BAB II
DASAR TEORI

Dalam perancangan sistem pengontrolan lampu ruangan menggunakan sensor infra merah diperlukan dasar teori dan komponen-komponen elektronik pendukung agar alat yang dimaksud dapat tercapai. Adapun dasar teori dan komponen-komponen elektronik yang digunakan yaitu :

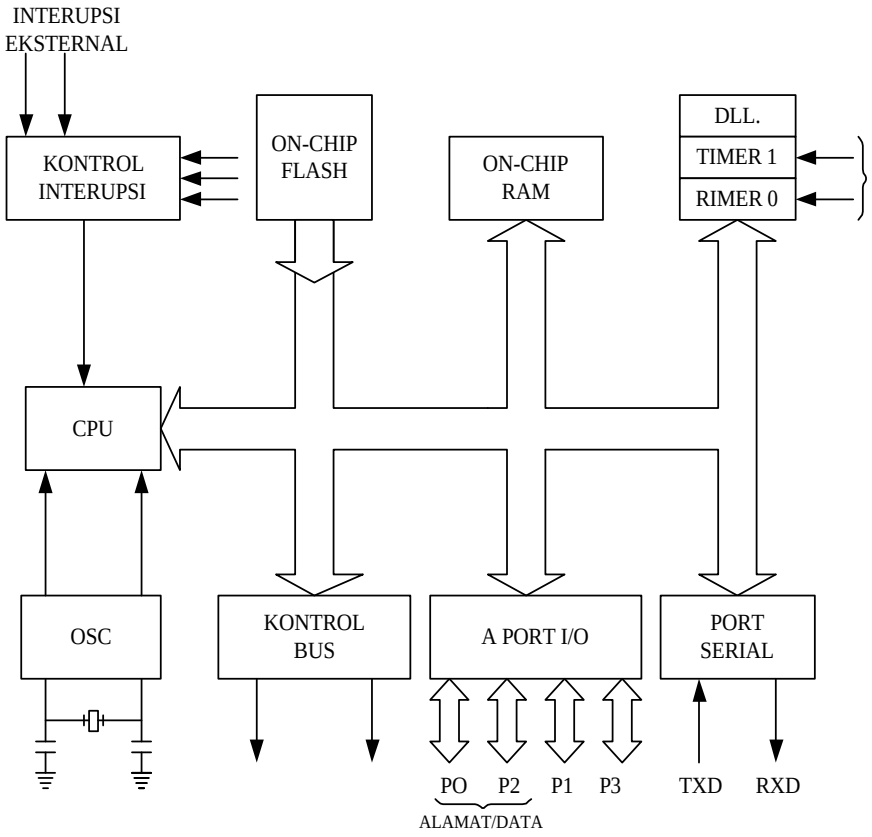
2. 1 Mikrokontroler AT89C51

Mikrokontroler adalah sebuah mikroprosesor yang telah dikombinasikan dengan *Input/Output* (I/O) dan memori sebagai komponen utamanya dalam bentuk s*ingle chip.* Lebih jauh lagi mikroprosesor yang telah dikombinasikan dengan *Input/Output* (I/O) dan memori baik RAM maupun ROM dengan kompleksitas tinggi dan mempunyai pemproses pusat (*Central Processing Unit*) akan disebut sebagai mikrokomputer. Secara umum sistem mikrokomputer terdiri dari :

a. Unit mikroprosesor (*microprocessing unit*)
b. RAM (*Random Access Memory*)
c. ROM (*Read Only Memory*)
d. I/O (*Input/Output unit*)

Mikrokontroler secara mandiri dapat melakukan pemrosesan terhadap data yang diterimanya, tetapi ada beberapa mikrokontroler yang memerlukan ROM tambahan (*external ROM*) agar dapat beroperasi. Mikrokontroler AT89C51 adalah mikrokontroler buatan ATMEL *Corporation,* termasuk ke dalam bagian keluarga mikrokontroler MCS-51, varian mikrokontroler 8051 dan memiliki ciri-ciri sebagai berikut :

a. CPU (*Central Processing Unit*) 8 bit untuk aplikasi pengontrolan
b. 4 kilobyte internal ROM
c. 128 byte internal RAM
d. 4 buah *port* I/O 8 bit
e. 2 buah timer 16 bit
f. 5 jalur interupsi, terdiri dari 3 interupsi internal dan 2 interupsi eksternal
g. Saluran komunikasi serial
h. 32 baris *Programmable* I/O
i. Mampu beroperasi statis secara penuh pada 0 Hz to 24 MHz



Gambar 2.1 *Blok Diagram Dasar AT89C51*

2.1.1 Deskripsi Fungsi Pin Mikrokontroler AT89C51

1. *Port* 0

   Port 0 yang berada pada pin 32 hingga pin 39 merupakan saluran 8 bit *bidirectional* yang dapat dijadikan saluran input maupun saluran *output*. Bila digunakan untuk mengakses eksternal memori, *port* 0 akan memultipleks alamat memori dan data.

2. *Port* 1

   Port 1 yang berada pada pin 1 hingga pin 8 merupakan saluran 8 bit *bidirectional* yang dapat dijadikan saluran *input* maupun saluran *output*.

3. *Port* 2

   Port 2 berada pada pin 21 hingga pin 28 merupakan saluran 8 bit *bidirectional* yang dapat berfungsi saluran *input* maupun saluran *output*. Bila digunakan untuk mengakses eksternal memori, *port* 2 akan mengirimkan byte atau alamat ke eksternal memori.

4. *Port* 3

   Port 3 yang berada pada pin 10 hingga pin 17 merupakan saluran 8 bit *bidirectional* yang dapat dijadikan saluran *input* maupun saluran *output*. Selain itu *port* 3 juga berfungsi sebagai saluran komunikasi serial dan sebagai jalur interupsi.

   Berikut ini merupakan fungsi khusus dari *port* 3 :

   Tabel 2.1 *Fungsi Khusus Port 3*

PIN *PORT*	**FUNGSI**
P3.0	RXD (*Port Serial Input*)
P3.1	TXD (*Port Serial Output*)
P3.2	INT0 (*Input Interupsi External* 0)
P3.3	INT1 (*Input Interupsi External* 1)
P3.4	T0 (*Input Interupsi Timer* 0)
P3.5	T1 (*Input Interupsi Timer* 1)
P3.6	WR (Sinyal Eksternal Tulis Memory Data)
P3.7	RD (Sinyal Eksternal Baca Memory Data)

   Fungsi-fungsi alternatif pada tabel 2.1 hanya dapat diaktifkan jika bit-bit pengancing (*latch*) *port* yang bersangkutan berisi ‘1’.

5. RST (*Reset*)

   Pin *reset* berfungsi untuk memaksa sistem mikrokontroler bekerja dari awal (alamat 00h) dengan cara memasukkan *input high* pada pin *reset*.

6. ALE / $\overline{\text{PROG}}$

   ALE (*Address Latch Enable*) digunakan untuk pengaturan *address bus* dan *data bus* pada pengoperasian dengan memori eksternal, sinyal ALE akan menahan alamat memori eksternal selama pelaksanaan instruksi. Pin ini juga berfungsi sebagai pulsa program (PROG) pada proses pemrograman *flash*.

7. $\overline{\text{PSEN}}$

   Program Store Enable merupakan pin untuk membaca eksternal program memori pada mikrokontroler yang dioperasikan dengan program memori eksternal.

8. $\overline{EA}$ / VPP

   Eksternal Access Enable merupakan pin yang dihubungkan dengan Vcc bila menggunakan memori program internal yang beralamat 0000H hingga 0FFFH. Apabila menggunakan memori eksternal maka alamat dari memori eksternal ialah mulai 1000H hingga FFFFH. EA harus dihubungkan dengan *ground* agar dapat mengakses memori program eksternal yang beralamat mulai 0000H hingga FFFFH.

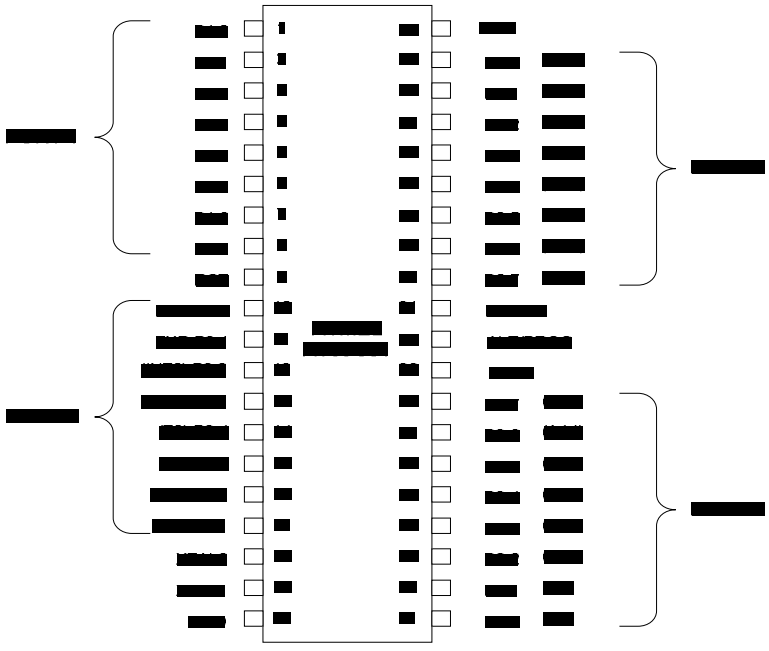
9. VCC

   Pin Vcc merupakan *input* catu daya untuk mikrokontroler dan diberi tegangan sebesar 5 volt.

10. VSS

    Pin Vss adalah pin *ground* tegangan catu daya untuk mikrokontroler dan diberi *input* tegangan 0 volt.

