

# **MODUL PRAKTIKUM**

## **SISTEM TERTANAM**



**DEPARTEMEN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS SUMATERA UTARA**

**2020**

## DAFTAR ISI

|   | Hal |
|---|-----|
| DAFTAR ISI  | 1   |
| <b>MODUL TEORI</b>  |     |
| BAB I    PENGANTAR TEKNOLOGI MIKROKONTROLER                               | 2   |
| BAB II    MIKROKONTROLER PLATFORM ARDUINO                                 | 7   |
| BAB III    PEMROGRAMAN ARDUINO  | 15  |
| <b>MODUL PRAKTIKUM</b>  |     |
| 1        Percobaan ADC dan Pengatur Kecerahan LED                         | 19  |
| 2        Percobaan LDR (Light Dependent Resistor)                         | 20  |
| 3        Percobaan Input Digital dengan Push Button                       | 21  |
| 4        Percobaan LCD (Liquid Crystal Display)                           | 22  |
| 5        Percobaan Motor Servo  | 23  |
| 6        Percobaan Sensor Ultrasonik HC-SR04                              | 24  |
| 7        Percobaan Read/Write Data pada EEPROM                            | 25  |
| 8        Percobaan Komunikasi SPI (Serial Peripheral Interface)           | 28  |
| 9        Percobaan Komunikasi I <sup>2</sup> C (Inter-Integrated Circuit) | 31  |
| 10       Lampiran   | 33  |

# BAB I

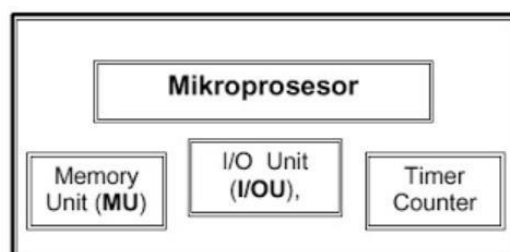
## PENGANTAR TEKNOLOGI MIKROKONTROLER

### A. Sejarah Mikrokontroler

Sejarah awal perkembangan mikrokontroler dimulai pada awal tahun 1970- an. Motorola mengeluarkan seri mikrokontroler 6800 yang terus dikembangkan menjadi 68HC05, 68HC08, 68HC11, 68HC12, dan 68HC16. Zilog juga mengeluarkan seri mikroprosesor Z80-nya yang terkenal dan terus dikembangkan hingga kini menjadi Z180 dan kemudian diadopsi juga oleh mikroprosesor Rabbit. Tahun 1974, Texas Instrument mengenalkan mikrokontroler dengan seri TMS 1000 yang merupakan mikrokontroler 4 bit. TMS 1000 adalah mikrokomputer dalam sebuah chip, lengkap dengan RAM dan ROM. Tahun 1976, Intel mengeluarkan mikrokontroler 8-bit seri Intel 8748 yang merupakan mikrokontroler dari keluarga MCS 48. Intel juga mengeluarkan mikrokontroler-nya yang populer di dunia yaitu

8051, yang karena begitu populernya maka arsitektur 8051 tersebut kemudian diadopsi oleh vendor lain seperti Phillips, Siemens, Atmel, dan vendor-vendor lain. Selain itu masih ada mikrokontroler populer lainnya seperti Basic Stamps, PIC dari Microchip, MSP 430 dari Texas Instrument dan masih banyak lagi.

Mikrokontroler (*microcontroller*) atau disingkat dengan “*micron*” adalah pengendali yang merupakan suatu komputer kecil yang terletak di dalam sebuah chip atau IC (*integrated circuit*) yang berisikan inti prosesor, memori, dan komponen input/output yang dapat diprogram. Mikrokontroler biasa digunakan pada produk dan perangkat yang dapat dikontrol secara otomatis, seperti sistem kontrol mesin mobil (*engine control*), perangkat medis (*medical devices*), pengendali jarak jauh (*remote control*), mesin perkantoran (*office machines*), dan juga mainan (*games*). Penggunaan mikrokontroler lebih ekonomis dibandingkan sebuah desain sistem yang berisikan mikroprosesor, memori, dan perangkat input/ouput terpisah. Mikrokontroler adalah komputer mikro dalam satu chip tunggal. Mikrokontroler memadukan CPU, ROM, RWM, I/O paralel, I/O seri, counter-timer, dan rangkaian clock dalam satu chip tunggal seperti terlihat pada Gambar 1.1.



