

# SISTEM MONITORING LEVEL PADA HOPPER BERBASIS MIKROKONTROLLER ATMEGA 16 DAN PC DI PT. HANAMPI SEJAHTERA KAHURIPAN

<sup>1)</sup>Samsul Fahmi, <sup>2)</sup>Misbah, <sup>3)</sup>Hendra Ariwinarno,

<sup>1, 2, 3)</sup>Jurusan Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Gresik

Jl. Sumatera No 101, Gresik 61121, Jawa Timur

E-mail:over\_fahmi@yahoo.co.id <sup>1)</sup>, misbah.grs@gmail.com <sup>2)</sup>, hendra.umg@gmail.com <sup>3)</sup>

## ABSTRAK

PT Hanampi Sejahtera Kahuripan perusahaan pembuat pupuk pelepasan terkendali terdapat beberapa tempat penyimpanan untuk menampung bahan baku produksi maupun hasil produksi dipantau dengan *level transmitter*. Kerusakan pada *level transmitter* buatan China yang terhubung dengan program DCS *Advantool* menyebabkan terganggunya proses produksi menyebabkan kerugian. Untuk mengatasinya maka dibuat alat alternatif monitoring berbasis mikrokontroler ATMEGA 16 dengan menggunakan sensor ultrasonik. Untuk memantau level diunit-unit hopper.

Dalam penelitian ini dibuat desain monitoring menggunakan komputer sebagai master yang berkomunikasi serial memakai RS 232 dengan mikrokontroler dan sensor ultrasonic sebagai komponen pengirim dan penerima gelombang suara. Metode pengujian alat ini memanfaatkan refleksi gelombang ultrasonik pada medium yang dilalui mulai dari kerapatan rendah pada zat cair hingga padat. Jika gelombang ultrasonik berjalan melalui sebuah medium, secara matematis besarnya jarak dapat dihitung.

Dari hasil Penelitian ini desain pengendali secara terprogram dapat ditampilkan di komputer menggunakan program *Borland Delphi* sebagai pusat *monitoring* dan *controlling* terhadap dinamika proses berupa data level untuk mengetahui persentase volume material di dalam hopper, penggunaan clock 16.0 MHz diperlukan untuk menstabilkan penerimaan dan pengiriman data.

Kata kunci : Sensor ultrasonik, mikrokontroler, dan komputer.

## 1. PENDAHULUAN

PT. Hanampi Sejahtera Kahuripan adalah perusahaan pupuk pertama di Indonesia yang menggunakan metode pembuatan pupuk pelepasan terkendali.

Pelepasan terkendali adalah proses pembuatan pupuk dengan cara melapisi pupuk urea dengan belerang dan lilin, hal ini bertujuan agar nutrisi pada pupuk urea tersebut tidak cepat terurai atau mencair ketika terkena udara bebas. Pupuk ini

diberi nama SCU (*Sulfur Coating Urea*).

Dalam proses produksi di perusahaan terdapat beberapa *hopper*, *hopper* adalah tempat penampungan bahan pendukung untuk produksi pupuk diantaranya: urea, belerang dan lilin. Pada *hopper* tersebut terdapat beberapa *level switch* yang terhubung dengan PLC (*Programable Logic Controller*) dan DCS (*Distributed Control System*) merk SUPCON, material tersebut dalam proses produksi ditampung pada *hopper* untuk material padat dan *Meltitank* untuk cair yang lokasinya saling berjauhan. Sebelumnya di perusahaan tersebut menggunakan program DCS *Advantool* keluaran SUPCON *Made In China*. Untuk memonitoring dan menjalankan proses produksi secara keseluruhan, tetapi saat ini transmitter buatan SUPCON yang ada pada beberapa *hopper* jika mengalami kerusakan mengakibatkan beberapa material yang ada di dalam *hopper* meluber keluar atau kehabisan stok di *hopper* karena tidak terpantau dapat menyebabkan kerugian pada perusahaan. Perusahaan selalu memperbaiki atau menggantinya dengan mendatangkan *vendor* SUPCON dari China selaku pemilik lisensi yang membutuhkan waktu dan biaya yang sangat mahal. Oleh karena itu perlu adanya sistem monitoring terpusat terpisah dari program *Advantool*. Sistem *monitoring* tersebut menggunakan mikrokontroler, mikrokontroler mempunyai beberapa keunggulan disamping sebagai *chip programmable*, mempunyai fitur-fitur yang tidak kalah dengan DCS dan PLC. Disamping itu mikrokontroler dapat dianalogikan dengan sebuah sistem komputer yang dikemas dalam sebuah *chip*. Artinya bahwa didalam sebuah IC mikrokontroler sudah

