DAFTAR PUSTAKA

[1] Y. Prayitno, R. Pramana, E. Prayetno, and J. Teknik Elektro, "Rancang Bangun Prototipe Sistem Kontrol Ballast Otomatis Pada Kapal," *J. Tek. Elektro Umr.*, vol. 1, pp. 1–9, 2017.

[2] I Tandra Novian, "PERANCANGAN ONE DESK MONITORING SYSTEM PADA LOADING UNLOADING KAPAL TANKER PERANCANGAN ONE DESK MONITORING SYSTEM PADA LOADING UNLOADING KAPAL," ITS, 2007.

[3] M. Corps, *Design , Monitoring*. Oregon, U.S.A, 2005.

[4] R. Wirhatnolo, *Monitoring evaluasi dan pengendalian: Konsep dan Pembahasan*. 2008.

[5] E. Amsler , G. M. Findley, HM & Ingram, "Performance Monitoring : Guidance for the modern workplace," vol. 70, pp. 12–19, 2009.

[6] M. Surveyor, "Cara Menghitung Minyak Di Kapal Tanker," 2014. http://marine-surveyor-indonesia.blogspot.com/2014/03/cara-menghitung-minyak-di-kapal-tanker.html?m=1.

[7] P. dkk Ongga, "Konsepsi mahasiswa tentang tekanan hidrostatis," 2009, pp. 181–185.

[8] F. Aulia, *Scilab (5.4.0)*, no. April. 2013.

[9] A. Kadir, *Buku Pintar Pemrograman Arduino*. Yogyakarta: Mediakom, 2015.

[10] F. Semiconductor, *Freescale Semiconductor Integrated Silicon Pressure Sensor On-Chip Signal Conditioned , Temperature Compensated and Calibrated*. Texas, Amerika Serikat, 2009.

[11] I. M. Yuliara, "Regresi Linier Sederhana," Jurusan Fisika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Udayana, 2016.

Lampiran 1. Coding Arduino

Tab 1 : Arduino_Code_Pengukuran_Volume_Tangki.ino

```
#include<OneWire.h>
long  dT = 1000;
int const pinAn[] = {A0, A1, A2, A3, A4, A5};
int const puluhan[] = {1,10,100,1000,10000};
int  pinIdxAn[]={A0,A0,A0,A0,A0,A0};
int  readValue[]={0,0,0,0,0,0};
char var;
uint8_t  nSample;  //Hz
uint16_t TSample = 1000/nSample;
int  nSensor;
uint8_t i,j,k;

uint8_t pinOneWire = 4;
uint8_t  numDev,numActiveDev,activeDev[10];

byte address[8];
byte ArrayAddr[10][8],tempArrayAddr[10][8];
byte scratchPad[9];
byte scratchPadDev[10][9];
float devReadData[10];

OneWire ds(pinOneWire);

void setup() {
  while(!Serial) { }
  Serial.begin(115200);
  while(!setupPin()){ }
   Serial.println(" starting the measurement process ");
```

```
  delay(100);
 // put your setup code here, to run once:
}

void loop() {
 // Format command:
 // setup : SP0123!   --> S; setup P; pin analog  0123; pin A0, A1, A2, A3
(maksimal A5, Uno)   !; penutup
 //       : ST20!     --> S; setup T; pengukuran berulang     20; jumlah pengukuran
sebanyak 20 kali
 //       : SO        --> S; setup O; pencarian alamat / address device
 // Read  : RA        --> R; read sensor    A; Analog  sensor
 //       : RO        --> R; read sensor    O; onewire sensor
 //
 if(Serial.available()) { //cek jika ada permintaan
     var = Serial.read();           //setup
     //Serial.println(var);
     delay(5);
     switch(var) {
       case 'S':                  //setup sampling rate
          while(!setupPin()){ }
       break;
       case 'R':                  //Read sensor
          var = Serial.read();
          if(var=='A')  readSensorAnalog();
          if(var=='O')  printOneWireData();
       break;
       case 'I':
          Serial.print("Jumlah Sensor Analog = ");
          Serial.println(nSensor);
          Serial.print("Jumlah Iterasi = ");
```

```
            Serial.println(nSample);
         break;
       } // end of switch
  }      // end of if check
}        // end of loop
```

