LAMPIRAN A

Listing Program Arduino Pada *Node Transmitter*

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Keypad.h>
#include <DHT.h>
DHT dht(A14, DHT11);
LiquidCrystal_I2C lcd(0x27, 20, 4);
#include <DS3231.h>
#include "RTClib.h"
int jam, menit, detik;
RTC_DS3231 rtc;
String myTime;
unsigned long MillisAwal = 0;
unsigned long Mi = 0;
char key;
long duration;
float distance, jarak;
float tinggi;
volatile byte rpmcount;
volatile unsigned long last_micros;
unsigned long timeold;
unsigned long timemeasure = 2.00;
int timetoSleep = 1;
unsigned long sleepTime = 15;
unsigned long timeNow;
int countThing = 0;
int GPIO_pulse = 2;
float rpm, rps;
float radius = 0.125;
anemometer wing
float velocity_kmh;
float angin;
float omega = 0;
float calibration_value = 1.1;
int kelembaban ;
int suhu ;
```

```
const int pin_interrupt = 3;
long int jumlah_tip = 0;
float curah_hujan = 0.00;
float curah_hujan_per_menit = 0.00;
float curah_hujan_per_jam = 0.00;
float curah_hujan_per_hari = 0.00;
float curah_hujan_hari_ini = 0.00;
float temp_curah_hujan_per_menit = 0.00;
float temp_curah_hujan_per_jam = 0.00;
float temp_curah_hujan_per_hari = 0.00;
float milimeter_per_tip = 1.7;
String cuaca;
volatile boolean flag = false;
#define trig 49
#define echo 51
unsigned long waktuSesudah;
unsigned long waktusebelum = 0;
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {A0, A1, A2, A3};
byte colPins[COLS] = {A4, A5, A6, A7};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins,
ROWS, COLS);
int value = 122;
int count = 0;

void setup() {
  Serial.begin(9600);
  Serial1.begin(115200);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  if (! rtc.begin()) {
```

```
    Serial.println("tidak menemukan RTC");
    while (1);
  }
  if (rtc.lostPower()) {
    Serial.println("daya RTC hilang, set waktu");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("     Sistem EWS      ");
  lcd.setCursor(0, 1);
  lcd.print("  LoRa  Transmiter  ");
  delay(2000);
  lcd.clear();
  setkecepatan_angin();
  set_rain();
  dht.begin();
}

void loop() {
  DateTime now = rtc.now();
  detik = now.second();
  jam = now.hour();
  menit = now.minute();
  kecepatan_angin();
  sen_rain();
  ultrasonik();
  Serial.println(distance);
  dht_11();
  TAMPILLCD();
   String kirimData = "(" + String(tinggi) + "," + String(angin) + "," +
   String(curah_hujan) + "," + String(curah_hujan_per_menit) + "," +
   String(suhu) + "," + String(kelembaban) + ")";
  Serial1.println(kirimData);
  Serial.println(kirimData);
  }
```

```
void setkecepatan_angin() {
  pinMode(GPIO_pulse, INPUT);
  digitalWrite(GPIO_pulse, LOW);
  detachInterrupt(digitalPinToInterrupt(GPIO_pulse));
  delay(10);
  attachInterrupt(digitalPinToInterrupt(GPIO_pulse), rpm_anemometer,
RISING);
  rpmcount = 0;
  rpm = 0;
  timeold = 0;
  timeNow = 0;
}
void kecepatan_angin()
{
  if ((millis() - timeold) >= timemeasure * 1000)
  {
    countThing++;
    detachInterrupt(digitalPinToInterrupt(GPIO_pulse));
    rps = float(rpmcount) / float(timemeasure);
    rpm = 60 * rps;
    omega = 2 * PI * rps;
    angin = omega * radius * calibration_value;
    velocity_kmh = angin * 3.6;
    if (countThing == 1)
    {
      countThing = 0;
    }
    timeold = millis();
    rpmcount = 0;
    attachInterrupt(digitalPinToInterrupt(GPIO_pulse), rpm_anemometer,
RISING);
  }}
void rpm_anemometer()
{
  if (long(micros() - last_micros) >= 5000)
  {
    rpmcount++;
    last_micros = micros();
  }}
```

