

**ROBOT MOBILE AVOIDER MENGGUNAKAN SENSOR  
INFRAMERAH**



**OLEH:**

**KELOMPOK IV**

|                        |               |
|------------------------|---------------|
| Abdi Riyanto           | (09110303014) |
| Afifah Fasywah         | (09110303044) |
| Awliya Ikhsan Harahap  | (09110303010) |
| Ketut Sri Rahayu       | (09110303030) |
| Pratikto Aditya Wiguna | (09110303018) |
| Mentari                | (09110303040) |
| Ricky Cantona          | (09110303012) |

Kelas

: Teknik Komputer 4B

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS SRIWIJAYA PALEMBANG  
TAHUN AJARAN 2012 - 2013**

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Sebuah komputer mampu mengendalikan sebuah rangkaian elektronika menggunakan sebuah chip IC yang dapat diisi program dan logika yang disebut teknologi Mikroprosessor. Sebagai tugas perkuliahan akhir semester IV dengan Dosen Pembimbing Ibu Dr. Ir. Siti Nurmaini M.T., dalam mata kuliah Pengantar Robotika penulis mengambil judul **ROBOT MOBILE AVOIDER MENGGUNAKAN SENSOR INFRAMERAH**. Adapun alat tersebut merupakan serangkaian komponen elektronika berbentuk *prototype* robot yang dapat bergerak dan apabila menemui halangan akan menghindari secara otomatis yang telah dikontrol menggunakan program mikrokontroler.

Tugas robot berjalan secara terus menerus tanpa membentur suatu halangan dan dapat mencari jalan keluar apabila robot menemui jalan buntu. Sehingga Robot penghindar halangan dapat dikatakan sebagai robot penjelajah. Robot penghindar halangan termasuk jenis *mobile* robot yang menyerupai fungsi makhluk hidup yang dapat berpindah. Konstruksi Robot penghindar halangan ini digerakkan oleh dua penggerak. Penggerak menyebabkan gerakan maju atau mundur, sedangkan arah pergerakan dipengaruhi oleh kombinasi arah putar kedua penggerak. Kedua penggerak motor yang bekerja secara berbeda, tidak diperlukan suatu kemudi.

Mekanisme dari robot yang dibuat penulis yaitu kemudi robot *mobile Avoider*, penulis menggunakan sensor inframerah. Sedangkan kontrol robot dan penggerak motor dibutuhkan mikrokontroler yang berfungsi menerima data dari sensor inframerah sehingga mikrokontroler dapat menggerakkan motor sesuai data yang diterima dari sensor inframerah, dengan adanya sensor inframerah dan penggerak motor, maka robot dapat berjalan tanpa

membentur halangan. Agar fungsi robot dapat menghindari halangan tercapai, maka di lakukan pengujian pada ruangan yang memiliki beberapa halangan.

## **1.2 Tujuan dan Manfaat Penelitian**

### **1.2.1 Tujuan Penelitian**

Adapun tujuan dari tugas ini adalah untuk

- Sebagai sebuah robot yang dapat bekerja secara otomatis tanpa kendali lagi oleh manusia dan dapat dikembangkan untuk robot lain yang lebih cerdas.
- Untuk menuntaskan mata kuliah Pengantar Robotika
- Membuat program yang digunakan untuk mengendalikan standar servo
- Melakukan pengujian terhadap keseluruhan sistem meliputi kemampuan sensor inframerah , sudut perputaran *standard servo* dan PWM motor DC.

### **1.2.2 Manfaat Penelitian**

Adapun manfaat yang diperoleh dari tugas ini adalah sebagai berikut:

1. Penulis dapat mengetahui *hardware* dan *software* dalam perancangan robot *Avoider* atau robot penghindar halangan.
2. Penulis dapat mengerti dan memahami bahasa C untuk hardware.

## **1.3 Batasan Masalah**

Dalam menganalisa robot penghindar halangan ini, perlu diberikan pembatasan atau ruang lingkup pembahasan untuk mempermudah dalam pemecahan serta pembahasannya. Adapun batasan – batasannya adalah sebagai berikut :

1. Robot bergerak secara otomatis untuk penghindar halangan.

2. Robot akan berhenti jika ada halangan di depan, kanan dan kiri.
3. Robot tidak dapat menghindari halangan yang ada pada bagian belakang robot.
4. Halangan yang dapat dideteksi oleh sensor berjarak 30 cm.
5. Bahasa pemrograman menggunakan bahasa C.

#### **1.4 Metode Penelitian**

Adapun metodologi penulisan yang akan digunakan dalam penelitian ini adalah sebagai berikut:

1. Metode Observasi

Merupakan metode dengan cara mencari dan mengumpulkan sumber-sumber bacaan, literatur dari media cetak maupun media elektronik yang berhubungan atau menunjang perancangan alat dan juga sebagai referensi dalam pembuatan tugas akhir ini.

2. Metode Konsultasi

Metode Konsultasi yaitu dengan melakukan diskusi serta tanya jawab dengan dosen dan laboran pengajar secara teori maupun praktek, serta orang-orang yang memiliki pengetahuan dan wawasan terhadap permasalahan yang akan dibahas didalam tugas ini.

3. Metode Perancangan

Metode Perancangan yaitu melakukan analisa terhadap rangkaian yang dibuat dimana pembuatan alat ini, berupa *layout* PCB, pemasangan komponen pada PCB agar mendapatkan hasil rangkaian yang dapat bekerja sesuai keinginan, serta melindungi komponen dari guncangan.

#### 4. Metode Pengujian

Metode pengujian yaitu tahap pengujian alat, untuk mengetahui apakah alat tersebut bekerja dengan baik dan sesuai dengan harapan.

### 1.5 Percencanaan Biaya

Beberapa Alat atau komponen yang akan digunakan dengan harga perkiraan yang sesuai di toko online, sebagai berikut:

| No           | Nama Barang                  | Banyaknya<br>(Buah) | Harga          |
|--------------|------------------------------|---------------------|----------------|
| 1.           | DC Motor Servo MT203 6 Volt  | 2                   | Rp. 80.000 /pc |
| 2.           | Sistem Minimum AVR 16 USB    | 1                   | Rp. 210.000,-  |
| 3.           | IC L293D                     | 1                   | Rp. 22.000,-   |
| 4.           | Battery Li-Po                | 1                   | Rp. 150.000,-  |
| 5.           | Sensor SHARP <i>GP2Y0A21</i> | 1                   | Rp. 210.000,-  |
| 6.           | Case Battery                 | 1                   | Rp. 5.000,-    |
| 7.           | Standard Servo Hitech HS-311 | 1                   | Rp. 145.000,-  |
| 8.           | Acrylic                      | 1 (30cm x<br>15cm)  | Rp. 20.000,-   |
| 9.           | Wheels                       | 1                   | Rp. 12.000,-   |
| <b>TOTAL</b> |                              |                     | Rp. 934.000,-  |

## BAB II

### DASAR TEORI

#### 2.1 Robot *Avoider*

Robot *Avoider* adalah robot beroda atau berkaki yang diprogram untuk dapat menghindari jika ada halangan, misalnya dinding atau benda lain yang dapat dideteksi oleh sensor sebagai halangan. Robot *Avoider* membutuhkan secara umum menggunakan 3 buah sensor sebagai pendeteksi halangan yaitu sensor depan, sensor kanan, dan sensor kiri. Dalam hal ini penulis menggunakan sensor inframerah, dan hanya menggunakan sensor sebanyak satu buah. Sensor untuk mendeteksi halangan melakukan dengan metode *scanning* dengan dibantu oleh *Standard Servo* yang digunakan untuk membuat sensor bergerak berputar kekanan dan kekiri.

