

SISTEM PENGAMANAN LAHAN KELAPA SAWIT MENGUNAKAN MULTI-SENSOR DIOPERSIKAN JARAK JAUH

Morisius Riyo¹, Jannus Marpaung², Neilcy Tjahjamoonsih³, Redi Ratiandi Yacoub⁴,
Leonardus Sanddy Ade Putra⁵

morisius@student.untan.ac.id¹, jannus.marpaung@ee.untan.ac.id²,
neilcy.mooniarsih@ee.untan.ac.id³, rediyacoub@ee.untan.ac.id⁴, leonardusandy@ee.untan.ac.id⁵
Universitas Tanjungpura Pontianak

ABSTRAK

Dalam lahan kelapa sawit sensor suara bisa ditempatkan disekitar perimeter lahan atau ditempat strategis misalnya didekat akses jalan masuk, ketika ada aktivitas suara yang mencurigakan sistem ini. deteksi sensor suara (db, waktu dalam menentukan kebutuhan meskipun dalam deteksi suara ada durasi yang mencakup semua). Sensor gerak (motion sensor) sensor ini digunakan untuk mendeteksi pergerakan yang terjadi dilahan yang mencurigakan didaerah yang rawan pencurian agar dapat mencegah terjadinya pencurian. Alat ini sering kali berbasis inframerah pasif (PIR). Sensor berat (Load Sensor) tidak mendeteksi perubahan berat padat terjadinya kendaraan yang keluar masuk dilokasi. Kombinasi multi-sensor secara lebih efektif jika diintegrasikan dengan teknologi modern seperti sistem LoRa (Long Range) sensor-sensor ini dapat dikoneksikan ke pusat pengendalian melalui LoRa sehingga informasi bisa dikirim secara real-time ke penjaga atau pemilik lahan.

Kata Kunci: Keamanan Lahan Sawit, Multi-Sensor, Internet Of Things, LoRa.

ABSTRACT

In oil palm plantations, sound sensors can be placed around the perimeter of the land or in strategic places, for example near the access road, when there is suspicious sound activity, this system. Sound sensor detection (db, time in determining needs even though in sound detection there is a duration that covers all). Motion sensor This sensor is used to detect movements that occur on suspicious land in areas prone to theft in order to prevent theft. This tool is often based on passive infrared (PIR). The weight sensor (Load Sensor) does not detect changes in the weight of solid vehicles entering and leaving the location. The combination of child multisensors is more effective if integrated with modern technology such as the LoRa (Long Range) system. These sensors can be connected to the control center via LoRa so that information can be sent in real time to the guard or land owner.

Keywords: Palm Oil Land Security, Multi-Sensor, Internet Of Things, LoRa.

PENDAHULUAN

Keamanan lahan kelapa sawit sering kali menjadi perhatian penting bagi para pemilik lahan dan pengelola. Terutama karena kerentanan terhadap pencurian hasil panen, pembalakan liar, atau kerusakan lingkungan. Beberapa faktor yang menjadi latar belakang penting untuk pengembangan sistem pengaman lahan sawit meliputi Kejahatan di Lahan Sawit. Lahan sawit rentan terhadap berbagai jenis kejahatan, termasuk pencurian hasil panen, pembalakan liar, dan penambangan ilegal. Dapat menyebabkan kerugian finansial yang signifikan bagi pemilik lahan dan pengelola.

Keterbatasan Pengawasan Manusia Pengawasan lahan sawit dengan menggunakan tenaga manusia saja sering kali tidak memadai untuk mencakup area yang luas secara efektif, terutama di lahan-lahan yang terpencil atau sulit dijangkau. Dibutuhkan solusi teknologi untuk meningkatkan efisiensi dan efektivitas pengawasan.

Tuntutan Regulasi dan Sosial Pemilik lahan dan pengelola sawit juga dihadapkan pada tuntutan untuk mematuhi regulasi lingkungan dan sosial yang semakin ketat [4]. Meningkatkan keamanan lahan sawit dapat membantu memenuhi persyaratan ini dan mendukung praktik pertanian berkelanjutan. Dengan memahami latar belakang ini,

pengembangan sistem pengaman lahan sawit yang menggunakan teknologi sensor dan komunikasi jarak jauh dapat menjadi langkah penting dalam meningkatkan keamanan, produktivitas, dan keberlanjutan industri sawit secara keseluruhan.

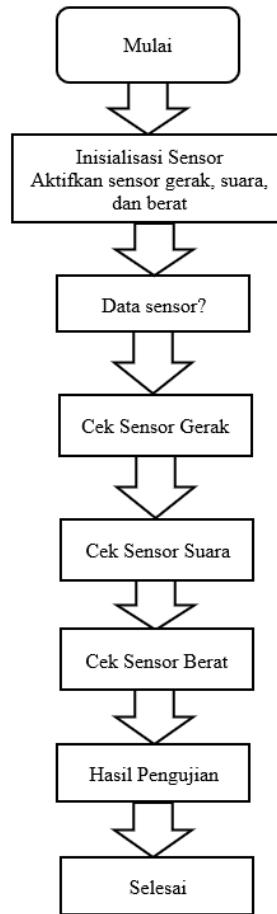
HASIL DAN PEMBAHASAN

1. Materi Pengujian

Pada setiap sensor diuji secara individu, kemudian dilakukan pengujian integrasi dengan menghubungkan semua sensor ke sistem utama yaitu IoT. Data dari masing-masing sensor bisa dikombinasikan untuk analisis yang lebih akurat. Penelitian ini data-data sensor gerak, sensor suara dan sensor berat dari berapa skenario pengujian yang dilakukan. Sebelum melakukan pengujian keseluruhan, terlebih dahulu diuji perangkat individual sensor, pengujian komunikasi LoRa dari ke tiga sensor merupakan keamanan lahan kelapa sawit. Selanjutnya data-data yang dihasilkan dibahas untuk mendapatkan kinerja dari setiap sensor.

