

# **Laporan Penugasan Akhir Elektronika Medis Monitoring Suhu Tubuh dan Detak Jantung**

**Dosen: Dr. Ir. Fatchul Arifin, M.T.**



Oleh :

Andrean Rizky Mahardika

Farhan Ramadhani

Muhammad Gufron

Mohammad Nurdin Prastya H

**D4 Teknik Elektronika**

**JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA DAN  
INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI YOGYAKARTA**

**2022**

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Detak jantung dan suhu tubuh merupakan dua parameter penting yang digunakan oleh para medis untuk mengetahui kesehatan fisik maupun kondisi mental seseorang. Pengukuran yang dilakukan saat ini masih menggunakan cara yang konvensional, cara ini akan memakan waktu yang lama dan harus dilakukan pengecekan secara berkala sehingga kurang efektif apabila ingin memantau perkembangan detak jantung dan suhu tubuh pasien dari jarak jauh. Pada penelitian ini metode yang digunakan yaitu sistem monitoring yang dirancang untuk memantau dan mengukur detak jantung dan suhu tubuh pada manusia secara otomatis. Pengukuran detak jantung dan suhu tubuh menggunakan Pulse Sensor dan sensor MLX sebagai pembaca parameter. Hasil dari pengukuran kemudian ditampilkan pada LCD. Indikator yang akan menginformasikan keadaan pasien atau seseorang yang sedang dipantau sehingga akan ada penanganan yang dilakukan oleh staf medis. Alat ukur detak jantung dan suhu tubuh menggunakan NodeMCU ESP8266 menggunakan pulse sensor dan MLX dengan memanfaatkan aplikasi *blynk* sebagai media dari *internet of things* sebagai sistem monitoring. Alat penelitian ini dapat menampilkan data suhu tubuh dan detak jantung manusia melalui *blynk*.

### **B. Rumusan Masalah**

Dari latar belakang diatas maka dapat ditarik rumusan masalah sebagai berikut?

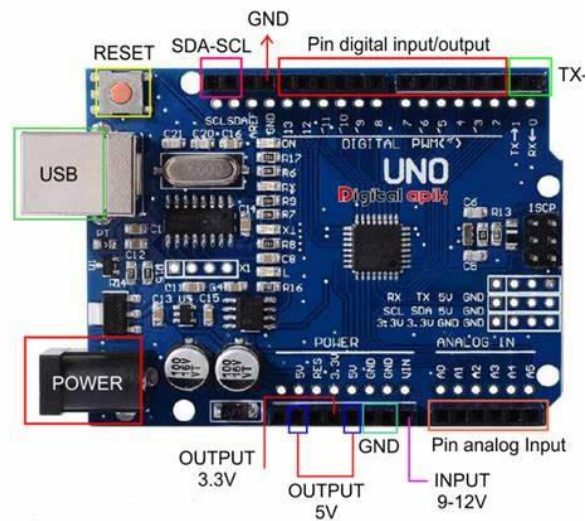
1. Bagaimana merancang alat tersebut hingga dapat digunakan
2. Bagaimana menyusun program dan menjalankannya?

## BAB II

### KAJIAN PUSTAKA

#### A. Arduino Uno

Arduino Uno adalah salah satu jenis papan mikrokontroler berbasis ATmega328, dan Uno adalah istilah bahasa Italia yang artinya satu. Arduino Uno dinamai untuk menandai peluncuran papan mikrokontroler yang akan datang yaitu Arduino Uno Board 1.0. Papan ini mencakup pin-14 I / O digital, colokan listrik, i / ps-6 analog, resonator keramik-A16 MHz, koneksi USB, tombol RST, dan header ICSP. Semua ini dapat mendukung mikrokontroler untuk operasi lebih lanjut dengan menghubungkan papan ini ke komputer. Catu daya papan ini dapat dilakukan dengan bantuan adaptor AC ke DC, kabel USB, atau baterai. Artikel ini membahas tentang apa itu mikrokontroler Arduino Uno, konfigurasi pin, spesifikasi atau fitur Arduino Uno, dan aplikasi.



