128)       0
 g2D)

 conv2d_3 (Conv2D)           (None, 21, 33, 256)       295168

 max_pooling2d_3 (MaxPoolin  (None, 10, 16, 256)       0
 g2D)

 flatten (Flatten)           (None, 40960)             0

 dense (Dense)               (None, 512)               20972032

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 21361987 (81.49 MB)
Trainable params: 21361987 (81.49 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Model Compile and Fit

In [12]:
```
model.compile(loss='categorical_crossentropy', #menggunakan 'categorical_cross
              optimizer=tf.optimizers.Adam(),
              metrics=['accuracy'])
```

In [13]:
```
history = model.fit( #di set ke variabel 'history' untuk plotting data epochs
      train_generator, # train data
      steps_per_epoch=25, # Total number of steps (batches of samples) before
      epochs=30, # 15 epochs baru bisa membedakan gunting dari lainnya dengan
      validation_data=validation_generator, # validation data
      validation_steps=5, # Total number of steps (batches of samples) to draw
      verbose=2) #verbosity mode for information per epoch

# source for .fit: https://www.tensorflow.org/api_docs/python/tf/keras/Model#f
```

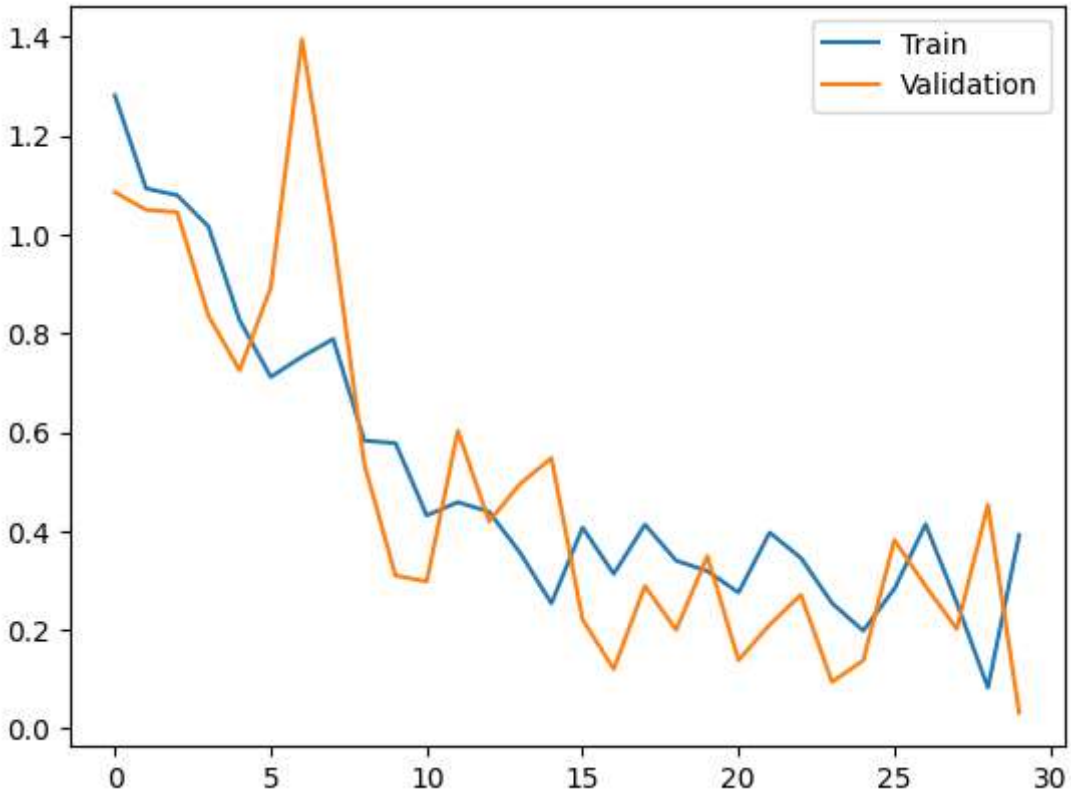
```
Epoch 1/30
25/25 - 46s - loss: 1.2809 - accuracy: 0.3900 - val_loss: 1.0853 - val_accura
cy: 0.5500 - 46s/epoch - 2s/step
Epoch 2/30
25/25 - 31s - loss: 1.0926 - accuracy: 0.3000 - val_loss: 1.0496 - val_accura
cy: 0.4000 - 31s/epoch - 1s/step
Epoch 3/30
25/25 - 30s - loss: 1.0790 - accuracy: 0.4200 - val_loss: 1.0446 - val_accura
cy: 0.7500 - 30s/epoch - 1s/step
Epoch 4/30
25/25 - 30s - loss: 1.0157 - accuracy: 0.5800 - val_loss: 0.8356 - val_accura
cy: 0.4500 - 30s/epoch - 1s/step
Epoch 5/30
25/25 - 29s - loss: 0.8262 - accuracy: 0.6600 - val_loss: 0.7247 - val_accura
cy: 0.6000 - 29s/epoch - 1s/step
Epoch 6/30
25/25 - 29s - loss: 0.7116 - accuracy: 0.6800 - val_loss: 0.8921 - val_accura
cy: 0.6000 - 29s/epoch - 1s/step
Epoch 7/30
25/25 - 29s - loss: 0.7524 - accuracy: 0.6800 - val_loss: 1.3949 - val_accura
cy: 0.4500 - 29s/epoch - 1s/step
Epoch 8/30
25/25 - 29s - loss: 0.7882 - accuracy: 0.6900 - val_loss: 1.0005 - val_accura
cy: 0.4500 - 29s/epoch - 1s/step
Epoch 9/30
25/25 - 29s - loss: 0.5827 - accuracy: 0.7800 - val_loss: 0.5351 - val_accura
cy: 0.8000 - 29s/epoch - 1s/step
Epoch 10/30
25/25 - 29s - loss: 0.5775 - accuracy: 0.7800 - val_loss: 0.3099 - val_accura
cy: 0.9000 - 29s/epoch - 1s/step