11. XTAL1 dan XTAL2 XTAL 1 merupakan *input* oscillator internal. XTAL 2 merupakan *output* oscilator eksternal.



Gambar 2.2 *Konfigurasi Pin Mikrokontroler AT89C51*

2.1.2 *Port* Input/Output AT89C51

Mikrokontroler AT89C51 mempunyai empat buah *port*, yaitu *Port* 0, *Port* 1, *Port* 2 dan *Port* 3 yang terletak pada alamat 80H, 90H, A0H dan B0H. Namun, jika digunakan memori eksternal atau pun fungsi-fungsi khusus, seperti interupsi eksternal,

serial atau pun timer eksternal, *Port* 0, *Port* 2, dan *Port* 3 tidak dapat digunakan sebagai *port* dengan fungsi umum. Untuk itu disediakan *port* 1 yang dikhususkan untuk *port* dengan fungsi umum.

Port 0, *Port* 2, dan *Port* 3 memiliki fungsi alternatif. Masing-masing pin dari *port-port* ini dapat digunakan sebagai jalur *input*/*output* digital secara umum atau alternatifnya dapat digunakan untuk fungsi keduanya. Fungsi kedua dari *Port* 0 dan 2 adalah untuk menghubungkan memori eksternal. Ketika program eksternal atau memori data sedang diakses, *Port* 2 mengeluarkan byte tinggi dari alamat 16-bit. *Port* 0 awalnya mengeluarkan byte rendah dari alamat 16-bit, kemudian mengirim atau menerima byte data. Semua *port* ini dapat diakses dengan pengalamatan secara bit sehingga dapat dilakukan perubahan *output* pada tiap-tiap pin dari *port* ini tanpa mempengaruhi pin-pin yang lainnya.

Ketika sebuah *port* digunakan sebagai *port output*, data diletakkan pada *SFR (Special Function Register)* yang sesuai. Lebih spesifik nilai yang tertulis di SFR akan diteruskan sampai dengan operasi penulisan selesai. Nilai dari *port output* akan berubah bila nilai baru dimasukkan.

Ketika sebuah *port* digunakan sebagai *port input*, nilai FFH harus dituliskan pertama kali ke *port*, kemudian setiap *input* yang menggunakan tegangan rendah akan dianggap sebagai nilai 0, dan *port* tersebut dapat dibaca dari SFR yang sesuai. Lebih spesifik pembacaan SFR yang sesuai, membaca nilai pin *port*. *Output latch* menggerakkan pin *port* ke level logika 1 jika tidak ada penurunan arus rangkaian eksternal pada pin tersebut.

Harus diperhatikan, meskipun SFR yang sama sedang digunakan, terdapat dua perbedaan operasi yang keluar, kapan sebuah *port* ditulis dan kapan *port* dibaca. Operasi-operasi dilaksanakan seluruhnya di perangkat keras dan pengguna dapat menjaga arah jalur di mana data tersebut dipindahkan.

Fungsi alternatif dari pin-pin *Port* 3 termasuk *interrupt* dan *input timer*, serial *port input* dan *output*, dan sinyal kontrol untuk menghubungkan dengan memori eksternal.

2.1.3 Struktur Memori

AT89C51 mempunyai struktur memori yang terdiri dari :

1. RAM Internal

   RAM internal memiliki memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara, dialamati oleh *RAM Address Register* (Register Alamat RAM). RAM internal terdiri atas :

   a. *Register Banks*

      89C51 memiliki delapan buah register yang terdiri dari R0 sampai R7 yang tereletak pada alamat 00H hingga 07H pada setiap kali *reset*.

   b. *Bit Addressable* RAM

      RAM dengan alamat 20H hingga 2FH dapat diakses secara pengalamatan bit *(bit addressable)* sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat di-set, *clear*, AND dan OR.

   c. RAM dengan fungsi umum

      RAM dengan fungsi umum dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu *operand* merupakan bilangan yang dialamati. Sedangkan pengalamatan tak langsung pada lokasi dari RAM Internal ini adalah akses data dari memori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1 yang dapat digunakan sebagai *pointer* dari lokasi memori pada RAM Internal.

2. Register Fungsi Khusus (*Special Function Register*)

   Memori yang berisi register-register yang memiliki fungsi khusus yang tersediakan oleh mikrokontroler, seperti timer, serial dan lain-lain. 89C51 memiliki 21 *Special Function Register* yang terletak pada alamat 80H hingga FFH dengan rincian pada tabel 2.2. Salah satu contoh dari *Special Function Register* adalah *Accumulator*, register ini terletak pada alamat E0H. Semua operasi aritmatika dan operasi logika dan proses pengambilan dan pengiriman data ke memori selalu menggunakan register ini.

Tabel 2.2 *Alamat register fungsi khusus*

Register	**Mnemonic**	**Alamat**
P0	*Port* 0 Latch	80H
SP	Stack Pointer	81H
DPTR	Data Pointer	82H-83H
DPL	Data Pointer Low Byte	82H
DPH	Data Pointer High Byte	83H
PCON	Power Control	87H
TCON	Timer/Counter Control	88H
TMOD	Timer/Counter Mode Control	89H
TL0	Timer/Counter 0 Low Byte	8AH
TL1	Timer/Counter 1 Low Byte	8BH
TH0	Timer/Counter 0 High Byte	8CH
TH1	Timer/Counter 1 High Byte	8DH
P1	*Port* 1 Latch	90H
SCON	Serial *Port* Control	98H
SBUF	Serial Data *Port*	99H
P2	*Port* 2 Latch	A0H
IE	Interrupt Enable	A8H
P3	*Port* 3 Latch	B0H
IP	Interrupt Priority Control	B8H
PSW	Program Status Word	D0H
ACC	Accumulator	E0H
B	Register B	F0H

3. *Flash* PEROM

Memori yang digunakan untuk menyimpan instruksi-instruksi MCS-51 dialamati oleh Register Alamat Program (*Program Address Register*). AT89C51 memiliki 4 Kb *Flash* PEROM yang menggunakan *Atmel's High-Density Non Volatile Technology*.

Program yang ada pada *Flash* PEROM akan dijalankan jika pada saat sistem di-*reset*, pin EA/VP berlogika satu maka mikrokontroler aktif berdasarkan program yang ada pada *Flash* PEROM-nya. Namun jika pin

EA/VP berlogika nol, mikrokontroler aktif berdasarkan program yang ada pada memori eksternal.

2.1.4 Timer AT89C51

Mikrokontroler AT89C51 mempunyai dua buah timer, yaitu Timer 0 dan Timer 1, setiap timer terdiri dari 16 bit timer yang tersimpan dalam dua buah register yaitu THx untuk *Timer High Byte* dan TLx untuk *Timer Low Byte* yang keduanya dapat berfungsi sebagai *counter* maupun sebagai timer. Secara fisik timer juga merupakan rangkaian T Flip-flop yang dapat diaktifkan dan dinonaktifkan setiap saat. Perbedaan keduanya terletak pada sumber *clock* dan aplikasinya.

Timer mempunyai sumber *clock* dengan frekuensi tertentu yang sudah pasti sedangkan *counter* mendapat sumber *clock* dari pulsa yang hendak dihitung jumlahnya. Aplikasi dari timer atau pewaktu biasa digunakan untuk aplikasi menghitung lamanya suatu kejadian yang terjadi sedangkan *counter* atau penghitung biasa digunakan untuk aplikasi menghitung jumlah kejadian yang terjadi dalam periode tertentu.

Perilaku dari register THx dan TLx diatur oleh register TMOD dan register TCON. Timer dapat diaktifkan melalui perangkat keras maupun perangkat lunak.

Perioda waktu *timer*/*counter* dapat dihitung menggunakan rumus 2.1 dan 2.2 sebagai berikut :

1. Sebagai *timer/counter* 8 bit

$$T = (255 - TLx) * \frac{12}{frekuensi.XTAL}\ \mu s \quad \text{..... (2.1)}$$

2. Sebagai *timer/counter* 16 bit

$$T = (65535 - THx\ TLx) * \frac{12}{frekuensi.XTAL}\ \mu s \quad \text{..... (2.2)}$$

di mana : THx = isi register TH0 atau TH1

TLx = isi register TL0 atau TL1

Dapat diakses secara bit

	Register Timer				Register Interupsi			
88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
	TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0

Tidak dapat diakses secara bit

	Timer 1				Timer 0			
89H	Gate	C/T	M1	M0	Gate	C/T	M1	M0

Gambar 2.3 *Register TCON dan TMOD*

Pengontrolan kerja *timer/counter* diatur oleh register TCON. Register ini bersifat *bit addressable* sehingga bit TF1 dapat disebut TCON.7 dan seterusnya hingga bit IT0 sebagai TCON.0. Register ini hanya mempunyai 4 bit saja yang berhubungan dengan timer seperti diperlihatkan gambar 2.3 dan dijelaskan pada tabel 2.3.

Tabel 2.3 *Fungsi bit register TCON yang berhubungan dengan timer*

Nama Bit	Fungsi
TF1	*Timer* 1 *Overflow Flag* yang akan diset jika *timer overflow*.
TR1	Membuat timer 1 aktif (set) dan nonaktif (*clear*)
TF0	Timer 0 *Overflow Flag* yang akan diset jika *timer overflow*.
TR0	Membuat timer 0 aktif (set) dan nonaktif (clear)

Register TMOD berfungsi untuk pemilihan mode operasi *timer*/*counter* dengan fungsi setiap bitnya adalah sebagai berikut:

Gate : Pada saat TRx = 1, *timer* akan berjalan tanpa memperhatikan nilai pada *Gate* (timer dikontrol software).

