

BAB III

PERANCANGAN DAN PEMODELAN

Pada bab ini akan membahas mengenai perancangan dan pemodelan serta realisasi dari perangkat keras dan perangkat lunak untuk alat pengukur kecepatan dengan sensor infra merah.

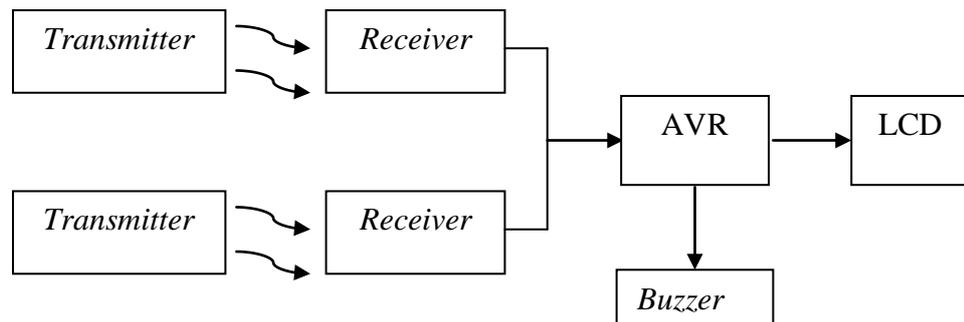
3.1. Perancangan Perangkat Keras (*Hardware*)

Perancangan alat pengukur kecepatan dengan sensor infra merah dimulai dengan perancangan *hardware* yaitu : sensor infra merah (*receiver* dan *transmitter*), *ATMega 16*, LCD 2x16, *buzzer*, dan kabel.

3.1.1. Blok Diagram

Alat akan dirancang menggunakan 2 buah sensor infra merah yang diberi jarak antar sensor sejauh 1 meter. Data untuk proses perhitungan didapat dari benda yang terdeteksi oleh sensor infra merah. Setelah data diproses di *microcontroller*, didapat kecepatan benda dan akan dikonversi dalam satuan kilometer per jam (*km/h*). Sedangkan data panjangnya benda dalam satuan *centimeter* (*cm*). Hasil berupa kecepatan dan panjangnya benda akan ditampilkan ke LCD 2x16. Jika kecepatan melebihi batas kecepatan yang sudah ditentukan yaitu 3.96 *km/h*, *buzzer* akan berbunyi.

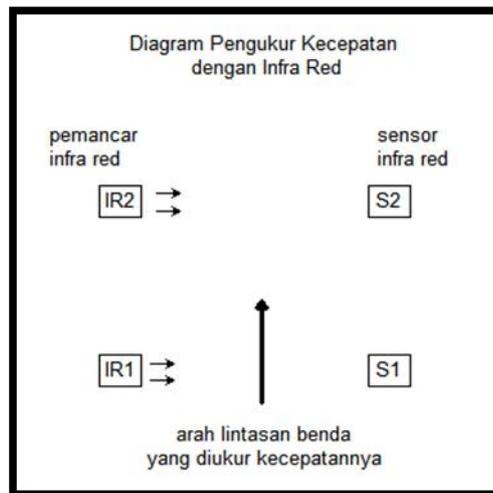
Batas kecepatan berbunyinya *buzzer* yaitu 3.96 km/h ditentukan berdasarkan percobaan yang telah dilakukan. Percobaan yang dilakukan menggunakan benda yang digerakkan dengan tangan sehingga didapat ketika kecepatan 3.96 km/h tangan sudah bergerak cepat.



Gambar 3.1. Blok Diagram

3.1.2. Sensor Infra Merah

Alat pengukur kecepatan benda ini bekerja berdasarkan respon sensor yang dihubungkan ke *microcontroller*. Sensor yang dipakai berupa *transmitter* dan *receiver* infra merah sebanyak dua pasang. *Transmitter* yang digunakan berupa LED yang memancarkan sinar infra merah sedangkan *receiver* yang digunakan adalah fotodioda. Fotodioda bekerja berdasarkan terdeteksinya sinar infra merah. Perancangan penempatan sensor seperti gambar 3.2.

