

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA**



SKRIPSI

**PERANCANGAN DAN PEMBUATAN
ROBOT *LINE FOLLOWER* DENGAN KENDALI KECEPATAN
MENGUNAKAN METODE *PULSE WIDTH MODULATION*
BERBASIS *MIKROKONTROLLER ATMEGA 16***

Disusun Oleh :

**RIZA PAHLEVI
NIM : 05.12.208**

Maret 2011



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Riza Pahlevi
NIM : 05.12.208
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan Dan Pembuatan Robot *Line Follower*
Dengan Kendali Kecepatan Menggunakan Metode
Pulse Width Modulation Berbasis Mikrokontroler
ATMEGA 16

Dipertahankan dihadapan majelis penguji Skripsi jenjang Strata satu (S-1) pada :

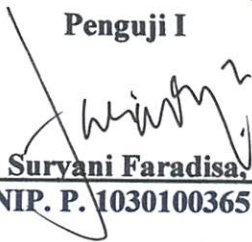
Hari : Sabtu
Tanggal : 21 Agustus 2010
Dengan Nilai : 85,5 (A)

Panitia Ujian Skripsi
Ketua Majelis Penguji

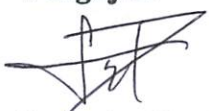
Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Anggota Majelis Penguji

Penguji I


Irmalia Suryani Faradisa, ST., MT
NIP. P. 1030100365

Penguji II


Sonny Prasetyo, ST., MT
NIP. P. 1031000433

**PERANCANGAN DAN PEMBUATAN
ROBOT *LINE FOLLOWER* DENGAN KENDALI KECEPATAN
MENGUNAKAN METODE *PULSE WIDTH MODULATION*
BERBASISKAN *MIKROKONTROLLER ATMEGA 16***

Riza Pahlevi
Konsentrasi Teknik Elektronika
Jurusan Teknik Elektro S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Email : riza.pahlevi@ymail.com

Dosen Pembimbing : I. Ir. Yusuf Ismail Nakhoda, MT
II. Dr. Eng. Aryunto Soetedjo ST., MT

Robot adalah sebuah alat yang dirancang untuk dapat membantu manusia dalam melakukan suatu pekerjaan manusia. Untuk mengenal maupun memperluas ilmu pengetahuan khususnya dalam bidang ilmu robotika, penulis berusaha untuk membuat suatu robot *line follower* dengan kendali kecepatan menggunakan metode *Pulse Width Modulation*

Pada rangkaian sensor garis yang digunakan, setiap masukan sensor mempunyai nilai *PWM* masing-masing, hal ini bertujuan untuk keakuratan pembacaan sensor dan untuk mendapat pergerakan robot yang lebih halus. Pembangkitan pulsa *PWM* didapatkan secara *software* dengan memanfaatkan interupsi *timer* pada *mikrokontroller ATMEGA 16*. Metode *open loop* digunakan untuk pengontrolan kecepatan motor DC

Kata Kunci : *Line Follower, Pulse Width Modulation, ATMEGA 16*

KATA PENGANTAR

Dengan memanjatkan puji syukur kepada Tuhan Yang Maha Esa, atas berkat, rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Perancangan dan Pembuatan Robot *Line Follower* dengan Kendali Kecepatan Menggunakan Metode *Pulse Width Modulation* Berbasis *Mikrocontroller ATMEGA 16*" ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan studi di jurusan Teknik Elektro S-1 konsentrasi Teknik Elektronika Institut Teknologi Nasional Malang dan untuk mencapai gelar sarjana teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan beberapa pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE. Selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Sidik Noertjahjono, MT. Selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Ir. Yusuf Ismail Nakhoda, MT. Selaku Ketua Jurusan Teknik Elektro S-1 dan Dosen Pembimbing I
4. Bapak Dr. Eng. Aryunto Soetedjo ST., MT. Selaku Sekretaris Jurusan dan Dosen Pembimbing II
5. Abi dan Umi serta saudara (Kak JJ, Almh.Kak Eli, Kak Anis, Adikku Taufik), ponakanku tercinta (Si Kak Ros "Ganis" dan Si Ipin "Jalu" yang nyengir), *and special thanks to* "Hoeget Sam", kesemuanya yang telah memberikan doa restu, motivasi serta biaya

6. My Lovely Angel “*Ana Nurfitasari*”, without u i’ m nothing honey
7. Rekan-rekan instruktur Laboratorium Perancangan Elektronika yang telah membantu
8. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini

Penyusun telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan skripsi ini, untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan skripsi ini.

Harapan penyusun semoga laporan ini memberikan manfaat bagi perkembangan ilmu pengetahuan bagi pembaca,

Malang, Maret 2011

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
BERITA ACARA UJIAN SKRIPSI	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan	2
1.5. Metodologi Penulisan	3
1.6. Sistematika Penulisan	3
BAB II LANDASAN TEORI	
2.1. Pendahuluan	5
2.2. <i>Mikrokontroler</i> ATMEGA 16	6
2.2.1. Arsitektur <i>Mikrokontroler</i> ATMEGA 16	6
2.2.2. Konfigurasi <i>Port</i> ATMEGA 16	8
2.2.3. <i>Timer</i>	10

2.3. Catu Daya Sistem	11
2.4. <i>Driver</i> Motor DC	12
2.5. <i>Pulse Width Modulation</i>	13
2.6. LED <i>Infra Red</i>	16
2.7. LED <i>Superbight</i>	18
2.8. <i>Photodioda</i>	19
2.9. Pengkondisi Sinyal	21
2.10. Detektor Cahaya	22
2.11. <i>Smitch Trigger</i>	22

BAB III PERANCANGAN ALAT

3.1. Perancangan Perangkat Keras (<i>Hardware</i>)	24
3.1.1. Perancangan <i>Minimum Sytem</i> ATMEGA 16	27
3.1.1.1. Rangkaian <i>Clock Generator</i>	27
3.1.1.2. Alokasi Penggunaan <i>Port Mikrokontroller</i>	28
3.1.2. Perancangan Rangkaian Pengkondisi Sinyal	29
3.1.3. Perancangan Rangkaian <i>Smitch Trigger</i>	30
3.1.4. Perancangan Rangkaian <i>Push Button</i>	31
3.1.5. Perancangan <i>Driver</i> Motor DC	32
3.1.5.1. Cara Kerja Rangkaian <i>Driver</i> Motor DC	33
3.1.6. Perancangan Rangkaian <i>LCD</i>	34
3.1.7. Perancangan Catu Daya Sistem	35
3.2. Perancangan Perangkat Lunak (<i>Software</i>)	37
3.2.1. Diagram Alir Program <i>Real</i>	36
3.2.2. Diagram Alir Program Simulasi	37

BAB IV PENGUJIAN ALAT

4.1. Pengujian <i>Minimum System</i> ATMEGA 16	38
4.1.1. Tujuan.....	38
4.1.2. Rangkaian Pengujian.....	39
4.1.3. Peralatan Yang Digunakan.....	39
4.1.4. Langkah – Langkah Pengujian.....	39
4.1.5. Hasil Pengujian	40
4.1.5.1. Pengujian Pada <i>Port A</i>	40
4.1.5.2. Pengujian Pada <i>Port B</i>	42
4.1.5.3. Pengujian Pada <i>Port C</i>	43
4.1.5.4. Pengujian Pada <i>Port D</i>	44
4.2. Pengujian Rangkaian <i>Regulator</i> Tegangan	45
4.2.1. Tujuan	45
4.2.2. Rangkaian Pengujian <i>Regulator</i> Tegangan.....	46
4.2.3. Peralatan Yang Digunakan.....	46
4.2.4. Langkah – Langkah Pengujian.....	46
4.2.5. Hasil Pengujian <i>Regulator</i> Tegangan.....	47
4.3. Pengujian Rangkaian Pengkondisi Sinyal.....	48
4.3.1. Tujuan	48
4.3.2. Rangkaian Pengujian Pengkondisi Sinyal	48
4.3.3. Peralatan Yang Digunakan	49
4.3.4. Langkah-Langkah Pengujian	49
4.3.5. Hasil Pengujian Rangkaian Pengkondisi Sinyal	50

4.3.5.1. Hasil Pengujian Rangkaian Pengkondisi Sinyal Sensor <i>Scanning</i>	50
4.3.5.2. Hasil Pengujian Rangkaian Pengkondisi Sinyal Sensor <i>Counting</i>	53
4.3.5.3. Hasil Pengujian Rangkaian Pengkondisi Sinyal Detektor Cahaya	53
4.4. Pengujian Rangkaian <i>Smitch Trigger</i>	53
4.4.1. Tujuan	53
4.4.2. Rangkaian Pengujian <i>Smitch Trigger</i>	54
4.4.3. Peralatan Yang Diperlukan.....	54
4.4.4. Langkah-Langkah Pengujian Rangkaian <i>Smitch Trigger</i>	54
4.4.5. Hasil Pengujian <i>Rangkaian Smitch Trigger</i>	55
4.5. Pengujian Rangkaian <i>Driver Motor DC</i>	55
4.5.1. Tujuan	55
4.5.2. Rangkaian Pengujian <i>Driver Motor DC</i>	56
4.5.3. Peralatan Yang Diperlukan	56
4.5.4. Langkah-Langkah Pengujian Rangkaian <i>Driver Motor</i>	57
4.5.5. Hasil Pengujian Rangkaian <i>Driver Motor DC</i>	57
4.6. Pengujian Rangkaian <i>LCD (Liquid Crstal Display)</i>	58
4.6.1. Tujuan	58
4.6.2. Rangkaian Pengujian <i>LCD</i>	59
4.6.3. Peralatan Yang Diperlukan	59
4.6.4. Langkah-Langkah Pengujian Rangkaian <i>LCD</i>	59
4.6.5. Hasil Pengujian Rangkaian <i>LCD</i>	59

4.7. Pengujian Keseluruhan Sistem.....	60
4.7.1. Hasil Pengukuran Sistem	60
BAB V PENUTUP	
5.1 Kesimpulan.....	63
5.2 Saran.....	64
DAFTAR PUSTAKA	65
LAMPIRAN

DAFTAR GAMBAR

Gambar 2-1 Blok Diagram <i>Mikrokontroler</i> ATMEGA 16	6
Gambar 2-2 Konfigurasi Pin ATMEGA 16	7
Gambar 2-3 Blok Diagram <i>Timer/Counter</i>	11
Gambar 2-4 <i>Transistor</i> LM 7805	12
Gambar 2-5 <i>Transistor</i> TIP 2955, dan TIP 3055	13
Gambar 2-6 <i>Transistor</i> TIP 41	13
Gambar 2-7 Bentuk Gelombang <i>PWM</i>	14
Gambar 2-8 <i>T - ON</i> Dan <i>T - OFF</i> Pada <i>PWM</i>	14
Gambar 2-9 <i>Duty Cycle</i> 10%, 50% dan 90%	15
Gambar 2-10 <i>LED Infrared</i>	17
Gambar 2-11 Simbol dari <i>LED</i> Infra Red	17
Gambar 2-12 <i>Led Superbright</i>	18
Gambar 2-13 Struktur dari <i>Photodiode</i>	19
Gambar 2-14 Bentuk Fisik <i>Photodiode</i>	20
Gambar 2-15 Konfigurasi <i>Pin</i> LM 324	21
Gambar 2-16 <i>IC</i> LM 324	22
Gambar 2-17 Konfigurasi <i>Pin</i> 74LS14	23
Gambar 2-18 <i>Inverter Schmitt Trigger</i> <i>IC</i> 74LS14	23
Gambar 3-1 Blok Diagram Sistem	24
Gambar 3-2 Perancangan <i>Minimum System</i> ATMEGA 16	27
Gambar 3-3 Perancangan Rangkaian <i>Clock Generator</i>	28
Gambar 3-4 Alokasi Penggunaan <i>Port</i> <i>Mikrokontroller</i>	28

Gambar 3-5 Rangkaian Pengkondisi Sinyal	29
Gambar 3-6 Rangkaian <i>Smitch Trigger</i> 74LS14	31
Gambar 3-7 Rangkaian <i>Push Button</i>	31
Gambar 3-8 Rangkaian <i>Driver Motor DC</i>	32
Gambar 3-9 Rangkaian <i>LCD</i>	34
Gambar 3-10 Rangkaian <i>Catu Daya</i>	35
Gambar 3-11 Diagram Alir Program <i>Real</i>	36
Gambar 3-12 Diagram Alir Program <i>Simulasi</i>	37
Gambar 4-1 Rangkaian Pengujian <i>Minimum System ATMEGA 16</i>	39
Gambar 4-2 Tegangan Keluaran <i>Mikrokontroller PortA</i>	
Kondisi <i>High</i>	41
Gambar 4-3 Tegangan Keluaran <i>Mikrokontroller PortA</i>	
Kondisi <i>Low</i>	41
Gambar 4-4 Tegangan Keluaran <i>Mikrokontroller PortB</i>	
Kondisi <i>High</i>	42
Gambar 4-5 Tegangan Keluaran <i>Mikrokontroller PortB</i>	
Kondisi <i>Low</i>	43
Gambar 4-6 Tegangan Keluaran <i>Mikrokontroller PortC</i>	
Kondisi <i>High</i>	43
Gambar 4-7 Tegangan Keluaran <i>Mikrokontroller PortC</i> Kondisi <i>Low</i>	44
Gambar 4-8 Tegangan Keluaran <i>Mikrokontroller PortD</i> Kondisi <i>High</i>	44
Gambar 4-9 Tegangan Keluaran <i>Mikrokontroller PortD</i> Kondisi <i>Low</i>	45
Gambar 4-10 Rangkaian <i>Mikrokontroller ATMEGA 16</i>	45
Gambar 4-11 Rangkaian Pengujian <i>Regulator Tegangan</i>	46

Gambar 4-12 Tegangan Masukan <i>Regulator</i> Tegangan <i>LM 7805</i>	47
Gambar 4-13 Tegangan Keluaran <i>Regulator</i> Tegangan <i>LM 7805</i>	47
Gambar 4-14 Rangkaian <i>Regulator</i> Tegangan <i>LM 7805</i>	48
Gambar 4-15 Rangkaian Pengujian Pengkondisi Sinyal	48
Gambar 4-16 Tegangan Masukan <i>Sensor Scanning</i>	50
Gambar 4-17 V_{OUT} <i>Sensor</i> Saat Terkena Garis	51
Gambar 4-18 V_{OUT} <i>Buffer</i> Saat Terkena Garis	51
Gambar 4-19 V_{OUT} <i>Sensor</i> Saat Terkena Lantai	51
Gambar 4-20 V_{OUT} <i>Buffer</i> Saat Terkena Lantai	52
Gambar 4-21 Rangkaian Pengkondisi Sinyal	52
Gambar 4-22 Rangkaian Susunan <i>Sensor</i>	52
Gambar 4-23 Pengujian Rangkaian <i>Smitch Trigger</i>	54
Gambar 4-24 Pengujian Rangkaian <i>Driver Motor DC</i>	56
Gambar 4-25 Tegangan <i>Enable PWM Digital</i> Dengan <i>Input</i> <i>Duty Cycle 94,12 %</i>	58
Gambar 4-26 Tegangan <i>Enable PWM Digital</i> Dengan <i>Input</i> <i>Duty Cycle 50,19 %</i>	58
Gambar 4-27 Pengujian Rangkaian <i>LCD</i>	59
Gambar 4-28 Pengujian <i>LCD</i>	59
Gambar 4-29 Susunan Pemasangan sensor Garis	60
Gambar 4-30 Tampilan Saat Sensor Paling Kiri Terkena Garis	61
Gambar 4-31 Tampilan Saat Sensor Ke-Dua Dari Kiri Terkena Garis	61
Gambar 4-32 Tampilan Saat Sensor Ke-Tiga Dari Kiri Terkena Garis ..	61

Gambar 4-33 Tampilan Saat Sensor Ke-Tiga Dan Ke-Empat Dari Kiri Terkena
Garis62

Gambar 4-34 Tampilan Saat Sensor Paling Kanan Terkena Garis62

Gambar 4-35 Tampilan Saat Sensor Ke-Dua Dari Kanan Terkena Garis62

DAFTAR TABEL

Tabel 2-1 Fungsi-Fungsi <i>Port</i> ATMEGA 16	10
Tabel 4-1 Hasil Pengujian Pada <i>Port A Minimum System</i>	40
Tabel 4-2 Hasil Pengujian Pada <i>Port B Minimum System</i>	42
Tabel 4-3 Hasil Pengujian Pada <i>Port C Minimum System</i>	43
Tabel 4-4 Hasil Pengujian Pada <i>Port D Minimum System</i>	44
Tabel 4-5 Hasil Pengujian Rangkaian <i>Regulator</i> Tegangan	47
Tabel 4-6 Hasil Pengujian Rangkaian Pengkondisi Sinyal <i>Sensor Scanning</i> ..	50
Tabel 4-7 Hasil Pengujian Rangkaian Pengkondisi Sinyal <i>Sensor Counting</i> ..	53
Tabel 4-8 Hasil Pengujian Rangkaian Pengkondisi Sinyal Detektor Cahaya ..	53
Tabel 4-9 Hasil Pengukuran Tegangan Pada Rangkaian <i>Smitch Trigger</i>	55
Tabel 4-10 Hasil Pengukuran Pada Rangkaian <i>Driver</i> Motor DC.....	57
Tabel 4-11 Hasil Pengamatan Dan Pengukuran Sistem	61

DAFTAR LAMPIRAN

1. *Listing* Program
2. Foto Alat



BAB I

PENDAHULUAN

1.1. Latar Belakang

Robot *Line Follower* diciptakan untuk berbagai tujuan, dapat diaplikasikan layaknya *forklift* otomatis di dunia industri, dan dalam dekade akhir-akhir ini, robot *line follower* dikompetisikan dalam berbagai kontes robot baik nasional maupun dalam kancah internasional.

Robot *line follower* dapat dikendalikan kecepatannya, hal ini bertujuan untuk lebih memudahkan dalam pengendalian dari robot itu sendiri. Salah satu metode dalam pengendalian motor dc, yaitu dengan menggunakan metode *Pulse Width Modulation*, dimana tidak hanya dikenal kondisi 0 dan 1 atau kondisi motor dc mati atau aktif, tetapi juga bisa ditentukan kondisi setengah kecepatan motor dc dan berbagai *range* kecepatan lainnya. Kendali kecepatan menggunakan *Pulse Width Modulation* menerapkan pengaturan lebar pulsanya, dimana dalam kecepatan penuh sama dengan 100% *duty cycle*, setengah kecepatan sama dengan 50% *duty cycle*, dan saat kondisi motor berhenti, maka nilai *duty cycle* 0%.

Dalam *website* <http://insansainsprojects.files.wordpress.com>, dijelaskan bahwa pulsa *PWM* bisa dibangkitkan secara *software* dengan memanfaatkan fungsi *timer/counter* pada fitur yang tersedia di *Mikrokontroler* ATMEGA 16. Kelebihan dari metode pembangkitan dengan cara *software* adalah pulsa *PWM* yang dihasilkan memiliki *ripple* yang lebih sedikit dibandingkan dengan pembangkitan melalui *hardware*.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, permasalahan yang akan dibahas pada laporan akhir ini adalah :

1. Bagaimana merencanakan membuat *hardware* dan *software* dari robot berbasis *mikrokontroller* ATMEGA 16.
2. Bagaimana mengendalikan kecepatan motor DC dengan menggunakan *Metode Pulse Width Modulation*.

1.3. Batasan Masalah

Penulis mengambil batasan masalah sebagai berikut :

1. Pembangkitan pulsa *PWM* digital diperoleh secara program
2. Desain robot beroda.
3. Jalur robot sudah ditentukan
4. Tidak membahas motor DC yang digunakan
5. Kontrol kecepatan dengan metode *open loop*

1.4. Tujuan

Tujuan dari penulisan skripsi ini adalah :

1. Pengontrolan kecepatan motor DC
2. Keakuratan pembacaan sensor garis

1.5. Metodologi Penulisan

Metodologi penulisan yang dipakai dalam pembuatan skripsi ini.

1. Studi Literatur

Pengumpulan bahan sebagai referensi yang berkaitan dengan metode *PWM*

2. Perancangan dan Pembuatan Alat

Perancangan *hardware* dan pemrograman

3. Pelaksanaan uji coba alat

4. Pengambilan dan Analisa Data

5. Menarik Kesimpulan

1.6. Sistematika Penulisan

Penulisan skripsi ini terbagi dalam lima bab dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Berisi latar belakang tujuan, permasalahan, batasan masalah, metodologi, dan sistematika penulisan.

BAB II LANDASAN TEORI

Membahas teori-teori dasar penunjang, perancangan dan pembuatan alat.

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Membahas tentang perancangan alat baik perangkat keras maupun perangkat lunak, serta cara kerja blok diagram.

BAB IV PENGUJIAN ALAT

Mencakup pembahasan tentang proses pengujian alat yang terdiri dari peralatan yang digunakan, langkah kerja dan analisa hasil pengujian.

