

1. **Tujuan Aplikasi**

Tujuan utama keamanan kantor adalah untuk melindungi aset dan fasilitas organisasi Anda. Oleh karena itu tanggung jawab untuk melindungi karyawan sangat penting karena mereka merupakan aset penting bagi perusahaan. Keamanan mereka adalah prioritas Anda bersamaan dengan mengamankan fasilitas

1. **Landasan Teori**

Informasi yang merupakan aset harus dilindungi keamanannya. Keamanan secara umum diartikan sebagai „quality or state of being secure-to be free from danger„. Untuk menjadi aman adalah dengan cara dilindungi dari musuh dan bahaya. Contoh tinjauan keamanan informasi dari Whitman dan Mattord (2011) sebagai berikut.

a. Physical Security yang memfokuskan strategi untuk mengamankan pekerja atau anggota organisasi, aset fisik, dan tempat kerja dari berbagai ancaman meliputi bahaya kebakaran, akses tanpa otorisasi, dan bencana alam.

b. Personal Security yang overlap dengan „phisycal security’ dalam melindungi orang-orang dalam organisasi

c. Operation Security yang memfokuskan strategi untuk mengamankan kemampuan organisasi atau perusahaan untuk bekerja tanpa gangguan.

d. Communications Security yang bertujuan mengamankan media komunikasi, teknologi komunikasi dan isinya, serta kemampuan untuk memanfaatkan alat ini untuk mencapai tujuan organisasi.

e. Network Security yang memfokuskan pada pengamanan peralatan jaringan data organisasi, jaringannya dan isinya, serta kemampuan untuk menggunakan jaringan tersebut dalam memenuhi fungsi komunikasi data organisasi.

Masing-masing komponen diatas berkontribusi dalam program keamanan informasi secara keseluruhan. Keamanan informasi adalah perlindungan informasi, termasuk system dan perangkat yang digunakan, menyimpan, dan mengirimkannya. Keamanan informasi melindungi informasi dari berbagai ancaman untuk menjamin kelangsungan usaha, meminimalisasi kerusakan akibat terjadinya ancaman, mempercepat kembalinya investassi dan peluang usaha.

1. **Alat Dan Bahan**

Ada pun alat dan bahan yang di gunakan dalam pengembangan sistem ini adalah:

>.alat : 1.Satu perangkat laptop

2.Framework Berbasis Java(NEATBEANS)