C/T : Pemilihan fungsi timer (0) atau counter (1).

M1 & M0 : Untuk memilih mode timer dengan variasi seperti pada tabel 2.4.

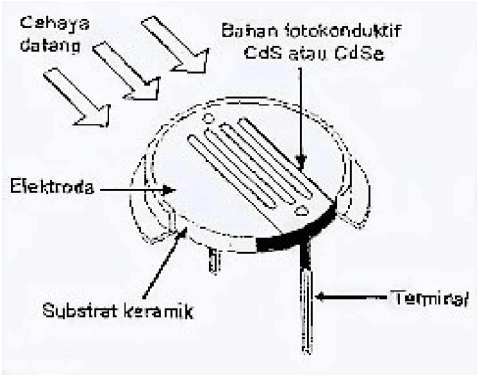
Tabel 2.4 *Mode timer*

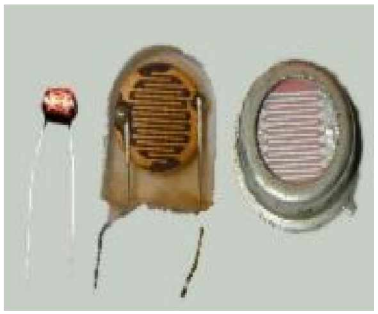
M1	M0	Mode	Operasi
0	0	0	Timer 13 bit
0	1	1	Timer/counter 16 bit
1	0	2	Timer 8 bit di mana nilai timer tersimpan pada TLx. Register THx berisi nilai isi ulang yang akan dikirim ke TLx setiap *overflow*.
1	1	3	Pada mode ini, AT89C51 bagaikan memiliki 3 buah timer. Timer 0 terpisah menjadi 2 buah timer 8 bit (TL0 – TF0 dan TH0 – TF1) dan timer 1 tetap 16 bit.

2.2 *Light Dependent Resistor* (LDR)

Light Dependent Resistor (LDR) adalah sebuah komponen elektronika yang mempunyai sifat seperti tahanan variabel. LDR dibuat dari bahan semikonduktor (yaitu bahan yang bukan konduktor dan bukan pula isolator), yang karakteristik listriknya berubah-ubah sesuai dengan intensitas cahaya yang mengenai permukaannya. Dua jenis bahan semikonduktor yang digunakan dalam pembuatan LDR adalah C*admium Sulfida* (CdS) dan *Cadmium Selenida* (CdSe).

Kedua bahan ini mempunyai respon yang lamban terhadap perubahan intensitas cahaya. Untuk CdSe responnya sekitar 10 ms, dan Cds sekitar 100 ms. Perbedaan yang utama pada kedua bahan tersebut adalah kepekaannya terhadap panas yang berbeda, pada CdS lebih stabil untuk suhu ruang dari pada CdSe. Bahan-bahan ini juga paling sensitif terhadap cahaya dalam spektrum tampak, dengan panjang gelombang berkisar antara 400 nm sampai 750 nm, untuk CdS memuncak sekitar 600 nm dan untuk CdSe memuncak sekitar 720 nm. Hal ini memperlihatkan mengapa CdS lebih popular untuk rangkaian yang dikemudikan cahaya daripada CdSe, namun CdSe juga cukup peka pada beberapa daerah cahaya tampak. Gambar 2.4(a) memperlihatkan penampang dalam LDR dan beberapa bentuk LDR diperlihatkan pada gambar 2.4(b).



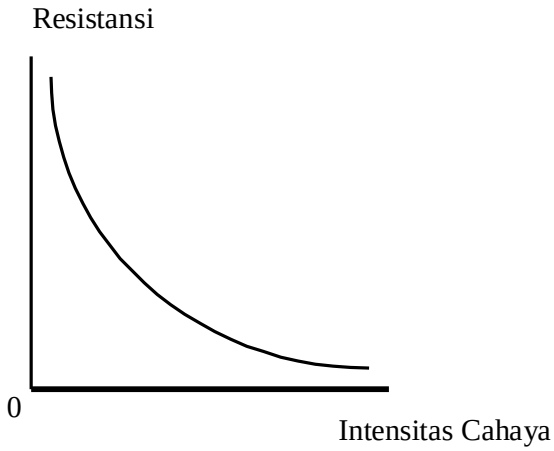


(a) (b)

Gambar 2.4 *(a) Penampang dalam LDR, (b) Bebarapa bentuk LDR*

Prinsip kerja dari LDR adalah sebagai berikut, apabila seberkas cahaya mengenai permukaannya ternyata nilai tahanannya menjadi berkurang dan apabila ditempatkan dalam ruangan yang lebih gelap nilai tahanannya menjadi lebih besar. Jadi nilai tahanannya berbanding terbalik terhadap intensitas cahaya yang mengenainya. Umumnya pada keadaan gelap LDR mempunyai tahanan beberapa mega Ohm,

sedangkan pada keadaan terang akan mempunyai tahanan beberapa ratus Ohm. Grafik karakteristik LDR diperlihatkan pada gambar 2.5.



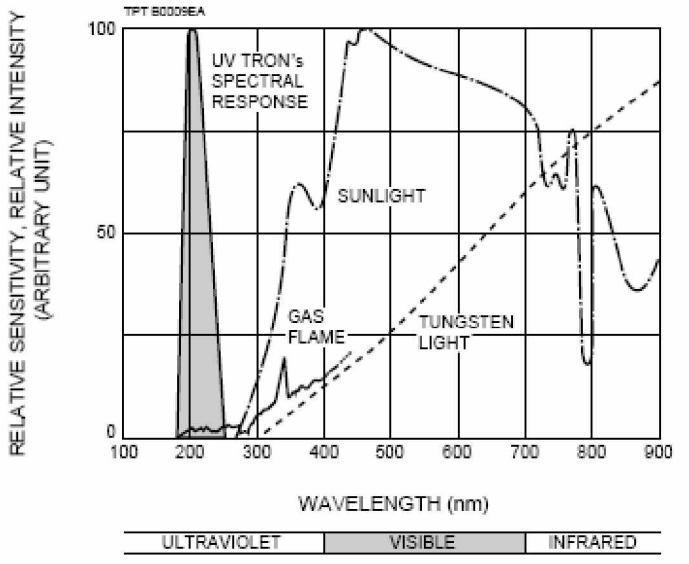
Gambar 2.5 *Kurva karakteristik LDR terhadap cahaya tampak*

Dengan karakteristik tersebut maka LDR dapat dimanfaatkan sebagai alat sensor cahaya dengan mengasumsikan bahwa objek yang di deteksi mempunyai intensitas cahaya yang cukup kuat untuk mempengaruhi nilai resistansi dari komponen ini. Komponen ini akan mengeluarkan tegangan keluaran yang berubah-ubah sejalan dengan intensitas cahaya yang di terima, di mana sinyal ini dapat dijadikan sebagai sinyal masukan untuk keadaan cahaya dalam ruangan.

2.3 Infra Merah

Cahaya infra merah merupakan cahaya yang tidak tampak. Jika di lihat dengan spektroskop cahaya maka radiasi cahaya infra merah akan nampak pada spektrum elektromagnet dengan panjang gelombang di atas panjang gelombang cahaya merah.

Infra merah merupakan suatu bentuk energi elektromagnetik dengan panjang gelombang sekitar 750 nm sampai 1 mm, dengan panjang gelombang ini maka cahaya infra merah tidak tampak oleh mata dan tidak dapat menembus benda padat seperti halnya cahaya tampak yang diperlihatkan oleh gambar 2.6.



Gambar 2.6 *Beberapa spektrum gelombang cahaya*

2.3.1 Pemancar Infra Merah (*Transmitter*)

Pemancar infra merah adalah suatu rangkaian yang dibuat agar dapat mengirim suatu informasi melalui media cahaya. Agar informasi dapat di terima oleh *receiver*, maka sinyal informasi tersebut harus diubah ke dalam besaran listrik dengan menggunakan piranti-piranti tertentu.

Infra merah dapat digunakan baik untuk memancarkan data maupun sinyal suara. Keduanya membutuhkan sinyal *carrier* untuk membawa sinyal data maupun sinyal suara tersebut hingga sampai pada *receiver*.

Untuk transmisi sinyal biasanya digunakan rangkaian *voltage to frequency converter* yang berfungsi untuk merubah tegangan sinyal menjadi frekuensi. Untuk transmisi data biasanya sinyal ditransmisikan dalam bentuk pulsa-pulsa. Informasi tersebut di kirim dalam bentuk frekuensi. Agar informasi sampai ke modul penerima, diperlukan teknik pengiriman frekuensi. Adapun teknik pengiriman frekuensi yaitu :

a. Teknik tanpa Modulasi

   Teknik ini hanya memanfaatkan intensitas cahaya pantulan yang di terima. Kelemahan dari teknik ini adalah rentan terhadap gangguan cahaya luar yang intensitasnya lebih kuat.

b. Teknik dengan Modulasi

   Teknik ini memodulasi intensitas cahaya dengan frekuensi tertentu. Ilustrasi dari teknik ini adalah pancaran cahaya yang termodulasi seperti

cahaya yang berkedip. Cahaya ini masih dapat terlihat walaupun diganggu oleh sinar lain.

Teknik Modulasi

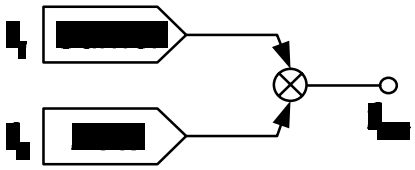
Ada dua cara dalam memodulasi yaitu :

1. Modulasi dengan frekuensi tunggal

   Teknik ini memodulasi cahaya dengan satu frekuensi saja, misalnya frekuensi 1kHz. Untuk memodulasi, teknik ini memanfaatkan *bandpass filter*.