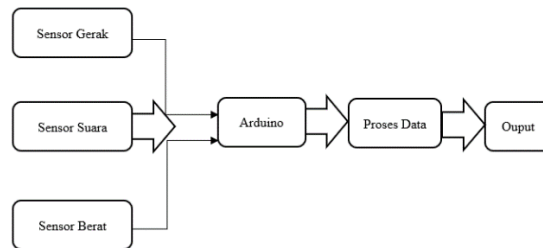
1) Mekanisma Pengujian Sensor

Berdasarkan pengujian yang dilakukan terhadap ketiga sensor (gerak, suara dan berat) dalam sistem keamanan lahan kelapa sawit. Sensor gerak cukup efektif untuk mendeteksi intrusi diarean perkebunan, tapi perlu dikombinasikan dengan algoritma pemrosesan data untuk mengurangi false alarm. Pada pemasangan sensor di ketinggian yang optimal (sekitar 1,5 - 2 meter) dapat meningkatkan akurasi dan mengurangi gangguan dari hewan kecil atau dedaunan yang bergerak. Pada sensor suara sangat berguna untuk mendeteksi aktivitas ilegal, terutama saat pencurian buah kelapa sawit menggunakan alat berat atau kendaraan. Penggunaan filter noise untuk mengurangi gangguan dari suara angin dan hujan kombinasi dengan sensor lain (misalnya gerak) dapat meningkatkan akurasi deteksi. Sensor berat mampu mendeteksi beban berlebihan seperti kendaraan yang melintas atau tandan buah sawit yang dicuri akurasi pengukuran cukup lebih tinggi tetapi sensor perlu dipasang ditempat strategis seperti jalan masuk keperkebunan atau jalan keluar kearah pertambangan buah sawit perlu sistem pemilihan sensor secara rutin untuk memastikan sensor tidak tertutup lumpur atau rusak akibat cuaca ekstrim. Pengujian data sensor (gerak, suara dan berat) dalam diringkas digambarkan alir flowchart seperti ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Pengujian ke Tiga Sensor (gerak, suara, dan berat)

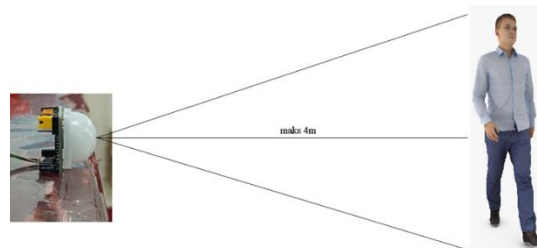
Gambar 1 merupakan diagram alir yang menunjukkan bagaimana dari kerja dari sensor (gerak suara dan berat) hingga bisa menghasilkan data. Pertama dimulai ini adalah titik awal sebelum sensor diaktifkan dan diuji pada tahap ini semua sensor (gerak, suara, dan berat) diaktifkan agar siap membaca data dari lingkungan sekitar. Sistem membaca data dari ketiga sensor yang telah diaktifkan setiap sensor mulai mengukur parameter yang sesuai sensor gerak mendeteksi adanya pergerakan, sensor suara mendeteksi suara disekitar sensor berat mengukur berat atau suatu objek. Cek sensor gerak sistem ini untuk memeriksa apakah sensor gerak mendeteksi pergerakan jika YA, hasil pengujian dicatat atau ditampilkan dan jika TIDAK lanjut ke pemeriksaan sensor suara. Cek sensor suara jika ada pergerakan sistem memeriksa sensor suara jika YA hasil pengujian dicatat atau ditampilkan jika TIDAK lanjut pemeriksaan sensor berat. Cek sensor berat jika tidak ada deteksi suara sistem memeriksa apakah ada perubahan pada sensor berat jika YA hasil pengujian dicatat atau ditampilkan jika TIDAK sistem kembali membaca data dari awal. Tampilkan hasil pengujian setelah semua sensor diuji sistem menampilkan hasil pengujian berdasarkan deteksi dari ketiga sensor. Proses pengujian selesai dan sistem siap untuk digunakan kembali atau melakukan pengujian ulang. Diagram Blok pembaca sensor (berat, gerak, dan berat) Ditujukan pada Gambar 2.



Gambar 2. Diagram Blok Pengujian Sensor (gerak, suara, dan berat)

2. Pengujian Sensor Gerak

Pergerakan benda berada di depan sensor, pengujian yang dilakukan di laboratorium pergerakan benda dilakukan oleh manusia, dapat berupa pergerakan badan ataupun pergerakan tangan. Sensor dapat menangkap pergerakan benda sejauh 4 m yang berada di depannya. Yang dimaksud di depan sensor tidak hanya berada tegak lurus dengan sensor, melainkan di samping sensor masih bisa terdeteksi.



Gambar 3. Pengujian Sensor Gerak

Pengujian pergerakan di depan sensor dilakukan pada jarak 1m, 2m, 3m dan 4m. Pada setiap jarak dilakukan sebanyak 10 kali dan hasilnya ditunjukkan pada Tabel 4.1.

Tabel 1. Hasil Pengujian Sensor Gerak (PIR)

Pengujian ke	Jarak Sensor ke Objek Bergerak			
	1 m	2m	3m	4m
1	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
2	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
3	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
4	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
5	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
6	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
7	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
8	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
9	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!
10	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!	Motion detected! Motion ended!

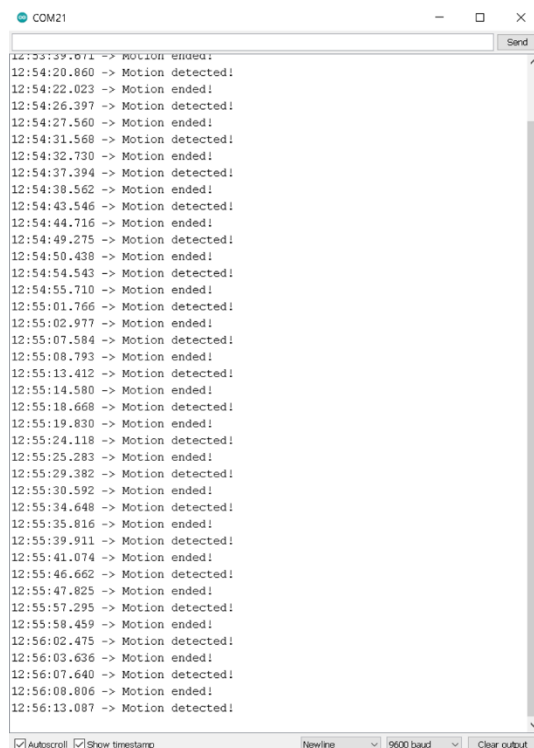
Dari percobaan 10 kali pergerakan di depan sensor dengan jarak yang diberikan objek yang bergerak terdeteksi dengan komentar Motion detected! Dan diakhiri dengan Motion ended!.

