DAFTAR LAMPIRAN

Lampiran -1 *Listing* program halaman activity_main.xml

```
<?xmlversion="1.0"
encoding="utf-8"?>
<androidx.constraintlayout.w
idget.ConstraintLayout
xmlns:android="http://schem
as.android.com/apk/res/andro
id"

xmlns:app="http://schemas.a
ndroid.com/apk/res-auto"

xmlns:tools="http://schemas.
android.com/tools"

android:layout_width="match
_parent"

android:layout_height="matc
h_parent"

tools:context=".ui.MainActivi
ty">


<androidx.appcompat.widget.
AppCompatImageView

android:id="@+id/ellipseTop
Start"

android:layout_width="wrap
_content"

android:layout_height="wrap
_content"

android:src="@drawable/ic_e
llipse_4"

app:layout_constraintTop_to
TopOf="parent"

app:layout_constraintEnd_to
EndOf="parent"
    />


<androidx.appcompat.widget.
AppCompatImageView

android:id="@+id/ellipseMid
dle"

android:layout_width="wrap
_content"

android:layout_height="wrap
_content"

android:src="@drawable/ic_e
llipse_6"

app:layout_constraintTop_to
TopOf="parent"

app:layout_constraintStart_to
StartOf="@+id/ellipseTopSta
rt"

app:layout_constraintEnd_to
EndOf="@id/ellipseTopEnd"
```

```
tools:ignore="RtlSymmetry"
/>


<androidx.appcompat.widget.
AppCompatImageView

android:id="@+id/ellipseTop
End"

android:layout_width="wrap
_content"

android:layout_height="wrap
_content"

android:src="@drawable/ic_e
llipse_5"

app:layout_constraintTop_to
TopOf="parent"

android:layout_marginEnd="
120dp"

app:layout_constraintEnd_to
EndOf="@id/ellipseMiddle"

app:layout_constraintStart_to
StartOf="parent"
      />


<androidx.appcompat.widget.
AppCompatImageView

android:id="@+id/imageVie
wHeader"

android:layout_width="match
_parent"

android:layout_height="wrap
_content"

app:layout_constraintEnd_to
EndOf="parent"

app:layout_constraintStart_to
StartOf="parent"

app:layout_constraintTop_to
BottomOf="@id/ellipseTopS
tart"

android:layout_marginStart="
16dp"

android:layout_marginEnd="
16dp"

app:layout_constraintBottom
_toTopOf="@id/containerBut
ton"

app:srcCompat="@drawable/
logo_splash" />


<androidx.constraintlayout.w
idget.ConstraintLayout

android:id="@+id/containerB
utton"

android:layout_width="match
_parent"

android:layout_height="wrap
_content"

android:paddingBottom="16d
p"

android:paddingTop="64dp"

android:elevation="10dp"

app:layout_constraintBottom
_toBottomOf="parent"
```

```
android:background="@drawable/bg_rounded_half"
        >

<androidx.constraintlayout.widget.ConstraintLayout

android:elevation="10dp"

android:id="@+id/button_image_to_text"

android:layout_width="match_parent"

android:layout_height="wrap_content"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toEndOf="parent"

android:layout_marginStart="25dp"

android:layout_marginEnd="25dp"

android:padding="8dp"

android:background="@drawable/bg_btn_detection"
            >

<androidx.appcompat.widget.AppCompatImageView

android:id="@+id/ivDeteksi"

android:layout_width="50dp"

android:layout_height="50dp"

android:src="@drawable/ic_loupe"

app:layout_constraintTop_toTopOf="parent"

android:layout_marginStart="8dp"

android:padding="8dp"

app:layout_constraintStart_toStartOf="parent"
                />

<androidx.appcompat.widget.AppCompatTextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Mulai Deteksi"

android:textColor="#ffff"

app:layout_constraintTop_toTopOf="parent"

android:textSize="24sp"

android:textStyle="bold"

android:layout_marginTop="8dp"

app:layout_constraintStart_toEndOf="@id/ivDeteksi"
```

```
app:layout_constraintEnd_to
EndOf="parent"

android:layout_marginEnd="
20sp"
            />

</androidx.constraintlayout.
widget.ConstraintLayout>


<androidx.constraintlayout.w
idget.ConstraintLayout

android:id="@+id/button_tex
t_to_image"

android:layout_width="match
_parent"

android:layout_height="wrap
_content"

app:layout_constraintTop_to
BottomOf="@id/button_terje
mah"

app:layout_constraintStart_to
StartOf="parent"

app:layout_constraintEnd_to
EndOf="parent"

android:layout_marginStart="
25dp"

android:layout_marginEnd="
25dp"

android:padding="8dp"

android:elevation="10dp"

android:layout_marginTop="
16dp"

android:background="@draw
able/bg_btn_detection"
        >

<androidx.appcompat.widget.
AppCompatImageView

android:id="@+id/ivPetunjuk
Isyarat"

android:layout_width="50dp"

android:layout_height="50dp
"

android:src="@drawable/ic_p
etunjuk_isyarat"

app:layout_constraintTop_to
TopOf="parent"

android:layout_marginStart="
8dp"

android:padding="8dp"

app:layout_constraintStart_to
StartOf="parent"
            />

<androidx.appcompat.widget.
AppCompatTextView

android:layout_width="wrap
_content"

android:layout_height="wrap
_content"

android:text="Petunjuk
Isyarat"

android:textColor="#ffff"
```

```
app:layout_constraintTop_to
TopOf="parent"

android:textSize="24sp"

android:textStyle="bold"

android:layout_marginTop="
8dp"

app:layout_constraintStart_to
EndOf="@id/ivPetunjukIsyar
at"

app:layout_constraintEnd_to
EndOf="parent"

android:layout_marginEnd="
20sp"
            />

</androidx.constraintlayout.
widget.ConstraintLayout>


<androidx.constraintlayout.w
idget.ConstraintLayout

android:elevation="1dp"

android:id="@+id/button_terj
emah"

android:layout_width="match
_parent"

android:layout_height="wrap
_content"

app:layout_constraintTop_to
BottomOf="@id/button_ima
ge_to_text"

app:layout_constraintStart_to
StartOf="parent"

app:layout_constraintEnd_to
EndOf="parent"

android:layout_marginStart="
25dp"

android:layout_marginEnd="
25dp"

android:padding="8dp"

android:layout_marginTop="
16dp"

android:background="@draw
able/bg_btn_detection"
            >

<androidx.appcompat.widget.
AppCompatImageView

android:id="@+id/ivTerjema
h"

android:layout_width="50dp"

android:layout_height="50dp
"

android:src="@drawable/tran
slator"

app:layout_constraintTop_to
TopOf="parent"

android:layout_marginStart="
8dp"

android:padding="8dp"

app:layout_constraintStart_to
StartOf="parent"
            />
```

```
<androidx.appcompat.widget.
AppCompatTextView

android:layout_width="wrap
_content"

android:layout_height="wrap
_content"

android:text="Huruf ke
isyarat"

android:textColor="#ffff"

app:layout_constraintTop_to
TopOf="parent"

android:textSize="24sp"

android:textStyle="bold"

android:layout_marginTop="
8dp"

app:layout_constraintStart_to
EndOf="@id/ivTerjemah"

app:layout_constraintEnd_to
EndOf="parent"

android:layout_marginEnd="
20sp"
        />

</androidx.constraintlayout.
widget.ConstraintLayout>


<androidx.constraintlayout.w
idget.ConstraintLayout

android:elevation="1dp"

android:id="@+id/button_ab
out_us"

android:layout_width="match
_parent"

android:layout_height="wrap
_content"

app:layout_constraintTop_to
BottomOf="@id/button_text
_to_image"

app:layout_constraintStart_to
StartOf="parent"

app:layout_constraintEnd_to
EndOf="parent"

android:layout_marginStart="
25dp"

android:layout_marginEnd="
25dp"

android:padding="8dp"

android:layout_marginTop="
16dp"

android:background="@draw
able/bg_btn_detection"
        >

<androidx.appcompat.widget.
AppCompatImageView

android:id="@+id/ivInformat
ion"

android:layout_width="50dp"

android:layout_height="50dp
"

android:src="@drawable/ic_i
nformation"
```

```
app:layout_constraintTop_to
TopOf="parent"

android:layout_marginStart="
8dp"

android:padding="8dp"

app:layout_constraintStart_to
StartOf="parent"
            />

<androidx.appcompat.widget.
AppCompatTextView

android:layout_width="wrap
_content"

android:layout_height="wrap
_content"

android:text="Informasi"

android:textColor="#ffff"

app:layout_constraintTop_to
TopOf="parent"

android:textSize="24sp"
android:textStyle="bold"

android:layout_marginTop="
8dp"

app:layout_constraintStart_to
EndOf="@id/ivInformation"

app:layout_constraintEnd_to
EndOf="parent"

android:layout_marginEnd="
20sp"
            />

</androidx.constraintlayout.
widget.ConstraintLayout>

</androidx.constraintlayout.
widget.ConstraintLayout>

</androidx.constraintlayout.widget.C
onstraintLayout>
```

