

LAMPIRAN

Lampiran 1 Surat Kesediaan Membimbing TA Pembimbing I

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Ida Afriliana, ST., M.kom.
NIDN : 0624047703
NIPY : 12.013.168
Jabatan Struktural : Dosen Tetap
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Anwar Fuad
NIM : 21041071
Program Studi : D3 Teknik Komputer
Judul TA : SISTEM ABSENSI KARYAWAN GUDANG
BERBASIS *FACE RECOGNITION*

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui
Ka. Prodi DIII Teknik Komputer,



Ida Afriliana, ST, M.Kom
NIPY. 12.013.168

Tegal, 2024

Dosen Pembimbing I,

Ida Afriliana, ST, M.Kom
NIPY. 12.013.168

Lampiran 2 Surat Kesiediaan Membimbing TA Pembimbing II

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Abdul Basit, S.Kom., MT
NIDN : 0608129106
NIPY : 01.015.198
Jabatan Struktural : Dosen Tetap
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Anwar Fuad
NIM : 21041071
Program Studi : D3 Teknik Komputer
Judul TA : SISTEM ABSENSI KARYAWAN GUDANG
BERBASIS *FACE RECOGNITION*

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui
Ka Prodi DIII Teknik Komputer,



Ida Airliana, ST, M.Kom
NIPY. 12.013.168

Tegal, 2024

Dosen Pembimbing II,

Abdul Basit, S.Kom., MT
NIPY. 01.015.198

Lampiran 3 Surat Observasi



POLITEKNIK HARAPAN BERSAMA
The True Vocational Campus

D-3 Teknik Komputer

No. : 046.03/KMP.PHB/V/2024
Lampiran : -
Perihal : Permohonan Izin Observasi Tugas Akhir (TA)

Kepada Yth.
Kepala CV.GUNA JAYA
Debong Wetan, Kec. Dukuhturi, Kabupaten Tegal, Jawa Tengah

Dengan Hormat,

Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester VI (Genap) Program Studi D III Teknik Komputer Politeknik Harapan Bersama Tegal, Maka dengan ini kami mengajukan izin observasi pengambilan data di CV.GUNA JAYA yang Bapak / Ibu Pimpin, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:

No.	NIM	Nama	No. HP
1	21041071	ANWAR.FUAD	089524173146
2	21041075	HIMMATUL ULYA	0895359543859

Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.

Tegal, 27 Mei 2024

Kap. Prodi DIII Teknik Komputer
Politeknik Harapan Bersama Tegal



Ida Airliana, ST, M.Kom
NIP. 12.013.168

Jl. Mataram No. 9 Kota Tegal 52143, Jawa Tengah, Indonesia.

(0283)352000

komputer@poltektegal.ac.id

poltektegal.ac.id

Lampiran 4 Source Code

```
#include <ArduinoWebsockets.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <time.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "Base64.h"
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "camera_index.h"
#include "Arduino.h"
#include "fd_forward.h"
#include "fr_forward.h"
#include "fr_flash.h"
#include "fb_gfx.h"

const char* ssid = "Puad";
const char* password = "katasandi";

#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7
#define FLASH_LED_PIN 4

#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"

using namespace websockets;
WebsocketsServer socket_server;

camera_fb_t *fb = NULL;

long current_millis;
long last_detected_millis = 0;

unsigned long absensi_berhasil_millis = 0;
long interval = 5000; // open lock for ... milliseconds
bool face_recognised = false;
bool flashState = false;

void app_facenet_main();
void app_httpserver_init();
void send_attendance_data(const char *face_id, const uint8_t
*image_data, size_t image_length);

typedef struct {
    uint8_t *image;
    box_array_t *net_boxes;
    dl_matrix3d_t *face_id;
} http_img_process_result;

static inline mtmn_config_t app_mtmn_config() {
```

```

mtmn_config_t mtmn_config = {0};
mtmn_config.type = FAST;
mtmn_config.min_face = 80;
mtmn_config.pyramid = 0.707;
mtmn_config.pyramid_times = 4;
mtmn_config.p_threshold.score = 0.6;
mtmn_config.p_threshold.nms = 0.7;
mtmn_config.p_threshold.candidate_number = 20;
mtmn_config.r_threshold.score = 0.7;
mtmn_config.r_threshold.nms = 0.7;
mtmn_config.r_threshold.candidate_number = 10;
mtmn_config.o_threshold.score = 0.7;
mtmn_config.o_threshold.nms = 0.7;
mtmn_config.o_threshold.candidate_number = 1;
return mtmn_config;
}
mtmn_config_t mtmn_config = app_mtmn_config();

face_id_name_list st_face_list;
static dl_matrix3du_t *aligned_face = NULL;

httpd_handle_t camera_httpd = NULL;

typedef enum {
    START_STREAM,
    START_DETECT,
    SHOW_FACES,
    START_RECOGNITION,
    START_ENROLL,
    ENROLL_COMPLETE,
    DELETE_ALL,
} en_fsm_state;
en_fsm_state g_state;

typedef struct {
    char enroll_name[ENROLL_NAME_LEN];
} httpd_resp_value;

httpd_resp_value st_name;

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;

