

II. TINJAUAN PUSTAKA

A. Perkembangan Mobil

MOBIL pertama dengan prinsip kendaraan bermotor berhasil ditemukan pada tahun 1769 [2]. Penemunya adalah Nicholas Joseph Cugnot (1725-1804), dengan mobil bermesin tenaga uap air. Kemudian teknologi mesin uap ini lebih banyak digunakan sebagai tenaga penggerak lokomotif kereta api. Seperti roda mobil, revolusi mesin penggerak kendaraan terus menggelinding. Setelah temuan mesin uap, kini giliran mesin dengan tenaga uap minyak bumi yang diterapkan untuk menggerakkan kendaraan. Henry Ford, dialah orang yang berhasil membuat mobil dengan mesin uap minyak bumi. Henry Ford lahir pada tanggal 30 Juli 1863.

Karl Friedrich Benz adalah sarjana dari Jerman yang dikenal sebagai penemu dari mobil dengan bahan bakar bensin (gazoline). Walaupun pada saat yang bersamaan Gottlieb Daimler yang berpasangan dengan Wilhelm Maybach juga bekerja meneliti mesin dengan bahan bakar bensin juga, Benz terlebih dahulu menyelesaikan penemuan itu dan mempatenkan penemuan tersebut pada tahun 1879.

Tahun 1885, Karl Benz membangun Motorwagen, mobil pertama yang dijual secara komersil. Mobil tersebut adalah mobil dengan mesin empat langkah dengan bahan bakar bensin hasil rancangannya. Benz juga merupakan penemu dari komponen mobil seperti pengapian mobil, busi, sistem transmisi mobil, radiator air dan karburator. Motorwagen, mobil pertama dengan bahan bakar bensin. Motorwagen menjadi cikal-bakal dari mobil modern yang kita kendarai sekarang.

Penemuan – penemuan mobil tersebut belum membuat para ahli dan penemu mobil tersebut bangga dengan penemuannya, justru ingin terus melanjutkan penelitiannya, diantaranya model dan design mobil tersebut, karena tanpa didesign secara cermat, setelah diteliti ternyata berpengaruh terhadap kecepatan dari gaya gesekan terhadap udara.

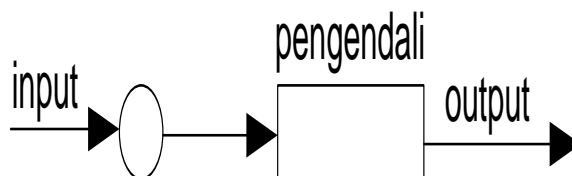
Setelah Perang Dunia II, desain mobil-mobil Jerman menggunakan nilai-nilai baru, di mana performa mobil dan karakter merek harus tercermin dalam tampilan dan sosok mobil. Dengan demikian, orang-orang yang bekerja di departemen desain menghadapi tugas baru yang lebih berat, seiring dengan semakin lebarnya rentang model yang dikeluarkan masing-masing merek mobil. Pada saat yang sama, mereka pun diharapkan muncul dengan desain-desain yang menggabungkan keaslian dengan kosmopolitan.

B. Dasar Sistem Kontrol

Sistem kontrol atau sistem kendali merupakan suatu sistem yang keluarannya atau outputnya dikendalikan pada suatu nilai tertentu atau untuk merubah beberapa ketentuan yang telah ditetapkan dari masukan atau inputan ke sistem [3]. Untuk merancang suatu sistem yang dapat merespon perubahan tegangan dan mengeksekusi perintah berdasarkan situasi yang terjadi, maka diperlukan pemahaman tentang sistem kendali (*controll system*). Sistem kendali merupakan suatu kondisi dimana sebuah perangkat (*device*) dapat di kontrol sesuai dengan perubahan situasi.

1. Sistem kendali kalang terbuka (*open loop*).

Loop terbuka atau *open loop* merupakan sebuah sistem yang tidak dapat merubah dirinya sendiri terhadap perubahan situasi yang ada. Hal ini disebabkan karena tidak adanya umpanbalik (*feedback*) pada sebuah sistem kalang terbuka. Sistem ini masih membutuhkan campur tangan manusia yang bekerja sebagai operator. Dapat dilihat pada blok diagram dari sebuah sistem kalang terbuka, sebagai berikut:

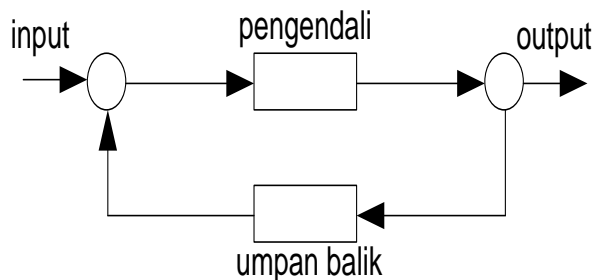


Gambar 2.1. Rangkaian Kalang Terbuka (sumber : google)

Pada sistem kalang terbuka inputan di kendalikan oleh manusia sebagai operator, dan perubahan kondisi lingkungan tidak akan langsung direspon oleh sistem, melainkan di kontrol oleh manusia.

2. Sistem kendali kalang tertutup (*close loop*).

Loop tertutup merupakan sebuah sistem kontrol yang sinyal atau nilai keluarannya memiliki pengaruh langsung terhadap aksi pengendalian yang dilakukan. Pada rangkaian loop tertutup sinyal *error* yang merupakan selisih antara sinyal masukan dan sinyal umpanbalik (*feedback*), lalu diumpankan pada komponen pengendalian (*controller*), umpan balik ini dilakukan untuk memperkecil kesalahan nilai keluaran (*output*) sistem semakin mendekati nilai yang diinginkan.



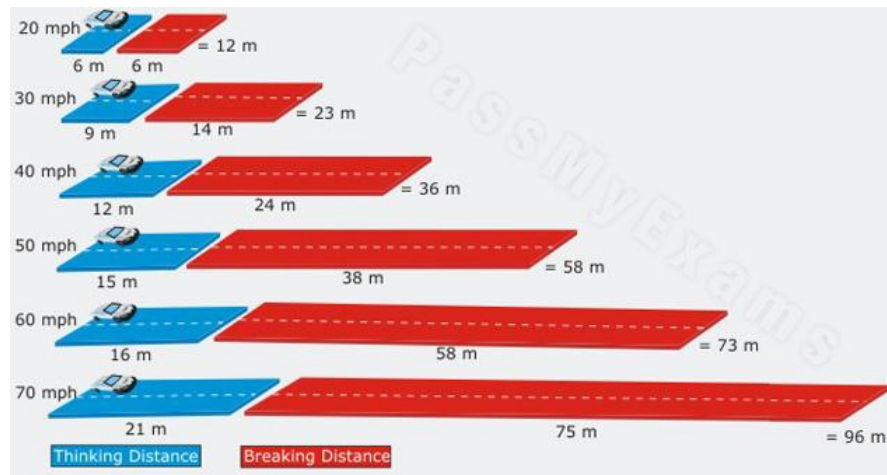
Gambar 2.2. Rangkaian Kalang Tertutup (sumber : google)

Keuntungan dari rangkaian sistem loop tertutup ini adalah adanya pemanfaatan nilai umpanbalik yang dapat membuat respon sistem kurang peka terhadap gangguan eksternal dan perubahan internal pada parameter sistem. Kemudian kerugiannya

adalah tidak dapat mengambil aksi perbaikan terhadap suatu gangguan sebelum gangguan tersebut mempengaruhi nilai prosesnya.