Robot *Avoider* ini memiliki kekurangan dan kelebihan, jika dilihat dari kekurangannya, Robot *Avoider* ini tidak bisa mendeteksi halangan yang ada dibelakangnya terkecuali memakai sensor yang lebih banyak dan diletakkan di bagian belakang, jika dilihat dari sisi kelebihannya, Robot *Avoider* ini merupakan dasar sebuah Robot yang bisa dikembangkan untuk robot-robot lain yang lebih cerdas. Sebagai penggerak roda robot, penulis menggunakan 2 buah motor DC yang ter-*integrasi* dengan rangkaian driver motor. Rangkaian driver menggunakan IC L293D, rangkaian ini berfungsi untuk mengendalikan motor DC.

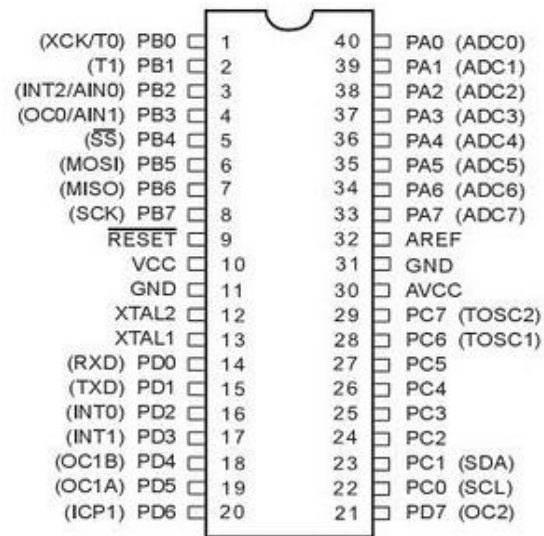
## 2.2 Sistem Mikrokontroler

Mikrokontroler, merupakan salah satu sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering disebut dengan *single chip microcomputer*. Mikrokontroler ini sudah terdapat komponen-komponen dengan bus internal yang saling berhubungan. Komponen-komponen tersebut adalah RAM, ROM, time komponen I/O paralel dan serial, dan *interrupt controller*. Mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC yang memiliki beragam fungsi. Perbedaan yang lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dan mikrokontroler. Dalam mikrokontroler ROM jauh lebih besar dibandingkan dengan RAM, sedangkan dalam komputer atau PC RAM jauh lebih besar dibanding ROM.

### 2.2.1 Mikrokontroler *ATMega16*

Mikrokontroler *ATMega 16* merupakan keluarga mikrokontroler *Atmel*. Banyak sekali fitur-fitur yang ada pada mikrokontroler *ATMega 16* sebut saja pada kecepatan *transfer data*, mikro ini sangat cepat (*high performance*) dan *low power*nya 8 bit. Serta dapat baca tulis sebanyak 100.000 kali. Jadi mikro ini dapat di *flash* sebanyak 100 ribu kali, tentu ini akan membuat mikro ini memiliki kemampuan yang lebih dan fasilitas 32 *I/O lines* serta jumlah keseluruhan pin yaitu 40 pin.

### 2.2.2 Konstruksi Mikrokontroler



**Gambar 2.1** Mikrokontroler *ATMega 16*

Dari gambar diatas di atas dapat dilihat bahwa *ATMega 16* memiliki bagian sebagai berikut:

1. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah Timer/Counter dengan kemampuan pembandingan.
4. CPU yang terdiri atas 32 buah *register*.
5. *Watchdog Timer* dengan *osilator internal*.
6. SRAM sebesar 512 byte.
7. Memori *flash* sebesar 8 kb dengan kemampuan *Read While Write*.
8. Unit interupsi *internal* dan *eksternal*.
9. *Port* antarmuka SPI.
10. EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi
11. Antarmuka komparator analog.
12. *Port* USART untuk komunikasi serial.



Fitur *ATMega 16* Kapabilitas detail dari *ATMega 16* adalah sebagai berikut:

1. *System* mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 Mhz.
2. Kapabilitas *memory flash* 8kb, SRAM sebesar 512 *byte*, dan EEPROM (*electrically Erasable Programmable Read Only Memory*) sebesar 512 *byte*.
3. ADC *internal* dengan fidelitas 10 bit sebanyak 8 *channel*.
4. Portal komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode sleep menghemat penggunaan daya listrik

### 2.2.3 Konfigurasi pin *ATMega 16*

Secara fungsional konfigurasi pin *ATMega 16* sebagai berikut:

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin *ground*.
3. *Port A* (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. *Port B* (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *timer/counter*, komparator analog, dan SPI.
5. *Port C* (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog dan *Timer Oscillator*.
6. *Port D* (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal dan komunikasi serial.
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan *clock eksternal*.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.

Untuk memprogram mikrokontroler dapat menggunakan bahasa *assembler* atau bahasa tingkat tinggi yaitu bahasa C. Bahasa yang digunakan memiliki keunggulan tersendiri, untuk bahasa *assembler* dapat diminimalisasi penggunaan memori program, sedangkan dengan bahasa C menawarkan kecepatan dalam pembuatan program. Untuk bahasa *assembler* dapat ditulis dengan menggunakan *text editor* setelah itu dapat dikompilasi dengan *tool* tertentu misalnya *asm51* untuk MCS51 dan AVR Studio untuk AVR.

## 2.2 Motor DC

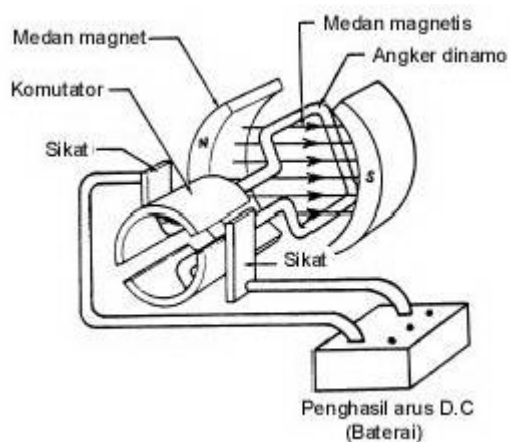
Motor DC banyak digunakan sebagai penggerak dalam berbagai peralatan, baik kecil maupun besar, lambat maupun cepat. Motor DC juga banyak dipakai karena cukup dapat dikendalikan dengan mudah pada kebanyakan kasus. Cara pengendalian motor *DC* bisa secara **ON/OFF** biasa. Pemilihan cara pengendalian akan tergantung dari kebutuhan terhadap gerakan motor *DC* itu sendiri. Pada Motor *DC* biasa, akan berputar dan berputar terus selama *power supply* ada. Tidak ada rangkaian cerdas tertentu yang diperlukan untuk mengendalikan motor tersebut, kecuali hanya memperlambat putara atau membalik putaran, dengan menerapkan polaritas balik. Elemen utama motor *DC* adalah:

- a. Magnet
- b. *Armatur* atau *rotor*
- c. *Commutator*
- d. Sikat (*Brushes*)
- e. As atau poros (*Axle*)

Motor *DC* atau dengan istilah lain dikenal sebagai dinamo merupakan motor yang paling sering digunakan untuk *mobile robot*, motor DC tidak berisik dan dapat memberikan daya yang memadai untuk tugas-tugas berat. Motor *DC* standar berputar secara bebas. Gambar 2.2 ialah bentuk dari motor DC sedangkan gambar 2.3 cara kerja motor DC.



