DAFTAR PUSTAKA

[1] Alimuddin, ST., MT. 2015. *Sistem Pengendalian Kadar pH, Suhu, dan Level Air Pada Model Miniatur Tambak Udang.* Jurnal Electro Luceat (JEC) Vol 1 No 1 Juli 2015.

[2] Al Barqi, Urwah, dkk. 2019. *Sistem Monitoring Online Pada Budidaya Udang Menggunakan Wireless Sensor Network dan Internet of Things*. Kumpulan Artikel Mahasiswa Pendidikan Informatika (KARMAPA TI) Vol. 8 No. 2 tahun 2019.

[3] Chandane, Mrinal Parikshit. 2016. *Real Time Operating System: A Complete Overview*. International Journal of Electrical and Electronics Engineers Vol No 8 Issue 1 January-June 2016.

[4] Espressif Systems (Shanghai) CO., LTD. 2016. *ESP-MESH*. Available from: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/mesh.html.

[5] Liu, Yu, dan Dr. Kin-Fai Tong. 2017. *Wireless Network in IoT Networks*. International Workshop on Electromagnetics: Application and Student Innovation Competition.

[6] Nur, Iswahyudi. 2017. *Pengendalian Sirkulasi dan Pengukuran Ph Air Pada Tambak Udang Berbasis Arduino*. Skripsi Fakultas Sains dan Teknologi UIN Alauddin Makassar.

[7] Nursan, Rio. 2013. *Analisis Loss Packet Pada Proses Download di Wide Area Network Menggunakan Wireshark*. Skripsi Fakultas Sains dan Teknologi UIN Sultan Syarif Kasim Pekanbaru.

[8] Setiawan, Sulhan. 2016. *Teknik Pemrograman dan Multithreading pada Mikrokontroller*. Yogyakarta: Andi.

[9] Suparta, Adrian, Boni P. Lapanporo, dan Apriyansyah. 2018. *Rancang Bangun Alat Ukur Salinitas dan Suhu Menggunakan Mikrokontroler ATMEGA328P Berbasis Data Logger yang Terintegrasi dengan GPS*. Jurnal PRISMA FISIKA, Vol VI No 1 (2018).

[10] Susilawati, Tati, dan Iwan Awaludin. 2019. *Eksplorasi Sensor, GPS, dan Moda Komunikasi Nirkabel Internet of Things.* Jurnal IKRA-ITH Vol 3 No 2 Juli 2019.

[11] Wisnu Jatmiko, dkk. 2004. *Teori & Aplikasi Realtime Operating System*. Depok: Fakultas Ilmu Komputer Universitas Indonesia.