C. Breaking distance dan Jarak Aman Berkendara

Braking Distance adalah jarak yang dibutuhkan kendaraan untuk berhenti total mulai dari pengemudi mengoperasikan rem [4]. Bila kecepatan kendaraan semakin cepat, *braking distance* akan semakin panjang. Berarti waktu yang dibutuhkan kendaraan untuk berhenti akan semakin lama. Selain itu *braking distance* juga tergantung pada kondisi permukaan jalan. Oleh karena itu pada kondisi jalan licin atau berpasir pengemudi diharap mengurangi kecepatan dan lebih berhati-hati. Selain adanya *braking distance* jika kendaraan manual tanpa adanya *auto brake sistem* seperti yang diterapkan pada simulator ini, maka pada saat berkendara dibutuhkan juga adanya *thinking distance*. *Thinking distance* adalah jarak saat dimana pengemudi menyadari harus mengerem. Kalau diumpamakan sebagai waktu, maka *empty distance* berkisaran 0,7 detik.



Gambar 2.3 diagram thinking distance dan breaking distance

(sumber : kompas.com)

Dari diagram pola diatas dapat dilihat hubungan antara kecepatan mobil dan waktu menghentikan. Waktu reaksi total rata-rata sopir adalah sekitar 0.7 detik, semakin kecepatan mobil meningkat dengan waktu reaksi yang sama maka semakin jauh juga jarak reaksinya, dan dapat dilihat dari hubungan;

$$\text{Jarak tempuh kecepatan} = (X) \text{ waktu reaksi} \dots\dots\dots(2.1)$$

Begitu juga dengan jarak pengereman yang juga meningkat pada kecepatan yang lebih tinggi. Mobil akan memiliki lebih banyak energi kinetik. Energi ini ditransformasikan ke rem, sehingga jarak pengereman juga semakin panjang.

Hubungan antara kecepatan dan energi kinetik,

$$\text{Kinetic energy (joule)} = \frac{1}{2} \times \text{massa} \times \text{kecepatan}^2 \dots\dots(2.2)$$

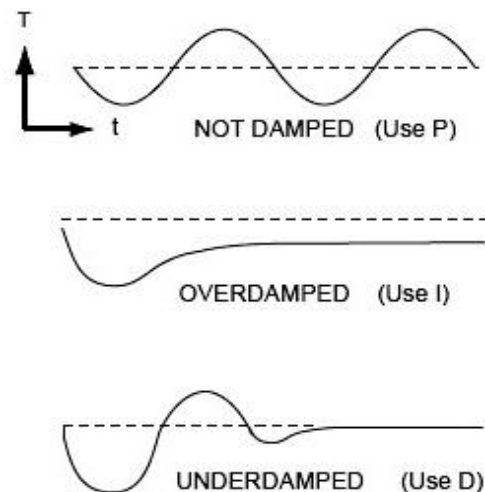
Oleh karena itu jika mobil melaju tiga kali kecepatan memiliki 9 kali energi kinetik yang lebih besar, berarti jarak pengereman akan sembilan kali lebih lama. Artinya memiliki resiko kecelakaan sembilan kali lebih besar.

Berkendara dengan aman adalah dengan mengikuti aturan berkendara dengan menjaga jarak aman dengan kendaraan didepan. Jarak aman berkendara dalam meter adalah sama dengan kecepatan laju. Dimisalkan sebuah kendaraan melaju dengan kecepatan 60 km/jam, maka jarak amak kendaraan tersebut adalah 60 m.

D. Pengendali PID

PID merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut.

Komponen kontrol PID ini terdiri dari tiga jenis yaitu **Proportional**, **Integratif** dan **Derivatif**. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant.



Gambar 2.4 Bentuk sinyal PID (sumber : wikipedia)

1. Kontrol Proporsional

Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka

$$u = K_p \cdot e \quad \dots\dots\dots (2.3)$$

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

$$u(t) = \left[\int e(t) dt \right] K_i \quad \dots\dots\dots (2.4)$$

dengan K_i adalah konstanta Integral, dan dari persamaan di atas, $G(s)$ dapat dinyatakan sebagai

$$u = K_d \cdot \left[\frac{\Delta e}{\Delta t} \right] \quad \dots\dots\dots (2.5)$$

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 pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan

Ki yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde sistem.

3.Kontrol Derivatif

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai

$$G(s) = s.Kd \dots\dots\dots (2.6)$$

Dari persamaan di atas, 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 error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri

E. Putaran motor, Putaran Roda, Keliling Roda dan Gear Box

Final reduction adalah perbandingan antara putaran *input* dan *output* pada *differential*, sebelum daya putaran diberikan ke roda [5]. *Final reduction* akan menurunkan putaran dan meningkatkan torsi. Pada sepeda motor, *final reduction* adalah perbandingan roda gigi rantai (*sprocket*) yang besar pada roda belakang, dibagi dengan sprocket yang kecil pada output transmisi. Beberapa kendaraan

didesain dengan beberapa rasio, seperti rasio primer dan sekunder, *low*, *high*, dan lain-lain. Jika rasio-rasio transmisi tersebut bekerja pada kecepatan tertentu, maka rasio-rasio tersebut harus juga dicantumkan dalam kalkulasi untuk kecepatan yang bersangkutan.

$$\text{Putaran roda} = \text{putaran engine} / \text{rasio transmisi (Gear box)} \dots(2.7)$$

Ukuran roda akan mempengaruhi kecepatan kendaraan. Untuk menghitung kecepatan dengan berpatokan pada RPM engine, maka dibutuhkan data keliling ban. Pengukuran keliling ban dilakukan dengan mengukur tinggi roda dari tanah ke titik pusat as roda (*tire rolling radius*). Perlu diketahui bahwa tinggi roda dari tanah akan sedikit lebih kecil dari radius ban, hal ini karena adanya bagian yang flexibel dari ban dan akibat berat kendaraan maka tingginya sedikit menjadi lebih rendah.

$$\text{Kecepatan} = \text{Putaran roda} \times \text{Keliling ban} \times 60 / 1.000.000 \dots\dots(2.8)$$

$$\text{Keliling Roda} = \text{Diameter Roda} / 22/7 \dots\dots\dots(2.9)$$

Untuk mencari kecepatan kendaraan secara keseluruhan dapat dicari berdasarkan persamaan berikut.

