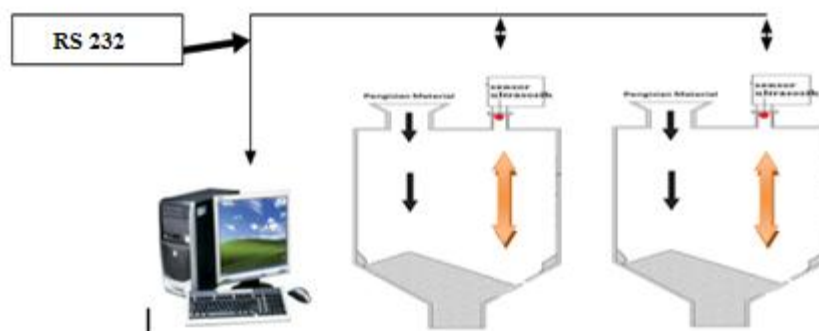


BAB III

PERENCANAAN DAN PEMBUATAN ALAT

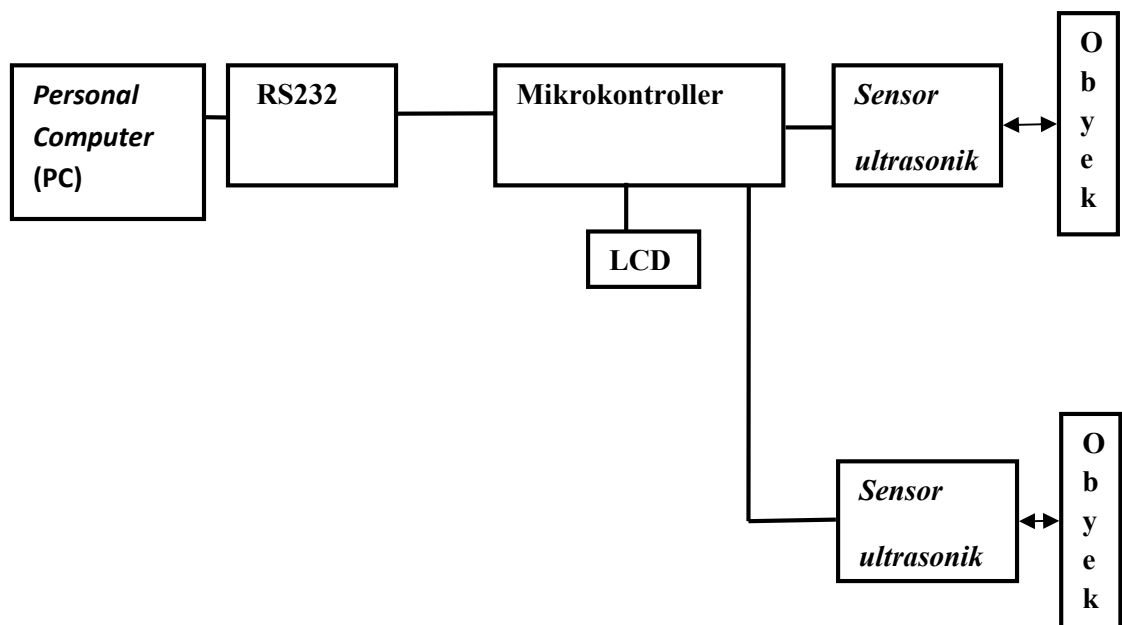
3.1. Perancangan *Hardware*

Dalam suatu *plant* produksi di PT. Hanampi Sejahtera Kahuripan bisa terdapat dua atau lebih unit *hopper* yang digunakan untuk menampung bahan baku atau bahan hasil olahan produksi berkapasitas 12 ton sampai 18 ton tergantung bahan material dan berat jenisnya. Untuk membuat sistem monitoring yang terintegrasi, pemantauan dapat dilakukan oleh operator produksi dari ruang control produksi melalui *Personal Computer* (PC) sebagai master yang berkomunikasi secara serial dengan mikrokontroler dan sensor ultrasonik. Komunikasi menggunakan RS-485 *multidrop network*. Sensor ultrasonik dan mikrokontroler dipasang diatas masing-masing *unit hopper* akan mengirim dan menerima sinyal-sinyal ke computer (PC).