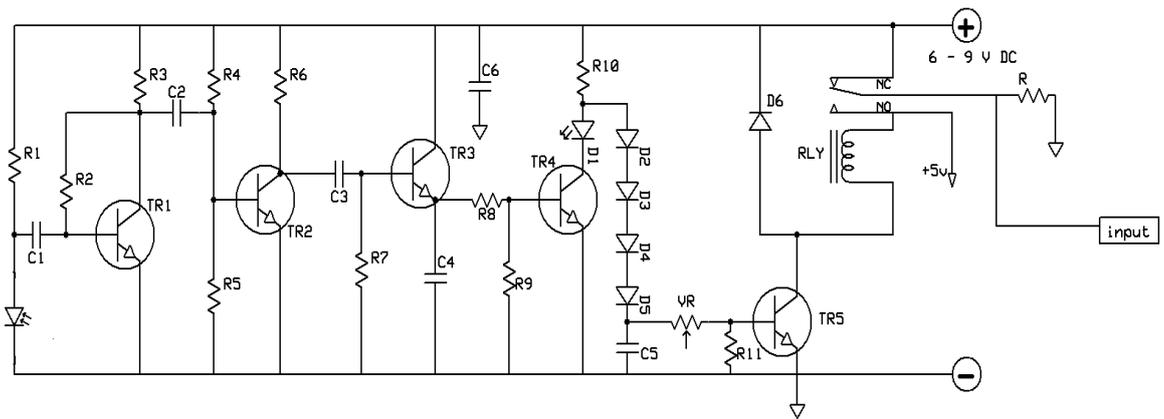


Gambar 3.2. Penempatan Sensor

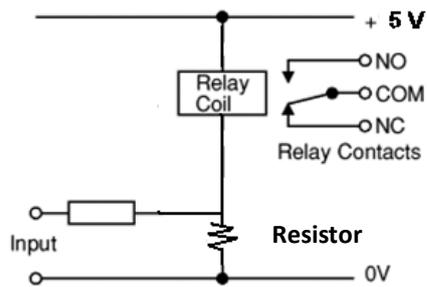
Pada kondisi normal kedua sensor S1 dan S2 akan menerima cahaya infra merah dari IR1 dan IR2. Jika ada benda melintas sesuai dengan arah lintasan (tanda panah ke atas pada gambar 3.2), maka benda tersebut akan menghalangi jalur cahaya infra merah IR1 untuk dapat mencapai sensor S1. Kondisi ini oleh *microcontroller* sebagai saat 'start' perhitungan waktu. Kemudian benda akan bergerak sampai melewati IR2 dan S2, kondisi ini dianggap oleh *microcontroller* sebagai saat 'stop' perhitungan waktu.

Tegangan *output* yang dihasilkan *receiver* infra merah akan masuk ke *microcontroller*. *Output receiver* infra merah berasal dari *relay*. Kondisi saat *receiver* menerima sinar infra merah dari *transmitter* dibaca oleh *microcontroller* sebagai *logic low*, sedangkan kondisi saat *receiver* tidak menerima sinar infra merah akan dibaca oleh *microcontroller* sebagai *logic high*.

Receiver infra merah sebagai penerima sinyal, akan terhubung dengan rangkaian penguat transistor seperti gambar 3.3. *Output* dari penguat transistor ini akan terhubung dengan *relay*. *Output* yang dihasilkan oleh *relay* akan terhubung dengan suatu rangkaian (gambar 3.4). *Output* dari rangkaian (gambar 3.4) menjadi *input microcontroller*.