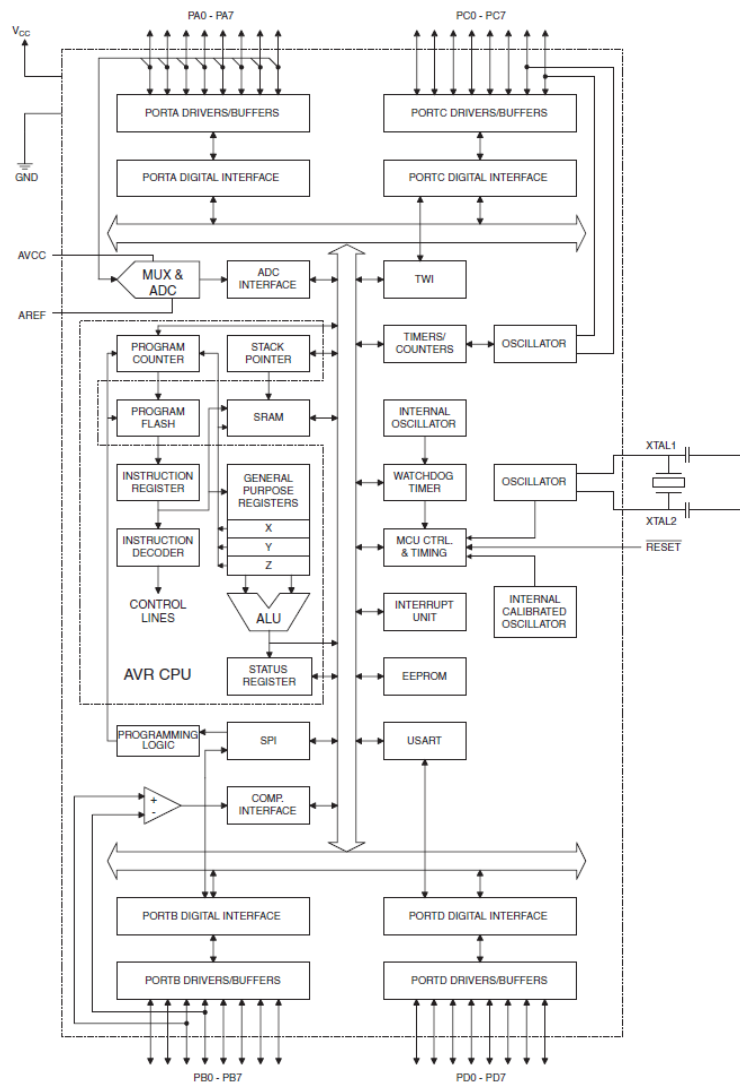
$$\text{Kecepatan Kendaraan} = \text{Jarak} / \text{Waktu} \dots\dots\dots(2.10)$$

F. Mikrokontroler ATmega 8535

Mikrokontroler adalah suatu keping *IC* dimana terdapat mikroprosesor dan memori program (*ROM*) serta memori serbaguna (*RAM*), bahkan ada beberapa jenis mikrokontroler yang memiliki fasilitas *ADC*, *PPL*, *EEPROM* dalam suatu kemasan. Penggunaan mikrokontroler dalam bidang kontrol sangat luas dan populer [6]. Ada beberapa *vendor* yang membuat mikrokontroler diantaranya Intel, Microchip, Winbond, Atmel, Philips, Xemics dan lain-lain buatan Atmel.

Mikrokontroler ATmega8535 merupakan generasi *AVR* (*Alf and Vegard's Risk processor*). Mikrokontroler *AVR* memiliki arsitektur *RISC* (*Reduced Instruction Set Computing*) 8 bit, dimana semua instruksi dalam kode 16-bit (16-bit *word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*. *AVR* menjalankan sebuah instruksi komponen eksternal dapat dikurangi. Mikrokontroler *AVR* didesain menggunakan arsitektur *Harvard*, di mana ruang dan jalur bus bagi memori program dipisahkan dengan memori data. Karena mikrokontroler ini memiliki berbagai macam fitur dan tidak terlalu sulit dalam pemrogramannya karena didukung dengan software program yang sederhana. Dalam pemrogramannya mikrokontroler ATmega8535 ini menggunakan 2 bahasa program yakni, dengan bahasa C dan bahasa assembly. Dalam penelitian ini pemrograman mikrokontroler ini menggunakan bahasa C, yang menurut penulis tidak terlalu sulit dalam pemahaman struktur bahasanya.

1. Arsitektur Mikrokontroler ATmega8535



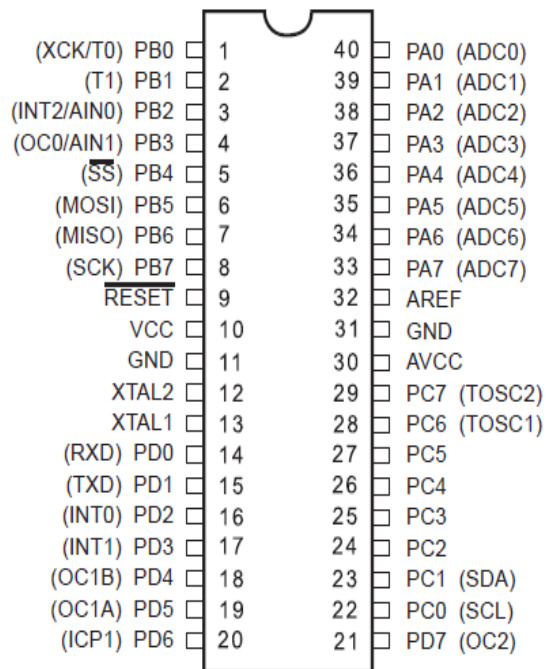
Gambar 2.5. BlokDiagram Fungsional ATmega8535 (sumber : ATMEL)

Dari Gambar 2.4 dapat dilihat bahwa ATmega8535 memiliki bagian sebagai berikut:

- a. Saluran *I/O* sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
- b. *ADC* 10 bit sebanyak 8 saluran.
- c. Tiga buah *Timer/Counter* dengan kemampuan pembanding
- d. *CPU* yang terdiri atas 32 buah register.
- e. *Watchdog Timer* dengan osilator internal.
- f. *SRAM* sebesar 512 byte.
- g. Memori *Flash* sebesar 8 Kb dengan kemampuan *Read While Write*.
- h. Unit interupsi internal dan eksternal.
- i. *Port* antarmuka *SPI*.
- j. *EEPROM*(*Electrically Erasable Programmable Read Only Memori*) sebesar 512 byte yang diprogram saat operasi.
- k. Antarmuka komparator *analog*.
- l. *Port USART* untuk komunikasi serial dengan kecepatan maksimal 12,5 Mbps.
- m. Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.

2. Konfigurasi Pin ATmega8535

Konfigurasi pin pada mikrokontroler ATmega8535 yaitu :



Gambar 2.6. Konfigurasi Pin ATmega 8535 (sumber : ATMEL)

Konfigurasi *pin* pada mikrokontroller ATmega8535 dapat dilihat pada Gambar 2.3.

