

BAB III

PERANCANGAN, PEMBUATAN DAN PENGUJIAN

3.1 Perancangan

Sebelum melakukan perancangan terlebih dahulu yang harus dilakukan adalah menganalisa kebutuhan alat yang akan dibuat. Agar dalam pembuatan alat sesuai dengan kebutuhan dan dapat berfungsi dengan baik sehingga tujuan dapat tercapai.

Analisis kebutuhan merupakan batasan masalah pada tujuan yang diharapkan dari sistem yang di bangun yaitu alat ukur tekanan ban otomatis. Analisis kebutuhan dari alat yang akan di bangun adalah sebagai berikut:

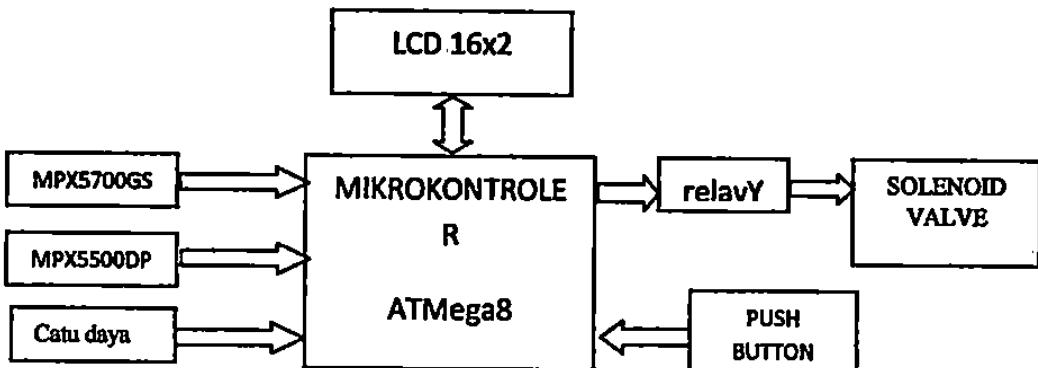
- Alat ini menggunakan sensor tekanan yang komponen utamanya yaitu MPX5700 untuk tekananan ban dan MPX5500 untuk tekanan tabung, dimana sensor ini mengubah besaran tekanan pada ban atau tabung menjadi besaran listrik (tegangan).
- Tegangan keluaran dari sensor, diproses oleh mikroprosesor atmega8, kemudian besarnya tekanan yang terukur ditampilkan pada LCD.
- Alat ini dapat bekerja secara otomatis yaitu, jika pada waktu pemopaan, tekanan yang dikehendaki (*set point*) telah tercapai maka alat akan berhenti bekerja dengan sendirinya. Begitu pula jika tekanan pada tabung

- Proses buka tutup pipa angin untuk menambah tekanan angin, digunakan *solenoid valve*.

Setelah menganalisis kebutuhan dari alat yang dibuat, kita dapat menentukan spesifikasi alat. Secara umum alat ukur tekanan ban dan tabung otomatis mempunyai spesifikasi sebagai berikut:

1. Sensor untuk membaca tekanan angin dalam ban dan tabung
2. Rangkaian mikrokontroler sebagai pengolah input dari rangkaian sensor.
3. LCD untuk menampilkan besarnya tekanan yang terukur.
4. *Solenoid valve* yang berfungsi untuk membuka dan menutup pipa angin.
5. Catu daya untuk mensuplai rangkaian menggunakan aki
6. *Software* untuk memprogram Mikrokontroler ATMega8 menggunakan AvrCodevision

Penjelasan spesifikasi diatas dapat dilihat pada Gambar 3.1 Diagram blok sistem kompresor otomatis diagram pada gambar sebagai berikut :



3.1.1 Perancangan Perangkat Keras

3.1.1.1 Perangkat Mekanik

3.1.1.1.1 Pengisi Angin

Karena sumber tegangan dari aki kendaraan, maka kita menggunakan motor bertegangan 12 volt. Motor akan bergerak setelah mendapat tegangan dari pengontrol sehingga dapat menggerakan seker/kelep dengan bantuan gir dan stang seker. Kelep akan terus menghasilkan angin yang akan di teruskan ke tabung melewati dop. Kalau angin dah melewati dop maka angin tidak bisa masuk lagi ke kelep. Jadi kelep hanya bisa mengeluarkan angin tidak bisa menyedot angin.

3.1.1.1.2 Tabung Kompresor

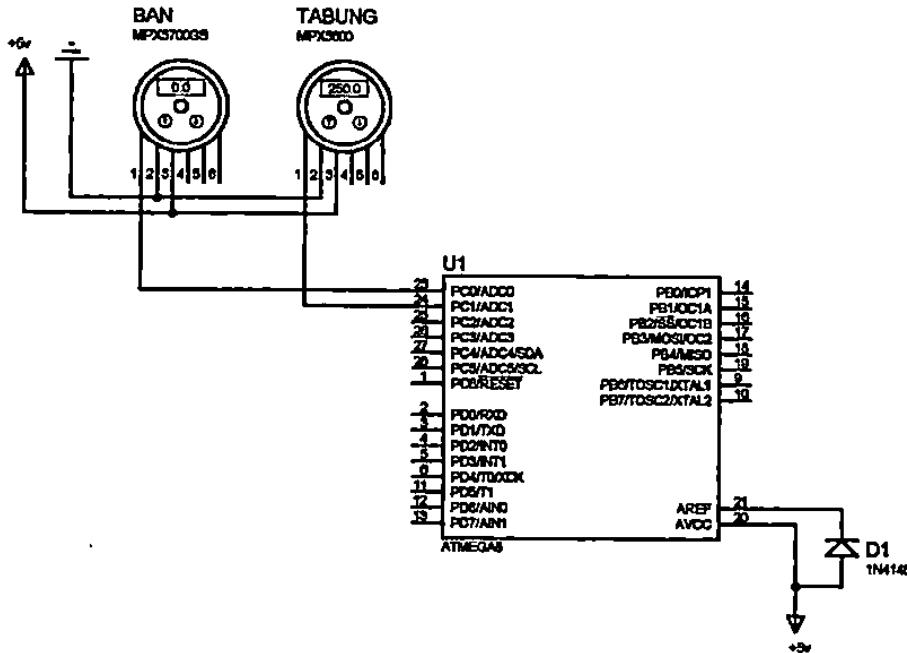
Tabung di buat dari plat besi yang tebalnya 1mm, karena tekanan angin sangat besar. Tabung memiliki kapasitas tekanan angin sampai ~ 40 psi. Untuk keselamatan pemakai maka tabung di beri *setpoint* 40 psi saja, dengan setpoint ini maka pengisian otomatis dari kelp ketabung akan

