

DAFTAR PUSTAKA

- [1] Zainal, N. I., Sidek, K. A., Gunawan, T. S., Manser, H., & Kartiwi, M. (2014, January 23). Design and development of portable classroom attendance system based on Arduino and fingerprint biometric. 2014 the 5th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2014. <https://doi.org/10.1109/ICT4M.2014.7020601>
- [2] Pamungkas, Damar Putra, Fajar Rohman Hariri, "Implementasi Metode PCA dan City Block Distance untuk Presensi Mahasiswa Berbasis Wajah," Seminar Nasional Teknologi Informasi, Komunikasi, dan Aplikasinya (SNATIKA) Vol. 04, 2017, 190-194. <http://jurnal.stiki.ac.id/SNATIKA/article/view/126>
- [3] Sunaryono, D., Siswanto, J., & Anggoro, R. (2021). An android based course attendance system using face recognition. *Journal of King Saud University - Computer and Information Science*
- [4] Artotomo, W., Vekky, V., Repi, R., & Azman, N. (2014). Perancangan Sistem Identifikasi Wajah dengan Teknik Feature-Based pada Aplikasi Safety Box. *Jurnal Ilmiah GIGA*, 17(2), 55–62.
- [5] Singh, S., & Prasad, S. V. A. V. (2018). Techniques and challenges of face recognition: A critical review. *Procedia Computer Science*, 143, 536–543. <https://doi.org/10.1016/j.procs.2018.10.427>
- [6] Indra, E., Diarmansyah Batubara, M., Yasir, M., & Chau, S. (2019). Desain dan Implementasi Sistem Absensi Mahasiswa Berdasarkan Fitur Pengenalan Wajah dengan Menggunakan Metode Haar-Like Feature. *Jurnal Penelitian Teknik Informatika UNPRI*.
- [7] Rusimamto, P. W., Endryansyah, Anifah, L., Harimurti, R., & Anistyasari, Y. (2021). Implementation of arduino pro mini and ESP32 cam for temperature monitoring on automatic thermogun IoT-based. *Indonesian Journal of Electrical Engineering and Computer Science*

- [8] Putra, A., Susilo, M., Darlis, D., & Nurmantris, D. A. (2021). PENGENALAN WAJAH BERBASIS ESP32-CAM UNTUK SISTEM KUNCI SEPEDA MOTOR ESP32-CAM-BASED FACE RECOGNITION FOR MOTORCYCLE LOCK SYSTEM. 8(2), 1091–1103. <https://doi.org/10.25124/jett.v8i2.4199>
- [9] Prasetyo, D., Brin, A., Dani, P., & Adi, P. (2021). Performance evaluation of ESP32 Camera Face recognition for various projects Real-time and continuous output power monitoring of photovoltaic (PV) systems View project Analyzes of Radio Frequency and QoS LoRa 915 MHz (920-928 MHz) japan View project Performance evaluation of ESP32 Camera Face recognition for various projects. 02, 1. <https://doi.org/10.31763/iota.v2i1.512>
- [10] Viola, P., Jones, M.J. (2004). Robust real-time face detection. *Int. J. Computer Vision* 57
- [11] Tyagi, Mrinal. “Viola Jones Algorithm and Haar Cascade Classifier.” Medium, 16 July 2021, <https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8>.
- [12] Bhosale, S., Dhanashri, G., & Namrata, M. (2018). Face Detection and Recognition Techniques: A Survey. *IJARCCCE*, 7(11), 207–215. <https://doi.org/10.17148/ijarccce.2018.71145>
- [13] how2electronics.com, “DIY Surveillance CCTV Camera using ESP32 CAM & Blynk”. 16 September 2022, <https://how2electronics.com/diy-surveillance-cctv-camera-using-esp32-cam-blynk/>
- [14] randomnerdtutorials.com, “ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained | Random Nerd Tutorials”. 10 March 2020, <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>.
- [15] Febryan, A., & Teknik Elektro, J. (n.d.). RANCANG BANGUN SISTEM KEAMANAN RUMAH BERBASIS TELEGRAM MENGGUNAKAN ESP 32 CAM. 15(1), 2023.