BAB V PENUTUP

Penulis menarik kesimpulan dari apa yang telah diuraikan pada bab-bab sebelumnya dan mengemukakan saran saran yang mungkin akan bermanfaat bagi laporan akhir ini.

BAB II

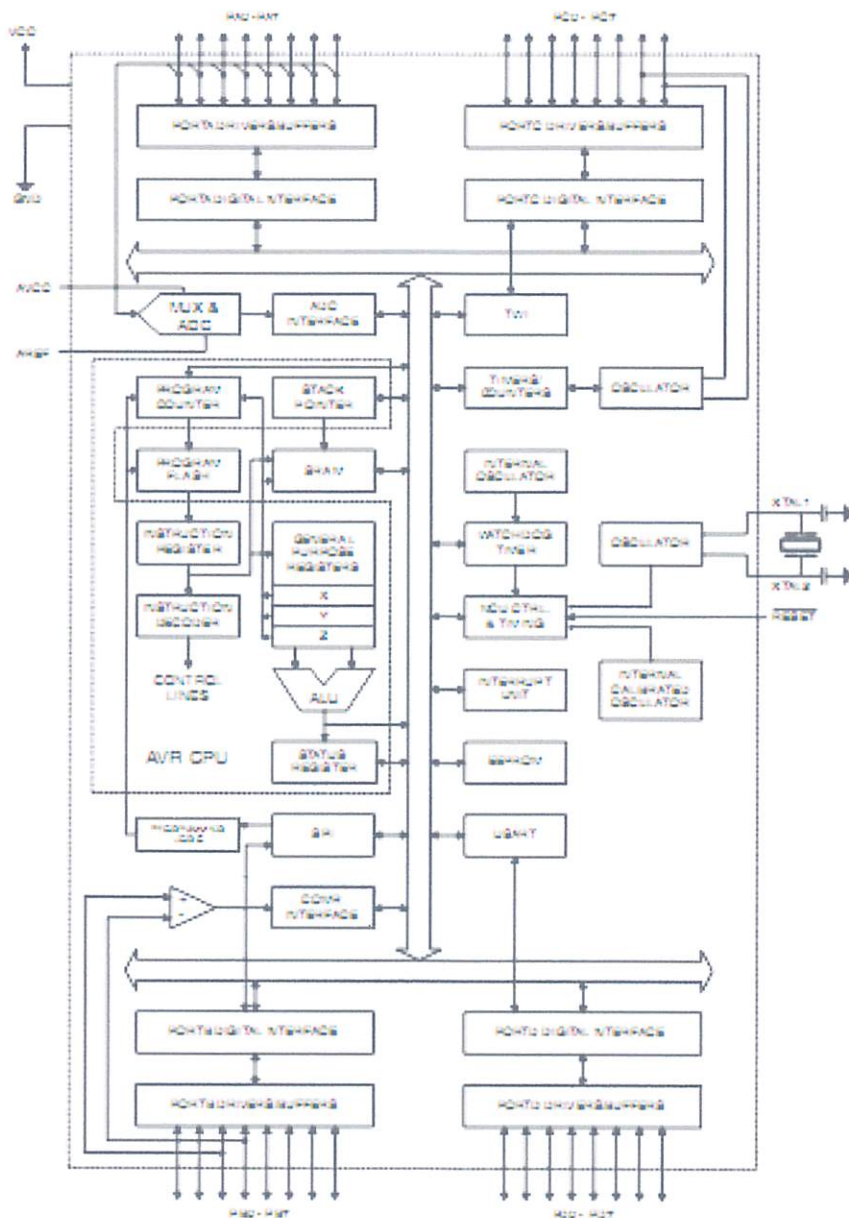
LANDASAN TEORI

2.1. Pendahuluan

Landasan teori sangat membantu untuk dapat memahami suatu sistem, selain daripada itu, dapat juga dijadikan sebagai bahan literatur, teori dasar, serta referensi yang berguna sebagai acuan, rujukan perhitungan dan landasan acuan di dalam merencanakan suatu sistem. Dengan pertimbangan hal-hal tersebut, maka landasan teori merupakan bagian yang harus dipahami untuk pembahasan selanjutnya. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi tentang dasar – dasar teori yang menunjang dalam perencanaan dan pembuatan alat tersebut. Uraian teori dalam bab ini meliputi *Mikrokontroler ATmega 16*, rangkaian catu daya, *driver* motor DC, *Pulse Width Modulation*, *LED Superbright*, *Photodiode*, rangkaian pengkondisi sinyal, rangkaian detektor cahaya, rangkaian *Schmitt Trigger*, dan rangkaian *LCD (Liquid Crystal Display)*

2.2. Mikrokontroler ATMEGA 16

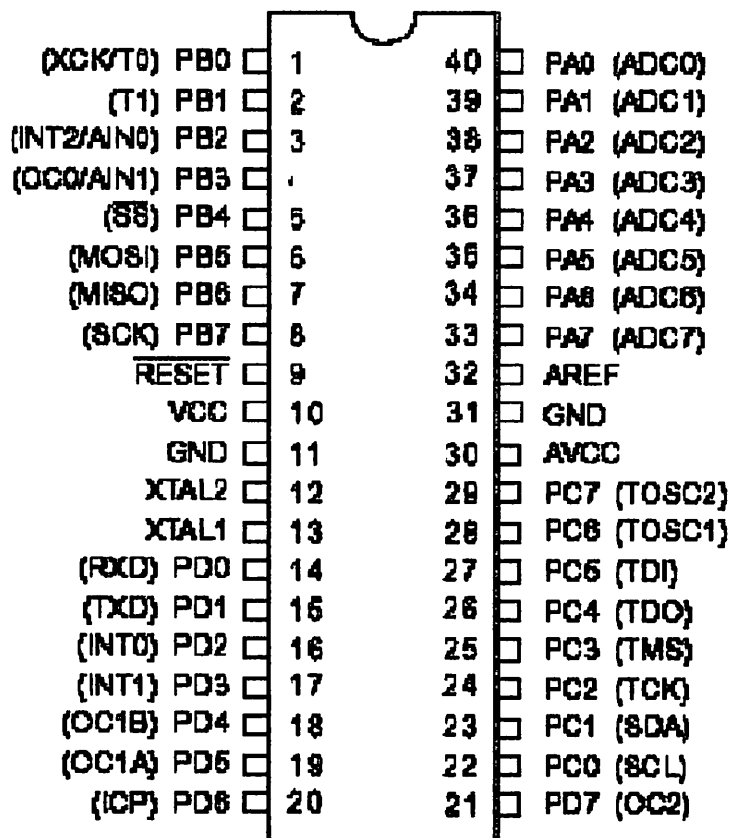
2.2.1. Arsitektur Mikrokontroler ATMEGA 16



Gambar 2-1 Blok Diagram Mikrokontroler ATMEGA 16

ATMEGA 16 (tergolong dalam keluarga AVR) merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register *general-purpose*, timer/counter fleksibel dengan mode *compare*,

interrupt internal dan eksternal, serial UART, programmable Watchdog Timer, dan mode power saving. Mempunyai ADC dan PWM internal. AVR juga mempunyai In-System Programmable Flash on-chip yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. ATMEGA 16 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus clock, Atmega 16 mempunyai throughput mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.



Gambar 2-2 Konfigurasi Pin ATMEGA 16

2.2.2. Konfigurasi Port ATMEGA 16

ATMEGA 16 mempunyai empat buah *port* yang bernama *PortA*, *PortB*, *PortC*, dan *PortD*. Keempat port tersebut merupakan jalur *bi-directional* dengan pilihan internal *pull-up*. Tiap port mempunyai tiga buah *register* bit, yaitu DDxn, PORTxn, dan PINxn. Huruf 'x' mewakili nama huruf dari *port* sedangkan huruf 'n' mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam *register* DDRx (*Data Direction Register*) menentukan arah pin. Bila DDxn diset 1 maka Px berfungsi sebagai pin *output*. Bila DDxn diset 0 maka Px berfungsi sebagai pin *input*. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin *input*, maka *resistor pull-up* akan diaktifkan. Untuk mematikan *resistor pull-up*, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin *output*. Pin port adalah *tri-state* setelah kondisi *reset*. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin *output* maka pin port akan berlogika 1. Dan bila PORTxn diset 0 pada saat pin terkonfigurasi sebagai pin *output* maka pin port akan berlogika 0. Saat mengubah kondisi port dari kondisi *tri-state* (DDxn=0, PORTxn=0) ke kondisi *output high* (DDxn=1, PORTxn=1) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* (DDxn=0, PORTxn=1) atau kondisi *output low* (DDxn=1, PORTxn=0). Biasanya, kondisi *pull-up enabled* dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak

memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah *pull-up*. Jika ini bukan suatu masalah, maka bit *PUD* pada register *SFIOR* dapat diset 1 untuk mematikan semua *pull-up* dalam semua port. Peralihan dari kondisi input dengan *pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama. Kita harus menggunakan kondisi *tri-state* ($DDxn=0$, $PORTxn=0$) atau kondisi *output high* ($DDxn=1$, $PORTxn=0$) sebagai kondisi transisi. Berikut penjelasan dari masing-masing port:

- Port A

Port A menyediakan *input* analog untuk *ADC*, Port A juga bisa digunakan sebagai jalur *bi-directional I/O* jika fitur *ADC* tidak digunakan

- Port B

Port B menyediakan fitur *bi-directional I/O* dan beberapa fungsi pin khusus, antara lain *timer/counter*, komparator analog, dan *SPI*

- Port C

Port C menyediakan fitur *bi-directional I/O* dan beberapa fungsi pin khusus, antara lain : *TWI*, komparator analog, dan *Timer Oscillator*

- Port D

Port D menyediakan fitur *bi-directional I/O* dan beberapa fungsi pin khusus, antara lain : komparator analog, interupsi eksternal, dan komunikasi *serial*

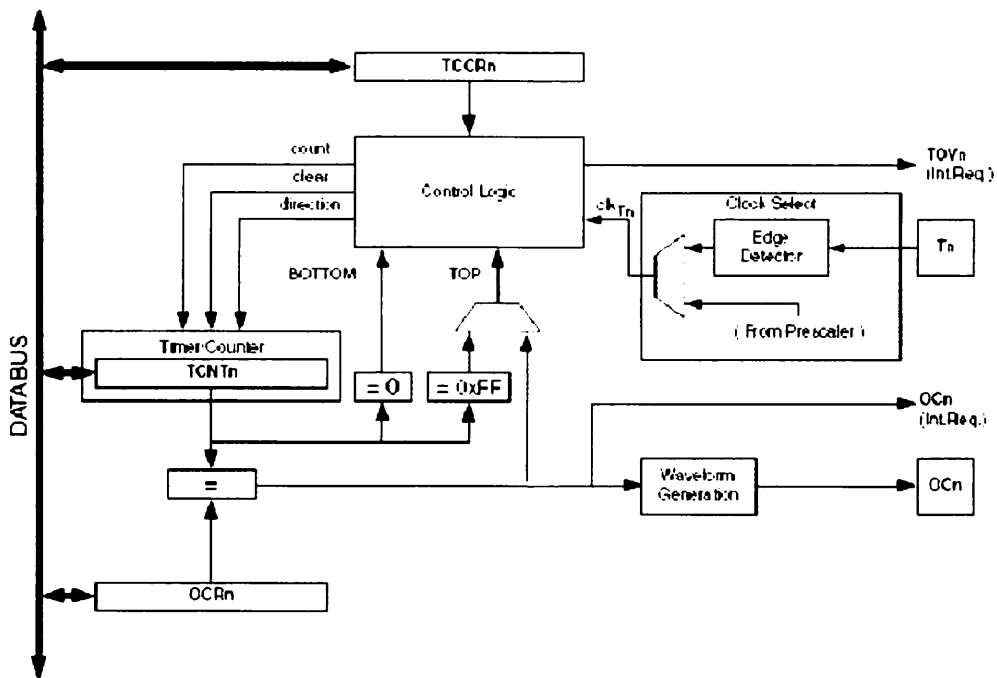
Tabel 2-1 Fungsi-Fungsi *Port* ATMEGA 16

Nama Port	Pin	Fungsi
VCC	10	Pin Catu Daya
GND	11	Pin <i>Ground</i>
XTAL1	13	<i>Input Clock</i> Eksternal
XTAL2	12	<i>Input Clock</i> Eksternal
RESET	9	Mereset <i>Mikrokontroler</i>
AVCC	30	<i>Input</i> Tegangan Untuk <i>ADC</i>
AREF	32	<i>Input</i> Tegangan Referensi
RXD	14	Penerima Komunikasi Serial
TXD	15	Pengirim Komunikasi Serial
OC1B	18	<i>Output PWM Channel B</i>
OC1A	19	<i>Output PWM Channel A</i>

2.2.3. *Timer*

Timer atau *counter* adalah tujuan umum *single channel*, modul 8 bit *timer/counter*. Beberapa fasilitas *channel* dari *timer counter* antara lain:

- *Counter channel* tunggal
- Pengosongan data *timer* sesuai dengan data pembandingan
- Bebas *glitch*, tahap yang tepat untuk *Pulse Width Modulator (PWM)*
- Pembangkit *frekuensi*
- *Event counter external*

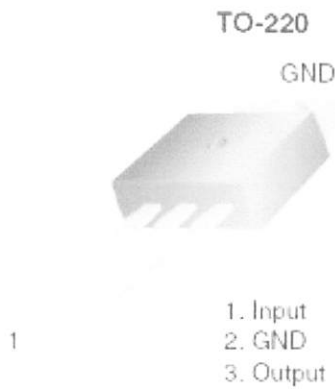


Gambar 2-3 Blok Diagram *Timer/Counter*

2.3. Catu Daya Sistem

Pada sistem yang digunakan, tidak semua rangkaian mempunyai level tegangan *input* catu daya yang sama, dimana untuk sistem dari robot digunakan catu daya sebesar +5 Volt, sedangkan untuk *output* motor digunakan catu daya sebesar +24 Volt. Untuk mendapatkan tegangan +5 Volt dari masukan 12 Volt dari baterai, maka dibutuhkan suatu rangkaian *regulator* tegangan sehingga dapat dihasilkan catu daya untuk sistem robot sebesar +5 Volt.

Untuk me-regulasikan tegangan dari 12 Volt menjadi +5 Volt, digunakan *Transistor* LM 7805.



Gambar 2-4 Transistor LM 7805

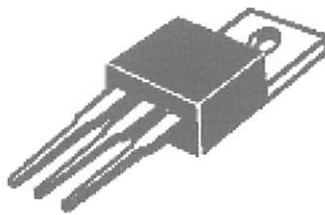
2.4. Driver Motor DC

Untuk menggerakkan motor *DC*, dibutuhkan *driver* motor *DC* yang berfungsi sebagai penguat arus untuk motor yang akan dikendalikan, karena arus yang dihasilkan oleh *mikrokontroller* sangat kecil. *Driver* motor *DC* pada umumnya menggunakan kombinasi rangkaian *H-Bridge* yang terdiri dari kombinasi *transistor* yang berfungsi untuk menentukan polaritas dari motor sehingga motor *DC* bisa bergerak maju atau mundur. Selain *transistor* untuk kombinasi *H-Bridge*, juga dibutuhkan sebuah *transistor* untuk pemicu arus dari inputan *mikrokontroller* sebelum masuk ke dalam rangkaian *H-Bridge*, *transistor* inilah yang akan men *switching transistor* mana yang akan saturasi dan *transistor* mana yang akan *cut-off* sehingga motor bisa dikendalikan secara dua arah

Rangkaian *driver* motor ini menggunakan *Transistor TIP 2955* dan *TIP 3055* yang disusun secara *H-Bridge* dan untuk men-*switching transistor* mana yang saturasi atau transistor mana yang *cut-off*, digunakan *transistor TIP 41*.



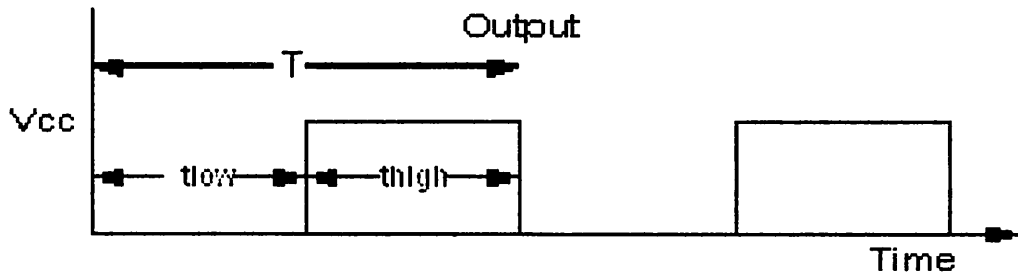
Gambar 2-5 *Transistor* TIP 2955, dan TIP 3055



Gambar 2-6 *Transistor* TIP 41

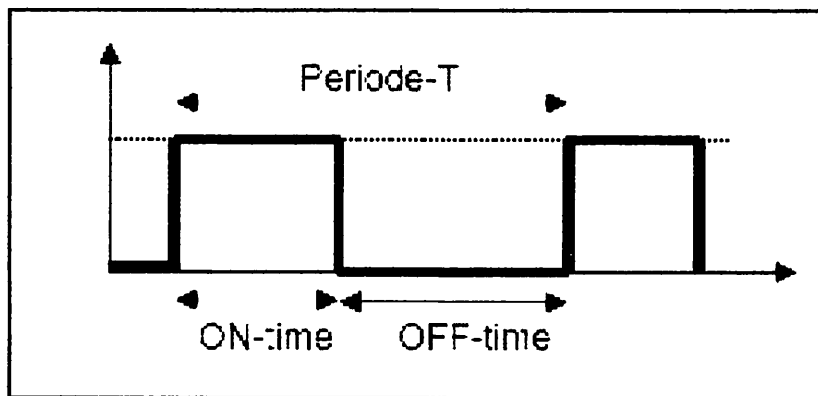
2.5. *Pulse Width Modulation*

PWM (Pulse Width Modulation) merupakan suatu metode untuk mendapatkan sinyal *DC* yang bervariasi yang dikendalikan secara *digital* yang bertujuan untuk mengontrol kecepatan motor. Sinyal *DC* tersebut adalah suatu pulsa dengan *frekuensi* dan *amplitudo* tetap dan lebar pulsa yang dapat diubah – ubah. *PWM* bisa dibangkitkan secara *software* maupun *hardware*. Adapun secara *software* bisa menggunakan *mikrokontroller* dengan memanfaatkan fasilitas interupsi *timer*, sehingga dapat dibentuk gelombang dengan *duty cycle* yang dapat diatur sesuai dengan program. Adapun bentuk gelombang *PWM* dapat dilihat pada Gambar 2-7



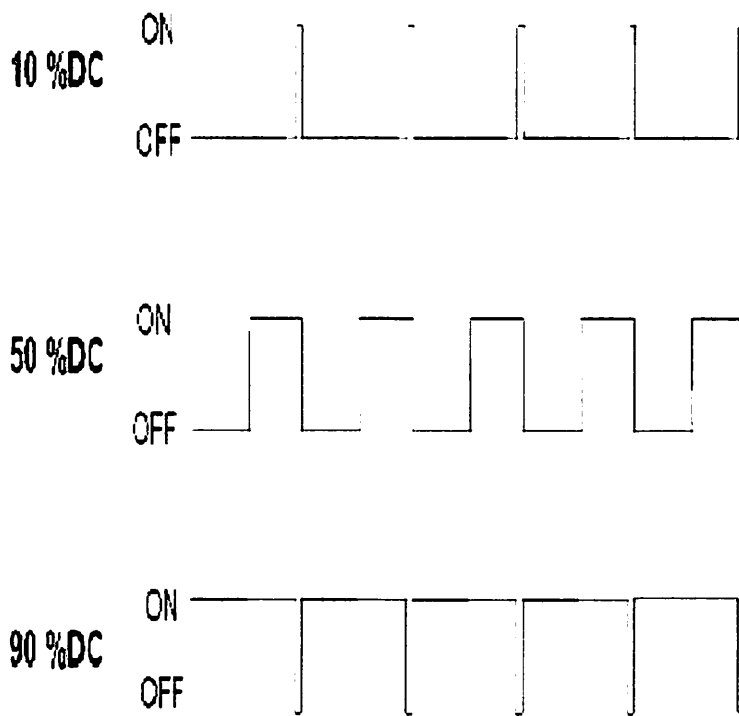
Gambar 2-7 Bentuk Gelombang *PWM*^[1]

PWM pada dasarnya adalah menyalakan (*ON*) dan mematikan (*OFF*) motor DC dengan cepat, Kuncinya adalah mengatur berapa lama waktu *ON* dan *OFF*.



Gambar 2-8 *T - ON* Dan *T - OFF* Pada *PWM*^[1]

Duty Cycle adalah rasio lebar pulsa (dalam satuan waktu) *ON* terhadap periodenya/waktu total (waktu total = *ON* + *OFF*). *Duty cycle* 10% artinya lebar pulsa *high* adalah 10% dari periode sinyal kotak, begitu juga 50% dan 90%.



