DAFTAR PUSTAKA

Achmad, H., Ikhsan, S., Rony, D., & Usep. S. G. (2015). PERANCANGAN MODUL SISTEM DETEKSI DAN TRANSMITTER SIGNAL PADA PERANGKAT ANALISA UNSUR DENGAN TEKNIK XRF UNTUK INDUSRI. Tangerang selatan

Ariawan, A. (2021). SISTEM MONITORING SUHU TUBUH MENGGUNAKAN SENSOR INFRAMERAH BERBASIS ARDUINO. Makassar. Universitas Hasanuddin.

Cahyono, T. H., & Suprayitno, E. A. (2018). *ALAT UKUR BERAT BADAN, TINGGI BADAN DAN SUHU BADAN DI POSYANDU BERBASIS ANDROID.* Sidoarjo: Universitas Muhammadiyah Sidoarjo.

Ring, E. F., & Ammer, K. (2012). *Infrared thermal imaging in medicine.* Vienna: University of Glamorgan.

Safitri, M., & Dinata, G. A. (2019). *NON-CONTACT THERMOMETER BERBASIS INFRA MERAH.* Yogyakarta: Universitas Muhammadiyah Yogyakarta.

Sollu, T. S., Alamsyah, Bachtiar, M., Amir, A., & Bontong, B. (2018). *Sistem Monitoring Detak Jantung dan Suhu Tubuh Menggunakan Arduino.* Palu: Universitas Tadulako.

Suyudi, A. (2017). *Tanggung Jawab Pelayanan Jasa Transportasi Laut Oleh PT. Pelni Terhadap Penumpang.* Makassar: Universitas Hasanuddin.

Vadivambal, R., & Jayas, D. S. (2015). *Thermal Imaging.* Winnipeg: Springer Science+Business Media.

Varihar, V. R., Tonge, A. Y., & Ganorkar, P. D. (2017). *Heartbeat and Temperature Monitoring System for Remote Patients using Arduino.* Amravati: Prof Ram Meghe College of Engineering and Management.

Wirawan Dwiky (2018). *ANALISA PENGARUH BESAR SUDUT DAN JARAK TITIK FOKUS REFELKTOR CORNER PADA ANTENA HELIX MODE AXIAL UNTUK SISTEM KOMUNIKASI RADIO 433 MHZ.* Jawa Timur : Universitas Jember.

Zebua, J. S., Suraatmadja, M. S., & Qurthobi, A. (2016). *PERANCANGAN TERMOMETER DIGITAL TANPA SENTUHAN*. Bandung: Universitas Telkom.

LAMPIRAN KODE

PROGRAM ARDUINO

MLX90614

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Adafruit_MLX90614.h>
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {

  Serial.begin(9600);

  mlx.begin();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("Connected");
  lcd.setCursor(2,1);
  lcd.print("Successfully");
}

void loop() {
```

```
  Serial.print("T");
  Serial.print(mlx.readObjectTempC()+2.5);
  Serial.println();

  delay(1000);

}
```

LAMPIRAN KODE

PROGRAM

TRANSMITTER RF

433Hz

```
#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Adafruit_MLX90614.h>

Adafruit_MLX90614 mlx = Adafruit_MLX90614();

LiquidCrystal_I2C lcd(0x27,16,2);

//#include "DHT.h";

//#define DHTPIN 7
//#define DHTTYPE DHT11

float hum;
float temp;

String str_humid;
String str_temp;
String str_out;

RH_ASK rf_driver;

//DHT dht(DHTPIN, DHTTYPE);

void setup()
```

```
{
  Serial.begin(9600);

  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.noCursor();

  rf_driver.init();
  //Serial.begin(9600);  // Debugging only
  //dht.begin();
  mlx.begin();
}

void loop()
{
    lcd.setCursor(0,1); lcd.print("T= ");
    lcd.setCursor(2,1); lcd.print(mlx.readObjectTempC()+2.5);
    lcd.setCursor(7,1); lcd.print(" C");

  delay(1000);

    hum= mlx.readObjectTempC()+2.5;
    //temp= dht.readTemperature();
    temp= mlx.readObjectTempC()+2.5;

    str_humid = String(hum);
    str_temp = String(temp);
```

```
    str_out = str_humid + "," + str_temp;

    static char *msg = str_out.c_str();

    rf_driver.send((uint8_t *)msg, strlen(msg));
    rf_driver.waitPacketSent();

