

BAB II
TEORI PENUNJANG

2.1. Penghematan Energi Listrik

Penggunaan saklar otomatis di ruang baca merupakan salah satu cara operasi yang digunakan untuk mengendalikan beban listrik. Ide penggunaan saklar otomatis ini muncul sebagai upaya menghindari pemborosan energi listrik. Selain itu juga saklar otomatis dapat memudahkan operasi.

Dari segi ekonomis, dengan memasang saklar otomatis, maka keborosan energi listrik dapat dihindari. Penggunaan energi listrik menjadi terkontrol. Sebagai contoh, bila seseorang lupa mematikan lampu penerangan 40 Watt dalam ruangan selama 5 jam, maka akan terjadi pemborosan energi listrik sebesar :

$$E = P \times t \dots\dots\dots (2.1)$$

Dimana :

- W = Energi listrik dalam satuan (kWh)
- P = Daya listrik dalam satuan (Watt)
- t = Waktu pemakaian dalam satuan (jam)

Maka :

$$\begin{aligned} W &= P \times t \\ &= 40 \times 5 \\ &= 200 \text{ Wh (sehari)} \end{aligned}$$

Dalam sebulan :

$$\begin{aligned} W &= 30 \times 200 \text{ Wh} \\ &= 6000 \text{ Wh} \\ &= 6 \text{ kWh} \end{aligned}$$

Jika biaya per kWh misalkan Rp. 500,00 maka biaya yang harus dibayar adalah $\text{Rp.}500,00 \times 6 = 3000,00/\text{bulan}$

2.2. *Smart Building*

Perkembangan teknologi melaju dengan sangat pesat. Perkembangan teknologi ini merupakan hasil kerja keras dari rasa ingin tahu manusia terhadap suatu hal yang pada akhirnya diharapkan akan mempermudah kehidupan manusia. Salah satu cara untuk mempermudah kehidupan manusia adalah dengan membangun *smart building*.

Smart building sudah menjadi salah satu topik pembahasan dalam karya tulis fiksi ilmiah selama bertahun-tahun, namun baru mulai dicoba untuk diwujudkan pada awal abad ke 20 ketika pemakaian energi listrik telah meluas hingga ke rumah-rumah dan mulai majunya teknologi informasi.

Smart building mengacu pada penggunaan teknologi informasi dan komputer untuk mengendalikan peralatan yang berada di dalam bangunan tersebut (seperti jedela atau lampu). Sistem yang beropersi dalam *smart building* dapat berupa sistem kontrol jarak jauh yang sederhana dari lampu-lampu hingga sebuah sistem yang berbasis komputer atau mikrokontroler yang memiliki tingkat kecerdasan yang bervariasi yang

secara otomatis mengontrol peralatan yang berada di dalam bangunan.

Elemen dalam sebuah sistem *smart building* terdiri dari sensor (seperti sensor cahaya atau sensor suhu), pengontrol (seperti komputer atau mikrokontroler) dan aktuator (seperti motor, *relay*, *servo* atau sakelar lampu). Sebuah antarmuka antara manusia dengan mesin diperlukan, agar pemilik bangunan dapat berinteraksi dengan sistem untuk memonitor atau mengontrolnya.

Antarmuka ini dapat berupa terminal khusus atau aplikasi yang berjalan pada telepon genggam atau komputer tablet. Antarmuka tersebut berkomunikasi dengan sistem melalui jaringan kabel atau *wireless* menggunakan satu atau lebih protokol.

2.3. Arduino

Arduino adalah platform pembuatan prototipe elektronik yang bersifat *open-source hardware* yang berdasarkan pada perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan. Arduino ditujukan bagi para seniman, desainer, dan siapapun yang tertarik dalam menciptakan objek atau lingkungan yang interaktif.

Arduino pada awalnya dikembangkan di Ivrea, Italia. Nama *arduino* adalah sebuah nama maskulin yang berarti teman yang kuat. Platform *arduino* terdiri dari *arduino board*, *shield*, bahasa pemrograman *arduino*, dan *arduino development environment*. *Arduino board* biasanya memiliki sebuah *chip* dasar mikrokontroler Atmel AVR ATmega8 berikut turunannya. Blok diagram *arduino board* yang sudah disederhanakan dapat

dilihat pada Gambar 2.1. *Shield* adalah sebuah papan yang dapat dipasang di atas *arduino board* untuk menambah kemampuan dari *arduino board*.

Bahasa pemrograman *arduino* adalah bahasa pemrograman yang umum digunakan untuk membuat perangkat lunak yang ditanamkan pada *arduino board*. Bahasa pemrograman *arduino* mirip dengan bahasa pemrograman C++.



Gambar 2.1 : Blok diagram *arduino board*

Arduino Development Environment adalah perangkat lunak yang digunakan untuk menulis dan meng-*compile* program untuk *arduino*. *Arduino Development Environment* juga digunakan untuk meng-*upload* program yang sudah di-*compile* ke memori program *arduino board*.

2.3.1. Arduino Uno

Arduino Uno adalah *arduino board* yang menggunakan mikrokontroler ATmega328. *Arduino Uno* memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog, sebuah 16 MHz osilator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah header ICSP, dan sebuah tombol reset. *Arduino Uno* memuat segala hal yang dibutuhkan untuk mendukung sebuah mikrokontroler. Hanya dengan menghubungkannya ke sebuah komputer melalui USB atau

memberikan tegangan DC dari baterai atau adaptor AC ke DC sudah dapat membuanya bekerja. Arduino Uno menggunakan ATmega16U2 yang diprogram sebagai *USB-to-serial converter* untuk komunikasi serial ke computer melalui port USB. Tampak atas dari arduino uno dapat dilihat pada Gambar 2.2.

Adapun data teknis *board* Arduino Uno R3 adalah sebagai berikut:

1. Mikrokontroler : ATmega328
2. Tegangan Operasi : 5V
3. Tegangan Input (recommended) : 7 - 12 V
4. Tegangan Input (limit) : 6-20 V
5. Pin digital I/O : 14 (6 diantaranya pin PWM)
6. Pin Analog input : 6
7. Arus DC per pin I/O : 40 mA
8. Arus DC untuk pin 3.3 V : 150 mA
9. Flash Memory : 32 KB dengan 0.5 KB digunakan untuk bootloader
10. SRAM : 2 KB
11. EEPROM : 1 KB
12. Kecepatan Pewaktuan : 16 Mhz

The image features a large, semi-transparent blue circular logo of Universitas Medan Area in the background. The logo contains the text 'UNIVERSITAS' at the top and 'MEDAN AREA' at the bottom, with a central emblem depicting a building and a star.