~~berhenti jika tekanan tabung telah mencapai 40 psi~~

3.1.2 Perancangan Elektrik

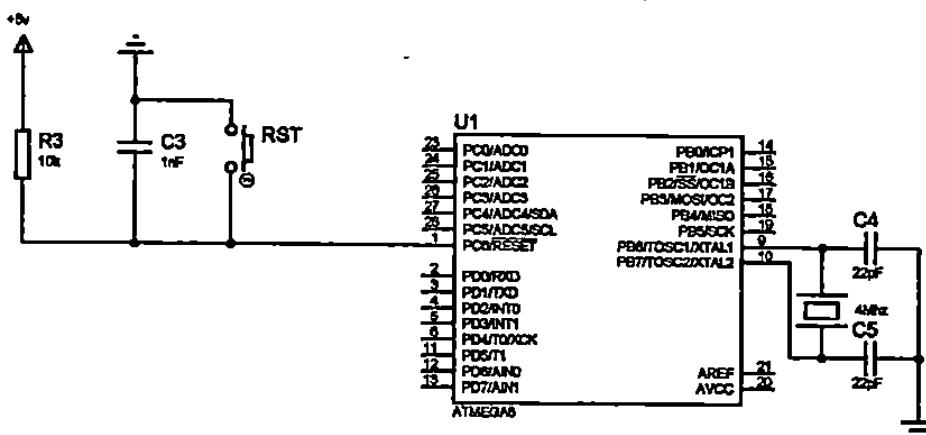
3.1.2.1 Sensor Tekanan

Untuk membaca tekanan ban dan tekanan tabung digunakan sensor tekanan MPX5700 dan MPX5500. Sensor MPX5700 mampu membaca tekanan 15 kPa – 700 kPa (2.18 psi – 101.5 psi) sedangkan MPX5500 mampu membaca 15 kPa – 500 kPa (0 psi – 75.5 psi). Keluaran dari kedua sensor ini linier yaitu 0.2 volt/kPa. Tegangan ini dikonversikan ke dalam data numerik oleh ADC internal pada ATMega8. Agar dapat memanfaatkan ADC internal tersebut pin AVCC dihubungkan dengan VCC dan AGND dengan GND *microcontroller*. Tegangan referensi ADC dapat dipilih dari tiga sumber yang tersedia, yaitu AVCC, AREF, dan internal VREF sebesar 5 volt.



3.1.2.2 Microcontroller ATMega8

Microcontroller ATMega16 memerlukan minimal catu daya 5V, *clock*, dan reset untuk dapat bekerja. Sumber *clock* diperoleh dari sebuah kristal 4Mhz yang dipasang pada kaki 9 dan 10, seperti terlihat pada Gambar 3.3. Sedangkan tombol reset yang bersifat aktif *low* digunakan untuk mereset pelaksanaan program dalam *microcontroller* sehingga dimulai dari awal (*restart*). Resistor R3 yang dipasang pada kaki reset dan terhubung pada VCC (+5V) digunakan *pull-up*, yaitu untuk mempertahankan nilai 1 (*high*) pada kaki reset selama tombol reset tidak ditekan.



Gambar 3.3 Sistem Minimum ATMega8

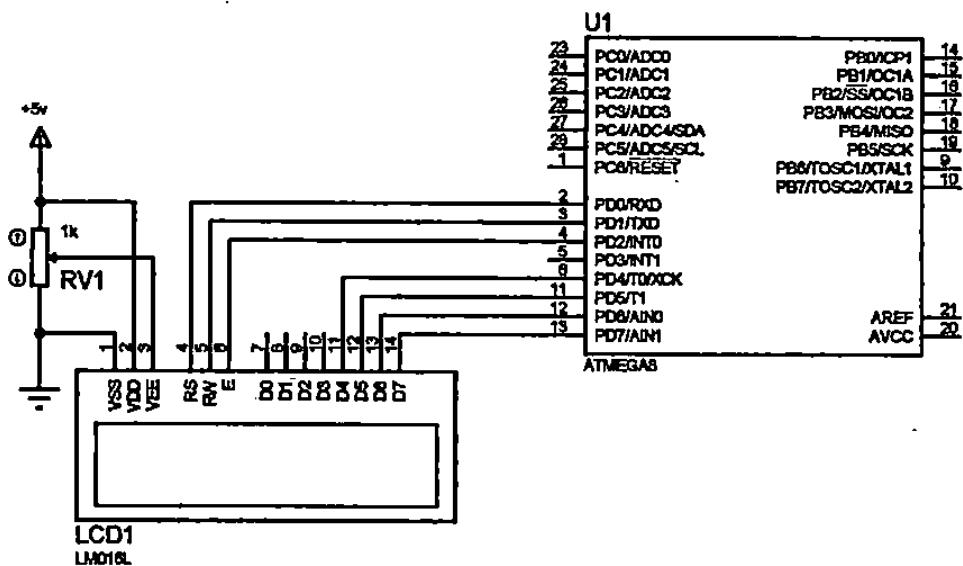
3.1.2.3 Rangkaian Penampil LCD 16X2

Rangkaian penampil menggunakan *LCD* pada penampilnya. Rangkaian *LCD* sebagai penampil dihubungkan dengan mikrokontroler.

modul LCD menggunakan sistem pengiriman data 4 bit. Sehingga hanya 4 pin dari *bus* data LCD yang digunakan yaitu D4, D5, D6, dan D7.

Pin E, RS, dan R/W digunakan untuk mengendalikan operasi LCD. Untuk semua operasi LCD, pin E (*enable*) harus dalam kondisi 1 (*high*). RS digunakan untuk menentukan jenis input, yaitu *Data Input* atau *Instruction Input*. Sedangkan R/W digunakan untuk menentukan jenis operasi yaitu *Read* atau *Write* dengan mengeset *high* atau *low*.

Pin VDD dihubungkan dengan sumber tegangan +5V dan VSS dihubungkan dengan GND. Sedangkan VEE digunakan untuk mengatur kontras LCD.

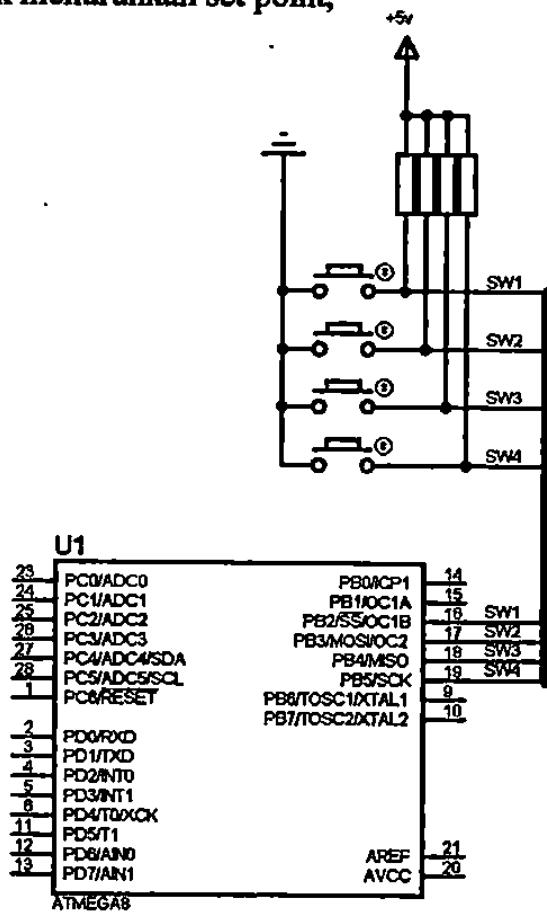


Gambar 3.4 Papan milik ICD 16x2

3.1.2.4 Rangkaian Tombol

Rangkaian ini berfungsi sebagai *interface* alat dengan dunia luar, atau lebih khususnya digunakan untuk memasukkan *set point* yang diinginkan.