Tab 2. _setupPin

```
bool setupPin(void) {
i = 0;
delay(5);
if(Serial.available()) {                    //
  switch(Serial.read()) {
    case 'P':
      while(1) {
       // if(Serial.available()) {
        int temp = Serial.read();
          if(temp>=48 && temp<=53) {
            pinIdxAn[i] = pinAn[temp-48];
            i=i+1;
            }
        else if(temp == '!') {
          break; }
        }
       nSensor = i;
       return 0;
    case 'O':
        searchDevice();
        if(tempArrayAddr[0][0]>0) {          // jika ada device onewire lanjutkan
          for(i=0;i<numDev;i++) {
            for(j=0;j<7;j++) {
              Serial.print(tempArrayAddr[i][j]);
```

```
        Serial.print(',');
       }
        Serial.println(ArrayAddr[i][7]);
     }
  numActiveDev = i;
  while(1) {
   i = 0;
   if(Serial.available()) {
    uint8_t temp = Serial.read();
     if(temp>=48 && temp<=53) {

      for(j=0;j<8;j++) {
       ArrayAddr[i][j] = tempArrayAddr[temp-48][j];
      }
      i=i+1;
       }
    else if(temp == '!') {
     numActiveDev = i;
     break; }
    }
  }
   }
   return 0;
case 'X':
   Serial.println(" Your setup ended.... ");
   return 1;
case 'T':
  uint8_t tempT[]={0,0,0,0,0,0};
  j=0;  uint8_t temp;
  while(1){
    temp = Serial.read();
```

```
        if(temp>=48 && temp<=57) {
          tempT[j] = temp-48;
          j=j+1;
          }
        else if(temp == '!') {
          break; }
        }
       uint8_t Value = 0;
       for(i=0;i<j;i++) {
         Value = Value + tempT[i] * puluhan[j-i-1];  //Hz
       }
       nSample = Value;
       return 0;
     }
  }
  return 0;
}
```

Tab 3. _1readAnalogSensor

```
void readSensorAnalog(void) {
   int16_t sensor; int j;
   for(j=0;j<nSensor;j++) {
     sensor = 0;
     if(nSample==0) nSample=1;
     for(int i=0;i<nSample;i++)  {
        sensor = sensor + analogRead(pinIdxAn[j]);
        }
        readValue[j] = sensor;
   }
   Serial.print('$');
   Serial.print('A');
```

```
   Serial.print(nSample);
   Serial.print(';');
   for(j=0;j<nSensor;j++){
     Serial.print(readValue[j]);
     if(j<nSensor-1) {
       Serial.print(';');
       }
     else {
       Serial.println('!');
     }
   }
}
```

