

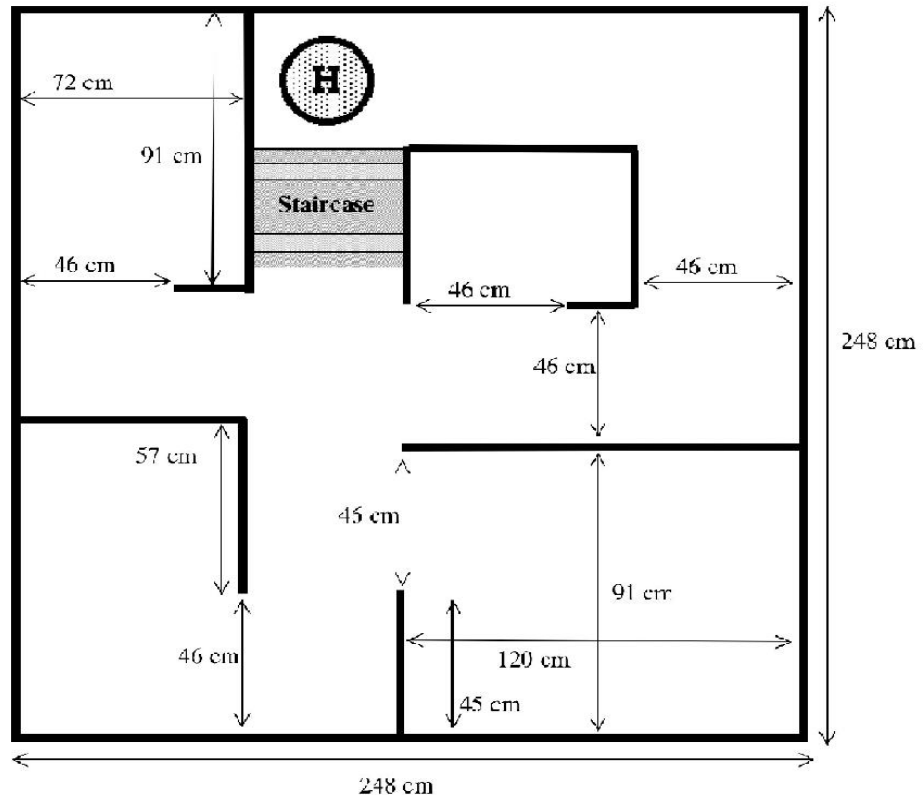
3. PERENCANAAN

Pada bab ini akan dibahas mengenai perencanaan seluruh sistem robot yang dibuat, baik *hardware* maupun *software*. Perencanaan *hardware* meliputi perencanaan dan pembuatan mekanik serta elektrik robot. Perencanaan *software* meliputi *flowchart-flowchart* yang mendukung jalannya robot.

3.1. Perencanaan Sistem

Perencanaan sistem yang dibuat harus sesuai dengan target dan peraturan kontes robot KRCI-2006. Robot yang dibuat harus dapat melintas di atas struktur arena pertandingan, kemudian mencari lilin di sebuah ruangan, mematikan lilin tersebut dalam waktu singkat, lalu kembali ke posisi *start*.

Arena pertandingan ini merupakan simulasi dari sebuah rumah pada kenyataannya. Terdapat empat buah ruangan dengan ukuran yang berbeda. Berikut ini adalah gambar dari arena pertandingan yang akan dipakai :



Gambar 3.1. Denah Arena Pertandingan

Untuk mempermudah dalam merencanakan sistem ini, maka keseluruhan dari sistem ini dibagi menjadi empat bagian.

Bagian pertama adalah sistem kerja robot melintasi lantai rumah (arena pertandingan). Warna lantai secara keseluruhan adalah hitam. Dan sedangkan warna tembok secara keseluruhan adalah putih. Pada bagian ini, robot harus mampu mendeteksi jarak dirinya dengan tembok rumah, serta mendeteksi adanya ruangan. Agar sistem ini dapat bekerja dengan baik, maka robot dilengkapi dengan sensor-sensor. Untuk dapat mendeteksi jarak robot dengan tembok, agar robot tidak menyentuh tembok, digunakan sensor ultrasonik yang diletakkan pada bagian depan robot dan samping kanan robot. Secara keseluruhan denah rumah telah diprogramkan pada mikrokontroler robot, sehingga dengan bantuan sensor ultrasonik robot dapat melintasi denah rumah tersebut dengan baik. Untuk mendeteksi adanya ruangan atau robot telah memasuki ruangan, maka robot ini dilengkapi juga dengan *photoreflector*. Sensor ini digunakan untuk mendeteksi adanya garis putih yang terletak pada setiap pintu masuk empat ruangan yang ada.

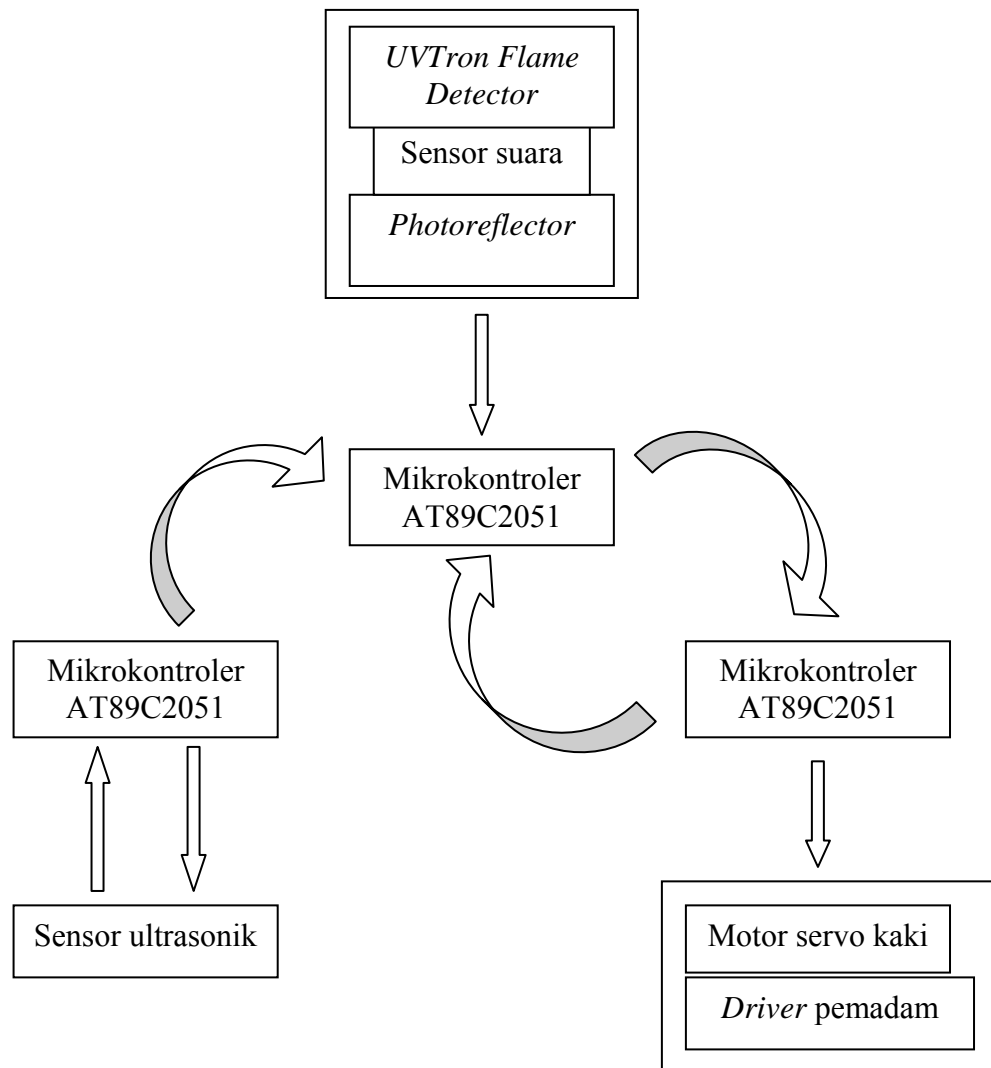
Bagian kedua adalah sistem robot mengenali cahaya lilin. Pada bagian ini robot akan masuk pada tiap-tiap ruangan untuk mendeteksi adanya lilin yang menyala. Lilin hanya akan dinyalakan pada salah satu ruangan pada rumah secara random. Untuk mendeteksi adanya cahaya lilin digunakan *UVTron flame detector*.

Bagian ketiga adalah sistem dari robot untuk mematikan api lilin. Robot harus mematikan cahaya lilin dalam radius kurang dari 30 centimeter dari lilin tersebut (harus ada sebagian dari robot yang memasuki daerah radius 30 centimeter ini). Pada radius 30 centimeter di sekeliling lilin akan diberikan warna putih. Untuk mengenali warna putih tersebut digunakan *photoreflector*. Setelah robot masuk radius tersebut, maka robot akan mematikan cahaya lilin tanpa menyentuh lilin tersebut. Alat yang digunakan robot untuk mematikan api lilin adalah kipas angin (motor DC). Kipas angin (motor DC) ini akan diarahkan ke hadapan lilin. Keseluruhan waktu sampai robot mematikan api lilin ini maksimum adalah 5 menit.

Bagian keempat adalah sistem robot untuk kembali ke posisi *start*. Setelah robot menemukan lilin dan mematikan lilin tersebut di salah satu ruangan di dalam rumah (yang disimulasikan dalam sebuah arena), maka robot akan langsung

kembali ke posisi *start* dalam kurun waktu maksimum selama 2 menit. Sehingga total waktu robot beroperasi adalah selama 7 menit. Cara kerja sistem ini hampir sama dengan cara kerja sistem robot melintasi lantai rumah.

Jadi ada empat buah sistem yang bekerja untuk dapat mengoperasikan robot ini. Berikut ini adalah bagan utama dari keseluruhan sistem robot cerdas pemadam api:



Gambar 3.2. Bagan Sistem Lengkap Robot Pemadam Api

3.2. Perencanaan Hardware

Di dalam perencanaan *hardware* ini terbagi menjadi 2 bagian utama, yaitu perencanaan mekanik dan perencanaan elektrik. Perencanaan mekanik ini terdiri dari perencanaan desain robot, perencanaan mekanik kaki robot dan mekanik

pemadam api. Perencanaan elektrik terdiri dari 4 bagian, yaitu *Mainboard AT89C2051*, sensor suara, *driver Flame Detector*, dan *Photoreflector*.

3.2.1. Perencanaan Mekanik

Perencanaan mekanik ini meliputi tiga bagian, yaitu perencanaan desain robot, perencanaan mekanik kaki robot dan perencanaan mekanik penyemprot.

