

DAFTAR PUSTAKA

- Parta Setiawan. 2019. pengertian Viskositas dan Vaktornya.
<https://www.gurupendidikan.co.id/viskositas/>
- Bayu Niko Dahlan. 2015. Mengenal dan Memulai Pemrograman java
<https://www.codepolitan.com/mengenal-dan-memulai-pemrograman-java-belajar-java>
- Derry Fajriawan. 2017. Pengenalan OpenCV
<http://derryfajirwan.blogspot.com/2017/10/pengenalan-opencv.html>
- Ivanj. 2018 Simpel Thresholding menggunakan OpenCV python
<https://www.ivanjul.com/simple-thresholding-menggunakan-opencv-python/>
- S Rahman. 2013. Lectrures Blog STT - Harapan Medan.
<http://sayutirahman.stth-medan.ac.id/2013/12/thresholding.html>.
- Anonim. 2016. Metode Algoritma. <http://www.metode-algoritma.com/2016/01/region-of-interest-roi.html> .
- Basuki, Lutfi Febriandita. 2016. *Impelementasi Metode Histograms of Oriented Gradients dengan Optimasi Algoritma Frei-Chen untuk Deteksi Citra Manusia.* Diakses di <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikomp-gdl-lutfifebri-35958> pada tanggal 7 Februari 2018.
- Jalled, Fares. 2016. *Object Detection Using Image Processing.* <https://arxiv.org/pdf/1611.07791.pdf>. Diakses pada 6 Februari 2018
- Le, Hoang Huy. 2015. *Dvision Intro.* https://www.academia.edu/27143145/Dvision_intro. Diakses pada 6 Februari 2018.
- Munir, Rinaldi, 2004, Pengolahan Citra Digital dengan pendekatan Algoritmik, Penerbit Informatika, Bandung.
- Basri. 2015. Metode Gaussian Mixture Models untuk Optimalisasi Penghitungan Kendaraan Dalam Sistem Transportasi Cerdas.

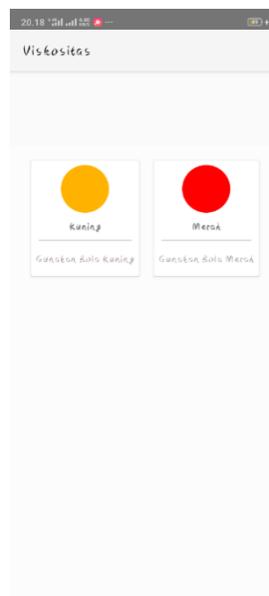
Zakurenov V.M, Konyakhin V.P., Nozdrev V.F. 1975. The Viscosities of Ethanol-Cyclohexane Mixtures. Zh.Fiz.Khim. 49. 548-549

.
.

LAMPIRAN

✓ **Petunjuk manual atau langkah-langkah menggunakan aplikasi untuk menguji viskositas dengan bola-bola.**

- Langkah pertama yaitu siapkan alat-alat seperti buret, statif, tripod dan fluida yang ingin di uji viskositasnya
- Pastikan buret yang di pakai bersih, tidak ada bekas cairan lain misalkan bekas minyak.
- Selanjutnya, buret di letakkan pada statif. Statif disini digunakan untuk memegang buret agar tidak goyang dan tidak jatuh.
- Langkah selanjutnya letakkan smartphone pada tripod, dan diletakkan di depan buret. Kemudian buka aplikasi lalu pilih warna bola yang ingin di gunakan dalam pengujian viskositas. Contoh :



- Selanjutnya stel jarak yang akan di pakai, misalkan kita menggunakan ukuran 30cm pada tabung buret, sesuaikan dengan

garis awal yang ada pada layar kamera pada aplikasi dengan titik 0, kemudian garis akhir di tempatkan pada titik 40cm. Contoh :



- Setelah titik awal dan titik akhir sudah pas, langkah selanjutnya yaitu masukkan fluida atau cairan, misalkan Alkohol, Aquades, dll.
- Setelah itu klik tombol play yang ada pada layar, tombol play tersebut berfungsi mendeteksi warna bola yang digunakan dalam percobaan.
- Selanjutnya masukkan bola-bola kedalam tabung buret yang telah diisi fluida, warna bola yang di pakai tidak boleh beda dari yang telah di set pada aplikasi, misalkan pada aplikasi telah di set warna kuning, berarti bola yang di masukkan ke dalam buret yaitu warna kuning juga.

- Setelah itu klik tombol stop pada layar, setelah tombol stop di klik maka akan muncul hasil waktu yang telah di tembus bola dari titik awal sampai titik akhir.
- Selanjutnya masukkan nilai-nilai untuk rumus viskositas seperti berat jenis fluida, berat jenis bola, kuadrat jari-jari bola, kecepatan gravitasi, dan jarak yang digunakan. Contoh :

Viskositas

Waktu Laju bola (detik):
1.4300

Berat jenis fluida (gr) 0.876

Berat jenis bola (gr) 1.1021

kuadrat Jarijari bola (cm) 0.05320

Jarak (cm) 20

HITUNG VISKOSITAS

Hasil Hitung Viskositas adalah :
Viskositas : 0.009964

- Setelah itu nilai-nilai telah dimasukkan selanjutnya klik tombol hitung viskositas, lalu akan muncul hasil hitung viskositas dari percobaan yang telah di gunakan.