Rangkaian ini terdiri dari 4 buah push button, yaitu SW1, SW2, SW3 dan SW4 yang dihubungkan pada port I/O atmega8. SW1 berfungsi Start/stop, SW2 untuk *default*, SW3 untuk menaikkan nilai set point, SW4 berfungsi untuk menurunkan set point,



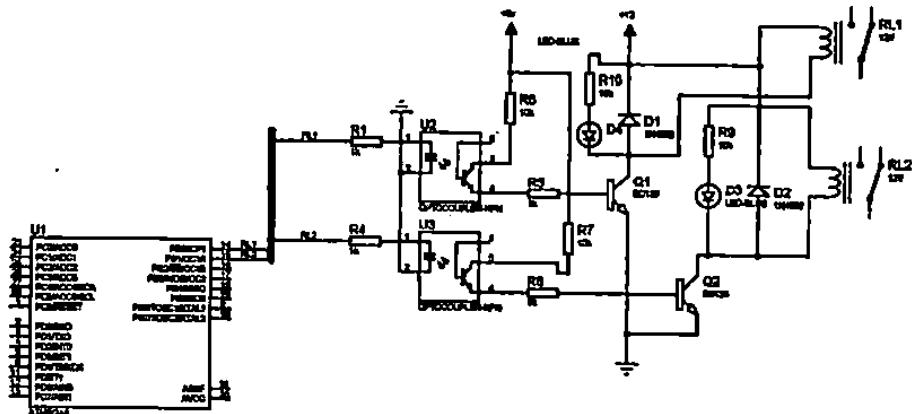
3.1.2.5 Rangkaian Relay

Rangkaian ini terdiri dari: resistor, optocoupler-npn, transistor BD139, dioda 1N4002, dan relay 12 volt. *Optocoupler* merupakan gabungan dari LED infra merah dengan *Photo transistor* yang terbungkus menjadi satu *chip*. Prinsip kerja dari *Optocoupler* adalah mengubah efek cahaya yang dipancarkan dari LED menjadi sinyal listrik oleh *Photo transistor*. Penggunaan optocoupler-npn dimaksudkan untuk menghindari *error* pada atmega8 jika terjadi gangguan pada relay, karena optocoupler ini bekerja dengan memanfaatkan cahaya. Jadi diharapkan tidak ada kontak langsung antara atmega8 dengan relay.

Transistor BD139 berfungsi untuk memberikan tegangan referensi pada relay. Resistor berfungsi untuk membatasi arus yang masuk ke rangkaian, sedangkan dioda 1N4002 sebagai indikator relay mana yang sedang bekerja.

Relay merupakan saklar elektromagnetik yang berfungsi untuk memutuskan, atau mengubah satu atau lebih kontak elektrik. *Relay* terdiri dari kumparan dan kontak NC (*normally close*). Prinsip kerjanya adalah ketika diberi tegangan referensi dari transistor BD139, maka akan terjadi medan magnet pada kumparan yang akan menarik saklar dan posisinya dari terbuka menjadi tertutup. Pada keadaan ini maka *solenoid valve* akan mendapat tegangan sebesar 12 volt sehingga katup pada solenoid valve

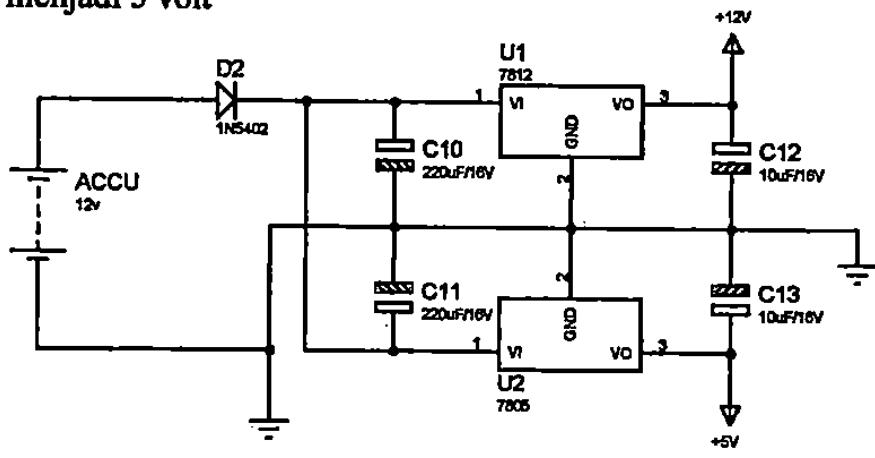
Begitu tegangan referensi dari transistor BD139 diputuskan, maka dengan segera kontak pada *relay* akan terbuka kembali. Pada keadaan ini, katub pada *solenoid valve* kembali dalam keadaan tertutup



Gambar 3.6 Relay

3.1.2.6 Rangkaian Catu Daya

Tegangan yang dibutuhkan oleh system yaitu 5 dan 12 volt, tegangan bersumber dari aki 12 volt. Untuk mendapat tegangan sebesar 5 volt dipasang IC regulator 7805 yang berfungsi untuk menurunkan tegangan stabil menjadi 5 volt

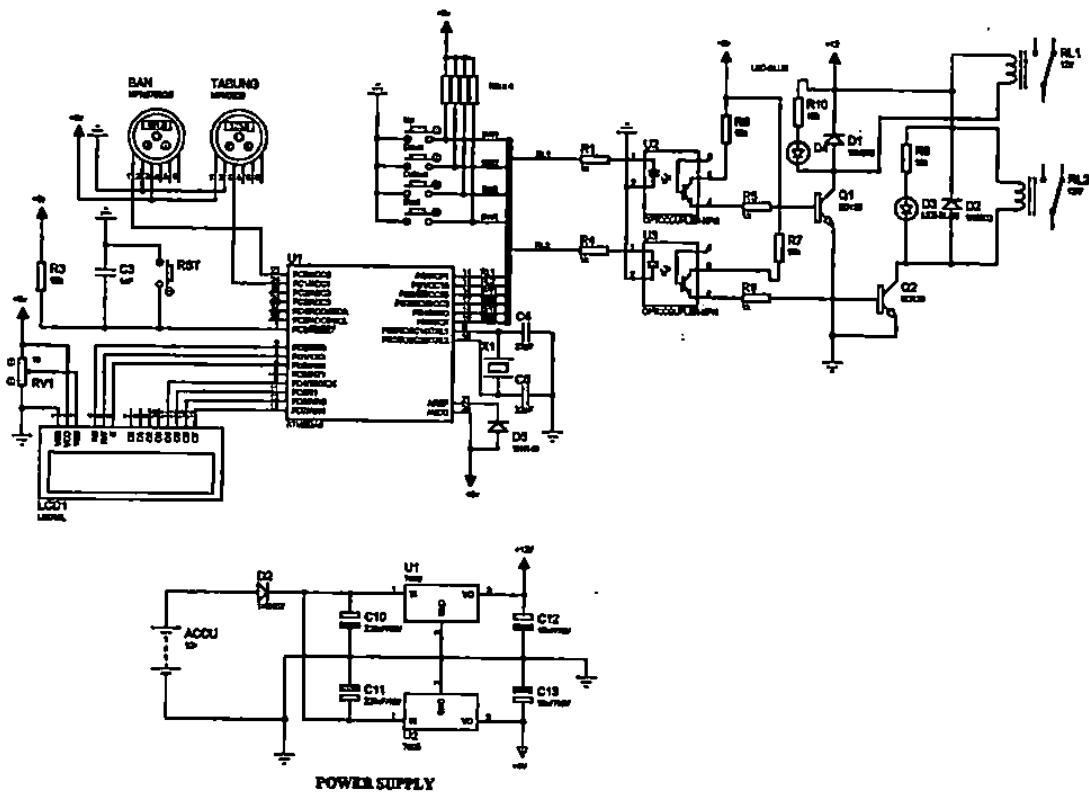


POWER SUPPLY

Gambar 3.7 Catu Daya

3.1.2.7 Rangkaian Keseluruhan

Rangkaian keseluruhan merupakan gabungan dari sistem-sistem yang telah dijelaskan diatas. Sehingga dengan penggabungan rangkaian sistem diatas terbentuklah suatu sistem kesatuan yang utuh yaitu kompressor otomatis.