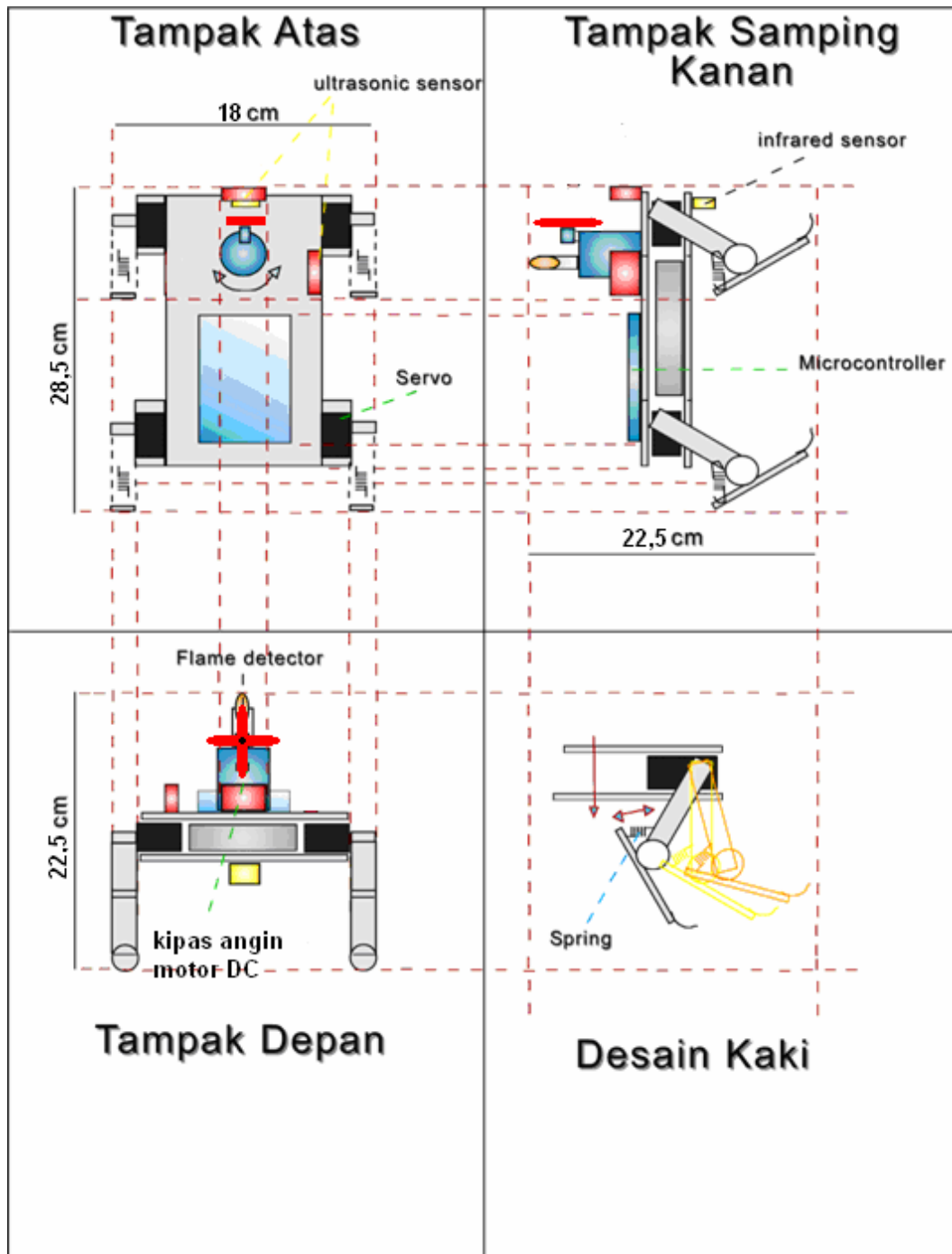
3.2.1.1. Perencanaan Desain Robot

Pada desain robot ini disesuaikan dengan ketentuan yang ada pada peraturan KRCI 2006 yang berlaku pada kategori robot berkaki yaitu sebagai berikut:

- Ukuran robot maksimum untuk robot berkaki adalah panjang = 46 cm, lebar = 31 cm, dan tinggi = 31 cm.
- Berat dari robot tidak memiliki batasan
- Material dari badan robot bebas, bisa dari material apa saja.

Dari ketentuan robot yang diperbolehkan dalam kontes ini, maka dibuat desain robot berkaki sebagai berikut:

- Dimensi robot: panjang = 28,5, lebar = 18 cm, dan tinggi = 22,5 cm
- Kecepatan dari robot adalah 7,85 cm / det.
- Berat Robot adalah 2,7 kg
- Alat pemadam adalah kipas angin motor DC
- Material kerangka robot menggunakan lempeng PCB dan aluminium.

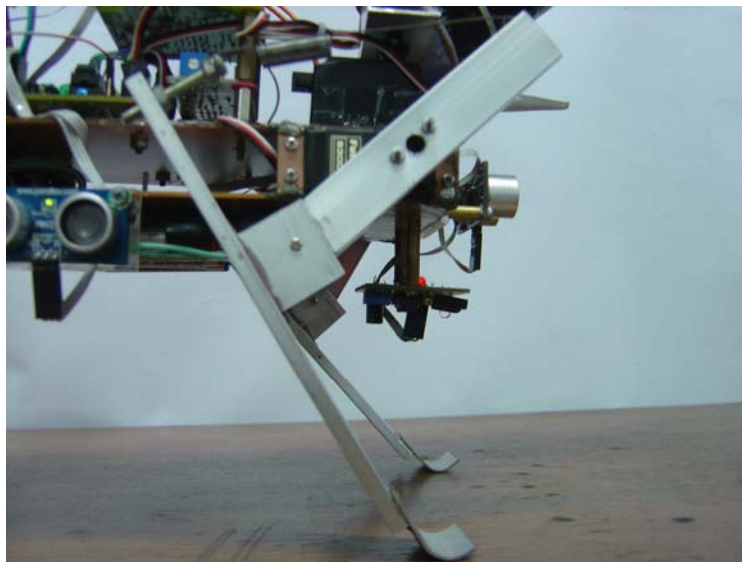


Gambar 3.3. Gambar Desain Robot Cerdas Pemadam Api

3.2.1.2. Perencanaan Mekanik Kaki Robot

Mekanik kaki robot dirancang menyerupai kaki pada hewan berkaki empat. Terdapat empat buah kaki yang masing-masing terpasang pada sebuah motor servo sebagai penggerakannya. Sehingga robot ini hanya memiliki satu buah derajat kebebasan (DOF) untuk setiap kakinya. Pada kaki robot dipasangkan

sebuah pegas untuk setiap kakinya. Pegas ini digunakan untuk mengembalikan posisi kaki robot setelah robot melakukan kayuhan ke lantai.

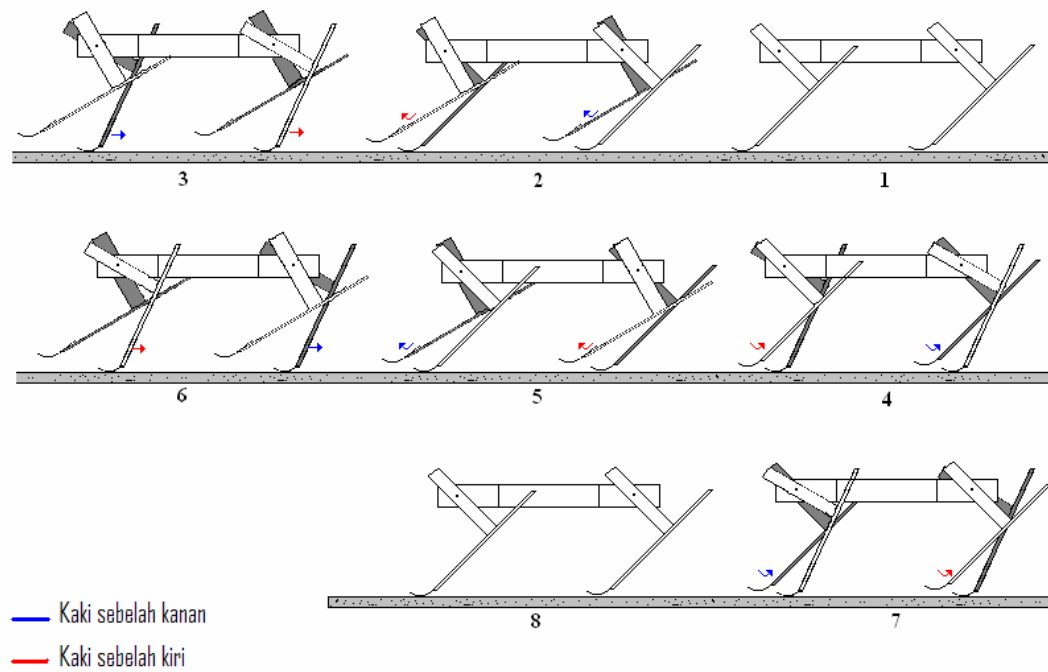


Gambar 3.4. Konstruksi Kaki Robot Pemadam Api

Pada robot diprogramkan lima model jalan untuk dapat menyelesaikan permasalahan, yaitu jalan maju, jalan maju ke kanan, jalan maju ke kiri, belok kanan, dan belok kiri. Cara jalan yang dilakukan pada robot yaitu kaki robot bagian kiri depan dan bagian kanan belakang bergerak secara bersamaan, kemudian bergantian dengan dengan kaki bagian kanan depan dan bagian kiri belakang yang juga bergerak secara bersamaan. Untuk jalan maju ke kanan memiliki cara jalan yang hampir sama, namun yang berbeda adalah pada kaki sebelah kanan mengangkat kaki jauh lebih lebar dibandingkan kaki yang sebelah kiri. Begitu juga pada saat menapakkan kaki ke lantai untuk kaki yang berada di sebelah kanan lebih menapakkan kaki (seperti membuat dorongan ke belakang) dibandingkan kaki yang berada di sebelah kiri. Hal ini berbanding terbalik dengan cara jalan maju ke kiri, yaitu pada kaki sebelah kiri mengangkat kaki jauh lebih lebar dibandingkan kaki yang sebelah kanan. Begitu juga pada saat menapakkan kaki ke lantai untuk kaki yang berada di sebelah kiri lebih menapakkan kaki dibandingkan kaki yang berada di sebelah kanan.

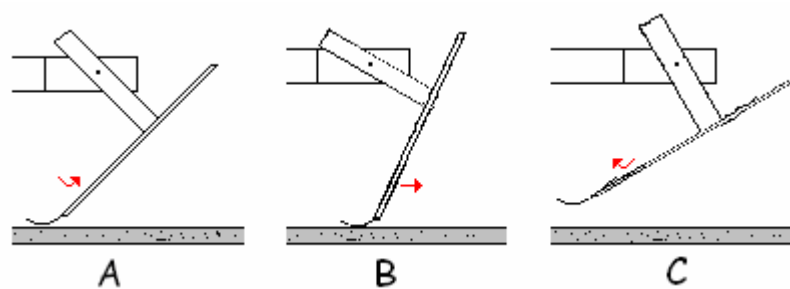
Cara jalan maju ke kanan dan maju ke kiri juga terjadi pada saat robot belok kanan atau ke kiri. Yang berbeda adalah perbandingan antara kaki kanan

dan kiri pada saat mengangkat kaki dan menapakkan kaki. Keseluruhan cara jalan robot ini dapat diilustrasikan pada gambar berikut.



Gambar 3.5. Cara Jalan Kaki Robot Pemadam Api

Dari gambar di atas dapat dijelaskan cara jalan robot untuk maju. Pada program langkah kaki ini dibagi dalam tiga posisi kaki. Tiga posisi kaki tersebut dapat dilihat pada gambar 3.6 di bawah ini.