Lampiran -2 *Listing* program halaman MainActivity.kt

```
package org.tensorflow.lite.examples.classification.ui

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.constraintlayout.widget.ConstraintLayout
import org.tensorflow.lite.examples.classification.ClassifierActivity
import org.tensorflow.lite.examples.classification.R

class MainActivity : AppCompatActivity() {
  lateinit var button_image_to_text: ConstraintLayout
  lateinit var button_about_us: ConstraintLayout
  lateinit var button_text_to_image: ConstraintLayout
  lateinit var button_terjemahan: ConstraintLayout

  override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity_main)


button_image_to_text = findViewById<ConstraintLayout>(R.id.button_image_to_text)
    button_about_us = findViewById<ConstraintLayout>(R.id.button_about_us)

button_text_to_image = findViewById<ConstraintLayout>(R.id.button_text_to_image)
    button_terjemahan = findViewById<ConstraintLayout>(R.id.button_terjemah)


button_image_to_text.setOnClickListener {
      val intent = Intent(this@MainActivity,
```

```
ClassifierActivity::class.java)

startActivity(intent)
        }

button_about_us.setOnClickListener {
            val intent = Intent(this@MainActivity, AboutUsActivity::class.java)

startActivity(intent)
        }

button_text_to_image.setOnClickListener {
            val intent = Intent(this@MainActivity, PetunjukIsyaratActivity::class.java)

startActivity(intent)
        }


button_terjemahan.setOnClickListener {
            val intent = Intent(this@MainActivity, TerjemahanIsyarat::class.java)

startActivity(intent)
        }
    }
}
```