Gambar 2-9 Duty Cycle 10%, 50% dan 90% ^[1]

Variasi *duty cycle* ini memberikan harga tegangan rata – rata yang berbeda-beda. Sinyal *PWM* dengan *duty cycle* yang besar memiliki harga rata-rata yang lebih besar dibandingkan yang memiliki *duty cycle* kecil. Nilai tegangan yang diberikan sebanding dengan nilai *duty cycle* yang diberikan. Adapun rumus *duty cycle*, yaitu :

$$\text{duty cycle} = t \text{ high} / (t \text{ high} + t \text{ low}) \times 100\%$$

Metode *Pulse Width Modulation* (PWM) merupakan metode yang cukup efektif untuk mengendalikan kecepatan motor *DC*. *PWM* ini bekerja dengan cara membuat gelombang persegi yang memiliki perbandingan pulsa *high* terhadap pulsa *low* yang telah ditentukan, biasanya diskalakan dari 0 hingga 100% seperti pada Gambar Gelombang persegi ini memiliki *frekuensi* tetap (biasanya *maximum* 10 *KHz*) namun lebar pulsa *high* dan *low* dalam 1

periode yang akan diatur. Perbandingan pulsa *high* terhadap *low* ini akan menentukan jumlah daya yang diberikan ke motor *DC*.

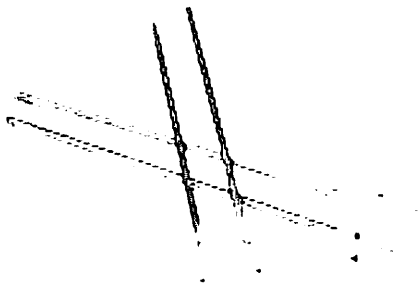
Penggunaan sinyal *PWM* untuk mengendalikan kecepatan motor *DC* sudah umum dilakukan. Salah satu faktor penting dalam penggunaan sinyal ini adalah besarnya *frekuensi* yang dipilih. Membangkitkan sinyal dengan *frekuensi* rendah lebih mudah dibandingkan *frekuensi* tinggi. Tetapi sinyal *PWM* dengan *frekuensi* rendah tidak memberikan hasil yang memuaskan jika diaplikasikan pada motor *DC*. Putaran motor *DC* pada sinyal *PWM* frekuensi rendah akan tersendat-sendat.

2.6. LED *Infra Red*

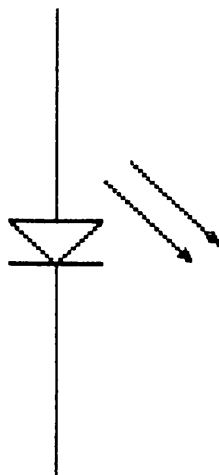
Spektrum cahaya termasuk gelombang *elektromagnetik* dan dapat di kelompokkan dalam beberapa macam, yaitu cahaya tampak dan cahaya tak tampak. *Infra red* termasuk golongan cahaya tak tampak oleh mata kita, memiliki panjang gelombang kurang lebih $0,7-100 \times 10^{-6}$ m. Gelombang elektromagnetik dapat dibagi dalam beberapa daerah, yaitu gelombang radio, sinar x, sinar *gama*, sinar *infra red*, dan sinar *ultraviolet*.

Sifat dari spektrum *infra red* tidak berbeda jauh dengan cahaya tampak, di antaranya yaitu dapat memancarkan cahaya yang merupakan satu garis lurus dan dapat menembus benda – benda transparan serta dapat dipantulkan oleh benda yang memantulkan cahaya, misalnya cermin. Spektrum *infra red* banyak digunakan untuk keperluan *elektromagnetik*, terutama untuk alat yang menggunakan *remote control*.

Cahaya *infra red* dapat dibangkitkan melalui *difusi* pada *dioda semi konduktor* yang disebut *LED*. Sedangkan *dioda* sendiri juga banyak jenisnya termasuk yang bisa memancarkan cahaya saat dialiri arus *forward* atau arus masuk *elektron* dari pita *konduksi* yang melewati *junction* dan jatuh ke dalam *hole* pita *valensi*, sehingga *elektron* memancarkan energi. Pada *dioda* energi di pancarkan melalui panas dan *dioda* yang tidak memancarkan cahaya contohnya *dioda zener* maupun *dioda* biasa. Lambang dari *LED Infra Red* dapat dilihat pada gambar di bawah ini :



Gambar 2-10 *LED Infrared* ^[2]



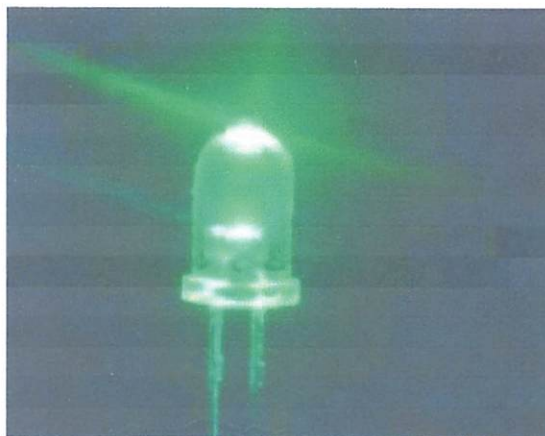
Gambar 2-11 Simbol dari *LED Infra Red* ^[2]

Keuntungan dalam menggunakan *Spektrum* Infra Red adalah sebagai berikut:

- *Infra Red* mempunyai panjang gelombang yang kecil pada daerah *frekuensi* tertentu dengan panjang gelombang 7 – 100 ($\bar{6}$) m.
- Pemancar *Infra Red* membutuhkan sumber tegangan yang kecil.
- Tidak mengganggu pemakaian udara untuk kepentingan umum, sehingga tidak memerlukan ijin khusus.
- *Spektrum Infra Red* merupakan cahaya yang tidak tampak, sehingga dapat digunakan untuk keperluan rahasia.

2.7. LED Superbright

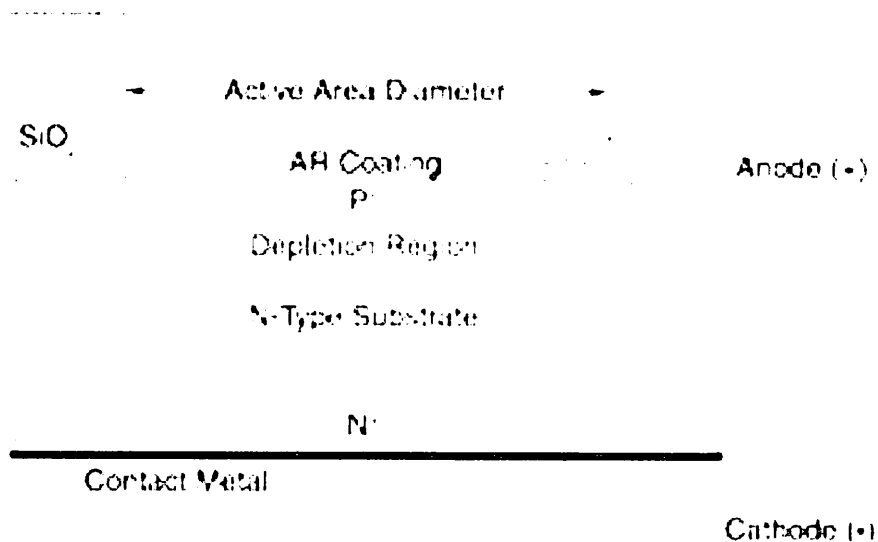
LED Superbright berfungsi sama halnya dengan *Infrared*, bedanya dalam hal ini *Infrared* merupakan cahaya tak tampak, sedangkan *Led Superbright* merupakan cahaya yang tampak. *Led Superbright* berfungsi sebagai pengirim cahaya ke garis untuk dibaca sensor. Kerjanya ketika sumber tegangan masuk pada *battery on*, maka arus masuk sehingga *led superbright* menyala dengan terang yang kemudian dibiaskan pada *photodiode*.



Gambar 2-12 *Led Superbright* ^[2]

2.8. Photodiode

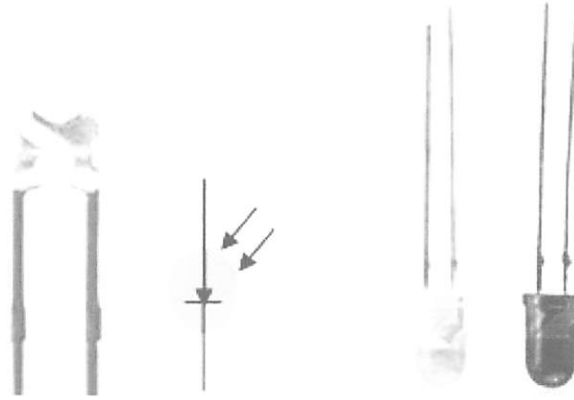
Photodiode mempunyai daerah *depleksi* dengan medan listrik yang tinggi yang menyebabkan pemisahan pasangan *elektron hole* secara *photogeneration*. Untuk operasi kerja yang tinggi, daerah *depleksi* harus dijaga ketipisannya untuk mereduksi waktu transit. Dengan kata lain peningkatan *efisiensi kuantum* (jumlah pasangan *elektron* dihasilkan per-*photon* yang timbul). Lapisan *depleksi* harus cukup tebal untuk menyerap sebagian besar cahaya. Gambar di bawah ini menunjukkan struktur dari *photodiode* :



Gambar 2-13 Struktur dari *Photodiode* ^[2]

Photodiode dapat dioperasikan *photovoltaic*, *photodiode* dalam keadaan tidak terbias dan terhubung pada impedansi beban sama seperti *solarsel* yang akan diuraikan. Selanjutnya bagaimanapun juga *desain* piranti ini berbeda secara dasarnya. Pada *photodiode* hanya *range* panjang gelombang yang sempit terpusat pada panjang gelombang sinyal optik merupakan hal yang penting.

Bentuk fisik *photodioda* adalah seperti ditunjukkan pada gambar di bawah ini yang mempunyai jenis warna yang bermacam – macam. Berikut ini ditunjukkan *photodioda* yang memilih warna gelap dan warna terang :



Gambar 2-14 Bentuk Fisik *Photodioda*^[2]

Photodioda sulit untuk memperkecil *kapasitansi junction* dan satu hal yang terpenting dari pada *photodioda* adalah *efisiensi kuantum*. P-I-N *photodioda* adalah salah satu *photodetector* yang paling umum oleh karena daerah *depleksi* yang lebar yang dapat mengoptimalkan *efisiensi kuantum* dan respon frekuensinya.

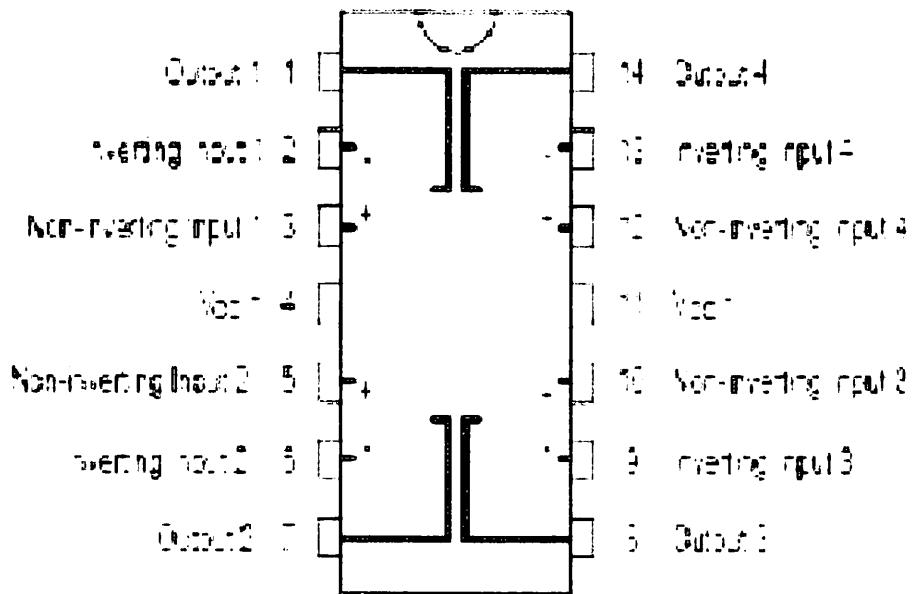
Penyerapan cahaya pada *semikonduktor* menghasilkan pasangan *elektron-hole*, pasangan *elektron-hole* menghasilkan daerah *depleksi* atau panjang *difusi* dari pasangan *elektron hole* akan dipisahkan oleh medan listrik dimana arus mengalir kedalam rangkaian *external* sebagai *carieer drift* melalui lapisan *depleksi*.

Ketebalan lapisan yang lebar dioptimalisasikan untuk panjang gelombang sinyal optik dan *frekuensi* modulasi.

2.9. Pengkondisi Sinyal

Pengkondisi sinyal adalah suatu rangkaian yang membandingkan tegangan *input* dari rangkaian *photodiode* untuk membedakan gelap terang jalur robot sehingga robot bisa bergerak mengikuti garis. Pada rangkaian ini digunakan *IC LM 324*

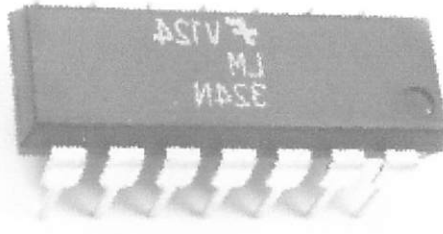
IC LM324 merupakan *IC Operational Amplifier*, *IC* ini mempunyai 4 buah *op-amp* yang berfungsi sebagai *komparator*. *IC* ini mempunyai tegangan kerja antara +5 V sampai +15V untuk +Vcc dan -5V sampai -15V untuk -Vcc. Adapun definisi dari masing-masing pin *IC LM 324* adalah sebagai berikut :



Gambar 2-15 Konfigurasi Pin LM 324

2.10. Rangkaian *Detektor Cahaya*

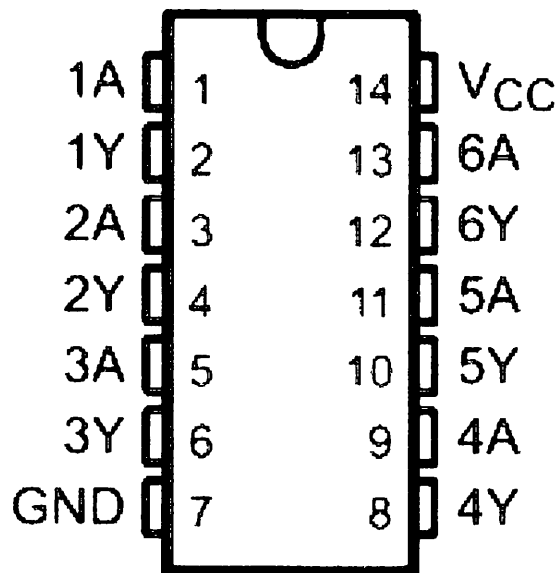
Untuk mendeteksi cahaya, maka digunakan suatu rangkaian *op-amp* (seperti yang digunakan pada sensor garis), namun dalam hal ini difungsikan sebagai pendeteksi cahaya, dimana intensitas cahaya yang dihasilkan api lebih besar daripada garis yang dilalui oleh robot, sehingga level tegangan yang dihasilkan lebih besar daripada garis arena, jauh diatas tegangan *referensi* yang ditentukan. *IC komparator* yang digunakan menggunakan *LM 324*.



Gambar 2-16 *IC LM 324*

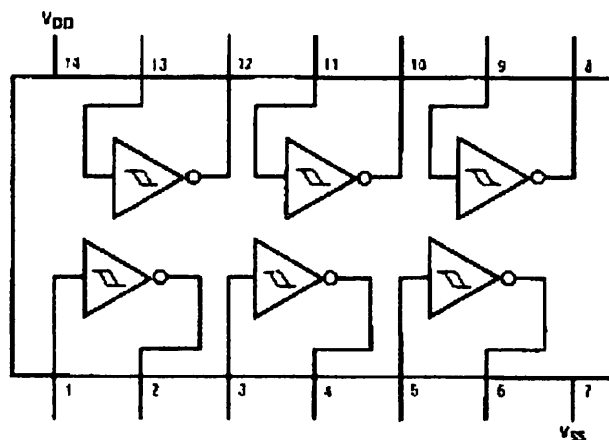
2.11. *Smitch Trigger*

IC 74LS14 merupakan *IC TTL* yang berfungsi sebagai penguat tegangan dan menegaskan masukan sinyal yang berupa sinyal bergerigi atau *sinus* menjadi sinyal kotak yang tegas. *IC* ini mempunyai 6 gerbang *inverter* yang masing-masing dengan *input* Schmitt trigger. Gambar konfigurasi pin-pinnya adalah sebagai berikut:



Gambar 2-17 Konfigurasi Pin 74LS14

Akibat adanya *inversi*, maka *histeresis IC TTL 74LS14* menjadi terbalik. Bila masukkan melampaui 1,9V maka keluaran berpindah ke keadaan rendah. Bila masukkan kurang dari 1V maka keluaran berpindah kembali ke keadaan tinggi, dengan kata lain, keluaran dari *inverter schmitt trigger* ialah kebalikan dari masukkannya. *Schmitt trigger* dengan gerbang *NOT* yang dikemas dalam bentuk *IC TTL 74LS14* diperlihatkan dalam Gambar 2-18



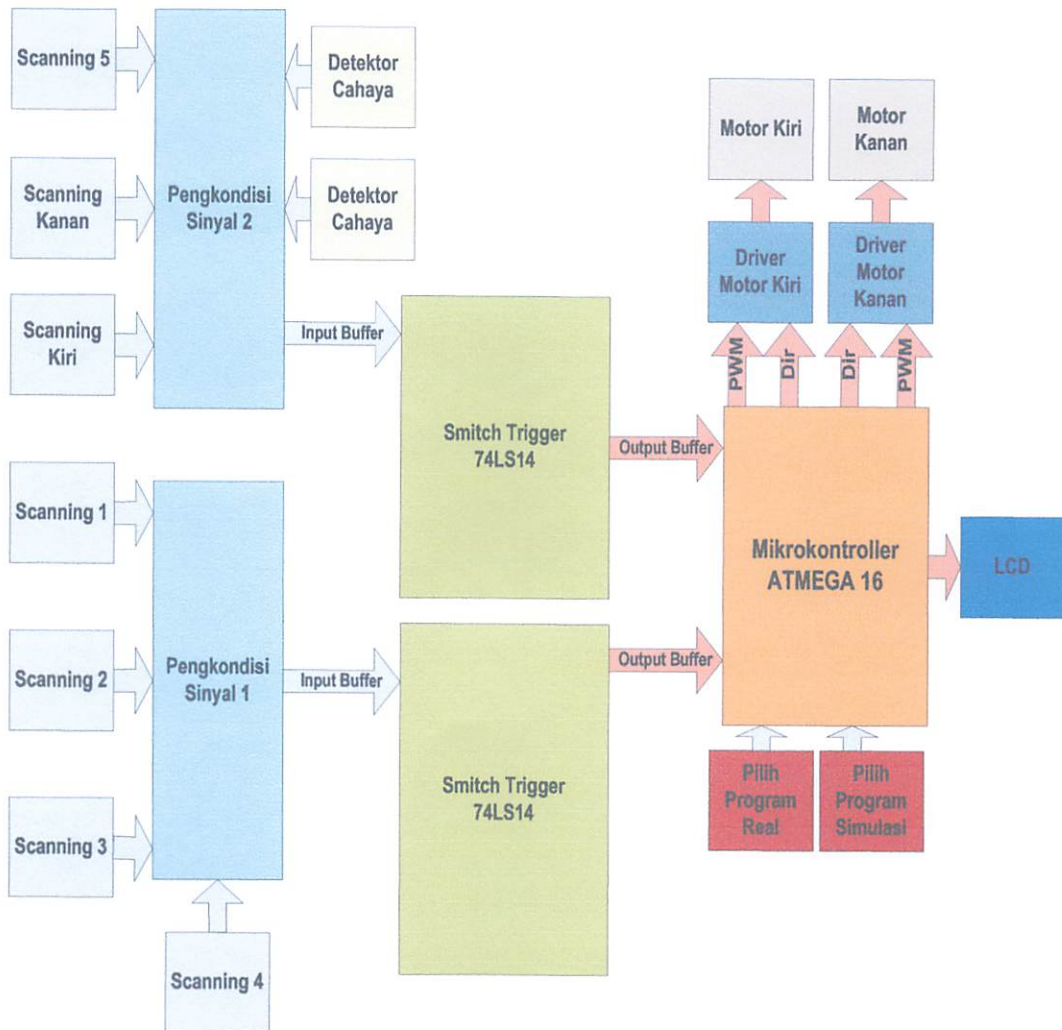
Gambar 2-18 Inverter Schmitt Trigger IC 74LS14

BAB III

PERANCANGAN ALAT

Pada bab ini akan dibahas mengenai perancangan dan pembuatan perangkat keras (*Hardware*) dan perangkat lunak (*software*) yang berfungsi sebagai sistem dari robot itu sendiri

3.1. Perancangan Perangkat Keras (*Hardware*)



Gambar 3-1 Blok Diagram Sistem

Keterangan Blok Diagram Sistem :

➤ *Sensor Scanning I*

Sensor yang berfungsi untuk mendeteksi garis putih, dan diletakkan pada ujung paling kiri rangkaian sensor.

