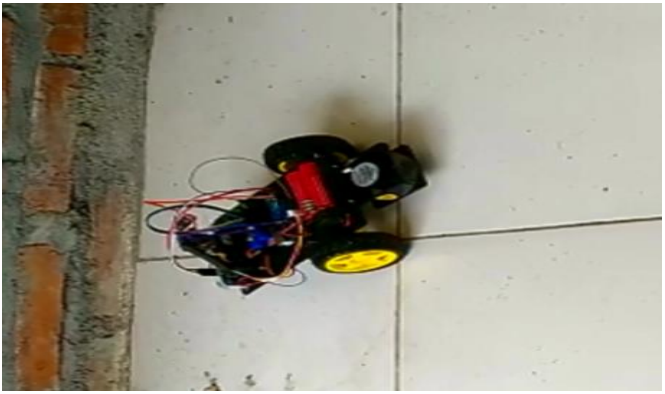
LAMPIRAN

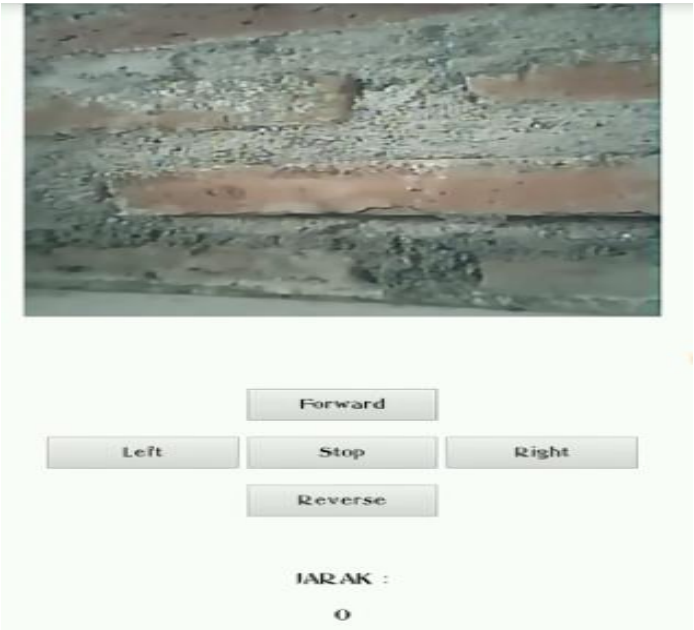
 # TAMPILAN SOFTWARE DAN HARDWARE

a. Pengujian Fan/Kipas 12V

b. Pengujian Sensor Ultrasonik HC-SR04



c. Pengujian Node MCU ESP 32 CAM untuk stream video dan mengcontrol robot pembersih debu.



