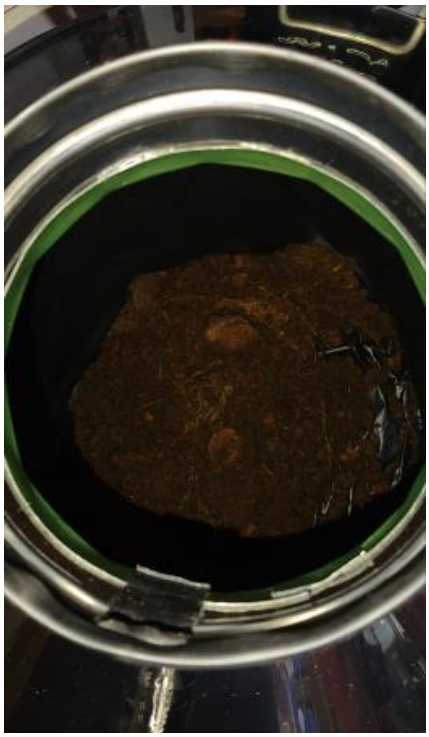
LAMPIRAN

Lampiran 1. Dokumentasi Percobaan Biodigester Limbah Serbuk Kayu

Lampiran 2. Dokumentasi Percobaan Biodigester Limbah Kulit Nanas

Lampiran 3. Dokumentasi Percobaan Biodigester Limbah Kotoran Bebek



Lampiran 4. Surat Kesediaan Pembimbing Tugas Akhir I

SURAT KESEDIAAN MEMBIMBING TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Bahrun Niam, M.T

NIPY : 09.015.277

Jabatan : Dosen Program Studi DIII Teknik Elektronika

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

Nama : Sona Aditya Putra

NIM : 24012001

Program Studi : DIII Teknik Elektronika

Judul Laporan TA : **RANCANG BANGUN BIODIGESTER PORTABEL BERBASIS MIKROKONTROLER ESP32**

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui,
Ka. Prodi DIII Teknik Elektronika,

Rony Darpono, M.T
NIPY. 09.015.282

Tegal, 25 Oktober 2024
Calon Dosen Pembimbing I,

Bahrun Niam, M.T
NIPY. 09.015.277

Lampiran 5. Surat Kesediaan Pembimbing Tugas Akhir II

SURAT KESEDIAAN MEMBIMBING TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Qirom, S.Pd, M.T

NIPY : 09.015.281

Jabatan : Dosen Program Studi DIII Teknik Elektronika

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

Nama : Sona Aditya Putra

NIM : 24012001

Program Studi : DIII Teknik Elektronika

Judul Laporan TA : **RANCANG BANGUN BIODIGESTER PORTABEL BERBASIS MIKROKONTROLER ESP32**

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui,
Ka. Prodi DIII Teknik Elektronika,

Rony Darpono, M.T
NIPY. 09.015.282

Tegal, 25 Oktober 2024
Calon Dosen Pembimbing II,

Qirom, S.Pd, M.T
NIPY. 09.015.281

Lampiran 6. Form Bimbingan Tugas Akhir I

FORM BIMBINGAN
TUGAS AKHIR

NAMA : SONA ADITYA PUTRA

NIM : 24012001

JUDUL TA : RANCANG BANGUN BIODIGESTER PORTABEL BERBASIS MIKROKONTROLER ESP32

Pembimbing 1

No	Hari/Tanggal	Uraian	Tanda Tangan
1.	25/10/2024	Pengajuan Judul dan Konsep TA	
2.	19/12/2024	Bimbingan BAB I - Objek penelitian - Cover halaman depan - Referensi max 5 tahun terakhir	
3.	19/12/2024	Bimbingan BAB II - III - Mengubah sensor (BME280) DHT22 - Penulisan kata asing	
4	23/01/2025	BAB IV - Pembahasan - Proses (langkah-langkah) hingga percobaan dimulai - Metana mulai nyala api (beapa ppm)	
5.	23/01/2025	BAB V Cover - Latbel - rumusan, tujuan, batasan masalah - Gambar rangkaian (jelasin fungsi) - Gambar alat jadi (jelasin proses) - Pengujian - hasil pembahasan - Gambar Cover 2D	