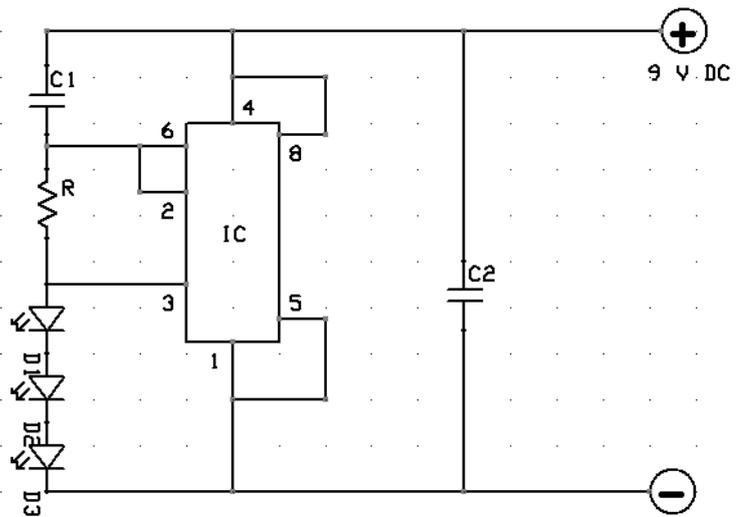


Gambar 3.3. Rangkaian *Receiver* Infra merah



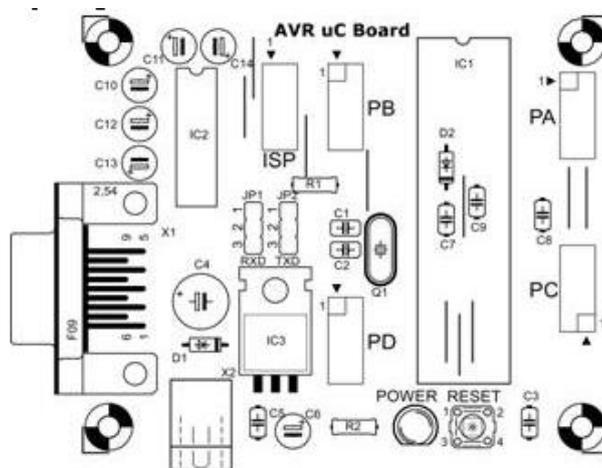
Gambar 3.4. Rangkaian *Output Relay*

Dalam rangkaian *transmitter* seperti gambar 3.5, pembangkit sinyal akan diteruskan ke LED yang kemudian akan memancarkan sinar infra merah. IC yang digunakan adalah IC 17555 yang berfungsi sebagai *timer*.



Gambar 3.5. Rangkaian *Transmitter* Infra Merah

3.1.3. *ATMega 16*



Gambar 3.6. *ATMega 16* Board

AVR *ATMega 16* ini dipakai sebagai tempat di mana program berada. *Port A* dipakai sebagai input untuk sensor infra merah. Dengan data *port input* nya adalah : PA.0 dan PA.7.

Pada *Port C* digunakan sebagai *port* untuk *output* tampilan LCD, tampilan berupa informasi-informasi yang di perlukan seperti kecepatan dan panjangnya kendaraan. Konfigurasinya adalah sebagai berikut:

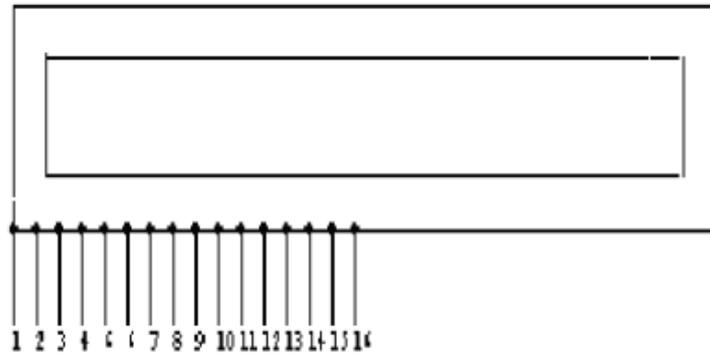
Tabel 3.1. Konfigurasi *Port C*

| | | |
|-------------------|-----|------------|
| <i>PORT</i> bit 0 | RS | LCD pin 4 |
| <i>PORT</i> bit 1 | RD | LCD pin 5 |
| <i>PORT</i> bit 2 | EN | LCD pin 6 |
| <i>PORT</i> bit 3 | – | FREE |
| <i>PORT</i> bit 4 | DB4 | LCD pin 11 |
| <i>PORT</i> bit 5 | DB5 | LCD pin 12 |
| <i>PORT</i> bit 6 | DB6 | LCD pin 13 |
| <i>PORT</i> bit 7 | DB7 | LCD pin 14 |

Port B digunakan sebagai *output*, yaitu untuk *buzzer* yang berfungsi sebagai *output* alarm. Dengan data *port output* PB.0.

3.1.4. LCD 2x16

LCD ini dipakai sebagai output dengan konfigurasi *pin* – *pin* nya seperti gambar 3.7.