```
void ultrasonik ()
{
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = (float)duration * 0.0343 / 2.00;
  tinggi = distance;
  delay(10);
}

void hitung_curah_hujan()
{
  flag = true;
  if (flag == true)
  {
    curah_hujan_per_menit += milimeter_per_tip;
    jumlah_tip++;
    flag = false;
  }
}
void set_rain()
{
  Serial.begin(9600);
  pinMode (pin_interrupt, INPUT_PULLUP);
  attachInterrupt (digitalPinToInterrupt(pin_interrupt),
hitung_curah_hujan, FALLING);
}
void sen_rain()
{
  if (millis() - waktusebelum > 60000) {
    curah_hujan += curah_hujan_per_menit;
    curah_hujan_per_menit = 0.00;
    jumlah_tip = 0;
    waktusebelum = millis();
  }
```

```
  if (jam == 23 && menit == 59 && detik > 58) {
    curah_hujan = 0.00;
    Serial.println("reset");
  }
}

void dht_11() {
  kelembaban = dht.readHumidity();
  suhu = dht.readTemperature();
}

void TAMPILLCD() {

  key = keypad.getKey();
  if (key == 'A') {
    count = 0;
    lcd.clear();
  }

  else if (key == 'B') {
    count = 1;
    lcd.clear();
  }

  else if (key == 'C') {
    count = 2;
    lcd.clear();
  }

  else if (key == 'D') {
    count = 3;
    lcd.clear();
  }
  else if (key == '#') {
    count = 4;
    lcd.clear();
  }
```

```
if (count == 0) {
  lcd.setCursor(0, 0);
  lcd.print("   Monitoring Air   ");
  lcd.setCursor(0, 2);
  lcd.print("Tinggi = ");
  lcd.print(tinggi);
  lcd.print(" Cm");
}
if (count == 1) {
  lcd.setCursor(0, 0);
  lcd.print("  Monitoring  Suhu  ");
  lcd.setCursor(0, 2);
  lcd.print("Kelembaban= ");
  lcd.print("77");
  lcd.print("%");
  lcd.setCursor(0, 3);
  lcd.print("Suhu      = ");
  lcd.print("29");
  lcd.print(" *C");
}
if (count == 2) {
  myTime = myTime +jam + "." + menit + "." + detik;
  lcd.setCursor(0, 0);
  lcd.print("  Monitoring Hujan  ");
  lcd.setCursor(0, 1);
  lcd.print("Jam   : " + myTime);
  myTime = "";
  lcd.print(" WIB");
  lcd.setCursor(0, 2);
  lcd.print("PerHri: ");
  lcd.print(curah_hujan);
  lcd.print(" ml/cm2");
  lcd.setCursor(0, 3);
  lcd.print("PerMnt: ");
  lcd.print(curah_hujan_per_menit);
  lcd.print(" ml/cm2");
}
```

```
if (count == 3) {
  lcd.setCursor(0, 0);
  lcd.print("  Monitoring Angin  ");
  lcd.setCursor(0, 2);
  lcd.print("Kec.Angin= ");
  lcd.print(angin);
  lcd.print("m/s");
}
if (count == 4) {
  if (key != NO_KEY) {
    switch (key) {
      case '1':
        value = (value * 10) + 1;
        break;
      case '2':
        value = (value * 10) + 2;
        break;
      case '3':
        value = (value * 10) + 3;
        break;
      case '4':
        value = (value * 10) + 4;
        break;
      case '5':
        value = (value * 10) + 5;
        break;
      case '6':
        value = (value * 10) + 6;
        break;
      case '7':
        value = (value * 10) + 7;
        break;
      case '8':
        value = (value * 10) + 8;
        break;
      case '9':
        value = (value * 10) + 9;
        break;
      case '0':
```

```
        value = (value * 10);
        break;
      case '*':
        value = 0;
        break;
    }
   }
   lcd.setCursor(0, 0);
   lcd.print("Kalibrasi : ");
   lcd.print(value);
   lcd.print(" cm");
  }
}
```

LAMPIRAN B

Listing Program LoRa Pada *Node Transmitter*