     Serial.print("T");
  Serial.print(mlx.readObjectTempC()+2.5);
  Serial.println();
}
```

LAMPIRAN KODE

PROGRAM RECEIVER

RF 433Hz

```
// ask_receiver.pde
// -- mode: C++ --
// Simple example of how to use RadioHead to receive messages
// with a simple ASK transmitter in a very simple way.
// Implements a simplex (one-way) receiver with an Rx-B1 module
// Tested on Arduino Mega, Duemilanova, Uno, Due, Teensy, ESP-12

#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile

String str_humid;
String str_temp;
String str_out;

RH_ASK rf_driver;

void setup()
{
  rf_driver.init();
  Serial.begin(9600);    // Debugging only
}

void loop()
```

```
{
    uint8_t buf[11];
    uint8_t buflen = sizeof(buf);

    if(rf_driver.recv(buf, &buflen)) // Non-blocking
    {
      str_out =String((char*)buf);

      for (int i = 0; i < str_out.length(); i++){
        if (str_out.substring(i, i+1) == ","){
        str_humid = str_out.substring(0, i);
        str_temp = str_out.substring(i+1);
        break;
        }
      }
      //Serial.print("Humidity: ");
      //Serial.print(str_humid);
      //Serial.print("Temperature = ");
     // Serial.println(str_temp);

        Serial.print("T");
        Serial.print(str_temp);
        Serial.println();
    }
}
```

LAMPIRAN KODE

PROGRAM

VISUAL BASIC

```
Imports System
Imports System.IO.Ports
Imports Excel = Microsoft.Office.Interop.Excel '-> It is
used to save data to Excel.

Public Class Form1
    Dim vpb_sy, vpb_ly As Integer
    Dim TempL As Integer
    Dim Temp, TempResult As String
    Dim ChartLimit As Integer = 30
    Dim StrSerialIn, StrSerialInRam As String
    Dim SB As Boolean = True '-> Helper variable to display
the result of heart rate calculation (BPM)
    Dim SR As Boolean = False '-> Variable trigger to start
the calculation of the heart rate (BPM)

    Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        Me.CenterToScreen()
        PanelConnection.Focus()
        ComboBoxBaudRate.SelectedIndex = 0

        For i = 0 To 30
            Chart2.Series("Temperature").Points.AddY(0)
            If Chart2.Series(0).Points.Count = ChartLimit
Then
                Chart2.Series(0).Points.RemoveAt(0)
            End If
        Next
        Chart2.ChartAreas(0).AxisY.Maximum = 50
        Chart2.ChartAreas(0).AxisY.Minimum = -10

Chart2.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled =
False
    End Sub

    Private Sub ComboBoxPort_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs) Handles
ComboBoxPort.SelectedIndexChanged
        PanelConnection.Focus()
    End Sub

    Private Sub ComboBoxPort_DropDown(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
ComboBoxPort.DropDown
```

```
        PanelConnection.Focus()
    End Sub

    Private Sub ComboBoxPort_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles ComboBoxPort.Click
        If LabelStatus.Text = "Status : Connected" Then
            MsgBox("Connce tion in progress, please
Disconnect to change COM.", MsgBoxStyle.Critical, "Warning
!!!")
            Return
        End If
    End Sub

    Private Sub ComboBoxBaudRate_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs) Handles
ComboBoxBaudRate.SelectedIndexChanged
        PanelConnection.Focus()
    End Sub

    Private Sub ComboBoxBaudRate_DropDown(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
ComboBoxBaudRate.DropDown
        PanelConnection.Focus()
    End Sub

    Private Sub ComboBoxBaudRate_Click(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
ComboBoxBaudRate.Click
        If LabelStatus.Text = "Status : Connected" Then
            MsgBox("Conncetion in progress, please
Disconnect to change Baud Rate.", MsgBoxStyle.Critical,
"Warning !!!")
            Return
        End If
    End Sub

    Private Sub ButtonScanPort_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonScanPort.Click
        PanelConnection.Focus()
        If LabelStatus.Text = "Status : Connected" Then
            MsgBox("Conncetion in progress, please
Disconnect to scan the new port.", MsgBoxStyle.Critical,
"Warning !!!")
            Return
```

