

BAB II

DASAR TEORI

2.1 Mikropengendali AVR[2,3]

Kata AVR merupakan singkatan dari *Alf and Vegard RISC (Reduced Instruction Set Computer)* sesuai dengan nama penggagasnya. Mikropengendali AVR yang menggunakan teknologi RISC dan menggunakan arsitektur Harvard ini pertama kali dikembangkan pada tahun 1996 oleh dua orang mahasiswa Norwegian Institute of Technology yaitu Alf Egil Bogen dan Vegard Wollan yang kemudian dikembangkan lebih lanjut oleh perusahaan Atmel. Seri pertama mikropengendali AVR yang dikeluarkan adalah mikropengendali 8 bit dengan nama AT90S8515 dengan konfigurasi pin yang sama dengan mikropengendali 8051, termasuk bus alamat dan bus data yang termultipleks. Mikropengendali AVR mempunyai set instruksi yang lebih sedikit dan mode pengalamatannya yang juga sederhana. Dalam AVR RISC 8 bit, semua instruksi berukuran 16 bit dan sebagian besar dieksekusi dalam satu siklus *clock* kecuali instruksi percabangan yang membutuhkan dua siklus *clock*.

Berbeda dengan mikropengendali MCS dengan menggunakan teknologi *Complex Instruction Set Computing* (CISC) yang instruksinya bervariasi antara 8 bit sampai 32 bit dan dieksekusi selama satu sampai empat siklus mesin, dimana satu siklus mesin membutuhkan 12 periode *clock*. Beberapa fitur yang dimiliki mikropengendali dengan arsitektur RISC diantaranya adalah instruksi yang sederhana, set instruksi yang sedikit, panjang instruksi yang sama untuk semua instruksi, register untuk penyimpanan *operand* yang jumlahnya besar, mempunyai arsitektur *load/store*, dan eksekusi instruksi yang lebih cepat sehingga memberikan kecepatan siklus instruksi rata-rata satu *clock* per instruksi. Dalam perkembangannya, AVR dibagi menjadi enam kelas yaitu kelompok ATtiny, AT90Sxx, ATmega, AT90CAN, AT90PWM, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah kapasitas memori, peripheral, fungsinya dan beberapa fitur tambahannya. Sedangkan dari segi arsitektur dan set instruksi yang digunakan hampir sama. Pengelompokan jenis mikropengendali AVR sesuai varian dan tipe mikropengendalinya ditunjukkan oleh tabel 2.1 [2].

Tabel 2.1 Jenis Varian Mikropengendali AVR

No	Varian	Tipe Mikropengendali
1	ATtiny	ATTINY10, ATTINY11, ATTINY12, ATTINY13, ATTINY15, ATTINY2313, ATTINY24, ATTINY25, ATTINY261, ATTINY44, ATTINY 45, ATTINY461, ATTINY 48, ATTINY48_32PIN, ATTINY84, ATTINY85, ATTINY861, ATTINY88, ATTINY88_32PIN.
2	AT90S	AT90S1200, AT90S2313, AT90S2323, AT90S2333, AT90S2343, AT90S4433, AT90S4434, AT90S8515, AT90S8535.
3	AT90USB	AT90USB646, AT90USB1286.
4	ATMega	ATMega103, ATMega128, ATMega1280, ATMega1281, ATMega1284P, ATMega16, ATMega162, ATMega164P, ATMega165, ATMega165P, ATMega168, ATMega168-32PIN, ATMega168P, ATMega168P_32PIN, ATMega169, ATMega169P, ATMega2560, ATMega2561, ATMega32, ATMega324P, ATMega325, ATMega325P, ATMega3250, ATMega3250P, ATMega328P, ATMega328P, ATMega328P_32PIN, ATMega329, ATMega3290, ATMega3290P, ATMega48, ATMega48P, ATMega48P_32PIN, ATMega64, ATMega640, ATMega644, ATMega644P, ATMega645, ATMega6450, ATMega649, ATMega6490, ATMega8, ATMega8515, ATMega8535, ATMega88, ATMega88_32PIN, ATMega88P.

2.2 Mikropengendali ATmega 128 [3,14]

2.2.1 Arsitektur ATmega 128

Mikropengendali ATmega 128 merupakan mikropengendali 8-bit teknologi CMOS dengan konsumsi daya rendah yang berbasis arsitektur enhanced RISC AVR.. Arsitektur yang dihasilkan adalah arsitektur yang kode operasinya lebih efisien serta pencapaian *throughput*-nya hingga sepuluh kali lebih cepat daripada mikropengendali *Complex Instruction Set Computer* (CISC) konvensional. Beberapa fitur utama yang tersedia pada ATmega 128 diantaranya adalah :

1. *Port I/O* sebanyak 53 bit, yang dikelompokkan dalam *Port A*, *Port B*, *Port C*, *Port D*, *Port E*, *Port F* dan *Port G*.
2. *Analog to Digital Converter* 10 bit sebanyak 8 *input*.
3. *Timer/Counter* sebanyak 3 buah.
4. CPU 8 bit yang terdiri dari 32 *register*.
5. *Watchdog Timer* dengan *oscillator internal*.
6. SRAM sebesar 4 Kbyte.
7. Memori *Flash* sebesar 128 Kbyte dengan kemampuan *read while write*.
8. *Interrupt* internal maupun eksternal.

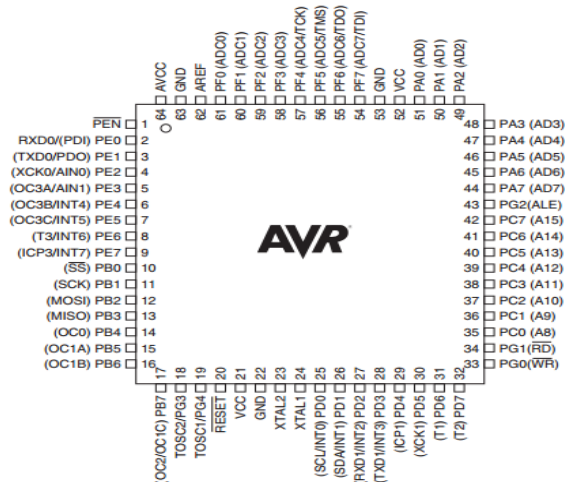
9. *Port* komunikasi SPI dan Analog komparator.
10. EEPROM sebesar 4 Kbyte yang dapat diprogram saat operasi.
11. Komunikasi serial standar USART dengan kecepatan maksimal 2,5 Mbps.
12. Frekuensi *clock* maksimum 16 MHz
13. Dua Kanal 8-bit PWM dan 6 Kanal PWM dengan program resolusi dari 2-16 bits.

