

BAB II

LANDASAN TEORI

2.1. Pengertian Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC, sehingga sering disebut *single chip microcomputer*. Mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan *PC* (*Personal Computer*) yang memiliki beragam fungsi. Perbedaan lainnya adalah perbandingan *RAM* dan *ROM* yang sangat berbeda antara komputer dengan mikrokontroler.

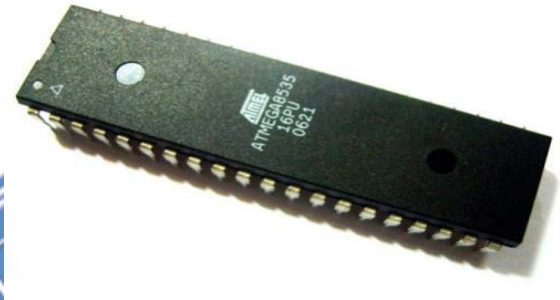
Mikrokontroler adalah sebuah *system microprocessor* dimana didalamnya sudah terdapat *CPU*, *ROM*, *RAM*, *I/O*, *Clock* dan peralatan *internal* lainnya yang sudah saling terhubung dan terorganisasi (teralamat) dengan baik oleh pabrik pembuatnya dan dikemas dalam satu *chip* yang siap pakai. Sehingga kita tinggal memprogram isi *ROM* sesuai aturan penggunaan oleh pabrik yang membuatnya menurut Winoto (2008:3).

Teknologi yang digunakan pada mikrokontroler *AVR* berbeda dengan mikrokontroler seri *MCS-51*. *AVR* berteknologi *RISC* (*Reduced Instruction Set Computer*), sedangkan seri *MCS-51* berteknologi *CISC* (*Complex Instruction Set Computer*). Mikrokontroler *AVR* dapat dikelompokkan menjadi empat kelas, yaitu keluarga *ATtiny*, keluarga *AT90Sxx*, keluarga *ATMega*, dan keluarga *AT89RFxx*. Pada dasarnya yang membedakan masing-masing kelas adalah memori, kelengkapan peripheral dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan mereka bisa dikatakan hampir sama. Oleh karena itu, dipergunakan salah satu *AVR* produk *Atmel*, yaitu *ATMega8535*.

2.1.1 Mikrokontroler AVR ATMega8535

Mikrokontroler adalah suatu keping *IC* dimana terdapat mikroprosesor dan memori program (disebut: *ROM*) serta memori serba-guna (disebut: *RAM*), bahkan ada beberapa jenis mikrokontroler yang memiliki fasilitas *ADC*, *PLL*, *EEPROM* dalam satu kemasan. Penggunaan mikrokontroler dalam bidang kontrol sangat luas dan populer. Ada beberapa vendor yang membuat mikrokontroler

diantaranya *Intel, Microchip, Winbond, Atmel, Philips, Xemics* dan lain - lain. Dari beberapa vendor tersebut, yang paling populer digunakan adalah mikrokontroler buatan *Atmel*. Adapun mikrokontroler *ATMega8535* dapat dilihat pada gambar 2.1 dibawah ini :



Gambar 2.1: Mikrokontroler AVR ATMega8535

ATMega8535 adalah mikrokontroler *CMOS 8 bit* daya rendah berbasis arsitektur *RISC*. Instruksi dikerjakan pada satu siklus *clock*, *ATMega8535* mempunyai *throughput* mendekati *1 MIPS* per MHz, hal ini membuat *ATMega8535* dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah. Mikrokontroler *ATmega8535* memiliki beberapa fitur atau spesifikasi yang menjadikannya sebuah solusi pengendali yang efektif untuk berbagai keperluan. Fitur-fitur tersebut antara lain:

- 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 perbandingan.
- 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* sebesar 512 *byte* yang dapat diprogram saat operasi.
- k) Antar-muka *komparator analog*.
- l) *Port USART* untuk komunikasi serial.

Mikrokontroler AVR (*Advance Versatile RISC processor*) memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-bits word) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock, berbeda dengan instruksi MCS 51 yang membutuhkan 12 siklus clock. Tentu saja itu terjadi karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda.

Selain mudah didapatkan dan lebih murah ATmega8535 juga memiliki fasilitas yang lengkap. Untuk tipe AVR ada 3 jenis yaitu AT Tiny, AVR klasik, ATmega. Perbedaannya hanya pada fasilitas dan I/O yang tersedia serta fasilitas lain seperti ADC, EEPROM dan lain sebagainya. Salah satu contohnya adalah ATmega8535. Memiliki teknologi RISC dengan kecepatan maksimal 16 MHz membuat ATmega8535 lebih cepat bila dibandingkan dengan varian MCS 51. Dengan fasilitas yang lengkap tersebut menjadikan ATmega8535 sebagai mikrokontroler yang powerfull.

2.1.2 Konfigurasi Pin ATmega8535

Mikrokontroler AVR ATmega8535 memiliki 40 pin dengan 32 pin diantaranya digunakan sebagai port paralel. Satu port paralel terdiri dari 8 pin, sehingga jumlah port pada mikrokontroler adalah 4 port, yaitu port A, port B, port C dan port D. Sebagai contoh adalah port A memiliki pin antara port A.0 sampai dengan port A.7, demikian selanjutnya untuk port B, port C, port D. Diagram pin mikrokontroler dapat dilihat pada gambar 2.2 berikut:

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5
(TXD) PD1	15	26	PC4
(INT0) PD2	16	25	PC3
(INT1) PD3	17	24	PC2
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

Gambar 2.2: Konfigurasi Pin ATmega8535

Berikut ini tabel 2.1 penjelasan mengenai *pin* yang terdapat pada mikrokontroler *ATMega8535*.