```
        End If
        ComboBoxPort.Items.Clear()
        Dim myPort As Array
        Dim i As Integer
        myPort = IO.Ports.SerialPort.GetPortNames()
        ComboBoxPort.Items.AddRange(myPort)
        i = ComboBoxPort.Items.Count
        i = i - i
        Try
            ComboBoxPort.SelectedIndex = i
            ButtonConnect.Enabled = True
            ButtonExportToExcel.Enabled = False
            ButtonClearRecording.Enabled = False
            ButtonRecord.Enabled = False
        Catch ex As Exception
            MsgBox("Com port not detected",
MsgBoxStyle.Critical, "Warning !!!")
            ComboBoxPort.Text = ""
            ComboBoxPort.Items.Clear()
            Return
        End Try
        ComboBoxPort.DroppedDown = True
    End Sub

    Private Sub ButtonConnect_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonConnect.Click
        If PanelConnection.Focus() Then
            Try
                SerialPort1.BaudRate =
ComboBoxBaudRate.SelectedItem
                SerialPort1.PortName =
ComboBoxPort.SelectedItem
                SerialPort1.Open()
                TimerSerial.Start()
                ButtonConnect.BackColor = Color.Green
                ButtonConnect.ForeColor = Color.White
                LabelStatus.Text = "Status : Connected"
                ButtonDisconnect.BackColor =
Color.WhiteSmoke
                ButtonDisconnect.ForeColor = Color.Black
                ButtonRecord.Enabled = True
                PictureBoxStatusConnection.BackColor =
Color.Green
            Catch ex As Exception
```

```
                MsgBox("Please check the Hardware, COM, Baud
Rate and try again.", MsgBoxStyle.Critical, "Connection
failed !!!")
            End Try
        Else
            Try
                TimerSerial.Stop()
                Threading.Thread.Sleep(500)
                SerialPort1.Close()
                Threading.Thread.Sleep(500)
                ButtonConnect.Text = "Connect"
                ButtonRecord.Enabled = False
                ButtonScanPort.Enabled = True
                ButtonRecord.Enabled = False
            Catch ex As Exception
            End Try
        End If
    End Sub

    Private Sub ButtonDisconnect_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonDisconnect.Click
        PanelConnection.Focus()
        TimerSerial.Stop()
        SerialPort1.Close()
        ButtonDisconnect.SendToBack()
        ButtonDisconnect.BackColor = Color.Red
        ButtonDisconnect.ForeColor = Color.White
        ButtonConnect.BackColor = Color.WhiteSmoke
        ButtonConnect.ForeColor = Color.Black
        LabelStatus.Text = "Status : Disconnect"
        PictureBoxStatusConnection.Visible = True
        PictureBoxStatusConnection.BackColor = Color.Red
    End Sub
    'The function to convert temperature values to
PictureBoxPBTemp size so that it looks like a progress bar.
    Function MapVPB(ByVal X As Single, ByVal In_min As
Single, ByVal In_max As Single, ByVal Out_min As Single,
ByVal Out_max As Single) As Integer
        Dim A As Single
        Dim B As Single
        A = X - In_min
        B = Out_max - Out_min
        A = A * B
        B = In_max - In_min
```

```
        A = A / B
        MapVPB = A + Out_min
    End Function

    Private Sub TimerSerial_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TimerSerial.Tick
        Try
            StrSerialIn = SerialPort1.ReadExisting '-->
Read incoming serial data

            Dim TB As New TextBox
            TB.Multiline = True
            TB.Text = StrSerialIn   '--> Enter serial data
into the textbox

            If TB.Lines.Count > 0 Then
                If TB.Lines(0) = "Failed to read from
sensor!" Then '--> Check Arduino if it fails to read the DHT
sensor, if this happens the connection is disconnected
                    TimerSerial.Stop()
                    SerialPort1.Close()
                    LabelStatus.Text = "Status : Disconnect"
                    ButtonDisconnect.SendToBack()
                    ButtonConnect.BringToFront()
                    PictureBoxStatusConnection.Visible =
True
                    PictureBoxStatusConnection.BackColor =
Color.Red
                    MsgBox("Please check the Hardware and
Please connect again.", MsgBoxStyle.Critical, "Connection
failed !!!")
                    Return
                End If

                StrSerialInRam = TB.Lines(0).Substring(0, 1)
                If StrSerialInRam = "T" Then
                    Temp = TB.Lines(0)
                    TempL = Temp.Length
                Else
                    Temp = Temp
                End If
                TempResult = Mid(Temp, 2, TempL)
                LabelTemperature.Text = TempResult & " °C"
```

