LAMPIRAN 1 :

Sketch RTU

```
#include <Ethernet.h>
#include <SPI.h>
#include "Mudbus.h"
Mudbus Mb;                          //Port 502 (defined in Mudbus.h) MB_PORT

#include <SimpleModbusMaster.h>
#define baud 9600                   // Modbus RTU baudrate
#define timeout 2000                // Modbus master comm timeout
#define polling 200                 // the scan rate
#define retry_count 100
#define TxEnablePin 2
#define TOTAL_NO_OF_REGISTERS 30    // The total amount of available
memory on the master to store data

#include <RBD_Timer.h>              //
https://github.com/alextaujenis/RBD_Timer
RBD::Timer timer1;
RBD::Timer timer2;
RBD::Timer timer3;
RBD::Timer timer4;

byte mac[] = { 0x70, 0xA2, 0xDA, 0x0E, 0xAD, 0x8A };

IPAddress ip(192, 168, 1, 10);
IPAddress gateway(192,168,1,1);
IPAddress dns_server(192,168,1,1);
IPAddress subnet(255,255,255,0);

//timer declaration
```

```
unsigned long cycle_time = 0;              //capture Processor cycle time
unsigned long cycle_time_last = 0;         //last Processor cycle time
unsigned long cycle_time_current = 0;      //current Processor cycle time

//Digital Output signal Declaration
const int DO_0_pin = 7;                    //DO  close
const int DO_1_pin = 6;                    //DO  open
const int DO_2_pin = 5;                    //DO  close
const int DO_3_pin = 3;                    //DO  open

boolean DO_0 = false;                      // DO 0
boolean DO_1 = false;                      // DO 1
boolean DO_2 = false;                      // DO 2
boolean DO_3 = false;                      // DO 3

//Digital Input signal Declaration
const int DI_0_pin = A0;                   //DI pin 36
const int DI_1_pin = A1;                   //DI pin 37
const int DI_2_pin = A2;                   //DI pin 38
const int DI_3_pin = A3;                   //DI pin 39

boolean DI_0 = false;                      // DI 0
boolean DI_1 = false;                      // DI 1
boolean DI_2 = false;                      // DI 2
boolean DI_3 = false;                      // DI 3

//Internal TAG declaration
word discrete_array_1=0;                   // Temporary variable for DI 16 bit
word discrete_array_2=0;                   // Temporary variable for DO command
from modbus master 16 bit
```

```
boolean MBUS_DO_0 = false;          // DO 0
boolean MBUS_DO_1 = false;          // DO 1
boolean MBUS_DO_2 = false;          // DO 2
boolean MBUS_DO_3 = false;          // DO 3

enum
{ PACKET1,
 //PACKET2,
 //PACKET3,
 //PACKET4,
 TOTAL_NO_OF_PACKETS                // leave this last entry};

// Create an array of Packets to be configured
Packet packets[TOTAL_NO_OF_PACKETS];
// Masters register array
unsigned int regs[TOTAL_NO_OF_REGISTERS];

//---------------------------------------------
//---------------------------------------------

// Procedure read Digital Input from Arduino pin 36 to 51
void Read_DigitalInput() {
  DI_0 = digitalRead(DI_0_pin);
  DI_1 = digitalRead(DI_1_pin);
  DI_2 = digitalRead(DI_2_pin);
  DI_3 = digitalRead(DI_3_pin);

  //Copy value of digital input to variable discrete_array_1
  bitWrite(discrete_array_1,0, DI_0);
  bitWrite(discrete_array_1,1, DI_1);
  bitWrite(discrete_array_1,2, DI_2);
```

```
  bitWrite(discrete_array_1,3, DI_3);
 }

// Procedure write Digital Output arduino pin 22 to 31
void Write_Digital_Output()
{ digitalWrite(DO_0_pin, DO_0);
  digitalWrite(DO_1_pin, DO_1);
  digitalWrite(DO_2_pin, DO_2);
  digitalWrite(DO_3_pin, DO_3);

  //Copy value of variable discrete_array_2 into temporary variable
MBUS_DO_XX
  MBUS_DO_0 = bitRead(discrete_array_2,0);
  MBUS_DO_1 = bitRead(discrete_array_2,1);
  MBUS_DO_2 = bitRead(discrete_array_2,2);
  MBUS_DO_3 = bitRead(discrete_array_2,3); }