LAMPIRAN 1: *Source Code* Node Sensor

```
#include "painlessMesh.h"
#include "DallasTemperature.h"

#define MESH_PREFIX     "PainlessMesh"
#define MESH_PASSWORD   "painlessmesh"
#define MESH_PORT       5555
#define ONE_WIRE_BUS 21

OneWire oneWire (ONE_WIRE_BUS);
DallasTemperature sensors (&oneWire);

Scheduler userScheduler;    //untuk mengontrol task
painlessMesh mesh;

float temperature, salinitas;
int tinggi, jarak, pH, sensorValue = 0, buf[10],temp;
const int trig = 23, echo = 22, pinpH = 35, pinSalinitas = 34;
long durasi;
unsigned long awal, akhir, hasil;
unsigned long int avgValue;

void receivedCallback(uint32_t from, String &msg){
  Serial.printf("Data diterima dari %u msg=%s\n", from, msg.c_str());
  mesh.sendSingle(309500513, msg);
}

void newConnectionCallback (uint32_t nodeId){
```

```
  Serial.printf("--> New Connection, nodeId = %u\n", nodeId);
}

void changedConnectionCallback(){
  Serial.printf("Changed Connections\n");
}

void nodeTimeAdjustedCallback(int32_t offset){
  Serial.printf("Adjusted time %u. offset = %d\n", mesh.getNodeTime(), offset);
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  sensors.begin();
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  xTaskCreate(
              taskSuhu,
              "TaskSuhu",
              10000,
              NULL,
              2,
              NULL);
  xTaskCreate(
              taskSalinitas,
              "TaskSalinitas",
              10000,
              NULL,
```

```
        2,
        NULL);
  xTaskCreate(
        taskTinggi,
        "TaskTinggi",
        10000,
        NULL,
        2,
        NULL);
  xTaskCreate(
        taskpH,
        "TaskpH",
        10000,
        NULL,
        2,
        NULL);
  mesh.setDebugMsgTypes( ERROR | STARTUP );
  mesh.init(MESH_PREFIX, MESH_PASSWORD, &userScheduler,
MESH_PORT);
  mesh.onReceive(&receivedCallback);
  mesh.onNewConnection(&newConnectionCallback);
  mesh.onChangedConnections(&changedConnectionCallback);
  mesh.onNodeTimeAdjusted(&nodeTimeAdjustedCallback);

  xTaskCreate(
        sendMessage,
        "SendMessage",
        10000,
```

```
            NULL,
            1,
            NULL);
}

void loop() {
  // put your main code here, to run repeatedly:
  mesh.update();
}

void taskSuhu(void * parameter){
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
  float temp,temp1;
  for (int i;;){
    temp1=0;
    for (int j=0;j<60;j++){
      vTaskDelayUntil(&xLastWakeTime, xFrequency);
      sensors.requestTemperatures();
      temp = sensors.getTempCByIndex(0);
      temp1 = temp1+temp;
    }
    temperature = temp1/60;
  }
}

void taskSalinitas(void * parameter){
```

```
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
  float sal,sal1;
  for (int i;;){
    sal1=0;
    for (int j=0;j<60;j++){
      vTaskDelayUntil(&xLastWakeTime, xFrequency);
      sal = analogRead(pinSalinitas)*0.5543-138.24;
      sal1 = sal1+sal;
    }
    salinitas=sal1/60;
  }
}

void taskTinggi(void * parameter){
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
  int h,h1;
  for (int i;;){
    h1=0;
    for (int j=0;j<60;j++){
      vTaskDelayUntil(&xLastWakeTime, xFrequency);
      digitalWrite(trig, HIGH);
      delayMicroseconds(10);
      digitalWrite(trig, LOW);
      durasi = pulseIn(echo, HIGH);
```

```
    jarak = durasi*0.034/2;
    h = 100 - jarak;
    h1 = h1+h;
   }
   tinggi = h1/60;
  }
}

void taskpH(void * parameter){
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
  int pHVol1,pHVol;
  for (int i;;){
   pHVol1=0;
   for (int j=0;j<60;j++){
    vTaskDelayUntil(&xLastWakeTime, xFrequency);
    for(int i=0;i<10;i++)
    {
     buf[i]=analogRead(pinpH);
    }
    for(int i=0;i<9;i++)
    {
     for(int j=i+1;j<10;j++)
     {
      if(buf[i]>buf[j])
      {
       temp=buf[i];
```

```
      buf[i]=buf[j];
      buf[j]=temp;
     }
    }
  }
  avgValue=0;
  for(int i=2;i<8;i++)
  avgValue+=buf[i];
  pHVol=avgValue/6;
  Serial.println(pHVol);
  pHVol1 = pHVol+pHVol1;
}
pHVol=pHVol1/60;
if (pHVol >= 0 && pHVol <= 300){
  pH = 14;
} else if (pHVol > 300 && pHVol <= 600){
  pH = 13;
} else if (pHVol > 600 && pHVol <= 900){
  pH = 12;
} else if (pHVol > 900 && pHVol <= 1200){
  pH = 11;
} else if (pHVol > 1200 && pHVol <= 1500){
  pH = 10;
} else if (pHVol > 1500 && pHVol <= 1800){
  pH = 9;
} else if (pHVol > 1800 && pHVol <= 2100){
  pH = 8;
} else if (pHVol > 2100 && pHVol <= 2400){
```

```
    pH = 7;
   } else if (pHVol > 2400 && pHVol <= 2700){
    pH = 6;
   } else if (pHVol > 2700 && pHVol <= 3000){
    pH = 5;
   } else if (pHVol > 3000 && pHVol <= 3300){
    pH = 4;
   } else if (pHVol > 3300 && pHVol <= 3600){
    pH = 3;
   } else if (pHVol > 3600 && pHVol <= 3900){
    pH = 2;
   } else pH = 1;
  }
 }


void sendMessage(void * parameter){
 TickType_t xLastWakeTime;
 const TickType_t xFrequency = 1000 * 60 / portTICK_PERIOD_MS;
 xLastWakeTime = xTaskGetTickCount();
 for (int i;;){
  vTaskDelayUntil(&xLastWakeTime, xFrequency);
  String msg = "{";
  msg += "\"Suhu_B\": ";
  msg += temperature;
  msg += ", \"Salinitas_B\": ";
  msg += salinitas;
  msg += ", \"Tinggi_Air_B\": ";
```

```
    msg += tinggi;
    msg += ", \"pH_B\": ";
    msg += pH;
    msg += ", \"T \": ";
    msg += hasil;
    msg += "}";
    mesh.sendSingle(309500513, msg);
    Serial.printf("Mengirim Data: %s\n", msg.c_str());
    akhir = millis();
    hasil = akhir-awal;
    Serial.println(hasil);
    awal = millis();
  }
}
```