```

```

config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

pinMode(FLASH_LED_PIN, OUTPUT);
digitalWrite(FLASH_LED_PIN, LOW);

#if defined(CAMERA_MODEL_ESP_EYE)
    pinMode(13, INPUT_PULLUP);
    pinMode(14, INPUT_PULLUP);
#endif

esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t *s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE)
    s->set_vflip(s, 1);
    s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

app_httpserver_init();
app_facenet_main();
socket_server.listen(82);

```

```

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");

// Inisialisasi NTP
configTime(0, 0, "pool.ntp.org"); // Waktu UTC
setenv("TZ", "WIB-7", 1); // Zona waktu untuk WIB (GMT+7)
}

static esp_err_t index_handler(httpd_req_t *req) {
    httpd_resp_set_type(req, "text/html");
    httpd_resp_set_hdr(req, "Content-Encoding", "gzip");
    return httpd_resp_send(req, (const char *)index_ov2640_html_gz,
index_ov2640_html_gz_len);
}

httpd_uri_t index_uri = {
    .uri      = "/",
    .method   = HTTP_GET,
    .handler  = index_handler,
    .user_ctx = NULL
};

void app_httpserver_init () {
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    if (httpd_start(&camera_httpd, &config) == ESP_OK)
        Serial.println("httpd_start");
    {
        httpd_register_uri_handler(camera_httpd, &index_uri);
    }
}

String getFormattedDate() {
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) {
        Serial.println("Failed to obtain time");
        return "";
    }

    char date[11]; // Format: YYYY-MM-DD
    strftime(date, 11, "%Y-%m-%d", &timeinfo);
    return String(date);
}

String getFormattedTime() {
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) {
        Serial.println("Failed to obtain time");
        return "";
    }

    char time[9]; // Format: HH:MM:SS
    strftime(time, 9, "%H:%M:%S", &timeinfo);
    return String(time);
}

String urlencode(String str) {
    String encodedString = "";

```



```

char c;
char code0;
char code1;
char code2;

for (int i = 0; i < str.length(); i++) {
    c = str.charAt(i);
    if (c == ' ') {
        encodedString += '+';
    } else if (isalnum(c)) {
        encodedString += c;
    } else {
        code1 = (c & 0xf) + '0';
        if ((c & 0xf) > 9) {
            code1 = (c & 0xf) - 10 + 'A';
        }
        c = (c >> 4) & 0xf;
        code0 = c + '0';
        if (c > 9) {
            code0 = c - 10 + 'A';
        }
        code2 = '\\0';
        encodedString += '%';
        encodedString += code0;
        encodedString += code1;
    }
    yield();
}
return encodedString;
}

void send_attendance_data(const char *face_id, const uint8_t
*image_data, size_t image_length) {
    WiFiClient client;
    HTTPClient http;

    String postData = "face_id=";
    postData += String(face_id);
    postData += "&image=data:image/jpeg;base64,";
    String checkindate = getFormattedDate(); // Tanggal saat ini
    String checkintime = getFormattedTime(); // Waktu saat ini

    // Mengonversi gambar ke Base64
    String base64Image = base64::encode(image_data, image_length);
    postData += urlencode(base64Image);
    postData += "&checkindate=" + checkindate + "&checkintime=" +
checkintime;

    Serial.println("Data yang dikirim:");
    Serial.println(postData);

    http.begin(client, "https://himmatululya.my.id/absensi.php");
    http.addHeader("Content-Type", "application/x-www-form-
urlencoded");

    int httpResponseCode = http.POST(postData);