```
#include "heltec.h"
#include "images.h"
#define BAND    915E6
String FullData;
unsigned int counter = 0;
String rssi = "RSSI --";
String packSize = "--";
String packet ;
unsigned long terima;
unsigned long MillisAwal = 0;

void logo()
{
  Heltec.display->clear();
  Heltec.display->drawXbm(0, 5, logo_width, logo_height, logo_bits);
  Heltec.display->display();
}

void setup()
{
  Heltec.begin(true /*DisplayEnable Enable*/, true
/*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true
/*PABOOST Enable*/, BAND /*long BAND*/);
  Heltec.display->init();
  Heltec.display->flipScreenVertically();
  Heltec.display->setFont(ArialMT_Plain_10);
  logo();
  delay(1500);
  Heltec.display->clear();

  Heltec.display->drawString(0, 0, "Heltec.LoRa Initial success!");
  Heltec.display->display();
  delay(1000);
}
```

```
void loop() {
  if (Serial.available()) {
  FullData = Serial.readStringUntil('\n');
  Serial.println(FullData);
  }
  Heltec.display->clear();
  Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
  Heltec.display->setFont(ArialMT_Plain_10);
  Heltec.display->drawString(0, 0, "KIRIM PAKET : ");
  Heltec.display->drawString(90, 0, String(counter));
  Heltec.display->display();
  unsigned long MillisSekarang = millis();
  if (MillisSekarang - MillisAwal >= 1000) {
    MillisAwal = MillisSekarang;
    LoRa.beginPacket();
    LoRa.setTxPower(20, RF_PACONFIG_PASELECT_PABOOST);
    LoRa.print(FullData);
    LoRa.endPacket();
    counter++;
  }
}
```

LAMPIRAN C

Listing Program LoRa Pada *Node Receiver*

```
#include "heltec.h"
#include "images.h"
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
#define tombol 37
int itung = 1;
#define alarm 22

#define BAND    915E6
String rssi = "RSSI --";
String packSize = "--";
String packet;
unsigned long MillisAwal = 0;
unsigned long Millisreset = 0;
float tinggi, angin, hujan1, hujan2, suhu, kelembaban;
String tinggistring, anginstring, hujan1string, hujan2string, suhustring,
kelembabanstring;

void setup() {
  Heltec.begin(true /*DisplayEnable Enable*/, true
/*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true
/*PABOOST Enable*/, BAND /*long BAND*/);
  Heltec.display->init();
  Heltec.display->flipScreenVertically();
  Heltec.display->setFont(ArialMT_Plain_10);
  logo();
  delay(1500);
  Heltec.display->clear();
  Heltec.display->drawString(0, 0, "Heltec.LoRa Initial success!");
  Heltec.display->drawString(0, 10, "Wait for incoming data...");
  Heltec.display->display();
  LoRa.receive();
  lcd.backlight();
  lcd.begin();
  pinMode(alarm, OUTPUT);
```

```
  pinMode(tombol, INPUT_PULLUP);
  digitalWrite(alarm, LOW);
  lcd.setCursor(0, 0);
  lcd.print("     Sistem EWS      ");
  lcd.setCursor(0, 1);
  lcd.print("   LoRa  Receiver  ");
  lcd.clear();
}

void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    cbk(packetSize);
    Serial.println(packet);
    parsing(packet);
    antares.add("Tinggi Air", tinggi);
    antares.add("Curah hujan", hujan1);
    antares.add("Kecepatan angin", angin);
    antares.add("Suhu", suhu);
    antares.add("Kelembaban", kelembaban);
    lcdprint();
  }
}

void lcdprint() {
  if (digitalRead(tombol) == LOW ) {
    lcd.clear();
    itung++;
    delay(200);
  }
  if (itung == 1) {
    lcd.setCursor(0, 0);
    lcd.print("   Monitoring Air   ");
    lcd.setCursor(0, 2);
    lcd.print("Tinggi = ");
    lcd.print(tinggi);
    lcd.print(" Cm");
  }
```

