DAFTAR PUSTAKA

Anas,Nasri.(2019). Implementasi Algoritma genetika untuk pengendalian perangkat listrik pada miniature ruangan cerdas. Universitas Hasanuddin.

Amini,Nurlatifah.,2022.Implementasi Algoritma Genetika untuk seleksi fitur pada klasifikasi genre music menggunakan metode random forest. Jurnal informatika polinema(JIP). ISSN: 2614-6371 E-ISSN: 2407-070X. Vol.9,Edisi 1.

Ardiansyah, H., & Junianto, M. B. S., (2022). Penerapan algoritma genetika untuk penjadwalan mata pelajaran. Jurnal Media Informatika Budidarma, 6(1), 329-336. http://dx.doi.org/10.30865/mib.v6i1.3418

Arrobi,Ridho,Muhammad.,Nuralam.2022.Rancangan Bangun Robot Lego Sebagai Media Pembelajana Sekolah Dasar Berbasis Arduino Mega, TEKNIK Vol 2 No. 2

Ayu,Monavia,Rizaty., Oktober 2022.Survei:1 dari 3 Remaja Indonesia Punya Masalah Kesehatan Mental.dataInonesia.id.

Azizah,Muftiyatul.,2022.Perbandingan Hybrid Algoritma Genetika Dengan MultiLayer perceptron dan geometric Brownian Motion Untuk Memprediksi Saham.Journal of mathematis education and science,Vol.5.,No.2.

BlissLights LLC',How Lighting Affects Mood and Human Behavior'. Available :https://blisslights.com/blogs/blisslights/how-lighting-affects-mood-and-human-behaviour. Accessed Januari 2023

Blume,Christine, Garbazza,Corrado, & Spitschan, Manuel. 2019. Effects of light on human circadian rhythms,sleep and mood. Somnologie (Berl). 2019; 23(3): 147–156. Published online 2019 Aug 20. doi: 10.1007/s11818-019-00215-x.

Cheng.2020.Facial Expression Recognition Method Based on Improved VGG Convolutional Neural Network. Internasional Journla of Pattern Recognition and Arttificial Intelegence. Vo.34.no.7. DOI: 10.1142/S0218001420560030

Chul.,B.,K.2018. A Brief Review of Facial Emotion Recognition Based on Visual Information. Sensors, 18, 401; doi:10.3390/s18020401

Desiana, Eva. (2016). Performance algoritma genetika (GA) pada penjadwalan mata pelajaran. InfoTekjar. Vol 1,No.1.

Desmira.,Aribowo,Didik.,dkk,2022,Aplikasi Sensor LDR (light Dependent Resistor) untuk Efisiensi Energi Pada Lampu Penerangan Jalan Umum, Jurnal PROSISKO Vol. 9 No.1.

E. Mest, "AccorHotels details new Smart Room concept.," 2018. [Online]. Available: https://www.hotelmanagement.net/operate/accorhotels-detailsnew-smart-room-concept..[Accessed 18 September 2022].

Golmohammadi,Rostam., Yousefi, Hanieh,.(2021). Effects of Light on Attention and Reaction Time : A systematic Review. doi: 10.34172/jrhs.2021.66

H. Awal, "Perancangan Prototype Smart Home Dengan Konsep Internet Of Thing (IoT) Berbasis Web Server," Majalah Ilmiah, vol. 26, pp. 64-79, 2019.

Jiayi Bao.”Effect of lighting illuminance and colour temperature on mental workload in an office setting”.Juli 2021.

M. Anshar, A. U. Ejah, M. Zaenab, R. Ida, A. Emir, Alwi, Nurindasari, H. Dicky, N. Anas, K. Ahmad and M. Akkas, "Smart Room Design: A Pilot Project," Proc. The 1st EPI Int. Conf. on Science and Engineering, vol. 1, no. 4, pp. 19-26, 2017.

M. Anshar and N. Anas, "Hardware and User Perspective Assessment on Application of Smart Door Access," Proceeding The 2nd EPI International Confrence on Science and Engineering, 2018.

M.B.C. Aries, F. Beute, G. Fischl (2020). Assessment protocol and effects of two dynamic light patterns on human well-being and performance in a simulated and operational office environment. https://doi.org/10.1016/j.jenvp.2020.101409

Made Sujiwa Putra, I.(2018). Penerapan Algoritma Genetika daln Implementasi dalam Matlab. Universitas Udayana.