Gambar 2.2 : Arduino uno

2.3.1.1. Pin Masukan dan Keluaran Arduino Uno

Masing-masing dari 14 pin digital arduino uno dapat digunakan sebagai masukan atau keluaran menggunakan fungsi `pinMode()`, `digitalWrite()` dan `digitalRead()`. Setiap pin beroperasi pada tegangan 5 volt. Setiap pin mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki resistor *pull-up* internal (diputus secara default) sebesar 20-30 KOhm. Sebagai tambahan, beberapa pin masukan digital memiliki kegunaan khusus yaitu:

1. Komunikasi serial: pin 0 (RX) dan pin 1 (TX), digunakan untuk menerima(RX) dan mengirim(TX) data secara serial.
2. External Interrupt: pin 2 dan pin 3, pin ini dapat dikonfigurasi untuk memicu sebuah interrupt pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
3. Pulse-width modulation (PWM): pin 3,5,6,9,10 dan 11, menyediakan keluaran PWM 8-bit dengan menggunakan fungsi

analogWrite().

4. Serial Peripheral Interface (SPI): pin 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), pin ini mendukung komunikasi SPI dengan menggunakan SPI library.
5. LED: pin 13, terdapat built-in LED yang terhubung ke pin digital 13. Ketika pin bernilai HIGH maka LED menyala, sebaliknya ketika pin bernilai LOW maka LED akan padam.

Arduino Uno memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap pin menyediakan resolusi sebanyak 10 bit (1024 nilai yang berbeda). Secara default pin mengukur nilai tegangan dari ground (0V) hingga 5V, walaupun begitu dimungkinkan untuk mengganti nilai batas atas dengan menggunakan pin AREF dan fungsi analog Reference(). Sebagai tambahan beberapa pin masukan analog memiliki fungsi khusus yaitu pin A4 (SDA) dan pin A5 (SCL) yang digunakan untuk komunikasi *Two Wire Interface (TWI)* atau *Inter Integrated Circuit (I2C)* dengan menggunakan Wire library.

2.3.1.2. Sumber Daya dan Pin Tegangan Arduino Uno

Arduino uno dapat diberi daya melalui koneksi USB (Universal Serial Bus) atau melalui power supply eksternal. Jika arduino uno dihubungkan ke kedua sumber daya tersebut secara bersamaan maka arduino uno akan memilih salah satu sumber daya secara otomatis untuk digunakan. Power supply external (yang bukan melalui USB) dapat berasal dari adaptor AC ke DC atau baterai. Adaptor dapat dihubungkan ke soket power pada arduino uno. Jika menggunakan baterai, ujung kabel

yang dibubungkan ke baterai dimasukkan kedalam pin GND dan Vin yang berada pada konektor Power.

Arduino uno dapat beroperasi pada tegangan 6 sampai 20 Volt. Jika arduino uno diberi tegangan di bawah 7 Volt, maka pin 5V akan menyediakan tegangan di bawah 5 Volt dan arduino uno mungkin bekerja tidak stabil. Jika diberikan tegangan melebihi 12 Volt, penstabil tegangan kemungkinan akan menjadi terlalu panas dan merusak arduino uno. Tegangan rekomendasi yang diberikan ke arduino uno berkisar antara 7 sampai 12 Volt.

Pin-pin tegangan pada arduino uno adalah sebagai berikut:

1. Vin adalah pin untuk mengalirkan sumber tegangan ke arduino uno ketika menggunakan sumber daya eksternal (selain dari koneksi USB atau sumber daya yang teregulasi lainnya). Sumber tegangan juga dapat disediakan melalui pin ini jika sumber daya yang digunakan untuk arduino uno dialirkan melalui soket *power*.
2. 5V adalah pin yang menyediakan tegangan teregulasi sebesar 5 Volt berasal dari *regulator* tegangan pada arduino uno.
3. 3V3 adalah pin yang menyediakan tegangan teregulasi sebesar 3,3 Volt berasal dari *regulator* tegangan pada arduino uno.
4. GND adalah pin *ground*.

2.3.1.3. Peta Memori Arduino Uno

Arduino Uno adalah arduino *board* yang menggunakan mikrokontroler ATmega328. Maka peta memori arduino uno sama dengan peta memori pada mikrokontroler ATmega328.

2.3.1.3.1. Memori Program

ATMega328 memiliki 32K byte *On-chip In-System Reprogrammable Flash Memory* untuk menyimpan program. Memori *flash* dibagi kedalam dua bagian, yaitu bagian program *bootloader* dan aplikasi seperti terlihat pada Gambar 2.3. *Bootloader* adalah program kecil yang bekerja pada saat sistem dimulai yang dapat memasukkan seluruh program aplikasi ke dalam memori prosesor.



Gambar 2.3 : Peta memori program ATMega 328

2.3.1.3.2. Memori Data

Memori data ATMega328 terbagi menjadi 4 bagian, yaitu 32 lokasi untuk register umum, 64 lokasi untuk register I/O, 160 lokasi untuk register I/O tambahan dan sisanya 2048 lokasi untuk data SRAM internal. Register umum menempati alamat data terbawah, yaitu 0x0000 sampai 0x001F. Register I/O menempati 64 alamat berikutnya mulai dari 0x0020 hingga 0x005F. Register I/O tambahan menempati 160 alamat berikutnya mulai dari 0x0060 hingga 0x00FF. Sisa alamat berikutnya mulai dari 0x0100 hingga 0x08FF digunakan untuk SRAM internal. Peta memori data dari ATMega 328 dapat dilihat pada Gambar 2.4.



Gambar 2.4 : Peta memori data atmega 328

2.3.1.3.3. Memori Data EEPROM

Arduino uno terdiri dari 1 KByte memori data EEPROM. Pada memori EEPROM, data dapat ditulis/dibaca kembali dan ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM dimulai dari 0x000 hingga 0x3FF.