```
                'Result
                If TempResult >= "37,5" Then
                    LabelResult1.Text = "Over"
                    LabelResult2.Text = "The"
                    LabelResult3.Text = "Limit"
                    LabelResult1.Visible = True
                    LabelResult3.Visible = True
                    LabelResult2.AutoSize = True
                    LabelResult1.ForeColor = Color.Red
                    LabelResult2.ForeColor = Color.Red
                    LabelResult3.ForeColor = Color.Red
                Else
                    LabelResult2.Text = "Normal"
                    LabelResult2.ForeColor = Color.RoyalBlue
                    LabelResult1.Visible = False
                    LabelResult3.Visible = False
                End If
                '-------------------------------------------------------------------------------------------------


Chart2.Series("Temperature").Points.AddY(TempResult)
                If Chart2.Series(0).Points.Count =
ChartLimit Then
                    Chart2.Series(0).Points.RemoveAt(0)
                End If
                '-------------------------------------------------------------------------------------------------

                '-----------If the Then connection Is
successful And running, PictureBoxStatusConnection will
blink----
                If PictureBoxStatusConnection.Visible = True
Then
                    PictureBoxStatusConnection.Visible =
False
                ElseIf PictureBoxStatusConnection.Visible =
False Then
                    PictureBoxStatusConnection.Visible =
True
                End If
                '-------------------------------------------------------------------------------------------------
            End If
        Catch ex As Exception
```

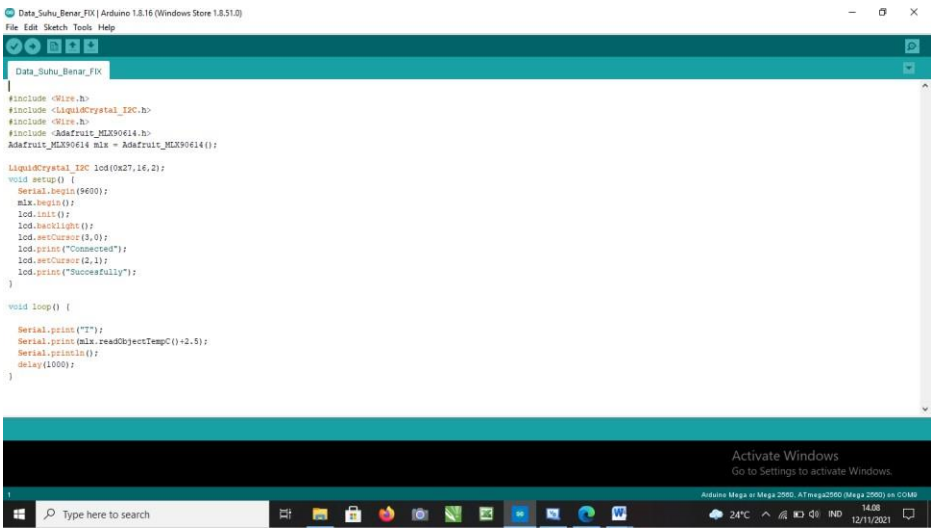
```
            TimerSerial.Stop()
            SerialPort1.Close()
            LabelStatus.Text = "Status : Disconnect"
            ButtonDisconnect.SendToBack()
            ButtonConnect.BringToFront()
            PictureBoxStatusConnection.BackColor = Color.Red
            MsgBox("Please check the Hardware and Please
connect again." & ex.Message, MsgBoxStyle.Critical,
"Connection failed !!!")
            Return
        End Try
    End Sub

    Private Sub ButtonExportToExcel_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonExportToExcel.Click
        Try
            If DataGridViewBT.Rows.Count > 0 Then
                Dim filename As String = ""
                Dim SV As SaveFileDialog = New
SaveFileDialog()
                SV.Filter = "EXCEL FILES|*.xlsx;*.xls"
                Dim result As DialogResult = SV.ShowDialog()

                If result = DialogResult.OK Then
                    Me.Text = "Monitoring Body Temperature
(Saving to Excel. Please wait...)"
                    ProgressBarSave.Visible = True
                    ProgressBarSave.Value = 2
                    filename = SV.FileName
                    Dim multiselect As Boolean =
DataGridViewBT.MultiSelect
                    DataGridViewBT.MultiSelect = True
                    DataGridViewBT.SelectAll()
                    DataGridViewBT.ClipboardCopyMode =
DataGridViewClipboardCopyMode.EnableAlwaysIncludeHeaderText