Marcella,dewi.,dkk.2022. Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19. Universitas Multi Data Palembang. Vol. 3, No. 1. E–ISSN: 2775-8796

Minaee,Shervin.,dkk.,2021. Deep-Emotion:Facial Expression Recognition Using Attentional Convolutional Network.Expression Group. Received: 15 February 2021 / Revised: 20 April 2021 / Accepted: 23 April 2021 / Published: 27 April 2021.https://doi.org/10.3390/s21093046.

Mutia,Cindy,Annur., 27 Oktober 2022. Presentase Masyarakat Indonesia yang Punya Masalah Kesehatan Mental di Indonesia.Databoks. https://databoks.katadata.co.id/datapublish/2022/10/27/survei-populix-1-dari-2-penduduk-indonesia-punya-masalah-kesehatan-mental....Diakses pada tanggal 23 Januari 2023

Münch,Mirjam.,Schmieder,Michael.,Bieler,Katharina.(2022). Bright Light Delights: Effects of Daily Light Exposure on Emotions, Restactivity Cycles, Sleep and Melatonin Secretion in Severely Demented Patients.

Nur,Yuanita,Indah,Sari.15 Juni 2022.Mengenal Emosi Negatif, ketahui penyebab, Jenis,bahaya dan Cara Mengendalikannya. SehatQ:Kesehatan Mental. https://www.sehatq.com/artikel/mengenal-emosi-negatif-dan-cara-mengatasinya...diakses pada tanggal 23 Januari 2023.

Prayoga,Senanjung,dkk.,2020.Auto Tuning PID pada Lebview Menggunakan Metode Algortma Genetika.Kournal of Applied Science,Electrical Engineering and computer technology. Vol2.,No.1.

Ruijun Chen, Meng-Chun Tsai and Yaw-Shyan Tsay (2022). Effect of Color Temperature and Illuminance on Psychology, Physiology, and Productivity: An Experimental Study. https://doi.org/10.3390/en15124477.

Setiawan,wahyudi.2019. Perbandingan arsitektur convolutional neural network untuk klasifikasi fundus. Universitas Trunojoyo. Vol.7,no.2. ISSN 2088-2130.

Zhang,Zhihui., Maria ,Fort ,Mirand,Josep., Mateu, Gimenez,Lluis., 2022.The Effects of White versus Coloured Light in Waiting Rooms on People's Emotions, Received: 30 July 2022 / Revised: 26 August 2022 / Accepted: 30 August 2022 / Published: 1 September 2022.

WuJie1010,Facial-Expression-Recognition.Pytorch, github, https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch.

Yulianti, Vira Nevriza. 2023. Penerapan metode backward untuk menentukan persamaan regresi linear berganda pada duagaan tindak pidana di kota binjai. Universitas Sumatra Utara. ARABI: Jurnal Matematika dan Pendidikan Matematika Volume 6 Nomor 1.

Zahir, L. A. (2022). Algoritma genetika sebagai solusi permasalahan persamaan linear matematika. Journal of Research in Science and Mathematics Education (J-RSME), 1(2), 87-95. https://doi.org/10.56855/jrsme.v1i2.62

Zhang,Zhuihui.,dkk (2022). The Effects of White versus Coloured Light in Waiting Rooms on People's Emotions. https://doi.org/10.3390/buildings12091356