Tabel 2.1: Penjelasan *Pin ATMega8535*

VCC	Tegangan suplai (5 Volt)
GND	<i>Ground</i>
<i>Reset</i>	<i>Input reset level rendah, pada pin ini selama lebih dari panjang pulsa minimum akan menghasilkan reset walaupun clock sedang berjalan. RST pada pin 9 merupakan reset dari AVR. Jika pada pin ini diberi masukan low selama minimal 2 machine cycle maka sistem akan di-reset</i>
XTAL 1	<i>Input penguat osilator inverting dan input pada rangkaian operasi clock internal</i>
XTAL 2	<i>Output dari penguat osilator inverting</i>
Avcc	<i>Pin tegangan suplai untuk port A dan ADC. Pin ini harus dihubungkan ke Vcc walaupun ADC tidak digunakan, maka pin ini harus dihubungkan ke Vcc melalui low pass filter</i>
Aref	<i>pin referensi tegangan analog untuk ADC</i>
AGND	<i>pin untuk analog ground. Hubungkan kaki ini ke GND, kecuali jika board memiliki analog ground yang terpisah</i>

Fitur *ATMega8535* kapabilitas detail dari *ATMega8535* adalah sebagai berikut:

1. Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 Mhz.
2. Kapabilitas *memory flash* 8KB, *SRAM* sebesar 512 byte, dan *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte.

3. *ADC internal* dengan *fidelitas 10 bit* sebanyak *8 channel*.
4. Portal komunikasi *serial (USART)* dengan kecepatan maksimal *2,5 Mbps*.
5. Enam pilihan mode *sleep* menghemat penggunaan daya listrik.
6. Berperformen tinggi dan dengan konsumsi daya rendah (*low power*).
7. Fitur Peripheral
 - a) Dua *Timer/Counter 8-bit* dengan *Separate Prescaler* (sumber *clock* yang dapat diatur) dan *Mode* pembanding.
 - b) Satu *Timer/Counter 16-bit* dengan *Separate Prescaler*, *Mode* pembanding dan *Capture Mode*.
 - c) *Real Time Counter* dengan sumber osilator terpisah.
 - d) Terdapat delapan saluran *ADC* dengan resolusi sepuluh bit *ADC*.
 - e) Empat saluran *Pulse Width Modulation (PWM)*.
 - f) Terdapat *Two Serial Interface*.
 - g) *Programmable serial USART*.
 - h) *Master/Serial SPI Serial Interface*.
 - i) *Programmable Watchdog Timer* dengan *On-Chip Oscillator*.
 - j) *On-Chip Analog Comparator*.
8. I/O dan kemasan
 - a) *32 programmable* saluran I/O.
 - b) *40 pin PDIP*, *44 pin TQFP*, *44 PIN PLCC* dan *44 pin MLF*.
9. Tegangan kerja
 - a) *2,7 – 5,5V* untuk *ATmega8535L*.
 - b) *4,5 – 5,5V* untuk *ATmega8535*.
10. Kelas kecepatan
 - a) *0 – 8 Mhz* untuk *ATmega8535L*
 - b) *0 – 16 Mhz* untuk *ATmega8535*

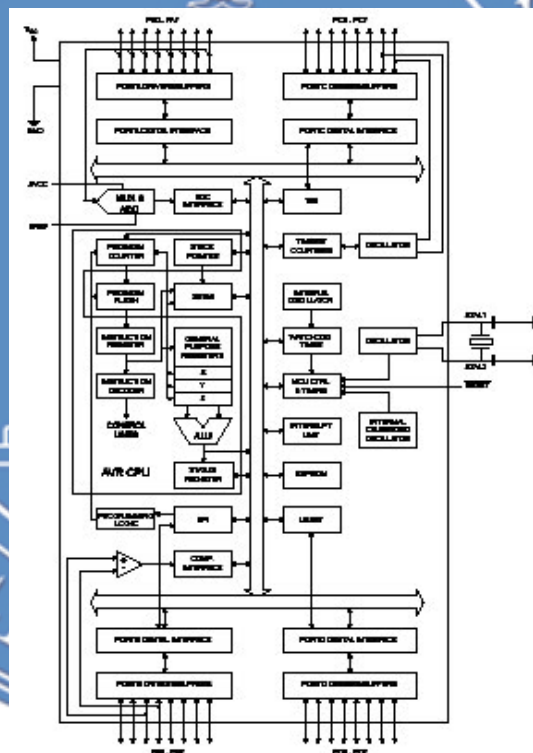
Konfigurasi *pin ATMega8535* konfigurasi *pin ATMega8535* bisa dilihat pada gambar 2.2 Dari gambar tersebut dapat dijelaskan secara *funksional* konfigurasi *ATMega8535* menurut *port*-nya masing-masing sebagai berikut:

1. *Port A (PA0..PA7)* merupakan *pin I/O* dua arah dan pin masukan *ADC*.
2. *Port B (PB0..PB7)* merupakan *pin I/O* dua arah dan *pin* fungsi khusus, yaitu *Timer/Counter*, *komparator analog*, dan *SPI*.

3. Port C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog dan Timer Oscillator.
4. Port D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial.

