

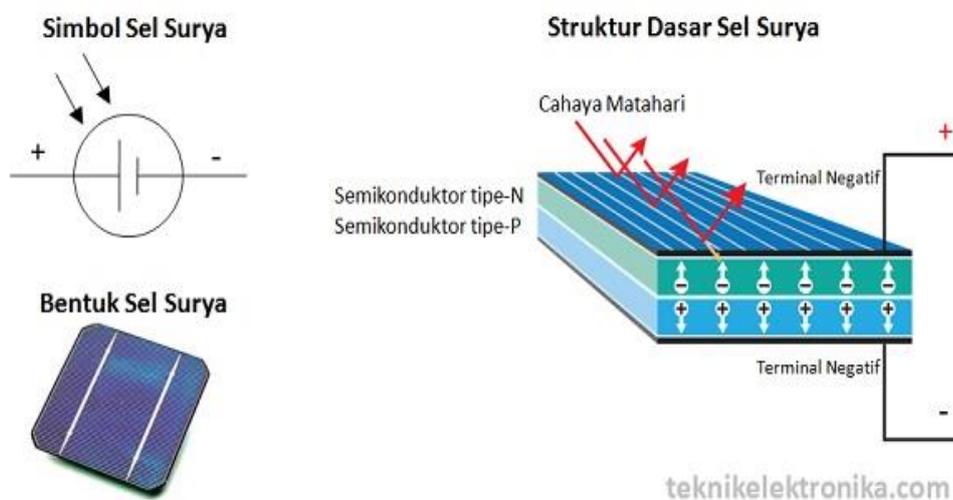
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Solar Cell

Sel Surya atau *Solar Cell* adalah suatu perangkat atau komponen yang dapat mengubah energi cahaya matahari menjadi energi listrik dengan menggunakan prinsip efek *Photovoltaic*. Yang dimaksud dengan Efek *Photovoltaic* adalah suatu fenomena dimana munculnya tegangan listrik karena adanya hubungan atau kontak dua *elektroda* yang dihubungkan dengan sistem padatan atau cairan saat mendapatkan energi cahaya. Oleh karena itu, Sel Surya atau *Solar Cell* sering disebut juga dengan Sel *Photovoltaic*. (Henri Becquerel, 1839)

Arus listrik timbul karena adanya energi foton cahaya matahari yang diterimanya berhasil membebaskan elektron-elektron dalam sambungan semikonduktor tipe N dan tipe P untuk mengalir. Sama seperti Dioda Foto (*Photodiode*), Sel Surya atau *Solar Cell* ini juga memiliki kaki Positif dan kaki Negatif yang terhubung ke rangkaian atau perangkat yang memerlukan sumber listrik.



**Gambar 2.1** Sollar Cell

(Sumber : <http://teknikelektronika.com/pengertian-sel-surya-solar-cell-prinsip-kerja-sel-surya>)

## 2.2 Solar Charge Controller

*Solar Charge Controller* adalah peralatan elektronik yang digunakan untuk mengatur arus searah yang diisi ke baterai dan diambil dari baterai ke beban. Solar charge controller mengatur overcharging (kelebihan pengisian - karena baterai sudah 'penuh') dan kelebihan voltase dari solar module. Kelebihan voltase dan pengisian akan mengurangi umur baterai. Charge controller menerapkan teknologi Pulse width modulation (PWM) untuk mengatur fungsi pengisian baterai dan pembebasan arus dari baterai ke beban. Solar module 12 Volt umumnya memiliki tegangan output 16 - 21 Volt. Jadi tanpa solar charge controller, baterai akan rusak oleh over-charging dan ketidakstabilan tegangan. (Junial Heri, 2011)

Beberapa fungsi detail dari solar charge controller adalah sebagai berikut:

- Mengatur Mengatur arus untuk pengisian ke baterai, menghindari overcharging, dan overvoltage.
- Arus yang dibebaskan/ diambil dari baterai agar baterai tidak 'full discharge', dan overloading.
- Monitoring temperatur baterai Untuk membeli solar charge controller yang harus diperhatikan adalah: Voltage 12 Volt DC / 24 Volt DC
- Kemampuan (dalam arus searah) dari controller. Misalnya 5 Ampere, 6 Ampere, 10 Ampere, dsb.
- Full charge dan low voltage cut



