DAFTAR PUSTAKA

Adriansyah,Andi.(2013). Rancang Bangun Prototipe Elevator Menggunakan Microcontroller Arduino Atmega 328p. Universitas Mercu Buana.

Atyabi, Seyed A., Ebrahim A.,Wei S., & Xinguang C. (2022). Numerical Simulation Of Membrane Humidifier With Different Flow Fields For Medical Application. Medicine In Novel Technology And Devices. https://journals.elsevier.com/medicine-in-novel-technology-and-devices/.

Bipasha Biswas, S., & Tariq Iqbal, M. (2018). Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller. *Canadian Conference on Electrical and Computer Engineering*, *2018-May*(October). https://doi.org/10.1109/CCECE.2018.8447749.

Handoko, Ruri. (2022). Uji Karakteristik Humidifier Sebagai Sistem Pengkabutan Rumah Sarang Walet. Universitas Hasanuddin.

Khaenury, V. F., Darlis, D., & Mulyana, A. (2020). Rancang Bangun AlatMedical Check Up Berbasis Internet Of Things. *E-Proceeding of Engineering*, *6*(2), 2468–2475.

Maier, A., Sharp, A., & Vagapov, Y. (2017). Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. *2017 Internet Technologies and Applications, ITA 2017 - Proceedings of the 7th International Conference*, *September*, 143–148. https://doi.org/10.1109/ITECHA.2017.8101926.

Mila, Siti Muslikatul.(2006). Hubungan Antara Masa Kerja, Pemakaian Alat Pelindung Pernapasan (Masker) Pada Tenaga Kerja Bagian Pengamplasan Dengan Kapasitas fungsi Paru PT. Accent House Pecangaan Jepara. Universitas Negeri Semarang.

Nasrullah, E., Trisanto, A., & Utami, L. (2011). Rancang Bangun Sistem Penyiraman Tanaman Secara Otomatis Menggunakan Sensor Suhu LM35 Berbasis Mikrokontroler ATMega8535. *Bina Sarana Informatika Teknologi Elektro*, *5*(3), 182–192.

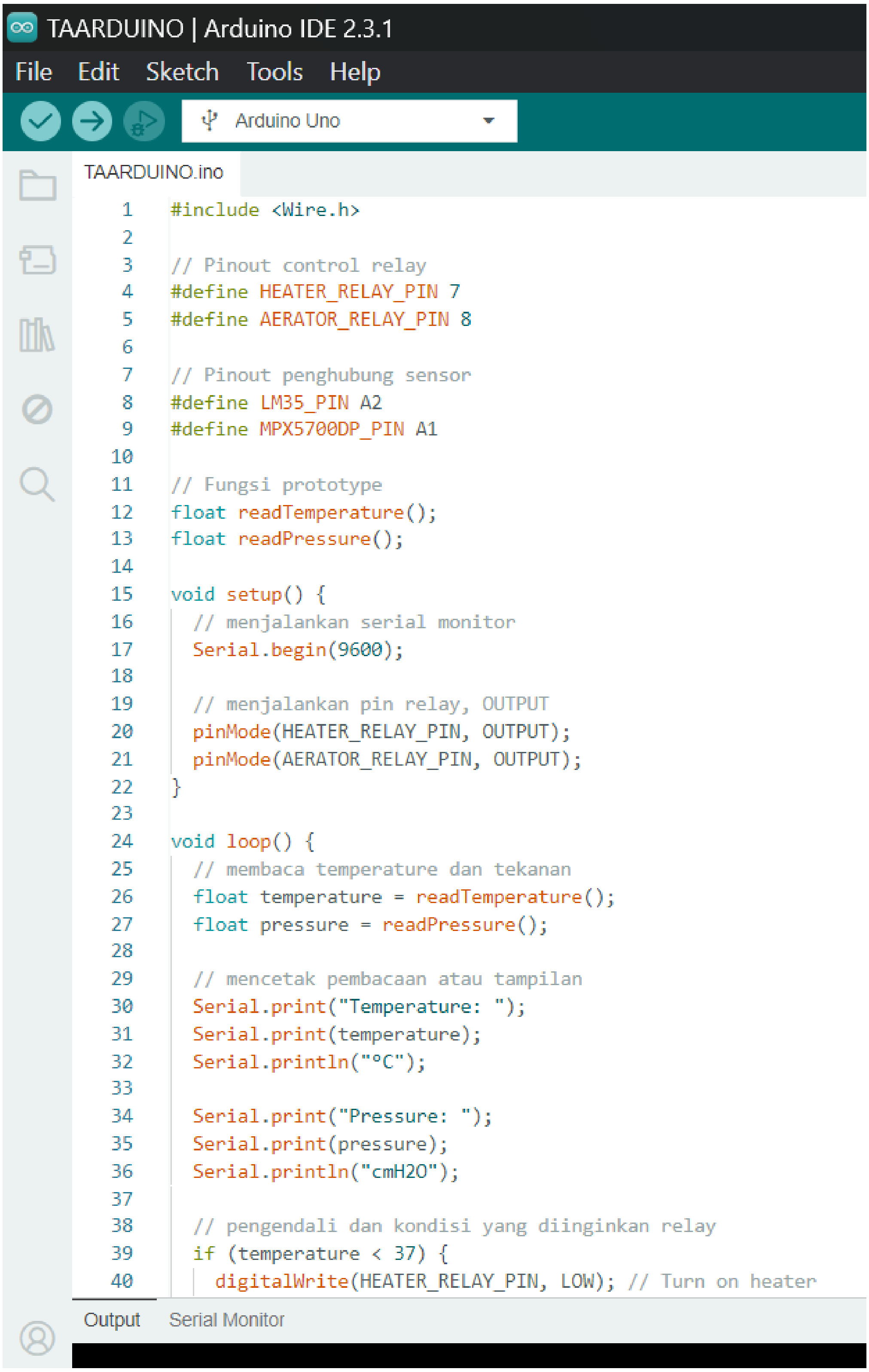
Safitri, Ariani Nur Rizkiya.(2019). Desain Monitoring Pernafasan Berbasis Serat Optik Dengan Konsep Compact Dan Konsep Compact Dan Portable. Institut