Tab 4. _2OneWire

```
// Membaca semua alamat device, simpan pada alamat global ArrayAddr
void searchDevice(void) {
  i=0;
  while(ds.search(address)) {                    //berhenti setelah semua device address dibaca
     for(j=0;j<8;j++) {
       if(OneWire::crc8(address,7)==address[7]) {     // cek crc8 addres yang dibaca benar
          tempArrayAddr[i][j]=address[j];              // Simpan ke array address (kumpulan alamat)
          i++;
       }
     }
   // Read all address byte
   // byte    0: device model
```

```
   //           DS18S20MODEL / DS1820MODEL 0x10
   //           DS18B20MODEL 0x28
   //           DS1822MODEL  0x22
   //           DS1825MODEL  0x3B
   //           DS28EA00MODEL 0x42
   // byte 1-6 : unik dev address
   // byte    7: CRC dev Address
  }
  numDev = i;     //Jumlah alamat
}

bool readDeviceData(void) {
  // sends command for all devices on the bus to perform a temperature conversion
  ds.reset();
  //ds.select(address);
  ds.skip();            // semua device
  ds.write(0x44,1);      // start conversion, with parasite power on at the end
  delay(1000);          // maybe 750ms is enough, maybe not
  // we might do a ds.depower() here, but the reset will take care of it.
  // present = ds.reset();

  for(uint8_t j=0;j<numDev;j++) {       // baca data semua device
    for(uint8_t i=0;i<8;i++) { address[i] = ArrayAddr[j][i]; }
      ds.select(address);               // pilih device
      ds.write(0xBE);                   // Read Scratchpad
      for (uint8_t i = 0; i < 9; i++) {  // we need 9 bytes
        scratchPad[i] = ds.read();
          // Read all registers in a scratchPad
          // byte 0: temperature LSB
          // byte 1: temperature MSB
          // byte 2: high alarm temp
```

```
      // byte 3: low alarm temp
      // byte 4: DS18S20: store for crc
      //         DS18B20 & DS1822: configuration register
      //         TEMP_9_BIT  0x1F //  9 bit
      //         TEMP_10_BIT 0x3F // 10 bit
      //         TEMP_11_BIT 0x5F // 11 bit
      //         TEMP_12_BIT 0x7F // 12 bit
      // byte 5: internal use & crc
      // byte 6: DS18S20: COUNT_REMAIN
      //         DS18B20 & DS1822: store for crc
      // byte 7: DS18S20: COUNT_PER_C
      //         DS18B20 & DS1822: store for crc
      // byte 8: SCRATCHPAD_CRC
    }
// check CRC;
    if(OneWire::crc8(scratchPad,8)==scratchPad[8]) {
      // Convert the data to actual temperature
      // because the result is a 16 bit signed integer, it should
      // be stored to an "int16_t" type, which is always 16 bits
      // even when compiled on a 32 bit processor.
      int16_t raw = (scratchPad[1] << 8) | scratchPad[0];
      if (scratchPad[0]==0x10) {
        raw = raw << 3; // 9 bit resolution default
        if (scratchPad[7] == 0x10) {
          // "count remain" gives full 12 bit resolution
          raw = (raw & 0xFFF0) + 12 - scratchPad[6];
        }
      } else {
        byte cfg = (scratchPad[4] & 0x60);
        // at lower res, the low bits are undefined, so let's zero them
        if      (cfg == 0x00) raw = raw & ~7;   // 9 bit resolution, 93.75 ms
```

```
        else if (cfg == 0x20) raw = raw & ~3;   // 10 bit res, 187.5 ms
        else if (cfg == 0x40) raw = raw & ~1;   // 11 bit res, 375 ms
        //// default is 12 bit resolution, 750 ms conversion time
       }
       //celsius    = (float)raw / 16.0;
       //fahrenheit = celsius * 1.8 + 32.0;
       devReadData[j]=(float)raw / 16.0;
     }
  }
}

void printOneWireData(void) {
  delay(5);
  readDeviceData();
  Serial.print('$');
  Serial.print('O');
  if(numDev==0) {Serial.print(0);} else {Serial.print(1);}
  Serial.print(';');
  for (i=0;i<numDev;i++){
    if (i==numDev-1) {
      Serial.print(devReadData[i]);
      Serial.println('!');
      }
    else {
      Serial.print(devReadData[i]);
      Serial.print(';');
      }
   }
}
```