>.Bahan : 1.Analisa Lapangan

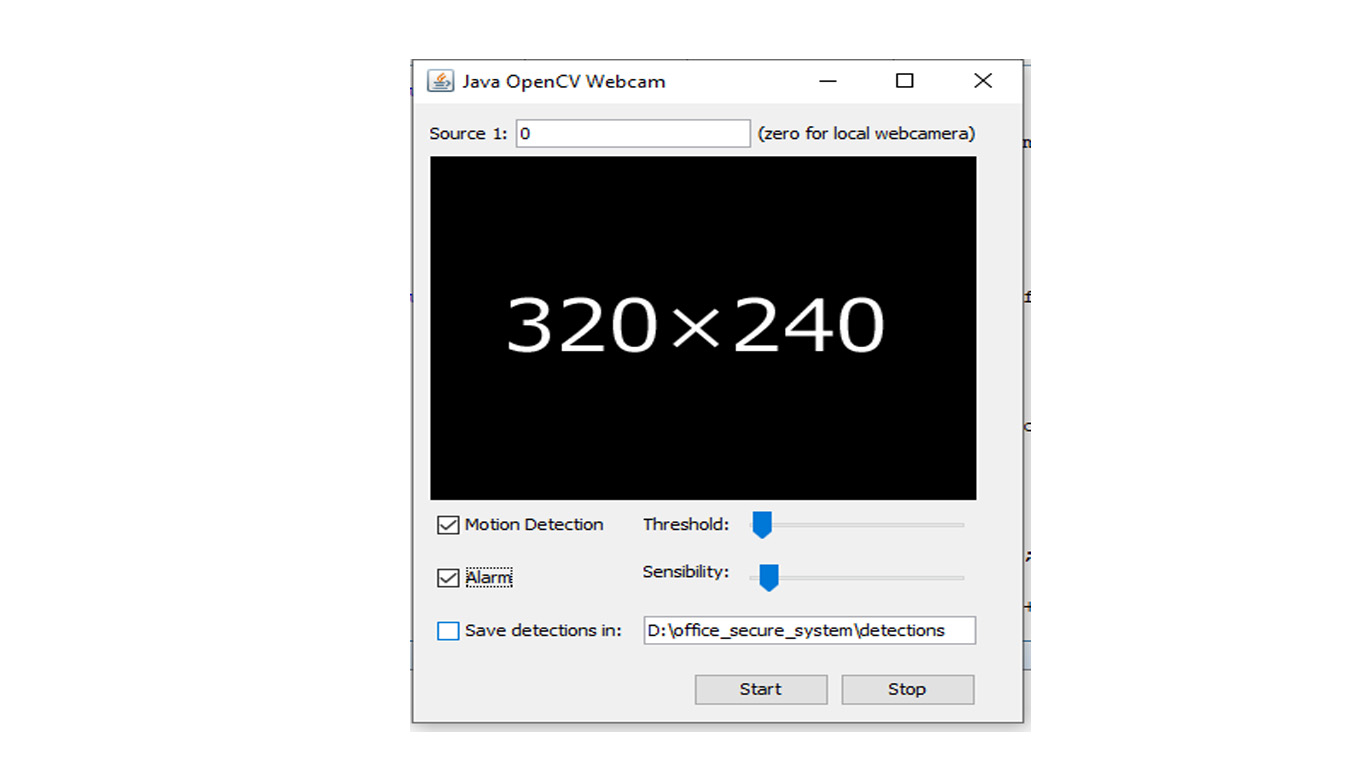
2.Inspirasi

1. **Prosedur Aplikasi Motion Detector**

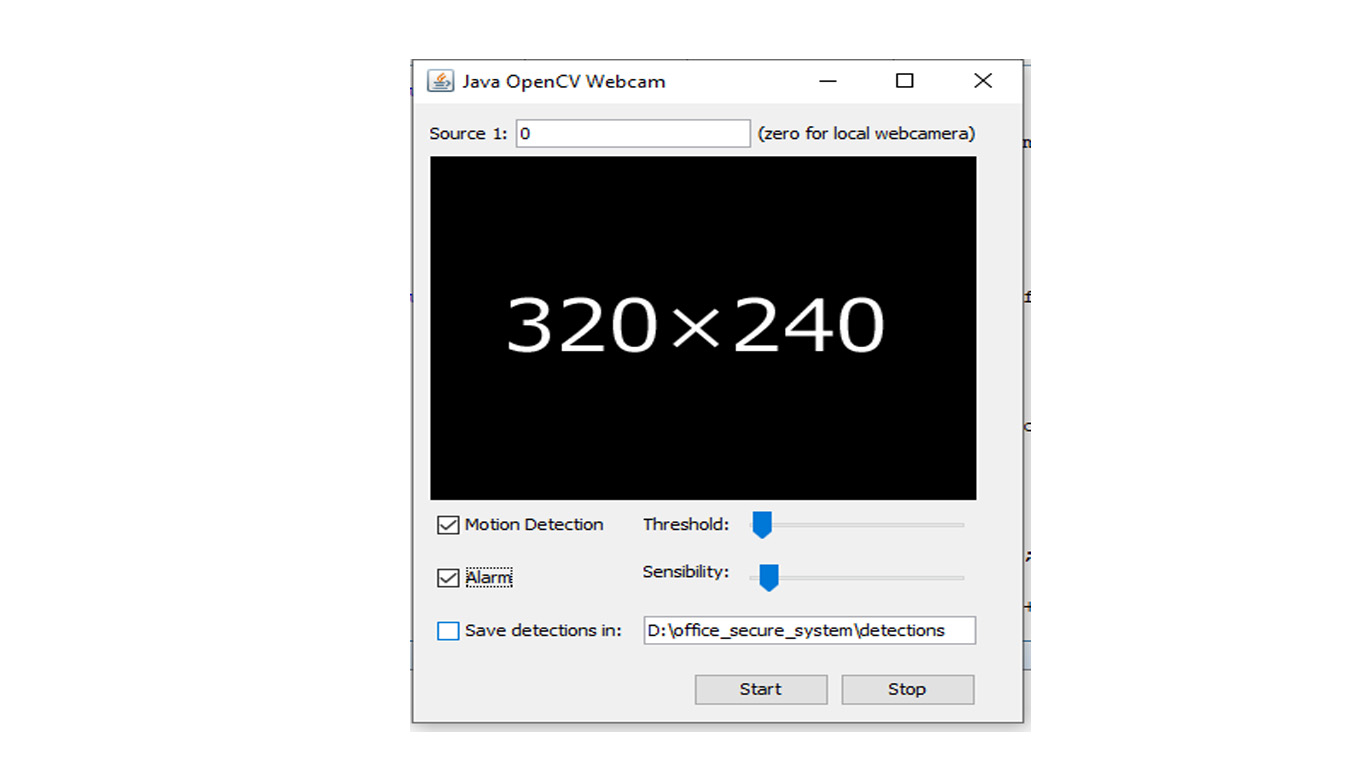
1.User Membuka Aplikasi (HALAMAN UTAMA)