```
 if (itung == 2) {
  lcd.setCursor(0, 0);
  lcd.print("  Monitoring  Suhu  ");
  lcd.setCursor(0, 2);
  lcd.print("Kelembaban= ");
  lcd.print("77");
  lcd.print("%");
  lcd.setCursor(0, 3);
  lcd.print("Suhu      = ");
  lcd.print("29");
  lcd.print(" *C");
 }

 if (itung == 3) {
  lcd.setCursor(0, 0);
  lcd.print("  Monitoring Hujan  ");
  lcd.setCursor(0, 2);
  lcd.print("PerHri: ");
  lcd.print(hujan1);
  lcd.print(" ml/cm2");
  lcd.setCursor(0, 3);
  lcd.print("PerMnt: ");
  lcd.print(hujan2);
  lcd.print(" ml/cm2");
 }

 if (itung == 4) {
  lcd.setCursor(0, 0);
  lcd.print("  Monitoring Angin  ");
  lcd.setCursor(0, 2);
  lcd.print("Kec.Angin= ");
  lcd.print(angin);
  lcd.print("m/s");
 }
 if (itung == 5) {
  itung = 1;
 }
 delay(5);
}
```

```
void parsing(String data) {
  byte awal = data.indexOf('(');
  byte koma1 = data.indexOf(',');
  byte koma2 = data.indexOf(',' , koma1 + 1);
  byte koma3 = data.indexOf(',' , koma2 + 1);
  byte koma4 = data.indexOf(',' , koma3 + 1);
  byte koma5 = data.indexOf(',' , koma4 + 1);
  byte akhir = data.indexOf(')');

  tinggistring = data.substring(awal + 1, koma1);
  anginstring = data.substring(koma1 + 1, koma2);
  hujan1string = data.substring(koma2 + 1, koma3);
  hujan2string = data.substring(koma3 + 1, koma4);
  suhustring = data.substring(koma4 + 1, koma5);
  kelembabanstring = data.substring(koma5 + 1, akhir);

  tinggi = tinggistring.toFloat();
  angin = anginstring.toFloat();
  hujan1 = hujan1string.toFloat();
  hujan2 = hujan2string.toFloat();
  suhu = suhustring.toFloat();
  kelembaban = kelembabanstring.toFloat();
  if (tinggi > 100) {
    digitalWrite(alarm, HIGH);
  }
  else {
    digitalWrite(alarm, LOW);
  }
}
```

LAMPIRAN D

Listing Program Arduino Pada *Node Receiver*

```
#include <String.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
unsigned long MillisAwal = 0;
unsigned long milislcd;
unsigned long Mi = 0;
int berhasil = 0;
int jmlkirim = 0;
float tinggi, angin, hujan1, hujan2, suhu, kelembaban;
String tinggistring, anginstring, hujan1string, hujan2string, suhustring,
kelembabanstring;
unsigned long terima, kirim, cd;
String FullData;
String Write_API_key = "MQOPIYXIH4A5C8QV";
String apn = "TSEL-SNS";

void setup() {
  Serial.begin(115200);
  Serial1.begin(9600);
  Serial3.begin(115200);
  SetupModule();
}

void loop() {
  if (Serial3.available()) {
    String FullData;
    FullData = Serial3.readStringUntil('\n');
    parsing(FullData);
  }
  unsigned long MillisSekarang = millis();
  if (MillisSekarang - MillisAwal >= 50) {
    MillisAwal = MillisSekarang;
    kirimdata();
  }
}
```

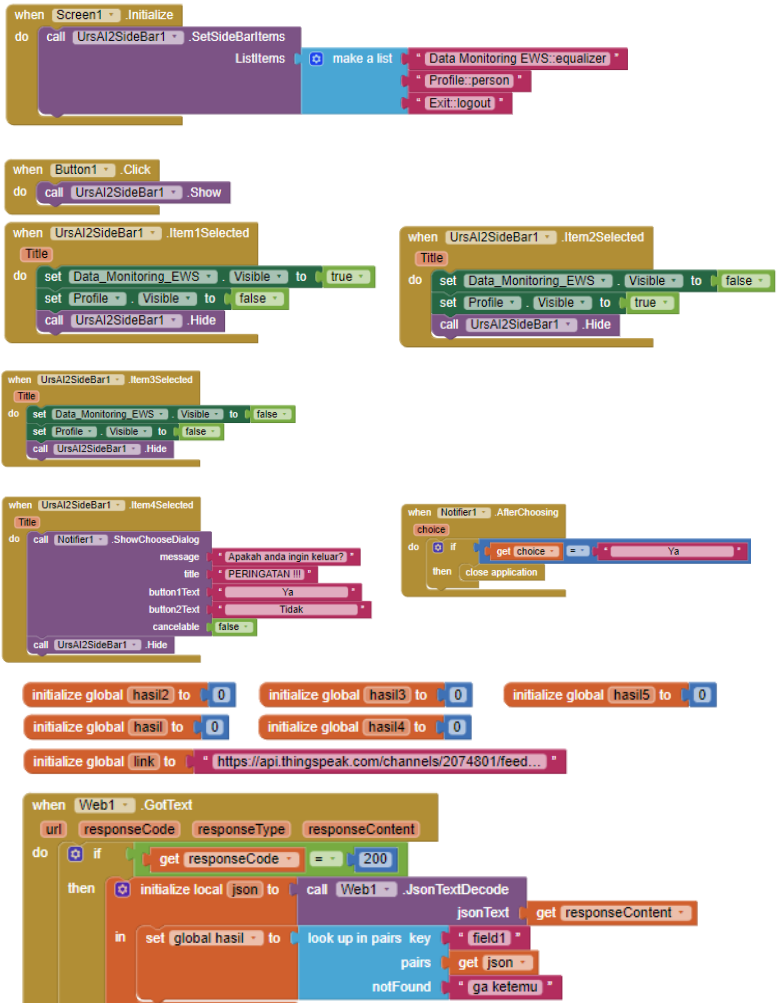
```
int data1, data2, data3, data4, data5;

