

High Speed Line Follower

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh

Gelar Sarjana Teknik Elektro



Disusun oleh :

Nama : Puguh Jatmiko Krisna Aji

No.Mahasiswa : 04524069

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2012

LEBAR PENGESAHAN PENGUJI

HIGH SPEED LINE FOLLOWER

Tugas Akhir

Oleh :

Nama : Puguh Jatmiko Krisna Aji

No. Mahasiswa : 04524069

Telah Di Pertahankan Didepan Sidang Penguji Sebagai Salah Satu syarat

Untuk Memperoleh Gelar Sarjana Teknik Elektro

Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Januari 2012

Tim Penguji

Ketua

Medilla Kusriyanto, ST., M.Eng.

.....

Anggota I

Ir. Hj. Budi Astuti, MT.

.....

Anggota II

Tito Yuwono , ST., M.Sc.

.....

Mengetahui :

Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri

Universitas Islam Indonesia

Tito Yuwono , ST.,M.Sc.

LEMBAR PENGESAHAN PEMBIMBING

HIGH SPEED LINE FOLLOWER

TUGAS AKHIR



Oleh:

Nama : Puguh Jatmiko Krisna Aji

No. Mahasiswa : 04524069

Yogyakarta, November 2011

Menyetujui,

Pembimbing

Medilla Kusriyanto S.T, M.Eng.

ABSTRAK

Secara sederhana, robot high speed line follower adalah robot yang dapat bergerak mengikuti garis secara otomatis. Kemajuan ilmu pengetahuan dan teknologi telah mendorong manusia untuk berusaha mengatasi segala permasalahan yang timbul di sekitarnya. Keterbatasannya untuk terjun langsung melakukan proses pengendalian karena jauh, banyaknya plant yang dikendalikan dan membutuhkan waktu yang lama, lambat laun telah dapat diatasi dengan ditemukannya teknologi-teknologi baru, salah satunya adalah sistem gerak pada Robot pengikut garis (Line Follower). Robot ini menggunakan ATmega16 sebagai sistem pengendali utama dengan algoritma PID didalamnya. Kendali ini di dukung mux 4051 dan 12 sensor garis. Cara kerja robot ini adalah sensor membaca adanya garis yang kemudian di proses didalam mikrokontroler untuk kemudian menjalankan motor dc yang terhubung pada driver motor. Selanjutnya kecepatan motor di kontrol oleh algoritma PID yang tertanam pada mikrokontroler. Hasil yang di peroleh dari perancangan robot ini adalah sistem dapat bekerja dinamis antara sensor, mikrokontroler, dan driver motor.

Keyword : PID, mux 4051, Line detector

MOTTO

كُتِبَ عَلَيْكُمُ الْقِتَالُ وَهُوَ كُرْهُ لَكُمْ وَعَسَىٰ أَن تَكْرَهُوا شَيْئًا
وَهُوَ خَيْرٌ لَّكُمْ وَعَسَىٰ أَن تُحِبُّوا شَيْئًا وَهُوَ شَرٌّ لَّكُمْ وَاللَّهُ يَعْلَمُ
وَأَنْتُمْ لَا تَعْلَمُونَ ﴿٢١٦﴾

Diwajibkan atas kamu berperang, padahal berperang itu adalah sesuatu yang kamu benci. Boleh jadi kamu membenci sesuatu, padahal ia amat baik bagimu, dan boleh jadi (pula) kamu menyukai sesuatu, padahal ia amat buruk bagimu; Allah mengetahui, sedang kamu tidak mengetahui.

(Q.S. Al Baqarah :216)

وَلَوْ أَنَّمَا فِي الْأَرْضِ مِنْ شَجَرَةٍ أَقْلَمٌ وَالْبَحْرُ يَمُدُّهُ مِنْ بَعْدِهِ سَبْعَةُ
أَبْحُرٍ مَا نَفِدَتْ كَلِمَاتُ اللَّهِ إِنَّ اللَّهَ عَزِيزٌ حَكِيمٌ ﴿٢٧﴾

Dan seandainya pohon-pohon di bumi menjadi pena dan laut, ditambahkan kepadanya tujuh laut sesudahnya, niscaya tidak akan habis-habisnya kalimat Allah. Sesungguhnya Allah Maha Perkasa lagi Maha Bijaksana.

(QS. Al Lukman 27)

SELANGKAH LEBIH MAJU DALAM ILMU AMALIAH DAN AMAL ILMIAH

HALAMAN PERSEMBAHAN

*Kupersembahkan karya ini untuk :
Orang tua dan keluargaku tercinta
yang telah mencurahkan kasih sayang, do'a dan dukungan_nya
yang telah mengajarkan keseimbangan dalam diriku
kesungguhan, ketekunan, kesabaran, welas asih dan cara mencintai makhluk
yang menjadikanku merasa bersyukur diantara kelebihan dan kekuranganku
yang tetap membuatku ingat untuk selalu menunduk dihadapan – Nya
Semoga Allah SWT mencatatnya sebagai amal kebajikan.*

KATA PENGANTAR



Assalamu'alaikum warahmatullahi wabarakatuh,

Alhamdulillahirabbil'alamin, segala puji syukur saya panjatkan kehadirat Allah SWT yang telah melimpahkan segala karunia-Nya, sehingga saya dapat menyelesaikan Tugas Akhir ini sebagai salah satu persyaratan program S-1 Jurusan Teknik Industri, Fakultas Teknik Elektro, Universitas Islam Indonesia. Shalawat dan salam semoga tercurah pada junjungan kita Rasulullah Muhammad SAW beserta keluarga, dan pengikutnya yang setia hingga akhir zaman.

Merupakan suatu kelegaan ketika akhirnya saya dapat menyelesaikan bagian akhir dari amanah yang panjang ini dengan begitu banyak kemudahan dan kekuatan yang Allah berikan melalui banyak pihak dengan bantuan, dukungan dan do'a. Untuk itulah, saya sangat ingin menghaturkan terima kasih kepada :

1. Bapak dan Ibu tercinta yang telah memberikan do'a serta dukungan, materi sehingga dapat menyelesaikan dan menyusun Tugas Akhir ini. Bapak, Ibu.... Puguh minta maaf.
2. Bapak Medilla Kusriyanto, ST, M.Eng selaku dosen pembimbing yang telah menuntun bukan menuntut setiap proses bimbingan dalam penyusunan Tugas Akhir ini.
3. Bapak Tito Yuwono, ST, M.Sc selaku ketua jurusan Teknik Elektro Universitas Islam Indonesia.

4. Dekan fakultas Teknologi Industri Universitas Islam Indonesia.
5. Saudara-saudaraku, M. Khoirul Musta'an, Edi Suseno, Ilham Arto Suryo, Alvin Syahroni, Asep Juliansyah, Muhamad Hafizh, Handi Alaina, Muhamad Azra'i Bin Bahrum, Muhamad Wahyudi, Septa Aris Prasetya, Hanung, Habib Sholeh dan teman2 lainnya yang tidak bisa saya sebut satu persatu, terima kasih banyak kawan atas semua fasilitas-fasilitas yang telah kalian semua berikan. Semoga Allah SWT membalas budi baik kalian, dilapangkan rizkinya. Amin.
6. Saudaraku Wahyu Tri Utomo dan Sudiyono teima kasih atas waktu dan belajar kilatnya.
7. Terkhusus Siti Kamsiyah beserta keluarga. Kalian juga mengajarkan untuk selalu menjadi orang yang mawas diri.

Saya menyadari sepenuhnya bahwa tugas akhir ini memiliki banyak kekurangan dan masih jauh dari idealisme seorang peneliti maupun karya ilmiah. Untuk itulah saya meminta maaf yang setulus-tulusnya dan menunggu masukan yang berguna untuk perbaikan selanjutnya.

Akhir kata, semoga tugas akhir ini dapat memberi manfaat bagi para pembacanya.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Yogyakarta,

Puguh Jatmiko Krisna Aji

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
MOTTO	iv
KATA PENGANTAR	vi
ABSTRAK	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	xii
LEMBAR PENGESAHAN PEMBIMBING	xiii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penulisan	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA.....	
2.1 Pendahuluan.....	6
2.2 Sensor Photodiode	9
2.3 Infra Merah (LED).....	11
2.4. Mikrokontroler ATmega16	13
2.4.1 Deskripsi PinATmega16	16
2.5 Kontrol PID (<i>Proportional-Integral-Derivative</i>)	19
BAB III PERANCANGAN SISTEM	
3.1 Perencanaan	22
3.2 Sistem Sensor.....	27
3.3 Mikrokontroler ATmega16	28
3.4 Driver Penggerak Motor DC	30
3.5 Rangkaian Regulator	31
3.6 Perancangan dan Pembuatan Perangkat Lunak	32
3.6.1 Algoritma Kontrol PID	31
3.6.2 Algoritma Kontrol PID	33
BAB IV PENGUJIAN DAN ANALISIS KINERJA SISTEM	
4.1 Metode Pengujian	38
4.2 Pengujian dan Analisa Sistem Kerja Sensor.....	38
4.3 Pengujian dan Analisa Rangkaian Mikrokontroler	40
4.4 Pengujian dan Analisa Rangkaian Driver Motor	41

4.5	Pengujian dan Analisa Motor DC.....	42
BABV PENUTUP.....		
5.1	Kesimpulan	44
5.2	Saran-saran	45
DAFTAR PUSTAKA		
LAMPIRAN.....		

DAFTAR TABEL

	Halaman
2.1 Fungsi Khusus PORT B ATmega16.....	17
2.2 Fungsi Khusus PORT C ATmega16.....	17
2.3 Fungsi Khusus PORT D ATmega16	18
4.1 Hasil Pengujian Tegangan Pada Rangkaian Sensor Depan	39
4.2 Hasil Pengujian Tegangan Pada Rangkaian Sensor Samping Kanan	39
4.3 Hasil Pengujian Tegangan Pada Rangkaian Sensor Samping Kiri	40
4.4 Hasil Pengujian Pada Rangkaian Driver Motor	42
4.4 Hasil Pengujian Pada Rangkaian Driver Motor DC.....	43
4.5 Hasil Pengujian RPM Pada Motor DC.....	43

DAFTAR GAMBAR

	Halaman
2.1 Photodioda	10
2.2 kurva Karakteristik Photodioda	10
2.3 Rangkaian Pengkondisi Dari Photodioda	11
2.4 Infra Merah (LED).....	12
2.5 Block Diagram Fungsional ATmega16	15
2.6 Konfigurasi Pin ATmega16	16
3.1 Rancangan Robot Tamapak Atas.....	21
3.2 Rancangan Robot tampak Samping.....	22
3.3 Block Diagram <i>High Speed Line Follower</i>	22
3.4 <i>Flowchart High Speed Line Follower</i>	23
3.5 Konfigurasi posisi sensor.....	24
3.6 Posisi Sensor Pada Lintasan	25
3.7 Skema Rangkaian Sensor Depan	26
3.8 Skema Rangkaian Sensor Samping	27
3.9 Skema Rangkaian Sistem Minimum ATmega16.....	28
3.10 Skema Rangkaian Mosfet.....	29
3.11 Skema Rangkaian LM2576	31
3.12 Halaman Editor Bascom AVR	32
3.13 Bar Fungsi Pada Bascom AVR	32
3.14 Bar Fungsi Pada File.....	33
3.15 Bar Fungsi Pada Edit	34
3.16 Bar fungsi Pada Program.....	35
3.17 Bar Fungsi Pada Tools.....	36
3.18 Bar Fungsi Pada Options	37
3.19 Bar Fungsi Pada Menu Compiler	38
4.1 Konstruksi Motor Dc	42

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Secara sederhana, robot line follower adalah robot yang dapat bergerak mengikuti garis secara otomatis. Kemajuan ilmu pengetahuan dan teknologi telah mendorong manusia untuk berusaha mengatasi segala permasalahan yang timbul di sekitarnya. Keterbatasannya untuk terjun langsung melakukan proses pengendalian karena jauh, banyaknya plant yang dikendalikan dan membutuhkan waktu yang lama, lambat laun telah dapat diatasi dengan ditemukannya teknologi-teknologi baru, salah satunya adalah sistem gerak pada Robot pengikut garis (Line Follower). Sistem penggerak ini menggunakan motor DC digunakan untuk menjalankan Robot Pengikut Garis (Line Follower). Motor DC merupakan penggerak utama Robot ini. Prinsip dasar dari motor arus searah adalah jika sebuah kawat berarus diletakkan melintang di antara dua kutub magnet, maka pada kawat itu akan bekerja suatu gaya yang menggerakkannya. Dengan ini, kecepatan putar dari motor DC dapat diatur sesuai kebutuhan. Dalam hal ini dikehendaki perlambatan. Jadi motor DC akan bergerak sesuai dengan garis yang telah ditentukan, dan akan bergerak ketika ada masukan dari sensor photodiode.

1.2 Rumusan Masalah

Bagaimana melakukan perancangan dan realisasi sistem gerak robot line follower menggunakan motor DC berbasis mikrokontroller Atmega16 dengan sensor photodioda sehingga dihasilkan kecepatan maksimum robot line follower.

1.3 Batasan Masalah

Berikut adalah hal-hal yang menjadi batasan masalah pembuatan Tugas Akhir ini:

1. Mikrokontroler yang digunakan adalah ATmega16 sebagai sistem pemrosesan kontrol untuk semua sistem.
2. Algoritma pemrograman control yang digunakan adalah PID.
3. Sensor yang digunakan adalah photodioda 5mA sebanyak 12 buah dengan LED sebagai transmitter.