Teknologi Sepuluh Nopember Surabaya.

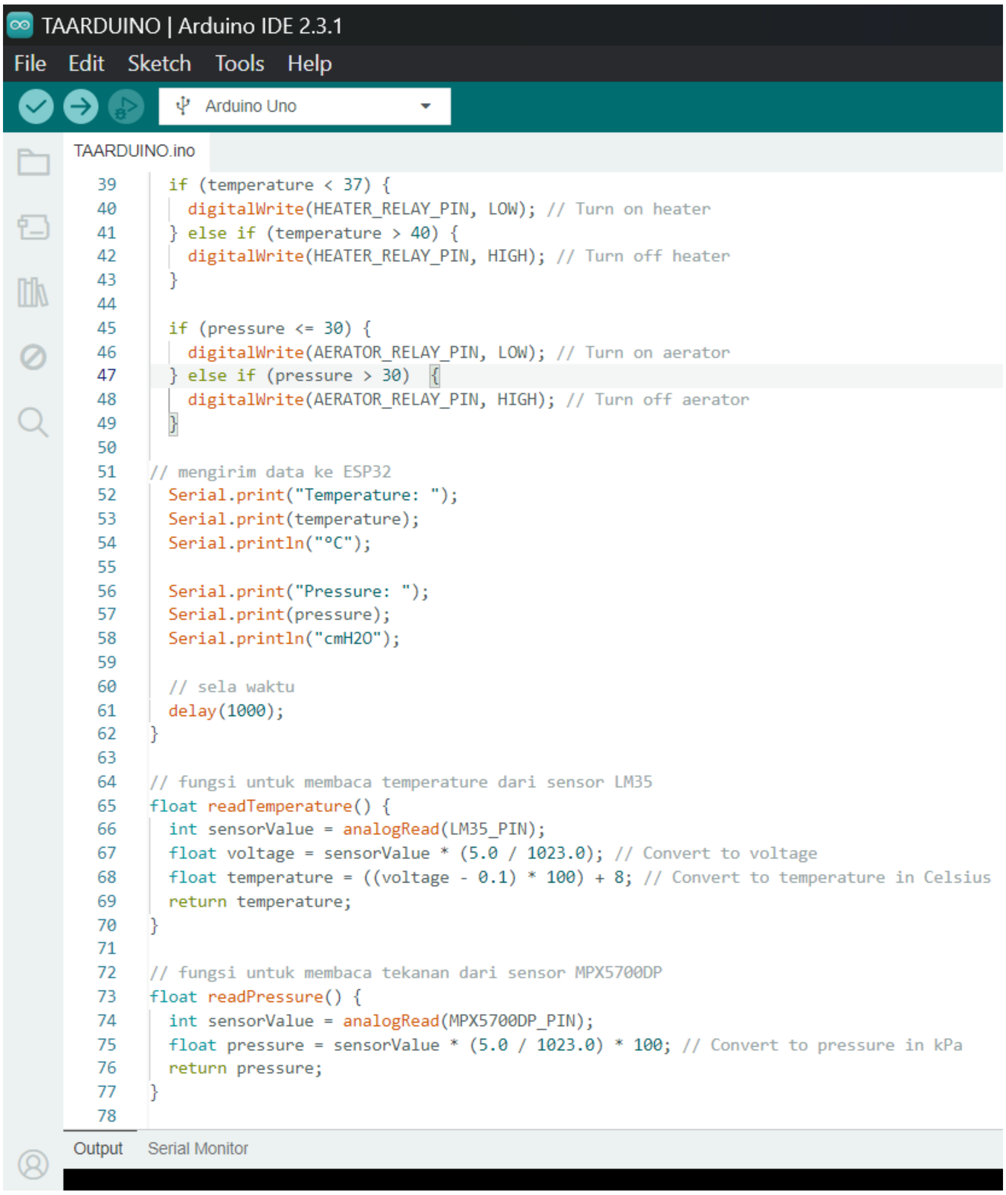
Shaputra, R. (2019). Kran Air Otomatis Pada Tempat Berwudhu Menggunakan Sensor Ultrasonik Berbasis Arduino Uno. Sigma Teknika, 2(2), 192. https://doi.org/10.33373/sigma.v2i2.2085.