void kirimdata() {
Serial1.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"
");
  delay(1000);
  ShowSerialData();
  Serial1.println("AT+CIPSEND");
  delay(1000);
  ShowSerialData();
  String str = "GET https://api.thingspeak.com/update?api_key=" +
Write_API_key + "&field1=" + String(suhu) + "&field2=" +
String(kelembaban) + "&field3=" + String(tinggi) + "&field4=" +
String(angin) + "&field5=" + String(hujan1);
  Serial1.println(str);
  delay(2000);
  Serial1.println((char)26);
  delay(2000);
  Serial1.println("AT+CIPSHUT");
  delay(500);
  ShowSerialData();
  delay(1000);
  ShowSerialData();
  delay(1000);
  str = "";
  delay(1000);
}

void SetupModule() {
  if (Serial1.available())Serial.write(Serial1.read());
  Serial1.println("AT"); delay(2000);
  ShowSerialData();
  Serial1.println("AT+CPIN?"); delay(2000);
  ShowSerialData();
  Serial1.println("AT+CREG?"); delay(2000);
  ShowSerialData();
  Serial1.println("AT+CGATT?"); delay(2000);
  ShowSerialData();
  Serial1.println("AT+CIPSHUT"); delay(2000);
```

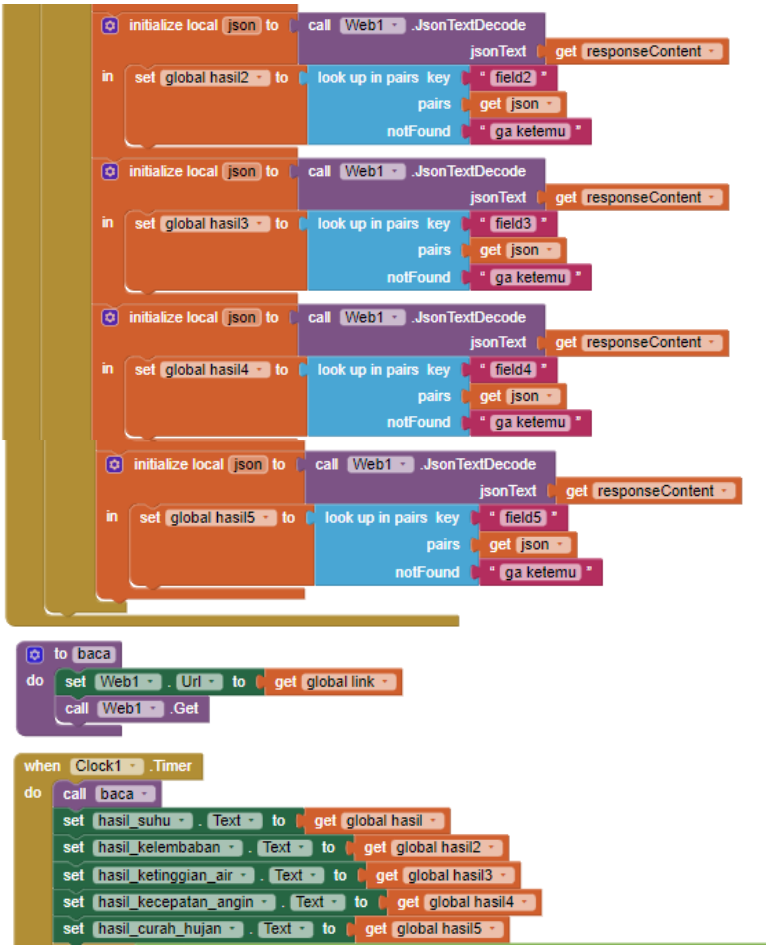
```
  ShowSerialData();
  Serial1.println("AT+CIPSTATUS"); delay(2000);
  ShowSerialData();
  Serial1.println("AT+CIPMUX=0");  delay(2000);
  ShowSerialData();
  Serial1.println("AT+CSTT=\"" + apn + "\""); delay(2000);
  ShowSerialData();
  Serial1.println("AT+CIICR");    delay(2000);
  ShowSerialData();
  Serial1.println("AT+CIFSR");    delay(2000);
  ShowSerialData();
  Serial1.println("AT+CIPSPRT=0"); delay(2000);
  ShowSerialData();
}