Lampiran -3 *Listing* program halaman CameraActivity.java

```
package org.tensorflow.lite.examples.classification;

import android.Manifest;
import android.app.Fragment;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.hardware.Camera;
import android.hardware.camera2.CameraAccessException;
import android.hardware.camera2.CameraCharacteristics;
import android.hardware.camera2.CameraManager;
import android.hardware.camera2.params.StreamConfigurationMap;
import android.media.Image;
import android.media.Image.Plane;
import android.media.ImageReader;
import android.media.ImageReader.OnImageAvailableListener;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.Trace;
import androidx.annotation.NonNull;
import androidx.annotation.UiThread;
import androidx.appcompat.app.AppCompatActivity;

import android.util.Log;
import android.util.Size;
import android.view.Surface;
import android.view.View;
```

```
import android.view.ViewTreeObserver;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.material.bottomsheet.BottomSheetBehavior;
import java.nio.ByteBuffer;
import java.util.List;
import org.tensorflow.lite.examples.classification.env.ImageUtils;
import org.tensorflow.lite.examples.classification.env.Logger;
import org.tensorflow.lite.examples.classification.ui.MainActivity;
import org.tensorflow.lite.examples.classification.tflite.Classifier.Device;
import org.tensorflow.lite.examples.classification.tflite.Classifier.Model;
import org.tensorflow.lite.examples.classification.tflite.Classifier.Recognition;

public abstract class CameraActivity extends AppCompatActivity
    implements OnImageAvailableListener,

Camera.PreviewCallback,

View.OnClickListener,

AdapterView.OnItemSelectedListener {
  private static final Logger LOGGER = new Logger();

  private static final int PERMISSIONS_REQUEST = 1;
```

```
 private static final String PERMISSION_CAMERA = Manifest.permission.CAMERA;
 protected int previewWidth = 0;
 protected int previewHeight = 0;
 private Handler handler;
 private HandlerThread handlerThread;
 private boolean useCamera2API;
 private boolean isProcessingFrame = false;
 private byte[][] yuvBytes = new byte[3][];
 private int[] rgbBytes = null;
 private int yRowStride;
 private Runnable postInferenceCallback;
 private Runnable imageConverter;
 private LinearLayout bottomSheetLayout;
 private LinearLayout gestureLayout;
 private BottomSheetBehavior<LinearLayout> sheetBehavior;
 protected TextView recognitionTextView, recognitionValueTextView,
     composed_text;
 protected TextView inferenceTimeTextView;
 protected ImageView bottomSheetArrowImageView;
 private ImageView plusImageView, minusImageView;
 private Spinner deviceSpinner;
 private TextView threadsTextView;

 private Model model = Model.QUANTIZED_MOBILENET;
 private Device device = Device.CPU;
 private int numThreads = -1;

 protected Button
         button_space,
         button_delete,
         button_sweep;

 protected ImageView button_back;

 protected String active_latter;
 protected int active_latter_number = 0, composedSpeed = 5;
```

```
  @Override
  protected void onCreate(final Bundle savedInstanceState) {
    LOGGER.d("onCreate " + this);
    super.onCreate(null);

getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);


setContentView(R.layout.tfe_ic_activity_camera);

    if (hasPermission()) {
      setFragment();
    } else {
      requestPermission();
    }

    threadsTextView = findViewById(R.id.thread);
    plusImageView = findViewById(R.id.plus);
    minusImageView = findViewById(R.id.minus);
    deviceSpinner = findViewById(R.id.device_spinner);
    bottomSheetLayout = findViewById(R.id.bottom_sheet_layout);
    gestureLayout = findViewById(R.id.gesture_layout);
    sheetBehavior = BottomSheetBehavior.from(bottomSheetLayout);

bottomSheetArrowImageView = findViewById(R.id.bottom_sheet_arrow);
    button_back = findViewById(R.id.button_back);
    button_delete = findViewById(R.id.delete);
/*  button_space = findViewById(R.id.space);
    button_sweep = findViewById(R.id.sweep);*/
    composed_text = findViewById(R.id.composed_text);


button_delete.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        String str = composed_text.getText().toString().trim();
        if(str.length()!=0){
```

```
        str  = str.substring(
0, str.length() - 1 );

composed_text.setText  (
str );
       }
     }
   });


/*button_space.setOnClic
kListener(new
View.OnClickListener()
{
     @Override
     public              void
onClick(View view) {

composed_text.setText(c
omposed_text.getText()
+ " ");
     }
   });


button_sweep.setOnClick
Listener(new
View.OnClickListener()
{
     @Override
     public              void
onClick(View view) {

composed_text.setText("
");
     }
   });*/
```

```
button_back.setOnClickL
istener(new
View.OnClickListener()
{
     @Override
     public              void
onClick(View view) {
       Intent intent = new
Intent(CameraActivity.th
is, MainActivity.class);
       startActivity(intent);
     }
   });

   ViewTreeObserver vto
=
gestureLayout.getViewTr
eeObserver();

vto.addOnGlobalLayout
Listener(
       new
ViewTreeObserver.OnGl
obalLayoutListener() {
         @Override
         public              void
onGlobalLayout() {
           if
(Build.VERSION.SDK_I
NT                          <
Build.VERSION_CODE
S.JELLY_BEAN) {

gestureLayout.getViewTr
eeObserver().removeGlo
balOnLayoutListener(thi
s);
         } else {
```

```
gestureLayout.getViewTreeObserver().removeOnGlobalLayoutListener(this);
        }
        // int width = bottomSheetLayout.getMeasuredWidth();
        int height = gestureLayout.getMeasuredHeight();


sheetBehavior.setPeekHeight(height);
        }
      });

sheetBehavior.setHideable(false);


sheetBehavior.setBottomSheetCallback(
      new BottomSheetBehavior.BottomSheetCallback() {
        @Override
        public void onStateChanged(@NonNull View bottomSheet, int newState) {
          switch (newState) {
            case BottomSheetBehavior.STATE_HIDDEN:
              break;
            case BottomSheetBehavior.STATE_EXPANDED:
              {

bottomSheetArrowImageView.setImageResource(R.drawable.icn_chevron_down);
              }
              break;
            case BottomSheetBehavior.STATE_COLLAPSED:
              {

bottomSheetArrowImageView.setImageResource(R.drawable.icn_chevron_up);
              }
              break;
            case BottomSheetBehavior.STATE_DRAGGING:
              break;
            case BottomSheetBehavior.STATE_SETTLING:

bottomSheetArrowImageView.setImageResource(R.drawable.icn_chevron_up);
              break;
          }
        }
```

```
        @Override
        public void
onSlide(@NonNull View
bottomSheet, float
slideOffset) {}
      });

   recognitionTextView
=
findViewById(R.id.detec
ted_item);

recognitionValueTextVie
w =
findViewById(R.id.detec
ted_item_value);


inferenceTimeTextView
=
findViewById(R.id.infer
ence_info);


deviceSpinner.setOnItem
SelectedListener(this);


plusImageView.setOnCli
ckListener(this);

minusImageView.setOn
ClickListener(this);

   device =
Device.valueOf(deviceSp
inner.getSelectedItem().t
oString());

   numThreads =
Integer.parseInt(threadsT
extView.getText().toStri
ng().trim());
  }

  protected int[]
getRgbBytes() {
   imageConverter.run();
   return rgbBytes;
  }

  protected int
getLuminanceStride() {
   return yRowStride;
  }

  protected byte[]
