

LAMPIRAN

PERNYATAAN ORISINALITAS SKRIPSI

Yang bertanda tangan di bawah ini

Nama	Dwi Darmawan
NIM	1612219
Jurusan / Peminatan	Teknik Elektro S-1/Elektronika
ID KTP / Paspor	NIK. 3579012807980003
Alamat	Jl Lahor, Perum Lahor Agung no A9, RT 03 RW 12 Pesanggrahan, Batu
Judul Skripsi	Rancang Bangun Monitoring Hujan, Kecepatan Angin dan Tanah Longsor Berbasis Internet of Things(IOT)

Dengan ini menyatakan bahwa skripsi yang saya buat merupakan hasil karya sendiri bukan hasil plagiarisme dari orang lain. Dalam skripsi ini tidak memuat karya orang lain kecuali dicantumkan sumber yang digunakan sesuai dengan ketentuan yang berlaku.

Apabila ternyata di dalam skripsi ini dapat dibuktikan terdapat unsur-unsur plagiatisme, maka saya bersedia skripsi ini di gugurkan dan gelar akademik yang telah saya peroleh (S-1) di batalkan, serta di proses sesuai dengan perundang-undangan yang berlaku.

Malang, 22 September 2020

Yang membuat pernyataan



Dwi Darmawan)
1612219

**RANCANG BANGUN MONITORING HUJAN,
KECEPATAN ANGIN DAN TANAH LONGSOR
BERBASIS INTERNET OF THINGS(IOT)**

SKRIPSI

oleh:

Dwi Darmawan

1612219

Diajukan Untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Program Studi Teknik Elektronika
Institut Teknologi Nasional Malang

Diperiksa dan Disetujui:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Aryanto Soetedjo, ST., MT

NIP.Y. 1030800417

Sotyohadi, ST., MT

NIP. Y. 1039700309

Mengetahui:

Ketua Program Studi Teknik Elektro S-1

Dr. Eng. I Komang Somawirata, ST., MT

NIP P 1030100361

MALANG
September, 2020



INSTITUT TEKNOLOGI NASIONAL
MALANG

PERKUMPULAN PEMERUSA PERDIKSI UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Telpon : 0341-8610000 No. 2. Fax : 0341-8610001
E-mail : info@itn.ac.id itn@itn.ac.id

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa	Dwi Darmawan
NIM	1612219
Program Studi	Teknik Elektro S-1
Peminatan	Teknik Elektronika
Masa Bimbingan	Semester Genap 2019-2020
Judul Skripsi	Rancang Bangun Monitoring Hujan, Kecepatan Angin dan Tanah Longsor Berbasis Internet of Things(IOT)
Diperlihatkan dihadapan (S-1) Pada	Majelis Penguji Skripsi Jenjang Strata Satu
Hari	Selasa
Tanggal	4 Agustus 2020
Nilai	85,75 (A)

Panitia Ujian Skripsi

Ketua Majelis Penguji

Dr. Eng. I Komang Somawirata, ST., MT.
NIP. P. 1030100361

Sekretaris Majelis Penguji

Sotvuhadi, ST., MT.
NIP. V. 1039700309

Anggota Penguji

Dosen Penguji I

M. Ibrahim Ashari, ST., MT.
NIP. P. 1030100358

Dosen Penguji II

Dr. Eng. I Komang Somawirata, ST., MT.
NIP. P. 1030100361



PERKUMPULAN PEMBILDA PERDIKSIAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

INSTITUT TEKNOLOGI NASIONAL
MALANG

Jalan Sekeloa Selatan No. 1, Desa Sekeloa, Kecamatan Blimbing, Kabupaten Malang 65132
Telp. (0341) 821111, Fax. (0341) 821111, Email: info@itn.ac.id

LEMBAR PERSETUJUAN PERBAIKAN SKRIPSI

Nama Mahasiswa Dwi Darmawan
NIM 1612219
Program Studi Teknik Elektro S-1
Peminatan Teknik Elektronika S-1
Masa Bimbingan Semester Genap 2019-2020
Judul Skripsi **Rancang Bangun Monitoring Hujan,
Kecepatan Angin dan Tanah Longsor
Berbasis *Internet Of Things(IOT)***

Tanggal	Uraian	Paraf
Penguji I	-	

Disetujui,

Dosen Penguji I

M. Ibrahim Ashuri, ST., MT.
NIP. P. 1030100358

Mengetahui,

Dosen Pembimbing I

Dr. Eng. Aryanto Soetedjo, ST., MT.
NIP.Y. 1030800417

Dosen Pembimbing II

Sutvohadi, ST., MT.
NIP. Y 1039700309



PERKUMPULAN PEMBINA PERDIKLAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