Gambar 3.6. Tiga Posisi Kaki Pada Tiap Kaki Robot

Urutan dari langkah kaki robot dimulai dari no 1 hingga no 8. Berikut adalah penjelasan dari urutan langkah yang telah diprogramkan pada robot:

1. Posisi awal seluruh kaki robot berada pada posisi A. Posisi ini merupakan posisi robot untuk berdiri.

2. Pada urutan ini posisi kaki bagian kanan depan dan kiri belakang berada pada posisi yang sama yaitu posisi A, sedangkan untuk kaki bagian kiri depan dan kanan belakang akan berada pada posisi C.
3. Posisi kaki bagian kiri depan dan kanan belakang akan berada pada posisi tetap yaitu posisi C. Pada urutan ini posisi kaki bagian kanan depan dan kaki bagian kiri belakang akan berada pada posisi B. Dalam hal ini mengakibatkan robot mengayuh kebelakang (kondisi menendang ke belakang).
4. Kemudian pada urutan ini posisi kaki kiri bagian depan dan kaki kanan bagian belakang akan merubah posisinya menjadi posisi A. Namun pada urutan ini posisi kaki bagian kanan depan dan kiri belakang masih tetap pada posisi B.
5. Posisi yang tetap terjadi pada kaki bagian kiri depan dan kaki bagian kanan belakang. Pada kaki bagian kanan depan dan kiri belakang langsung berganti posisi menjadi posisi C.
6. Urutan ini berkebalikan dengan urutan nomor 3, yaitu kaki kiri depan dan kaki kaki kanan belakang berada pada posisi B (menendang ke belakang guna mengimbangi jalan robot yang telah terjadi pada urutan nomor 3). Dan pada kaki bagian kanan depan dan kaki bagian kiri belakang tetap pada posisi sebelumnya.
7. Dengan posisi kaki yang sama pada kaki bagian kiri depan dan kaki bagian kanan belakang, maka pada kaki bagian kanan depan dan kaki bagian kiri belakang berganti pada posisi A.
8. Kemudian pada urutan terakhir ini posisi dari ke empat kaki tersebut akan kembali pada posisi A secara keseluruhan, sama seperti pada posisi awal robot sebelum berjalan.

Dari keseluruhan penjelasan cara jalan robot di atas, maka dapat dijelaskan mengenai algoritma motor servo yang dipakai untuk menjalankan robot ini. Untuk menggerakkan motor servo sesuai dengan cara jalan di atas, maka akan didapatkan sebuah algoritma seperti pada program di bawah ini (diambil dari software mikrokontroler jalan pada lampiran halaman 92).

```

pwm: djnz  r_kidep,out
      cpl   P1.7
      mov   r_kidep,#0ffh
      jnb   P1.7,out
      mov   r_kidep,r1_kidep
out:  ret

```

Dari contoh program di atas hanya diambil untuk satu buah motor servo. Dari program tersebut terdapat variabel yang harus diisi agar dapat membentuk suatu pulsa PWM yang diinginkan. Misalkan pada variabel (*r_kidep* dan *r1_kidep*) tersebut diberi nilai 15, maka servo akan berada pada posisi *center* (posisi robot berdiri tegak). Hal ini didapatkan dengan bantuan *interrupt timer 0* (mode *8 bit auto reload* setiap 0,1 ms) yang digunakan.

Pertama kali *r_kidep* dan *r1_kidep* diisi dengan nilai 15, kemudian program akan mengurangi nilai tersebut. Jika nilai tidak sama dengan nol, maka program akan keluar dari rutin dan menunggu adanya *interrupt timer* lagi. Jika nilai tersebut sudah nol, maka program akan membalik nilai pada P1.7 ('1' menjadi '0'). Kemudian *r_kidep* akan diisi dengan FF_H yang digunakan untuk besar pulsa *low*-nya. Dan akan dicek apakah nilai pada P1.7 itu nol? jika nilai pada P1.7 adalah nol, maka program akan keluar dari rutin dan kembali seperti proses saat memberikan pulsa *high* pada P1.7. Setelah melakukan proses yang sama, maka nilai pada P1.7 akan berubah menjadi '1'. Kemudian akan dilakukan pengisian nilai ulang oleh *r1_kidep* kepada *r_kidep*. Dan proses ini akan berlangsung terus menerus, hingga terjadi perubahan nilai pada *r_kidep* dan *r1_kidep*. Hal inilah yang menyebabkan terjadinya suatu PWM yang digunakan untuk menggerakkan servo.

Seperti yang telah dijelaskan bahwa pada robot diprogramkan lima model jalan, yaitu jalan maju, jalan maju kanan, jalan maju kiri, belok kanan, dan belok kiri. Untuk itu didapatkan suatu kombinasi nilai variabel yang digunakan untuk membangkitkan PWM guna menjalankan robot. Berikut adalah tabel dari nilai variabel tersebut.

Tabel 3.1. Nilai Variabel Yang Digunakan Untuk Jalan Robot

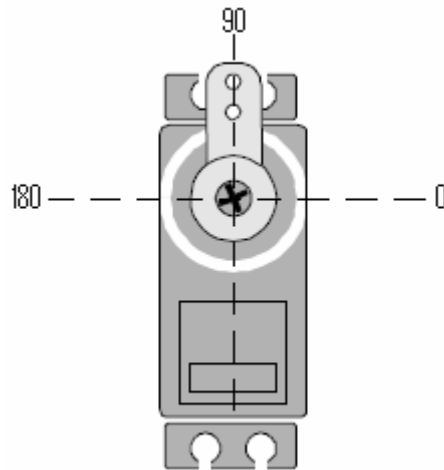
Langkah	Urutan	Servo I Kiri Dpn	Servo II Kanan Dpn	Servo III Kiri Blk	Servo IV Kanan Blk
Maju	1	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°
	2	1,2 ms	1,4 ms	1,6 ms	1,7 ms
		127°	99°	81°	72°
	3	1,2 ms	1,2 ms	1,8 ms	1,7 ms
		127°	127°	63°	72°
	4	1,5 ms	1,2 ms	1,8 ms	1,5 ms
		90°	127°	63°	90°
5	1,5 ms	1,7 ms	1,4 ms	1,5 ms	
	90°	72°	99°	90°	
6	1,7 ms	1,7 ms	1,4 ms	1,2 ms	
	72°	72°	99°	127°	
7	1,7 ms	1,4 ms	1,6 ms	1,2 ms	
	72°	99°	81°	127°	
8	1,5 ms	1,4 ms	1,6 ms	1,5 ms	
	90°	99°	81°	90°	
Maju kanan	1	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°
	2	1,2 ms	1,4 ms	1,6 ms	1,6 ms
		127°	99°	81°	81°
	3	1,2 ms	1,2 ms	1,8 ms	1,6 ms
		127°	127°	63°	81°
	4	1,5 ms	1,2 ms	1,8 ms	1,5 ms
		90°	127°	63°	90°
5	1,5 ms	1,6 ms	1,4 ms	1,5 ms	
	90°	81°	99°	90°	
6	1,7 ms	1,6 ms	1,4 ms	1,3 ms	
	72°	81°	99°	118°	
7	1,7 ms	1,4 ms	1,6 ms	1,3 ms	
	72°	99°	81°	118°	
8	1,5 ms	1,4 ms	1,6 ms	1,5 ms	
	90°	99°	81°	90°	
Maju kiri	1	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°
	2	1,3 ms	1,4 ms	1,6 ms	1,7 ms
		118°	99°	81°	72°
	3	1,3 ms	1,2 ms	1,8 ms	1,7 ms
		118°	127°	63°	72°
	4	1,5 ms	1,2 ms	1,8 ms	1,5 ms
		90°	127°	63°	90°
5	1,5 ms	1,7 ms	1,5 ms	1,5 ms	
	90°	72°	90°	90°	
6	1,6 ms	1,7 ms	1,5 ms	1,2 ms	
	81°	72°	90°	127°	
7	1,6 ms	1,4 ms	1,6 ms	1,2 ms	
	81°	99°	81°	127°	

Tabel 3.2. Nilai Variabel Yang Digunakan Untuk Jalan Robot (Lanjutan)

Langkah	Urutan	Servo I Kiri Dpn	Servo II Kanan Dpn	Servo III Kiri Blk	Servo IV Kanan Blk
Maju kiri	8	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°
Belok kanan	1	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°
	2	1,1 ms	1,4 ms	1,6 ms	1,6 ms
		136°	99°	81°	81°
	3	1,1 ms	1,4 ms	1,9 ms	1,6 ms
		136°	99°	54°	81°
	4	1,5 ms	1,4 ms	1,9 ms	1,5 ms
		90°	99°	54°	90°
	5	1,5 ms	1,5 ms	1,4 ms	1,5 ms
		90°	90°	99°	90°
	6	1,8 ms	1,5 ms	1,4 ms	1,5 ms
		63°	90°	99°	90°
	7	1,8 ms	1,4 ms	1,6 ms	1,5 ms
		63°	99°	81°	90°
	8	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°
Belok kiri	1	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°
	2	1,4 ms	1,4 ms	1,6 ms	1,7 ms
		99°	99°	81°	72°
	3	1,4 ms	1,1 ms	1,6 ms	1,7 ms
		99°	136°	81°	72°
	4	1,5 ms	1,1 ms	1,6 ms	1,5 ms
		90°	136°	81°	90°
	5	1,5 ms	1,8 ms	1,5 ms	1,5 ms
		90°	63°	90°	90°
	6	1,5 ms	1,8 ms	1,5 ms	1,2 ms
		90°	63°	90°	127°
	7	1,5 ms	1,4 ms	1,6 ms	1,2 ms
		90°	99°	81°	127°
	8	1,5 ms	1,4 ms	1,6 ms	1,5 ms
		90°	99°	81°	90°

Dari tabel di atas dapat dilihat bahwa untuk memodelkan jalan robot ini hanya memberikan *range* pulsa yang berbeda. Misalkan pada jalan maju dengan jalan maju kanan, terjadi perbedaan pada nilai yang diberikan. Hal ini juga dilakukan pada model jalan yang lain. Antara nilai yang diberikan pada tiap urutan terdapat suatu *delay* yang mengatur jalannya robot ini, sehingga kaki ini terlihat jalan bergantian.

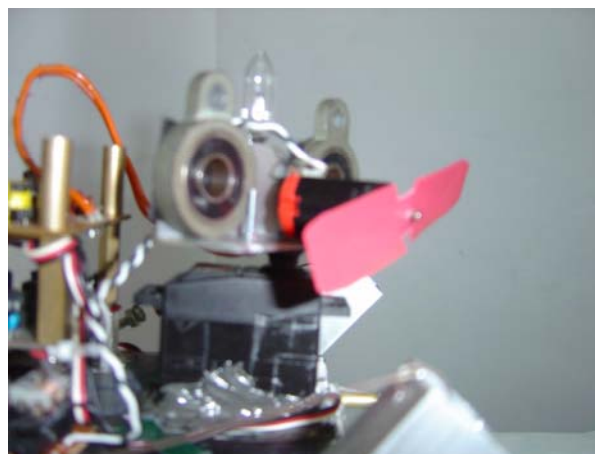
Pada tabel terdapat nilai yang menunjukkan posisi servo dalam satuan derajat. Posisi ini diasumsikan pada posisi servo yang seperti tampak pada gambar di bawah ini.



Gambar 3.7. Posisi servo dalam derajat

3.2.1.3. Perencanaan Mekanik Pemadam Api

Mekanik pemadam api ini menggunakan sebuah motor DC yang diberi baling-baling, sehingga bekerja seperti sebuah kipas angin. Kipas ini diletakkan di atas sebuah servo agar kipas ini dapat bergerak ke kanan hingga ke kiri pada saat melakukan proses pemadaman api. Pada gambar 3.8 menunjukkan gambar kipas yang telah terpasang pada badan robot.



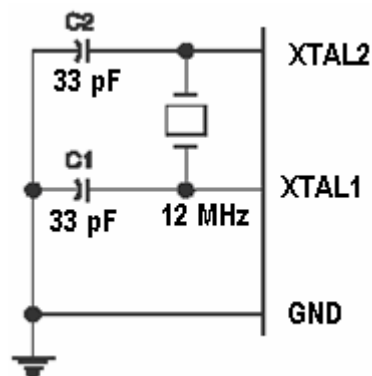
Gambar 3.8. Kipas Pemadam Api

3.2.2. Perencanaan Elektrik

Perencanaan elektrik ini terdiri dari 4 bagian, yaitu minimum sistem AT89C2051, sensor suara, *driver Flame Detector*, *Photoreflector* dan *driver* kipas pemadam api.