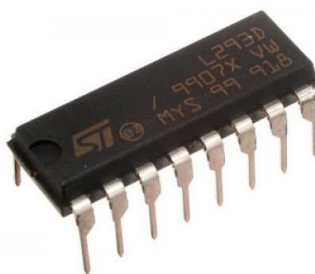
**Gambar 2.2** Motor DC



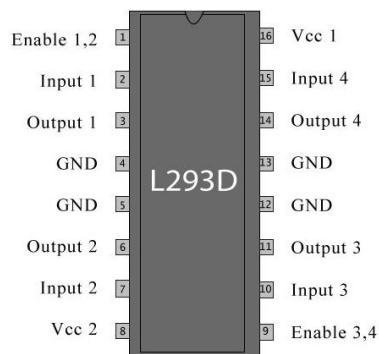
**Gambar 2.3** Cara kerja motor DC

### 2.3 IC L293D

IC L293D biasanya digunakan untuk mengendalikan motor DC. IC ini juga sering disebut *driver* motor. L293D dirancang untuk mengendalikan 2 motor DC. Lihat gambar 2.4 ialah bentuk dari IC L293D dan gambar 2.5 skema dari L293D.



**Gambar 2.4** Bentuk IC L293D



**Gambar 2.5** Skema IC L293D

Keterangan gambar:

- Pin 1, 8, 9 dan 16 menerima masukan Vcc
- Pin 2, 7, 10, dan 15 sebagai masukan dari mikrokontroler yang memberi logika 1 atau 0
- Pin 3, 6, 11, 14 sebagai keluaran ke motor DC
- Pin 4, 5, 12, 13 sebagai *ground*

## 2.4 Sensor SHARP GP2Y0A21

Sensor SHARP GP2Y0A21 digunakan untuk membaca jarak. Sensor ini menggunakan prinsip pantulan sinar infra merah. Dalam aplikasi ini nilai tegangan keluran dari sensor yang berbanding terbalik dengan hasil pembacaan jarak dikomparasi dengan tegangan referensi komparator. Rangkaian sistem komparator pembacaan jarak dengan sensor SHARP GP2Y0A21 ini disajikan pada Gambar berikut:



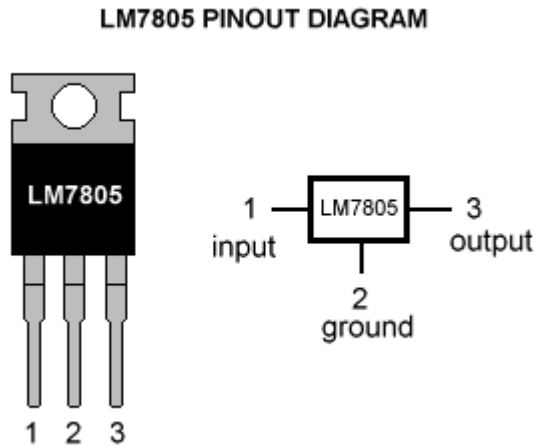
**Gambar 2.6** Sensor SHARP *GP2Y0A21*

Prinsip kerja dari rangkaian komparator sensor SHARP *GP2Y0A21* pada gambar diatas adalah jika sensor mengeluarkan tegangan melebihi tegangan referensi, maka keluaran dari komparator akan berlogika rendah. Jika tegangan referensi lebih besar dari tegangan sensor maka keluaran dari komparator akan berlogika tinggi. Selain menggunakan komparator, untuk mengakses sensor jarak SHARP *GP2Y0A21* dapat dengan menggunakan prinsip ADC, atau dengan kata lain mengolah sinyal analog dari pembacaan sensor SHARP *GP2Y0A21* ke bentuk digital dengan bantuan pemrograman. Dalam pemrograman BASCOM-AVR untuk mengakses sensor ini dapat menggunakan fasilitas akses ADC yang cukup mudah. Berikut ini link program akses ADC menggunakan BASCOM-AVR

## 2.5 LM 7805

Sirkuit terpadu seri **78xx** (kadang-kadang dikenal sebagai **LM78xx**) adalah sebuah keluarga sirkuit terpadu regulator tegangan linier monolitik bernilai tetap. Keluarga 78xx adalah pilihan utama bagi banyak sirkuit elektronika yang memerlukan catu daya teregulasi karena mudah digunakan dan harganya relatif murah. Untuk spesifikasi IC individual, xx digantikan dengan angka dua digit yang mendindikasikan tegangan keluaran yang didesain,

contohnya 7805 mempunyai keluaran 5 volt dan 7812 memberikan 12 volt. Keluarga 78xx adalah regulator tegangan positif, yaitu regulator yang didesain untuk memberikan tegangan keluaran yang relatif positif terhadap ground bersama.



**Gambar 2.7 IC 7805**

## 2.6 Standard Servo

Aktuator yang penulis gunakan untuk mendeteksi adanya halangan adalah sensor sedangkan sensor digerakkan oleh Motor Servo untuk memutar 180° untuk melakukan pemindaian daerah yang ada berada di depan sensor. Driver motor digunakan sebagai penghubung antara mikrokontroller ke motor DC. Digunakan driver motor karena arus yang keluar dari mikrokontroller tidak mampu mencukupi kebutuhan dari motor DC. Motor Servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Motor Servo tampak pada gambar berikut.



**Gambar 2.8** Motor Servo

Motor Servo merupakan sebuah motor DC yang memiliki rangkaian *control* elektronik dan *internal gear* untuk mengendalikan pergerakan dan sudut angularnya. Motor servo merupakan motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena *internal gear*nya. Lebih dalam dapat digambarkan bahwa sebuah motor servo memiliki :

- Jalur kabel : power, ground, dan control
- Sinyal control mengendalikan posisi
- Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum.
- Konstruksi didalamnya meliputi internal gear, potensiometer, dan feedback Control.

Jenis-Jenis Motor Servo :

- Motor Servo Standar 180°

Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°.