```

```

if (httpResponseCode > 0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    String payload = http.getString();
    Serial.println(payload);
} else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}

http.end();
}
//untuk menginisialisasi list wajah yang sudah terdaftar
void app_facenet_main() {
    face_id_name_init(&st_face_list, FACE_ID_SAVE_NUMBER,
ENROLL_CONFIRM_TIMES);
    aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT,
3);
    read_face_id_from_flash_with_name(&st_face_list); //membaca list
wajah dari penyimpanan flash
}
//proses pendaftaran wajah baru
static inline int do_enrollment(face_id_name_list *face_list,
dl_matrix3d_t *new_id) {
    ESP_LOGD(TAG, "START ENROLLING");
    int left_sample_face =
enroll_face_id_to_flash_with_name(face_list, new_id,
st_name.enroll_name);
    ESP_LOGD(TAG, "Face ID %s Enrollment: Sample %d",
st_name.enroll_name, ENROLL_CONFIRM_TIMES - left_sample_face);
    return left_sample_face;
}

static esp_err_t send_face_list(WebsocketsClient &client) {
    client.send("delete_faces");
    face_id_node *head = st_face_list.head;
    char add_face[64];
    for (int i = 0; i < st_face_list.count; i++) {
        sprintf(add_face, "listface:%s", head->id_name);
        client.send(add_face);
        head = head->next;
    }
}

static esp_err_t delete_all_faces(WebsocketsClient &client) {
    delete_face_all_in_flash_with_name(&st_face_list);
    client.send("delete_faces");
}

void handle_message(WebsocketsClient &client, WebsocketsMessage
msg) {
    if (msg.data() == "stream") {
        g_state = START_STREAM;
        client.send("STREAMING");
    }
    if (msg.data() == "detect") {

```

```

    g_state = START_DETECT;
    client.send("DETECTING");
}
if (msg.data().substring(0, 8) == "capture:") {
    g_state = START_ENROLL;
    char person[FACE_ID_SAVE_NUMBER * ENROLL_NAME_LEN] = {0,};
    msg.data().substring(8).toCharArray(person, sizeof(person));
    memcpy(st_name.enroll_name, person, strlen(person) + 1);
    client.send("CAPTURING");
}
if (msg.data() == "recognise") {
    g_state = START_RECOGNITION;
    client.send("RECOGNISING");
}
if (msg.data().substring(0, 7) == "remove:") {
    char person[ENROLL_NAME_LEN * FACE_ID_SAVE_NUMBER];
    msg.data().substring(7).toCharArray(person, sizeof(person));
    delete_face_id_in_flash_with_name(&st_face_list, person);
    send_face_list(client);
}
if (msg.data() == "delete_all") {
    delete_all_faces(client);
}
}

void absensi_berhasil(WebsocketsClient &client, const char*
id_name) {
    client.send("absensi_berhasil");
    absensi_berhasill_millis = millis();

    fb = esp_camera_fb_get();
    if (fb) {
        delay(10000);
        send_attendance_data(id_name, fb->buf, fb->len);
        esp_camera_fb_return(fb);
    }
    delay(20000);
    client.send("STREAMING");
}

void loop() {
    auto client = socket_server.accept();
    client.onMessage(handle_message);
    dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, 320, 240,
3);
    http_img_process_result out_res = {0};
    out_res.image = image_matrix->item;

    send_face_list(client);
    client.send("STREAMING");

    while (client.available()) {
        client.poll();

        if (millis() - interval > absensi_berhasill_millis) {
            digitalWrite(FLASH_LED_PIN, LOW);

