LAMPIRAN A
Program Kamera

```
import math
import cv2
import numpy as np
import config
import serialArduino
import utils
from capture import cam, camFrnt, fps2, frameModule,
frameModuleFrnt, makeText
from circleMasking import frameCircleModule
from masking import maskedFrameModule
from millis import millis
from objectDetect import objectDetectBall, objectDetectBallFrnt
from updateJson import (brightnessValue, brightnessValueFrnt,
focusValue, focusValueFrnt, settings, update_json_trackbar)
cv2.namedWindow('main', cv2.WINDOW_AUTOSIZE)
cv2.setWindowProperty('main', cv2.WND_PROP_FULLSCREEN,
cv2.WND_PROP_FULLSCREEN)
cv2.namedWindow('slave', cv2.WINDOW_AUTOSIZE)
cv2.setWindowProperty('slave', cv2.WND_PROP_FULLSCREEN,
cv2.WND_PROP_FULLSCREEN)

objectSelected = 0
dataObjectArr = [None]*len(config.hsvBola)
lastDataObjectConditionedArr = [None]*len(config.hsvBola)
countObjectFixNoneArr = [0]*len(config.hsvBola)

dataObjectArrFrnt = [None]*len(config.hsvBola)
lastDataObjectConditionedArrFrnt = [None]*len(config.hsvBola)
countObjectFixNoneArrFrnt = [0]*len(config.hsvBola)
prevMillis = [0]
serialTime = 0
firstPointY = 0
firstPointX = 0
Dx = 0
Dy = 0
detectStatus = 0
```

```
last= [0,0]
detectStatus = 0
error = 9999
XYcon = [0,0]
distFrnt = 9999
firstPointX = 0
firstPointY = 0
data = ""
directPre = ""
startMillis = [0]
millisKirim = [0]
thresholdRead = 10
Ydist = 0
preNew =[0,0]
lastTime = 0
intervalPre = 0

focalLength = (49 * 400)/20 #FL = (widhtBallFrame *
distanceMeasuement) / realWidthBall

def set_brghtnes (val):
   cam.set (cv2.CAP_PROP_BRIGHTNESS, val)

def set_focus(val):
   cam.set(28, val)

def set_brghtnesFrnt (val):
   camFrnt.set (cv2.CAP_PROP_BRIGHTNESS, val)

def set_focusFrnt(val):
   camFrnt.set(28, val)

cv2.createTrackbar('Brightness', 'main', settings['Brightness'], 255,
lambda x: update_json_trackbar(x, 'Brightness', 'Brightness', 'main'))
cv2.createTrackbar('Focus', 'main', settings['Focus'], 255, lambda x:
update_json_trackbar(x, 'Focus', 'Focus', 'main'))

cv2.createTrackbar('Brightness', 'slave', settings['BrightnessFrnt'], 255,
lambda x: update_json_trackbar(x, 'Brightness', 'BrightnessFrnt', 'slave'))
```

```
cv2.createTrackbar('Focus', 'slave', settings['FocusFrnt'], 255, lambda x:
update_json_trackbar(x, 'Focus', 'FocusFrnt', 'slave'))

def mapping(x,in_min,in_max, out_min, out_max): #function mapping
kalo make fungsi ga bisa float
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min

def lastObject(x, y, interval) :
    global lastTime
    if (millis() > lastTime):
        lastTime = millis() + interval
        last[0] = x
        last[1] = y
    return last

def convertXY(X,Y) :
    if X > 320 :
        # Xnew = mapping(X, 320, 640, 0, 320)
        Xnew = X -320
    elif X < 320 :
        # Xnew = mapping(X, 0, 320, -320, 0)
        Xnew = X - 320
    elif X > 640:
        Xnew = 320
    elif X < 0:
        Xnew = -320
    else:
        Xnew = 0
    if Y > 500 :
        Ynew = 0
    else :
        Ynew = abs(480 - Y)
    return Xnew, Ynew

def setErrorPre(X):
    if X > 10 or X < -10 :
        error= X/10
    else :
        error = 0
    return error

def on_keyboard(key):
    if key == ord('a') or key == ord('A'):
        # tambahkan logic jika sudah dibuka, jangan eksekusi dibawah
        if config.useSerial != True:
            serialArduino.buka_koneksi(config.serialAddress, 9600)
            serialArduino.kirim_perintah("a")

    if key == ord('c') or key == ord("C"):
        if config.useSerial != False:
            serialArduino.tutup_koneksi()

def predicDirect(Dx , Dy):
    if (Dx < thresholdRead and Dx > -thresholdRead)  and (Dy < -
thresholdRead):
        directPre = "atas"
    elif (Dx < thresholdRead and Dx > -thresholdRead)  and (Dy >
thresholdRead):
        directPre = "bawah"
    elif((Dx > thresholdRead)  and (Dy < thresholdRead and Dy > -
thresholdRead)):
        directPre = "kanan"
    elif((Dx < -thresholdRead)  and (Dy < thresholdRead and Dy > -
thresholdRead)):
        directPre = "kiri"
    elif((Dx > thresholdRead)  and (Dy > thresholdRead)):
        directPre = "kananBawah"
    elif((Dx < -thresholdRead)  and (Dy > thresholdRead)):
        directPre = "kiriBawah"
    elif((Dx > thresholdRead)  and (Dy < -thresholdRead)):
        directPre = "kananAtas"
    elif((Dx < -thresholdRead)  and (Dy < -thresholdRead)):
        directPre = "kiriAtas"
    else:
        directPre = ""
    return directPre
```

```
def gain(lastX,D):
    if((lastX > 320 and D > 0) and (lastX < 320 and D < 0)):
        gainOut = 5
    else:
        gainOut = 1
    return gainOut

while 1:
    frame = frameModule()
    frame = frameCircleModule(frame)

    frameFrnt = frameModuleFrnt()
    brightnessValue = cv2.getTrackbarPos('Brightness', 'main')
    focusValue = cv2.getTrackbarPos('Focus', 'main')
    brightnessValueFrnt = cv2.getTrackbarPos('Brightness', 'slave')
    focusValueFrnt = cv2.getTrackbarPos('Focus', 'slave')

    set_brghtnes(brightnessValue)
    set_focus(focusValue)
    set_brghtnesFrnt(brightnessValueFrnt)
    set_focusFrnt(focusValueFrnt)

    for i in range(len(config.hsvBola)) :
        # arduino.get()
        dataObject, dataObjectConditioned = objectDetectBall(frame)
        dist = 0.1201 * (dataObjectConditioned[0]**(2)) - (12.036 *
dataObjectConditioned[0]) + 317.23
        dist = int(dist)
        # arduino.send('b1', dataObjectConditioned[0],
dataObjectConditioned[1], dist)
        lastDataObjectConditionedArr[i] = dataObjectConditioned
        dataObjectArr[i] = dataObject
    i = 0

    for i in range(len(config.hsvBolaFrnt)) :
        # arduino.get()
        dataObjectFrnt, dataObjectConditionedFrnt, widthBall =
objectDetectBallFrnt(frameFrnt)
        # print(dataObjectConditionedFrnt[0])
        # arduino.send('b1', dataObjectConditioned[0],
dataObjectConditioned[1], dist)
        lastDataObjectConditionedArrFrnt[i] = dataObjectConditionedFrnt
        dataObjectArrFrnt[i] = dataObjectFrnt
    i = 0

    for dataObject in dataObjectArr :
    #    arduino.get()
        frameMasked = maskedFrameModule(frame)
        makeText(frameMasked, 20,50,str(dataObjectConditioned[1]))
        if dataObject[0] != -1 :
            utils.masking(frameMasked, config.hsvBola[i]['name'],
            dataObject[0], dataObject[1],1)
        i = i + 1
    i = 0

    for dataObjectFrnt in dataObjectArrFrnt :
    #    arduino.get()
        if dataObjectFrnt[0] != -1 :
            utils.masking(frameFrnt,str(dataObjectFrnt[0]) + ","
            + str(dataObjectFrnt[1]), dataObjectFrnt[0], dataObjectFrnt[1],0)
            distFrnt = (20 * focalLength)/float(widthBall)
            Ydist = mapping(dataObjectFrnt[1], 0, 480, 480, 0)
            distFrnt = utils.regressJarak(distFrnt)
            lastXY = lastObject(dataObjectFrnt[0], dataObjectFrnt[1], 1000)
            XYcon = convertXY(dataObjectFrnt[0], dataObjectFrnt[1])

            Dx = dataObjectFrnt[0] - lastXY[0]
            Dy = dataObjectFrnt[1] - lastXY[1]
            R = math.sqrt((Dy * Dy) + (Dx * Dx))