d. Pengujian Fan/Kipas 12V

 # PROGRAM

a. app_httpd

```
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "camera_index.h"
#include "Arduino.h"

extern int leftmotor1;
extern int leftmotor2;
extern int rightmotor1;
extern int rightmotor2;

int a, b;
String aa;

void terima_serial();

extern String Camerafeed;

void Motor(int L1, int L2, int R1, int R2);
```

```
typedef struct {
  size_t size; //number of values used for filtering
  size_t index; //current value index
  size_t count; //value count
  int sum;
  int * values; //array to be filled with values
} ra_filter_t;

typedef struct {
  httpd_req_t *req;
  size_t len;
} jpg_chunking_t;

#define PART_BOUNDARY "123456789000000000000987654321"
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n";

static ra_filter_t ra_filter;
httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;
```

```
static ra_filter_t * ra_filter_init(ra_filter_t * filter, size_t
sample_size) {
  memset(filter, 0, sizeof(ra_filter_t));

  filter->values = (int *)malloc(sample_size * sizeof(int));
  if (!filter->values) {
    return NULL;
  }
  memset(filter->values, 0, sample_size * sizeof(int));

  filter->size = sample_size;
  return filter;
}

static int ra_filter_run(ra_filter_t * filter, int value) {
  if (!filter->values) {
    return value;
  }
  filter->sum -= filter->values[filter->index];
  filter->values[filter->index] = value;
  filter->sum += filter->values[filter->index];
  filter->index++;
  filter->index = filter->index % filter->size;
```

```
  if (filter->count < filter->size) {
    filter->count++;
  }
  return filter->sum / filter->count;
}

static size_t jpg_encode_stream(void * arg, size_t index, const void*
data, size_t len) {
  jpg_chunking_t *j = (jpg_chunking_t *)arg;
  if (!index) {
    j->len = 0;
  }
  if (httpd_resp_send_chunk(j->req, (const char *)data, len) !=
ESP_OK) {
    return 0;
  }
  j->len += len;
  return len;
}

static esp_err_t capture_handler(httpd_req_t *req) {
  camera_fb_t * fb = NULL;
  esp_err_t res = ESP_OK;
  int64_t fr_start = esp_timer_get_time();
```

```
  fb = esp_camera_fb_get();
  if (!fb) {
    Serial.printf("Camera capture failed");
    httpd_resp_send_500(req);
    return ESP_FAIL;
  }

  httpd_resp_set_type(req, "image/jpeg");
  httpd_resp_set_hdr(req, "Content-Disposition", "inline;
filename=capture.jpg");

  size_t fb_len = 0;
  if (fb->format == PIXFORMAT_JPEG) {
    fb_len = fb->len;
    res = httpd_resp_send(req, (const char *)fb->buf, fb->len);
  } else {
    jpg_chunking_t jchunk = {req, 0};
    res = frame2jpg_cb(fb, 80, jpg_encode_stream, &jchunk) ?
ESP_OK : ESP_FAIL;
    httpd_resp_send_chunk(req, NULL, 0);
    fb_len = jchunk.len;
  }
  esp_camera_fb_return(fb);
```

```
  int64_t fr_end = esp_timer_get_time();
  //  Serial.printf("JPG: %uB %ums", (uint32_t)(fb_len),
(uint32_t)((fr_end - fr_start) / 1000));
  return res;
}

static esp_err_t stream_handler(httpd_req_t *req) {
  camera_fb_t * fb = NULL;
  esp_err_t res = ESP_OK;
  size_t _jpg_buf_len = 0;
  uint8_t * _jpg_buf = NULL;
  char * part_buf[64];

  static int64_t last_frame = 0;
  if (!last_frame) {
    last_frame = esp_timer_get_time();
  }

  res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
  if (res != ESP_OK) {
    return res;
  }

  while (true) {
```

```
    fb = esp_camera_fb_get();
    if (!fb) {
      Serial.printf("Camera capture failed");
      res = ESP_FAIL;
    } else {
      if (fb->format != PIXFORMAT_JPEG) {
        bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf,
&_jpg_buf_len);
        esp_camera_fb_return(fb);
        fb = NULL;
        if (!jpeg_converted) {
          Serial.printf("JPEG compression failed");
          res = ESP_FAIL;
        }
      } else {
        _jpg_buf_len = fb->len;
        _jpg_buf = fb->buf;
      }
    }
    if (res == ESP_OK) {
      size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART,
_jpg_buf_len);
      res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
    }
```

```
    if (res == ESP_OK) {
      res = httpd_resp_send_chunk(req, (const char *)_jpg_buf,
_jpg_buf_len);
    }
    if (res == ESP_OK) {
      res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
    }
    if (fb) {
      esp_camera_fb_return(fb);
      fb = NULL;
      _jpg_buf = NULL;
    } else if (_jpg_buf) {
      free(_jpg_buf);
      _jpg_buf = NULL;
    }
    if (res != ESP_OK) {
      break;
    }
    int64_t fr_end = esp_timer_get_time();

    int64_t frame_time = fr_end - last_frame;
    last_frame = fr_end;
    frame_time /= 1000;
```

```
    uint32_t avg_frame_time = ra_filter_run(&ra_filter,
frame_time);
    //    Serial.printf("MJPG: %uB %ums (%.1ffps), AVG: %ums
(%.1ffps)"
    //                , (uint32_t)(_jpg_buf_len),
    //                (uint32_t)frame_time, 1000.0 / (uint32_t)frame_time,
    //                avg_frame_time, 1000.0 / avg_frame_time
    //               );
  }

  last_frame = 0;
  return res;
}

static esp_err_t cmd_handler(httpd_req_t *req) {
  char*  buf;
  size_t buf_len;
  char variable[32] = {0,};
  char value[32] = {0,};

  buf_len = httpd_req_get_url_query_len(req) + 1;
  if (buf_len > 1) {