INSTITUT TEKNOLOGI NASIONAL
MALANG

Jalan Sekeloa Selatan 1 No. 1, A, Malang 65132
Telp. (0341) 422100 Fax. (0341) 422101

LEMBAR PERSETUJUAN PERBAIKAN SKRIPSI

Nama Mahasiswa Dwi Darmawan
NIM 1612219
Program Studi Teknik Elektro S-1
Peminatan Teknik Elektronika S-1
Masa Bimbingan Semester Genap 2019-2020
Judul Skripsi **Rancang Bangun Monitoring Hujan,
Kecepatan Angin dan Tanah Longsor
Berbasis *Internet Of Things (IOT)***

Tanggal	Uraian	Paraf
Penguji II	Perlu di simpulkan dari data-data yang didapat untuk menyatakan tanah longsor	

Disetujui,

Dosen Penguji II

Dr. Eng. I Komang Somawirata, ST., MT.

NIP. P. 1030100361

Mengetahui,

Dosen Pembimbing I

Dr. Eng. Aryanto Soetedjo, ST., MT

NIP. 1030800417

Dosen Pembimbing II

Servohadi, ST., MT

NIP. Y. 1039700309



ITN DAN PASCASARJANA MALANG
www.itn.ac.id/ptn/MA/PTN

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Sembang I : J. Soekarno Sigitjoko No. 2, Telp. (041) 814311 (Ha. Bg), Fax. (041) 8143119 Malang 65145
Sembang II : J. Raya Gegerok, Km. 3 Telp. (041) 417333 (Ha. Bg), Fax. (041) 417341 (Sembang)

MONITORING BIMBINGAN SKRIPSI
SEMESTER GANJIL TAHUN AKADEMIK 2019-2020

Nama Mahasiswa : Dwi Darmawan
NIM : 1612219
Nama Pembimbing : Dr. Eng. Aryuanto Soetedjo, ST., MT
Judul Skripsi : Rancang Bangun Monitoring Hujan,
Kecepatan Angin dan Tanah Longsor
Berbasis *Internet Of Things (IoT)*

No.	Hari Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1.	Kamis 10/10/19	13.00	Latar belakang	
2.	Senin 21/10/19	13.00	Perancangan konsep alat dan sensor-sensor yang digunakan	
3.	Kamis 07/11/19	10.00	Dijijinkan Sempro	
4.	Kamis 14/11/19	11.00	Revisi Topologi WSN yang digunakan	
5.	Senin 02/03/20	10.00	Konsultasi sensor raindrop dan anemometer	
6.	Rabu 11/03/20	09.00	Konsultasi sensor anemometer	
7.	Kamis 26/03/20	13.00	Konsultasi prototype alat	
8.	Jum'at 03/04/20	09.17	Konsultasi topologi WSN yang digunakan	
9.	Sabtu 04/04/20	09.00	ACC makalah progres	
10.	Senin 06/04/20	11.15	Konsultasi komunikasi antar node	



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

ITN DR. PETEDJO MALANG
www.itn-malang.ac.id

Alamat : J. Soekarno Rahayu No. 2, Telp. (041) 814311 (4 lsg), Fax. (041) 8143119 Malang 65145
Gedung : J. Soekarno Rah. No. 2 Telp. (041) 817222 (2 lsg), (041) 41724 (2 lsg)

11.	Sabtu 18/04/20	14.00	Konsultasi komunikasi menggunakan NRF	
12.	Senin 05/05/20	18.00	ACC sembas	
13.	Minggu 26/07/20	19.31	Revisi Bab 3,4,5	
14.	Senin 27/07/20	17.13	ACC kompre	
15.	Senin 25/08/20	17.57	Revisi Bab 5 dan daftar pustaka	
16.	Selasa 26/07/20	14.52	ACC Buku Skripsi	

Malang, 2020
Dosen Pembimbing

Dr. Eng. Aryuanto Soetedjo, ST., MT
NIP.Y. 1030800417



FT dan PENSIPIL MALANG
BANK MADA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus 1 : Jl. Bendungan Sepuluh Nopember No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553515 Malang 65145
Kampus 2 : Jl. Raya Karangrejo, Km. 2 Telp. (0341) 417835 Fax. (0341) 417834 Malang

MONITORING BIMBINGAN SKRIPSI
SEMESTER GANJIL TAHUN AKADEMIK 2019-2020

Nama Mahasiswa : Dwi Darmawan
NTM : 1612219
Nama Pembimbing : Sotyohadi, ST., MT
Judul Skripsi : Rancang Bangun Monitoring Hujan,
Kecepatan Angin dan Tanah Longsor
Berbasis *Internet Of Things (IoT)*