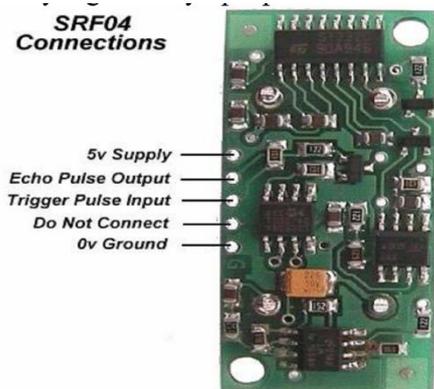
terdapat kebutuhan agar *mikroprosesor* dapat bekerja, yaitu meliputi *mikroprosesor*, ROM, RAM, I/O, dan *clock* seperti halnya yang dimiliki sebuah komputer (Bejo, 2008). Fitur lain dari mikrokontroler yaitu adanya komunikasi serial. Komunikasi ini bisa dimanfaatkan sebagai pemantau program yang sedang berjalan. Untuk memantau suhu, tekanan, level kemudian hasilnya dapat dilihat dari monitor dengan ditransmisikan ke komputer secara serial (Heriyanto dan Wisnu, 2008). Integrasi antara komputer dan mikrokontroler dirancang sebagai *monitoring* dan *controlling* alternatif di unit-unit *Chiller* AHU PT. Sakatama (Hippong, 2011)

Pada penelitian ini akan dibuat sistem *monitoring* level masing-masing *hopper* berbasis mikrokontroler dan komputer untuk mengatasi permasalahan *monitoring* level *hopper* yang ada pada salah satu perusahaan pupuk pelepasan terkendali PT. Hanampi Sejahtera Kahuripan. Sistem ini menggunakan komputer sebagai *monitoring* dan *controlling* yang berkomunikasi serial RS232 atau RS485 dengan mikrokontroler dan sensor ultrasonik yang terpasang di atas masing-masing *hopper* material.

## 2. DASAR TEORI

Pada bab ini diuraikan beberapa landasan teori yang dipergunakan untuk menyelesaikan masalah dan pengerjaan pada penelitian ini yang mencakup, prinsip kerja sensor ultrasonic, mikrokontroler ATmega16, komunikasi serial, aplikasi program *Borland Delphi* dan *CodeVision AVR*.

## 2. 1. Sensor Ultrasonik



Gambar 2.1. Sensor Ultrasonik  
( sumber : [www.parallax.com](http://www.parallax.com) )

Berdasarkan gambar 2.1 Sensor ultrasonik adalah salah satu piranti elektronika yang dapat mengukur jarak didepannya dengan menggunakan sifat dari gelombang suara yang dipantulkan. Sensor (jarak) ultrasonik menggunakan media suara untuk melakukan pengukuran jarak. Sensor ultrasonik terdiri dari 2 bagian, yaitu bagian pemancar dan bagian penerima, bagian pemancar ultrasonik akan memancarkan gelombang ultrasonik pada frekwensi lebih dari 20kHz, biasanya menggunakan frekwensi 40kHz. Kemudian sinyal tersebut akan dipantulkan kembali setelah mencapai suatu benda di depannya. Sinyal tersebut akan di terima kembali oleh bagian penerima sinyal. Jarak antara sensor dengan bidang pantul nya dapat diketahui dengan menggunakan rumus :  $S = V.t/2$  Dimana S adalah jarak sensor dengan bidang pantul, V adalah kecepatan suara (340m/s), dan t adalah selisih waktu antara pemancaran sinyal sampai diterima kembali oleh penerima ultrasonik. Sensor ultrasonik tersebut di hubungkan ke mikrokontroler untuk melakukan penghitungan waktu beserta jarak pantulnya. ( [www.parallax.com](http://www.parallax.com) )

## 2.2 Mikrokontroler ATmega 16

Mikrokontroler ATmega 16 adalah merupakan salah satu keluarga mikrokontroler AVR. Mikrokontroler AVR adalah mikrokontroler RISC (*Reduce instruction set compute*) 8 bit keluaran Atmel, yang merupakan salah satu mikrokontroler yang banyak digunakan saat ini.