Gambar 3.8 Rangkaian Keseluruhan

3.1.3 Perancangan Perangkat Lunak

3.1.3.1. Spesifikasi Perangkat Lunak

Pemrograman *microcontroller* harus memperhatikan

definisi fungsi masing-masing port / pin yang terhubung dengan komponen lain sebagai pendukung operasi *microcontroller*. Apabila program yang dibuat tidak sesuai dengan definisi fungsi port / pin maka sistem tidak akan bekerja dengan benar.

Perangkat lunak untuk sistem ini dibangun dengan bahasa C dan menggunakan *CodeVisionAVR* versi 1.25.9 Standard sebagai kompilenya. Dalam IDE (*Integrated Development Environment*) *CodeVisionAVR* telah disertakan berbagai *library* untuk mendukung kemudahan pemrograman. *CodeVisionAVR* juga dilengkapi dengan *tool* tambahan seperti *CodeWizardAVR* dan Programmer. *CodeWizardAVR* digunakan untuk membuat kerangka program dengan pendefinisian fungsi per langkah (*wizard*). Sedangkan Programmer digunakan untuk mentransfer program hasil kompilasi ke dalam chip AVR.

Pendefinisian port / pin dan fungsi-fungsi *library* yang digunakan dalam pemrograman dijelaskan sebagai berikut:

a. ADC

Inisialisasi ADC diletakkan pada fungsi main dengan statemen sebagai berikut:

```
// ADC initialization
```

```
// ADC Clock frequency 21.050 kHz
```

```
// ADC Voltage Reference: AREF pin
```

```
ADMUX=ADC_VREF_TYPE & 0xff;
```

```
ADCSRA=0x87;
```

ATMega8 mampu mengkonversi data tegangan analog ke dalam data digital dalam 10 bit. Rumus untuk ADC pada ATMega8 adalah:

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

Sehingga ketelitian input untuk mengubah bit LSB hasil konversi ADC adalah: (VREF = 5 V)

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

$$V_{IN} = \frac{ADC \cdot V_{REF}}{1024}$$

$$V_{IN} = \frac{ADC \cdot 5}{1024}$$

$$V_{IN} = 0,0048 \text{ ADC}$$

Dari persamaan di atas diketahui bahwa setiap perubahan tegangan input sebesar 4,8mV akan mengubah bit LSB pada ADC 10 bit.

Konversi ADC diatur dalam mode *free running*, yaitu *interrupt*

lalu ada ADC internal ATMega8 yang akan mengambil data

analog menjadi data digital secara terus menerus. Proses tersebut dikerjakan oleh statemen sebagai berikut:

```
#define ADC_VREF_TYPE 0x00

// ADC interrupt service routine

// with auto input scanning

interrupt [ADC_INT] void adc_isr(void)

{

register static unsigned char input_index=0;

// Read the AD conversion result

adc_data[input_index]=ADCW;

// Select next ADC input

if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))

    input_index=0;

ADMUX=(FIRST_ADC_INPUT | (ADC_VREF_TYPE &

0xff))+input_index;

// Delay needed for the stabilization of the ADC input voltage
```

LCD dipasang pada port C dan menggunakan *library* standar CodeVisionAVR sehingga didefinisikan sebagai berikut:

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

```
#asm
```

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

```
#endasm
```

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

Fungsi-fungsi *library* lcd.h yang digunakan dalam operasi *microcontroller* adalah:

- `unsigned char lcd_init(unsigned char lcd_columns);`

Fungsi ini digunakan untuk menginisialisasi LCD, membersihkan tampilan dan meletakkan posisi tulis pada baris 0 dan kolom 0. Dalam perintah inisialisasi ini jumlah kolom LCD harus ditentukan, sehingga untuk LCD 16x2 diinisialisasi dengan `lcd_init(16);`. Fungsi inisialisasi ini akan memberikan nilai 1 apabila modul LCD terdeteksi dan memberikan nilai 0 apabila modul LCD tidak berhasil dideteksi. Fungsi inisialisasi ini harus dijalankan sebelum memulai fungsi LCD lainnya.

Fungsi ini digunakan untuk membersihkan tampilan dan meletakkan posisi tulis pada baris 0 dan kolom 0.

- `void lcd_puts(char *str);`

Fungsi ini digunakan untuk menampilkan string str, yang terletak di SRAM, pada posisi *cursor* LCD.

- `void lcd_putsf(char flash *str);`

Fungsi ini digunakan untuk menampilkan string str, yang terletak di memori Flash, pada posisi *cursor* LCD.

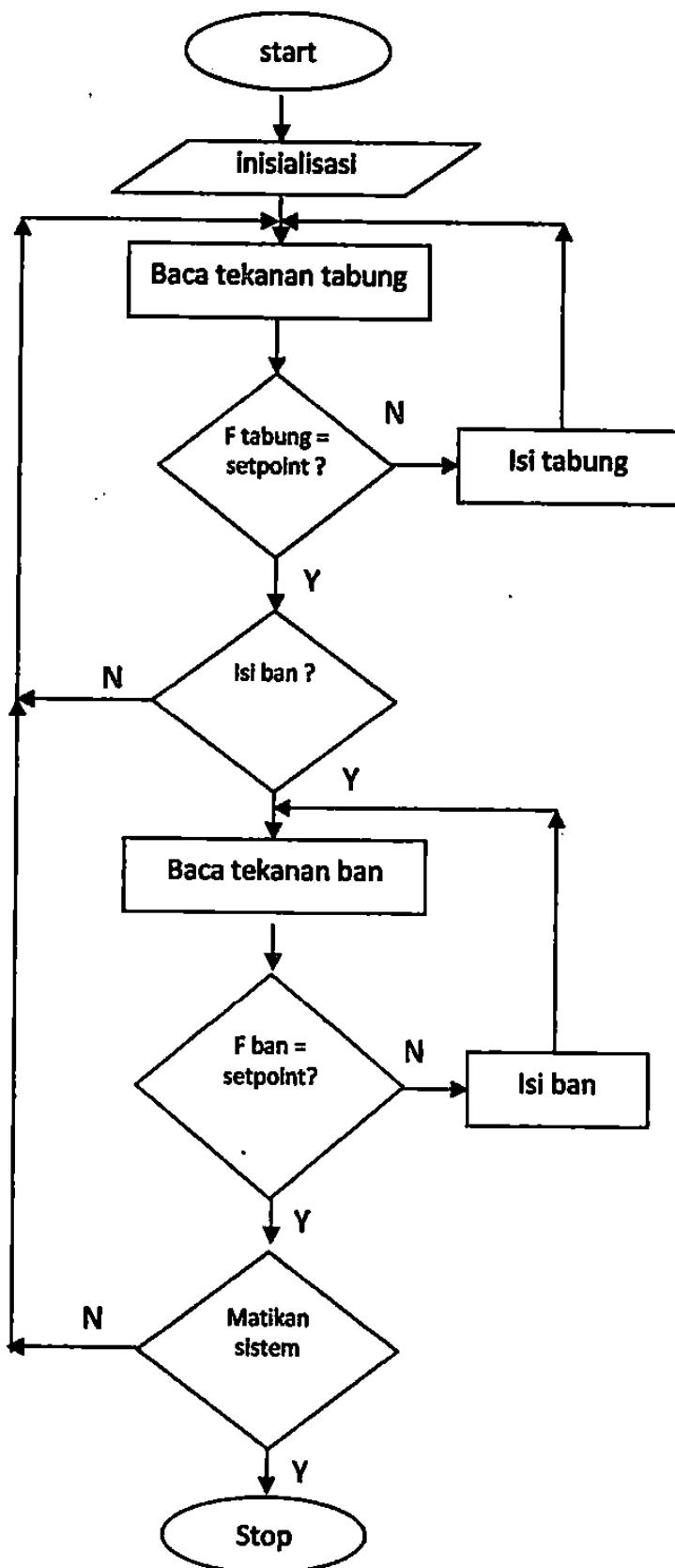
- `void lcd_gotoxy(unsigned char x, unsigned char y);`

Fungsi ini digunakan untuk meletakkan posisi tulis LCD pada kolom x dan baris y. Nomor kolom dan baris dimulai dari 0 (nol).