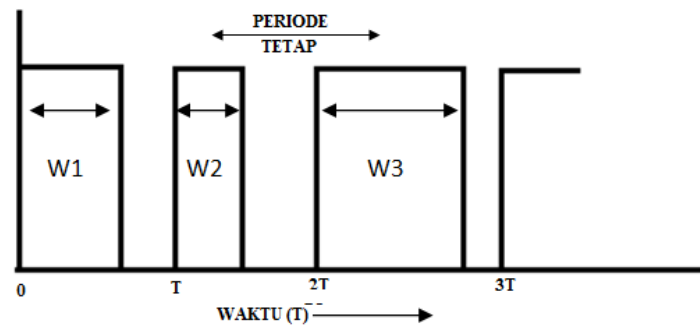
- Motor Servo *Continuous*/ 360°

Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

## 2.7 PWM ( Pulse Width Modulation )

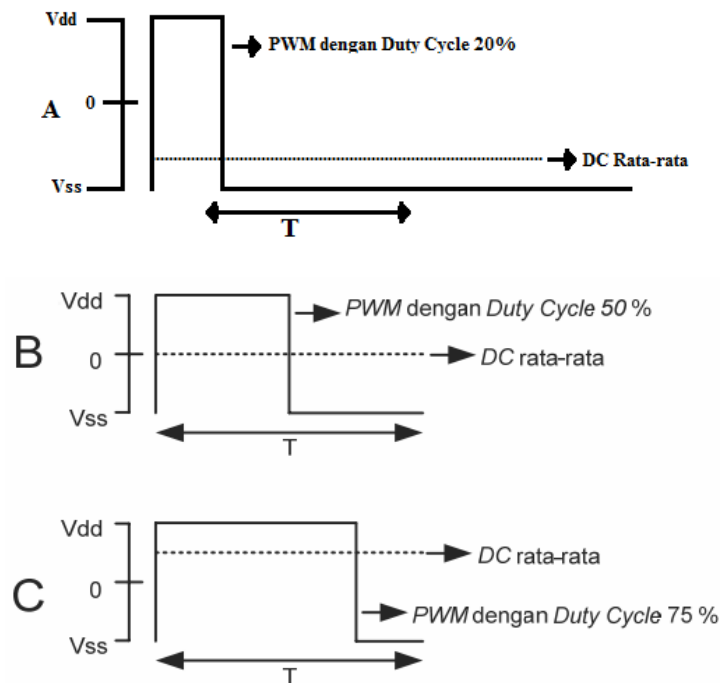
PWM merupakan sebuah mekanisme untuk membangkitkan sinyal keluaran yang periodenya berulang antara high dan low dimana kita dapat mengontrol durasi sinyal high dan low sesuai dengan yang kita inginkan. Duty cycle merupakan prosentase periode sinyal high dan periode sinyal, prosentase duty cycle akan berbanding lurus dengan tegangan rata-rata yang dihasilkan. Berikut ilustrasi sinyal PWM, misalkan kondisi high 5 V dan kondisi low 0 V. Pengaturan lebar pulsa modulasi atau PWM merupakan salah satu teknik yang “ampuh” yang digunakan dalam sistem kendali (control system) saat ini. Pengaturan lebar modulasi dipergunakan di berbagai bidang yang sangat luas, salah satu diantaranya adalah: speed control (kendali kecepatan), power control (kendali sistem tenaga), measurement and communication (pengukuran atau instrumentasi dan telekomunikasi).





**Gambar 2.9** Gelombang *PWM*

Karena hanya ada 2 kondisi amplitudo sinyal *PWM* (yaitu *Low* dan *High*) maka dapat juga dikatakan bahwa sinyal *PWM* adalah sinyal yang informasinya terletak pada lebar pulsa.



**Gambar 2.10** Tegangan rata-rata suatu *PWM*

Jika gelombang *PWM* yang dihasilkan merupakan dari sintesis gelombang sinus maka tegangan DC rata-rata akan menunjukkan sinyal sinus. Gelombang *PWM* yang difilter

dapat digunakan untuk mengendalikan perangkat analog, dan menjadikan rangkaian pemfilter sebagai suatu *digital-to-analog converter (DAC)*.

Ada beberapa alasan untuk memilih *PWM*:

1. Dalam pembangkitan *PWM* hanya butuh 1 bit mikrokontroler (bila dibandingkan dengan *DAC* konvensional butuh 8 bit, bahkan ada yang 10 bit).
2. Pada hakikatnya sinyal *PWM* merupakan sinyal yang *ON* dan *OFF*, *driver*-nya dapat dibangun dengan rangkaian *BJT* sedangkan *DAC* dibutuhkan rangkaian *driver* lebih komplek.
3. *PWM* lebih sering dipakai didalam dunia industri.

## 2.8 Catu daya

Catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah robot. Tanpa bagian ini robot tidak akan berfungsi. Begitu juga bila pemilihan catu daya tidak tepat, maka robot tidak akan bekerja dengan baik. Penentuan sistem catu daya yang akan digunakan ditentukan oleh banyak faktor, diantaranya :

- Tegangan

Setiap modul sensor atau aktuator tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap disain catu daya. Tegangan tertinggi dari salah satu modul sensor atau aktuator akan menentukan nilai tegangan catu daya.

- Arus

Arus memiliki satuan Ah (*Ampere-hour*). Semakin besar Ah, semakin lama daya tahan baterai bila digunakan pada beban yang sama.

- Teknologi Baterai

Baterai isi ulang ada yang dapat diisi hanya apabila benar-benar kosong, dan ada pula yang dapat di isi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong.

Pada perancangan robot *Avoider* ini, penulis menggunakan catu daya *Lithium Polymer* (Li-Po) dengan tegangan 7,4 volt dan arus sebesar 1000 mAh. Baterai Li-Po dibuat dari beberapa kombinasi bahan seperti *Lithium-Cobalt*, *Lithium-Phospat* dan *Lithium-Mangan*. Kebanyakan baterai yang dijual kepada konsumen terbuat dari kombinasi *Lithium-Cobalt*.



**Gambar 2.9** Baterai Li-Po

Penulis menggunakan baterai jenis ini karena memiliki keunggulan dibanding dengan baterai tipe lain seperti NiCad atau NiMH yaitu, diantaranya :

- Baterai Li-Po memiliki bobot yang ringan dan tersedia dalam berbagai macam bentuk dan ukuran
- Baterai Li-Po memiliki kapasitas penyimpanan energi listrik yang besar

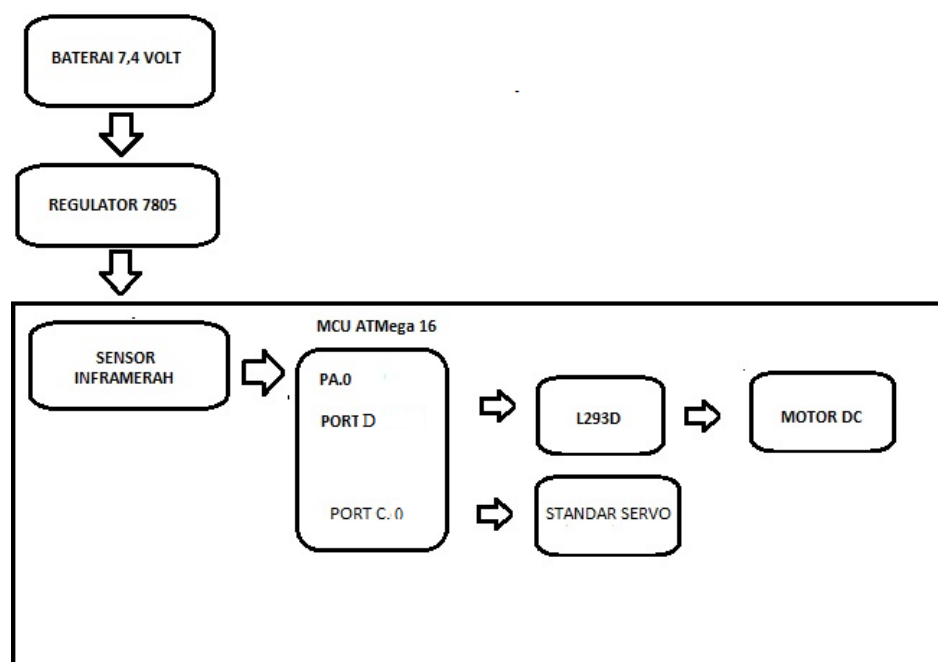
- Baterai Li-Po memiliki tingkat discharge rate energi yang tinggi, yang artinya pemakaian baterai ini bisa digunakan sampai baterai benar-benar mencapai titik krusialnya.
- Baterai Li-Po tidak memiliki “*Memory Effect*” seperti pada baterai jenis Ni-Cd (*Nickel-Cadmium*) dan Ni-MH (*Nickel-Metal Hydride*). Jadi kita tidak harus menunggu hingga baterai Li-Po benar-benar habis jika ingin melakukan isi ulang (*recharge*)
- Baterai Lipo bisa menghasilkan energi lebih besar dari pada Li-ion standar (dengan ukuran fisik yang sama), atau dengan kata lain, dengan kapasitas yang sama, baterai Li-po bisa dibuat lebih kecil ukuran fisiknya.