Catatan : jarak maksimal yang dapat dipakai tergantung dengan besar kecilnya bola yang digunakan, di sini jarak maksimal yang didapatkan peneliti yaitu 40cm, karena bola-bola yang digunakan sangat kecil. Selain jarak pencahayaan juga berpengaruh terhadap

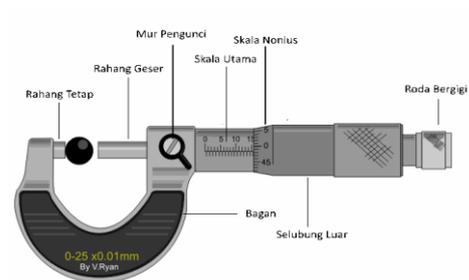
pendeteksian bola, jadi harus menggunakan cahaya yang terang jangan menggunakan cahaya yang redup.

✓ **Gambar alat-alat yang digunakan**

1. Tripod



2. Mikrometer



3. Bola-bola



4. Buret



5. Statif



6. Piknometer



7. Viscometer



✓ **Dokumentasi pengujian**





✓ Source code MainActivity

```
package com.example.defafm10opencvhough;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.SurfaceView;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.TextView;

import org.opencv.android.BaseLoaderCallback;
import org.opencv.android.CameraBridgeViewBase;
import org.opencv.android.LoaderCallbackInterface;
import org.opencv.android.OpenCVLoader;
import org.opencv.core.Core;
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.util.ArrayList;

public class MainActivity extends AppCompatActivity implements
CameraBridgeViewBase.CvCameraViewListener2 {
    private static final String TAG = "MainActivity";

    private CustomizableCameraView javaCameraView;
    private TextView tvAverage, tvFrameCount, touchDetection, tvTimer;
    private ArrayList<MatOfPoint> countours;
    private Float frameCount = 0f;
    private Float average = 0f;
    private Float BolayyaNgaseng = 0f;
    private boolean runIdentification = false, isRecord = false;
    private Button btnStart, btnStop, btnRecord;
    private int test = 0;
    int rectX1, rectX2;
    private String bolayya;

    double sec;

    long SecondInterval = 0;
```

```

        Mat mRgba;
        Mat mRgbaFiltered;

        long start, end;

        private BaseLoaderCallback mLoaderCallback = new
BaseLoaderCallback(this) {
            @Override
            public void onManagerConnected(int status) {
                switch (status) {
                    case LoaderCallbackInterface.SUCCESS:
                    {
                        Log.i(TAG, "OpenCV loaded successfully");
                        javaCameraView.setMaxFrameSize(1920, 1080);
                        javaCameraView.enableView();
                    } break;
                    default:
                    {
                        super.onManagerConnected(status);
                    } break;
                }
            }
        };

        public MainActivity() {
            Log.i(TAG, "Instantiated new " + this.getClass());
        }

        /** Called when the activity is first created. */
        @Override
        public void onCreate(Bundle savedInstanceState) {
            Log.i(TAG, "called onCreate");
            super.onCreate(savedInstanceState);

            tvTimer = (TextView) findViewById(R.id.tvTimer);

            javaCameraView = (CustomizableCameraView) findViewById(R.id.java_camera_view);
            javaCameraView.setMaxFrameSize(1080, 1920);
            javaCameraView.setVisibility(SurfaceView.VISIBLE);
            javaCameraView.setCvCameraViewListener(this);
            btnRecord.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    if(!isRecord) {
                        isRecord=true;
                        btnRecord.setBackgroundResource(R.drawable.ic_stop_black_24dp);
                        runIdentification = true;
                    }
                }
            });
        }
    }
}

```

```

        frameCount = 0f;
        average = 0f;
        BolayyaNgaseng = 0f;
    }
    else{
        runIdentification = false;
        isRecord=false;
        btnRecord.setBackgroundResource(R.drawable.ic_play);
        // btnRecord.setText("Start");
        Intent intent = new Intent(MainActivity.this, Add.class);
        intent.putExtra("waktuBola", tvTimer.getText());
        startActivity(intent);
        test=0;
        sec = 0;
        start = 0;
        end = 0;
        SecondInterval = 0;
    }
}

@Override
public void onPause()
{
    super.onPause();
    if (javaCameraView != null) ;
    javaCameraView.disableView();
}

@Override
public void onResume()
{
    super.onResume();
    if (!OpenCVLoader.initDebug()) {
        Log.d(TAG, "Internal OpenCV library not found. Using OpenCV Manager for
initialization");
        OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_3_0_0, this,
mLoaderCallback);
    } else {
        Log.d(TAG, "OpenCV library found inside package. Using it!");
        mLoaderCallback.onManagerConnected(LoaderCallbackInterface.SUCCESS);
    }
}

@Override
public void onPause()
{
    super.onPause();
    if (javaCameraView != null) ;
    javaCameraView.disableView();
}

