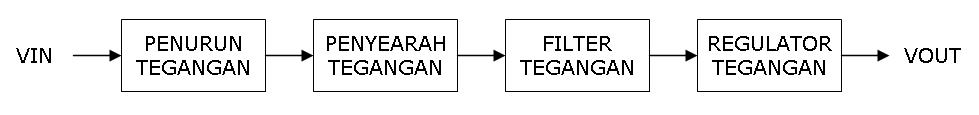
**BAB II**

**TINJAUAN PUSTAKA**

**2.1 Catu Daya**

Pencatu daya (*power supply*) adalah sebuah piranti [elektronika](http://id.wikipedia.org/wiki/Elektronika) yang berguna sebagai sumber daya untuk piranti lain, terutama daya [listrik](http://id.wikipedia.org/wiki/Listrik)[[1]](#footnote-2).

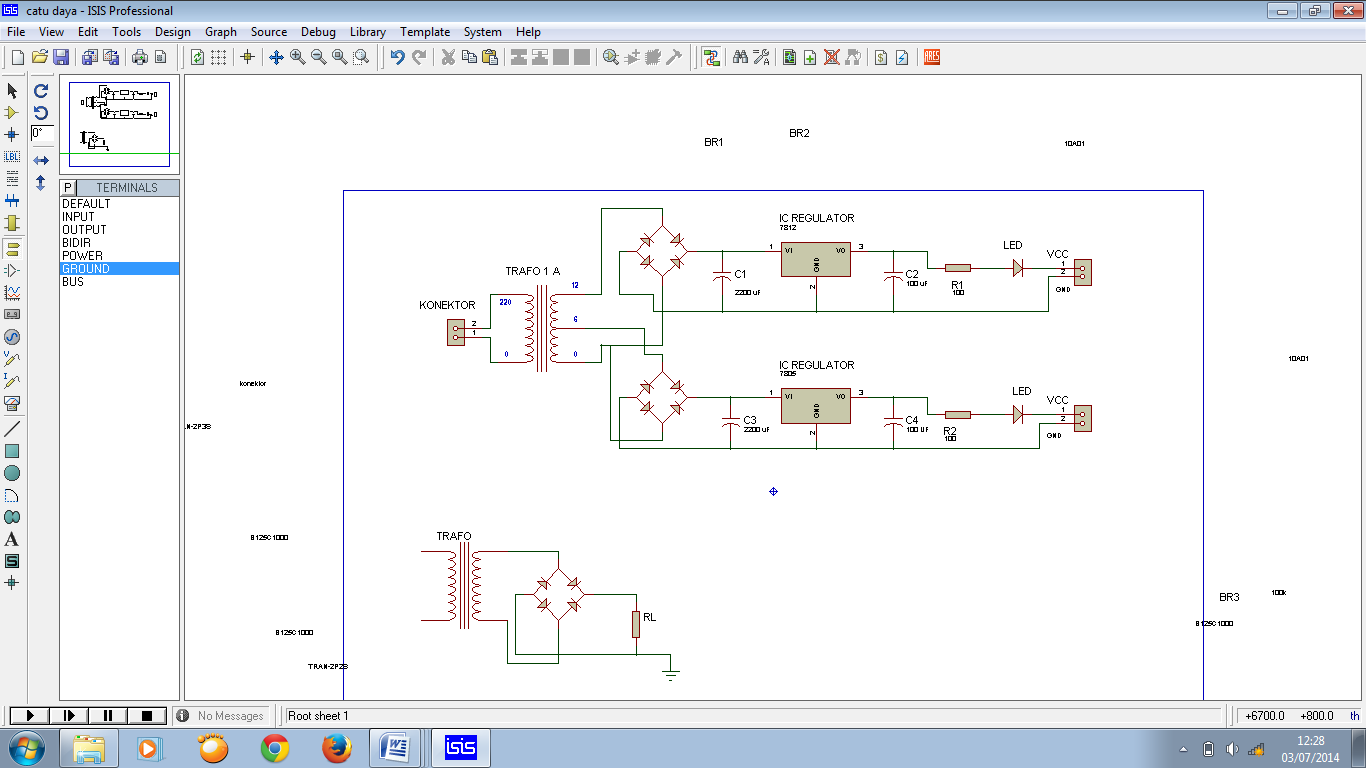
Catu daya terdiri dari beberapa komponen yang memiliki fungsinya masing-masing yang mana semua komponen tersebut saling bekerja sama untuk menghasilkan tegangan yang diinginkan. Catu daya yang dibutuhkan untuk alat ini adalah sebesar 5 volt dan 12 volt. Tegangan 5 volt dan 12 volt ini dapat dihasilkan dengan menggunakan IC *regulator* 7805 dan 7812 pada keluaran catu daya. Gambar 2.1 di bawah ini merupakan blok diagram dari catu daya yaitu sebagai berikut.



Gambar 2.1 Blok Diagram Catu Daya

**2.1.1 Penurun Tegangan**

*Transformator* adalah suatu komponen yang berguna untuk menaikkan atau menurunkan tegangan, misalnya tegangan 220 volt diturunkan menjadi tegangan yang kita inginkan. Gambar 2.2 di bawah ini merupakan penggunaan *transformator* pada rangkaian penyearah gelombang penuh dengan yaitu sebagai berikut.



Gambar 2.2 Rangkaian Penyearah Gelombang Penuh

Pada dasarnya *transformator* terdiri dari dua gulungan (lilitan) yaitu lilitan *primer* dan lilitan *sekunder*. Ada dua jenis *transformator* yang sering digunakan yaitu *transformator step up* untuk menaikkan tegangan dan *transformator step down* untuk menurunkan tegangan. Pada pembuatan catu daya pada alat ini digunakan jenis *transformator step down*. *Transformator step down* memilikijumlah lilitan *sekunder* lebih sedikit daripada lilitan *primer* sehingga berfungsi sebagai penurun tegangan[[2]](#footnote-3).

*Transformator* bekerja berdasarkan prinsip [induksi](http://id.wikipedia.org/w/index.php?title=Induksi&action=edit&redlink=1) [*elektromagnetik*](http://id.wikipedia.org/wiki/Elektromagnet)*.* Tegangan masukan bolak-balik yang membentangi *primer* menimbulkan [*fluks* magnet](http://id.wikipedia.org/w/index.php?title=Fluks_magnet&action=edit&redlink=1) yang idealnya semua bersambung dengan lilitan *sekunder*. [*Fluks*](http://id.wikipedia.org/wiki/Fluks) bolak-balik ini menginduksikan gaya gerak listrik ([GGL](http://id.wikipedia.org/w/index.php?title=GGL&action=edit&redlink=1)) dalam lilitan *sekunder*. Jika *efisiensi* sempurna, semua daya pada lilitan *primer* akan dilimpahkan ke lilitan *sekunder*.

Besar tegangan dan kuat arus pada *transformator* tergantung pada banyaknya lilitan. Semakin banyak jumlah lilitan maka tegangan yang dihasilkan akan semakin besar.  Hal ini berlaku untuk lilitan *primer* dan *sekunder* pada *transformator*. *Transformator* dikatakan ideal jika tidak ada energi yang hilang menjadi kalor, yaitu ketika jumlah energi yang masuk pada kumparan *primer*sama dengan jumlah energi yang keluar pada kumparan *sekunder*[[3]](#footnote-4).

**2.1.2 Penyearah, *Filter* dan *Regulator***

Penyearah adalah suatu rangkaian yang dapat mengubah tegangan AC menjadi tegangan DC[[4]](#footnote-5). Pada dasarnya, suatu penyearah terbagi menjadi tiga jenis, yaitu penyearah gelombang penuh dengan jembatan, penyearah setengah gelombang, dan penyearah gelombang penuh. Pada rangkaian ini penyearah yang digunakan adalah penyearah gelombang penuh dengan jembatan.