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

. Sesuai dengan gambar 3.1. blok diagram sistem monitoring *hopper*, untuk mendesain sistem monitoring *hopper* berbasis PC dan mikrokontroler dengan memanfaatkan USART (*Universal Synchronous Asynchronous Receiver Transmitter*) yang tersedia pada *Personal Computer* (PC) dan mikrokontroler, maka yang perlu direncanakan dan dirancang adalah pertama desain meliputi kombinasi rangkaian sistem minimum mikrokontroler Atmega16, rangkaian input untuk mikrokontroler terisolasi menggunakan sensor ultrasosik serta *IC line driver* RS-232



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.

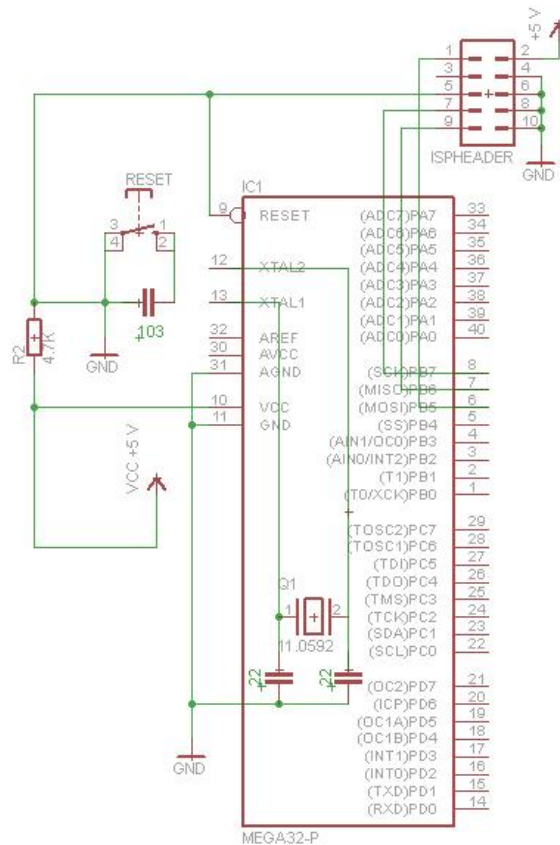
3. Mikrokontroler dan LCD

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 .

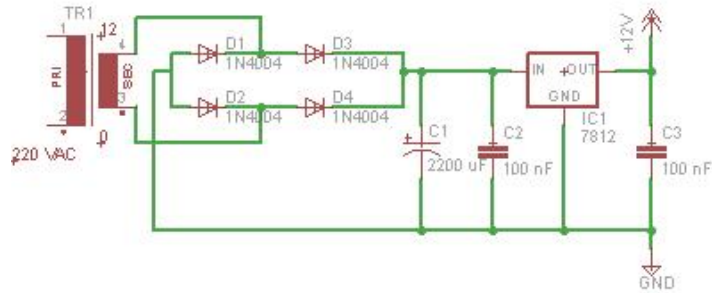
3.1.1. Sistem Minimum Mikrokontroler Atmega16



Gambar 3.3. Rangkaian Sistem Minimum ATmega 16.

Pada gambar 3.3. sistem minimum merupakan rangkaian yang diperlukan untuk mikrokontroler dapat bekerja. Meliputi pemberian tegangan supply (VCC) 5 VDC, rangkaian *clock* 11 Mhz, *push button* untuk *reset* dan soket *header* untuk memasukkan program ke mikrokontroler melalui komunikasi *Serial Peripheral Interface* (SPI).