1.4 Tujuan Penulisan

Tujuan yang akan dicapai dalam penulisan Tugas Akhir ini adalah membangun sebuah mekanisme robot line follower untuk dapat menghasilkan kecepatan robot *line follower* yang maksimum.

1.5 Metodologi Penelitian

Metodologi Penelitian dalam tugas akhir ini akan dilakukan dengan tahapan berikut :

1. Studi literatur, yaitu mengumpulkan data dengan cara mempelajari literatur/buku, artikel yang berkaitan dengan topik tugas akhir. Termasuk artikel dan data dari internet.
2. Perancangan, yaitu merancang sistem berdasarkan teori yang diperoleh saat studi literatur dan menguji sistem agar dapat dianalisis dan diambil kesimpulan.
3. Merancang *software* dengan algoritma kontrol PID.
4. Analisis hasil yang telah dilakukan.

1.6 Sistematika Penulisan

Dari hasil penelitian yang telah dilaksanakan, sistematika penulisan laporannya adalah sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan berisi tentang Latar Belakang Masalah, Maksud dan Tujuan, Batasan Masalah, Perumusan Masalah, Metode Penelitian, serta Sistematika Penulisan pada Laporan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab studi pustaka membahas tentang penelitian yang telah dilakukan sebelumnya dan penjelasan secara terperinci mengenai teori-teori yang digunakan sebagai landasan untuk pemecahan masalah.

BAB III PERANCANGAN SISTEM

Berisi pembahasan tentang perancangan perangkat keras yang meliputi sensor sebagai media pendeteksi lintasan, mikrokontroler sebagai kendali utamanya dan program untuk mengontrol aksi gerak mekanis robot di lintasan.

BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN

Bagian ini menjelaskan tentang pengujian dan analisa terhadap hasil pengamatan dari sistem yang telah dibuat.

BAB V PENUTUP

Bab penutup membahas tentang kesimpulan dan saran-saran yang menunjang agar alat tersebut dapat berkerja lebih optimal, sehingga tugas akhir ini dapat dikembangkan lebih lanjut, dengan harapan dapat digunakan untuk mendukung penerapannya dalam kehidupan masyarakat

BAB II

TINJAUAN PUSTAKA

2.1 Pendahuluan

Eko Prasetyo, 2010. “SISTEM GERAK ROBOT LINE FOLLOWER MENGGUNAKAN MOTOR DC BERBASIS MIKROKONTROLER ATmega8535 DENGAN SENSOR PHOTODIODA” Sistem yang dirancang terdiri atas perangkat keras dan perangkat lunak. Perangkat keras terdiri atas mikrokontroler Atmega8535, rangkaian LCD (*Liquid Crystal Display*), motor DC dan *driver* motor DC IC L293D, sensor photodiode. Perangkat lunak mikrokontroler dalam penelitian ini dibuat dengan menggunakan bahasa *c*. Sistem ini bekerja setelah ditekannya tombol pengatur pada mikrokontroler kemudian mikrokontroler menerima data dari pengatur tersebut, dari mikrokontroler akan mengirim data ke *driver* motor DC L293D. Setelah mengirim data ke *driver* motor DC, mikrokontroler mengontrol sistem gerak dari motor DC. Dan mikrokontroler akan mengirim data ke LCD dan akan ditampilkan dalam bentuk karakter huruf dan angka. Sistem ini telah terealisasi dan dapat menggerakkan motor DC sesuai dengan kecepatan yang diinginkan. LCD menampilkan hasil kecepatan motor DC yang bergerak serta mendeteksi sensor yang bekerja.

Muhammad Azhar Alfarezell, 2010. “Robot Line Follower Analog” Dalam penelitian ini akan dibahas mengenai robot *Line Follower*. Robot ini merupakan

salah satu bentuk robot beroda yang memiliki komponen utama diantaranya, seperti resistor, dioda, transistor, Led yang dirangkai untuk menghasilkan jenis kendaraan yang berjalan secara otomatis dengan kecepatan tertentu mengikuti garis. Di dalam rangkaian *Line Follower* terdapat 3 bagian utama, yaitu bagian sensor, komparator dan *driver*. Untuk bagian sensor digunakan photodioda sebagai sensor cahaya, sedangkan komparatornya menggunakan IC LM 324 sebagai pembanding tegangan dan untuk drivernya digunakan 2 buah motor sebagai penggerak rodanya. Hasil uji coba rangkaian *Line Follower* ini menunjukkan performa yang mampu berjalan di beberapa medan, diantaranya medan lurus, belok, naik, dan menurun.

Penerapan robot *High Speed Line follower* robot ini sendiri dalam kehidupan nyata adalah untuk pengendalian kereta api listrik dalam kota. Kereta api ini dapat berjalan dengan cepat mengikuti lintasan garis yang di lalukannya.

Adapun dasar pengoperasian *High Speed Line follower* robot secara lengkap adalah sebagai berikut :

1. Untuk membaca garis, robot dilengkapi dengan sensor optik yang diletakkan di ujung depan dari robot tersebut. Sensor merupakan suatu piranti elektronika yang berfungsi untuk mengubah besaran-besaran fisik yang ada di alam menjadi besaran elektrik yang dapat dimengerti oleh rangkaian elektronika.

Dari sudut pandang robot, sensor dapat diklasifikasikan dalam dua kategori, yaitu sensor lokal (*on-board*) yang dipasang di tubuh robot, dan

sensor global yaitu sensor yang diinstal diluar robot tapi masih dalam lingkungannya dan data sensor global ini dikirim balik ke robot melalui komunikasi nirkabel.

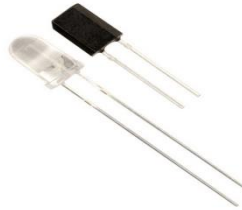
Dalam perancangan sebuah *high speed line follower* robot, sensor merupakan salah satu bagian sistem terpenting. Karena kemampuan robot untuk mengikuti garis, akan tergantung pada aktivitas dan sensitifitas sensornya. Sensor *line follower* robot biasanya menggunakan sensor intensitas cahaya yang difungsikan untuk mendeteksi adanya garis putih pada lapangan dengan warna hitam ataupun mendeteksi garis hitam pada alas berwarna putih. Alasan penggunaan sensor intensitas cahaya yaitu pertimbangan kemudahan pembacaan garis oleh sensor melalui pantulan cahaya yang diterimanya.

Line follower robot biasanya menggunakan sensor intensitas cahaya yang difungsikan untuk mendeteksi adanya garis putih pada lapangan dengan warna hitam ataupun mendeteksi garis hitam pada alas berwarna putih. Alasan penggunaan sensor intensitas cahaya yaitu pertimbangan kemudahan pembacaan garis oleh sensor melalui pantulan cahaya yang diterimanya. *Photo-reflectors*, *photo-transsistors* ataupun *photo-dioda* merupakan beberapa contoh sensor yang menggunakan intensitas cahaya dan biasa digunakan pada rangkaian sensor *line follower* robot. Kesemuanya pada dasarnya menggunakan prinsip *infra red* atau pantulan dari led.

2. Untuk mengendalikan robot diatas *track*, digunakan beberapa pengendali mekanik, dan yang digunakan disini digunakan motor DC sebagai penggeraknya kemudian menggunakan sebuah pengontrol untuk mengendalikan motor tersebut dengan algoritma dan aturan yang disesuaikan pula.
3. Pengendalian kecepatan sangat bergantung pada batas putaran dan pergesekan antara ban robot dengan lantainya. Agar sebuah robot dapat dikatakan sebagai *high speed* maka motor dc yang digunakan harus mempunyai nilai diatas 200 rpm.
4. Ada dua jenis garis yang mampu dibaca oleh *Line follower* robot. Garis putih dan garis hitam. Sesuai dengan *setting* yang ditentukan. Biasanya lebarnya berkisar antara 15 – 25 mm.

2.2 Sensor Photodioda

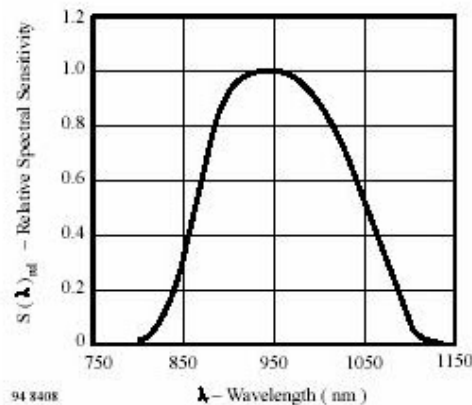
Dioda yang terbuat dari semikonduktor dengan bahan yang populer adalah silikon atau *gallium arsenide*, dan lain-lain termasuk InSb, *Carbuncle*, PbSe. Photodioda Ini (BPW41N) dapat mendeteksi cahaya infra merah dengan panjang gelombang berkisar 800-1100 nm dengan sensitivitas puncak pada panjang gelombang 950 nm sebagaimana ditunjukkan pada gambar di bawah.



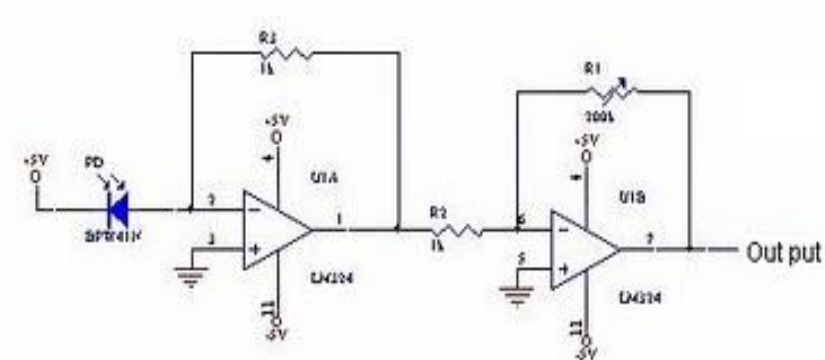
Gambar 2.1 Photodioda

Photodioda ini sering digunakan sebagai penangkap gelombang cahaya yang dipancarkan oleh Infra merah. Jumlah arus tegangan atau listrik yang dihasilkan oleh ukuran tergantung radiasi yang dipancarkan oleh photodioda infra merah.

Sebuah *foton* (unit energi cahaya) dari sumber cahaya diserap, hal ini menghasilkan elektron dan menghasilkan sepasang pembawa muatan tunggal. Arah arus yang melalui semikonduktor berlawanan dengan gerakan pembawa muatan. Arus yang berada dalam *Photodiode* digunakan untuk mengumpulkan *foton* (seperti arus atau tegangan) mengalir / terbentuk di bagian elektroda.



Gambar 2.2 Kurva karakteristik photodioda



Gambar 2.3 Rangkaian Pengkondisi Sinyal dari Photodioda

Rangkaian ini digunakan sebagai konverter arus ke tegangan dengan faktor dari 1000 kali (diatur oleh R3) karena berbagai photodioda arus balik dari 1 sampai 100 UA output dari U1A adalah berkisar dari -1 sampai -100 mV. Tegangan negatif ini kemudian diperkuat lagi oleh rangkaian penguat pembalik dengan nilai sesuai dengan tingkat input kebutuhan, jumlah penguatan dilakukan dengan menyesuaikan nilai R1 potensiometer.

2.3 Infra Merah (LED)

Infra merah ialah sinar elektromagnet yang panjang gelombangnya lebih dari cahaya nampak yaitu di antara 700 nm sampai 1 mm. Sinar infra merah merupakan cahaya yang tidak tampak. Jika dilihat dengan dengan *spektroskop* cahaya maka radiasi cahaya infra merah akan nampak pada *spectrum* elektromagnet dengan panjang gelombang di atas panjang gelombang cahaya merah.



Gambar 2.4 Infra Merah (LED)

Dengan panjang gelombang ini maka cahaya infra merah ini akan tidak tampak oleh mata, namun radiasi panas yang ditimbulkannya masih terasa/dideteksi.

Infra merah dapat dibedakan menjadi tiga daerah yakni:

- *Near* Infra Merah : 0,75 - 1,5 μm
- *Mid* Infra Merah : 1,5 - 10 μm
- *Far* Infra Merah : 10 - 100 μm

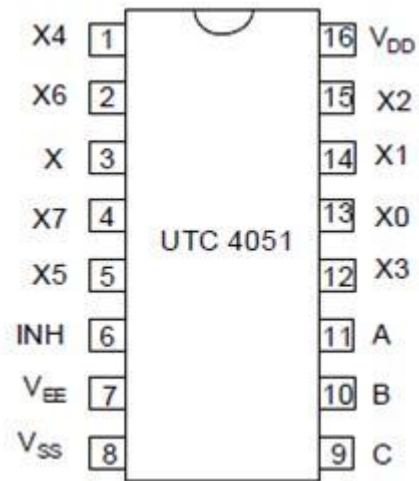
Contoh aplikasi sederhana untuk *far infra red* adalah terdapat pada alat-alat kesehatan. Sedangkan untuk *mid infra red* ada pada alat ini untuk sensor *alarm* biasa dan *near infra red* digunakan untuk pencitraan pandangan malam seperti pada *nightscoop*. Penggunaan infra merah sebagai media transmisi data mulai diaplikasikan pada berbagai peralatan seperti televisi, handphone sampai pada transfer data pada PC. Media infra merah ini dapat digunakan baik untuk kontrol aplikasi lain maupun transmisi data. Sifat-sifat cahaya infra merah:

1. Tidak tampak manusia
2. Tidak dapat menembus materi yang tidak tembus pandang
3. Dapat ditimbulkan oleh komponen yang menghasilkan panas
4. Panjang gelombang pada inframerah memiliki hubungan yang berlawanan atau berbanding terbalik dengan suhu. Ketika suhu mengalami kenaikan, maka panjang gelombang mengalami penurunan.

