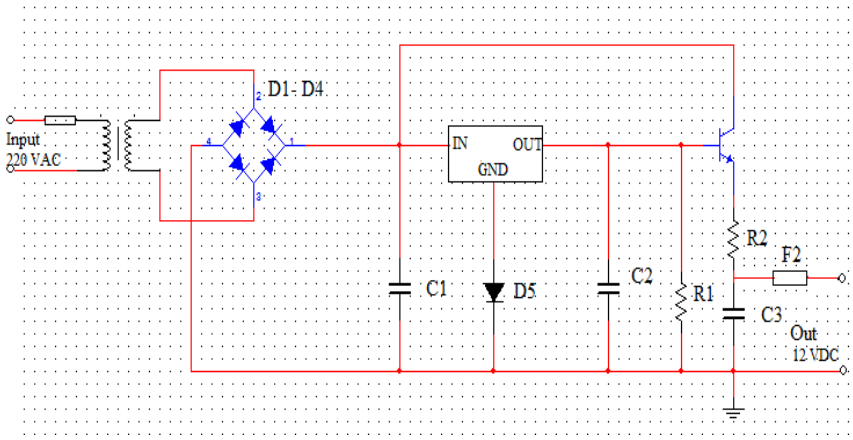
BAB II
DASAR TEORI

2.1 Teori Catu Daya

Pengertian *transformator* atau yang biasa kita kenal dengan *trafo* adalah komponen elektronika yang berfungsi untuk menaikkan atau menurunkan tegangan listrik. *Transformator* berperan dalam menyalurkan tenaga atau daya listrik dari tegangan tinggi ke tegangan yang rendah atau sebaliknya, namun dengan frekuensi yang sama. *Transformator* bekerja berdasarkan prinsip kerja induksi elektromagnetik. Dimana apabila terjadi suatu perubahan fluks magnet pada kumparan primer, maka aka n diteruskan ke kumparan sekunder dan menghasilkan suatu gaya gerak listrik (GGL) induksi dan arus induksi. Agar selalu terjadi perubahan fluks magnet, maka arus yang masuk (*input*) ini harus berupa arus bolak balik (AC). Fungsi *tranformator* yang diaplikasikan dalam kehidupan sehari-hari diantaranya :

1. *Trafo step up*, fungsi *transformator* ini digunakan untuk menaikkan tegangan AC, *trafo* jenis ini dipakai dalam rangkaian-rangkaian pembangkit tegangan pada perangkat elektronika seperti *trafo inverter* monitor LCD, *trafo* inverter TV.
2. *Trafo step-down* adalah kebalikannya, fungsi *transformator* ini untuk menurunkan tegangan AC, contoh pemakaiannya pada *adaptor*. Rangkaian *adaptor* ditunjukkan pada gambar 2.1.



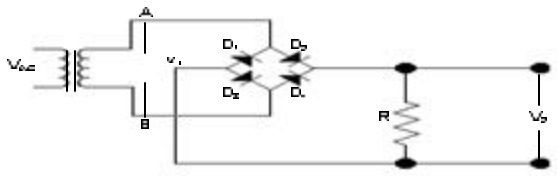
Gambar 2.1 Rangkaian *Adaptor*

Adaptor adalah sebuah rangkaian elektronika yang dapat mengubah tegangan AC menjadi DC. Rangkaian ini adalah alternatif pengganti dari sumber tegangan DC, misalnya batu baterai dan *accumulator*. Keuntungan dari *adaptor* dibanding dengan batu baterai atau *accumulator* adalah sangat praktis berhubungan dengan ketersediaan tegangan karena *adaptor* dapat di ambil dari sumber tegangan AC yang ada di rumah, di mana pada jaman sekarang ini setiap rumah sudah menggunakan listrik. Selain itu, *adaptor* mempunyai jangka waktu yang tidak terbatas asal ada tegangan AC, tegangan AC ini sudah merupakan kebutuhan primer dalam kehidupan manusia. *Adaptor* sederhana terdiri dari bagian *input* tegangan yang merupakan bagian yang berfungsi sebagai penghubung sumber tegangan AC dari stop kontak yang ada di dalam rumah. Bagian ini terdiri dari *jack/steker* dan kabel *input*. Stop kontak adalah konektor sumber tegangan AC dari listrik PLN yang digunakan untuk menyalurkan tegangan pada *adaptor* melalui kabel *input* tegangan. Bagian penurun tegangan yang berfungsi untuk menurunkan tegangan AC 220 *Volt* menjadi tegangan yang lebih kecil, misalnya 3 *volt*, 4,5 *volt*, 6 *volt*, 7,5 *volt*, 9 *volt*, atau 12 *volt*.

Untuk memilih *output* tegangan ini digunakan *rotary switch*/saklar putar/saklar 1 induk 6 anak. *Trafo* yang digunakan adalah jenis *step down*, dapat menggunakan *trafo* dengan arus 500 mA (mili Ampere). Bagian penyearah, yaitu mengubah tegangan AC menjadi tegangan DC. Komponen utamanya adalah dioda. Dioda yang digunakan berjumlah 4 dirangkai sedemikian rupa sehingga membentuk jembatan dioda atau *bridge dioda*. Bagian *filter* atau penyaring yang berfungsi untuk menghilangkan tegangan AC yang masih lewat. Efek dari tegangan AC yang lewat ini adalah munculnya suara dengung. Komponen yang dibutuhkan antara lain IC penstabil tegangan dan elco. Bagian *output* tegangan yang berfungsi sebagi keluaran tegangan berupa tegangan DC. Besar keluaran tegangan DC ini sesuai dengan tegangan *output* pada *trafo step down* yang diatur oleh *rotary switch* sesuai yang diinginkan. Fungsi dioda pada rangkaian sangat berhubungan dengan sistem pengendalian arus tegangan. Dioda penyearah (*rectifier*) berfungsi sebagai penyearah tegangan / arus dari arus bolak-balik (AC) ke arus searah (DC) atau mengubah arus AC menjadi DC [1].

2.1.1. Penyearah Jembatan

Rangkaian penyearah ini memerlukan empat buah dioda yang dipasang dengan konfigurasi jembatan seperti terlihat pada gambar 2.2



Gambar 2.2. Penyearah Jembatan [1]

Gambar 2.2. menunjukan prinsip kerja dari penyearah jembatan. Dari gambar tersebut menunjukan D_2 dan D_3 mengalami *forward* bias sementara D_1 dan D_4 mengalami *reverse* bias sehingga sama dengan rangkaian terbuka. Arus akan mengalir dari sekunder trafo titik A melalui D_2, *resistor* beban (R_L), D_3 dan kembali ke sekunder trafo titik B. Dioda D_1 dan D_4 mengalami *forward* bias sebaliknya dioda D_1 dan D_2 mengalami *reverse* bias sehingga sama dengan rangkaian terbuka. Arus akan mengalir dari sekunder *trafo* titik B melalui D_4, R_L, D_1 dan kembali ke sekunder trafo titik A. Gambar 2.2. (a), menunjukan pada saat D_1 dan D_4 mendapat tegangan berupa setengah gelombang negatif, yaitu selama t_1-t_2. Sedangkan pada gambar 2.2. (b) menunjukan saat D_2 dan D_3 mendapat tegangan berupa setengah gelombang positif, yaitu selama t_0-t_1.