2.1.3 Diagram Blok ATmega8535

Pada diagram blok ATmega8535 digambarkan 32 *general purpose Working register* yang dihubungkan secara langsung dengan *Arithmetic Logic Unit (ALU)*. Sehingga memungkinkan dua *register* yang berbeda dapat diakses dalam satu siklus *clock*. Adapun diagram bloknya dapat dilihat pada gambar 2.3 berikut :

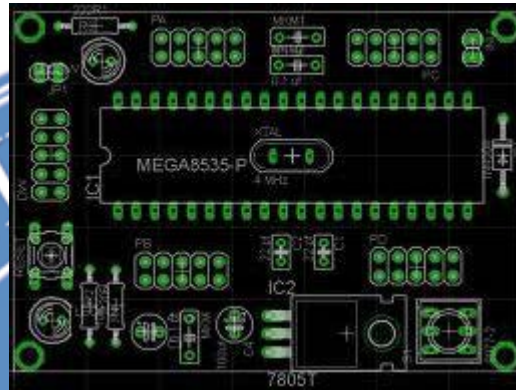


Gambar 2.3: Diagram Blok ATmega8535

2.2. Sistem Minimum ATmega8535

Sistem minimum (sismin) mikrokontroler adalah rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC mikrokontroler. Sistem ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi

tertentu. Dikeluarga mikrokontroler AVR, seri 8535 adalah salah satu seri yang sangat banyak digunakan. Mikrokontroler *ATMega8535* telah dilengkapi dengan *osilator internal*, sehingga tidak diperlukan kristal atau *resonator eksternal* untuk sumber *clock CPU*. Namun osilator ini maksimal 8 MHz jadi disarankan untuk tetap memakai kristal *eksternal*. Skema rangkaian sistem minimum dapat dilihat pada gambar 2.4 berikut :



Gambar 2.4: Sistem Minimum

Osilator internal oleh pabriknya telah di-setting 1 MHz, dan untuk merubahnya perlu merubah *setting* pada *fuse bit*. Namun untuk pengaturan *fuse bit* perlu berhati-hati, sebab pengaturan ini begitu rawan karena bila salah menyettingnya bisa menyebabkan mikrokontroler rusak. Sistem minimum AVR sangat sederhana dimana hanya menghubungkan *VCC* dan *AVCC* ke +5 V dan *GND* dan *AGND* ke *ground* serta *pin reset* tidak dihubungkan apa-apa (diambangkan). *Chip* akan *reset* jika tegangan nol atau *pin reset* dipaksa nol dan ini merupakan sistem minimum tanpa memakai kristal. Untuk yang memakai kristal rangkaian di atas ditambah kristal pada *pin XTAL1* dan *XTAL2*.

2.3. LDR (Light Dependent Resistor)

LDR merupakan resistor yang nilai resistansinya dapat diatur (*adjust*) berdasarkan besarnya intensitas cahaya yang diterima pada bagian *photoconductive* dipermukaan atasnya. Oleh sebab itu komponen ini disebut dengan *Light Dependent Resistor (LDR)* karena nilai resistansinya bergantung

pada ada dan tidaknya cahaya yang diterima. Bentuk dan *symbol* LDR dapat dilihat pada gambar 2.5 dan 2.6 berikut :