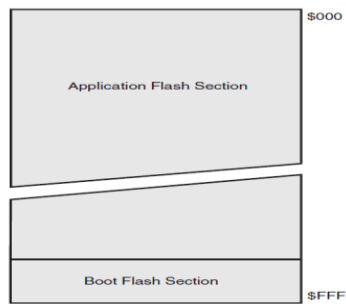
fungsi dari *pin* ATmega 8535 tersebut adalah sebagai berikut:

- a. *VCC* merupakan *Pin* yang berfungsi sebagai *pin* masukan catudaya
- b. *GND* merupakan *PinGround*
- c. *Port A* (PA0...PA7) merupakan *pin I/O* dan *pin* masukan ADC
- d. *Port B* (PB0...PB7) merupakan *pin I/O* dua arah dan *pin* yang mempunyai fungsi khusus yaitu *Timer/Counter*, komparator *analog* dan SPI
- e. *Port C* (PC0...PC7) merupakan *port I/O* dua arah dan *pin* yang mempunyai fungsi khusus, yaitu komparator *analog* dan *Timer Oscillator*

- f. *Port D (PD0...PD7)* merupakan *port I/O* dua arah dan *pin* fungsi khusus yaitu komparator *analog* dan *interrupt* eksternal serta komunikasi serial
- g. *RESET* merupakan *pin* yang digunakan untuk mengembalikan kondisi mikrokontroler seperti semula
- h. *XTAL1* dan *XTAL2* *pin* untuk eksternal *clock*
- i. *AVCC* adalah *pin* masukan untuk tegangan *ADC*
- j. *AREF* adalah *pin* masukan untuk tegangan referensi eksternal *ADC*

3. Peta Memori

Mikrokontroler ATmega8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian yaitu : 32 buah *register* umum, 64 buah *register I/O*, dan 512 byte *SRAMinternal*. *Register* untuk keperluan umum menempati ruang data pada alamat terbawah yaitu \$00 sampai \$1F. Sementara itu *register* khusus untuk menangani *I/O* dan kontrol terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari \$20 sampai \$5F. *Register* tersebut merupakan *register* yang khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral* mikrokontroler, seperti kontrol *register*, *timer/counter*, fungsi fungsi *I/O*, dan sebagainya. Alamat memori berikutnya digunakan untuk *SRAM* 512 byte, yaitu pada lokasi \$60 sampai dengan \$25F.



Gambar 2.7. Memori Data AVR ATmega8535 (Sumber : ATMEL)

Memori program yang terletak pada *Flash Perom* tersusun dalam *word* atau 2 byte karena setiap instruksi memiliki lebar 16-bit atau 32-bit. AVR ATmega8535 memiliki 4KByte x 16 Bit *Flash Perom* dengan alamat mulai dari \$000 sampai \$FFF. AVR tersebut memiliki 12 bit *Program Counter (PC)* sehingga mampu mengamati isi *Flash*. Selain itu AVR ATmega8535 juga memiliki memori data berupa *EEPROM* 8 bit sebanyak 512 byte. Alamat *EEPROM* dimulai dari \$000 sampai \$1FF.

4. Fitur-fitur dari ATmega 8535

a. ADC(*Analog to digital converter*)

Fitur yang terdapat pada mikrokontroler ATmega8535 ini berfungsi sebagai pengkonversi nilai *analog* ke *digital*, karena nilai masukan atau keluaran pada ATmega 8535 menggunakan logika *input-output* digital yakni 1/0 (*high / low*). Penggunaan ADC ini dikarenakan sensor yang digunakan pada pembacaan konversi tegangan dari nilai 0 – 5 Volt, sedangkan ATmega 8535 tidak dapat membaca nilai sensor tegangan yang bernilai 1,5 atau 3,5 Volt. Dengan menggunakan fitur ADC ini nilai pembacaan sensor tersebut dapat dibaca oleh ATmega 8535, karena fitur ADC mempunyai resolusi pembacaan dari 0-1024. Untuk pembacaan nilai sensor yang

berada di tengah-tengah antara 0-5 Volt dapat menggunakan persamaan konversi ADC dibawah ini :

$$\text{Nilai konversi} = \frac{V_{cc}}{V_{ref}} \times 1024 \quad (2.11)$$

Dimana :

Nilai konversi = nilai yang telah terkonversi dalam bentuk resolusi ADC

V_{cc} = nilai tegangan dari sensor

V_{ref} = nilai tegangan referensi pada ATmega (5 volt)

1024 = nilai resolusi 10 bit ADC ATmega 8535

Nilai hasil dari konversi tersebut kemudian dimasukkan pada bahasa program (*software*). Dari pembacaan konversi nilai sensor diatas *pin-pinoutput* yang telah ditentukan bekerja sesuai dengan kode program yang ditulis pada *software* dan seterusnya pada saat perubahan nilai sensor yang terbaca oleh *channel* ADC.

b. Timer dan Counter

Timer dan counter adalah dua fasilitas yang memiliki perangkat yang sama, seperti halnya register penampungnya (**TCNTx**). Ketika difungsikan sebagai timer, maka register penampung tersebut berisikan jumlah waktu yang terlampaui tiap selang waktu tertentu. Besar selang waktu tersebut dapat disetting sesuai dengan kebutuhan. Jika dipakai sebagai counter, maka register penampung tersebut digunakan untuk menyimpan data hasil perhitungan terakhir. Saat difungsikan sebagai counter, maka

masuk melewati pin **TO** dan **T1**. Register untuk mengatur kapan timer difungsikan sebagai timer dan kapan sebagai counter adalah **TCCR_x**.

ATMEGA8535 memiliki fasilitas 3 buah timer/counter yaitu **timer/counter0 8 bit**, **timer/counter1 16 bit**, dan **timer/counter2 8 bit**. 8 bit dan 16 bit adalah jumlah data yang bisa ditampung pada register penampungnya. Pada bab ini akan didemonstrasikan 2 aplikasi praktik, dengan tujuan peserta bisa membedakan fungsi dari timer dan counter pada mikrokontroller.

c. Interupsi

Interupsi adalah suatu kondisi dimana *microchip* akan berhenti sementara dari program utama untuk melayani atau menjalankan kode interupsi yang ditulis pada kode program interupsi, kemudian *microchip* akan menjalankan program utama. Atmega 8535 menyediakan 3 interupsi eksternal, yaitu INT0, INT1 dan INT2. Interupsi ini dapat dilakukan dengan cara memberikan *input low* (0) pada *pin* yang dipasang sebagai *pin* interupsi.

d. PWM

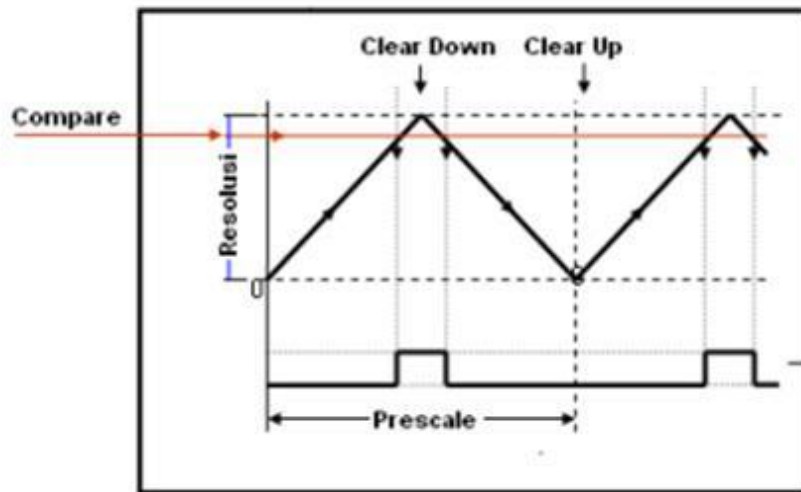
PWM merupakan suatu teknik teknik dalam mengatur kerja suatu peralatan yang memerlukan arus *pull in* yang besar dan untuk menghindari disipasi daya yang berlebihan dari peralatan yang akan dikontrol. PWM merupakan suatu metoda untuk mengatur kecepatan perputaran motor dengan cara mengatur prosentase lebar pulsa

high terhadap perioda dari suatu sinyal persegi dalam bentuk tegangan periodik yang diberikan ke motor sebagai sumber daya. Semakin besar perbandingan lama sinyal high dengan perioda sinyal maka semakin cepat motor berputar.