Epoch 11/30
25/25 - 28s - loss: 0.4315 - accuracy: 0.8300 - val_loss: 0.2978 - val_accura
cy: 0.9000 - 28s/epoch - 1s/step
Epoch 12/30
25/25 - 28s - loss: 0.4583 - accuracy: 0.8200 - val_loss: 0.6025 - val_accura
cy: 0.7500 - 28s/epoch - 1s/step
Epoch 13/30
25/25 - 30s - loss: 0.4389 - accuracy: 0.8400 - val_loss: 0.4192 - val_accura
cy: 0.8000 - 30s/epoch - 1s/step
Epoch 14/30
25/25 - 29s - loss: 0.3556 - accuracy: 0.9100 - val_loss: 0.4957 - val_accura
cy: 0.8000 - 29s/epoch - 1s/step
Epoch 15/30
25/25 - 28s - loss: 0.2538 - accuracy: 0.9200 - val_loss: 0.5475 - val_accura
cy: 0.6500 - 28s/epoch - 1s/step
Epoch 16/30
25/25 - 29s - loss: 0.4068 - accuracy: 0.8200 - val_loss: 0.2209 - val_accura
cy: 0.9500 - 29s/epoch - 1s/step
Epoch 17/30
25/25 - 28s - loss: 0.3132 - accuracy: 0.9000 - val_loss: 0.1206 - val_accura
cy: 0.9500 - 28s/epoch - 1s/step
Epoch 18/30
25/25 - 36s - loss: 0.4131 - accuracy: 0.8500 - val_loss: 0.2887 - val_accura
cy: 0.9000 - 36s/epoch - 1s/step
Epoch 19/30
25/25 - 29s - loss: 0.3400 - accuracy: 0.8900 - val_loss: 0.1998 - val_accura
cy: 0.9500 - 29s/epoch - 1s/step
```

```
Epoch 20/30
25/25 - 29s - loss: 0.3189 - accuracy: 0.8400 - val_loss: 0.3492 - val_accura
cy: 0.8500 - 29s/epoch - 1s/step
Epoch 21/30
25/25 - 29s - loss: 0.2756 - accuracy: 0.9000 - val_loss: 0.1386 - val_accura
cy: 0.9500 - 29s/epoch - 1s/step
Epoch 22/30
25/25 - 31s - loss: 0.3967 - accuracy: 0.8800 - val_loss: 0.2107 - val_accura
cy: 0.9000 - 31s/epoch - 1s/step
Epoch 23/30
25/25 - 29s - loss: 0.3455 - accuracy: 0.8800 - val_loss: 0.2707 - val_accura
cy: 0.9500 - 29s/epoch - 1s/step
Epoch 24/30
25/25 - 29s - loss: 0.2540 - accuracy: 0.9000 - val_loss: 0.0947 - val_accura
cy: 1.0000 - 29s/epoch - 1s/step
Epoch 25/30
25/25 - 28s - loss: 0.1981 - accuracy: 0.9592 - val_loss: 0.1381 - val_accura
cy: 0.9000 - 28s/epoch - 1s/step
Epoch 26/30
25/25 - 29s - loss: 0.2831 - accuracy: 0.9100 - val_loss: 0.3818 - val_accura
cy: 0.9000 - 29s/epoch - 1s/step
Epoch 27/30
25/25 - 29s - loss: 0.4136 - accuracy: 0.8400 - val_loss: 0.2885 - val_accura
cy: 0.9500 - 29s/epoch - 1s/step
Epoch 28/30
25/25 - 28s - loss: 0.2548 - accuracy: 0.9000 - val_loss: 0.2019 - val_accura
cy: 0.9000 - 28s/epoch - 1s/step
Epoch 29/30
25/25 - 28s - loss: 0.0832 - accuracy: 0.9700 - val_loss: 0.4528 - val_accura
cy: 0.8000 - 28s/epoch - 1s/step
Epoch 30/30
25/25 - 29s - loss: 0.3915 - accuracy: 0.8776 - val_loss: 0.0325 - val_accura
cy: 1.0000 - 29s/epoch - 1s/step
```