Komunikasi Infra merah dilakukan dengan menggunakan dioda infra merah sebagai pemancar dan modul penerima infra merah sebagai penerimanya. Untuk jarak yang cukup jauh, kurang lebih tiga sampai lima meter, pancaran data infra merah harus dimodulasikan terlebih dahulu untuk menghindari kerusakan data akibat *noise*.

2.4 Multiplexer

Multiplexer atau disingkat MUX adalah alat atau komponen elektronika yang bisa memilih input (masukan) yang akan diteruskan ke bagian output(keluaran). Pemilihan input mana yang dipilih akan ditentukan oleh signal yang ada di bagian kontrol (kendali) *Select*.

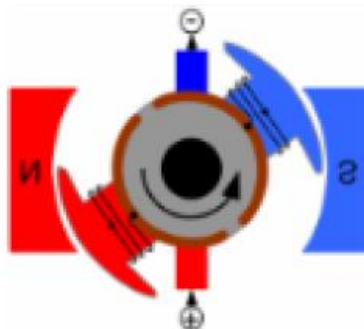


Gambar 2.5 Multiplexer 4051

IC 4051 dapat difungsikan sebagai multiplexer 8 input ke 1 output, dengan menggunakan 3 pin kontrol...yang terdapat pada pin 9, 10 dan 11.

2.5 Motor DC

Motor DC merupakan alat penggerak dari robot. Motor DC memiliki konstruksi yang cukup sederhana, dapat dilihat pada gambar di bawah ini:



Gambar 2.6 Konstruksi Motor DC

Cara kerja dari motor Dc ini adalah pada saat ada arus yang masuk kedalam motor melalui kutub-kutub motor maka akan terjadi momen kopel yang akan memberikan *force* (gaya) untuk memutar motor. Untuk arah putaran motor tergantung dari arah arus yang masuk ke motor.

2.6 Mikrokontroler ATmega16

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*, 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. ATmega16. ATmega16 mempunyai *throughput* mendekati 1MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

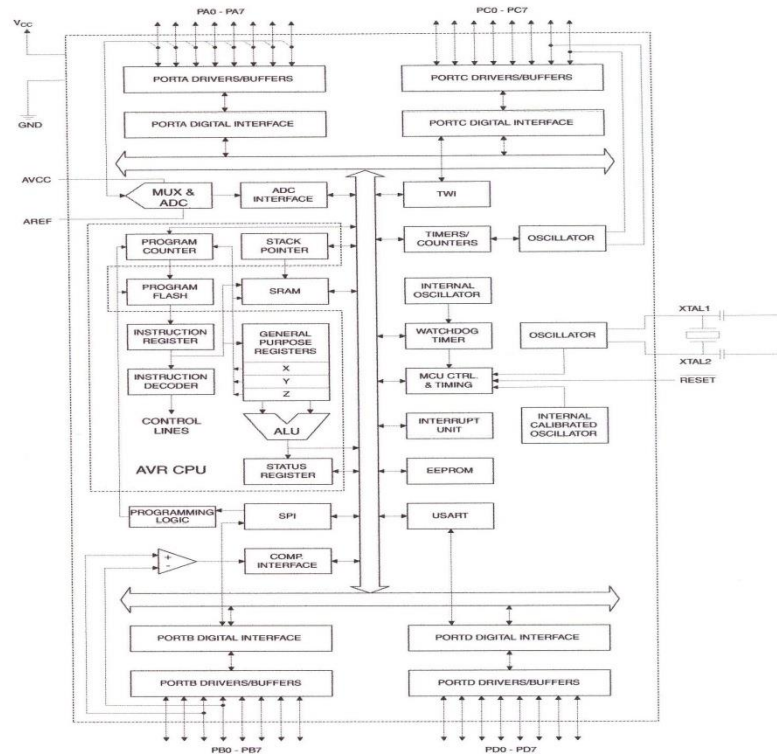
Fitur dasar yang dimiliki Mikrokontroler AVR ATmega16 adalah sebagai berikut :

Fitur-fitur pada mikrokontroler ATmega16 antara lain:

1. Saluran I/O ada 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
2. ADC 10 bit sebanyak 8 *channel*.

3. CPU yang terdiri dari 32 buah *register*.
4. *Watchdog Timer* dengan osilator internal
5. Dan fitur-fitur lain yang mempermudah dalam penggunaan.
6. Tegangan kerja berkisar 4-5 *Volt*.
7. Memori Flash 8 *Kbytes* untuk program
8. Memori EEPROM 512 *bytes* untuk data
9. Memori SRAM 512 *bytes* untuk data
10. 20 *interrupt*
11. Satu 16-bit *timer* dan dua 8-bit *timer*
12. Komunikasi serial melalui SPI dan USART
13. Analog komparator
14. 4 I/O PWM
15. Fasilitas *In System Programming* (ISP)

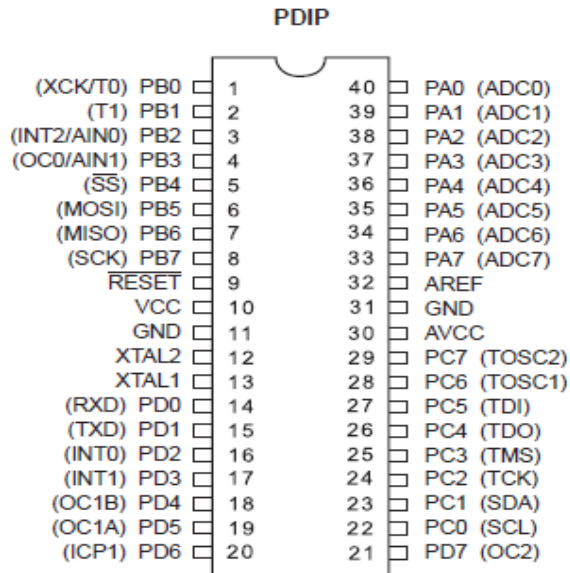
Dengan fitur-fitur seperti diatas, pembuatan alat menggunakan ATmega16 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak. Agar lebih jelasnya dapat dilihat pada gambar dibawah ini:



Gambar 2.7 Blok diagram Fungsional ATmega16

2.6.1 Deskripsi Pin ATmega16

ATmega16 memiliki 32 pin yang digunakan untuk input/output, pin-pin tersebut terdiri dari 8 pin sebagai *port A*. 8 pin sebagai *port B*. 8 pin sebagai *port C*. 8 pin sebagai *port D*. Dalam komunikasi serial, maka hanya *port D* yang dapat digunakan karena fungsi khusus yang dimilikinya. Untuk lebih jelas akan ditunjukkan pada tabel-tabel fungsi khusus *port*. Susunan pin Mikrokontroler ATmega16 diperlihatkan pada gambar berikut:



Gambar 2.8 Konfigurasi Pin ATmega16

- Pin 1 sampai 8 (Port B) merupakan *port* paralel 8 bit dua arah (*input/output*) dan pin fungsi khusus.

Tabel 2.1 Fungsi Khusus Port B ATmega16

Port Pin	Alternate Functions
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	\overline{SS} (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)

- Pin 9 (Reset) merupakan pin yang digunakan untuk mereset mikrokontroller.
- Pin 10 (VCC) merupakan pin masukan catu daya.
- Pin 11 & 31 (GND) merupakan pin ground.
- Pin 12 (XTAL) & Pin (XTAL1) merupakan pin masukan clock eksternal.
- Pin 14 sampai 21 (Port D) merupakan *port* parallel 8 bit dua arah (*input/output*) dan pin fungsi khusus

Tabel 2.2 Fungsi Khusus Port D ATmega16

Port Pin	Alternate Function
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

- Pin 22 sampai 29 (Port C) merupakan *port* parallel 8 bit dua arah (*input/output*) dan pin fungsi khusus.

Tabel 2.3 Fungsi Khusus Port C ATmega16

Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin 2)
PC6	TOSC1 (Timer Oscillator Pin 1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

- Pin 30 (AVCC) merupakan pin masukan tegangan untuk ADC.
- Pin 32 (AREF) merupakan pin masukan tegangan referensi ADC.
- Pin 33 sampai 40 (Port A) merupakan pin I/O dua arah dan pin masukan ADC.

2.7 Kontrol PID (*Proportional–Integral–Derivative*)

Algoritma pengontrolan yang akan di pakai untuk merancang robot *High Speed Line Follower* ini adalah algoritma control PID. Komponen control PID terdiri dari 3 jenis yaitu *Proportional*, *Integratif* dan *Derivatif* . ketiganya dapat dipakai bersamaan ataupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant*.

1. Kontrol Proporsional

Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka $u = K_p \cdot e$ dengan K_p adalah konstanta Proporsional. K_p berlaku sebagai Gain (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki *respon transien* khususnya *rise time* dan *setting time*.

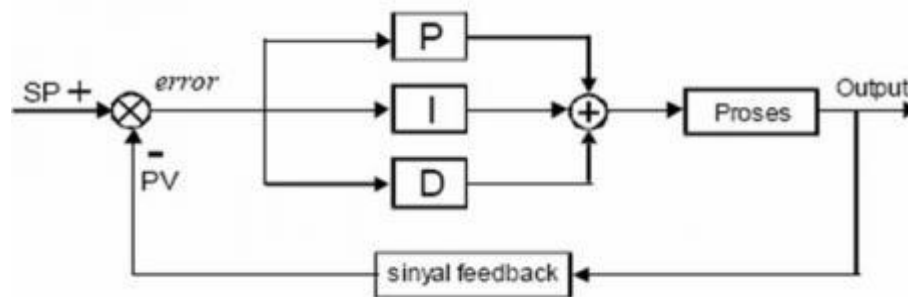
2. Kontrol Integratif jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = [\int e(t) dt] K_i$ dengan K_i adalah konstanta integral, dan dari persamaan diatas, $G(s)$ dapat dinyatakan sebagai $u = K_d \cdot \frac{de}{dt}$ jika $e(T)$ mendekati konstan (bukan nol) maka $u(T)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika $e(T)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon *steady-state*, namun memiliki K_i yang tidak tepat dapat menyebabkan respon *transien* yang tinggi sehingga dapat menyebabkan ketidak stabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi karena dapat menambah orde sistem

3. Kontrol Derivatif

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s) = s \cdot K_d$ Dari persamaan diatas, nampak bahwa sifat dari kontrol D ini dalam konteks “kecepatan” atau *rate* dari *error*. Dengan sifat ini ia dapat digunakan

untuk memperbaiki respon *transien* dengan memprediksi respon yang akan terjadi. Kontrol Derivatif hanya berubah saat ada perubahan *error* sehingga saat *error* statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivatif tidak dapat dipakai sendiri.

Setelah menggunakan kendali PID artinya kendali ini bertujuan mengolah suatu sinyal kesalahan atau error, nilai *error* tersebut diolah dengan formula PID untuk dijadikan suatu sinyal kendali atau sinyal kontrol yang akan diteruskan ke aktuator. Perancangan kendali PID pada robot *line follower* blok diagram umpan balik loop tertutup dapat di lihat pada gambar berikut ini:



Gambar 2.9 Blok Diagram Pada Kendali PID

Dari blok diagram diatas dapat di jelaskan sebagai berikut:

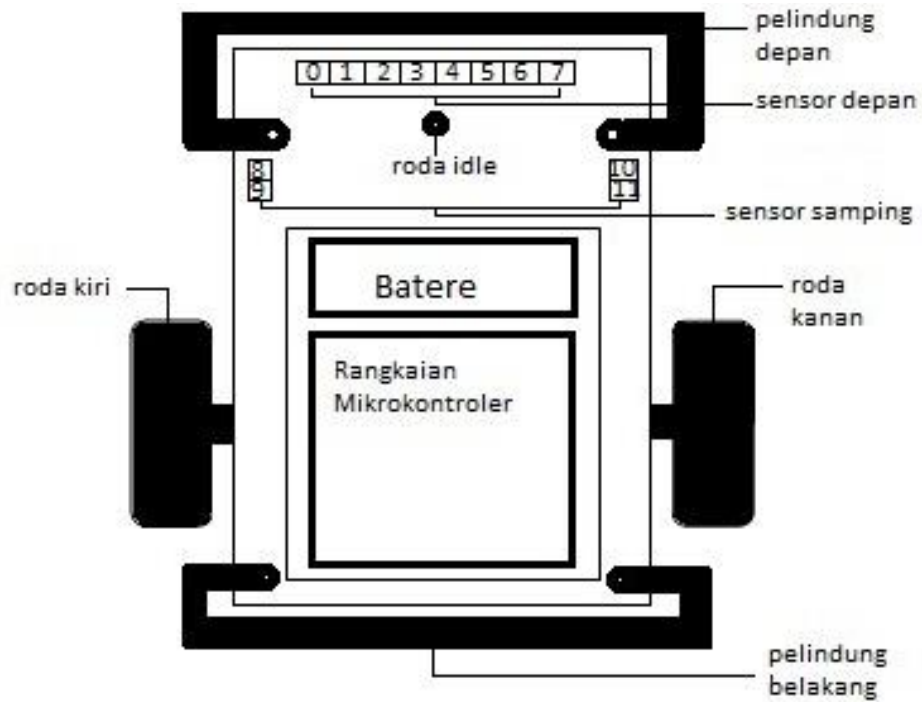
1. $SP = Set\ point$, secara sederhana, maksudnya ialah suatu parameter nilai acuan atau nilai yang di inginkan.
2. $PV = Present\ Value$, maksudnya ialah nilai bobot pembacaan sensor saat itu atau variabel terukur yang di umpan balikan oleh sensor (sinyal *feedback* dari sensor).
3. $Error =$ nilai kesalahan, ialah *Deviasi* atau simpangan antar variabel terukur atau bobot sensor (PV) dengan nilai acuan (SP)