2.2.2 Konfigurasi Pin ATmega 128[14]

Mikropengendali ATmega 128 mempunyai pin sebanyak 53 dengan *port* I/O yang dikelompokkan dalam empat *port* yaitu *Port A*, *Port B*, *Port C*, *Port D*, *Port E*, *Port F* dan *Port G*. Masing-masing *port* tersebut harus diatur jika difungsikan sebagai *input* atau *output* karena *port-port* tersebut dapat digunakan untuk jalur *input* maupun *output*. Semua *port* yang ada bisa digunakan sebagai masukan atau keluaran. Mikropengendali AVR mempunyai set instruksi yang lebih sedikit dan mode pengalamatannya yang juga sederhana. Dalam AVR RISC 8 bit, semua instruksi berukuran 16 bit dan sebagian besar dieksekusi dalam satu siklus *clock* kecuali instruksi percabangan yang membutuhkan

dua siklus *clock*. Tipe data mikropengendali AVR hanya mempunyai satu tipe data yaitu 8-bit dan ukuran setiap register juga 8-bit. Pada dasarnya ada empat cara untuk merepresentasikan sebuah byte data yang dinyatakan dalam format hex, biner, decimal dan ASCII. Jika dilihat lebih detail pada bagian pemroses mikropengendali, terdapat unit CPU utama untuk memastikan eksekusi program. CPU juga bisa mengakses memori, melakukan kalkulasi, pengontrolan dan penanganan interupsi dengan menggunakan arsitektur Harvard (bus untuk memori, program, dan data terpisah), sehingga dihasilkan performa yang tinggi. Hal ini dikarenakan instruksi pada memori program dieksekusi dengan *single level pipe lining*, dengan demikian pada saat sebuah instruksi dieksekusi, instruksi berikutnya dapat diakses dari memori program. Konsep ini memungkinkan instruksi-instruksi dieksekusi pada setiap siklus *clock*. Pin I/O pada mikropengendali AVR dapat dikonfigurasi masukan atau keluaran dengan cara mengubah isis I/O register DDR (*Data Direction Register*). Selain itu beberapa *port* pada ATmega 128 memiliki fungsi khusus. Konfigurasi Pin mikropengendali AVR ATmega 128 ditunjukkan

pada gambar 2.1 sedangkan untuk konfigurasi pin ditunjukkan pada tabel 2.2.



Gambar 2.1 Konfigurasi Pin ATMega 128

Berikut ini penjelasan umum mengenai deskripsi pin dari Atmega 128 :

- VCC merupakan pin masukan positif catu daya
- GND sebagai pin *ground*
- Port A (PA0...PA7) merupakan 8 bit port I/O dua arah dan pin fungsi khusus sebagai *interface* memori eksternal.

- Port B (PB0...PB7) merupakan 8 bit port I/O dua arah dan pin fungsi khusus yaitu *counter/timer*, dan SPI.
- Port C (PC0...PC7) merupakan 8 bit port I/O dua arah dan pin fungsi khusus yaitu sebagai *interface* memori eksternal.
- Port D (PD0...PD7) merupakan 8 bit port I/O dua arah dan pin fungsi khusus yaitu *timer/counter*, interupsi internal, dan UART.
- Port E (PE0...PE7) merupakan 8 bit port I/O dua arah dan fungsi khusus yaitu eksternal interrupt, UART, dan analog komparator.
- Port F (PF0...PF7) merupakan 8 bit port I/O pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC.
- Port G (PG0...PG4) merupakan 5 bit port I/O pin I/O dua arah.
- XTAL 1 dan XTAL 2 sebagai pin masukan clock eksternal.
- AVCC sebagai tegangan masukan untuk port F dan *A/D Converter*.

- AREF sebagai pin tegangan referensi *A/D Converter*.
- PEN merupakan pin untuk menghidupkan program untuk mode SPI serial programming. PEN tidak mempunyai fungsi selama pengoprasian normal.

Tabel 2.2 Deskripsi Pin ATMEGA8535

Nama Port	Keterangan
Port A : PA0-PA7	<i>Port I/O dua arah dilengkapi internal pull-up resistor. Fungsi khusus untuk pengalamatan byte rendah dan untuk interface memori eksternal</i>
Port B : PB0-PB7	<p><i>Port I/O dua arah dilengkapi internal pull-up resistor. Fungsi khusus:</i></p> <ul style="list-style-type: none"> • PB0 : SS (<i>SPI slave select input</i>) • PB1 : SCK (<i>SPI bus serial clock</i>) • PB2 : MOSI (<i>SPI bus master output/slave input</i>) • PB3 : MISO (<i>SPI bus master input/slave output</i>) • PB4 : OC0 (<i>Output compare dan PWM output timer/counter0</i>) • PB5 : OC1A (<i>Output compare dan PWM output A timer/counter1</i>) • PB6 : OC1B (<i>Output compare dan PWM output B timer/counter1</i>) • PB7 : OC2/OC1C (<i>Output compare dan PWM output C timer/counter2</i>)

Nama Port	Keterangan
Port C : PC0-PC7	<i>Port I/O dua arah dilengkapi internal pull-up resistor. Fungsi khusus untuk pengalamatan byte tinggi dan untuk interface memori eksternal.</i>
Port D : PD0-PD7	<p><i>Port I/O dua arah dilengkapi internal pull-up resistor.</i></p> <p>Fungsi khusus:</p> <ul style="list-style-type: none"> • PD0 : INT1/SCL (<i>External interrupt 0 input dan TWI Serial Clock</i>) • PD1 : INT1/SDA (<i>External interrupt 1 input dan TWI Serial Data</i>) • PD2 : INT2/RXD1 (<i>External interrupt 2 input dan UART1Receive Pin</i>) • PD3 : INT3/TXD1 (<i>External interrupt 3 input dan UART1Transmit Pin</i>) • PD4: ICP1 (<i>Timer/Counter Input Capture Pin</i>) • PD5 : XCK1 (<i>USART1 External Clock Input/Output</i>) • PD6 : OC1A (<i>timer/counter 1 output compare A match output</i>) • PD6 : T1 (<i>Timer/Counter1 Clock Input</i>) • PD7 : T2 (<i>Timer/Counter2 Clock Input</i>)
Port E : PE0-PE7	<p><i>Port I/O dua arah dilengkapi internal pull-up resistor.</i></p> <p>Fungsi khusus:</p> <ul style="list-style-type: none"> • PE0 : PDI/RXD0 (<i>Programming Data Input or UART0 Receive Pin</i>) • PE1 : PDO/TXD0 (<i>Programming Data Output or UART0 Transmit Pin</i>) • PE2 : AIN0/XCK0 (<i>Analog Comparator Positive Input or USART0 external clock input/output</i>)