**Gambar Arduino uno tipe smd.**

Arduino Uno dapat mendeteksi lingkungan dari input. Di sini inputnya adalah berbagai macam sensor dan ini dapat mempengaruhi sekitarnya melalui pengontrolan motor, lampu, aktuator lain, dll. Mikrokontroler ATmega328 pada papan Arduino dapat diprogram dengan bantuan bahasa pemrograman Arduino dan IDE (Integrated Development Environment ). Proyek Arduino dapat berkomunikasi dengan perangkat lunak saat berjalan di PC.

## B. MLX90614(temperatur)

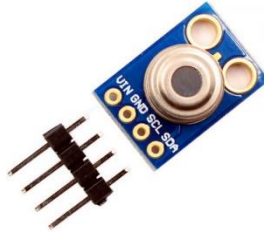
MLX90614 adalah **Sensor Suhu Digital Inframerah Tanpa Kontak (IR)** yang dapat digunakan untuk mengukur suhu objek tertentu mulai dari  $-70^{\circ}\text{C}$  hingga  $382,2^{\circ}\text{C}$ . Sensor menggunakan sinar IR untuk mengukur suhu objek tanpa kontak fisik dan berkomunikasi dengan mikrokontroler menggunakan protokol I2C.

Seperti disebutkan sebelumnya, sensor MLX90614 dapat mengukur suhu suatu benda tanpa kontak fisik dengannya. Hal ini dimungkinkan dengan hukum yang disebut **Stefan-Boltzmann Law**, yang menyatakan bahwa semua benda dan makhluk hidup memancarkan Energi IR dan intensitas energi IR yang dipancarkan ini akan berbanding lurus dengan suhu benda atau makhluk hidup tersebut. Jadi sensor MLX90614 menghitung suhu suatu benda dengan mengukur jumlah energi IR yang dipancarkan darinya.

### Spesifikasi Sensor Suhu MLX90614

- Tegangan Operasi: 3.6V hingga 5V (tersedia dalam versi 3V dan 5V)
- Pasokan saat ini: 1.5mA
- Kisaran Suhu Objek:  $-70^{\circ}\text{C}$  hingga  $382,2^{\circ}\text{C}$
- Kisaran Suhu Sekitar:  $-40^{\circ}\text{C}$  hingga  $125^{\circ}\text{C}$
- Akurasi:  $0,02^{\circ}\text{C}$
- Bidang Pandang:  $80^{\circ}$
- Jarak antara objek dan sensor: 2cm-5cm (perkiraan)

Pin No.	Nama Pin	Deskripsi
1	Vdd (Catu daya)	Vdd dapat digunakan untuk memberi daya pada sensor, biasanya menggunakan 5V
2	Tanah	Logam juga dapat bertindak sebagai tanah
3	SDA – Serial Data	Pin data serial yang digunakan untuk Komunikasi I2C
4	SCL – Jam Serial	Pin Jam Serial yang digunakan untuk Komunikasi I2C



**Gambar sensor MLX90614(temperatur)**

### **C. Pulse Sensor (BPM)**

Pulse Heart Rate Sensor adalah sebuah sensor denyut jantung. Sensor ini dapat mendeteksi denyut nadi pada jari telunjuk tangan dengan cara menggabungkan data denyut nadi dari sensor dengan program di mikrokontroler bisa di dapatkan nilai bpm, Heart rate sendiri merupakan detak jantung per satuan waktu yang biasanya dinyatakan dalam beats per menit (bpm).