Penyearah gelombang penuh dengan sistem jembatan bekerja dalam satu siklus yaitu setengah siklus positif dan setengah siklus negatif. Pada saat setengah siklus positif, arus I1 mengalir melalui D1, C, RL dan D3 hal ini dikarenakan D1 dan D3 bekerja karena mendapatkan bias maju. sedangkan D2 dan D3 mati karena mendapat bias mundur.

Pada saat siklus negatif maka arus I2 akan mengalir melalui D2, C, RL, dan D4. Hal ini dikarenakan D2 Dan D4 mendapat bias maju dan D1 dan D3 mendapat bias mundur.

Pada penyearah gelombang penuh, harga rata-rata dari sinyal gelombang penuh sama dengan dua kali harga rata-rata dari sinyal setengah gelombang sehingga[[5]](#footnote-6):

Vrms = 0,707Vp .................................. (2.1)

V2(peak) = .................................... (2.2)

Vout(peak) = ............................. (2.3)

Vdc = 0,636Vout(peak) . ......................... (2.4)

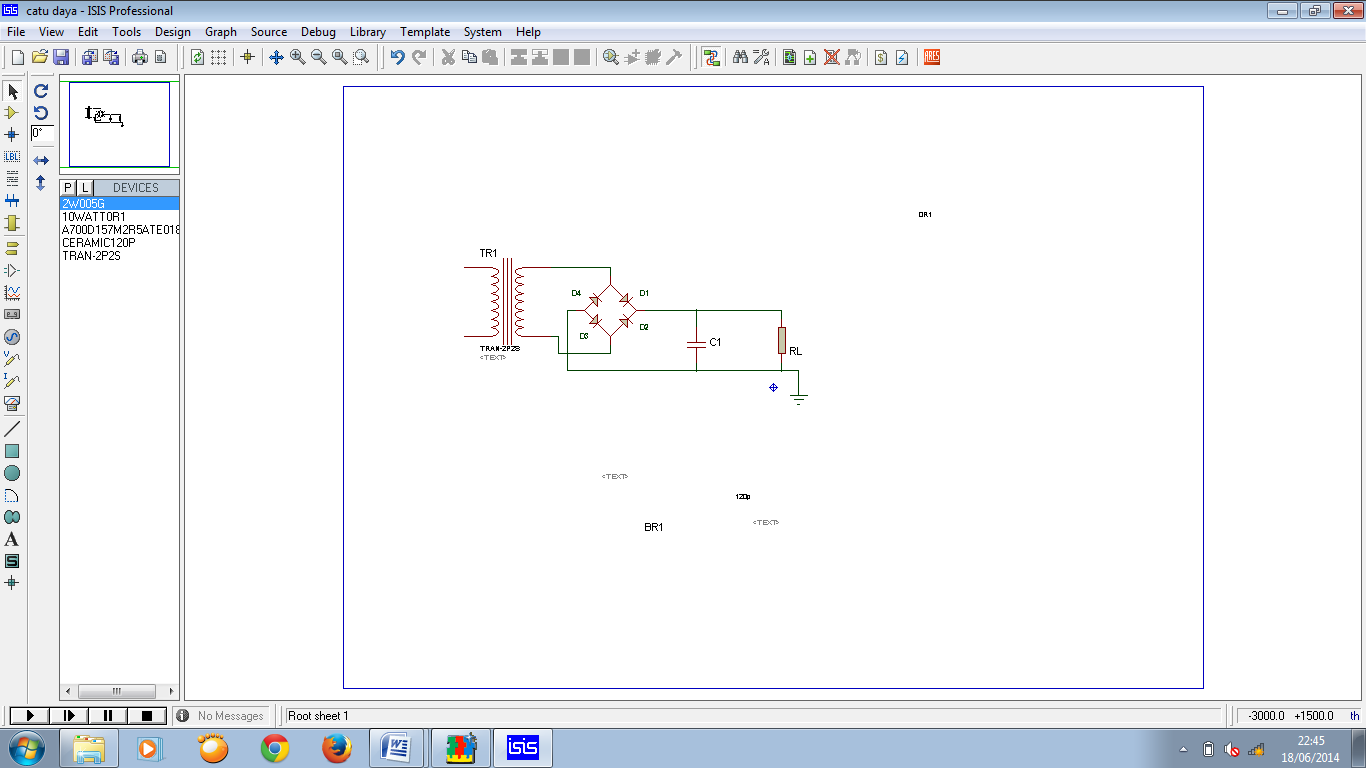
PIV = V2(peak) ..................................... (2.5)

Idc = .............................................(2.6)

I diode = .........................................(2.7)

Dalam rangkaian penyearah baik itu penyearah setengah gelombang ataupun gelombang penuh dengan menggunakan dioda, tegangan DC yang dihasilkan masih terdapat *ripple*. *Ripple* adalah sinyal AC yang tidak diinginkan.

Untuk menghasilkan tegangan DC yang benar-benar rata maka diperlukan rangkaian penyaring berupa *filter kapasitif* (berupa kapasitor) untuk mengurangi tegangan *ripplenya*. Semakin besar nilai kapasitor maka akan semakin kecil nilai tegangan *ripple.* Pada gambar 2.3 di bawah ini dapat dilihat rangkaian penyearah gelombang penuh jembatan dengan *filter* C.



Gambar 2.3 Rangkaian Penyearah Gelombang Penuh dengan Filter C

Setelah mendapatkan tegangan DC yang benar-benar rata maka tegangan tersebut kemudian akan distabilkan oleh IC *regulator* agar tegangan yang dihasilkan catu daya sesuai dengan keperluan. Dalam pembuatan alat ini menggunakan 2 buah IC *regulator* yaitu 7805 untuk menghasilkan tegangan sebesar 5 volt dan 7812 untuk menghasilkan tegangan sebesar 12 volt. Gambar 2.4 di bawah ini merupakan rangkaian *power supply* yaitu sebagai berikut.

****

Gambar 2.4 Rangkaian *Power Supply*

**2.2 Arduino[[6]](#footnote-7)**

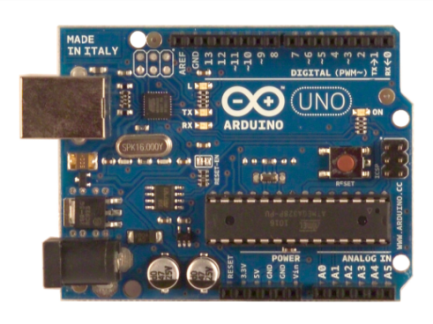
Arduinodikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. *Physical computing* adalah membuat sebuah sistem atau perangkat fisik dengan menggunakan *software* dan *hardware* yang sifatnya interaktif yaitu dapat menerima rangsangan dari lingkungan dan merespon balik. Dalam penggunaan arduino dikenal dengan kata *processing*. *Processing*ini adalah bahasa pemrograman yang digunakan untuk menulis program di dalam Arduino. *Processing* adalah bahasa pemrograman tingkat tinggi yang dialeknya sangat mirip dengan C++ dan Java.

Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi arduino adalah kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory* mikrokontroler.

Jenis-jenis papan arduino yang dijual di pasaran yaitu sebagai berikut:

1. Arduino USB

Jenis arduino yang menggunakan USB sebagai antar muka pemrograman atau komunikasi komputer. Contohnya yaitu arduino *uno,* arduino *duemilanove,* arduino *diecimila,* arduino *ng rev. c,* arduino *ng (nuova generazione),* arduino *extreme* dan arduino *extreme v2,* arduinoUSB dan arduino USB v2.0. Foto dari bentuk fisik arduino USB dapat dilihat pada gambar 2.5 di bawah ini.