BAB III

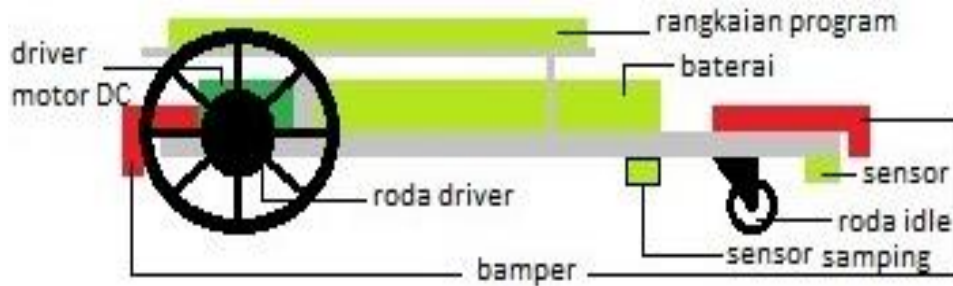
PERANCANGAN SISTEM

3.1 Perancangan

Rancangan robot *High Speed Line Follower* yang rencananya akan dirancang ditunjukkan pada gambar dibawah ini:



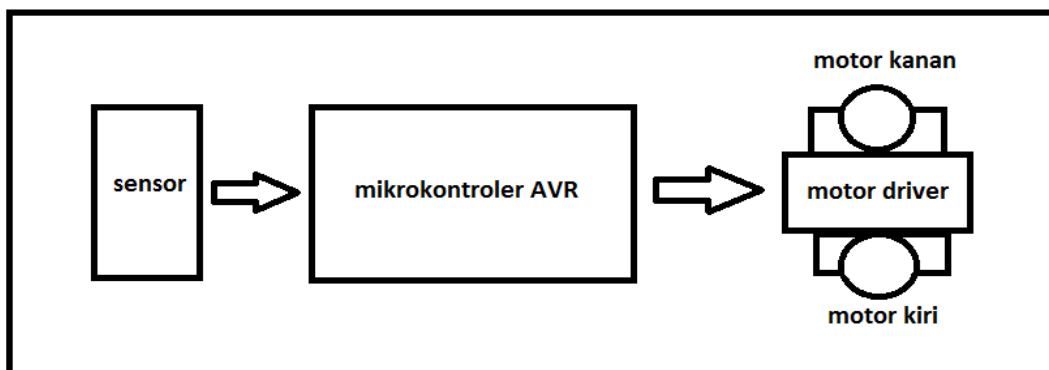
Gambar 3.1 Rancangan Robot Tampak Atas



Gambar 3.2 Rancangan Robot Tampak Samping

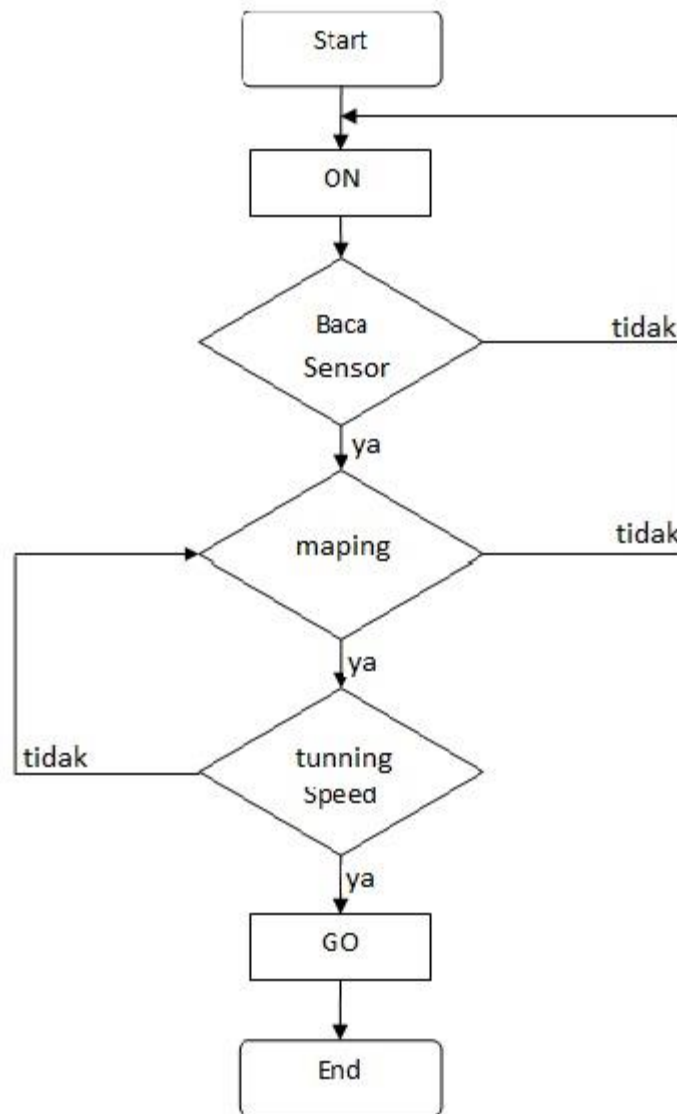
Konstruksi fisik suatu robot menjadi dasar tumpuan dari rangkaian elektronis dan *software* kontroler. Konstruksi fisik *high speed line follower robot* didesain untuk melakukan konsep atau tujuan untuk mengikuti garis. Sehingga hasil konstruksi fisik *high speed line follower robot* juga harus menyesuaikan dengan tujuan seperti tersebut di atas, seperti digunakannya roda karena dibutuhkan suatu perpindahan posisi robot.

Diagram blok robot *high speed line follower* secara keseluruhan dapat di lihat pada gambar berikut ini:



Gambar 3.3 Blok Diagram *high speed line follower*

Adapun sistem kerja alat di tunjukan pada *flowchart* di bawah ini:

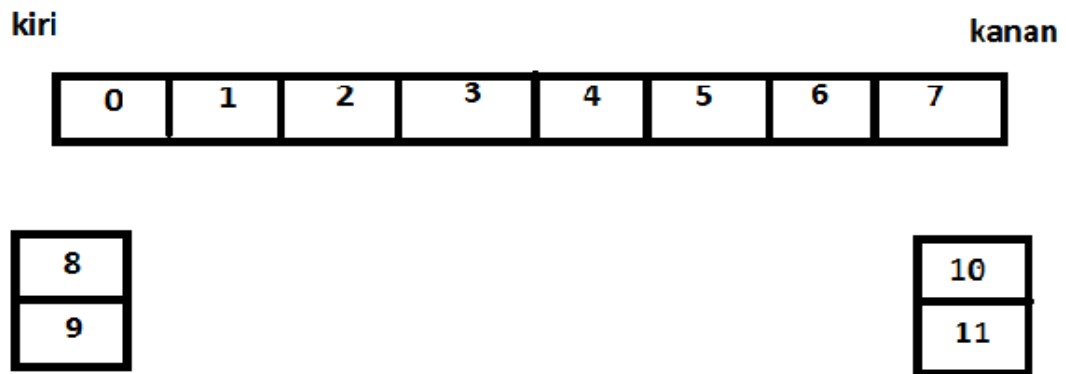


Gambar3.4 *Flowchart High Speed Line Follower*

Robot menggunakan sensor infra merah (IR) untuk mendeteksi garis pembimbing. Masing-masing sensor terdiri dari pasangan *infra red emitting diode* (IRED) sebagai pemancar (Tx) dan photodioda sebagai sensor (Rx).

Jumlah sensor IR yang dipasang terdiri dari 8 buah dipasang sejajar dan 2 buah disamping kanan dan kiri menghadap ke lantai. Keluaran dari sensor masih berupa sinyal analog yang bergantung dari jumlah pancaran sinar infra merah yang dipantulkan dan diterima oleh sensor photodiode. Sensor dipasang pada rangkaian pengkondisi sinyal yang berfungsi sebagai pembanding (komparator), untuk menghasilkan keluaran berupa logika “0” dan “1”. Keluaran dari rangkaian ini dihubungkan ke mikrokontroler ATmega16.

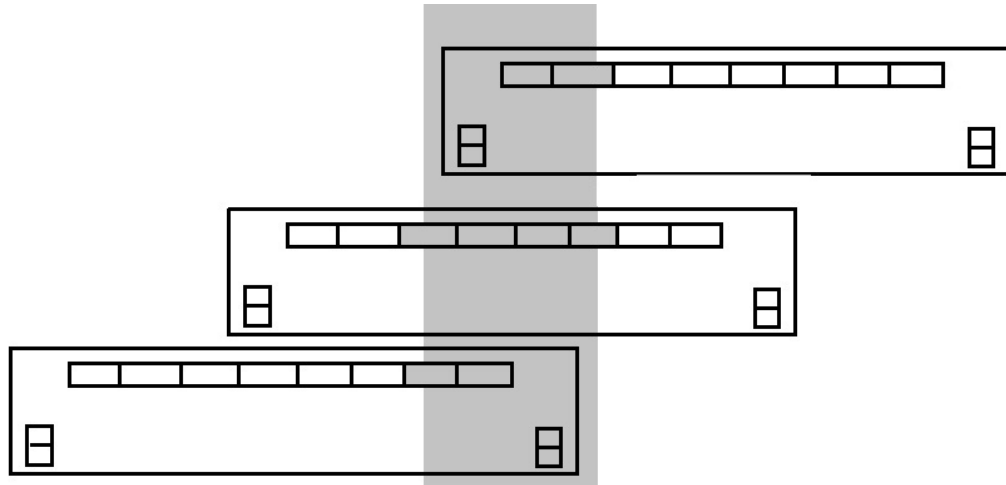
Konfigurasi pasangan sensor adalah sebagai berikut:



Gambar 3.5 Konfigurasi Posisi Sensor

Mulai dari kiri ke kanan, sensor diberi nama 0,1,2,3,4,5,6,7. Kemudian sensor 8,9,10,11 berada disamping kiri dan kanan robot. Pemakaian sensor IR pada robot ini di asumsikan bahwa, jika sensor berada pada garis, hal ini adalah garis hitam, maka keluaran dari sensor berlogika “1” dan jika sensor tidak berada pada pada garis, dalam hal ini latar berwarna putih, maka keluaran sensor akan berlogika “0”.

Pemasangan sensor yang tampak seperti pada gambar tampak atas, maka kemungkinan posisi sensor adalah:



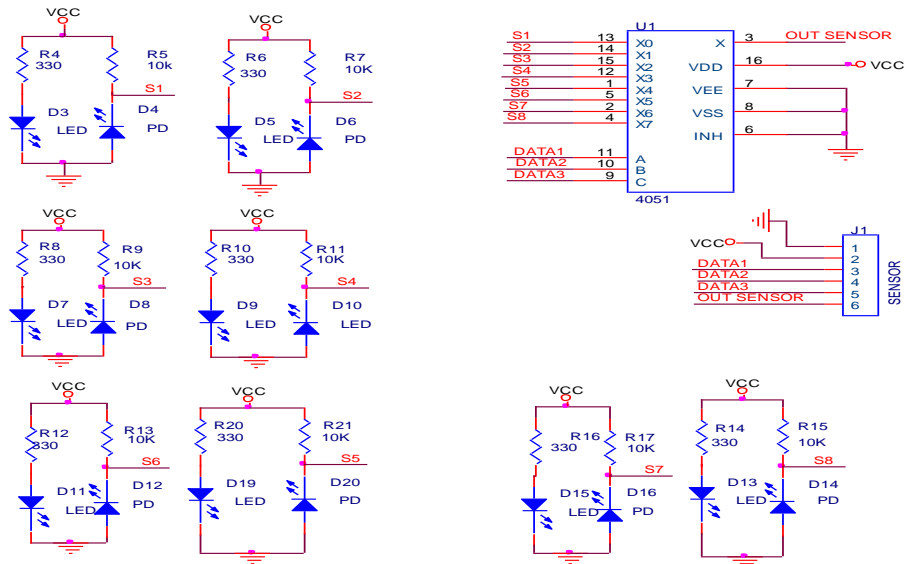
Gambar 3.6 Posisi Sensor Pada Lintasan

Mikrokontroler sebagai sistem navigasi dari robot akan memutar kedua motor DC secara *differensial*, dengan teknik PWM (*Pulse Width Modulation*) untuk menggerakkan robot mengikuti garis hitam. Keputusan mikrokontroler untuk menggerakkan motor DC berdasarkan kemungkinan posisi sensor seperti pada gambar 3.5 diatas, menggunakan Algoritma pemrograman PID.