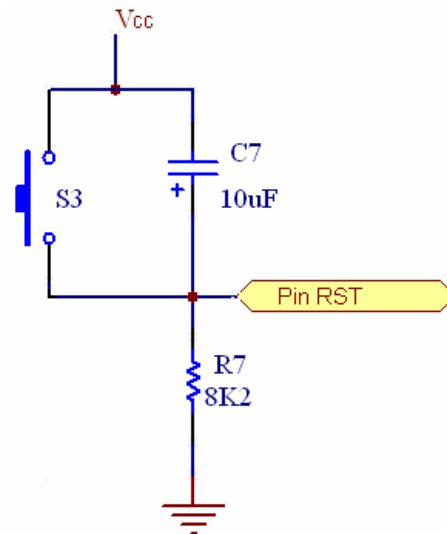
3.2.2.1. Mainboard AT89C2051

Untuk membuat *mainboard* ini diperlukan tiga buah minimum sistem AT89C2051. Secara umum untuk membuat sebuah minimum sistem hanya membutuhkan rangkaian untuk mengaktifkan *internal clock* dan *power on reset* pada mikrokontrolernya. Untuk mengaktifkan rangkaian *internal clock* diperlukan komponen tambahan komponen sebuah *crystal* dan dua buah kapasitor. Nilai kapasitor yang dipakai sesuai datasheet adalah 30pF hingga 33pF. Sedangkan *crystal* yang digunakan bernilai 12 MHz.



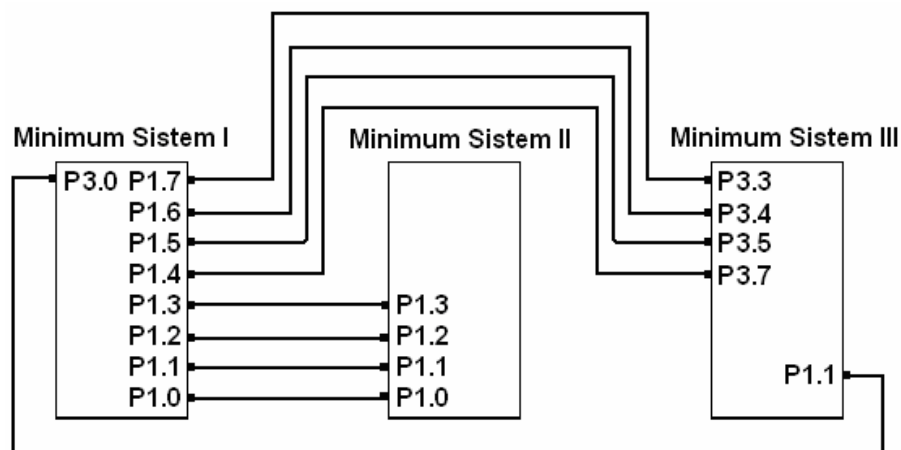
Gambar 3.9. Rangkaian Osilator

Rangkaian *reset* terdiri dari sebuah kapasitor sebesar 10 uF, dan resistor 8,2K Ω dan sebuah *push button* untuk *reset* manual. Nilai komponen tersebut diperoleh dari *datasheet* (yang dapat dilihat pada lembar lampiran). Pada saat mikrokontroler diberi tegangan ($t=0$), tidak diperlukan *reset* secara manual (karena tegangan pada *pin reset* sama dengan Vcc). Setelah selang waktu tertentu maka tegangan pada resistor akan berkurang sesuai dengan rumus $V_R = 5 e^{-t(8,2K \times 10^{-6})}$. Sampai pada saat tertentu dimana tegangan pada *pin reset* lebih kecil dari V_{IH} minimum, maka mikrokontroler akan bekerja.



Gambar 3.10. Rangkaian *Power On Reset*

Secara keseluruhan, ketiga minimum sistem memiliki rangkaian dasar yang sama seperti yang telah dijelaskan sebelumnya. *Mainboard* ini terdiri dari tiga buah minimum sistem yang berkomunikasi secara paralel, dimana masing-masing minimum sistem tersebut memiliki tugas sendiri. Diagram koneksi komunikasi antara ketiga minimum sistem dapat ditunjukkan pada gambar 3.11 di bawah ini.



Gambar 3.11. Diagram Koneksi Komunikasi Ketiga Minimum Sistem

Pembagian tiga tugas minimum sistem yang ada pada *mainboard* ini di antaranya yaitu minimum sistem I (Pusat) bertugas untuk mengolah data yang diterima dari minimum sistem lainnya dan juga meng-*handle* data yang diberikan oleh sensor api, sensor suara, dan sensor garis. Minimum sistem II mempunyai

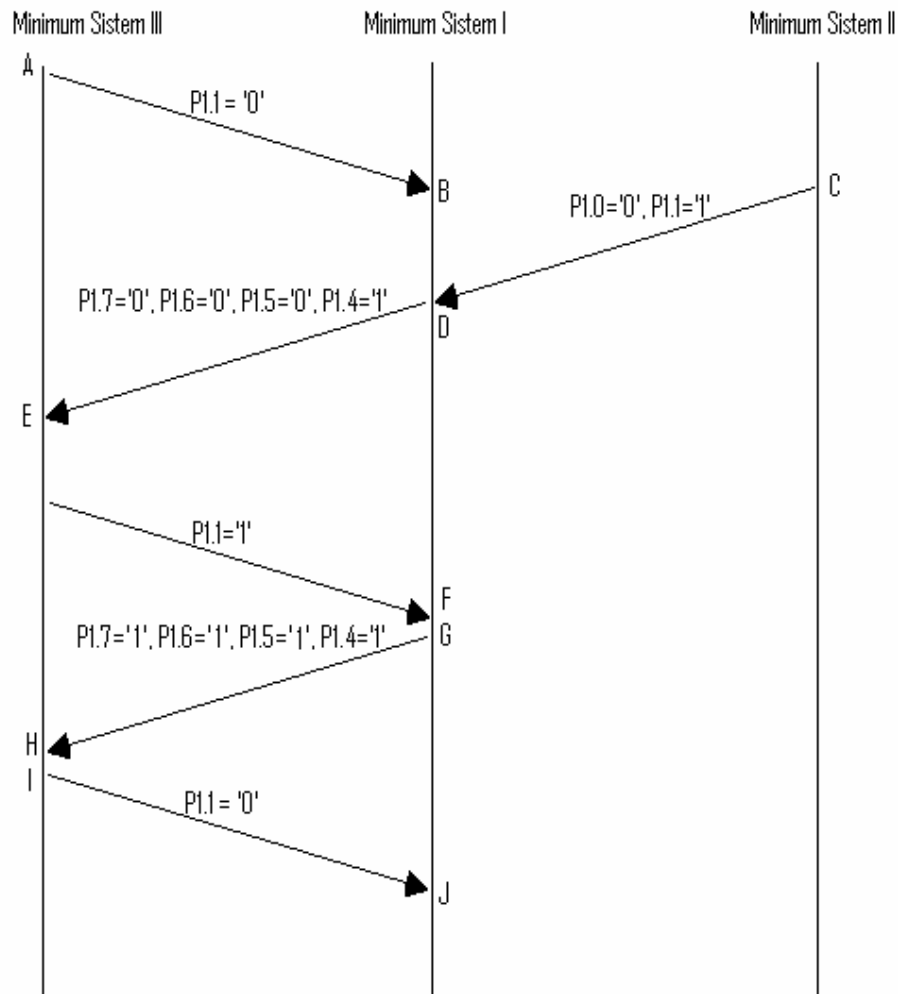
tugas untuk meng-*handle* kerja dari sensor ultrasonik dan mengolah data yang dihasilkan sensor ultrasonik kemudian dikirimkan ke bagian minimum sistem pusat untuk diolah lagi. Hasil olahan data tersebut digunakan untuk memberikan data kepada minimum sistem III. Tugas dari minimum sistem III ini adalah untuk meng-*handle* kerja dari seluruh motor servo yang ada pada robot.

Pada diagram koneksi komunikasi seperti yang tampak pada gambar 3.11. dapat dijelaskan alur sistem komunikasinya sebagai berikut:

- Hubungan koneksi antara minimum sistem I dengan minimum sistem II diatur oleh *pin-pin* pada *port* 1 (P1.0, P1.1, P1.2, P1.3 pada masing-masing minimum sistem). Proses yang dilakukan oleh minimum sistem I adalah menunggu data yang dikeluarkan oleh minimum sistem II (yang mengolah data dari sensor ultrasonik) pada *port* yang telah terkoneksi, sehingga data ini dapat diolah oleh minimum sistem I agar data tersebut dapat mengatur jalannya robot untuk menelusuri arena dengan mengetahui jarak robot dengan dinding pada arena.
- Setelah minimum sistem I menerima data yang diberikan oleh minimum sistem II, maka minimum sistem I akan mengolah data tersebut untuk dikirimkan kepada minimum sistem III. Pada minimum sistem III inilah model jalan robot telah diprogramkan diantaranya jalan maju, jalan maju ke kanan, jalan maju ke kiri, belok ke kanan, dan belok ke kiri.
- Sistem yang digunakan oleh kedua minimum sistem ini (minimum sistem I dan minimum sistem III) adalah dengan komunikasi paralel 4 pin. Dari 4 *pin* yang saling terhubung secara paralel ini, maka akan didapatkan 16 macam kemungkinan data yang bisa dikirimkan oleh minimum sistem I ke minimum sistem III. Karena konfigurasi yang dipakai adalah mulai dari 0000_B sampai 1111_B.
- Dari komunikasi yang dilakukan antara minimum sistem I dengan minimum sistem III terdapat satu *pin* yang saling terhubung, yaitu P3.0 pada minimum sistem I dan P1.1 pada minimum sistem III. Fungsi dari *pin* ini adalah *feedback* yang dikirimkan oleh minimum sistem III bahwa tugas yang telah diberikan oleh minimum sistem I telah dikerjakan. Hal ini

digunakan agar jalan dari robot benar-benar dapat diatur sepenuhnya oleh minimum sistem I.

Secara keseluruhan dapat dilihat pada diagram sekuensial komunikasi yang ada pada gambar 3.12 di bawah ini.