2.3.2. Arduino Ethernet Shield

Ethernet Shield menambah kemampuan arduino board agar terhubung ke jaringan komputer. Ethernet shield berbasis cip ethernet Wiznet W5100. Ethernet *library* digunakan dalam menulis program agar arduino board dapat terhubung ke jaringan dengan menggunakan arduino ethernet shield.

Pada ethernet shield terdapat sebuah slot micro-SD, yang dapat digunakan untuk menyimpan file yang dapat diakses melalui jaringan.

Onboard micro-SD card reader diakses dengan menggunakan *SD library*.

Arduino board berkomunikasi dengan W5100 dan SD card menggunakan bus SPI (Serial Peripheral Interface). Komunikasi ini diatur oleh library SPI.h dan Ethernet.h. Bus SPI menggunakan pin digital 11, 12 dan 13 pada Arduino Uno. Pin digital 10 digunakan untuk memilih W5100 dan pin digital 4 digunakan untuk memilih SD card. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk input/output umum ketika kita menggunakan ethernet shield.

Karena W5100 dan SD card berbagi bus SPI, hanya salah satu yang dapat aktif pada satu waktu. Jika kita menggunakan kedua perangkat dalam program kita, hal ini akan diatasi oleh *library* yang sesuai. Jika kita tidak menggunakan salah satu perangkat dalam program kita, kiranya kita perlu secara eksplisit men-*deselect*-nya. Untuk melakukan hal ini pada SD card, set pin 4 sebagai output dan menuliskan logika tinggi padanya, sedangkan untuk W5100 yang digunakan adalah pin 10. DFRduino Ethernet shield adalah sebuah *clone* dari arduino Ethernet shield yang dibuat oleh DFRobot. Penampakan DFRduino Ethernet shield dapat dilihat pada Gambar 2.5.

Gambar 2.5 : Ethernet shield

2.3.3. Bahasa Pemrograman Arduino

Arduino board merupakan perangkat yang berbasis mikrokontroler. Perangkat lunak (*software*) merupakan komponen yang membuat sebuah mikrokontroler dapat bekerja. Arduino board akan bekerja sesuai dengan perintah yang ada dalam perangkat lunak yang ditanamkan padanya. Bahasa Pemrograman Arduino adalah bahasa pemrograman utama yang digunakan untuk membuat program untuk arduino board. Bahasa pemrograman arduino menggunakan bahasa pemrograman C sebagai dasarnya.

Karena menggunakan bahasa pemrograman C sebagai dasarnya, bahasa pemrograman arduino memiliki banyak sekali kemiripan, walaupun beberapa hal telah berubah.

2.3.3.1. Struktur

Setiap program dalam arduino board terdiri dari dua fungsi utama yaitu `setup()` dan `loop()`. Instruksi yang berada dalam fungsi `setup()` dieksekusi hanya sekali, yaitu ketika arduino board pertama kali dihidupkan. Biasanya instruksi yang berada pada fungsi `setup()` merupakan konfigurasi dan inisialisasi dari arduino board. Instruksi yang berada pada fungsi `loop()` dieksekusi berulang-ulang hingga arduino board dimatikan (catu daya diputus). Fungsi `loop()` merupakan tugas utama dari arduino board. Jadi setiap program yang menggunakan bahasa pemrograman arduino memiliki struktur sebagai berikut:

Program di atas dapat dianalogikan dalam bahasa C sebagai berikut:

2.3.3.2. Konstanta

Konstanta adalah variable yang sudah ditetapkan sebelumnya dalam bahasa pemrograman arduino. Konstanta digunakan agar program lebih mudah untuk dibaca dan dimengerti. Konstanta dibagi menjadi 3 kelompok yaitu:

1. Konstanta yang digunakan untuk menunjukkan tingkat logika (konstanta Boolean), yaitu *true* dan *false*
2. Konstanta untuk menunjukkan keadaan pin, yaitu HIGH dan LOW
3. Konstanta untuk menunjukkan fungsi pin, yaitu INPUT, INPUT_PULLUP dan OUTPUT

Konstanta yang digunakan untuk menunjukkan benar atau salah dalam bahasa pemrograman arduino adalah *true* dan *false*. *False* lebih mudah didefinisikan daripada *true*. *False* didefinisikan sebagai 0(nol). *True* sering didefinisikan sebagai 1(satu), yang mana hal ini benar, tetapi *true* memiliki definisi yang lebih luas. Setiap integer yang bukan nol adalah *true* dalam pengertian Boolean. Jadi -2, 3 dan -100 semuanya didefinisikan sebagai *true*, juga dalam pengertian Boolean. Tidak seperti konstanta yang lain *true* dan *false* diketik dengan menggunakan huruf kecil.

Ketika membaca atau menulis ke sebuah pin digital, terdapat hanya dua nilai yang dapat diberikan atau diterima, yaitu HIGH dan LOW. HIGH memiliki arti yang berbeda tergantung apakah sebuah pin dikonfigurasi menjadi masukan atau keluaran. Ketika pin dikonfigurasi sebagai masukan dengan fungsi `pinMode()`, lalu kemudian dibaca dengan fungsi `digitalRead()`, mikrokontroler akan melaporkan nilai HIGH jika tegangan yang ada pada pin tersebut berada pada tegangan 3 volt atau lebih.

Ketika sebuah pin dikonfigurasi sebagai masukan, dan kemudian dibuat bernilai HIGH dengan fungsi `digitalWrite()`, maka resistor *pull-up* internal dari chip ATmega akan aktif, yang akan membawa pin masukan ke nilai HIGH kecuali pin tersebut ditarik (*pull-down*) ke nilai LOW oleh sirkuit dari luar.

Ketika pin dikonfigurasi sebagai keluaran dengan fungsi `pinMode()`, dan diset ke nilai HIGH dengan fungsi `digitalWrite()`, maka pin berada pada tegangan 5 volt. Dalam keadaan ini, pin tersebut dapat

memberikan arus, sebagai contoh, untuk menghidupkan LED yang terhubung seri dengan resistor dan ground, atau pin lain yang dikonfigurasi sebagai keluaran dan diberi nilai LOW.

Sama seperti HIGH, LOW juga memiliki arti yang berbeda bergantung pada konfigurasi pin. Ketika pin dikonfigurasi sebagai masukan, maka mikrokontroler akan melaporkan nilai LOW jika tegangan yang terdapat pada pin berada pada tegangan 2 volt atau kurang. Ketika pin dikonfigurasi sebagai keluaran dan diberi nilai LOW maka pin berada pada tegangan 0 volt.