3.1.2. Rangkaian Catu Daya (*Power Supply*)



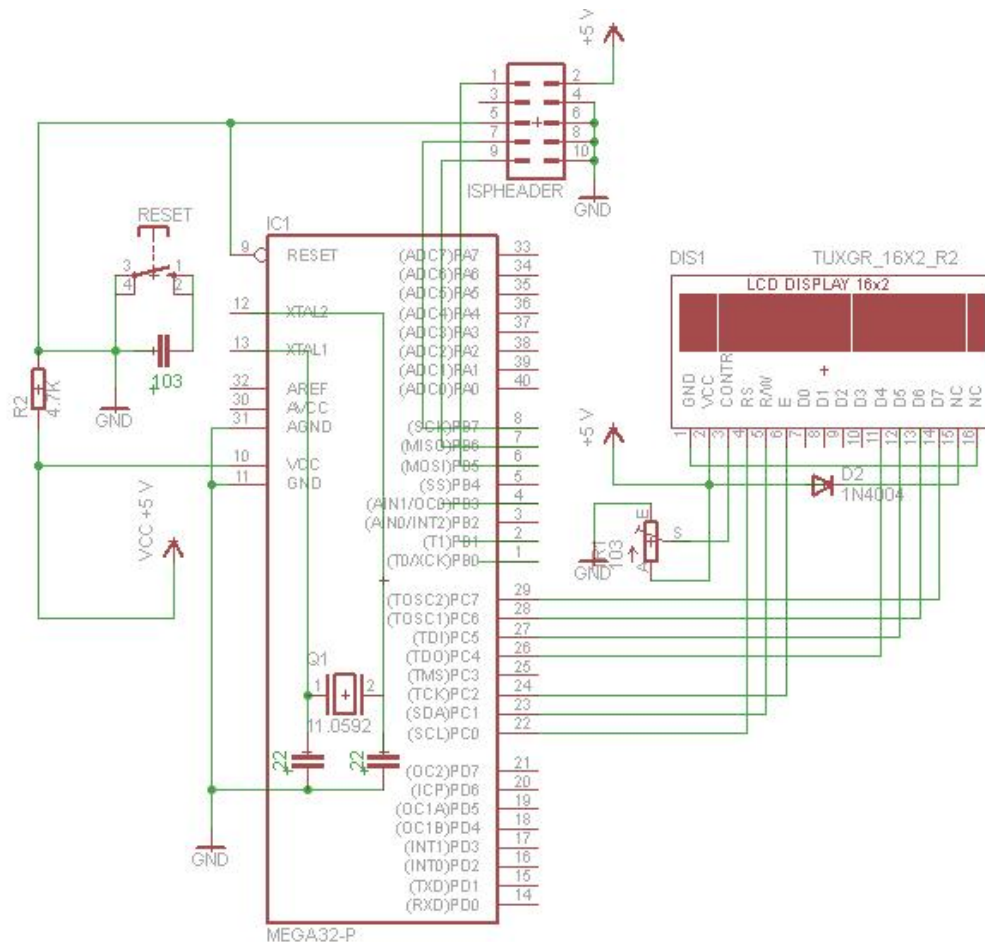
Gambar 3.4. Rangkaian Catu Daya 12 VDC

Rangkaian catu daya atau *power supply* digunakan untuk menyuplai tegangan keseluruhan komponen alat ini, yaitu pada sistem minimum dan *line driver* RS-485, serta converter RS-485 ke RS-232. Tegangan 220 VAC diturunkan oleh trafo menjadi 12 VAC selanjutnya oleh *diode bridge* disearahkan menjadi 12 VDC. Keluaran dari diode bridge kemudian masuk ke IC regulator LM7812 yang fungsinya adalah untuk menstabilkan tegangan pada 12 V.

3.1.3. Rangkaian LCD (*Liquid Crystal Display*) 16 x 2

LCD 2x16 sangat mudah diaplikasikan pada mikrokontroler AVR ATmega32. LCD 2X16 artinya LCD tersebut didesain untuk menampilkan data sebanyak 2 baris 16 kolom. Pada gambar 3.5. merupakan rangkaian *interface* LCD dengan mikrokontroler AVR ATmega16. Pada rangkaian tersebut dapat dilihat Pin 1 dari LCD terhubung dengan *ground*, Pin 2 merupakan VCC dari LCD dihubungkan dengan pin VCC mikrokontroler AVR ATmega16. Sedangkan pin 3 berfungsi mengatur kontras LCD, sehingga pada rangkaian diatas Pin 3