No.	Hari Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1.	Rabu 06/11/19	13.30	Revisi latar belakang dan abstrak	
2.	Rabu 13/11/19	13.00	Revisi Judul	
3.	Kamis 14/11/19	11.00	Diizinkan Sempro	
4.	Senin 20/01/20	12.00	Revisi Bab 1 dan 2	
5.	Jum'at 01/05/20	12.05	Revisi flowchat	
6.	Sabtu 02/05/20	19.17	Konsultasi sensor hujan dan anemometer	
7.	Selasa 05/05/20	18.37	Diizinkan semhas	
8.	Jum'at 17/07/20	13.28	Revisi bab 2	
9.	Rabu 22/07/20	11.36	Revisi bab 4 dan sumber Referensi belum ada	
10.	Kamis 23/07/20	14.25	Revisi bab 5	



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. IRE PERSEKUTU MALANG
BANK NUSA MALANG

Kampus I : J. Dendang Sigit-guna No. 3 Telp. (0341) 551431 (Kuning), Fax. (0341) 553015 Malang 65145
Kampus II : J. Raya Kasongo, Km 2 Telp. (0341) 417838 Fax. (0341) 417834 Malang

11.	Jum'at 24/07/20	08.04	Dirizinkan ujian kompre Dan acc buku skripsi	

Malang, 2020
Dosen Pembimbing

Sotyo Hadi, ST., MT
NIP. Y. 1039700309



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BSI (PERSERO) MALANG
BANK NAGA MALANG

Kampus 1 J. Bondongan Segara-gara No. 2 Telp. (0341) 551431 (Pusat); Fax. (0341) 550018 Malang 65145
Kampus 2 J. Raya Kacanglo, Km 2 Telo. 0341/417630 Fax. (0341) 417634 Malang

Nomor Surat ITN-457/EL-FTI/2019
Lampiran -
Perihal BIMBINGAN SKRIPSI

12 Oktober 2019

Kepada : Yth. Dr. Eng. Aryunto Soetedjo, ST., MT.
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa:

Nama DWI DARMAWAN
NIM 1612219
Fakultas **Teknologi Industri**
Program Studi **Teknik Elektro S-1**
Peminatan T Elektronika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu

"Semester Ganjil Tahun Akademik 2019/2020"

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Mengetahui
Kepua Program Studi Teknik Elektro S-1

Dr. Eng. I Komang Somawirata, ST, MT.
NIP P 1030100361





PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BSI (PERSERO) MALANG
BANK NAGA MALANG

Kampus I
Kampus II

J. Bondongan Sempur-gara No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553315 Malang 65145
J. Raya Karanglo, Kiri 2 Telp. (0341) 417635 Fax. (0341) 417634 Malang

Nomor Surat ITN-457/EL-FTI/2019
Lampiran -
Perihal BIMBINGAN SKRIPSI

12 Oktober 2019

Kepada : Yth. Sotyohadi, ST., MT.
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa:

Nama DWI DARMAWAN
NIM 1612219
Fakultas **Teknologi Industri**
Program Studi **Teknik Elektro S-1**
Peminatan T Elektronika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu

"Semester Ganjil Tahun Akademik 2019/2020"

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Mengetahui
Ketua Program Studi Teknik Elektro S-1

Dr. Eng. I Komang Somawirata, ST, MT.
NIP. P 1030100361



Rancang Bangun Monitoring Hujan, Kecepatan Angin dan Tanah Longsor Berbasis Internet of Things(IoT)

ORIGINALITY REPORT

3%

SIMILARITY INDEX

2%

INTERNET SOURCES

0%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1**Submitted to Universitas Sebelas Maret**

Student Paper

3%

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On



**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Semester Ganjil Tahun Akademik 2019/2020**

PEMINATAN		T. Elektronika		
1	Nama Mahasiswa	DWI DARMAWAN	NIM	1612219
2	Keterangan Pelaksanaan	Tanggal	Waktu	Tempat
3	Judul Proposal yang Diseminarkan Mahasiswa	Rancang Bangun Monitoring Hujan Lokal, Kecepatan Angin dan Tanah Longsor Berbasis Internet of Things (IoT)		
4	Perubahan Judul yang Disarankan (bila ada)		
5	Masukan yang harus ditambahkan dalam skripsi: - Dr. Co. Supalaji (Wsu) yg sesua			
Persetujuan Judul Skripsi				
Disetujui, Dosen Keahlian (.....)				
Disetujui, Dosen Pembimbing				
Pembimbing I		Pembimbing II		
 Dr. Eng. Aryuanto Soetedjo, ST., MT		Sotyohadi, ST., MT		
Mengetahui, Ketua Program Studi Teknik Elektro S-1				
 Dr. Eng. I Komang Somawirata, ST., MT NIP. P*1030100361				