2.1.2. Kapasitor Perata Tegangan

Tegangan DC yang dihasilkan dari penyearah tegangan bolak-balik (AC) masih berupa pulsa-pulsa. Sedangkan untuk keperluan *supply* rangkaian elektronika dibutuhkan tegangan DC yang mendekati murni. Untuk mendapatkan tegangan DC yang demikian, maka keluaran dari penyearah masih perlu diratakan atau diperhalus, yaitu dengan menggunakan kapasitor perata tegangan atau *filter*[2].

2.2. Teori Komponen

2.2.1. Kapasitor

Pada dasarnya kapasitor merupakan salah satu komponen pasif yang terdiri dari dua penghantar yang saling tersekat dengan bahan isolasi yang disebut *dielaktrik*. Sifat utama kapasitor adalah dapat menyimpan muatan listrik. Kapasitor terbagi dalam dua kelompok yaitu kapasitor elektrolit dan kapasitor *non* elektrolit, yang dibedakan oleh simbol elektronnya [1]. Hal ini seperti yang ditunjukan pada gambar 2.3.

Gambar 2.3 Simbol Kapasitor [3]

Kemampuan menyimpan muatan ini disebut juga kapasitansi. Kapasitansi adalah suatu ukuran jumlah pengisian kapasitor yang dapat disimpan sewaktu diberikan tegangan[4].

2.2.2. *Resistor*

Resistor adalah komponen elektronika yang mempunyai fungsi sebagai pembatas arus listrik dan membagi tegangan (*voltage divider*) dalam suatu rangkaian tertutup. Untuk perhitungan besarnya arus yang mengalir melalui sebuah tahanan maka berlaku hukum *ohm*. Hukum *ohm* menyatakan bahwa "tegangan yang ada pada berbagai jenis penghantar adalah berbanding lurus dengan arus yang mengalir pada penghantar tersebut [3]. Resistor merupakan salah satu komponen dasar elektronika yang mempunyai sifat *resistif* terbuat dari bahan karbon. *Resistor* mampu membatasi atau menghambat arus listrik dalam suatu alur yang melewati dalam suatu rangkaian. *Resistor* menentukan aliran arus dalam jalur dari sebuah rangkaian. Dimana ada resistensi yang tinggi di rangkaian mengakibatkan aliran arus kecil, dimana resistensi rendah aliran arus besar. Resistor digunakan untuk mengatur *Output* dari *input* arus yang masuk ke rangkaian. Resistor mampu memblok aliran arus. Nilai atau hambatan dari sebuah dinyatakan dengan resistansi dengan satuan

ohm (?). Komponen resistor hampir ditemukan pada rangkaian elektronik. Hal ini seperti yang ditunjukan pada tabel 2.1.

Tabel 2.1 Warna *Resistor*

Warna	Nilai	Faktor Pengali	Toleransi
Hitam	0	1	
Coklat	1	10	1%
Merah	2	100	2%
Jingga	3	1.000	
Kuning	4	10.000	
Hijau	5	100.000	
Biru	6	10^6	
Violet	7	10^7	
Abu-abu	8	10^8	
Putih	9	10^9	
Emas	-	0.1	5%
Perak	-	0.01	10%
Tanpa Warna	-	-	20%

2.2.3. *Light Emitting Dioda* (LED)

LED (*Light emiting diode*) atau dioda penghasil cahaya adalah dioda yang dapat memancarkan cahaya jika mendapat arus listrik atau tegangan. Dengan kata lain, dioda mengkonversikan energi listrik menjadi energi cahaya. Hal ini disebabkan terjadinya tumbukan dalam proses penggabungan antara elektron dan *hole*. Saat dioda diberi tegangan maju maka kutub negatif baterai menyuntikkan elektron ke layar N *(katoda)* dan elektron ini akan menuju kearah persambungan. Ini juga akan menyebabkan *hole* pada layar P menuju kearah persambungan. Penggabungan antara elektron dan *hole* pada daerah pertemuan akan menghasilkan energi cahaya bila elektron memiliki energi.

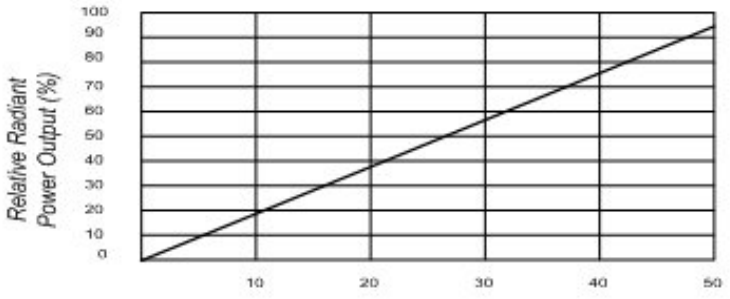
Warna cahaya yang dihasilkan LED ditentukan oleh bahan pembentuknya. Sebagai contoh, LED yang terbuat dari bahan *gallium arsenida fosfida* (GaAsP) akan memancarkan cahaya merah dengan panjang gelombang kira-kira 5600 *angstroms*. Bahan GaAsP juga memiliki jangkauan panjang gelombang yang cukup lebar yang dapat dihasilkan,

yaitu dengan mengatur jumlah *fosfida* dalam bahan. Dengan pengaturan ini, panjang gelombang yang dapat dihasilkan berkisar antara 5500 sampai dengan 9100 *angstroms*.

LED dengan bahan *gallium arsenida* (GaAs) menghasilkan cahaya dengan panjang gelombang kira-kira 9000 *angstroms* yang masuk dalam spektrum cahaya *infrared* sehingga tidak dapat dilihat oleh mata manusia. Hal ini seperti yang ditunjukan pa da gambar 2.4 dan gambar 2.5.

Gambar 2.4 Simbol LED [4]

Keluaran daya cahaya dari LED diperlihatkan seperti Gambar 2.5



Gambar 2.5 Karakteristik LED dari bahan GaAsP [5]

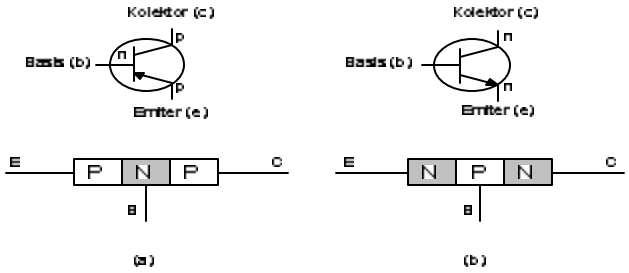
Dalam penerapannya, LED dapat digunakan sebagai *indikator* dan sebagai penampil digital dalam peralatan, kalkulator elektronik, jam *digital*, dan *arloji*. LED pada umumnya dipasang seri dengan tahanan untuk membatasi arus agar tidak melebihi kemampuan dari LED itu sendiri, sehingga arus yang mengalir tidak merusak LED.