Gambar 3.7. Konfigurasi LCD

- V_{ss} (*pin 1*), merupakan *pin* tegangan referensi 0 Volt (*ground*).
- V_{cc} (*pin 2*), merupakan *pin* tegangan +5 Volt.
- V_{ee} (*pin 3*), berfungsi untuk mengatur *contrast* LCD. V_{ee} dihubungkan dengan *ground*.
- RS (*pin 4*), *Port C* bit 0.
- R/W (*pin 5*), *Port C* bit 1.
- EN (*pin 6*), *Port C* bit 2.
- DB4 (*pin 11*) – DB7 (*pin 14*), *Port C* bit 4 – 7.
- V (*pin 15*), dihubungkan ke pin 2 dengan resistor 220 Ω
- V (*pin 16*), dihubungkan ke pin 1.

3.1.5. Buzzer

Buzzer merupakan alat yang berfungsi sebagai alarm penanda. Anoda *buzzer* dihubungkan ke VCC. Sedangkan

katodanya dihubungkan sebagai *input* ATmega 16 di Port B, dengan konfigurasi PB.0.

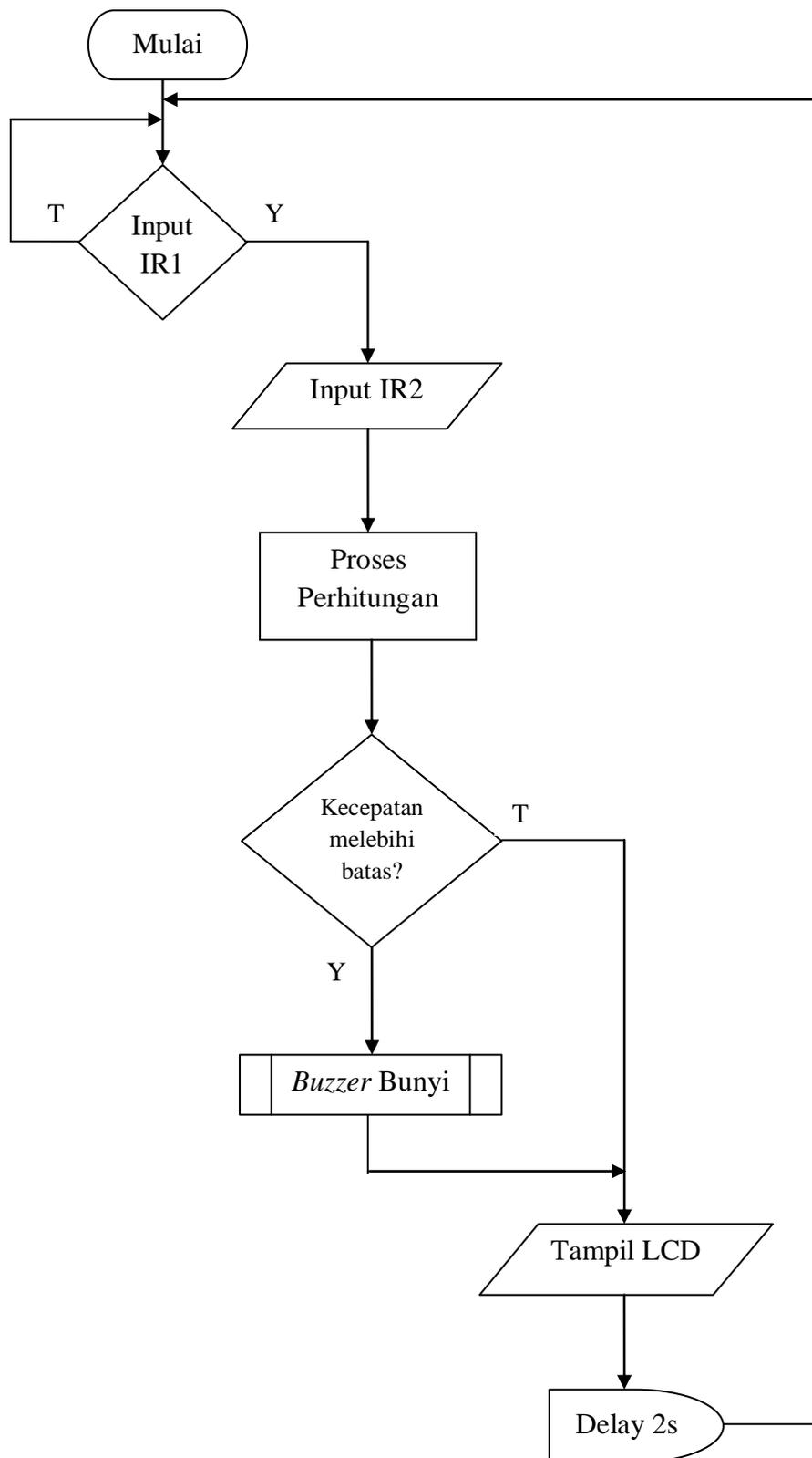
3.2. Perancangan Perangkat Lunak (*Software*)