- **Coding Sink**

```
#include <SPI.h>
#include "RF24.h"
double got_time;
bool radioNumber = 1;

//RF24 radio(34, 36);
RF24 radio(8, 7);//tester
//RF24 radio(5, 10);//timbangan
byte addresses[][6] = {"1Node", "2Node"};
bool role = 0;

void setup()
{
  Serial.begin(9600);
  Serial.println(F("RF24/examples/reciever..."));
  Serial.println(F("**** PRESS 'T' to begin transmitting to the other
node"));
  radio.begin();
  radio.setPALevel(RF24_PA_LOW);
// radio.setPALevel(RF24_PA_MAX);
// radio.setDataRate(RF24_2MBPS);
// radio.setChannel(124);
// Open a writing and reading pipe on each radio, with opposite
addresses
  if (radioNumber)
  {
    radio.openWritingPipe(addresses[1]);
    radio.openReadingPipe(1, addresses[0]);
  } else {
    radio.openWritingPipe(addresses[0]);
    radio.openReadingPipe(1, addresses[1]);
  }
  // Start the radio listening for data
  radio.startListening();
}
void loop()
{
```

```

while ( radio.available())
{
  radio.startListening();
  // Variable for the received timestamp
  // if (radio.available()) {           // While there is data
  ready
    radio.read( &got_time, sizeof(got_time) );      // Get the payload
  // }
  // radio.stopListening();           // First, stop
  listening so we can talk
  // radio.write( &got_time, sizeof(unsigned long) );      // Send
  the final one back.
  // radio.startListening();           // Now, resume
  listening so we catch the next packets.
  Serial.print("RECEIVED DATA: ");
  Serial.println(got_time);
  delay(100);
}
} // Loop

```

- **Coding ESP sink**

```

#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
#define SSID_wifi "ZENSKAA"
#define SSID_pass "tahupetis"

char ssid[] = SSID_wifi;
char pass[] = SSID_pass;
int keyIndex = 0;
WiFiClient client;
unsigned long myChannelNumber = 1033865;
const char * myWriteAPIKey = "9TVKN2Y5MESYOU3C";

float a, b, c, d, e, f, g, h;
String myStatus = "";

void setup() {
  Serial.begin(9600);
  WiFi.mode(WIFI_STA);

```

```

ThingSpeak.begin(client);
}
void loop()
{
  if (WiFi.status() != WL_CONNECTED)
  {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SSID_wifi);
    while (WiFi.status() != WL_CONNECTED)
    {
      WiFi.begin(ssid, pass);
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }
  if (Serial.available())
  {
    if (Serial.find("se"))
    {
      a = Serial.parseFloat();
      b = Serial.parseFloat();
      c = Serial.parseFloat();
      d = Serial.parseFloat();
      e = Serial.parseFloat();
      f = Serial.parseFloat();
      g = Serial.parseFloat();
      h = Serial.parseFloat();

      Serial.print("a:"); Serial.print(a);
      Serial.print("\t b:"); Serial.print(b);
      Serial.print("\t c:"); Serial.print(c);
      Serial.print("\t d:"); Serial.print(d);

      Serial.print("\t e:"); Serial.print(e);
      Serial.print("\t f:"); Serial.print(f);
      Serial.print("\t g:"); Serial.print(g);
      Serial.print("\t h:"); Serial.print(h);
      Serial.println();
    }
  }
}

```

```

ThingSpeak.setField(1, a);
ThingSpeak.setField(2, b);
ThingSpeak.setField(3, c);
ThingSpeak.setField(4, d);
ThingSpeak.setField(5, e);
ThingSpeak.setField(6, f);
ThingSpeak.setField(7, g);
ThingSpeak.setField(8, h);
}
}
ThingSpeak.setStatus(myStatus);
int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
if (x == 200)
    Serial.println("Channel update successful.");
delay(100);
}

```

- **Coding Node 1**

```

#include <SPI.h>
#include "RF24.h"
double start_time;
RF24 Radio(8, 7);
byte addresses[][6] = {"1Node", "2Node"};

#include <Wire.h>
#include <MPU6050.h>
MPU6050 mpu;
long int Akselero, x, y, z, xx, yy, zz;
int nolkan_x, nolkan_y, nolkan_z, MPU;

int MAP;
float MAP2;
int nilaiTanah, cnt;
float rangetanah;
float tanah;