**Gambar 2.2** *Solar Charge Controller*

(Sumber : <http://www.smartclima.com/solar-charge-controller-pwm.htm>)

### 2.3 Battery

*Battery* adalah alat yang menyimpan daya yang dihasilkan oleh panel surya yang tidak segera digunakan oleh beban. Daya yang disimpan dapat digunakan saat periode radiasi matahari rendah atau pada malam hari. Komponen baterai kadang-kadang dinamakan akumulator (*accumulator*). *Battery* menyimpan listrik dalam bentuk daya kimia. *Battery* yang paling biasa digunakan dalam aplikasi surya adalah *Battery* yang bebas pemeliharaan bertimbang asam (*maintenance-free lead-acid batteries*), yang juga dinamakan baterai recombinant atau VRLA (klep pengatur asam timbal atau *valve regulated lead acid*).

*Battery* memenuhi dua tujuan penting dalam sistem *fotovoltaik*, yaitu untuk memberikan daya listrik kepada sistem ketika daya tidak disediakan oleh array panel-panel surya, dan untuk menyimpan kelebihan daya yang ditimbulkan oleh panel-panel setiap kali daya itu melebihi beban. Baterai tersebut mengalami proses siklis menyimpan dan mengeluarkan, tergantung pada ada atau tidak adanya sinar matahari. Selama waktu adanya matahari, array panel menghasilkan daya listrik. Daya yang tidak digunakan dengan segera dipergunakan untuk mengisi baterai. Selama waktu tidak adanya matahari, permintaan daya listrik disediakan oleh baterai, yang oleh karena itu akan mengeluarkannya. (Junial Heri, 2011)



**Gambar 2.3** Battery

(Sumber : <https://privatmarket.ua/product/vision-12v-100ah-6fm100p-x>)

## 2.4 Inverter

*Inverter* adalah suatu alat yang dapat digunakan untuk mengubah tegangan masukan dc menjadi tegangan keluaran ac. Keluaran inverter dapat berupa tegangan yang dapat diatur dan tegangan yang tetap. Sumber tegangan masukan inverter dapat menggunakan baterai, tenaga surya, atau sumber tegangan dc yang lain. Tegangan keluaran yang biasa dihasilkan adalah 120 V, 220 V dan 115 V. (Sukmawidjaja. 2006)



**Gambar 2.4** *Inverter*

(Sumber : <http://www.dcac-power-inverter.com/grid%20tie%20inverter.html>)

## 2.5. Mikrokontroler

Mikrokontroler sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi kebutuhan pasar (market need) dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara masal (dalam jumlah banyak) membuat harganya menjadi lebih murah dibandingkan mikroprosesor.

Mikrokontroller memiliki banyak kelebihan daripada mikroprosesor. Adapun kelebihan dari mikrokontroller adalah sebagai berikut :

1. Penggerak pada mikrokontoler menggunakan bahasa pemrograman assembly dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem.

2. Mikrokontroler tersusun dalam satu chip dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan kontrol sistem.
3. Sistem running bersifat berdiri sendiri tanpa tergantung dengan komputer. Sedangkan parameter komputer hanya digunakan untuk download perintah instruksi atau program.
4. Pada mikrokontroler tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.
5. Harga untuk memperoleh alat ini lebih murah dan mudah didapat.

### 2.1.1. Mikrokontroler ATmega328

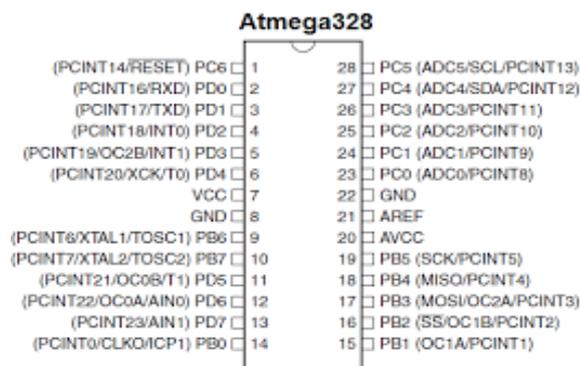
Mikrokontroler adalah sebuah komputer kecil (“special purpose computers”) di dalam satu IC yang berisi CPU, memori, timer, saluran komunikasi serial dan parallel, Port input/output, ADC. Mikrokontroler digunakan untuk suatu tugas dan menjalankan suatu program. (Andrianto, 2013)