Lampiran

Lampiran 1. Program Arduino Uno pada aplikasi Arduino IDE

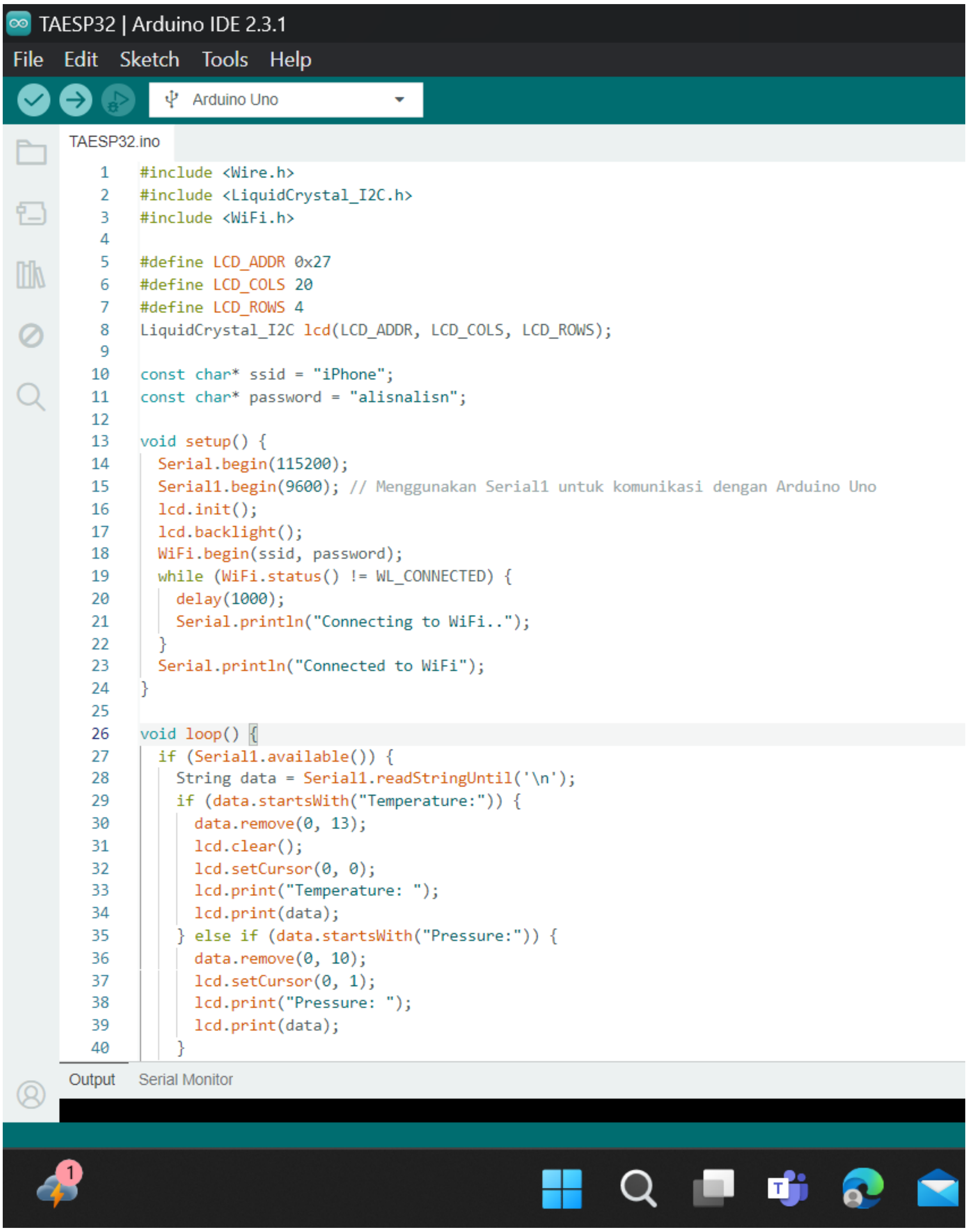


```
#include <Wire.h>

// Pinout control relay
#define HEATER_RELAY_PIN 7
#define AERATOR_RELAY_PIN 8

// Pinout penghubung sensor
#define LM35_PIN A2
#define MPX5700DP_PIN A1

// Fungsi prototype
float readTemperature();
float readPressure();

void setup() {
  // menjalankan serial monitor
  Serial.begin(9600);

  // menjalankan pin relay, OUTPUT
  pinMode(HEATER_RELAY_PIN, OUTPUT);
  pinMode(AERATOR_RELAY_PIN, OUTPUT);
}

void loop() {
  // membaca temperature dan tekanan
  float temperature = readTemperature();
  float pressure = readPressure();

  // mencetak pembacaan atau tampilan
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println("°C");

  Serial.print("Pressure: ");
  Serial.print(pressure);
  Serial.println("cmH2O");

  // pengendali dan kondisi yang diinginkan relay
  if (temperature < 37) {
    digitalWrite(HEATER_RELAY_PIN, LOW); // Turn on heater
```



```
  if (temperature < 37) {
    digitalWrite(HEATER_RELAY_PIN, LOW); // Turn on heater
  } else if (temperature > 40) {
    digitalWrite(HEATER_RELAY_PIN, HIGH); // Turn off heater
  }

  if (pressure <= 30) {
    digitalWrite(AERATOR_RELAY_PIN, LOW); // Turn on aerator
  } else if (pressure > 30) {
    digitalWrite(AERATOR_RELAY_PIN, HIGH); // Turn off aerator
  }

// mengirim data ke ESP32
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println("°C");

  Serial.print("Pressure: ");
  Serial.print(pressure);