LAMPIRAN 2: ***Source Code Mqtt Bridge***

```
#include <Arduino.h>
#include <painlessMesh.h>
#include <PubSubClient.h>
#include <WiFi.h>

#define   MESH_PREFIX     "PainlessMesh"
#define   MESH_PASSWORD   "painlessmesh"
#define   MESH_PORT       5555
#define   STATION_SSID     "JELEK SINYAL BOSS"
#define   STATION_PASSWORD "D42115008"
#define HOSTNAME "MQTT_Bridge"
#define TOKEN "8XIwnc32Y7uWblcVSymF"
#define thingsboardServer "cloud.thingsboard.io"

float temperature;
float salinitas;
int tinggi;
float pH;

// Prototypes
void receivedCallback( const uint32_t &from, const String &msg );

painlessMesh  mesh;

WiFiClient wificlient;
PubSubClient client(wificlient);

void setup() {
  Serial.begin(115200);

  xTaskCreate(
                taskSuhu,
                "TaskSuhu",
                10000,
                NULL,
                2,
                NULL);
  xTaskCreate(
                taskSalinitas,
                "TaskSalinitas",
                10000,
                NULL,
                2,
                NULL);
```

```
  xTaskCreate(
                taskTinggi,
                "TaskTinggi",
                10000,
                NULL,
                2,
                NULL);
  xTaskCreate(
                taskpH,
                "TaskpH",
                10000,
                NULL,
                2,
                NULL);


  mesh.setDebugMsgTypes( ERROR | STARTUP | CONNECTION );
  mesh.init( MESH_PREFIX, MESH_PASSWORD, MESH_PORT,
WIFI_AP_STA, 6 );
  mesh.onReceive(&receivedCallback);
  mesh.stationManual(STATION_SSID, STATION_PASSWORD);
  mesh.setHostname(HOSTNAME);

  xTaskCreate(
                sendMessage,
                "SendMessage",
                10000,
                NULL,
                1,
                NULL);

  mesh.setRoot(true);
  mesh.setContainsRoot(true);
  client.setServer(thingsboardServer,1883);

  WiFi.begin(STATION_SSID,STATION_PASSWORD);
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.println("Connected");

}

void loop() {
```

```
  mesh.update();
   if ( !client.connected() ) {
    reconnect();
  }
}

void receivedCallback( const uint32_t &from, const String &msg ) {
  Serial.printf("bridge: Received from %u msg=%s\n", from, msg.c_str());
  Serial.println(msg.c_str());
  client.publish("v1/devices/me/telemetry",msg.c_str());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Connecting to Thingsboard node ...");
    if ( client.connect("ESP32", TOKEN, NULL) ) {
      Serial.println( "[DONE]" );
    } else {
      Serial.print( "[FAILED] [ rc = " );
      Serial.print( client.state() );
      Serial.println( " : retrying in 1 seconds]" );
      delay( 1000 );
    }
  }
}

void taskSuhu(void * parameter){
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
  float temp, temp1;
  for (int i;;){
    temp = 0;
    for (int j=0;j<60;j++){
      vTaskDelayUntil(&xLastWakeTime, xFrequency);
      temp1 = random(26,30);
      temp = temp+temp1;
    }
  temperature=temp/60;
  }
}

void taskSalinitas(void * parameter){
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
```

```
  float temp,temp1;
  for (int i;;){
    temp1=0;
    for (int j=0;j<60;j++){
      vTaskDelayUntil(&xLastWakeTime, xFrequency);
      temp = random(1500,2000);
      temp1 = temp1+temp;
    }
    salinitas = temp1/60;
  }
}

void taskTinggi(void * parameter){
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
  int temp,temp1;
  for (int i;;){
    temp1 = 0;
    for (int j=0;j<60;j++){
      vTaskDelayUntil(&xLastWakeTime, xFrequency);
      temp = random (80,120);
      temp1 = temp1+temp;
    }
    tinggi = temp1/60;
  }
}

void taskpH(void * parameter){
  TickType_t xLastWakeTime;
  const TickType_t xFrequency = 1000 / portTICK_PERIOD_MS;
  xLastWakeTime = xTaskGetTickCount();
  float temp,temp1;
  for (int i;;){
    temp1 = 0;
    for (int j=0;j<60;j++){
      vTaskDelayUntil(&xLastWakeTime, xFrequency);
      temp = random(7,8);
      temp1 = temp1+temp;
    }
    pH = temp1/60;
  }
}

void sendMessage(void * parameter){
  TickType_t xLastWakeTime;
```

```
 const TickType_t xFrequency = 1000 * 60 / portTICK_PERIOD_MS;
 xLastWakeTime = xTaskGetTickCount();
 for (int i;;){
  vTaskDelayUntil(&xLastWakeTime, xFrequency);
  String msg = "{";
  msg += "\"Suhu_D\": ";
  msg += temperature;
  msg += ", \"Salinitas_D\": ";
  msg += salinitas;
  msg += ", \"Tinggi_Air_D\": ";
  msg += tinggi;
  msg += ", \"pH_D\": ";
  msg += pH;
  msg += "} ";
  client.publish("v1/devices/me/telemetry",msg.c_str());
  Serial.printf("Mengirim Data: %s\n", msg.c_str());
 }
}
```