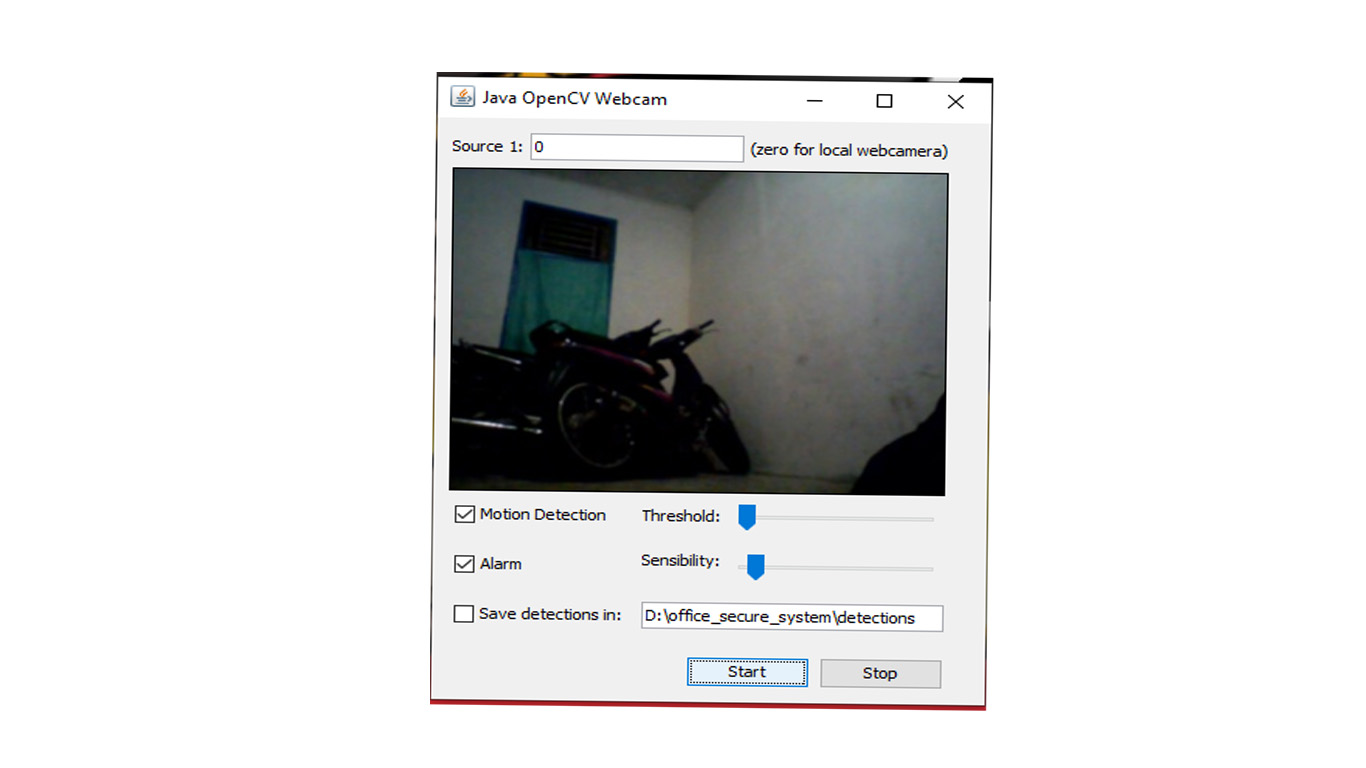
2. User Mengaktifkan Sistem Pendeteksi Gerakan Dan Alarm Dengan Cara Mencentang Kotak Check Box



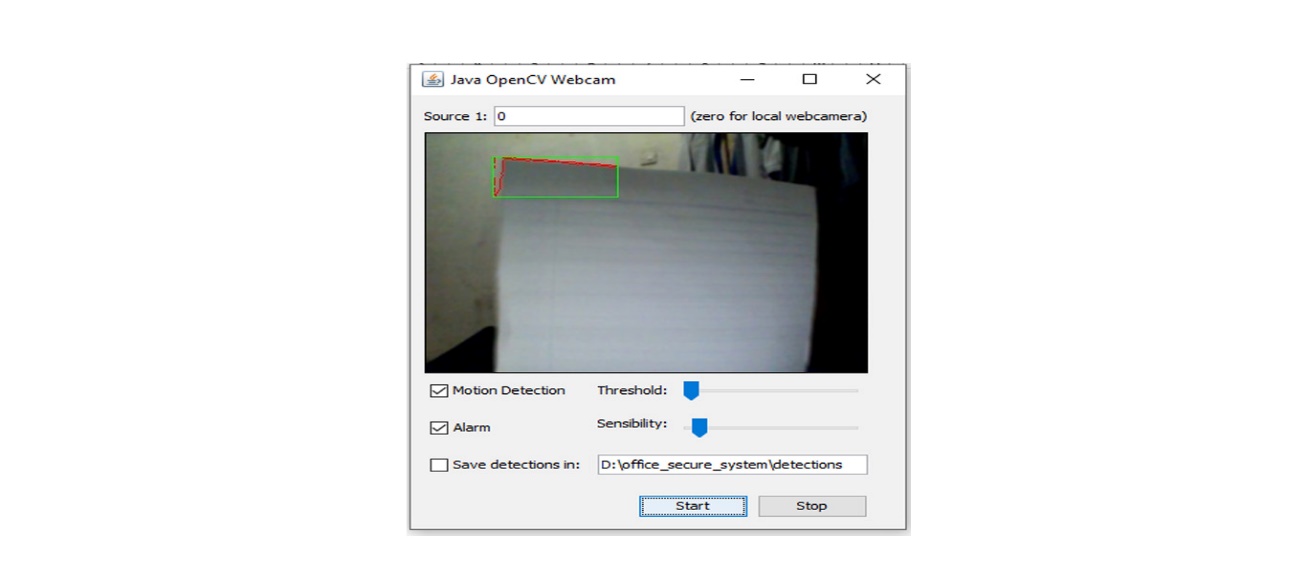
5. User Menentukan Lokasi Di mana Nanti Hasil Tangkapan gerakan akan di simpan



6. Setelah user menentukan lokasi dan mencentang semua chekbox maka selanjutnya dengan menekan start maka rekaman video akan tampil