- [16] arducam.com, "Arducam OV2640 Camera Module, 2MP Mini CCM Compact Camera Modules Compatible with Arduino ESP32 ESP8266 Development Board with DVP 24 Pin Interface." Arducam, https://www.arducam.com/product/arducam-ov2640-camera-module-2mp-mini-ccm-compact-camera-modules-compatible-with-arduino_m0031esp32-esp8266-development-board-with-dvp-24-pin-interface_/
- [17] Li, Stan Z., Jain, Anil K.. Institute of Automation, Center Biometrics Research & Security. (2011). Handbook of Face recognition. second edition.
- [18] Prathapagiri, D., & Eethamakula, K. (2021). Wi-Fi Door Lock System Using ESP32 CAM Based on IoT.
- [19] Ariansyah, W., Ilham, D. N., Khairuman, K., & Candra, R. A. (2021). Opening Doors Using Internet of Things (IoT) Based Face recognition. Brilliance: Research of Artificial Intelligence, 1(2). <https://doi.org/10.47709/brilliance.v1i2.1095>
- [20] hallroad.org, ESP32 CAM MB WIFI Bluetooth Development Board MICRO-USB Interface CH340G USB to Serial Port and OV2640 Camera Module In Pakistan. <https://hallroad.org/esp32-cam-ch340g-wifi-bluetooth-development-board-and-ov2640-camera-module-in-pakistan.html>
- [21] Viola, P., & Jones, M. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features*.
- [22] Setiawan, W., Jurusan Teknik Elektro Dan Komputer, M., Teknik Universitas Udayana, F., & Pengajar Jurusan Teknik Elektro Dan Komputer, S. (2016). *Sistem Pengenalan Wajah Dengan Metode Face Features* (Vol. 3, Issue 2).
- [23] Muhammad, K., & al Banjari, A. (2020). PENGOLAHAN CITRA DIGITAL DAN HISTOGRAM DENGAN PHYTON DAN TEXT EDITOR PHYCHARM. In *Technologia* (Vol. 11, Issue 3).
- [24] Septyanto, M. W., Sofyan, H., Jayadianti, H., Samuel, O., Boedi, D., Teknik, J., Fakultas, I., Industri, T., Veteran, U. ", Yogyakarta, ", Babarsari, J., &

- Yogyakarta, T. (2019). APLIKASI PRESENSI PENGENALAN WAJAH DENGAN MENGGUNAKAN ALGORITMA HAAR CASCADE CLASSIFIER. In *TELEMATIKA* (Vol. 16, Issue 2).
- [25] Gonzalez, R. C., Woods, R. E., & Prentice Hall, P. (2008). Digital Image Processing Third Edition Pearson International Edition prepared by Pearson Education.
- [26] Raheja, S., & Kumar, A. (2021). Edge detection based on type-1 fuzzy logic and guided smoothing. *Evolving Systems*, 12(2), 447–462. <https://doi.org/10.1007/s12530-019-09304-6>
- [27] pjm.ppu.edu, “Image deblurring by periodic and Neumann boundary conditions in a wavelet domain”. https://pjm.ppu.edu/sites/default/files/papers/PJM_May_2022_617_to_629.pdf
- [28] Reddy, A. P., & Riyajuddin, K. (2022). Image deblurring by periodic and Neumann boundary conditions in a wavelet domain. In *Palestine Journal of Mathematics* (Vol. 11, Issue 2).
- [29] researchgate.net, “Compression of Boat. (a) The original image. (b), (c) The image is compressed using the adaptive SFQ with directionlets at the bit rates 0.10 and 0.15 bpp”. https://www.researchgate.net/figure/Compression-of-Boat-a-The-original-image-b-c-The-image-is-compressed-using-the_fig12_6233005
- [30] researchgate.net, “Demonstration of image sharpening (Lena)”. https://www.researchgate.net/figure/Demonstration-of-image-sharpening-Lena_fig12_336386971
- [31] Velisavljević, V., Beferull-Lozano, B., & Vetterli, M. (2007). Space-frequency quantization for image compression with directionlets. *IEEE Transactions on Image Processing*, 16(7), 1761–1773. <https://doi.org/10.1109/TIP.2007.899183>