  Serial.println("cmH2O");

  // sela waktu
  delay(1000);
}

// fungsi untuk membaca temperature dari sensor LM35
float readTemperature() {
  int sensorValue = analogRead(LM35_PIN);
  float voltage = sensorValue * (5.0 / 1023.0); // Convert to voltage
  float temperature = ((voltage - 0.1) * 100) + 8; // Convert to temperature in Celsius
  return temperature;
}

// fungsi untuk membaca tekanan dari sensor MPX5700DP
float readPressure() {
  int sensorValue = analogRead(MPX5700DP_PIN);
  float pressure = sensorValue * (5.0 / 1023.0) * 100; // Convert to pressure in kPa
  return pressure;
}

```

Lampiran 2. Program ESP32 pada aplikasi Arduino IDE



```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>

#define LCD_ADDR 0x27
#define LCD_COLS 20
#define LCD_ROWS 4
LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);

const char* ssid = "iPhone";
const char* password = "alisnalisn";

void setup() {
  Serial.begin(115200);
  Serial1.begin(9600); // Menggunakan Serial1 untuk komunikasi dengan Arduino Uno
  lcd.init();
  lcd.backlight();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to WiFi");
}

void loop() {
  if (Serial1.available()) {
    String data = Serial1.readStringUntil('\n');
    if (data.startsWith("Temperature:")) {
      data.remove(0, 13);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Temperature: ");
      lcd.print(data);
    } else if (data.startsWith("Pressure:")) {
      data.remove(0, 10);
      lcd.setCursor(0, 1);
      lcd.print("Pressure: ");
      lcd.print(data);
    }
```