3.1.3.2. Operasional Perangkat Lunak

Saat pertama kali sistem dinyalakan, *microcontroller* akan menjalankan program dari awal, yaitu dari inisialisasi hingga proses pengiriman data ke LCD. Urutan kerja program pada saat pertama kali dijalankan ditunjukkan oleh arah panah dalam Gambar 3.9. Pada proses tersebut dilakukan seluruh inisialisasi dan pengecekan semua komponen library yang dipergunakan. Setelah proses inisialisasi selesai mikrokontroller akan membaca sensor

tekanan tabung, apabila tekanan tidak sesuai dengan setpoint maka mikrokontroller memerintahkan solenoid untuk membuka katub dan motor untuk menggerakkan pompa agar tabung terisi angin. Demikian juga dengan ban apabila ban yang mau dipompa tekanannya tidak sesuai dengan set point maka secara otomatis solenoid terbuka untuk mengisi ban dan apabila telah sesuai dengan setpoint, secara otomatis akan berhenti sendiri. Kemudian data ditampilkan ke penampil LCD, proses ini berlangsung terus-



3.2 Proses Pembuatan dan Pengerjaan Alat

3.2.1 Pembuatan Alat Mekanik

3.2.1.1 Pengadaan Alat dan Bahan

- Peralatan**

1. Bor besi
2. Las karbit
3. Gergaji besi
4. Grinda besi
5. Mesin bubut
6. Pemanas acrylic

- Bahan**

1. Pipa besi ukuran diameter 7 cm panjang ~ 21 cm

8. Plat kuningan

9. Dop ban Sepeda & Dop ban motor

10. Sekrup

11. Pencabangan selang, selang & knop dop

12. Acrilic (tempat rangkaian elektrik)

3.2.1.2 Proses Pengerjaan

- 1. Membuat gambar kompresor**
- 2. Membuat tabung kompresor beserta kaki, pegangan, lubang angin**
- 3. Membuat dudukan rangkaian pengisi angin kompresor & dudukan alat elektronik/pengontrol kompresor**
- 4. Membuat stang seker, seker buat kelep & bos kelep beserta tutup bos kelep**
- 5. Membuat dudukan motor & dudukan bos kelep**
- 6. Pengeboran**
- 7. Merangkai & pengelasan**

3.2.2 Pembuatan Alat Elektris

- **Peralatan**

1. Solder
2. Timah, Pelarut
3. Papan PCB
4. Bor, Gergaji besi
5. Komputer
6. Software pendukung AvrCodevision

- **Bahan**

1. Sensor Optocoupler
2. Catu Daya
3. Relay
4. Tombol push button
5. AT Mega8
6. LCD
7. Solenoid Valve
8. Komponen pendukung (Elco, resistor, dioda, Kapasitor, dll)

3.2.2.2 Proses Penggerjaan

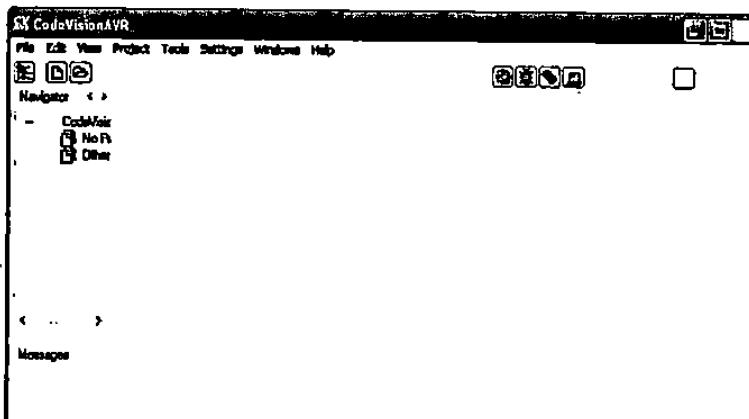
2. Membuat rangkaian PCB pada Proteus-ARES
3. Mengkonversi gambar PCB dari proteus ARES kedalam PCB

Yaitu dengan cara mencetak gambar kedalam kertas glossy kemudian gambar tersebut disetrika pada papan PCB polos, dengan demikian jalur PCB yang dibuat dapat menempel dalam papan PCB

4. Melarutan papan PCB yang telah tertempel papan dengan Ferry Clorida (F_3CL_3).
- . 5. Pengeboran papan PCB
6. Penyolderan

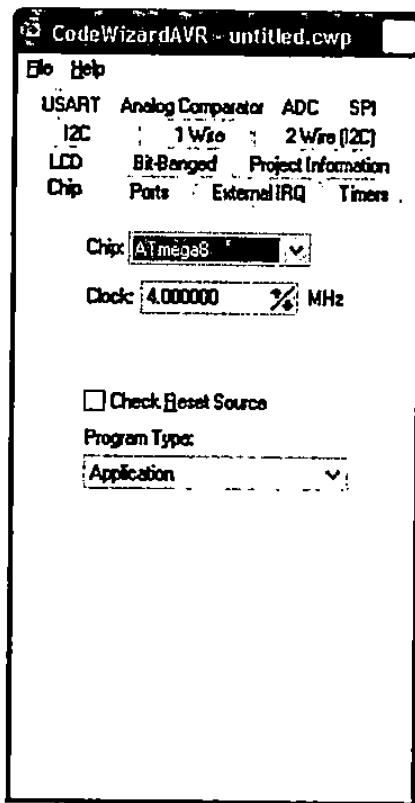
3.2.3 Proses Penggerjaan Perangkat Lunak AVRCodevision 1.25.9

Untuk memulai pemrograman C dengan AVR ,Buka program CodeVisionAVR caranya klik Start- AllPrograms- CodeVisionAVR- CodeVisionAVR C Compiler.



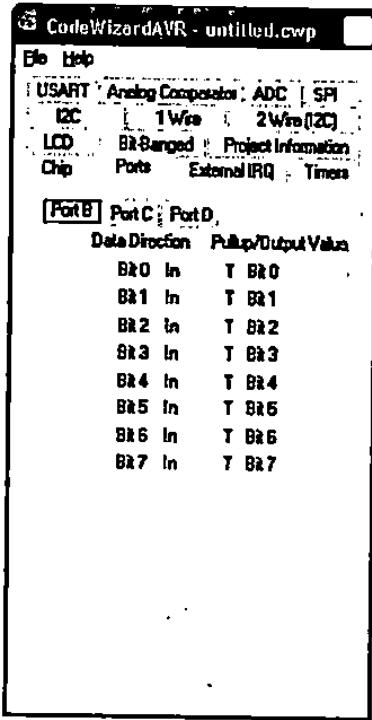
Untuk memulai membuat project baru klik File-New maka akan keluar 2 pilihan yaitu source atau project. Karena kita akan membuat project maka pilih Project. Setelah itu akan keluar kotak konfirmasi apakah kita akan menggunakan fasilitas CodeWizartAVR atau tidak. Pilih Yes.

Pilih chip yang digunakan yaitu ATMega8 dan clock yang digunakan 4MHz.