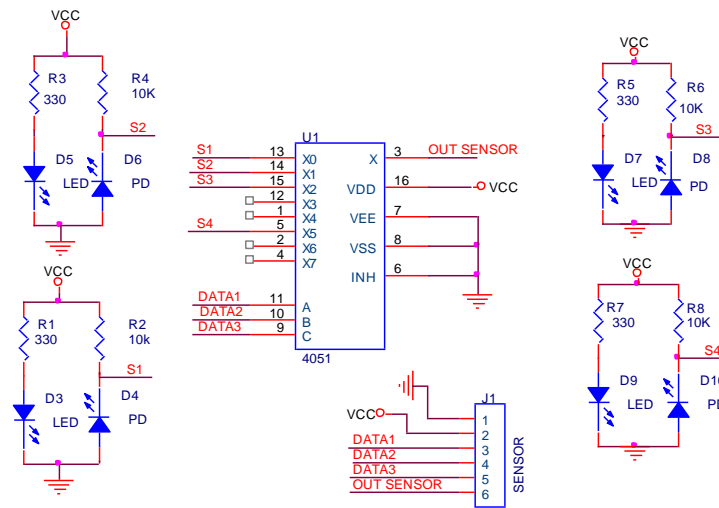
3.2 Sistem Sensor

Perancangan sensor *high speed line follower robot* pada proyek tugas akhir ini menggunakan sensor photodiode. Sensor photodiode adalah salah satu sensor yang beroperasi secara biner. Photodiode merupakan sebuah modul sensor

yang didalamnya terdapat kombinasi pancaran led yang diterima oleh photodiode dengan tingkat sensitif yang tinggi melalui media pantulan suatu obyek atau media yang kemudian diperkuat dengan penguat dan hasil *output* photodiode dikomparasi terlebih dahulu sehingga menghasilkan *output* 1 atau 0. Sistem kerja photodiode adalah *output* akan berlogika *High* (1) apabila di depannya terdapat media yang terang dan akan berlogika *Low* (0) apabila menemukan media yang lebih gelap. Jarak antara sensor dengan media yang dideteksi berkisar ± 1 cm. Bentuk rangkaian dari photodiode dapat dilihat pada gambar berikut.



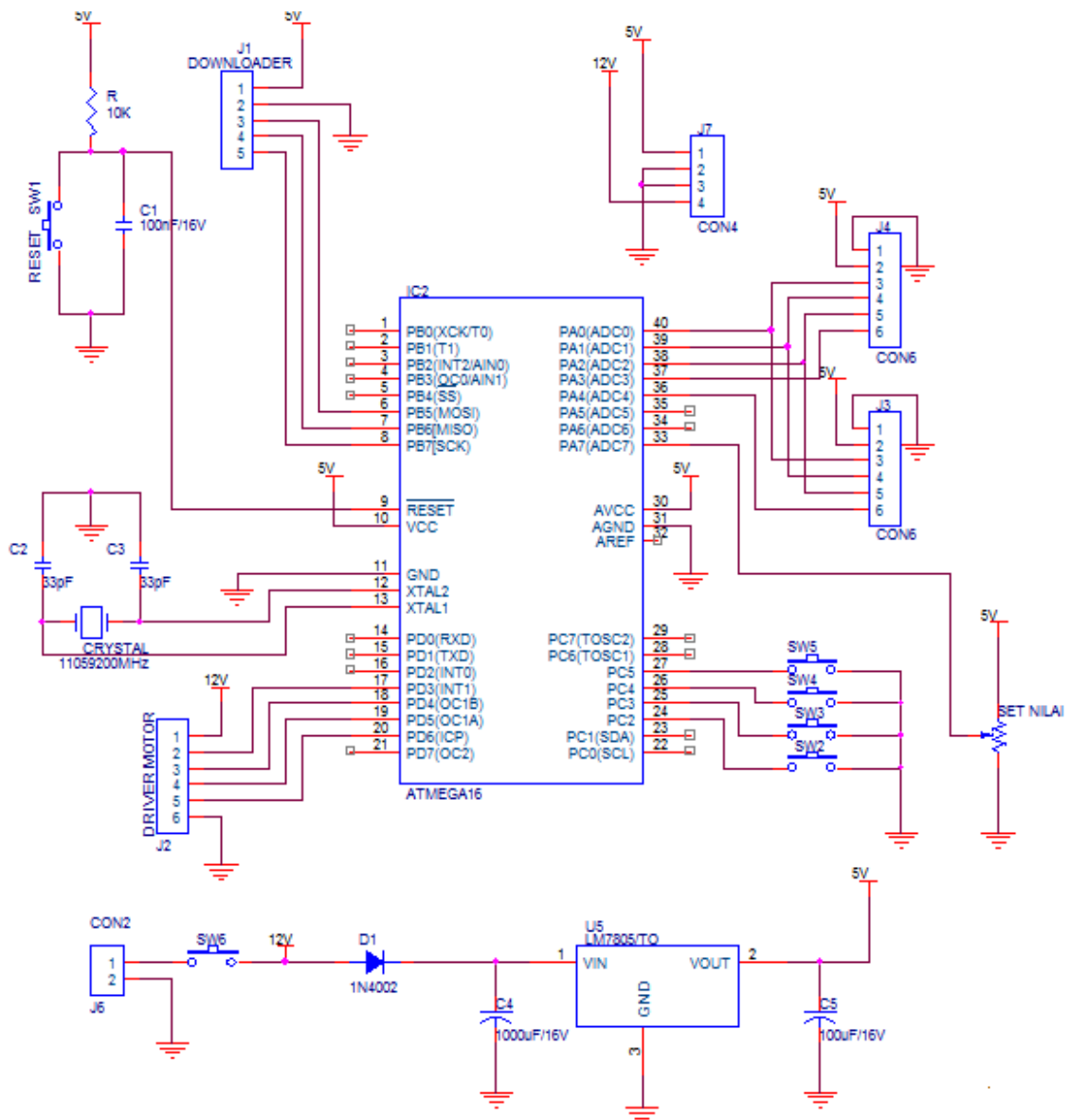
Gambar 3.7 Bentuk Skema Rangakain Sensor Depan



Gamabr 3.8 Bentuk Skema Rangkaian Sensor Samping

3.3 Mikrokontroller ATmega16

Seperti yang telah dijelaskan pada batasan masalah bahwa pembahasan lebih mendalam sistem mikokrontroler yang digunakan adalah menggunakan Mikrokontroler ATmega16. Disebut sistem minimal karena pemakaian komponen *hardware* yang digunakan merupakan kebutuhan yang paling minimal agar sebuah prosesor dapat bekerja. Gambar 3.8 adalah gambar skematik sistem minimal mikrokontroller ATmega16 yang sudah ditambah dengan *header* ISP (*In System Programming*) yang memungkinkan mikrokontroller diprogram secara langsung di dalam *board* robot melalui kabel ISP, tanpa harus bongkar pasang. Sensor dihubungkan ke PORT A.

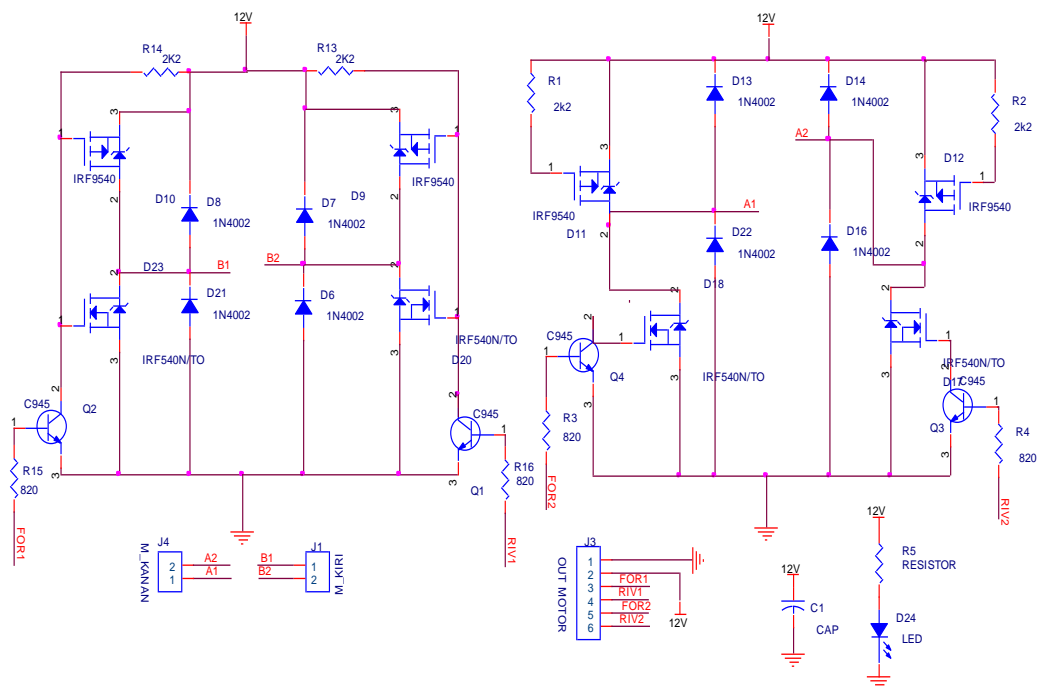


Gambar 3.9 Bentuk Skema Rangkaian Sistem Minimum ATmega16

3.4 Driver Penggerak Motor DC

Pada rancangan robot *High Speed Line Follower* ini *driver* penggerak yang digunakan menggunakan rakiaian mosfet. Hal ini dilakukan karena reaksi

yang didapat akan lebih cepat dibandingkan apabila menggunakan rangkaian *driver* LM298. Keputusan untuk memakai rangkaian mosfet ini juga didasari oleh karena untuk mendapatkan torsi yang besar tidak bisa menggunakan rangkain *driver* biasa.



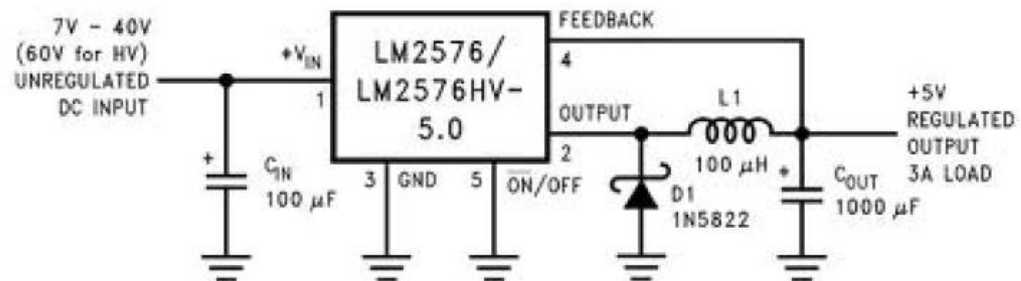
Gambar 3.10 Skema Rangkaian Mosfet

Transistor MOSFET (*Metal oxide FET*) memiliki *drain*, *source* dan *gate*. Namun perbedaannya *gate* terisolasi oleh suatu bahan oksida. *Gate* sendiri terbuat dari bahan metal seperti aluminium. Oleh karena itulah transistor ini dinamakan *metal-oxide*. Karena *gate* yang terisolasi, sering jenis transistor ini disebut juga IGFET yaitu *insulated-gate FET*. Ada dua jenis MOSFET, yang

pertama jenis *depletion-mode* dan yang kedua jenis *enhancement-mode*. Jenis MOSFET yang kedua adalah komponen utama dari gerbang logika dalam bentuk IC (*integrated circuit*), uC (*micro controller*) dan uP (*micro processor*) yang tidak lain adalah komponen utama dari komputer modern saat ini.

3.5 Rangkaian Regulator

Setiap mikrokontroler pasti membutuhkan sumber tegangan, lazimnya adalah 5 Volt (meskipun ada beberapa yang bisa memakai tegangan 2.85 – 3.6 Volt). Tapi, jika masih ingin memakai LM7805 untuk sumber tegangan 5Volt, diperlukan heatsink yang besar karena panasnya terlalu panas. Tetapi pada saat ini LM7805 sudah tidak layak lagi, karena *National Semiconductor* sudah mengeluarkan LM2576 untuk regulator 5 volt berjenis ‘*Simple Switcher*’. Rangkaianannya (diambil dari *datasheet* LM2576):



Gambar 3.11. Skema Rangkaian LM 2576

3.6 Perancangan dan Pembuatan Perangkat Lunak

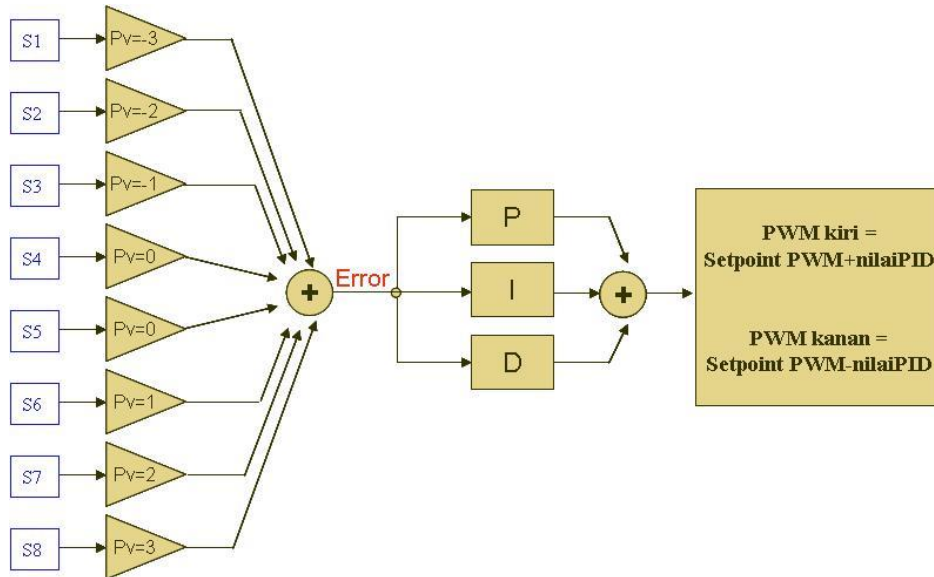
3.6.1 Basic Compiler AVR

Bascom AVR atau yang biasa disebut *basic compiler* adalah suatu piranti lunak yang termasuk bahasa tingkat tinggi yang sangat mudah untuk dipelajari. Sebagai *compiler*, yaitu pengubah instruksi dari bahasa *basic* ke *file* yang berbentuk *hexa* dengan tujuan dimengerti oleh mesin atau mikrokontroler, sehingga mikrokontroler mampu menterjemahkan instruksi-instruksi yang telah di buat dengan benar dan tepat. Piranti lunak inilah yang nantinya akan di gunakan sebagai media perancangan dalam pembuatan perangkat lunak.