LAMPIRAN

LAMPIRAN 1 : Facial Expression Recognition

```
from __future__ import print_function
import argparse
import os
import cv2
import pandas as pd
import numpy as np
import torch
from torchvision import transforms
from PIL import Image
from torch.utils.data import DataLoader, Dataset
import torch.nn.functional as F
from deep_emotion import Deep_Emotion
from data_loaders import Plain_Dataset, eval_data_dataloader
import smbus
import os
from time import sleep

device = torch.device("cuda:0" if torch.cuda.is_available() else
"cpu")

transformation =
transforms.Compose([transforms.ToTensor(),transforms.Normalize((0.5,)
,(0.5,))])
dataset = Plain_Dataset(csv_file='data'+'/'+'train.csv',img_dir =
'data'+'/'+'test/',datatype = 'finaltest',transform = transformation)
test_loader =  DataLoader(dataset,batch_size=64,num_workers=0)

net = Deep_Emotion()
print("Deep Emotion:-", net)
model = 'deep_emotion1-2-128-0.005.pt'
net.load_state_dict(torch.load(model))
net.to(device)
net.eval()

#Model Evaluation on test data
classes = ('Angry', 'Angry', 'Angry' 'Happy','Sad', 'Happy' , 'Sad')
total = []
test_acc = False
if test_acc:
    with torch.no_grad():
        for data, labels in test_loader:
            data, labels = data.to(device), labels.to(device)
            outputs = net(data)
            pred = F.softmax(outputs,dim=1)
            classs = torch.argmax(pred,1)
            wrong = torch.where(classs !=
labels,torch.tensor([1.]).cuda(),torch.tensor([0.]).cuda())
            acc = 1- (torch.sum(wrong) / 64)
            total.append(acc.item())
```

```
    print('Accuracy of the network on the test images: %d %%' % (100
* np.mean(total)))


#helper_function for real time testing
def load_img(path):
    img = Image.open(path)
    img = transformation(img).float()
    img = torch.autograd.Variable(img,requires_grad = True)
    img = img.unsqueeze(0)
    return img.to(device)
cam = True
if cam:
    cap = os.system('libcamera-still -o /home/iascr/Downloads/Deep-
Emotion-master/imgs/oke.jpg -t 4000 --width 740 --height 780')
    while True:
        face_cascade =
cv2.CascadeClassifier('./cascade_model/haarcascade_frontalface_defaul
t.xml')
        # Read the frame
        img = cv2.imread("/home/iascr/Downloads/Deep-Emotion-
master/imgs/oke.jpg")
        # Convert to grayscale
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        # Detect the faces
        faces = face_cascade.detectMultiScale(gray, 1.1, 4)
        # Draw the rectangle around each face
        #roi = cv2.resize(gray,(48,48))
        #cv2.imwrite("D:/Kuliah/Tugas-akhir/Realtime-Face-Emotion-
Recognition-main/oke.jpg", roi)
        facess = face_cascade.detectMultiScale(img)
        if len(facess) == 0:
            print("no face detected")
        else:
            imgg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            imgg = cv2.resize(imgg,(48,48))
            imgg = np.expand_dims(imgg, axis = 0)
            imgg = np.expand_dims(imgg, axis = 0)
            imgg = imgg/255.0
            data = torch.from_numpy(imgg)
            data = data.type(torch.FloatTensor)
            data = data.to(device)
            out = net(data)
            pred = F.softmax(out, dim = 1)
            classs = torch.argmax(pred,1)
            wrong = torch.where(classs !=
3,torch.tensor([1.]).cpu(),torch.tensor([0.]).cpu())
            classs = torch.argmax(pred,1)
            prediction = classes[classs.item()]
            print(prediction)
```

LAMPIRAN 2 : Algoritma Genetika

```
%Mainsmatroom
clear
clc
besar_populasi=10;
laju_mutasi = 0.05;
total_generasi = 150;
%cahayaT = 200;
prev.gen = [0 0 0 0];

prev.fitness = 0;
delay = 0;
global s;
s = serialport('COM3',9600);
fopen(s);
isLooping = true;
data2 =
zeros(total_generasi,9);
data1 =
zeros(total_generasi,8);

i = 1;
for i = 1:1
    i = i+1;

[ga,data,populasi_awal,cahayaT]
= GAmain(data1,data2, prev,
besar_populasi,laju_mutasi,
total_generasi);
gen(1) = prev.gen(1)-ga.gen(1);
gen(2) = prev.gen(2) -
ga.gen(2);
gen(3) = prev.gen(3) -
ga.gen(3);
gen(4) = prev.gen (4) -
ga.gen(4);
%fitness GAcalculateFitness
(prev,ga.gen,suhuId,cahayaT,
suDa,suLu,caDa,caLu);
    pause(delay);
    serialWrite(gen,s)
end
clear s
```

```
%calculate nilai fitness
function [Euclidean_distanse ,
Euclidean_distanse_cahaya,
Euclidean_distanse_Suhu, luxTambahan,
luxKiraan]=
GAcalculateFitness(prev,gen,suhuId,
cahayaT,suDa,suLu,caDa,caLu)

    lampu1       = gen(1);
    kipas       = gen(2);
    jendela     = gen(3);
    lampu2       = gen(4);
    suhuTarget  = suhuId;
    luxTambahan =
luxLampu(lampu1,lampu2,jendela);
    %disp(luxTambahan)
    %luxSebelum  =
luxLampu(prev.gen(1),prev.gen(4),prev.gen(3))+
luxJendela(caLu, prev.gen(3));
    luxKiraan   = abs(caDa + luxTambahan);

    if suDa < suLu && suDa < suhuTarget
        suhuKiraan = suDa +
suhu(kipas,jendela);
    elseif suDa > suLu && suDa > suhuTarget
        suhuKiraan =  suDa -
suhu(kipas,jendela);

    elseif suDa <= suLu && suDa >= suhuTarget
        suhuKiraan = suDa +
suhu(kipas,jendela);
    elseif suDa >= suLu && suDa <= suhuTarget
        suhuKiraan = suDa -
suhu(kipas,jendela);
    end

   Euclidean_distanse_Suhu     =suhuId -
suhuKiraan;
   Euclidean_distanse_cahaya  = abs(cahayaT -
luxKiraan);
   Euclidean_distanse        = abs(cahayaT -
luxKiraan);

end
```