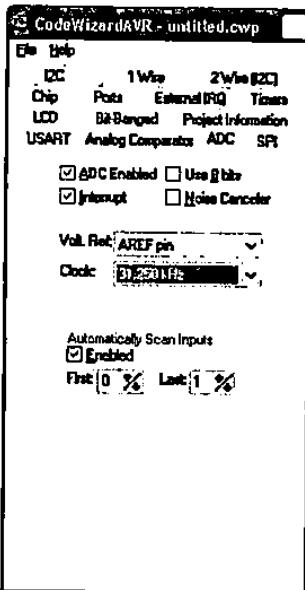
Gambar 3.11. Setting Chip yang digunakan

Pilih tab Port|Port D atur semua pin (Bit 0-bit 7) pada port D

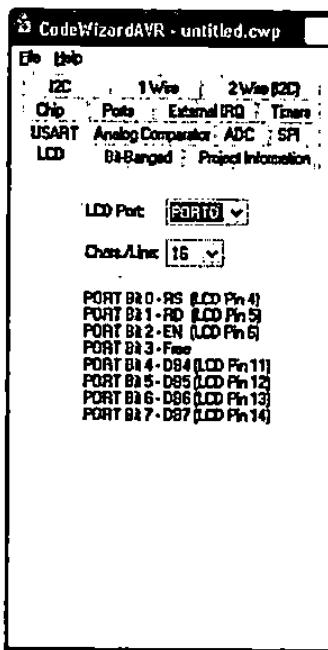


Gambar 3.12. Setting Port yang digunakan

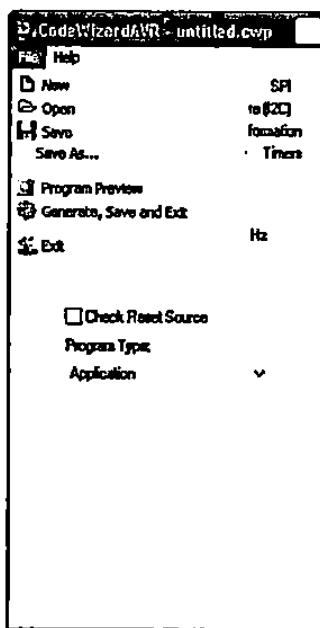
Pilih tab ADC atur untuk mengatur penggunaan ADC



Pilih tab LCD pilih Port C



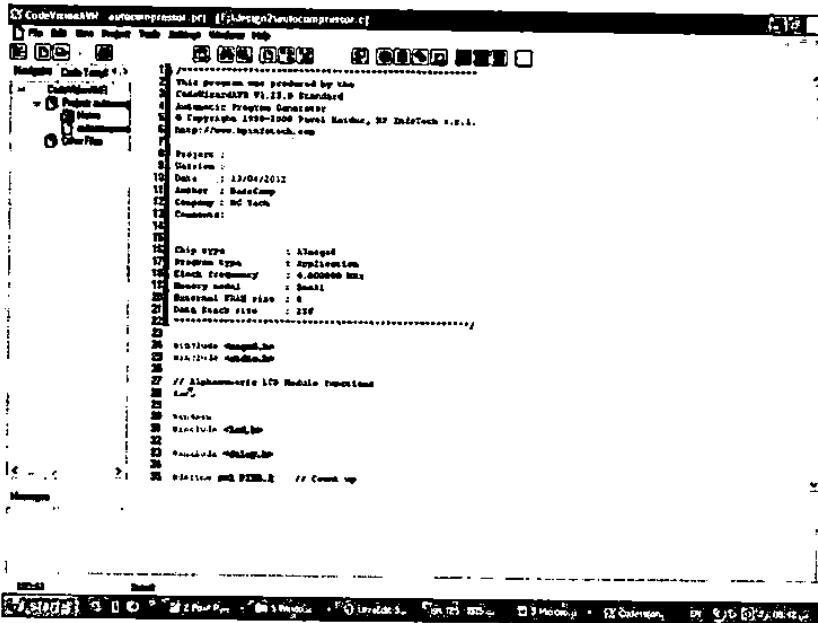
Gambar 3.14. Setting LCD



Gambar 3.15. Pengimpostan Project

Setelah semua konfigurasi/ Setting Klik File- Generate and Save and Exit. Simpan Project pada folder yang telah di tentukan

Kemudian akan dihasilkan tampilan sebagai berikut



The screenshot shows the CodeWarrior IDE interface with a project configuration file named "Project.cfg" open in the main window. The file contains various configuration parameters for the project. Key sections include:

```
/* This program was produced by the
   CodeWarrior V1.23.0 Standard
   Automatic Program Generator
   Copyright 1999-2000 Power Machines, NY DataTech v.1.1.
   http://www.datatech.com */

Project:
  Status:
    Date : 13/04/2012
    Author : Radamp
    Company : MC Tech
    Comments:

  Chip type : Allegro
  Program type : Application
  Clock frequency : 4.000000 kHz
  Memory size : Small
  External Flash size : 256
  Data Flash size : 256
  /* Aligned memory size
  RAM/ROM alignment
  // Alignment error 170 bytes detected
  */

  Test mode
  External flash
  Flashable nibbles
  Flashable pages

  0x00000000 0x0000FFFF // Current config
  0x00000000 0x0000FFFF // Default config

  0x00000000 0x0000FFFF // Current config
  0x00000000 0x0000FFFF // Default config
```

Gambar 3.16. Project yang siap dikerjakan

3.3 Pengujian

3.3.1 Pengujian Catu Daya

Tegangan yang dibutuhkan untuk pembuatan alat adalah 5 volt. Untuk tegangan 5 volt menggunakan IC *regulator* LM7805. Pengujian dilakukan dengan menggunakan multimeter, adaptor

Tabel 3.1. Hasil Validasi Catu Daya untuk *Regulator LM7805*

No	Tegangan Input (DC)	Tegangan Output	Keterangan
1	2,79 volt	0,13 volt	Gagal
2	4,58 volt	1,68 volt	Gagal
3	5,87 volt	4,83 volt	OK
4	7,26 volt	4,95 volt	OK
5	9,45 volt	4,96 volt	OK

Berdasarkan data pengujian menunjukkan bahwa hubungan antara tegangan *input* dan tegangan *output* LM7805 bersifat stabil. Tegangan keluaran yang berbeda disebabkan oleh banyak hal seperti kondisi pengukuran yang berbeda saat pengambilan data atau dapat pula disebabkan oleh kondisi IC LM7805 itu sendiri karena hasil produksi pabrikan tak ada yang sempurna tepat mencapai tegangan 5 Volt. Tetapi berdasarkan data hasil pengukuran prosentase nilai *error* masih jauh dari batas nilai

3.3.2 Pengujian Sensor

Pengujian sensor tekanan ini bertujuan untuk melihat karakteristik keluaran tegangan dari sensor tersebut. Pengujian dilakukan dengan mengukur hasil keluaran sensor pada output dengan menggunakan multimeter, dan ban kompresor sebagai penguji tekanan. Maka dari hasil pengujian tersebut diperoleh :

Tabel 3.2 Pengujian sensor MPX5700

No	Tekanan (Psi)	Keluaran (Volt)
1	1	0.045
2	5	0.223
3	10	0.444
4	15	0.666
5	20	0.887
6	25	1.12
7	30	1.33
8	35	1.56
9	40	1.77
10	45	2.1