Sinyal PWM dapat dibangun dengan banyak cara, dapat menggunakan metode analog menggunakan rangkaian op-amp atau dengan menggunakan metode digital. Dengan metode analog setiap perubahan PWM-nya sangat halus, sedangkan menggunakan metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Misalkan PWM digital 8 bit berarti PWM tersebut memiliki resolusi $2^8 = 256$, maksudnya nilai keluaran PWM ini memiliki 256 variasi, variasinya mulai dari 0 – 255 yang mewakili *duty cycle* 0 – 100% dari keluaran PWM tersebut. Pada perancangan driver ini, sinyal PWM akan diatur secara digital yang dibangkitkan oleh mikrokontroler ATMEGA 8535.

Pengaturan PWM menggunakan mikrokontroler ATMEGA :

Proses pembangkitan sinyal PWM pada mikrokontroler AVR ATMEGA 8535 adalah sebagai berikut.

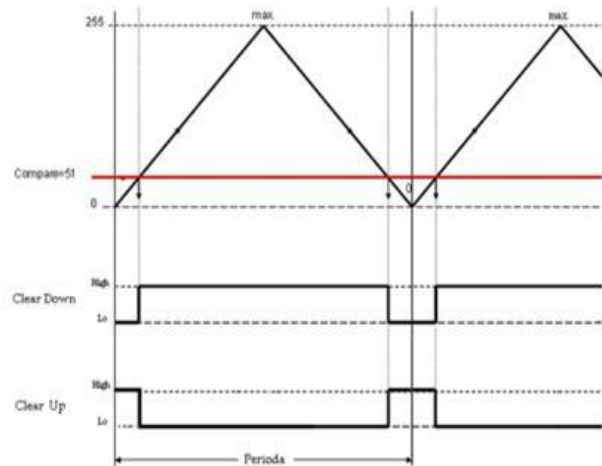


Gambar 2.8 Proses pembangkitan sinyal PWM (Sumber : Google)

Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Misalkan suatu PWM memiliki resolusi 8 bit berarti PWM ini memiliki variasi perubahan nilai sebanyak $2 \text{ pangkat } 8 = 256$ variasi mulai dari 0 – 255 perubahan nilai. Compare adalah nilai pembanding. Nilai ini merupakan nilai referensi *duty cycle* dari PWM tersebut. Nilai compare bervariasi sesuai dengan resolusi dari PWM. Dalam gambar nilai compare ditandai dengan garis warna merah, dimana posisinya diantara dasar segitiga dan ujung segitiga.

Clear digunakan untuk penentuan jenis komparator apakah komparator *inverting* atau *non-inverting*. Mikrokontroler akan membandingkan posisi keduanya, misalkan bila PWM diset pada kondisi clear down, berarti apabila garis segitiga berada dibawah garis merah (compare) maka PWM akan mengeluarkan logika 0. Begitu pula sebaliknya apabila garis segitiga berada diatas garis merah (compare) maka

PWM akan mengeluarkan logika 1. Lebar sempitnya logika 1 ditentukan oleh posisi compare, lebar sempitnya logika 1 itulah yang menjadi nilai keluaran PWM, dan kejadian ini terjadi secara harmonik terus-menerus. Maka dari itu nilai compare inilah yang dijadikan nilai *duty cycle* PWM. Clear Up adalah kebalikan (invers) dari Clear Down pada keluaran logikanya.



Gambar 2.9 Bentuk nilai compare PWM (sumber : google)

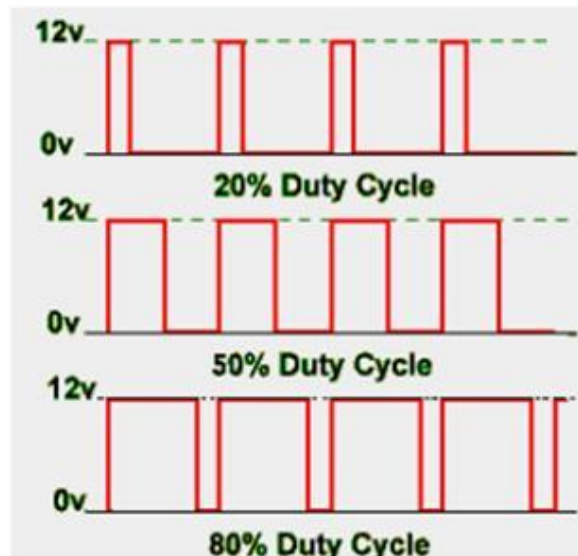
Prescale digunakan untuk menentukan waktu periode dari pada PWM. Nilai *prescale* bervariasi yaitu 1, 8, 32, 64, 128, 256, 1024. Misalkan jika *prescale* diset 64 berarti timer/PWM akan menghitung 1 kali bila clock di CPU sudah 64 kali, Clock CPU adalah clock mikrokontroler itu sendiri. Periode dari PWM dapat dihitung menggunakan rumus:

$$T = \left(\frac{1}{\text{Clock CPU}} \right) \times \text{prescale} \times \text{resolusi} \dots\dots\dots (2.7)$$

Setting prescale disini digunakan untuk mendapatkan frekuensi dan periode kerja PWM sesuai dengan spesifikasi yang dibutuhkan. **Perhitungan duty cycle PWM** dengan cara mengatur lebar pulsa “on” dan “off” dalam satu periode gelombang melalui pemberian besar sinyal referensi output dari suatu PWM akan didapat duty cycle yang diinginkan. *Duty cycle* dari PWM dapat dinyatakan sebagai :

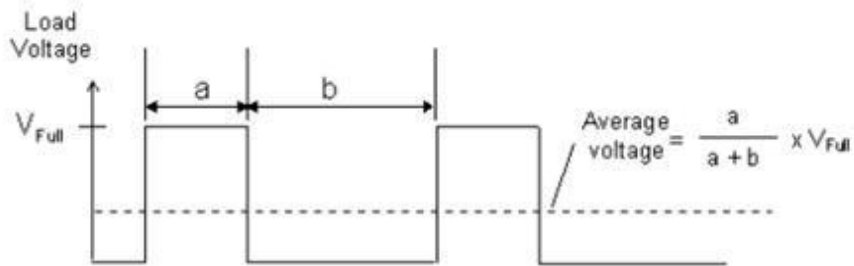
$$duty\ cycle = \frac{t_{on}}{t_{off}} \times 100\% \quad \dots\dots\dots(2.12)$$

Duty cycle 100% berarti sinyal tegangan pengatur motor dilewatkan seluruhnya. Jika tegangan catu 100V, maka motor akan mendapat tegangan 100V. pada *duty cycle* 50%, tegangan pada motor hanya akan diberikan 50% dari total tegangan yang ada, begitu seterusnya.



Gambar 2.10 Duty Cycle (sumber : google)

Perhitungan Pengontrolan tegangan output motor dengan metode PWM cukup sederhana.