Lampiran 2. Coding Program GUI Scilab

```
// This GUI file is generated by guibuilder version 3.0
```

```
//////////
global h;
global Offsets;
global panel;
global NOffsets;
f=figure('figure_position',[376,40],'figure_size',[921,654],'auto_resize','on','background',[33],
'figure_name','Graphic window number %d');
//////////
delmenu(f.figure_id,gettext('File'))
delmenu(f.figure_id,gettext('?'))
delmenu(f.figure_id,gettext('Tools'))
toolbar(f.figure_id,'off')
handles.dummy = 0;
handles.Connect=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','center','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.0707371,0.22
45652,0.1383047,0.067971],'Relief','default','SliderStep',[0.01,0.1],'String','connect','Style','p
ushbutton','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','Connect','Callback','Con
nect_callback(handles)')
handles.Start=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','center','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.2207371,0.22
45652,0.1383047,0.067971],'Relief','default','SliderStep',[0.01,0.1],'String','start','Style','push
button','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','Start','Callback','Start_callb
ack(handles)')
handles.stop=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','center','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.0707371,0.14
45652,0.1383047,0.067971],'Relief','default','SliderStep',[0.01,0.1],'String','Stop','Style','push
button','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','stop','Callback','stop_callba
ck(handles)')
handles.close=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','center','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.2207371,0.14
45652,0.1383047,0.067971],'Relief','default','SliderStep',[0.01,0.1],'String','Close','Style','pus
hbutton','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','close','Callback','close_call
back(handles)')
handles.gr= newaxes();handles.gr.margins = [ 0 0 0 0];handles.gr.axes_bounds =
[0.4015487,0.1892754,0.5331858,0.6594928];
handles.volume=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.0829204,0.6255
797,0.255531,0.0778986],'Relief','default','SliderStep',[0.01,0.1],'String','
Volume','Style','text','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','volume','Callb
ack','')
handles.port=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.0829204,0.7255
```

```
797,0.255531,0.0778986],'Relief','default','SliderStep',[0.01,0.1],'String','
Port','Style','text','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','port','Callback','')
handles.tilt=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.0829204,0.5355
797,0.255531,0.0778986],'Relief','default','SliderStep',[0.01,0.1],'String','
Tilt','Style','text','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','tilt','Callback','')
handles.depth=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.0829204,0.4455
797,0.255531,0.0778986],'Relief','default','SliderStep',[0.01,0.1],'String','
Depth','Style','text','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','depth','Callback'
,'')
handles.nbport=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.1714602,0.7378
986,0.119469,0.057971],'Relief','default','SliderStep',[0.01,0.1],'String','
