

MANUAL BOOK PROGRAM PEMBELAJARAN
“BENTUK BENDA”
(BANGUN DATAR) Versi 1.0



Oleh:

Teja Endra Eng Tju, S.T., M.Kom.¹

Elizabeth Nurmiyati Tamatjita, S.Kom, MM, M.CS., M.H.²

PROGRAM STUDI SISTEM INFORMASI¹
FAKULTAS TEKNOLOGI INFORMASI¹
UNIVERSITAS BUDI LUHUR¹

TEKNIK INFORMATIKA²
PTK – SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER
(STMIK) WIDURI²

DAFTAR ISI

I. INFORMASI UMUM APLIKASI	3
II. PANDUAN PENGGUNAAN	8
III. <i>SOURCE CODE</i> PROGRAM PEMBELAJARAN “BENTUK BENDA” (BANGUN DATAR)	12

I. INFORMASI UMUM APLIKASI

“Bentuk Benda” (Bangun Datar)

“Bentuk Benda” (Bangun Datar) adalah program pembelajaran berbasis Android yang bertujuan sebagai sarana pengenalan bentuk bangun datar dua dimensi untuk anak-anak di jenjang Taman Kanak-kanak.

Program pembelajaran pengenalan bangun datar ini bertujuan untuk meningkatkan kecerdasan, kreativitas, dan imajinasi pada anak dan dapat dilakukan dengan beberapa cara salah satunya dengan pengenalan program permainan “Bentuk Benda.” Adanya teknologi yang sudah sangat berkembang, sarana untuk pembelajaran menjadi semakin banyak dan beragam seperti menggunakan komputer, *laptop* maupun telepon genggam (*smartphone*). Teknologi permainan sebagai sarana belajar yang sudah sangat berkembang dapat mempermudah anak-anak dalam mengenal perbedaan antara bentuk bangun segitiga, segi empat, segi lima dan lingkaran.

Cara memainkan permainan pengenalan bangun datar ini adalah dengan mengarahkan kamera ke sebuah atau beberapa bidang datar, lalu aplikasi akan mendeteksi bidang datar tersebut. Agar memunculkan objek 2 (dua) dimensi, pemain harus memencet tombol KAMERA agar bidang datar yang sudah dideteksi dan objek bangun datar tersebut akan muncul pada bidang datar tersebut. Permainan dimulai dan pemain dapat memfoto objek-objek yang tersebar di sekitar latar belakang terang untuk mendapatkan bentuk sesuai dengan namanya yaitu setelah tombol KAMERA dipilih, maka jepretlah salah satu objek atau beberapa yang dituju. Kemudian setelah difoto, pilihlah tombol centang (✓), agar muncul jawabanya dari program permainan tersebut dengan nama objek bangun datar yang dikenali oleh program permainan “Bentuk Benda” (Bangun Datar) ini, seperti terlihat pada Gambar 1, Gambar 2 dan Gambar 3.



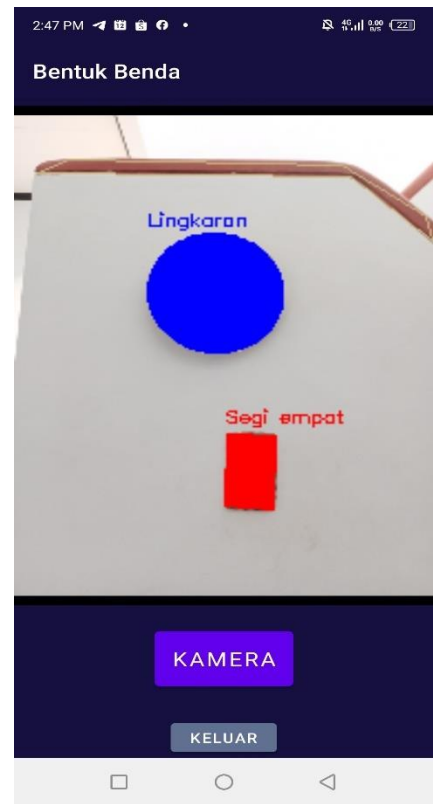
Gambar 1. Saat Program

dijalankan.