getLuminance() {
   return yuvBytes[0];
  }

  /** Callback for
android.hardware.Camer
a API */
  @Override
  public void
onPreviewFrame(final
byte[] bytes, final
Camera camera) {
   if (isProcessingFrame)
{

LOGGER.w("Dropping
frame!");
     return;
   }

   try {
```

```
    // Initialize the storage bitmaps once when the resolution is known.
    if (rgbBytes == null) {
      Camera.Size previewSize = camera.getParameters().getPreviewSize();
      previewHeight = previewSize.height;
      previewWidth = previewSize.width;
      rgbBytes = new int[previewWidth * previewHeight];

onPreviewSizeChosen(new Size(previewSize.width, previewSize.height), 90);
    }
  } catch (final Exception e) {
    LOGGER.e(e, "Exception!");
    return;
  }

  isProcessingFrame = true;
  yuvBytes[0] = bytes;
  yRowStride = previewWidth;

  imageConverter =
      new Runnable() {
        @Override
        public void run() {

ImageUtils.convertYUV420SPToARGB8888(bytes, previewWidth, previewHeight, rgbBytes);
        }
      };

  postInferenceCallback =
      new Runnable() {
        @Override
        public void run() {

camera.addCallbackBuffer(bytes);

isProcessingFrame = false;
        }
      };
  processImage();
 }

 /** Callback for Camera2 API */
 @Override
 public void onImageAvailable(final ImageReader reader) {
  // We need wait until we have some size from onPreviewSizeChosen
  if (previewWidth == 0 || previewHeight == 0) {
    return;
  }
```

```
    if (rgbBytes == null) {
      rgbBytes = new
int[previewWidth *
previewHeight];
    }
    try {
      final Image image =
reader.acquireLatestImag
e();

      if (image == null) {
        return;
      }

      if
(isProcessingFrame) {
        image.close();
        return;
      }
      isProcessingFrame =
true;

Trace.beginSection("ima
geAvailable");
      final Plane[] planes =
image.getPlanes();
      fillBytes(planes,
yuvBytes);
      yRowStride =
planes[0].getRowStride()
;
      final int uvRowStride
=
planes[1].getRowStride()
;
      final int
uvPixelStride =
planes[1].getPixelStride()
;

      imageConverter =
          new Runnable() {
            @Override
            public void run()
{

ImageUtils.convertYUV
420ToARGB8888(
                yuvBytes[0],
                yuvBytes[1],
                yuvBytes[2],

previewWidth,

previewHeight,
                yRowStride,
                uvRowStride,
                uvPixelStride,
                rgbBytes);
            }
          };


postInferenceCallback =
          new Runnable() {
            @Override
            public void run()
{
              image.close();

isProcessingFrame =
false;
            }
          };

      processImage();
    } catch (final
Exception e) {
```

```
      LOGGER.e(e,
"Exception!");
      Trace.endSection();
      return;
    }
    Trace.endSection();
  }

  @Override
  public synchronized
void onStart() {
    LOGGER.d("onStart "
+ this);
    super.onStart();
  }

  @Override
  public synchronized
void onResume() {

LOGGER.d("onResume
" + this);
    super.onResume();

    handlerThread = new
HandlerThread("inferenc
e");
    handlerThread.start();
    handler = new
Handler(handlerThread.g
etLooper());
  }

  @Override
  public synchronized
void onPause() {
    LOGGER.d("onPause
" + this);

handlerThread.quitSafely
();
    try {
      handlerThread.join();
      handlerThread = null;
      handler = null;
    } catch (final
InterruptedException e) {
      LOGGER.e(e,
"Exception!");
    }

    super.onPause();
  }

  @Override
  public synchronized
void onStop() {
    LOGGER.d("onStop "
+ this);
    super.onStop();
  }

  @Override
  public synchronized
void onDestroy() {

LOGGER.d("onDestroy
" + this);
    super.onDestroy();
  }

  protected synchronized
void
runInBackground(final
Runnable r) {
    if (handler != null) {
      handler.post(r);
```

```
    }
  }

  @Override
  public void
onRequestPermissionsRe
sult(
      final int requestCode,
final String[]
permissions, final int[]
grantResults) {

super.onRequestPermissi
onsResult(requestCode,
permissions,
grantResults);
    if (requestCode ==
PERMISSIONS_REQU
EST) {
      if
(allPermissionsGranted(g
rantResults)) {
        setFragment();
      } else {
        requestPermission();
      }
    }
  }

  private static boolean
allPermissionsGranted(fi
nal int[] grantResults) {
    for (int result :
grantResults) {
      if (result !=
PackageManager.PERMI
SSION_GRANTED) {
        return false;
      }
    }
    return true;
  }

  private boolean
hasPermission() {
    if
(Build.VERSION.SDK_I
NT >=
Build.VERSION_CODE
S.M) {
      return
checkSelfPermission(PE
RMISSION_CAMERA)
==
PackageManager.PERMI
SSION_GRANTED;
    } else {
      return true;
    }
  }

  private void
requestPermission() {
    if
(Build.VERSION.SDK_I
NT >=
Build.VERSION_CODE
S.M) {
      if
(shouldShowRequestPer
missionRationale(PERM
ISSION_CAMERA)) {
        Toast.makeText(

CameraActivity.this,
            "Camera
permission is required for
this demo",
```

```
Toast.LENGTH_LONG)
                .show();
        }

requestPermissions(new
String[]
{PERMISSION_CAME
RA},
PERMISSIONS_REQU
EST);
    }
  }

  // Returns true if the
device supports the
required hardware level,
or better.
  private boolean
isHardwareLevelSupport
ed(

CameraCharacteristics
characteristics, int
requiredLevel) {
    int deviceLevel =
characteristics.get(Camer
aCharacteristics.INFO_S
UPPORTED_HARDWA
RE_LEVEL);
    if (deviceLevel ==
CameraCharacteristics.I
NFO_SUPPORTED_HA
RDWARE_LEVEL_LE
GACY) {
      return requiredLevel
== deviceLevel;
    }
    // deviceLevel is not
LEGACY, can use
numerical sort
    return requiredLevel
<= deviceLevel;
  }

  private String
chooseCamera() {
    final CameraManager
manager =
(CameraManager)
getSystemService(Conte
xt.CAMERA_SERVICE)
;
    try {
      for (final String
cameraId :
manager.getCameraIdLis
t()) {
        final
CameraCharacteristics
characteristics =
manager.getCameraChar
acteristics(cameraId);

        // We don't use a
front facing camera in
this sample.
        final Integer facing
=
characteristics.get(Camer
aCharacteristics.LENS_F
ACING);
        if (facing != null
&& facing ==
CameraCharacteristics.L
ENS_FACING_FRONT)
{
```

```
        continue;
      }

      final
StreamConfigurationMap
map =

characteristics.get(Camer
aCharacteristics.SCALE
R_STREAM_CONFIGU
RATION_MAP);

      if (map == null) {
        continue;
      }

      // Fallback to
camera1 API for internal
cameras that don't have
full support.
      // This should help
with legacy situations
where using the camera2
API causes
      // distorted or
otherwise broken
previews.
      useCamera2API =
          (facing ==
CameraCharacteristics.L
ENS_FACING_EXTER
NAL)
              ||
isHardwareLevelSupport
ed(

characteristics,
CameraCharacteristics.I
NFO_SUPPORTED_HA
RDWARE_LEVEL_FU
LL);

LOGGER.i("Camera API
lv2?:
useCamera2API);
        return cameraId;
      }
    } catch
(CameraAccessExceptio
n e) {
      LOGGER.e(e, "Not
allowed to access
camera");
    }

    return null;
  }

  protected void
setFragment() {
    String cameraId =
chooseCamera();

    Fragment fragment;
    if (useCamera2API) {

CameraConnectionFrag
ment camera2Fragment =

CameraConnectionFrag
ment.newInstance(
            new
CameraConnectionFrag
ment.ConnectionCallbac
k() {
              @Override
              public void
onPreviewSizeChosen(fi
```