### BAB III

#### PERANCANGAN ALAT

##### 3.1 Perancangan Blok Diagram

Perancangan alat ini meliputi rangkaian sensor inframerah, Mikrokontroler *ATMega 16*, rangkaian Motor DC dengan L293D, Gambar 3.1 menunjukkan diagram blok system yang dirancang.



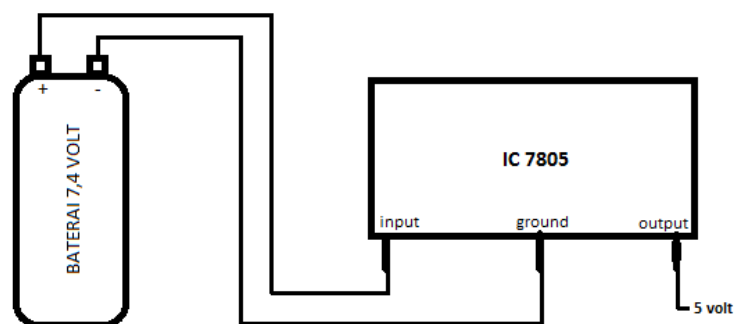
**Gambar 3.1** Diagram Blok Rangkaian

1. Sensor inframerah, sebagai pemberi masukan informasi sinyal yang dapat dihubungkan ke mikrokontroler.
2. Mikrokontroler Unit (MCU), merupakan komponen pengolah data masukan dari sensor, dan juga untuk mengirimkan data ke *driver* motor DC.
3. L293D digunakan sebagai ic driver motor dc.
4. Standar servo untuk menggerakkan sensor pada saat memindai halangan yang ada.

### 3.2 Rancangan Catu Daya Menggunakan IC 7805

Catu daya adalah sebuah rangkaian yang memberikan tegangan DC yang dibutuhkan. Rangkaian catu daya ini menggunakan sebuah baterai *charge* 9,6 volt dan IC 7805 untuk menjaga kesetabilan tegangan.

Untuk rangkaian sensor, mikrokontroler, dan IC L293D membutuhkan tegangan yang kecil, karena baterai yang digunakan 7,4 volt maka dibutuhkan sebuah IC 7805 untuk merubah tegangan baterai 7,4 volt menjadi tegangan 5 volt. Pada kaki masukan IC 7805 terhubung dari baterai 7,4 volt dan kaki keluaran memberikan keluaran sebesar 5 volt. Gambar 3.2 merupakan rangkaian baterai 7,4 volt dan IC 7805:

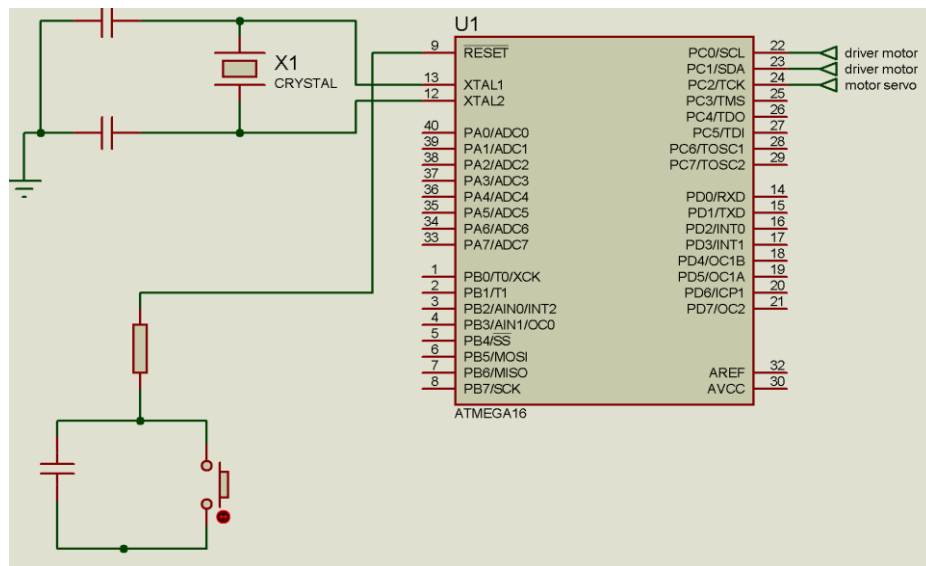


**Gambar 3.2** rangkaian catu daya menggunakan IC 7805

### 3.3 Mikrokontroler ATmega 16

Sebagai “otak” robot, digunakan mikrokontroler AVR jenis ATmega 16 yang akan membaca data dari sensor inframerah lalu mengolahnya sesuai dengan program yang telah di *download*. Pada robot *Avoider* ini, inputan sesor inframerah dihubungkan ke Port A.0 dan Por A.1 dengan output ke *driver* motor yang dihubungkan ke port C.0 s/d Port C.3.

Untuk output ke motor DC control dihubungkan pada port D.0 dan D.1 dapat dijelaskan seperti terlihat pada gambar berikut:

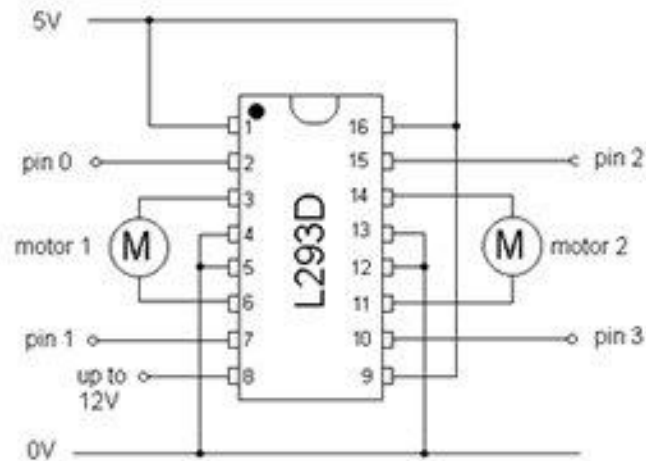


**Gambar 3.3** Mikrokontroler ATMega 16

### 3.4 Rangkaian IC L293D Sebagai Driver Motor DC

L293D adalah driver motor untuk robot *track linefollower*, motor DC 6 volt sebelah kiri dihubungkan pada kaki 3 (1A) dan 6 (2A) pada IC L293D, begitu juga pada motor sebelah kanan dihubungkan pada kaki 10 (3A) dan 16 (4A), pin 1 pada kaki L293D dihubungkan ke 5 volt DC sedangkan kaki 8 pada IC L293D diberi tegangan 5 volt, ini bertujuan agar putaran motor semakin cepat.