```

```

int saya = 1 ;
int nilaiHujan;
float rangehujan;
float hujan;

double data_kirim[6];
# define windPin 2      // Receive the data from sensor
// Constants definitions
const float pi = 3.14159265; // pi number
int period = 1000;     // Measurement period (milliseconds)
int delaytime = 1000; // Time between samples (milliseconds)
int radio = 90;       // Distance from center windmill to outer cup
(mm)
int jml_celah = 18;   // jumlah celah sensor
float anemo;
// Variable definitions
unsigned int Sample = 0; // Sample number
unsigned int counter = 0; // B/W counter for sensor
unsigned int RPM = 0;    // Revolutions per minute
float speedwind = 0;    // Wind speed (m/s)
long previousMillis = 0;
long interval = 10000;
unsigned long currentMillis;
//=====
=====
//=====
=====

void setup()
{
  delay(3000);
  Serial.begin(9600);
  pinMode(2, INPUT);
  digitalWrite(2, HIGH);
  while (!mpu.begin(MPU6050_SCALE_2000DPS,
MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check
wiring!");
    delay(500);
  }
}

```

```

}
akselero();
tampil();
nolkan_x = xx;
nolkan_y = yy;
nolkan_z = zz;
// Splash screen
Serial.println("ANEMOMETER");
Serial.println("*****");
Serial.println("Based on depoinovasi anemometer sensor");
Serial.print("Sampling period: ");
Serial.print(period / 1000);
Serial.print(" seconds every ");
Serial.print(delaytime / 1000);
Serial.println(" seconds.");
Serial.println("*** You could modify those values on code ***");
Serial.println();
Serial.println();
Radio.begin();
Radio.setPALevel(RF24_PA_LOW);
// Radio.setPALevel(RF24_PA_MAX);
/*
 * openWritingPipe (const uint8_t *address)
 * openReadingPipe (uint8_t number, const uint8_t *address)
 */
Radio.openWritingPipe(1);
// Radio.openReadingPipe(1, 1);
Radio.stopListening();
}
//=====
=====
//=====
=====

void loop()
{
  timer();
  anemo = anemometer();
  soilMoisture();
  rainDrop();
}

```



```

    akselero();
    gempa();
    tampil();
}
//=====
=====
//=====
=====

void kirim()
{
    Radio.stopListening();
    delay(100);
    data_kirim[0] = saya;
    data_kirim[1] = hujan;
    data_kirim[2] = tanah;
    data_kirim[3] = speedwind;
    data_kirim[4] = MPU;

    Radio.write(&data_kirim, sizeof(data_kirim));

    Serial.println();
    Serial.println("Kirim Data.....");
    Serial.println();

}
//=====
=====
//=====
=====

void timer()
{
    currentMillis = millis();//penempatan awal
    if (currentMillis - previousMillis > interval) //penempatan akhir
    {
        kirim();
        previousMillis = currentMillis;
    }
}
}

```

```

//=====
=====
//=====
=====
void tampil()
{
    Serial.print("saya : ");
    Serial.print(saya);
    Serial.print(" Wind speed : ");
    Serial.print(speedwind);
    Serial.print(" [m/s]");
    Serial.print("\t Kondisi tanah : ");
    Serial.print( tanah );
    Serial.print("\t Kondisi hujan : ");
    Serial.print( hujan );
    // Serial.print("\t Xraw = ");
    // Serial.print(x);
    // Serial.print("\t Yraw = ");
    // Serial.print(y);
    // Serial.print("\t Zraw = ");
    // Serial.print(z);
    Serial.print("\t Xnorm = ");
    Serial.print(xx);
    Serial.print(" Ynorm = ");
    Serial.print(yy);
    Serial.print(" Znorm = ");
    Serial.print(zz);
    Serial.print(" MPU = ");
    Serial.print(MPU);
    Serial.println();
}
//=====
=====
//=====
=====
float anemometer ()
{
    Sample++;
    // Serial.print(Sample);

```

```

// Serial.print(": Start measurement...");
windvelocity();
// Serial.println(" finished.");
// Serial.print("Counter: ");
// Serial.print(counter);
// Serial.print("; RPM: ");
RPMcalc();
// Serial.print(RPM);
// Serial.print("; Wind speed: ");
WindSpeed();
// Serial.print(speedwind);
// Serial.print(" [m/s]");
// Serial.println();
// delay(5000);
return;
}
// Measure wind speed
void windvelocity()
{
    speedwind = 0;
    counter = 0;
    attachInterrupt(0, addcount, CHANGE);
    unsigned long millis();
    long startTime = millis();
    while (millis() < startTime + period) {}
    detachInterrupt(1);
}
void RPMcalc()
{
    RPM = ((counter / jml_celah) * 60) / (period / 1000); // Calculate
    revolutions per minute (RPM)
}

void WindSpeed()
{
    speedwind = ((2 * pi * radio * RPM) / 60) / 1000; // Calculate wind
    speed on m/s
}
void addcount()