Koding program untuk sensor gerak adalah sebagai berikut:

```
#define pirPin 4
int val = 0;
bool motionState = false;
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(pirPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  val = digitalRead(pirPin);
  if (val == HIGH)
  {
    digitalWrite(LED_BUILTIN, HIGH);
    if (motionState == false)
    {
      Serial.println(" Motion detected!");
      motionState = true;
    }
  }
  else
  {
    digitalWrite(LED_BUILTIN, LOW);
    if (motionState == true)
    {
      Serial.println(" Motion ended!");
      motionState = false;
    }
  }
}
```

Hasil selengkapnya ditunjukkan pada tangkapan layar seperti ditunjukkan pada Gambar 4.



Gambar 4. Hasil Tangkapan Layar Pergerakan Objek pada Di depan Sensor Gerak

3. Pengujian Sensor Berat

Sensor berat yang digunakan adalah load cell dengan beban maksimum 5kg dengan modul driver HX711. Nilai berat yang dihasilkan sensor dikalibrasi dengan timbangan dengan ketelitian tinggi. Namun, beban maksimum timbangan tersebut adalah 500 gram. Sehingga skala perbandingan sensor berat terhadap alat kalibrasi adalah 1:10.

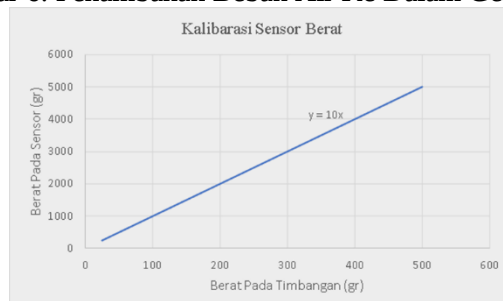
Beban yang digunakan adalah air yang dimasukkan pada gelas ukur standart. Mula-mula berat yang ditimbang adalah 25 gr kemudian diletakkan di atas sensor dan hasilnya dibaca pada serial monitor. Peneliti mengusahakan kenaikan berat 25 gr namun hasilnya berbeda sedikit saat ditimbang, namun hasil pembacaan pada timbangan selalu sama seperti pada layar serial monitor.



Gambar 5. Penggunaan Gelas Ukur Standart dan Timbangan Presisi



Gambar 6. Penambahan Beban Air Ke Dalam Gelas Ukur



Gambar 7. Penambahan Beban Air Ke Dalam Gelas Ukur

Tabel 2. Data Sensor Berat

Timbangan	Sensor Berat	
	gr	kg
25	250	0.25

51	510	0.51
76	760	0.76
99	990	0.99
124	1240	1.24
150	1500	1.5
175	1750	1.75
202	2020	2.02
227	2270	2.27
252	2520	2.52
276	2760	2.76
301	3010	3.01
325	3250	3.25
352	3520	3.52
375	3750	3.75
400	4000	4
425	4250	4.25
450	4500	4.5
475	4750	4.75
500	5000	5

Grafik hubungan antara berat pada timbangan dan sensor ditunjukkan pada Gambar 4.

Model matematis untuk kalibrasi sensor adalah berupa persamaan garis linier (lurus) dengan kemiringan positif 10.

$$y=10x$$

Dimana x adalah berat yang ditunjukkan pada timbangan sedangkan y adalah berat yang ditampilkan oleh sensor.

Selanjutnya, untuk penelitian ini nilai sensor diskalakan sehingga nilainya menjadi tonase atau ton. Untuk berat yang tertimbang 100gr dinyatakan sebagai x_1 sama dengan 1000000 gr (1 ton) dinyatakan sebagai y_1 , sedangkan untuk 500gr dinyatakan sebagai x_2 sama dengan 12000000 gr (12 ton) dinyatakan sebagai y_2 . Dengan menggunakan interpolasi linier diperoleh model matematika hasil penskalaan sensor sebagai berikut:

$$y = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) + y_1$$

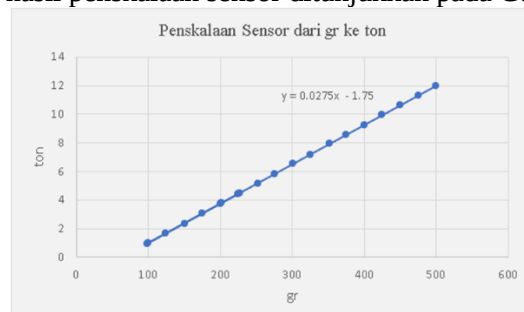
$$y = \frac{12000000 - 1000000}{500 - 100} (x - 100) + 1000000$$

$$y = 27500x - 1750000 \text{ dalam satuan gram}$$

Dalam satuan ton

$$y = 0,0275x - 1,75$$

Grafik hasil penskalaan sensor ditunjukkan pada Gambar 8.



Gambar 8. Penskalaan Sensor Berat dari gram ke ton

Potongan kodung program untuk panskalaan sensor berat adalah sebagai berikut:

```
#include "HX711.h"
#define DOUT 3
#define CLK 2
HX711 scale;
float calibration_factor = -398; //Hasil Kalibrasi
float units;
void setup()
{
  Serial.begin(9600);
  scale.begin(DOUT, CLK);
  scale.set_scale();
  scale.tare(); //Reset the scale to 0
  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: ");
  Serial.println(zero_factor);
}
void loop()
{
  scale.set_scale(calibration_factor); //Adjust to this calibration factor
  Serial.print("Reading: ");
  units = scale.get_units(1);
  float x=units-221.73; //pengurangan berat gelas ukur
  float y=0.0275*x+1.75;
  if (y < 0)
  {
    y = 0.00;
  }
  Serial.print("Real Berat: ");
  Serial.print(x );
  Serial.print(" Gram");
  Serial.println();
  Serial.print("Skala Berat: ");
  Serial.print(y);
  Serial.print(" ton");
  Serial.println();
  delay(1000);
}
```

4. Pengujian Sensor Suara

Berikut ini ditunjukkan hasil pengujian sensor suara.

Tabel 3. Data Hasil Dari Sensor Suara

Waktu (detik)	Nilai Sensor Suara	Keterangan
0	320	Kondisi normal (background noise)
1	330	Tidak ada suara signifikan
2	480	Suara terdeteksi (misakan tepuk tangan)
3	500	Puncak intensitas suara
4	450	Suara mulai mereda
5	340	Kondisi normal

Pada Gambar 7. Data sensor suara nilai yang dibaca sensor berada dala, rentang 0-1023 nilai yang lebih tinggi waktu pengukuran dalam data diambil setiap detik sehingga dapat diperoleh memperoleh hasil dari sensor suara berubah seiring waktu. Pada detik 0 dan 1 nilai relatif stabil menunjukan tidak ada suara yang sigingifkan selain backgrund noise. Dan terjadi pada detik 2 dan 3 terjadinya peningkatan nilai yang mengindikasikan adanya suara atau kejadian yang menibukan penikatkan intensitas suara setelah nilai menurun secara bertahap hingga kembali ke level normal mendakan suara mereda data seperti ini berguna untuk aplikasi yang memerlukan deteksi analisis suara.

