**LAPORAN KERJA PRAKTEK ANTARES TELKOM**

**SMARTQUARIUM**

**Raymond Tirta, Kafin Mufid**

**Fisika ITB 2017**

1. **Latar Belakang**

Pada awalnya, kami berpikir untuk membuat aquascape dikarenakan dengan pembuatan aquascape ini harapan kedepannya, aquascape tersebut dapat menjadi cuci mata dan penenang hati saat sedang stress bekerja di kantor. Dengan melihat tumbuhan dan hewan dalam sebuah aquarium dapat menenangkan hati. Sehingga kami berpikir untuk membuat aquascape serta membuat alat monitoringnya juga agar jika pada hari libur, semua parameter tetap dapat terukur dan menyala.

Aquascape adalah seni bercocok tanam dalam air untuk mengatur dan mendekorasi sebuah set dari elemen natural seperti tumbuhan air, batu, kayu, substrat, dan fauna dalam air. Pada dasarnya, aquascape dapat berjalan dengan baik jika kita dapat memperhatikan parameter air dengan baik seperti suhu, *total dissolved solids* (TDS), pH, turbidity, dan lain sebagainya. Di mana dari parameter tersebut, maka fauna dan flora yang ada dalam tank aquascape juga akan senang dan dapat bertumbuh sesuai aslinya. Setiap tumbuhan dan hewan dalam aquascape mempunyai parameter sendiri dengan rentang suhu antara 21-29OC, TDS untuk soft water berkisar antara 50-200 ppm, turbidity baik di kisaran 1-500 NTU, dan terakhir pH kisaran 6 – 7,5.

Kami merancang sebuah alat di mana alat tersebut dapat membaca TDS, suhu, pH dan turbidity, kemudian merancang aplikasi yang dapat mengatur parameter tersebut, dan mengaplikasikan parameter tambahan seperti nyala lampu dan kipas pada aplikasi tersebut. Kami juga merancang web design sederhana untuk pengukuran dan untuk mengatur parameter. Tetapi kemudian kami tidak menggunakan pH sensor meter dikarenakan pH meter tidak bisa mengukur bersamaan dengan adanya TDS meter. Maka dalam hal ini kami mengukur pH secara manual, dan menggunakan TDS. \

1. **Teori Dasar**

Aquascape adalah seni bercocok tanam dalam air untuk mengatur dan mendekorasi sebuah set dari elemen natural seperti tumbuhan air, batu, kayu, substrat, dan fauna dalam air. Setelah akuarium menjadi bagian dari rumah, maka akuarium tersebut dapat saja menjadi kebanggaan dan masuk ke bahasan percakapan sehari hari. Pembuatan aquascape ini juga dapat membuat kita kreatif, imaginatif, dan mempunyai artistic yang besar. Selain menumbuhkan aspek dari hortikultura akuatik, dapat melibatkan physiology, pemangkasan, ekologi dan pemeliharaan akuarium juga menyiratkan aspek desain dan tata letak dari aquascape.

*Total Dissolved Solids* atau biasa juga di sebut TDS adalah sebuah parameter air di mana TDS mengindikasikan *general hardness* (GH), kesadahan (KH), nitrat, nitrit, ammonia, dsb. TDS meter adalah alat yang digunakan untuk mengukur TDS dalam satuan *part per million* (ppm). Konsentrasi TDS yang tinggi dapat mengurangi kejernihan air dan mengurangi fotosintesis pada tumbuhan dalam air, juga dapat memperlihatkan banyak racun dan logam yang menyebabkan temperature meningkat. TDS yang baik untuk ikan beragam ada yang mendekati nol TDS, sampai 300-400 ppm TDS. Dapat terbilang .

Nitrat dan nitrit dalam akuarium yang baik adalah nol, dan bisa ditoleransi sampai 40 ppm. Klorin yang baik adalah 0 ppm. Kemudian GH adalah kombinasi dari ion magnesium dan kalsium yang ditemukan di air. GH beragam tergantung penggunaan flora dan fauna pada aquarium. Sedangkan KH yang baik berkisar antara 120-180 ppm pada air jernih.