terhubung dengan sebuah *varibel resistor* (VR) 10 K Ω . LCD memerlukan tiga jalur kontrol dan delapan jalur data (untuk mode 8 *bit*) atau empat jalur data (untuk mode 4 *bit*). Ketiga jalur kontrol yang dimaksud adalah Pin 4 (RS), Pin 5 (R/W), Pin 6 (E). Pin 4 (RS) merupakan Pin *register selection signal* yang berfungsi memilih *register* yang akan digunakan, Pin ini terhubung pada Pin 14 (PC.0). Untuk Pin 5 (R/W) merupakan Pin *read/write signal* yang bila diberi logika '0', akan terpilih *write*. dan bila diberi logika '1', maka akan terpilih *read*. Pada saat Pin R/W berlogika rendah, informasi pada jalur data berupa pengiriman data ke LCD (*write*). Sedangkan ketika Pin R/W berlogika tinggi, berarti sedang dilaksanakan pengambilan data dari LCD (*read*). Pin 5 LCD ini dihubungkan dengan Pin 15 (PC.1). Sedangkan Pin 6 (E) adalah Pin *Enable* yang berfungsi mengaktifkan *read/write* data, yang digunakan untuk memberitahu LCD kalau kita akan berkomunikasi dengannya. Sebelum mengirim data ke LCD Pin ini di buat berlogika tinggi dahulu. Kemudian jalur kontrol yang lain di-setting, pada saat bersamaan data yang akan dikirim ditempatkan pada jalur data. Setelah semua siap, Pin *Enable* dibuat berlogika rendah. Transisi dari logika tinggi ke logika rendah ini akan memberitahu LCD untuk mengambil data pada jalur kontrol dan jalur data. Pin *Enable* dihubungkan dengan Pin 16 (PC.2) pada mikrokontroler AVR ATMega16.