Nama Port	Keterangan
	<ul style="list-style-type: none"> • PE3 : AIN1/OC3A (<i>Analog Comparator Negative Input or Output Compare and P WM Output A for Timer/Counter3</i>) • PE4 : INT4/OC3B (<i>External Interrupt4 Input or Output Compare and P WM Output B forTimer/Counter3</i>) • PE5: INT5/OC3C (<i>External Interrupt 5 Input or Output Compare and P WM Output C for Timer/Counter3</i>) • PE6 : INT6/ T3 (<i>External Interrupt 6 Input or Timer/Counter3 Clock Input</i>) • PE7 : INT7/ICP3 (<i>External Interrupt 7 Input or Timer/Counter3 Input Capture Pin</i>)
Port F : PF0-PF7	<p>Port I/O dua arah dilengkapi internall pull-up resistor. Port ini juga dimultipleks dengan masukan analog ke ADC 8 kanal. Fungsi khusus :</p> <ul style="list-style-type: none"> • PF0 : ADC0 (<i>ADC input channel 0</i>) • PF1 : ADC1 (<i>ADC input channel 1</i>) • PF2 : ADC2 (<i>ADC input channel 2</i>) • PF3: ADC3 (<i>ADC input channel 3</i>) • PF4: ADC4/TCK (<i>ADC input channel 4 or JTAG Test Clock</i>) • PF5: ADC5/TMS(<i>ADC input channel 5 or JTAG Test Mode Select</i>) • PF6: ADC6/TDO (<i>ADC input channel 6 or JTAG Test Data Output</i>) • PF7: ADC7/TDI (<i>ADC input channel 7 or JTAG Test Data Input</i>)

Nama Port	Keterangan
Port G : PG0-PG4	<p>Port I/O dua arah dilengkapi <i>internal pull-up resistor</i>.</p> <p>Fungsi khusus:</p> <ul style="list-style-type: none"> • PG0 : WR (<i>Write strobe to external memory</i>) • PG1 : RD (<i>Read strobe to external memory</i>) • PG2 : ALE (<i>Address Latch Enable to external memory</i>) • PG3: TOSC2 (<i>RTC Oscillator Timer/Counter0</i>) • PG4: TOSC1 (<i>RTC Oscillator Timer/Counter0</i>)
RESET	Memasukkan <i>reset</i> . Sebuah <i>reset</i> terjadi jika sebuah pin ini diberi logika <i>low</i> melebihi periode minimum yang diperlukan.
VCC	Catu Daya
GND	Ground
XTAL1	Keluaran dari <i>inverting oscillator amplifier</i>
XTAL2	Masukan ke <i>inverting oscillator amplifier</i> dan masukan ke rangkaian internal <i>clock</i>
AVCC	Catu daya untuk <i>port A</i> dan ADC
AGND	Analog Ground
AREF	Referensi masukan analog untuk ADC

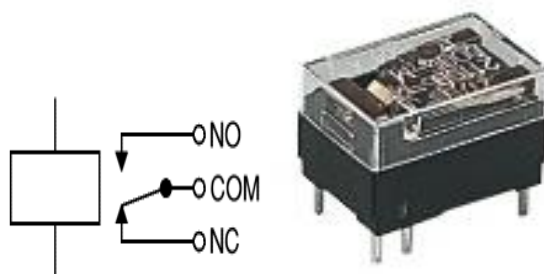
2.3 RELAY [4]

Relay merupakan suatu tombol elektrik yang bekerja berdasarkan prinsip elektro magnet untuk menutup atau membuka kontak saklar secara otomatis. *Relay* terdiri dari

suatu lilitan dan *switch* mekanik. *Switch* mekanik akan bergerak jika ada arus listrik yang mengalir melalui lilitan. Susunan kontak *relay* sebagai berikut :

1. *Normally switch (NC)* : COM akan terhubung apabila kumparan *relay off*.
2. *Normally Open (NO)* : COM akan terhubung apabila kumparan relay diberi arus listrik.
3. *Change Over Relay* adalah kutub acuan (COM), memiliki kontak tengah yang akan melepaskan diri dan membuat kontak lainnya terhubung.

Pada saat lilitan disuplai tegangan, maka arus akan mengalir pada kumparan, sehingga pada inti besi yang dililit oleh kumparan akan timbul medan magnet. Karena inti besi bersifat magnetis, maka jangkar akan tertarik ke inti besi sehingga mengaktifkan kontak *relay*. *Symbol* dan bentuk fisik dari *relay* ditunjukkan pada gambar 2.2



Gambar 2.2 Simbol dan Bentuk Fisik *Relay*

Berdasarkan jumlah kutub pada *relay*, maka *relay* dibedakan menjadi 4 jenis. *Pole* adalah jumlah common dan throw adalah jumlah terminal *output* (NO dan NC) dari *relay*. Untuk jenis relay diantaranya yaitu SPST atau *single pole single throw*, SPDT atau *single pole double throw*, DPST atau *double pole single throw* dan DPDT atau *double pole double throw*.

2.4 MOTOR *DIRRECT CURRENT* (DC) [3,5]

Motor bekerja berdasarkan prinsip induksi magnetik. Sirkuit internal motor DC terdiri dari kumparan/lilitan konduktor. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Konduktor dibentuk menjadi sebuah loop sehingga ada dua bagian konduktor yang berada di dalam medan magnet pada saat yang sama. Konfigurasi konduktor seperti ini akan menghasilkan distorsi pada medan magnet utama dan menghasilkan gaya dorong pada masing-masing konduktor. Pada saat konduktor ditempatkan pada rotor, gaya dorong yang timbul akan menyebabkan rotor berputar searah jarum jam. Kontruksi dasar motor DC dapat dilihat pada gambar 2.3. Arah garis gaya medan magnet yang dihasilkan oleh magnet permanen adalah dari kutub utara menuju ke selatan. Sementara pada konduktor yang

dekat dengan kutub, arah garis gaya magnet disisi sebelah bawah searah dengan garis gaya magnet permanen sedangkan disisi sebelah atas arah garis gaya magnet berlawanan arah dengan garis gaya magnet permanen. Ini menyebabkan medan magnet di sisi sebelah bawah lebih rapat daripada sisi sebelah atas. Dengan demikian konduktor akan terdorong ke arah atas. Sementara pada konduktor yang dekat dengan kutub utara, arah garis gaya magnet permanen. Ini menyebabkan medan magnet disisi sebelah atas lebih rapat daripada sisi sebelah bawah. Pada akhirnya konduktor akan membentuk gerakan berputar berlawanan dengan jarum jam.