- [32] Hasugian, A. H., & Zufria, I. (2018). Perancangan Sistem Restorasi Citra Dengan Metode Image Inpainting. ALGORITMA: Jurnal Ilmu Komputer Dan Informatika, 1.
- [33] ankurm.com, “Implementing edge detection in matlab”.
<https://ankurm.com/implementing-edge-detection-in-matlab/?shared=email&msg=fail>
- [34] Bayu Prasetio, A., & Yudi Sobari, M. (n.d.). IMPLEMENTASI SEGMENTASI CITRA DENGAN METODE THRESHOLD PADA PENGOLAHAN CITRA DIGITAL TANAMAN BERACUN DI INDONESIA.
- [35] Tinaliah, Elizabeth, T. (2019). Analisis Hasil Resolusi Citra Dengan Metode Interpolasi Nearest Neighbor, Interpolasi Bilinear, dan Interpolasi Bicubic. JURNAL SISTEM & TEKNOLOGI INFORMASI KOMUNIKASI 1
- [36] Gautam Krishna, C., Adusumilli, T., Puripandla, V., Vemulapalli, R., Venkata Krishna Chaitanya, P., & Engineering Department, E. (2013). Analysis of Adaptive Algorithms. IJSRD-International Journal for Scientific Research & Development, 1(8). www.ijrd.com



LAMPIRAN

A. Coding C++Arduino IDE

```
#include <WebServer.h>
```

```
#include <WiFi.h>
```

```
#include <esp32cam.h>
```

```
const char* WIFI_SSID = "rapap";
```

```
const char* WIFI_PASS = "rapap123";
```

```
WebServer server(80);
```

```
static auto loRes = esp32cam::Resolution::find(320, 240);
```

```
static auto midRes = esp32cam::Resolution::find(350, 530);
```

```
static auto hiRes = esp32cam::Resolution::find(800, 600);
```

```
void serveJpg()
```

```
{
```

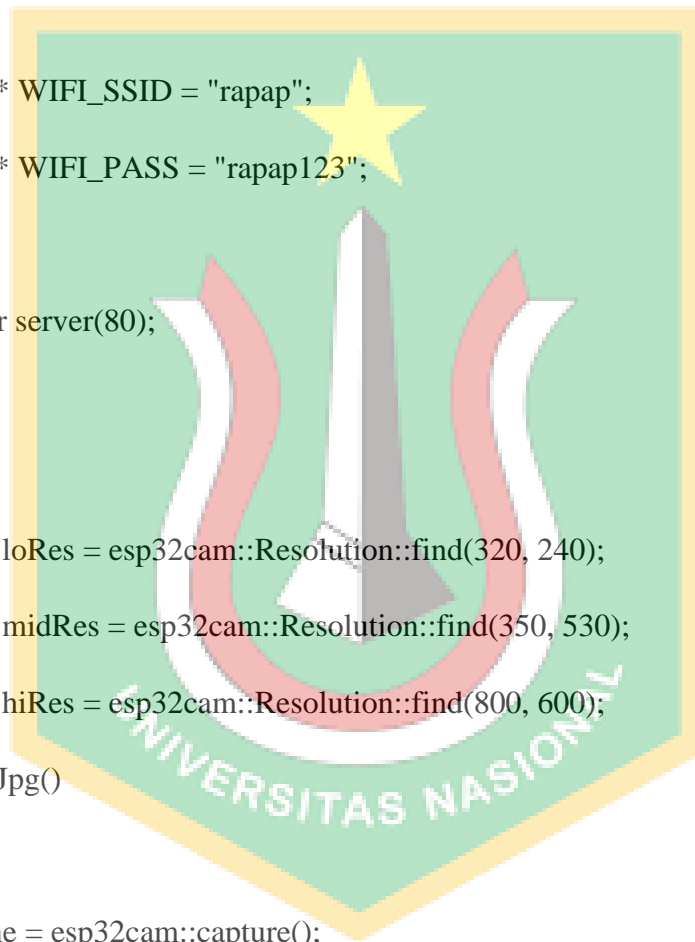
```
    auto frame = esp32cam::capture();
```

```
    if (frame == nullptr) {
```

```
        Serial.println("CAPTURE FAIL");
```

```
        server.send(503, "", "");
```

```
        return;
```



```

}

Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame-
>getHeight(),

    static_cast<int>(frame->size()));

server.setContentLength(frame->size());

server.send(200, "image/jpeg");
WiFiClient client = server.client();
frame->writeTo(client);
}

void handleJpgLo()
{
if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
}
serveJpg();
}

void handleJpgHi()
{
if (!esp32cam::Camera.changeResolution(hiRes)) {
    Serial.println("SET-HI-RES FAIL");
}
}