void ProgramTask(){

  //-----------------------------------------------------------------
  //-----------------------------------------------------------------
  //General treatment for DO pin 22 - 31 when modbus command ON arrival
  //than after 1 sec back to OFF

  //read Modbus address 40018 bit at 0 ; when value is TRUE
  //than DO pin 22 value ON for 1 seconds than soon back to OFF
  if (MBUS_DO_0 == true) {DO_0 = true ; timer1.restart();}
  if (timer1.isExpired()) {DO_0 = false ; timer1.stop();}


  //read Modbus address 40018 bit at 1 ; when value is TRUE
```

```
  //than DO pin 23 value ON for 1 seconds than soon back to OFF
  if (MBUS_DO_1 == true) {DO_1 = true ; timer2.restart();}
  if (timer2.isExpired()) {DO_1 = false; timer2.stop();}

  //read Modbus address 40018 bit at 2 ; when value is TRUE
  //than DO pin 24 value ON for 1 seconds than soon back to OFF
  if (MBUS_DO_2 == true) {DO_2 = true ; timer3.restart();}
  if (timer3.isExpired()) {DO_2 = false ; timer3.stop();}

  //read Modbus address 40018 bit at 3 ; when value is TRUE
  //than DO pin 25 value ON for 1 seconds than soon back to OFF
  if (MBUS_DO_3 == true) {DO_3 = true ; timer4.restart();}
  if (timer4.isExpired()) {DO_3 = false ; timer4.stop();}


  Mb.R[18] = 0;                    //re-write modbus register 40018 modbus
command after bounching treatment}

/*-----------------------------------------------------------------------
---PROCEDURE PROCESSOR DIAGNOSTIC
-----------------------------------------------------------------------
*/
// Procedure Frame work
void Frame_work() {
  //Create MCU cycle time
  cycle_time = millis();
  cycle_time_current = cycle_time - cycle_time_last;
  cycle_time_last = cycle_time;}

//Procedure MODBUS TCP slave (arduino)
void Modbus_COMM() {
```

```
  Mb.Run();
  //Digital Input
  Mb.C[0] = digitalRead(A0);
  Mb.C[1] = digitalRead(A1);
  Mb.C[2] = digitalRead(A2);
  Mb.C[3] = digitalRead(A3);

  //mapping power meter
  Mb.R[0] = (regs[0]);//kwh
  Mb.R[1] = (regs[1]);//kwh
  Mb.R[2] = (regs[2]);//tegangan
  Mb.R[3] = (regs[3]);//arus
  Mb.R[4] = (regs[4]);//arus
  Mb.R[5] = (regs[5]);//daya
  Mb.R[6] = (regs[6]);//daya
  Mb.R[7] = (regs[11]);//frekwensi

  //Digital inputs
  Mb.R[17]=(discrete_array_1);                // Modbus Data Digital Input

  //Digital outputs
  discrete_array_2 = Mb.R[18];               // Modbus Data Digital Output

  Mb.R[19] = (cycle_time_current);            // Modbus Data Processor cycle time
mSec}

void setup(){ Ethernet.begin(mac, ip, gateway, subnet);
Serial.begin (9600);

// Digital Input-Output initialization
  pinMode(A0, INPUT);
```

```
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);
  pinMode(A3, INPUT);

  pinMode(DO_0_pin, OUTPUT);
  pinMode(DO_1_pin, OUTPUT);
  pinMode(DO_2_pin, OUTPUT);
  pinMode(DO_3_pin, OUTPUT);


  modbus_construct(&packets[PACKET1], 5, READ_HOLDING_REGISTERS,
0, 15, 0);//
 //modbus_construct(&packets[PACKET2], 2, READ_HOLDING_REGISTERS,
103, 7, 7);//
 //modbus_construct(&packets[PACKET3], 3, READ_HOLDING_REGISTERS,
103, 7, 14);//

 modbus_configure(&Serial, baud, SERIAL_8N1, timeout, polling, retry_count,
TxEnablePin, packets, TOTAL_NO_OF_PACKETS, regs);

  //Timer preset value ; 1000 = 1000 mSec
  timer1.setTimeout(1000);
  timer2.setTimeout(1000);
  timer3.setTimeout(1000);
  timer4.setTimeout(1000);
  }

void loop()
{ modbus_update();                    //Initiate pooling data modbus Master RTU
(from other arduino slave)
  Frame_work();                      //Call procedure Frame_work
```

```
Read_DigitalInput();            //Call procedure Read_DigitalInput
Write_Digital_Output();         //Call procedure Write_Digital_Output
ProgramTask();                  //Call procedure ProgramTask