2.2.4. *Transistor*

Transistor adalah komponen *semikonduktor* yang dapat digunakan untuk memperkuat sinyal listrik. Disamping itu transistor dapat difungsikan

pula sebagai saklar *elektronik. Transistor* dibuat dari bahan *semikonduktor* kristal silikon atau *germaniu m* yang disusun tiga lapis. Sesuai dengan bahan pembuatan serta penyusunan lapisan muatannya, transistor dibedakan dalam dua tipe. Hal ini seperti yang ditunjukan pada gambar 2.6.

1. Transistor tipe PNP
2. Transistor tipe NPN



terhubung singkat karena tahanan sangat kecil bahkan mendekati nol dan arus *collector* (Ic) mencapai maksimum, sehingga kondisi ini dapat di katakan kondisi saturasi. Kondisi kedua adalah *"OFF"*, Pada kondisi ini *transistor non conduct* (tidak bekerja) sehingga kaki *collector* dan *emitter* dari *transistor* seolah-olah terbuka karena tahanannya sangat besar bahkan mendekati tak terhingga dan arus *collector* (Ic) sangat minimum (mendekati nol), sehingga kondisi ini dapat dikatakan kondisi *cut off* [3].

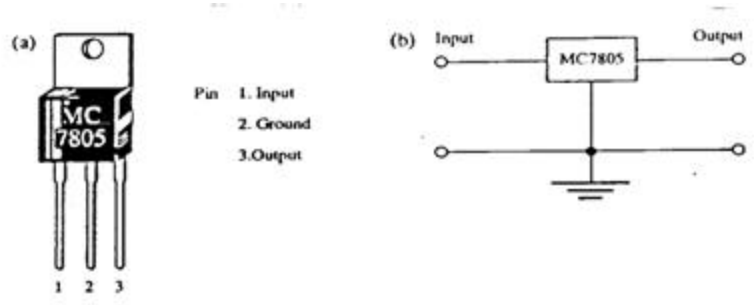
2.2.5. IC Pengatur Tegangan *(Regulator)*

Regulator tiga terminal adalah "*Integrated Voltage Regulator Circuit*" yang dirancang untuk mempertahankan tegangan *output*nya tetap dan mudah untuk dirangkai.

Keuntungannya adalah :

1. Membutuhkan penambahan komponen luar yang sangat sedikit, ukuran kecil
2. Mempunyai proteksi terhadap arus hubung singkat.
3. Mempunyai *automatic thermal shutdown* .
4. Mempunyai tegangan *output* yang sangat konstan.
5. Mempunyai arus rendah.
6. Mempunyai *ripple output* yang sangat kecil.
7. Pembiyaan rendah

Gambar 2. 7 memperlihatkan contoh IC *Regulator* tegangan positif tiga terminal MC 7805.



Gambar 2. 7 Bentuk IC *Regulator* Dan Simbol Rangakain [7]

Seri LM 7805 adalah *Regulator* dengan tiga terminal, dapat diperoleh dengan berbagai tegangan tetap. Beberapa IC *Regulator*

mempunyai kode yang dibuat oleh pabrik pembuat komponen, sebagai contoh : IC LM.7805 AC Z yang artinya sebagai berikut:

- LM (*Linear Monolithic*)
- 78L (Bagian nomor dasar yang menyatakan tegangan positip)
- 06 (Tegangan *output*)
- AC (Standart kerapatan)
- Z (Tipe pembungkus)
- TO-92 (*plastic*)

Seri LM 7805C dapat diperoleh dalam kemasan TO-3 alamunium, arus keluaran (*output*) 1A, boleh lebih asalkan IC *Regulator* lengkapi dengan pendingin (*heat sink*). *Regulator* LM 7805C mudah dipakai dan tambahan komponen- komponen ektern tidak banyak

Sifat-sifat IC *Regulator* LM 7805 adalah sebagai berikut :

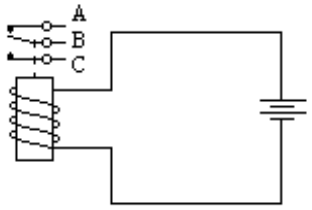
1. Arus keluaran melebihi 1A .
2. Pengamanan pembebanan lebih termik.
3. Tidak diperlukan komponen tambahan.
4. Ada pengamanan untuk transistor keluaran (*output*)
5. Dapat diperoleh dalam kemasan TO-3 aluminium

2.2.6. *Relay*

Sebuah kawat yang lurus bila dialiri arus listrik maka disekelilingnya akan timbul medan magnet, untuk memusatkan medan magnet yang dihasilkan oleh arus yang mengalir pada kawat itu maka kawat harus digulung atau dililit melingkar [1].

Dengan demikian medan magnet di sekeliling kawat yang dililit melingkar akan saling menambah sehingga menghasilkan medan magnet yang kuat pada ujung-ujung kumparan, *elektromagnet* semacam ini disebut *solenoid*. Bila kumparan dialirkan arus dengan arah yang tetap maka kutub magnet yang dihasilkan akan tetap, tetapi sebaliknya apabila arus bolak-balik yang mengalir pada kumparan maka kutub magnet juga akan bergantian sesuai dengan arah arusnya.

Sebagai inti dari kumparan dapat berupa gulungan kertas, plastik dan besi lunak. Untuk mendapatkan medan magnet yang kuat maka sebaiknya ini gulungan adalah dari besi lunak. Pada dasarnya *relay* terdiri dari sebuah lilitan kawat tembaga (kumparan) yang terlilit dari suatu inti yang berasal dari besi lunak. Jika kumparan dialiri arus listrik, maka besi lunak berubah menjadi magnet. Hal ini akan menolak lidah (pegas) dan lidahpun terlepas. Suatu contoh konstruksi *relay* seperti pada gambar 2.8.



Gambar 2.8 Salah satu bentuk konstruksi *relay* [8]

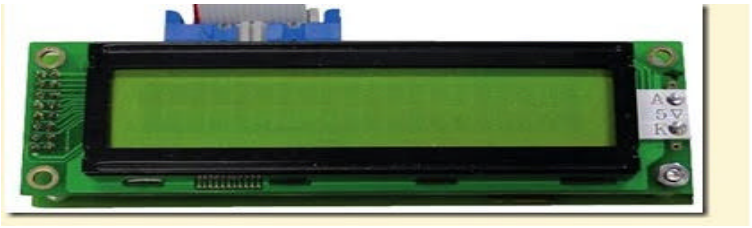
Jika kumparan dialiri arus maka inti besi lunak menjadi magnet dan inti akan menarik jangkar sehingga kontak A dan B akan terputus *(open)*, serat kontak B dan C akan terhubung *(close)*. Jenis *relay* semacam ini dinamaka *relay* kontak tukar.

2.2.7. Penampilan LCD (*Liquid Crystal Display*) [9]

Banyak sekali kegunaan LCD dalam perancangan suatu sistem yang menggunakan mikrokontroler. LCD berfungsi menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi mikrokontroler. LCD yang digunakan adalah jenis LCD M1632. LCD M1632 merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya rendah. Modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD.