Model Epoch Plotting

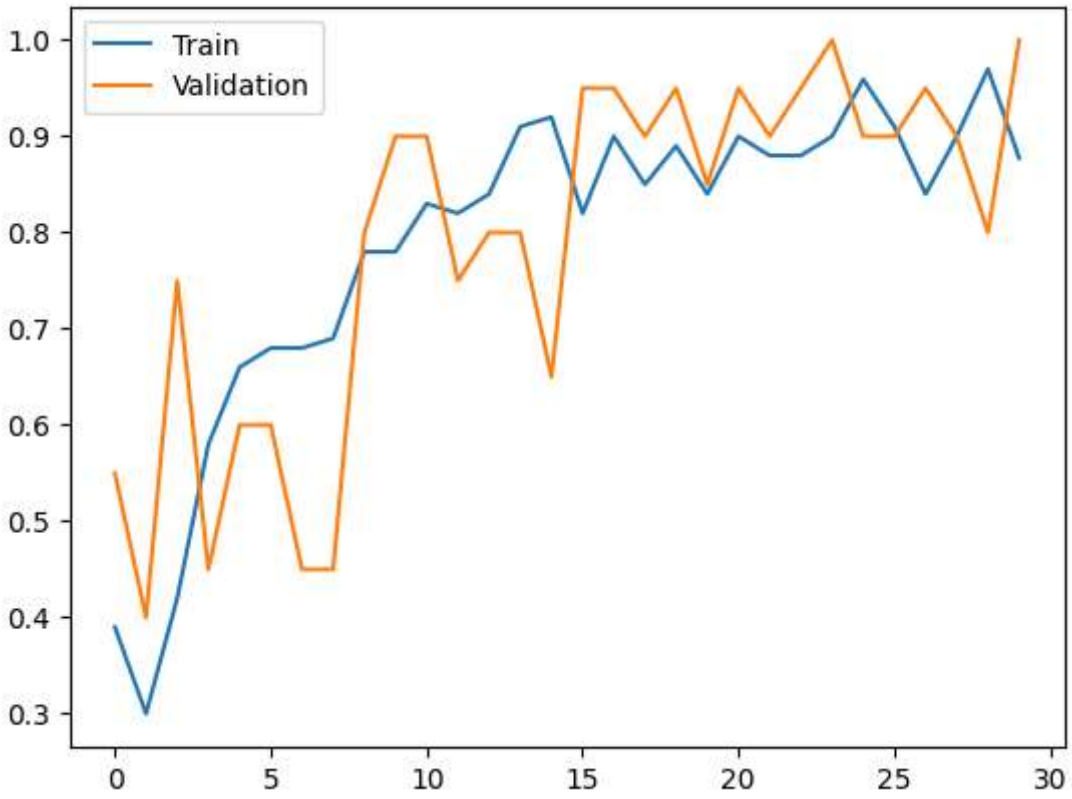
In [14]:
```
plt.plot(history.history['loss'],label="Train")
plt.plot(history.history['val_loss'],label="Validation")
plt.legend()
```