GAMain

```
function [mutant1, data,populasi_awal,cahayaT] = GAmain(data,data2,
prev, besar_populasi,laju_mutasi,total_generasi)
clear s
global s
[populasi,cahayaT,suhuId] = GAcreatePopulation(prev, besar_populasi);
populasi_awal = populasi;
isLooping = true;
generasi = 0;
check = 0;
ED=0;
ulang = true;
solusi = struct();
while isLooping
    if ulang == true
    [parent1,parent2] = GAselection(populasi);
    [child1,child2] = GAcrossover(parent1,parent2);
    disp("Crossover selesai");
    mutant1 = GAmutation(child1,laju_mutasi);
    mutant2 = GAmutation(child2,laju_mutasi);
    %disp(mutant1)
    %disp(mutant2)
    %disp("Mutasi selesai")
    [~, suDa, suLu, caDa, caLu, ~] = serialRead(s);
    [mutant1.Euclidean_distanse,mutant1.Euclidean_distanseCahaya,
mutant1.Euclidean_distanseSuhu, mutant1.luxlampu,
mutant1.lumencapaianindividu] =
GAcalculateFitness(prev,mutant1.gen,suhuId,cahayaT,suDa,suLu,caDa,caLu
);
    [mutant2.Euclidean_distanse,mutant2.Euclidean_distanseCahaya,
mutant2.Euclidean_distanseSuhu, mutant2.luxlampu,
mutant2.lumencapaianindividu] =
GAcalculateFitness(prev,mutant2.gen,suhuId,cahayaT,suDa,suLu,caDa,caLu
);
    disp("Mengirim inidvidu baru");
    %populasi = GAregeneration(children,populasi);
    for p = 1 : length(populasi)
        ED = ED + populasi(p).Euclidean_distanse;
    end
    ED1 = ED + mutant1.Euclidean_distanse +
mutant2.Euclidean_distanse;
    mutant1.fitness = 1 -  (mutant1.Euclidean_distanse/ED1);
    mutant2.fitness = 1 -  (mutant2.Euclidean_distanse/ED1);
    [solusi1,~] = GAselection1(populasi);
    ulang = false;
    coba1 = [mutant1.gen(1),mutant1.gen(4),
mutant1.gen(2),mutant1.gen(3)];
    coba2 = [mutant2.gen(1),mutant2.gen(4),
mutant2.gen(2),mutant2.gen(3)];
    end
    if generasi >= 2
           LX = ismember(coba1,data(1:generasi,2:5),"rows");
```

```
            disp(LX);
            disp("masukmi");
            if LX == 1
                LX2 = ismember(coba2,data(1:generasi,2:5),"rows");
                if LX2 == 1
                    ulang = true;
                    disp("masukmi2");
                else
                     solusi = mutant2;
                     generasi = generasi + 1;
                     disp("Generasi sekarang");
                     disp(generasi);
                     data(generasi,:) = [generasi,
solusi.gen(1),solusi.gen(4), solusi.gen(2), solusi.gen(3),
solusi.Euclidean_distanse, solusi.luxlampu,
solusi.lumencapaianindividu];
                end
            else
                    solusi = mutant1;
                     generasi = generasi + 1;
                     disp("Generasi sekarang");
                     disp(generasi);
                     data(generasi,:) = [generasi,
solusi.gen(1),solusi.gen(4), solusi.gen(2), solusi.gen(3),
solusi.Euclidean_distanse, solusi.luxlampu,
solusi.lumencapaianindividu];
            end
    else
        solusi = mutant1;
        generasi = generasi + 1;
        disp("Generasi sekarang");
        disp(generasi);
        data(generasi,:) = [generasi, solusi.gen(1),solusi.gen(4),
solusi.gen(2), solusi.gen(3),   solusi.Euclidean_distanse,
solusi.luxlampu,  solusi.lumencapaianindividu];
    end
    children = [solusi mutant2];
    populasi = GAregeneration(children,populasi);
   if generasi == total_generasi
                isLooping = false;
                disp("masukmi3");
   else
       ulang= true;
       disp(solusi);
   end
 end
end
```