```



```

serveJpg();
}

void handleJpgMid()
{
  if (!esp32cam::Camera.changeResolution(midRes)) {
    Serial.println("SET-MID-RES FAIL");
  }
  serveJpg();
}

void setup(){
  Serial.begin(115200);
  Serial.println();
  {
    using namespace esp32cam;

    Config cfg;

    cfg.setPins(pins::AiThinker);

    cfg.setResolution(hiRes);

    cfg.setBufferCount(2);

    cfg.setJpeg(80);

```




```

bool ok = Camera.begin(cfg);

    Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");

}

WiFi.persistent(false);

WiFi.mode(WIFI_STA);

WiFi.begin(WIFI_SSID, WIFI_PASS);

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

}

Serial.print("http://");

Serial.println(WiFi.localIP());

Serial.println(" /cam-lo.jpg");

Serial.println(" /cam-hi.jpg");

Serial.println(" /cam-mid.jpg");

server.on("/cam-lo.jpg", handleJpgLo);

server.on("/cam-hi.jpg", handleJpgHi);

server.on("/cam-mid.jpg", handleJpgMid);

server.begin();

}

void loop()

```



```
{
    server.handleClient();
}
```

B. Coding Python Visual Studio

```
import pandas as pd
import cv2
import urllib.request
import numpy as np
import os
from datetime import datetime
import face_recognition

path = r'C:\Attendance\Attendance\image_folder'
url='http://192.168.29.230/cam-hi.jpg'
##"cam.bmp / cam-lo.jpg / cam-hi.jpg / cam.mjpeg ""

if 'Attendance.csv' in os.listdir(os.path.join(os.getcwd(),'Attendance')):
    print("there iss..")
    os.remove("Attendance.csv")
else:
    df=pd.DataFrame(list())
    df.to_csv("Attendance.csv")

images = []
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
```

```
images.append(curImg)
    classNames.append(os.path.splitext(c1)[0])
print(classNames)
```

```
def findEncodings(images):
```

```
    encodeList = []
```

```
    for img in images:
```

```
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
        encode = face_recognition.face_encodings(img)[0]
```

```
        encodeList.append(encode)
```

```
    return encodeList
```

```
def markAttendance(name):
```

```
    with open("Attendance.csv", 'r+') as f:
```

```
        myDataList = f.readlines()
```

```
        nameList = []
```

```
        for line in myDataList:
```

```
            entry = line.split(',')
```

```
            nameList.append(entry[0])
```

```
        if name not in nameList:
```

```
            now = datetime.now()
```

```
            dtString = now.strftime('%H:%M:%S')
```

```
            f.writelines(f'\n{name},{dtString}')
```

```
encodeListKnown = findEncodings(images)
```

```
print('Encoding Complete')
```

```
#cap = cv2.VideoCapture(0)
```

```

while True:
    #success, img = cap.read()
    img_resp=urllib.request.urlopen(url)
    imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)
    img=cv2.imdecode(imgnp,-1)
# img = captureScreen()
    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS,
facesCurFrame)

    for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown,
encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown,
encodeFace)
# print(faceDis)
        matchIndex = np.argmin(faceDis)
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
# print(name)
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1 + 6, y2 - 6),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
            markAttendance(name)