2. Modulasi dengan dua frekuensi

   Teknik ini memodulasi cahaya dengan dua frekuensi, misalnya frekuensi pertama pada 38 kHz (sebagai *carrier*), sedangkan frekuensi yang kedua adalah frekuensi informasi (data) yang biasanya lebih rendah dari yang pertama misalnya berkisar 500 Hz. Teknik ini sangat efektif agar informasi yang diterima tidak salah.



Gambar 2.7 *Diagram blok Modulasi*

Sumber frekuensi osilasi

Terdapat beberapa cara untuk mendapatkan frekuensi osilasi yang akan memodulasi cahaya infra merah, diantaranya:

1. Kristal

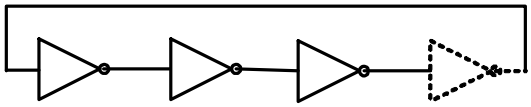
Komponen ini biasanya digunakan untuk membangkitkan *clock* mikrokontroler. Keuntungan dari penggunaan komponen ini adalah penentuan frekuensi osilator dengan tepat.

2. *Schmitt-trigger*

Schmitt-trigger adalah aplikasi komparator yang men-*switch* keluaran negatif ketika *input* melewati tegangan referensi positif. Kemudian

menggunakan balikan negatif untuk mencegah *switching* kembali ke bagian lain sampai *input* melewati tegangan *threshold* yang lebih rendah, kemudian penstabilan *switching* terhadap *trigger* yang cepat oleh *noise.*

Penggabungan beberapa gerbang *inverter* akan menghasilkan frekuensi osilasi dengan menggunakan delay propagasi, seperti pada gambar 2.8 dan berikut adalah rumusan untuk mencari nilai frekuensi osilasi :



Gambar 2.8 *Osilasi dengan beberapa gerbang inverter*

$$f = \frac{1}{2nTp} \quad \text{.....(2.3)}$$

Keterangan:

F = Frekuensi osilasi

Tp = Delay propagasi tiap gerbang

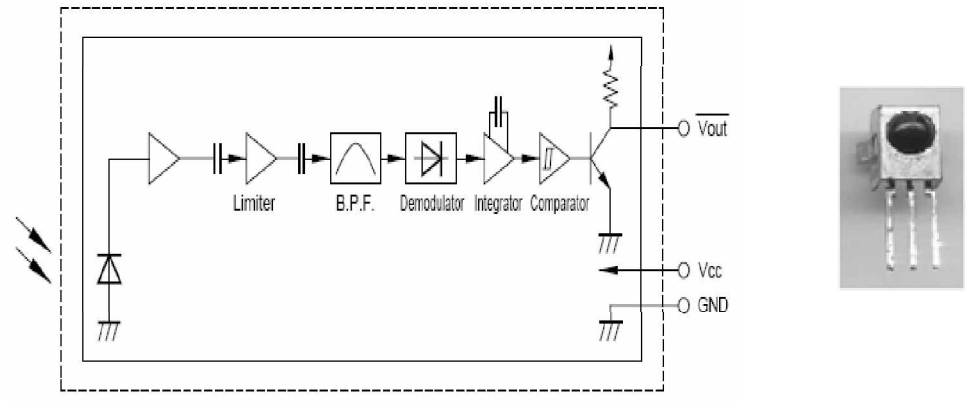
N = Jumlah gerbang yang digunakan

2.3.2 Penerima Infra Merah (*Receiver*)

Komponen yang peka terhadap cahaya seperti photodioda atau phototransistor merupakan komponen yang dapat menerima cahaya infra merah yang dipancarkan oleh *transmitter*. Komponen ini akan mengubah energi cahaya infra merah menjadi energi listrik sebanyak mungkin, sehingga pulsa sinyal listrik yang dihasilkannya mempunyai kualitas yang baik.

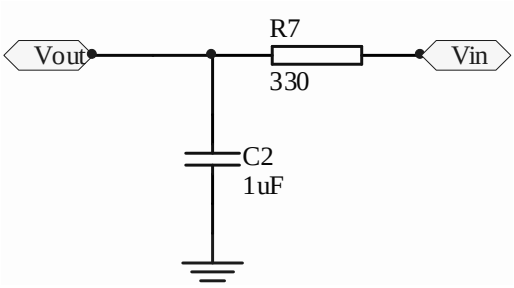
Pada *receiver* infra merah, sinyal ini merupakan sinyal infra merah yang termodulasi. Pemodulasian sinyal data dan sinyal *carrier* pada frekuensi tertentu dapat membuat jarak transmisi data sinyal infra merah menjadi jauh.

Sebuah *receiver* infra merah dilengkapi dengan lensa cembung yang mempunyai sifat mengumpulkan dan mem-*filter* cahaya, atau lebih dikenal sebagai *optical filter*, yang hanya melewatkan cahaya infra merah saja. Walaupun demikian cahaya yang tampakpun masih dapat mengganggu kerja dari *receiver* infra merah, karena tidak semua cahaya tampak bisa di-*filter* dengan baik. Oleh karena itu harus di-*filter* pada frekuensi sinyal *carrier* yaitu pada 30 kHz sampai 40 kHz.



Gambar 2.9 *Modul Penerima Infra merah*

Karena pada modul pemancar infra merah menggunakan teknik modulasi dua frekuensi, maka pada modul penerima infra merah harus men-demodulasi informasi yang dikirimkan pemancar. Teknik yang digunakan adalah *low-pass filter* yaitu sebuah *filter* yang melewatkan frekuensi rendah dan menahan frekuensi yang lebih tinggi dari frekuensi *cutoff*.



Gambar 2.10 *Contoh Low-pass filter*

Frekuensi *cutoff* (*fc*) dapat dicari menggunakan rumus:

$$fc = \frac{1}{2\pi RC} \quad \text{..... (2.4)}$$

2.4 *Buzzer*

Buzzer adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara. Pada umumnya *buzzer* digunakan untuk alarm, karena penggunaannya yang cukup mudah yaitu dengan memberikan tegangan *input* maka *buzzer* akan mengeluarkan bunyi. Frekuensi suara yang dikeluarkan oleh *buzzer* yaitu antara 1 sampai 5 kHz. Contoh buzzer diperlihatkan pada gambar 2.11.



Gambar 2.11 *Contoh Buzzer*

2.5 Relay

Relay adalah alat yang dioperasikan dengan listrik dan secara mekanis mengontrol penghubungan rangkaian listrik, relay dioperasikan sebagai saklar (*switch*) listrik yang bermanfaat untuk kontrol jarak jauh. Relay akan bekerja jika ada masukan sinyal listrik berupa arus dan tegangan.

Pada relay terdapat dua bagian utama, yaitu koil dan kontak. Koil terdiri dari kumparan yang merupakan lilitan kawat tembaga, di mana kumparan tersebut akan di aliri arus listrik agar dapat menghasilkan medan magnet pada inti besi. Inti besi dan koil juga memiliki jangkar yang terbuat dari besi lunak yang digunakan untuk mengaktifkan kontak relay setelah tertarik pada inti besi. Relay dan bagian dalam relay diperlihatkan pada gambar 2.12.

Sedangkan kontak yang merupakan saklar terdapat dua macam kondisi dari kontak tersebut, yaitu :

a. *Normally Open* (NO), yaitu kontak (*switch*) akan terbuka pada saat koil diberi suplai tegangan.
b. *Normally Closed* (NC), yaitu kontak (*switch*) akan terbuka pada saat koil tidak disuplai tegangan.

Pada komponen relay yaitu bagian koilnya disuplai tegangan, yang mana besar tegangan yang akan di suplai harus sesuai dengan tegangan yang dibatasi oleh koil relay. Maka arus akan mengalir pada kumparan, sehingga pada inti besi yang dililiti oleh kumparan akan timbul atau menghasilkan medan magnet, setelah inti besi bersifat magnetis maka jangkar akan tertarik ke inti besi sehingga akan mengaktifkan kontak relay.

Jangkar dapat ditarik dari inti besi jika gaya magnet pada inti besi dapat mengalahkan gaya pegas pada jangkar yang melawannya, besarnya gaya magnet ditetapkan oleh kuat medan magnet yang ada di dalam udara diantara jangkar dan inti

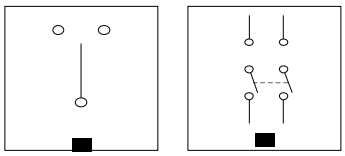
besi, adapun gaya magnet ini bergantung pada banyaknya lilitan kumparan dari kuat arus yang ada pada kumparan.



(a) (b)

Gambar 2.12 *(a) Contoh Relay, (b) Bagian dalam Relay*

Ada dua macan relay yaitu SPDT (*Single Pole Double Though*) relay, dengan cara kerjanya yaitu ketika *switch* yang satu membuka maka yang lain menutup, sedangkan DPDT (*Double Pole Double Though*) relay mempunyai dua pelat yang akan bergerak secara bersamaan ketika ada arus masuk pada relay tersebut. Contoh kedua relay tersebut seperti pada gambar 2.13.



Gambar 2.13 *(a) SPDT relay, (b) DPDT relay*

2.6 Penyearahan (*Rectifier*)

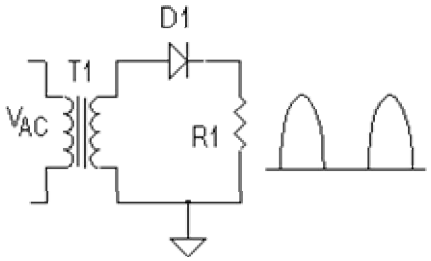
Penyearahan adalah proses menyearahkan arus bolak-balik menjadi arus searah. Arus bolak balik-balik ini berasal dari tegangan jala-jala. Komponen utama yang diperlukan dalam penyearahan adalah transformator, dioda dan kapasitor elektrolit.