Turbidity adalah pengukuran kekeruhan dalam air, di mana parameternya tergantung kejernihan dari air. Air yang tidak jernih, dapat membuat aquarium sulit di lihat, dan mengganggu pandangan mata. Air yang keruh dapat dibersihkan dengan peggunaan filter dan juga alat bantu lainnya seperti *eachem clarity*, atau *microbe-lift clarifier plus freshwater*, dan sebagainya. Turbidity disebabkan oleh partikel yang tertangguhkan dan seringkali disebabkan oleh partikel substrat dan apapun yang berada di bawah tank oleh ikan atau saat dekorasi tank. Parameter yang baik untuk turbidity aquarium adalah di kisaran 1-600 satuan turbidity.

Adapun sensor-sensor yang digunakan pada alat ini adalah DS18B20 sensor suhu, gravity: TDS analog meter, dan gravity: Turbidity meter. Kemudian digunakan relay 2 channel untuk kipas serta lampu, dan menggunakan mikrokontroller ESP32.

1. **Skema dan Coding**

Adapun skema rangkaian sebagai berikut

A picture containing looking, photo, different, sitting

Description automatically generated

Gambar 1. Gambar skema rangkaian.

Sedangkan, kodingan Arduino adalah sebagai berikut.

|  |
| --- |
| #include <DallasTemperature.h>  #include "time.h"  #include <AntaresESP32HTTP.h>  #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  LiquidCrystal\_I2C lcd(0x27, 16, 2);  #define PinSuhu 32 //digital  #define PinTDS 34 //analog  #define PinPH 39 //analog  #define PinTurbidity 35 //analog  #define PinLampu 33 //IN1  #define PinKipas 25 //IN2  #define I2C\_SDA 22  #define I2C\_SCL 26  OneWire oneWire(PinSuhu);  DallasTemperature sensorSuhu(&oneWire);  #define offsettds 0.5  #define VCC 3.3  #define ADCResolution 4096  #define SCOUNT 10 //sum of sample poin for TDS  #define pHLength 10 //sum of sample poin for pH  #define tempLength 10 // sum of sample poin for temp  #define turbidityLength 10 // sum of sample poin for turbidity  #define Offset 0.00  #define turbidityOffset -0.7  const char \*ntpServer = "pool.ntp.org";  const long gmtOffset\_sec = 7 \* 3600;  const int daylightOffset\_sec = 0;  int relaytime\_hour;  int relaytime\_min;  int analogBuffer[SCOUNT];  int analogBufferTemp[SCOUNT];  int analogBufferIndex = 0, copyIndex = 0;  float avgV = 0, TDSValue = 0, tempSet = 25;  int pHArray[pHLength];  int pHArrayIndex = 0, tempArrayIndex = 0, turbidityArrayIndex = 0;  int tempArray[tempLength];  int turbidityArray[turbidityLength];  #define samplingInterval 200  #define printInterval 8000  const char \*ACCESSKEY = "38727ec4c4a0a1aa:7de4a820d4d6c54c"; // Ganti dengan access key akun Antares anda  const char \*WIFISSID = "LAB IOT Dev 5GHz"; // Ganti dengan SSID WiFi anda  const char \*PASSWORD = "labiot2018"; // Ganti dengan password WiFi anda  const char \*projectName = "testingkafin1"; // Ganti dengan application name Antares yang telah dibuat  const char \*sendName = "espsend"; // Ganti dengan device Antares yang telah dibuat  const char \*subName = "espreceive";  const unsigned long publishdelay = 60000;  const unsigned long subscribedelay = 15000;  const unsigned long samplingdelay = 100;  const unsigned long dataprocessdelay = 1000;  const unsigned long rtcdelay = 36E5;  const unsigned long lampudelay = 15000;  const unsigned long kipasdelay = 5000;  unsigned long publishtime = publishdelay; //tunggu sampling sebelum kirim data pertama  unsigned long subscribetime = subscribedelay; //inisiasi dilakukan di setup  unsigned long samplingtime;  unsigned long dataprocesstime = dataprocessdelay; //tunggu sampling sebelum diprocess  unsigned long rtctime = rtcdelay; //inisiasi dilakukan di setup  unsigned long lamputime;  unsigned long kipastime;  float pHValue, voltagePH, voltageTB, turbidityValue, tdsValue, tempValue, tempNow;  AntaresESP32HTTP antares(ACCESSKEY);  struct tm timeinfo;  //nilai awal default  float pHset = 6.5;  int temperatureset = 25;  int relayontime = 900;  int relayofftime = 1500;  int relayontime\_hour = relayontime / 100;  int relayontime\_min = relayontime && 100;  int relayofftime\_hour = relayofftime / 100;  int relayofftime\_min = relayofftime && 100;  void publishdata()  {  // Tambahkan variabel ke penampungan data sementara  antares.add("temperature", tempValue);  antares.add("pH", pHValue);  antares.add("turbidity", turbidityValue);  