**Gambar 2.5** Arduino Uno

(Sumber : [http://www.geeetech.com/wiki/index.php/Arduino\\_Uno](http://www.geeetech.com/wiki/index.php/Arduino_Uno))

### 2.1.2. Konfigurasi Pin ATmega328



**Gambar 2.6.** Konfigurasi Pin ATmega328

(Sumber : <http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller>)

ATMega328 memiliki 28 Pin, yang masing-masing pinnya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki ATmega8 yaitu sebagai berikut :

1. VCC  
Merupakan supply tegangan digital.
2. GND  
Merupakan ground untuk semua komponen yang membutuhkan grounding.
3. 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. Sebagai input, pin-pinyang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika pull-up resistor diaktifkan. Khusus PB6 dapat 8 digunakan sebagai input Kristal (inverting oscillator amplifier) dan input ke rangkaian clock internal, bergantung pada pengaturan Fuse bit yang digunakan untuk memilih sumber clock. Sedangkan untuk PB7 dapat digunakan sebagai output Kristal (output oscillator amplifier) bergantung pada pengaturan Fuse bit yang digunakan untuk memilih sumber clock. Jika sumber clock yang dipilih dari oscillator internal, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan Asynchronous Timer/Counter2 maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran input timer.
4. 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) ataupun mengeluarkan arus (source).

5. 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. Dan jika level tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsaminimum, maka akan menghasilkan suatu kondisi reset meskipun clock-nya tidak bekerja.

6. Port D (PD7...PD0)

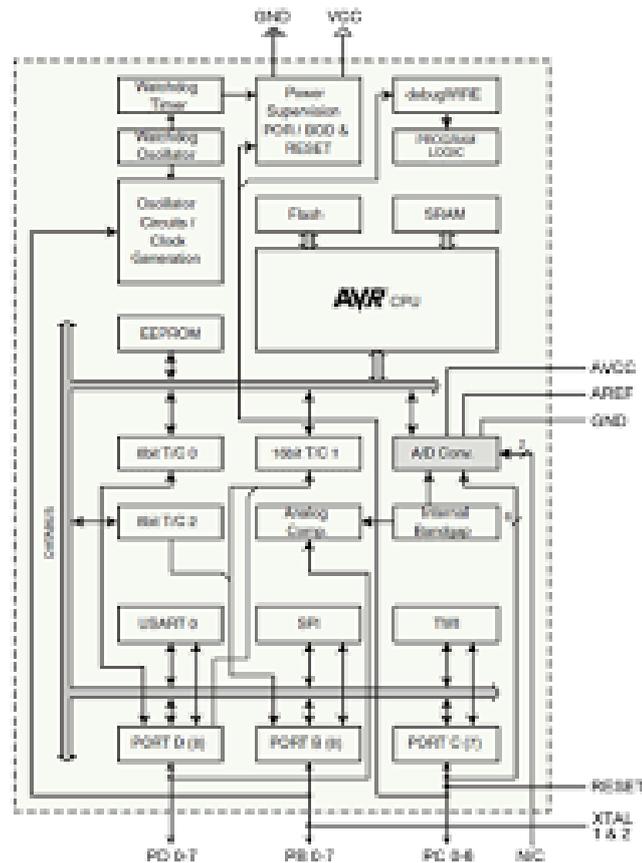
Port D merupakan 8-bit bi-directional I/O dengan internal pull-up resistor. Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada port ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

7. 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. Bahkan jika ADC pada AVR tidak digunakan tetap saja 9 disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui low pass filter.