','Style','edit','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','nbport','Callback','')
handles.nbvol=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.1714602,0.6378
986,0.119469,0.057971],'Relief','default','SliderStep',[0.01,0.1],'String','
','Style','edit','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','nbvol','Callback','')
handles.nbmiring=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.1714602,0.5478
986,0.119469,0.057971],'Relief','default','SliderStep',[0.01,0.1],'String','
','Style','edit','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','nbmiring','Callback','')
handles.nbdepth=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[12],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','left','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.1714602,0.4578
986,0.119469,0.057971],'Relief','default','SliderStep',[0.01,0.1],'String','
','Style','edit','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','nbdepth','Callback','')
handles.judul=uicontrol(f,'unit','normalized','BackgroundColor',[-1,-1,-
1],'Enable','on','FontAngle','normal','FontName','Tahoma','FontSize',[16],'FontUnits','points','
FontWeight','normal','ForegroundColor',[-1,-1,-
1],'HorizontalAlignment','center','ListboxTop',[],'Max',[1],'Min',[0],'Position',[0.2565044,0.86
68116,0.4745575,0.1033333],'Relief','default','SliderStep',[0.01,0.1],'String',' MONITORING
VOLUME
TANGKI','Style','text','Value',[0],'VerticalAlignment','middle','Visible','on','Tag','judul','Callbac
k','')

//////////
// Callbacks are defined as below. Please do not delete the comments as it will be used in
coming version
//////////

function Connect_callback(handles)
```

```
//Write your callback for  Connect  here
   global h;
   global Offsets;
   global panel;
   com = handles.nbport.string;
   [h]=openSerialPort(strtod(com));
   disp(h);
   txt = mgetl(uigetfile(".txt"),3);    // file txt (notepad) yang isinya baris 1 nilai
Rhanyanilainyaajatanpatanda [ ].
   R = csvTextScan(txt(1,:),' ','.');   // file txt (notepad) yang isinya baris 2 nilai X hanya nilainya
saja tanpa tanda [ ].
   X = csvTextScan(txt(2,:),' ','.');   // file txt (notepad) yang isinya baris 3 nilai n hanya nilainya
saja tanpa tanda [ ].
   n = csvTextScan(txt(3,:),' ','.');
   [Offsets,panel]=genOffsets(R,X,n);
endfunction

function Start_callback(handles)
//Write your callback for  Start  here
global NOffsets;
tic;
   writeserial(h,'SP01!T1!X');
   writeserial(h,'RA');
   sleep(5000);
   Data = readserial(h);
   [readValue,sensor]=encode(Data);
   disp(readValue);
   a = readValue(1);
   b = readValue(2);
   disp(a);disp(b);
   t1 = (0.057*b)-35.877;
   t2 = (0.057*b)-35.877;
   depth = (t1+t2)/2;
   dt = t2 - t1;
   t = -16.33+((t1+t2)/2);
   disp(t);disp(depth);
   handles.nbdepth.string = string(depth);
   //Area=areapanel(Offsets,panel);
   //[vol] = volume(Offsets,panel,t);
   R = 16.75;A=6;
   vol1 = vol(R,depth,A);
   handles.nbvol.string = string(vol1);
   handles.nbmiring.string = string(atan(dt/63)*(180/%pi));
   An = atan(dt/63)*(180/%pi);
   Angle = [0 An 0];
   [NOffsets]=transformasi(Offsets,Angle);
   sleep(5000);
   delete(handles.gr.children);
   k = [16.33;16.33;-16.33;-16.33;16.33]
   l = [0;73;73;0;0]
   m = [t;t;t;t;t]
   plot3d(l,k,m)
      nx = length (NOffsets);
      n = 50;
```