5. Pengujian Komunikasi LoRa Point To Point

Komunikasi point to point berarti ada dua perangkat LoRa berkomunikasi langsung atau sama yang lain tanpa perantara seperti gateway atau infrastruktur jaringan lainnya. Ada dua node node A dan node B saling bertukar data secara langsung konfigurasi ini cocok untuk jarak antar perangkat cukup jauh dan tidak diperlukan infrastruktur jaringan kompleks.

1) Koding Pengirim Data Sensor Dengan Komunikasi Point To Point

Modul pengirim segera mengirimkan data sensor sekaligus dengan data sensor gerak, sensor suara dan sensor berat data yang dikirimkan melalui LoRa adalah serangkaian data dalam format teks atau angka. Data data dikirimkan dipisahkan dengan menyisipkan karakter khusus dalam hal ini digunakan larakter % format data pemancar string data = Data1 1% Data 2% dan Data 3%..... Sampai dengan data akhir. Data-data sensor yang dikirimkan melalui pemancar sistem keamanan seperti sensor gerak, sensor suara dan sensor berat. Koding program terdiri dari koding master (penerima) dan slave (pemancar data sensor), komunikasi master dan slave didasarkan pada komunikasi half duplex. Master mengirimkan permintaan data, selanjutnya slave memberikan respon dengan memancar data sensor-sensor.

Koding Pemancar

```
#include <SPI.h>
#include <LoRa.h>
#include <DHT.h>
#include "HX711.h"
HX711 scale;
float CallF=400;
float units;
#define ss 5
#define rst 14
#define dio0 2
#define SoundData 12
#define SoundLED 13
#define SoundAnalog 15
#define PIRData 26
#define PIRLED 27
#define LoadCellData 33
#define LoadCellCLK 34
#define LoadCellLED 35
#define LoRaLED 4
int val =0;
bool motionState =false;
boolean valD=0;
String outgoing;
byte msgCount = 0;
byte Slave1ID = 0xAA;
byte Master= 0xFF;
long lastSendTime = 0;
int interval = 2000;
String DataSlave1="";
int SF=12;
int CR=5;
String CR1="";
long int BW=250E3;
long int freq=868E6;

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

  scale.begin(LoadCellData,LoadCellCLK);
  scale.set_scale();
  scale.tare();
  long zero_factor=scale.read_average();
  Serial.println("LoRa Duplex");
  LoRa.setPins(ss, rst, dio0);
  if (!LoRa.begin(freq))
  {
    Serial.println("LoRa init failed. Check
your connections.");
    while (true);
  }
  LoRa.setSpreadingFactor(SF);
  LoRa.setCodingRate4(CR);
  LoRa.setSignalBandwidth(BW);
  Serial.println("LoRa init succeeded.");
}

void loop()
{
  onReceive(LoRa.parsePacket());
}

void sendMessage(String outgoing, byte
Master, byte Slave1ID)
{
  LoRa.beginPacket();
  LoRa.write(Master);
  LoRa.write(Slave1ID);
  LoRa.write(msgCount);

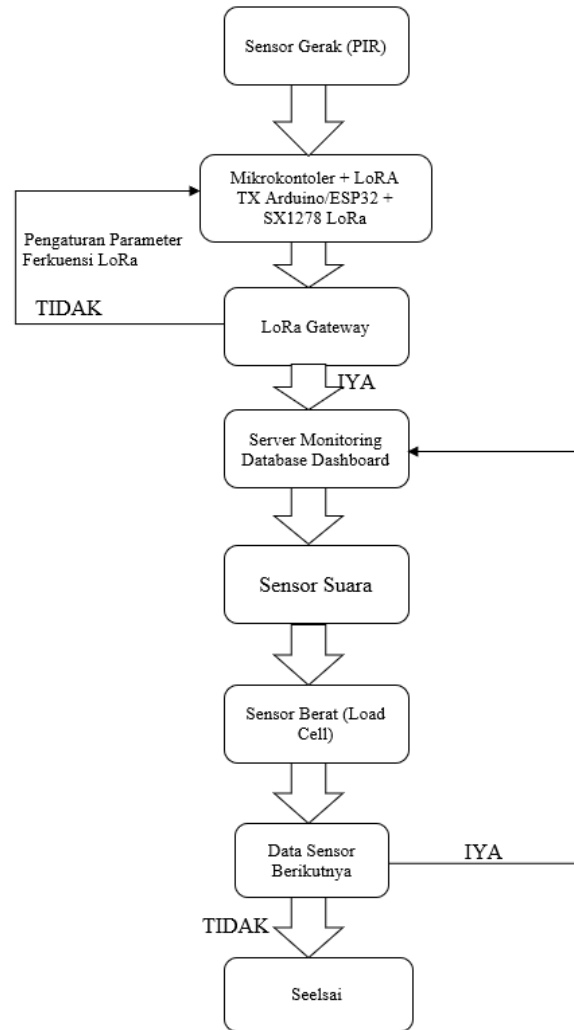
  if (motionState==false)
  {
    motionState=true;
  }
}
else
{
  digitalWrite(PIRLED,LOW);
  PIRs=0;
  if (motionState==true)
```

```

    byte sender = LoRa.read();
    byte incomingMsgId = LoRa.read();
    byte incomingLength = LoRa.read();
    while (LoRa.available())
    {
        incoming += (char)LoRa.read();
    }
    Serial.println(incoming);
    int Val = incoming.toInt();
    if(Val == 33)
    {
        Data_Slave1();
        String message =DataSlave1;
        Serial.println("Sending: " + DataSlave1);
        sendMessage(message,Master,Slave1ID);
        delay(100);
        DataSlave1="";
    }
    Serial.println("RSSI: " + String(LoRa.packetRssi()));
    Serial.println("Snr: " + String(LoRa.packetSnr()));
    Serial.println();
}

void Data_Slave1()
{
    val=digitalRead(PIRData);
    int PIRs=0;
    if(val==HIGH)
    {
        digitalWrite(PIRLED,HIGH);
        PIRs=1;
    }
    {
        motionState=false;
    }
    }
    int Sounds=0;
    valD=digitalRead(SoundData);
    int valA=analogRead(SoundAnalog);
    if(valA<=513)
    {
        digitalWrite(SoundLED,HIGH);
        Sounds=1;
    }
    else
    {
        digitalWrite(SoundLED,LOW);
        Sounds=0;
    }
    scale.set_scale(CallF);
    units=scale.get_units(),1;
    if(units<0)
    {
        units=0.00;
    }
    Serial.println();
    Serial.println(units);
    DataSlave1=String("1")+String("$")+
    String("2000")+String("$")+String(PIRs)+
    String("$")+String(Sounds);
}