Secara umum penyearah dibagi menjadi tiga kategori yaitu :

- Penyearah setengah gelombang
- Penyearah gelombang penuh sistem CT
- Penyearah gelombang penuh sistem jembatan
- Penyearah Teregulasi

2.6.1 Penyearah Setengah Gelombang

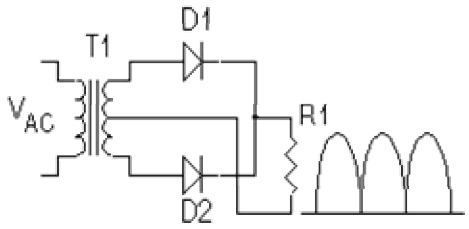
Penyearah ini bekerja dengan menggunakan satu dioda, sehingga hanya pulsa positif yang dapat terambil. Penyerah ini praktis sederhana, tetapi kekurangannya adalah bahwa gelombang keluaran bukan gelombang penuh sehingga rentan sekali akan *ripple.*



Gambar 2.14 *Penyearah Setengah Gelombang*

2.6.2 Penyearah Gelombang Penuh Sistem CT

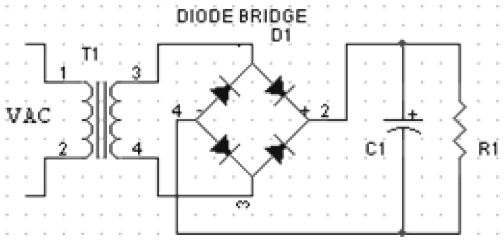
Penyearah ini menggunakan transformator jenis CT dengan dua buah dioda sebagai penyearah. Dioda bekerja secara bergantian untuk mengambil pulsa positif dan negatif, sehingga keluaran berupa gelombang penuh.



Gambar 2.15 *Penyearah Sistem CT*

2.6.3 Penyearah Gelombang Penuh Sistem Jembatan

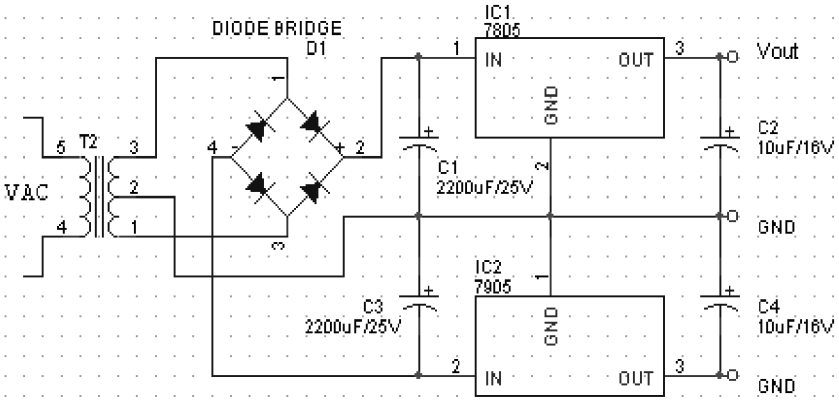
Penyearah ini menggunakan 4 buah dioda sebagai penyearah. Pada siklus pertama dua dioda bekerja untuk menyearahkan atau mengambil pulsa positif. Siklus selanjutnya dua dioda berikutnya yang bekerja untuk mengambil pulsa negatif. Keuntungan penyearah ini adalah bahwa keluaran berupa gelombang penuh dan jika salah satu dioda rusak, maka dioda yang satunya lagi akan tetap bekerja.



Gambar 2.16 *Penyearah dengan Sistem Jembatan*

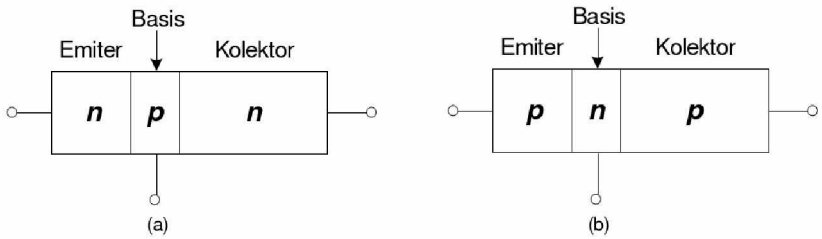
2.6.4 Penyearah Teregulasi

Tegangan hasil penyearah belum tentu stabil pada satu titik yang diinginkan, untuk itu harus ada proses untuk menstabilkan tegangan tersebut. Hal ini dapat dilakukan dengan menambahkan komponen pada keluaran penyearah, diantaranya menggunakan dioda zener, penguat operasional atau dengan IC regulator.



Gambar 2.17 *Penyearah Teregulasi menggunakan IC*

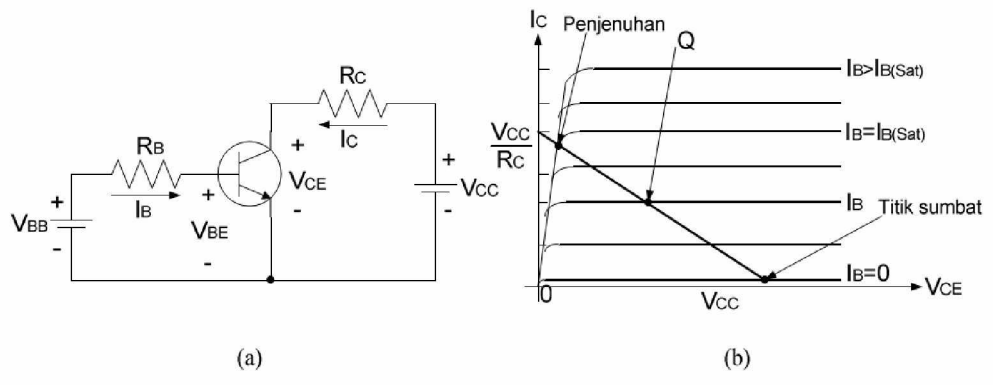
2.7 Transistor Sebagai Saklar



Gambar 2.18 *(a) Transistor npn, (b) Transistor pnp*

Transistor pada gambar 2.18(a) mempunyai dua sambungan (*junction*), satu di antaranya emiter dan basis dan basis dan kolektor. Karena inilah, sebuah transistor sama seperti dua buah dioda. Transistor *npn* disebut juga dioda emiter basis atau singkatnya dioda emiter. Transistor *pnp* juga disebut dioda kolektor basis atau dioda kolektor.

Gambar 2.18(b) menunjukan kemungkinan yang lain, yaitu sebuah transistor *pnp*. Transistor *pnp* merupakan komplemen dari transistor *npn*. Pembawa muatan mayoritas pada emiter adalah *hole*, sebagai pengganti dari muatan bebas. Ini berarti, pada transistor *pnp* dibutuhkan arus dan tegangan yang berlawanan dengan transistor *npn*.



Gambar 2.19 *(a) Rangkaian bias basis, (b) Garis beban dc*

Gambar 2.19(a) memperlihatkan rangkaian *bias basis*. Sebuah sumber tegangan V_{BB} membias maju dioda emiter melalui resistor R_B yang juga berfungsi membatasi arus. Penjumlahan tegangan di sekitar *loop input* memberikan :

$$I_B R_B + V_{BE} - V_{BB} = 0 \quad \text{.....(2.6)}$$

sehingga arus bias pada basis adalah :

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} \quad \text{.....(2.7)}$$

Dengan V_{BE} = 0,7 V untuk transistor silikon dan 0,3 V untuk germanium. Dalam rangkaian kolektor, sumber tegangan Vcc membias balik dioda kolektor melalui R_C. Persamaan tegangan kolektor emiter dapat diperoleh melalui hukum ohm, yaitu :

$$V_{CE} = V_{CC} - I_C R_C \quad \text{.....(2.8)}$$

Dalam rangkaian bias basis yang diperlihatkan Gambar 2.19(a), V_{CC} dan R_C adalah konstan. Pada persamaan 2.8 apabila disederhanakan akan dapat ditentukan besarnya arus Ic, seperti terlihat pada persamaan 2.9.

$$I_C = \frac{V_{CC} - V_{CE}}{R_C} \quad \text{.....(2.9)}$$

Gambar 2.19(b) menunjukkan grafik dari persamaan 2.9 memotong kurva dari kolektor. Perpotongan vertikal adalah pada V_{CC}/R_C dan perpotongan horizontal pada V_{CC}. Garis ini disebut *garis beban dc* karena garis ini menyatakan semua titik operasi yang mungkin. Perpotongan dari garis beban DC dengan arus basis adalah titik operasi kerja dari transistor. Titik di mana garis beban memotong kurva I_B = 0 disebut *titik sumbat* (*cutoff*). Pada titik ini, arus basis nol dan arus kolektor sangat kecil, sehingga dapat diabaikan (hanya ada arus bocor I_{CEO}). Pada titik sumbat, dioda emiter tidak lagi

dibias maju, dan transistor kehilangan kerja normalnya. Untuk itu digunakan suatu pendekatan, bahwa tegangan kolektor-emiter adalah :

$$V_{CE(cutoff)} \cong V_{CC} \quad \text{.....(2.10)}$$

Perpotongan dari garis beban dan kurva $I_B = I_{B(sat)}$ disebut saturasi. Pada titik ini, arus basis sama dengan $I_{B(sat)}$ dan arus kolektor adalah maksimum. Pada saturasi, dioda kolektor tidak lagi dibias balik, dan transistor kehilangan kerja normalnya. Untuk itu digunakan suatu pendekatan, bahwa arus kolektor pada saturasi adalah seperti di perlihatkan pada persamaan 2.9.