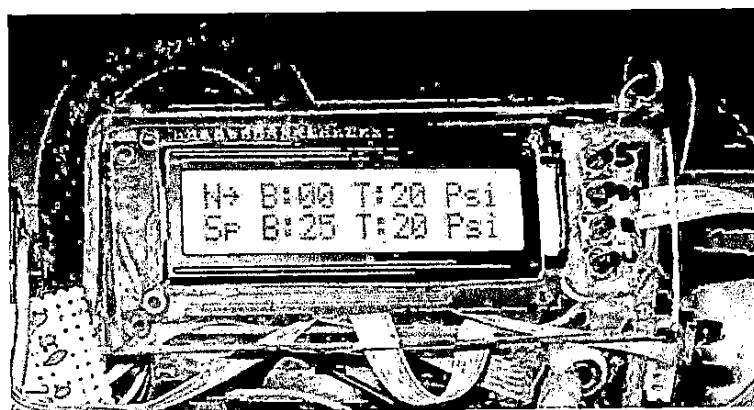
Tabel 3.3 Pengujian sensor MPX5500

No	Tekanan (Psi)	Keluaran (Volt)
1	1	0.062
2	5	0.31
3	10	0.62
4	15	0.93
5	20	1.24
6	25	1.55
7	30	1.86
8	35	2.17
9	40	2.48
10	45	2.79

Dari Pengujian tersebut ternyata bentuk keluaran dari sensor MPX5700 adalah linier yaitu 0.045 volt/Psi sedangkan

3.3.3 Pengujian Penampil

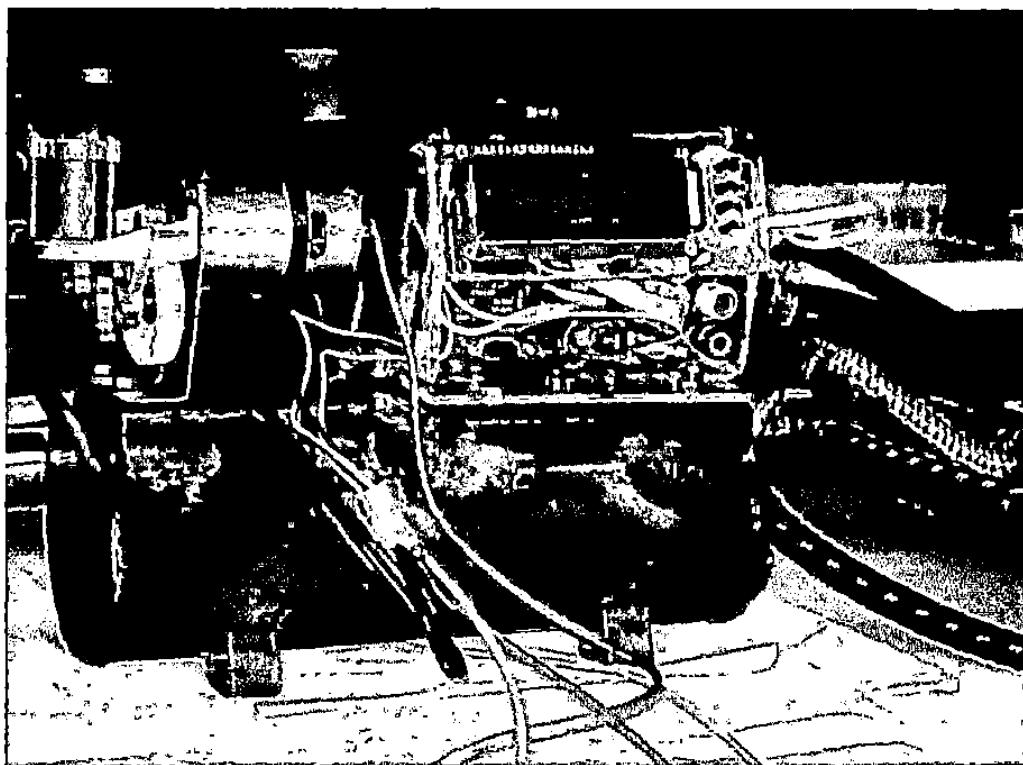
Pengujian penampil bertujuan untuk mengetahui bekerja tidaknya display pada LCD, bagian yang diuji termasuk ketajaman tampilan dan lampu backlight sehingga dapat bekerja maksimal. Cara pengujian dilakukan dengan cara memutar trimpot yang terhubung pada kaki VEE / Kontras LCD pada pin 3 untuk mengatur ketajaman karakter.



Gambar 3.17. Tampilan LCD

3.3.4 Pengujian Keseluruhan Alat

Pengujian kali ini melibatkan keseluruhan rangkaian yang telah dirancang. Pengujian rangkaian mikrontroler adalah pengujian terhadap semua fungsi yang ada dalam rangkaian tersebut yang telah terintegrasi dalam suatu rangkaian yang utuh, meliputi pengujian LCD , pengujian sensor, catudaya, penampil, dan pengujian keypad. Hasil dari pengujian ini secara keseluruhan dapat dilihat pada tabel sebagai berikut :



Gambar 3.18 Alat Keseluruhan

Tabel 3.4. Hasil Validasi Terhadap Fungsi Bagian-bagian Sistem

No	Kerja Alat	Kondisi	Deskripsi kerja	Status
1	Saklar reset	OFF	Alat bekerja normal	OK
		ON	Mereset <i>microcontroller</i> dan mengulang pelaksanaan program dari awal (<i>restart</i>)	OK
2	Sensor	Tekanan semakin besar vout semakin besar	<ul style="list-style-type: none"> • Vout linier 0.045 volt/Psi untuk MPX 5700 • Vout linier 0.062 volt/Psi untuk MPX 5500 	OK
3	LCD sebagai penampil menu dan informasi		<ul style="list-style-type: none"> • Menampilkan Tekanan • Menampilkan set point 	OK
4	Solenoid	ON OFF	Membuka Menutup	OK
5	Mekanik		Telah berkerja sesuai fungsinya	OK
4	Rangkaian relay sebagai antarmuka penggerak	Logika 0	terputus	OK
		Logika 1	terhubung	

3.4 Kalibrasi Alat

Dalam pengukuran tekanan dibutuhkan keakuratan agar nilai tekanan yang diukur memiliki nilai tekanan yang sama dengan nilai yang sesungguhnya. Perubahan – perubahan karakteristik dari komponen yang terdapat didalamnya akan berpengaruh pada ketelitian hasil pengukuran. Oleh karena itu dibutuhkan alat agar mampu menyetting nilai agar sesuai dengan hasil yang sebenarnya.

3.4.1 Peralatan dan Bahan kalibrasi

Untuk mengkalibrasi dibutuhkan alat dan bahan sebagai berikut:

- a. Alat yang telah dibuat
- b. Alat ukur tekanan digital pabrikan merk *Air Driver*
- c. Komputer

3.4.2 Langkah-langkah kalibrasi

- a. Isi tabung dengan udara sampai tekanan mencapai ~ 35 Psi pada alat
- b. Colokkan pada tabung saluran tekanan untuk ban
- c. Bandingkan tekanan alat dengan *Air driver*
- d. Perhatikan nilai tekanan pada layar LCD apabila nilai belum sama

3.5 Analisa data dan pengukuran

3.5.1 Pengujian tekanan tabung

Tabel 3.5a. Pengujian tekanan tabung

No	Alat standar (Psi)	Alat (Psi)	Error (%)
1	5	5	0.0
2	10	10	0.0
3	15	16	6.7
4	20	21	5.0
5	25	26	4.0
6	30	30	0.0
7	35	36	2.9
Rata - rata			2.6

Tabel 3.5b. Pengujian tekanan tabung

No	Alat standar (Psi)	Alat (Psi)	Error (%)
1	5	5	0.0
2	10	11	10.0
3	15	15	0.0
4	20	21	5.0
5	25	25	0.0
6	30	31	3.3
7	35	36	2.9
Rata - rata			3.0