            theta = math.degrees(math.atan2(Dy, Dx))
            Dx2 = R * math.cos(math.radians(theta))
            Dy2 = R * math.sin(math.radians(theta))
            if(R > thresholdRead or R < -thresholdRead) and Ydist > 150
and Dy2 > -5:
                directPre = predicDirect(Dx2 , Dy2)
                if(distFrnt < 250 and (Dx2 > 10 or Dx2 < -10)): gainP = 3
                else : gainP = 1
                preX = dataObjectFrnt[0] + Dx2
                preY = dataObjectFrnt[1] + Dy2
                preNew = convertXY(preX, preY)
                error = (setErrorPre(preNew[0]))
                error = error * gainP
                makeText(frameFrnt, 500, 300, "err:" + str(int(error)))
                cv2.line(frameFrnt,(dataObjectFrnt[0], dataObjectFrnt[1]),
(int(preX), int(preY)), (0,255,0),2)
            else :
                preNew = [0,0]
                theta = 0
                Dx = 0
                Dy = 0
                preX =0
                preY = 0
                directPre = ""
                error = 0

            cv2.line(frameFrnt,(lastXY[0], lastXY[1]), (dataObjectFrnt[0],
dataObjectFrnt[1]), (0,0,255),2)
            # cv2.line(frameFrnt,(lastXY[0], lastXY[1]), (firstPointX,
firstPointY), (0,0,255),2)
            makeText(frameFrnt, 50, 450, "dir:" + directPre)
            makeText(frameFrnt, 50, 50, "Xc:" + str(dataObjectFrnt[0]))
            makeText(frameFrnt, 50, 100, "Yc:" + str(dataObjectFrnt[1]))
            makeText(frameFrnt, 50, 350, "dist:" + str(int(distFrnt)))
            makeText(frameFrnt, 50, 150, "XL:" + str(lastXY[0]))
            makeText(frameFrnt, 50, 200, "YL:" + str(lastXY[1]))
            makeText(frameFrnt, 50, 250, "Dx:" + str(Dx))
            makeText(frameFrnt, 50, 300, "Dy:" + str(Dy))
            makeText(frameFrnt, 500, 50 , "Teta:" + str(int(theta)))
            makeText(frameFrnt, 500, 100 , "Dx2:" + str(int(Dx2)))
            makeText(frameFrnt, 500, 150 , "Dy2:" + str(int(Dy2)))
            makeText(frameFrnt, 500, 200, "preX:" + str(int(preNew[0])))
            makeText(frameFrnt, 500, 250, "preY:" + str(int(preNew[1])))
        else :
            Dx = 0
            Dy = 0
            Ydist = 0
            error = 9999
            preX =0
            XYcon = [9999,9999]
            preNew = [9999,0]
            distFrnt = 0

        i = i + 1
    i = 0

    data =
"errorPre,"+str(error)+",Xfrnt,"+str(XYcon[0])+",Yfrnt,"+str(XYcon[1])
+",jarakB,"+str(Ydist)+",bolaT,"+str(dataObjectConditioned[1])+",dx,"
+str(Dx)+",dy,"+str(Dy)+","+'\n'
    print(data)
    makeText(frame, 50, 430, "Ser::" + str(config.useSerial))
    # if delay_millis(millisKirim, 100):
    # if(millis() > serialTime):
    # serialTime = millis() + 100
    serialArduino.sendData(data)
    cv2.imshow('main', frameMasked)
    cv2.imshow('slave', frameFrnt)
    key = cv2.waitKey(1)
    on_keyboard(key)
    if key == ord('q') or key == ord('Q'):
        break
```

LAMPIRA B
Program Arduino

```
#include "variable_nasional.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4); // set the LCD address to 0x27 for a
16 chars and 2 line display

int buttonA, buttonB, buttonC, button;
float Vy, Vx, V0;
int d1, d2, d3;

//pneumatic control
#define relayKn 45
#define relayAt 47
#define relayKr 49
#define proxyKn 37
#define proxyAt 35
#define proxyKr 33

#define echo1 5
#define echo2 4
#define echo3 3
int jarakKn, jarakBk, jarakKr;
bool readProxy, readKn, readKr, readAt;

float elapsed;
float yaw, mapyaw, inversmap, newYaw;
int setAwal = 0;