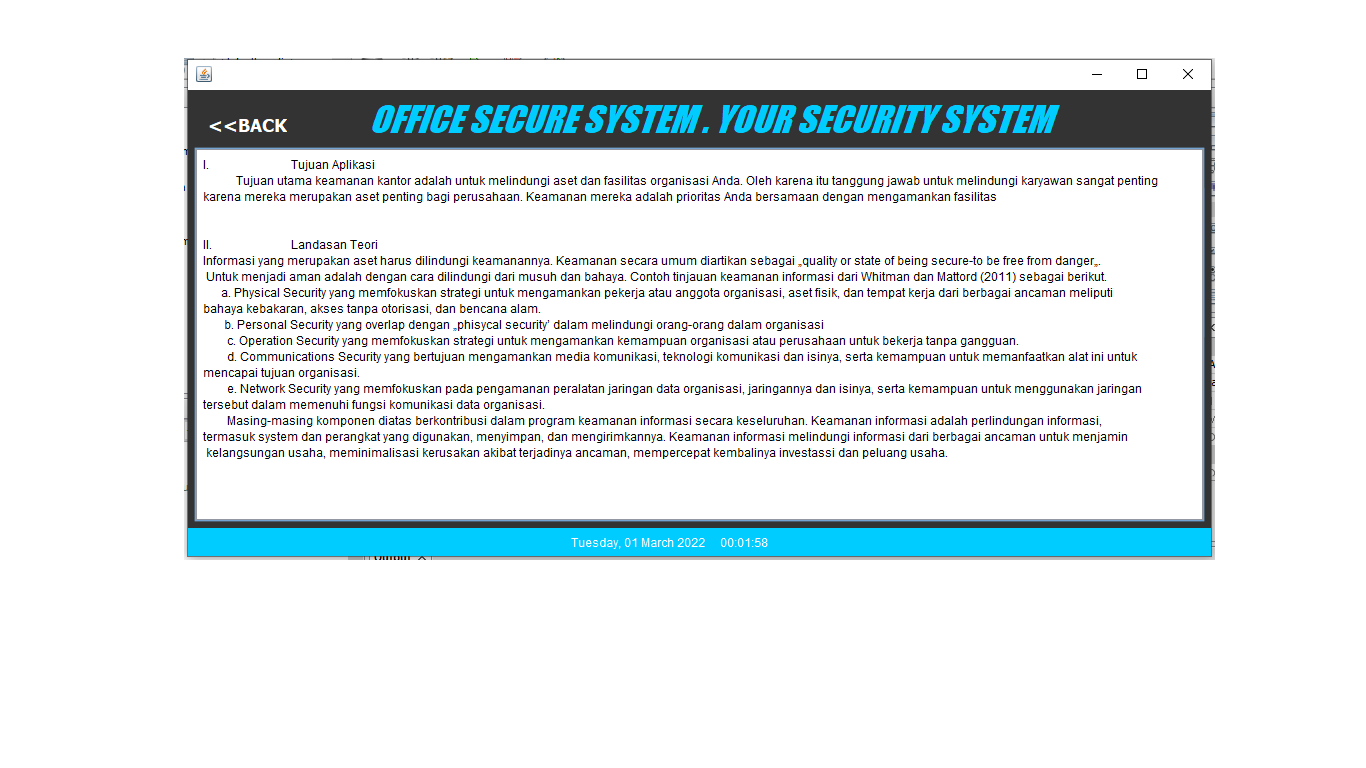


7. Jika ada gerakan yang tertangka kamera maka akan langsung tereteksi oleh sistem



8. Stop Button Untuk Menghentikan sistem/Rekaman.

9.halaman tentang aplikasi



1. Hasil Pengamatan
2. OPEN CV

OpenCV (Open Source Computer Vision Library), adalah sebuah library open source yang dikembangkan oleh intel  yang fokus untuk menyederhanakan programing terkait citra digital. Di dalam OpenCV sudah mempunyai banyak fitur, antara lain : pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, dan berbagai jenis metode AI (Artificial Intellegence). Dan menyediakan berbagai algoritma sederhana terkait Computer Vision untuk low level API.

OpenCV merupakan open source computer vision library untuk bahasa pemrograman C/C++, dan telah dikembangkan ke phyton, java, matlab. Library dapat diunduh melalui link http://sourceforge’net/projects/opencvlibrary atau melalui <http://opencv.org/releases.html>. Saat ini versi yang terbaru adalah 3.3.0 dengan tanggal rilis 3 Agustus 2017.

Intel meluncurkan versi pertama dari OpenCV pada 1999, dan awalnya memerlukan library dari Intel Image Processing Library. Kemudian dependecy tersebut ahkirnya dihilangkan sehingga terciptalah OpenCV seperti yang sekarang sebagai standalone library. OpenCV mendukung multiplatform, dapat mendukung baik windows ataupun linux, dan sekarang telah mendukung MacOSX dan android

OpenCV mempunyai banyak fitur yang dapat dimanfaatkan, berikut ini adalah fitur utama dari OpenCV antara lain :

* Image and video I/O

Dengan antar muka ini kita dapat membaca data gambar dari file, atau dari umpan video langsung. Dan juga dapat menciptakan file gambar maupun video.

* Computer Vision secara umum dan pengolahan citra digital ( untuk low dan mid level API)

Dengan antar muka ini kita dapat melakukan experimen uji coba dengan berbagai standar algorima computer vision. Termasuk juga deteksi garis, tepi, pucuk, proyeksi elips, image pyramid untuk pemrosesan gambar multi skala, pencocokan template, dan berbagai transform ( Fourier, cosine diskrit, distance transform) dan lain lain.

* Modul computer vision high level

Di dalam OpenCV juga termasuk kemampuan “high level”, seperti kemampuan tambahan untuk deteksi wajah, pengenalan wajah, termasuk optical flow

* Metode untuk AI dan machine learning

Applikasi computer vision sering kali memerlukan machine learning atau metode AI lainnya, beberapa metode tersebut tersedia dalam paket OpenCV machine learning.