%s",

```
nal Size size, final int
rotation) {

previewHeight =
size.getHeight();
            previewWidth
= size.getWidth();

CameraActivity.this.onPr
eviewSizeChosen(size,
rotation);
             }
            },
            this,
            getLayoutId(),

getDesiredPreviewFrame
Size());


camera2Fragment.setCa
mera(cameraId);
    fragment =
camera2Fragment;
   } else {
    fragment =
       new
LegacyCameraConnectio
nFragment(this,
getLayoutId(),
getDesiredPreviewFrame
Size());
   }


getFragmentManager().b
eginTransaction().replace
(R.id.container,
fragment).commit();
  }

  protected void
fillBytes(final Plane[]
planes, final byte[][]
yuvBytes) {
   // Because of the
variable row stride it's
not possible to know in
   // advance the actual
necessary dimensions of
the yuv planes.
   for (int i = 0; i <
planes.length; ++i) {
    final ByteBuffer
buffer =
planes[i].getBuffer();
    if (yuvBytes[i] ==
null) {

LOGGER.d("Initializing
buffer %d at size %d", i,
buffer.capacity());
      yuvBytes[i] = new
byte[buffer.capacity()];
    }

buffer.get(yuvBytes[i]);
   }
  }

  protected void
readyForNextImage() {
   if
(postInferenceCallback
!= null) {

postInferenceCallback.ru
n();
```

```
    }
  }

  protected int
getScreenOrientation() {
    switch
(getWindowManager().g
etDefaultDisplay().getRo
tation()) {
      case
Surface.ROTATION_27
0:
        return 270;
      case
Surface.ROTATION_18
0:
        return 180;
      case
Surface.ROTATION_90:
        return 90;
      default:
        return 0;
    }
  }

  @UiThread
  protected void
showResultsInBottomSh
eet(List<Recognition>
results) {
    if (results != null &&
results.size() >= 1) {
      Recognition
recognition =
results.get(0);
      if (recognition !=
null) {
        float confi = 100*
recognition.getConfidenc
e();
        Log.d("persen",
String.valueOf(confi));
        if
(recognition.getTitle() !=
null){

recognitionTextView.set
Text(recognition.getTitle
());

          if
(recognition.getTitle().eq
uals("space")){

recognitionTextView.set
Text("Boşluk
Bırakıldı.");

composed_text.setText(c
omposed_text.getText()
+ " ");
          }
          else if
(recognition.getTitle().eq
uals("del")){

recognitionTextView.set
Text("Harf Silindi.");

button_delete.callOnClic
k();
          }
          else if
(recognition.getTitle().eq
uals("nothing")){
```

```
recognitionTextView.setText("Tidak ada Isyarat terdeteksi");
          // do nothing
         /*} else {

          if(confi>80){
//composed_text.setText(composed_text.getText() + recognition.getTitle());

composed_text.setText(recognition.getTitle());
          }*/
         }
         else {
          if (recognition.getTitle() == active_latter) {

active_latter_number += 1;
           if (active_latter_number >= 7) {

composed_text.setText(composed_text.getText() + active_latter);

active_latter_number = 1;
            } else {
              //do nothing
            }
           } else {
                active_latter = recognition.getTitle();

active_latter_number = 1;
           }
          }
         }
         if (recognition.getConfidence() != null)
//recognitionValueTextView.setText(String.format("%.2f", (10000 * recognition.getConfidence())) + "%");

recognitionValueTextView.setText(String.format("%.2f", (100 * recognition.getConfidence())) + "%");
      }
    }
  }

  protected void showInference(String inferenceTime) {

inferenceTimeTextView.setText(inferenceTime);
  }

  protected Model getModel() {
    return model;
  }
```

```
  protected Device
getDevice() {
    return device;
  }

  private void
setDevice(Device device)
{
    if (this.device !=
device) {

LOGGER.d("Updating
device: " + device);
      this.device = device;
      final boolean
threadsEnabled = device
== Device.CPU;

plusImageView.setEnabl
ed(threadsEnabled);

minusImageView.setEna
bled(threadsEnabled);

threadsTextView.setText
(threadsEnabled ?
String.valueOf(numThre
ads) : "N/A");

onInferenceConfiguratio
nChanged();
    }
  }

  protected int
getNumThreads() {
    return numThreads;
  }

  private void
setNumThreads(int
numThreads) {
    if (this.numThreads !=
numThreads) {

LOGGER.d("Updating
numThreads: " +
numThreads);
      this.numThreads =
numThreads;

onInferenceConfiguratio
nChanged();
    }
  }

  protected abstract void
processImage();

  protected abstract void
onPreviewSizeChosen(fi
nal Size size, final int
rotation);

  protected abstract int
getLayoutId();

  protected abstract Size
getDesiredPreviewFrame
Size();

  protected abstract void
onInferenceConfiguratio
nChanged();

  @Override
```

```
  public void onClick(View v) {
    if (v.getId() == R.id.plus) {
      String threads = threadsTextView.getText().toString().trim();
      int numThreads = Integer.parseInt(threads);
      if (numThreads >= 9) return;

      setNumThreads(++numThreads);

      threadsTextView.setText(String.valueOf(numThreads));
    } else if (v.getId() == R.id.minus) {
      String threads = threadsTextView.getText().toString().trim();
      int numThreads = Integer.parseInt(threads);
      if (numThreads == 1) {
        return;
      }
      setNumThreads(--numThreads);

      threadsTextView.setText(String.valueOf(numThreads));
    }
  }

  @Override
  public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {
    if (parent == deviceSpinner) {

      setDevice(Device.valueOf(parent.getItemAtPosition(pos).toString()));
    }
  }

  @Override
  public void onNothingSelected(AdapterView<?> parent) {
    // Do nothing.
  }
}
```