Interface LCD merupakan sebuah parallel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8 bit dikirim ke LCD secara 4 atau 8 bit pada satu waktu. Jika mode 4 bit yang digunakan, maka 2 *nibble* data dikirim untuk membuat sepenuhnya 8 bit (pertama

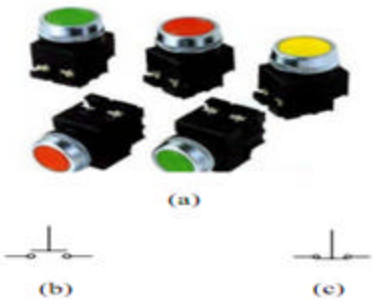
dikirim 4 bit MSB lalu 4 bit LSB dengan pulsa *clock* EN setiap *nibble*). Gambar 2.9 memperlihatkan contoh LCD 16*2.



Gambar 2.9 Bentuk LCD 16*2[9]

2.2.8. *Push Botton*[9]

Prinsip kerja *Push Button* adalah apabila dalam keadaan normal tidak ditekan maka kontak tidak berubah, apabila ditekan maka kontak NC akan berfungsi sebagai *stop* (memberhentikan) dan kontak *NO* akan berfungsi sebagai *start* (menjalankan) biasanya digunakan pada sistem pengontrolan motor induksi untuk menjalankan atau mematikan motor pada industri. Gambar 2.10 memperlihatkan contoh *Push Botton*



Gambar 2.10 *Push Botton*[9]

2.3. Mikrokontroler ATMega8 [10]

AVR ATMega8 adalah mikrokontroler CMOS 8-*bit* berarsitektur AVR RISC yang memiliki 8K *byte in-System Programmable Flash*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16MIPS pada frekuensi 16MHz. Jika dibandingkan dengan ATMega8L perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATMega8 tipe L, mikrokontroler ini dapat bekerja dengan tegangan

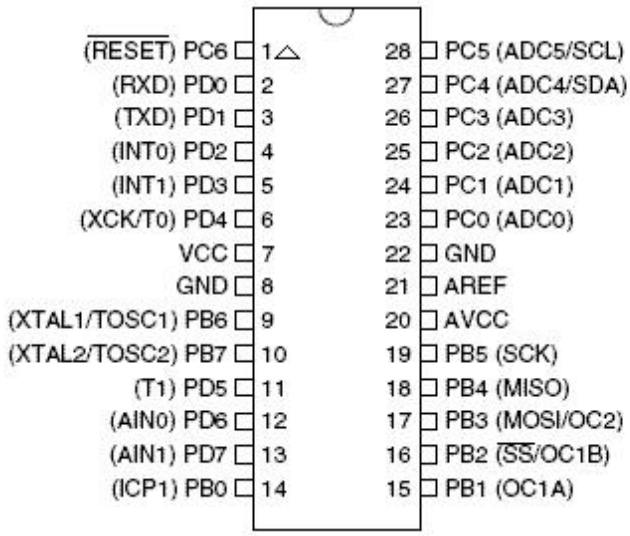
antara 2,7-5,5 V sedangkan untuk ATMega8 hanya dapat bekerja pada tegangan antara 4,5–5,5 V.

AVR merupakan salah satu jenis mikrokontroler yang didalamnya terdapat berbagai macam fungsi. Perbedaannya pada mikro yang pada umumnya digunakan seperti MCS51 adalah pada AVR tidak perlu menggunakan *oscillator eksternal* karena di dalamnya sudah terdapat internal*oscillator.*

Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset,* yaitu tidak perlu ada tombol *reset* dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset.* Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 *byte* sampai dengan 512 *byte.*

2.3.1. Konfigurasi Pin ATMega8

Berikut konfugurasi Pin ATMega8 seperti pada gambar 2.11.



Gambar 2.11 Konfigurasi Pin ATMega8[10]

Pada Gambar 2.11 menjelaskan ATMega8 memiliki 28 *pin,* yang masing-masing *pin* nya memiliki fungsi yang berbeda-beda baik sebagai *port* maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki ATMega8 :

- VCC Merupakan *supply* tegangan digital.
- GND Merupakan *ground* untuk semua komponen yang membutuhkan *grounding*.
- *Port* B (PB7...PB0) Didalam *Port* B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah *Port* B adalah 8 buah *pin*, mulai dari *pin* B.0 sampai dengan B.7. Tiap *pin* dapat digunakan sebagai *input* maupun *output*. *Port* B merupakan sebuah 8-*bit bi-directional* I/O dengan *internal pull-up* resistor.
- *Port* C (PC5…PC0) *Port* C merupakan sebuah 7-*bit bi-directional* I/O *port* yang di dalam masing-masing *pin* terdapat *pull-up* resistor. Jumlah *pin* nya hanya 7 buah mulai dari *pin* C.0 sampai dengan *pin* C.6. Sebagai keluaran/*output port* C memiliki karakteristik yang sama dalam hal menyerap arus (*sink*) atau pun mengeluarkan arus *(source*).
- RESET/PC6 Jika RSTDISBL *Fuse* diprogram, maka PC6 akan berfungsi sebagai *pin* I/O. *Pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada *port* C lainnya. Namun jika RSTDISBL *Fuse* tidak diprogram, maka *pin* ini akan berfungsi sebagai *input reset*.
- *Port* D (PD7…PD0) *Port* D merupakan 8-*bit bi-directional* I/O dengan internal *pull-up* resistor. Fungsi dari *port* ini sama dengan *port-port* yang lain.
- AVcc *Pin* ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk analog saja.
- AREF Merupakan *pin* referensi jika menggunakan ADC.

Pada AVR status *register* mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi instruksi aritmatik. Informasi ini digunakan untuk altering arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Register ini di *update* setelah

operasi *Arithmetic Logic Unit* (ALU) hal tersebut seperti yang tertulis dalam *datasheet* khususnya pada bagian *Instruction Set Reference.*

Dalam hal ini untuk beberapa kasus dapat membuang pengunaan kebutuhan instrukasi perbandingan yang telah di dedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat.

Register ini tidak secara otomatis tersimpan ketika mamasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal tersebut harus dilakukan melalui *software*. Berikut Register ATMega8 seperti pada gambar 2.12.

Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C

Gambar 2.12 Register ATMega8

Read/wriet	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Keterangan:

- Bit 7(I) Merupakan *bit Global Interrupt Enable.*
- Bit 6(T) Merupakan *bit Copy Storage.*
- Bit 5(H) Merupakan *bit Half Carry Flag.*
- Bit 4(S) Merupakan *Sign bit.*
- Bit 3(V) Merupakan *bit Two's Complement Overflow Flag.*
- Bit 2(N) Merupakan *bit Negative Flag.*
- Bit 1(Z) Merupakan *bit Zero Flag.*
- Bit 0(C) Merupakan *bit Carry Flag.*

2.3.2. Komunikasi Serial Pada ATMega8

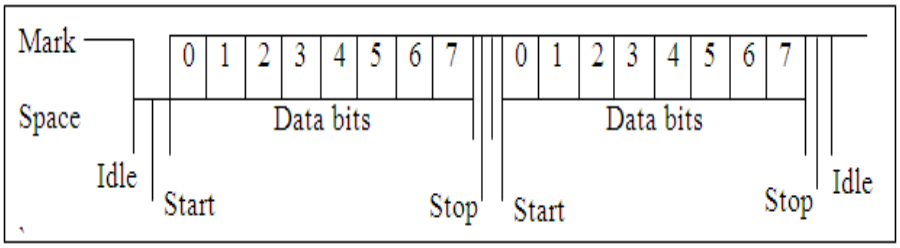
Mikrokontroler AVR ATMega 8 memiliki *Port* USART pada *Pin* 2 dan *Pin* 3 untuk melakukan komunikasi data antara mikrokontroler dengan mikrokontroler ataupun mikrokontroler dengan komputer. USART dapat

difungsikan sebagai transmisi data *sinkron*, dan *asinkron*. *Sinkron* berarti *clock* yang digunakan antara *transmiter* dan *receiver* satu sumber *clock*. Sedangkan *asinkron* berarti *transmiter dab receiver* mempunyai sumber *clock* sendiri-sendiri. USART terdiri dalm tiga blok yaitu *clock generator*, *transmiter*, dan *receiver*. Komunikasi ini memiliki kecepatan komunikasi yang rendah tetapi sangat mendukung untuk komunikasi jarak jauh. Pada setiap perangkat komputer bisanya sudah terdapat minimal satu *port* serial yang membuat komunikasi ini umum digunakan. Serial *port* pada komputer memungkinkan untuk melakukan komunikasi dua arah atau *full duplex* dimana dapat mengirim dan menerima data secara bersamaan. Terdapat beberapa perangkat lain, yang dapat melakukan komunikasi secara serial tetapi hanya satu arah saja atu *half duplex*.

Komunikasi serial memiliki beberapa parameter yang harus ditentukan yaitu:

1. *Baud Rate* kecepatan transmisi data.
2. *Start Bit* untuk memulai pengiriman data.
3. *Data Bit* data yang dikirim formatnya yang dipakai 8 bit.
4. *Parity bit* yang terdiri dari *odd* dan *even parity*, digunakan untuk *error cheking*.
5. *Stop Bit* untuk menandakan akhir dari satu pengiriman.

Format data yang digunakan pada komunikasi serial adalah 1 *start bit* (*low*), 8 *bit* data, 1 *stop bit* (*high*). Berikut Format Data Serial dapat dilihat pada gambar 2.13.



Gambar 2.13 Format Data Serial[10]

Standart pengaturan dalam komunikasi serial menggunakan *asinkronus start-stop*. *Standart* tersebut digunakan untuk mengatur

kecepatan data, banyaknya jumlah *bit* tiap karakternya, *parity*, dan jumlah banykanya *stop bit* tiap karakternya. Dalam perkembanganya serial komunikasi telah menggunakan USART sistem, maksud dari semua aturan yang ada sudah dikontrol berbasis perangkat lunak.

Serial *port* menggunakan dua *level* sinyal, sehingga data *rate* per detik akan sama dengan simbol *rate* per detik (*baud*). Antar perangkat yang dihuhbungkan harus memiliki kecepatan yang sama agar dapat berkomunikasi. Pada perangkat-perangkat baru yang memiliki kecepatan tinggi seperti komputer akan otomatis mengikuti standar kecepatan perangkat lain yang terhubung.

Standar jumlah *bit* dalam satu karakter dapat bervariasi mulai dari 5 hingga 9 *bit*, tetapi standar yang sering digunakan adalah 8 *bit* setiap karakternya. Seringkali teknik pengiriman data serial dimulai dari *bit* yang paling rendah terlebih dahulu, tetapi hal ini tidak bersifat mutlak. Teknik pengiriman dapat dirubah sebelum pengiriman data.

Parity merupakan cara untuk memonitor kesalahan yang terjadi saat pengiriman data. Ketika pengiriman suatu data menggunakan *parity*, maka pada setiap karakter akan ditambahkan satu *bit parity*. Terdapat lima macam *parity*, yaitu *none, mark, space, odd, even*. *None parity* berarti tidak ada *bit* untuk *parity*. *Mark* berarti *parity bit* dalam satu karakter selalu diset pada logika 1. *Space* merupakan 32. Kebalikan dari *mark*, yaitu *bit parity* selalu diset pada logika 0. Ketiga teknik di atas jarang digunakan, teknik yang sering digunakan untuk 8 *bit* data setiap karakternya adalah *even* dan *odd*. *Even parity* dan *odd parity* akan memeriksa setiap *bit* dalam karakter, khusunya *bit* bernilai 1, apabila jumlah *bit* 1 genap maka digunakan *even parity* dan juga sebaliknya.

Serial *port* akan menggunakan sinyal dalam suatu perangkat untuk memberhentikan dan memulai suatu pengiriman data. Oleh karena itu diperluakn suatu perjanjian antara perangkat yang berkomunikasi atau sering disebut dengan *handshake*. Kebanyakan perangkat menggunakan standar RS-232 dalam melakukan komunikasi. Selain itu terdapat standar lain yang dapat digunakan yaitu *Xon/Xoff*, maksudnya *Xon* akan dikirim

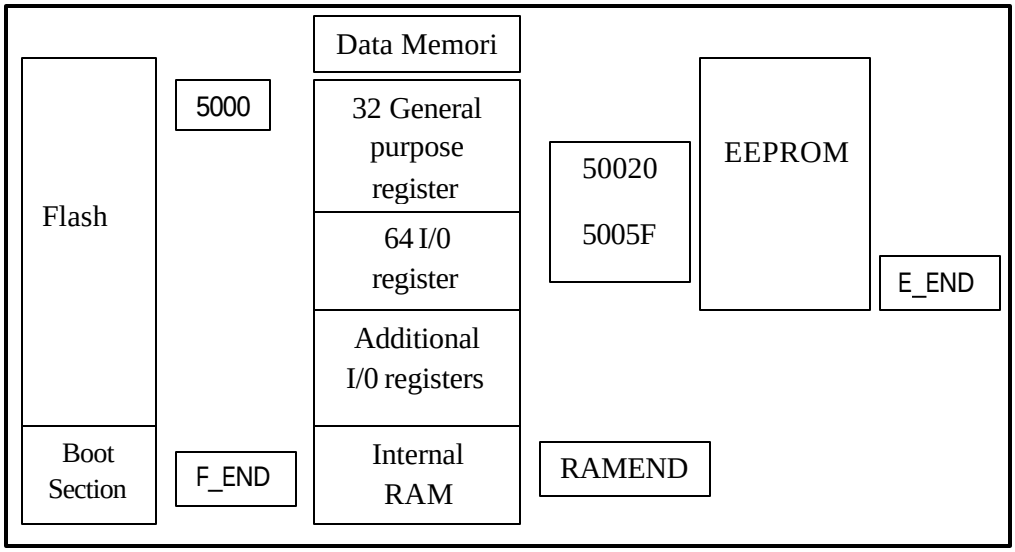
ketika penerima data siap untuk menerima dan *Xoff* akan dikirim ketika memori penerima penuh.