```

```

}

@Override
public void onResume()
{
    super.onResume();
    if (!OpenCVLoader.initDebug()) {
        Log.d(TAG, "Internal OpenCV library not found. Using OpenCV Manager for
initialization");
        OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_3_0_0, this,
mLoaderCallback);
    } else {
        Log.d(TAG, "OpenCV library found inside package. Using it!");
        mLoaderCallback.onManagerConnected(LoaderCallbackInterface.SUCCESS);
    }
}

public void onDestroy() {
    super.onDestroy();
    if (javaCameraView != null) ;
    javaCameraView.disableView();
}

public void onCameraViewStarted(int width, int height) {

    mRgba = new Mat(height, width, CvType.CV_8UC1);
    mRgbaFiltered = new Mat(height, width, CvType.CV_8UC4);
}

public void onCameraViewStopped() {
    mRgba.release();
}

public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
    mRgba = inputFrame.rgba();
    Imgproc.cvtColor(mRgba, mRgbaFiltered, Imgproc.COLOR_RGB2HSV);
    //-----DETECTION LINE-----
    -----

    double
    Point pt1 = new Point(x1, y1);
    Point pt2 = new Point(x2, y2);

    Point pt3 = new Point(x3, y3);
    Point pt4 = new Point(x4, y4);

    Imgproc.line(mRgba, pt3, pt4, new Scalar(0, 0, 255), 3);
}

```

```

Imgproc.line(mRgba, pt1, pt2, new Scalar(0,0,255), 3);

if(runIdentification) {

    int sensitivity = 20;
    //-----END OF DETECTION LINE-----
    //----- VARIABEL WARNA KUNING -----
    Scalar lower = new Scalar ( 20 - sensitivity, 100, 100);
    Scalar upper = new Scalar( 30 + sensitivity, 255, 255);
    //-----END OF VARIABEL WARNA KUNING -----

    //----- VARIABEL WARNA MERAH -----
    Scalar lower1 = new Scalar( 190 - sensitivity, 70, 50);
    Scalar upper1 = new Scalar( 160 + sensitivity, 255, 255);
    //----- END OF VARIABEL WARNA MERAH -----

    if(bolayya.equals("Merah")) {
        Core.inRange(mRgbaFiltered, lower1, upper1, mRgbaFiltered);
    }
    else{
        Core.inRange(mRgbaFiltered, lower, upper, mRgbaFiltered);
    }

    ArrayList<MatOfPoint> contours = new ArrayList<MatOfPoint>();
    Mat hierarchy = new Mat();
    Imgproc.findContours(mRgbaFiltered, contours, hierarchy,
    Imgproc.RETR_TREE, Imgproc.CHAIN_APPROX_SIMPLE);
    for (int i = 0; i < contours.size(); i++) {

        if(contours.get(i).total() > 25) {

            MatOfPoint2f approxCurve = new MatOfPoint2f();
            MatOfPoint2f contour2f = new MatOfPoint2f( contours.get(i).toArray() );

            //Melakukan Proses Pada mMOP2f1 yang ada dalam variabel MatOfPoint2f
            double approxDistance = Imgproc.arcLength(contour2f, true)*0.02;
            Imgproc.approxPolyDP(contour2f, approxCurve, approxDistance, true);

            //Konversi Kembali Ke MatOfPoint
            MatOfPoint points = new MatOfPoint( approxCurve.toArray() );

            Rect rect = Imgproc.boundingRect(points);

            Imgproc.rectangle(mRgba, new Point(rect.x, rect.y), new Point(rect.x +
            rect.width, rect.y + rect.height), new Scalar (255, 0, 0, 255), 2);

```

```

        rectX1 = rect.x + rect.width;
        rectX2 = rect.x;

        BolayyaNgaseng ++;
    }
    Log.i(TAG, i+" = "+contours.get(i).total());
}
// ----- UNTUK MENGETAHUI TITIK X AWAL RECT OBJECT SAMPAI TITIK X AKHIR RECT
OBJECT BILA TERSENTUH Imgproc.Line-----

//          ----- CounterMeter -----

    if (rectX1 >= 100 ) {
        test=1;
    }
    if (rectX1 >= 150 ){
        test=0;
        rectX1 = 0;
    }

//          ----- End OF Counter Meter -----

frameCount++;
average = BolayyaNgaseng/frameCount;

MainActivity.this.runOnUiThread(new Runnable() {
    @Override
    public void run() {

        if(test==1){
            tvTimer.setVisibility(View.VISIBLE);
            start = System.currentTimeMillis();

        }

        else if(test==0){
            end = System.currentTimeMillis();
        }

        sec = (start - end) / 1000F;

        if(sec>0) {
            tvTimer.setText(String.format("%.4f", sec));
        }
        else{

        }
    }
}

```

```
    }  
    });  
  }  
  return mRgba;  
}  
}
```