3.6.2 Algoritama Kontrol PID

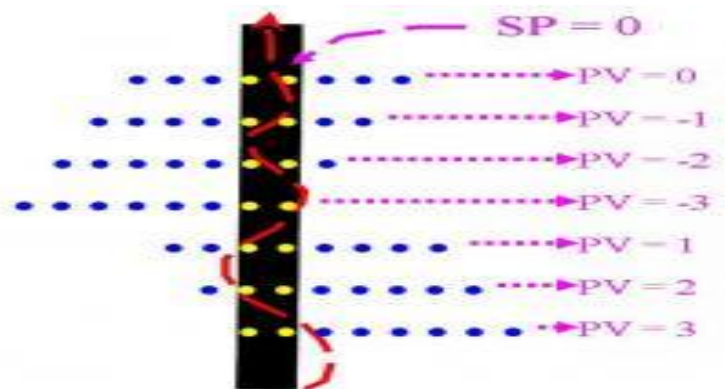
Tunning kontrol PID bertujuan untuk menentukan parameter aksi kontrol *Proportional, Integratif, Derivatif* pada robot *line follower*. Proses ini dapat dilakukan dengan cara *trial and error* . Keunggulan cara ini adalah tidak perlu mengidentifikasi *plant*, membuat model matematis *plant*, menentukan parameter *plant* dengan grafis ataupun analitis, cukup dengan cara coba-coba memberikan konstanta P-I-D pada formula PID hingga di peroleh hasil yang di inginkan, dengan mengacu pada karakteristik masing-masing kontrol P-I-D.

Penempatan aksi dari kontrol PID ini sendiri dapat dilihat pada ilustrasi gambar dibawah ini:



Gambar 3.12 ilustrasi penempatan aksi kontrol P-I-D

Gambar di bawah ini adalah ilustrasi pemberian bobot sensor (nilai kesalahan pembacaan sensor) pada robot line follower:



Gambar 3.13 Ilustrasi Pemberian Bobot Sensor

kondisi ideal robot terjadi saat kondisi robot pada $PV=0$ (misal kondisi nilai sensor = 11100111, nilai 0 merepresentasikan sensor mengenai garis). Atau $SP=0$ ialah kondisi ideal dari robot *line follower*. Jika PV tidak sama dengan 0 berarti robot tidak berada pada kondisi ideal dan artinya ada sinyal kesalahan (*error*). Pada kondisi *error* inilah formula PID akan menentukan hasil sinyal kendalinya.

Proporsional kontrol

Kondisi ideal pada robot adalah bergerak maju lurus mengikuti garis, dengan kata lain $PV=0$ (nilai sensor = 11100111). Dari sini dapat diasumsikan bahwa *Set Point* (SP) / kondisi ideal adalah saat $SP=0$. Nilai sensor yang dibaca oleh sensor disebut *Process Variable* (PV) / nilai aktual pembacaan. Menyimpangnya posisi robot dari garis disebut sebagai *error* (e), yang didapat dari $e = SP - PV$. Dengan mengetahui besar *error*, mikrokontroler dapat memberikan nilai PWM motor kiri dan kanan yang sesuai agar dapat menuju ke posisi ideal ($SP=0$). Besarnya nilai PWM ini dapat diperoleh dengan menggunakan kontrol Proporsional (P), dimana $P = e \cdot K_p$ (K_p adalah konstanta proporsional yang nilainya di set sendiri dari hasil tuning).

Saat Sensor = 11111111 nilai PV dapat ditentukan dengan cara membandingkan kondisi PV sebelumnya, jika PV lebih besar dari 0, maka

Semakin cepat bergerak dari satu sisi ke sisi lainnya, maka semakin besar nilai D. Konstanta D (K_d) digunakan untuk menambah atau mengurangi imbas dari *derivatif*. Dengan mendapatkan nilai K_d yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional bisa diminimalisasi. Nilai D didapat dari $D = K_d/T_s * rate$, dimana T_s ialah *time sampling* atau waktu cuplik dan $rate = e(n) - e(n-1)$. Dalam program nilai *error* ($SP - PV$) saat itu menjadi nilai *last_error*, sehingga *rate* didapat dari $error - last_error$. Untuk menambahkan kontrol D, program dimodifikasi menjadi:

Perhitungan kendali *proporsional + Derivatif*

$Sp_sensor = 0$	‘setpoint sensor
$Error = Sp_sensor - Pv$	‘nilai error
$P = K_p * Error$	‘proporsional kontrol
$D1 = K_d * 10$	‘derivatif kontrol
$D2 = D1 / T_s$	
$D3 = Error - Last_error$	‘rate
$D = D2 * D3$	
$Last_error = Error$	‘error lampau
$Pd = P + D$	‘proporsional-derivatif kontrol

Aplikasi kontrol proporsional dan *derivatif* pada PWM ialah sebagai berikut:

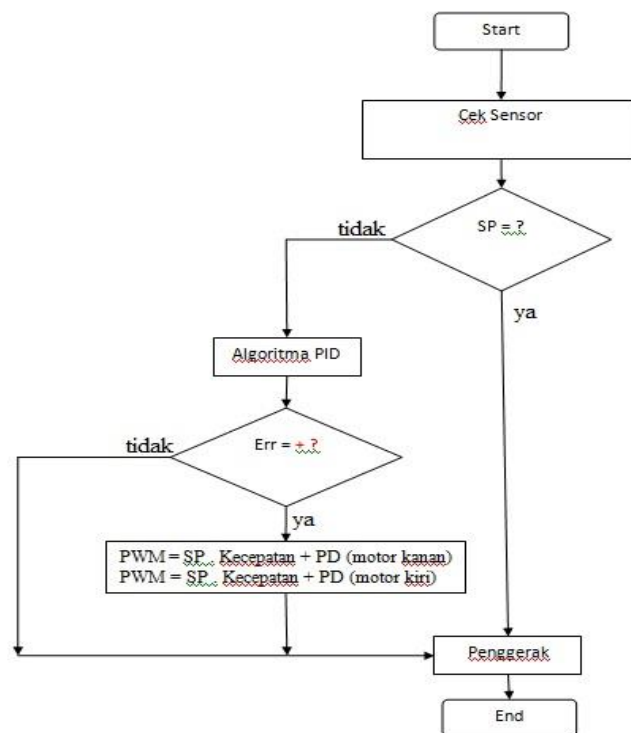
$$Pwm = Sp_kecepatan + Pd \quad \text{'motor kiri}$$

$$Pwm = Sp_kecepatan - Pd \quad \text{'motor kanan}$$

Integratif kontrol

Dengan P + D sudah membuat pergerakan robot cukup *smooth*, maka penambahan *Integratif* menjadi opsional. Dan pada penelitian ini kontrol I ini tidak di gunakan.

Algoritma untuk perancangan perangkat lunak pengendali yang telah dirancang dapat di tunjukan seperti gambar *flowchart* berikut:



Gambar 3.14 *Flowchart* Algoritma untuk perancangan kendali PID

BAB IV

PENGUJIAN, ANALISIS DAN PEMBAHASAN

4.1 Metode Pengujian

Pada sistem kerja kontrol *High Speed Line Follower* ini, terdapat beberapa sistem yang dapat dianalisa yaitu analisa sistem kerja sensor, pengujian dan analisa rangkaian sistem mikrokontroler, pengujian rangkaian *driver* motor, dan pengujian motor DC.

4.2 Pengujian dan Analisa Sistem Kerja Sensor

Sensor garis terdiri atas 12 buah sensor cahaya yang diatur sedemikian rupa agar dapat mengoptimalkan proses deteksi garis. Peletakan sensor adalah sebagai berikut:

1. 8 buah sensor pada bagian depan
2. 2 buah sensor bagian samping kanan
3. 2 buah sensor pada bagian samping kiri

Sensor garis ini bekerja dengan mendeteksi ada atau tidak adanya garis/jalur hitam pada area pergerakannya. Setiap perubahan kondisi garis/jalur akan dibaca sebagai perubahan bit *high/low* oleh sensor. Pembacaan ini akan mengakibatkan perubahan pergerakan pada penggerak kendaraan. Rangkaian sensor garis tersebut mendeteksi cahaya yang dipancarkan oleh LED *ultra bright*. Karena cahaya yang dipancarkan oleh LED *ultra bright* cukup terang sehingga dapat dipantulkan.

Analisa hasil pengujian rangkaian sensor dapat di jelaskan pada tabel di bawah ini:

Table 4.1 Hasil Pengujian Tegangan Pada Rangkaian Sensor Depan

Sensor Depan	Low	High
0	0.24 V	3.98 V
1	0.17 V	1.72 V
2	0.19 V	2.31 V
3	0.20 V	3.96 V
4	0.23 V	4.30 V
5	0.18 V	3.02 V
6	0.20 V	3.72 V
7	0.18 V	2.36 V

Table 4.2 Hasil Pengujian Tegangan Pada Rangkaian Samping Kanan

Sensor Samping Kanan	Low	High
0	0.27 V	2.39 V
1	0.18 V	3.21 V

Table 4.3 Hasil Pengujian Tegangan Pada Rangkaian Sensor Samping Kiri

Sensor Samping Kiri	Low	High
0	0.22 V	2.19 V
1	0.19V	2.87 V

Nilai tegangan kondisi *LOW* lebih rendah daripada pada saat kondisi *High*, hal ini menunjukkan stabilnya sistem pada rangkaian sensor.

4.3 Pengujian dan Analisa Rangkaian Mikrokontroler.

Mikrokontroler ini berjumlah 4 PORT dan 40 pin yang masing-masing PORT mempunyai jumlah 8 pin. Adapun fungsi semua PORT dalam Mikrokontroler adalah sebagai berikut:

1. PORTA (PA0-PA5) adalah pin yang terhubung ke rangkaian sensor.
2. PORTB (PB2-PB7) adalah pin yang terhubung ke *ISP downloader*.
3. PORTC (PC2-PC5) adalah pin yang terhubung ke tombol fungsi pada menu.
4. PORTD (PD3-PD6) adalah pin yang terhubung ke rangkaian driver motor.

Mikrokontroler ini harus disuplai tegangan 5 Volt dc. Dan sebagai sumber clock digunakan *clock eksternal* yang dibangkitkan oleh osilator kristal 16 MHz. Terdapat potensiometer yang berguna untuk mengatur nilai PWM atau mengatur nilai pada menu sesuai keinginan yang dihubungkan ke pin PA7. Dan sebuah *tripot* terpasang berguna sebagai pengatur contrast pada tampilan LCD. LCD sendiri terpasang pada PORT yang sama dengan yang terhubung pada sambungan *ISP downloader*.

4.4 Pengujian dan Analisa Rangkaian Driver Motor

Pada rangkaian driver motor ini menggunakan IC MOSFET IRF9540 dan IRF540 masing-masing dihubung seri sebanyak 4 buah. Penggunaan kedua jenis IC ini berfungsi sebagai pengendali besarnya tegangan yang di terima oleh motor DC. Untuk pengaturan kecepatan motornya pada penelitian ini di gunakan teknik PWM

(*Pulse Width Modulation*), dengan menggunakan metode ini pengendalian motor DC dapat dilakukan secara efektif.

PWM ini bekerja dengan cara membuat gelombang persegi yang memiliki perbandingan pulsa *high* terhadap pulsa *low* yang telah tertentu, diskalakan dari 0 hingga 100%. Gelombang persegi ini memiliki frekuensi tetap (max 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.

Analisa hasil pengujian rangkaian *Driver* Motor dapat di jelaskan pada tabel di bawah ini:

Table 4.4 Hasil Pengujian Pada Rangkaian Driver Motor

Logika	Respon Driver
1 0	ON
0 1	ON
1 1	Break
0 0	OFF

Pada saat salah satu pin pada *driver* di beri logika 1 (*high*) atau 0 (*Low*), maka *driver* motor dapat bergerak sesuai dengan kondisi logika yang di berikan.

4.5 Pengujian dan Analisa Motor DC

Analisa hasil pengujian rangkaian *Driver* Motor dapat di jelaskan pada tabel di bawah ini:

Table 4.5 Hasil Pengujian Hambatan Pada Motor DC

Hambatan Dalam Motor	Motor 1	Motor 2
Resistansi	1 Ohm	1 Ohm

Table 4.6 Hasil Pengujian RPM Pada Motor DC

Tegangan	Motor 1	Motor 2
3 V	5809.4 Rpm	5813 Rpm
5 V	10950 Rpm	10937 Rpm
7 V	17391 Rpm	17390 Rpm
9 V	21398 Rpm	21424 Rpm
11V	30785 Rpm	30802 Rpm
13V	34668 Rpm	34527 Rpm

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan dari analisa hasil pengamatan perancangan, pembuatan, pengujian alat, dan pembahasan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Tegangan sensor pada kondisi *high* akan selalu lebih tinggi dibandingkan tegangan pada kondisi *low*. Hal ini menandakan bahwa sensor dapat berfungsi dengan baik. Sensitivitas reaksi sensor ditentukan oleh jarak dan kondisi pencahayaan pada saat robot tersebut di operasikan. Semakin ideal jarak antara lantai dan sensor maka semakin baik pula respon tanggapan sensor terhadap garis.
2. Ukuran robot dan pemilihan bahan mekanik mempengaruhi performa dan akselerasi robot ketika berada pada jalur lintasan.
3. Dengan menggunakan Mikrokontroler ATmega16 ini baik PORT, pin maupun *clock* sudah dapat memenuhi kebutuhan untuk sebuah program *High Speed Line Follower*.
4. Hambatan terbesar dalam pemilihan motor DC ini adalah pemilihan motor DC yang mempunyai nilai RPM dan torsi yang pas. Maka, sebaiknya untuk setiap pemilihan motor DC harus memperhatikan

motor DC yang mempunyai respon tinggi tetapi RPM dan torsiya juga sesuai.