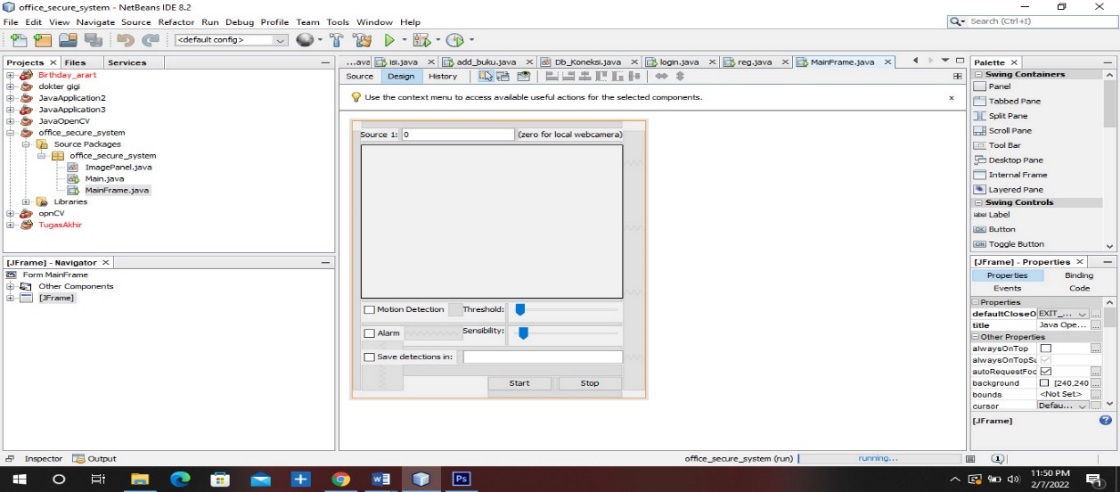
* Sampling gambar dan transformasi

Di dalam OpenCV sudah terdapat antar muka untuk substraksi subregion dari gambar, random sampling, rotating, dan lain lain.

* Metode untuk menciptakan dan menganalisa gambar biner
* Metode untuk memperhitungkan pemodelan 3D

Fungsi ini sangat bermanfaat untuk mapping dan localization, baik untuk stereo camera ataupun satu kamera dengan berbagai sudut pandang.

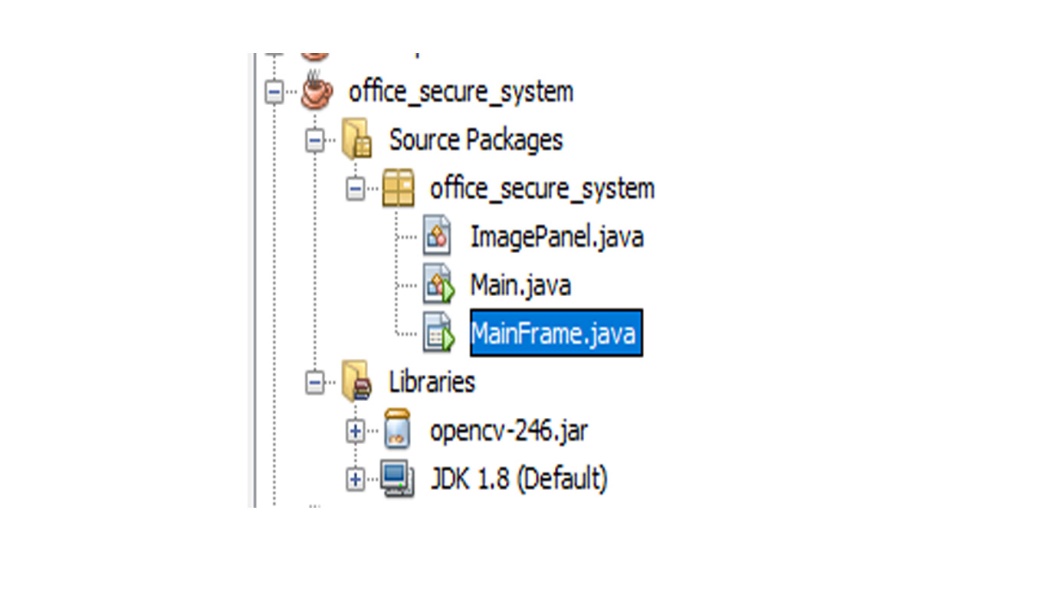
1. Desain aplikasi



Di dalam mendesain tampilan User Interface Terdapat satu kotak yang akan berisi nantinya video rekaman dari kamera dan ada tiga check box

Yang berfungsi untuk mengaktifkan suatu methode setelah itu ada tombol start dan finish yang kita ketahui bersama fungsi biasanya ke dua tombol ini ,untuk fungsi bunyi alarm belum bisa di jalankan pada OSS versi tester ini dan mungkin akan di lebih di kembangkan lagi di fungsi beikutnya