Setiap pin pada arduino dapat dikonfigurasi sebagai masukan, masukan dengan resistor *pull-up* atau keluaran. Untuk mengkonfigurasi fungsi pin pada arduino digunakan konstanta INPUT, INPUT_PULLUP dan OUTPUT. Pin arduino yang dikonfigurasi sebagai masukan dengan fungsi pinMode() dikatakan berada dalam kondisi berimpedansi tinggi. Pin yang dikonfigurasi sebagai masukan memiliki permintaan yang sangat kecil kepada sirkuit yang di-sampling-nya, setara dengan sebuah resistor 100 Megaohm dipasang seri dengan pin tersebut. Hal ini membuat pin tersebut berguna untuk membaca sensor, tetapi tidak untuk menghidupkan sebuah LED.

Cip ATmega pada arduino memiliki resistor *pull-up* internal (resistor yang terhubung ke sumber tegangan secara internal) yang dapat digunakan. Untuk menggunakan resistor *pull-up* internal ini kita menggunakan konstanta INPUT_PULLUP pada fungsi pinMode().

Pin yang dikonfigurasi menjadi sebuah keluaran dikatakan berada dalam kondisi berimpedansi rendah. Hal ini berarti pin tersebut dapat menyediakan sejumlah besar arus ke sirkuit yang lain. Pin pada ATmega mampu menyediakan arus hingga 40 mA.

2.3.3.3. Fungsi Masukan dan Keluaran Digital

Arduino memiliki 3 fungsi untuk masukan dan keluaran digital pada arduino board, yaitu `pinMode()`, `digitalWrite()` dan `digitalRead()`.

Fungsi `pinMode()` mengkonfigurasi pin tertentu untuk berfungsi sebagai masukan atau keluaran. Sintaksis untuk fungsi `pinMode()` adalah sebagai berikut:

`pinMode(pin, mode)`

Parameter: pin = angka dari pin digital yang akan dikonfigurasi

mode = konfigurasi yang diinginkan (INPUT, INPUT_PULLUP dan OUTPUT).

Fungsi `digitalWrite()` berfungsi untuk memberikan nilai HIGH atau LOW suatu digital pin. Sintaksis untuk fungsi `digitalWrite()` adalah sebagai berikut:

`digitalWrite(pin, value)`

Parameter: pin = angka dari pin digital yang akan dikonfigurasi value = nilai yang diinginkan (HIGH atau LOW).

Fungsi `digitalRead()` bertujuan untuk membaca nilai yang ada pada pin arduino uno. Sintaksis untuk fungsi `digitalRead()` adalah sebagai berikut:

digitalRead(pin)

Parameter: pin = angka dari pin digital yang akan dibaca

Berikut ini adalah contoh penggunaan fungsi masukan dan keluaran digital dalam sebuah program:

```
int ledPin = 13;    // LED terhubung ke pin digital 13
int inPin = 7;     // pushbutton terhubung ke pin digital 7
int val = 0;      // variable untuk menyimpan sebuah nilai

void setup()
{
  pinMode(ledPin, OUTPUT); // set pin digital 13 sebagai keluaran
  pinMode(inPin, INPUT);  // set pin digital 13 sebagai masukan
}

void loop()
{
  val = digitalRead(inPin); // baca nilai pin input
  digitalWrite(ledPin, val); // sets LED sesuai dengan nilai val
}
```

2.3.3.4. Progmem

Terkadang program dibuat memerlukan memori yang lebih besar, misalnya kita perlu menyimpan banyak teks dalam sebuah program penampil LCD. Hal ini bisa kita atasi dengan menyimpan data pada flash memori (memori program). Besar memori program pada arduino uno adalah 32 KByte.

Kata kunci **PROGMEM** adalah sebuah *variable modifier*,

PROGMEM hanya bisa digunakan untuk tipe data yang di definisikan pada *library pgmspace.h*. PROGMEM menjelaskan pada kompiler “letakkan informasi berikut pada memori program”, menggantikan SRAM, dimana biasanya informasi tersebut disimpan.

PROGMEM adalah bagian dari *library pgmspace.h*, jadi kita perlu memasukkan *library pgmspace.h* ke program kita. Sintaksis untuk menggunakan PROGMEM adalah sebagai berikut:

```
dataType variableName[] PROGMEM = {dataInt0, dataInt1, ...};
```

dataType – tipe variable

variableName – nama array data

Karena PROGMEM adalah sebuah *variable modifier*, tidak ada aturan yang baku untuk tempat peletaknya. Kompiler arduino menerima semua difinisi yang ditunjukkan di bawah.

```
dataType variableName[] PROGMEM
```

```
= {}; dataType PROGMEM
```

```
variableName[] = {}; PROGMEM
```

```
dataType variableName[] = {};
```

PROGMEM juga bisa digunakan untuk menyimpan satu variabel, tetapi karena kerepotan dalam penggunaannya, sebaiknya hanya digunakan untuk menyimpan data yang sangat banyak, yang biasanya lebih mudah menyimpannya di dalam array.

Dalam menggunakan PROGMEM kita perlu melakukan 2 langkah prosedur. Pertama adalah menyimpan data pada memori program, kemudian kita menggunakan fungsi khusus, yang juga didefinisikan di

library pgmspace.h, untuk membaca data tersebut dari memori program dan menuliskannya ke SRAM, sehingga kita dapat melakukan sesuatu dengan data tersebut.