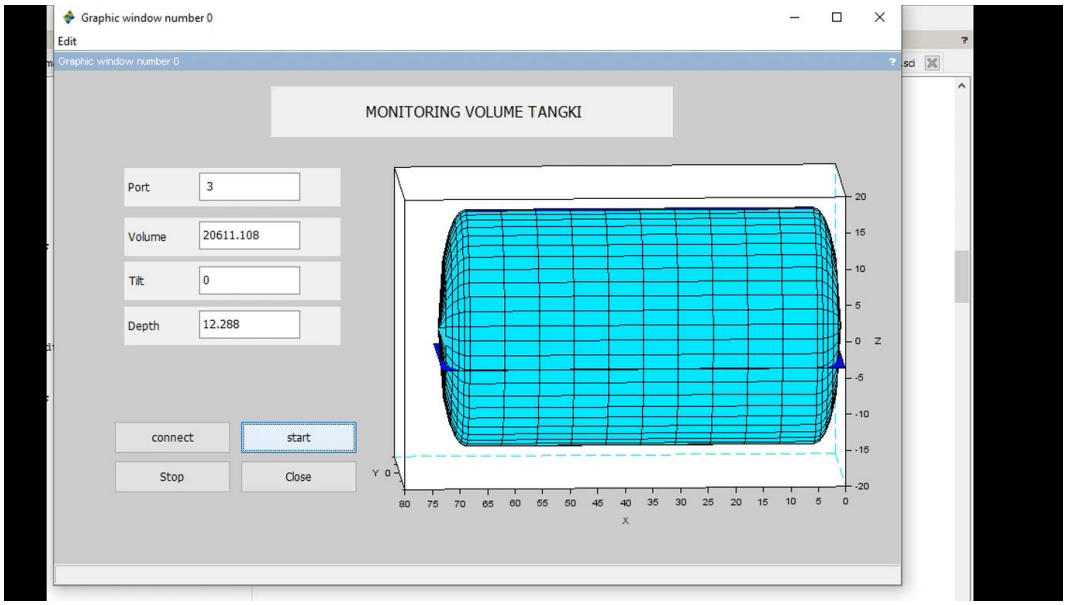
```
        for (i=1:nx)
          idx1 = i;
          idx2 = i+1;
          idx3 = n+i+2;
          idx4 = n+i+1;
          o = [NOffsets(idx1,1); NOffsets(idx2,1); NOffsets(idx3,1); NOffsets(idx4,1);
NOffsets(idx1,1)];
          p = [NOffsets(idx1,2); NOffsets(idx2,2); NOffsets(idx3,2); NOffsets(idx4,2);
NOffsets(idx1,2)];
          q = [NOffsets(idx1,3); NOffsets(idx2,3); NOffsets(idx3,3); NOffsets(idx4,3);
NOffsets(idx1,3)];
         plot3d(o,p,q);
        end

  //plot3dTempOffsets ();
toc;
endfunction

function stop_callback(handles)
//Write your callback for  stop  here
  disp(closeserial(h));
endfunction

function close_callback(handles)
//Write your callback for  close  here

endfunction

function [Offsets, panel]=genOffsets(R, X, n)
  // R array jari-jari lingkaran tabung R =[r1,r2,r3,dst]
  // n jumlah potongan / slice radius
  // L array jarak2; L =[m x1,x2,x3, dst]
  // Jumlah L dan R harus sama
  nX = length(X);
  nP = 1;
  Offsets = [];
  offPanelX = [];    offPanelY = [];    offPanelZ = [];
  for(i=1:nX)
   [y,z]=offsetCircle(R(i),n);
    x = X(i);
    ny = length(y);
    for(ii=1:ny)
      if(abs(y(ii,1))<0.000000001)  y(ii,1) =0; end
      if(abs(z(ii,1))<0.000000001)  z(ii,1) =0; end
    end
   Xx = ones(ny,1)*x; Yy = y; Zz = z;
   Offsets = [Offsets; Xx Yy Zz];
   //susun offset node
   if(i>1) then   // susunan node dimuali pada index 2
     for(j=1:n)
        idx1 = (i-2)*(n+1)+j;
        idx2 = (i-2)*(n+1)+j+1;
        idx3 = (i-1)*(n+1)+j+1;
        idx4 = (i-1)*(n+1)+j;
```