Out[14]:
```
<matplotlib.legend.Legend at 0x7b1a7c2deda0>
```



In [15]:
```
plt.plot(history.history['accuracy'],label="Train")
plt.plot(history.history['val_accuracy'],label="Validation")
plt.legend()
```

Out[15]:
```
<matplotlib.legend.Legend at 0x7b1a6ff26860>
```



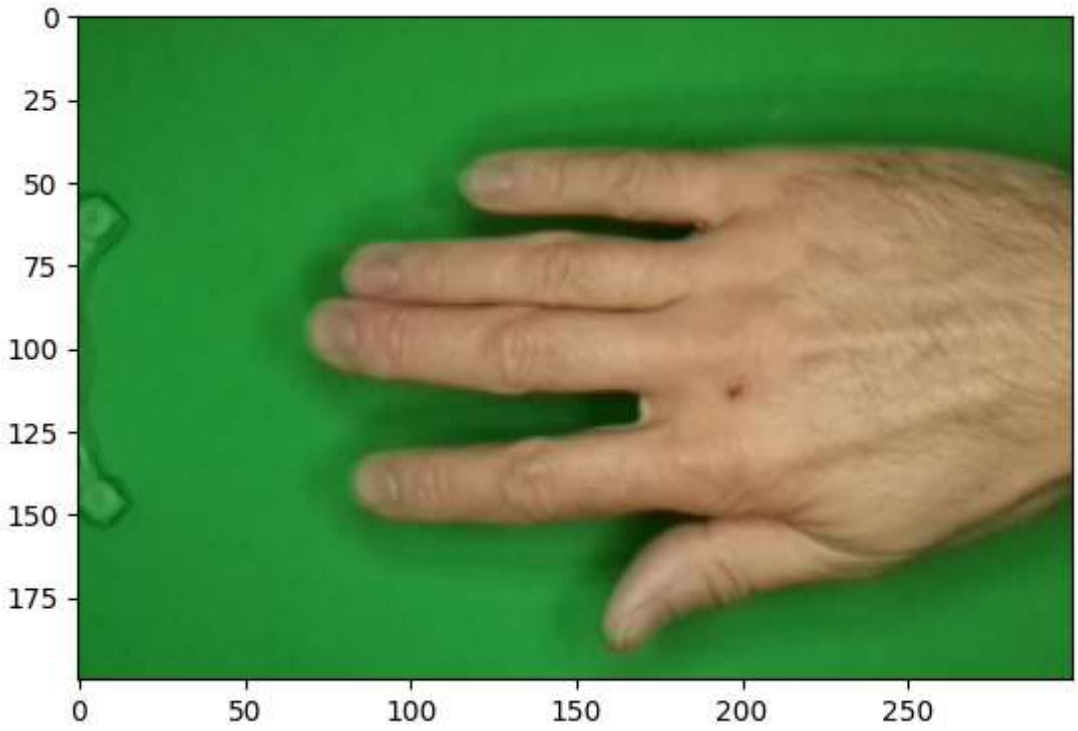
Predicting image from user input

In [16]:
```
%matplotlib inline

uploaded = files.upload()

for fn in uploaded.keys():

  path = fn
  img = image.load_img(path, target_size=(200,300))

  imgplot = plt.imshow(img)
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)

  images = np.vstack([x])
  classes = model.predict(images, batch_size=10)
  classed = np.argmax(classes)

  print(classes)
  print(classed)
  print("Object is:")

  if classes[0,0] == 1:
    print('paper')
  elif classes[0,1] == 1:
    print('rock')
  elif classes[0,2] == 1:
    print('scissors')
  else:
    print('Unknown...')
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving paper.png to paper.png
1/1 [==============================] - 0s 238ms/step
[[1. 0. 0.]]
0
Object is:
paper
```



kekurangan dari dataset ini adalah model yang dibuat menggunakan dataset ini hanya bisa memprediksi bentuk tangan dengan baik jika backgroundnya seperti yang di dataset (hijau, dan tidak banyak objek dibelakangnya)