Lampiran 3. Program Web pada Visual Studio Code

HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ventilator Monitoring - Universitas Hasanuddin</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <div class="logo">
      <img src="LogoUH.png" alt="Logo Universitas Hasanuddin">
    </div>
    <h1>Ventilator Monitoring System</h1>
  </header>
  <main>
    <div class="ventilator-info">
      <div class="ventilator-image">
        <img src="ventilator.png" alt="Gambar Ventilator">
      </div>
      <div class="ventilator-description">
        <h2>Ventilator</h2>
        <p>
          Ventilator pada umumnya merupakan alat yang digunakan untuk membantu pernapasan pasien yang mengalami kesulitan bernapas.
          Dengan menggunakan prinsip PVP (positive pressure ventilation), alat ini dapat mengatur aliran udara yang keluar dari ventilator ke paru-paru
          berupa suhu dan tekanan. data sensor tersebut akan ditampilkan sesuai dengan hasil kinerja dari alat ventilator ini.
        </p>
      </div>
    </div>
    <div class="sensor-data">
      <h2>Data Sensor</h2>
      <div class="sensor-reading" id="temperature">Suhu: 31.22 °C</div>
      <div class="sensor-reading" id="pressure">Tekanan: 19.06 Pa</div>
    </div>
  </main>
  <footer>
    <p>&copy; 2024 Ventilator Monitoring System - Universitas Hasanuddin</p>
  </footer>
</body>
</html>
```

Css :

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #0990ff; /* Warna background */
  color: #fff; /* Warna teks */
}

header {
  background-color: #006dcd; /* Warna header */
  padding: 20px; /
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

```
.logo img {
  max-width: 60px; /* Ukuran logo */
}

h1 {
  margin: 0;
  text-align: center;
  font-size: 24px; /* Ukuran judul */
  flex-grow: 1;
}

main {
  padding: 20px;
}

.ventilator-info {
  display: flex;
  align-items: center;
  justify-content: space-between;
  margin-bottom: 20px;
}

.ventilator-image img {
  max-width: 150px; /* Ukuran gambar ventilator */
  border-radius: 8px; /* Sudut bulat */
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); /* Bayangan */
}

.ventilator-description {
  flex-grow: 1;
  margin-left: 20px;
}
```

```
.sensor-data {
  background-color: #5da0f8; /* Warna latar belakang data */
  padding: 20px;
  border-radius: 8px; /* Sudut bulat */
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); /* Bayangan */
  display: flex;
  justify-content: space-between;
}

.sensor-reading {
  flex-grow: 1;
  margin: 0 5px;
  padding: 10px;
  border: 1px solid #0015ff;
  border-radius: 5px;
}

footer {
  background-color: #006dcd; /* Warna footer */
  color: #fff;
  padding: 10px;
  text-align: center;
  margin-top: 20px;
}
```

Script.Js :

```
// Fungsi untuk mengirim permintaan ke ESP32
function sendRequest(action) {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "192.168.43.27" + action, true);
    xhr.send();
}

// Fungsi untuk memperbarui data temperatur dan tekanan pada LCD
function updateLCD(data) {
   var temperature = data.temperature;
   var pressure = data.pressure;

   var xhr = new XMLHttpRequest();
   xhr.open("POST", "192.168.43.27/print", true);
   xhr.setRequestHeader("Content-Type", "application/json");
   xhr.send(JSON.stringify({ temperature: temperature, pressure: pressure }));
}

// Event listener untuk tombol on/off
document.getElementById("toggleButton").addEventListener("click", function()
{
   var action = this.textContent.toLowerCase();
   sendRequest(action);
});

// Fungsi untuk memperbarui data secara berkala (contoh setiap 2 detik)
setInterval(function() {
   var xhr = new XMLHttpRequest();
   xhr.onreadystatechange = function() {
      if (xhr.readyState == 4 && xhr.status == 200) {
         var data = JSON.parse(xhr.responseText);
         updateLCD(data);
      }
   };
   xhr.open("GET", "192.168.43.27/data", true);
   xhr.send();
}, 2000);
```