Gambar 2. Saat pilih tombol

KAMERA, kemudian cari objek-objek yang dituju. Setelah itu tekan tombol



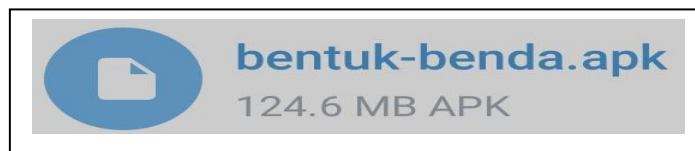
Gambar 3. Saat

tombol  di pilih, maka akan tampil objek-

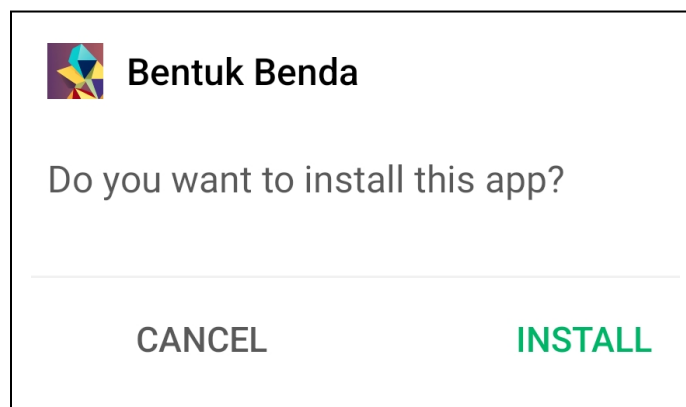
objek bangun datar, sesuai namanya (program dapat mengenali 2 (dua) buah Bangun Datar sesuai namanya.

Permainan di atas dikembangkan pada platform Java dan Xml serta berjalan pada Sistem Operasi Android dengan nama **bentuk-**

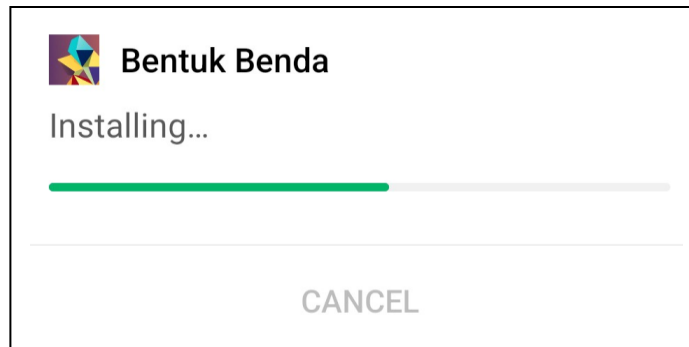
benda.apk. Gambar 3 menunjukkan bahwa program aplikasi pengenalan bangun datar dengan nama “Bentuk Benda” dapat mengenali objek jam sebagai lingkaran (objek lingkaran berwarna biru dan tulisan lingkaran berwarna biru) dan objek hiasan dinding di bawah objek jam sebagai bangun datar segi empat (objek hiasan berwarna merah dan tulisan segi empat berwarna merah). Jika objek benda lainnya terdeteksi seperti segitiga maka objek segitiga akan berwarna hijau beserta tulisan segitiga berwarna hijau juga dan objek segi lima jika terdeteksi akan berwarna kuning beserta dengan tulisan segi lima berwarna kuning yang sama. Saat file program bentuk-benda.apk dilakukan instalasi maka akan terlihat seperti Gambar 4, Gambar 5, Gambar 6, Gambar 7 dan Gambar 8.



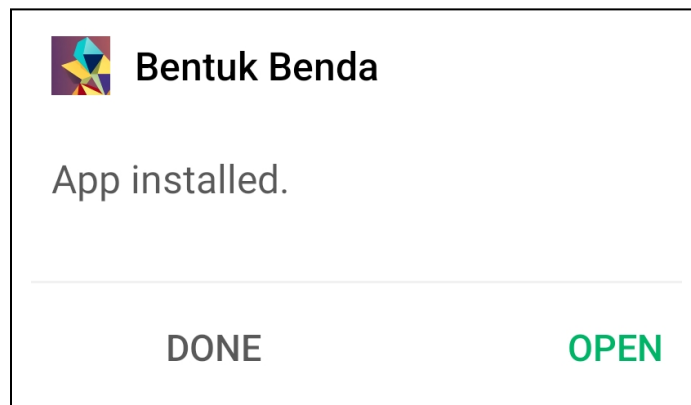
Gambar 4. File bentuk-benda.apk



Gambar 5. Pilih tombol *INSTALL* file Bentuk-benda.apk



Gambar 6. *Installing* file bentuk-benda.apk



Gambar 7. Aplikasi Bentuk Benda sudah *diinstall* pada *smartphone* Android



Gambar 8. Setelah tombol *OPEN* dipilih, maka Aplikasi “Bentuk Benda” dapat digunakan

II. PANDUAN PENGGUNAAN

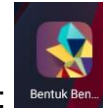
Panduan Pengguna: Aplikasi (Android) Pengenalan Bangun Datar untuk Siswa Taman Kanak-kanak

Panduan ini akan membantu penggunaan aplikasi Pengenalan Bangun Datar. Pastikan *file installer* **bentuk-benda.apk** telah disiapkan sebelum memulai.