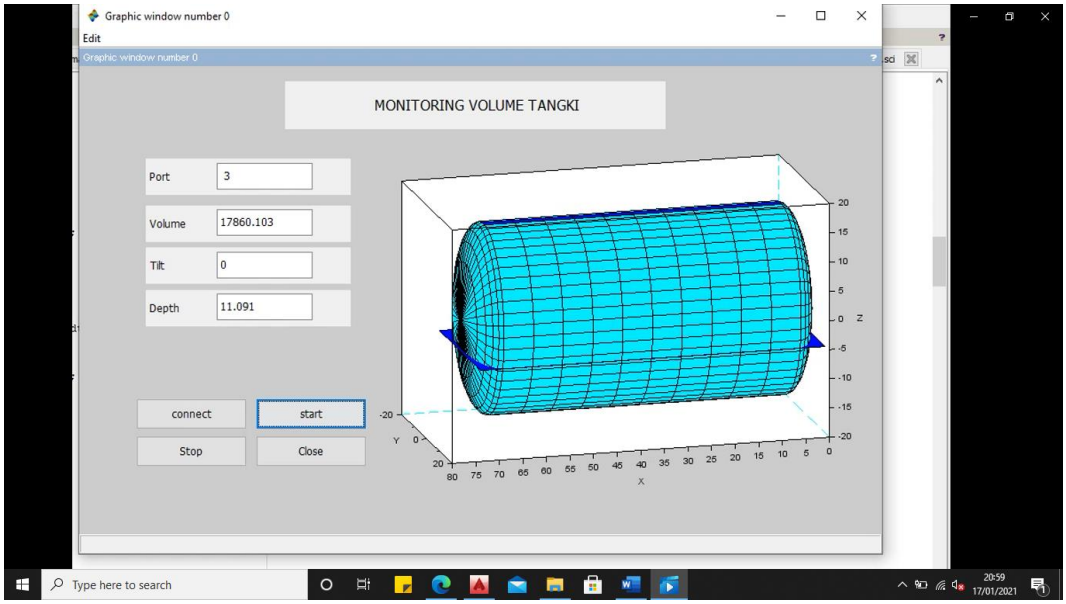
```
            panel(nP,:) = [idx1 idx2 idx3 idx4 idx1];
          offPanelX(:,j) = [Offsets(idx1,1); Offsets(idx2,1); Offsets(idx3,1); Offsets(idx4,1);
Offsets(idx1,1)];
          offPanelY(:,j) = [Offsets(idx1,2); Offsets(idx2,2); Offsets(idx3,2); Offsets(idx4,2);
Offsets(idx1,2)];
          offPanelZ(:,j) = [Offsets(idx1,3); Offsets(idx2,3); Offsets(idx3,3); Offsets(idx4,3);
Offsets(idx1,3)];
          nP = nP+1;
        end
        plot3d(offPanelX,offPanelY,offPanelZ);
     end
   end
endfunction

function Area=areapanel(Offsets, panel)
   nP = size(panel,1);
   for(i=1:nP)
    OffsetPan = Offsets(panel(i,:),:);
    Area(i,:)=areapoligon(OffsetPan);
   end
endfunction

function [y, z]=offsetCircle(r, n)
   alfa = 0:360/n:360;
   alfa = alfa' * (%pi/180);
   y = r*cos(alfa);
   z = r*sin(alfa);
endfunction

// buat coding mencari luasan dan menghitung titik berat
function [Area]=areapoligon(OffsetsPanel)
   X = OffsetsPanel(:,1)';
   Y = OffsetsPanel(:,2)';
   Z = OffsetsPanel(:,3)';
   N = size (OffsetsPanel,1)-1;
   //inisiasi area
   Axy = 0;
   Ayz = 0;
   Azx = 0;
   for (i = 1:N)
      Axy =Axy+(X(i)*Y(i+1))-(X(i+1)*Y(i));
      Ayz =Ayz+(Y(i)*Z(i+1))-(Y(i+1)*Z(i));
      Azx =Azx+(Z(i)*X(i+1))-(Z(i+1)*X(i));
   end
   Axy = Axy/2;   if(abs(Axy)<1e-13) then Axy=0; end
   Ayz = Ayz/2;   if(abs(Ayz)<1e-13) then Ayz=0; end
   Azx = Azx/2;   if(abs(Azx)<1e-13) then Azx=0; end
   //Menghitung Luasan
   Area = [Axy Ayz Azx];
endfunction

function [vol]=volume(Offsets, panel, t)
   np = size (panel,1);
   vol =0;
```