2.3.3. Fitur – fitur ATMega8[11]

Beberapa fitur dari ATmega8 adalah sebagai berikut:

1. 8 Kbyte *Flash* Program
2. 512 Kbyte EEPROM
3. 1 Kbyte SRAM
4. 2 *timer* 8 *bit* dan 1 *timer* 16 *bit*
5. Analog to digital *converter*
6. USART
7. Analog *comparator*
8. *Two wire interface* (I2C)

2.3.4. Peta Memori



Gambar 2.14 Peta Memori ATMega8[6]

Memori ATMega8 terbagi menjadi tiga buah yaitu:

1. Memori *Flash*

   Memori *flash* adalah memori ROM tempat kode-kode program berada. Kata *flash* menunjukan jenis ROM yang dapat ditulis dan dihapus secara elektrik. Memori *flash* terbagi menjadi dua bagian yaitu

bagian aplikasi dan bagian *boot*. Bagian aplikasi adalah bagian kode-kode program aplikasi berada. Bagian *boot* adalah bagian yang digunakan khusus untuk *booting* awal yang dapat diprogram untuk menulis bagian aplikasi tanpa melalui *programmer*/*downloader*, misalnya melalui USART.

2. Memori data

   Memori data adalah memori RAM yang digunakan untuk keperluan program. Memori data terbagi menjadi empat bagian yaitu 32 *Generaln Purphose Register* (GPR) adalah *register* khusus yang bertugas untuk membantu eksekusi program oleh *Arithmatich Logic Unit* (ALU) , dalam instruksi assembler setiap instruksi harus melibatkan GPR. Dalam bahasa C biasanya digunakan untuk variabel global atau nilai balik fungsi dan nilai-nilai yang dapat memperingan ALU. Dalam istilah *processor* komputer sehari-hari GPR dikenal sebagai "*chace memory*". I/O *register* dan *Aditional* I/O *register* adalah *register* yang difungsikan khusus untuk mengendalikan berbagai *pheripheral* dalam mikrokontroler seperti *pin port, timer*/*counter*, USART dan lain-lain. *Register* ini dalam keluarga mikrokontroler MCS51 dikenal sebagai *Special Function Register* (SFR).

3. EEPROM

   EEPROM adalah memori data yang dapat mengendap ketika *chip* mati (*off*), digunakan untuk keperluan penyimpanan data yang tahan terhadap gangguan catu daya.

2.3.5. Bahasa C++ [12]

Bahasa C++ diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories pada awal 1980an sebagai pengembangan dari bahasa C. Pada mulanya bahasa ini dikenal sebagai "C *with Classes*" (nama C++ digunakan sejak 1983, setelah diusulkan oleh Rick Mascitti). Pada tahun 1985 bahasa ini mulai disebarluaskan oleh AT&T dengan mengeluarkan perangkat lunak *cfront* yang berfungsi sebagai C++ translator (*cfront* menerima masukan program bahasa C++ dan menghasilkan kode bahasa C).

Perancangan bahasa C++ didasarkan pada bahasa C, simula67, algol68, dan ada. Sebagai contoh, konsep *class*" diambil dari bahasa simula67, konsep operator *overloading* dan kemungkinan penempatan deklarasi di antara instruksi diambil dari bahasa Algol68, konsep template dan *exception* diambil dari bahasa Ada. Bahasa C++ memperluas kemampuan bahasa C dalam beberapa hal yaitu:

1. Memberikan dukungan untuk menciptakan dan memanfaatkan abstraksi data
2. Memberikan dukungan untuk *object-oriented* programming
3. Memperbaiki beberapa kemampuan yang sudah ada pada bahasa C.

Upaya pembakuan terhadap bahasa C++ sudah dilakukan sejak beberapa tahun terakhir dan pada bulan Oktober - November 1998, ANSI telah mengeluarkan standard untuk bahasa C++.

A. *Kompatibilitas* antara C++ dan C [12]

Program yang dituliskan dalam bahasa C seharusnya dapat dikompilasi oleh kompilator C++. Namun demikian, ada beberapa hal yang harus diperhatikan

1. Program tidak dapat menggunakan kata kunci dari C++ sebagai nama *identivier*.
2. Dalam C++ deklarasi fungsi "f()" berarti bahwa, f tidak memiliki parameter formal satupun, dalam C ini berarti bahwa f dapat menerima parameter dari jenis apapun.
3. Dalam C++, tipe dari *konstanta* karakter adalah *char*, sedangkan dalam C, tipe tersebut adalah *int*. Akibatnya *sizeof* (`a') memberikan nilai 1 di C++ dan 4 di C pada mesin yang memiliki *representasi* integer 4 *byte*.
4. Setiap fungsi harus dideklarasikan (harus memiliki *prototype*).
5. Fungsi yang bukan bertipe *void*, harus memiliki instruksi return

```
char ch[3] = "\C++";          /* C: OK, \C++: error */
char ch[] = "\C++";           /* OK untuk C dan \C++ */
```

B. Saran untuk C programmers

Hindarilah penggunaan (#*define*) dalam program C++ seperti untuk mendefinisikan konstanta seperti pada

#*define* MAXBUFF 1500

Gunakanlah *const* untuk mendefinisikan konstanta.

1. Gunakan inline untuk menghindari *function-calling overhead*.
2. Deklarasikan setiap fungsi (*procedure*) dan spesifikasikan semua tipe parameter formalnya.
3. Jangan gunakan *malloc*() maupun *free*(). Sebagi gantinya, gunakanlah *new* dan *delete*. Kedua operator baru ini tidak hanya sekedar mengalokasikan memori melainkan juga secara otomatis melibatkan *constsructor* dan *destructor.*
4. *Union* pada umumnya tidak memerlukan *tag name*, gunakanlah *anonymous union*.

C. *Class*

Konsep kelas dalam C++ ditujukan untuk menciptakan tipe data baru. Sebuah tipe terdiri dari kumpulan bit yang merepresentasikan nilai abstrak dari instansiasi tipe tersebut serta kumpulan operasi terhadap tipe tersebut. Sebagai contoh *int* adalah sebuah tipe karena memiliki representasi bit dan kumpulan operasi seperti penambahan dua variabel bertipe *int*", \perkalian dua variabel bertipe *int*".

Dengan cara yang sama, sebuah kelas juga menyediakan sekumpulan operasi (biasanya *public*) dan sekumpulan data bit (biasanya *non-public*) yang menyatakan nilai abstrak objek dari kelas tersebut. Hubungan antara kelas dengan objek dapat dinyatakan dengan analogi berikut:

class vs. object = type vs. variable

Pendeklarasian kelas (terutama fungsi anggotanya) menentukan perilaku objek dalam operasi penciptaan, pemanipulasian, pemusnahan objek dari kelas tersebut. Dalam pemrograman dengan bahasa yang berorientasi objek dapat dilihat

adanya peran perancang kelas dan pengguna kelas. Perancang kelas menentukan representasi internal objek yang berasal kelas yang dirancangnya.