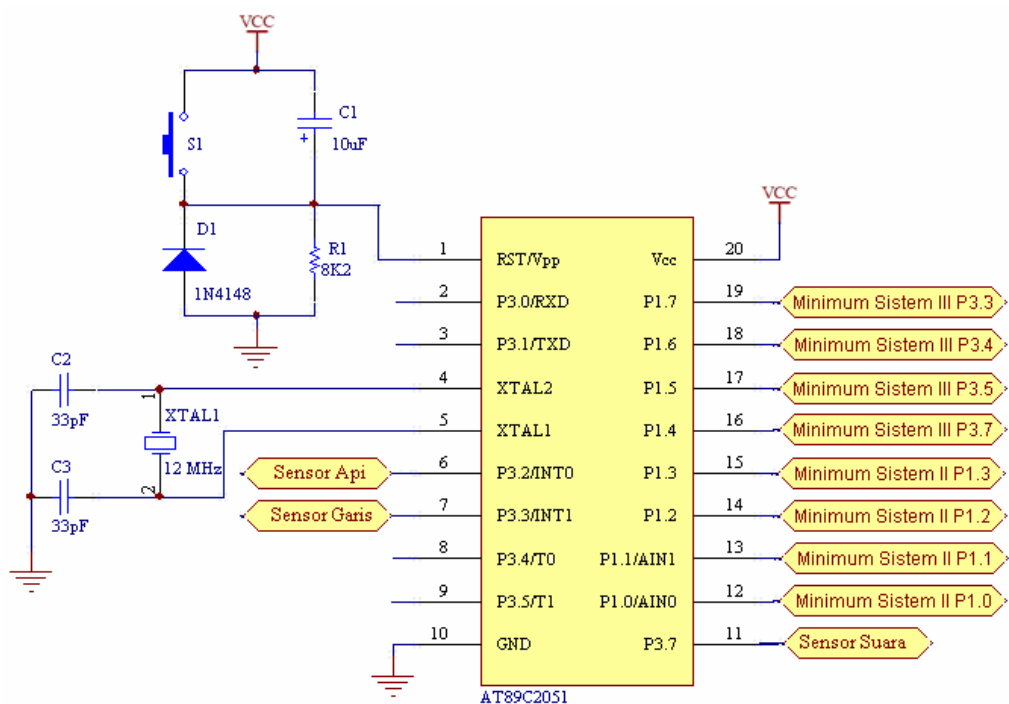


Gambar 3.12. Diagram Sekuensial Komunikasi Minimum Sistem I – Minimum Sistem II – Minimum Sistem III

Dari diagram koneksi dapat dijelaskan bahwa pertama kali yang dilakukan oleh minimum sistem I adalah menunggu adanya *feedback low* dari P1.1 (minimum sistem III) yang diterima P3.0 (minimum sistem I), yang ditunjukkan oleh titik A menuju titik B. Kemudian minimum sistem I akan menunggu data yang dikirimkan oleh minimum sistem II (C → D), yang digunakan untuk

menentukan jalannya robot menelusuri arena. Setelah minimum sistem I menerima data dari minimum sistem II, maka minimum sistem I akan mengirimkan data yang digunakan untuk memerintahkan robot berjalan ke minimum sistem III ($D \rightarrow E$). Setelah minimum sistem III menerima data yang digunakan untuk mengatur jalannya robot dari minimum sistem I, maka minimum sistem III ini akan mengirimkan *feedback high* yang digunakan untuk menandakan bahwa minimum sistem III telah melaksanakan tugas yang telah diperintahkan oleh minimum sistem I.

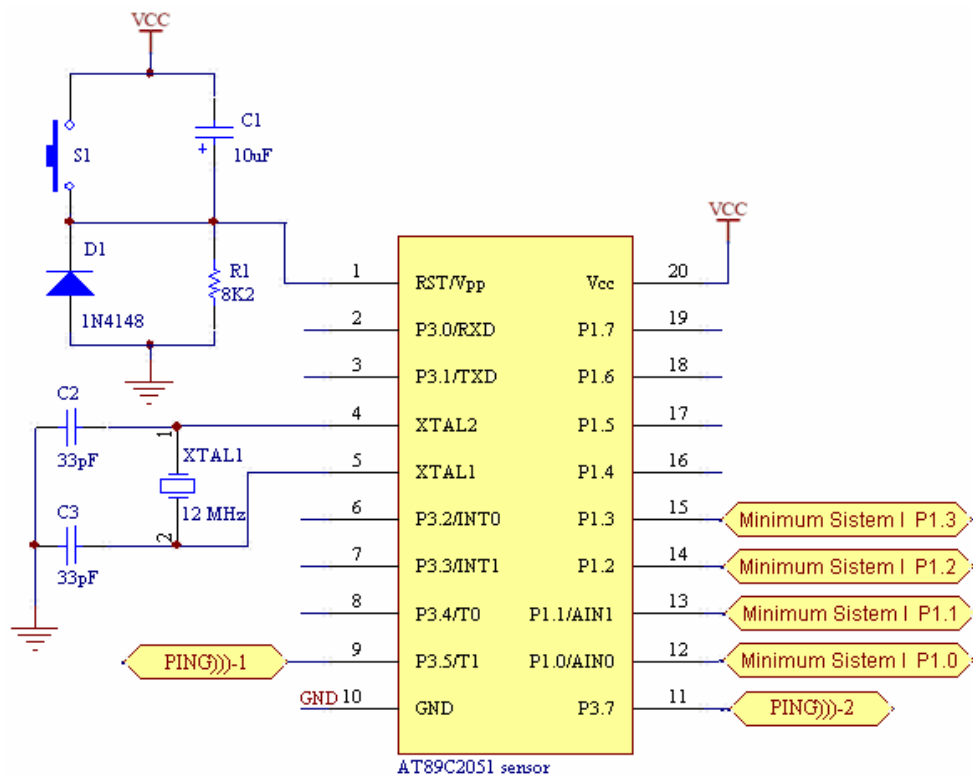
Setelah minimum sistem I menerima *feedback high* dari minimum sistem III, maka minimum sistem I akan mengirimkan data yang digunakan untuk *reset* data yang telah dikirimkan. Kemudian minimum sistem III akan menerima data terakhir dari minimum sistem I dan akan memberikan *feedback low* agar minimum sistem I dapat melakukan proses selanjutnya. Proses ini dilakukan secara terus-menerus sesuai dengan program yang telah diberikan (dapat dilihat pada lampiran pada halaman 74 – 98).



Gambar 3.13. Minimum Sistem I

Selain bertugas untuk mengatur jalannya robot, pada minimum sistem I ini juga berfungsi untuk menerima input dari sensor suara, sensor *UVTron Flame Detector*, dan Sensor *Photoreflector*. Hubungan *pin* pada minimum sistem I dengan sensor tersebut adalah sensor suara terhubung pada *pin* P3.7, sensor *UVTron Flame Detector* terhubung dengan *pin* P3.2, dan sensor *Photoreflector* terhubung dengan *pin* P3.3. Gambar koneksi minimum sistem I dengan ketiga sensor tersebut dapat dilihat pada gambar 3.13.

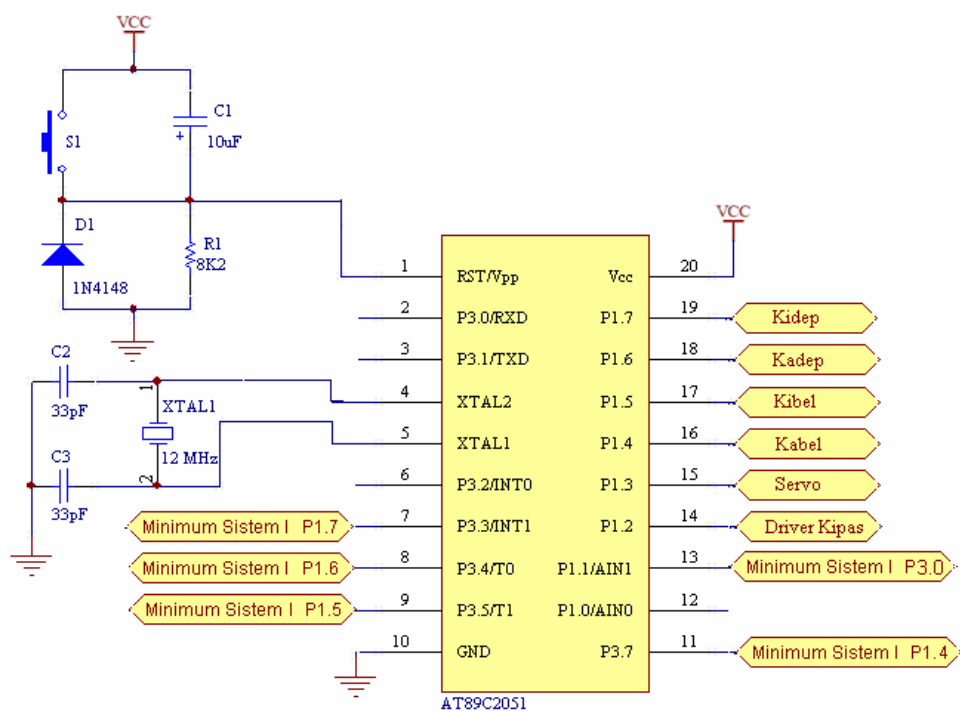
Seperti yang telah dijelaskan di atas bahwa pada minimum sistem II bertugas untuk mengatur kerja dari sensor ultrasonik dan mengolah data yang diperoleh dari pengukuran jarak dari sensor ultrasonik. Pada minimum sistem ini sensor ultrasonik terhubung pada *pin* P3.5 dan P3.7. *Pin* P3.5 terhubung dengan sensor ultrasonik bagian depan, sedangkan *pin* P3.7 terhubung dengan sensor ultrasonik pada sebelah kanan. Gambar koneksi minimum sistem II dengan sensor ultrasonik dapat dilihat pada gambar 3.14 di bawah ini.



Gambar 3.14. Minimum Sistem II

Pada minimum sistem III ini bertugas untuk meng-*handle* jalannya servo sebagai penggerak robot dan *driver* pemadam api pada robot. Pada minimum

sistem III ini meng-*handle* 5 buah motor servo, yaitu 4 buah motor servo untuk kaki robot dan 1 buah motor servo untuk pemadam api. Empat buah motor servo tersebut terhubung dengan *pin* P1.7 (kaki kiri bagian depan), P1.6 (kaki kanan bagian depan), P1.5 (kaki kiri bagian belakang), P1.4 (kaki kanan bagian belakang). Dan satu buah motor servo sisanya terhubung dengan *pin* P1.3. Pada minimum sistem ini *driver* pemadam api akan diatur oleh *pin* P1.2. Hubungan koneksi minimum sistem III dengan 5 buah motor servo dan *driver* pemadam api dapat dilihat pada gambar 3.15 di bawah ini.



Gambar 3.15. Minimum Sistem III

3.2.2.2. Sensor Suara

Sensor ini digunakan untuk mendeteksi adanya suara yang dihasilkan oleh sebuah *buzzer* (sekitar 3 KHz). Besarnya frekuensi yang dapat dideteksi oleh sensor ini dapat diatur sesuai dengan kebutuhan. Sensor ini hanya terdiri dari dua bagian terpenting yaitu rangkaian penguat *mic* dan rangkaian *tone decoder*.



Gambar 3.16. Sensor Suara

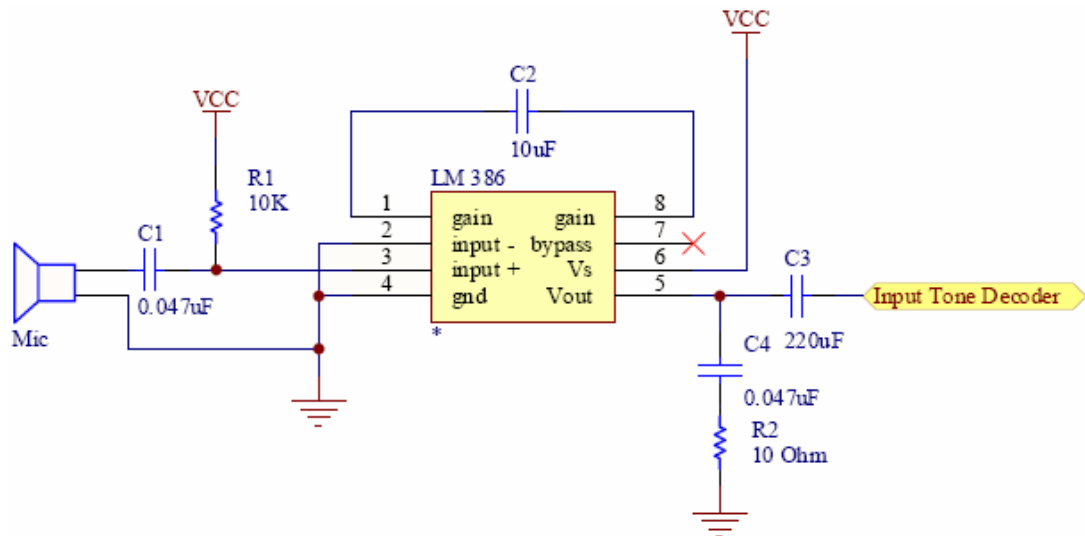
Cara kerja dari sensor ini cukup sederhana. Pertama *mic* (digunakan *mic condenser* sebagai *input* suara) di *pull-up* dan kemudian dihubungkan dengan rangkaian penguat *mic* sehingga sinyal yang nantinya dikeluarkan dapat dibaca oleh rangkaian *tone decoder*. Kemudian keluaran dari rangkaian penguat *mic* tersebut dihubungkan dengan rangkaian *tone decoder*.