$$I_{C(sat)} \cong \frac{Vcc}{Rc} \quad \text{.....(2.11)}$$

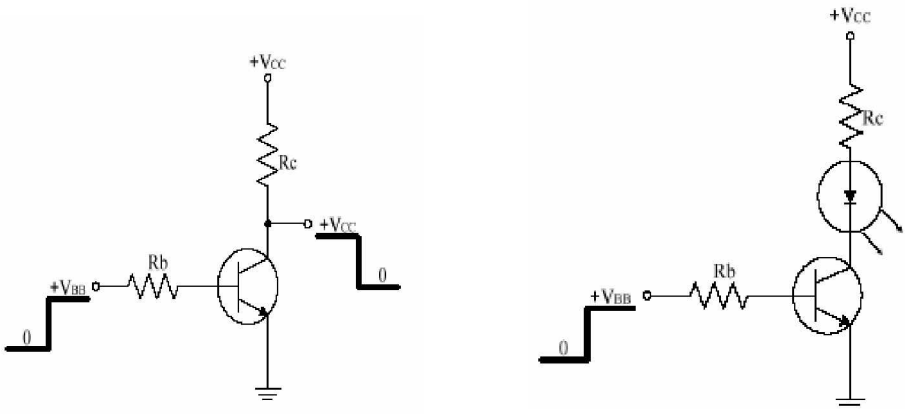
dan arus basis yang tepat menimbulkan saturasi adalah seperti diperlihatkan pada persamaan 2.12.

$$I_{B(sat)} = \frac{I_{C(sat)}}{\beta dc} \quad \text{.....(2.12)}$$

Dengan β_{dc} merupakan penguatan arus. Salah satu penggunaan dari transistor adalah sebagai *switch* atau saklar, artinya bahwa mengoperasikan transistor pada salah satu dari saturasi atau titik sumbat, tetapi tidak di tempat-tempat sepanjang garis beban. Jika sebuah transistor berada dalam keadaan saturasi, transistor tersebut seperti sebuah *switch* yang tertutup dari kolektor ke emiter. Jika transistor tersumbat (*cutoff*), transistor seperti sebuah *switch* yang terbuka. Dalam transistor dikenal istilah aturan disain *Soft saturation* dan *Hard saturation*. *Soft saturation* berarti kita membuat transistor hampir saturasi, di mana arus basis hanya cukup untuk mengoperasikan transistor pada ujung atas dari garis beban. *Soft saturation* tidak dapat diandalkan pada produksi masal karena adanya perubahan-perubahan pada β_{dc} dan I_B(sat). *Soft saturation* akan mengacu pada rancangan di mana transistor akan jenuh secara terbatas, dalam hal ini penguatan arus hanya sedikit lebih kecil daripada penguatan arus aktif.

Pada kondisi *Hard saturation*, berarti terdapat arus basis yang cukup kuat untuk membuat transistor saturasi pada semua harga dari β_{dc}. Untuk keadaan yang paling buruk dari temperatur dan arus, hampir semua transistor silikon sinyal kecil mempunyai β_{dc} lebih besar daripada 10. Karena itu, suatu pedoman disain untuk *hard saturation* adalah mempunyai arus basis kira-kira sepersepuluh dari harga saturasi arus kolektor; ini menjamin *hard saturation* pada semua kondisi kerja. Sebagai contoh, jika ujung atas garis beban mempunyai arus kolektor sebesar 10 mA, maka akan didapatkan arus basis sebesar 1 mA. Hal ini menjamin keadaan saturasi untuk semua transistor, arus,

temperatur dan sebagainya. Di gunakan aturan 10 : 1 dalam proses mendisain rangkaian *switching* transistor, kecuali jika di tentukan lain. Ingat, ini hanya suatu pedoman. Jika nilai tahanan standar menghasilkan perbandingan I_C/I_B sedikit lebih besar daripada 10, hampir setiap transistor sinyal kecil akan menuju keadaan *hard saturation*.



(a) (b)

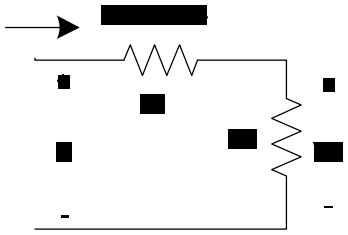
Gambar 2.20 *Contoh transistor yang digunakan sebagai switch*

Gambar 2.20(a) menunjukkan sebuah rangkaian *switching* transistor yang digerakkan oleh tegangan step. Jika tegangan *input* nol, transistor tersumbat (*cutoff*). Dalam hal ini, transistor kelihatannya seperti sebuah *switch* yang terbuka. Dengan tidak adanya arus yang mengalir melalui tahanan kolektor, maka tegangan *output* sama dengan $+V_{BB}$.

Gambar 2.20(b) menunjukan rangkaian *switching* transistor dengan sedikit variasi. Rangkaian ini disebut *LED driver*, karena transistor mengendalikan LED. Jika tegangan *input* rendah (*low*), transistor akan tersumbat (*cutoff*) dan LED padam. Jika tegangan *input* tinggi (*high*), transistor saturasi dan LED menyala.

2.8 Pembagi Tegangan

Dengan mengkombinasikan tahanan-tahanan dan sumber tegangan, maka didapatkan suatu metode untuk memperpendek kerja dalam menganalisis sebuah rangkaian, metode tersebut adalah rangkaian pembagi tegangan. Pembagian tegangan digunakan untuk menyatakan tegangan melintasi salah satu di antara dua tahanan seri, dinyatakan dalam tegangan melintasi kombinasi itu.



Gambar 2.21 *Rangkaian Pembagi Tegangan*

Pada gambar 2.21 nilai tegangan yang melewati R_1 dapat di hitung dengan :

$$V_1 = \frac{R_1}{R_1 + R_2} xV \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.13)$$

Dan nilai tegangan yang melewati R_2 yaitu :

$$V_2 = \frac{R_2}{R_1 + R_2} xV \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.14)$$

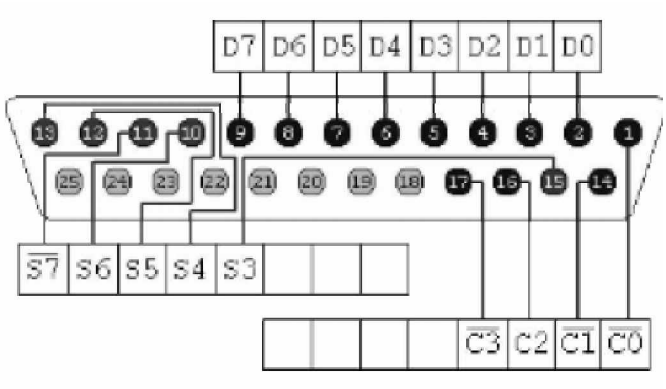
2.9 *Port* Paralel

Menurut cara komunikasi datanya, I/O pada PC dapat dibedakan menjadi dua jenis yaitu Serial dan Paralel. I/O jenis serial, mengirim dan menerima data dengan format seri bit demi bit dengan diawali *bit Start* dan diakhiri *bit Stop* untuk sinkronisasi. I/O serial memiliki efisiensi terhadap media transmisi yang lebih sedikit, yakni hanya membutuhkan dua buah jalur komunikasi untuk pengiriman dan penerimaan data, dan satu buah jalur *ground*. Namun, kompleksitas masalah dan kecepatan transmisi yang cenderung lambat, menjadi suatu kelemahan yang dimilikinya. Kerenanya, I/O jenis serial ini banyak digunakan untuk komunikasi jarak jauh.

Untuk komunikasi jarak dekat, I/O jenis paralel lebih banyak memberikan keuntungan. Sebab, I/O paralel memiliki kesederhanaan rangkaian dan kecepatan akses data yang pasti lebih cepat dibanding I/O serial. Hal tersebut adalah suatu konsekwensi terhadap cara komunikasi format paralel, yang dilakukan I/O jenis paralel. I/O ini mentransmisikan 8 bit data secara serempak, seluruh bit pada satu format data dalam satu siklus mesin. *Port* paralel pada PC umumnya digunakan untuk printer, dengan konektor female DB-25 sebagai saluran fisiknya. Aslinya, saluran ini memiliki total keluaran digital sebanyak 12 pin dan total masukan digital sebanyak 5 pin, yang digolongkan menjadi 3 bagian (seperti dalam gambar 2.22), yakni:

- Saluran Data, pin 2 sampai 9
- Saluran Status, pin 10, 11, 12, 13, dan 15, dan

- Saluran Kendali, pin 1, 14, 16, dan 17.



Gambar 2.22 *Posisi Pin Port Paralel*

Standar *port* paralel yang baru ialah IEEE 1284 di mana dikeluarkan tahun 1994. Standar ini mendefinisikan 5 mode operasi *port* paralel sebagai berikut :

1. Mode kompatibilitas
2. Mode *nibble*
3. Mode *byte*
4. Mode EPP (*enhanced parallel port*)
5. Mode ECP(*Extended capability port*)

Cara *compatibility, nibble,* dan *byte* terdapat pada *port* paralel asli PC. Sedangkan cara EPP dan ECP dibangun dengan penambahan perangkat keras sedemikian rupa, sehingga *port* paralel dapat bekerja lebih cepat dan memiliki tambahan fungsi.

Normalnya, keluaran *port* paralel adalah sesuai dengan level TTL. Arus yang dialirkan *port* paralel sekitar 12mA, sementara kecepatan pemindahan data ada diantara 50 - 150 kbps. Pada tabel 2.5 diperlihatkan penetapan pin-pin penghubung dari *port* paralel.