Berikut adalah tipe data yang didukung oleh *library* pgmspace.h:

prog_char a signed char (1 byte) -127 sampai 128

prog_uchar - an unsigned char (1 byte) 0 sampai 255

prog_int16_t - a signed int (2 bytes) -32,767 sampai 32,768

prog_uint16_t - an unsigned int (2 bytes) 0 sampai 65,535

prog_int32_t - a signed long (4 bytes) -2,147,483,648 sampai *
2,147,483,647.

prog_uint32_t - an unsigned long (4 bytes) 0 sampai 4,294,967,295



Berikut ini adalah contoh program untuk menggunakan PROGMEM

```
#include <avr/pgmspace.h>

// simpan beberapa unsigned int

PROGMEM

M      prog_uint16_t charSet[]      = { 65000, 32796, 16843, 10,
11234};

// simpan beberapa karakter
prog_uchar signMessage[] PROGMEM = {"UNIVERSITAS SUMATERA
UTARA"}
;
unsigned int displayInt;
int k;      // variabel counter
char myChar;
// baca int
displayInt = pgm_read_word_near(charSet + k)
// baca karakter
myChar = pgm_read_byte_near(signMessage + k);
```

Di bawah ini adalah contoh program untuk menyimpan array string pada program memori. Karena string sendiri adalah sebuah array, contoh di bawah ini juga merupakan contoh penggunaan array dua dimensi.

```
#include <avr/pgmspace.h>

prog_char string_0[] PROGMEM = "String 0";
prog_char string_1[] PROGMEM = "String 1";
prog_char string_2[] PROGMEM = "String 2";
```

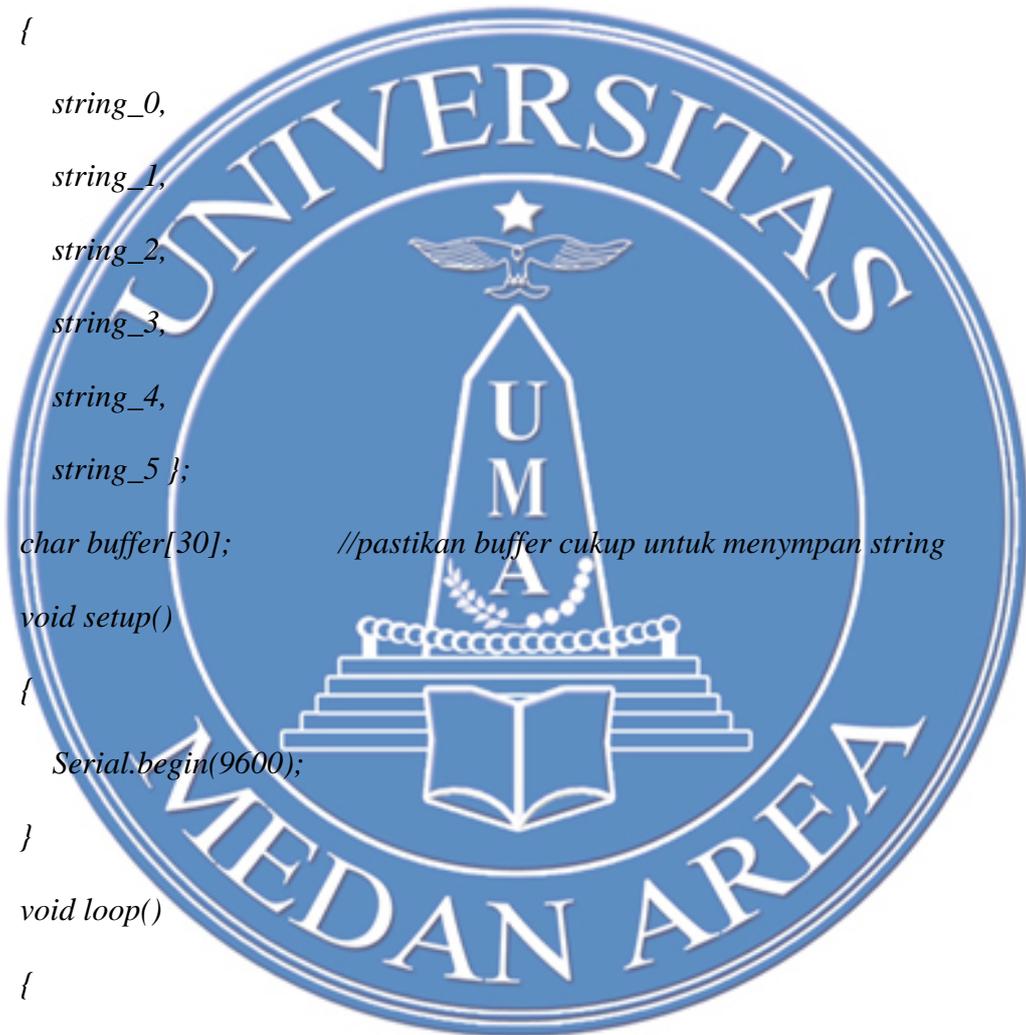
```

prog_char string_3[] PROGMEM = "String 3";
prog_char string_4[] PROGMEM = "String 4";
prog_char string_5[] PROGMEM = "String 5";

//buat table untuk menunjukkan string.
PROGMEM const char *string_table[] =
{
  string_0,
  string_1,
  string_2,
  string_3,
  string_4,
  string_5 };
char buffer[30]; //pastikan buffer cukup untuk menyimpan string
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  /*menggunakan table string pada memori program membutuhkan
fungsi special untuk mengambil data tersebut.

  fungsi strcpy_P ("buffer") menyalin sebuah string dari program
memori sebuah string di SRAM ("buffer").

```



Pastikan bahwa SRAM cukup untuk menampung setiap string yang diambil dari memori program*/