2.4.1 Pengontrolan Motor DC[3]

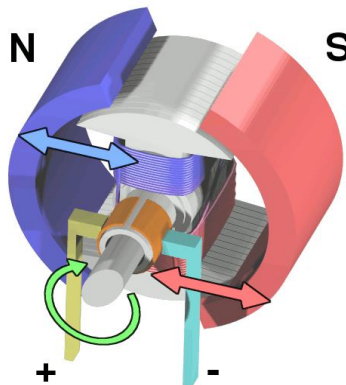
Pada dasarnya ada beberapa cara yang dapat digunakan untuk mengontrol motor DC berdasarkan pemberian pemiccuan dan karena itu berkenaan dengan jenis driver yang harus diberikan pada motor DC tersebut. Di sini akan dibahas dua macam *driver* yang bisa diterapkan berdasarkan kebutuhan. Pertama adalah dengan cara memberikan pengontrolan pada motor DC dengan memberikan *driver* berupa transistor unggal dari jenis NPN. Jenis *driver* ini hanya mengatur kecepatan motor DC saja tanpa bisa mengatur arah putarannya (searah jarum

jam/CW atau berlawanan arah jarum jam/CCW). Umumnya catu daya motor DC adalah sebesar 12V, sedangkan *output* PWM dari mikropengendali maksimal sebesar 5V. Karena itu gunakanlah transistor sebagai penguat tegangan dan sekaligus sebagai *buffer* terhadap mikropengendali. Jenis driver yang kedua adalah metode jembatan H (*H-bridge*). Jenis driver ini digunakan untuk mengontrol baik kecepatan maupu arah motor DC tersebut. *Driver H-bridge* dibangun dengan empat buah transistor jenis TIP120. Sedangkan untuk pengaturan kecepatannya dapat menghubungkan *output* PWM ke kaki basis transistor 1 untuk putaran searah jarum jam. Dan untuk putaran yang dua berlawanan arah jarum jam, *output* PWM dapat dihubungkan ke kaki basis transistor 2.

2.4.2 Pengontrolan Dengan Teknik PWM [3]

Pengontrolan dengan teknik PWM (*Pulse Width Modulation*) adalah sebuah cara memanipulasi lebar pulsa dalam perioda yang konstan untuk mendapatkan tegangan rata-rata yang berbeda. Sebagai contoh jika suply tegangan sebesar 5V dan *duty cycle* 25%, maka akan dihasilkan tegangan 1,25%. Kecepatan motor tergantung pada tiga faktor yaitu beban motor, tegangan listrik pada motor dan arus listrik pada motor. Untuk motor dengan beban yang

tetap/konstan, maka kecepatan motor dapat dipertahankan dengan menggunakan *Pulse Width Modulation* (PWM). Dengan melakukan perubahan (modulasi) lebar pulsa terhadap motor yang digunakan, maka dapat diatur untuk menaikkan atau menurunkan kecepatan motor. Semakin lebar pulsa yang diberikan pada motor maka kecepatan motor juga semakin besar.

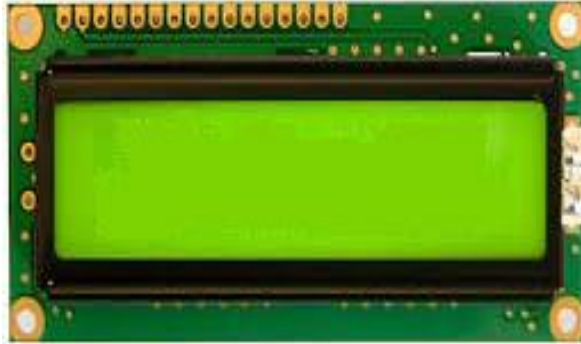


Gambar 2.3 *Motor DC*

2.5 **LIQUID CRYSTAL DISPLAY (LCD)**[6,7]

Liquid Crystal Display (LCD) adalah komponen yang dapat menampilkan tulisan. Salah satu jenisnya memiliki dua baris dengan setiap baris terdiri atas enam belas karakter dan biasa disebut LCD 16x2. LCD 16x2 bearti mempunyai ukuran lebar *display* 2 baris dan mempunyai 16

kolom atau biasa disebut dengan LCD karakter 16x2. Pada gambar 2.4 menunjukkan tampilan dari LCD 16x2.



Gambar 2.4 LCD 16x2

LCD karakter 16x2 tersebut memiliki 16 pin dengan fungsi masing-masing pin yang berbeda. Pemakaian pin pada LCD tergantung dengan kebutuhan pin yang digunakan. Pin-pin pada LCD diantaranya berupa pin power, pin data *input* dan *output*. Pada tabel 2.3 adalah tabel pin dan keterangan pada LCD.

Tabel 2.3 Pin LCD dan Keterangan

PIN	Nama PIN	I/O	Keterangan
1	VSS	Power	Ground voltage (0 V)
2	VCC	Power	+ 5 V
3	VEE	Power	Pengatur kontras

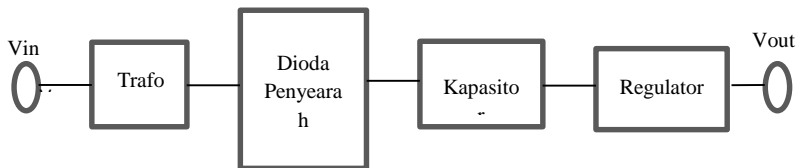
PIN	Nama PIN	I/O	Keterangan
4	RS	<i>Input</i>	<i>Register select</i> <ul style="list-style-type: none"> • RS= HIGH : untuk mengirim data • RS=LOW: untuk mengirim instruksi
5	R / W	<i>Input</i>	<i>Read / Write control bus</i> <ul style="list-style-type: none"> • R/W = HIGH : Mode untuk membaca data pada LCD • R/W = LOW : Mode untuk penulisan <i>data</i> pada LCD
6	E	<i>Input</i>	<i>Data Enable</i> untuk mengontrol ke LCD ketika bernilai LOW, LCD tidak dapat diakses
7	DB0	<i>Data Bus</i>	<i>Data 0</i>
8	DB1	<i>Data Bus</i>	<i>Data 1</i>
9	DB2	<i>Data Bus</i>	<i>Data 2</i>
10	DB3	<i>Data Bus</i>	<i>Data 3</i>
11	DB4	<i>Data Bus</i>	<i>Data 4</i>
12	DB5	<i>Data Bus</i>	<i>Data 5</i>
13	DB6	<i>Data Bus</i>	<i>Data 6</i>
14	DB7	<i>Data Bus</i>	<i>Data 7</i>
15	BLA	<i>Power</i>	Catu Daya Layar , positif
16	BLK	<i>Power</i>	Catu Daya Layar , negatif