```



Gambar 9. Diagram Alir Koding Pemancaran Data Sensor

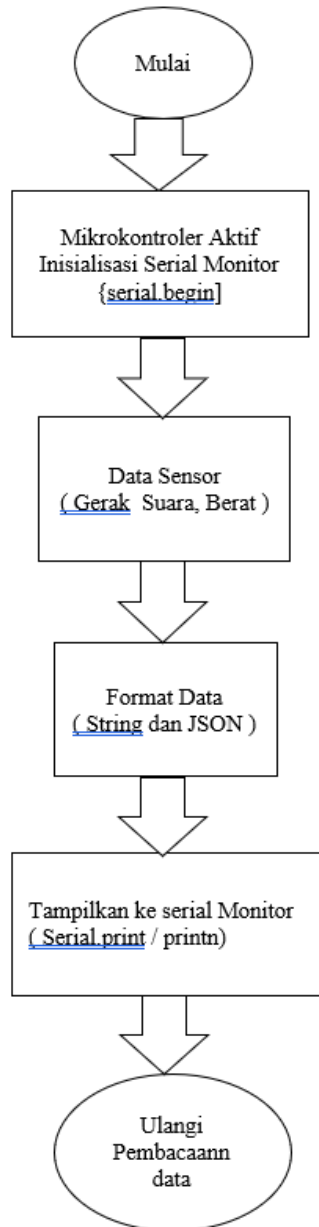
Gambar 9. merupakan diagram alir format pengiriman sensor yang menunjukan Proses pengiriman data yang telah didapatkan dari sensor dimulai dengan memasukkan LoRa TX dan RSSI, SNR kemudian variabel dari sensor suara dan berat. Setelah itu pengambilan data dari setiap sensor sehingga bisa ditampilkan pada serial monitor.

6. Koding Pada Penerima

LoRa menerima data berupa serangkaian teks (string) sesuai dengan format data dari pengirim. Data dengan tipe string tidak dapat dikenali dan digunakan untuk ditampilkan pada platform Thingier.io diuraikan dan dipisahkan serta dimasukkan pada variabel-variabel yang ditetapkan sehingga dapat dipisahkan.

Rangkaian yang perlu dilakukan data LoRa yang diterima dimasukkan ke dalam variabel array bernama `buff[i]`. `i` adalah variabel angka integer mulai dari 0 sampai dengan fungsi `delimite`. Setiap data yang sudah dipisahkan dimasukkan ke dalam array data yang disebut `arrLoRadata[index]` dimana `index` adalah variabel integer dengan nilai mulai dari 0 sampai dengan data yang diperoleh.

Pada sensor yang digunakan di titik 1 2 3 dan 4 merupakan output yang dikenali saat pembuatan widget pada platform Thingier.io pada parameter komunikasi LoRa antara pemancar dan penerima yang ditampilkan pada platform IoT. Komunikasi LoRa digunakan standar perintah LoRa dan disimpan dalam variabel `Nilai_RSSI` dan `Nilai_SNR1`.



Gambar 10. Diagram Alir Penampilan Data Pada Serial Monitor Dan Thingier.io.

```

#include <SPI.h>
#include <LoRa.h>
#define ss 15
#define rst 16
#define dio0 4
int SF=12;
int CR=5;
long int BW=250E3;
long int freq=868E6;
byte msgCount = 0;
byte localAddress = 0xBB;
byte destination = 0xFF;
long lastSendTime = 0;
int interval = 2000;
String arrayData[50];
String Tx, Rx, DataSF,DataCR,DataBW;
String Datafreq, Berat, Sound, PIR;
float Nilai_RSSI,Nilai_SNR;

String message = "Minta Data dari
Pemancar";
sendMessage(message);
Serial.println("Sending: " + message);
lastSendTime = millis();
interval = random(2000) + 1000;
}
onReceive(LoRa.parsePacket());
thing.handle();
}
void sendMessage(String outgoing)
{
  LoRa.beginPacket();
  LoRa.write(destination);
  LoRa.write(localAddress);
  LoRa.write(msgCount);
  LoRa.write(outgoing.length());
  LoRa.print(outgoing);

```

```

void setup()
{
  Serial.begin(115200);
  while (!Serial);
  LoRa.setPins(ss, rst, dio0);
  if (!LoRa.begin(freq))
  {
    Serial.println("LoRa init failed");
    while (true);
  }
  LoRa.setSpreadingFactor(SF);
  LoRa.setCodingRate4(CR);
  LoRa.setSignalBandwidth(BW);
  Serial.println("LoRa init succeeded.");
  thing.add_wifi(SSID, SSID_PASSWORD);
  Serial.println("Thing Connect!");
  thing["Data1"] >>[](pson& out)
  {
    out["Berat"] = Berat;
    out["Sound"] = Sound;
    out["PIR"] = PIR;
    out["SF"] = DataSF;
    out["CR"] = DataCR;
    out["BW"] = DataBW;
    out["freq"] = Datafreq;
    out["RSSI"] = Nilai_RSSI;
    out["SNR1"] = Nilai_SNR;
    out["Pemancar"] = Rx;
    out["Penerima"] = Tx;
  };
}

void loop()
{
  if (millis() - lastSendTime > interval)
  {
    if(body!="")
    {
      int index =0;
      for(int i2=0;i2<body.length();i2++)
      {
        char delimiter ='@';
        if(body[i2] !=delimiter)
          arrayData[index]+=body[i2];
        else
          index++;
      }
      if(index==10)
      {

      }
    }
    Serial.println("Berat      : "+arrayData[0]);
    Serial.println("Sound      : "+arrayData[1]);
    Serial.println("PIR        : "+arrayData[2]);
    Serial.println("DataSF     : "+arrayData[3]);
    Serial.println("DataCR     : "+arrayData[4]);
    Serial.println("DataBW     : "+arrayData[5]);
    Serial.println("Datafreq   : "+arrayData[6]);
    Serial.println("Nilai_RSSI :

    LoRa.endPacket();
    msgCount++;
    Tx=String(localAddress,HEX);
    Rx=String(destination,HEX);
  }
  void onReceive(int packetSize)
  {
    if (packetSize == 0) return;
    int recipient = LoRa.read();
    byte sender = LoRa.read();
    byte incomingMsgId = LoRa.read();
    byte incomingLength = LoRa.read();
    String incoming = "";
    String Dat[100];
    if (packetSize)
    {
      int i=0;
      while (LoRa.available())
      {
        Dat[i] = (char)LoRa.read();
        i++;
      }
      int k = i;
      String body ="";
      for(i= 0;i<k;i++)
      {
        body = body+Dat[i];
      }
      Serial.println("Data yang diterima:
      "+String(body));
      body = body.substring(0);
      BB=body.length();
      Serial.print("Panjang Data: ");
      Serial.println(BB);
      body.trim();