Langkah 1: Instal Aplikasi

1. Buka (klik dua kali) *file* **bentuk-benda.apk** pada perangkat Android.
2. Tunggu hingga proses instalasi selesai.

3. Hasil instalasi berupa ikon **Bentuk Benda**:



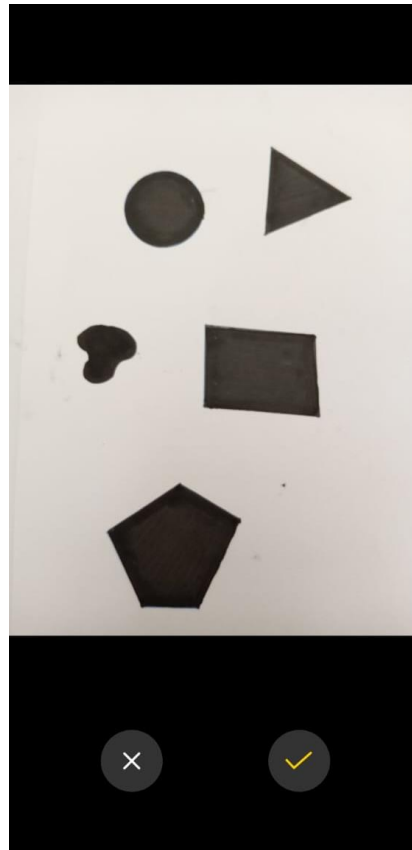
Langkah 2: Buka Aplikasi

1. Setelah instalasi selesai, buka (klik dua kali) ikon aplikasi di layar beranda perangkat Android.
2. Aplikasi akan terbuka dengan tampilan:



Langkah 3: Tombol **KAMERA** atau **KELUAR**

1. Tekan tombol **KAMERA** untuk menangkap objek atau benda dengan kamera perangkat Android, atau Tekan **KELUAR** untuk menutup aplikasi.
2. Setelah tombol **KAMERA** ditekan akan terbuka kamera dan siap menangkap objek atau benda dengan hasil tangkapan seperti berikut:



3. Selanjutnya tekan tanda silang (x) untuk membatalkan hasil tangkapan kamera atau tekan tanda check (✓) untuk melakukan pengujian dengan hasil seperti berikut ini:

Bentuk Benda

Lingkaran



Segi tiga



Segi empat



Segi lima



KAMERA

KELUAR

III. *SOURCE CODE* “BENTUK BENDA” (BANGUN DATAR)

Kode Program:

Pemrograman aplikasi Android menggunakan Android Studio terdiri dari 2 (dua) bagian utama yaitu tampilan layar (didefinisikan melalui file **activity_main.xml**) dan program aplikasinya (**MainActivity.java**).

File **activity_main.xml** didefinisikan bahwa tampilan aplikasi akan memiliki 3 (tiga) buah komponen yaitu **ImageView** sebagai tempat meletakkan citra foto setelah diidentifikasi bentuk bangunnya, serta 2 (dua) buah tombol yaitu “KAMERA” dan “KELUAR” masing-masing untuk mengambil gambar dari kamera telepon genggam (*smartphone*) dan untuk keluar dari aplikasi.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="400dp"
        android:layout_height="500dp"
        android:id="@+id/imageView"
        android:background="#000"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="kamera"
        android:id="@+id/imageView"
        android:layout_below="@id/select"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        />

    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/exitbut"
        android:text="keluar"
        android:layout_below="@id/camera"
```

```
        android:layout_centerHorizontal="true"  
        android:layout_marginTop="20dp"  
    />  
  