GAcreatePopulation

```
function[ populasi,cahayaT,suhuId] = GAcreatePopulation(prev,
besar_populasi)
```

```
clear s
    populasi = struct();
    ED = 0;
    fitness = 0;
    gensekarang = 0;
    prev.gen = [0 0 0 0];
    global s;
    for k=1:besar_populasi
        creategen = true;
        while creategen
        gen = GAcreateGen();
        populasi(k).gen = gen;
        %a = serialport('COM3',9600);
             gen = populasi(k).gen;
            [emotion, suDa, suLu, caDa, caLu, ~] = serialRead(s);
            emosi = [700 700 700];
            suhuId = 23;
            cahayaT= emosi(emotion);
            [Euclidean_distanse, Euclidean_distanseCahaya,
Euclidean_distanseSuhu, luxlampu, lumencapaianindividu] =
GAcalculateFitness(prev,gen,suhuId,cahayaT,suDa,suLu,caDa,caLu);
            ED = ED + Euclidean_distanse;
            populasi(k).Euclidean_distanse = Euclidean_distanse;
            populasi(k).Euclidean_distanseCahaya =
Euclidean_distanseCahaya;
            populasi(k).Euclidean_distanseSuhu =
Euclidean_distanseSuhu;
            populasi(k).luxlampu = luxlampu;
            populasi(k).lumencapaianindividu = lumencapaianindividu;
            disp(Euclidean_distanse);
            creategen = false;
            if Euclidean_distanse > 50
                creategen = true;
            else
                creategen = false;
            end
        end
        gensekarang = gensekarang + 1;
        disp("gensekarang");
        disp(gensekarang);
    end
         for i = 1:besar_populasi
            Euclidean_distanse = populasi(i).Euclidean_distanse;
            fitness = 1 - (Euclidean_distanse./ED);
            populasi(i).fitness = fitness;
             % populasi(i).luxSebelum = luxSebelum;
        end
clear s
end
```

serialWrite

```
function [] = serialWrite(ga,s)
a = int2str(ga(1));
b = int2str(ga(2));
c = int2str(ga(3));
d = int2str(ga(4));

fwrite(s,a);
pause(15);
fwrite(s,b);
pause(15);
fwrite(s,c);
pause(15);
fwrite(s,d);
pause(15);
end
```

luxLampu

```
function luxLampu =
luxLampu(lampu1,lampu2,jendela)

    m1 = 1.70436367920813;
    m2 = 1.58152272466225;
    m3 = -0.630000000000001;
    c = 267.908933713251;

    luxLampu = ( (m1*(lampu1)) +
(m2*(lampu2)) + (jendela*(m3)) +
c);
end
```

```
function [child1,child2] =
GAcrossover(parent1,parent2)
    child1 = parent1;
    child2 = parent2;

    CP =
(1:(length((parent1.gen)/2)));

    child1.gen(1:CP) =
parent2.gen(1:CP);
    child2.gen(1:CP) =
parent1.gen(1:CP);
end
```

```
function [gen] = GAcreateGen()
    gen = zeros(1,4);
    gen(1) = randi([0,255]);
    gen(2) = randi([1,3]);
    gen(3) = randi([0,100]);
    gen(4) = randi([0,255]);
end
```