int maxPwmX = 250;
bool resetMpu;
float r, teta;
float dt_lap;
float r_lap, teta_lap, teta_xy, z_robot;
float errorx_lap, errory_lap;

//variable odometry
```

```
float enc_kanan, enc_kiri, enc_belakang;

struct lapangan {
  float data;
  float last;
  float out, lap;
  int last_lapangan;
};
lapangan X = { 0, 0, 0, 425};
lapangan Y = { 0, 0, 0, HomeY};
lapangan Z;

struct valPID {
  float Kp;
  float Ki;
  float Kd;
  int maxOut;
  int minOut;

  float integral;
  double error;
  double lastError;
  float Output;
  double Derivatif;
  double deltaError;
};

#define maks_odom 150
#define maks_pwm_kinematik 150

valPID PID_X = { 5, 0.01, 0.005, maks_odom, -maks_odom };
valPID PID_Y = { 5, 0.01, 0.005, maks_odom, -maks_odom };
valPID PID_Z = { 4, 0.009, 0.02, maks_odom, -maks_odom};
valPID PID_Xpre = { 10, 0, 0.01, maxPwmX, -maxPwmX};
valPID PID_XCam = { 5, 0, 0.01, maxPwmX, -maxPwmX};

void setup() {
  serial_setup();
  motor_setup();
  LCD_Setup();
  compas_setup();
  setup_ultra();
  setup_pneumatic();
}

void loop() {
  parsing_pc();
  get_last_bola();
  compas();
  serialSlave();
  odometry();
  peta();
  ultra();
  button_read();
  pneumatic();
  blockMode();
}

int LastX, checkLastPost, detect;
void blockMode() {
  if (udp_terima.KDBolaT != 9999) {
    checkLastPost = 1;
    PID_Xpre = PID_compute(0, -udp_terima.bolaX, PID_Xpre, 0, 0);
    PID_Y = PID_compute(0, Y.data, PID_Y, 0, 0);
    PID_Z = PID_compute(0, newYaw, PID_Z, 0, 0);
    gerak(PID_Xpre.Output, PID_Y.Output, PID_Z.Output);
    if (jarakKn == 0 && udp_terima.bolaX > 0 || jarakKr == 0 &&
udp_terima.bolaX < 0) {
      checkLastPost = 0;
      PID_X = PID_compute(LastX, X.lap, PID_X, 0, 0);
      PID_Y = PID_compute(0, Y.data, PID_Y, 0, 0);
      PID_Z = PID_compute(0, newYaw, PID_Z, 0, 0);
      gerak(0, PID_Y.Output, PID_Z.Output);
    }
  }
  else if (udp_terima.KDBolaT == 9999)  {
    if (udp_terima.bolaT != 404 && -100 < udp_terima.bolaT &&
udp_terima.bolaT < 100) {
      checkLastPost = 1;
      PID_XCam = PID_compute(0, 0, PID_XCam, 1, udp_terima.bolaT);
      PID_Y = PID_compute(0, Y.data, PID_Y, 0, 0);
      PID_Z = PID_compute(0, newYaw, PID_Z, 0, 0);
      gerak(PID_XCam.Output, PID_Y.Output, PID_Z.Output);
    }
    else {
      checkLastPost = 1;
      PID_X = PID_compute(425, X.lap, PID_X, 0, 0);
      PID_Y = PID_compute(0, Y.data, PID_Y, 0, 0);
      PID_Z = PID_compute(0, newYaw, PID_Z, 0, 0);
      gerak(PID_X.Output, PID_Y.Output, PID_Z.Output);
    }
  }
  if ((jarakKn == 0 && udp_terima.bolaT >= 0 && udp_terima.bolaT <=
120) || (jarakKr == 0 && udp_terima.bolaT < 0 && udp_terima.bolaT >=
-120)) {
    checkLastPost = 0;
    PID_X = PID_compute(LastX, X.lap, PID_X, 0, 0);
    PID_Y = PID_compute(0, Y.data, PID_Y, 0, 0);
    PID_Z = PID_compute(0, newYaw, PID_Z, 0, 0);
    gerak(PID_X.Output, PID_Y.Output, PID_Z.Output);
  }
  if (checkLastPost = 1)LastX = X.lap;
}
```

BIODATA PENULIS

Nama	:	Ardi Mujiarto
Tempat/Tanggal Lahir	:	Brebes / 30 September 2003
Alamat	:	Rt 01, Rw 01 Ds. Langkap Kec. Bumiayu Kab. Brebes
Hp	:	(+62) 81331975761
Hobi	:	Main Efootballe (PS-an)
Motto	:	Kalo hidupmu bermasalah, emang hidup kamu doang yang bermasalah??

Riwayat Pendidikan :

SD Negeri Langkap 01	Tahun 2009 – 2015
SMP Islam Ta'allumul Huda Bumiayu	Tahun 2015 – 2018
SMA Islam Ta'allumul Huda Bumiayu	Tahun 2018 – 2021
Politeknik Negeri Cilacap	Tahun 2021 – 2024

Penulis telah mengikuti seminar Tugas Akhir pada tanggal 6 Agustus 2024, sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A. Md).