KARTU BIMBINGAN SKRIPSI

Prodi S1 Teknik Informatika Universitas Hasanuddin

Stb.	Nama Mahasiswa
D42115008	Sabtian Juliana

Pembimbing.	Nama Pembimbing	Paraf & Tgl. Persetujuan Ujian Akhir
I	Adnan, S.T., M.T., Ph.D	
II	Ir. Christoforus Yohannes, M.T.	
No SK Pemb:		

Judul Skripsi:	Sistem Monitoring Kualitas Air Tambak Udang Menggunakan *Real Time Operating System*

No.	Tanggal Bimbingan	Uraian Kegiatan Bimbingan	Paraf Pemb.
1	17 Juni 2020	Konsultasi dengan Pembimbing I terkait kalibrasi sensor pH	
2	30 Juni 2020	Konsultasi dengan Pembimbing I terkait pengiriman data sistem	
3	19 Agustus 2020	Konsultasi dengan Pembimbing I terkait skema pengujian sistem	
4	30 Agustus 2020	Konsultasi dengan Pembimbing I terkait paper	
5	9 Oktober 2020	Konsultasi dengan Pembimbing I terkait perbaikan penulisan paper	
6	14 Oktober 2020	Konsultasi dengan Pembimbing II mengenai pengujian sistem	
7	21 Oktober 2020	Konsultasi dengan Pembimbing II mengenai pengujian sistem dan perbaikan penulisan	
8	1 November 2020	Konsultasi dengan Pembimbing II mengenai hasil pengujian	
9	4 November 2020	Konsultasi dengan Pembimbing II mengenai perbaikan penulisan dan presentasi seminar hasil	
10	4 November 2020	Konsultasi dengan Pembimbing I mengenai hasil pengujian dan penulisan skripsi	

LEMBAR PERBAIKAN SKRIPSI

"SISTEM MONITORING KUALITAS AIR TAMBAK UDANG MENGGUNAKAN *REAL TIME OPERATING SYSTEM*"

OLEH:

SABTIAN JULIANA
D421 15 008

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 29 Januari 2021. Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Adnan, S.T., M.T., Ph.D	
Sekretaris	Ir. Christoforus Yohannes, M.T.	
Anggota	Dr. Eng. Muhammad Niswar, S.T., M.I.T	
	Elly Warni, S.T., M.T.	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Adnan, S.T., M.T., Ph.D	
II	Ir. Christoforus Yohannes, M.T.	