Dari tabel 2.3 dapat dijelaskan *display* pada karakter LCD akan diatur oleh pin *EN*, *RS* dan *RW*. Jalur *EN* dinamakan *Enable*. Jalur ini digunakan untuk memberitahu *LCD* bahwa data sedang dikirimkan. Untuk mengirimkan data ke *LCD* maka Program *EN* harus berlogika 0 *active low* dan set pada jalur yang lain *RS* serta *RW*. Jalur *RS* adalah jalur *register select*. Ketika *RS* berlogika *low* “0” data akan dianggap sebagai sebuah perintah atau instruksi khusus. Sedangkan ketika berlogika *high* “1” data akan dikirim sebagai data *text* yang ditampilkan pada *LCD*. Jalur *RW* adalah jalur kontrol *read/write*. Ketika *RW* berlogika *low* “0” maka informasi pada *bus* data akan ditampilkan pada *LCD*. Ketika *RW* berlogika *high* “1” maka program akan melakukan pembacaan memori pada *LCD*. Sedangkan untuk jalur *bus* terdiri dari 4 atau 8 jalur bergantung pada *mode* operasi yang dipilih oleh *user*.

2.6 CATU DAYA[8]

Catu daya merupakan bagian terpenting pada sebuah rangkaian elektronika karena catu daya merupakan sumber energi dari sebuah rangkaian. Pada sebuah catu daya terdapat dua buah sumber tegangan yaitu DC (*Direct Current*) dan AC (*Alternating Current*). Sedangkan dalam kebiasaan sehari-hari banyak menggunakan arus

AC, maka dari itu diperlukan *power supply* untuk dapat mengubah sumber tegangan AC menjadi DC. *Power supply* sendiri merupakan kumpulan dari beberapa perangkat elektronika diantaranya adalah *trafo*, penyearah (*rectifier*), *filter* dan regulator. *Power supply* memperoleh sumber tegangan dari PLN sebesar 220 VAC yang kemudian diturunkan menjadi 12 VAC dengan menggunakan *trafo step down*. Tegangan 12 VAC lalu disearahkan dengan menggunakan *dioda bridge* sehingga menghasilkan tegangan DC keluaran dari *diode bridge* ini masuk ke dalam IC regulator yang berfungsi untuk menstabilkan tegangan. Pada gambar 2.5 merupakan blok diagram rangkaian catu daya dan cara kerja dari catu daya tersebut.



Gambar 2.5 Blok diagram rangkaian catu daya

Gambar 2.5 menjelaskan fungsi dari masing-masing blok, antara lain adalah sebuah trafo, dioda penyearah, filter dan regulator. Pada Blok diagram *power supply* diatas dapat dijelaskan sebagai berikut:

a) *Transformator*

Transformator merupakan suatu peralatan listrik elektromagnetik statis yang berfungsi untuk memindahkan dan mengubah daya listrik dari suatu rangkaian listrik ke rangkaian listrik lainnya, dengan frekuensi yang sama dan perbandingan transformasi tertentu melalui suatu magnet dan bekerja berdasarkan prinsip *induksi elektromagnetis*, dimana perbandingan tegangan antara sisi primer dan sisi sekunder berbanding lurus dengan perbandingan jumlah lilitan dan berbanding terbalik dengan perbandingan arusnya. Pada dasarnya jenis Transformator dibedakan menjadi 2 jenis yaitu *Trafo Step Up* dan *Trafo Step Down*. *Trafo Step Up* digunakan untuk menaikkan tegangan listrik sedangkan *Trafo Step Down* digunakan untuk menurunkan tegangan listrik. Selain itu *Trafo Step Up* memiliki ciri-ciri lilitan kumparan primer lebih sedikit dari pada lilitan kumparan sekunder dengan tegangan primer lebih kecil dari tegangan sekunder. Sedangkan ciri-ciri yang dimiliki oleh *Trafo Step Down* adalah lilitan kumparan primer lebih banyak daripada lilitan kumparan sekunder, kemudian

tegangan primer lebih tinggi dari tegangan sekunder.

b) *Rectifier*

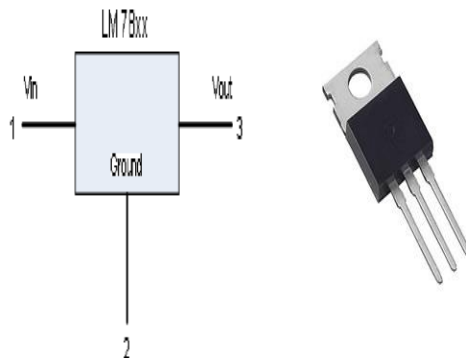
Rectifier merupakan komponen yang digunakan untuk mengubah sumber arus bolak-balik (AC) menjadi sumber arus searah (DC). Rangkaian *rectifier* banyak menggunakan *transformator step down* yang digunakan untuk menurunkan tegangan sesuai dengan perbandingan *transformator* yang digunakan.

c) *Filter*

Filter atau penyaring digunakan untuk mengurangi tegangan kerut (*ripple voltage*) sehingga dapat diperoleh tegangan keluaran yang lebih rata, baik untuk penyearah gelombang setengah maupun gelombang penuh. Filter diperlukan karena rangkaian elektronik memerlukan sumber tegangan DC yang tetap, baik untuk keperluan sumber daya dan pembiasan yang sesuai operasi rangkaian. Ada dua komponen yang umum digunakan sebagai rangkaian *filter*, yaitu induktor dan kapasitor.

d) *IC Regulator*

Regulator digunakan sebagai penstabil untuk memberikan tegangan keluaran yang konstan walaupun terdapat *fluktuasi* baik arus beban maupun tegangan input sumber. *IC Regulator* yang digunakan adalah LM 7805 untuk menghasilkan output tegangan sebesar 5 Volt. IC regulator ini akan menstabilkan tegangan DC. Selain IC regulator 7805 juga terdapat IC regulator yang lainnya missal LM 7812 menghasilkan tegangan keluaran +12 VDC, LM 7809 menghasilkan tegangan keluaran +9 VDC. Berikut skema dari IC regulator dan bentuk fisik LM 7805/7812 ditunjukkan pada gambar 2.6.