    buf = (char*)malloc(buf_len);
    if (!buf) {
```

```
    httpd_resp_send_500(req);
    return ESP_FAIL;
  }
  if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
    if (httpd_query_key_value(buf, "var", variable, sizeof(variable))
== ESP_OK &&
        httpd_query_key_value(buf, "val", value, sizeof(value)) ==
ESP_OK) {
    } else {
      free(buf);
      httpd_resp_send_404(req);
      return ESP_FAIL;
    }
  } else {
    free(buf);
    httpd_resp_send_404(req);
    return ESP_FAIL;
  }
  free(buf);
 } else {
  httpd_resp_send_404(req);
  return ESP_FAIL;
 }
```

```
  int val = atoi(value);
  sensor_t * s = esp_camera_sensor_get();
  int res = 0;

  if (!strcmp(variable, "framesize")) {
    if (s->pixformat == PIXFORMAT_JPEG) res = s->set_framesize(s, (framesize_t)val);
  }
  else if (!strcmp(variable, "quality")) res = s->set_quality(s, val);
  else if (!strcmp(variable, "contrast")) res = s->set_contrast(s, val);
  else if (!strcmp(variable, "brightness")) res = s->set_brightness(s, val);
  else if (!strcmp(variable, "saturation")) res = s->set_saturation(s, val);
  else if (!strcmp(variable, "gainceiling")) res = s->set_gainceiling(s, (gainceiling_t)val);
  else if (!strcmp(variable, "colorbar")) res = s->set_colorbar(s, val);
  else if (!strcmp(variable, "awb")) res = s->set_whitebal(s, val);
  else if (!strcmp(variable, "agc")) res = s->set_gain_ctrl(s, val);
  else if (!strcmp(variable, "aec")) res = s->set_exposure_ctrl(s, val);
  else if (!strcmp(variable, "hmirror")) res = s->set_hmirror(s, val);
  else if (!strcmp(variable, "vflip")) res = s->set_vflip(s, val);
  else if (!strcmp(variable, "awb_gain")) res = s->set_awb_gain(s, val);
  else if (!strcmp(variable, "agc_gain")) res = s->set_agc_gain(s, val);
```

```
  else if (!strcmp(variable, "aec_value")) res = s->set_aec_value(s,
val);

  else if (!strcmp(variable, "aec2")) res = s->set_aec2(s, val);

  else if (!strcmp(variable, "dcw")) res = s->set_dcw(s, val);

  else if (!strcmp(variable, "bpc")) res = s->set_bpc(s, val);

  else if (!strcmp(variable, "wpc")) res = s->set_wpc(s, val);

  else if (!strcmp(variable, "raw_gma")) res = s->set_raw_gma(s,
val);

  else if (!strcmp(variable, "lenc")) res = s->set_lenc(s, val);

  else if (!strcmp(variable, "special_effect")) res = s-
>set_special_effect(s, val);

  else if (!strcmp(variable, "wb_mode")) res = s->set_wb_mode(s,
val);

  else if (!strcmp(variable, "ae_level")) res = s->set_ae_level(s, val);

  else {

    res = -1;

  }


  if (res) {

    return httpd_resp_send_500(req);

  }


  httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");

  return httpd_resp_send(req, NULL, 0);

}
```

```
static esp_err_t status_handler(httpd_req_t *req) {
  static char json_response[1024];

  sensor_t * s = esp_camera_sensor_get();
  char * p = json_response;
  *p++ = '{';

  p += sprintf(p, "\"framesize\":%u,", s->status.framesize);
  p += sprintf(p, "\"quality\":%u,", s->status.quality);
  p += sprintf(p, "\"brightness\":%d,", s->status.brightness);
  p += sprintf(p, "\"contrast\":%d,", s->status.contrast);
  p += sprintf(p, "\"saturation\":%d,", s->status.saturation);
  p += sprintf(p, "\"special_effect\":%u,", s->status.special_effect);
  p += sprintf(p, "\"wb_mode\":%u,", s->status.wb_mode);
  p += sprintf(p, "\"awb\":%u,", s->status.awb);
  p += sprintf(p, "\"awb_gain\":%u,", s->status.awb_gain);
  p += sprintf(p, "\"aec\":%u,", s->status.aec);
  p += sprintf(p, "\"aec2\":%u,", s->status.aec2);
  p += sprintf(p, "\"ae_level\":%d,", s->status.ae_level);
  p += sprintf(p, "\"aec_value\":%u,", s->status.aec_value);
  p += sprintf(p, "\"agc\":%u,", s->status.agc);
  p += sprintf(p, "\"agc_gain\":%u,", s->status.agc_gain);
```

```
  p += sprintf(p, "\"gainceiling\":%u,", s->status.gainceiling);
  p += sprintf(p, "\"bpc\":%u,", s->status.bpc);
  p += sprintf(p, "\"wpc\":%u,", s->status.wpc);
  p += sprintf(p, "\"raw_gma\":%u,", s->status.raw_gma);
  p += sprintf(p, "\"lenc\":%u,", s->status.lenc);
  p += sprintf(p, "\"hmirror\":%u,", s->status.hmirror);
  p += sprintf(p, "\"dcw\":%u,", s->status.dcw);
  p += sprintf(p, "\"colorbar\":%u", s->status.colorbar);
  *p++ = '}';
  *p++ = 0;
  httpd_resp_set_type(req, "application/json");
  httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
  return httpd_resp_send(req, json_response, strlen(json_response));
}

static esp_err_t index_handler(httpd_req_t *req) {
  httpd_resp_set_type(req, "text/html");
  String page = "";
  //  page += "<TITLE>Surveillance Robot</TITLE>";
  //  page += "<H1>Surveillance Robot Using ESP32</H1>";

  page += "<TITLE> <p align=center> ...test... <p align=center>
</TITLE>";
```

```
  page += "<meta name=\"viewport\" content=\"width=device-
width, initial-scale=0.8, maximum-scale=0.8, user-scalable=0\">\n";

  page += "<script>var xhttp = new XMLHttpRequest();</script>";

  page += "<script>function getsend(arg) { xhttp.open('GET', arg
+'?' + new Date().getTime(), true); xhttp.send() } </script>";


  page += "<p align=center><IMG SRC='http://" + Camerafeed +
":81/stream' style='width:300px; height:300px;
transform:rotate(90deg);'></p><br/><br/>";