➤ *Sensor Scanning II*

Sensor yang berfungsi untuk mendeteksi garis putih, dan diletakkan pada urutan ke-dua paling kiri rangkaian sensor.

➤ *Sensor Scanning III*

Sensor yang berfungsi untuk mendeteksi garis putih, dan diletakkan pada urutan ke-tiga paling kiri rangkaian sensor.

➤ *Sensor Scanning IV*

Sensor yang berfungsi untuk mendeteksi garis putih, dan diletakkan pada urutan ke-empat paling kiri rangkaian sensor.

➤ *Sensor Scanning V*

Sensor yang berfungsi untuk mendeteksi garis putih, dan diletakkan pada urutan ke-lima paling kiri rangkaian sensor.

➤ *Sensor Scanning Kanan*

Sensor yang berfungsi untuk mendeteksi garis putih, dan diletakkan pada ujung kanan bagian depan robot, sensor ini berfungsi untuk indikator robot untuk berbelok

➤ *Sensor Scanning Kiri*

Sensor yang berfungsi untuk mendeteksi garis putih, dan diletakkan pada ujung kiri bagian depan robot, sensor ini berfungsi untuk indikator robot untuk berbelok

➤ **Detektor Cahaya**

Terdapat 2 buah detektor cahaya, 1 buah berfungsi untuk indikator robot melakukan start, dan satu buah sebagai indikator robot untuk berhenti

➤ **Pengkondisi Sinyal**

Rangkaian ini bertujuan untuk membandingkan tegangan referensi dengan tegangan masukan dari sensor garis

➤ **Smitch Trigger 74LS14**

Rangkaian ini berfungsi untuk membalik *logic* yang dihasilkan oleh pengkondisi sinyal.

➤ **Tombol Pilih Program Real**

Tombol untuk menjalankan program utama

➤ **Tombol Pilih Program Simulasi**

Tombol untuk menjalankan program simulasi

➤ **Mikrokontroller ATMEGA 16**

Berfungsi sebagai pengontrol utama dari sistem robot

➤ **Driver Motor Kanan**

Berfungsi sebagai penguat arus untuk menggerakkan motor kanan sebagai penggerak roda robot bagian kanan

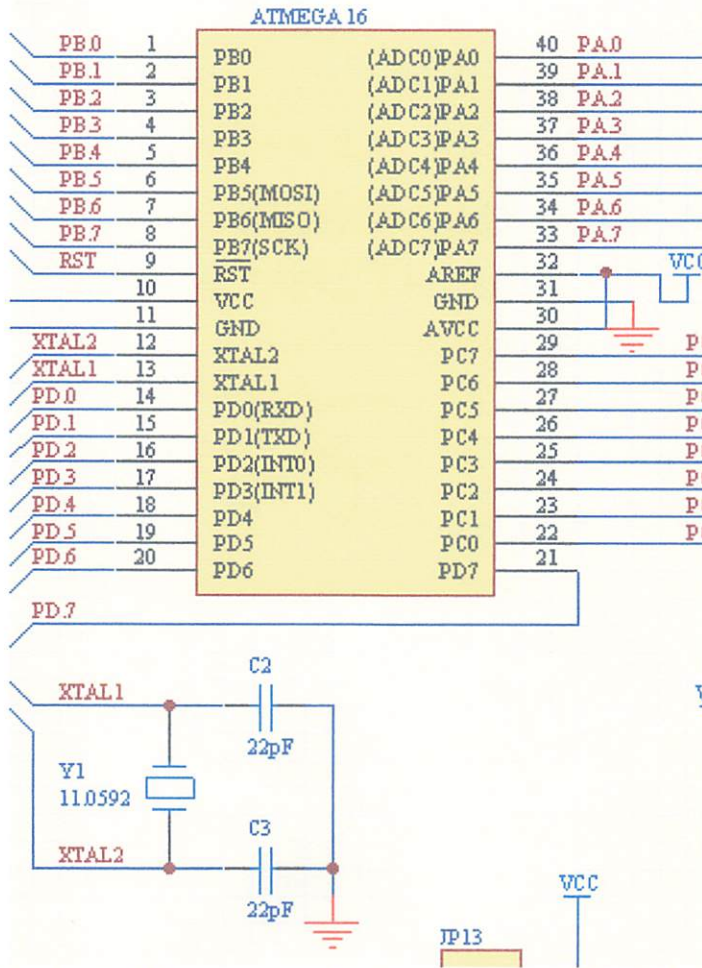
➤ **Driver Motor Kiri**

Berfungsi sebagai penguat arus untuk menggerakkan motor kiri sebagai penggerak roda robot bagian kiri

➤ **LCD**

Sebagai penampil untuk simulasi sistem

3.1.1. Perancangan *Minimum Sytem* ATMEGA 16

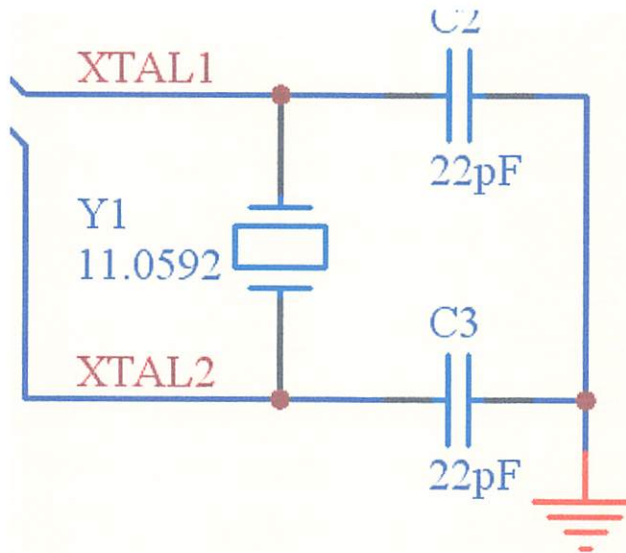


Gambar 3-2 Perancangan *Minimum System* ATMEGA 16

3.1.1.1. Rangkaian *Clock Generator*

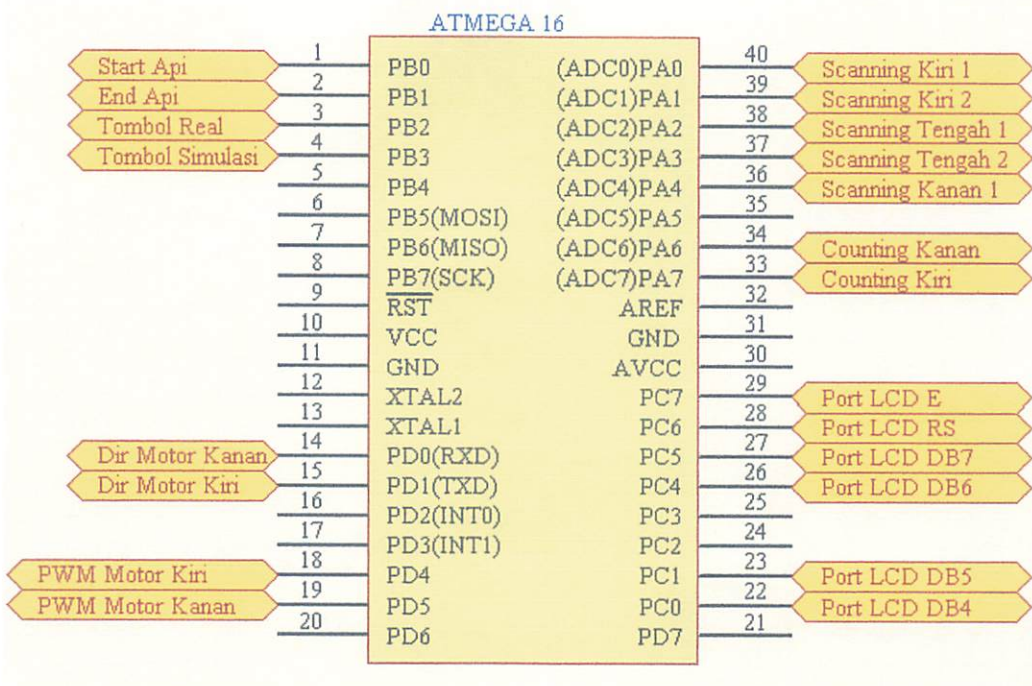
Mikrokontroller ATmega16 memiliki *osilator internal (on chip oscillator)* yang dapat digunakan sebagai sumber *clock* bagi CPU. Untuk menggunakan *osilator internal* diperlukan sebuah kristal antara pin *XTAL1* dan *XTAL2* serta kapasitor ke *ground* seperti Gambar 3.3. Untuk kristal dari *minimum system* ini menggunakan frekuensi 11.0592 MHz,

sedangkan untuk kapasitor menggunakan 22 pF sampai 22 pF. Nilai kapasitor ini diperoleh dari tabel *datasheet* tentang penggunaan kapasitor untuk rangkaian *osilator/sistem clock* pada ATmega 16.



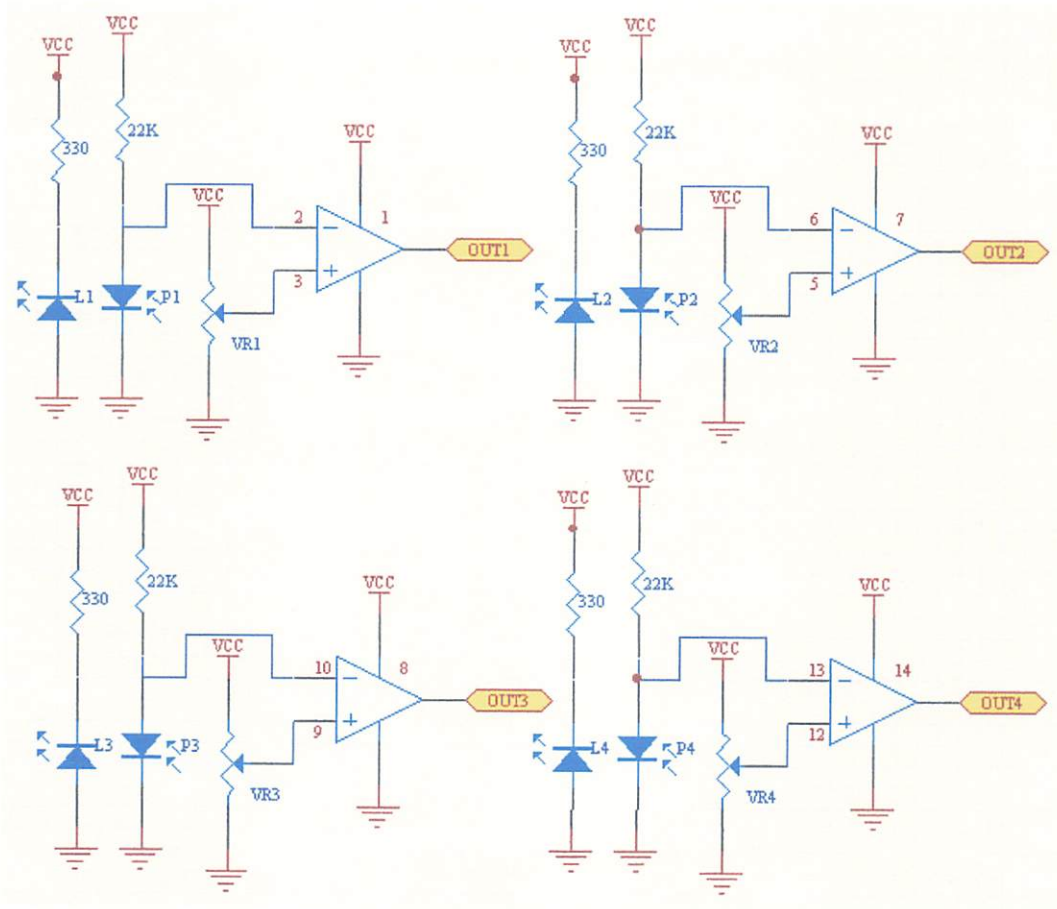
Gambar 3-3 Perancangan Rangkaian *Clock Generator*

3.1.1.2. Alokasi Penggunaan *Port Mikrokontroller ATMEGA 16*



Gambar 3-4 Alokasi Penggunaan *Port Mikrokontroller*

3.1.2. Perancangan Rangkaian Pengkondisi Sinyal



Gambar 3-5 Rangkaian Pengkondisi Sinyal

Pada perancangan rangkaian pengkondisi sinyal, sinyal dikuatkan sebesar 5,5 kali. Untuk menghasilkan penguatan sebesar 5,5 kali, digunakan rumus : $(R2/R1) + 1$, dimana $R2 = 100K$, dan $R1 = 22K$.

Cara kerja dari rangkaian pengkondisi sinyal ini adalah LED akan memancarkan cahaya ke lantai dan *photodiode* akan menerima cahaya yang dipantulkan oleh lantai tersebut. Intensitas cahaya yang diterima oleh *photodiode* akan mempengaruhi nilai *resistansi*-nya. Lantai dan garis memantulkan cahaya dengan intensitas yang berbeda. Garis akan memantulkan cahaya dengan intensitas yang lebih tinggi daripada

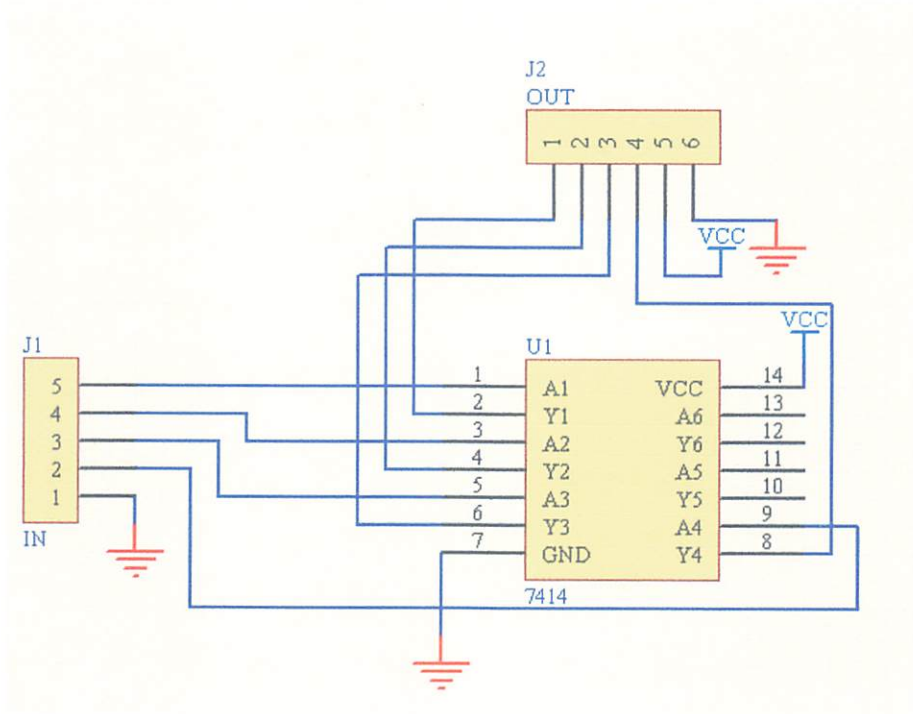
lantai, sehingga nilai *resistansi*-nya akan berbeda. Semakin besar intensitas cahaya yang diterima oleh *photodiode*, maka nilai *resistansi*-nya akan semakin kecil dan nilai tegangan *output*-nya akan semakin kecil pula.

Perbedaan nilai tegangan *output* dari *photodiode* saat menerima cahaya pantulan dari lantai dan garis akan dideteksi oleh rangkaian komparator. Tegangan referensi dapat diatur dengan memutar variabel resistor. Untuk dapat membedakan antara garis dan lantai, nilai tegangan referensi diatur sehingga memiliki nilai diantara nilai tegangan *output* dari *photodiode* saat menerima pantulan cahaya dari lantai dan garis. Pada saat *photodiode* menerima pantulan cahaya dari lantai, nilai tegangan *output*-nya akan lebih kecil dari tegangan referensi, sehingga *output* dari komparator akan bernilai “0”. Sedangkan saat *photodiode* menerima pantulan cahaya dari garis, nilai tegangan *output*-nya akan lebih besar dari tegangan referensi, sehingga *output* dari komparator bernilai “1”.

3.1.3. Perancangan Rangkaian *Smitch Trigger*

Karena level tegangan keluaran dari rangkaian pengkondisi sinyal masih dalam kondisi ambang, hal ini menyebabkan *mikrokontroller* tidak dapat melakukan instruksi dengan baik, karena input dari pengkondisi sinyal tidak sepenuhnya high, atau tidak sepenuhnya low, maka untuk mengatasi hal ini, digunakan rangkaian

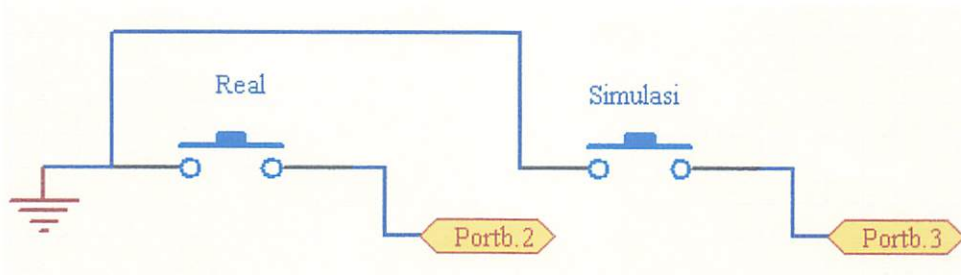
smitch trigger dengan menggunakan IC 74LS14 yang bertujuan untuk menegaskan tegangan yang dianggap ambang oleh *mikrokontroller*.