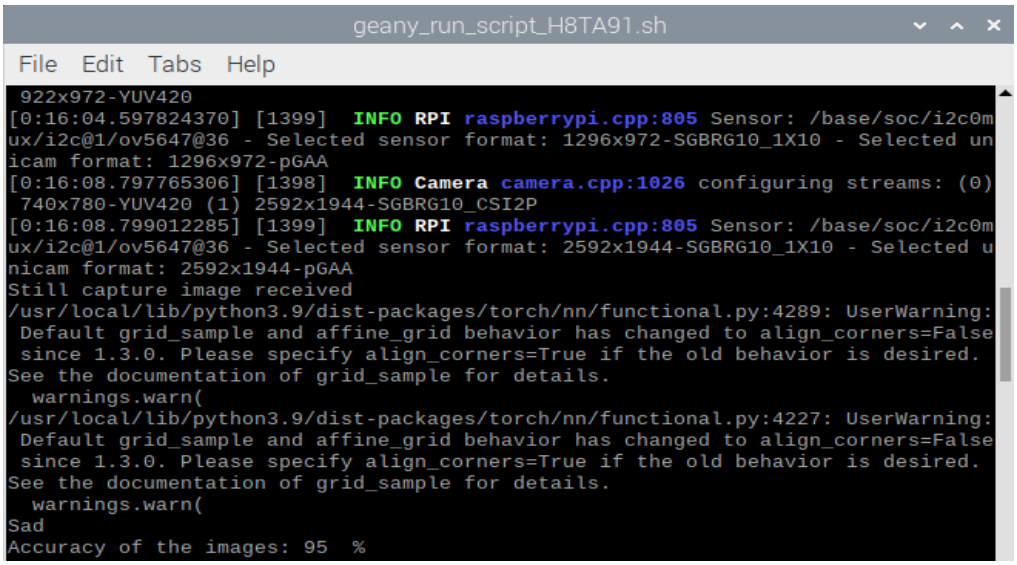
```
function [parent1,parent2] =
GAselection(populasi)

    fitness_data =
zeros(1,length(populasi));

    for i=1:length(populasi)
        fitness_data(i) =
populasi(i).fitness;
        %fitness_data(i) =
populasi(i).Euclidean_distanse;
    end

    [~,index] =max(fitness_data);
    parent1 = populasi(index);

    populasi(index) = [];
```

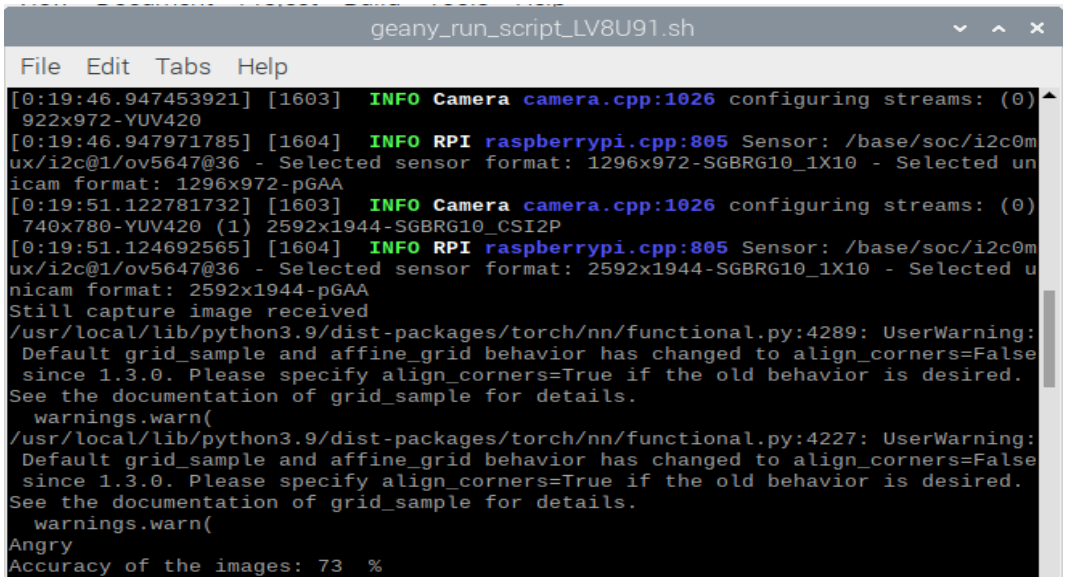
```
function mutant =
GAmutation(child,laju_mutasi)
    mutant = child;
    if rand <= laju_mutasi
        mutant.gen(1) =
randi([0,224]);
    end
    if rand <= laju_mutasi
        mutant.gen(2) =
randi([1,3]);
    end
    if rand <= laju_mutasi
        mutant.gen(3) =
randi([0,100]);
    end
    if rand <= laju_mutasi
```