Lampiran 7. Form Bimbingan Tugas Akhir II

FORM BIMBINGAN
TUGAS AKHIR

NAMA : SONA ADITYA PUTRA

NIM : 24012001

JUDUL TA : RANCANG BANGUN BIODIGESTER PORTABEL BERBASIS MIKROKONTROLER ESP32

Pembimbing 2

No	Hari/Tanggal	Uraian	Tanda Tangan
1.	25/10/2024	Pengajuan Judul dan Konsep TA	
2.	03/01/2025	BAB I	
3.	03/01/2025	Bimbingan BAB II - Sitasi subbab 2.2.2 ditambahkan - menggunakan sitasi IEEE pada subbab 2.2.4	
4.	03/01/2025	BAB III - Gambar desain 3.2 ditambahkan keterangan dimensi	
5.	23/01/2025	BAB IV dan V - Penambahan Proses pembuatan alat - Literatur di pembahasan	

Lampiran 8. Kode Program

```
//RTC--------------------------------
#include <RtcDS1302.h>

ThreeWire myWire(4, 16, 2); // IO, SCLK, CE
RtcDS1302<ThreeWire> Rtc(myWire);
//rtc end-----------------------------------------------

//dht----------------------------------------
#include <DHT22.h>  // Pastikan library DHT22 sudah terpasang
#define pinDATA 26
DHT22 dht22(pinDATA);  // Membuat objek untuk sensor DHT22
//dht end-----------------------------------------------

//mq4----------------------

const int analogPin = 34;
const float Vref = 330;  // Tegangan referensi ESP32 (3.3V)
const int ADCResolution = 4095;  // Resolusi ADC ESP32 (12-bit)

float calculatePPM(float voltage) {
  return (voltage) * 1000;  // Eksperimen untuk hasil lebih akurat
}
//mq4 end--------------------------
```

```
//wisner-------------------------------------
const int wisnerPin = 33; // Pin analog tempat sensor terhubung
int rawwisnerValue = 0;      // Variabel untuk menyimpan nilai analog
int tekanan = 0;      // Variabel untuk menyimpan nilai yang dimapping
//wisner end---------------------------------------

//lcd----------------------------------------
#include <Bonezegei_LCD1602_I2C.h>
Bonezegei_LCD1602_I2C lcd(0x27);
//lcd end------------------------------

//sd card-----------------
#include "SD.h"
#include "SPI.h"
#include <Wire.h>

#define SCK  18
#define MISO  19
#define MOSI  23
#define CS  5

SPIClass spi = SPIClass(VSPI);

//sd card end----------------------------

unsigned long lastSaveTime = 0; // Waktu terakhir data disimpan
```

```
const unsigned long saveInterval = 600000; // Interval penyimpanan data (30 menit)

void setup() {
  Serial.begin(115200);
  //rtc---------------------------------------
   Rtc.Begin();

    RtcDateTime compiled = RtcDateTime(_DATE, __TIME_);
    printDateTime(compiled);
    Serial.println();

    if (!Rtc.IsDateTimeValid())
    {
        Serial.println("RTC lost confidence in the DateTime!");
        Rtc.SetDateTime(compiled);
    }

    if (Rtc.GetIsWriteProtected())
    {
        Serial.println("RTC was write protected, enabling writing now");
        Rtc.SetIsWriteProtected(false);
    }

    if (!Rtc.GetIsRunning())
    {
        Serial.println("RTC was not actively running, starting now");
```

```
        Rtc.SetIsRunning(true);
    }

    RtcDateTime now = Rtc.GetDateTime();
    if (now < compiled)
    {
        Serial.println("RTC is older than compile time!  (Updating DateTime)");
        Rtc.SetDateTime(compiled);
    }
    else if (now > compiled)
    {
        Serial.println("RTC is newer than compile time. (this is expected)");
    }
    else if (now == compiled)
    {
        Serial.println("RTC is the same as compile time! (not expected but all is fine)");
    }

//rtc end-------------------------------------------

//lcd--------------------------------
  lcd.begin();