```

```

"+String(LoRa.packetRssi());
  Serial.println("Nilai_SNR :
"+String(LoRa.packetSnr());
  Serial.println();
  Berat      = arrayData[0];
  Sound      = arrayData[1];
  PIR        = arrayData[2];
  DataSF     = arrayData[3];
  DataCR     = arrayData[4];
  DataBW     = arrayData[5];
  Datafreq   = arrayData[6];
  Nilai_RSSI = LoRa.packetRssi();
  Nilai_SNR  = LoRa.packetSnr();
  delay(100);
  arrayData[0]= "";
  arrayData[1]= "";
  arrayData[2]= "";
  arrayData[3]= "";
  arrayData[4]= "";
  arrayData[5]= "";
  arrayData[6]= "";
}
}

```

7. Pengujian Sistem Pengamanan Sawit

Pengujian dilakukan prototype, meliputi 3 buah slave (lokasi berbeda) dan 1 buah master. Pada master dilengkapi dengan platform Iot thinger.io. Untuk dapat terhubung ke platform maka mikrokontroler terlebih dahulu melakukan koneksi dengan kanal wifi setempat, bisa dengan wifi Untan ataupun mode teathering menggunakan smartphone.

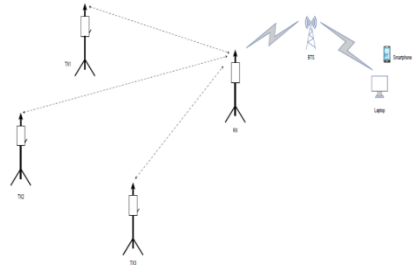
Untuk menampilkan data-data yang menarik digunakan fasilitas pembuatan widget pada platform thinger.io Pembuatan widget pada platform Thingier.io mulai dengan mendaftar atau subscribe pada platform dengan menyertakan username dan password. Setelah terhubung segera diperoleh konfirmasi pada email. Setelah terkonfirmasi maka dilakukan login pada platform dan selanjutnya membuat device dan dashboard.

Pembuatan device dibatasi sebanyak 2 device sedangkan pembuatan dashboard sebanyak 4 dashboard. Device yang dibuat disebut DEVICE_ID dengan nama Morisius Riyo dengan DEVICE_CREDENTIAL adalah mendeteksi pergerakan manusia dan kendaraan sedangkan nama dashboard yang dibuat adalah mendeteksi pergerakan manusia dan kendaraan.

Widget dibuat dalam dashboard oleh karena 4 data yang segera ditampilkan maka dibuat 6 widget dengan tipe grafis Tachometer setiap widget harus terhubung dengan nama dashboard dan nama-nama variabel dalam dashboard yang ditampilkan data-data sensor dan parameter komunikasi LoRa ditujukan pada Gambar 4.12.

1) Arsitektur Sistem

Sistem yang dibangun terdiri dari 3 pemancar dan 1 penerima, jaringan WiFi setempat ataupun BTS (base transceiver station) dan penampil pada layar laptop. Pemancar diatur pada jarak jauh antara 50 sampai dengan 150m. Pengujian dilakukan 2 kali, yaitu di sekitar halaman Perpustakaan Untan dan pada koridor di sekitar Fakultas Teknik Untan.



Gambar 12. Penskalaan Sensor Berat dari gram ke ton

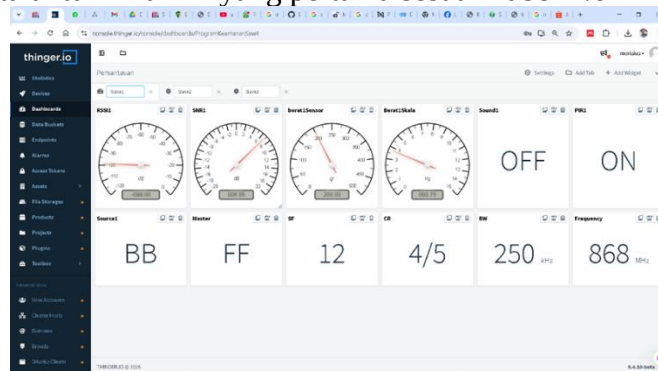
Untuk memberikan gambaran kerja sistem, pada setiap pemancar diberikan nilai-nilai yang sudah diuji pada lab, yang diringkaskan pada Tabel 4.6 berikut:

Tabel 4. Nilai-Nilai Sensor Yang Diimputkan pada Setiap Pemancar

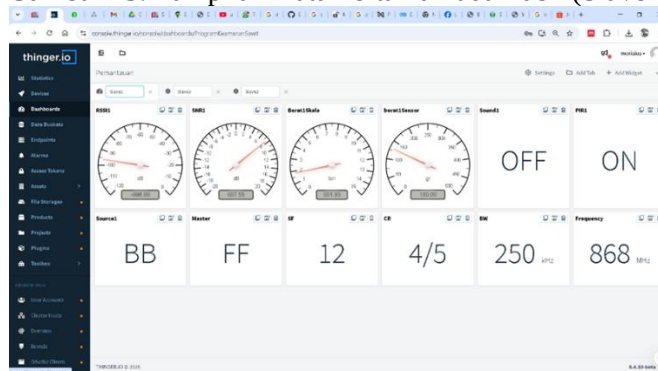
TX1			TX2			TX3		
Berat1 (ton)	Sound1	PIR1	Berat2 (ton)	Sound2	PIR2	Berat3 (ton)	Sound3	PIR3
3.75	OFF	ON	11.31	OFF	ON	1.83	OFF	ON
9.25	ON	ON	10.63	ON	ON	11.45	ON	ON
5.18	ON	OFF	7.19	ON	OFF	0.45	ON	OFF