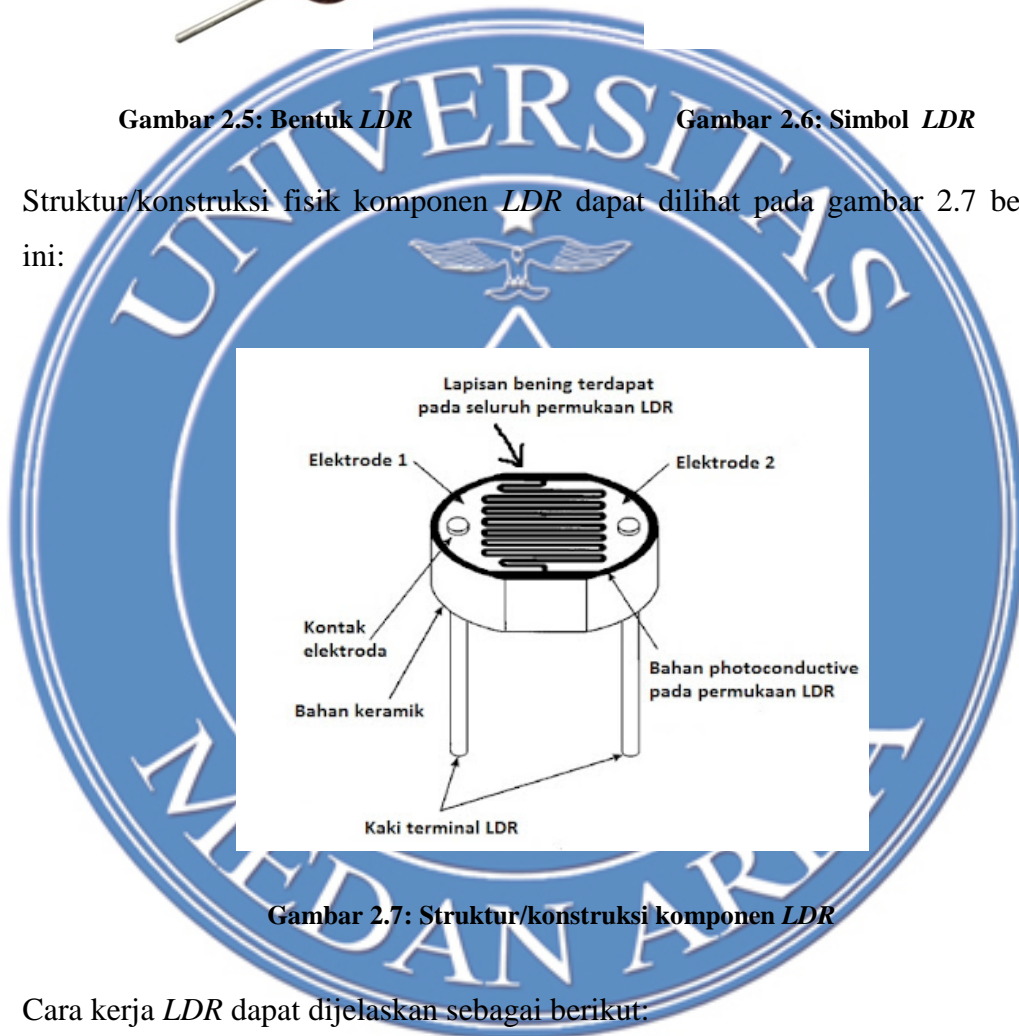


Gambar 2.5: Bentuk LDR



Gambar 2.6: Simbol LDR

Struktur/konstruksi fisik komponen LDR dapat dilihat pada gambar 2.7 berikut ini:



Gambar 2.7: Struktur/konstruksi komponen LDR

Cara kerja LDR dapat dijelaskan sebagai berikut:

1. LDR akan menghambat aliran arus listrik apabila pada tidak ada berkas cahaya yang mengenai permukaan atas LDR.
2. Apabila intensitas cahaya yang mengenai permukaan LDR kecil (sedikit), maka nilai resistansi LDR akan besar ($R_{LDR} = \text{besar}$).
3. Apabila intensitas cahaya yang mengenai permukaan LDR besar (banyak), maka nilai resistansi LDR akan kecil ($R_{LDR} = \text{kecil}$).

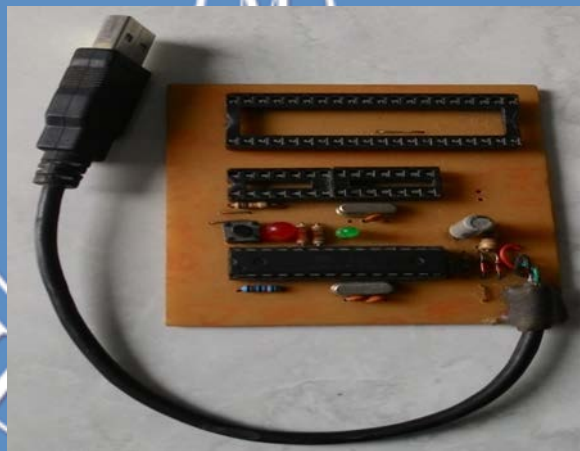
2.4. Rangkaian USB Downloader

Downloader USB ini dibuat oleh *Thomas Fischl* memanfaatkan *driver USB* dengan *AVR* yang sedang dikembangkan oleh *Objective Development GmbH*. *Downloader USB* ini diberi nama *USBaps*. *USBaps* terdiri atas *ATmega48/8* dan beberapa komponen pasif tanpa membutuhkan komponen *driver* lainnya. *Downloader* berfungsi untuk mengirim data berformat “*HEX*” ke dalam mikrokontroler yang ingin diprogram.

USB aps memiliki beberapa fitur antara lain :

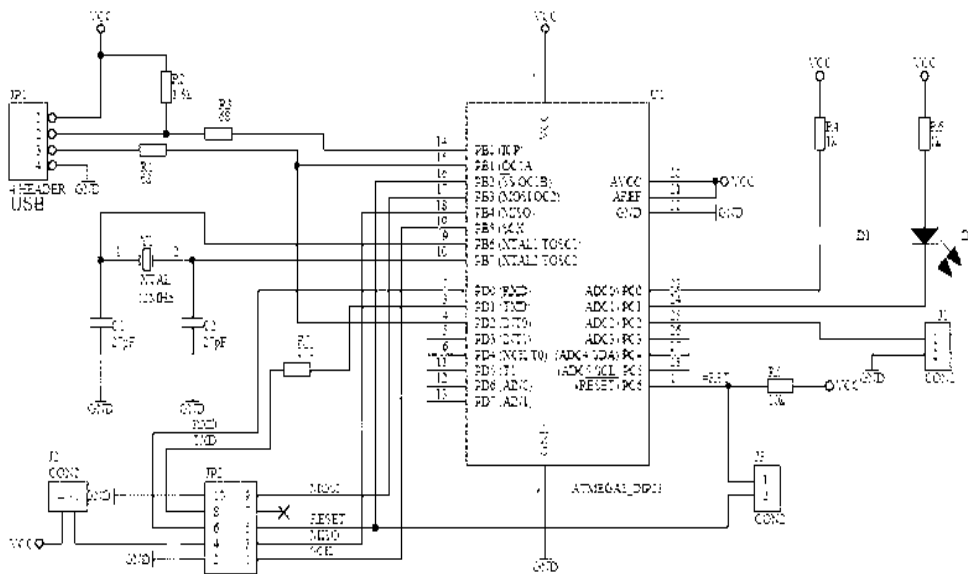
1. Dapat bekerja disystem operasi *Linux, Mac OS X, dan Windows*.
2. Tidak membutuhkan *controller* khusus atau komponen *SMD*.
3. Kecepatan *programming* sampai *5 kbyte/sec*.
4. Pilihan *SCK* untuk mendukung *target device* dengan kecepatan rendah (<1,5 *MHz*).