Pendeklarasian kelas dilakukan seperti pendenisian sebuah struktur namun dengan mengganti kata kunci *struct* dengan class. Kata kunci *class* dalam C++ dapat dipandang sebagai perluasan dari kata kunci *struct*, hanya perbedaannya nama kelas (*tag-name*) dalam C++ sekaligus merupakan tipe baru. [13]

D. WinAVR

Code vision AVR merupakan salah satu program bahas C yang berbasis Windows, keuntungan menggunakan code vision AVR lebih besar dibandingkan menggunakan program yang lain yang under DOS. Code vision AVR dalam pemrogramannya menggunakan bahasa C maupun bahasa C++. Namun dalam pembuatan Tugas Akhir penulis menggunakan code vision AVR untuk bahasa C.karena bahasa C sangat compatibel dengan mikrokontroler AVR terutama mikrokontroler ATMega 8

Adapun tiap tahap penyusunan perangkat lunak diantaranya adalah menyusun diagram alir (*flow chart*) program rangkaian kendali dan membuat perangkat lunak berdasarkan diagram alir yang telah disusun dengan menggunakan bahasa C. WinAVR digunakan untuk menuliskan *listing* program atau *coding*. Program yang sudah jadi kemudian disimpan dengan tipe *hex* yang kemudian akan diisikan ke mikrokontroler ATMega 8. Selain sebagai *compiler*, WinAVR digunakan untuk men-*download* program ke dalam mikrokontroler. Sebelum mikrokontroler diisi maka program sebelumnya harus di-*build* terlebih dahulu untuk mengetahui apakah terjadi kesalahan atau tidak. Kesalahan umumnya terdapat pada rutin dan subrutinnya yang tidak sesuai dengan alur pembacaan program. WinAVR akan menginformasikan titik dan letak terjadinya *error* sehingga mempermudah pembuat program dalam perbaikan. Jika tidak terjadi *error* maka WinAVR akan melakukan proses *flash* pada

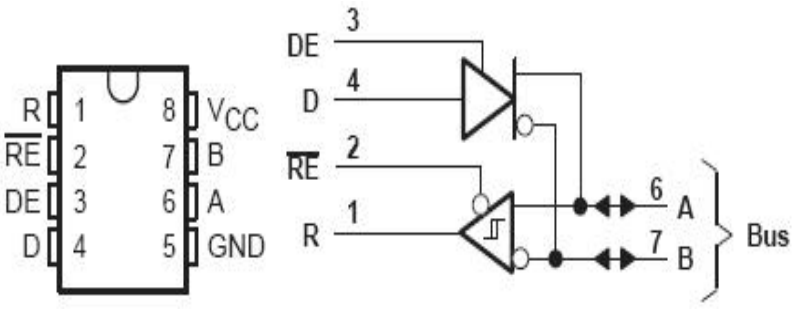
mikrokontroler untuk mulai mengisi program. Dengan demikian maka mikrokontroler akan mengeksekusi instruksi sesuai dengan program yang sudah ada. Selain itu WinAVR Studio juga dapat digunakan untuk melakukan pengaturan yang berhubungan dengan memori *flash,* EEPROM, dan penggunaan *oscillator* eksternal.

2.4. Rangkaian Terintegrasi

2.4.1. Komunikasi Serial RS485[6]

Komunikasi serial RS485 menggunakan sepasang kabel untuk mengirimkan satu sinyal. Tegangan antara kedua kabel saluran selalu berlawanan. Logika ditentukan dari beda tegangan antara kedua kabel tersebut.

SN75176 merupakan IC *multipoint* RS485 *transceiver.* Di dalam SN75176 terdapat sebuah *driver* dan *receiver* seperti pada Gambar 2.15.



Gambar 2.15 Bagan IC SN75176.

SN75176 dapat mendukung 32 unit paralel dalam satu jalur. Sensitivitas tegangan *input receiver* 0,2 V dan jarak maksimum 4000 feet. Pada mode pengiriman (*transmitting*), kaki *enable* kirim DE diberi logika 1. Keluaran A dan B ditentukan oleh masukan *driver* D, dimana keluaran A akan sesuai dengan logika *driver* D, sedangkan B berkebalikan. Jika *input* D berlogika 1, maka *output* A akan bertegangan 5 Volt dan *output* B 0 Volt. Sebaliknya jika *input* D berlogika 0 maka *output* A bertegangan 0 Volt dan *output* B 5 *Volt.*

Pada mode penerimaan (*receiving*), kaki *enable* terima RE diberi logika 0. *Output receiver* R ditentukan oleh tegangan diferensial antara *input* A dan B ($V_A - V_B$). Jika tegangan diferensial $V_A - V_B$ lebih besar dari +0,2 *volt*, maka *receiver* R akan berlogika1, sedangkan jika $V_A - V_B$ lebih kecil dari -0,2 *volt* maka *receiver* R akan berlogika 0. Untuk tegangan $V_A - V_B$ antara -0,2 *volt* sampai +0,2 *volt*, maka level logika tidak terdefinisi.

Mode pengiriman dan penerimaan data SN75176 ditunjukkan pada Tabel 2.2 dan 2.3 [6]

Tabel 2.2 Pengiriman Data (*Transmitting*)

INPUT	*ENABLE*	*OUTPUT*	
D	DE	A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

Tabel 2.3 Penerimaan Data (*Receiving*)

DIFFERENTIAL INPUTS (A-B)	*ENABLE* RE	*OUTPUT* R
V_{ID} = 0,2 V	L	H
-0,2V < V_{ID} < 0,2V	L	?
V_{ID} = -0,2V	L	L
X	H	Z
OPEN	L	?

Keterangan:

H = *High Level*

L = *Low Level*

x = *Irrelevant*

Z = *high impedance (off)*

? = Indeterminate

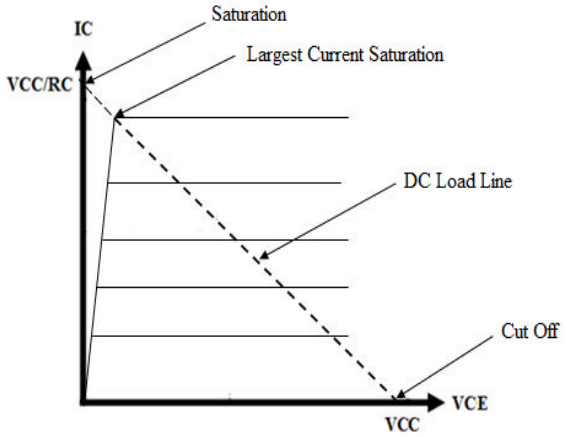
Jika terdapat gangguan listrik yang menimpa saluran transmisi, maka induksi tegangan gangguan akan diterima kedua kabel saluran sama besar. Karena *receiver* membandingkan selisih tegangan antara dua kabel saluran, maka induksi tegangan tidak akan berpengaruh pada *output.* Dengan kemampuan menangkal gangguan yang sangat baik ini, RS485 dapat dipakai untuk membangun saluran transmisi jarak jauh sampai 4000 feet dengan kecepatan tinggi.