Sensor ini berfungsi untuk menghitung jumlah detak jantung .caranya cukup sederhana dengan meletakkan jari ke sensor.Selanjutnya data pembacaan sensor tersebut di terima arduino melalui pin analog (A0), menggunakan fitur Analog to Digital Converter (ADC), diolah menjadi bpm (*Beats Per Minute*). Detak jantung normal (Resting Heart Rate) pada manusia adalah 60 - 100 bpm, biasanya faktor yg paling berpengaruh yg membedakan jumlah detak jantung per menit ( bpm ) masing2 orang adalah umur, kondisi jantung dan aktivitas olahraga yg sedang di lakukan.



**Gambar Pulse Sensor BPM**

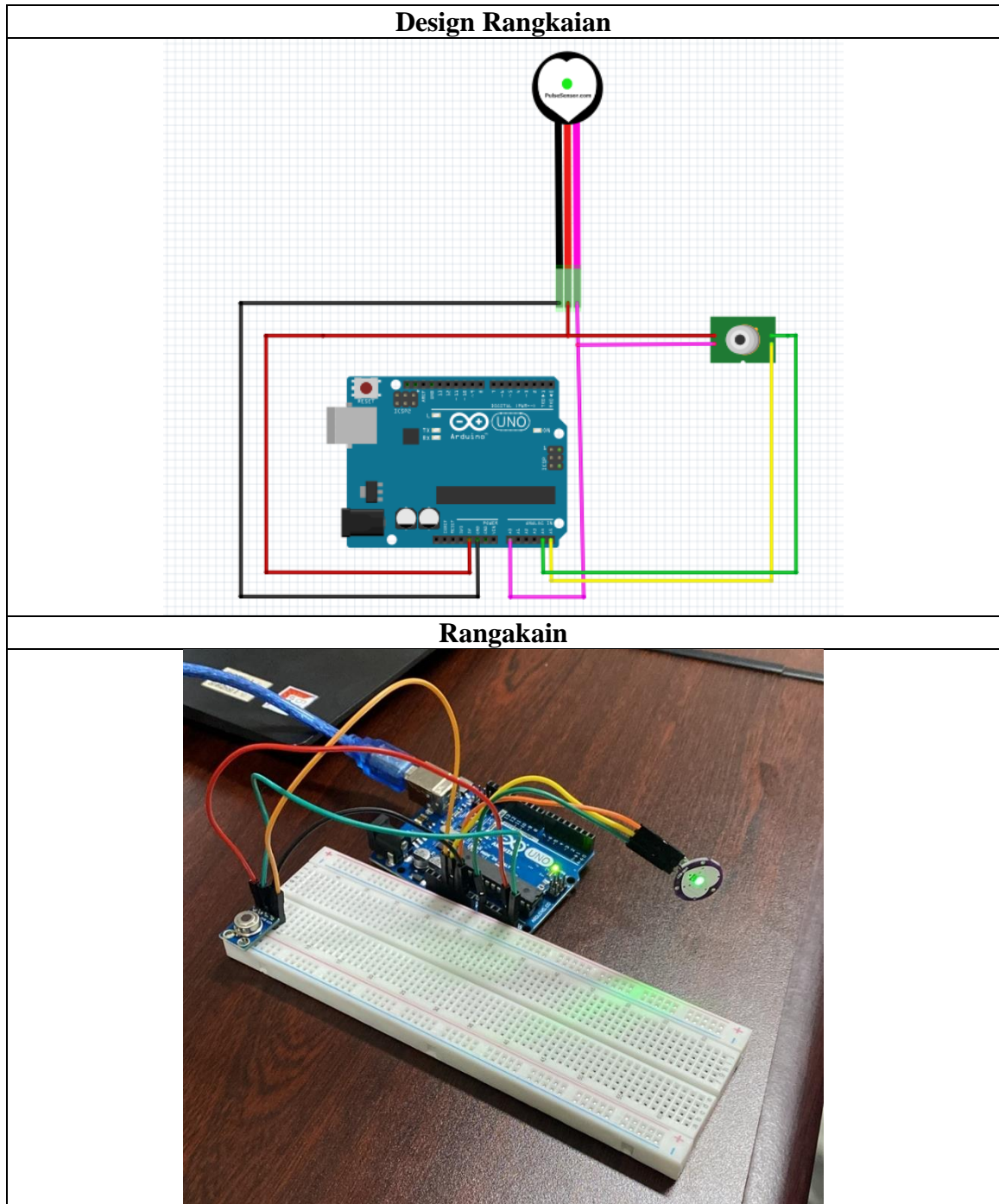
#### **Alokasi pin Arduino**

Heart Rate	Arduino
Signal	A0
Gnd	Gnd
VCC	+5V

## BAB III

### PEMBAHASAN

#### A. Rangkaian



## Keterangan

Pin Pulse sensor	Pin Arduino IDE
Signal	A0
Gnd	Gnd
VCC	+5V
Pin MLQ90614	Pin Arduino IDE
SCL	A4
SDA	A5
VSS	A0
VDD	+5V

## B. Program

Kode program
<pre>#include &lt;Adafruit_MLX90614.h&gt;  Adafruit_MLX90614 mlx = Adafruit_MLX90614();  int pulsePin = A0;  int blinkPin = 13;  volatile int BPM;  volatile int Signal;  volatile int IBI = 600;  volatile boolean Pulse = false;  volatile boolean QS = false;  static boolean serialVisual = true;  volatile int rate[10];  volatile unsigned long sampleCounter = 0;  volatile unsigned long lastBeatTime = 0;  volatile int P = 512;  volatile int T = 512;  volatile int thresh = 525;  volatile int amp = 100;</pre>

```

volatile boolean firstBeat = true;

volatile boolean secondBeat = false;

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

    while (!Serial);

    Serial.println("Adafruit MLX90614 test");

    if (!mlx.begin()) {

        Serial.println("Error connecting to MLX sensor. Check wiring.");

        while (1);

    };

    Serial.print("Emissivity = "); Serial.println(mlx.readEmissivity());