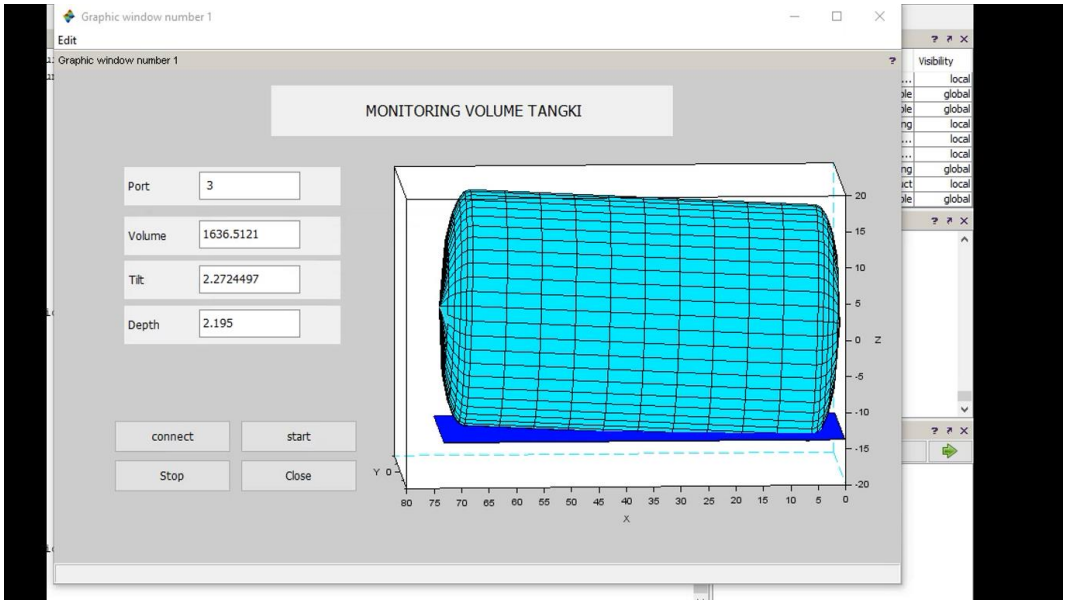
```
  for i = 1:np,
    OffsetsPanel = Offsets(panel(i,1:4),:);
    C = mean (OffsetsPanel,1);
    if C(3)< t then
      [Area]=areapoligon(OffsetsPanel);
      Az=Area(3);
      volz = (t-C(3)) * Az;   //1 cm3 = 0.001 liter
    else
      volz = 0;
    end
    vol = vol+volz;
  end
endfunction

function [h]=openSerialPort(com)
  h = openserial(com,"115200,n,8,1");
endfunction

function [readValue, sensor]=encode(Data)
  A=ascii(Data);
  m = length(A);
  j=1; k=0; ii=1;  perpuluhan=[1 10 100 1000 10000];
  for(i=1:m)
    if(A(i)==ascii('$'))   // start to read
      i = i + 1;
      if(A(i)==ascii('A')) sensor = "Analog";  end
      if(A(i)==ascii('O')) sensor = "OneWire"; end
      i = i + 1;
      while(1)
        if(A(i) == ascii(';') || A(i) == ascii('!'))
          sensorValue(ii) = perpuluhan(1,k:-1:1) * temp(1:k);
          k=0;
          ii = ii + 1;
        else
            k = k + 1;
          temp(k) = A(i)-48;
        end
        if(A(i)==ascii('!')) break; end
        i = i + 1;
      end
    end
  end
  readValue = sensorValue(2:ii-1)/sensorValue(1);
endfunction

function vol1=vol(R, depth, A)
  q = (R^2)*acos((R-depth)/R);
  w = (R-depth)*(((2*R*depth)-(depth^2))^(1/2));
  e = 22/7*A*(3*R-depth)*depth*depth/3/R;
  vol1 = (q-w)*63+e;
endfunction

function [NOffsets]=transformasi(Offsets, Angle)
  //Offset merupakan data offset gading semu yang terdiri atas X Y Z
```

```
  //Angle variabel sudut putar pada sumbu-x,y,z --> [ax ay az]
   a=Angle(1)*%pi/180;
   b=Angle(2)*%pi/180;
   c=Angle(3)*%pi/180;
   [m,n]=size(Offsets);
   T1 = [1 0 0; 0 cos(a) sin(a);0 -sin(a) cos(a)];  // bidang Y dan Z, X tetap
   T2 = [cos(b) 0 -sin(b); 0 1 0; sin(b) 0 cos(b)]; // bidang X dan Z, Y tetap
   T3 = [cos(c) sin(c) 0;-sin(c) cos(c) 0; 0 0 1];  // bidang X dan Y, Z tetap
   for(i=1:m)
      temp = Offsets(i,:)';
      temp = T1 * temp;
      temp = T2 * temp;
      temp = T3 * temp;
      NOffsets(i,:) = temp';
   end
endfunction

function plot3dTempOffsets()
   nx = length (NOffsets);
   n = 50;
         for (i=1:nx)
           idx1 = i;
           idx2 = i+1;
           idx3 = n+i+2;
           idx4 = n+i+1;
           o = [NOffsets(idx1,1); NOffsets(idx2,1); NOffsets(idx3,1); NOffsets(idx4,1);
NOffsets(idx1,1)];
           p = [NOffsets(idx1,2); NOffsets(idx2,2); NOffsets(idx3,2); NOffsets(idx4,2);
NOffsets(idx1,2)];
           q = [NOffsets(idx1,3); NOffsets(idx2,3); NOffsets(idx3,3); NOffsets(idx4,3);
NOffsets(idx1,3)];
           plot3d(o,p,q);
         end
endfunction
```

Lampiran 3. Dokumentasi Kegiatan






Graphic window number 1
Edit
Graphic window number 1
MONITORING VOLUME TANGKI
Port
3
Volume
1636.5121
Tilt
2.2724497
Depth
2.195
connect
start
Stop
Close
Visibility