Tujuan dari standar yang baru tersebut ialah untuk mendesain driver dan peralatan yang baru yang kompatibel dengan peralatan lainnya serta standar paralel *port* (SPP) sebelumnya yang diluncurkan tahun 1981. *Mode compatibilitas, nibble* dan *byte* digunakan sebagai standar perangkat keras yang tersedia di-*port* paralel orisinal di mana EPP dan ECP membutuhkan tambahan hardware di mana dapat berjalan dengan kecepatan yang lebih tinggi. Mode kompatibilitas atau (“*Mode Centronics*”) hanya

dapat mengirimkan data pada arah maju pada kecepatan 50 kbytes per detik hingga 150 Kbyte per detik. Untuk menerima data, anda harus mengubah mode menjadi *mode nibble* atau *byte*. *Mode nibble* dapat menerima 4 bit (*nibble*) pada arah yang mundur, misalnya dari alat ke komputer. *Mode byte* menggunakan fitur *bi-directional parallel* untuk menerima 1 byte (8 bit) data pada arah mundur. IRQ (*Interrupt Reques*) pada *port* paralel biasanya pada IRQ5 atau IRQ7. *Port parallel Extend* dan *Enhanced* menggunakan *hardware* tambahan untuk membangkitkan dan mengatur *handshaking*. Untuk mengeluarkan 1 *byte* ke *printer* menggunakan mode kompatibilitas, *software* harus :

1. Menulis byte ke data *port*.
2. Cek untuk melihat apakah printer sibuk, jika sibuk, ia tidak akan menerima data, sehingga data yang telah ditulis akan hilang.
3. Buat *strobe* (pin 1) rendah. Ini memberitahukan printer bahwa data yang benar telah berada di line data.
4. Buat *strobe* tinggi lagi setelah menunggu sekitar 5 mikro detik setelah membuat *strobe low.*

Hal ini membatasi kecepatan data. Sedangkan EPP dan ECP mengizinkan *hardware* mengecek jika *printer* sibuk dan mengeluarkan sinyal *strobe* atau *handshaking* lainnya. Ini berarti hanya 1 instruksi I/O yang harus dilakukan yang akan meningkatkan kecepatan *port ECP* juga mempunyai kelebihan menggunakan saluran DMA dan *buffer FIFO*, jadi data dapat digeser tanpa menggunakan instruksi I/O. Protokol EPP mempunyai 4 macam siklus transfer data yang berbeda yaitu :

1. Siklus baca data (Data read)
2. Siklus baca alamat (Address Read)
3. Siklus tulis data (data write)
4. Siklus tulis alamat (address write)

Siklus data digunakan untuk mentrasfer data antara host dan peripheral. Siklus alamat digunakan untuk mengirimkan alamat, channel atau informasi perintah dan control.

Berikut ialah tabel nama pin dari konekter DB25 yang umum digunakan di komputer sebagai *port* paralel.

Tabel 2.5 *NamaPin dariKonektorPort Paralel (DB-25)*

No. Pin	***Signal name***	***Direction***	***Register– bit***	***Inverted***
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

2.9.1 Pengalamatan *Port* Paralel Standar

Port Paralel mempunyai 3 alamat seperti pada tabel 2.6. 3BCH ialah alamat dasar yang diperkenalkan sejak munculnya *port* Paralel pada kartu video yang kemudian tidak digunakan lagi. LPT1 ialah *line* printer dengan alamat 378h, lalu LPT2 dengan alamat 278H meskipun alamat ini dapat dirubah. Saat ini, alamat 378H dan 278H umum digunakan sebagai alamat *port* Paralel.

Tabel 2.6 *Alamat Port Paralel*

Alamat	**Penjelasan**
3BCH- 3BFH	Digunakan untuk paralel *port* di kartu video, tidak mendukung alamat ECP
378H- 37FH	Alamat untuk LPT1
278H- 27FH	Alamat untuk LPT2

Ketika komputer dihidupkan, BIOS (*Basic Input/Output System*) akan menentukan jumlah *port* yang dimiliki dan mengeset nama LPT1, LPT2 & LPT3 ke masing-masing *port* tersebut. Pertama kali BIOS mencari alamat 3BCH. Jika *port* paralel ditemukan, maka BIOS akan mengeset dengan nama LPT1, lalu mencari alamat lainnya yaitu 378H. *Port* paralel standar memiliki 3 buah register, yaitu :

1. Register Data
2. Register Status
3. Register Kontrol

Tabel 2.7 *Register pada Port Parallel (LPT1)*

Register	Alamat
Register data (baseaddress + 0)	378H
Register status (baseaddress + 1)	379H
Register kontrol (baseaddress + 2)	37AH

2.10 Bahasa Assembly

Assembler adalah program komputer yang men-translitrasi program dari bahasa *assembly* ke bahasa mesin. Sedangkan bahasa *assembly* adalah ekuivalensi bahasa mesin dalam bentuk alpanumerik. *Mnemonics* alpanumerik digunakan sebagai alat bantu bagi programer untuk memprogram mesin komputer daripada menggunakan serangkaian 0 dan 1 (bahasa mesin) yang panjang dan rumit.

2.10.1 Konstruksi Program Assembly

Program sumber *assembly* terdiri dari kumpulan baris-baris perintah dan biasanya di simpan dengan extension .ASM dengan 1 baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian, yakni bagian label, bagian *mnemonic*, bagian operand yang bisa lebih dari satu dan terakhir bagian komentar. Program sumber (*source code*) dibuat dengan program *pinnacle* 52. Hasil kerja program yang telah dikompile dalam bahasa *assembler* ini adalah "*assembly listing*". dan juga "file dengan ekstensi HEX". File dengan ekstensi HEX inilah yang akan diisikan kedalam Chip Mikrokontroler. Ketentuan penulisan *source code* adalah sebagai berikut:

1. Masing-masing bagian dipisahkan dengan spasi atau TAB, khusus untuk *operand* yang lebih dari satu masing-masing operand dipisahkan dengan koma.
2. Bagian-bagian tersebut tidak harus semuanya ada dalam sebuah baris, jika ada satu

bagian yang tidak ada maka spasi atau TAB sebagai pemisah bagian tetap harus ditulis.

3. Bagian label ditulis mulai huruf pertama dari baris, jika baris bersangkutan tidak mengandung label maka label tersebut digantikan dengan spasi atau TAB, yakni sebagai tanda pemisah antara bagian label dan bagian *mnemonic*.

2.10.2 Instruksi MCS-51 Yang Digunakan

Beberapa instruksi yang digunakan dalam penyusunan program adalah sebagai berikut :

1. EQU
   Konstanta Data yang dideklarasikan dengan nilai pada variabel tertentu.
2. BIT
   Konstanta Bit yang dideklarasikan dengan 1 bit data dari variabel tertentu.
3. ORG
   Digunakan untuk menunjukkan lokasi memori tempat instruksi atau perintah yang ada di bawahnya disimpan.
4. CLR
   Memberikan nilai “nol” pada bit tertentu.
5. SETB
   Memberikan nilai “satu” pada bit tertentu.
6. CALL
   Instruksi melakukan lompatan dengan area sebesar 2 KByte.
7. LCALL
   Sama seperti instruksi CALL, hanya instruksi ini digunakan jika label yang dipanggil letaknya lebih jauh dari 2 KByte
8. MOV
   Instruksi ini melakukan pemindahan data dari variabel pada kode operasi kedua dan disimpan di variabel pada kode operasi pertama.
9. MOVC
   Instruksi ini melakukan pemindahan data dari variabel pada kode operasi kedua ditambah *carry* dan disimpan di variabel pada kode operasi pertama.

10. JMP

    Melakukan lompatan dan menjalankan program yang berada di alamat yang ditentukan oleh label tertentu.

11. SJMP

    Melakukan lompatan untuk jarak yang pendek (*Short JMP*).

12. LJMP

    Melakukan lompatan untuk jarak yang jauh (*Long JMP*).

13. JZ

    Lompat jika akumulator bernilai “nol”. Jika semua bit pada akumulator bernilai “nol”, maka lompat ke alamat yang telah ditentukan dan dilanjutkan ke instruksi berikutnya.

14. JB

    Lompat jika Bit tertentu bernilai “satu”, artinya jika bit yang telah ditentukan bernilai “satu”, maka lompat ke alamat yang telah ditentukan dan dilanjutkan ke instruksi berikutnya.

15. JNB

    Kebalikan dari JB, JNB merupakan instruksi untuk beralih ke alamat tertentu jika Bit tertentu bernilai “nol”.

16. DJNZ

    Melakukan pengurangan pada Rn (R0…R7) dengan 1 dan lompat ke alamat yang ditentukan apabila hasilnya bukan 00. Apabila hasilnya telah mencapai 00, maka program akan terus menjalankan instruksi dibawahnya.

17. CJNE

    Instruksi ini melakukan perbandingan antara data tujuan dan data sumber serta melakukan lompatan ke alamat yang ditentukan jika hasil perbandingan tidak sama.

18. RET

    Instruksi ini melakukan lompatan ke alamat yang disimpan dalam SP dan SP-1. Instruksi ini biasa digunakan pada saat kembali dari subrutin yang dipanggil dengan instruksi ACALL atau LCALL.

19. SWAP

    Baca 4 bit (*nibble*) data pertama LSB dari akumulator. Perintah SWAP memungkinkan untuk menukar 4 bit rendah menjadi 4 bit tinggi dari akumulator (berlaku sebaliknya).

20. PUSH

    Melakukan penyimpanan data dari suatu register atau memori ke dalam stack. Lokasi data dalam stack tersebut ditunjuk oleh *stack pointer*. Pada saat instruksi ini dijalankan, nilai dari stack pointer akan bertambah satu dan register atau memori yang di PUSH akan masuk ke dalam alamat yang ditunjuk oleh *stack pointer* tersebut.