Clipboard.SetDataObject(DataGridViewBT.GetClipboardContent()
)
                    Dim results =
System.Convert.ToString(Clipboard.GetData(DataFormats.Text))
                    DataGridViewBT.ClearSelection()
                    DataGridViewBT.MultiSelect = multiselect
                    Dim XCELAPP As
Microsoft.Office.Interop.Excel.Application = Nothing
```

```
                    Dim XWORKBOOK As
Microsoft.Office.Interop.Excel.Workbook = Nothing
                    Dim XSHEET As
Microsoft.Office.Interop.Excel.Worksheet = Nothing
                    Dim misValue As Object =
System.Reflection.Missing.Value
                    ProgressBarSave.Value = 4
                    XCELAPP = New Excel.Application()
                    XWORKBOOK =
XCELAPP.Workbooks.Add(misValue)
                    XCELAPP.DisplayAlerts = False
                    XCELAPP.Visible =  False
                    XSHEET = XWORKBOOK.ActiveSheet
                    ProgressBarSave.Value = 6
                    XSHEET.Paste()
                    XWORKBOOK.SaveAs(filename,
Excel.XlFileFormat.xlOpenXMLWorkbook)
                    XWORKBOOK.Close(False)
                    XCELAPP.Quit()
                    ProgressBarSave.Value = 8
                    Try

System.Runtime.InteropServices.Marshal.ReleaseComObject(XSHE
ET)

System.Runtime.InteropServices.Marshal.ReleaseComObject(XWOR
KBOOK)

System.Runtime.InteropServices.Marshal.ReleaseComObject(XCEL
APP)
                    Catch
                    End Try
                    Me.Text = "Monitoring Body Temperature"
                    ProgressBarSave.Value = 10
                    ProgressBarSave.Visible = False
                    MessageBox.Show("Save Succesfully")
                End If
            End If
        Catch ex As Exception
            Me.Text = "Data Log"
            MessageBox.Show(ex.Message, "Error Message",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub
```

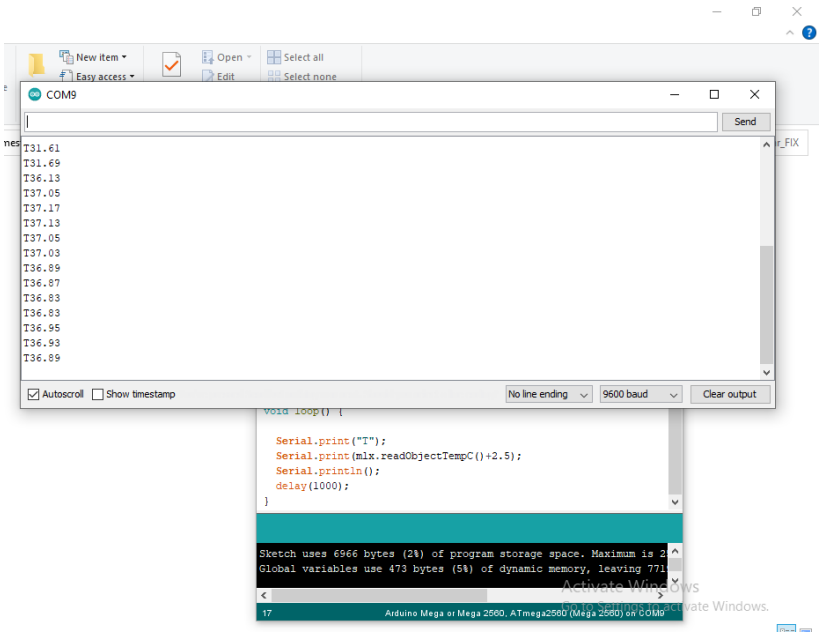
```
    Private Sub ButtonRecord_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonRecord.Click
        If ButtonRecord.Text = "Start Recording" Then
            ButtonRecord.Text = "Stop Recording"
            ButtonRecord.BackColor = Color.Red
            ButtonRecord.ForeColor = Color.WhiteSmoke
            DataGridViewBT.Rows.Clear()
            Chart2.Series("Temperature").Points.Clear()
            ButtonConnect.Enabled = False
            ButtonExportToExcel.Enabled = False
            ButtonClearRecording.Enabled = False
            TimerDataLogRecord.Start()

        Else
            ButtonRecord.Text = "Start Recording"
            ButtonRecord.BackColor = Color.WhiteSmoke
            ButtonConnect.Enabled = True
            ButtonExportToExcel.Enabled = True
            ButtonClearRecording.Enabled = True
            TimerDataLogRecord.Stop()