8. AREF

Merupakan pin referensi jika menggunakan ADC



**Gambar 2.7.** Blok Diagram ATmega328

(Sumber : <http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller>)

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 ALU (*Arithmetic Logic Unit*) hal tersebut seperti yang tertulis dalam datasheet khususnya pada bagian *Instruction Set Reference*. Dalam hal ini untuk beberapa kasus dapat membuang penggunaan kebutuhan instruksi perbandingan yang telah didedikasikan serta 10 dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. Register ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal tersebut harus dilakukan melalui software.

## 2.6. LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama, selain itu LCD juga dapat digunakan untuk menampilkan karakter ataupun gambar. (Murethania, 2011)



**Gambar 2.8.** Liquid Crystal Display

( Sumber : <http://www.instructables.com/id/how-to-interface-LCD-16X2-to-arduino/>)

Modul LCD memiliki karakteristik sebagai berikut:

1. Terdapat 16 x 2 karakter huruf yang bisa ditampilkan.
2. Setiap terdiri dari 5 x 7 dot-matrix cursor.
3. Terdapat 192 macam karakter.
4. Terdapat 80 x 8 bit display RAM ( maksimal 80 karakter ).
5. Memiliki kemampuan penulisan dengan 8 bit maupun dengan 4 bit.
6. Dibangun oleh osilator lokal.
7. Satu sumber tegangan 5 Volt.
8. Otomatis reset saat tegangan dihidupkan.
9. Bekerja pada suhu 0oC sampai 55oC

### 2.6.1. Konfigurasi Pin LCD

**Tabel 2.1** Konfigurasi PIN pada LCD

PIN	Nama	Fungsi
1	Vss	GND
2	Vcc	+5v
3	Vee	LCD
4	RS	1 = Input data, 0 = Input Intruksi
5	R/W	1 = Read, 0 = Write
6	E	Enable
7	D0	Data 0

8	D1	Data 1
9	D2	Data 2
10	D3	Data 3
11	D4	Data 4
12	D5	Data 5
13	D6	Data 6
14	D7	Data 7
15	VBL+	4 – 4.2 volt
16	VBL-	GND

Dalam modul LCD terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan register, memori yang digunakan adalah:

- **DDRAM** (*Display Data Random Access Memory*) merupakan memori tempat karakter yang ditampilkan
- **CGRAM** (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat berubah-ubah sesuai dengan keinginan.
- **CGROM** (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD.

## 2.7 Sensor Arus

Sensor arus adalah suatu komponen elektronika yang berfungsi untuk mendeteksi besar arus listrik yang mengalir. Salah satu jenis sensor arus adalah ACS712. Sensor arus ACS712 menggunakan metode *hall effect* sensor. *Hall effect* sensor bekerja dengan hukum fisika dimana sensor yang digunakan bekerja dengan mendeteksi medan magnet.

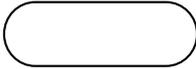
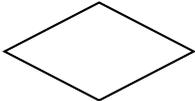


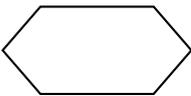
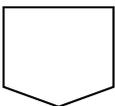
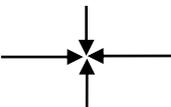
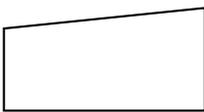
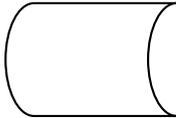
( Sumber : <https://electronilab.co/tienda/modulo-sensor-de-corriente-ac712-30/>)

## 2.8 Flowchart

*Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman. (Adelia, 2011)

**Tabel 2.2** Simbol - Simbol *Flowchart*

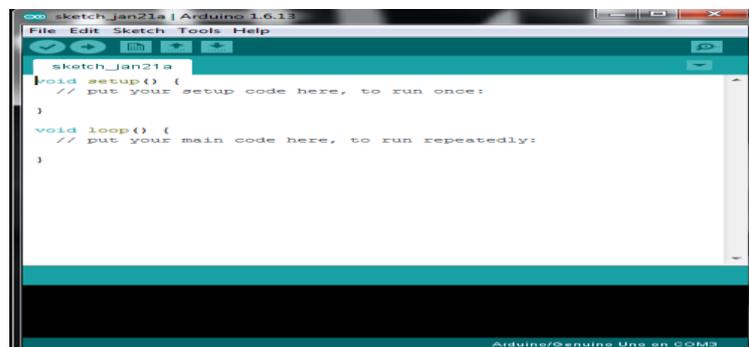
No	Simbol	Fungsi
1	Terminal 	Simbol untuk memulai dan mengakhiri suatu program
2	Proses 	Simbol untuk menyatakan suatu tindakan (proses) yang dilakukan oleh computer
3	Manual Operator 	Simbol untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
4	Input – Output 	Simbol untuk menyatakan proses input atau output tanpa tergantung jenis peralatannya
5	<i>Decision</i> 	Simbol untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak

6	<p><i>Predefined Process</i></p> 	Simbol untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan didalam storage
7	<p><i>Connector</i></p> 	Simbol untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
8	<p><i>Off Line Connector</i></p> 	Simbol untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
9	<p>Arus atau <i>Flow</i></p> 	Garis untuk menghubungkan arah tujuan simbol flowchart yang satu dengan yang lainnya
10	<p><i>Manual Input</i></p> 	Simbol untuk memasukkan data secara manual dengan menggunakan on-line keyboard
11	<p><i>Punched Card</i></p> 	Simbol untuk menyatakan input berasal dari kartu atau output ditulis ke kartu
12	<p><i>Document</i></p> 	Simbol untuk mencetak keluaran dalam bentuk dokumen (melalui printer)
13	<p><i>Disk Storage</i></p> 	Simbol untuk menyatakan input berasal dari disk atau output disimpan ke disk

## 2.9 Arduino Development Environment

Arduino *Development Environment* terdiri dari editor teks untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah *toolbar* dengan tombol- tombol untuk fungsi yang umum dan beberapa menu. *Arduino Development Environment* terhubung ke arduino board untuk meng-*upload* program dan juga untuk berkomunikasi dengan arduino board.

Perangkat lunak yang ditulis menggunakan *Arduino Development Environment* disebut *sketch*. *Sketch* ditulis pada editor teks. *Sketch* disimpan dengan file berekstensi .ino. area pesan memberikan informasi dan pesan error ketika kita menyimpan atau membuka *sketch*. Konsol menampilkan output teks dari *Arduino Development Environment* dan juga menampilkan pesan error ketika kita mengkompilasi *sketch*. Pada sudut kanan bawah jendela *Arduino Development Environment* menunjukkan jenis board dan port serial yang sedang digunakan. Tombol *toolbar* digunakan untuk mengecek dan meng-*upload sketch*, membuat, membuka, atau menyimpan *sketch*, dan menampilkan serial monitor. Lingkungan *open-source Arduino* memudahkan untuk menulis kode dan meng-*upload* ke *board Arduino*. Ini berjalan pada Windows, Mac OS X, dan Linux. Berdasarkan Pengolahan, avr-gcc, dan perangkat lunak sumber terbuka lainnya. *Arduino Development Environment* dapat dilihat pada gambar 2.9.



**Gambar 2.9** Tampilan Arduino IDE

Berikut ini merupakan tombol-tombol *toolbar* serta fungsinya :



**Gambar 2.10** Tombol *Verify*

Tombol *Verify* berfungsi untuk mengecek error pada kode program.



**Gambar 2.11** Tombol *Upload*

Tombol *Upload* berfungsi untuk meng-*compile* dan meng-*upload* program ke Arduino board.



**Gambar 2.12** Tombol *New*

Tombol *New* berfungsi untuk membuat *sketch* baru.



**Gambar 2.13** Tombol *Open*

Tombol *Open* berfungsi untuk menampilkan sebuah menu dari seluruh *sketch* yang berada di dalam *sketchbook*.



**Gambar 2.14** Tombol *Save*

Tombol *Save* berfungsi untuk menyimpan *sketch*.