  page += "<p align=center> <button style=width:90px;height:30px;
onmousedown=getsend('forward')  ontouchstart=getsend('forward')
>Forward</button> </p>";

  page += "<p align=center>";

  page += "<button style=width:90px;height:30px
onmousedown=getsend('leftturn')  ontouchstart=getsend('leftturn')
>Left</button> ";

  page += "<button style=width:90px;height:30px
onmousedown=getsend('stop')
onmouseup=getsend('stop')>Stop</button> ";

  page += "<button style=width:90px;height:30px
onmousedown=getsend('rightturn')
ontouchstart=getsend('rightturn') >Right</button>";

  page += "</p>";

  page += "<p align=center><button style=width:90px;height:30px
onmousedown=getsend('reverse')  ontouchstart=getsend('reverse')
>Reverse</button></p>";

  page += "</p><br/>";
```

```
  page += "<p align=center>JARAK : </p>";
  page += "<p align=center>" + String(a) + " </p>";

  page += "</p>";

  return httpd_resp_send(req, &page[0], strlen(&page[0]));
}

static esp_err_t forward_handler(httpd_req_t *req)
{
  //  if (a > 0 && a < 15)
  //  {
  //    Serial.println("stop ultrasonik.....");
  //    Motor(LOW, LOW, LOW, LOW);
  //  }
  //  else
  //    Motor(LOW, HIGH, HIGH, LOW);
  //  Serial.println("Forward");
  //delay(4000);
  Serial.println("se1");

  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, "OK", 2);
```

```
}

static esp_err_t reverse_handler(httpd_req_t *req)
{
  //  Motor(HIGH, LOW, LOW, HIGH);
  //  Serial.println("Reverse");
  Serial.println("se2");
  // delay(4000);
  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, "OK", 2);
}

static esp_err_t leftturn_handler(httpd_req_t *req)
{
  //  Motor(HIGH, LOW, HIGH, LOW);
  //  Serial.println("Left");
  Serial.println("se3");
  //delay(6000);
  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, "OK", 2);
}

static esp_err_t rightturn_handler(httpd_req_t *req)
```

```
{
  //  Motor(LOW, HIGH, LOW, HIGH);
  //  Serial.println("Right");
  Serial.println("se4");
  //delay(6000);
  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, "OK", 2);
}

static esp_err_t stop_handler(httpd_req_t *req)
{
  //  Motor(LOW, LOW, LOW, LOW);
  //  Serial.println("Stop");
  Serial.println("se0");
  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, "OK", 2);
}

void startCameraServer() {
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();

  httpd_uri_t forward_uri = {
    .uri       = "/forward",
```

```
  .method    = HTTP_GET,
  .handler   = forward_handler,
  .user_ctx  = NULL
};

httpd_uri_t reverse_uri = {
  .uri       = "/reverse",
  .method    = HTTP_GET,
  .handler   = reverse_handler,
  .user_ctx  = NULL
};

httpd_uri_t stop_uri = {
  .uri       = "/stop",
  .method    = HTTP_GET,
  .handler   = stop_handler,
  .user_ctx  = NULL
};

httpd_uri_t leftturn_uri = {
  .uri       = "/leftturn",
  .method    = HTTP_GET,
  .handler   = leftturn_handler,
```

```
  .user_ctx  = NULL
};

httpd_uri_t rightturn_uri = {
  .uri       = "/rightturn",
  .method    = HTTP_GET,
  .handler   = rightturn_handler,
  .user_ctx  = NULL
};


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

httpd_uri_t status_uri = {
  .uri       = "/status",
  .method    = HTTP_GET,
  .handler   = status_handler,
  .user_ctx  = NULL
```

```
};

httpd_uri_t cmd_uri = {
  .uri       = "/control",
  .method    = HTTP_GET,
  .handler   = cmd_handler,
  .user_ctx  = NULL
};

httpd_uri_t capture_uri = {
  .uri       = "/capture",
  .method    = HTTP_GET,
  .handler   = capture_handler,
  .user_ctx  = NULL
};

httpd_uri_t stream_uri = {
  .uri       = "/stream",
  .method    = HTTP_GET,
  .handler   = stream_handler,
  .user_ctx  = NULL
};
```

```
  ra_filter_init(&ra_filter, 20);
  Serial.printf("Starting web server on port: ' % d'", config.server_port);
  if (httpd_start(&camera_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(camera_httpd, &index_uri);
    httpd_register_uri_handler(camera_httpd, &forward_uri);
    httpd_register_uri_handler(camera_httpd, &reverse_uri);
    httpd_register_uri_handler(camera_httpd, &stop_uri);
    httpd_register_uri_handler(camera_httpd, &leftturn_uri);
    httpd_register_uri_handler(camera_httpd, &rightturn_uri);

  }

  config.server_port += 1;
  config.ctrl_port += 1;
  Serial.printf("Starting stream server on port: ' % d'", config.server_port);
  if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &stream_uri);
  }
}

void Motor(int L1, int L2, int R1, int R2)
```

```
{
  digitalWrite(leftmotor1,  L1);
  digitalWrite(leftmotor2,  L2);
  digitalWrite(rightmotor1, R1);
  digitalWrite(rightmotor2, R2);

  //  ledcWrite(leftmotor1,  L1);
  //  ledcWrite(leftmotor2,  L2);
  //  ledcWrite(rightmotor1, R1);
  //  ledcWrite(rightmotor2, R2);
}

void terima_serial()
{
  if (Serial.available())
  {
    if (Serial.find("se"))
    {
      a = Serial.parseInt();
      b = Serial.parseInt();
      Serial.println();
      Serial.println();
      Serial.print("a:"); Serial.print(a);
```

```
    Serial.print("\t b:"); Serial.print(b);
    Serial.println();
    Serial.println();
    aa = String(a);
    if (a > 0 && a < 15)
    {
      Serial.println("stop ultrasonik.....");
      Motor(LOW, LOW, LOW, LOW);
    }
   }
  }
}
```