Gambar 2.5 Foto Bentuk Fisik Arduino USB

(Sumber <http://arduino.cc/en/main/boards>)

1. Arduino *Serial*

Jenis arduino ini menggunakan RS232 sebagai antar muka pemrograman atau komunikasi komputer. Contohnya yaitu arduino *serial* dan arduino *serial* v2.0. Foto dari bentuk fisik arduino *serial* dapat dilihat pada gambar 2.6 di bawah ini.

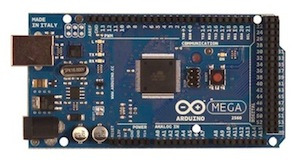


Gambar 2.6 Foto Bentuk Fisik Arduino Serial

(Sumber <http://arduino.cc/en/main/boards>)

1. Arduino Mega

Jenis papan arduino dengan *spesifikasi* yang lebih tinggi, dilengkapi tambahan *pin digital*, *pin analog*, *port serial* dan sebagainya. Contohnya yaitu arduino mega, arduino mega 2560. Foto dari bentuk fisik arduino mega dapat dilihat pada gambar 2.7 di bawah ini.

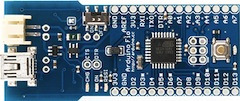


Gambar 2.7 Foto Bentuk Fisik Arduino Mega

(Sumber <http://arduino.cc/en/main/boards>)

1. Arduino FIO

Jenis arduino yang digunakan untuk penggunaan *nirkabel*. Arduino FIO memiliki *header* untuk radio XBee, *konektor* untuk baterai Lipo, dan rangkaian pengisian baterai. Bentuk dari arduino FIO dapat dilihat pada gambar 2.8 di bawah ini.

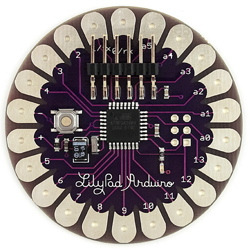


Gambar 2.8 Foto Bentuk Fisik Arduino FIO

(Sumber <http://arduino.cc/en/main/boards>)

1. Arduino *Lilypad*

Jenis papan arduino dengan bentuk yang melingkar. Contohnya yaitu *lilypad* arduino 00, *lilypad* arduino 01*, lilypad* arduino 02*, lilypad* arduino 03*, lilypad* arduino 04. Foto dari bentuk fisik arduino *lilypad* dapat dilihat pada gambar 2.9 di bawah ini.



Gambar 2.9 Foto Bentuk Fisik Arduino *Lilypad*

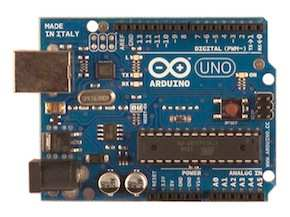
(Sumber <http://arduino.cc/en/main/boards>)

Komponen utama di dalam papan arduino adalah sebuah mikrokontroler 8 *bit* dengan *merk* Atmega yang dibuat oleh perusahaan *Atmel Corporation*. Berbagai papan arduino menggunakan tipe Atmega yang berbeda-beda tergantung dari *spesifikasinya*, sebagai contoh arduino uno menggunakan Atmega328 sedangkan arduino mega 2560 yang lebih canggih menggunakan Atmega2560. Pada pembuatan alat ini digunakan arduino jenis USB yaitu arduino uno.

**2.2.1 Arduino Uno**

**Arduino uno adalah salah satu produk berlabel arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler Atmega 328 (sebuah keping yang secara fungsional bertindak seperti sebuah komputer)**[[7]](#footnote-8)**.**

Fotodari bentuk fisik arduino uno dapat dilihat pada gambar 2.10 di bawah ini.

****

Gambar 2.10 Foto Bentuk Fisik Arduino Uno

(Sumber http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2011-2-01650-SK%20Bab2001.pdf)

Arduino uno memiliki 14 *digital* *input/output* pin (dimana 6 *pin* dapat digunakan sebagai *output PWM*), 6 *input* *analog*, 16 MHz *osilator* kristal, *koneksi* USB, *jack* listrik dan tombol *reset*. *Pin-pin* ini berisi semua yang diperlukan untuk mendukung mikrokontroler[[8]](#footnote-9).

Berikut adalah *spesifikasi* dari arduino uno yang ditunjukkan pada tabel 2.1 di bawah ini :

Tabel 2.1 *Spesifikasi* Arduino Uno

|  |  |
| --- | --- |
| Mikrokontroler | ATmega328 |
| Tegangan pengoperasian | 5V |
| Tegangan *input* yang disarankan | 7-12V |
| Batas tegangan *input* | 6-20V |
| Jumlah *pin* I/O *digital* | 14 (6 di antaranya menyediakan keluaran PWM) |
| Jumlah *pin* *input analog* | 6 |
| Arus DC tiap *pin* I/O | 40 Ma |
| Arus DC untuk *pin* 3.3V | 50 mA |
| Memori *Flash* | 32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| *Clock Speed* | 16 MHz |

**2.2.2 Komunikasi pada Arduino Uno**

Arduino uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, arduino lain, atau mikrokontroler lain. Arduino uno dapat terhubung ke komputer dengan menggunakan kabel USB. Dalam hal ini, kebutuhan listrik disuplai oleh *Personal Computer* (PC), sedangkan apabila arduino uno hanya digunakan sendiri maka diperlukan sumber tegangan *eksternal* sebesar 9 Volt. Apabila arduino uno sudah terhubung ke komputer, ada dua indikator yang menyatakan bahwa arduino tidak bermasalah yaitu sebagai berikut:

* Indikator pertama yaitu berupa lampu kecil berlabel ON yang akan menyala.
* Indikator kedua yaitu berupa lampu kecil yang terhubung ke *pin* 13 yang akan berkedip-kedip[[9]](#footnote-10).

Kedua keadaan yang menyatakan arduino tidak bermasalah bukan berarti menunjukkan bahwa arduino sudah dapat digunakan. Diperlukan sebuah *driver* arduino yang harus diinstal terlebih dahulu agar dapat terhubung ke PC.

**2.2.3** **Program pada Arduino Uno**

Bagi pemula, *software* arduino yang digunakan adalah berupa *driver* dan IDE. IDE arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE arduino terdiri dari:

* Editor program yaitu sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
* *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*. Mikrokontroler dapat memahami dalam kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
* *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan arduino.

Dalam penggunaan arduino uno dikenal dengan istilah *sketch*. *Sketch* adalah istilah yang digunakan di arduino uno untuk menyatakan program.

**2.2.4 Dasar Pemrograman Arduino Uno**

**2.2.4.1 Variabel dan Tipe Data[[10]](#footnote-11)**

Variabel adalah suatu nama di memori mikrokontroler yang berguna untuk menyimpan nilai dan nilainya dapat diubah sewaktu-waktu. Dalam penggunaan variabel pada *sketch*, variabel tersebut harus dideklarasikan terlebih dahulu. Hal ini dilakukan sebagai suatu pemberitahuan bahwa terdapat variabel yang akan digunakan di dalam *sketch,* sekaligus digunakan sebagai penentuan tipe data untuk variabel tersebut.

Tipe data menyatakan bentuk data yang dapat disimpan di dalam variabel. Tipe data yang dapat digunakan di *sketch* cukup beragam. Tipe data yang tersedia dapat dilihat pada tabel 2.2 di bawah ini.