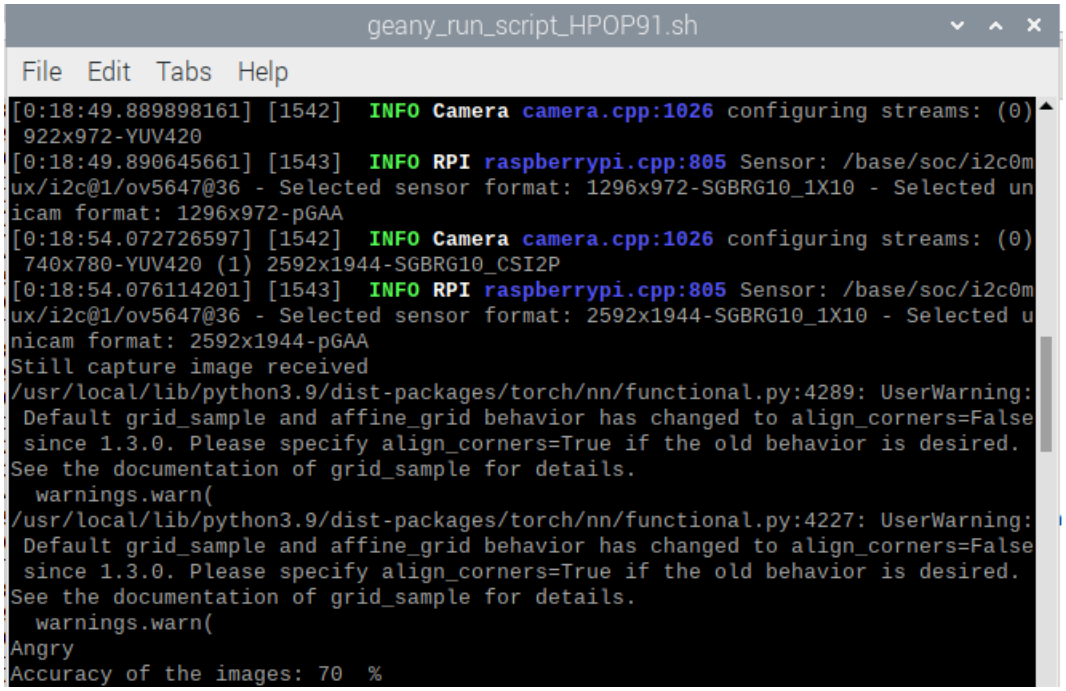
```
    fitness_data(index) = [];

    [~,index] = max(fitness_data);
    parent2 = populasi(index);
end
```

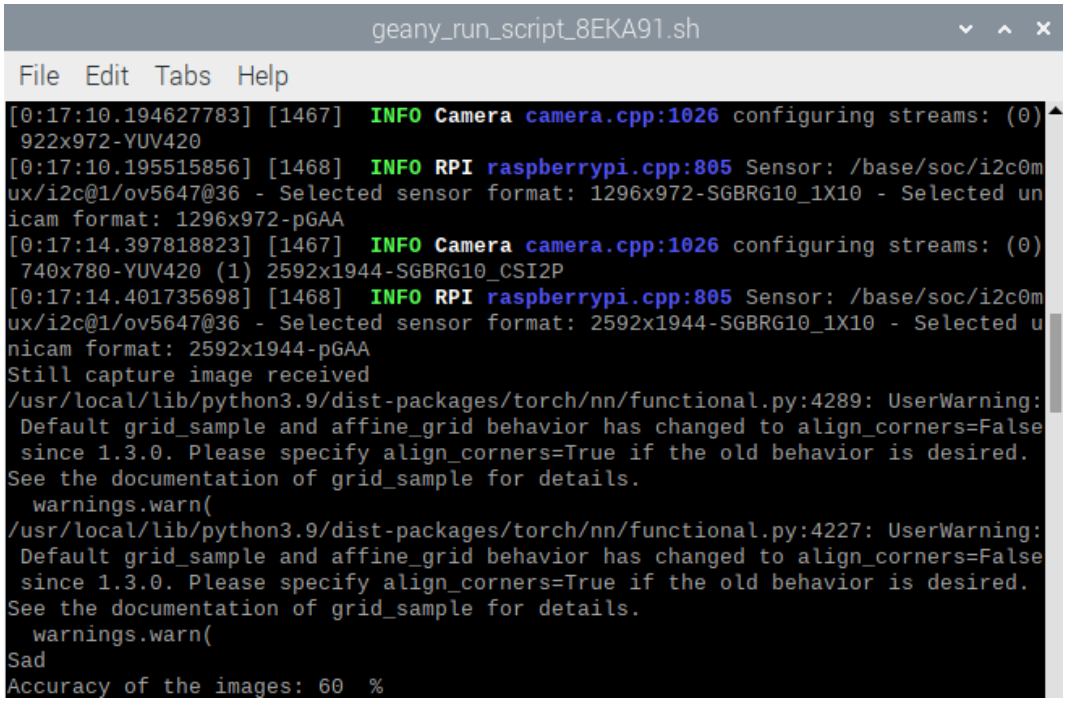
```
        mutant.gen(4) =
randi([0,255]);
    end
end
```