b. Camera_car_v6

```
/*
  esp32 1.0.4
  partitions : huge APP
  esp32 wrover module
*/

#include "esp_camera.h"
#include <WiFi.h>
#define CAMERA_MODEL_AI_THINKER
```

```
#include "camera_pins.h"

const char* ssid = "nicola";
const char* password = "12345678";

//extern int leftmotor1  = 4; // Left Motor
//extern int leftmotor2  = 2;
//extern int rightmotor1 = 14; // Right Motor
//extern int rightmotor2 = 15;

extern int leftmotor1  =  2; // Left Motor
extern int leftmotor2  = 14;
extern int rightmotor1 = 15; // Right Motor
extern int rightmotor2 = 13;

extern String Camerafeed = "";

//extern int a, b;

void startCameraServer();
void terima_serial();
```

```
void setup()
{
  //  delay(3000);
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  pinMode(leftmotor1, OUTPUT);
  pinMode(leftmotor2, OUTPUT);
  pinMode(rightmotor1, OUTPUT);
  pinMode(rightmotor2, OUTPUT);

  //initialize
  digitalWrite(leftmotor1, LOW);
  digitalWrite(leftmotor2, LOW);
  digitalWrite(rightmotor1, LOW);
  digitalWrite(rightmotor2, LOW);

  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;
//init with high specs to pre-allocate larger buffers
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;
}

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

//drop down frame size for higher initial frame rate
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF);
//  s->set_framesize(s, FRAMESIZE_QVGA);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
```

```
  Serial.println("WiFi connected");

  startCameraServer();

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

void loop()
{
  terima_serial();
}
```