Lampiran -4 *Listing* program halaman activity_terjemahan_isyarat .xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.core.widget.NestedScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout_width="match_parent"

android:layout_height="match_parent"

tools:context=".ui.TerjemahanIsyarat">
   <LinearLayout

android:id="@+id/linaerlayoutTerjemah"

android:layout_width="match_parent"

android:layout_height="wrap_content"
       android:orientation="vertical"
       android:paddingBottom="8dp"
       >


<androidx.appcompat.widget.AppCompatEditText
           android:id="@+id/edtHuruf"

android:layout_marginTop="16dp"
           android:padding="42dp"

android:layout_width="250dp"

android:layout_height="wrap_content"

android:layout_gravity="center"

android:background="@drawable/bg_terjemahan"
           />


<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btnTerjemah"

android:layout_marginTop="16dp"

android:layout_width="wrap_content"

android:layout_height="wrap_content"
           android:text="Terjemahkan"

android:layout_gravity="center"
           android:padding="8dp"

android:background="@drawable/bg_btn_detection"
           />


<androidx.appcompat.widget.AppCompatButton
```

```
        android:id="@+id/btnDelete"
android:layout_marginTop="16dp"
android:layout_width="wrap_content
"
android:layout_height="wrap_conten
t"
        android:text="Hapus"
android:layout_gravity="center"
        android:padding="8dp"
android:background="@drawable/bg
_btn_detection"
        />
<androidx.appcompat.widget.AppCo
mpatTextView
android:layout_width="wrap_content
"
android:layout_height="wrap_conten
t"
        android:text="Terjemahan"
        android:textSize="16sp"
android:layout_marginTop="16dp"
        android:textColor="#000000"
android:layout_marginStart="8dp"
        />
        <View
android:layout_width="400dp"
            android:layout_height="2dp"
android:background="#221E1E"
android:layout_gravity="center"
android:layout_marginTop="8dp"
            />
    <LinearLayout
android:id="@+id/linaerlayoutImage
"
android:layout_width="match_parent
"
android:layout_height="wrap_conten
t"
        android:orientation="vertical"
        android:paddingBottom="8dp"
        >
    </LinearLayout>
    </LinearLayout>
</androidx.core.widget.NestedScroll
View>
```

Lampiran -5 *Listing* program halaman TerjemahIsyrat.kt

```
package org.tensorflow.lite.examples.classification.ui

import android.os.Bundle
import android.util.Log
import android.util.TypedValue
import android.view.Gravity
import android.view.View
import android.view.ViewGroup
import android.widget.LinearLayout
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_terjemahan_isyarat.*
import org.tensorflow.lite.examples.classification.R
import pl.droidsonroids.gif.GifImageView


class TerjemahanIsyarat : AppCompatActivity() {

    private var ivGif: GifImageView? = null

    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_terjemahan_isyarat)


        btnTerjemah.setOnClickListener {

            val ambilHuruf = edtHuruf.text.toString()
            val chars: CharArray = toCharacterArray(ambilHuruf)

            println(chars.contentToString())

            val charArray = ambilHuruf.toCharArray()
            for (i in 0 until edtHuruf.length()) {

                tampilGambr(charArray[i].toString())
```

```
Toast.makeText(this,
"${charArray[i]}",
Toast.LENGTH_SHORT
).show()
            Log.d("kata",
"${charArray[i]}")
          }

      }


btnDelete.setOnClickList
ener {
          hapusGambar()
          hapusGambar()
      }
   }

   private              fun
toCharacterArray(str:
String): CharArray {
      return
str.toCharArray()
   }

   private              fun
tampilGambr(str:
String){
      ivGif              =
pl.droidsonroids.gif.GifI
mageView(applicationCo
ntext)

      when(str){
         "a"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_a)
         "b"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_b)
         "c"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_c)
         "d"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_d)
         "e"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_e)
         "f"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_f)
         "g"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_g)
         "h"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_h)
         "i"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_i)
         "j"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_j)
         "k"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_k)
         "l"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_l)
         "m"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_m)
         "n"             ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_n)
```