Tabel 2.2 Tipe Data Dasar

|  |  |  |
| --- | --- | --- |
| **Tipe Data** | **Keterangan** | **Kebutuhan Memori** |
| *boolean* | Tipe *boolean* hanya dapat digunakan untuk menampung dua nilai saja yaitu *true* atau *false*. Kedua nilai tersebut adalah *konstanta* yang sudah disediakan di bahasa arduino. variabel yang bertipe *boolean* biasa digunakan pada kondisi pernyataan seperti *if, while*, dan *for.* | 1 *byte* |
| *char* | Tipe *char* berguna untuk menyimpan sebuah nilai karakter seperti ‘A’, ‘9’, dan ‘\*’. *konstanta* karakter ditulis didalam tanda petik tunggal. Di dalam memori, karakter disimpan dalam bentuk bilangan (nilai ASCII), yang berkisar antara -128 hingga 127. | 1 *byte* |
| *Unsigned char* | Serupa dengan tipe *char,* tetapi bilangan akan dikodekan dalam bentuk bilangan positif antara 0 sampai dengan 255. | 1 *byte* |
| *byte* | Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara 0 hingga 255. | 1 *byte* |
| *int* | Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara -32768  hingga 32767. | 2 *byte* |

**2.2.4.2 Ekspresi, Operand, dan Operator**

Ekspresi atau ungkapan adalah suatu rangkaian operator, variabel, fungsi, atau konstanta yang ditujukan untuk menghasilkan sebuah nilai dengan tipe tertentu[[11]](#footnote-12). Operand adalah sebuah objek yang ada pada operasi matematika yang dapat digunakan untuk melakukan operasi[[12]](#footnote-13). Semua operator yang tergolong sebagai operator aritmatika dapat dilihat pada tabel 2.3 di bawah ini.

Tabel 2.3 Operator Aritmatika

|  |  |  |
| --- | --- | --- |
| **Operator** | **Keterangan** | **Contoh** |
| + | Tanda plus | +3 |
| - | Tanda minus | -3 |
| + | Penjumlahan  Tipe hasil ekspresi bergantung pada tipe operand.   * jika keseluruhan operand berupa bilangan bulat, hasil ekspresi berupa bilangan bulat. * jika salah satu operand berupa bilangan real, hasil ekspresi berupa bilangan real. | 3 + 2 5  3.0 + 2 5.0 |
|  | Pengurangan  Tipe hasil ekspresi bergantung pada tipe operand.   * jika keseluruhan operand berupa bilangan bulat, hasil ekspresi berupa bilangan bulat. * jika salah satu operand berupa bilangan real, hasil ekspresi berupa bilangan real. | 3 – 2 1  3.0 – 2 1.0 |
| \* | Perkalian  Tipe hasil ekspresi bergantung pada tipe operand.   * jika keseluruhan operand berupa bilangan bulat, hasil ekspresi berupa bilangan bulat. * jika salah satu operand berupa bilangan real, hasil ekspresi berupa bilangan real. | 3 \* 2 6  3.0 \* 2 6.0 |
| / | Pembagian  Tipe hasil ekspresi bergantung pada tipe operand.   * jika keseluruhan operand berupa bilangan bulat, hasil ekspresi berupa bilangan bulat. * jika salah satu operand berupa bilangan real, hasil ekspresi berupa bilangan real. | 7 /2 3  7 / 2.0 3.5 |
| % | Sisa pembagian  Operator ini hanya berlaku untuk bilangan bulat | 5 % 2 3  5 % 3 2  8 % 2 0 |

**2.2.4.3 Pernyataan *If [[13]](#footnote-14)***

Pernyataan *if* sangat diperlukan untuk menangani masalah yang memerlukan pengambilan keputusan. Bentuk pernyataan *if* yang paling sederhana seperti berikut:

***if***(kondisi)

pernyataan

Pada bentuk seperti itu, bagian pernyataan hanya akan dijalankan jika kondisi bernilai benar (*true*). Bentuk if yang lain seperti berikut:

***if (***kondisi)

pernyataan\_A

***else***

pernyataan\_B

Pada bentuk seperti itu, pernyataan\_A dijalankan jika kondisi bernilai benar *(true*). Adapun pernyataan\_B (yang terletak pada bagian *else*) hanya akan dijalankan jika kondisi bernilai salah (*false*).

Pernyataan yang dijalankan di bagian *if* ataupun di *else* bila lebih dari satu. Pada keadaan seperti itu, pernyataan yang lebih dari satu perlu ditulis di dalam tanda { dan }. Jadi, bentuk *if* tanpa *else* yang digunakan untuk menjalankan beberapa pernyataan seperti berikut.

***if*** (kondisi){

pernyataan\_A1

....

pernyataan\_AN

}

***else***{

pernyataan\_B1

...

pernyataan\_BM

}

Pada bentuk di atas keberadaan pasangan { dan } mengapit beberapa pernyataan. Pernyataan\_A1 sampai pernyataan\_AN dijalankan jika kondisi bernilai benar. Adapun pernyataan\_B1 sampai pernyataan\_BM dijalankan jika kondisi bernilai salah.

**2.2.4.4 Komunikasi Arduino Uno dan Komputer[[14]](#footnote-15)**

Komunikasi antara arduino uno dan komputer dapat dilakukan melalui *port serial* (via USB). Dalam hal ini, arduino uno tidak hanya dapat membaca data dari komputer yang ada di *port serial*, melainkan juga dapat mengirim data ke komputer. Perintah yang berhubungan dengan komunikasi *serial* ditangani oleh objek bernama *serial*. Fungsi-fungsi yang tersedia di objek *serial* ditunjukkan pada tabel 2.4 di bawah ini.

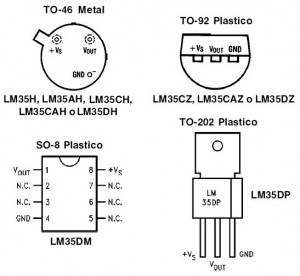
Tabel 2.4 Fungsi-fungsi untuk Berkomunikasi secara *Serial*

|  |  |
| --- | --- |
| Fungsi | Keterangan |
| Serial.begin(kecepatan) | Serial.begin() berguna untuk menentukan kecepatan pengiriman dan penerimaan data melalui *port serial*. Kecepatan yang umum digunakan 9600 bit per detik. Namun kecepatan hingga 115.200 didukung oleh arduino uno. |
| Serial.read() | Serial.read() berguna untuk membaca satu *byte* data yang terdapat di *port serial*. Setelah pemanggilan Serial.read(), jumlah data di *port* *serial* berkurang satu. Untuk membaca seluruh data, diperlukan perintah sebagai berikut:  Int data;  While (Serial.available()){  Data = Serial.read();  ...  } |
| Serial.print(data)  Serial.print(data,format) | Serial.print() berguna untuk mengirimkan data ke *port serial*. Apabila argumen format disertakan, data yang dikirimkan akan menyesuaikan dengan format tersebut. Dalam hal ini, format yang digunakan dapat berupa:   * DEC (format desimal atau basis 10) * HEX (format heksadesimal atau basis 16) * OCT (format oktal atau basis 8) * BIN (format biner atau basis 2) |
| Serial.println(data)  Serial.println(data,format) | Perintah ini secara prinsip sama dengan Serial.print(). Perbedaannya, Serial.println() menambahkan “\r\n” (*carriage return dan linefeed*) yang memberi efek perpindahan baris di dalam tampilan. |

**2.3 Sensor LM 35**

Sensor adalah alat yang digunakan untuk mendeteksi dan sering berfungsi untuk mengukur *magnitude* sesuatu. Sensor adalah jenis *transduser* yang digunakan untuk mengubah variasi mekanis, magnetis, panas, sinar dan kimia menjadi tegangan dan arus listrik. Sensor biasanya dikategorikan melalui pengatur dan memegang peranan penting dalam pengendalian proses pabrikasi modern. Sensor memberikan ekivalen mata, pendengaran, hidung, lidah untuk menjadi otak mikroprosesor dari sistem otomatisasi industri[[15]](#footnote-16).