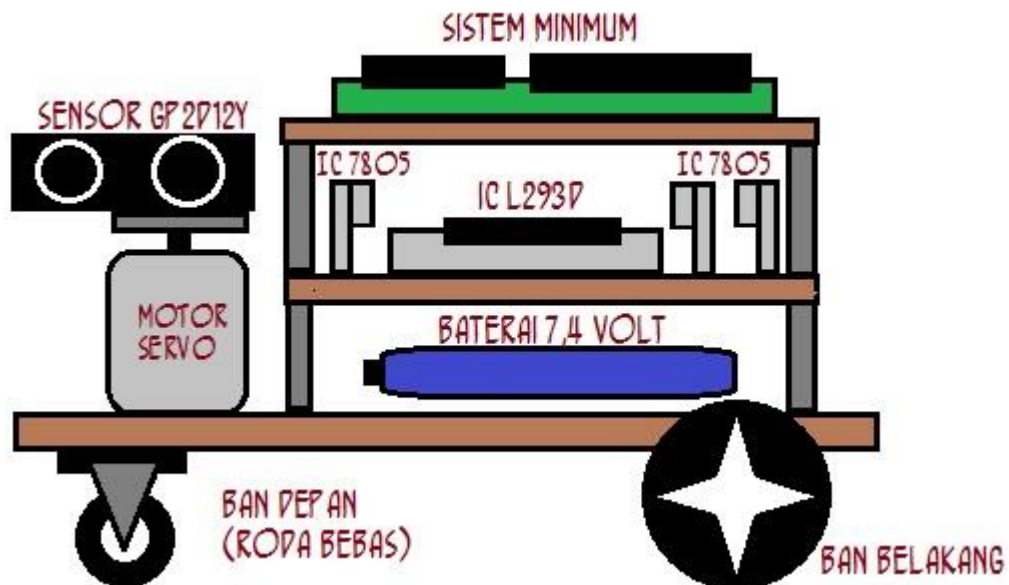
Jika RA3 diberikan logika 1 dan RA2 diberi logika 0, maka motor A akan berputar kebalikan arah jarum jam, dan sebaliknya jika RA3 diberikan logika 0 dan RA2 diberikan logika 1, maka motor A akan berputar searah jarum jam. Jika memberikan logika 1 pada RA3 dan RA2 motor A akan berhenti, begitu juga motor B. jika ingin mengganti kecepatan motor salah satu caranya adalah mengganti tegangan VCC2 atau VS. Rangkaian driver motor dapat dilihat pada gambar dibawah ini.



**Gambar 3.4** Rangkaian driver L293D motor DC

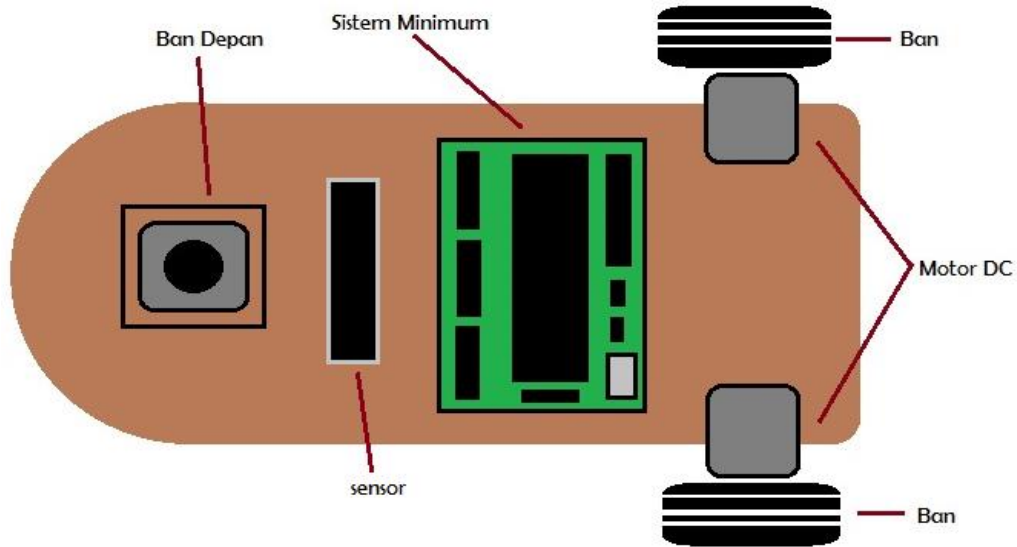
### 3.5 Rancangan Mekanik Robot

Rancangan mekanik robot tampak samping dan bawah dapat dilihat dari gambar dibawah ini, gambar dibawah ini menjelaskan rancangan dan tata letak komponen yang akan dipasangkan ke robot.



**Gambar 3.6** Rancangan Mekanik Robot tampak dari samping





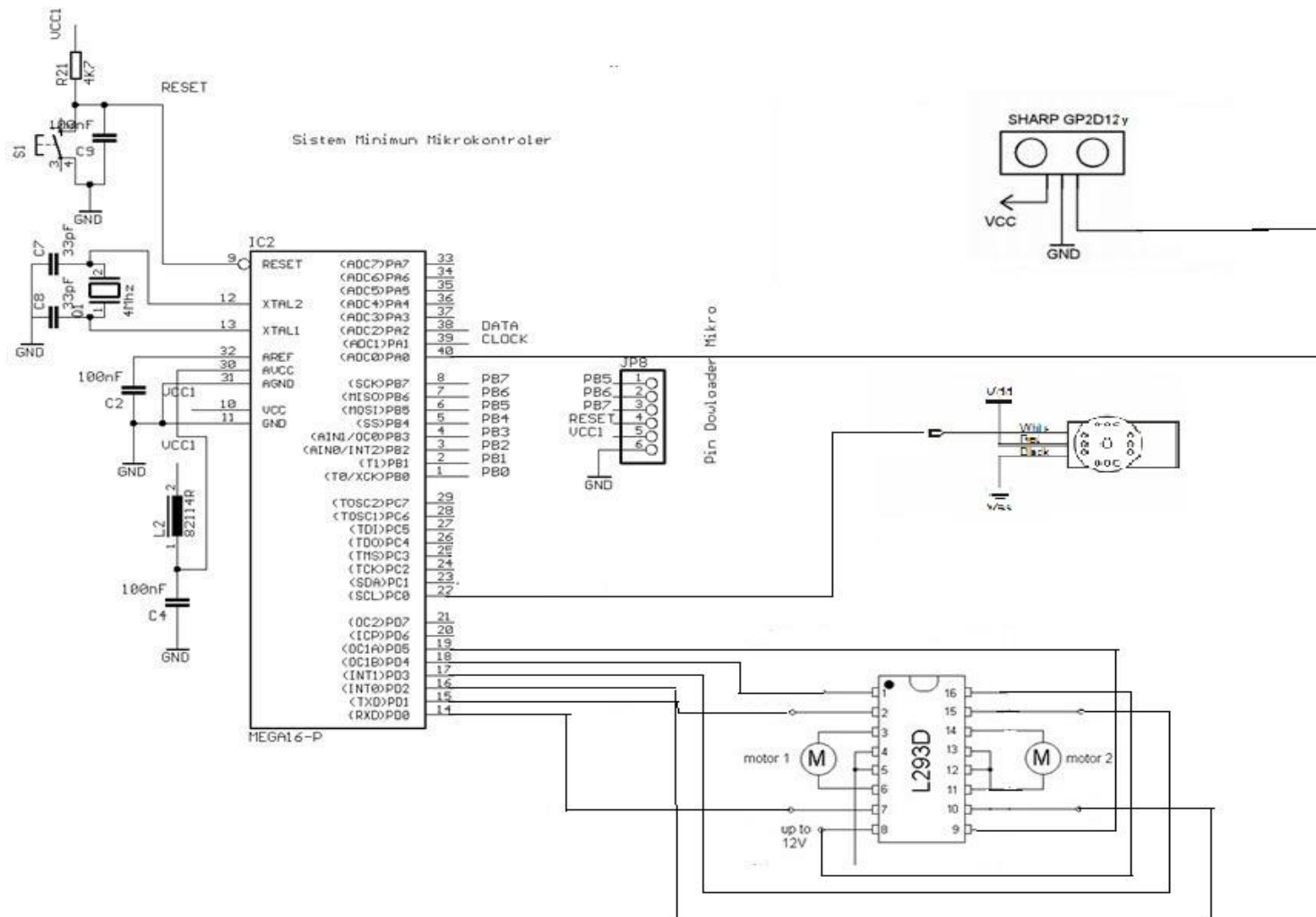
**Gambar 3.7** Rancangan Mekanik Robot tampak dari atas