Gambar 3-6 Rangkaian *Smitch Trigger* 74LS14

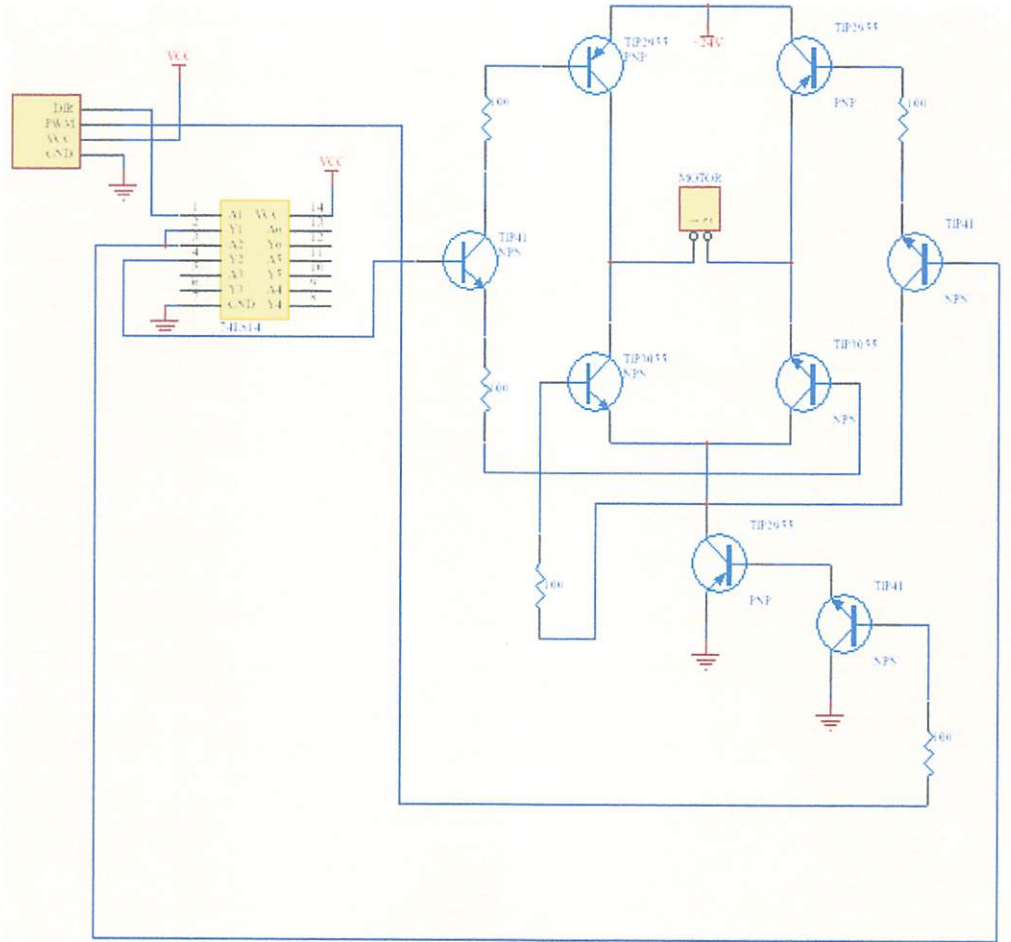
3.1.4. Perancangan Rangkaian *Push Button*

Tombol *push button* digunakan untuk memilih program, apakah untuk program yang sesungguhnya atau untuk *display* simulasi, hal ini bertujuan agar data yang ditampilkan oleh *LCD* bisa terlihat dengan jelas,



Gambar 3-7 Rangkaian *Push Button*

3.1.5. Perancangan *Driver Motor DC*



Gambar 3-8 Rangkaian *Driver Motor DC*

Motor DC yang digunakan memiliki arus beban maksimum sebesar $3,87 \text{ Ampere}$, oleh karena itu digunakan *transistor* 3055 dan 2955 yang mempunyai I_C maksimum hingga 15 Ampere sebagai penguat daya untuk motor, dan *transistor* TIP 41 sebagai *switching*. Untuk menentukan besarnya *resistor* yang digunakan, dapat menggunakan rumus sebagai berikut:

$$I_B = \frac{I_C}{h_{fe}} = \frac{3,87}{90} = 0,043 \text{ Ampere}$$

$$\begin{aligned}
 R_3/R_4 &= \frac{Vb - Vbe}{I_B} \\
 &= \frac{5 - 0,7}{0,043} \\
 &= \frac{4,3}{0,043} = 100 \Omega
 \end{aligned}$$

$$\begin{aligned}
 R_1/R_2 &= \frac{Vss - Vbe}{I_B} \\
 &= \frac{24 - 0,7}{0,043} \\
 &= \frac{23,3}{0,043} = 541,8 \Omega
 \end{aligned}$$

Karena *resistor* dengan nilai $541,8 \Omega$ tidak tersedia di pasaran, maka digunakan nilai yang mendekati, yaitu 560Ω

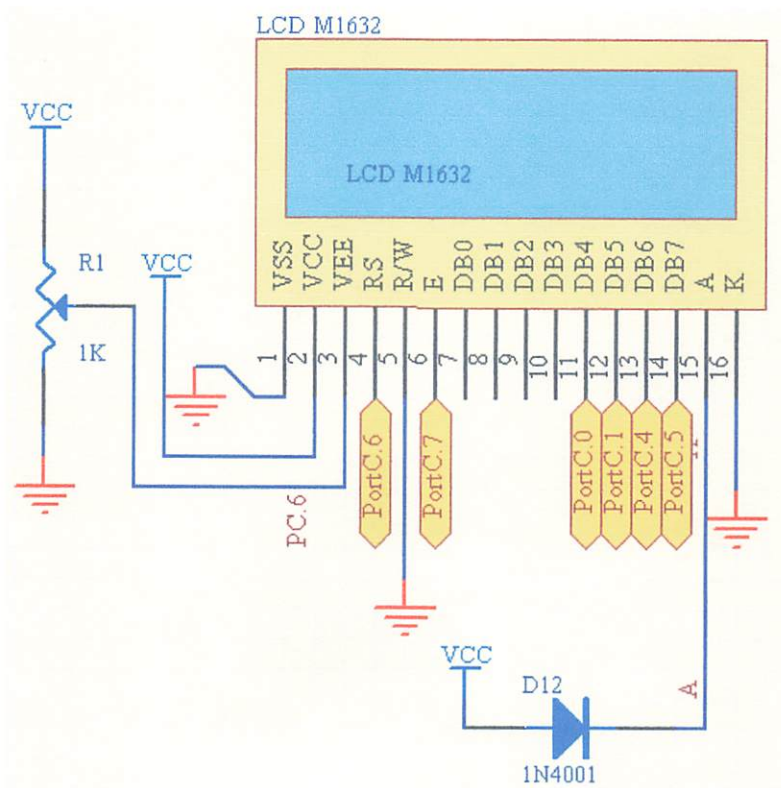
3.1.5.1. Cara Kerja Rangkaian *Driver Motor DC*

Transistor NPN dan *PNP* disusun secara *H-Bridge*, hal ini bertujuan untuk merubah polaritas motor, sehingga motor bisa bergerak dua arah. *IC 74LS14* digunakan bertujuan untuk menghemat *port mikrokontroller*, dimana dengan kombinasi *IC 74LS14*, hanya dibutuhkan 1 *port mikrokontroller* untuk mengatur satu buah motor. Keluaran dari A1 menjadi masukan bagi A2 dan masukan bagi *transistor TIP 41* sisi kanan atas, sedangkan keluaran dari A2 menjadi masukan bagi *transistor TIP 41* sisi kiri. Penggunaan *IC 74LS14* ini juga bertujuan untuk menghindarkan terjadinya *short* akibat kesalahan dalam hal

pemrograman *direction* motor DC, kondisi *logic* sama-sama “1” tidak mungkin terjadi, terkecuali IC 74LS14 telah rusak.

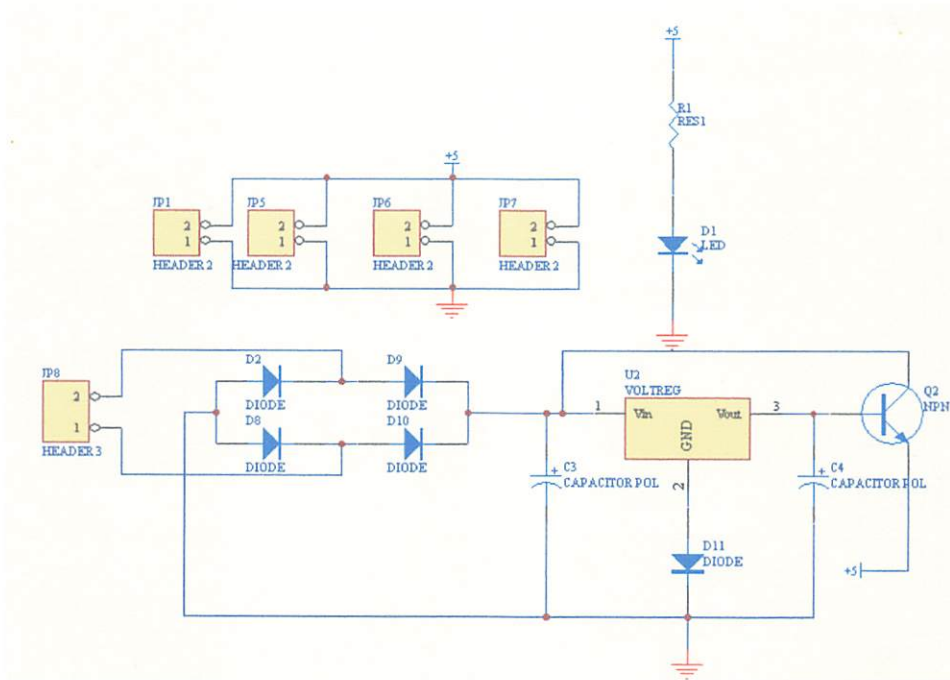
Transistor NPN bagian paling bawah, digunakan untuk *enable* motor, karena dalam hal ini digunakan metode *PWM*, maka *transistor TIP 41* ini diberi data *PWM* keluaran dari *mikrokontroler*.

3.1.6. Perancangan Rangkaian LCD



Gambar 3-9 Rangkaian LCD

3.1.7. Perancangan Catu Daya Sistem



Gambar 3-10 Rangkaian Catu Daya

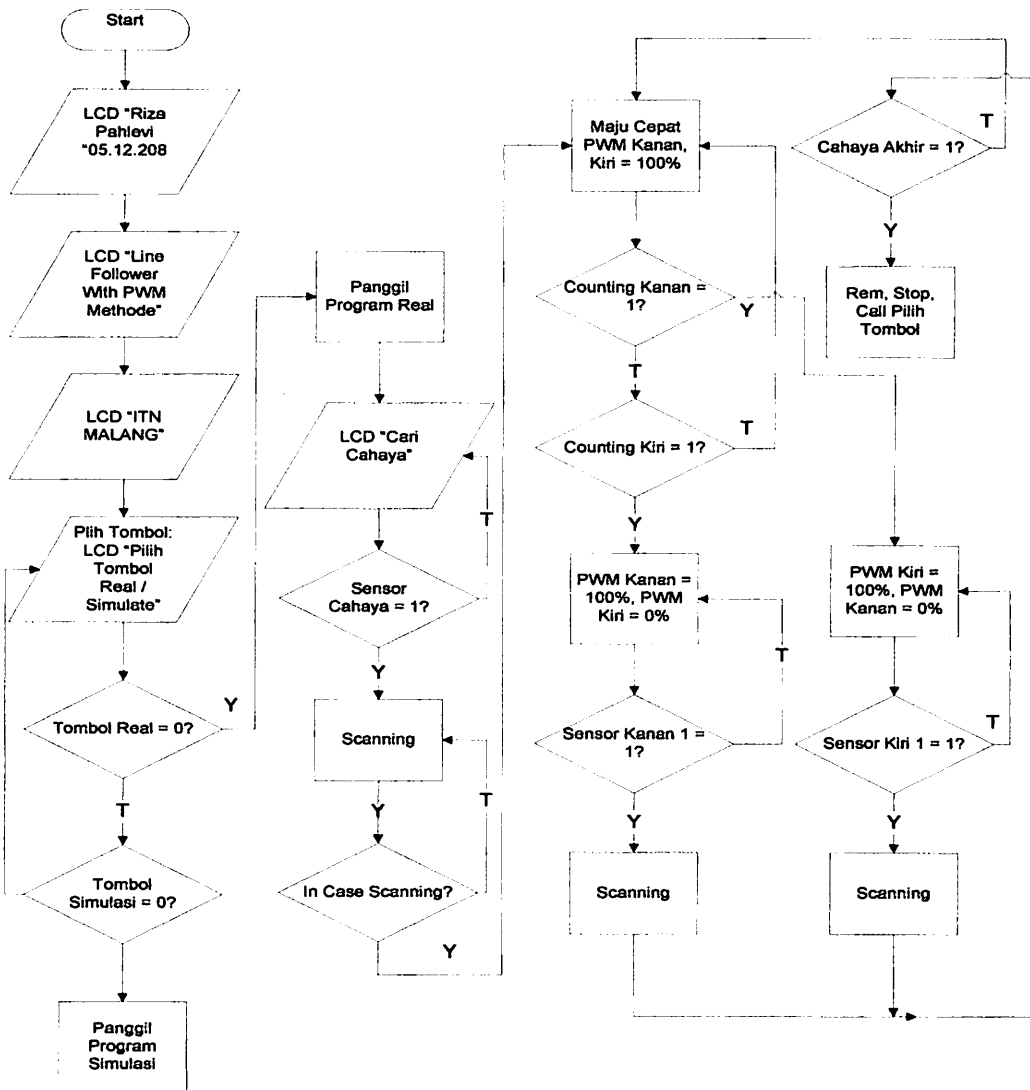
Pada sistem yang digunakan, tidak semua rangkaian mempunyai level tegangan *input* catu daya yang sama, dimana untuk sistem dari robot digunakan catu daya sebesar +5 Volt, sedangkan untuk *output* motor digunakan catu daya sebesar +24 Volt. Untuk mendapatkan tegangan +5 Volt dari masukan 12 Volt dari baterai, maka dibutuhkan suatu rangkaian *regulator* tegangan sehingga dapat dihasilkan catu daya untuk sistem robot sebesar +5 Volt.

Untuk me-regulasikan tegangan dari 12 Volt menjadi +5 Volt, digunakan *Transistor* LM 7805.

3.2. Perancangan Perangkat Lunak (Software)

Pemrograman *mikrokontroller* menggunakan bahasa *basic*, dengan *compiler* menggunakan program *Basic Compiler*, untuk meng-*upload* program pada *mikrokontroller* menggunakan program *SPI Programmer* dengan komunikasi *port parallel*.

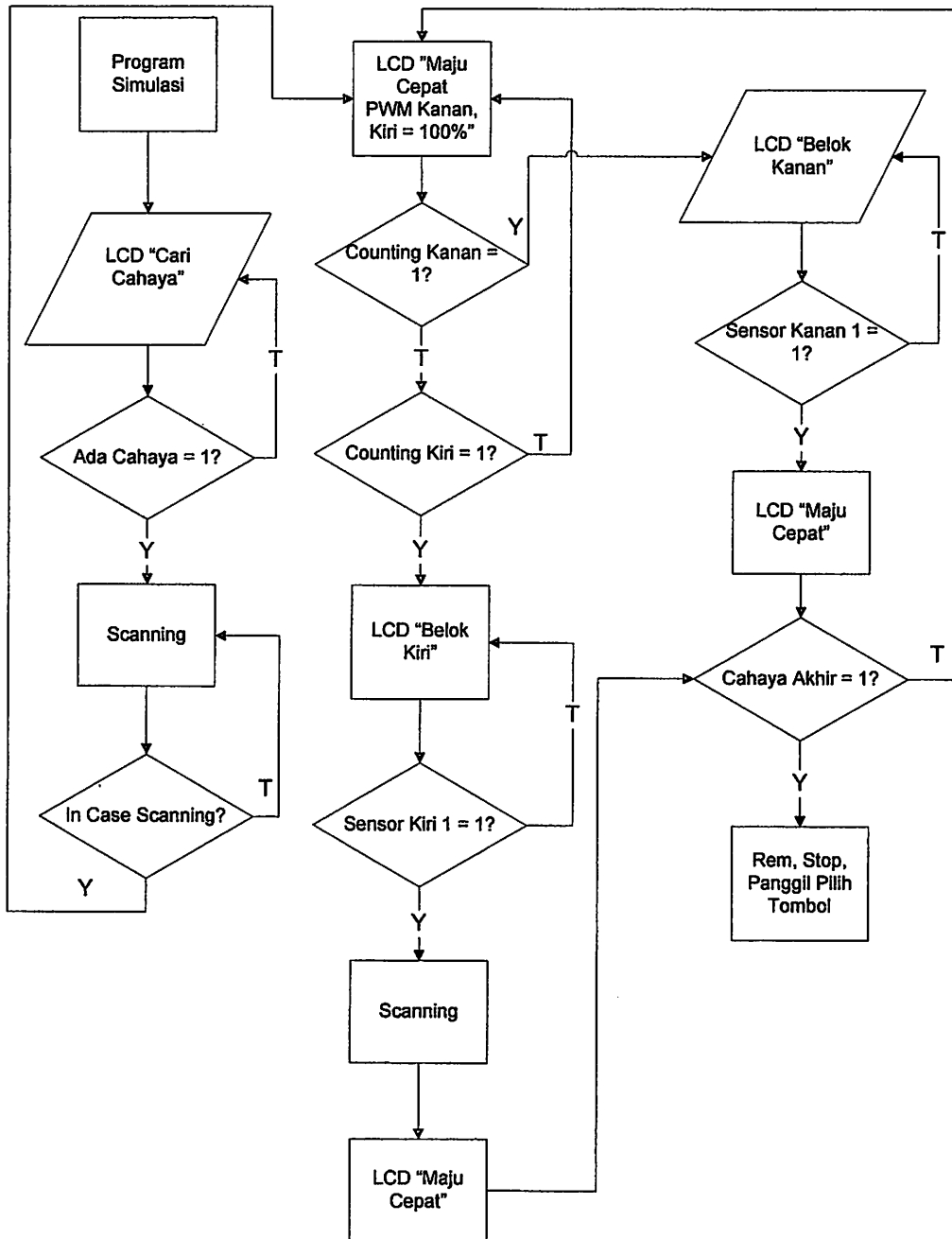
3.2.1. Diagram Alir Program Real



Gambar 3-11 Diagram Alir Program Real

- Keterangan :
Y = Ya
T = Tidak

3.2.2. Diagram Alir Program Simulasi



Gambar 3-12 Diagram Alir Program Simulasi

- Keterangan :
 Y = Ya
 T = Tidak

BAB IV

PENGUJIAN ALAT

Pada bab ini membahas cara pengujian dan analisa dari alat yang dirancang, sehingga dapat diketahui apakah alat tersebut dapat bekerja sesuai dengan yang telah direncanakan. Dalam rangka pengujian alat tersebut diuraikan percobaan yang dilakukan untuk mengetahui respon dari keseluruhan alat yang sudah dirancang.

Untuk mengetahui kemampuan alat dan sistem kerja sesuai dengan program yang telah dibuat maka dilakukan pengujian pada alat dan sistem kerja alat dengan prosedur pengujian per blok pada setiap sistem.

4.1. Pengujian *Minimum System* ATMEGA 16

4.1.1. Tujuan

Tujuan dari pengujian rangkaian *Minimum System* ATMEGA 16 adalah untuk menguji data yang telah diprogram pada *mikrokontroller* bisa dieksekusi sesuai dengan alur program.

4.1.5. Hasil Pengujian

4.1.5.1. Pengujian Pada *Port A*

Tabel 4-1 Hasil Pengujian Pada *Port A Minimum System*

<i>Port</i>	<i>Logic Masukan (Bit)</i>	<i>Tegangan Keluaran (Volt)</i>	<i>LED Indikator</i>
A.0	1	4.80	Mati
A.1	1	4.80	Mati
A.2	1	4.78	Mati
A.3	1	4.80	Mati
A.4	0	0.14	Nyala
A.5	0	0.13	Nyala
A.6	0	0.12	Nyala
A.7	0	0.12	Nyala

Mikrokontroller ATMEGA 16 untuk mengkonfigurasi *port* sebagai keluaran, maka *port* diberi logika “1” terlebih dahulu, dengan kata lain, keluaran *mikrokontroller* aktif *low*, sebaliknya untuk mengkonfigurasi *port* sebagai masukan, maka *port mikrokontroller* diberi logika “0” terlebih dahulu, dengan kata lain masukan *mikrokontroller* aktif *high*.



Gambar 4-2 Tegangan Keluaran *Mikrokontroller PortA*
Kondisi *High*



Gambar 4-3 Tegangan Keluaran *Mikrokontroller PortA*
Kondisi *Low*

4.1.5.2. Pengujian Pada Port B

Tabel 4-2 Hasil Pengujian Pada Port B Minimum System

Port	Logic Masukan (Bit)	Tegangan Keluaran (Volt)
B.0	1	3.23
B.1	1	3.38
B.2	1	4.70
B.3	1	4.71
B.4	0	0.01
B.5	0	0.01
B.6	0	0.02
B.7	0	0.01

Port B, tidak terhubung dengan led sebagai indikator, maka untuk pengujian hanya mengacu pada tegangan keluaran



Gambar 4-4 Tegangan Keluaran Mikrokontroller PortB Kondisi High

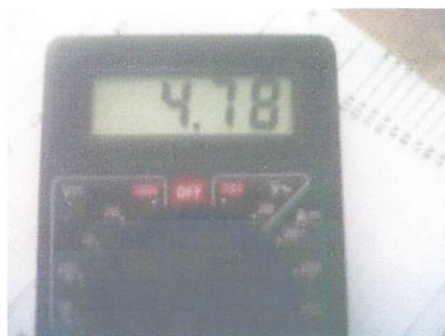


Gambar 4-5 Tegangan Keluaran *Mikrokontroller PortB*
Kondisi *Low*

4.1.5.3. Pengujian Pada *Port C*

Tabel 4-3 Hasil Pengujian Pada *Port C Minimum System*

<i>Port</i>	<i>Logic Masukan (Bit)</i>	<i>Tegangan Keluaran (Volt)</i>	<i>LED</i>
C.0	1	4.78	Mati
C.1	1	4.77	Mati
C.2	1	4.78	Mati
C.3	1	4.78	Mati
C.4	0	0.20	Nyala
C.5	0	0.21	Nyala
C.6	0	0.20	Nyala
C.7	0	0.15	Nyala



Gambar 4-6 Tegangan Keluaran *Mikrokontroller PortC*
Kondisi *High*



Gambar 4-7 Tegangan Keluaran *Mikrokontroller PortC*
Kondisi *Low*

4.1.5.4. Pengujian Pada *Port D*

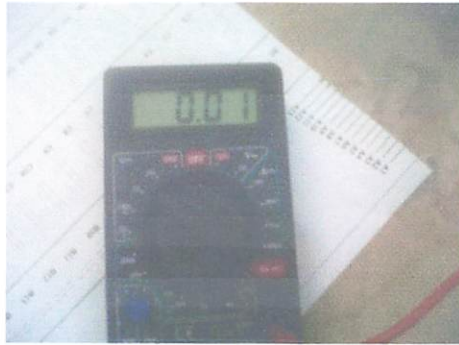
Tabel 4-4 Hasil Pengujian Pada *Port D Minimum System*

<i>Port</i>	<i>Logic Masukan (Bit)</i>	<i>Tegangan Keluaran (Volt)</i>
D.0	1	4.77
D.1	1	4.78
D.2	1	4.79
D.3	1	4.78
D.4	0	0.01
D.5	0	0.01
D.6	0	0.02
D.7	0	0.02

Port D, tidak terhubung dengan *led* sebagai indikator, maka untuk pengujian hanya mengacu pada tegangan keluaran



Gambar 4-8 Tegangan Keluaran *Mikrokontroller PortD*
Kondisi *High*



Gambar 4-9 Tegangan Keluaran *Mikrokontroler PortD*
Kondisi *Low*



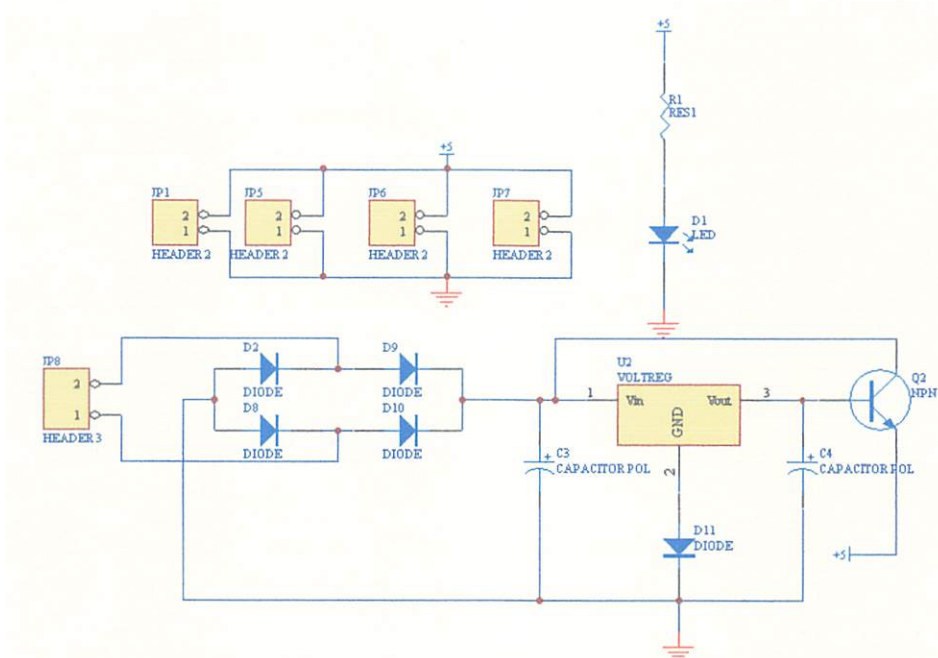
Gambar 4-10 Rangkaian *Mikrokontroler ATMEGA 16*

4.2. Pengujian Rangkaian *Regulator* Tegangan

4.2.1. Tujuan

Mengetahui keluaran tegangan dari rangkaian *regulator* tegangan LM7805, hal ini bertujuan untuk menghindari kelebihan level tegangan yang bisa mengakibatkan rusaknya perangkat keras sistem.