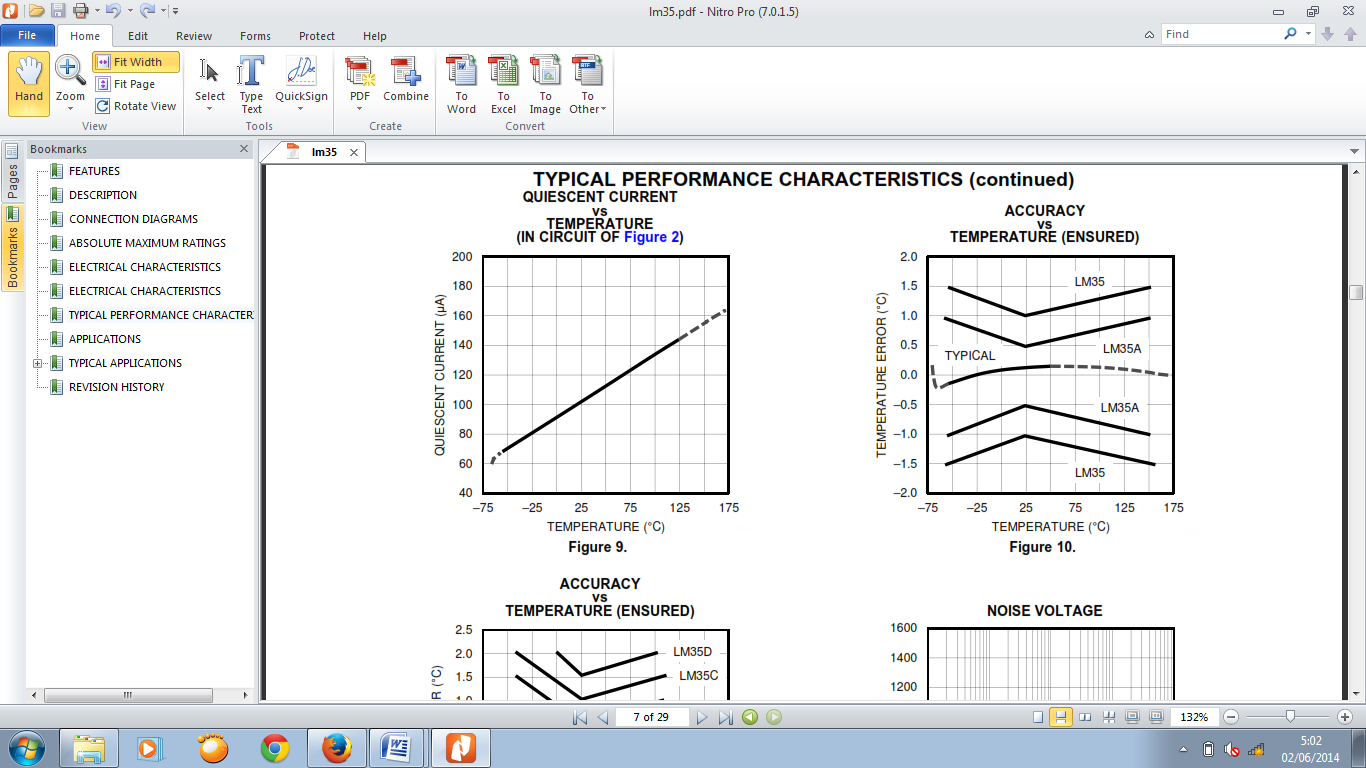
Sensor suhu LM 35 merupakan komponen elektronika yang berfungsi untuk mengubah besaran fisis yang berupa suhu menjadi besaran elektris[[16]](#footnote-17). Foto dari bentuk fisik sensor suhu LM 35 dapat dilihat pada gambar 2.11 di bawah ini.



Gambar 2.11 Foto Bentuk Fisik Sensor LM 35

(Sumber http://labdegaragem.com)

Sensor suhu LM 35 memiliki parameter bahwa setiap kenaikan 1ºC tegangan keluarannya naik sebesar 10 mV dengan batas maksimal keluaran sensor adalah 1,5 V pada suhu 150°C. Grafik dari akurasi LM 35 terhadap suhu dapat dilihat pada gambar 2.12 di bawah ini.



Gambar 2.12 Grafik Akurasi LM 35 Terhadap Suhu

(Sumber datasheet LM 35)

LM35 memiliki kelebihan – kelebihan sebagai berikut:

1. Di kalibrasi langsung dalam *celsius*

2. Memiliki faktor skala *linear* + 10.0 mV/°C

3. Memiliki ketetapan 0,5°C pada suhu 25°C

4. Jangkauan maksimal suhu antara -55°C sampai 150°C

5. Cocok untuk aplikasi jarak jauh

6. Harganya cukup murah

7. Bekerja pada tegangan catu daya 4 sampai 30 volt

8. Memiliki arus *drain* kurang dari 60 µAmp

9. Pemanasan sendiri yang lambat ( *low self-heating*)

10. *low self-heating* sebesar 0,08˚C di udara

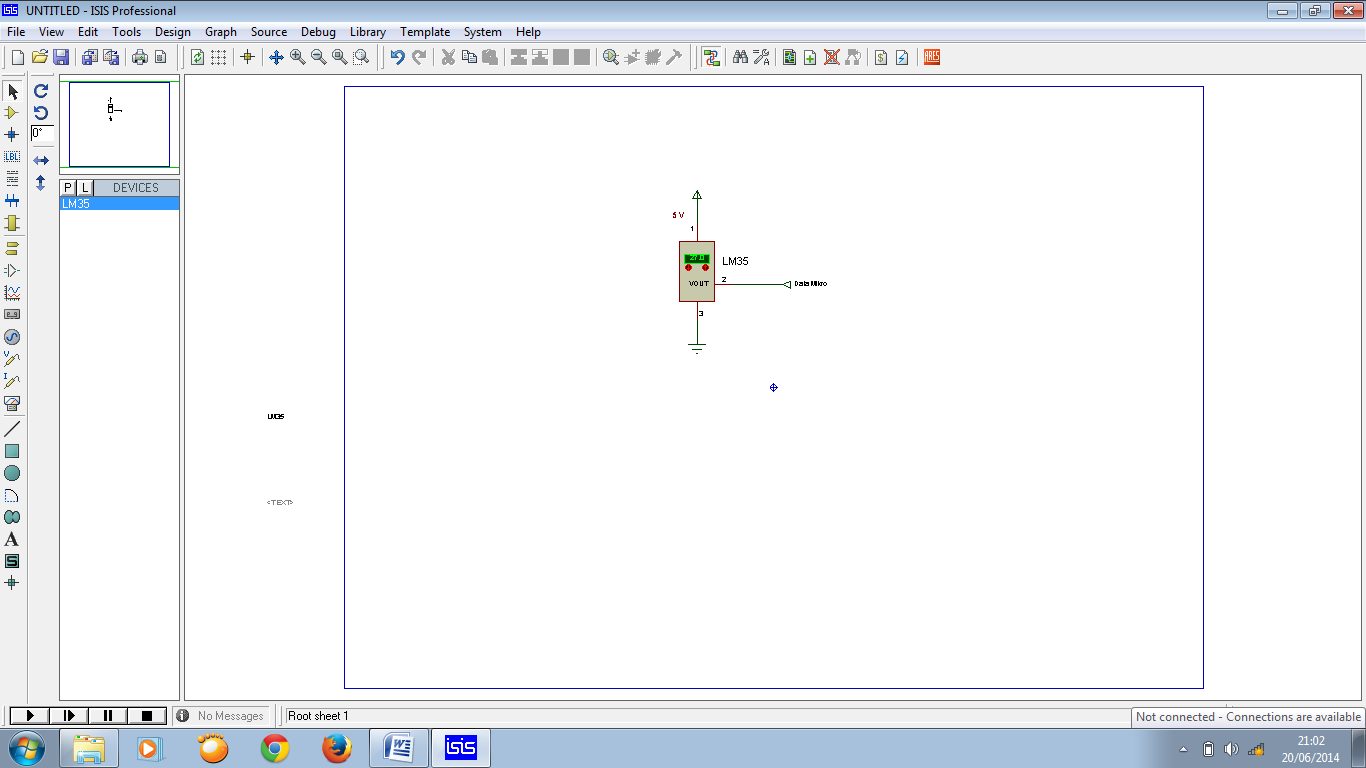
11. Ketidak linearanya hanya sekitar ±¼°C