antares.add("tds", tdsValue);  // Publish data ke database Antares dan juga broker MQTT Antares  antares.send(projectName, sendName);  }  void subscribedata()  {  antares.get(projectName, subName);  // Mendapatkan data individu  pHset = antares.getFloat("pH");  temperatureset = antares.getInt("temperature");  relayontime = antares.getInt("ON");  relayofftime = antares.getInt("OFF");  relayontime\_hour = relayontime / 100;  relayontime\_min = relayontime && 100;  relayofftime\_hour = relayofftime / 100;  relayofftime\_min = relayofftime && 100;  Serial.println(pHset);  Serial.println(temperatureset);  Serial.println(relayontime);  Serial.println(relayofftime);  }  void setup()  {  Wire.begin(I2C\_SDA, I2C\_SCL);  Serial.begin(115200); // Buka komunikasi serial dengan baudrate 115200  lcd.init();  // Print a message to the LCD.  lcd.backlight();  lcd.setCursor(0, 0);  lcd.print("Powering on...");  lcd.setCursor(0, 1);  lcd.print("Please wait");  lcd.setCursor(0, 2);  antares.setDebug(true); // Nyalakan debug. Set menjadi "false" jika tidak ingin pesan-pesan tampil di serial monitor  antares.wifiConnection(WIFISSID, PASSWORD); // Mencoba untuk menyambungkan ke WiFi  while (WiFi.status() != WL\_CONNECTED)  {  delay(500);  Serial.print(".");  }  Serial.println("");  sensorSuhu.begin(); //setup sensor suhu  pinMode(PinTurbidity, INPUT); //setup pin turbidity  pinMode(PinTDS, INPUT); //setup pin tds  pinMode(PinLampu, OUTPUT);  pinMode(PinKipas, OUTPUT);  //inisialisasi variabel  Serial.println("getting the bloody time, one moment");  while (!getLocalTime(&timeinfo))  {  Serial.print(".");  configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);  delay(500);  }  subscribedata();  lcd.clear();  lcd.setCursor(0, 0);  lcd.print("T");  lcd.setCursor(0, 1);  lcd.print("pH");  lcd.setCursor(7, 0);  lcd.print("TB");  lcd.setCursor(7, 1);  lcd.print("TDS");  }  void loop()  {  if (millis() - subscribetime > subscribedelay) //get newest parameter  {  subscribetime = millis();  subscribedata();  }  if (millis() - publishtime > publishdelay) //publish data  {  publishtime = millis();  publishdata();  }  if (millis() - samplingtime > samplingdelay) //sampling  {  samplingtime = millis();  //masukkin index pH  pHArray[pHArrayIndex++] = analogRead(PinPH);  if (pHArrayIndex == pHLength)  pHArrayIndex = 0;  //masukkin index Temp  tempArray[tempArrayIndex++] = ambilSuhu();  if (tempArrayIndex == tempLength)  tempArrayIndex = 0;  //masukkin index turbidity  turbidityArray[turbidityArrayIndex++] = analogRead(PinTurbidity);  if (turbidityArrayIndex == turbidityLength)  turbidityArrayIndex = 0;  //masukkin index tds  analogBuffer[analogBufferIndex++] = analogRead(PinTDS);  if (analogBufferIndex == SCOUNT)  analogBufferIndex = 0;  }  if (millis() - dataprocesstime > dataprocessdelay) //data processing  {  dataprocesstime = millis();  //masukkin index pH  voltagePH = averagearray(pHArray, pHLength) \* VCC / ADCResolution;  pHValue = 3.5 \* voltagePH + Offset;  //masukkin index Temp  tempValue = ambilSuhu();  //masukkin index turbidity  voltageTB = (averagearray(turbidityArray, turbidityLength) \* 1.515 \* VCC / ADCResolution) + turbidityOffset;  turbidityValue = -1120.4 \* voltageTB \* voltageTB + (5742.3 \* voltageTB) - 4352.9;  //setting for tds  tdsValue = calculatetds();  float tdsv = tdsValue / 10;  Serial.print("temp is ");  Serial.println(tempValue);  Serial.print("turbidity is ");  Serial.print(turbidityValue);  Serial.print(" voltage is ");  Serial.println(voltageTB);  Serial.print("tds is ");  Serial.println(tdsValue);  lcddisplay();  }  if (millis() - lamputime > lampudelay) //cek lampu  {  lamputime = millis();  if (checkshouldon())  digitalWrite(PinLampu, LOW);  else  digitalWrite(PinLampu, HIGH);  }  if (millis() - kipastime > kipasdelay) //cek kipas  {  kipastime = millis();  if (tempValue > temperatureset)  digitalWrite(PinKipas, LOW);  else  digitalWrite(PinKipas, HIGH);  }  if (millis() - rtctime > rtcdelay) //sinkronisasi NTP  {  rtctime = millis();  configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);  }  }  float calculatetds() //function for calculating tds  {  for (copyIndex = 0; copyIndex < SCOUNT; copyIndex++)  analogBufferTemp[copyIndex] = analogBuffer[copyIndex];  float averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) \* (float)VCC / ADCResolution; // read the analog value more stable by the median filtering algorithm, and convert to voltage value  float compensationCoefficient = 1.0 + 0.02 \* (tempValue - 25.0); //temperature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02\*(fTP-25.0));  float compensationVolatge = averageVoltage / compensationCoefficient; //temperature compensation  float result = (133.42 \* compensationVolatge \* compensationVolatge \* compensationVolatge - 255.86 \* compensationVolatge \* compensationVolatge + 857.39 \* compensationVolatge) \* 0.5; //convert voltage value to tds value  return result;  }  float ambilSuhu() //function for getting temp  {  sensorSuhu.requestTemperatures();  float suhu = sensorSuhu.getTempCByIndex(0);  return suhu;  }  int getMedianNum(int bArray[], int iFilterLen) //getmedian  {  int bTab[iFilterLen];  for (byte i = 0; i < iFilterLen; i++)  bTab[i] = bArray[i];  int i, j, bTemp;  for (j = 0; j < iFilterLen - 1; j++)  {  for (i = 0; i < iFilterLen - j - 1; i++)  {  if (bTab[i] > bTab[i + 1])  {  bTemp = bTab[i];  bTab[i] = bTab[i + 1];  bTab[i + 1] = bTemp;  }  }  }  if ((iFilterLen & 1) > 0)  bTemp = bTab[(iFilterLen - 1) / 2];  else  bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;  return bTemp;  }  double averagearray(int \*arr, int number) //averaging stuff  {  int i;  int max, min;  double avg;  long amount = 0;  if (number <= 0)  {  Serial.println("Error number for the array to avraging!/n");  return 0;  }  if (number < 5)  { //less than 5, calculated directly statistics  for (i = 0; i < number; i++)  {  amount += arr[i];  }  avg = amount / number;  return avg;  }  else  {  if (arr[0] < arr[1])  {  min = arr[0];  max = arr[1];  }  else  {  min = arr[1];  max = arr[0];  }  for (i = 2; i < number; i++)  {  if (arr[i] < min)  {  amount += min; //arr<min  min = arr[i];  }  else  {  if (arr[i] > max)  {  amount += max; //arr>max  max = arr[i];  }  else  {  amount += arr[i]; //min<=arr<=max  }  } //if  } //for  avg = (double)amount / (number - 2);  } //if  return avg;  }  bool checkshouldon() //check based on time if it should on  {  bool checkonhour;  bool rollover;  //periksa rollover  if (relayontime\_hour < relayofftime\_hour)  rollover = false; //periksa jam  else if (relayontime\_hour = relayofftime\_hour && relayontime\_hour < relayofftime\_hour)  rollover = false; //periksa hingga menit  else  rollover = true;  if (!getLocalTime(&timeinfo))  {  Serial.println("Failed to obtain time");  return false;  }  Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");  //cara 1  /\*  if (rollover)  { //jika rollover  if (timeinfo.tm\_hour > relayontime\_hour)  checkonhour = true; //periksa jam saja sebelum roll  else if (timeinfo.tm\_min > relayontime\_min && timeinfo.tm\_hour == relayontime\_hour)  checkonhour = true; //periksa jika jam sama dgn on time  else if (timeinfo.tm\_hour < relayofftime\_hour)  checkonhour = true; //periksa jam saja setelah roll  else if (timeinfo.tm\_min > relayofftime\_min && timeinfo.tm\_hour == relayontime\_hour)  checkonhour = true; //periksa jika jam sama dgn off time  else  checkonhour = false;  }  else  {  if (timeinfo.tm\_hour > relayontime\_hour && timeinfo.tm\_hour < relayofftime\_hour)  checkonhour = true; //periksa jam saja  else if (timeinfo.tm\_min > relayontime\_min && timeinfo.tm\_hour == relayontime\_hour)  checkonhour = true; //periksa jika jam sama dgn on time  else if (timeinfo.tm\_min < relayofftime\_min && timeinfo.tm\_hour == relayofftime\_hour)  checkonhour = true; //periksa jika jam sama dgn off time  else  checkonhour = false;  }  \*/  // cara 2  int timetotal = timeinfo.tm\_hour \* 100 + timeinfo.tm\_min;  if (rollover)  {  if ((timetotal >= relayontime) || (timetotal < relayofftime))  checkonhour = true;  else  checkonhour = false;  }  else  {  if ((timetotal >= relayontime) && (timetotal < relayofftime))  checkonhour = true;  else  checkonhour = false;  }  return checkonhour;  }  void lcddisplay() //print lcd  {  lcd.setCursor(2, 0);  lcd.print(tempValue, 1);  lcd.setCursor(3, 1);  lcd.print(pHValue, 1);  lcd.setCursor(10, 0);  lcd.print(turbidityValue);  lcd.setCursor(11, 1);  lcd.print(tdsValue);  } |