AVR memiliki keunggulan dibanding mikrokontroler lain, keunggulan mikrokontroler AVR yaitu memiliki kecepatan eksekusi program yang lebih cepat karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*. Selain itu mikrokontroler AVR memiliki memiliki fitur yang lengkap (ADC internal, *EEPROM internal*, *Timer/Counter*, *Watchdog Timer*, PWM, Port I/O, Komunikasi serial, I2C dll [1].

## 2.3 Komunikasi Serial

Komunikasi Serial adalah pengiriman data satu persatu secara berurutan dalam waktu tertentu. Komunikasi serial ada 2 macam, yaitu komunikasi sinkron dan asinkron. Komunikasi sinkron dilakukan dengan menambahkan sinyal sinkronisasi. Komunikasi asinkron dilakukan dengan menetapkan kecepatan bit (*baud rate*) dan menyisipkan beberapa bit protokol, yaitu bit start, *parity* dan bit stop.

## 2.4 Komunikasi Serial RS-232

Standar RS-232 merupakan aturan mengenai level tegangan, konektor dan aturan komunikasi. Standar RS232 memiliki level tegangan antara -3 sampai -15 Volt untuk logika *high*, dan antara +3

sampai +15 Volt untuk logika *low*. Level tegangan antara -3 sampai +3 Volt tidak didefinisikan, sebab di daerah ini kemungkinan adalah *noise*.

## 2.4 Code Vision AVR

*Code Vision AVR* adalah suatu alat bantu pemrograman (*Programming tools*) yang bekerja dalam lingkungan pengembangan perangkat lunak yang terintegrasi. *CodeVision AVR* merupakan software bahasa pemrograman bahasa C yang digunakan dalam menghasilkan program bahasa *assembly* untuk dimasukkan kedalam mikrokontroler keluarga AVR dari ATMEL Corp.

Khusus untuk library fungsi, disamping library standar (seperti fungsi-fungsi matematik, manipulasi *string*, pengaksesan memori dan sebagainya), *CodeVisionAVR* juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi library yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I2C, IC RTC (*Real time Clock*), sensor suhu LM75, SPI (*Serial Peripheral Interface*) dan lain sebagainya.

LCD (*Liquid Cristal Display*) 2X16 character

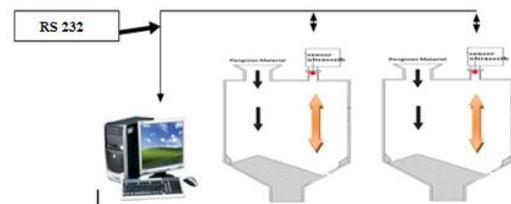
LCD (*Liquid Cristal Display*) adalah suatu *display* dari bahan cairan kristal yang pengoprasiannya menggunakan sistem *dot matriks*. LCD banyak digunakan sebagai *display* dari alat-alat elektronika seperti kalkulator, *multitester digital*, Jam digital dan sebagainya. Bentuk fisik LCD 2x16 dapat dilihat pada gambar 2 dibawah ini.



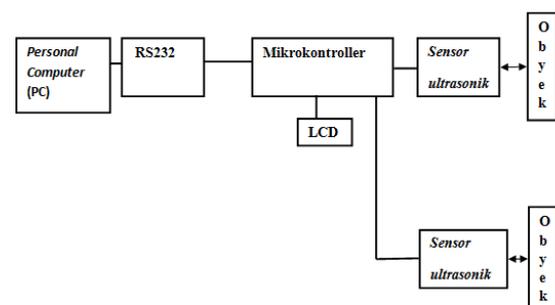
Gambar 2. LCD 2x16

## 3. PERENCANAAN DAN PEMBUATAN ALAT

Perencanaan dimulai dari pembuatan blok sistem monitoring, seperti pada gambar 3 dibawah ini.