        End If
        '-----------------------------------------------------
    End Sub

    Private Sub ButtonClearRecording_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonClearRecording.Click
        For i = 0 To 30 Step 1

Chart2.Series("Temperature").Points.AddXY(DateTime.Now.ToLon
gTimeString, 0)
            If Chart2.Series(0).Points.Count = ChartLimit
Then
                Chart2.Series(0).Points.RemoveAt(0)
            End If
        Next
        DataGridViewBT.Rows.Clear()
    End Sub

    Private Sub TimerDataLogRecord_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TimerDataLogRecord.Tick
```

```
        Dim Temp_Log As String
        Dim DT As DateTime = Now

        Temp_Log = Mid(Temp, 3, TempResult)
        DataGridViewBT.Rows.Add(New String()
{DataGridViewBT.RowCount, TempResult, DT.ToLongTimeString,
DT.ToString("dd-MM-yyyy")})
        Me.DataGridViewBT.FirstDisplayedScrollingRowIndex =
Me.DataGridViewBT.RowCount - 1


Chart2.Series("Temperature").Points.AddXY(DateTime.Now.ToLon
gTimeString, TempResult)
        If Chart2.Series(0).Points.Count = ChartLimit Then
            Chart2.Series(0).Points.RemoveAt(0)
        End If
    End Sub
End Class
```

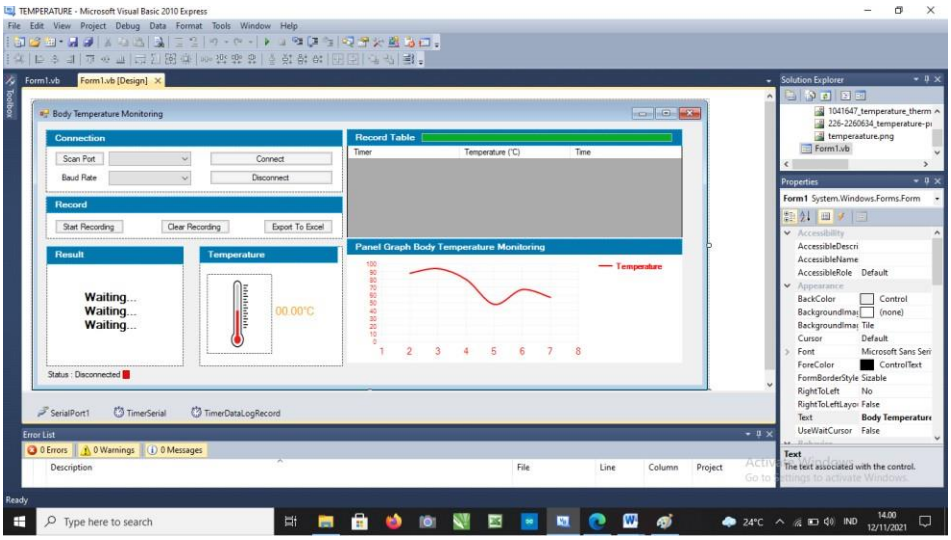
LAMPIRAN PROSES PENGAMBILAN DATA



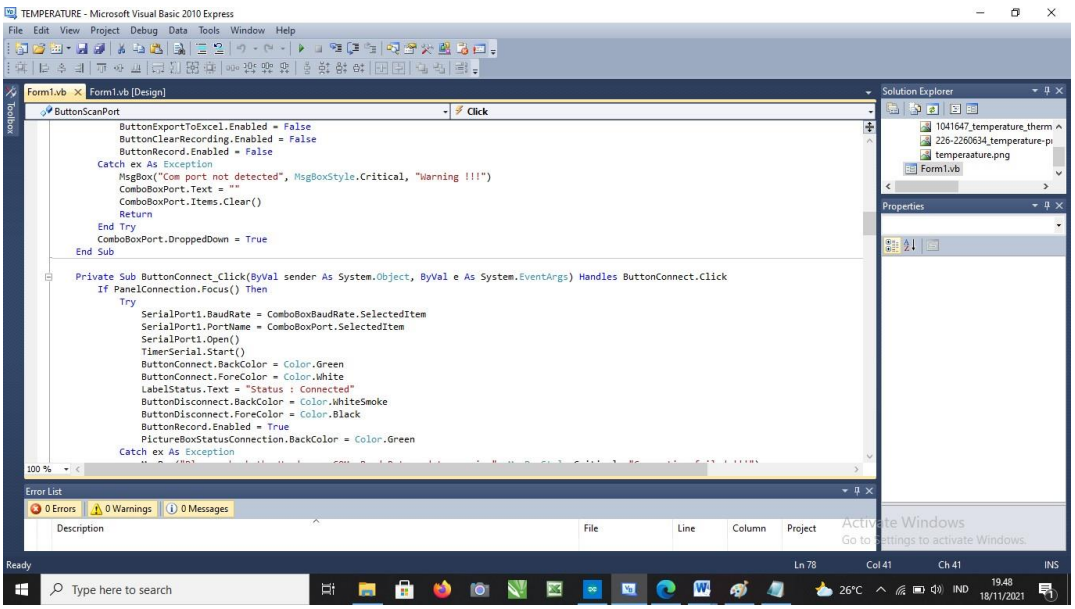