    Serial.println("=====");

    pinMode(blinkPin, OUTPUT);

    interruptSetup();
}

void loop()
{
    Serial.print("*C\tObject = "); Serial.print(mlx.readObjectTempC()); Serial.println("*C");

    Serial.println();

    delay(2000);

    serialOutput();

    if (QS == true)
    {
        serialOutputWhenBeatHappens();
    }
}

```



```
    QS = false;

}

delay(20);

}

void interruptSetup()

{

    TCCR2A = 0x02;

    TCCR2B = 0x06;

    OCR2A = 0X7C;

    TIMSK2 = 0x02;

    sei();

}

void serialOutput()

{

    if (serialVisual == true)

    {

        arduinoSerialMonitorVisual('-', Signal);

    }

    else

    {

        sendDataToSerial('S', Signal);

    }

}

void serialOutputWhenBeatHappens()

{
```

```

if (serialVisual == true)
{
    Serial.print(" Heart-Beat Found ");

    Serial.print("BPM: ");

    Serial.println(BPM);
}
else
{
    sendDataToSerial('B', BPM);

    sendDataToSerial('Q', IBI);
}
}

void arduinoSerialMonitorVisual(char symbol, int data )
{
    const int sensorMin = 0;

    const int sensorMax = 1024;

    int sensorReading = data;

    int range = map(sensorReading, sensorMin, sensorMax, 0, 11);
}

void sendDataToSerial(char symbol, int data )
{
    Serial.print(symbol);

    Serial.println(data);
}

ISR(TIMER2_COMPA_vect)

```

```

{

cli();

Signal = analogRead(pulsePin);

sampleCounter += 2;

int N = sampleCounter - lastBeatTime;

if (Signal < thresh && N > (IBI / 5) * 3)

{

    if (Signal < T)

    {

        T = Signal;

    }

}

if (Signal > thresh && Signal > P)

{

    P = Signal;

}

if (N > 250)

{

    if ( (Signal > thresh) && (Pulse == false) && (N > (IBI / 5) * 3) )

    {

        Pulse = true;

        digitalWrite(blinkPin, HIGH);

        IBI = sampleCounter - lastBeatTime;

        lastBeatTime = sampleCounter;

        if (secondBeat)