Gambar 2.11 Tegangan keluaran PWM (sumber : google)

Dengan menghitung *duty cycle* yang diberikan, akan didapat tegangan output yang dihasilkan. Sesuai dengan rumus yang telah dijelaskan pada gambar.

$$\text{Average Voltage} = \frac{a}{a+b} \times V_{full} \quad \dots\dots\dots (2.13)$$

Average voltage merupakan tegangan output pada motor yang dikontrol oleh sinyal PWM. *a* adalah nilai *duty cycle* saat kondisi sinyal “on”. *b* adalah nilai *duty cycle* saat kondisi sinyal “off”. V_{full} adalah tegangan maximum pada motor. Dengan menggunakan rumus diatas, maka akan didapatkan tegangan output sesuai dengan sinyal kontrol PWM yang dibangkitkan. Untuk menentukan tegangan keluaran berdasarkan pengaturannya dapat dituliskan :

$$255/PWM \text{ set} = V_{in}/V_{set} \quad \dots\dots\dots (2.14)$$

G. Pemrograman Mikrokontroller

1. Bahasa C

Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories [7]. Meskipun C dibuat untuk memprogram sistem dan jaringan komputer namun bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. C secara luar biasa memengaruhi bahasa populer lainnya, terutama C++ yang merupakan ekstensi dari C.

Berikut ini adalah contoh program sederhana yang akan mencetak kalimat "Hello, World!" dengan menggunakan pustaka `stdio.h` (ANSI C):

```
#include <stdio.h>
#include <conio.h>
int main(void) {
    printf("Hello, World!\n");
    return 0;
}
```

2. CodeVision AVR

Dalam memprogram chip mikrokontroller atmega8535 pada proyek tugas akhir ini digunakan software CodeVision AVR. Code vision AVR merupakan Software untuk membuat code program microcontroller AVR. kebanyakan programmer memakai software ini karena fasilitas-fasilitas yang disediakan CodeVision AVR sangatlah

memudahkan bagi programmer dalam membuat code. HP InfoTech menyajikan versi baru (lebih dari 9500 pengguna terdaftar) yang paling populer C Compiler komersial untuk Atmel AVR.

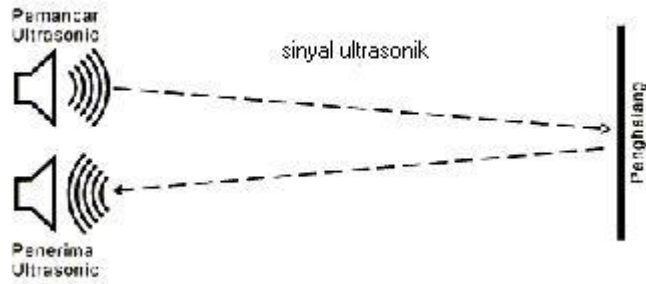
H. Sensor Ultrasonik SRF05



Gambar 2.12 SRF05 (Sumber : Parallax)

Gelombang ultrasonik adalah gelombang dengan besar frekuensi diatas frekuensi gelombang suara yaitu lebih dari 20 KHz. Seperti telah disebutkan bahwa sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).

Prinsip kerja dari sensor ultrasonik dapat ditunjukkan dalam gambar dibawah ini :



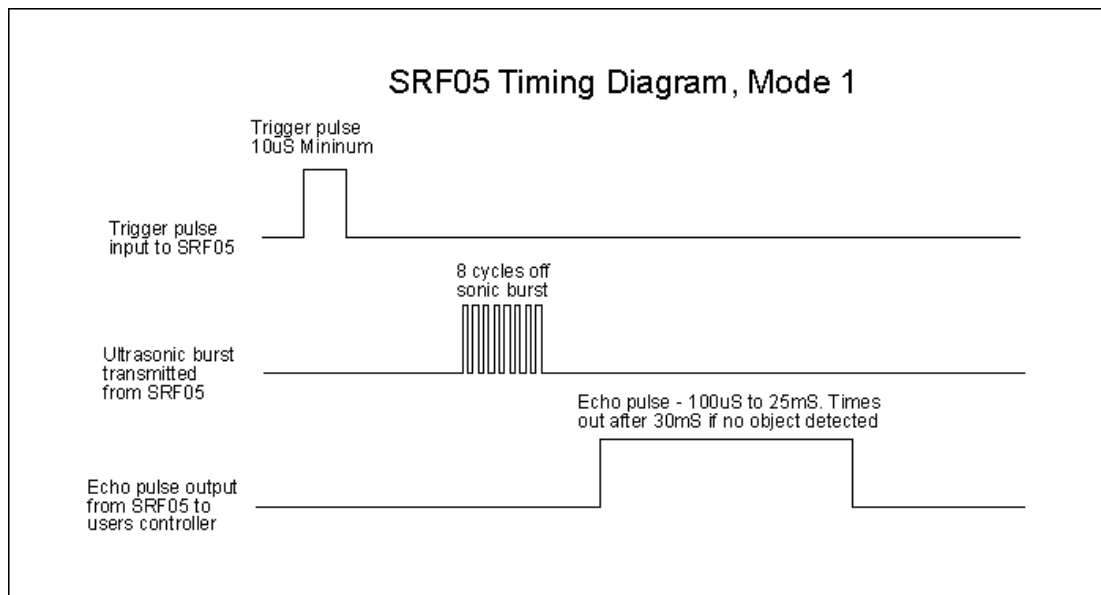
Gambar 2.13 Prinsip Kerja Sensor Ultrasonik (sumber : google)

Prinsip kerja dari sensor ultrasonik adalah sebagai berikut :

1. Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20kHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40kHz. Sinyal tersebut di bangkitkan oleh rangkaian pemancar ultrasonik.
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal / gelombang bunyi dengan kecepatan bunyi yang berkisar 340 m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima Ultrasonik.
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya. Jarak dihitung berdasarkan rumus :

$$S = 340.t/2 \dots\dots (2.15)$$

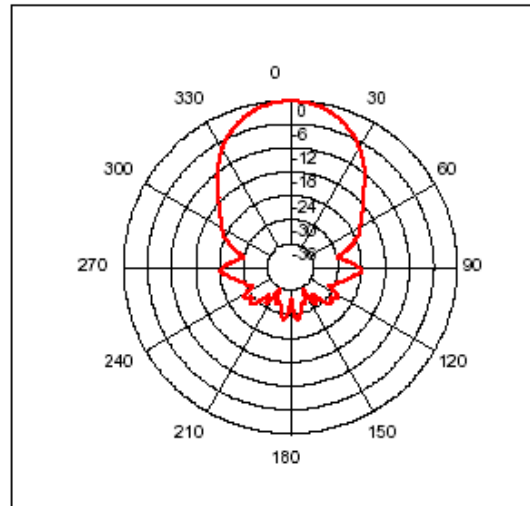
dimana S adalah jarak antara sensor ultrasonik dengan bidang pantul, dan t adalah selisih waktu antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik.



Gambar 2.14 SRF05 timing diagram (sumber : parallax)

SRF05 adalah sensor non-kontak pengukur jarak menggunakan ultrasonik. Prinsip kerja sensor ini adalah transmitter mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan obyek, sehingga jarak sensor dengan obyek dapat ditentukan persamaan $\text{jarak} = \text{kecepatan_suara} \times \text{waktu_pantul} / 2$. SRF05 dapat mengukur jarak dalam rentang antara 3 cm – 4 m dengan output panjang pulsa yang sebanding dengan jarak obyek. Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu *TRIGGER*

dan *ECHO*. Untuk mengaktifkan SRF05 mikrokontroler mengirimkan pulsa positif melalui pin *TRIGGER* minimal 10 us, selanjutnya SRF04 akan mengirimkan pulsa positif melalui pin *ECHO* selama 100 us hingga 18 ms, yang sebanding dengan jarak obyek.