void ShowSerialData() {
  while (Serial1.available() != 0)
    Serial.write(Serial1.read());
  delay(1000);
}

void TerimaLORA() {
  if (Serial3.available()) {
    String FullData;
    FullData = Serial3.readStringUntil('\n');
    parsing(FullData);
    Serial.println(FullData);
  }
}
void parsing(String data) {
  byte awal = data.indexOf('(');
  byte koma1 = data.indexOf(',');
  byte koma2 = data.indexOf(',' , koma1 + 1);
  byte koma3 = data.indexOf(',' , koma2 + 1);
  byte koma4 = data.indexOf(',' , koma3 + 1);
  byte koma5 = data.indexOf(',' , koma4 + 1);
  byte akhir = data.indexOf(')');
```

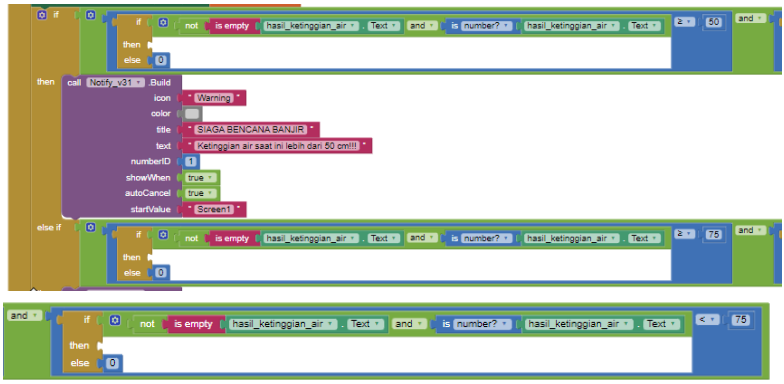
```
  tinggistring = data.substring(awal + 1, koma1);
  anginstring = data.substring(koma1 + 1, koma2);
  hujan1string = data.substring(koma2 + 1, koma3);
  hujan2string = data.substring(koma3 + 1, koma4);
  suhustring = data.substring(koma4 + 1, koma5);
  kelembabanstring = data.substring(koma5 + 1, akhir);