Cara kerja dari *tone decoder* ini didesain dengan mengkondisikan sebuah transistor tersaturasi yang terhubung dengan *ground* (*active low*), jika *input* dari *tone decoder* menerima *input* signal yang sesuai dengan frekuensi yang telah ditentukan.

Tone decoder ini terdiri dari I (*input*) dan Q (*output*) *detector* yang dikontrol oleh sebuah *voltage controlled oscillator* yang disesuaikan dengan *center frequency* dari *decoder*.

3.2.2.2.1. Rangkaian Penguat *Mic*

Untuk membuat rangkaian penguat *mic* ini dipakai sebuah IC LM386 yaitu penguat daya audio bertegangan rendah. Sesuai yang tertulis pada *datasheet* LM 386, secara internal penguatan yang diatur sebesar 20 kali. Hal ini guna memperkecil komponen-komponen tambahan lainnya. Namun penambahan kapasitor pada *pin* 1 dan *pin* 8 akan menaikkan penguatan sampai 200 kali. Pada rangkaian penguat *mic* dipilih rangkaian dengan penguatan sebesar 200 kali. Hal ini dikarenakan agar dapat menerima *input* suara (*buzzer*) yang telah diatur pada jarak yang cukup jauh. Rangkaian penguat *mic* tersebut adalah sebagai berikut:

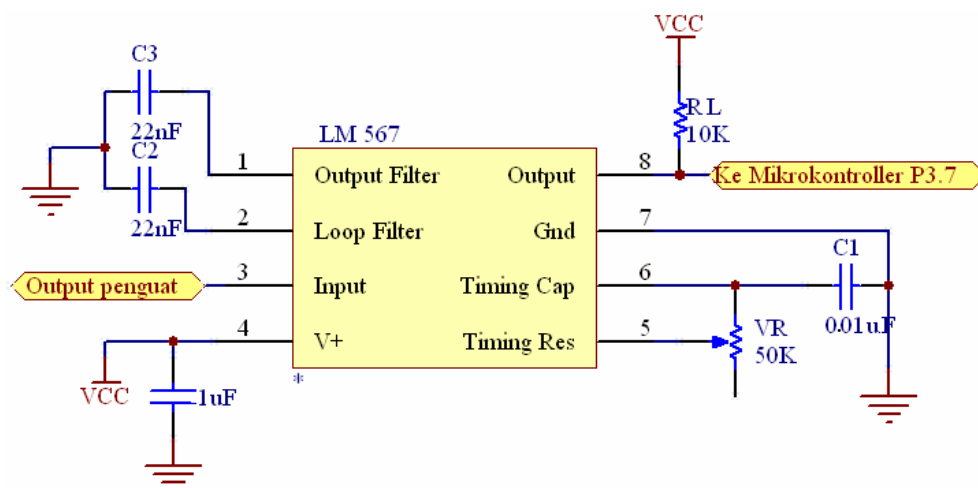


Gambar 3.17. Rangkaian Penguat Mic

Sumber : National Semiconductors. *LM386 Datasheet*. 27 Januari 2006.
<<http://www.national.com/ds/LM/LM386.pdf>>

3.2.2.2.2. Rangkaian *Tone Decoder*

Untuk membuat rangkaian *tone decoder* digunakan IC tipe LM 567. Sesuai dengan *datasheet* yang ada, rangkaian yang dipakai untuk mendeteksi frekuensi *center* adalah sebagai berikut (dalam hal ini frekuensi sebesar 3KHz).

Gambar3.18. Rangkaian *Tone Decoder*

Perhitungan dari rangkaian diatas untuk mencari frekuensi *center* adalah sebagai berikut:

$$f_0 \approx \frac{1}{1,1.VR.C1} \quad (3.1)$$

Dipilih $C1 = 0,01 \mu\text{F}$, dan frekuensi *center* yang diinginkan adalah sebesar 3 KHz maka,

$$f_0 = \frac{1}{1,1 \cdot VR \cdot 0,01 \cdot 10^{-6}}$$

$$3000 = \frac{1}{VR \cdot 11 \cdot 10^{-9}}$$

$$VR = \frac{10^9}{33000}$$

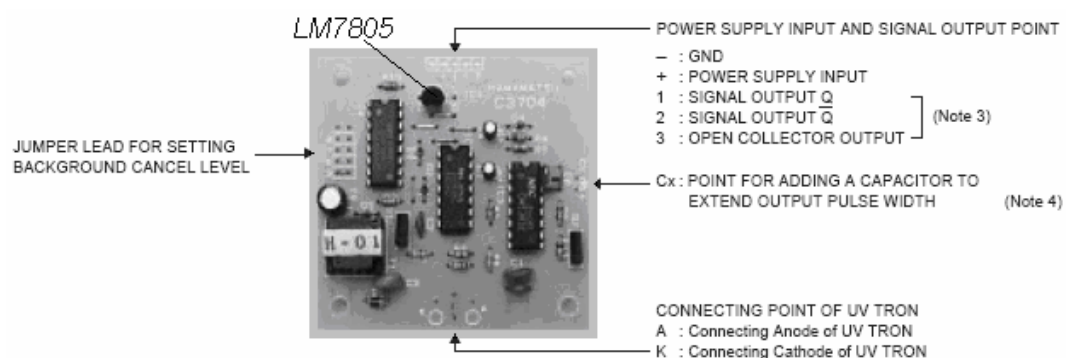
$$VR = 30303,0303 \Omega$$

Dipilih $VR = 50 \text{ K}\Omega$

Untuk mendapatkan *bandwidth* dari frekuensi yang akan dideteksi, maka bagian yang paling berperan adalah $C2$. Dalam rangkaian ini dipilih $C2$ sebesar $0,0047 \mu\text{F}$ (pada *datasheet* tertera besarnya $C2 = 0,005 \mu\text{F}$). Sesuai dengan data yang diambil dari *datasheet*, nilai dari $C3$ adalah $0,02 \mu\text{F}$, maka diambil nilai pendekatan sehingga diberikan nilai kapasitor sebesar $0,022 \mu\text{F}$.

3.2.2.3. Driver Flame Detector

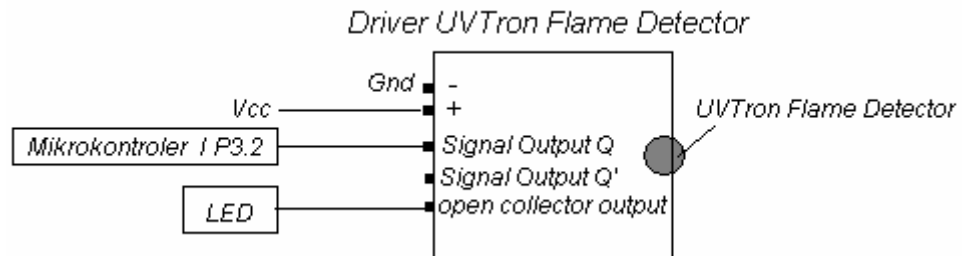
Sesuai dengan *datasheet driver UVTron Flame Detector*, dibutuhkan tegangan sebesar 5V. Karena itu, digunakan IC *regulator 7805* sebagai sumber tegangan pada *driver UVTron Flame Detector* ini. Pada *driver* ini sudah terdapat IC *regulator 7805* yang ditunjukkan pada gambar 3.17 dibawah ini.



Gambar 3.19. Driver UVTron Flame Detector

Dari rangkaian di atas terdapat keterangan *pin input power supply* dan *pin output* dari rangkaian *driver UVTron Flame Detector*. Sehingga dalam

penggunaan rangkaian ini didesain seperti yang tampak pada gambar 3.20 di bawah ini:



Gambar 3.20. Desain Koneksi *UVTron Flame Detector* Dengan Mikrokontroler

Dari desain yang tampak di atas dapat dilihat pada saat sensor menerima sinar *ultraviolet*, maka *output* yang dibaca oleh mikrokontroler adalah *logic high*. Dan sebaliknya jika sensor tidak menerima sinar *ultraviolet*, maka *output* sensor yang dibaca mikrokontroler adalah *logic low*. Hal ini dikarenakan yang terhubung dengan mikrokontroler adalah *pin signal output Q*. Kondisi ini akan berbanding terbalik dengan *output* sensor yang dibaca mikrokontroler, jika *pin signal output Q'* yang terhubung.

3.2.2.4. Photoreflector

Photoreflector adalah rangkaian sensor yang arah pancarannya berupa garis sinar. Adapun sensor yang digunakan dapat berupa infra merah, laser *pointer*, dsb. *Photoreflector* ini akan bernilai *high*, bila sinar pemancar (*transmitter*) tidak diterima oleh penerima (*receiver*) atau mengenai lantai yang berwarna gelap (tidak memantulkan cahaya). Dan sebaliknya jika mengenai lantai yang berwarna terang (yang dapat memantulkan cahaya), maka *Photoreflector* ini akan bernilai *low*.



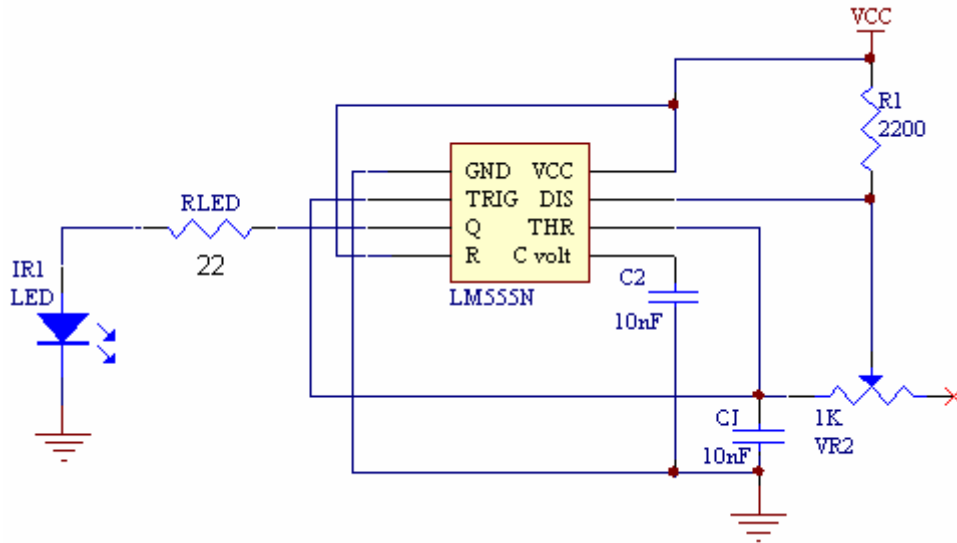
Gambar 3.21. Sensor *Photoreflexor*

Sensor *Photoreflexor* ini terdiri dari dua rangkaian yaitu rangkaian *transmitter* dan rangkaian *receiver*. Rangkaian *transmitter* ini digunakan untuk memancarkan sinar infra merah yang dimodulasi. Rangkaian *receiver* ini digunakan modul *receiver* infra merah yang sudah jadi.