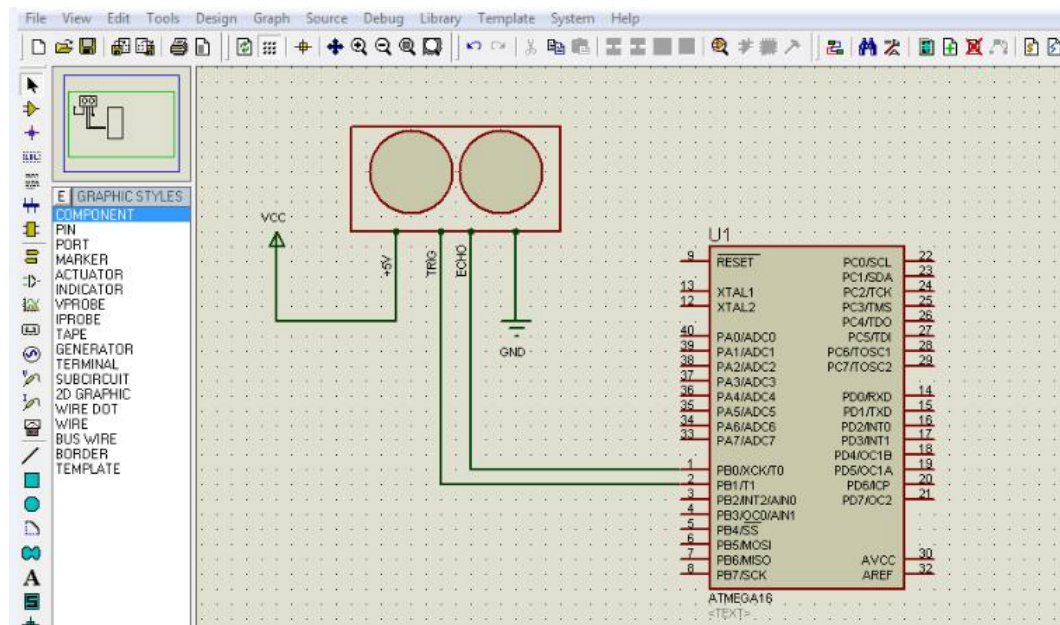


Gambar 3.5. Rangkaian LCD 2x16 dan mikrokontroler

3.1.4. 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

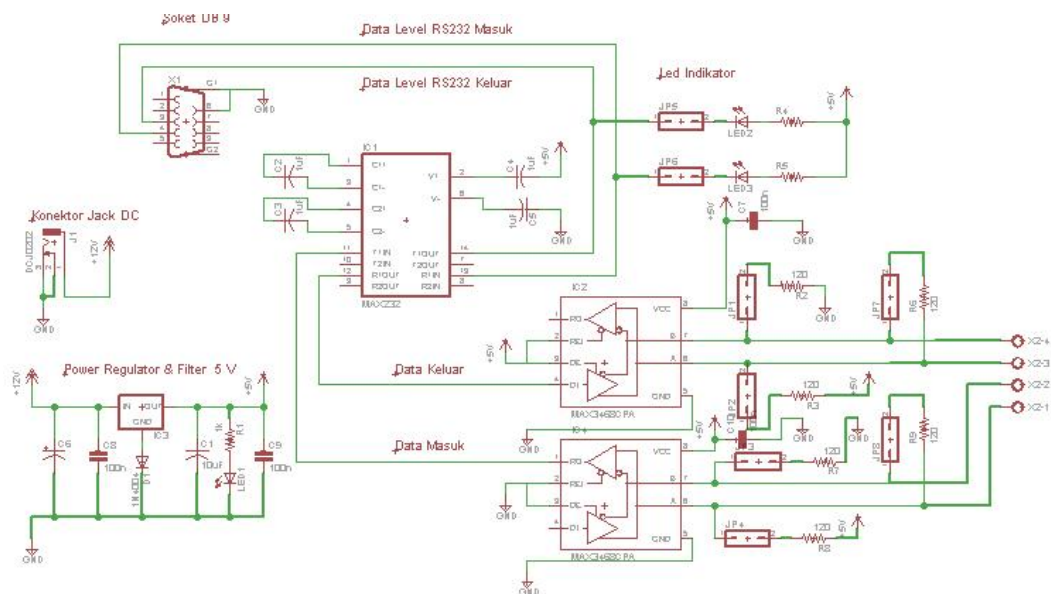
kemudian akan menyediakan pulsa keluaran yang berhubungan langsung dengan waktu yang dibutuhkan bagi pantulan gelombang ultrasonik untuk mencapai sensor.



Gambar 3.6. Rangkaian Sensor ultrasonik dan mikrokontrol

3.1.5. Rangkaian *Converter* RS-232

Level tegangan komunikasi serial antara mikrokontroler dan komputer tidak sama, Mikrokontroler bekerja pada 0V untuk logika rendah dan 5V untuk logika tinggi. Sedangkan komputer +25V untuk logika rendah (*space*) dan -25V untuk logika tinggi (*mark*). Untuk itu diperlukan rangkaian penerjemah (*converter*). Sehingga komunikasi *master slave* dapat berlangsung.

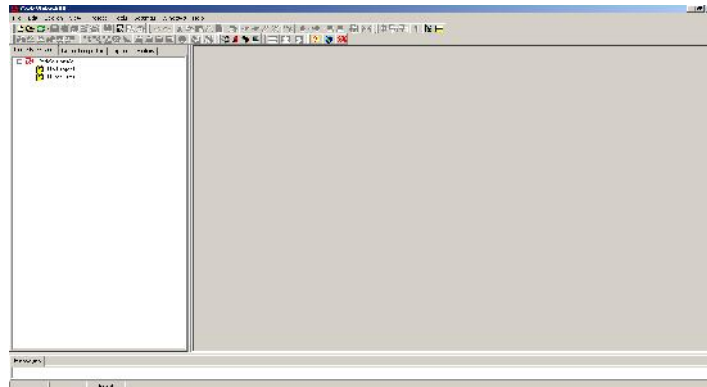


Gambar 3.7. Rangkaian Converter RS-232

IC MAX232 merupakan RS-232 *line driver*, berfungsi untuk menerjemahkan data serial dari level RS-232 ke RS-485 atau sebaliknya dari level RS-485 ke RS232.

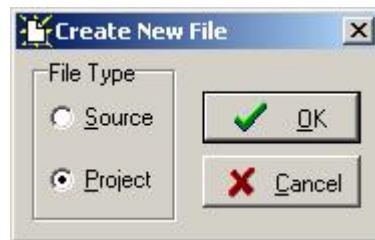
3.1.6. Pemrograman Menggunakan CodeVisionAVR

CodeVisionAVR merupakan sistem *software C-cross compiler*, dimana program dapat ditulis menggunakan bahasa C. Dengan menggunakan pemrograman bahasa C diharapkan waktu disain (*development time*) akan menjadi lebih singkat. Setelah program dalam bahasa C ditulis dan dilakukan kompilasi tidak terdapat kesalahan/*error*, maka proses download dapat dilakukan. Mikrokontroler AVR ATmega 32 mendukung sistem download secara ISP (*In-System Programming*).