```

```

{
  counter++;
}
//=====
=====
//=====
=====
void soilMoisture()
{
  nilaiTanah = analogRead(A1);
  rangetanah = nilaiTanah * (100 / 1023.0);
  tanah = rangetanah;
  // Serial.println();
  // Serial.print("Kondisi tanah : ");
  // Serial.println( tanah );
  //
  // // Kering = 61 - 100 | Lembab = 21 - 60 | Berlumpur 0 - 20
  //
  // if (tanah >= 61 ) {
  //   Serial.println(" Kering ");
  // } else if (tanah >= 20 ) {
  //   Serial.println(" Lembab ");
  // } else {
  //   Serial.println(" Berlumpur ");
  // }
}
//=====
=====
//=====
=====
void rainDrop()
{
  nilaiHujan = analogRead(A0);
  rangehujan = nilaiHujan * (3.36 / 417);
  hujan = 8.24287 - rangehujan;
  // Serial.println();
  // Serial.print("Kondisi hujan : ");
  // Serial.println( hujan );
  //

```

```

// // Terang = 98 - 100 | Gerimis = 71 - 97 | Sedang = 53 - 70 | Deras =
0 - 52
//
// if ( hujan >= 98 ) {
//   Serial.println(" Terang ");
// } else if (hujan >= 71 ) {
//   Serial.println(" Gerimis ");
// } else if (hujan >= 53) {
//   Serial.println(" Sedang ");
// } else {
//   Serial.println(" Deras ");
// }
// delay(1000);
}
//=====
=====
//=====
=====

void gempu()
{
  // MPU = abs(xx - nolkan_x) + abs(yy - nolkan_y) + abs(zz -
nolkan_z);
  MPU = MAP;
}
void akselero()
{
  Vector rawAccel = mpu.readRawAccel();
  Vector normAccel = mpu.readNormalizeAccel();
  // x = abs(normAccel.XAxis);
  // y = abs(normAccel.YAxis);
  // z = abs(normAccel.ZAxis);
  xx = abs(normAccel.XAxis);
  yy = abs(normAccel.YAxis);
  zz = normAccel.ZAxis * 10.0;
  // int giro = zz * 10.0;
  MAP = map (zz, 100, -93, 0, 180);
  // MAP2 = MAP / 10.0;
  delay(10);
}

```

```

//=====
=====
//=====
=====
void checkSettings()
{
  Serial.println();
  Serial.print(" * Sleep Mode:      ");
  Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");
  Serial.print(" * Clock Source:      ");
  switch (mpu.getClockSource())
  {
    case MPU6050_CLOCK_KEEP_RESET:  Serial.println("Stops the
clock and keeps the timing generator in reset"); break;
    case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL
with external 19.2MHz reference"); break;
    case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL
with external 32.768kHz reference"); break;
    case MPU6050_CLOCK_PLL_ZGYRO:   Serial.println("PLL with
Z axis gyroscope reference"); break;
    case MPU6050_CLOCK_PLL_YGYRO:   Serial.println("PLL with
Y axis gyroscope reference"); break;
    case MPU6050_CLOCK_PLL_XGYRO:   Serial.println("PLL with
X axis gyroscope reference"); break;
    case MPU6050_CLOCK_INTERNAL_8MHZ:
Serial.println("Internal 8MHz oscillator"); break;
  }
  Serial.print(" * Accelerometer:    ");
  switch (mpu.getRange())
  {
    case MPU6050_RANGE_16G:         Serial.println("+/- 16 g");
break;
    case MPU6050_RANGE_8G:          Serial.println("+/- 8 g"); break;
    case MPU6050_RANGE_4G:          Serial.println("+/- 4 g"); break;
    case MPU6050_RANGE_2G:          Serial.println("+/- 2 g"); break;
  }
  Serial.print(" * Accelerometer offsets: ");
  Serial.print(mpu.getAccelOffsetX());
  Serial.print(" / ");

```



```

Serial.print(mpu.getAccelOffsetY());
Serial.print(" / ");
Serial.println(mpu.getAccelOffsetZ());
Serial.println();
}
//=====
=====
//=====
=====