Gambar 3.1 Potongan melintang Hopper, posisi sensor ultrasonik, dan computer



Gambar 3.2 Skema blok diagram sistem

Keterangan dari gambar 3.2 skema blok diagram dia atas adalah sebagai berikut :

1. PC/laptop  
Sebagai *host* sistem komunikasi untuk *monitoring* dan *controlling* unit-unit

hopper. Menggunakan *Software Borland Delphi* . berada di control room.

## 2. Komunikasi Serial RS232

Sebagai penerima dan pengirim sinyal-sinyal *input/output* dari mikrokontroler ke komputer atau PC.

## 3. Mikrokontroler dan LCD

Sebagai *Hardware* yang dapat berinteraksi dengan *plant* dan berkomunikasi pada komputer. LCD sebagai media untuk menampilkan data % level

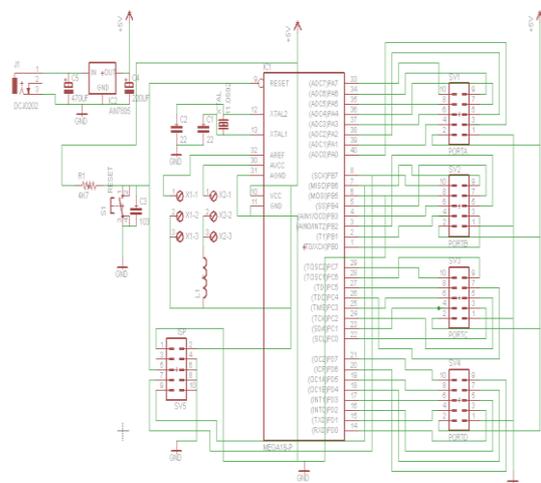
## 4. Sensor Ultrasonik

Sebagai pemancar dan penerima frekuensi gelombang suara untuk mendapatkan nilai jarak benda, akan ditempatkan 5cm diatas permukaan maket hopper .

### 3.1. Perancangan Hardware

#### Sistem Minimum ATmega16

Mikrokontroler ATmega16 dalam tugas akhir ini digunakan untuk memproses data dari modul ACS712 yang berupa sinyal analog dan merubahnya menjadi sinyal digital melalui ADC yang ada pada mikrokontroler tersebut, memprosesnya dan menampilkannya pada LCD 2x16.

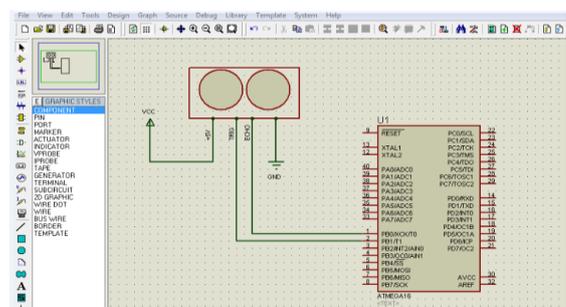


Gambar 3.3. Sistem Minimum ATmega16

Dari gambar sistem minimum diatas menggambarkan rangkaian mikrokontroler ATmega32 secara *singlechip*. Rangkaian sistem minimum ini terdiri atas kristal, kapasitor, dan resistor dengan *suplai* tegangan sebesar 5 Volt DC. Pada *pin reset* terhubung dengan kapasitor 10 nF, resistor 4K7  $\Omega$ , dan sebuah *push button*, rangkaian *reset* tersebut berfungsi untuk mengembalikan mikrokontroler ke kondisi awal.