3.2.2.4.1. Rangkaian Pemancar *Photoreflexor*

Rangkaian ini terdiri dari satu buah LED infra merah. Agar jangkauan pancaran dapat mencapai jarak yang cukup jauh, dikarenakan peletakan *photoreflexor* pada robot cukup tinggi, maka sinar infra merah dipancarkan dengan frekuensi yang tinggi. Dalam hal ini LED infra merah dipancarkan dengan frekuensi 40 KHz. Hal ini memudahkan pembuatan di bagian penerima (dengan tidak perlunya penambahan filter untuk frekuensi), karena bisa digunakan modul VISHAY TSOP12XX *Infrared Receiver/Demodulator* (yang sudah termasuk *filter* 40 KHz di dalamnya). Dimana bila *receiver* VISHAY TSOP12XX terkena sinyal 40 KHz, maka *output*-nya berlogika rendah. Sebaliknya bila sinar infra merah terhalang, maka *output* dari VISHAY TSOP12XX akan berlogika tinggi.

Untuk membangkitkan frekuensi sebesar 40 KHz digunakan IC LM555 yang difungsikan sebagai *astable multivibrator*. Alasan digunakan IC LM555 sebagai pembangkit frekuensi 40 KHz karena *photoreflexor* akan bekerja secara terus menerus, sehingga tidak mempengaruhi kinerja dari mikrokontroler untuk mengeksekusi program.



Gambar 3.22. Rangkaian Pemancar *Photoreflector*

Perhitungan pembangkit frekuensi 40 KHz:

$$f = \frac{1,44}{(R1 + 2VR2)C} \quad (3.2)$$

Dipilih $R1 = 2200\Omega$

$C = 10\text{nF}$

$$f = \frac{1,44}{(2200 + 2VR2)C}$$

$$40000 = \frac{1,44}{(2200 + 2VR2) \cdot 10 \cdot 10^{-9}}$$

$$2200 + 2VR2 = 3600$$

$$2VR2 = 3600$$

$$VR2 = 700 \Omega$$

Dipilih $VR2 = 1 \text{ K}\Omega$

Dilakukan pengukuran tegangan pada infra merah dengan avometer, didapat $V_{IR} = 1,05 \text{ Volt}$ Menurut *data book* LED infra merah, LED infra merah jika dipancarkan dengan frekuensi 1 KHz, arus LED dapat mencapai 1000mA. Sehingga diambil $I_{LED} = 90 \text{ mA}$. Pemilihan ini didasarkan bahwa jarak lantai agak jauh dari robot.

$$V_{out} = V_{RLED} + V_{LED} \quad (3.3)$$

$$3,3 = V_{\text{RLED}} + 1,05$$

$$V_{\text{RLED}} = 2,25 \text{ volt}$$

sehingga,

$$V_{\text{RLED}} = I_{\text{LED}} \cdot R_{\text{LED}} \quad (3.4)$$

$$2,25 = 90 \cdot 10^{-3} \cdot R_{\text{LED}}$$

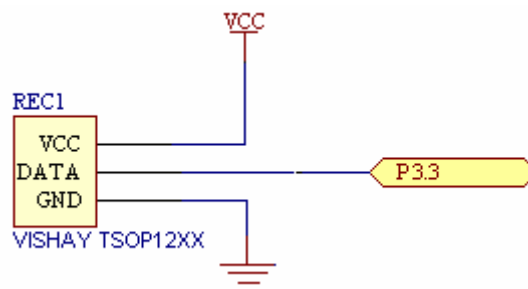
$$R_{\text{LED}} = \frac{2,25}{90 \cdot 10^{-3}}$$

$$R_{\text{LED}} = 25 \Omega$$

Dipilih $R_{\text{LED}} = 22 \Omega$

3.2.2.4.2. Rangkaian Penerima *Photoreflector*

Menggunakan *receiver* dengan merk VISHAY TSOP 12XX dimana pada *receiver* ini telah dilengkapi dengan *filter* 40 KHz, sehingga yang terbaca pada *receiver* ini sudah berupa data asli. Karena yang dipancarkan tidak berupa data, hanya frekuensi 40 KHz terus menerus, maka *output receiver* berlogika rendah.



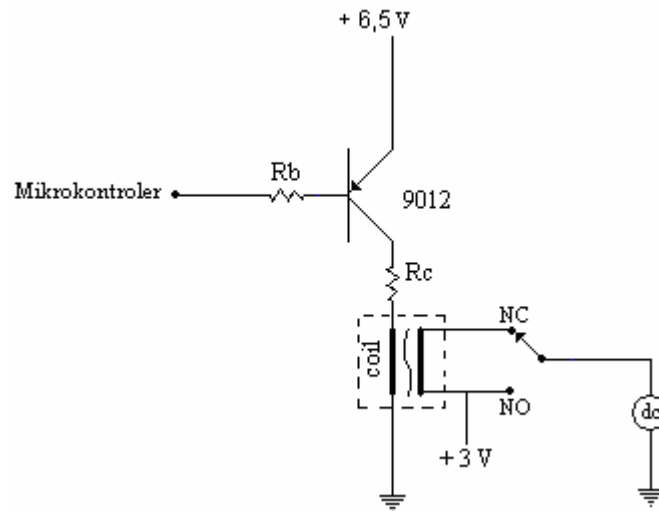
Gambar 3.23. Rangkaian Receiver Infra Merah

Receiver infra merah VISHAY TSOP 12XX pada saat menerima sinar infra merah, *output*-nya berlogika rendah. Sehingga *port* mikrokontroler akan menerima *logic low* (untuk memberikan interupsi pada mikrokontroler). Sedangkan bila *receiver* tidak menerima cahaya infra merah, maka *output receiver* berlogika *high* sehingga *port* mikrokontroler (tidak terjadi interupsi).

3.2.2.5. *Driver* Kipas Pemadam Api

Kipas pemadam api yang digunakan memerlukan sebuah *driver* untuk menjalankannya. *Driver* ini terdiri dari rangkaian transistor dan sebuah relay

untuk menghubungkan antara motor DC dengan tegangan. *Driver* yang dimaksud seperti yang ditunjukkan pada gambar 3.24.



Gambar 3.24. *Driver* Kipas Pemadam Api

Perhitungan rangkaian *driver* kipas pemadam api :

Relay yang digunakan adalah relay 5 V_{DC} dengan arus pada *coil* 0,05A.

maka,

$$\text{Dipilih } h_{fe} = 50$$

$$h_{fe} = \frac{I_c}{I_b} \quad (3.5)$$

$$50 = \frac{0,05}{I_b}$$

$$I_b = \frac{0,05}{50}$$

$$I_b = 1 \text{ mA}$$

sehingga,

$$V_{CC} = V_{R_b} + V_{EB} \quad (3.6)$$

$$6,5 = I_b \cdot R_b + 0,7$$

$$5,8 = 0,001 \cdot R_b$$

$$R_b = \frac{5,8}{0,001}$$

$$R_b = 5800 \Omega$$

$$\text{Dipilih } R_b = 4,7 \text{ K}\Omega$$

Dengan dipilihnya R_b sebesar $4,7 \text{ K}\Omega$, maka didapatkan besarnya I_b adalah:

$$V_{cc} = V_{R_b} + V_{EB} \quad (3.7)$$

$$6,5 = I_b \cdot R_b + 0,7$$

$$5,8 = I_b \cdot 4700$$

$$I_b = \frac{5,8}{4700}$$

$$I_b = 0,0012 \text{ A (mendekati nilai } I_b \text{ yang didapatkan)}$$

Sehingga dengan h_{fe} yang sama I_c didapatkan:

$$h_{fe} = \frac{I_c}{I_b} \quad (3.8)$$

$$50 = \frac{I_c}{0,0012}$$

$$I_c = 0,06 \text{ A (memenuhi)}$$

Dengan nilai I_c yang diambil dari spesifikasi relay, maka:

$$V_{cc} = V_{Relay} + V_{R_c} + V_{EC} \quad (3.9)$$

$$6,5 = 5 + V_{R_c} + 0,2$$

$$V_{R_c} = 1,3$$

$$R_c = \frac{1,3}{0,05}$$

$$R_c = 26 \Omega$$

Dipilih $R_c = 22 \Omega$

Dengan perbandingan I_c yang telah didapat dari besarnya $R_b = 4,7 \text{ K}\Omega$, maka:

$$V_{cc} = V_{Relay} + V_{R_c} + V_{EC} \quad (3.10)$$

$$6,5 = 5 + 0,06 \cdot R_c + 0,2$$

$$R_c = \frac{1,3}{0,06}$$

$$R_c = 21,6667\Omega \text{ (mendekati nilai } 22 \Omega)$$

3.3. Perencanaan Software

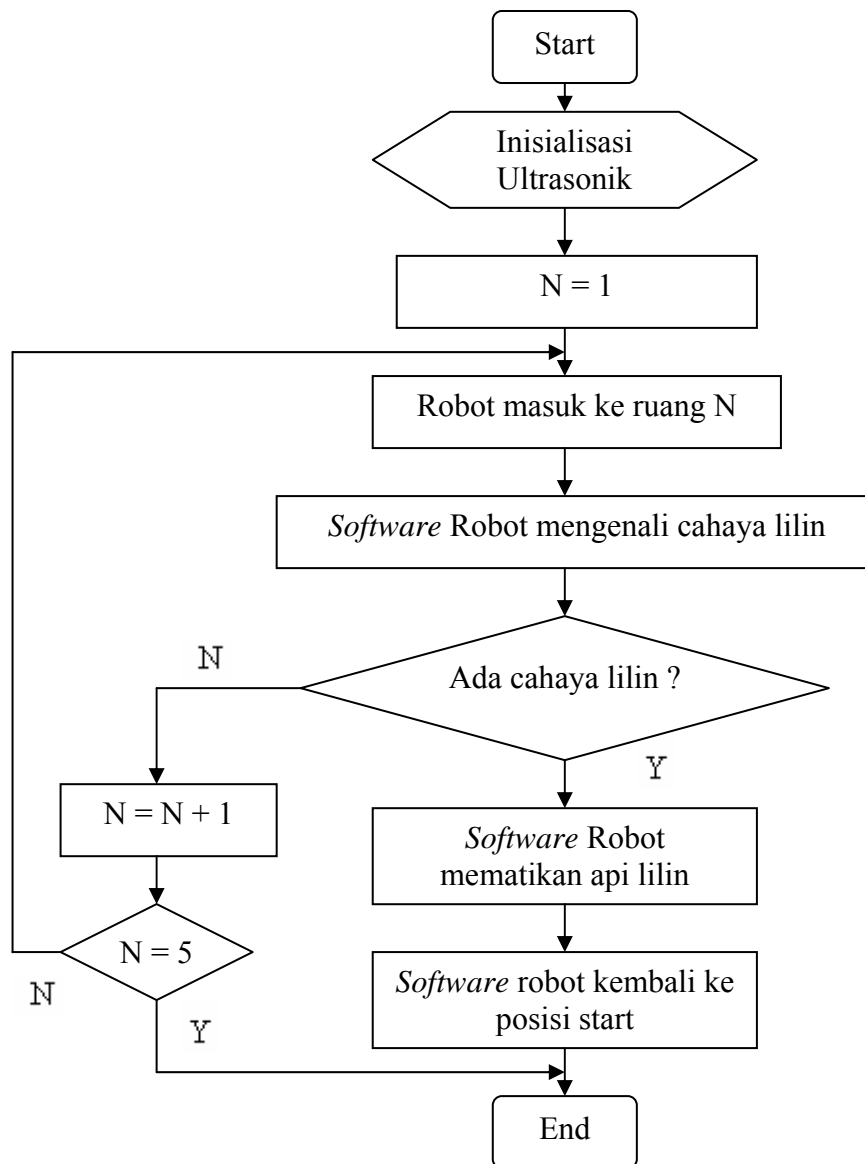
Software yang dibuat untuk memfungsikan robot ini memiliki 4 bagian utama, yaitu:

- Software utama robot melintasi arena
- Software *sub routine* robot mengenali cahaya lilin

- *Software sub routine* robot mematikan api lilin
- *Software sub routine* robot kembali ke posisi *start*

3.3.1. *Software* Utama Robot Melintasi Arena

Pada bagian ini, *software* yang dibuat bertujuan agar robot dapat berjalan melintasi arena tanpa menyentuh dinding di sebelah kanan dan kiri serta untuk memetakan robot memasuki ruangan. Pada bagian ini berhubungan dengan motor servo sebagai penggerak robot dan sensor ultrasonik yang dipakai untuk mendeteksi jarak robot terhadap dinding. *Pin* yang dipakai untuk meng-*handle* servo jalan robot adalah P1.7 (kiri depan), P1.6 (kanan depan), P1.5 (kiri belakang), P1.4 (kanan belakang) terletak pada minimum sistem III. *Port* yang digunakan untuk meng-*handle* PING))) *ultrasonic sensor* adalah P3.7 dan P3.5 pada minimum sistem II.