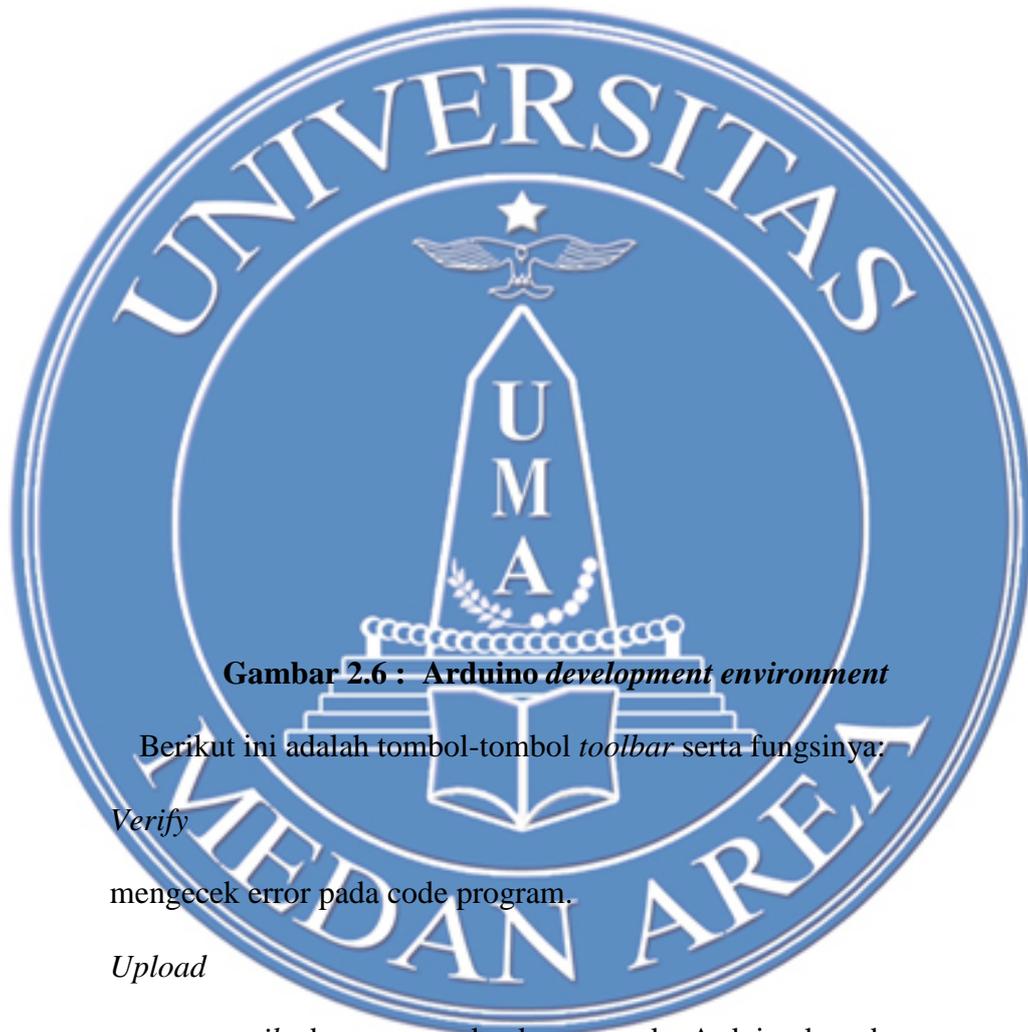
```
for (int i = 0; i < 6; i++)  
{  
    strcpy_P(buffer, (char*)pgm_read_word(&(string_table[i]))); // salin  
string ke buffer  
  
    Serial.println( buffer ); delay(  
500 );  
}  
}
```

2.3.4. 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 dari jendela *Arduino Development Environment* menunjukkan jenis board dan port serial yang sedang digunakan. Tombol *toolbar* digunakan untuk mengecek dan

mengupload *sketch*, membuat, membuka atau menyimpan *sketch*, dan menampilkan *serial monitor*.



Gambar 2.6 : Arduino *development environment*

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

Verify

mengecek error pada code program.

Upload

meng-*compile* dan meng-*upload* program ke Arduino board.

New

membuat *sketch* baru.

Open

menampilkan sebuah menu dari seluruh *sketch* yang berada di dalam

sketchbook.

Save

menyimpan sketch.

Serial Monitor

membuka serial

monitor.

Dalam lingkungan arduino digunakan sebuah konsep yang disebut *sketchbook*, yaitu tempat standar untuk menampung program (*sketch*). *Sketch* yang ada pada *sketchbook* dapat dibuka dari menu **File > Sketchbook** atau dari tombol *open* pada *toolbar*. Ketika pertama kali menjalankan arduino development environment, sebuah direktori akan dibuat secara otomatis untuk tempat penyimpanan *sketchbook*. Kita dapat melihat atau mengganti lokasi dari direktori tersebut dari menu **File > Preferences**.

Serial monitor menampilkan data serial yang sedang dikirim dari arduino board. Untuk mengirim data ke board, masukkan teks dan klik tombol *send* atau tekan *enter* pada *keyboard*.

Sebelum meng-*upload* program, kita perlu mensetting jenis board dan port serial yang sedang kita gunakan melalui menu **Tools > Board** dan **Tools > Serial Port**. Pemilihan board berguna untuk mengeset parameter (contohnya: kecepatan mikrokontroler dan *baud rate*) yang digunakan ketika meng-*compile* dan meng-*upload sketch*.

Setelah memilih board dan port serial yang tepat, tekan tombol *upload* pada *toolbar* atau pilih menu **File > Upload**. Arduino board akan me-*reset* secara otomatis dan proses *upload* akan dimulai. Pada kebanyakan board, LED RX dan

TX akan berkedip ketika program sedang di-*upload*. *Arduino development environment* akan menampilkan pesan ketika proses *upload* telah selesai, atau menampilkan pesan error.

Ketika sedang meng-*upload* program, arduino bootloader sedang digunakan, Arduino bootloader adalah sebuah program kecil yang telah ditanamkan pada mikrokontroler yang berada pada arduino board. Bootloader ini memungkinkan kita meng-*upload* program tanpa menggunakan perangkat keras tambahan.

2.4. Sensor

Sensor adalah alat yang digunakan untuk mendeteksi dan sering berfungsi untuk mengukur magnitudo sesuatu. Dengan menggunakan sensor kita dapat mengubah mekanis, magnetis, panas, cahaya dan kimia menjadi tegangan dan arus listrik. Sensor yang digunakan pada sistem kontrol intensitas penerangan lampu pada ruang baca ini adalah sensor *Infra Red*.

2.4.1. Led Infra Red

Infra merah merupakan gelombang cahaya yang tak tampak oleh mata manusia. Panjang gelombangnya berada pada rentang 700 nm sampai 1 mm. Panjang gelombang ini lebih panjang daripada panjang gelombang cahaya tampak, tetapi tidak lebih panjang daripada panjang gelombang radio. Sumber cahaya infra merah hampir sama dengan sumber cahaya tampak yaitu *LED*, tetapi bahan pembuatnya berbeda. *LED* infra merah dibuat dari bahan gallium arsenide (GaAs). *LED* infra merah hanya memerlukan tegangan masukan sebesar 3 Volt. Berikut adalah Gambar 2.7 yang menampilkan bentuk fisik dari *led* infra merah :



Gambar 2.7 : *Led* infra merah

2.4.2. Fotodioda

Fotodioda adalah komponen elektronik yang terbuat dari bahan semikonduktor dan dapat mengubah intensitas cahaya menjadi arus listrik dan berikut bentuk fisiknya yang ditunjukkan pada Gambar 2.8 berikut :



Gambar 2.8 : Fotodioda

Jika diperhatikan, skematik *LED* dan fotodioda hampir sama. Perbedaan hanya terletak pada arah panahnya saja. Pada skematik *LED*, arah panahnya keluar, yang dapat diartikan bahwa *LED* memancarkan cahaya. Pada skematik fotodioda, arah panahnya masuk, yang dapat diartikan bahwa fotodioda menyerap atau memproses cahaya.