LEMBAR PERNYATAAN ORISINALITAS SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama	: Muhammad Mas'ud Zulkifli
NIM	: 16.12.236
Program Studi	: Teknik Elektronika S-1
Fakultas	: Fakultas Teknologi Industri

Dengan ini menyatakan sesungguhnya bahwa skripsi saya yang berjudul: **"ROBOT PEMBERSIH DEBU OTOMATIS BERBASIS ARDUINO MENGGUNAKAN ANDROID"** merupakan hasil karya sendiri dan bukan hasil karya ilmiah orang lain atau plagiarisme.

Demikian pernyataan ini saya buat dengan sebenar-benarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika pada kemudian hari pernyataan ini tidak benar.

Malang, 14 April 2021

Yang Membuat Pernyataan

(Muhammad Mas'ud Zulkifli)

16.12.236

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-457/EL-FTI/2019 12 Oktober 2019
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Dr. Eng. Aryuanto Soetedjo, ST., MT.
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa:

Nama	: MUHAMMAD MAS`UD ZULKIFLI
NIM	: 1612236
Fakultas	: **Teknologi Industri**
Program Studi	: **Teknik Elektro S-1**
Peminatan	: T. Elektronika S1

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu :

"Semester Genap Tahun Akademik 2019/2020"

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih

Mengetahui
Ketua Program Studi Teknik Elektro S-1

Dr. Eng. I Komang Somawirata, ST, MT.
NIP. P. 1030100361

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

MONITORING BIMBINGAN SKRIPSI
SEMESTER GENAP TAHUN AKADEMIK 2019/2020

Nama : MUHAMMAD MAS'UD ZULKIFLI
NIM : 1612236
Nama Pembimbing I : Dr. Eng. Aryuanto Soetedjo, ST., MT.
Judul Skripsi : **Robot Pembersih Debu Otomatis Berbasis Arduino Menggunakan Android**

No	Hari Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	Senin 2/3/2020	10.16	Konsultasi progres minggu 1	
2	Jum'at 13/03/2020	14.59	Konsultasi judul	
3	Senin 6/4/2020	09.00	Penyesuaian komponen	
4	Jum'at 10/4/2020	07.05	Konsultasi progres alat penyedot debu	
5	Senin 13/4/2020	10.36	Konsultasi pengujian kamera	