```

```
cv2.imshow('Webcam', img)
key=cv2.waitKey(5)
if key==ord('q'):
    break
cv2.destroyAllWindows()
cv2.imread
```



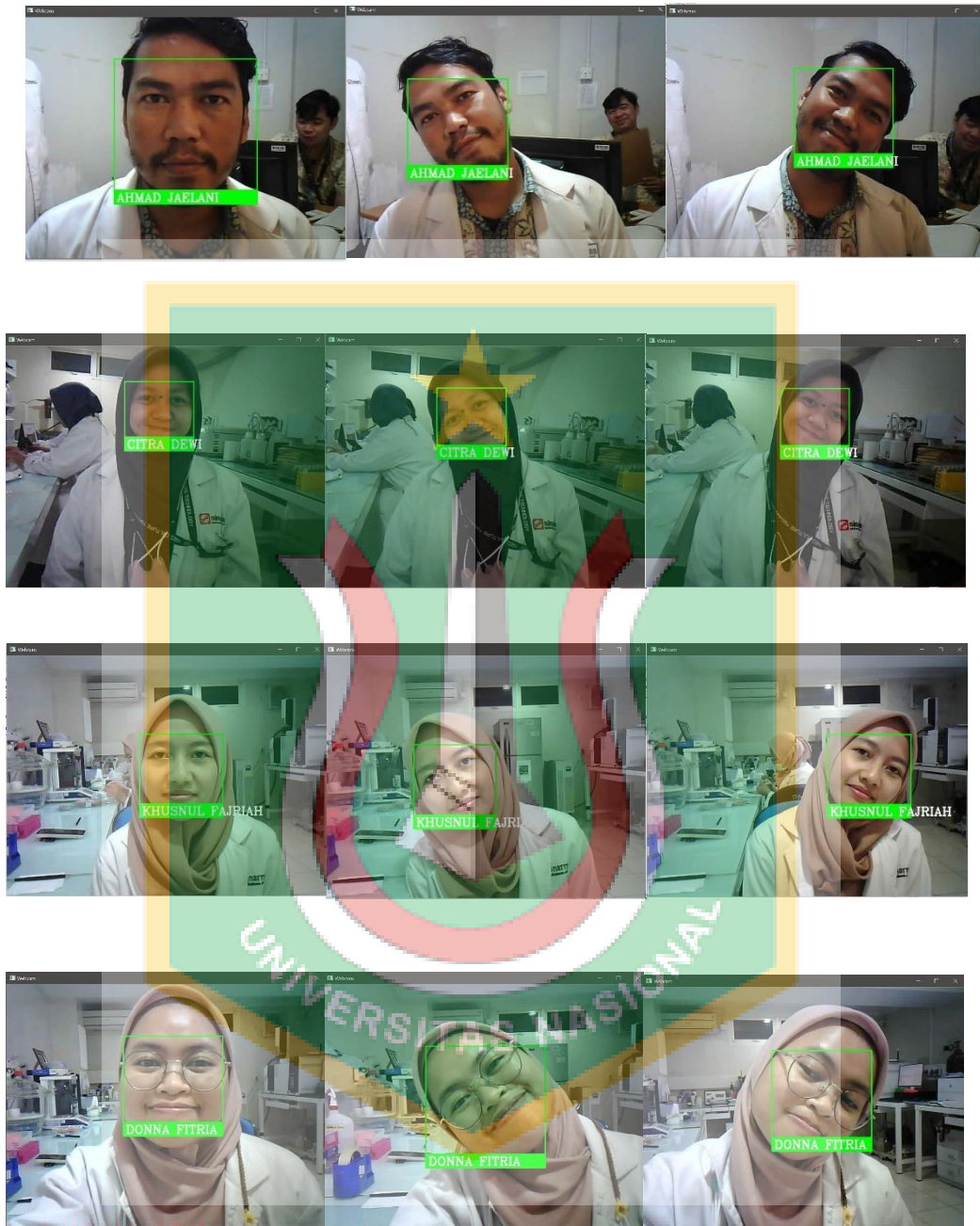
C. Tabel waktu pengenalan wajah

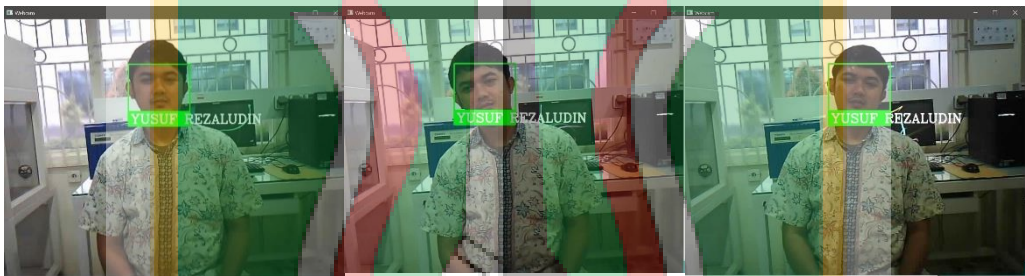
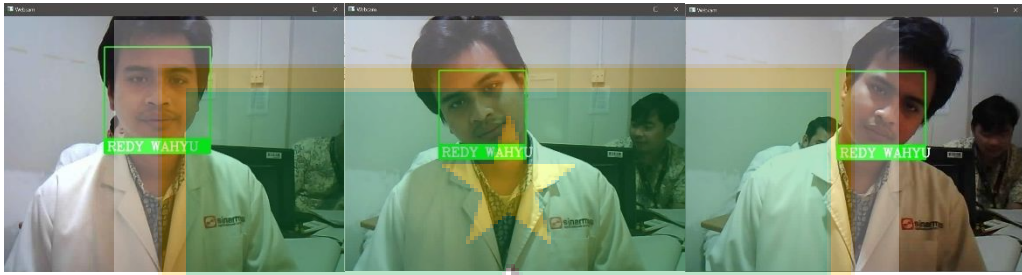
Tabel Waktu Pengenalan Wajah	
Ulangan	Waktu Pengukuran (s)
1	2.17
2	2.3
3	2.45
4	2.19
5	2.26
6	2.05
7	2.51
8	2.13
9	2.41
10	2.34
11	2.12
12	2.18
13	2.07
14	2.29
15	2.28
Total	33.75
Rata-rata	2.25

D. Tabel waktu pengenalan sidik jari

Tabel Waktu Absensi Fingerprint	
Ulangan	Waktu Pengukuran (s)
1	3.13
2	3.45
3	3.58
4	3.15
5	3.24
6	3.52
7	3.23
8	3.16
9	3.62
10	3.11
11	3.42
12	3.37
13	3.02
14	3.41
15	3.37
Total	49.78
Rata-rata	3.32

E. Hasil pembacaan wajah oleh alat.





laporan TA_REIANSYAH

ORIGINALITY REPORT

29%
SIMILARITY INDEX

28%
INTERNET SOURCES

6%
PUBLICATIONS

17%
STUDENT PAPERS

PRIMARY SOURCES

1	www.mathaelectronics.com Internet Source	3%
2	Submitted to Universitas Nasional Student Paper	3%
3	stackoverflow.com Internet Source	2%
4	ejournal.unesa.ac.id Internet Source	1%
5	www.trivusi.web.id Internet Source	1%
6	repository.uin-suska.ac.id Internet Source	1%
7	www.researchgate.net Internet Source	1%
8	journals.telkomuniversity.ac.id Internet Source	1%
9	pt.scribd.com Internet Source	<1%