Gambar 2.6 IC Regulator 7805/12

2.7 *Transmission Control Protocol/Internet Protocol (TCP/IP)*[9]

Transmission Control Protocol (TCP) adalah suatu protokol atau perantara yang dapat mentransmisikan data persegmen, artinya paket data dipecah dalam jumlah yang sesuai dengan besaran paket, kemudian dikirimkan satu persatu hingga selesai. Selain itu TCP bertugas menangani pengiriman pesan dan juga mendefinisikan suatu mekanisme pengiriman dari semua jenis data pada suatu jaringan. Sedangkan *Internet Protocol (IP)* adalah protokol yang mengatur routing dari pentransmisian melewati jaringan antara pengirim dan penerima, termasuk juga yang berkaitan dengan pengalamatan jaringan dan komputer, sehingga dapat dikatakan bahwa *Internet Protocol* merupakan perantara komunikasi antar komputer dengan menggunakan alamat IP (*IP Address*) sebagai suatu identitas dari jaringan maupun komputer. Pengiriman data akan dikemas dalam paket dengan label berupa IP Address pengirim dan penerima paket data. Apabila penerima melihat pengiriman paket tersebut dengan identitas, maka paket data tersebut akan diambil dan disalurkan ke TCP melalui port, dimana aplikasi menunggu. Sehingga dapat dikatakan bahwa TCP merupakan protocol level tertinggi yang disebut sebagai protocol aplikasi. Layer ini bertugas untuk komunikasi antar

aplikasi user-visible interface pada TCP/IP. *Transmission Control Protocol/Internet Protocol* (TCP/IP) merupakan standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP stack.

Protokol TCP/IP dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas. TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme transport jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang memungkinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di Internet. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti

Microsoft Windows dan keluarga UNIX) untuk membentuk jaringan yang heterogen. *Transmission Control Protocol/Internet Protocol* (TCP/IP) adalah satu set aturan standar komunikasi data yang digunakan dalam proses transfer data dari satu komputer ke komputer lain di jaringan komputer tanpa melihat perbedaan jenis hardware. Protokol TCP/IP selalu berevolusi seiring dengan waktu, mengingat semakin banyaknya kebutuhan terhadap jaringan komputer dan Internet. Pengembangan ini dilakukan oleh beberapa badan, seperti halnya Internet Society (ISOC), Internet Architecture Board (IAB), dan Internet Engineering Task Force (IETF). Macam-macam protokol yang berjalan di atas TCP/IP, skema pengalamatan, dan konsep TCP/IP didefinisikan dalam dokumen yang disebut sebagai Request for Comments (RFC) yang dikeluarkan oleh IETF. TCP/IP mempunyai beberapa layer, layer-layer itu adalah :

1. IP (*internet protocol*) yang berperan dalam pentransmisiian paket data dari node ke node. IP mendahului setiap paket data berdasarkan 4 byte (untuk versi IPv4) alamat tujuan (nomor IP). *Internet authorities* menciptakan *range* angka untuk organisasi yang berbeda. Organisasi menciptakan grup dengan nomornya untuk departemen. IP bekerja pada mesin *gateway* yang

memindahkan data dari departemen ke organisasi kemudian ke region dan kemudian ke seluruh dunia.

2. TCP (*transmission transfer protocol*) berperan didalam memperbaiki pengiriman data yang benar dari suatu klien ke *server*. Data dapat hilang di tengah-tengah jaringan. TCP dapat mendeteksi error atau data yang hilang dan kemudian melakukan transmisi ulang sampai data diterima dengan benar dan lengkap.
3. Sockets yaitu merupakan nama yang diberikan kepada paket yang menyediakan akses ke TCP/IP pada kebanyakan sistem.

2.8 ADC Pada Mikropengendali ATmega 128 [3,14]

Kelebihan mikropengendali AVR Atmega 128 dan kebanyakan AVR lainnya karena telah dilengkapi dengan ADC internal sehingga tidak perlu memasang *chip* ADC di luar mikropengendali untu kebutuhan pengubahan sinyal analog menjadi sinyal digital. Sebelum membahas tentang proses tesebut. mulai dengan perbedaan antara sinyal analog dan sinyal digital. Sinyal adalah sekumpulan nilai yang meepresentasikan keadaan dimana dari suatu variabel fisik. Kumpulan tersebut bisa berupa satu nilai yang kecil atau

dapat juga memiliki banyak nilai sesuai yang diinginkan. Dalam dunia rekayasa, biasanya menyusun nilai-nilai tersebut secara berurut, misalnya terhadap waktu, atau terhadap sebuah sumbu *spatial* untuk menampilkan informasi. Variabel waktu dan *spatial* disebut variabel independen, karena mereka tidak dipengaruhi oleh kepentingan variabel-variabel fisik. Proses yang terjadi dalam ADC adalah *Sampling* (Pencuplikan), *Quantization* (Penguantisasian) dan *Coding* (Pengkodean). Tiga hal penting dalam ADC adalah *sampling*, *quantization* dan *encoding*. Pertama adalah *sampling*. *Sampling* adalah proses mengambil suatu nilai pasti (diskrit) dalam suatu data kontinu dalam satu titik tertentu dengan periode yang tetap. *Sampling* sangat penting karena jika ingin merepresentasikan sebuah sinyal analog dalam suatu sistem digital, misalnya dalam kompute, harus menggunakan *sampling rate* yang tepat untuk meng-‘capture’ sinyal analog untuk representasi yang sesuai/tepat dalam sistem digital. *Quantization* adalah proses pengelompokan data diskrit yang didapatkan pada proses pertama ke dalam kelompok-kelompok data. Kuantisasi, dalam matematika dan pemrosesan sinyal digital, adalah proses pemetaan nilai *input* seperti nilai pembulatan. Setiap sistem digital mempunyai jumlah bit yang digunakan sebagai unit dasar untuk