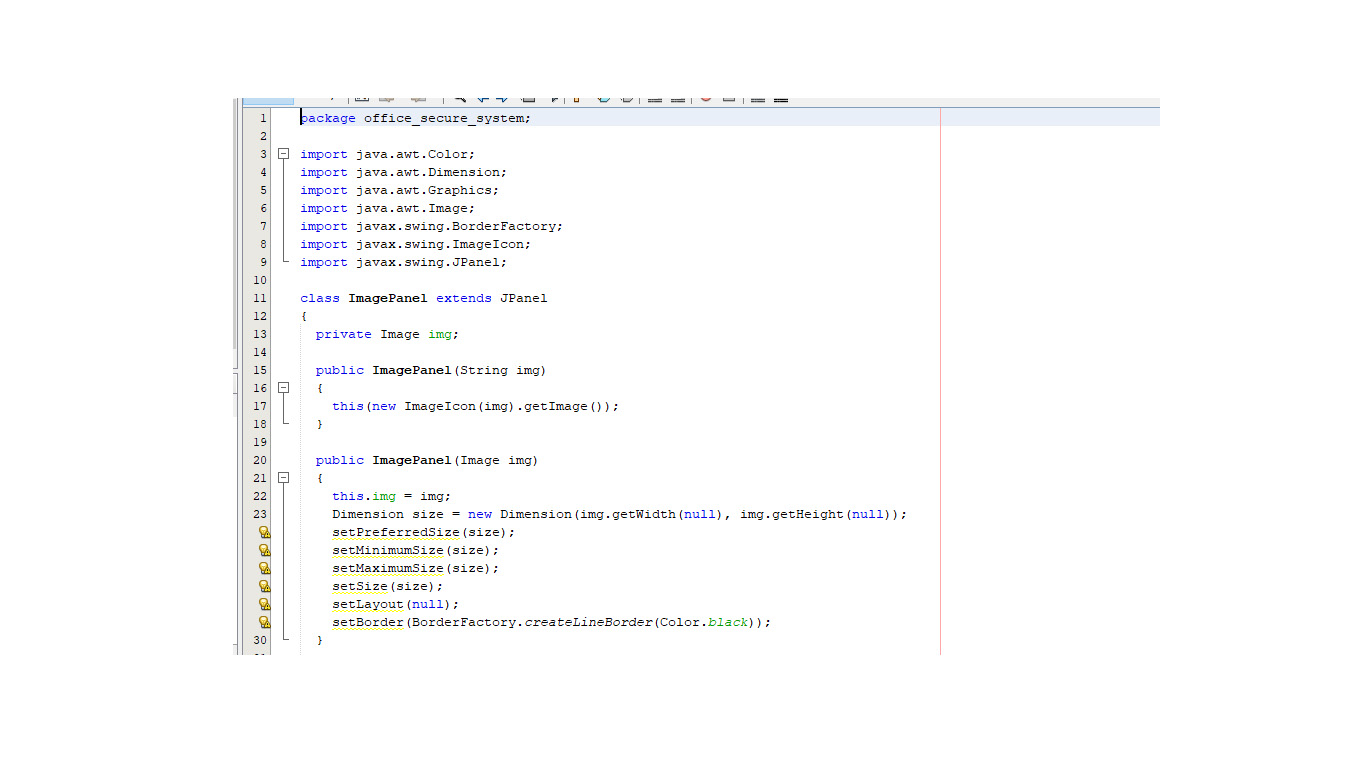
1. File Java



Terdapat 3 file ber ekstensi java (panah biru) yang mana memliki fungsi yang di dalamnya untuk menjalankan sistem

>.ImagePanel.java

Disisni terdapat beberapa fungsi seperti yang ada pada gambar

 Yang bertanda warna biru adalah Import Pada [Java](https://www.java.com/) yang  Merupakan Suatu Perintah Untuk Memasukan suatu Method atau perintah dalam Bahasa Pemrograman [Java](https://www.java.com/) sehingga perintah tersebut dapat Aktif dan digunakan atau berfungsi. Bisa kita lihat ada beberapa import yang di tunjuk oleh panah biru contohnya **“import.java.awt.image”** Yang mana berfungsi untuk memanggil perintah untuk memasukkan gambr ke dalam projek

Setelah itu yang di tunjuk oleh panah berwarna orange adalah class yang berisi source source yang berguna untuk mengatur tampilan gambar thumbnail nantinya pada saat di jalanakan

>. main.java



Gambar source di atas adalah methode yang akan berjalan ke tika tombol start di tekan



**Public void run()** adalah fungsi yang berguna ketika program mulai berjalan atau setelah tombol start di klik



Ketika tombol STOP di tekan maka fungsi ini yang akan di panggil dan program pun akan berhenti

>.Main.java

Souce code yang ada pada file java ini adalah

package office\_secure\_system;

import java.awt.BorderLayout;

import java.awt.image.BufferedImage;

import java.io.ByteArrayInputStream;

import java.io.File;

import java.io.InputStream;

import java.nio.file.Paths;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.Iterator;

import java.util.List;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import org.opencv.core.Core;

import org.opencv.core.CvType;

import org.opencv.core.Mat;

import org.opencv.core.MatOfByte;

import org.opencv.core.MatOfPoint;

import org.opencv.core.Point;

import org.opencv.core.Rect;

import org.opencv.core.Scalar;

import org.opencv.core.Size;

import org.opencv.highgui.Highgui;

import org.opencv.highgui.VideoCapture;

import org.opencv.imgproc.Imgproc;

import org.opencv.video.BackgroundSubtractorMOG;

public class MainFrame extends javax.swing.JFrame

{

static

{

System.loadLibrary(Core.NATIVE\_LIBRARY\_NAME);