2) Tampilan Data Pada Platform IoT

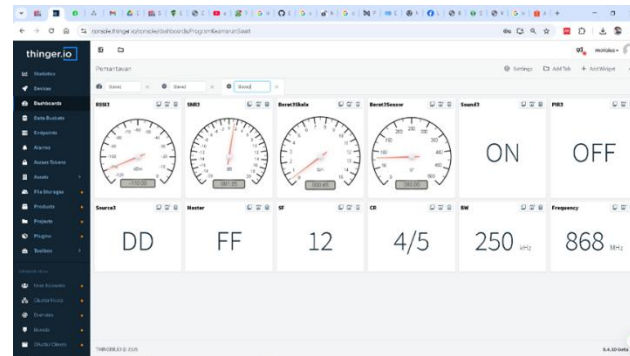
Setelah koding program di upload pada masing-masing perangkat dan pemancar Tx (slave) diletakkan pada jangkauan 50m-150m terpisah dengan Rx (master) dan pembuatan widget pada platform sudah rampung untuk setiap variabel yang ditetapkan maka didapatkan data-data untuk nilai Tx yang pertama sesuai Tabel 4.6



Gambar 13. Tampilan Data Pertama Pada Tab1 (Slave1)

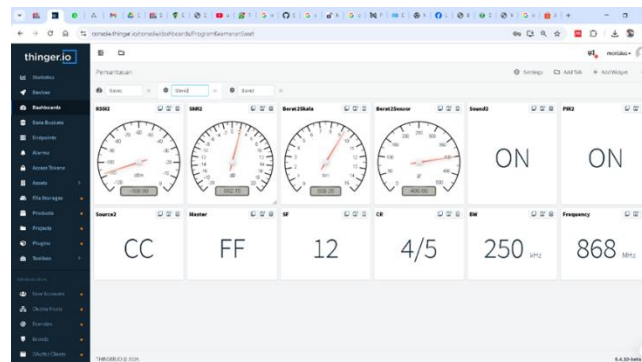


Gambar 14. Tampilan Data Pertama Pada Tab2 (Slave2)

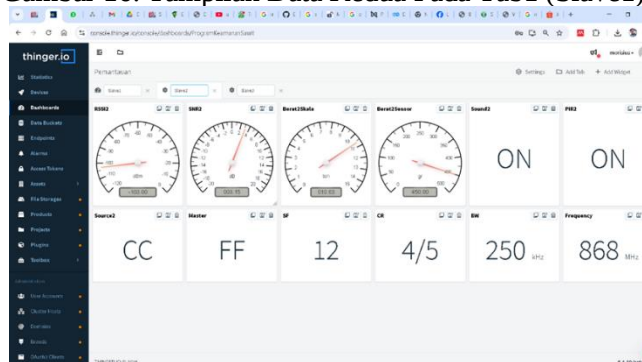


Gambar 15. Tampilan Data Pertama Pada Tab3 (Slave3)

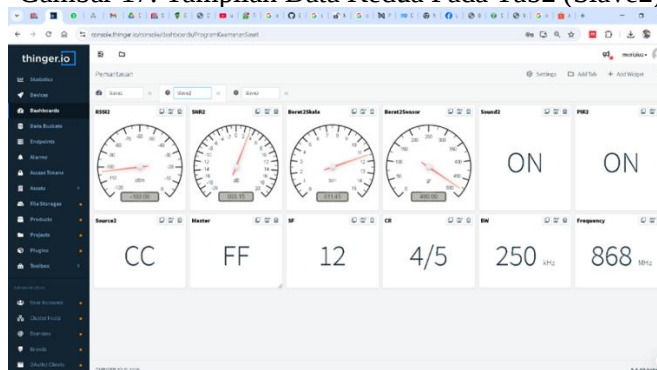
Untuk data kedua yang diinputkan pada setiap slave diperoleh tampilan data-data sebagai berikut:



Gambar 16. Tampilan Data Kedua Pada Tab1 (Slave1)

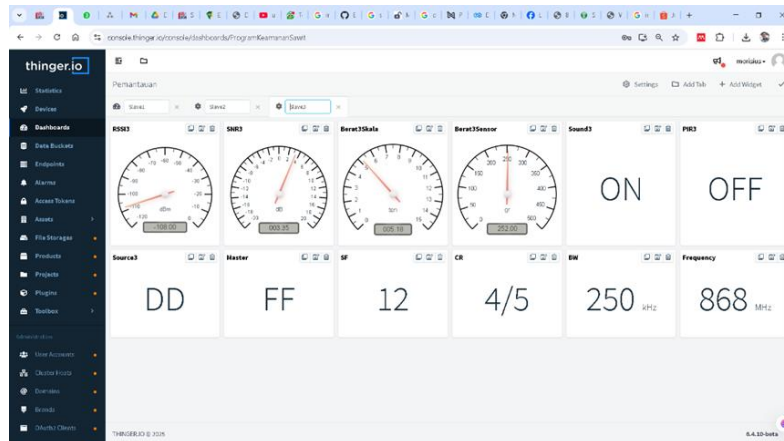


Gambar 17. Tampilan Data Kedua Pada Tab2 (Slave2)

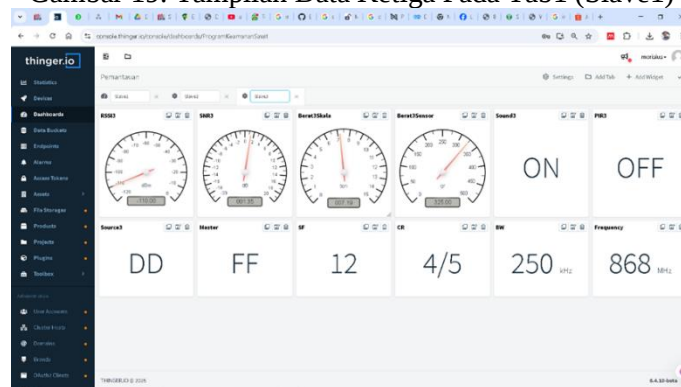


Gambar 18. Tampilan Data Kedua Pada Tab3 (Slave3)