Kemudian koding dengan menggunakan MIT APP Inventor adalah sebagai berikut.

Untuk screen 1, di mana screen ini digunakan sebagai home dari aplikasi tersebut.

A screenshot of a cell phone

Description automatically generated

Gambar 2. Gambar screen 1

Penggunaan screen ini adalah sebagai layar untuk melanjutkan ke page selanjutnya dengan click button.

Kemudian untuk screen 2 adalah sebagai berikut

A screenshot of a social media post

Description automatically generated

Gambar 3. Gambar inisiasi awal screen 2.

Gambar 3 merupakan inisiasi awal untuk mengambil URL dari Antares, dan header untuk menerima data dari server Antares menggunakan HTTP.

A screenshot of a cell phone

Description automatically generated

Gambar 4. Gambar screen 2 get\_data from web.

Koding di atas adalah untuk mendapatkan data atau menerima data terakhir atau terbaru dari website Antares, dan mengubahnya menjadi temperature, pH, turbidity, dan TDS.

A picture containing screenshot

Description automatically generated

Gambar 5. Gambar screen 2 procedure, dan set screen.

Pada gambar 5 tersebut adalah koding untuk membuat check live measurement button tidak ter ceklis dan untuk set warna dari text setiap label pada screen tersebut, dan juga procedure untuk mengambil parsing data dari web.