### 3.6 Rangkaian Keseluruhan

Robot *Avoider* ini menggunakan sensor inframerah sebagai masukan, dimana jarak objek dengan sensor dapat dihitung dengan mengukur besarnya output pada sensor, output dari sensor berupa tegangan, dimana semakin dekat jarak objek dengan sensor maka semakin tinggi pula tegangan yang dikeluarkan oleh output sensor. Robot akan berjalan lurus jika berlogika 1 dan berhenti jika berlogika 0, dimana dalam bahasa digital berarti jika logika 1 maka 5 volt dan logika 0 jika 0, volt. Output dari sensor dihubungkan ke ADC0. Sedangkan output dari ATmega ini pada port D.0, port D.1, port D.2, port D.3, port D.4 digunakan untuk driver motor dimana port D.0 dan port D.1 digunakan untuk motor driver L293D sebelah kiri sedangkan port D.2 dan port D.3 digunakan untuk motor driver L293D sebelah kanan, dan port D.4 digunakan untuk motor servo. Untuk logika roda bergerak jalan lurus, belok kanan, belok kiri, dan berhenti bias dilihat pada tabel berikut:

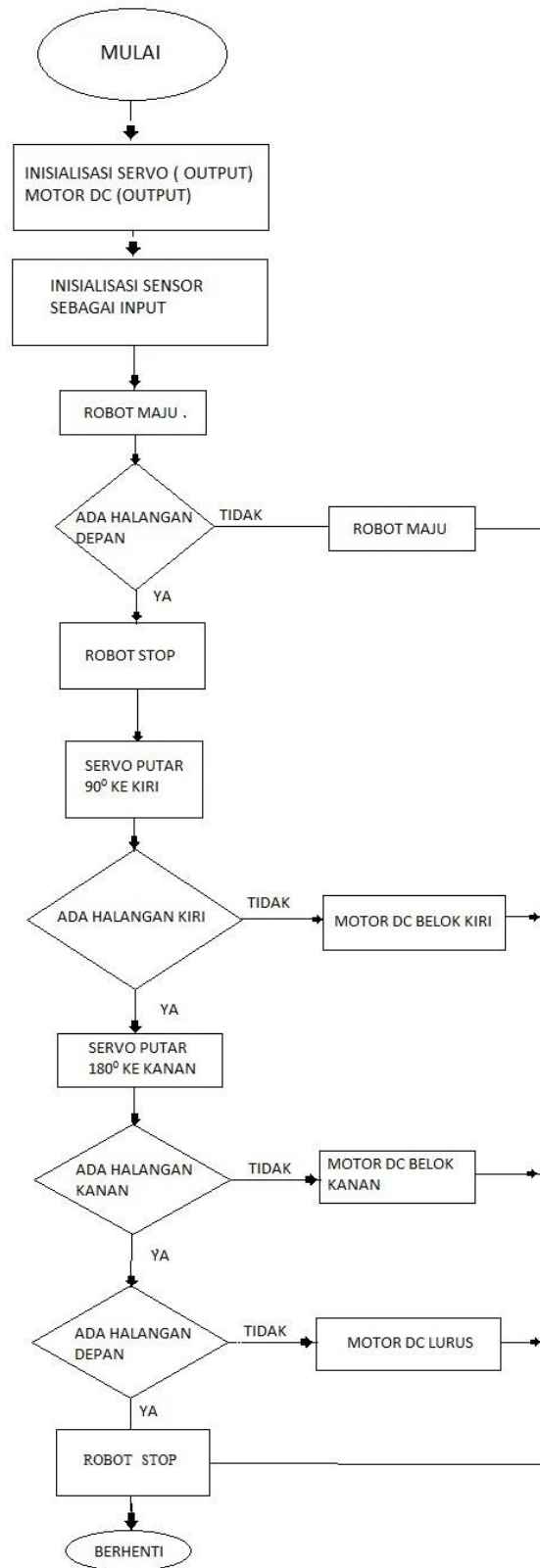
| Kecepatan Motor |     |            |
|-----------------|-----|------------|
| Motor Kanan     |     | Motor Kiri |
| Maju Lurus      | 255 | 255        |
| Belok Kanan     | 250 | 255        |
| Belok Kiri      | 255 | 250        |

Untuk menggerakkan robot ini diperlukan bahasa pemrograman, bahasa yang digunakan yaitu bahasa C, dan *codevisionAVR* sebagai *compiler* nya, untuk *downloader* nya digunakan port USB yang terhubung ke PC dan system minimum.



**Gambar 3.8** Rangkaian keseluruhan

### 3.7 Flowchat *Mobile Robot Avider*



**Gambar 3.9** *Flowchart Robot*

### 3.7.2 Cara kerja robot

- Sensor melakukan pemindaian halangan, dimana sensor digerakkan oleh standar servo.
- Sensor mencari jarak terjauh dari halangan
- Sensor akan memberikan output ke minsys untuk menggerakkan motor DC untuk bergerak ke arah kanan, kiri, maju, atau berhenti dan melukan pemindaian lagi.
- Apabila sensor mendeteksi halangan yang berada di depan, kanan, kiri maka robot berhenti.