  tinggi = tinggistring.toFloat();
  angin = anginstring.toFloat();
  hujan1 = hujan1string.toFloat();
  hujan2 = hujan2string.toFloat();
  suhu = suhustring.toFloat();
  kelembaban = kelembabanstring.toFloat();
}
```

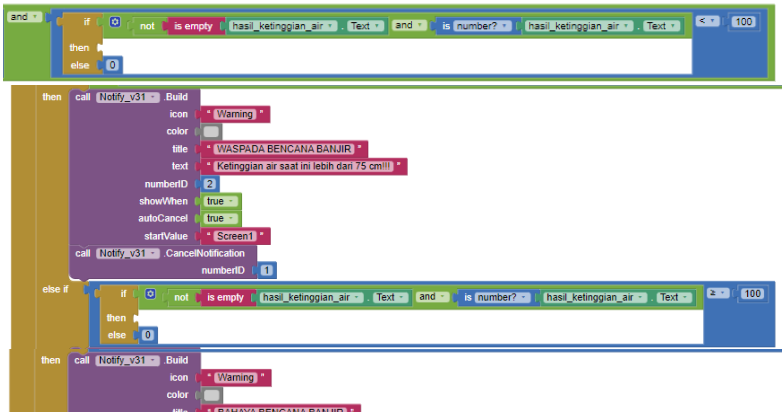
LAMPIRAN E

Listing Program Pada MIT App Inventor




initialize local json to call Web1 .JsonTextDecode
jsonText get responseContent
in set global hasil2 to look up in pairs key " field2 "
pairs get json
notFound " ga ketemu "
initialize local json to call Web1 .JsonTextDecode
jsonText get responseContent
in set global hasil3 to look up in pairs key " field3 "
pairs get json
notFound " ga ketemu "
initialize local json to call Web1 .JsonTextDecode
jsonText get responseContent
in set global hasil4 to look up in pairs key " field4 "
pairs get json
notFound " ga ketemu "
initialize local json to call Web1 .JsonTextDecode
jsonText get responseContent
in set global hasil5 to look up in pairs key " field5 "
pairs get json
notFound " ga ketemu "
to baca
do set Web1 . Url to get global link
call Web1 .Get
when Clock1 .Timer
do call baca
set hasil_suhu . Text to get global hasil
set hasil_kelembaban . Text to get global hasil2
set hasil_ketinggian_air . Text to get global hasil3
set hasil_kecepatan_angin . Text to get global hasil4
set hasil_curah_hujan . Text to get global hasil5


if
not
is empty
hasil_ketinggian_air
Text
and
is number?
≥
50
then
else
0
call Notify_v31 .Build
icon
Warning
color
title
SIAGA BENCANA BANJIR
text
Ketinggian air saat ini lebih dari 50 cm!!!
numberID
1
showWhen
true
autoCancel
startValue
Screen1
else if
75
and
<


and
if
not
is empty
hasil_ketinggian_air
Text
is number?
<
100
then
else
0
call Notify_v31 .Build
icon
Warning
color
title
WASPADA BENCANA BANJIR
text
Ketinggian air saat ini lebih dari 75 cm!!!
numberID
2
showWhen
true
autoCancel
startValue
Screen1
call Notify_v31 .CancelNotification
numberID
1
else if
≥
100
BAHAYA BENCANA BANJIR
Ketinggian air saat ini lebih dari 100 cm!!!
3
call Player1 .Start
50
call Notify_v31 .CancelAllNotifications

LAMPIRAN F
Hasil Alat

A. Gambar Tampak Depan *Node Trasmitter*

B. Gambar Tampak Depan *Node Trasmitter*

C. Gambar Tampak Samping

D. Gambar Keseluruhan EWS

E. Gambar *Panel Box Transmitter*

F. Gambar *Panel Box Receiver*

G. Data Iradiasi Matahari Dari PVsyst

Geographical site parameters, new site

Geographical Coordinates | Monthly meteo | Interactive Map

Site: **Politeknik Negeri Cilacap (Indonesia)**

Data source: Meteonorm 8.0 (1991-2009), Sat=100%

	Global horizontal irradiation	Horizontal diffuse irradiation	Temperature	Wind Velocity	Linke turbidity	Relative humidity
	kWh/m²/mth	kWh/m²/mth	°C	m/s	[-]	%
January	129.4	68.6	26.2	3.80	4.624	80.0
February	155.8	67.6	26.6	3.19	4.040	79.7
March	181.4	78.1	27.6	2.30	3.899	79.3
April	182.9	71.8	28.5	1.41	3.537	80.4
May	174.8	79.6	29.2	1.79	3.379	79.9
June	148.0	79.0	28.2	1.80	3.576	82.7
July	149.9	80.5	28.3	2.39	3.719	81.4
August	158.4	82.0	28.3	2.61	3.870	81.1
September	160.8	81.0	27.7	2.10	3.662	83.6
October	150.9	80.5	27.9	1.61	3.792	82.9
November	108.9	69.0	27.3	2.40	4.109	84.1
December	105.6	73.3	26.9	3.59	5.147	80.5
Year	**1807.0**	**910.9**	**27.7**	**2.4**	**3.946**	**81.3**