### 3.2. Perancangan Sensor ultrasonik

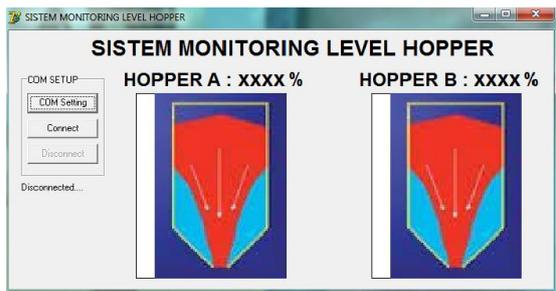
Sensor ini memiliki 3 kaki (Pin) yang berfungsi sebagai berikut Pin Ground, Pin Supply 5V, Pin Input dan Output, seperti Gambar 4.6. Dalam perancangan alat ini, pin input output sensor ultrasonik dihubungkan dengan port B.0 pada AVR sedangkan pin supply 5V dan Ground dihubungkan dengan catu daya. Sensor ini berfungsi untuk mengubah besaran fisis yang berupa jarak menjadi besaran elektris tegangan. Sensor ini bekerja dengan mengirimkan gelombang ultrasonik yang memiliki frekuensi diatas pendengaran manusia kemudian akan menyediakan pulsa keluaran yang berhubungan langsung dengan waktu yang dibutuhkan bagi pantulan gelombang ultrasonik untuk mencapai sensor.



Gambar 3.4. Rangkaian Sensor ultrasonik dan mikrokontrol

### 3.3. Perancangan Software

Software yang dirancang adalah software yang mendukung perangkat keras (*hardware*). Sehingga terdapat dua bagian software yaitu software untuk mikrokontroler Atmega16 menggunakan CodeVisionAVR dan perancangan tampilan antarmuka (*interfacing*) menggunakan form visual dengan bahasa pemrograman Borland Delphi 7. Yang merupakan form utama untuk menampilkan sistem monitoring terintegrasi, yaitu terdapat tampilan-tampilan *indicator* angka dalam persen yang merepresentasikan volume atau isi hopper dimonitoring. Selama proses monitoring, apabila terjadi angka menunjukkan 0% sampai 100 % pada hopper yang dideteksi secara rutin,



Gambar 3.5 Tampilan Monitoring Borland Delphi

### 4. PENGUJIAN DAN ANALISA ALAT

Untuk mengetahui keberhasilan dari perancangan dan pembuatan sistem, maka dilakukan pengujian dari alat tersebut. Pengujian Elektronika dilakukan di Laboratorium Elektronika, dan pengujian alat secara keseluruhan dilakukan di sekitar kampus Universitas Muhammadiyah Gresik. Dari pengujian alat didapatkan hasil sebagai berikut;

Tabel 1. hasil pengujian alat pada benda cair (air)

Real (cm)	Sensor A (cm)	Sensor B (cm)	Error A (cm)	Error B (cm)
5	5	5	0	0
9	9.12	9.09	0.01	0.013
18	18.1	18	0.005	0
24	24.06	24.02	0.0025	0.001
30	30	30	0	0

Tabel 2. hasil pengujian alat pada benda padat (pasir)

Real (cm)	Sensor A (cm)	Sensor B (cm)	Error A (cm)	Error B (cm)
5	5	5	0	0
9	9.08	9.03	0.01	0.003
18	18.01	18.21	0.001	0.01
24	24.07	24.1	0.003	0.004
30	30	30	0	0

Tabel 3. hasil pengujian alat pada plat datar

Real (cm)	Sensor A (cm)	Sensor B (cm)	Error A (cm)	Error B (cm)
5	5	5	0	0
10	10	10	0	0
15	15	15	0	0
20	20	20	0	0
30	30	30	0	0

Dari hasil percobaan di atas Data Tabel.4 menunjukkan bahwa data dari sensor untuk plat datar relative stabil, sedangkan untuk Data Tabel.3 percobaan pada benda padat (butiran pasir) dan Data Tabel.2 percobaan pada air memiliki selisih antar 0.1 mm sampai 0.4 mm jika dibandingkan dengan alat ukur mistar, nilai ini didapat dengan menjumlahkan semua nilai error dari setiap pengujian dibagi jumlah pengujian (5 kali).