Seperti pada *LED*, fotodioda juga memiliki dua kaki. Kaki yang lebih panjang adalah anoda, dan kaki yang lebih pendek adalah katoda. Pemasangan komponen fotodioda pada rangkaian elektronik berkebalikan dengan pemasangan

LED. Kutub positif atau anoda fotodioda dihubungkan dengan kutub negatif sumber tegangan, dan kutub negatif atau katoda fotodioda dihubungkan dengan kutub positif sumber tegangan.

2.5. Lampu Pijar

Lampu pijar adalah sumber cahaya buatan yang dihasilkan melalui penyaluran arus listrik melalui filamen yang kemudian memanaskan dan menghasilkan cahaya. Kaca yang menyelubungi filamen panas tersebut menghalangi udara untuk berhubungan dengannya sehingga filamen tidak akan langsung rusak akibat teroksidasi. Lampu pijar dipasarkan dalam berbagai macam bentuk dan tersedia untuk tegangan (voltase) kerja yang bervariasi dari mulai 1,25 volt hingga 300 volt. Energi listrik yang diperlukan lampu pijar untuk menghasilkan cahaya yang terang lebih besar dibandingkan dengan sumber cahaya buatan lainnya seperti lampu pendar dan dioda cahaya, maka secara bertahap pada beberapa negara peredaran lampu pijar mulai dibatasi. Di samping memanfaatkan cahaya yang dihasilkan, beberapa penggunaan lampu pijar lebih memanfaatkan panas yang dihasilkan, contohnya adalah pemanas kandang ayam, dan pemanas inframerah dalam proses pemanasan di bidang industri.

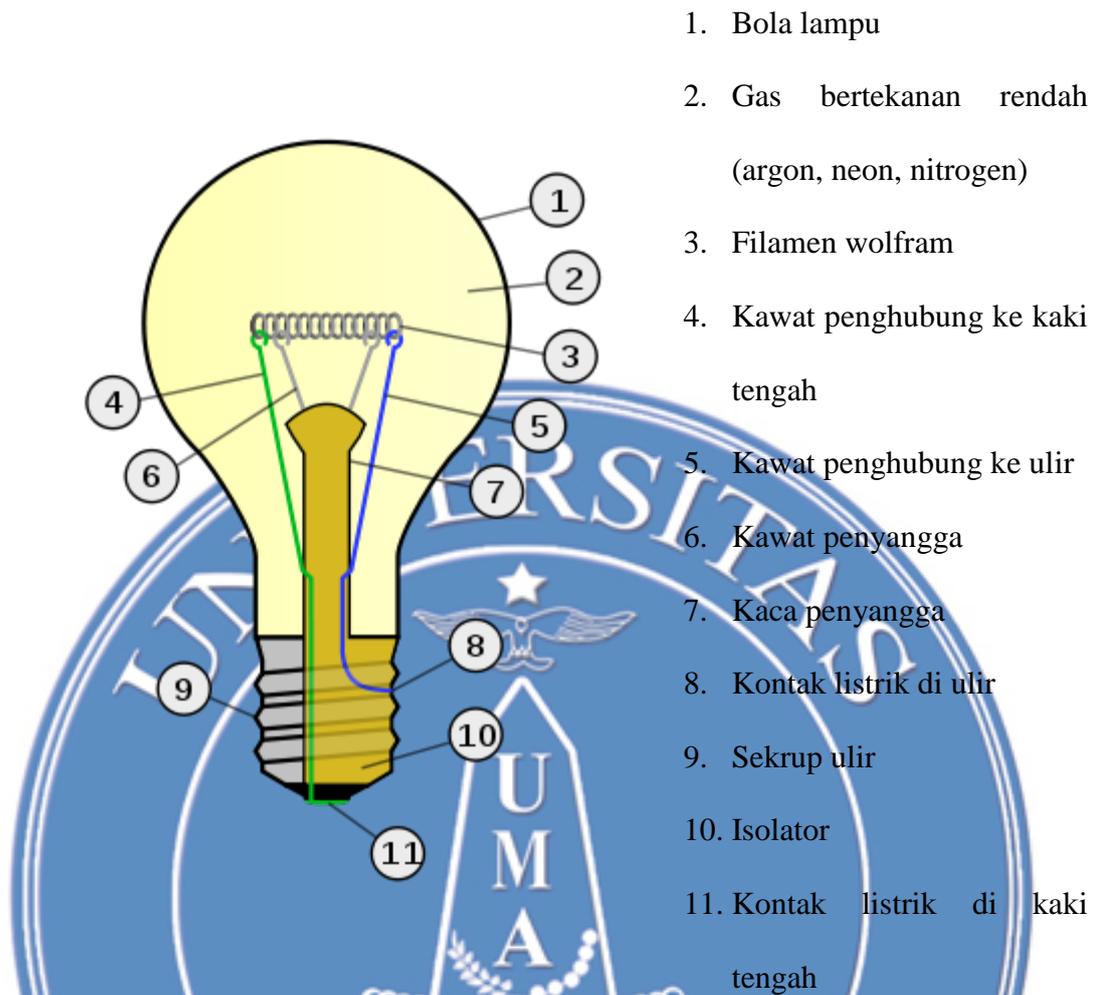
Pengembangan lampu pijar sudah dimulai pada awal abad XIX. Sejarah lampu pijar dapat dikatakan telah dimulai dengan ditemukannya tumpukan volta oleh Alessandro Volta. Pada tahun 1802, Sir Humphry Davy menunjukkan bahwa arus listrik dapat memanaskan seuntai logam tipis hingga menyala putih. Lalu, pada tahun 1820, Warren De la Rue merancang sebuah lampu dengan cara menempatkan sebuah kumparan logam mulia platina di dalam sebuah tabung lalu

mengalirkan arus listrik melaluinya. Hanya saja, harga logam platina yang sangat tinggi menghalangi pendayagunaan penemuan ini lebih lanjut. Elemen karbon juga sempat digunakan, namun karbon dengan cepat dapat teroksidasi di udara; oleh karena itu, jawabannya adalah dengan menempatkan elemen dalam vakum. Pada tahun 1870-an, seorang penemu bernama Thomas Alva Edison dari Menlo Park, negara bagian New Jersey, Amerika Serikat, mulai ikut serta dalam usaha merancang lampu pijar. Dengan menggunakan elemen platina, Edison mendapatkan paten pertamanya pada bulan April 1879.