4.2.2. Rangkaian Pengujian *Regulator* Tegangan



Gambar 4-11 Rangkaian Pengujian *Regulator* Tegangan

4.2.3. Peralatan Yang Digunakan.

- *Multimeter Digital.*
- *Catu Daya 12 Volt*

4.2.4. Langkah-Langkah Pengujian.

Langkah- langkah pengujian antara lain :

- 1) Memberikan masukan tegangan 12 *Volt* pada terminal masukan *regulator* tegangan
- 2) Mengukur tegangan keluaran dari *regulator* tegangan
- 3) Mencatat hasil pengamatan yang dilakukan

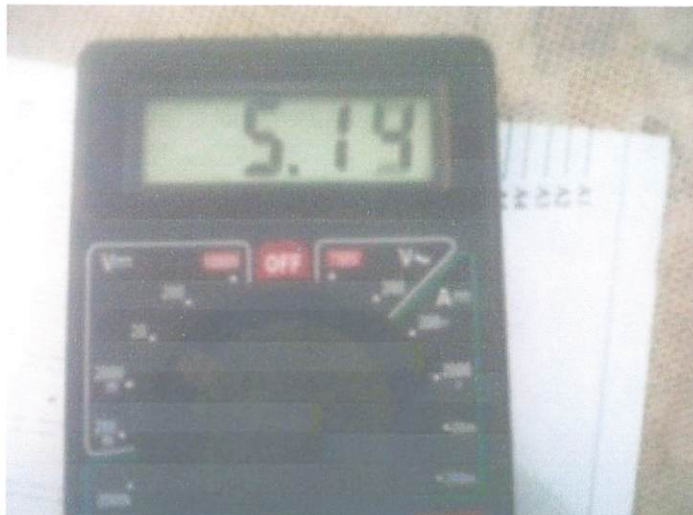
4.2.5. Hasil Pengujian *Regulator* Tegangan

Tabel 4-5 Hasil Pengujian Rangkaian *Regulator* Tegangan

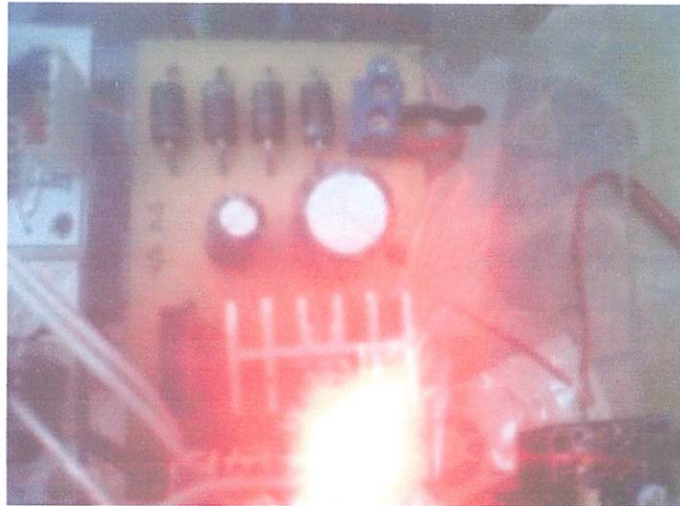
Tegangan Masukan (Volt)	Tegangan Keluaran (Volt)
12.9	5.14



Gambar 4-12 Tegangan Masukan *Regulator* Tegangan LM 7805



Gambar 4-13 Tegangan Keluaran *Regulator* Tegangan LM 7805



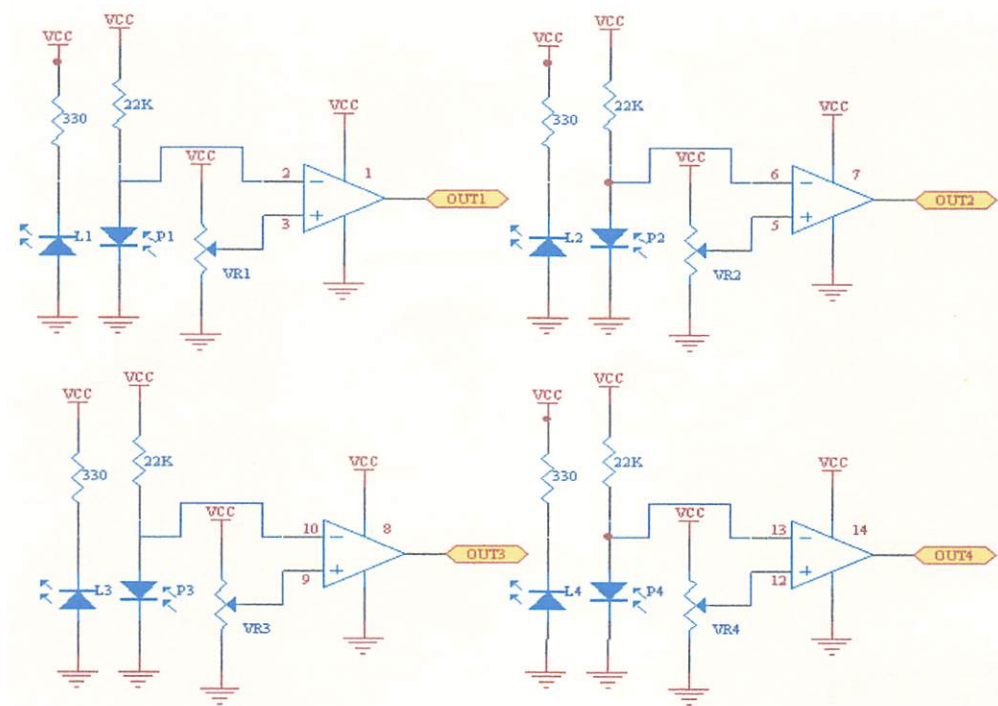
Gambar 4-14 Rangkaian *Regulator* Tegangan LM 7805

4.3. Pengujian Rangkaian Pengkondisi Sinyal

4.3.1. Tujuan

Pengujian ini perlu dilakukan untuk mengetahui perbandingan level tegangan saat sensor mengenai garis dan saat sensor mengenai lantai

4.3.2. Rangkaian Pengujian Pengkondisi Sinyal



Gambar 4-15 Rangkaian Pengujian Pengkondisi Sinyal

4.3.3. Peralatan Yang Digunakan

- Catu Daya 5 *Volt* keluaran dari *regulator* tegangan LM7805
- *Multimeter Digital*
- Obeng *Trimmer*
- Kertas Putih Sebagai Objek Pembanding Gelap Terang

4.3.4. Langkah-Langkah Pengujian

- 1) Menghubungkan rangkaian pengkondisi sinyal dengan catu daya 5 *Volt*
- 2) Meletakkan kertas putih dengan jarak ± 2 Cm dari ujung permukaan sensor
- 3) Men-*setting variable resistor* dengan obeng *trimmer*, bergerak memutar hingga indikator led menyala, kemudian memutar sedikit hingga indikator mati, kemudian memutar perlahan hingga indikator led menyala kembali
- 4) Mengukur tegangan keluaran pengkondisi sinyal saat sensor mengenai lantai
- 5) Mengukur tegangan keluaran pengkondisi sinyal saat sensor mengenai garis putih
- 6) Mencatat hasil pengukuran tegangan keluaran pengkondisi sinyal

4.3.5. Hasil Pengujian Rangkaian Pengkondisi Sinyal

4.3.5.1. Hasil Pengujian Rangkaian Pengkondisi Sinyal Sensor Scanning

Tabel 4-6 Hasil Pengujian Rangkaian Pengkondisi Sinyal *Sensor Scanning*

<i>Sensor Scanning</i>	Tegangan Masukan (Volt)	V_{out} Sensor Terkena Garis	V_{out} Buffer Terkena Garis	V_{out} Sensor Terkena Lantai	V_{out} Buffer Terkena Lantai
1	5.16	0.15	4.71	4.71	0.15
2	5.16	0.15	4.73	4.73	0.15
3	5.16	0.15	4.72	4.72	0.15
4	5.16	0.14	4.72	4.72	0.14
5	5.16	0.14	4.73	4.73	0.14

Keterangan : Sensor *scanning* dihitung mulai dari ujung paling kiri sensor



Gambar 4-16 Tegangan Masukan *Sensor Scanning*



Gambar 4-17 V_{OUT} Sensor Saat Terkena Garis



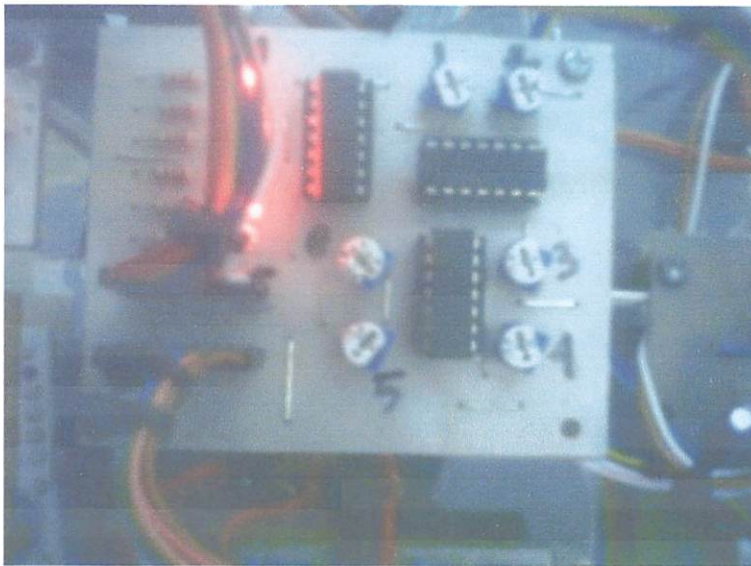
Gambar 4-18 V_{OUT} Buffer Saat Terkena Garis



Gambar 4-19 V_{OUT} Sensor Saat Terkena Lantai



Gambar 4-20 V_{OUT} Buffer Saat Terkena Lantai



Gambar 4-21 Rangkaian Pengkondisi Sinyal



Gambar 4-22 Rangkaian Susunan Sensor

4.3.5.2. Hasil Pengujian Rangkaian Pengkondisi Sinyal Sensor Counting

Tabel 4-7 Hasil Pengujian Rangkaian Pengkondisi Sinyal Sensor Counting

Sensor Counting	V_{in} (Volt)	V_{out} Terkena Garis (Volt)	V_{out} Terkena Lantai (Volt)
Kanan	5.16	4.72	0.15
Kiri	5.16	4.71	0.14

4.3.5.3. Hasil Pengujian Rangkaian Pengkondisi Sinyal Detektor Cahaya

Tabel 4-8 Hasil Pengujian Rangkaian Pengkondisi Sinyal Detektor Cahaya

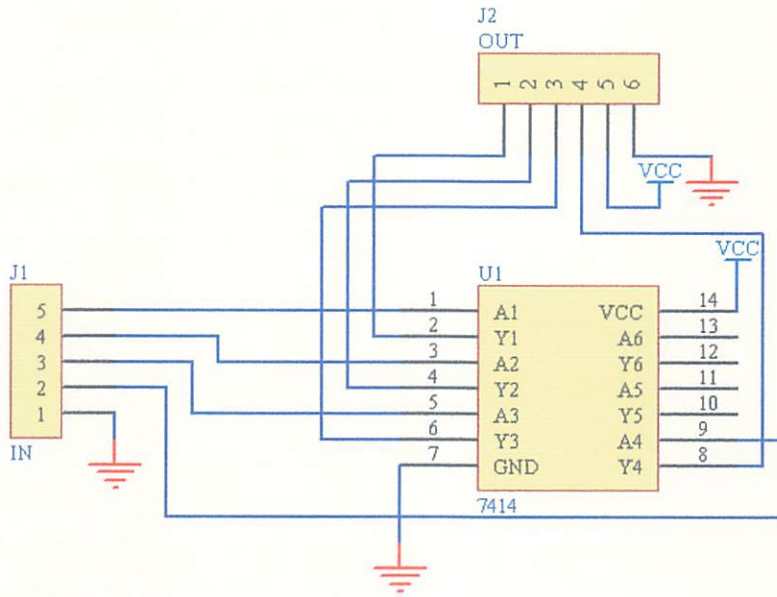
Detektor Cahaya	V_{in} (Volt)	V_{out} Terkena Garis (Volt)	V_{out} Terkena Lantai (Volt)
Awal	5.16	4.71	0.15
Akhir	5.16	4.73	0.15

4.4. Pengujian Rangkaian *Smitch Trigger*

4.4.1. Tujuan

Tujuan dari pengujian rangkaian *smitch trigger* ini adalah untuk mengetahui keluaran dari rangkaian *smitch trigger*, apakah sudah sesuai dengan masukan dari pengkondisi sinyal, keluaran dari rangkaian *smitch trigger* berkebalikan dengan masukannya.

4.4.2. Rangkaian Pengujian *Smitch Trigger*



Gambar 4-23 Pengujian Rangkaian *Smitch Trigger*

4.4.3. Peralatan Yang Diperlukan

- Catu Daya 5 Volt
- Multimeter Digital

4.4.4. Langkah-Langkah Pengujian Rangkaian *Smitch Trigger*

- 1) Menghubungkan Rangkaian *smitch trigger* dengan catu daya 5 Volt
- 2) Memberikan tegangan pada pin masukan IC 74LS14
- 3) Mengukur tegangan pada pin keluaran IC 74LS14
- 4) Mencatat hasil pengukuran tegangan masukan dan keluaran pada rangkaian *smitch trigger*

4.4.5. Hasil Pengujian Rangkaian *Smitch Trigger*

Tabel 4-9 Hasil Pengukuran Tegangan Pada Rangkaian *Smitch Trigger*

<i>Input Ke-</i>	Tegangan Masukan (<i>Volt</i>)	Tegangan Keluaran (<i>Volt</i>)
1	0.15	4.73
2	0.15	4.73
3	0.15	4.71
4	0.15	4.73
5	0.15	4.71
6	0.15	4.73

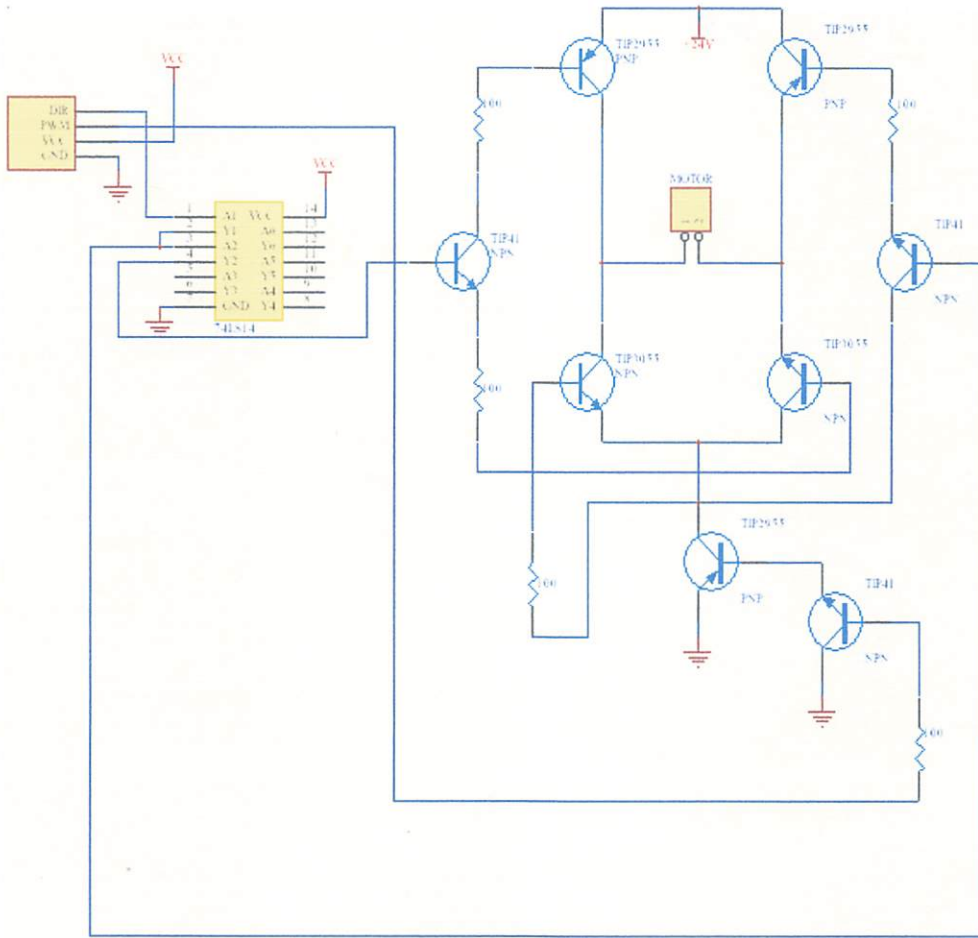
4.5. Pengujian Rangkaian *Driver Motor DC*

4.5.1. Tujuan

Tujuan dari pengujian rangkaian *driver motor* ini adalah untuk mengetahui apakah arah dari motor DC sudah bisa dikendalikan secara program.

Pengujian ini bertujuan juga untuk menganalisa data *PWM* keluaran dari *mikrokontroller*.