merepresentasikan data. Bit adalah unit dasar yang dinyatakan dalam representasi 0 dan 1. Sedangkan istilah *nibble* digunakan untuk menyatakan 4 bit, serta *byte* untuk menyatakan 8 bit. Ketika dilakukan sampling pada sebuah sinyal, sistem digital memerlukan sejumlah arti untuk merepresentasikan sample yang di-*'capture'*. Kuantisasi dari sebuah sinyal yang di-*'sample'* adalah bagaimana sinyal tersebut direpresentasikan sebagai satu level kuantisasi. Pada sistem digital umumnya, sinyal *in/out* adalah sinyal-sinyal tegangan. Sinyal tegangan merupakan sinyal yang pertama diperoleh dari sinyal fisik dengan bantuan transduser, seperti mikropon, sensor temperatur dan sebagainya. Sinyal tegangan kemudian dikondisikan untuk dipetakan dalam rentang mereka dengan rentang masukan dari sebuah sistem digital, tipikalnya adalah 0 sampai 5 V. Dengan n bit akan diizinkan untuk membagi rentang sinyal masukan dalam sebuah sistem digital menjadi 2^n level kuantisasi. Misalnya jika ADC 8 bit maka akan diperoleh sebanyak $2^8 = 256$ level kuantisasi. *Encoding* adalah mengkodekan data hasil kuantisasi ke dalam bentuk digital (0/1) atau dalam suatu nilai biner. Proses pengkodean dari hasil kuantisasi menjadi bilangan biner digital. Anggap menggunakan 8 bit untuk melakukan kuantisasi sebuah sinyal analog yang telah di-*'sample'*. Level kuantisasi ditentukan oleh 8 bit, dan setiap

sinyal yang di-*'sample'* adalah satu kuantisasi dari 256 level kuantisasi.

2.8.1 Fitur ADC ATmega 128[3,14]

Salah satu fitur yang sangat maju pada era *single chip computer* adalah ditanamkannya sebuah pengubah sinyal analog ke digital (ADC, *analog to digital converter*) dalam sebuah *chip* mikropengendali dalam sebuah chip mikropengendali. Mikropengendali AVR ATmega 128 merupakan salah satu turunan mikrokontroler yang dibekali dengan ADC *internal*. ADC yang dimiliki ATmega 128 mempunyai 8 buah saluran masukan analog yang multiplex, serta mempunyai resolusi hingga 10 bit. Artinya bahwa tegangan analog antara 0 dan 5 V akan dikodekan menjadi salah satu dari 1024 representasi biner antara 000H dan 3FFH. Ini memberikan ATmega 128 tegangan dengan resolusi ser 4,88 mv. Dalam mode operasinya ADC ATmega 128 dapat dioperasikan sebagai masukan tunggal (*single ended input*) ataupun sebagai masukan jamak (*different input*) yang dimultiplex. Selain itu, ADC ATmega 128 memiliki konfigurasi pewaktuan, tegangan referensi, mode operasi, dan kemampuan menyaring noise (gangguan) yang cukup fleksibel sehingga dapat dengan mudah disesuaikan dengan kebutuhan. Selain itu juga

disediakan fitur pembanding analog (*analog comparator*) yang terdapat di dalam mikropengendali Atmega 128 (internal) sebagai fasilitas untuk melakukan pembandingan tegangan analog yang diumpankan dari luar sistem. Fitur ATmega 128 sebagai ADC 10-bit *successive approximation*. ADC dihubungkan dengan *multiplexer* 8-kanal yang mengizinkan delapan masukan tegangan tunggal (*Single-ended voltage*) yang dibangun dari pin-pin Port F. Masukan tegangan tunggal merujuk ke 0V (GND).

2.8.2 Tegangan Referensi dan *Step Size* [10,11]

Tegangan referensi adalah tegangan masukan yang digunakan untuk tegangan referensi. Tegangan dihubungkan dengan pin ini, bersama dengan resolusi ADC mempengaruhi *step size*. *Step size* adalah perubahan nilai terkecil yang dapat dirasakan/dibedakan oleh ADC. V_{RH} adalah tegangan referensi *high* ADC dan V_{RL} adalah tegangan referensi *low* ADC, b adalah jumlah bit yang tersedia untuk dikonversi. Untuk ATmega 128 dengan tegangan referensi 5 VDC dan tersedia resolusi 10 bit untuk konversi, maka resolusinya adalah 4,88 Mv. Akurasi absolut ditetapkan ± 2 LSB pada resolusi ini adalah 9,66 mV. Besarnya nilai ADC merupakan hasil dari persamaan (2.1)

$$ADC = \frac{V_{in} \times 1024}{V_{ref}} \dots \dots \dots (2.1)$$

Sehingga berdasarkan rumus tersebut hubungan antara nilai ADC dan Tegangan *Input* (V_{in}) dapat dicari yaitu sebagai berikut :

$$V_{in}=(ADC \times V_{ref})/ 1024 \dots\dots\dots(2.2)$$

2.8.3 Antarmuka Sensor Temperatur LM35[3]

Sensor temperatur LM₃₅/LM_{35A} mempunyai rentang temperatur -55°C s.d. +150°C dengan skala keluaran 10mV/°C atau dengan kata lain bahwa sensor temperatur LM₃₅/LM_{35A} mempunyai step size 10 mV, di mana setiap perubahan 1°C. ADC AVR mempunyai resolusi 10-bit yang berarti mempunyai 1024 langkah (maksimum). Sekarang, jika menggunakan ADC dengan *step size* 10 mV, maka V_{out} yang akan diperoleh adalah tegangan skala penuh sebesar 10240 mV (10,24 V). Hal ini tidak dapat diterima walaupun temperatur maksimum yang dirasakan oleh LM₃₅/LM_{35A} adalah 150°C, dan keluaran tertinggi yang akan diperoleh adalah 1500 mV (1,5 V). Selain itu, jika menggunakan tegangan referensi internal 2,56 V, maka step size akan menjadi $2,56V/1024 = 2,5$ mV. Hal ini memberikan bilangan *output* untuk ADC empat kali temperatur ril karena sensor menghasilkan 10 mV untuk setiap derajat perubahan temperatur dan *step size* adalah 2,5 mV ($10mV/2,5$ mV =

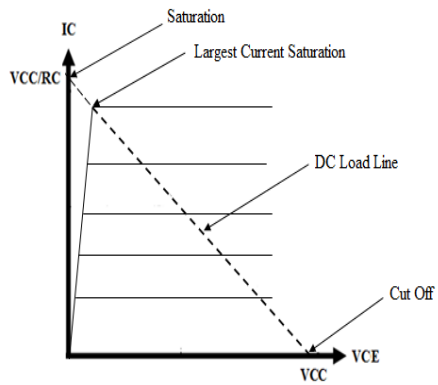
- 4). Untuk dapat melakukan penskalaan dapat dengan membaginya dengan 4 untuk mendapatkan bilangan ril untuk temperatur.