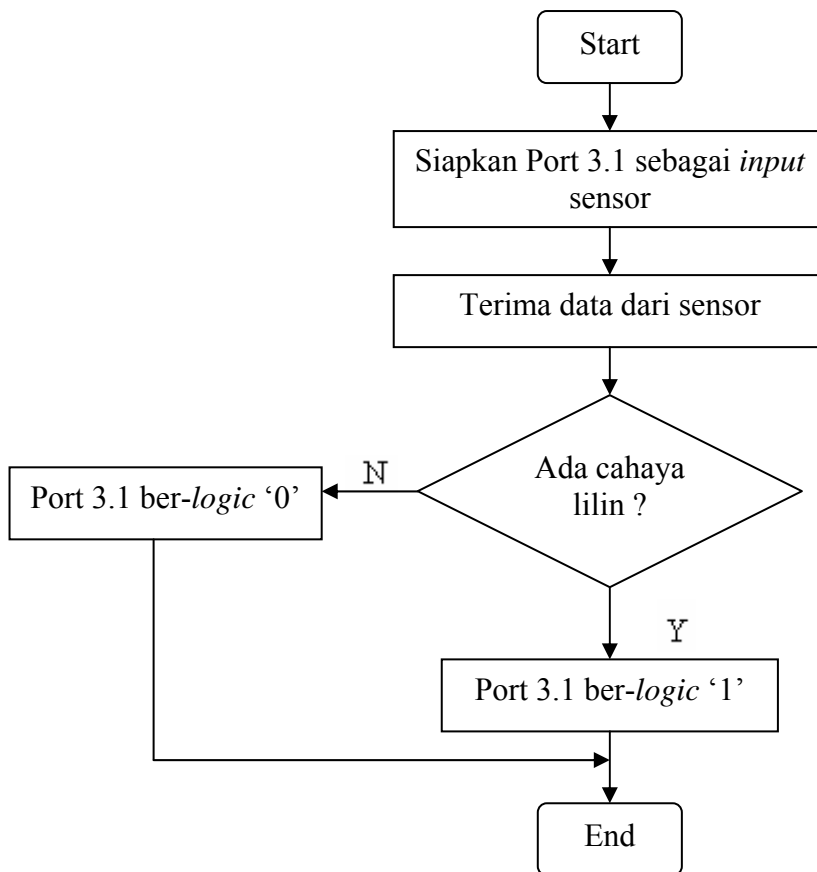


Gambar 3.25. Flowchart Software Utama Robot Melintasi Arena

3.3.2. Software Sub Routine Robot Mengenali Cahaya Lilin

Setiap robot memasuki ruangan, robot masuk pada bagian *software* ini untuk mengenali ada atau tidak cahaya lilin di ruangan tersebut. *Software* ini berhubungan dengan *UVTron Flame Detector* (minimum sistem I). *Port* yang digunakan untuk menerima pulsa dari *UVTron Flame Detector* ini adalah P3.2. Bila *software* mendeteksi adanya pulsa *high* yang menandakan adanya cahaya lilin di ruangan tersebut, maka *software* akan melanjutkannya perintah selanjutnya yaitu memerintahkan robot untuk terus meneruskan perjalanannya dalam ruang tersebut untuk mematikan api lilin. Namun jika robot tidak mendeteksi adanya

cahaya lilin dalam ruangan tersebut, maka robot akan keluar dari ruangan tersebut dan memasuki ruangan yang lain.



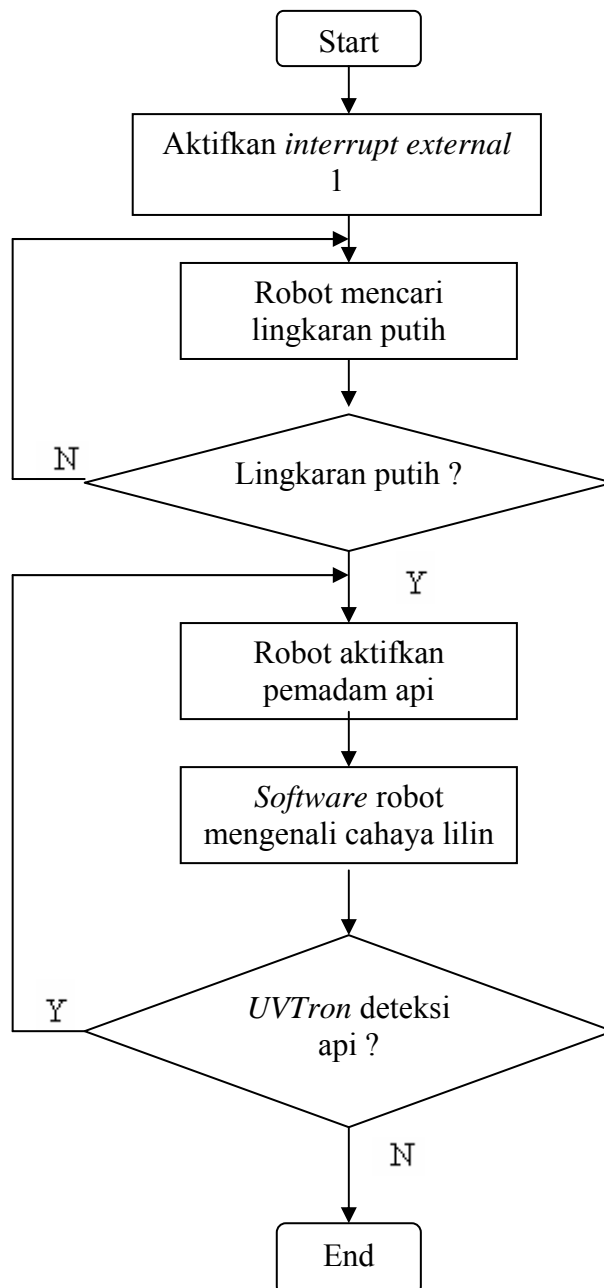
Gambar 3.26. Flowchart Software Sub Routine Robot Mengenali Cahaya Lilin

3.3 3. Software Sub Routine Robot Mematikan Api Lilin

Setelah robot mendeteksi adanya cahaya lilin di salah satu ruangan dalam arena (data dari *UVTron Flame Detector*), maka robot akan mematikan api lilin tersebut. Cara kerja robot mematikan api lilin tersebut adalah sebagai berikut :

Pertama kali robot akan masuk lebih dalam ke ruangan untuk mencari lingkaran putih dengan menggunakan *photoreflector* (minimum sistem I). Port yang digunakan untuk menerima data dari *photoreflector* adalah P3.3, yaitu *pin interrupt external 1*. setelah mikrokontroler menerima pulsa turun (dari '1' ke '0') yang menandakan adanya garis putih, maka program akan menuju alamat *vector* dari *interrupt external 1* yaitu 0013_H. Setelah itu robot akan berhenti dan mulai mengaktifkan prosedur untuk memadamkan api. Prosedur tersebut adalah dengan mengaktifkan kipas DC dan menggerakannya dari arah kanan hingga ke

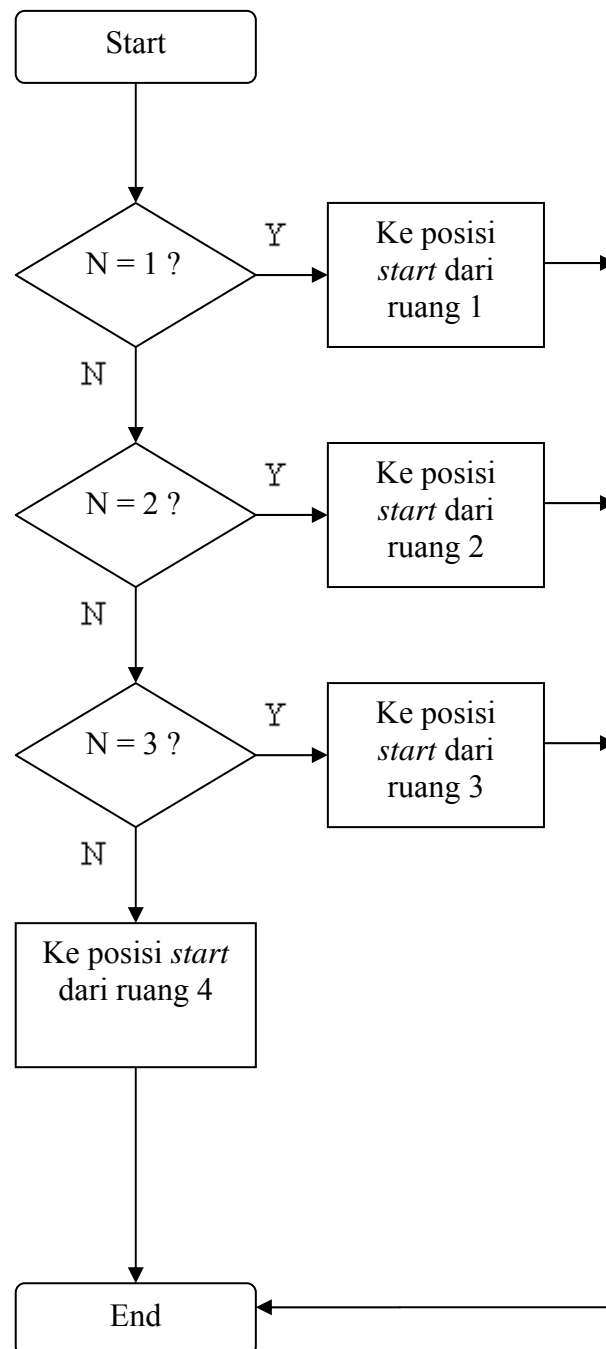
kiri, sesuai dengan program yang dibuat. Untuk menggerakkan kipas DC tersebut digunakan sebuah motor servo. Setelah itu akan dicek dengan *UVTron Flame Detector* apakah lilin sudah mati. Bila lilin belum mati, maka robot akan mengulangi proses semula. *Pin* yang digunakan untuk mengaktifkan motor servo penggerak kipas DC adalah P1.1 (pada minimum sistem III).



Gambar 3.27. *Flowchart Software Sub Routine Robot Mematikan Api Lilin*

3.3.4. *Software Sub Routine Robot Kembali ke Posisi Start*

Setelah robot mematikan api lilin, maka *software* akan mengenali ruang dimana robot berada. Setelah itu robot akan bergerak menuju posisi *start* sesuai dengan denah yang sudah dikenali. Dan robot akan berhenti setelah mendeteksi adanya *home circle* pada posisi *start*.



Gambar 3.28. *Flowchart Software Sub Routine Robot Kembali ke Posisi Start*