Secara rumus untuk mencari nilai konversi jarak (cm) real ke level

volume dalam persen adalah sebagai berikut ;

$$X = \frac{30 \text{ cm} - A \text{ cm}}{25 \text{ cm}} \times 100 = \dots \%$$

A= nilai jarak (cm) dari pembacaan sensor

Contoh perhitungan ;

$$1. \quad X = \frac{30 \text{ cm} - 30 \text{ cm}}{25 \text{ cm}} \cdot 100 \%$$

$$= 0 \%$$

$$2. \quad X = \frac{30 \text{ cm} - 18 \text{ cm}}{25 \text{ cm}} \cdot 100 \%$$

$$= 48 \%$$

$$3. \quad X = \frac{30 \text{ cm} - 9 \text{ cm}}{25 \text{ cm}} \cdot 100 \%$$

$$= 84 \%$$

Untuk mencari nilai *Error* dari setiap pengukuran menggunakan rumus sebagai berikut :

$$X = \frac{\text{Nilai Real} - \text{Nilai Aktual}}{\text{Nilai Real}} \cdot 100 \%$$

Contoh perhitungan ;

$$X = \frac{5 \text{ cm} - 5 \text{ cm}}{5 \text{ cm}} \cdot 100 \% = 0 \%$$

$$X = \frac{18 \text{ cm} - 18.21 \text{ cm}}{18 \text{ cm}} \cdot 100 \% = 0.01 \%$$

$$X = \frac{24 \text{ cm} - 24.07 \text{ cm}}{24 \text{ cm}} \cdot 100 \% = 0.003 \%$$



Gambar 4.1 Tampilan uji coba alat

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Berdasarkan pengujian dan analisa yang telah dilakukan pada proyek akhir, maka dapat disimpulkan bahwa :

1. Kesalahan pengukuran jarak dengan transduser Ultrasonik relatif kecil yaitu sekitar 0.003 persen(%) sampai 0.01 persen(%). Dalam implementasinya arah transduser *transmitter* dan *receiver* harus

tegak lurus terhadap permukaan air dalam hal ini sebagai media percobaan.

2. Pengukuran jarak yang dilakukan oleh sensor Ultrasonik terhadap benda padat memiliki prosentase *error* antara 0,005 persen(%) sampai 0,01 persen(%) yang relatif lebih kecil dibandingkan terhadap benda cair.

3. Adanya perbedaan antara hasil pengukuran yang dilakukan oleh sensor Ultrasonik terhadap benda padat, cair dan plat datar yang relative stabil.

4. Hasil pengujian komunikasi serial RS232 mikrokontroler, dalam pengiriman data protokol ke PC secara *real time* dilakukan tiap data per 1 detik secara kontinyu respon penerimaan data yang ditampilkan ke LCD dan PC kurang begitu stabil pada percobaan hopper yang lebih kecil

5. Pada pengerjaan proyek akhir ini sudah dapat melakukan monitoring dan pengambilan data level material yang ada di *Hopper* dengan menggunakan aplikasi *Borland Delphi*.

### 5.2 Saran

Dalam pengerjaan proyek akhir tentunya tidak lepas dari berbagai macam kekurangan dan kelemahan baik sistem maupun peralatan yang dibuat, untuk itu demi kesempurnaan proyek akhir dapat memberikan beberapa catatan :

Pastikan komunikasi serial antara mikrokontroler dengan PC bekerja dengan baik, sehingga sensor dapat saling kirim-terima dengan baik pula.

Penggunaan modul *transceiver* dengan jarak yang lebih jauh, dan penempatan yang tepat agar proses kirim-terima dari alat ukur ke Mikrokontroler dapat berjalan dengan baik dan tidak terjadi kehilangan data.

Untuk menstabilkan data yang dikeluarkan maka *clock* pada program *CodeVision AVR* perlu mengganti komponen *crystal* 11,059 MHz menjadi 16,0 MHz.

## 5.DAFTAR PUSTAKA

[1]. Andrianto, Heri. 2008. Pemrograman Mikrokontroler AVR ATMEGA 16 Menggunakan Bahasa C (Code Vision AVR). Informatika:Bandung.