Berikut ini rangkaian *USB downloader* dapat dilihat pada gambar 2.8 berikut :



Gambar 2.8 : Rangkaian USB Downloader

Adapun skema rangkaian *USB downloader* dapat dilihat pada gambar 2.9 berikut ini :



Gambar 2.9: Skema Rangkaian USB Downloader

2.5. LED (Light Emitting Diode)

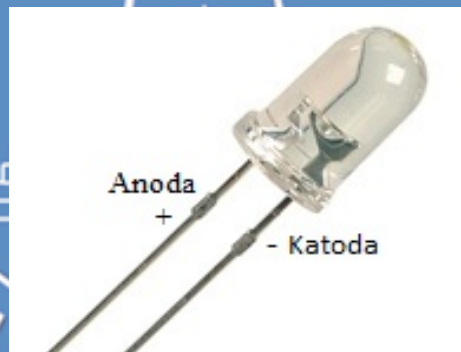
Pengertian dan cara kerja LED (*Light Emitting Diode*) adalah komponen elektronika yang dapat memancarkan cahaya *monokromatik* ketika diberikan tegangan maju. LED merupakan keluarga dioda yang terbuat dari bahan semikonduktor. Warna-warna cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada *remote control* TV ataupun *remote control* perangkat elektronik lainnya.

Bentuk LED mirip dengan sebuah lampu pijar (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika. Berbeda dengan lampu pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya. Oleh karena itu, saat ini LED (*Light Emitting Diode*) yang bentuknya kecil telah banyak digunakan sebagai lampu **penerang** dalam LCD TV yang mengganti lampu *tube*.

LED merupakan keluarga dari dioda yang terbuat dari semikonduktor. Cara kerjanya pun hampir sama dengan dioda yang memiliki dua kutub yaitu kutub Positif (P) dan kutub Negatif (N). LED hanya akan memancarkan cahaya apabila dialiri tegangan maju (*bias forward*) dari Anoda menuju ke Katoda. LED

terdiri dari sebuah *chip* semikonduktor yang didoping sehingga menciptakan *junction* P dan N. Yang dimaksud dengan proses doping dalam semikonduktor adalah proses untuk menambahkan ketidakmurnian (*impurity*) pada semikonduktor yang murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan. Ketika *LED* dialiri tegangan maju atau bias *forward* yaitu dari Anoda (P) menuju ke Katoda (K), kelebihan elektron pada N-Type material akan berpindah ke wilayah yang kelebihan *hole* (lubang) yaitu wilayah yang bermuatan positif (*P-Type material*). Saat elektron berjumpa dengan *hole* akan melepaskan *photon* dan memancarkan cahaya (satu warna).

LED (*Light Emitting Diode*) yang memancarkan cahaya ketika dialiri tegangan maju ini juga dapat digolongkan sebagai *transduser* yang dapat mengubah energi listrik menjadi energi cahaya. Ciri-ciri terminal anoda pada *LED* adalah kaki yang lebih panjang dan juga *lead frame* yang lebih kecil. Sedangkan ciri-ciri terminal katoda adalah kaki yang lebih pendek dengan *lead frame* yang besar serta terletak di sisi yang *flat*. Adapun bentuk fisik *LED* dapat dilihat pada gambar 2.10 dibawah ini :



Gambar 2.10: Polaritas *LED*

Untuk mengetahui polaritas terminal Anoda (+) dan Katoda (-) pada *LED*. Kita dapat melihatnya secara fisik berdasarkan gambar 2.10 di atas.

a. Warna-warna *LED* (*Light Emitting Diode*)

Saat ini, *LED* telah memiliki beranekaragam warna, diantaranya seperti warna merah, kuning, biru, putih, hijau, jingga dan infra merah. Keanekaragaman warna pada *LED* tersebut tergantung pada *wavelength* (panjang gelombang) dan

senyawa semikonduktor yang dipergunakannya. Berikut ini adalah Tabel 2.2 senyawa semikonduktor yang digunakan untuk menghasilkan variasi warna pada *LED* :

Tabel 2.2: Variasi warna *LED*

Bahan Semikonduktor	<i>Wavelength</i>	Warna
<i>Gallium Arsenide (GaAs)</i>	850-940 nm	Infra Merah
<i>Gallium Arsenide Phosphide (GaAsP)</i>	630-660 nm	Merah
<i>Gallium Arsenide Phosphide (GaAsP)</i>	605-620 nm	Jingga
<i>Gallium Arsenide Phosphide Nitride (GaAsP:N)</i>	585-595 nm	Kuning
<i>Aluminium Gallium Phosphide (AlGaP)</i>	550-570 nm	Hijau
<i>Silicon Carbide (SiC)</i>	430-505 nm	Biru
<i>Gallium Indium Nitride (GaN)</i>	450 nm	Putih

b. Tegangan Maju (*Forward Bias*) *LED*

Masing-masing warna *LED (Light Emitting Diode)* memerlukan tegangan maju (*Forward Bias*) untuk dapat menyalakannya. Tegangan maju untuk *LED* tersebut tergolong rendah sehingga memerlukan sebuah resistor untuk membatasi arus dan tegangannya agar tidak merusak *LED* yang bersangkutan. Tegangan maju biasanya dilambangkan dengan tanda V_F . Adapun penjelasan tegangan *LED* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2.3: Tegangan LED