12. Memiliki impedansi keluaran yang kecil yaitu 0,1 *watt* untuk beban 1 mAmp.

Sensor suhu tipe LM 35 merupakan IC sensor temperatur yang akurat yang tegangan keluarannya linear dalam satuan *celcius*. Sensor LM 35 memiliki kelebihan dibandingkan sensor temperatur *linear* dalam satuan kelvin, karena tidak memerlukan pembagian dengan *konstanta* tegangan yang besar dan keluarannya untuk mendapatkan nilai dalam satuan *celcius* yang tepat. LM 35 memiliki impedansi keluaran yang rendah, keluaran yang *linear*, dan sifat ketepatan dalam pengujian membuat proses *interface* untuk membaca atau mengontrol sirkuit lebih mudah. *Pin* V+ dari LM 35 dihubungkan ke catu daya, *pin* GND dihubungkan ke *ground* dan *pin* *Vout* yang menghasilkan tegangan *analog* hasil pengindera suhu dihubungkan ke *Vin* (+) dan ADC pada arduino.

LM 35 dapat disuplai dengan tegangan mulai 4 volt -30 volt DC dengan arus pengurasan 60 mikroampere, memiliki tingkat efek *self-heating*yang rendah (0,08 derajad *celcius*).

*Self-heating*adalah efek pemanasan oleh komponen itu sendiri akibat adanya arus yang bekerja melewatinya. Untuk komponen sensor suhu, parameter ini harus dipertimbangkan atau di *handle* dengan baik karena hal ini dapat menyebabkan kesalahan pengukuran. Skematik rangkaian dasar dari sensor suhu LM 35 dapat dilihat pada gambar 2.13 di bawah ini.



Gambar 2.13 Rangkaian Sensor Suhu LM 35

**2.4 *Analog to Digital Converter* (ADC)**

*Analog to digital converter* (ADC) adalah pengubah *input* analog menjadi kode – kode digital. Umumnya ADC digunakan sebagai perantara antara sensor yang kebanyakan analog dengan sistem komputer seperti sensor suhu, cahaya, tekanan/berat, aliran dan sebagainya kemudian diukur dengan menggunakan sistem digital[[17]](#footnote-18). Konsep pengubah sinyal analog menjadi sinyal digital adalah *sampling* terhadap sinyal analog yang kemudian mewakilinya dengan bilangan digital dengan batas yang sudah diberikan[[18]](#footnote-19).

**2.4.1 ADC pada Arduino Uno**

Arduino uno terdiri dari 6 buah saluran analog yaitu A0 sampai A5 dan juga terdiri dari 10 bit data ADC. hal ini berarti arduino akan memetakan tegangan input dari 0 sampai 5 V menjadi nilai *integer* antara 0 dan 1023 yang menghasilkan *resolusi* pembacaan 5 Volt / 1024 unit[[19]](#footnote-20).

Cara menentukan nilai ADC adalah sebagai berikut:

ADC= x 1024 ..............................(2.8)

keterangan:

ADC = nilai ADC

Vin = tegangan masukan

Vref = tegangan referensi

**2.5 *Liquid Crystal Display* (LCD) 16 x 2[[20]](#footnote-21)**

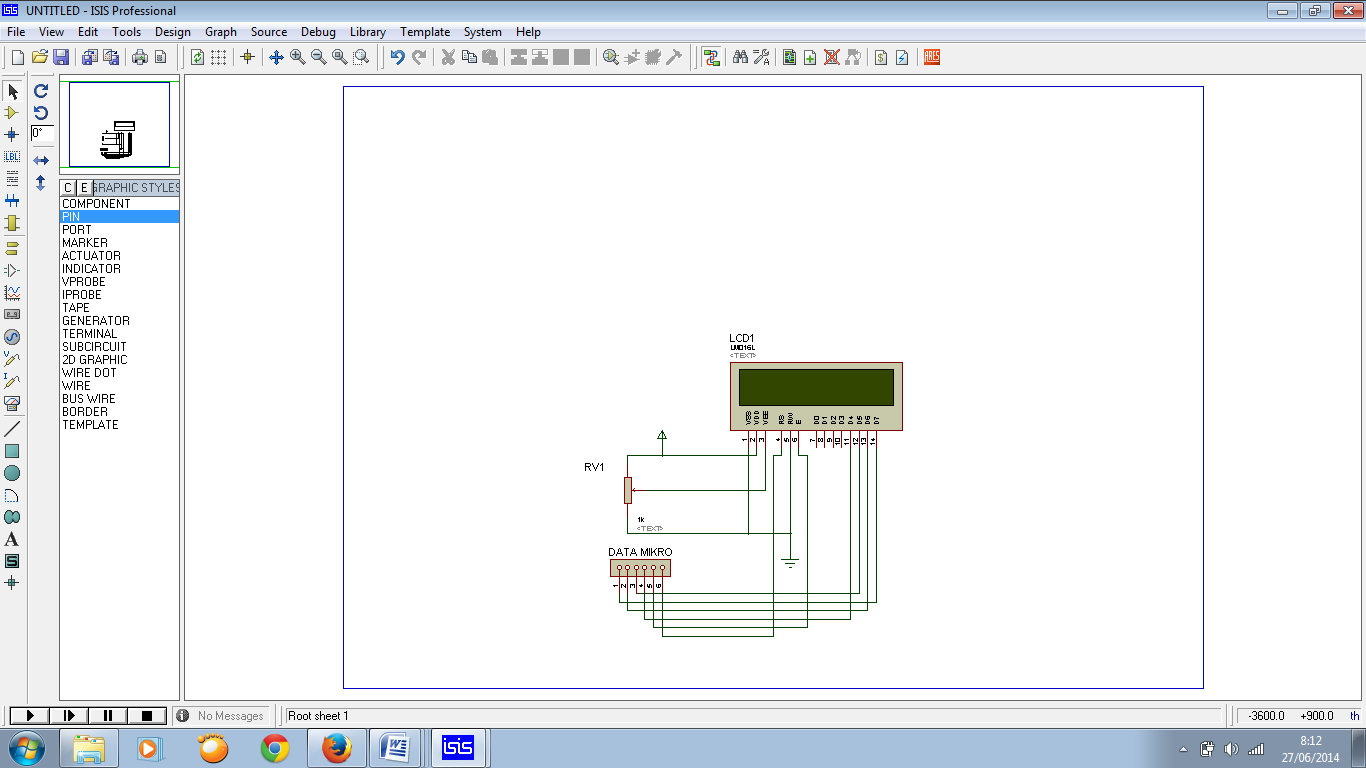
LCD berfungsi untuk menampilkan karakter angka, huruf ataupun simbol dengan lebih baik dan dengan konsumsi arus yang rendah. LCD merupakan modul LCD buatan hitachi. Modul LCD terdiri dari bagian penampil karakter yang berfungsi menampilkan karakter dan bagian sistem prosesor LCD dalam bentuk modul dengan mikrokontroler yang diletakan di bagian belakang LCD tersebut yang berfungsi untuk mengatur tampilan LCD serta mengatur komunikasi antara LCD dengan mikrokontroler yang menggunakan modul LCD tersebut.