Gambar 2.15 SRF05 Jangkauan (sumber : Parallax)

Ultrasonik bekerja berdasarkan efek Doppler. **Efek Doppler**, dinamakan mengikuti tokoh fisika, Christian Andreas Doppler, adalah perubahan frekuensi atau panjang gelombang dari sebuah sumber gelombang yang diterima oleh pengamat, jika sumber suara/gelombang tersebut bergerak relatif terhadap pengamat/pendengar. Untuk gelombang yang umum dijumpai, seperti gelombang suara yang menjalar dalam medium udara, perhitungan dari perubahan frekuensi ini, memerlukan kecepatan pengamat dan kecepatan sumber relatif terhadap medium di mana gelombang itu disalurkan.

Efek Doppler total, f , dapat merupakan hasil superposisi dari gerakan sumber dan/atau gerakan pengamat, sesuai dengan rumusan berikut:

$$f = \left(\frac{v + v_r}{v + v_s} \right) f_0 \dots\dots\dots (2.16)$$

di mana

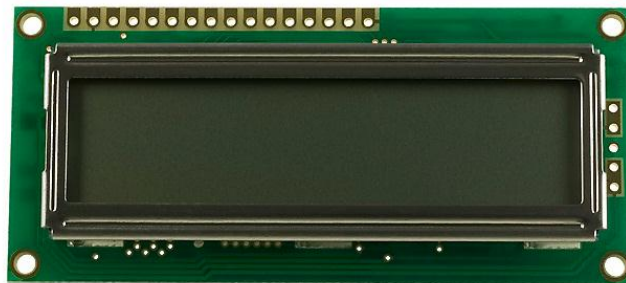
v adalah kecepatan Rambat gelombang

v_s adalah kecepatan sumber gelombang relatif terhadap medium; positif jika pengamat mendekati sumber gelombang/suara.

v_r adalah kecepatan pengamat (receiver) relatif terhadap medium; positif jika sumber menjauhi pengamat.

I. LCD

LCD merupakan suatu jenis penampil (*display*) yang menggunakan *Liquid Crystal* sebagai media refleksinya. *LCD* juga sering digunakan dalam perancangan alat yang menggunakan mikrokontroler. *LCD* dapat berfungsi untuk menampilkan suatu nilai hasil sensor ,menampilkan *teks*, atau menampilkan menu pada aplikasi mikrokontroler. Tergantung dengan perintah yang ditulis pada mikrokontroler.



Gambar 2.16 LCD 2 x 16 Karakter (sumber : depok.instrument)

LCD yang akan digunakan dalam pembuatan alat prototype ini adalah *LCD* dengan tipe karakter 2 x 16 yaitu alat penampil yang dibuat pabrikan umum dijual dipasaran standar dan dapat menampilkan karakter 2 baris dengan tiap baris 16 karakter. Pada pembuatan alat ini *LCD* akan digunakan sebagai penampil jarak antara bagian depan mobil dengan media didepanya.

J. BUZER



Gambar 2.17 Buzzer (sumber : google)

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).

K. Motor DC

Motor DC merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik [8]. Energi mekanik ini digunakan untuk, misalnya, memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (mixer, bor listrik, fan angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industri.

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

1. Tegangan dinamo – meningkatkan tegangan dinamo akan meningkatkan kecepatan
2. Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan rolling mills, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak

berbahaya sebab resiko percikan api pada sikatnya. Motor DC juga relatif mahal dibanding motor AC.

Hubungan antara kecepatan, flux medan dan tegangan dinamo ditunjukkan dalam persamaan berikut:

$$\text{Gaya elektromagnetik: } E = K\Phi N \dots\dots\dots (2.17)$$

$$\text{Torque: } T = K\Phi I_a \dots\dots\dots (2.18)$$

$$N = \frac{V - I_a R_a}{K \Phi} \dots\dots\dots (2.19)$$

Dimana:

E = gaya elektromagnetik yang dikembangkan pada terminal dinamo (volt)

Φ = flux medan yang berbanding lurus dengan arus medan

N = kecepatan dalam RPM (putaran per menit)

T = *torque* elektromagnetik

I_a = arus dinamo

K = konstanta persamaan

V = Tegangan Terminal

R_a = Tahanan Jangkar

1. Mekanisme Kerja Motor DC

Mekanisme kerja untuk seluruh jenis motor secara umum sama

1. Arus listrik dalam medan magnet akan memberikan gaya

2. Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran/*loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
3. Pasangan gaya menghasilkan tenaga putar/ *torque* untuk memutar kumparan.
4. Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban mengacu kepada keluaran tenaga putar/ *torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan kedalam tiga kelompok :

- a. ***Beban torque konstan*** adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torque* nya tidak bervariasi. Contoh beban dengan *torque* konstan adalah *conveyors*, *rotary kilns*, dan pompa *displacement* konstan.
- b. ***Beban dengan variabel torque*** adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa sentrifugal dan fan (*torque* bervariasi sebagai kwadrat kecepatan).
- c. ***Beban dengan energi konstan*** adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

2. Komponen Utama Motor DC

Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan *torque* yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas.

Sebuah motor DC yang memiliki tiga komponen utama:

- **a. Kutub medan.** Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.
- **b. Dinamo.** Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.

c. *Commutator*. Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.

3. Jenis Motor DC

a. Motor DC Sumber Daya Terpisah/ *Separately Excited*

Jika arus medan dipasok dari sumber terpisah maka disebut motor DC sumber daya terpisah/ *separately excited*.

b. Motor DC Sumber Daya Sendiri/ *Self Excited*: motor *shunt*

Pada motor *shunt*, gulungan medan (medan *shunt*) disambungkan secara paralel dengan gulungan dinamo. Oleh karena itu total arus dalam jalur merupakan penjumlahan arus medan dan arus dinamo.

Berikut tentang kecepatan motor *shunt* :

1. Kecepatan pada prakteknya konstan tidak tergantung pada beban (hingga *torque* tertentu setelah kecepatannya berkurang, lihat Gambar 4) dan oleh karena itu cocok untuk penggunaan komersial dengan beban awal yang rendah, seperti peralatan mesin.
2. Kecepatan dapat dikendalikan dengan cara memasang tahanan dalam susunan seri dengan dinamo (kecepatan berkurang) atau dengan memasang tahanan pada arus medan (kecepatan bertambah).

c. Motor DC daya sendiri: motor seri

Dalam motor seri, gulungan medan (medan *shunt*) dihubungkan secara seri dengan gulungan dinamo. Oleh karena itu, arus medan sama dengan arus dinamo. Berikut tentang kecepatan motor seri (Rodwell International Corporation, 1997; L.M. Photonics Ltd, 2002):

1. Kecepatan dibatasi pada 5000 RPM
2. Harus dihindarkan menjalankan motor seri tanpa ada beban sebab motor akan mempercepat tanpa terkendali.