```

- **Coding node 2**

```

#include <SPI.h>
#include "RF24.h"
double start_time;
RF24 Radio(8, 7);

byte addresses[][6] = {"1Node", "2Node"};

#include <Wire.h>
#include <MPU6050.h>
MPU6050 mpu;
long int Akselero, x, y, z, xx, yy, zz;
int nolkan_x, nolkan_y, nolkan_z, MPU;

int MAP;
float MAP2;
int nilaiTanah, cnt;
float rangetanah;
float tanah;
int saya = 2 ;

int nilaiHujan;
float rangehujan;
float hujan;

double data_kirim[6];
# define windPin 2 // Receive the data from sensor
// Constants definitions
const float pi = 3.14159265; // pi number

```

```

int period = 1000; // Measurement period (milliseconds)
int delaytime = 1000; // Time between samples (milliseconds)
int radio = 90; // Distance from center windmill to outer cup
(mm)
int jml_celah = 18; // jumlah celah sensor
float anemo;
// Variable definitions
unsigned int Sample = 0; // Sample number
unsigned int counter = 0; // B/W counter for sensor
unsigned int RPM = 0; // Revolutions per minute
float speedwind = 0; // Wind speed (m/s)

long previousMillis = 0;
long interval = 10000;
unsigned long currentMillis;
//=====
=====
//=====
=====

void setup()
{
  delay(3000);
  Serial.begin(9600);

  pinMode(2, INPUT);
  digitalWrite(2, HIGH);

  while (!mpu.begin(MPU6050_SCALE_2000DPS,
MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check
wiring!");
    delay(500);
  }

  akselero();
  tampil();
  nolkan_x = xx;
  nolkan_y = yy;

```

```

nolkan_z = zz;

// Splash screen
Serial.println("ANEMOMETER");
Serial.println("*****");
Serial.println("Based on depoinovasi anemometer sensor");
Serial.print("Sampling period: ");
Serial.print(period / 1000);
Serial.print(" seconds every ");
Serial.print(delaytime / 1000);
Serial.println(" seconds.");
Serial.println("*** You could modify those values on code ***");
Serial.println();
Serial.println();

Radio.begin();
Radio.setPALevel(RF24_PA_LOW);
// Radio.setPALevel(RF24_PA_MAX);
/*
 * openWritingPipe (const uint8_t *address)
 * openReadingPipe (uint8_t number, const uint8_t *address)
 */
Radio.openWritingPipe(1);
// Radio.openReadingPipe(1, 1);

Radio.stopListening();
}
//=====
=====
//=====
=====

void loop()
{
  timer();
  anemo = anemometer();
  soilMoisture();
  rainDrop();
  akselero();
  gempu();
}

```

```

    tampil();
}
//=====
=====
//=====
=====

void kirim()
{
    Radio.stopListening();
    delay(100);
    data_kirim[0] = saya;
    data_kirim[1] = hujan;
    data_kirim[2] = tanah;
    data_kirim[3] = speedwind;
    data_kirim[4] = MPU;
    Radio.write(&data_kirim, sizeof(data_kirim));
    Serial.println();
    Serial.println("Kirim Data.....");
    Serial.println();
}
//=====
=====
//=====
=====

void timer()
{
    currentMillis = millis();//penempatan awal
    if (currentMillis - previousMillis > interval) //penempatan akhir
    {
        kirim();
        previousMillis = currentMillis;
    }
}
//=====
=====
//=====
=====

void tampil()
{

```

```

Serial.print("saya : ");
Serial.print(saya);
Serial.print(" Wind speed : ");
Serial.print(speedwind);
Serial.print(" [m/s]");
Serial.print("\t Kondisi tanah : ");
Serial.print( tanah );
Serial.print("\t Kondisi hujan : ");
Serial.print( hujan );
// Serial.print("\t Xraw = ");
// Serial.print(x);
// Serial.print("\t Yraw = ");
// Serial.print(y);
// Serial.print("\t Zraw = ");
// Serial.print(z);
Serial.print("\t Xnorm = ");
Serial.print(xx);
Serial.print(" Ynorm = ");
Serial.print(yy);
Serial.print(" Znorm = ");
Serial.print(zz);
Serial.print(" MPU = ");
Serial.print(MPU);
Serial.println();
}
//=====
=====
//=====
=====

float anemometer ()
{
  Sample++;
  // Serial.print(Sample);
  // Serial.print(": Start measurement...");
  windvelocity();
  // Serial.println(" finished.");
  // Serial.print("Counter: ");
  // Serial.print(counter);
  // Serial.print("; RPM: ");