LAMPIRAN 3. Hasil pengujian Facial Expression Recognition




geany_run_script_LV8U91.sh
File Edit Tabs Help
[0:19:46.947453921] [1603] INFO Camera camera.cpp:1026 configuring streams: (0) 922x972-YUV420
[0:19:46.947971785] [1604] INFO RPI raspberrypi.cpp:805 Sensor: /base/soc/i2c0mux/i2c@1/ov5647@36 - Selected sensor format: 1296x972-SGBRG10_1X10 - Selected unicam format: 1296x972-pGAA
[0:19:51.122781732] [1603] INFO Camera camera.cpp:1026 configuring streams: (0) 740x780-YUV420 (1) 2592x1944-SGBRG10_CSI2P
[0:19:51.124692565] [1604] INFO RPI raspberrypi.cpp:805 Sensor: /base/soc/i2c0mux/i2c@1/ov5647@36 - Selected sensor format: 2592x1944-SGBRG10_1X10 - Selected unicam format: 2592x1944-pGAA
Still capture image received
/usr/local/lib/python3.9/dist-packages/torch/nn/functional.py:4289: UserWarning: Default grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Please specify align_corners=True if the old behavior is desired.
See the documentation of grid_sample for details.
warnings.warn(
/usr/local/lib/python3.9/dist-packages/torch/nn/functional.py:4227: UserWarning: Default grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Please specify align_corners=True if the old behavior is desired.
See the documentation of grid_sample for details.
warnings.warn(
Angry
Accuracy of the images: 73 %


geany_run_script_HPOP91.sh
File Edit Tabs Help
[0:18:49.889898161] [1542] INFO Camera camera.cpp:1026 configuring streams: (0) 922x972-YUV420
[0:18:49.890645661] [1543] INFO RPI raspberrypi.cpp:805 Sensor: /base/soc/i2c0mux/i2c@1/ov5647@36 - Selected sensor format: 1296x972-SGBRG10_1X10 - Selected unicam format: 1296x972-pGAA
[0:18:54.072726597] [1542] INFO Camera camera.cpp:1026 configuring streams: (0) 740x780-YUV420 (1) 2592x1944-SGBRG10_CSI2P
[0:18:54.076114201] [1543] INFO RPI raspberrypi.cpp:805 Sensor: /base/soc/i2c0mux/i2c@1/ov5647@36 - Selected sensor format: 2592x1944-SGBRG10_1X10 - Selected unicam format: 2592x1944-pGAA
Still capture image received
/usr/local/lib/python3.9/dist-packages/torch/nn/functional.py:4289: UserWarning: Default grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Please specify align_corners=True if the old behavior is desired.
See the documentation of grid_sample for details.
warnings.warn(
/usr/local/lib/python3.9/dist-packages/torch/nn/functional.py:4227: UserWarning: Default grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Please specify align_corners=True if the old behavior is desired.
See the documentation of grid_sample for details.
warnings.warn(
Angry
Accuracy of the images: 70 %


geany_run_script_8EKA91.sh
File Edit Tabs Help
[0:17:10.194627783] [1467] INFO Camera camera.cpp:1026 configuring streams: (0) 922x972-YUV420
[0:17:10.195515856] [1468] INFO RPI raspberrypi.cpp:805 Sensor: /base/soc/i2c0mux/i2c@1/ov5647@36 - Selected sensor format: 1296x972-SGBRG10_1X10 - Selected unicam format: 1296x972-pGAA
[0:17:14.397818823] [1467] INFO Camera camera.cpp:1026 configuring streams: (0) 740x780-YUV420 (1) 2592x1944-SGBRG10_CSI2P
[0:17:14.401735698] [1468] INFO RPI raspberrypi.cpp:805 Sensor: /base/soc/i2c0mux/i2c@1/ov5647@36 - Selected sensor format: 2592x1944-SGBRG10_1X10 - Selected unicam format: 2592x1944-pGAA
Still capture image received
/usr/local/lib/python3.9/dist-packages/torch/nn/functional.py:4289: UserWarning: Default grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Please specify align_corners=True if the old behavior is desired.
See the documentation of grid_sample for details.
warnings.warn(
/usr/local/lib/python3.9/dist-packages/torch/nn/functional.py:4227: UserWarning: Default grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Please specify align_corners=True if the old behavior is desired.
See the documentation of grid_sample for details.
warnings.warn(
Sad
Accuracy of the images: 60 %