Gambar 1.1. Blok Diagram Mikrokontroller

Sama halnya dengan mikroprosesor, mikrokontroler adalah perangkat yang dirancang untuk kebutuhan umum (*specific purpose*). Sesuai dengan fungsinya sebagai pengendali, mikrokontroler berisikan sepaket chip lengkap yang terdiri dari fitur-fitur pengolah data yang juga terdapat dalam mikroprosesor, ditambah RAM, ROM, I/O, dan fitur lain yang terintegrasi di dalamnya. Contohnya dapat ditemui pada perangkat otomotif, mesin industri, elektronik dan perangkat- perangkat lain yang memiliki embedded sistem di dalamnya. Mikrokontroler sebagai sebuah chip telah mengalami perkembangan baik dari sisi arsitektur, teknologi dan kemampuannya. Untuk melihat perbedaan konsep diantara mikroprosesor dan mikrokontroler di bawah ini ditunjukkan Tabel 1.1 perbandingan konfigurasi, arsitektur, dan set instruksi diantara mikroprosesor Z-80 CPU dengan mikrokontroler AT89C51.

Tabel 1.1. Perbedaan Mikroprosesor dan Mikrokontroler

| No  | Konfigurasi Pin      | Mikroprosesor<br>Z-80 CPU | Mikrokontroler<br>AT89C51 |
|-----|----------------------|---------------------------|---------------------------|
| 1.  | Jumlah pin           | 40                        | 40                        |
| 2.  | Jumlah pin alamat    | 16                        | 16                        |
| 3.  | Jumlah pin data      | 8                         | 8                         |
| 4.  | Jumlah pin interrupt | 2                         | 2                         |
| 5.  | Pin I/O              | -                         | 32                        |
| 6.  | Register 8 bit       | 20                        | 34                        |
| 7.  | Register 16 bit      | 4                         | 2                         |
| 8.  | Ukuran Stack         | 64 K                      | 128                       |
| 9.  | ROM Internal         | -                         | 4 Kbyte                   |
| 10. | RAM Internal         | -                         | 128 byte                  |
| 11. | Memori eksternal     | 64 Kbyte                  | 128 Kbyte                 |
| 12. | Flag                 | 6                         | 4                         |
| 13. | Timer                | 0                         | 2                         |
| 14. | Port Paralel         | 0                         | 4 x 8 bit                 |
| 15. | Port Serial          | 0                         | 1                         |

Terdapat perbedaan yang signifikan antara mikrokontroler dan mikroprosesor. Perbedaan yang utama antara keduanya dapat dilihat dari dua faktor utama yaitu arsitektur perangkat keras (*hardware architecture*) dan aplikasi masing-masing.

- 1) Ditinjau dari segi arsitekturnya, mikroprosesor hanya merupakan *single chip* CPU, sedangkan mikrokontroler dalam IC-nya selain CPU juga terdapat device lain yang memungkinkan mikrokontroler berfungsi sebagai suatu *single chip* komputer. Dalam sebuah IC mikrokontroler telah terdapat ROM, RAM, EPROM, serial interface dan paralel interface, timer, interrupt controller, konverter Analog ke Digital, dan lainnya (tergantung feature yang melengkapi mikrokontroler tersebut).
- 2) Sedangkan dari segi aplikasinya, mikroprosesor hanya berfungsi sebagai *Central Processing Unit* yang menjadi otak komputer, sedangkan mikrokontroler, dalam bentuknya yang mungil, pada umumnya ditujukan untuk melakukan tugas-tugas yang berorientasi kendali (*control*) pada rangkaian yang membutuhkan jumlah komponen minimum dan biaya rendah (*low cost*).



- 3) **Mikrokontroler 16 Bit;** keterbatasan-keterbatasan yang ada pada mikrokontroler 8 bit berkaitan dengan semakin kompleknya pengolahan data dan pengendalian serta kecepatan tanggap/respon disempurnakan dengan menggunakan mikrokontroler 16 bit. Salah satu solusinya adalah dengan menaikkan kecepatan clock, dan ukuran data. Mikrokontroler 16 bit digunakan untuk mengatur tangan robot, dan aplikasi *Digital Signal Processing* (DSP).
- 4) **Mikrokontroler 32 Bit;** ditargetkan untuk aplikasi Robot, Instrumen cerdas, Avionics, Image Processing, Telekomunikasi, Automobil, dan sebagainya. Program-program aplikasinya bekerja dengan sistim operasi dan dipadukan dengan perangkat pintar lainnya.

Karena kebutuhan yang tinggi terhadap “*smart chip*” dengan berbagai fasilitasnya, maka berbagai vendor juga berlomba untuk menawarkan produk-produk mikrokontrolernya. Selain mikroprosesor dan mikrokontroler, sebenarnya telah bemunculan chip-chip pintar lain seperti DSP prosesor dan *Application Spesific Integrated Circuit* (ASIC). Di masa depan, chip-chip mungil berkemampuan sangat tinggi akan mendominasi semua desain elektronik di dunia sehingga mampu memberikan kemampuan komputasi yang tinggi serta meminimumkan jumlah komponen-komponen konvensional.