**2.5.1 Fungsi Pin *Liquid Cristal Display* (LCD)**

Pada LCD terdiri dari pin- pin sebagai berikut:

* DB0 – DB7 adalah jalur data (data bus) yang berfungsi sebagai jalur komunikasi untuk mengirimkan dan menerima data atau instruksi dari mikrokontroler ke modul LCD.
* RS adalah pin yang berfungsi sebagai selektor register (*register sellect)* yaitu dengan memberikan logika *low* (0) sebagai register perintah dan logika *high* (1) sebagai register data.
* R/W adalah pin yang berfungsi untuk menentukan *mode* baca atau tulis dari data yang terdapat pada DB0 – DB7 yaitu dengan memberikan logika *low* (0) untuk fungsi *read* dan logika *high* (1) untuk *mode* *write*.
* *Enable* (E), berfungsi sebagai *Enable Clock* LCD, logika 1 setiap kali pengiriman atau pembacaan data.

Berikut adalah gambar dari rangkaian LCD 16 x2 yang dapat dilihat pada gambar 2.14 di bawah ini.



Gambar 2.14 Rangkaian LCD 16 x 2

**2.5.2 Penulisan Data *Register* Perintah *Liquid Cristal Display (*LCD)**

Penulisan data ke registerperintah dilakukan dengan tujuan mengatur tampilan LCD, inisialisasi dan mengatur *Address Counter* maupun *Address* Data. Kondisi RS berlogika 0 menunjukkan akses data ke register perintah. RW berlogika 0 yang menunjukkan proses penulisan data akan dilakukan. *Nibble* tinggi (bit 7 sampai bit 4) terlebih dahulu dikirimkan dengan diawali pulsa logika 1 pada E *Clock*. Kemudian *Nibble* rendah (bit 3 sampai bit 0) dikirimkan dengan diawali pulsa logika 1 pada E *Clock* lagi. Untuk mode 8 bit *interface*, proses penulisan dapat langsung dilakukan secara 8 bit (bit 7 … bit 0) dan diawali sebuah pulsa logika 1 pada E *Clock*.

**2.5.3 Pembacaan Data *Register* Perintah *Liquid Cristal Display* (LCD)**

Proses pembacaan data pada *register* perintah biasa digunakan untuk melihat status *busy* dari LCD atau membaca *Address Counter*. RS diatur pada logika 0 untuk akses ke *register* perintah, R/W diatur pada logika 1 yang menunjukkan proses pembacaan data. 4 bit *nibble* tinggi dibaca dengan diawali pulsa logika 1 pada E *Clock* dan kemudian 4 bit *nibble* rendah dibaca dengan diawali pulsa logika 1 pada E *Clock*. Untuk Mode 8 bit *interface*, pembacaan 8 bit (*nibble* tinggi dan rendah) dilakukan sekaligus dengan diawali sebuah pulsa logika 1 pada E *Clock*.

**2.5.4 Penulisan Data *Register* Data *Liquid Cristal Display* (LCD)**

Penulisan data pada *register* data dilakukan untuk mengirimkan data yang akan ditampilkan pada LCD. Proses diawali dengan adanya logika 1 pada RS yang menunjukkan akses ke *register* data, kondisi R/W diatur pada logika 0 yang menunjukkan proses penulisan data. Data 4 bit *nibble* tinggi (bit 7 hingga bit 4) dikirim dengan diawali pulsa logika 1 pada sinyal E *Clock* dan kemudian diikuti 4 bit *nibble* rendah (bit 3 hingga bit 0) yang juga diawali pulsa logika 1 pada sinyal E *Clock*.

**2.5.5 Pembacaan Data *Register* Data *Liquid Cristal Display* (LCD)**

Pembacaan data dari *register* data dilakukan untuk membaca kembali data yang tampil pada LCD. Proses dilakukan dengan mengatur RS pada logika 1 yang menunjukkan adanya akses ke *Register* Data. Kondisi R/W diatur pada logika tinggi yang menunjukkan adanya proses pembacaan data. Data 4 bit *nibble* tinggi (bit 7 hingga bit 4) dibaca dengan diawali adanya pulsa logika 1 pada E *Clock* dan dilanjutkan dengan data 4 bit *nibble* rendah (bit 3 hingga bit 0) yang juga diawali dengan pulsa logika 1 pada E *Clock*.

**2.6 *Relay***

*Relay* adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, *relay* merupakan tuas saklar dengan lilitan kawat pada batang besi (*solenoid)* di dekatnya. Ketika *solenoid* dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada *solenoid* sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. *Relay* biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 ampere AC 220 V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 ampere 12 Volt DC). *Relay* yang paling sederhana ialah *relay elektromekanis* yang memberikan pergerakan mekanis saat mendapatkan energi listrik.

Penggunaan *relay* perlu memperhatikan tegangan pengontrolnya serta kekuatan *relay* men*switch* arus/tegangan. Biasanya ukurannya tertera pada *body relay*. Misalnya *relay* 12VDC/4 A 220V, artinya tegangan yang diperlukan sebagai pengontrolnya adalah 12Volt DC dan mampu men-*switch* arus listrik (maksimal) sebesar 4 ampere pada tegangan 220 Volt. Sebaiknya *relay* difungsikan 80% saja dari kemampuan maksimalnya agar aman, lebih rendah lagi lebih aman. *Relay* jenis lain ada yang namanya *reedswitch* atau *relay* lidi. *Relay* jenis ini berupa batang kontak terbuat dari besi pada tabung kaca kecil yang dililitin kawat. Pada saat lilitan kawat dialiri arus, kontak besi tersebut akan menjadi magnet dan saling menempel sehingga menjadi saklar yang *on*. Ketika arus pada lilitan dihentikan medan magnet hilang dan kontak kembali terbuka (*off*). Foto dari bentuk fisik *relay* dapat dilihat pada gambar 2.15 di bawah ini.



Gambar 2.15 Foto Bentuk Fisik *Relay*

(sumber <http://www.meriwardanaku.com/2011/11/prinsip-kerja-relay.html>)

*Relay* yang sering digunakan dalam rangkaian ada dua jenis, yaitu:

1. *Relay* Magnetik

*Relay* magnetik adalah *relay* yang bekerja berdaasarkan magnet listrik untuk menggerakkan kontak-kontak mekaniknya dalam keadaan NO maka akan berubah menjadi NC dan sebaliknya.

1. *Relay* Elektronik

*Relay* elektronik merupakan *relay* yang bekerja dengan menggunakan komponen-komponen elektronika pada waktu pensuplaiannya.

**2.6.1 Sifat- Sifat *Relay***

Sifat-sifat *relay* adalah sebagai berikut:

* Kuat arus yang diperlukan untuk pengoperasian *relay* ditentukan oleh pabrik pembuatnya. *Relay* dengan tahanan kecil memerlukan arus yang besar dan juga sebaliknya*, relay* dengan tahanan besar memerlukan arus yang kecil.
* Tegangan yang diperlukan untuk menggerakkan suatu *relay* akan sama dengan kuat arus yang dikalikan dengan tahanan atau hambatan *relay*.
* Daya yang diperlukan untuk menggerakkan *relay* sama dengan tegangan yang dikalikan dengan arus.