Rancangan ini relatif tidak praktis namun Edison tetap berusaha mencari elemen lain yang dapat dipanaskan secara ekonomis dan efisien. Di tahun yang sama, Sir Joseph Wilson Swan juga menciptakan lampu pijar yang dapat bertahan selama 13,5 jam. Sebagian besar filamen lampu pijar yang diciptakan pada saat itu putus dalam waktu yang sangat singkat sehingga tidak berarti secara komersial.^[2] Untuk menyelesaikan masalah ini, Edison kembali mencoba menggunakan untaian karbon yang ditempatkan dalam bola lampu hampa udara hingga pada tanggal 19 Oktober 1879 dia berhasil menyalakan lampu yang mampu bertahan selama 40 jam.

2.5.1. Konstruksi

Komponen utama dari lampu pijar adalah bola lampu yang terbuat dari kaca, filamen yang terbuat dari wolfram, dasar lampu yang terdiri dari filamen, bola lampu, gas pengisi, dan kaki lampu. Berikut ini adalah Gambar 2.9, yang menjelaskan tentang bola lampu :



Gambar 2.9 : Bola lampu dan keterangannya

2.5.2. Bola Lampu

Selubung gelas yang menutup rapat filamen suatu lampu pijar disebut dengan bola lampu. Macam-macam bentuk bola lampu antara lain adalah bentuk bola, bentuk jamur, bentuk lilin, dan bentuk *lustre*. Warna bola lampu antara lain yaitu bening, warna susu atau buram, dan warna merah, hijau, biru, atau kuning.

2.5.3. Operasi

Pada dasarnya filamen pada sebuah lampu pijar adalah sebuah resistor. Saat dialiri arus listrik, filamen tersebut menjadi sangat panas, berkisar antara 2800 derajat Kelvin hingga maksimum 3700 derajat Kelvin. Ini menyebabkan

warna cahaya yang dipancarkan oleh lampu pijar biasanya berwarna kuning kemerahan. Pada temperatur yang sangat tinggi itulah filamen mulai menghasilkan cahaya pada panjang gelombang yang kasatmata. Hal ini sejalan dengan teori radiasi benda hitam. Indeks renderasi warna menyatakan apakah warna obyek tampak alami apabila diberi cahaya lampu tersebut dan diberi nilai antara 0 sampai 100. Angka 100 artinya warna benda yang disinari akan terlihat sesuai dengan warna aslinya. Indeks renderasi warna lampu pijar mendekati 100. Berikut adalah Gambar 2.10, yang menunjukkan bentuk kawat filamen lampu pijar.



Gambar 2.10 : Kawat filamen

2.6. AC/DC Adaptor

Sebagian besar piranti elektronika membutuhkan tegangan *DC* untuk bekerja. Meskipun baterai berguna dalam piranti yang bisa dibawa-bawa atau piranti berdaya rendah, akan tetapi waktu operasinya terbatas. Sumber daya yang mudah dapat dibuat dari sebuah rangkaian yang dapat mengubah tegangan *AC* menjadi tegangan *DC*. Sebuah adaptor dapat dibuat dengan tiga buah komponen utama, yaitu transformer, dioda penyearah, dan kapasitor *filter*. Transformator penurun tegangan adalah transformator yang diperlukan untuk menurunkan tegangan primer yang tinggi misalnya sebesar 220 Volt atau 380 Volt, menjadi tegangan yang lebih rendah pada bagian sekundernya, 6 Volt, 9 Volt, 12 Volt,

atau 24 Volt. Berikut adalah Gambar 2.11 yang memperlihatkan bentuk fisik AC-DC adaptor :



Gambar 2.11 : AC-DC Adaptor

2.7. Dasar - dasar ISP (*In-System Chip Programming*)

In-System Chip Programming (ISP) adalah sebuah fitur bagi sebuah *microcontroller* agar dapat didownload dengan program tanpa mencabut *microcontroller* dari system-nya. Sehingga *Microcontroller* tetap pada kedudukannya semua dan dihubungkan dengan *ISP*. Dan dilakukan proses download. Begitu pula saat memutuskan hubungan antara *Downloader* dan *Microcontroller*, kita hanya cukup memutuskan kabelnya saja, tanpa lagi perlu mencucuk-cabut mikrokontroler. Cara semacam ini adalah cara yang sangat hemat waktu terutama dalam proses pengembangan sebuah program.

In-System Chip Programming (ISP) buatan *ATMEL* adalah sebuah komunikasi serial yang menggunakan bus *SPI (Serial Peripheral Interface)* yang menggunakan *Shift Register* sebagai komponen utamanya. Ada 2 kabel data yang disebut sebagai *MISO (Master In Slave Out)*, dan *MOSI (Master Out Slave In)*. Sesuai dengan namanya jika *ISP* ini adalah sebuah *downloader* dan *microcontroller*, maka dapat dikatakan Master adalah *downloader* dan *Slave* adalah *Microcontroller*. *MOSI* adalah kabel yang mengirimkan data kepada *Slave*,

sedang *MISO* kabel tempat master menerima data. Transfer data tersebut memerlukan sebuah kabel lagi, yang dinamakan sebagai *shynchronization*. Dalam hal ini kabel tersebut dinamakan dengan *SCK (Serial Clock)*. Data (*MISO* dan atau *MOSI*) akan dianggap valid hanya saat *SCK* dalam keadaan tinggi. Berikut ini adalah Gambar 2.12 yang menampilkan bentuk fisiknya :



Gambar 2.12 : USB – AT ISP programmer download adapter