21. POP

    Melakukan pengambilan data dari dalam stack ke suatu register atau memori. Lokasi data dalam stack tersebut ditunjuk oleh *stack pointer*. Pada saat instruksi ini dijalankan, data di alamat yang ditunjuk oleh stack pointer akan dipindah ke register atau memori dilanjutkan dengan pengurangan nilai *stack pointer* dengan 1.

22. ANL

    Logika AND untuk variabel tertentu. Perintah ANL bekerja dengan melakukan operasi AND antara variabel yang ditentukan dengan nilai tertentu untuk kemudian hasilnya disimpan pada alamat yang ditunjuk oleh variabel yang ditentukan.

23. ORL

    Logika OR untuk variabel tertentu. Perintah ORL bekerja dengan melakukan operasi OR antara variabel yang ditentukan dengan nilai tertentu untuk kemudian hasilnya disimpan pada alamat yang ditunjuk oleh variabel yang ditentukan.

24. CPL

    Tiap Bit dari akumulator yang secara logika di komplemen satu. Misalnya nilai yang asalnya 1, setelah dikomplemen berubah menjadi 0. Dari faktor tersebut, CPL dapat diartikan sebagai instruksi untuk menukar nilai 0 menjadi 1 dan sebaliknya.

25. INC

    Menambahkan nilai variabel yang ditunjuk dengan 1 dan hasilnya disimpan di variabel tersebut.

26. SUBB

    Mengurangkan nilai variabel yang ditunjuk dengan nilai lain dan hasilnya disimpan di variabel tersebut.

27. ADD

    Menjumlahkan nilai variabel yang ditunjuk dengan nilai lain dan hasilnya di simpan di variabel tersebut.

28. ADDC

    Menjumlahkan nilai variabel yang ditunjuk dengan nilai lain dan ditambah dengan nilai *carry* yang kemudian hasilnya disimpan di variabel yang di tunjuk tersebut.

29. NOP

    No Operation, yang berarti tidak melakukan apa – apa. Tujuan dari instruksi ini adalah hanya untuk menambahkan siklus pulsa pewaktu sebesari 1 siklus.

30. DW

    Define Word, yang artinya mengacu pada data word (data 16 bit) tertentu.

31. DB

    Define Byte, yang artinya mengacu pada byte (data 8 bit) tertentu.

32. END

    END biasanya diletakkan di akhir baris dari file program sumber *assembler* sebagai tanda akhir pernyataan (*statement*) bagi program *assembler* dalam melakukan proses *assembly*.

2.11 Komponen Instrumentasi

2.11.1 LM 7805

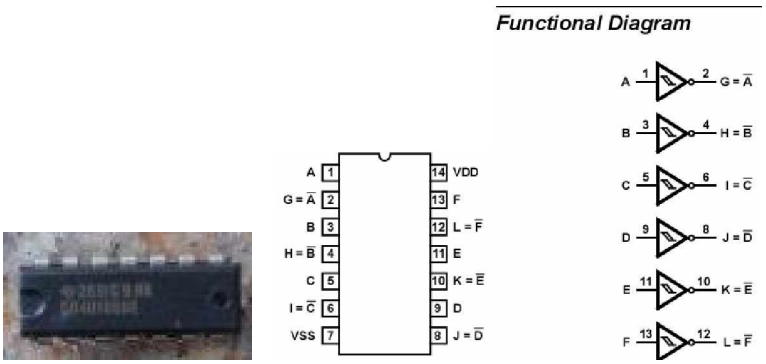
IC ini mempunyai tiga kaki yang digunakan sebagai komponen pendukung dari Vcc untuk menghasilkan tegangan 5V. IC *regulator* ini berfungsi untuk menstabilkan tegangan 5V dan dapat bekerja dengan baik jika tegangan *input* (V_{in}) lebih besar daripada tegangan *output* (V_{out}). Biasanya perbedaan tegangan *input* dengan *output* yang direkomendasikan tertera pada *datasheet* komponen tersebut. Contoh LM7805 diperlihatkan oleh gambar 2.23.



Gambar 2.23 *IC LM7805*

2.11.2 IC CD40106BE

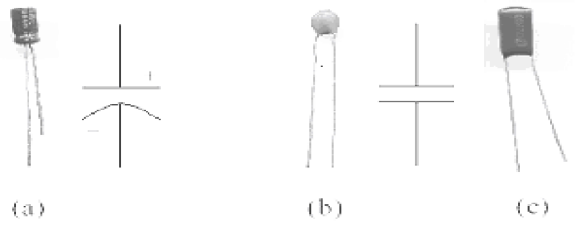
CD40106BE adalah IC yang memiliki enam rangkaian *schmitt-trigger*. Tiap rangkaian berfungsi sebagai *inverter*. Penggabungan beberapa gerbang *inverter* akan menghasilkan frekuensi osilasi dengan menggunakan waktu tunda propagasi (*propagation delay*). *Schmitt-trigger* adalah aplikasi komparator yang men-*switch* keluaran negatif ketika input melewati tegangan referensi positif. Kemudian menggunakan balikan negatif untuk mencegah *switching* kembali ke bagian lain sampai input melewati tegangan *threshold* yang lebih rendah, kemudian penstabilan *switching* terhadap *trigger* yang cepat oleh *noise*.



Gambar 2.24 *CD40106BE*

2.11.3 Kapasitor

Kapasitor disebut juga Kondensator, yaitu komponen yang berfungsi untuk menyimpan muatan/tegangan listrik atau menahan arus searah. Kapasitor ELCO (*Electrolit Capasitor*) terbuat dari keping aluminium dan elektrolit yang dikandung dalam lembaran kertas berpori. Plat aluminium bersifat sebagai isolator dan elektrolit berfungsi sebagai konduktor. Kapasitor ELCO memiliki kekutuban atau polaritas yaitu tanda positif dan tanda negatif. Jika dalam pemasangan kutub-kutub ELCO terbalik maka kapasitor akan rusak. Untuk satuan dari ELCO adalah mikro, kapasitor keramik adalah piko dan kapasitor milar adalah nano. Simbol dan contoh dari kapasitor diperlihatkan oleh gambar 2.25.



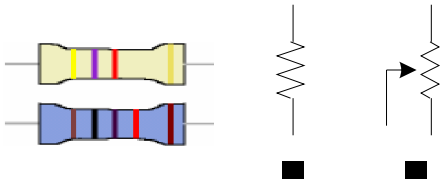
Gambar 2.25 *(a) ELCO, (b) Kapasitor keramik dan (c) Kapasitor milar*

2.11.4 Resistor

Resistor adalah komponen dasar elektronika yang digunakan untuk membatasi jumlah arus yang mengalir dalam satu rangkaian, bila tahanannya semakin besar maka semakin kecil arusnya, sebaliknya bila nilai tahanannya kecil, maka besar pula arusnya. 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. Satuan resistansi dari suatu resistor disebut *Ohm*(Ω).

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 pemakai mengenali besar resistansi tanpa mengukur besarnya dengan Ohmmeter.

Variabel resistor merupakan salah satu jenis dari resistor tiga kaki yang digunakan untuk mengetahui frekuensi maksismum dari suatu komponen dengan memutar-mutar arah variabel resistor. Simbol dan contoh resistor dan variabel resistor diperlihatkan pada gambar 2.26.



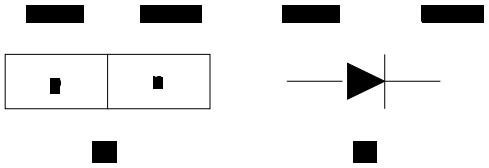
Gambar 2.26 *(a)Resistor dan Simbol (b) Simbol Variabel Resistor*

2.11.5 Dioda

Dioda adalah suatu komponen elektronik yang dapat melewatkan arus pada satu arah saja. Ada berbagai macam dioda, yaitu dioda tabung, dioda sambungan *p-n*, dioda kontak titik (*point-contact diode*) dan sebagainya.

Dioda memegang peranan yang sangat penting dalam elektronika, diantaranya adalah untuk menghasilkan tegangan searah dari tegangan bolak-balik, untuk mengeset gelombang radio, untuk membuat berbagai bentuk sinyal gelombang, untuk mengatur tegangan searah agar tidak berubah dengan beban maupun dengan perubahan tegangan jala-jala (PLN), untuk saklar elektronik, LED, laser semikonduktor, mengeset gelombang mikro dan lain-lain.

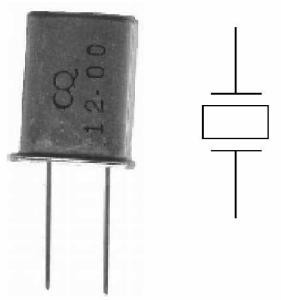
Bentuk dioda yang lazim digunakan terdiri dari semikonduktor jenis p yang dibuat bersambung dengan semikonduktor jenis n. Secara skematis dioda diperlihatkan pada gambar 2.27.



Gambar 2.27 *(a) Susunan dioda sambungan p-n, (b) Simbol dioda*

2.11.6 Kristal

Kristal merupakan pembangkit *clock* internal yang menentukan rentetan kondisi-kondisi (*state*) yang membentuk sebuah siklus mesin mikrokontroler. Siklus mesin tersebut diberi nomor S1 hingga S6, masing-masing kondisi panjangnya 2 periode osilator, dengan demikian satu siklus mesin paling lama dikerjakan dalam 12 periode osilator atau 1µs. Satuan kristal biasanya dalam skala mega dengan bentuk dan simbol kristal diperlihatkan oleh gambar 2.28.



Gambar 2.28 *Osilator/ Kristal*