}

private Boolean begin = false;

private Boolean firstFrame = true;

private VideoCapture video = null;

private CaptureThread thread = null;

private MatOfByte matOfByte = new MatOfByte();

private BufferedImage bufImage = null;

private InputStream in;

private Mat frameaux = new Mat();

private Mat frame = new Mat(240, 320, CvType.CV\_8UC3);

private Mat lastFrame = new Mat(240, 320, CvType.CV\_8UC3);

private Mat currentFrame = new Mat(240, 320, CvType.CV\_8UC3);

private Mat processedFrame = new Mat(240, 320, CvType.CV\_8UC3);

private ImagePanel image;

private BackgroundSubtractorMOG bsMOG = new BackgroundSubtractorMOG();

private int savedelay = 0;

String currentDir = "";

String detectionsDir = "detections";

public MainFrame()

{

initComponents();

image = new ImagePanel(new ImageIcon("figs/320x240.gif").getImage());

jPanelSource1.add(image, BorderLayout.CENTER);

currentDir = Paths.get(".").toAbsolutePath().normalize().toString();

detectionsDir = currentDir + File.separator + detectionsDir;

// new java.io.File( "." ).getCanonicalPath();

//System.out.println("Current dir: " + currentDir);

//System.out.println("Detections dir: " + detectionsDir);

jTextFieldSaveLocation.setText(detectionsDir);

}

private void start()

{

//System.out.println("You clicked the start button!");

if(!begin)

{

int sourcen = Integer.parseInt(jTextFieldSource1.getText());

System.out.println("Opening source: " + sourcen);

video = new VideoCapture(sourcen);

if(video.isOpened())

{

thread = new CaptureThread();

thread.start();

begin = true;

firstFrame = true;

}

}

}

private void stop()

{

//System.out.println("You clicked the stop button!");

if(begin)

{

try

{

Thread.sleep(500);

}

catch(Exception ex)

{

}

video.release();

begin = false;

}

}

public static String getCurrentTimeStamp()

{

SimpleDateFormat sdfDate = new SimpleDateFormat("yyyy-MM-dd\_HH-mm-ss");//dd/MM/yyyy

Date now = new Date();

String strDate = sdfDate.format(now);

return strDate;

}

public ArrayList<Rect> detection\_contours(Mat frame, Mat outmat)

{

Mat v = new Mat();

Mat vv = outmat.clone();

List<MatOfPoint> contours = new ArrayList();

Imgproc.findContours(vv, contours, v, Imgproc.RETR\_LIST, Imgproc.CHAIN\_APPROX\_SIMPLE);

double maxArea = 100;

int maxAreaIdx;

Rect r;

ArrayList<Rect> rect\_array = new ArrayList();

for(int idx = 0; idx < contours.size(); idx++)

{

Mat contour = contours.get(idx);

double contourarea = Imgproc.contourArea(contour);

if(contourarea > maxArea)

{

// maxArea = contourarea;

maxAreaIdx = idx;

r = Imgproc.boundingRect(contours.get(maxAreaIdx));

rect\_array.add(r);

Imgproc.drawContours(frame, contours, maxAreaIdx, new Scalar(0, 0, 255));

}

}

v.release();

return rect\_array;

}

class CaptureThread extends Thread

{

@Override

public void run()

{

if(video.isOpened())

{

while(begin == true)

{

//video.read(frameaux);

video.retrieve(frameaux);

Imgproc.resize(frameaux, frame, frame.size());

frame.copyTo(currentFrame);

if(firstFrame)

{

frame.copyTo(lastFrame);

firstFrame = false;

continue;

}

if(jCheckBoxMotionDetection.isSelected())

{

Imgproc.GaussianBlur(currentFrame, currentFrame, new Size(3, 3), 0);

Imgproc.GaussianBlur(lastFrame, lastFrame, new Size(3, 3), 0);

//bsMOG.apply(frame, processedFrame, 0.005);

Core.subtract(currentFrame, lastFrame, processedFrame);

//Core.absdiff(frame,lastFrame,processedFrame);

Imgproc.cvtColor(processedFrame, processedFrame, Imgproc.COLOR\_RGB2GRAY);

//

int threshold = jSliderThreshold.getValue();

//Imgproc.adaptiveThreshold(processedFrame, processedFrame, 255, Imgproc.ADAPTIVE\_THRESH\_MEAN\_C, Imgproc.THRESH\_BINARY\_INV, 5, 2);

Imgproc.threshold(processedFrame, processedFrame, threshold, 255, Imgproc.THRESH\_BINARY);

ArrayList<Rect> array = detection\_contours(currentFrame, processedFrame);

///\*

if(array.size() > 0)

{

Iterator<Rect> it2 = array.iterator();

while(it2.hasNext())

{

Rect obj = it2.next();

Core.rectangle(currentFrame, obj.br(), obj.tl(),

new Scalar(0, 255, 0), 1);

}

}

//\*/

if(jCheckBoxAlarm.isSelected())

{

double sensibility = jSliderSensibility.getValue();

//System.out.println(sensibility);

double nonZeroPixels = Core.countNonZero(processedFrame);

//System.out.println("nonZeroPixels: " + nonZeroPixels);

double nrows = processedFrame.rows();

double ncols = processedFrame.cols();

double total = nrows \* ncols / 10;

double detections = (nonZeroPixels / total) \* 100;

//System.out.println(detections);

if(detections >= sensibility)

{

//System.out.println("ALARM ENABLED!");

Core.putText(currentFrame, "MOTION DETECTED",

new Point(5,currentFrame.cols()/2), //currentFrame.rows()/2 currentFrame.cols()/2

Core.FONT\_HERSHEY\_TRIPLEX , new Double(1), new Scalar(0,0,255));

if(jCheckBoxSave.isSelected())

{

if(savedelay == 2)