Tabel 3.5c. Pengujian tekanan tabung

No	Alat standar (Psi)	Alat (Psi)	Error (%)
1	5	6	20.0
2	10	10	0.0
3	15	16	6.7
4	20	20	0.0
5	25	25	0.0
6	30	32	6.7
7	35	35	0.0
Rata - rata			4.8

Dari perbandingan data alat buatan dan data alat standar maka dapat diketahui prosentase kesalahan dan nilai rerata kesalahan dengan rumus:

$$\% \text{ kesalahan} = \frac{\text{nilai alat standar} - \text{nilai alat buatan}}{\text{nilai alat standar}}$$

$$\% \text{ kesalahan} = \frac{\text{nilai alat standar}}{\sum \% \text{ kesalahan}}$$

$$\% \text{ rerata kesalahan} = \frac{\sum \% \text{ kesalahan}}{\sum \text{ pengukuran}} \times 100\%$$

Dari Tabel 3.5a diambil sampel data ke 3

$$0+0+6.7+5+4+0+2.9$$

$$\% \text{ rerata kesalahan} = \frac{\text{jumlah kesalahan}}{7} \times 100\%$$

7

$$= 2.6\%$$

Dari Tabel 3.5b diambil sampel data ke 3

$$10 - 11$$

$$1. \% \text{ kesalahan} = \frac{\text{jumlah kesalahan}}{10} \times 100\%$$

10

$$= 10 \%$$

$$0+10+0+5+0+3.3+2.9$$

$$\% \text{ rerata kesalahan} = \frac{\text{jumlah kesalahan}}{7} \times 100\%$$

7

$$= 3.0\%$$

Dari Tabel 3.5c diambil sampel data ke 2

$$15 - 16$$

$$1. \% \text{ kesalahan} = \frac{\text{jumlah kesalahan}}{10} \times 100\%$$

10

$$= 6.7 \%$$

$$20+0+6.7+0+0+6.7+0$$

$$\% \text{ rerata kesalahan} = \frac{\text{jumlah kesalahan}}{7} \times 100\%$$

7

$$= 4.8\%$$

$$2.6+3.0+4.8$$

$$\% \text{ rerata kesalahan total} = \frac{\text{_____}}{3} \times 100\%$$

$$= 3.4\%$$

Dengan rata-rata kesalahan 3.4 % maka dapat disimpulkan bahwa alat telah memiliki tingkat akurasi yang cukup baik sehingga telah sesuai dengan standar

3.5.2 Pengujian tekanan ban

Tabel 3.6a. Pengujian tekanan ban

No	Alat standar (Psi)	Alat (Psi)	Error (%)
1	5	5	0.0
2	10	10	0.0
3	15	16	6.7
4	20	21	5.0
5	25	26	4.0
6	30	31	3.3
7	32	33	3.1
Rata - rata			3.2

Tabel 3.6b. Pengujian tekanan ban

No	Alat standar (Psi)	Alat (Psi)	Error (%)
1	5	5	0.0
2	10	10	0.0
3	15	15	0.0
4	20	21	5.0
5	25	26	4.0
6	30	32	6.7
7	32	33	3.1
Rata - rata			2.7

Tabel 3.6c. Pengujian tekanan ban

No	Alat standar (Psi)	Alat (Psi)	Error (%)
1	5	5	0.0
2	10	10	0.0
3	15	16	6.7
4	20	20	0.0
5	25	26	4.0
6	30	31	3.3
7	32	33	3.1
Rata - rata			2.4

Dari perbandingan data alat buatan dan data alat standar maka dapat diketahui
persentase kesalahan dan nilai rata-rata kesalahan dengan rumus:

nilai alat standar – nilai alat buatan

$$\% \text{ kesalahan} = \frac{\text{nilai alat standar} - \text{nilai alat buatan}}{\text{nilai alat standar}}$$

$$\sum \% \text{ kesalahan}$$

$$\% \text{ rerata kesalahan} = \frac{\sum \% \text{ kesalahan}}{\sum \text{ pengukuran}} \times 100\%$$

$$\sum \text{ pengukuran}$$

Dari Tabel 3.6a diambil sampel data ke 3

15 – 16

$$1. \% \text{ kesalahan} = \frac{15}{16} \times 100\%$$

15

$$= 6.7 \%$$

$$0+0+6.7+5+4+3.3+3.1$$

$$\% \text{ rerata kesalahan} = \frac{0+0+6.7+5+4+3.3+3.1}{7} \times 100\%$$

7

$$= 3.2\%$$

Dari Tabel 3.6b diambil sampel data ke 4

20 – 21

$$1. \% \text{ kesalahan} = \frac{20}{21} \times 100\%$$

20

$$= 5 \%$$

$$0+0+6.7+0+4+6.7+3.1$$

Dari Tabel 3.6c diambil sampel data ke 2

15 – 16

$$1. \% \text{ kesalahan} = \frac{\text{-----}}{10} \times 100\%$$

$$= 6.7 \%$$

$$0+0+6.7+0+4+3.3+3.1$$

$$\% \text{ rerata kesalahan} = \frac{\text{-----}}{7} \times 100\%$$

$$= 2.4\%$$

$$3.2+2.7+2.4$$

$$\% \text{ rerata kesalahan total} = \frac{\text{-----}}{3} \times 100\%$$

$$= 2.7 \%$$

Dengan rata-rata kesalahan 2.7 % maka dapat disimpulkan bahwa alat

telah memenuhi tingkat akurasi yang diinginkan. Tetapi sebaiknya tetap dilakukan pengecekan kembali.

3.5.3 Pengujian aksi solenoid dan motor pada waktu pengisian ban

Tabel 3.7a. Pengujian aksi solenoid dan motor pada waktu pengisian ban

No	Set point	Tekanan		Aksi		Keterangan
	Tabung	Ban	Tabung	Ban	Motor	Solenoid
1	35	25	35	1	Diam	Buka/tutup
2	35	25	34	4	Putar	Buka/tutup
3	35	25	33	8	Putar	Buka/tutup
4	35	25	34	14	Putar	Buka/tutup
5	35	25	33	16	Putar	Buka/tutup
6	35	25	32	18	Putar	Buka/tutup
7	35	25	33	20	Putar	Buka/tutup
8	35	25	33	22	Putar	Buka/tutup
9	35	25	32	23	Putar	Buka/tutup
10	35	25	34	24	Putar	Buka/tutup
11	35	25	32	25	Putar	Buka/tutup
12	35	25	32	25	Diam	Tutup
						Stop

Tabel 3.7b. Pengujian aksi motor dan solenoid pada pengisian ban

No	Set point		Tekanan		Aksi		Keterangan
	Tabung	Ban	Tabung	Ban	Motor	Solenoid	
1	35	30	35	1	Putar	Buka/tutup	Start
2	35	30	33	6	Putar	Buka/tutup	
3	35	30	33	9	Putar	Buka/tutup	
4	35	30	34	13	Diam	Buka/tutup	
5	35	30	33	17	Putar	Buka/tutup	
6	35	30	32	18	Putar	Buka/tutup	
7	35	30	34	20	Putar	Buka/tutup	
8	35	30	32	22	Putar	Buka/tutup	
9	35	30	33	24	Putar	Buka/tutup	
10	35	30	32	26	Putar	Buka/tutup	
11	35	30	33	27	Putar	Buka/tutup	
12	35	30	34	28	Putar	Buka/tutup	
13	35	30	32	29	Putar	Buka/tutup	
14	35	30	34	30	Putar	Buka/tutup	
15	35	30	33	30	Diam	Tutup	Stop