```

```

{
    secondBeat = false;

    for (int i = 0; i <= 9; i++)
    {
        rate[i] = IBI;
    }
}

if (firstBeat)
{
    firstBeat = false;
    secondBeat = true;

    sei();

    return;
}

word runningTotal = 0;

for (int i = 0; i <= 8; i++)
{
    rate[i] = rate[i + 1];

    runningTotal += rate[i];
}

rate[9] = IBI;

runningTotal += rate[9];

runningTotal /= 10;

BPM = 60000 / runningTotal;

QS = true;

```

```

    }

    }

    if (Signal < thresh && Pulse == true)
    {
        digitalWrite(blinkPin, LOW);

        Pulse = false;

        amp = P - T;

        thresh = amp / 2 + T;

        P = thresh;

        T = thresh;
    }

    if (N > 2500)
    {
        thresh = 512;

        P = 512;

        T = 512;

        lastBeatTime = sampleCounter;

        firstBeat = true;

        secondBeat = false;
    }

    sei();
}

```

### C. Hasil Pembacaan Sensor

```
20:28:05.123 -> Adafruit MLX90614 test
20:28:05.157 -> Emissivity = 1.00
20:28:05.157 -> =====
20:28:05.224 -> *C      Object = 28.97*C
20:28:05.224 ->
20:28:07.188 -> *C      Object = 29.69*C
20:28:07.222 ->
20:28:09.221 -> *C      Object = 29.71*C
20:28:09.255 ->
20:28:11.242 -> *C      Object = 30.47*C
20:28:11.277 ->
20:28:13.240 -> Heart-Beat Found BPM: 35
20:28:13.273 -> *C      Object = 30.43*C
20:28:13.306 ->
20:28:15.265 -> Heart-Beat Found BPM: 53
20:28:15.300 -> *C      Object = 30.45*C
20:28:15.334 ->
20:28:17.301 -> Heart-Beat Found BPM: 102
20:28:17.334 -> *C      Object = 30.39*C
20:28:17.367 ->
20:28:19.321 -> Heart-Beat Found BPM: 115
20:28:19.354 -> *C      Object = 28.11*C
20:28:19.388 ->
```

### D. Analisis Data

Terdapat 2 jenis keluaran data mentah, yaitu data BPM dan suhu object(tubuh). Jadi pada percobaan ini data yang ditampilkan adalah nilai BPM dan suhu tubuh dalam Celcius.

Seperti yang dapat dilihat pada gambar hasil pembacaan sensor suhu MLX90614 cukup stabil, akan tetapi pada pembacaan pulse sensor kurang stabil. Pengetesan pembacaan sensor dilakukan dengan cara mendekatkan ujung jari kita ke arah pancaran radiasi sensor. Dengan begitu nilai suhu dan BPM akan naik karena sensor mendeteksi adanya objek. Respon pembacaan pada sensor BLX90614 cukup baik karena bisa langsung mendeteksi suhu objek jika didekatkan. Sedangkan pada pulse sensor pembacaan BPM kurang baik dimana hasil pembacaan data berubah-ubah dan sering tidak terdeteksi. Hal ini mungkin disebabkan oleh sensor yang kondisinya kurang baik atau program Arduino IDE yang konfigurasinya kurang baik.

Data yang dibaca kedua sensor kemudian ditampilkan pada serial monitor seperti gambar hasil percobaan. Data juga dapat ditampilkan pada web menggunakan aplikasi Blynk dan modul MCU ESP8266. Namun pada percobaan kali ini tidak menggunakan modul MCU ESP8266.