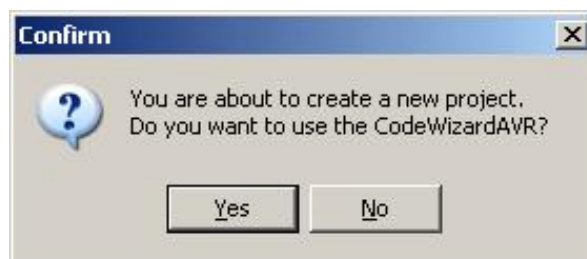
Gambar 3.8 Tampilan *CodeVision AVR*

Untuk memulai pemrograman menggunakan *CodeVisionAVR* pilih pada menu **File>New**, kemudian akan muncul kotak dialog pada gambar sebagai berikut:.



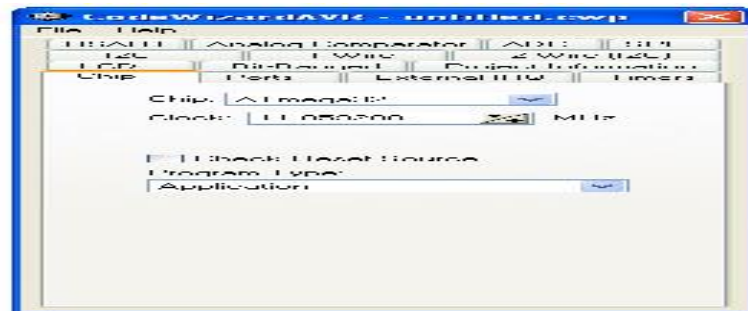
Gambar 3.9. Tampilan *New File*

Pilih project kemudian tekan OK, maka akan muncul kotak dialog pada gambar berikut.



Gambar 3.10. Tampilan *option di CodeWizard AVR*

Pilih Yes untuk menggunakan *CodeWizardAVR*. *CodeWizardAVR* digunakan untuk membantu dalam mengenerate program, terutama dalam konfigurasi *Port*, *Timer*, penggunaan fasilitas-fasilitas seperti LCD, *interrupt* dan sebagainya. Pada sistem kunci elektronik ini hanya memerlukan beberapa dari fasilitas *CodeWizardAVR*, antara lain setting chip, *port* setting, LCD, dan komunikasi serial menggunakan USART. Pertama kali yang harus dilakukan adalah mensetting chip dan *clock* (kristal) yang digunakan. Pada gambar 4.15. terlihat chip dipilih ATmega32 karena sistem distribusi listrik digital ini berbasis mikrokontroler AVR ATmega16, sedangkan *clock* dipilih 11.059200 sesuai dengan kristal yang digunakan pada rangkaian minimum mikrokontroler AVR.