Warna	Tegangan Maju @20 mA
Infra Merah	1,2 V
Merah	1,8 V
Jingga	2,0 V
Kuning	2,2 V
Hijau	3,5 V
Biru	3,6 V
Putih	4,0 V

2.6. Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*. Adapun bentuk *buzzer* dapat dilihat pada gambar 2.11 dibawah ini :



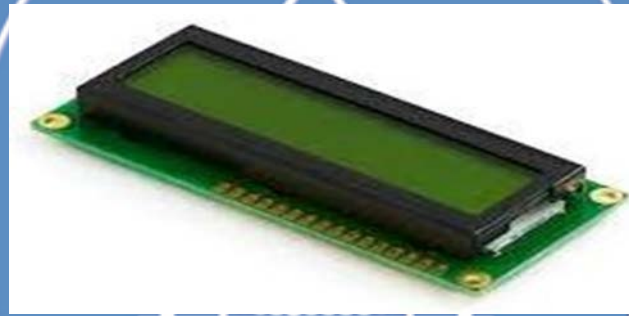
Gambar 2.11: Buzzer

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 ke luar, 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*).

2.7. LCD 2x16

LCD (Liquid Crystal Display) adalah suatu jenis media tampil yang menggunakan Kristal cair sebagai penampil utama. *LCD* sudah digunakan diberbagai bidang misalnya alat-alat elektronik seperti televisi, kalkulator, ataupun layar computer. Pada postingan aplikasi *LCD* yang digunakan ialah *LCD* dot matrik dengan jumlah karakter 2x16. Adapun bentuk *LCD* dapat dilihat pada gambar 2.12 berikut ini :



Gambar 2.12: LCD 2x16

LCD (Liquid Crystal Display) merupakan salah satu perangkat *display* yang umum dipakai dalam sebuah *system instrumentasi*. Dengan *LCD* kita bisa menampilkan sebuah informasi dari sebuah pengukuran data sensor, menu pengaturan *instrument*, ataupun yang lainnya dengan konsumsi daya rendah.

ATMega8535 juga didukung dengan penampil *LCD*, yang berfungsi untuk menampilkan nilai atau perintah-perintah yang ditulis pada kode program. Dengan *LCD* ini perintah-perintah yang diberikan akan mudah dibaca baik benar atau salah.

Spesifikasi *pin LCD* 2x16 dapat dilihat pada tabel 2.4 berikut ini :

Tabel 2.4 : Keterangan *Pin LCD*

Pin	Keterangan
1	Ground
2	Vcc
3	Pengatur kontras
4	“RS” (<i>Instruction/RegesterSelect</i>)
5	“R/W” (<i>Read/While</i>) <i>LCD Rigesters</i>
6	“EN” (<i>Enable</i>)
7	Data I/O <i>Pins</i>

2.8. *Bascom AVR*

Bahasa pemrograman basic terkenal di dunia sebagai bahasa pemrograman yang handal, bahasa ini sebenarnya bahasa yang memiliki kemampuan tingkat tinggi. Bahkan banyak para programmer terkenal dunia memakai bahasa pemrograman ini sebagai senjata ampuhnya. Bahasa pemrograman basic banyak digunakan untuk aplikasi mikrokontroler jenis AVR karena kompatibel dan didukung dengan *compiler* pemrograman berupa *software BASCOM AVR*. Bahasa basic memiliki penulisan program yang mudah dimengerti walaupun untuk orang awam sekalipun, karena itu bahasa ini dinamakan bahasa basic. Jenis perintah programnya seperti *do, loop, if, then*, dan sebagainya masih banyak lagi.

BASCOM AVR sendiri adalah salah satu tool untuk pengembangan / pembuatan program untuk kemudian ditanamkan dan dijalankan pada mikrokontroler terutama mikrokontroler keluarga AVR. *BASCOM AVR* juga bisa disebut sebagai *IDE (Integrated Development Environment)* yaitu lingkungan kerja yang terintegrasi, karena disamping tugas utamanya meng-compile kode program menjadi *file hex / bahasa mesin*, *BASCOM AVR* juga memiliki

kemampuan / fitur lain yang berguna sekali seperti *monitoring* komunikasi serial dan untuk menanamkan program yang sudah *dcompile* ke mikrokontroler.

BASCOM AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat dengan pergerakan *LED* yang ada pada layar simulasi dan dapat juga langsung dilihat pada *LCD*, jika kita membuat aplikasi yang berhubungan dengan *LCD*. Intruksi yang dapat digunakan pada editor *BASCOM AVR* relatif cukup banyak dan tergantung dari tipe dan jenis *AVR* yang digunakan.

a. Konstruksi bahasa *BASIC* pada *BASCOM AVR*

Setiap bahasa pemrograman mempunyai standar penulisan program. Konstruksi dari program bahasa *BASIC* harus mengikuti aturan sebagai berikut:

\$regfile = "header"
'inisialisasi
'deklarasi variabel
'deklarasi konstanta
Do
'pernyataan-pernyataan
Loop
End

Tabel 2.3 berikut ini adalah beberapa perintah intruksi-intruksi dasar yang digunakan pada *BASCOM AVR*.