Motor-motor seri cocok untuk penggunaan yang memerlukan *torque* penyalan awal yang tinggi, seperti derek dan alat pengangkat *hoist*.

d. Motor DC Kompon/Gabungan

Motor Kompon DC merupakan gabungan motor seri dan *shunt*. Pada motor kompon, gulungan medan (medan *shunt*) dihubungkan secara paralel dan seri dengan gulungan dynamo (A) seperti yang ditunjukkan dalam gambar 6. Sehingga, motor kompon memiliki *torque* penyalan awal yang bagus dan kecepatan yang stabil. Makin tinggi persentase penggabungan (yakni persentase gulungan medan yang dihubungkan secara seri), makin tinggi pula *torque* penyalan awal yang dapat ditangani oleh motor ini. Contoh, penggabungan 40-50% menjadikan motor ini cocok untuk alat pengangkat *hoist* dan derek, sedangkan motor kompon yang standar (12%) tidak cocok.

L. Resistor

Resistor adalah sebuah komponen elektronika yang berfungsi menghambat arus dalam suatu rangkaian listrik [9]. Resistor biasa disebut juga tahanan atau pelawan. Yang paling banyak digunakan adalah resistor yang terbuat dari bahan karbon yang banyak terdapat di pasaran memiliki harga 0,1 Ω sampai 100 M Ω dengan tarif 1/8 Wattsampai 20 Watt.

Sesuai dengan namanya resistor bersifat resistif dan umumnya terbuat dari bahan karbon. Dari hukum Ohm diketahui, resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya.

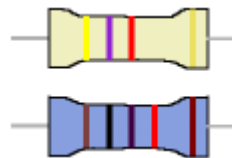
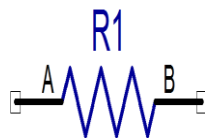
$$R = \frac{V}{I} \quad (2.20)$$

Dimana :

R = resitansi (Ohm)

V = tegangan (Volt)

I = arus (Ampere)



Gambar 2.18 Simbol dan Bentuk Fisik Resistor (sumber : google)

Satuan resistansi dari suatu resistor disebut Ohm atau dilambangkan dengan simbol Ω (Omega). Tipe resistor yang umum adalah berbentuk tabung dengan dua kaki tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang kode warna untuk memudahkan dalam mengenali besarnya resistansi pada resistor tersebut tanpa mengukur besarnya dengan Ohmmeter.

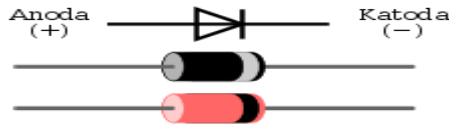
Tabel.2.1. Kode Nilai Warna Pada Resistor

Warna	Pita pertama	Pita kedua	Pita ketiga (pengali)	Pita keempat (toleransi)	Pita kelima (koefisien suhu)
Hitam	0	0	$\times 10^0$		
Cokelat	1	1	$\times 10^1$	$\pm 1\%$ (F)	100 ppm
Merah	2	2	$\times 10^2$	$\pm 2\%$ (G)	50 ppm
Oranye	3	3	$\times 10^3$		15 ppm
Kuning	4	4	$\times 10^4$		25 ppm
Hijau	5	5	$\times 10^5$	$\pm 0.5\%$ (D)	
Biru	6	6	$\times 10^6$	$\pm 0.25\%$ (C)	
Ungu	7	7	$\times 10^7$	$\pm 0.1\%$ (B)	
Abu-abu	8	8	$\times 10^8$	$\pm 0.05\%$ (A)	
Putih	9	9	$\times 10^9$		
Emas			$\times 10^{-1}$	$\pm 5\%$ (J)	
Perak			$\times 10^{-2}$	$\pm 10\%$ (K)	
Kosong				$\pm 20\%$ (M)	

M. Dioda

Dioda mempunyai dua [elektrode](#) yang aktif dimana arus listrik dapat mengalir dari anoda ke katoda, dan kebanyakan dioda digunakan karena karakteristik satu arah yang dimilikinya. Fungsi paling umum dari dioda adalah untuk mengalirkan arus

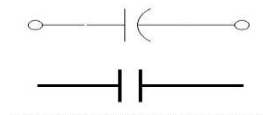
listrik mengalir dalam suatu arah yang sering juga disebut kondisi bias maju (*forward*) dan untuk menahan arus dari arah sebaliknya sering disebut kondisi bias mundur (*reverse*).



Gambar 2.19 Schematic dan Konstruksi Dioda (sumber : google)

Dioda akan bekerja pada saat tegangan *breakdown* diberi tegangan minimal 0.7 Volt agar arus listrik dapat mengalir dari anoda ke katoda. Dioda sering juga digunakan sebagai penyearah tegangan dari AC ke DC dan juga sering digunakan pada rangkaian catu daya (*power supply*).

N. Kapasitor



Gambar 2.20 Simbol Kapasitor (sumber : google)

Kapasitor merupakan komponen pasif elektronika yang sering dipakai didalam merancang suatu sistem yang berfungsi untuk mengeblok arus DC, Filter, dan menyimpan energi listrik. Didalamnya 2 buah pelat elektroda yang saling berhadapan dan dipisahkan oleh sebuah *insulator*. Sedangkan bahan yang digunakan sebagai *insulator* dinamakan dielektrik. Ketika kapasitor diberikan tegangan DC maka energi listrik disimpan pada tiap elektrodanya. Selama kapasitor melakukan pengisian, arus

mengalir. Aliran arus tersebut akan berhenti bila kapasitor telah penuh. Yang membedakan tiap - tiap kapasitor adalah dielektriknya. Jenis-jenis kapasitor yang banyak dipakai antara lain *Electrolytic Capacitor, Tantalum Capacitor, Ceramic Capacitor, Multilayer Ceramic Capacitor, Polyester Film Capacitor, Polypropylene Capacitor, Kapasitor Mika, Polystyrene Film Capacitor, Electric Double Capacitor (Super Capacitor), Trimmer Capacitor, Tuning Capacitor*

1. Nilai Kapasitor

Untuk mencari nilai dari kapasitor biasanya dilakukan dengan melihat angka/kode yang tertera pada badan kapasitor tersebut. Untuk kapasitor jenis elektrolit memang mudah, karena nilai kapasitansinya telah tertera dengan jelas pada tubuhnya. Sedangkan untuk kapasitor keramik dan beberapa jenis yang lain nilainya dikodekan. Biasanya kode tersebut terdiri dari 4 digit, dimana 3 digit pertama merupakan angka dan digit terakhir berupa huruf yang menyatakan toleransinya. Untuk 3 digit pertama angka yang terakhir berfungsi untuk menentukan 10^n , nilai n dapat dilihat pada tabel dibawah.

Tabel 2.2 Tabel nilai kapasitor

3rd Digit	Multiplier	Letter	Tolerance
0	1	D	0.5 pF
1	10	F	1 %
2	100	G	2 %
3	1,000	H	3 %
4	10,000	J	5 %
5	100,000	K	10 %
6,7	Not Used	M	20 %
8	.01	P	+100, -0 %
9	.1	Z	+80, -20 %

Misalnya suatu kapasitor pada badannya tertulis kode 474J, berarti nilai kapasitansinya adalah $47 + 10^4 = 470.000 \text{ pF} = 0.47 \mu\text{F}$ sedangkan toleransinya 5%. Yang harus diingat didalam mencari nilai kapasitor adalah satuannya dalam pF (*Pico Farad*).