4.5.2. Rangkaian Pengujian *Driver Motor DC*



Gambar 4-24 Pengujian Rangkaian *Driver Motor DC*

4.5.3. Peralatan Yang Diperlukan

- Catu Daya 24 Volt
- Catu Daya 5 Volt
- *Multimeter Digital*
- Motor DC
- *Oscilloscope Digital*

4.5.4. Langkah-Langkah Pengujian Rangkaian *Driver Motor*

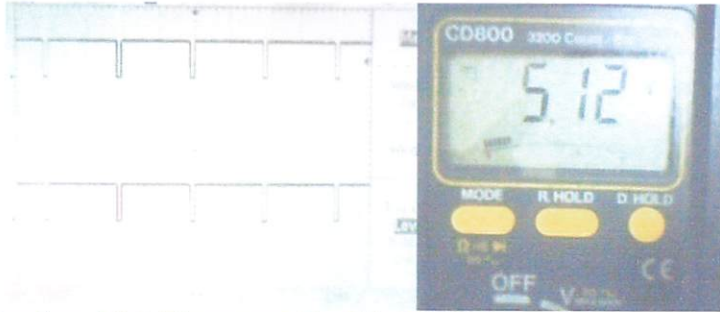
- 1) Menghubungkan rangkaian *driver motor dc* dengan catu daya 5 *Volt*
- 2) Menghubungkan motor dengan tegangan 24 *Volt*
- 3) Mengukur tegangan pada keluaran rangkaian *driver motor dc*
- 4) Menganalisa data *PWM* dengan menampilkannya pada *oscilloscope*

4.5.5. Hasil Pengujian Rangkaian *Driver Motor DC*

Tabel 4-10 Hasil Pengukuran Pada Rangkaian *Driver Motor DC*

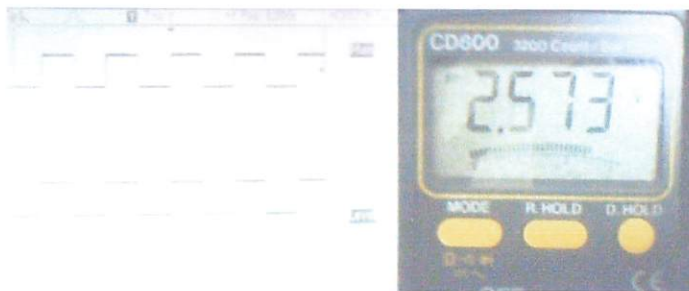
Motor DC	<i>Direction (Bit)</i>	<i>Duty Cycle (%)</i>	Arah Putaran Motor	$V_{OUT\ PWM}$
Kanan	1	94	Maju	5,12
Kanan	1	50	Maju	2,5
Kanan	0	94	Mundur	5,12
Kanan	0	50	Mundur	2,5
Kiri	1	94	Maju	5,12
Kiri	1	50	Maju	2,5
Kiri	0	94	Mundur	5,12
Kiri	0	50	Mundur	2,5

$$\begin{aligned} \text{Perhitungan } \textit{duty cycle} &= t_{\text{high}} / (t_{\text{high}} + t_{\text{low}}) \times 100\% \\ &= 240 / (240 + 15) \times 100\% \\ &= 94 \% \end{aligned}$$



Gambar 4-25 Tegangan *Enable PWM Digital* Dengan *Input Duty Cycle 94,12 %*

$$\begin{aligned} \text{Perhitungan } \textit{duty cycle} &= t_{\text{high}} / (t_{\text{high}} + t_{\text{low}}) \times 100\% \\ &= 128 / (128 + 127) \times 100\% \\ &= 50 \% \end{aligned}$$



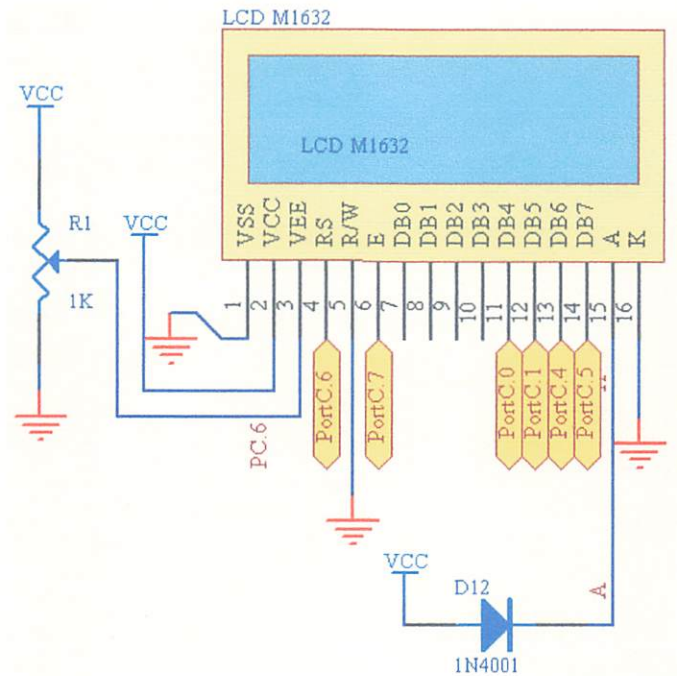
Gambar 4-26 Tegangan *Enable PWM Digital* Dengan *Input Duty Cycle 50,19 %*

4.6. Pengujian Rangkaian *LCD (Liquid Crstal Display)*

4.6.1. Tujuan

Tujuan dari pengujian rangkaian *LCD* adalah untuk menguji apakah *LCD* dapat menampilkan sesuai yang telah diprogram pada *mikrokontroller*

4.6.2. Rangkaian Pengujian LCD



Gambar 4-27 Pengujian Rangkaian LCD

4.6.3. Peralatan Yang Diperlukan

- Catu Daya 5 Volt

4.6.4. Langkah-Langkah Pengujian Rangkaian LCD

- 1) Menghubungkan rangkaian LCD dengan catu daya 5 Volt
- 2) Menampilkan tulisan “Riza Pahlevi”, dan pada baris selanjutnya menampilkan “05.12.208”

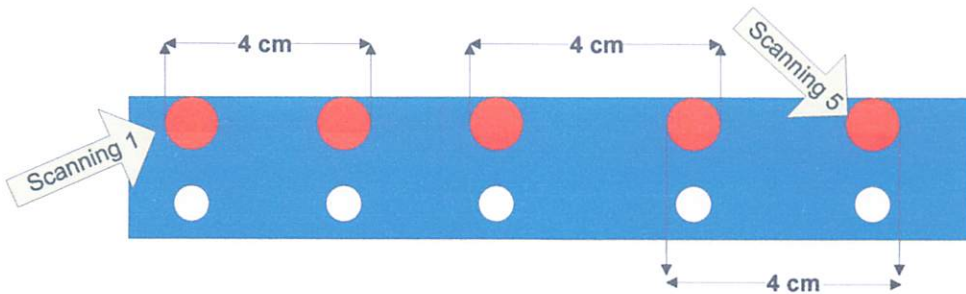
4.6.5. Hasil Pengujian Rangkaian LCD



Gambar 4-28 Pengujian LCD

4.7. Pengujian Keseluruhan Sistem

Pada pengujian keseluruhan sistem, robot berjalan mengikuti garis sesuai dengan masukan data yang diperoleh dari sensor garis, dan dikomparasi oleh pengkondisi sinyal, kemudian diproses oleh mikrokontroler untuk menentukan intruksi pada keluaran sesuai dengan masukan yang diperoleh, berikut susunan sensor yang dipasang pada robot



Gambar 4-29 Susunan Pemasangan sensor Garis

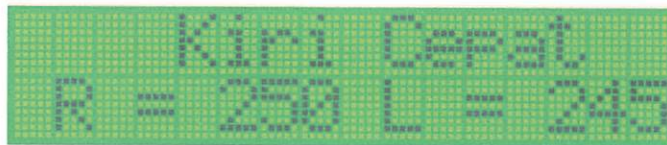
Dalam pengujian alat secara keseluruhan, setiap susunan sensor (dari *scanning 1* hingga *scanning 5*), mempunyai nilai *PWM* masing-masing, hal ini bertujuan agar pergerakan robot tidak terlalu kasar waktu melakukan *scanning* garis.

4.7.1. Hasil Pengukuran Sistem

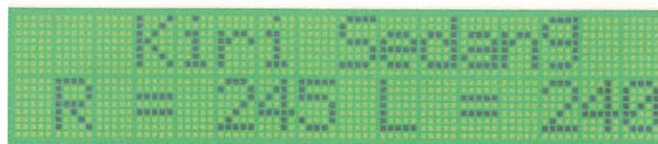
Untuk mengetahui pergerakan robot sesuai dengan alur program yang dibuat, maka pergerakan robot disimulasikan dengan tampilan LCD

Tabel 4-11 Hasil Pengamatan Dan Pengukuran Sistem

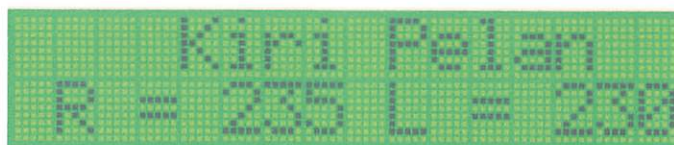
Sensor Terkena Garis	Arah Motor Kanan	Arah Motor Kiri	<i>PWM</i> Kanan (Byte)	<i>PWM</i> Kiri (Byte)
Scanning 1	Maju	Mundur	250	245
Scanning 2	Maju	Mundur	245	240
Scanning 3	Maju	Mundur	235	230
Scanning 4	Mundur	Maju	230	235
Scanning 5	Mundur	Maju	240	245



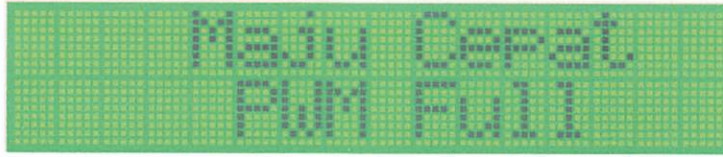
Gambar 4-30 Tampilan Saat Sensor Paling Kiri Terkena Garis



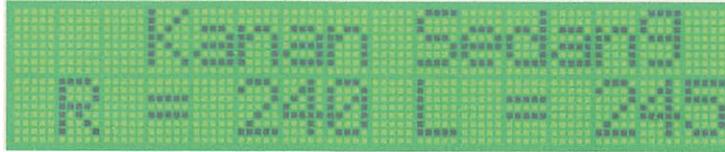
Gambar 4-31 Tampilan Saat Sensor Ke-Dua Dari Kiri Terkena Garis



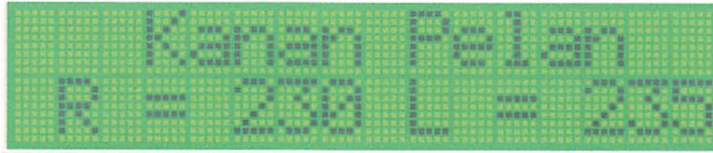
Gambar 4-32 Tampilan Saat Sensor Ke-Tiga Dari Kiri Terkena Garis



Gambar 4-33 Tampilan Saat Sensor Ke-Tiga Dan Ke-Empat Dari Kiri Terkena Garis



Gambar 4-34 Tampilan Saat Sensor Paling Kanan Terkena Garis



Gambar 4-35 Tampilan Saat Sensor Ke-Dua Dari Kanan Terkena Garis

BAB V

PENUTUP

Pada bab ini penulis akan menyampaikan kesimpulan dan saran setelah melakukan pengujian pada sistem yang telah dibuat

5.1. Kesimpulan

Setelah melakukan perencanaan, pembuatan, dan pengujian, dapat diambil beberapa kesimpulan sebagai berikut :

- a. Pergerakan robot lebih halus dengan menggunakan metode *Pulse Width Modulation* sebagai pengendali kecepatan
- b. Pada nilai *Duty Cycle* 50%, robot tidak dapat bergerak, hal ini dikarenakan masalah mekanik
- c. Untuk penggunaan *PWM 8 bit* yang tersedia pada *ATMEGA 16*, masih kurang bagus untuk keakuratan nilai *PWM* yang diberikan, dibutuhkan *range bit PWM* yang lebih besar lagi
- d. Sinyal *PWM* yang dibangkitkan melalui program, lebih bagus daripada pembangkitan melalui *hardware (PWM Analog)* yang memiliki *ripple* lebih banyak
- e. Penggunaan *smitch trigger* pada rangkaian *driver* motor lebih menghemat *port mikrokontroller*
- f. Penggunaan sensor garis yang difungsikan sebagai detektor cahaya, kurang efektif, karena dinilai terlalu sensitif

5.2. Saran

Alat yang dibuat ini masih memiliki keterbatasan, nantinya diharapkan dapat dikembangkan untuk mengatasi keterbatasan itu. Sehingga mendapatkan alat yang diharapkan dapat mendekati alat yang ideal.

- a. Bagi peneliti selanjutnya diharapkan bisa menambahkan metode *close loop* dengan menggunakan metode *PID*
- b. Penambahan *rotary encoder* sangat diperlukan untuk keakuratan pengontrolan motor.
- c. Jumlah sensor *scanning* perlu ditambah untuk mendapatkan pergerakan robot yang lebih halus lagi.
- d. Masih besarnya beban robot, oleh karena itu sebaiknya pembuatan mekanik robot lebih efisien dan efektif

DAFTAR PUSTAKA

- [1] http://en.wikipedia.org/wiki/File:Duty_cycle_general.png
- [2] <http://insansainsprojects.files.wordpress.com>
- [3] Tim Lab. *Mikroprosesor*. 2006. “ Pemograman *mikrokontroller* AT89S51 dengan *C/C++* dan *Assembler* “. Yogyakarta. Andi.
- [4] Suhata, ST. “Aplikasi *Mikrokontroller* Sebagai Pengendali Peralatan Elektronik via *Line telepon*”. Jakarta. PT Elex Media Komputindo.
- [5] John G. Proakis, Dimitris G. Manokalis. 1995 “Pemrosesan Sinyal Digital”. Jakarta. PT Prenhallindo.
- [5] <http://www.robotindonesia.com>
- [6] <http://www.elektroniclub.com>



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1

FORMULIR PERBAIKAN SKRIPSI

Nama : Riza Pahlevi
NIM : 05.12.208
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul : Perancangan Dan Pembuatan Robot *Line Follower*
Dengan Kendali Kecepatan Menggunakan Metode
Pulse Width Modulation Berbasis *Mikrokontroller*
ATMEGA 16
Hari / Tanggal Ujian Skripsi : Sabtu / 21 Agustus 2010

Tanggal	Dosen	Materi Perbaikan	Paraf
	Penguji I	Latar Belakang, Kesimpulan	
	Penguji II	Abstrak, Latar Belakang, <i>Flowchart</i> Sistem, Perhitungan <i>Baudrate</i> , Daftar Pustaka	

Disetujui,

Penguji I

Irmalia Suryani Faradisa, ST., MT
NIP. P. 1030100365

Penguji II

Sonny Prasetio, ST., MT
NIP. P. 1031000433

Mengetahui,

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Dr. Eng. Aryuanto Soetedjo ST., MT
NIP. Y. 1030800417

Lampiran

***Listing* Program**

```
'----- Created By : Riza Pahlevi'-----  
'----- Electronic Engineering' -----  
'----- National Technology Institute Of Malang'-----
```

```
$regfile = "m16def.dat"  
$crystal = 11059200  
$large
```

```
'-----Konfigurasi Port I/O-----
```

```
Config Porta = Input  
Config Portb = Input  
Config Portc = Output  
Config Portd = Output
```

```
Porta = 0
```

```
Portb.0 = 0  
Portb.1 = 0
```

```
Portb.2 = 1  
Portb.3 = 1
```

```
Portb.4 = 0  
Portb.5 = 0  
Portb.6 = 0  
Portb.7 = 0
```

```
Portc = 255  
Portd = 255
```

```
'----- Port Tombol Pilih -----
```

```
Real Alias Pinb.2      'Bulat  
Simulasi Alias Pinb.3  'Kotak
```

```
'----- Port Motor -----
```

```
'Roda Kanan      >> Portd.0  
'Roda Kiri       >> Portd.1
```

```
'1 = Maju  
'0 = Mundur
```

Motor_kanan Alias Portd.0
 Motor_kiri Alias Portd.1

'----- Port Sensor Garis -----

Kiri1 Alias Pina.0	'1
Kiri2 Alias Pina.1	'2
Tengah1 Alias Pina.2	'3
Tengah2 Alias Pina.3	'4
Kanan1 Alias Pina.4	'5
Counting_kanan Alias Pina.6	'7
Counting_kiri Alias Pina.7	'8

'----- Port Sensor Cahaya -----

Start_api Alias Pinb.0
 End_api Alias Pinb.1

'----- LCD Program -----

Config Lcdpin = Pin , Db4 = Portc.0 , Db5 = Portc.1 , Db6 = Portc.4
 , Db7 = Portc.5 , E = Portc.7 , Rs = Portc.6

Config Lcd = 16 * 2

Dim A As Byte

'----- Pwm Program -----

Pwm_kanan Alias Pwmla
 Pwm_kiri Alias Pwmlb

Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Down
 , Compare B Pwm = Clear Down , Prescale = 1

'----- Main Program -----

Enable Interrupts

```
Cls
Locate 1 , 3
Lcd "Riza Pahlevi"
Wait 1
```

```
Lowerline
Wait 1
Locate 2 , 4
Lcd "05.12.208"
Wait 2
```

```
For A = 1 To 15
Shiftlcd Left
Waitms 300
Next
```

```
Cls
Locate 1 , 3
Lcd "Line Follower"
Wait 1
```

```
Lowerline
Wait 1
Locate 2 , 2
Lcd "With PWM Method"
Wait 2
```

```
For A = 1 To 16
Shiftlcd Right
Waitms 300
Next
```

```
Cls
Locate 1 , 8
Lcd "ITN"
Wait 1
```

```
Lowerline
Locate 2 , 7
Lcd "MALANG"
Wait 2
```

```
Goto Pilih_tombol
```

```
Pilih_tombol:
```

```
Do
```

```
Cls
Locate 1 , 3
Lcd "Pilih Tombol"
```

```
Lowerline
Locate 2 , 2
Lcd "Real/Simulate?"
Waitms 200
```

```
If Real = 0 Then
```

```
    Goto Nyata
```

```
End If
```

```
If Simulasi = 0 Then
```

```
    Goto Simulate
```

```
End If
```

```
Loop
```

```
End
```

```
'----- Deklarasi Sub Routine Simulate -----'
```

```
Simulate:
```

```
Declare Sub Scanning_s  
Declare Sub Maju_cepat_s  
Declare Sub Kiri_cepat_s  
Declare Sub Kanan_sedang_s  
Declare Sub Kiri_sedang_s  
Declare Sub Kanan_pelan_s  
Declare Sub Kiri_pelan_s  
Declare Sub Belok_kanan_s  
Declare Sub Belok_kiri_s  
Declare Sub Rem_dikit_s  
Declare Sub Putar_balik_s  
Declare Sub Akhir_s
```

```
Tanya_api_s:
```

```
    Cls  
    Locate 1 , 6  
    Lcd "Start?"  
    Wait 1  
  
    Lowerline  
    Locate 2 , 3  
    Lcd "Nyalakan Api"  
    Bitwait Start_api , Set
```

```
Goto Ada_api_s
End

Ada_api_s:
  If Start_api = 1 Then
    Goto Mulai_s
  End If
End

Mulai_s:

  Do

    Scanning_s

    If Counting_kanan = 1 Then

      Rem_dikit_s
      Belok_kanan_s
      Bitwait Kanan1 , Set

    End If

    If Counting_kiri = 1 Then

      Rem_dikit_s
      Belok_kiri_s
      Bitwait Kiril , Set

    End If
```

```

If Counting_kanan = 1 Then
  If Counting_kiri = 1 Then

      Putar_balik_s
      Bitwait Counting_kanan , Set

  End If
End If

```

```

If End_api = 1 Then
  Rem_dikit_s
  Akhir_s
  Goto Pilih_tombol
End If

```

```

Loop
End

```

```

'SUB SCANNING

```

```

Sub Scanning_s

```

```

Mode 1

```

```

1

```

```

If Kiril = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then

```

```

        Akhir_s

```

```

      End If
    End If
  End If
End If

```

```

3

```

```

If Kiril = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then

```

Kanan_sedang_s

```

End If
  End If
    End If
      End If
        End If

```

----- 4 -----

```

If Kiril = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then

```

Kanan_pelan_s

```

End If
  End If
    End If
      End If
        End If

```

----- Mode 2 -----

----- 1 -----

```

If Kiril = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then

```

Kanan_pelan_s

```

End If
  End If
    End If
      End If
        End If

```

----- 3 -----

```

If Kiril = 0 Then
  If Kiri2 = 0 Then

```

```

If Tengah1 = 1 Then
  If Tengah2 = 0 Then
    If Kanan1 = 0 Then

```

```

      Kiri_pelan_s

```

```

    End If
  End If
End If
  End If
End If

```

```

----- 4 -----

```

```

If Kiril = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then

```

```

      Kiri_pelan_s

```

```

    End If
  End If
End If
  End If
End If

```

```

----- Mode 3 -----

```

```

----- 1 -----

```

```

If Kiril = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then

```

```

      Maju_cepat_s

```

```

    End If
  End If
End If
  End If
End If

```

```

----- 2 -----

```

```

If Kiril = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then

```

```

      Maju_cepat_s

```

```
End If
  End If
    End If
      End If
        End If
```

----- 3 -----

```
If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then
```

Kiri_sedang_s

```
End If
  End If
    End If
      End If
        End If
```

----- 4 -----

```
If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then
```

Akhir_s

```
End If
  End If
    End If
      End If
        End If
```

----- Mode 4 -----

----- 1 -----

```
If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then
```

Kanan_pelan_s

```
End If
  End If
    End If
```

```
End If
End If
```

'----- 2 -----'

```
If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then
```

```
          Kanan_pelan_s
```

```
        End If
      End If
    End If
  End If
End If
```