Tabel 2.5: Intruksi Pada *BASKOM AVR*

Intruksi	Keterangan
<i>DO....LOOP</i>	Perulangan
<i>GOSUB</i>	Memanggil prosedur
<i>IF....THEN</i>	Percabangan
<i>FOR.....NEXT</i>	Perulangan
<i>WAIT</i>	Waktu tanda detik

<i>WAITMS</i>	Waktu tanda mili detik
<i>WAITUS</i>	Waktu tanda micro detik
<i>GOTO</i>	Loncat ke alamat memori
<i>SELECT....CASE</i>	Percabangan

1. Pengarah Preprosesor

\$regfile = "m8535.dat" merupakan pengarah preprosesor bahasa *BASIC* yang memerintahkan untuk meyisipkan file lain, dalam hal ini adalah file *m8535.dat* yang berisi deklarasi *register* dari mikrokontroler *ATmega8535*, pengarah preprosesor lainnya yang sering digunakan ialah sebagai berikut:

\$crystal = 12000000 ‘menggunakan *crystal clock* 12 *MHz*

\$baud = 9600 ‘komunikasi serial dengan *baudrate* 9600

\$eeprom ‘menggunakan fasilitas *eeprom*

2. Tipe Data

Tipe data merupakan bagian program yang paling penting karena sangat berpengaruh pada program. Pemilihan tipe data yang tepat maka operasi data menjadi lebih efisien dan efektif. Adapun tipe data *BASKOM AVR* ditunjukkan pada tabel 2.4 dibawah ini :

Tabel 2.6: Tipe Data *BASKOM AVR*

No.	Tipe	Jangkauan
1	<i>Bit</i>	0 atau 10 – 255 – 32,768 – 32,7670 – 65535 – 2147483647
2	<i>Byte</i>	
3	<i>Integer</i>	
4	<i>Word</i>	
5	<i>Long</i>	
6	<i>Single</i>	

7	<i>Double</i>	5.0 x 10 ^{- 324} to 1.7 x 10 ³⁰⁸
8	<i>String</i>	>254 by

3. Konstanta

Konstanta merupakan suatu nilai dengan tipe data tertentu yang tidak dapat diubah-ubah selama proses program berlangsung. Konstanta harus didefinisikan terlebih dahulu diawal program.

Contoh : $K_p = 35, K_i=15, K_d=40$

4. Variabel

Variabel adalah suatu pengenal (*identifier*) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program yang dapat diubah-ubah sesuai dengan kebutuhan. Nama dari variabel terserah sesuai dengan yang diinginkan namun hal yang terpenting adalah setiap variabel diharuskan :

Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf, *max* 32 karakter. Tidak boleh mengandung spasi atau simbol-simbol khusus seperti : \$, ?, %, #, !, &, *, (,), -, +, = dan lain sebagainya kecuali *underscore*.

5. Deklarasi

Deklarasi sangat diperlukan bila akan menggunakan pengenal (*identifier*) dalam suatu program.

6. Deklarasi Variabel

Bentuk umum pendeklarasian suatu *variable* adalah *dim* nama_variabel *AS* tipe_data.

Contoh : *Dim x As Integer* 'deklarasi x bertipe *integer*

7. Deklarasi Konstanta

Dalam bahasa basic konstanta dideklarasikan langsung.

Contohnya : $S = \text{"hello world"}$ 'assign string

8. Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat dipanggil di manapun didalam program. Fungsi dalam bahasa basic ada yang sudah disediakan sebagai fungsi pustaka seperti *print*, *input* data dan untuk menggunakannya tidak perlu dideklarasikan.

9. Deklarasi buatan

Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh *programmer*. Bentuk umum deklarasi sebuah fungsi adalah :

Sub Test ()

Contohnya : *Sub Pwm(byval Kiri As In teger , Byval Kanan As Integer)*

10. Operator

a. Operator Penugasan

Operator Penugasan (*Assignment operator*) dalam bahasa basic berupa “=”.

b. Operator Aritmatika

* : untuk perkalian

/ : untuk pembagian

+ : untuk penambahan

- : untuk pengurangan

% : untuk sisa pembagian (*modulus*)

c. Operator Hubungan (Perbandingan)

Operator hubungan digunakan untuk membandingkan hubungan dua buah operand atau sebuah nilai / variable, misalnya :

= 'Equality X = Y

< 'Less than X < Y

> 'Greater than X > Y

<= 'Less than or equal to X <= Y

>= 'Greater than or equal to X >= Y

d. Operator Logika

Operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan. Operator logika ada empat macam, yaitu :

NOT 'Logical complement

AND 'Conjunction

OR 'Disjunction

XOR 'Exclusive or

e. Operator Bitwise

Operator *bitwise* digunakan untuk memanipulasi bit dari data yang ada di memori. Operator *bitwise* dalam bahasa basic :

Shift A, Left, 2 : Pergeseran bit ke kiri

Shift A, Right, 2 : Pergeseran bit ke kanan

Rotate A, Left, 2 : Putar bit ke kiri

Rotate A, right, 2 : Putar bit ke kanan

1. Pernyataan Kondisional (*IF-THEN – END IF*)

Pernyataan ini digunakan untuk melakukan pengambilan keputusan terhadap dua buah bahkan lebih kemungkinan untuk melakukan suatu blok pernyataan atau tidak. Konstruksi penulisan pernyataan *IF-THEN-ELSE-END IF* pada bahasa *BASIC* ialah sebagai berikut:

IF pernyataan kondisi 1 *THEN*

‘blok pernyataan 1 yang dikerjakan bila kondisi 1 terpenuhi

IF pernyataan kondisi 2 *THEN*

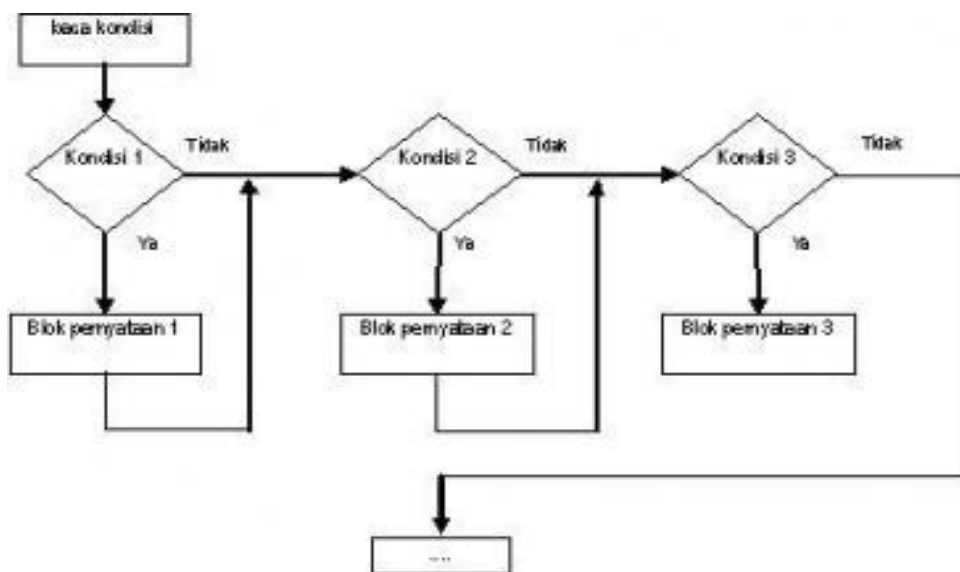
‘blok pernyataan 2 yang dikerjakan bila kondisi 2 terpenuhi

IF pernyataan kondisi 3 *THEN*

‘blok pernyataan 3 yang dikerjakan bila kondisi 3 terpenuhi

Setiap penggunaan pernyataan *IF-THEN* harus diakhiri dengan perintah *END IF* sebagai akhir dari pernyataan kondisional.

Adapun diagram alir pernyataan kondisional (*IF – THEN - END IF*) dapat dilihat pada gambar 2.13 dibawah ini :



Gambar 2.13: Diagram Alir Pernyataan Kondisional (*IF-THEN – END IF*)

2. Pernyataan Kondisional (*SELECT-CASE-END SELECT*)

Pernyataan ini digunakan untuk melakukan pengambilan keputusan terhadap banyak kondisi. Konstruksi penulisan pernyataan *SELECT-CASE-END SELECT* pada bahasa *BASIC* ialah sebagai berikut:

SELECT CASE var

CASE 'kondisi1 : 'blok perintah1

CASE 'kondisi2 : 'blok perintah2

CASE 'kondisi3 : 'blok perintah3

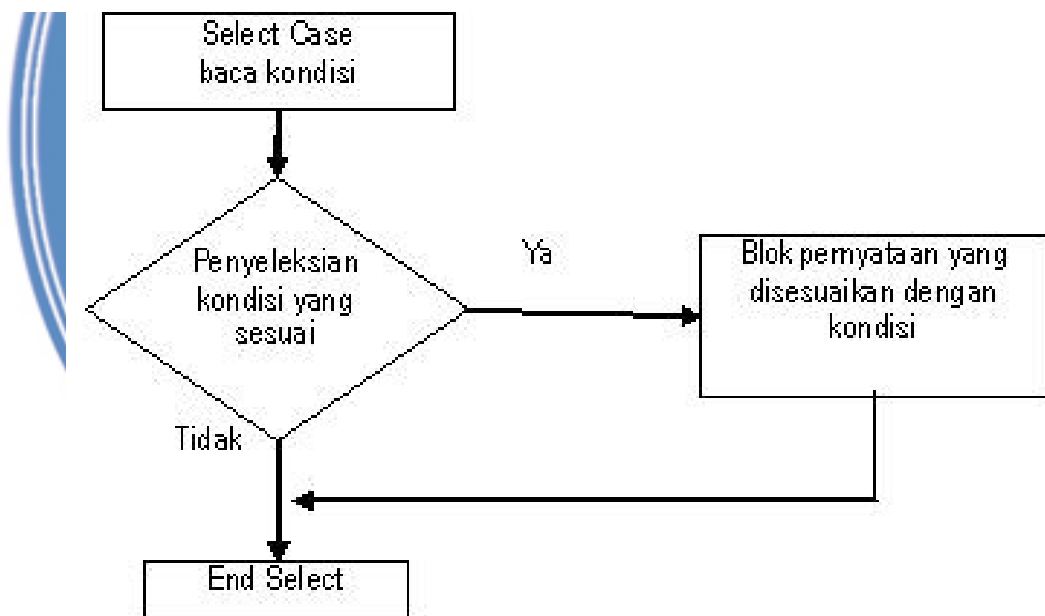
CASE 'kondisi4 : 'blok perintah4

CASE 'kondisi5 : 'blok perintah5

CASE 'kondisi'n' : 'blok perintah'n'

END SELECT 'akhir dari pernyataan SELECT CASE

Adapun diagram alirnya dapat dilihat pada gambar 2.14 berikut ini:



Gambar 2.14: Diagram Alir Pernyataan Kondisional (*SELECT-CASE-END SELECT*)