Tampilan Arduino Code RF 433 MHz Transmitter



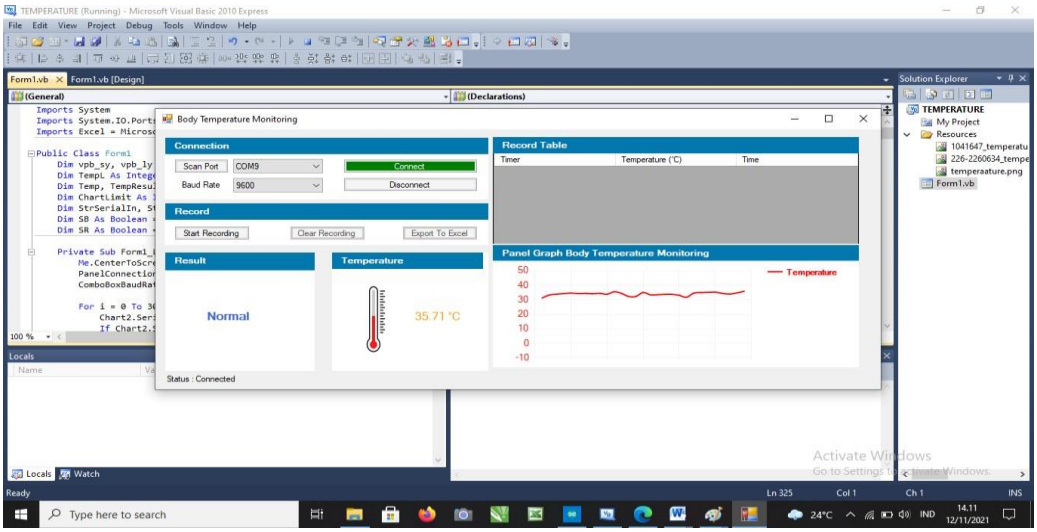
Tampilan Serial Monitor RF 433MHz Receiver



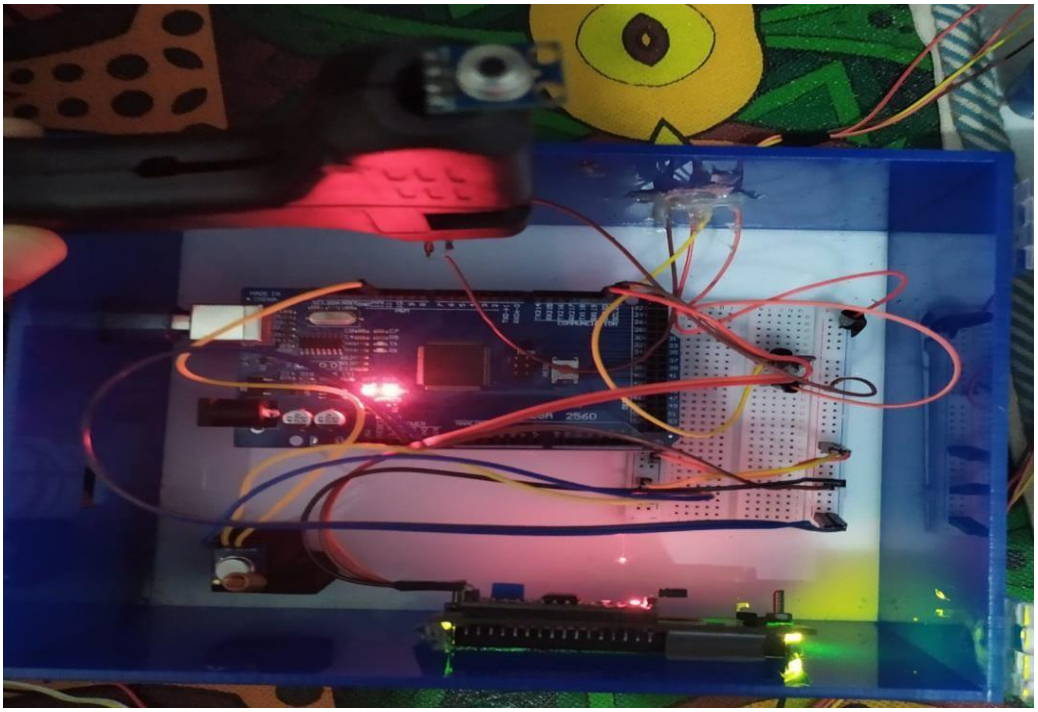
Interface Program Monitoring Suhu Tubuh Pada Visual Basic



Kode Program Sistem Monitoring Suhu Tubuh Pada *Visual Basic*

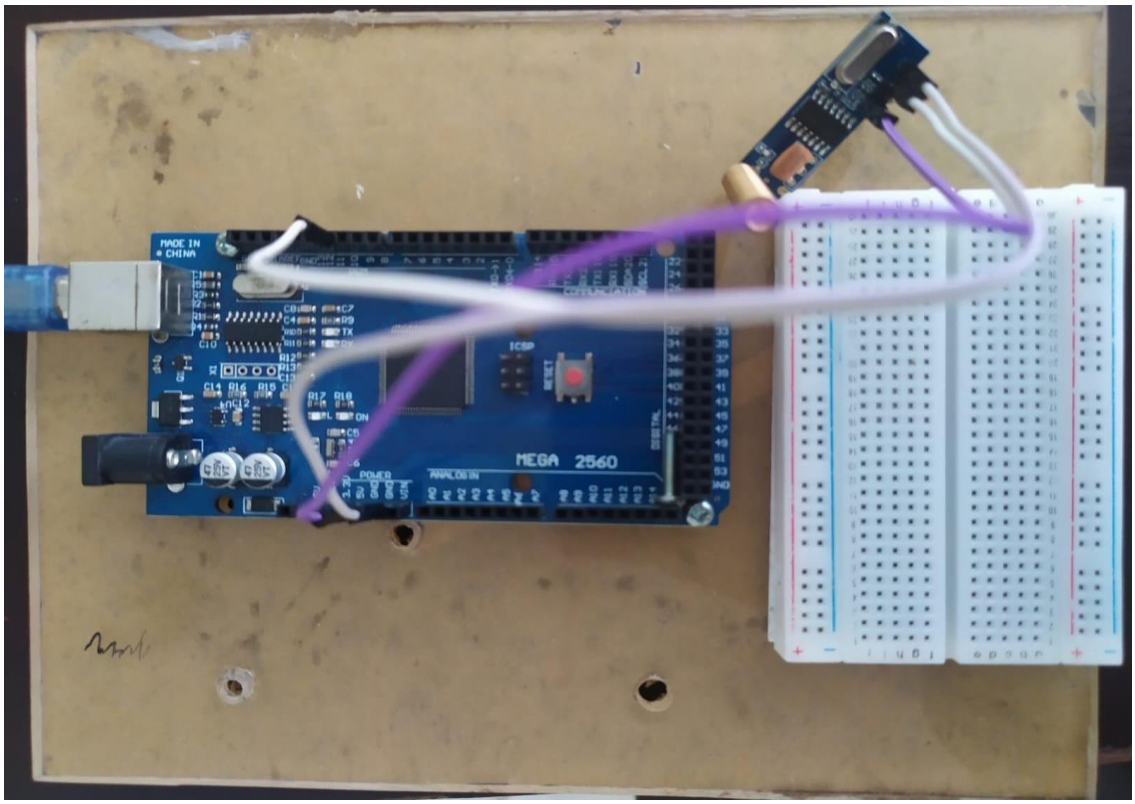


Pengambilan Data Monitoring Suhu Tubuh Pada *Visual Basic*



Komponen Arduino RF 433 MHz *Transmitter*

Tampak atas Komponen Arduino RF 433 MHz *Transmitter*



Komponen Arduino RF 433 MHz *Receiver*

Proses pengambilan data di Pelabuhan Paotere, Makassar

Proses pengambilan data di Pelabuhan ASDP Bira, Kabupaten bulukumba