[2]. Hippong, 2011. Sistem Monitoring Chiller Berbasis mikrokontroler ATmega 8535 dan PC di PT.Sakatama, Gresik, Tugas Akhir, Universitas Muhammadiyah Gresik.

[3]. Heriyanto M. Ary, P. Wisnu, Adi, 2008, *Pemrograman Bahasa C untuk Mikrokontroler ATmega8535*. Andi : Yogyakarta

[4]. Bejo, Agus. 2008. C dan AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATmega 8535. Graha Ilmu:Yogyakarta.

[5]. Ulfah,Mediaty.2011. Pengujian Sensor Ultrasonik PiNG untuk pengukuran Level Ketinggian dan Volume Air, jurnal Ilmiah “Elektrikal Enginiring” UNHAS.

[6]. Haiduc, Pavel, 2008, *Codevision AVR User Manual*, HP Info Tech s.r.l. Setiawan, Rahmad. 2006. *MIKROKONTROLER MCS-51*. Yogyakarta : Graha Ilmu.

[7]. Pranata, Antony 2003, *Pemrograman Borlan Delphi Edisi 6*. Andi : Yogyakarta

[8]. <http://www.atmel.com/ATMega32>.

[9]. <http://www.parallax.com>

## LAMPIRAN

Pembuatan programan dilakukan dengan bantuan *Shoftwar Borland Delphi* denngan listing sbb ;

```
unit sf;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, CPort, StdCtrls, Gauges, jpeg, ExtCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
ComPort1: TComPort;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Gauge1: TGauge;
```

```
Listbox1: TListBox;
```

```
Listbox2: TListBox;
```

```
Label5: TLabel;
```

```
Label6: TLabel;
```

```
GroupBox1: TGroupBox;
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
Gauge2: TGauge;
```

```
Label1: TLabel;
```

```
Image1: TImage;
```

```
Image2: TImage;
```

```
Label7: TLabel;
```

```
Label8: TLabel;
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure Button3Click(Sender: TObject);
```

```
procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
procedure ComPort1RxChar(Sender:
```

```
TObject; Count: Integer);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```

Form1: TForm1;

implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender:
TObject);
begin
comport1.ShowSetupDialog;
end;

procedure TForm1.Button2Click(Sender:
TObject);
begin
//try

ComPort1.Open;
//if comport1.Connected then
//begin
button2.Enabled:= false;
button3.Enabled:= true;
label1.Caption := 'Connected....';
//end

end;

procedure TForm1.Button3Click(Sender:
TObject);
begin
comport1.Close ;
button2.Enabled:= true;
button3.Enabled:= false;
label1.Caption := 'Disconnected....';

end;

procedure TForm1.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
comport1.Close;
end;

procedure TForm1.ComPort1RxChar(Sender:
TObject; Count: Integer);

begin
//PROGRAM AWAL
//Comport1.ReadStr (terima,rxcount);
//Comport1.ReadStr (terima,1); ~~~~~
//memo1.text := memo1.text + terima;
~~~~~
//label2.Caption:= terima;
//gauge1.Progress :=strtoint(terima);

//PROGRAM BARU
repeat
begin
comport1.ReadStr(A,1);
DataA:=DataA+A;
end;
until A='A';
s:=TStringList.Create;
s.Delimiter:='A'; //string 'A' dipisah dari
data
s.DelimitedText:=DataA; //data murni
dimasukkan ke variabel DataA
listbox1.Items:=s;
label2.Caption:= listbox1.Items[0];
gauge1.Progress :=strtoint(listbox1.Items[0]);
listbox1.Clear;

repeat
begin
comport1.ReadStr(B,1);
DataB:=DataB+B;
end;
until B='B';
s:=TStringList.Create;
s.Delimiter:='B'; //string 'A' dipisah dari
data
s.DelimitedText:=DataB; //data murni
dimasukkan ke variabel DataA
listbox2.Items:=s;
label5.Caption:= listbox2.Items[0];
gauge2.Progress := strtoint(listbox2.Items[0]);
listbox2.Clear;

end;

end.

Var
//rxcount : integer;
//terima : string;
A,B,DataA,DataB : string;
s :TStringList;

```