```

```

RPMcalc();
// Serial.print(RPM);
// Serial.print("; Wind speed: ");
WindSpeed();
// Serial.print(speedwind);
// Serial.print(" [m/s]");
// Serial.println();
// delay(5000);
return;
}
// Measure wind speed
void windvelocity()
{
    speedwind = 0;
    counter = 0;
    attachInterrupt(0, addcount, CHANGE);
    unsigned long millis();
    long startTime = millis();
    while (millis() < startTime + period) {}
    detachInterrupt(1);
}
void RPMcalc()
{
    RPM = ((counter / jml_celah) * 60) / (period / 1000); // Calculate
    revolutions per minute (RPM)
}
void WindSpeed()
{
    speedwind = ((2 * pi * radio * RPM) / 60) / 1000; // Calculate wind
    speed on m/s
}
void addcount()
{
    counter++;
}
//=====
=====
//=====
=====

```

```

void soilMoisture()
{
    nilaiTanah = analogRead(A1);
    rangetanah = nilaiTanah * (100 / 1023.0);
    tanah = rangetanah;
    // Serial.println();
    // Serial.print("Kondisi tanah : ");
    // Serial.println( tanah );
    //
    // // Kering = 61 - 100 | Lembab = 21 - 60 | Berlumpur 0 - 20
    //
    // if (tanah >= 61 ) {
    //     Serial.println(" Kering ");
    // } else if (tanah >= 20 ) {
    //     Serial.println(" Lembab ");
    // } else {
    //     Serial.println(" Berlumpur ");
    // }
}
//=====
=====
//=====
=====

void rainDrop()
{
    nilaiHujan = analogRead(A0);
    rangehujan = nilaiHujan * (3.36 / 417);
    hujan = 8.24287 - rangehujan;
    // Serial.println();
    // Serial.print("Kondisi hujan : ");
    // Serial.println( hujan );
    //
    // // Terang = 98 - 100 | Gerimis = 71 - 97 | Sedang = 53 - 70 | Deras =
    0 - 52
    //
    // if ( hujan >= 98 ) {
    //     Serial.println(" Terang ");
    // } else if (hujan >= 71 ) {
    //     Serial.println(" Gerimis ");

```

```

// } else if (hujan >= 53) {
//   Serial.println(" Sedang ");
// } else {
//   Serial.println(" Deras ");
// }
// delay(1000);
}
//=====
=====
//=====
=====

void gempa()
{
  // MPU = abs(xx - nolkan_x) + abs(yy - nolkan_y) + abs(zz -
nolkan_z);
  MPU = MAP;
}
void akselero()
{
  Vector rawAccel = mpu.readRawAccel();
  Vector normAccel = mpu.readNormalizeAccel();
  // x = abs(normAccel.XAxis);
  // y = abs(normAccel.YAxis);
  // z = abs(normAccel.ZAxis);
  xx = abs(normAccel.XAxis);
  yy = abs(normAccel.YAxis);
  zz = normAccel.ZAxis * 10.0;
  // int giro = zz * 10.0;
  MAP = map (zz, 100, -93, 0, 180);
  // MAP2 = MAP / 10.0;
  delay(10);
}
//=====
=====
//=====
=====

void checkSettings()
{
  Serial.println();
}

```



```

Serial.print(" * Sleep Mode:      ");
Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

Serial.print(" * Clock Source:    ");
switch (mpu.getClockSource())
{
  case MPU6050_CLOCK_KEEP_RESET:  Serial.println("Stops the
clock and keeps the timing generator in reset"); break;
  case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL
with external 19.2MHz reference"); break;
  case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL
with external 32.768kHz reference"); break;
  case MPU6050_CLOCK_PLL_ZGYRO:   Serial.println("PLL with
Z axis gyroscope reference"); break;
  case MPU6050_CLOCK_PLL_YGYRO:   Serial.println("PLL with
Y axis gyroscope reference"); break;
  case MPU6050_CLOCK_PLL_XGYRO:   Serial.println("PLL with
X axis gyroscope reference"); break;
  case MPU6050_CLOCK_INTERNAL_8MHZ:
Serial.println("Internal 8MHz oscillator"); break;
}

Serial.print(" * Accelerometer:   ");
switch (mpu.getRange())
{
  case MPU6050_RANGE_16G:         Serial.println("+/- 16 g");
break;
  case MPU6050_RANGE_8G:          Serial.println("+/- 8 g"); break;
  case MPU6050_RANGE_4G:          Serial.println("+/- 4 g"); break;
  case MPU6050_RANGE_2G:          Serial.println("+/- 2 g"); break;
}

Serial.print(" * Accelerometer offsets: ");
Serial.print(mpu.getAccelOffsetX());
Serial.print(" / ");
Serial.print(mpu.getAccelOffsetY());
Serial.print(" / ");
Serial.println(mpu.getAccelOffsetZ());

```

```
Serial.println();  
}  
//=====
```

```
=====
```

```
//=====
```

```
=====
```