Malang, Oktober 2020

Dosen Pembimbing I

Dr. Eng. Aryuanto Soetedjo, ST., MT.
NIP. Y. 1030800417

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-457/EL-FTI/2019 12 Oktober 2019
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Dr. F. Yudi Limpraptono, ST., MT.
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa:

Nama	: MUHAMMAD MAS`UD ZULKIFLI
NIM	: 1612236
Fakultas	: **Teknologi Industri**
Program Studi	: **Teknik Elektro S-1**
Peminatan	: T. Elektronika S1

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu :

"Semester Genap Tahun Akademik 2019/2020"

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih

Mengetahui
Ketua Program Studi Teknik Elektro S-1

Dr. Eng. I Komang Somawirata, ST, MT.
NIP. P. 1030100361

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

MONITORING BIMBINGAN SKRIPSI
SEMESTER GENAP TAHUN AKADEMIK 2019/2020

Nama : MUHAMMAD MAS'UD ZULKIFLI
NIM : 1612236
Nama Pembimbing II : Dr. F Yudi Limpraptono, ST., MT.
Judul Skripsi : **Robot Pembersih Debu Otomatis Berbasis Arduino Menggunakan Android**

No	Hari Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	Rabu 6/11/2019	10.16	Konsultasi Judul Skripsi	
2	Rabu 13/11/2019	09.00	Konsultasi flowchart sistem	
3	Jum'at 28/2/2020	13.30	Penyesuaian blok diagram	
5	Rabu 6/5/2020	10.36	Konsultasi integrasi android dengan sistem pengendalian	

Malang, Oktober 2020

Dosen Pembimbing II

Dr. F Yudi Limpraptono, ST., MT.
NIP. Y. 1039500274

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama	: MUHAMMAD MAS'UD ZULKIFLI
NIM	: 1612236
Program Studi	: Teknik Elektro S1
Peminatan	: Teknik Elektronika S1
Masa Bimbingan	: Semester Genap 2019-2020
Judul Skripsi	: **Robot Pembersih Debu Otomatis Berbasis Arduino Menggunakan Android**

Diperlihatkan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada:

Hari	:Rabu
Tanggal	: 29 Juli 2020
Nilai	: 75,95 (B)

Panitia Ujian Skripsi

Majelis Ketua Penguji

Dr. Eng. I Komang Somawirata, ST., MT
NIP. P. 1030100361

Sekretaris Majelis Penguji

Sotyohadi, ST., MT
NIP. Y. 1039700309

AnggotaPenguji

Dosen Penguji I

Ir. Kartiko Ardi Widodo, MT
NIP. Y. 1031400475

Dosen Penguji II

Sotyohadi, ST., MT
NIP. Y. 1039700309

LEMBAR PERSETUJUAN PERBAIKAN SKRIPSI

Nama	: MUHAMMAD MAS'UD ZULKIFLI
NIM	: 1612236
Program Studi	: Teknik Elektro S1
Peminatan	: Teknik Elektronika
Masa Bimbingan	:Semester Genap 2019-2020
Judul Skripsi	: **Robot Pembersih Debu Otomatis Berbasis Arduino Menggunakan Android**

Tanggal	Uraian	Paraf
Penguji II	1. Batasan masalah disempurnakan terumata untuk tujuan hanya pembersih debu lantai dan bekerja untuk penyedot debu	
	2. Untuk sensor infrared kalo memang tidak perlukan tidak usah dimasukkan dasar teori tentang cara kerja pembersih debu yang ada	
	3. Dalam perancangan minumal ada 3 blok, blok diagram disertai penjelasannya, yaitu blok diagram, rangkaian koneksi, diagram alir, diperbaiki simbol dan arah masing2 fungsi	
	4. Dalam pengujian bagian harus dilengkapi metode pengujian, hasil pengujian serta analisa. Dalam pengujian jara agar didetailkan dengan memberikan data sumpling sejumlah jarak. Perlu ada pengujian terhadap karakteristik bentuk benda yang ada di muka alat (temok, kaki meja atau benda lainnya). Perlu pengujian terhadap pergerakan motor. Perlu pengujian pengendalian android melalui modul wifi	