### 3.7.3 Algoritma

#### a. Algoritma Sensor

##### Deklarasi

float jarak

##### Deskripsi

```
while(1)
    transmit pulse()
    jarak= receive pulse()
    return(jarak)
    delay_ms(32)
endwhile
```

#### b. Algoritma Motor DC

##### Deklarasi

int gerak

##### Deskripsi

```
read (gerak)
switch (gerak)
case 1:          //Belok Kanan
    mrkir.cepat
    mrkan.lambat
```

```

case 2:           //Belok Kiri
    mrkir.lambat
    mrkan.cepat
case 3:           //Maju
    mrkir.cepat
    mkan.cepat

case4:             //Stop
    mrkir.mati
    mkan.mati

endcase

```

### **c. Motor Servo**

#### **Deklarasi**

```
int servo
```

#### **Deskripsi**

```
read (servo)
```

```
switch (servo)
```

```
case 1:
```

```
    sudut servo 00
```

```
case 2:
```

```
    sudut servo 900
```

```
case 3:
```

```
    sudut servo 1800
```

```
endcase
```

**d. Algoritma Utama**

**Deklarasi**

float jarak

int gerak

int servo

label ulang

**Deskripsi**

while(1)

    ulang:

    servo(2)

read (jarak)

if (jarak < 35)

        gerak(4)

        servo(3)

read (jarak)

if (jarak < 35)

        servo(1)

read (jarak)

if (jarak > 35)

        gerak(1)

    else

        gerak(2)

    else

        gerak(3)

goto ulang

endif

endwhile

**BAB IV**  
**PENGUJIAN ALAT**

**4.1 Ujian Sensor Sharp GP2Y0A21**

| Jarak ( cm ) | Tegangan |
|--------------|----------|
| 10           | 2.4 v    |
| 15           | 1.8 v    |
| 20           | 1.4 v    |
| 25           | 1.1 v    |
| 30           | 1 v      |
| 35           | 0.86 v   |

**4.2 Pengujian Motor Servo 180**

| Sudut            | Delay (mikro sekon) |
|------------------|---------------------|
| 0 <sup>0</sup>   | 1500                |
| 90 <sup>0</sup>  | 3500                |
| 180 <sup>0</sup> | 5220                |



#### 4.3 Pengujian Motor DC Dengan Menggunakan PWM ( Pulse Width Modulation )

| Kecepatan Motor |     |            |
|-----------------|-----|------------|
| Motor Kanan     |     | Motor Kiri |
| Maju Lurus      | 255 | 255        |
| Belok Kanan     | 250 | 255        |
| Belok Kiri      | 255 | 250        |

#### 4.4 Pengujian Rangkaian Keseluruhan

Robot avoider menggunakan mikrokontroler ATMEGA 16 sebagai pengendali robot dan sensor sharp yang diletakkan diatas motor servo. Sensor di hubungkan dengan PORTA ( adc 0 ) sedangkan servo di hubungkan dengan PORTC.0. Motor DC yang digunakan untuk menggerakkan roda dihubungkan pada PORTD. Motor DC menggunakan PWM untuk mengatur kecepatan motor. Pada saat di hidupkan, robot avoider akan berjalan maju dimana kecepatan motor kanan sama dengan kecepatan motor kiri. Robot akan menscan halangan di depannya, jika tidak ada maka robot akan terus maju kedepan,jika robot menemukan halangan maka motor servo bergerak 90 derajat kearah kiri dan menscan halangan di sebelah kiri. Jika ada halangan di sebelah kiri maka servo bergerak 180 derajat ke arah kanan dan belok ke arah kanan. Apabila tidak ada halangan, servo akan kembali ke sudut 90 derajat dan robot akan bergerak maju kedepan.

## 4.5 Program

```
#include <mega16.h>

#include <delay.h>

#define s PORTC.0

#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result

unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);

    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);

    // Start the AD conversion

    ADCSRA|=0x40;

    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);

    ADCSRA|=0x10;

    return ADCH;
}

// Declare your global variables here

int i;

unsigned char jarak(unsigned char x)
{
    if (read_adc(x)<21) return 80;

    else return (1560/read_adc(x)-2);
}
```

```

void p0()
{
    for (i=0;i<47;i++)
    {
        s=1;
        delay_us(1300);
        s=0;
        delay_ms(15);
    }
}

```

```

void p90()
{
    for (i=0;i<47;i++)
    {
        s=1;
        delay_us(3250);
        s=0;
        delay_ms(15);
    }
}

```

```

void p180()
{
    for (i=0;i<47;i++)
        PORTB = 0x05;
        // PORTB xxxx 01 01 = maju; =

    {
        s=1;
        delay_us(5200);
        s=0;
        delay_ms(15);
    }
}

```

```

void go()

{  PORTD = 0x05;

    OCR1A = 250;

    OCR1B = 255;

}

void stop()

{  PORTD = 0x00;

}

/*void maju()

{  OCR1A = OCR1B = 254;


    // delay_ms();

}*/

void kanan()          //  kiri kanan

{  PORTD = 0x04;      //xxxx 10 00

    //  OCR1A = 250  ;

    //  OCR1B = 200  ;

    delay_ms(1050);

}

void kiri()

{  PORTD = 0x01;      // xx 00 01

    //  OCR1A = 200  ;

    //  OCR1B = 250  ;

    delay_ms(1050);

}

void main(void)

{

```

```

// Declare your local variables here

    unsigned char a;

// 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=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

PORTB=0x05;

DDRB=0xff;

// Port C initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=Out
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=0

PORTC=0x00;

DDRC=0xf1;

// Port D 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

PORTD=0x00;

DDRD=0xFF;


// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

```

```

// Mode: Normal top=FFh

// OC0 output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: 16000.000 kHz

// Mode: Fast PWM top=0x00FF

// OC1A output: Non-Inv.

// OC1B output: Non-Inv.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0xA1;

TCCR1B=0x09;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

```

```

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=FFh

// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=0x00;

MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x00;

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;

// ADC initialization

// ADC Clock frequency: 1000.000 kHz

// ADC Voltage Reference: AVCC pin

// ADC Auto Trigger Source: None

// Only the 8 most significant bits of

```

```

// the AD conversion result are used

ADMUX=ADC_VREF_TYPE & 0xff;

ADCSRA=0x83;

OCR1A =255;

OCR1B = 250;

while (1)

{

// Place your code here

p90();

a = jarak(0);

if(a<35) // awal

{ stop();

p180();

//delay_ms(10);

a = jarak(0);

if(a<35)

{ p0();

a= jarak(0);

if(a>35)

{

kanan();

// delay_ms(100);

// p90();

}

}

else

{ kiri();

```



```
        // delay_ms(100);  
    }  
}  
else  
{  
    go();  
}  
delay_ms(10); // tutup 1  
}  
}
```

## **BAB V**

### **KESIMPULAN**

#### **5.1 Kesimpulan**

Setelah melalui tahap perancangan, pembuatan dan pemograman robot, maka kami mengambil kesimpulan sebagai berikut :

1. Tegangan sumber yang digunakan sangat mempengaruhi kinerja robot tersebut.
2. Delay kalibrasi servo berbeda dengan delay servo pada program utama, hal ini di sebabkan sensor diletakkan di atas servo.
3. Jenis halangan sangat mempengaruhi pembacaan sensor.
4. Delay pada program sangat mempengaruhi kecepatan eksekusi selanjutnya.
5. Secara keseluruhan, robot terdiri sensor yang diletakkan diatas servo sebagai inputan, MCU sebagai pemproses, driver motor DC sebagai pengatur gerak motor, dan motor DC sendiri sebagai output.

## DAFTAR PUSTAKA

Winoto,Ardi. Mikrokontroler AVR ATmega8/16/32/8535 dan pemrogramannya dengan bahasa C pada win AVR. Penerbit Informatika. Bandung. 2008.