5. Pemilihan penggunaan motor DC sangat menentukan hasil akhir kecepatan motor yang di inginkan. Reaksi kecepatan motor DC ini bergantung pula pada *driver* motor yang dipakai.
6. *Driver* motor mosfet IRF9540 dan IRF540 dapat berjalan dengan baik sebagai pengatur kecepatan motor akan tetapi IC ini akan terlalu panas. Maka untuk mengatasinya di perlukan suatu regulator tertentu.

5.2 Saran

1. Untuk dapat merancang dan membuat *High Speed Line Follower* yang handal, diperlukan pemilihan sensor yang mampu bekerja dan memiliki respon yang cepat.
2. Sebaiknya motor DC yang digunakan adalah motor DC yang mempunyai RPM tinggi namun dapat memberikan respon dengan cepat.

DAFTAR PUSTAKA

- Bejo, Agus, 2008, "*C dan AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATmega8535*" Graha Ilmu, Yogyakarta
- Atmel, 2011, "*ATMEGA16*",
<http://www.alldatasheet.com/datasheetpdf/pdf/164169/ATMEL/16.html>
- Eko Prasetyo, 2010. *Sistem Gerak Robot Line Follower Menggunakan Motor DC Berbasis Mikrokontroler ATmega8535 Dengan Sensor Photodiode*. Universitas Diponegoro. Semarang.
- Malvino, Albert Paul, 2003, "*Prinsip-prinsip Elektronika*" Terjemahan Alb Joko Sutoso, Salemba Teknik, Jakarta
- Malvino, Albert Paul, 2003, "*Elektronika komputer digital*" Terjemahan Alb Tjia may On, Erlangga, Jakarta
- Maulana Rendy Santika, 2010, "*Line Follower Robot Programing (Line Follower Robot) Using C language on the AVR*". Universita Gunadarma. Jakarta

```
$regfile = "m32def.dat"  
$crystal = 12000000
```

```
Config Timer1 = Pwm , Pwm = 8 , Prescale = 64 , Compare A Pwm = Clear Up ,  
Compare B Pwm = Clear Up  
Config Adc = Single , Prescaler = 8 , Reference = Avcc  
Start Adc
```

```
Config Lcdpin = Pin , Db4 = Portb.5 , Db5 = Portb.4 , Db6 = Portb.3 , Db7 = Portb.2  
, E = Portb.6 , Rs = Portb.7  
Config Lcdbus = 4  
Config Lcd = 16 * 2  
Cursor Off
```

```
Dim Status_darurat As Bit  
Dim Mode_darurat As Byte  
Dim Channel As Byte  
Dim Pilihan As Byte  
Dim Pilihan1 As Byte  
Dim Isi As Byte  
Dim Kecepatan_tahap As Byte
```

```
Dim Kp As Byte , Kd As Byte , Kecepatan(5) As Byte , Strategi(16) As Byte  
Dim Vref0(8) As Word  
Dim Vref1(8) As Word  
Dim Vref2(8) As Word
```

```
Dim Href0(8) As Word  
Dim Href1(8) As Word  
Dim Href2(8) As Word
```

```
Dim Lref0(8) As Word  
Dim Lref1(8) As Word  
Dim Lref2(8) As Word
```

```
Dim Analog0(8) As Word  
Dim Analog1(8) As Word  
Dim Analog2(8) As Word
```

```
Dim Potensio As Word
```

```
Dim Sensor_depan As Byte  
Dim Sensor_belakang As Byte
```

Dim Ngerem As Bit , Siaga_kiri As Bit , Siaga_kanan As Bit , Siaga_simpang As Bit
, Siaga_simpang1 As Bit , Siaga_simpang2 As Bit , Siaga_y As Bit , Salah_cut As
Bit
Dim Simpang(16) As Byte
Dim Perempatan As Byte , Coun_kir As Byte , Coun_kan As Byte , Siaga_darurat As
Bit

Dim Ekp As Eram Byte , Ekd As Eram Byte , Ekecepatan(5) As Eram Byte ,
Estrategi(16) As Eram Byte
Dim Evref0(8) As Eram Word , Evref1(8) As Eram Word , Evref2(8) As Eram Word

Dim Prop As Integer , Deriv As Integer , P As Integer , D As Integer , X As Byte
Dim Pwm_ka_max As Integer , Pwm_ki_max As Integer , Pwm_ka As Integer ,
Pwm_ki As Integer , Diff_speed As Integer , Init_speed As Integer
Dim Sum_error As Integer , Diff As Integer , Bobot As Integer , Bobot_lalu As
Integer

Config Porta.0 = Output
Config Porta.1 = Output
Config Porta.2 = Output

Config Portd.3 = Output
Config Portd.6 = Output
Set Portd.3
Set Portd.6

Config Pinc.2 = Input
Set Portc.2
Config Pinc.3 = Input
Set Portc.3
Config Pinc.4 = Input
Set Portc.4
Config Pinc.5 = Input
Set Portc.5

Tombol1 Alias Pinc.2
Tombol2 Alias Pinc.3
Tombol3 Alias Pinc.4
Tombol4 Alias Pinc.5

```
Dirkanan Alias Portd.3
Dirkiri Alias Portd.6
Declare Sub Baca_sensor
Declare Sub Kalibrasi_sensor
Declare Sub Cek_sensor
Declare Sub Cabut
Declare Sub Cekin
Declare Sub Seting
Declare Sub Seting_speed
Declare Sub Isi_bobot
Mulai:
Cls
Channel = 0
```

```
Upperline
Lcd "1.START | 2.CEK "
Lowerline
Lcd "3.KAL | 4.SET "
Waitms 20
```

```
Do
If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set
Call Cabut
End If
If Tombol2 = 0 Then
Waitms 100
Bitwait Tombol2 , Set
Call Cekin
End If
```

```
If Tombol3 = 0 Then
Waitms 100
Bitwait Tombol3 , Set
Call Kalibrasi_sensor
End If
```

```
If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Call Seting
```

```

End If
Loop
'=====mulai
jalan=====
=

```

```

Sub Cabut:
Kp = Ekp
Kd = Ekd
Kecepatan(1) = Ekecepatan(1)
Kecepatan(2) = Ekecepatan(2)
Kecepatan(3) = Ekecepatan(3)
Kecepatan(4) = Ekecepatan(1)
Kecepatan(5) = Kecepatan(4) / 3
For Channel = 1 To 8
Vref0(channel) = Evref0(channel)
Vref1(channel) = Evref1(channel)
Vref2(channel) = Evref2(channel)

```

```

Next Channel
For Channel = 1 To 16
Simpang(channel) = Estrategi(channel)
Next Channel
Cls

```

```

Perempatan = 1
Reset Siaga_darurat
Reset Siaga_simpang
Reset Siaga_simpang1
Reset Siaga_simpang2
Reset Siaga_y
Reset Salah_cut
Init_speed = 0
Coun_kir = 0
Coun_kan = 0
Upperline
Lcd " PENDADARAN "
Lowerline
Lcd " 3 Januari 2012 "
Waitms 100
Reset Status_darurat
Mode_darurat = 0
Utama:

```



```

Call Baca_sensor
If Init_speed < Kecepatan(1) Then
Init_speed = Init_speed + 5
Else
Init_speed = Kecepatan(1)
End If
If Perempatan = 3 Then Init_speed = 80
Pwm_ka_max = Init_speed
Pwm_ki_max = Init_speed

```

```

Call Isi_bobot
If Bobot = 0 Then
'Goto Lurus
End If

```

```

Isi_pwm:
P = Kp
Prop = P * Bobot
Diff = Bobot - Bobot_lalu
Deriv = Kd * Diff

```

```

Pwm_ka = Pwm_ka_max - Prop
Pwm_ki = Pwm_ki_max + Prop
Pwm_ka = Pwm_ka - Deriv
Pwm_ki = Pwm_ki + Deriv

```

```

Bobot_lalu = Bobot

```

```

If Pwm_ka >= 0 Then
Gosub Ka_maju
Else
Gosub Ka_mundur
End If

```

```

If Pwm_ki >= 0 Then
Gosub Ki_maju
Else
Gosub Ki_mundur
End If

```

```

If Pwm_ka < 0 Then Pwm_ka = 0 - Pwm_ka
If Pwm_ki < 0 Then Pwm_ki = 0 - Pwm_ki

```

```
If Pwm_ka > 255 Then Pwm_ka = 255
If Pwm_ki > 255 Then Pwm_ki = 255
```

```
'rumus meredam gerbang pada mosfet
If Portd.3 = 1 Then
Pwm_ka = Abs(pwm_ka)
```

```
Else
Pwm_ka = 255 - Pwm_ka
```

```
End If
```

```
If Portd.6 = 1 Then
Pwm_ki = Abs(pwm_ki)
```

```
Else
Pwm_ki = 255 - Pwm_ki
```

```
End If
```

```
Pwm1b = Pwm_ka           ' isi PWMnya
Pwm1a = Pwm_ki
'Lowerline
'Lcd Pwm_ka ; "|" ; Pwm_ki ; "|" ; Bobot ; "    "
'Waitms 20
X = 0
Response:
Waitms 1
Incr X
Call Cek_sensor
```

```
If Sensor_depan = 0 And Mode_darurat > 5 Then
If Sensor_belakang.1 = 1 Then
Gosub Cut_kanan
Goto Utama
End If
```

```
If Sensor_belakang.6 = 1 Then
Gosub Cut_kiri
Goto Utama
End If
```

End If

```
If Sensor_depan = 0 And Mode_darurat <= 5 Then
If Sensor_belakang.1 = 1 Or Sensor_belakang.0 = 1 Then
Gosub Cut_kanan
Goto Utama
End If
```

```
If Sensor_belakang.6 = 1 Or Sensor_belakang.7 = 1 Then
Gosub Cut_kiri
Goto Utama
End If
```

End If

```
If Tombol4 = 0 Then
Cls
Upperline
Lcd "Mode Darurat      "
Waitms 200
Bitwait Tombol4 , Set
Status_darurat = 1
End If
```

```
If Tombol2 = 0 Then
Cls
Upperline
Lcd "Normal          "
Waitms 200
Bitwait Tombol2 , Set
Status_darurat = 0
End If
```

```
If X = Kecepatan(3) Then Goto Utama
Goto Response
```

End Sub

```
'=====siaga
simpang=====
```

'=====

=====
Maju:
Set Dirkanan
Set Dirkiri
Pwm1a = Kecepatan(1)
Pwm1b = Kecepatan(1)
Waitms 10
Coun_kan = 0
Coun_kir = 0
Return

Ka_maju:
Set Dirkanan
Return

Ka_mundur:
Reset Dirkanan
Return

Ki_maju:
Set Dirkiri
Return
Ki_mundur:
Reset Dirkiri
Return

Cut_kiri:
'Reset Dirkanan
'Reset Dirkiri
'Pwm1a = Kecepatan(1)
'Pwm1b = Kecepatan(1)
'Waitms Kecepatan(2)
Incr Mode_darurat
Reset Dirkanan
Set Dirkiri
Pwm1a = Kecepatan(5)
Pwm1b = 255 - Kecepatan(4)
Do
Call Cek_sensor
Loop Until Sensor_depan.7 = 1 Or Sensor_depan.6 = 1 Or Sensor_depan.5 = 1 Or
Sensor_depan.4 = 1
Reset Siaga_kanan

```

Reset Siaga_kiri
Siaga_y = 0
Coun_kan = 0
Coun_kir = 0
Init_speed = 0
Return

```

```

Cut_kanan:
'Reset Dirkanan
'Reset Dirkiri
'Pwm1a = Kecepatan(1)
'Pwm1b = Kecepatan(1)
'Waitms Kecepatan(2)
Incr Mode_darurat
Set Dirkanan
Reset Dirkiri
Pwm1a = 255 - Kecepatan(4)
Pwm1b = Kecepatan(5)
Do
Call Cek_sensor
Loop Until Sensor_depan.0 = 1 Or Sensor_depan.1 = 1 Or Sensor_depan.2 = 1 Or
Sensor_depan.3 = 1
Reset Siaga_kanan
Reset Siaga_kiri
Reset Siaga_y
Coun_kan = 0
Coun_kir = 0
Siaga_y = 0
Init_speed = 0
Return

```

```

'=====
=====cut kusus=====

```

```

'=====isi

```

```

bobot=====

```

```

Sub Isi_bobot:
Utama2:
Call Cek_sensor

```

```

If Sensor_depan = &B00000001 Then Bobot = 15
If Sensor_depan = &B00000011 Then Bobot = 13
If Sensor_depan = &B00000010 Then Bobot = 11
If Sensor_depan = &B00000110 Then Bobot = 9