'----- 3 -----'

```
If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then
```

```
          Kiri_pelan_s
```

```
        End If
      End If
    End If
  End If
End If
```

'----- 4 -----'

```
If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then
```

```
          Kiri_pelan_s
```

```
        End If
      End If
    End If
  End If
End If
```

'----- Mode 5 -----'

```
'----- 1 -----
  If Kiril = 0 Then
    If Kiri2 = 1 Then
      If Tengah1 = 1 Then
        If Tengah2 = 1 Then
          If Kanan1 = 0 Then

            Maju_cepat_s

          End If
        End If
      End If
    End If
  End If

'----- 2 -----
  If Kiril = 0 Then
    If Kiri2 = 1 Then
      If Tengah1 = 1 Then
        If Tengah2 = 1 Then
          If Kanan1 = 1 Then

            Maju_cepat_s

          End If
        End If
      End If
    End If
  End If

'----- 3 -----
  If Kiril = 1 Then
    If Kiri2 = 0 Then
      If Tengah1 = 0 Then
        If Tengah2 = 0 Then
          If Kanan1 = 0 Then

            Kiri_cepat_s

          End If
        End If
      End If
    End If
  End If

'----- 4 -----
  If Kiril = 1 Then
    If Kiri2 = 0 Then
      If Tengah1 = 0 Then
        If Tengah2 = 0 Then
          If Kanan1 = 1 Then

            Akhir_s

          End If
        End If
      End If
    End If
  End If
```

```

End If
  End If
    End If
      End If
        End If

```

----- Mode 6 -----

----- 1 -----

```

If Kiril = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then

```

Kanan_pelan_s

```

End If
  End If
    End If
      End If
        End If

```

----- 2 -----

```

If Kiril = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then

```

Kanan_pelan_s

```

End If
  End If
    End If
      End If
        End If

```

----- 3 -----

```

If Kiril = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then

```

Kiri_pelan_s

```

End If

```

```
End If
  End If
    End If
      End If
```

```
----- 4 -----
If Kiri1 = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then
```

Kiri_pelan_s

```
End If
  End If
    End If
      End If
        End If
```

----- Mode 7 -----

```
----- 1 -----
If Kiri1 = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then
```

Maju_cepat_s

```
End If
  End If
    End If
      End If
        End If
```

```
----- 2 -----
If Kiri1 = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then
```

Maju_cepat_s

```
End If
  End If
    End If
      End If
        End If
```

----- 3 -----

```
If Kiril = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then
```

Kiri_sedang_s

```
End If
  End If
    End If
      End If
        End If
```

----- 4 -----

```
If Kiril = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then
```

Akhir_s

```
End If
  End If
    End If
      End If
        End If
```

----- Mode 8 -----

----- 1 -----

```
If Kiril = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then
```

Kanan_pelan_s

```
End If
  End If
    End If
      End If
        End If
```

'----- 2 -----'

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then
```

Kanan_pelan_s

```
End If
  End If
    End If
      End If
        End If
```

'----- 3 -----'

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then
```

Kiri_pelan_s

```
End If
  End If
    End If
      End If
        End If
```

'----- 4 -----'

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then
```

Kiri_pelan_s

```
End If
End If
```

```
End If
  End If
    End If
```

'----- Mode 9-----

'----- 1 -----

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then
```

Maju_cepat_s

```
End If
  End If
    End If
      End If
        End If
```

'----- 2 -----

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then
```

Maju_cepat_s

```
End If
  End If
    End If
      End If
        End If
```

End Sub

'----- Sub Maju Cepat -----'

Sub Maju_cepat_s

Motor_kanan = 1
Pwm_kanan = 255

Motor_kiri = 1
Pwm_kiri = 255

Cls
Locate 1 , 5
Lcd "Maju Cepat"
Waitms 100

Lowerline
Locate 2 , 6
Lcd "PWM Full"
Wait 1

End Sub

'----- Sub Kiri Cepat -----'

Sub Kiri_cepat_s

Motor_kanan = 1
Pwm_kanan = 250

Motor_kiri = 0
Pwm_kiri = 245

Cls
Locate 1 , 5
Lcd "Kiri Cepat"
Waitms 100

Lowerline
Locate 2 , 2
Lcd "R = 250 L = 245"
Wait 1

End Sub

'----- Sub Kanan Sedang -----'

Sub Kanan_sedang_s

Motor_kanan = 0
Pwm_kanan = 240

```
Motor_kiri = 1
Pwm_kiri = 245
```

```
Cls
Locate 1 , 4
Lcd "Kanan Sedang"
Waitms 100
```

```
Lowerline
Locate 2 , 2
Lcd "R = 240 L = 245"
Wait 1
```

```
End Sub
```

```
'----- Sub Kiri Sedang-----
```

```
Sub Kiri_sedang_s
```

```
Motor_kanan = 1
Pwm_kanan = 245
```

```
Motor_kiri = 0
Pwm_kiri = 240
```

```
Cls
Locate 1 , 4
Lcd "Kiri Sedang"
Waitms 100
```

```
Lowerline
Locate 2 , 2
Lcd "R = 245 L = 240"
Wait 1
```

```
End Sub
```

```
'----- Sub Kanan Pelan -----
```

```
Sub Kanan_pelan_s
```

```
Motor_kanan = 0
Pwm_kanan = 230
```

```
Motor_kiri = 1
Pwm_kiri = 235
```

```
Cls
Locate 1 , 4
Lcd "Kanan Pelan"
Waitms 100
```

```
Lowerline
Locate 2 , 2
Lcd "R = 230 L = 235"
Wait 1
```

End Sub

'----- Sub Kiri Pelan -----'

Sub Kiri_pelan_s

Motor_kanan = 1
Pwm_kanan = 235

Motor_kiri = 0
Pwm_kiri = 230

Cls
Locate 1 , 5
Lcd "Kiri Pelan"
Waitms 100

Lowerline
Locate 2 , 2
Lcd "R = 235 L = 230"
Wait 1

End Sub

'----- Sub Belok Kanan -----'

Sub Belok_kanan_s

Motor_kanan = 0
Pwm_kanan = 0

Motor_kiri = 1
Pwm_kiri = 240

Cls
Locate 1 , 5
Lcd "Turn Right"
Waitms 100

Lowerline
Locate 2 , 3
Lcd "R = 0 L = 240"
Wait 1

End Sub

'----- Sub Belok Kiri -----'

Sub Belok_kiri_s

Motor_kanan = 1
Pwm_kanan = 240

```
Motor_kiri = 0
Pwm_kiri = 0
```

```
Cls
Locate 1 , 5
Lcd "Turn Left"
Waitms 100
```

```
Lowerline
Locate 2 , 3
Lcd "R = 240 L = 0"
Waitms 1
```

```
End Sub
```

```
'----- Sub Rem Dikit -----
```

```
Sub Rem_dikit_s
```

```
Motor_kanan = 0
Pwm_kanan = 255
```

```
Motor_kiri = 0
Pwm_kiri = 255
```

```
Cls
Locate 1 , 8
Lcd "Rem"
Waitms 100
```

```
Lowerline
Locate 2 , 3
Lcd "Kelembaman = 0"
Wait 1
```

```
End Sub
```

```
'----- Sub Putar Balik -----
```

```
Sub Putar_balik_s
```

```
Motor_kanan = 0
Pwm_kanan = 0
```

```
Motor_kiri = 1
Pwm_kiri = 240
```

```
Cls
Locate 1 , 4
Lcd "Putar Balik"
Waitms 100
```

```

Lowerline
Locate 2 , 3
Lcd "R = 0 L = 240"
Waitms 1

```

```
End Sub
```

```
'----- Sub Akhir -----
```

```
Sub Akhir_s
```

```
Motor_kanan = 0
Pwm_kanan = 0
```

```
Motor_kiri = 0
Pwm_kiri = 0
```

```
Cls
Locate 1 , 8
Lcd "Stop"
Waitms 100
```

```
Lowerline
Locate 2 , 6
Lcd "PWM = 0"
Wait 1
```

```
End Sub
```

```
End
```

```
'----- Real -----
```

```
'----- Deklarasi Sub Routine Real-----
```

```
Nyata:
```

```
Declare Sub Scanning_r
Declare Sub Maju_cepat_r
```

```
Declare Sub Kiri_cepat_r
Declare Sub Kanan_sedang_r
Declare Sub Kiri_sedang_r
Declare Sub Kanan_pelan_r
Declare Sub Kiri_pelan_r
Declare Sub Belok_kanan_r
Declare Sub Belok_kiri_r
Declare Sub Rem_dikit_r
Declare Sub Putar_balik_r
Declare Sub Akhir_r
```

Tanya_api_r:

```
Cls
Locate 1 , 6
Lcd "Start?"
Wait 1

Lowerline
Locate 2 , 3
Lcd "Nyalakan Api"
Bitwait Start_api , Set
Goto Ada_api_r
```

End

Ada_api_r:

```
If Start_api = 1 Then
    Goto Mulai_r
End If
```

End

Mulai_r:

Do

Scanning_r

```
If Counting_kanan = 1 Then
```

```
Rem_dikit_r
Belok_kanan_r
Bitwait Kanan1 , Set
```

```
End If
```

```
If Counting_kiri = 1 Then
```

```
Rem_dikit_r
Belok_kiri_r

Bitwait Kiril , Set
```

```
End If
```

```
If Counting_kanan = 1 Then
  If Counting_kiri = 1 Then
```

```
    Putar_balik_r
    Bitwait Counting_kanan , Set
```

```
  End If
End If
```

```
If End_api = 1 Then
```

```
Rem_dikit_r
Akhir_r
Goto Pilih_tombol
```

```
End If
```

```
Loop
End
```

```
'SUB SCANNING
```

```
Sub Scanning_r
```

```
----- Mode 1-----
```

```
----- 1 -----
```

```
If Kiri1 = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then
```

```
          Akhir_r
```

```
        End If
      End If
    End If
  End If
End If
```

```
----- 3 -----
```

```
If Kiri1 = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then
```

```
          Kanan_sedang_r
```

```
        End If
      End If
    End If
  End If
End If
```

```
----- 4 -----
```

```
If Kiri1 = 0 Then
  If Kiri2 = 0 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then
```

```
          Kanan_pelan_r
```

```
        End If
      End If
    End If
  End If
End If
```

```
'----- Mode 2-----  
'----- 1 -----  
  If Kiril = 0 Then  
    If Kiri2 = 0 Then  
      If Tengah1 = 0 Then  
        If Tengah2 = 1 Then  
          If Kanan1 = 1 Then  
  
            Kanan_pelan_r  
  
          End If  
        End If  
      End If  
    End If  
  End If  
  
'----- 3 -----  
  If Kiril = 0 Then  
    If Kiri2 = 0 Then  
      If Tengah1 = 1 Then  
        If Tengah2 = 0 Then  
          If Kanan1 = 0 Then  
  
            Kiri_pelan_r  
  
          End If  
        End If  
      End If  
    End If  
  End If  
  
'----- 4 -----  
  If Kiril = 0 Then  
    If Kiri2 = 0 Then  
      If Tengah1 = 1 Then  
        If Tengah2 = 0 Then  
          If Kanan1 = 1 Then  
  
            Kiri_pelan_r  
  
          End If  
        End If  
      End If  
    End If  
  End If  
  
'----- Mode 3-----
```

```
'----- 1 -----
  If Kiril = 0 Then
    If Kiri2 = 0 Then
      If Tengah1 = 1 Then
        If Tengah2 = 1 Then
          If Kanan1 = 0 Then

            Maju_cepat_r

          End If
        End If
      End If
    End If
  End If

'----- 2 -----
  If Kiril = 0 Then
    If Kiri2 = 0 Then
      If Tengah1 = 1 Then
        If Tengah2 = 1 Then
          If Kanan1 = 1 Then

            Maju_cepat_r

          End If
        End If
      End If
    End If
  End If

'----- 3 -----
  If Kiril = 0 Then
    If Kiri2 = 1 Then
      If Tengah1 = 0 Then
        If Tengah2 = 0 Then
          If Kanan1 = 0 Then

            Kiri_sedang_r

          End If
        End If
      End If
    End If
  End If

'----- 4 -----
  If Kiril = 0 Then
    If Kiri2 = 1 Then
      If Tengah1 = 0 Then
        If Tengah2 = 0 Then
          If Kanan1 = 1 Then

            Akhir_r

          End If
        End If
      End If
    End If
  End If
```

```

End If
  End If
    End If
      End If
        End If

```

'----- Mode 4-----

'----- 1 -----

```

If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then

```

Kanan_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

'----- 2 -----

```

If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then

```

Kanan_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

'----- 3 -----

```

If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then

```

Kiri_pelan_r

```

End If

```

```

End If
  End If
    End If
      End If

```

----- 4 -----

```

If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then

```

Kiri_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

----- Mode 5 -----

----- 1 -----

```

If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then

```

Maju_cepat_r

```

End If
  End If
    End If
      End If
        End If

```

----- 2 -----

```

If Kiri1 = 0 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then

```

Maju_cepat_r

```

End If
  End If
    End If
      End If
        End If

```

```

----- 3 -----
    If Kiril = 1 Then
      If Kiri2 = 0 Then
        If Tengah1 = 0 Then
          If Tengah2 = 0 Then
            If Kanan1 = 0 Then

              Kiri_cepat_r

            End If
          End If
        End If
      End If
    End If

```

```

----- 4 -----
    If Kiril = 1 Then
      If Kiri2 = 0 Then
        If Tengah1 = 0 Then
          If Tengah2 = 0 Then
            If Kanan1 = 1 Then

              Akhir_r

            End If
          End If
        End If
      End If
    End If

```

```

----- Mode 6 -----

```

```

----- 1 -----
    If Kiril = 1 Then
      If Kiri2 = 0 Then
        If Tengah1 = 0 Then
          If Tengah2 = 1 Then
            If Kanan1 = 0 Then

              Kanan_pelan_r

            End If
          End If
        End If
      End If
    End If

```

```

----- 2 -----
    If Kiril = 1 Then

```

```

If Kiri2 = 0 Then
  If Tengah1 = 0 Then
    If Tengah2 = 1 Then
      If Kanan1 = 1 Then

```

Kanan_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

----- 3 -----

```

If Kiril = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then

```

Kiri_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

----- 4 -----

```

If Kiril = 1 Then
  If Kiri2 = 0 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then

```

Kiri_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

----- Mode 7 -----

----- 1 -----

```

If Kiril = 1 Then
  If Kiri2 = 0 Then

```

```
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then
```

```
          Maju_cepat_r
```

```
        End If
      End If
    End If
  End If
End If
```

```
----- 2 -----
  If Kiri1 = 1 Then
    If Kiri2 = 0 Then
      If Tengah1 = 1 Then
        If Tengah2 = 1 Then
          If Kanan1 = 1 Then
```

```
            Maju_cepat_r
```

```
          End If
        End If
      End If
    End If
  End If
```

```
----- 3 -----
  If Kiri1 = 1 Then
    If Kiri2 = 1 Then
      If Tengah1 = 0 Then
        If Tengah2 = 0 Then
          If Kanan1 = 0 Then
```

```
            Kiri_sedang_r
```

```
          End If
        End If
      End If
    End If
  End If
```

```
----- 4 -----
  If Kiri1 = 1 Then
    If Kiri2 = 1 Then
      If Tengah1 = 0 Then
        If Tengah2 = 0 Then
          If Kanan1 = 1 Then
```

Akhir_r

```
End If
  End If
    End If
      End If
        End If
```

'----- Mode 8-----'

'----- 1-----'

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then
```

Kanan_pelan_r

```
End If
  End If
    End If
      End If
        End If
```

'----- 2-----'

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 0 Then
      If Tengah2 = 1 Then
        If Kanan1 = 1 Then
```

Kanan_pelan_r

```
End If
  End If
    End If
      End If
        End If
```

'----- 3-----'

```
If Kiri1 = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 0 Then
```

Kiri_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

----- 4 -----

```

If Kiril = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 0 Then
        If Kanan1 = 1 Then

```

Kiri_pelan_r

```

End If
  End If
    End If
      End If
        End If

```

----- Mode 9 -----

----- 1 -----

```

If Kiril = 1 Then
  If Kiri2 = 1 Then
    If Tengah1 = 1 Then
      If Tengah2 = 1 Then
        If Kanan1 = 0 Then

```

Maju_cepat_r

```

End If
  End If
    End If
      End If
        End If

```

----- 2 -----

```

If Kiril = 1 Then
  If Kiri2 = 1 Then

```

```
If Tengah1 = 1 Then
  If Tengah2 = 1 Then
    If Kanan1 = 1 Then
```

```
      Maju_cepat_r
```

```
    End If
  End If
End If
  End If
End If
```

```
End Sub
```

```
----- Sub Maju Cepat -----
```

```
Sub Maju_cepat_r
```

```
  Motor_kanan = 1
  Pwm_kanan = 255
```

```
  Motor_kiri = 1
  Pwm_kiri = 255
```

```
  Cls
  Locate 1 , 5
  Lcd "Maju Cepat"
```

```
  Lowerline
  Locate 2 , 6
  Lcd "PWM Full"
```

```
End Sub
```

```
----- Sub Kiri Cepat -----
```

```
Sub Kiri_cepat_r
```

```
  Motor_kanan = 1
  Pwm_kanan = 250
```

```
Motor_kiri = 0
Pwm_kiri = 245
```

```
Cls
Locate 1 , 5
Lcd "Kiri Cepat"
```

```
Lowerline
Locate 2 , 2
Lcd "R = 250 L = 245"
```

```
End Sub
```

```
'----- Sub Kanan Sedang -----
```

```
Sub Kanan_sedang_r
```

```
Motor_kanan = 0
Pwm_kanan = 240
```

```
Motor_kiri = 1
Pwm_kiri = 245
```

```
Cls
Locate 1 , 4
Lcd "Kanan Sedang"
```

```
Lowerline
Locate 2 , 2
Lcd "R = 240 L = 245"
```

```
End Sub
```

```
'----- Sub Kiri Sedang-----
```

```
Sub Kiri_sedang_r
```

```
Motor_kanan = 1
Pwm_kanan = 245
```

```
Motor_kiri = 0
Pwm_kiri = 240
```

```
Cls
Locate 1 , 4
Lcd "Kiri Sedang"
```

```
Lowerline
Locate 2 , 2
Lcd "R = 245 L = 240"
```

End Sub

'----- Sub Kanan Pelan -----'

Sub Kanan_pelan_r

Motor_kanan = 0
Pwm_kanan = 230

Motor_kiri = 1
Pwm_kiri = 235

Cls
Locate 1 , 4
Lcd "Kanan Pelan"

Lowerline
Locate 2 , 2
Lcd "R = 230 L = 235"

End Sub

'----- Sub Kiri Pelan -----'

Sub Kiri_pelan_r

Motor_kanan = 1
Pwm_kanan = 235

Motor_kiri = 0
Pwm_kiri = 230

Cls
Locate 1 , 5
Lcd "Kiri Pelan"

Lowerline
Locate 2 , 2
Lcd "R = 235 L = 230"

End Sub

'----- Sub Belok Kanan -----'

Sub Belok_kanan_r

```
Motor_kanan = 0  
Pwm_kanan = 0
```

```
Motor_kiri = 1  
Pwm_kiri = 240
```

```
Cls  
Locate 1 , 5  
Lcd "Turn Right"
```

```
Lowerline  
Locate 2 , 3  
Lcd "R = 0 L = 240"
```

```
End Sub
```

```
----- Sub Belok Kiri -----
```

```
Sub Belok_kiri_r
```

```
Motor_kanan = 1  
Pwm_kanan = 240
```

```
Motor_kiri = 0  
Pwm_kiri = 0
```

```
Cls  
Locate 1 , 5  
Lcd "Turn Left"
```

```
Lowerline  
Locate 2 , 3  
Lcd "R = 240 L = 0"
```

```
End Sub
```

```
----- Sub Rem Dikit -----
```

```
Sub Rem_dikit_r
```

```
Motor_kanan = 0  
Pwm_kanan = 255
```

```
Motor_kiri = 0  
Pwm_kiri = 255
```

```
Cls  
Locate 1 , 8  
Lcd "Rem"
```

```
Lowerline
Locate 2 , 3
Lcd "Kelembaman = 0"
```

```
End Sub
```

```
'----- Sub Putar Balik -----
```

```
Sub Putar_balik_r
```

```
Motor_kanan = 0
Pwm_kanan = 0
```

```
Motor_kiri = 1
Pwm_kiri = 240
```

```
Cls
Locate 1 , 4
Lcd "Putar Balik"
```

```
Lowerline
Locate 2 , 3
Lcd "R = 0 L = 240"
```

```
End Sub
```

```
'----- Sub Akhir -----
```

```
Sub Akhir_r
```

```
Motor_kanan = 0
Pwm_kanan = 0
```

```
Motor_kiri = 0
Pwm_kiri = 0
```

```
Cls
Locate 1 , 8
Lcd "Stop"
```

```
Lowerline
Locate 2 , 6
Lcd "PWM = 0"
```

```
End Sub
```

Lampiran

Foto Robot