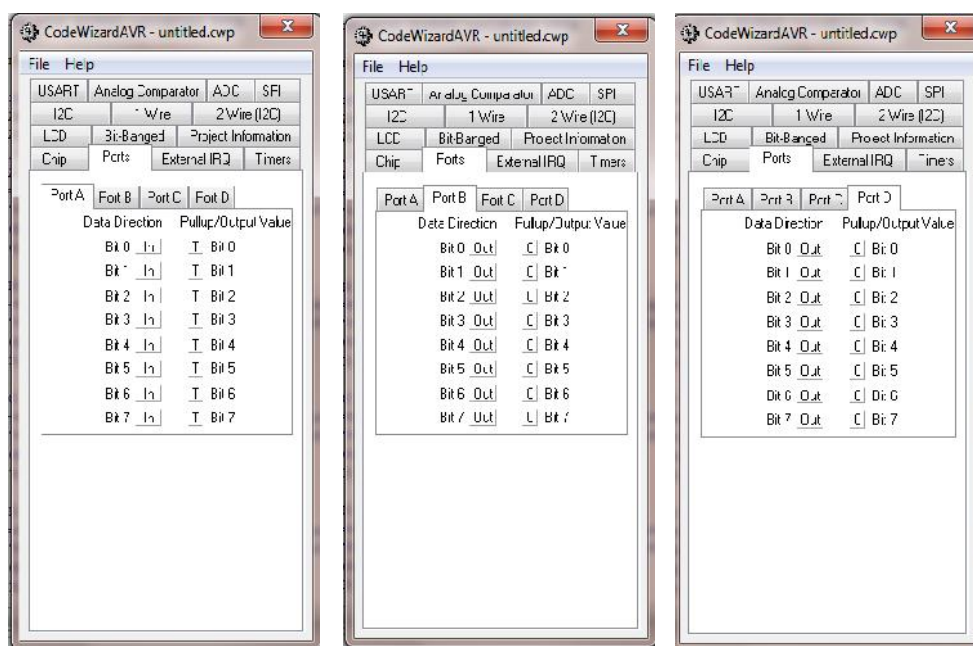


Gambar 3.11. Tampilan seting chip

3.1.7. Seting *Port* Menggunakan *CodeWizardAVR*

Setting port dilakukan untuk memilih port mana saja yang akan dijadikan *input* atau *output*. Pada Robot Penampung dan Pembuang Sampah Berbasis Mikrokontroler, Pin A (PA0, PA1, PA2, PA3, PA5, PA7) mikrokontroler AVR ATmega32 dipilih sebagai *inputan* dari sensor. Pin A (PA0, PA1, PA2, PA3) untuk *inputan* dari sensor garis, Pin A (PA5) untuk *inputan* sensor indicator penuh, dan A (PA7) berfungsi sebagai *inputan* dari sensor PIR (*Passive Infra*

Red). Sedangkan Port B (PB0, PB1, PB3) dan Port D (PD0, PD1, PD2, PD3, PD4, PD5, PD6, PD7) adalah *output* dari mikrokontroler AVR ATmega32. Port B (PB0, PB1, PB3) untuk inputan *driver* motor buka tutup, dan Port D (PD0, PD1, PD2, PD3, PD4, PD5, PD6, PD7) untuk inputan *driver* motor roda rantai. Seting Port dapat dilihat pada gambar 3.12.



Gambar 3.12. Tampilan Seting Port

List programnya akan tampil sebagai berikut :

```
#include <mega16.h>

// Input/Output Ports initialization

// Port A initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
```

```

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T

PORTA=0x00;

DDRA=0x00;

// Port B initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
    Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0
    State0=0

PORTB=0x00;

DDRB=0xFF;

// Port C initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
    Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0
    State0=0

PORTC=0x00;

DDRC=0xFF;

// Port D initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
    Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0
    State0=0

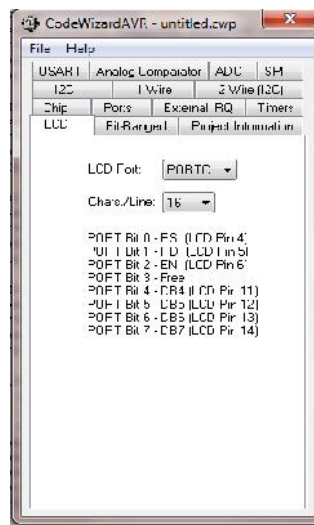
```

```
PORTD=0x00;
```

```
DDRD=0xFF;
```

3.1.9. Setting LCD CodeWizardAVR

Untuk menampilkan status robot maka digunakan LCD 2x16. Dengan menggunakan *CodeWizardAVR setting* LCD dapat dengan mudah dilakukan, seperti terlihat pada gambar 3.13. pada Robot Penampung dan Pembuang Sampah Berbasis Mikrokontroler, Port C dipilih sebagai jalur *inputan* LCD.