</RelativeLayout>
```

MainActivity.java

```
package com.example.camerashape;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

import org.opencv.android.OpenCVLoader;
import org.opencv.android.Utls;
import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.core.MatOfPoint;
import org.opencv.core.MatOfPoint2f;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.imgproc.Imgproc;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import android.Manifest;

public class MainActivity extends AppCompatActivity {

    Button camera, exitbut;
    ImageView imageView;
    Bitmap bitmap;
    Mat mat;
    int SELECT_CODE=100, CAMERA_CODE=101;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if(OpenCVLoader.initDebug()) Log.d("LOADED", "success");
        else Log.d("LOADED", "error");
    }
}
```

```

        getPermission();

        camera = findViewById(R.id.camera);
        exitbut = findViewById(R.id.exitbut);
        imageView = findViewById(R.id.imageView);

        camera.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(intent, CAMERA_CODE);
            }
        });

        exitbut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
                System.exit(0);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode==CAMERA_CODE){
            bitmap = (Bitmap) data.getExtras().get("data");
            imageView.setImageBitmap(bitmap);

            mat = new Mat();
            Utils.bitmapToMat(bitmap, mat);

            mat = filter(mat);

            Utils.matToBitmap(mat, bitmap);
            imageView.setImageBitmap(bitmap);
        }
    }

    void getPermission(){
        if(checkSelfPermission(Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED){
            requestPermissions(new String[]{Manifest.permission.CAMERA},
102);
        }
    }
}

```

```

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if(requestCode==102 && grantResults.length>0){
        if(grantResults[0]==PackageManager.PERMISSION_GRANTED){
            getPermission();
        }
    }
}

public Mat filter(final Mat src) {

    final Mat dst = new Mat(src.rows(), src.cols(), src.type());
    src.copyTo(dst);

    int height = dst.rows();
    int width = dst.cols();
    int imgarea = height * width;
    float aspect_ratio, w, h;
    double area, min_area, epsilon;
    Rect boundRect;
    String shp;

    Imgproc.cvtColor(dst, dst, Imgproc.COLOR_BGR2GRAY);
    Imgproc.threshold(dst, dst, 128,255,Imgproc.THRESH_BINARY_INV);

    final List<MatOfPoint> points = new ArrayList<>();
    final Mat hierarchy = new Mat();
    Imgproc.findContours(dst, points, hierarchy, 1, 2);

    if (hierarchy.size().height > 0 && hierarchy.size().width > 0) {
        int idx = 0;
        for (MatOfPoint cnt : points) {
            //Imgproc.drawContours(src, points, idx, new Scalar(255,
0, 0), 3);

            MatOfPoint2f thisContour2f = new MatOfPoint2f();
            MatOfPoint approxContour = new MatOfPoint();
            MatOfPoint2f approxContour2f = new MatOfPoint2f();

            cnt.convertTo(thisContour2f, CvType.CV_32FC2);
            epsilon = 0.01*Imgproc.arcLength(thisContour2f, true);
            Imgproc.approxPolyDP(thisContour2f, approxContour2f,
epsilon,true);

            area = Imgproc.contourArea(thisContour2f);
            min_area = imgarea*0.007;
            boundRect = Imgproc.boundingRect(cnt);
            w = boundRect.width;

```



```

        h = boundRect.height;
        aspect_ratio = w/h;

        approxContour2f.convertTo(approxContour, CvType.CV_32S);
        if(area>min_area){
            if(approxContour.size().height>2){
                if(approxContour.size().height==5){
                    shp="Segi lima";
                    Imgproc.drawContours(src, points, idx, new
Scalar(230, 230, 120), -1);
                } else if (approxContour.size().height==3) {
                    shp="Segi tiga";
                    Imgproc.drawContours(src, points, idx, new
Scalar(0, 255, 0), -1);
                } else if (approxContour.size().height==4) {
                    shp="Segi empat";
                    Imgproc.drawContours(src, points, idx, new
Scalar(255, 0, 0), -1);
                } else if (approxContour.size().height>8&&
approxContour.size().height<84) {
                    Imgproc.drawContours(src, points, idx, new
Scalar(0, 0, 255), -1);
                }
                if(aspect_ratio>1.05||aspect_ratio<0.95){
                    shp="Elips";
                } else {
                    shp="Lingkaran";
                }
            } else {
                shp = "Lainnya";
            }
            Imgproc.putText(mat, shp, new
Point(boundRect.x,boundRect.y), Imgproc.FONT_HERSHEY_DUPLEX, 0.6, new
Scalar(0, 0, 0));
        }
        idx++;
    }
}

//Imgproc.cvtColor(dst, dst, Imgproc.COLOR_GRAY2BGR);

return src;
}
}

```