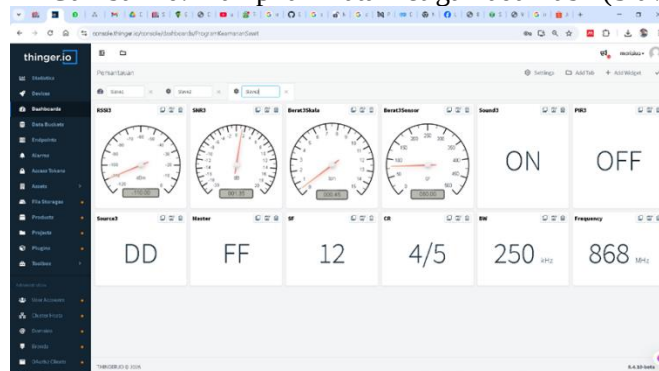
Untuk data ketiga yang diinputkan pada setiap slave diperoleh tampilan data-data sebagai berikut:



Gambar 19. Tampilan Data Ketiga Pada Tab1 (Slave1)



Gambar 20. Tampilan Data Ketiga Pada Tab2 (Slave2)



Gambar 21. Tampilan Data Ketiga Pada Tab3 (Slave3)

Data-data yang ditampilkan pada layar laptop sudah sesuai dengan data yang diinputkan pada setiap pemancar. Tambahan tampilan data berupa nilai RSSI dan SNR (live), nilai SF, CR, BW dan frekuensi kerja LoRa.

KESIMPULAN

Adapun berdasarkan hasil penelitian yang telah dilakukan maka kesimpulan yang dapat diambil ini adalah sebagai berikut:

1. Sistem keamanan berbasis multi sensor mampu mendeteksi aktivitas mencurigakan seperti pergerakan manusia, hewan, hingga kendaraan berat secara real-time.
2. Teknologi LoRa point-to-point terbukti efisien untuk komunikasi jarak jauh tanpa memerlukan jaringan internet, sangat cocok untuk wilayah kebun terpencil.
3. Pengujian dari TX1, TX2, dan TX3 menunjukkan bahwa data dari sensor dapat diterima RX dengan akurasi tinggi dan tanpa keterlambatan signifikan.
4. Data berhasil ditampilkan di dashboard Thingier.io, memudahkan proses pemantauan

jarak jauh secara digital.

5. Sistem bekerja dengan baik dalam jarak 300–600 meter antar node dengan kondisi kebun sawit terbuka hingga semi-terhalang.
6. Tidak terjadi kehilangan data (packet loss) yang berarti selama pengujian, menunjukkan kestabilan komunikasi antar perangkat.
7. Penerapan sistem ini mendukung upaya preventif terhadap pencurian dan pelanggaran akses ke lahan sawit tanpa harus menambah banyak tenaga keamanan manusia.

Saran

Saran yang perlu di pertimbangkan dari pengembangan berikut ini adalah pengembangan bertujuan membuat sistem lebih tangguh, otomatis dan mudah diakses secara jarak jauh serta sesuai untuk penerapan jangka panjang paling di industri perkebunan modern.

DAFTAR PUSTAKA

- Akyildiz, I. F., & Stuntebeck, E. P. (2006). Wireless underground sensor networks: Research challenges. *Ad Hoc Networks*, 4(6), 669–686. <https://doi.org/10.1016/j.adhoc.2006.04.003>
- Arifin, M., & Putra, Y. H. (2020). Rancang Bangun Sistem Monitoring Keamanan Kebun Kelapa Sawit Berbasis Kamera CCTV dan Sensor PIR Menggunakan Internet of Things (IoT). *Jurnal Teknologi dan Sistem Komputer*, 8(1), 15–21. <https://doi.org/10.14710/jtsiskom.8.1.2020.15-21>
- Hariyanto, D., & Santoso, A. (2020). Sistem monitoring suhu dan kelembaban berbasis IoT menggunakan LoRa. *Jurnal Teknologi dan Sistem Komputer*, 8(2), 123–130.
- Kurniawan, A., & Haryanto, A. (2019). Monitoring sistem keamanan rumah menggunakan sensor getar dan LoRa. *Jurnal Teknik ITS*, 8(2), A111–A115.
- Kusumawardhani, A., & Prasetyo, Y. (2021). Perancangan dan implementasi sistem deteksi gerakan berbasis sensor PIR dan LoRa. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, 7(1), 13–19.
- LoRa Alliance. (2015). A technical overview of LoRa® and LoRaWAN™. [White paper]. <https://loro-alliance.org>
- Nugroho, A. P., & Fauzan, M. (2019). Perancangan Sistem Keamanan Berbasis Sensor Gerak dan Kamera dengan Kendali Jarak Jauh Menggunakan ESP32 dan Telegram. *Prosiding Seminar Nasional Sistem Informasi Indonesia (SESINDO)*, 201–206.
- Nugroho, B., & Astuti, R. (2019). Sistem pemantauan lahan pertanian berbasis IoT menggunakan jaringan LoRa. *Jurnal Ilmiah Teknik Elektro dan Komputer*, 4(3), 77–83.
- Prasetyo, D., & Sari, R. M. (2021). Pemanfaatan Sensor PIR dan Kamera pada Sistem Keamanan Lahan Pertanian Berbasis Wireless Sensor Network. *Jurnal Ilmiah Teknologi dan Rekayasa*, 26(2), 101–110.
- Raharjo, B., & Widodo, E. (2022). Pengembangan Sistem Pemantauan Lahan Kelapa Sawit Berbasis Drone dan Sensor Multispektral. *Jurnal Teknik Pertanian Lampung*, 11(2), 55–63.
- Rahman, F., & Wibowo, H. (2021). Rancang bangun sistem keamanan otomatis berbasis sensor suara dan getar untuk lingkungan outdoor. *Jurnal Teknologi Informasi dan Komunikasi*, 10(1), 1–8.
- Rizki, M. A., & Sari, N. P. (2020). Analisis kinerja komunikasi LoRa untuk pengiriman data sensor pada area terbuka. *Jurnal Teknik Informatika dan Sistem Informasi*, 6(2), 140–148
- Siregar, H. A., & Tambunan, R. (2021). Implementasi IoT pada Sistem Keamanan Lahan Kelapa Sawit Menggunakan Sensor PIR dan Kamera IP. *Jurnal Teknologi Informasi dan Komputer*, 7(3), 142–149.
- Siregar, T. H., et al. (2018). Aplikasi Internet of Things (IoT) pada pertanian kelapa sawit. *Jurnal Agritech*, 38(3), 341–348.
- Widodo, A., & Handayani, D. (2020). Perancangan Sistem Keamanan Perkebunan Berbasis Sensor dan IoT Menggunakan NodeMCU ESP8266. *Jurnal Teknik Elektro dan Komputer*, 9(2), 78–85.