3.2.1. Diagram Alir

Program dimulai ketika alat dinyalakan dengan menggunakan sumber tegangan sebesar 7.5 Volt. Setelah itu program akan mulai membaca *input* yang berasal dari sensor infra merah pertama. Program akan terus menunggu sampai ada *input* infra merah pertama. Apabila *input* dari infra merah pertama sudah masuk, program akan melanjutkan pembacaan *input* yang berasal dari infra merah kedua. Program akan terus menunggu sampai ada *input* dari infra merah kedua.

Jika *input* dari infra merah pertama dan kedua sudah ada, program akan melanjutkan proses masukan dari kedua infra merah dengan menggunakan proses perhitungan dengan menggunakan rumus yang ada. Misalkan waktu untuk benda tersebut bergerak dari S_1 ke S_2 adalah t dan jarak antara S_1 dan S_2 adalah s , maka diperoleh kecepatan benda tersebut dengan rumus fisika sederhana: $v = s/t$. Sedangkan untuk menghitung panjang benda dengan rumus $x = v \cdot t_1$, di mana t_1 adalah waktu ketika S_1 putus sampai terhubung kembali.

Hasil perhitungan akan muncul di LCD 2x16. Apabila kecepatan benda melampaui batas kecepatan yang sudah ditentukan maka program akan memanggil *procedure* untuk menyalakan *buzzer*. Tampilan pada LCD akan menampilkan hasil perhitungan selama 2 detik. Program akan mengulang mendeteksi *input* infra merah pertama, dan melanjutkannya ke proses selanjutnya.



Gambar 3.8. Diagram Alir

3.2.2. CodeVision ATmega

Software yang dipakai dalam pembuatan program untuk alat pengukur kecepatan ini adalah dengan menggunakan CodeVision AVR. Code Vision AVR C *Compiler* pada dasarnya merupakan perangkat lunak pemrograman *microcontroller* keluarga AVR berbasis bahasa C.

Pada *tool* CodeVision AVR ini bisa ditentukan *port-port* dari *microkontroller* AVR yang berfungsi sebagai *input* maupun *output*, serta bisa juga ditentukan tentang penggunaan fungsi-fungsi *internal* dari AVR.

Dalam program ini terdapat proses berjalannya alat secara keseluruhan. Program ini diunduh secara langsung ke chip *ATmega 16*. Langkah – langkah pembuatan program dengan menggunakan CodeVision AVR adalah sebagai berikut:

1. Jalankan CodeVisionAVR, kemudian klik *File*→*New*, Pilih *Project*.
2. “*Do you want to use the CodeWizardAVR?*” Klik Yes.
3. Pilihlah *Chip* yang digunakan, *chip* : ATmega16, *clock* : 16.000000 MHz.
4. Lakukanlah setting sebagai berikut :
Port A = Sebagai *Input*
Port B = Sebagai *Output*
Port C = Sebagai *Output* Tampilan LCD
5. Klik *File*→*Generate, Save and Exit*.
6. Buatlah *source code*.

7. Klik *Project* → *Configure*, kemudian pilih menu *After Make* atau *After Build* dan aktifkan *Program*. Klik OK jika sudah.
8. Untuk meng-*compile project*, klik *Project* → *Make*. Pastikan program tidak ada yang *error*.
9. Jika tidak ada *error* maka *file* siap diunduh ke *chip*. Pastikan koneksi kabel *downloader* dan chip sudah terpasang dengan benar, klik → *Tools Chip Programmers*.
10. Nyalakan *power supply* dan klik *Program All*. Tunggu hingga proses pengunduhan *Chip* selesai.