```
            "o" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_o)
            "p" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_p)
            "q" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_q)
            "r" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_r)
            "s" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_s)
            "t" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_t)
            "u" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_u)
            "v" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_v)
            "w" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_w)
            "x" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_x)
            "y" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_y)
            "z" ->
ivGif!!.setImageResourc
e(R.drawable.isyarat_z)


        }
```

```
        ivGif!!.id =
View.generateViewId()
        println("muas =>
${ivGif!!.id}")

        // Create layout
parameters for
ImageView
        // Create layout
parameters for
ImageView
        val lp:
LinearLayout.LayoutPara
ms =
LinearLayout.LayoutPara
ms(

LinearLayout.LayoutPara
ms.WRAP_CONTENT,

LinearLayout.LayoutPara
ms.WRAP_CONTENT
        )
        val sizeInDP = 16

        val marginInDp =
TypedValue.applyDimen
sion(

TypedValue.COMPLEX
_UNIT_DIP,
sizeInDP.toFloat(),
resources

.displayMetrics).toInt()

        lp.gravity =
Gravity.CENTER
```

```
        lp.bottomMargin = marginInDp

        ivGif!!.layoutParams = lp;
        // Finally, add the ImageView to layout
        // linaerlayoutTerjemah.removeView(iv)

        linaerlayoutImage.addView(ivGif)
    }
        Log.d("ambil gambar", "sukses")

    }

    private fun hapusGambar(){

        linaerlayoutImage.removeAllViews()
    }
}
```

Lampiran -6 *Training Dataset*.xml

```
!pip install -q tflite-model-maker

import os

import numpy as np

import tensorflow as tf
assert tf.__version__.startswith('2')

from tflite_model_maker import
model_spec
from tflite_model_maker import
image_classifier
from tflite_model_maker.config
import ExportFormat
from tflite_model_maker.config
import QuantizationConfig
from
tflite_model_maker.image_classifier
import DataLoader

import matplotlib.pyplot as plt

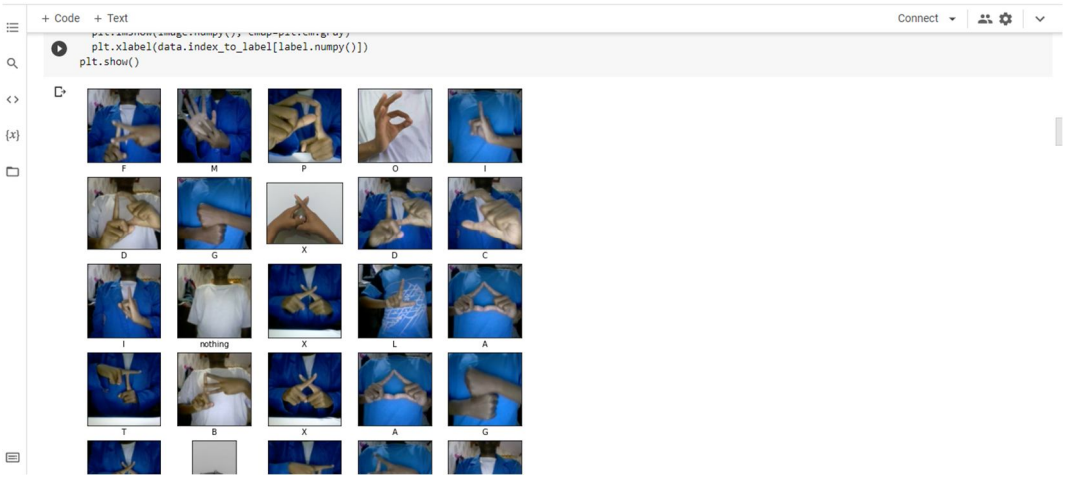
from google.colab import drive
drive.mount('/content/drive')
#base dir
%cd /content/drive/MyDrive/bisindo

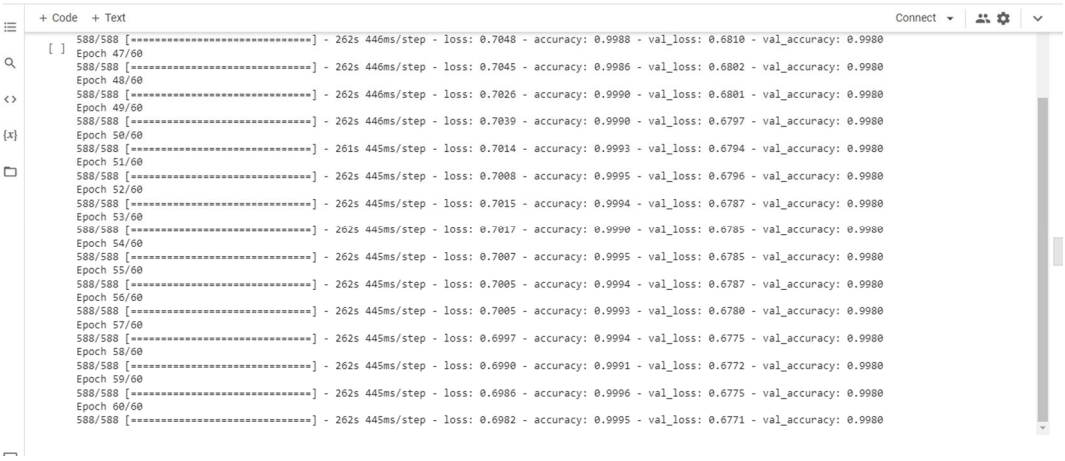
image_path = 'train'
image_path =
os.path.join(os.path.dirname(image_
path), 'train')
data =
DataLoader.from_folder(image_path
)
train_data, rest_data = data.split(0.9)
validation_data, test_data =
rest_data.split(0.5)