  lcd.setPosition(0, 0);      //param1 = X   param2 = Y
  lcd.print("MONITORING");
  lcd.setPosition(0, 1);
  lcd.print("BIODIGESTER            ");
```

```
 delay(2000);
 lcd.clear();
 lcd.setPosition(0, 0);
 lcd.print("IKHSAN            ");
 lcd.setPosition(0, 1);
 lcd.print("22010004          ");
 delay(1000);
 lcd.clear();
 lcd.setPosition(0, 0);
 lcd.print("GILANG            ");
 lcd.setPosition(0, 1);
 lcd.print("22010005          ");
 delay(1000);
 lcd.clear();
 lcd.setPosition(0, 0);
 lcd.print("SONA            ");
 lcd.setPosition(0, 1);
 lcd.print("24012001              ");
 delay(1000);
 lcd.clear();

 //lcd end-------------------------

 // SD Card Initialization
spi.begin(SCK, MISO, MOSI, CS);
```

```
  if (!SD.begin(CS,spi,80000000)) {
    Serial.println("Card Mount Failed");
    lcd.setPosition(0, 0);
    lcd.print("Card Mount");
    lcd.setPosition(0, 1);
    lcd.print("Failed           ");
    delay(2000);
    lcd.clear();
  }
  uint8_t cardType = SD.cardType();

  if(cardType == CARD_NONE){
    Serial.println("No SD card attached");
    lcd.setPosition(0, 0);
    lcd.print("No SD card");
    lcd.setPosition(0, 1);
    lcd.print("Attached         ");
    delay(2000);
    lcd.clear();
  }
}

void loop() {
  //rtc-----------------------------------------------------
   RtcDateTime now = Rtc.GetDateTime();
```

```
  printDate(now);
  Serial.println();
  printTime(now);
  Serial.println();

  if (!now.IsValid())
  {
     Serial.println("RTC lost confidence in the DateTime!");
  }
//rtc end-------------------------------------------------------

//dht-----------------------------------------
float suhu = dht22.getTemperature()-5;  // Membaca suhu
float kelembaban = dht22.getHumidity();     // Membaca kelembaban
if (dht22.getLastError() != dht22.OK) {
  // Menampilkan error jika terjadi masalah
  Serial.print("Last error: ");
  Serial.println(dht22.getLastError());
}
// Menampilkan nilai suhu dan kelembaban ke serial monitor
Serial.print("kelembaban = ");
Serial.print(kelembaban, 1);  // Menampilkan kelembaban dengan 1 digit desimal
Serial.print("\t");
Serial.print("suhu = ");
Serial.println(suhu, 1);
//dht end-------------------------------------------
```

```
  //mq4----------------------------------
  int rawValue = analogRead(analogPin);  // Baca nilai ADC
  float voltage = rawValue * Vref / ADCResolution;  // Konversi nilai ADC ke
tegangan
  float ppm = calculatePPM(voltage);  // Hitung nilai PPM
  ppm = max(ppm - 2000, 0.0f);

  // Tampilkan hasil
  Serial.print("Methane: ");
  Serial.print(ppm, 2);
  Serial.println(" ppm");
  //mq4----------------------------------

  //wisner-------------------------------------
   rawwisnerValue = analogRead(wisnerPin);
  tekanan = map(rawwisnerValue, 0, 6600, 1, 12);
  Serial.print("tekanan: ");
  Serial.print(rawwisnerValue);
  Serial.print(" / ");
  Serial.println(tekanan);
  Serial.println("");
  //wisner end----------------------------

  //sdcard----------------------------
    unsigned long currentMillis = millis();
```

```
  if (currentMillis - lastSaveTime >= saveInterval) {
    saveToCSV(now, suhu, kelembaban, tekanan, ppm);
    lastSaveTime = currentMillis; // Perbarui waktu terakhir penyimpanan
  }
  //endsdcard


  //lcd--------------------------
  char suhuStr[4]; // Buffer untuk menyimpan nilai (cukup untuk angka hingga 3 digit + null terminator)
  itoa(suhu, suhuStr, 10); // Konversi wisnerValue ke string