LEMBAR PERSETUJUAN PERBAIKAN SKRIPSI

	5. Kesimpulan disebutkan perpoint bagian sistem, baru item terakhir tentang pengujian secara keseluruhan, meliputi jarak minimal alat berhenti, permukaan apa yang bisa dideteksi, bagaimana pergerakan motor, bagaimana kesimpulan pengendalian melalui android	

Disetujui
Dosen Penguji I

Ir. Kartiko Ardi Widodo, MT
NIP. P. 1030400475

Mengetahui

Pembimbing I

Dr. Eng. Aryuanto Soetedjo, ST., MT
NIP. Y. 1030800417

Pembimbing II

Dr. F Yudi Limpraptono, ST. MT.
NIP. Y. 1039500274

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

LEMBAR PERSETUJUAN PERBAIKAN SKRIPSI

Nama	: MUHAMMAD MAS'UD ZULKIFLI
NIM	: 1612236
Program Studi	: Teknik Elektro S1
Peminatan	: Teknik Elektronika
Masa Bimbingan	:Semester Genap 2019-2020
Judul Skripsi	: **Robot Pembersih Debu Otomatis Berbasis Arduino Menggunakan Android**

Tanggal	Uraian	Paraf
Penguji I	1. Latar belakang yang menjadi dasar permasalahan perlu dijelaskan atau dapat mengacu dari sumber2 refrensi yang ada, bukan berdasarkan penilaian subjektif atau pribadi	
	2. Batasan masalah perlu diperbaiki mengacu kepada penelitian yang akan dikembangkan	
	3. Bab II tinjauan pustaka perlu disempurnakan atau diperbaiki karena bab II seharusnya menjelaskan mengenai dasar2 teori dan rumus tiap rangkaian tiap komponen serta elemen elektronik yang digunakan bukan sekedar gambar2 modul dari rangkaian	
	4. Transmiter infrared jika tidak digunakan tidak perlu dimasukkan dalam pembahasan	
	5. Komunikasi robot dengan	

LEMBAR PERSETUJUAN PERBAIKAN SKRIPSI

	android dijelaskan menggunakan apa dan bagaimana konsep algoritma komunikasi bekerja	
	6. Aplikasi yang digunakan dan dirancang diperangkat android dijelaskan di bab II sebagai teori dasarnya dan pengenalan aplikasinya dan bab III mengenai rancangan dari aplikasi android, sebagai pengendali robot atau alat pembersih	
	7. Digram alir atau sistem flowchart diperbaiki	
	8. Konsep rancangan robot, pergerakan robot dan pemrogaman robot dijelaskan di bab III	
	9. Konsep rancangan pengendalian robot dari smartphone (android) ke robot pembersih dijelaskan konseop algoritma dan metode pengendalian dijelaskan di bab III	
	10. Hasil pengujian atau pengukuran dilakukan lebih rinci di bab IV, pengukuran jarak, motor penyedot bekerja, motor gerak robot bekerja, pengiriman kamera diuji dan diukur semua	
	11. Kesimpulan disempurnakan dan diperbaikoi semua mengacu	

LEMBAR PERSETUJUAN PERBAIKAN SKRIPSI

	pada hasil pengujian atau pengukuran	
	12. Saran disempurnakan atau diperbaiki untuk penelitian selanjutnya dan hasil yang di capai pada penelitian anda	

Disetujui
Dosen Penguji II

Sotyohadi, ST., MT.
NIP. Y. 1039700309

Mengetahui

Pembimbing I

Dr. Eng. Aryuanto Soetedjo, ST., MT.
NIP. Y. 1030900417

Pembimbing II

Dr. F Yudi Limpraptono, ST., MT.
NIP. Y. 1039500274

ROBOT PEMBERSIH DEBU OTOMATIS BERBASIS ARDUINO MENGGUNAKAN ANDROID

ORIGINALITY REPORT

11%	11%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.scribd.com Internet Source	5%
2	scholar.unand.ac.id Internet Source	4%
3	e-jurnal.pnl.ac.id Internet Source	2%

Exclude quotes	On	Exclude matches	< 2%
Exclude bibliography	On		