### 2.8.1 Tipe-Tipe Data dalam Arduino

Setiap bagoan dari data yang anda simpan dalam program Arduino memiliki tipe datanya masing-masing. Tergantung pada kebutuhan anda, anda dapat memilih dari tipe-tipe data berikut ini :

1. Tipe data boolean mengambil satu byte memori dan dapat bernilai benar atau salah.
2. Tipe data char mengambil satu byte nomor memori dan menyimpan dari 1 sampai 127. Angka-angka ini biasanya mewakili karakter yang dikodekan dalam ASCII.
3. Tipe data int (integer) membutuhkan dua byte memori. Anda dapat menggunakannya untuk menyimpan angka dari -32.768 ke 32.767. unsigned int juga menghabiskan dua byte memori tetapi menyimpan angka dari 0 sampai 65.535.

4. Untuk angka yang lebih besar, digunakan tipe data long. Mengonsumsi empat byte memori dan menyimpan nilai dari -214783648 ke 2147483647. Unsigned long juga perlu empat byte tetapi menyimpan rentang nilai dari 0 sampai 4.294.967.295.
5. Tipe data float dan double adalah tipe data yang sama. Anda dapat menggunakan jenis tipe ini untuk menyimpan angka floating-point. Keduanya menggunakan empat byte memori dan mampu menyimpan nilai-nilai dari -3.4028235E+38 untuk 3.4028235E+38.
6. Tipe data void hanya untuk deklarasi fungsi. Ini menunjukkan bahwa fungsi tersebut tidak mengembalikan nilai.
7. Array menyimpan nilai yang memiliki tipe data yang sama.
8. Sebuah string adalah sebuah array nilai char.

## 2.10 Bahasa C

Bahasa C adalah bahasa pemrograman yang dapat dikatakan berada di antara bahasa beraras rendah dan beraras tinggi. Bahasa beraras rendah artinya bahasa yang berorientasi pada mesin dan beraras tinggi berorientasi pada manusia. Bahasa beraras rendah, misalnya bahasa *assembler*, bahasa ini ditulis dengan sandi yang dimengerti oleh mesin saja, oleh karena itu hanya digunakan bagi yang memprogram mikroprosesor. Bahasa beraras rendah merupakan bahasa yang membutuhkan kecermatan yang teliti bagi pemrogram karena perintahnya harus rinci, ditambah lagi masing-masing pabrik mempunyai sandi perintah sendiri. Bahasa tinggi relatif mudah digunakan, karena ditulis dengan bahasa manusia sehingga mudah dimengerti dan tidak tergantung mesinnya. Bahasa beraras tinggi biasanya digunakan pada komputer.

Pencipta bahasa C adalah Brian W. Kernighan dan Denis M. Ritchi, sekitar tahun 1972. Penulisan program dalam bahasa C dilakukan dengan membagi dalam blok-blok, sehingga bahasa C disebut dengan bahasa terstruktur. Bahasa C dapat digunakan di berbagai mesin dengan mudah, mulai dari PC sampai dengan mainframe, dengan berbagai sistem operasi misalnya DOS, UNIX, VMS dan lain-lain.

Program Bahasa C tidak mengenal aturan penulisan di kolom tertentu, jadi bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak dibaca. Berikut contoh penulisan Program Bahasa C:

```
#include <at89c51.h>
main ()
{ .....
.....
}
```

Program dalam bahasa C selalu berbentuk fungsi seperti ditunjukkan dalam **main ()**. Program yang dijalankan berada di dalam tubuh program yang dimulai dengan tanda kurung buka { dan diakhiri dengan tanda kurung tutup }. Semua yang tertulis di dalam tubuh program ini disebut dengan blok. Tanda () digunakan untuk mengapit **argumen** suatu fungsi. Argumen adalah suatu nilai yang akan digunakan dalam fungsi tersebut. Dalam tubuh fungsi antara tanda { dan tanda } ada sejumlah pernyataan yang merupakan perintah yang harus dikerjakan oleh prosesor. Setiap pernyataan diakhiri dengan tanda titik koma ; Baris pertama **#include <...>** bukanlah pernyataan, sehingga tak diakhiri dengan tanda titik koma (;). Baris tersebut meminta kompiler untuk menyertakan file yang namanya ada di antara tanda <...> dalam proses kompilasi. File-file ini (ber ekstensi .h) berisi deklarasi fungsi ataupun variable. File ini disebut **header**. File ini digunakan semacam perpustakaan bagi pernyataan yang ada di tubuh program. (Heryanto, 2008)