```

```

    }

    fb = esp_camera_fb_get();

    if (g_state == START_DETECT || g_state == START_ENROLL ||
g_state == START_RECOGNITION) {
        digitalWrite(FLASH_LED_PIN, HIGH);
        out_res.net_boxes = NULL;
        //data wajah yang sudah face recognirion disimpan dalam
variabel
        out_res.face_id = NULL;

        fmt2rgb888(fb->buf, fb->len, fb->format, out_res.image);
        out_res.net_boxes = face_detect(image_matrix, &mtmn_config);

        if (out_res.net_boxes) {
            if (align_face(out_res.net_boxes, image_matrix,
aligned_face) == ESP_OK) {
                out_res.face_id = get_face_id(aligned_face);
                last_detected_millis = millis();
                if (g_state == START_DETECT) {
                    client.send("FACE DETECTED");
                }

                if (g_state == START_ENROLL) {
                    int left_sample_face = do_enrollment(&st_face_list,
out_res.face_id);
                    char enrolling_message[64];
                    sprintf(enrolling_message, "SAMPLE NUMBER %d FOR %s",
ENROLL_CONFIRM_TIMES - left_sample_face, st_name.enroll_name);
                    client.send(enrolling_message);
                    if (left_sample_face == 0) {
                        ESP_LOGI(TAG, "Enrolled Face ID: %s",
st_face_list.tail->id_name);
                        g_state = START_STREAM;
                        char captured_message[64];
                        sprintf(captured_message, "FACE CAPTURED FOR %s",
st_face_list.tail->id_name);
                        client.send(captured_message);
                        send_face_list(client);
                    }
                }

                if (g_state == START_RECOGNITION && (st_face_list.count
> 0)) {
                    face_id_node *f =
recognize_face_with_name(&st_face_list, out_res.face_id);
                    if (f) {
                        char recognised_message[64];
                        sprintf(recognised_message, "ABSENSI BERHASIL FOR
%s", f->id_name);
                        absensi_berhasil(client, f->id_name);
                        client.send(recognised_message);
                    } else {
                        client.send("FACE NOT RECOGNISED");
                    }
                }
            }
        }
    }

```

```

        }
        dl_matrix3d_free(out_res.face_id);
    }
    else {
        if (g_state != START_DETECT) {
            client.send("NO FACE DETECTED");
        }
    }
    if (g_state == START_DETECT && millis() -
last_detected_millis > 500) {
        client.send("DETECTING");
    }
}

client.sendBinary((const char *)fb->buf, fb->len);

esp_camera_fb_return(fb);
fb = NULL;
}
}

```

Lampiran 5 Foto Dokumentasi






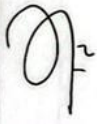
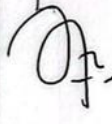

Lampiran 6 Form Pembimbing I

Lampiran 22
Bimbingan Proposal TA

IK P2M PHB d.5.1.e.1

NAMA MAHASISWA :
PEMBIMBING I:
PROPOSAL TA

BIMBINGAN

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	Kamis, 2 Mei 2024	- Revisi bab 1 - acc bab 2 dan 3 -	
1.	Rabu, 21 Februari 2024	- menjelaskan tentang tata cara alur bimbingan yang benar dan jelas	
2.	Jum, 23 Feb 2024	- sistematika dirapikan : semua bahasa asing italic - seora isi sudah benar - antar sub bab spasi	
3.	Rabu, 28 Feb 2024	- Terdapat typo kata - Block diagram dan flowchart - teliti spasi	
4.	Kamis, 9 Mei 2024	- Revisi bab 1 - bab 2 dan 3 acc	
5.	Selasa, 7 Mei 2024	- Acc Bab I	


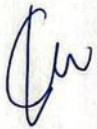

Lampiran 7 Form Pembimbing II

Lampiran 23
Bimbingan Laporan Pembimbing I TA

IK P2M PHB d.5.1.e.1

PEMBIMBING II:
LAPORAN TA

BIMBINGAN

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1	Selasa, 14 Mei 2024	- Block Diagram - Input, Proses, output	
2	Rabu, 15 Mei 2024	- menggunakan variabel - diterkirim ke Database atau menggunakan firebase - ESP32 dan PHP - Database	
	Selasa, 21/05/2024	Ya BAB 4	
	24/05/2024	Ya BAB 5 Ya Project Ya Slap upan 20.05.2024	