{

String filename = jTextFieldSaveLocation.getText() + File.separator + "capture\_" + getCurrentTimeStamp() + ".jpg";

System.out.println("Saving results in: " + filename);

Highgui.imwrite(filename, processedFrame);

savedelay = 0;

}

else

savedelay = savedelay + 1;

}

}

else

{

savedelay = 0;

//System.out.println("");

}

}

//currentFrame.copyTo(processedFrame);

}

else

{

//frame.copyTo(processedFrame);

}

currentFrame.copyTo(processedFrame);

Highgui.imencode(".jpg", processedFrame, matOfByte);

byte[] byteArray = matOfByte.toArray();

try

{

in = new ByteArrayInputStream(byteArray);

bufImage = ImageIO.read(in);

}

catch(Exception ex)

{

ex.printStackTrace();

}

//image.updateImage(new ImageIcon("figs/lena.png").getImage());

image.updateImage(bufImage);

frame.copyTo(lastFrame);

try

{

Thread.sleep(1);

}

catch(Exception ex)

{

}

}

}

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanelSource1 = new javax.swing.JPanel();

jLabelSource1 = new javax.swing.JLabel();

jTextFieldSource1 = new javax.swing.JTextField();

jButtonStart = new javax.swing.JButton();

jButtonStop = new javax.swing.JButton();

jCheckBoxMotionDetection = new javax.swing.JCheckBox();

jSliderThreshold = new javax.swing.JSlider();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jCheckBoxAlarm = new javax.swing.JCheckBox();

jLabel3 = new javax.swing.JLabel();

jSliderSensibility = new javax.swing.JSlider();

jTextFieldSaveLocation = new javax.swing.JTextField();

jCheckBoxSave = new javax.swing.JCheckBox();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Java OpenCV Webcam");

jPanelSource1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

javax.swing.GroupLayout jPanelSource1Layout = new javax.swing.GroupLayout(jPanelSource1);

jPanelSource1.setLayout(jPanelSource1Layout);

jPanelSource1Layout.setHorizontalGroup(

jPanelSource1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 318, Short.MAX\_VALUE)

);

jPanelSource1Layout.setVerticalGroup(

jPanelSource1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 238, Short.MAX\_VALUE)

);

jLabelSource1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabelSource1.setText("Source 1:");

jTextFieldSource1.setText("0");

jButtonStart.setText("Start");

jButtonStart.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonStartActionPerformed(evt);

}

});

jButtonStop.setText("Stop");

jButtonStop.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonStopActionPerformed(evt);

}

});

jCheckBoxMotionDetection.setText("Motion Detection");

jSliderThreshold.setMaximum(255);

jSliderThreshold.setPaintLabels(true);

jSliderThreshold.setPaintTicks(true);

jSliderThreshold.setValue(15);

jLabel1.setText("Threshold:");

jLabel2.setText("(zero for local webcamera)");

jCheckBoxAlarm.setText("Alarm");

jLabel3.setText("Sensibility:");

jSliderSensibility.setMinimum(1);

jSliderSensibility.setPaintLabels(true);

jSliderSensibility.setPaintTicks(true);

jSliderSensibility.setValue(10);

jCheckBoxSave.setText("Save detections in:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jCheckBoxMotionDetection)

.addGap(18, 18, 18)

.addComponent(jLabel1))

.addGroup(layout.createSequentialGroup()

.addComponent(jCheckBoxAlarm)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSliderThreshold, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addComponent(jSliderSensibility, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanelSource1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabelSource1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextFieldSource1, javax.swing.GroupLayout.PREFERRED\_SIZE, 138, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel2)))

.addGroup(layout.createSequentialGroup()

.addComponent(jCheckBoxSave)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jTextFieldSaveLocation, javax.swing.GroupLayout.PREFERRED\_SIZE, 195, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addComponent(jButtonStart, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButtonStop, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(27, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabelSource1)

.addComponent(jTextFieldSource1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanelSource1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSliderThreshold, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jCheckBoxMotionDetection)

.addComponent(jLabel1)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSliderSensibility, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 6, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextFieldSaveLocation, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jCheckBoxSave))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonStart)

.addComponent(jButtonStop)))

.addGroup(layout.createSequentialGroup()

.addComponent(jCheckBoxAlarm)

.addGap(0, 0, Short.MAX\_VALUE)))

.addContainerGap())

);

pack();

}// </editor-fold>

private void jButtonStartActionPerformed(java.awt.event.ActionEvent evt)

{

start();

}

private void jButtonStopActionPerformed(java.awt.event.ActionEvent evt)

{

stop();

}

public static void main(String args[])

{

* 1. **Penutup**

Sistem ke amanan kantor ini belum bisa di katakan sempurna di karnakan masih banyak ke kurangan yang ada pada salah satu aplikasi ke amanan kantor ini ,Mungkin seteleh aplikasi ini sempurna office secure system akan lebih mengmbangkan sistem tentang ke amanan kantor contohnya Kryptograph Pesan Rahasia

Maka dari itu kami dari pihak OSS(office secure system) memintak masukan dari pihak terkai tentang sistem MOTION DETECTOR ini karna masukan lebih berharga terhadap perkembangan sistem ini .

1. **Daftar Pustaka**

[www.slideshare.net](http://www.slideshare.net)

atalian.id

cemarakukeluarga.blogspot.com

binus.ac.id  
penelitianilmiah.com

repository.dinamika.ac.id

coolors.co

github.com

answers.opencv.org