2.5. Transistor Sebagai Saklar

Salah satu fungsi transistor adalah sebagai saklar yaitu bila berada pada dua daerah kerjanya yaitu daerah jenuh (saturasi) dan daerah mati (*cut-off*). Transistor akan mengalami perubahan kondisi dari menyumbat ke jenuh dan sebaliknya. Transistor dalam keadaan menyumbat dapat dianalogikan sebagai saklar dalam keadaan terbuka, sedangkan dalam keadaan jenuh seperti saklar yang menutup. Daerah kerja transistor pada keadaan jenuh adalah keadaan dimana transistor mengalirkan arus secara maksimum dari kolektor ke emitor sehingga transistor tersebut seolah-olah *short* pada hubungan kolektor–emitor. Pada daerah ini transistor dikatakan sebagai penghantar maksimum (sambungan CE terhubung maksimum). Untuk daerah aktif transistor merupakan daerah kerja yang biasanya digunakan sebagai penguat sinyal. Transistor dikatakan bekerja pada daerah aktif karena transistor selelu mengalirkan arus dari kolektor ke emitor walaupun tidak dalam proses penguatan sinyal, hal ini ditujukan untuk menghasilkan sinyal keluaran yang tidak cacat. Daerah aktif terletak antara daerah jenuh (saturasi) dan daerah mati (*Cut-off*). Daerah mati transistor atau daerah *cut-off* merupakan daerah kerja transistor dimana keadaan transistor menyumbat pada hubungan kolektor – emitor.

Daerah *cut-off* sering dinamakan sebagai daerah mati karena pada daerah kerja ini transistor tidak dapat mengalirkan arus dari kolektor ke emitor. Pada daerah *cut-off* transistor dapat di analogikan sebagai saklar terbuka pada hubungan kolektor–emitor. Untuk membuat transistor penghantar, pada masukan basis perlu diberi tegangan. Besarnya tegangan harus lebih besar dari V_{BE} (0,3 untuk *germanium* dan 0,7 untuk *silicon*). Dengan mengatur Ib > Ic/ß kondisi transistor

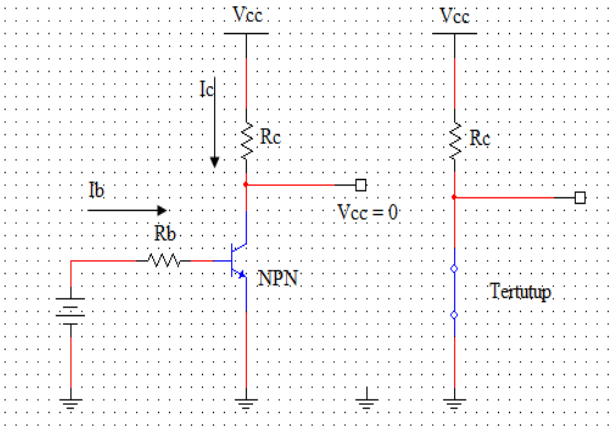
akan menjadi jenuh seakan kolektor dan emitor *short circuit.* Arus mengalir dari kolektor ke emitor tanpa hambatan dan Vce˜0. Besar arus yang mengalir dari kolektor ke emitor sama dengan Vcc/Rc. Keadaan seperti ini menyerupai saklar dalam kondisi tertutup (ON). Grafik hubungan daerah jenuh, daerah aktif dan daerah mati pada transistor dapat ditunjukkan pada gambar 2.16. Dari grafik tersebut dapat diketahui kondisi daerah transistor pada saat saturasi adalah pada saat transistor mengalirkan arus secara maksimum. Sedangkan untuk kondisi *cut off* merupakan kondisi sebaliknya yaitu karena transistor tidak dapat melewatkan arus. Untuk daerah aktif berada diantara daerah jenuh dan daerah *cut off* [7,10,11].



Gambar 2.16 Grafik Garis Beban DC

2.5.1. **Transistor Pada Kondisi Jenuh**

Untuk kondisi transistor pada kondisi jenuh atau saturasi ditunjukkan pada gambar 2.17.



Gambar 2.17 Transistor Pada Kondisi Jenuh

Untuk kondisi transistor pada keadaan jenuh atau saturasi maka besarnya tegangan kolektor-emitor VCE suatu transistor pada konfigurasi pada gambar 2.17 dapat diketahui sebagai berikut [11].

$$Vce = Vcc - Ic.Rc \quad \text{..... (2.1)}$$

Karena kondisi jenuh Vce = 0 Volt (transistor ideal) maka besarnya arus kolektor (Ic) adalah

$$Ic = \frac{Vcc}{Rc} \quad \text{.....(2.2)}$$

Besarnya arus yang mengalir agar transistor menjadi jenuh (saturasi) adalah

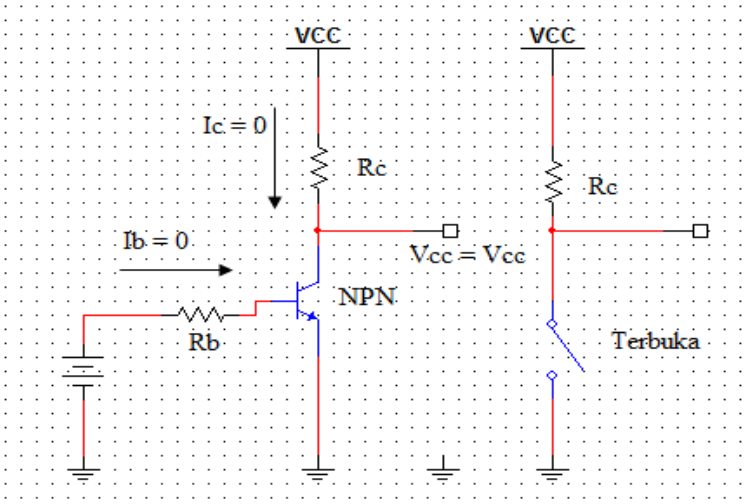
$$Rb - \frac{Vt - Vbe}{Ib} \quad \text{.....(2.3)}$$

Sehingga besarnya arus basis (Ib) pada kondisi saturasi adalah

$$Ib = \frac{Ic}{\beta} \quad \text{.....(2.4)}$$

2.5.2 Transistor Pada Kondisi Mati

Untuk kondisi transistor pada kondisi mati atau *cut off* ditunjukkan pada gambar 2.18.



Gambar 2.18 Transistor Pada Kondisi *Cut-off*

Dengan mengatur Ib = 0 atau tidak memberi tegangan pada bias basis atau basis diberi tegangan mundur terhadap emitor maka transistor akan dalam kondisi mati (*cut-off*), sehingga tak ada arus mengalir dari kolektor ke emitor (Ic˜ 0) dan Vce ˜ Vcc. Keadaan ini menyerupai saklar pada kondisi terbuka seperti ditunjukan pada gambar diatas.

Besarnya tegangan antara kolektor dan emitor transistor pada kondisi mati atau *cut off* adalah [11]

$$Vce = Vcc - Ic.Rc \quad \text{.....(2.1)}$$

Karena kondisi mati Ic=0 (transistor ideal) maka :

$$Vce = Vcc \quad \text{.....(2.5)}$$

Dengan demikian maka besar arus basis Ib adalah

$$Ib = \frac{Ic}{\beta} \quad \text{.....(2.4)}$$

$$Ib = 0$$