Gambar 3.13. Tampilan Seting LCD 2x16

List programnya akan tampil sebagai berikut :

```
#include <mega16.h>
```

```
#include <delay.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
// Alphanumeric LCD Module functions
```

```
#asm
```

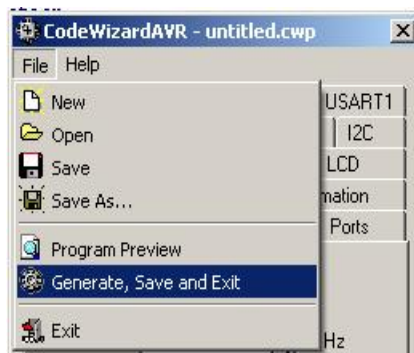
```
.equ __lcd_port=0x15 ;PORTC
```

```
#endasm
```

```
#include <lcd.h>
```

3.1.9. Menyimpan Data Program

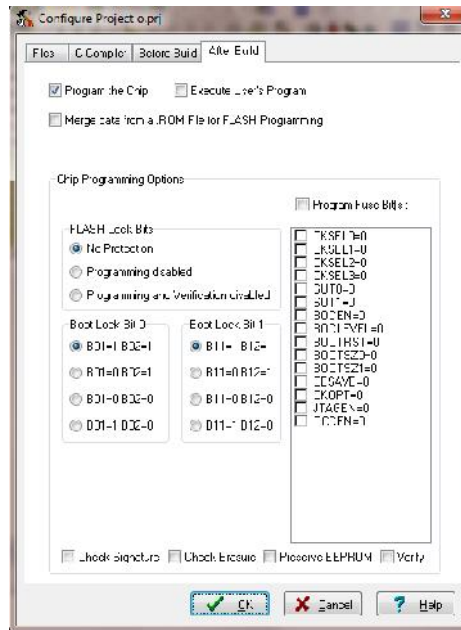
Setelah semua sudah setting sesuai kebutuhan dalam pemrograman, selanjutnya pilih menu *File*, pilih *Generate, Save and Exit* dan simpan pada direktori yang diinginkan pada gambar 3.16. List program yang tersebut diatas akan tergabung menjadi satu, kemudian ditambah beberapa program tambahan agar *software* sesuai dengan kinerja *hardware*.



Gambar 3.14. Tampilan Untuk Menyimpan *Project*

3.1.10. Mensetting Program

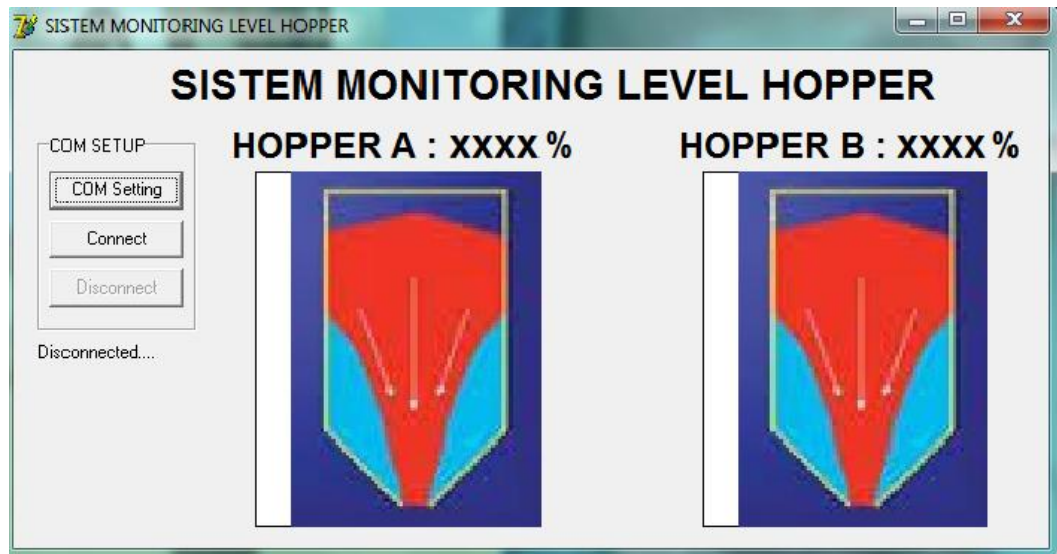
Program file pada *CodeVisionAVR* yang telah di *generate*. Selanjutnya di konfigurasi dengan memilih *Configure* di table *project*, dan mengaktifkan pemrograman IC dengan memilih *After Build* kemudian *Program the Chip*. Seperti terlihat pada gambar 3.17



Gambar 3.15. Tampilan Untuk Seting Project

3.1.12. Software untuk PC

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 3.16. Tampilan Monitoring *Borland Delphi*