```

```

If Sensor_depan = &B00000100 Then Bobot = 7
If Sensor_depan = &B00001100 Then Bobot = 3
If Sensor_depan = &B00001000 Then Bobot = 1
If Sensor_depan = &B00011000 Then Bobot = 0
If Sensor_depan = &B00010000 Then Bobot = -1
If Sensor_depan = &B00110000 Then Bobot = -3
If Sensor_depan = &B00100000 Then Bobot = -7
If Sensor_depan = &B01100000 Then Bobot = -9
If Sensor_depan = &B01000000 Then Bobot = -11
If Sensor_depan = &B11000000 Then Bobot = -13
If Sensor_depan = &B10000000 Then Bobot = -15

```

End Sub

```

'=====
seting=====
=====

```

Sub Seting:

```

Cls
Channel = 0
Do
Upperline
Lcd "1.KP ||| 2.KD "
Lowerline
Lcd "3.SPED||| 4.BACK"
Waitms 200

```

```

If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set

```

```

'===== KP=====

```

```

Cls
Upperline
Lcd "KP = "
Lowerline
Lcd "1. OK ||| 4.Back"
Waitms 20

```

```

'-----
Do
Potensio = Getadc(7)
Shift Potensio , Right , 2
Kp = Potensio
Locate 1 , 7

```

```

Lcd Kp ; "          "
Waitms 20
If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set
Ekp = Kp
Upperline
Lcd "telah tersimpan "
Wait 1
End If

If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Exit Do
End If
Loop
End If
'=====KD=====
If Tombol2 = 0 Then
Waitms 100
Bitwait Tombol2 , Set
Cls
Upperline
Lcd "KD = "
Lowerline
Lcd "1. OK ||| 4.Back"
Waitms 20
'-----
Do
Potensio = Getadc(7)
Shift Potensio , Right , 2
Kd = Potensio
Locate 1 , 7
Lcd Kd ; "          "
Waitms 20
If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set
Ekd = Kd
Upperline
Lcd "telah tersimpan "
Wait 1

```

End If

```
If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Exit Do
End If
Loop
End If
```

```
'=====SPD=====
If Tombol3 = 0 Then
Waitms 100
Bitwait Tombol3 , Set
Call Seting_speed
End If
```

```
If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Goto Seting_abis
End If
Loop
Seting_abis:
End Sub
```

```
'=====AKSES MULTIPLEX SENSOR=====
Sub Baca_sensor:
For Channel = 0 To 7
Pilihan = Channel + 1
'Porta.2 = Channel.0
'Porta.1 = Channel.1
'Porta.0 = Channel.2
Porta = Channel
Waitus 50
Analog0(pilihan) = Getadc(4)
Analog1(pilihan) = Getadc(5)
'Analog2(pilihan) = Getadc(5)
Next Channel
End Sub
```

```
Sub Kalibrasi_sensor
```



```

Upperline
Lcd "Sedang Kalibrasi"
Waitms 20
For Channel = 1 To 8
Href0(channel) = 0
Href1(channel) = 0
Href2(channel) = 0
Lref0(channel) = 1023
Lref1(channel) = 1023
Lref2(channel) = 1023
Next Channel

Do
Call Baca_sensor
For Channel = 1 To 8
If Analog0(channel) > Href0(channel) Then Href0(channel) = Analog0(channel)
If Analog1(channel) > Href1(channel) Then Href1(channel) = Analog1(channel)
If Analog2(channel) > Href2(channel) Then Href2(channel) = Analog2(channel)

If Analog0(channel) < Lref0(channel) Then Lref0(channel) = Analog0(channel)
If Analog1(channel) < Lref1(channel) Then Lref1(channel) = Analog1(channel)
If Analog2(channel) < Lref2(channel) Then Lref2(channel) = Analog2(channel)
Next Channel
Loop Until Tombol1 = 0

For Channel = 1 To 8
Vref0(channel) = Lref0(channel) + Href0(channel)
Vref1(channel) = Lref1(channel) + Href1(channel)
Vref2(channel) = Lref2(channel) + Href2(channel)

'BOLEH DIBAGI DENGAN ANGKA LAIN
Vref0(channel) = Vref0(channel) / 3
Vref1(channel) = Vref1(channel) / 3
Vref2(channel) = Vref2(channel) / 3

Evref0(channel) = Vref0(channel)
Evref1(channel) = Vref1(channel)
Evref2(channel) = Vref2(channel)

Next Channel
Upperline
Lcd "Kalibrasi Sukses"
Waitms 20

```

```

Cls
Do
Call Cek_sensor
Locate 1 , 5
Lcd Bin(sensor_depan)
Locate 2 , 5
Lcd Bin(sensor_belakang)
'Locate 2 , 7
'Lcd Sensorkanan
'Locate 2 , 8
'Lcd Sensorkiri

Waitms 50
Loop Until Tombol1 = 0

End Sub

```

```

Sub Cek_sensor

```

```

Call Baca_sensor
For Channel = 0 To 7
Pilihan = Channel + 1

```

```

    If Analog0(pilihan) > Vref0(pilihan) Then
Sensor_depan.channel = 1
    Else
Sensor_depan.channel = 0
    End If

```

```

    If Analog1(pilihan) > Vref1(pilihan) Then
Sensor_depan.pilihan1 = 1
    Else
Sensor_depan.pilihan1 = 0
    End If

```

```

    If Analog2(pilihan) > Vref2(pilihan) Then
Sensor_belakang.channel = 1
    Else
Sensor_belakang.channel = 0
    End If

```

```
Next Channel
End Sub
```

```
'=====setting
```

```
speed=====
```

```
Sub Seting_speed
```

```
Cls
```

```
Upperline
```

```
Lcd "Kec = "
```

```
Lowerline
```

```
Lcd "1. OK ||| 4.Back"
```

```
Waitms 20
```

```
'-----
```

```
Do
```

```
Potensio = Getadc(7)
```

```
Shift Potensio , Right , 2
```

```
Kecepatan(1) = Potensio
```

```
Locate 1 , 7
```

```
Lcd Kecepatan(1) ; " "
```

```
Waitms 20
```

```
If Tombol1 = 0 Then
```

```
Waitms 100
```

```
Bitwait Tombol1 , Set
```

```
Ekecepatan(1) = Kecepatan(1)
```

```
Upperline
```

```
Lcd "telah tersimpan "
```

```
Wait 1
```

```
End If
```

```
If Tombol4 = 0 Then
```

```
Waitms 100
```

```
Bitwait Tombol4 , Set
```

```
Exit Do
```

```
End If
```

```
Loop
```

```
'End If
```

```
'=====Limit
```

```
'If Tombol2 = 0 Then
```

```
'Waitms 100
```

```
'Bitwait Tombol2 , Set
```

```
'=====
```

```
Cls
```

```
Upperline
```

```

Lcd "Lim = "
Lowerline
Lcd "1. OK ||| 4.Back"
Waitms 20
'-----
Do
Potensio = Getadc(7)
Shift Potensio , Right , 2
Kecepatan(2) = Potensio
Locate 1 , 7
Lcd Kecepatan(2) ; "          "
Waitms 20
If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set
Ekecepatan(2) = Kecepatan(2)
Upperline
Lcd "telah tersimpan "
Wait 1
End If

If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Exit Do
End If
Loop
'End If

'=====sampling=====
'If Tombol3 = 0 Then
'Waitms 100
'Bitwait Tombol3 , Set
'=====
Cls
Upperline
Lcd "Sam = "
Lowerline
Lcd "1. OK ||| 4.Back"
Waitms 20
'-----
Do
Potensio = Getadc(7)

```

```

Shift Potensio , Right , 2
Kecepatan(3) = Potensio
Locate 1 , 7
Lcd Kecepatan(3) ; "          "
Waitms 20
If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set
Ekecepatan(3) = Kecepatan(3)
Upperline
Lcd "telah tersimpan "
Wait 1
End If

```

```

If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Exit Do
End If
Loop
'End If

```

```
'=====Cuter=====
```

```
=====
```

```
'If Tombol4 = 0 Then
'Waitms 100
'Bitwait Tombol4 , Set
'
```

```
'=====
```

```
Cls
Upperline
Lcd "Cut1= "
Lowerline
Lcd "1. OK ||| 4.Back"
Waitms 20
'
```

```
'-----
```

```
Do
Potensio = Getadc(7)
Shift Potensio , Right , 2
Kecepatan(4) = Potensio
Locate 1 , 7
Lcd Kecepatan(4) ; "          "
Waitms 20
If Tombol1 = 0 Then
Waitms 100

```

```

Bitwait Tombol1 , Set
Ekecepatan(4) = Kecepatan(4)
Upperline
Lcd "telah tersimpan "
Wait 1
End If

```

```

If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Goto Cut2
End If
Loop
'End If

```

```

'=====Cuter 2=====

```

```

Cut2:
'=====
Cls
Upperline
Lcd "Cut2= "
Lowerline
Lcd "1. OK ||| 4.Back"
Waitms 20
'-----

```

```

Do
Potensio = Getadc(7)
Shift Potensio , Right , 2
Kecepatan(5) = Potensio
Locate 1 , 7
Lcd Kecepatan(5) ; "      "
Waitms 20
If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set
Ekecepatan(5) = Kecepatan(5)
Upperline
Lcd "telah tersimpan "
Wait 1
End If

```

```

If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set

```

```
Exit Do
End If
Loop
```

```
'=====strategi=====
=====
```

```
Channel = 1
Cls
Do
Upperline
Lcd "perempatan= " ; Channel ; "      "
Lowerline
Lcd "1=1|2=2|3=0|4>  "
Waitms 20
```

```
If Tombol1 = 0 Then
Waitms 100
Bitwait Tombol1 , Set
Simpang(channel) = 1
Estrategi(channel) = Simpang(channel)
Upperline
Lcd "kan(1) tersimpan "
Wait 1
End If
```

```
If Tombol2 = 0 Then
Waitms 100
Bitwait Tombol2 , Set
Simpang(channel) = 2
Estrategi(channel) = Simpang(channel)
Upperline
Lcd "kir(2) tersimpan "
Wait 1
End If
```

```
If Tombol3 = 0 Then
Waitms 100
Bitwait Tombol3 , Set
Simpang(channel) = 0
Estrategi(channel) = Simpang(channel)
Upperline
Lcd "lur(0) tersimpan "
Wait 1
```

```

End If

If Tombol4 = 0 Then
Waitms 100
Bitwait Tombol4 , Set
Incr Channel
If Channel >= 17 Then Exit Do
End If
Loop

End Sub
'=====cek
eeprom=====
Sub Cekin:

Kp = Ekp
Kd = Ekd
Kecepatan(1) = Ekecepatan(1)
Kecepatan(2) = Ekecepatan(2)
Kecepatan(3) = Ekecepatan(3)
Kecepatan(4) = Ekecepatan(4)
Kecepatan(5) = Ekecepatan(5)
For Channel = 1 To 8
Vref0(channel) = Evref0(channel)
Vref1(channel) = Evref1(channel)
Vref2(channel) = Evref2(channel)

Next Channel

For Channel = 1 To 16
Simpang(channel) = Estrategi(channel)
Next Channel

Ngecek:
Cls
Upperline
Lcd "KP =" ; Kp ; "      "
Lowerline
Lcd "KD =" ; Kd ; "| 4.Next>  "
Waitms 200
Do

If Tombol4 = 0 Then

```



```

Waitms 200
Bitwait Tombol4 , Set
Goto Cek_speed
End If
Loop
'=====
Cek_speed:
Cls
Upperline
Lcd "Kecepatan =" ; Kecepatan(1) ; "    "
Lowerline
Lcd "Limit =" ; Kecepatan(2) ; "|4.==>    "
Waitms 200
Do

If Tombol4 = 0 Then
Waitms 200
Bitwait Tombol4 , Set
Goto Cek_sampling
End If
Loop

'=====
Cek_sampling:
Cls
Upperline
Lcd "Sampling =" ; Kecepatan(3) ; "    "
Lowerline
Lcd "Lanjut 4.==>    "
Waitms 200
Do

If Tombol4 = 0 Then
Waitms 200
Bitwait Tombol4 , Set
Goto Cek_cuter
End If
Loop

'=====
Cek_cuter:
Cls
Upperline
Lcd "cut1=" ; Kecepatan(4) ; "cut2=" ; Kecepatan(5) ; "    "

```

```

Lowerline
Lcd "Lanjut 4.==>      "
Waitms 200
Do

If Tombol4 = 0 Then
Waitms 200
Bitwait Tombol4 , Set
Goto Cek_strategi
End If
Loop
'=====strategi=====
Cek_strategi:
Cls
Upperline
Lcd Simpang(1) ; Simpang(2) ; Simpang(3) ; Simpang(4) ; Simpang(5) ; Simpang(6)
; Simpang(7) ; Simpang(8) ; Simpang(9) ; Simpang(10) ; Simpang(11) ; Simpang(12)
; Simpang(13) ; Simpang(14) ; Simpang(15) ; Simpang(16);
Lowerline
Lcd "Strategi^| 4.==>      "
Waitms 200
Do

If Tombol4 = 0 Then
Waitms 200
Bitwait Tombol4 , Set
Goto Cek_sensornya
End If
Loop

'=====sensor ceking=====

Cek_sensornya:
Do
Call Cek_sensor
Locate 1 , 5
Lcd Bin(sensor_depan)
Locate 2 , 1
Lcd Bin(sensor_belakang) ; "|<=1|4=> "
Waitms 50

If Tombol1 = 0 Then
Waitms 200

```

```
Bitwait Tombol1 , Set  
Goto Ngecek  
End If
```

```
If Tombol4 = 0 Then  
Waitms 200  
Bitwait Tombol4 , Set  
Goto Cek_selesai  
End If
```

```
Loop  
Cek_selesai:  
Goto Mulai  
End Sub
```