Modbus_COMM();                  //Call procedure Modbus_COMM }
```

LAMPIRAN 2

Kode Script HMI

```
Public Class FormView2
    Private iRTU As Integer = 0

    Private Sub btnBackHome_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        'Me.Panel1.Controls.Clear()
        'FormMain.Show()
        'Me.Panel1.Controls.Add(FormMain.PanelLogin)
        'FormMain.btnLogin.Visible = True
        'FormMain.btnConnect1.Visible = False
        'FormMain.btnDisconnect1.Visible = False
        'FormuUJI.Show()
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Dim iOpen1 = CoilReg(iRTU, 0)
        If iOpen1 = 1 Then
            If Not CBc.FillColor = Color.Red Then
                UpdateAlarm(3, Status_LastAlarmIndex(iRTU), "Inactive")
                CreateNewAlarm(iRTU, "LBS MOTORIZED CLOSE")
                Status_LastAlarmIndex(iRTU) = Me.dgAlarm.RowCount - 1
                CBc.FillColor = Color.Red
            End If
        Else
            CBc.FillColor = Color.Transparent
        End If
        Dim iOpen2 = CoilReg(iRTU, 1)
```

```
If iOpen2 = 1 Then
    If Not CBo.FillColor = Color.LimeGreen Then
        UpdateAlarm(3, Status_LastAlarmIndex(iRTU), "Inactive")
        CreateNewAlarm(iRTU, "LBS MOTORIZED OPEN")
        Status_LastAlarmIndex(iRTU) = Me.dgAlarm.RowCount - 1
        CBo.FillColor = Color.LimeGreen
    End If
Else
    CBo.FillColor = Color.Transparent
End If
Dim iOpen3 = CoilReg(iRTU, 2)
If iOpen3 = 1 Then
    If Not LR.Visible = False Then
        UpdateAlarm(3, Status_LastAlarmIndex(iRTU), "Inactive")
        CreateNewAlarm(iRTU, "LBS MOTORIZED - REMOTE")
        Status_LastAlarmIndex(iRTU) = Me.dgAlarm.RowCount - 1
        LR.Visible = False
    End If
Else
    If Not LR.Visible = True Then
        UpdateAlarm(3, Status_LastAlarmIndex(iRTU), "Inactive")
        CreateNewAlarm(iRTU, "LBS MOTORIZED - LOCAL")
        Status_LastAlarmIndex(iRTU) = Me.dgAlarm.RowCount - 1
        LR.Visible = True
    End If
End If

volt.Text = HoldReg(iRTU, 2) * 0.01 & ""
amp.Text = HoldReg(iRTU, 3) * 0.001 & ""
freq.Text = HoldReg(iRTU, 7) * 0.01 & ""
Daya.Text = HoldReg(iRTU, 5) * 0.001 & ""
```

```
        Dim iTime = Now.ToLocalTime
        ' Chart1.Series(0).Points.AddXY(iTime, HoldReg(iRTU, 2))
        Chart1.Series(0).Points.AddXY(iTime, HoldReg(iRTU, 3))
        'Chart1.Series(2).Points.AddXY(iTime, HoldReg(iRTU, 7))
        Chart1.Series(1).Points.AddXY(iTime, HoldReg(iRTU, 5))

    End Sub

    Private Sub FormView2_Disposed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Disposed
        For i = 0 To RTUsNum - 1
            Status_LastAlarmIndex(i) = 0
        Next
        '===
        'Private Sub FormAlarm_Disposed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Disposed
        '   For i = 0 To RTUsNum - 1
        'Status_LastAlarmIndex(i) = 0
        'Next

        '===
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
        ' Dim cTime = Now.ToLocalTime
        'Chart1.Series(0).Points.AddXY(cTime, HoldReg(iRTU, 0))
        'Chart1.Series(1).Points.AddXY(cTime, HoldReg(iRTU, 1))
        'Chart1.Series(2).Points.AddXY(cTime, HoldReg(iRTU, 2))
        'Chart1.Series(0).Points.AddXY(cTime, AI1.Text)
```

```
        'Chart1.Series(1).Points.AddXY(cTime, AI2.Text)
        'Chart1.Series(2).Points.AddXY(cTime, AI3.Text)
    End Sub

    Private Sub CB_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CBo.Click, CBc.Click
        FormSelect.Show()
        FormSelect.Timer1.Enabled = True
    End Sub

    Private Sub Back_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Back.Click
        FormMain.PanelMain.Controls.Clear()
        FormMain.PanelMain.Controls.Add(FormView1.Panel1)
    End Sub

End Class
```