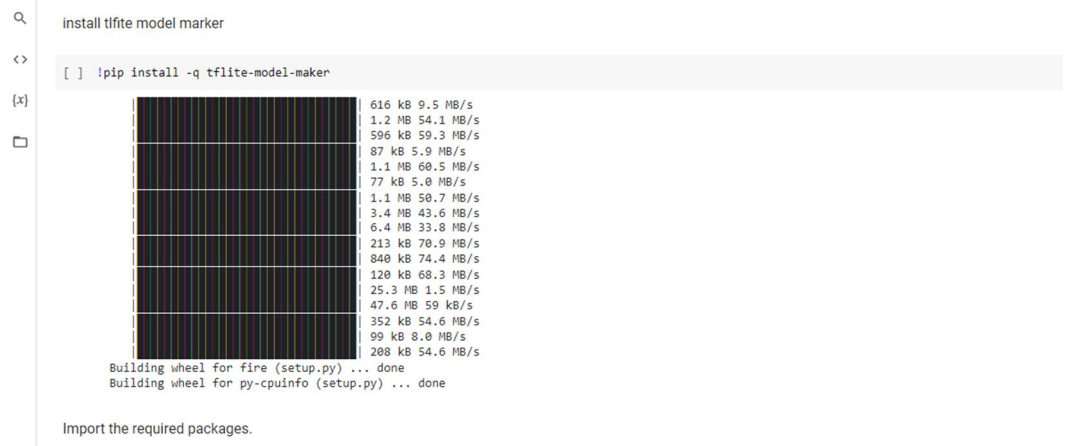
plt.figure(figsize=(10,10))
for i, (image, label) in
enumerate(data.gen_dataset().unbatc
h().take(25)):
  plt.subplot(5,5,i+1)
  plt.xticks([])
  plt.yticks([])
  plt.grid(False)
  plt.imshow(image.numpy(),
cmap=plt.cm.gray)

plt.xlabel(data.index_to_label[label.n
umpy()])
plt.show()

#Model New 23/12/2021
model =
image_classifier.create(train_data,
validation_data=validation_data,epoc
hs=70,learning_rate=0.001,batch_siz
e=32,model_spec=model_spec.get('
mobilenet_v2'))
```


+ Code
+ Text
Connect
plt.xlabel(data.index_to_label[label.numpy()])
plt.show()
F
M
P
O
I
D
G
X
D
C
I
nothing
X
L
A
T
B
X
A
G


+ Code
+ Text
Connect
588/588 [==========] - 262s 446ms/step - loss: 0.7048 - accuracy: 0.9988 - val_loss: 0.6810 - val_accuracy: 0.9980
Epoch 47/60
588/588 [==========] - 262s 446ms/step - loss: 0.7045 - accuracy: 0.9986 - val_loss: 0.6802 - val_accuracy: 0.9980
Epoch 48/60
588/588 [==========] - 262s 446ms/step - loss: 0.7026 - accuracy: 0.9990 - val_loss: 0.6801 - val_accuracy: 0.9980
Epoch 49/60
588/588 [==========] - 262s 446ms/step - loss: 0.7039 - accuracy: 0.9990 - val_loss: 0.6797 - val_accuracy: 0.9980
Epoch 50/60
588/588 [==========] - 261s 445ms/step - loss: 0.7014 - accuracy: 0.9993 - val_loss: 0.6794 - val_accuracy: 0.9980
Epoch 51/60
588/588 [==========] - 262s 445ms/step - loss: 0.7008 - accuracy: 0.9995 - val_loss: 0.6796 - val_accuracy: 0.9980
Epoch 52/60
588/588 [==========] - 262s 445ms/step - loss: 0.7015 - accuracy: 0.9994 - val_loss: 0.6787 - val_accuracy: 0.9980
Epoch 53/60
588/588 [==========] - 262s 445ms/step - loss: 0.7017 - accuracy: 0.9990 - val_loss: 0.6785 - val_accuracy: 0.9980
Epoch 54/60
588/588 [==========] - 262s 445ms/step - loss: 0.7007 - accuracy: 0.9995 - val_loss: 0.6785 - val_accuracy: 0.9980
Epoch 55/60
588/588 [==========] - 262s 445ms/step - loss: 0.7005 - accuracy: 0.9994 - val_loss: 0.6787 - val_accuracy: 0.9980
Epoch 56/60
588/588 [==========] - 262s 445ms/step - loss: 0.7005 - accuracy: 0.9993 - val_loss: 0.6780 - val_accuracy: 0.9980
Epoch 57/60
588/588 [==========] - 262s 445ms/step - loss: 0.6997 - accuracy: 0.9994 - val_loss: 0.6775 - val_accuracy: 0.9980
Epoch 58/60
588/588 [==========] - 262s 445ms/step - loss: 0.6990 - accuracy: 0.9991 - val_loss: 0.6772 - val_accuracy: 0.9980
Epoch 59/60
588/588 [==========] - 262s 445ms/step - loss: 0.6986 - accuracy: 0.9996 - val_loss: 0.6775 - val_accuracy: 0.9980
Epoch 60/60
588/588 [==========] - 262s 445ms/step - loss: 0.6982 - accuracy: 0.9995 - val_loss: 0.6771 - val_accuracy: 0.9980


install tlfite model marker
!pip install -q tflite-model-maker
616 kB 9.5 MB/s
1.2 MB 54.1 MB/s
596 kB 59.3 MB/s
87 kB 5.9 MB/s
1.1 MB 60.5 MB/s
77 kB 5.0 MB/s
1.1 MB 50.7 MB/s
3.4 MB 43.6 MB/s
6.4 MB 33.8 MB/s
213 kB 70.9 MB/s
840 kB 74.4 MB/s
120 kB 68.3 MB/s
25.3 MB 1.5 MB/s
47.6 MB 59 kB/s
352 kB 54.6 MB/s
99 kB 8.0 MB/s
208 kB 54.6 MB/s
Building wheel for fire (setup.py) ... done
Building wheel for py-cpuinfo (setup.py) ... done
Import the required packages.