Kemudian pada set screen adalah sebagai berikut

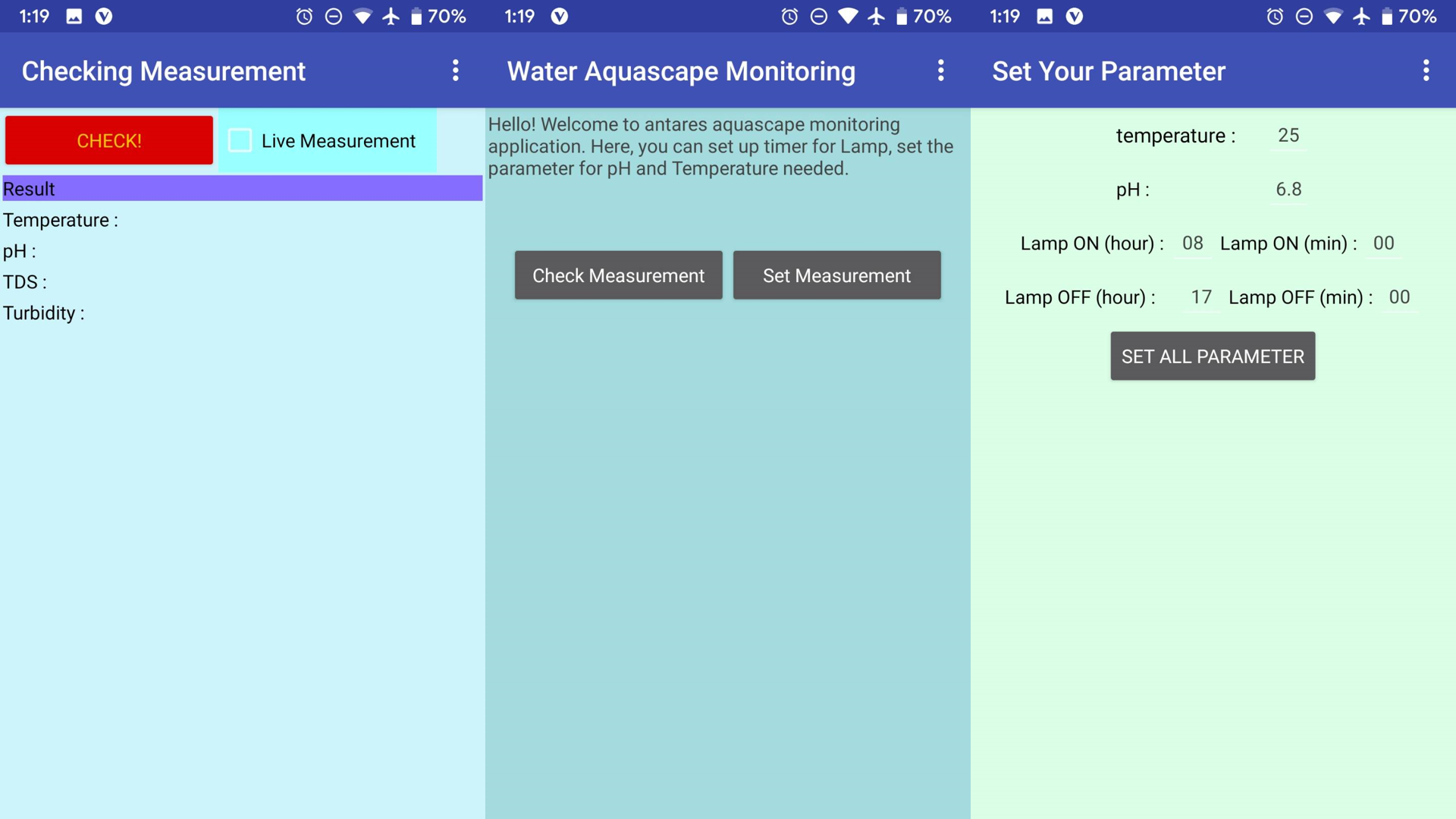
A screenshot of a cell phone

Description automatically generated

Gambar 6. Gambar *set measurement screen*.

Pada koding tersebut, url adalah untuk link Antares untuk mengirim data, secara HTTP.

Adapun design UI dari ketiga screen tersebut adalah sebagai berikut



Gambar 7. Gambar UI android.

1. **Hasil**

Hasil pembuatan aquascape dan alat monitoring aquascape dapat terlihat pada gambar berikut ini.

A picture containing indoor, window, sitting, small

Description automatically generated

Gambar 8. Gambar Awal aquarium.

A picture containing aquarium, indoor, vessel, table

Description automatically generated

Gambar 9. Gambar penyusunan awal aquascape

A large aquarium

Description automatically generated

Gambar 10. Hasil jadi aquarium.

A cluttered desk with a computer on a table

Description automatically generated

Gambar 11. Progress Alat monitoring.

A picture containing indoor, container, sitting, table

Description automatically generated

Gambar 12. Penyusunan sensor

A picture containing indoor, table, sitting, computer

Description automatically generated

Gambar 13. Hasil jadi alat monitoring

Kemudian Adapun berikut hasil keluaran monitoring pada Antares website

A screenshot of a social media post

Description automatically generated

Gambar 14. Gambar hasil penerimaan alat sensor di Antares.

A screenshot of a cell phone

Description automatically generated

Gambar 15. Gambar hasil set parameter untuk relay temperature dan jam.

1. **Daftar Pustaka**
2. https://aquascapinglove.com/learn-aquascaping/what-is-aquascaping/
3. https://atlantis-aquatics.com/tds-meter/#:~:text=TDS%20%3D%20(60%2D75%20ppm,%2C%20algae%20treatment%20chemicals%2C%20etc.
4. https://www.aquasabi.com/aquascaping-wiki\_aquarium\_cloudy-water