**2.6.2 Prinsip Kerja *Relay***

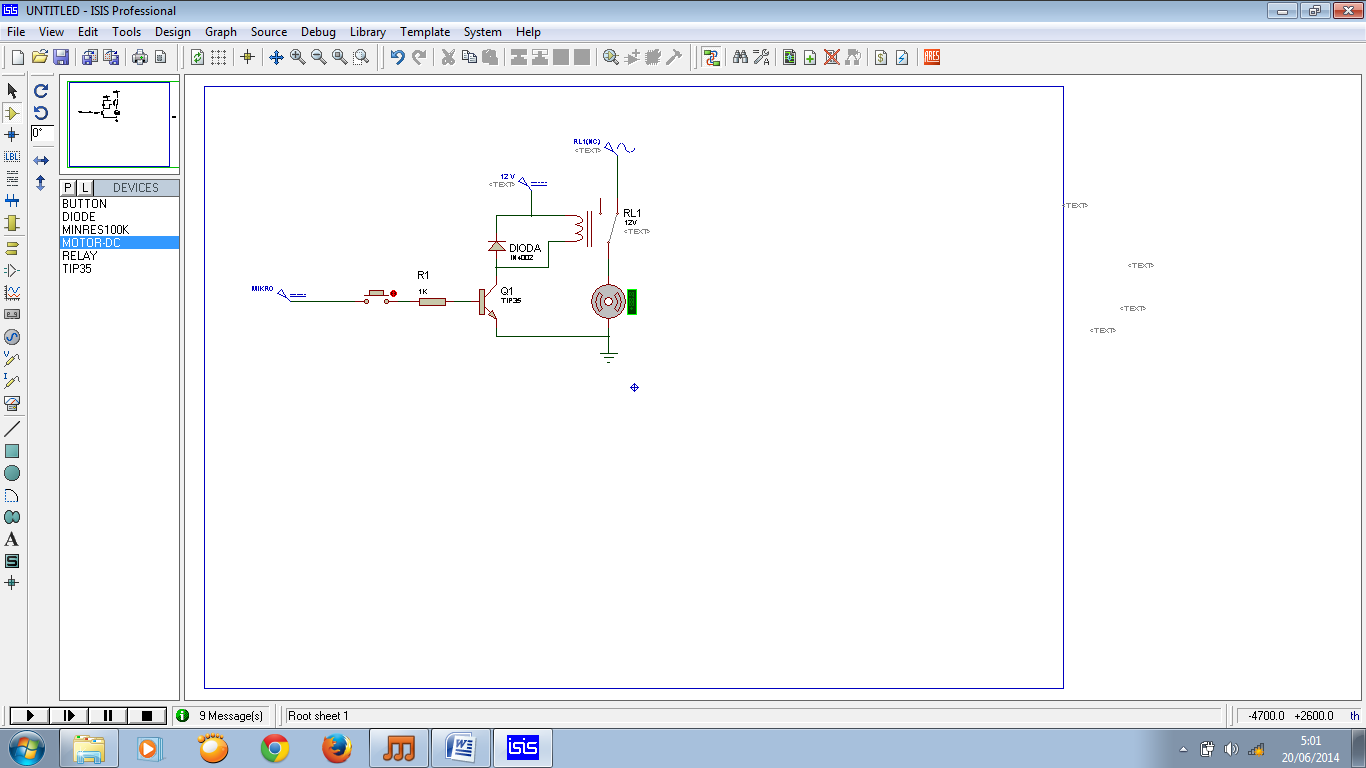
*Relay* terdiri dari *coil* dan *contact*. *Coil* adalah gulungan kawat yang mendapat arus listrik, sedangkan *contact* adalah  sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*.  *Contact* ada 2 jenis yaitu *normally open* (kondisi awal sebelum diaktifkan *open*) dan  *normally closed* (kondisi awal sebelum diaktifkan *close*).  Secara sederhana berikut ini adalah prinsip kerja dari *relay* yaitu sebagai berikut:

Ketika *coil* mendapat energi  listrik (*energized*) maka akan timbul gaya elektromagnet yang akan menarik *armature* yang berpegas dan *contact* akan menutup *(normally open).*

**2.6.3 Rangkaian *Driver Relay***

Penggunaan *relay* pada rangkaian, memerlukan sebuah rangkaian pengendali untuk dapat membuat *relay* beroperasi. Rangkaian *relay* ini dikenal dengan rangkaian *driver relay.*

Berikut adalah gambar rangkaian dari *driver relay* yang dapat dilihat pada gambar 2.16 di bawah ini.

****

Gambar 2.16 Rangkaian *Driver Relay*

(Sumber [www.google.com](http://www.google.com))

Rangkaian *relay* di atas memanfaatkan *common emitor*, apabila basis mendapat sinyal input logika 1 (sumber tegangan positif) maka transistor akan mendapat bias maju, sehingga transistor ON dan memberikan sumber tegangan ke *relay* dan *relay* menjadi ON. Agar dapat memberikan prategangan balik kepada dioda maka diperlukan harga VCE[[21]](#footnote-22).

VCE = VCB + VBE ................................(2.9)

Untuk transistor silikon, persamaan tersebut menjadi:

VCE = VCB + 0,7 V................................( 2.10)

Apabila VCB lebih besar dari nol, dioda akan memperoleh prategangan balik. jadi VCE harus melebihi 0,7 V untuk menghasilkan prategangan balik. Hambatan R1 mempunyai tegangan dari mikrokontroler pada ujung kiri dan tegangan dari VBE di ujung kanannya. Jadi beda tegangan lintas R1 adalah V mikro – VBE dan arus basisnya sama dengan[[22]](#footnote-23):

IB =

VBB -VBE

............................(2.11)

RB

Dalam daerah operasi aktif, arus kolektor diberikan oleh rumus:

IC = βdcIB .........................................(2.12)

1. http://id.wikipedia.org/wiki/Pencatu\_daya [↑](#footnote-ref-2)
2. http://id.wikipedia.org/wiki/Transformator [↑](#footnote-ref-3)
3. http://www.rumus-fisika.com/2014/03/transformator.html [↑](#footnote-ref-4)
4. Evelina. 2010. Laboratorium Analog 1. hal 19 [↑](#footnote-ref-5)
5. Malvino Barmawi. 1985. Aproksimasi Rangkaian Semikonduktor. Edisi ke Empat.hal 97 [↑](#footnote-ref-6)
6. [www.tobuku.com](http://www.tobuku.com). pengenalanan arduino. [↑](#footnote-ref-7)
7. Abdul Kadir. 2012.Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya menggunakan Arduino. hal 16 [↑](#footnote-ref-8)
8. http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2011-2-01650-SK%20Bab2001.pdf [↑](#footnote-ref-9)
9. Locit. abdul kadir. hal 17 [↑](#footnote-ref-10)
10. locit. abdul kadir. hal 56 [↑](#footnote-ref-11)
11. locit. abdul kadir. hal 63 [↑](#footnote-ref-12)
12. http://id.wikibooks.org/wiki/Pemrograman\_C/Operand\_dan\_Operator [↑](#footnote-ref-13)
13. locit. abdul kadir. hal 70 [↑](#footnote-ref-14)
14. locit.abdul kadir. hal 102 [↑](#footnote-ref-15)
15. Petruzella Frank D. 1996. Elektronika Industri. Edisi II. hal 157 [↑](#footnote-ref-16)
16. http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2011-2-01650-SK%20Bab2001.pdf [↑](#footnote-ref-17)
17. http://elektronika-dasar.web.id/teori-elektronika/adc-analog-to-digital-convertion/ [↑](#footnote-ref-18)
18. http://thesis.binus.ac.id/Asli/Bab2/2007-1-00229-SK-Bab%202.pdf [↑](#footnote-ref-19)
19. http://arduino.cc/en/Reference/analogRead [↑](#footnote-ref-20)
20. http://elektronika-dasar.web.id/komponen/lcd-liquid-cristal-display-dot-matrix-2x16-m1632/ [↑](#footnote-ref-21)
21. Malvino Barmawi. 1985. Aproksimasi Rangkaian Semikonduktor. Edisi ke Empat. hal 134 [↑](#footnote-ref-22)
22. ibid. Malvino Barmawi. hal 136 [↑](#footnote-ref-23)