2.9 Transistor Sebagai Saklar [12]

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 selalu 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.

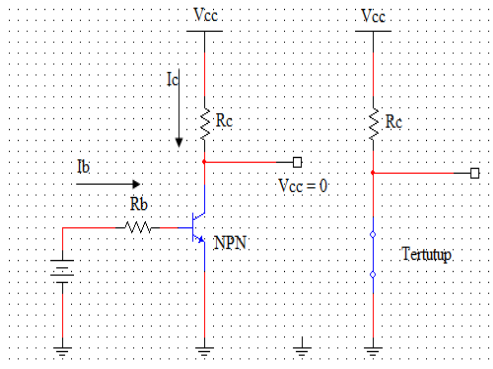
Daerah *cut off* sering dinamakan sebagai daerah mati karena pada daerah kerja ini transistor tidak dapat mengalirkan arus dari kolektor ke emitor. Pada daerah *cut off* transistor dapat di analogikan sebagai saklar terbuka pada hubungan kolektor–emitor. Untuk membuat transistor penghantar, pada masukan basis perlu diberi tegangan. Besarnya tegangan harus lebih besar dari V_{BE} (0,3 untuk *germanium* dan 0,7 untuk *silicon*). Dengan mengatur $I_b > I_c/\beta$ kondisi transistor akan menjadi jenuh seakan kolektor dan emitor *short circuit*. Arus mengalir dari kolektor ke emitor tanpa hambatan dan $V_{ce} \approx 0$. Besar arus yang mengalir dari kolektor ke emitor sama dengan V_{cc}/R_c . Keadaan seperti ini menyerupai saklar dalam kondisi tertutup (ON). Grafik hubungan daerah jenuh, daerah aktif dan daerah mati pada transistor dapat ditunjukkan pada gambar 2.7. 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*.



Gambar 2.7 Grafik Garis Beban DC

2.9.1 Transistor Pada Kondisi Jenuh [12]

Untuk kondisi transistor pada kondisi jenuh atau saturasi ditunjukkan pada gambar 2.8.



Gambar 2.8 Transistor Pada Kondisi Jenuh

Untuk kondisi transistor pada keadaan jenuh atau saturasi maka besarnya tegangan kolektor-emitor VCE suatu transistor pada konfigurasi pada gambar 2.12 dapat diketahui sebagai berikut [16]

$$V_{ce} = V_{cc} - I_c \cdot R_c \dots\dots\dots (2.3)$$

Karena kondisi jenuh $V_{ce} = 0$ Volt (transistor ideal) maka besarnya arus kolektor (I_c) adalah

$$I_c = \frac{V_{cc}}{R_c} \dots\dots\dots (2.4)$$

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

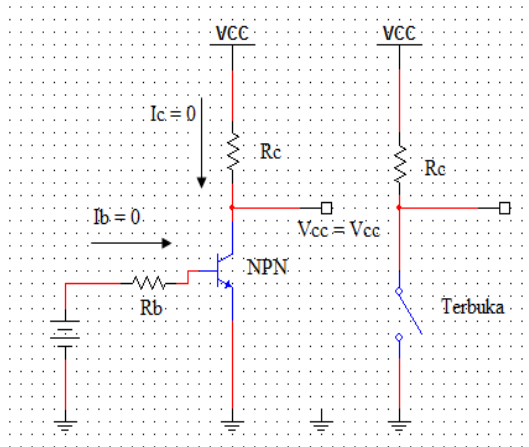
$$R_b = \frac{V_i - V_{be}}{I_b} \dots\dots\dots (2.5)$$

Sehingga besarnya arus basis (I_b) pada kondisi saturasi adalah

$$I_b = \frac{I_c}{\beta} \dots\dots\dots (2.6)$$

2.9.2 Transistor Pada Kondisi Mati [12]

Untuk kondisi transistor dalam kondisi mati adalah keadaan yang menyerupai saklar pada kondisi terbuka. Untuk kondisi transistor pada kondisi mati atau *cut off* ditunjukkan pada gambar 2.9.



Gambar 2.9 Transistor Pada Kondisi *Cut off*

Dengan mengatur $I_b = 0$ 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 ($I_c \approx 0$) dan $V_{ce} \approx V_{cc}$. Keadaan ini menyerupai saklar pada kondisi terbuka seperti ditunjukkan pada gambar diatas.

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

$$V_{ce} = V_{cc} - I_c \cdot R_c \dots\dots\dots(2.7)$$

Karena kondisi mati $I_c=0$ (transistor ideal) maka :

$$V_{ce} = V_{cc} \dots\dots\dots(2.8)$$

Dengan demikian maka besar arus basis Ib adalah

$$I_b = \frac{I_c}{\beta} \dots \dots \dots (2.9)$$

$$I_b = \dots \dots \dots (2.10)$$

2.10 BASCOM AVR[13]

BASCOM-AVR adalah program *basic compiler* berbasis windows untuk mikropengendali keluarga AVR merupakan pemrograman dengan bahasa tingkat tinggi ” BASIC ” yang dikembangkan dan dikeluarkan oleh MCS elektronika sehingga dapat dengan mudah dimengerti atau diterjemahkan. Dalam program BASCOM-AVR terdapat beberapa kemudahan, untuk membuat program *software* ATMega 128, seperti program simulasi yang sangat berguna untuk melihat, simulasi hasil program yang telah dibuat, sebelum program tersebut download ke IC atau ke mikropengendali. BASCOM-AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat dengan pergerakan LED yang ada pada layar simulasi dan dapat juga langsung dilihat pada LCD, jika membuat aplikasi yang berhubungan dengan LCD. Bahasa BASIC adalah salah satu bahasa pemrograman yang banyak digunakan untuk aplikasi mikropengendali karena kemudahan dan kompatibilitas terhadap mikropengendali jenis AVR

dan didukung oleh *compiler software* berupa BASCOM-AVR. Setiap bahasa pemrograman mempunyai standar penulisan program. Konstruksi dari program bahasa BASIC harus mengikuti aturan sebagai berikut :

\$regfile = "header"

'inisialisasi

'deklarasi variabel

'deklarasi konstanta

Do

'pernyataan-pernyataan

Loop

end