  char kelembabanStr[4]; // Buffer untuk menyimpan nilai (cukup untuk angka hingga 3 digit + null terminator)
  itoa(kelembaban, kelembabanStr, 10); // Konversi wisnerValue ke string


  char wisnerStr[4]; // Buffer untuk menyimpan nilai (cukup untuk angka hingga 3 digit + null terminator)
  itoa(tekanan, wisnerStr, 10); // Konversi wisnerValue ke string


  char ppmStr[6]; // Buffer untuk menyimpan nilai (cukup untuk angka hingga 3 digit + null terminator)
  itoa(ppm, ppmStr, 10); // Konversi wisnerValue ke string


  lcd.setPosition(0, 0);
  lcd.print("H: ");
  lcd.print(kelembabanStr);
  lcd.print("% ");
```

```
  lcd.print("M: ");
  lcd.print(ppmStr);
  lcd.print(" ppm");

  lcd.setPosition(0, 1);
  lcd.print("T: ");
  lcd.print(suhuStr);
  lcd.print("C");
  lcd.print(" ");
  lcd.print("P: ");
  lcd.print(wisnerStr);
  lcd.print(" Bar ");
  //lcd end------------------

  delay(5000);
  lcd.clear();
}

void saveToCSV(const RtcDateTime &dt, float suhu, float kelembaban, int tekanan,
float ppm) {
  const char *filePath = "/data.csv";

  // Buka file dalam mode append
  File file = SD.open(filePath, FILE_APPEND);
  if (file) {
    // Tulis header setiap kali program dimulai
```

```
    static bool isHeaderWritten = false;
    if (!isHeaderWritten) {
      file.println("   Date     ,   Time   , Temperature , Humidity , Pressure ,  Methane
");
      isHeaderWritten = true;
      Serial.println("Header written to CSV file.");
    }

    // Format data dengan satuan dan padding untuk rata tengah
    char buffer[128];
    snprintf(buffer, sizeof(buffer),
             " %02u/%02u/%04u , %02u:%02u:%02u ,   %7.2f C   ,   %7.2f %%  ,  %7.2d
bar , %7.2f ppm ",
             dt.Month(), dt.Day(), dt.Year(),
             dt.Hour(), dt.Minute(), dt.Second(),
             suhu, kelembaban, tekanan, ppm);
    file.println(buffer);
    file.close();

    Serial.println("Data successfully appended to CSV:");
    lcd.setPosition(0, 0);
    lcd.print("save succes          ");
    delay(5000);
    lcd.clear();
  } else {
    Serial.println("Error: Failed to open CSV file for appending.");
    lcd.setPosition(0, 0);
```

```
    lcd.print("save failed          ");
    delay(5000);
    lcd.clear();

  }
}

void printDateTime(const RtcDateTime &dt)
{
    char dateString[11]; // Format: MM/DD/YYYY
    char timeString[9];  // Format: HH:MM:SS

    snprintf_P(dateString,
            countof(dateString),
            PSTR("%02u/%02u/%04u"),
            dt.Month(),
            dt.Day(),
            dt.Year());

    snprintf_P(timeString,
            countof(timeString),
            PSTR("%02u:%02u:%02u"),
            dt.Hour(),
            dt.Minute(),
            dt.Second());
```

```
    Serial.print("Date: ");
    Serial.print(dateString);
    Serial.print("  Time: ");
    Serial.print(timeString);
}

void printDate(const RtcDateTime &dt)
{
    char dateString[11]; // Format: MM/DD/YYYY

    snprintf_P(dateString,
            countof(dateString),
            PSTR("%02u/%02u/%04u"),
            dt.Month(),
            dt.Day(),
            dt.Year());

    Serial.print("Date: ");
    Serial.print(dateString);
}

void printTime(const RtcDateTime &dt)
{
    char timeString[9]; // Format: HH:MM:SS

    snprintf_P(timeString,
```

```
            countof(timeString),
            PSTR("%02u:%02u:%02u"),
            dt.Hour(),
            dt.Minute(),
            dt.Second());

    Serial.print("Time: ");
    Serial.print(timeString);
}
```