### C. Implementasi Teknologi Mikrokontroler

Karena kemampuannya yang tinggi, bentuknya yang kecil, konsumsi dayanya yang rendah, dan harga yang murah maka mikrokontroler begitu banyak digunakan di dunia. Mikrokontroler digunakan mulai dari mainan anak-anak, perangkat elektronik rumah tangga, perangkat pendukung otomotif, peralatan industri, peralatan telekomunikasi, peralatan medis dan kedokteran, sampai dengan pengendali robot serta persenjataan militer. Terdapat beberapa keunggulan yang diharapkan dari alat-alat yang berbasis mikrokontroler (*microcontroller-based solutions*) :

1. Kehandalan tinggi (*high reliability*) dan kemudahan integrasi dengan komponen lain (*high degree of integration*)
2. Ukuran yang semakin dapat diperkecil (*reduced in size*)
3. Penggunaan komponen dipersedikit (*reduced component count*) yang juga akan menyebabkan biaya produksi dapat semakin ditekan (*lower manufacturing cost*)
4. Waktu pembuatan lebih singkat (*shorter development time*) sehingga lebih cepat pula dijual ke pasar sesuai kebutuhan (*shorter time to market*)
5. Konsumsi daya yang rendah (*lower power consumption*)

Penerapan teknologi di masyarakat akan memberikan banyak keuntungan. Mikrokontroler merupakan sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan penggunaan mikrokontroler ini maka:

1. Sistem elektronik akan menjadi lebih ringkas,
2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi,
3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak,

Teknologi mikrokontroller dapat diterapkan di berbagai bidang, baik di industri maupun di masyarakat. Dengan menerapkan teknologi Mikrokontroller di masyarakat akan memberikan banyak keuntungan, contohnya :

- 1) Kunci Kombinasi Digital dengan Remote Kontrol Inframerah
- 2) Jam Wekker Digital Bertampilan M1632 LCD
- 3) Alat penyiram tanaman otomatis
- 4) Pemberi pakan ikan otomatis
- 5) Deteksi kebakaran menggunakan sensor suhu dan mikrokontroller
- 6) Pendeteksi kebocoran gas
- 7) Sistem keamanan mobil
- 8) Pintu Air Otomatis
- 9) Pengendali lampu rumah dengan mikrokontroller melalui SMS
- 10) Robot Cerdas

## BAB II

### MIKROKONTROLER PLATFORM ARDUINO

#### A. Konsep Arduino

Arduino merupakan papan-tunggal mikrokontroler serba guna yang bisa diprogram dan bersifat *open-source*. Platform Arduino sekarang ini menjadi sangat populer dengan penambahan jumlah pengguna baru yang terus meningkat. Hal ini karena kemudahannya dalam penggunaan dan penulisan kode program. Tidak seperti kebanyakan papan sirkuit pemrograman sebelumnya, Arduino tidak lagi membutuhkan perangkat keras terpisah (disebut *programmer* atau *downloader*) untuk memuat atau meng-*upload* kode baru ke dalam mikrokontroler. Cukup dengan menggunakan kabel USB untuk mulai menggunakan Arduino. Selain itu, Arduino IDE menggunakan bahasa pemrograman C++ dengan versi yang telah disederhanakan, sehingga lebih mudah dalam belajar pemrograman. Arduino akhirnya berhasil menjadi papan sirkuit pemrograman paling disukai hingga menjadikannya sebagai bentuk standar dari fungsi mikrokontroler dengan paket yang mudah untuk diakses.

Hardware dan software Arduino dirancang bagi para seniman, desainer, enthusiast, hacker, pemula dan siapapun yang tertarik untuk menciptakan objek interaktif dan pengembangan lingkungan. Arduino mampu berinteraksi dengan tombol, LED, motor, speaker, GPS, kamera, internet, ponsel pintar bahkan dengan televisi. Fleksibilitas ini dihasilkan dari kombinasi ketersediaan software Arduino yang gratis, papan perangkat keras yang murah, dan keduanya yang mudah untuk dipelajari. Hal inilah yang menciptakan jumlah pengguna menjadi sebuah komunitas besar dengan berbagai kontribusinya yang telah dirilis pada berbagai proyek dengan berbasiskan Arduino.

Arduino Uno merupakan versi terbaru dari keluarga Arduino, berbasis mikrokontroler ATmega328, menyempurnakan tipe sebelumnya, Duemilanove. Perbedaan Arduino tersebut adalah tidak menggunakan IC FTDI (*Future Technology Devices International*) *USB to Serial* sebagai driver komunikasi USB-nya tetapi menggunakan mikrokontroler ATmega8U2 yang diprogram sebagai konverter USB ke serial. Uno sendiri diambil dari bahasa Italia yang artinya 1 (satu). Gambar 2.1 adalah *board* Arduino Uno dengan spesifikasi hardware :

- ☐ Microcontroller : ATmega328
- ☐ Tegangan Operasi : 5 V
- ☐ Tegangan Input : 7 – 12V
- ☐ Digital I/O : 14 pin
- ☐ PWM : 6 channel
- ☐ Analog Input : 6 channel



- Memory : 32KB Flash PEROM (0,5 KB digunakan oleh *bootloader*), 2KB SRAM dan 1KB EEPROM
- Frekuensi Clock : 16 MHz



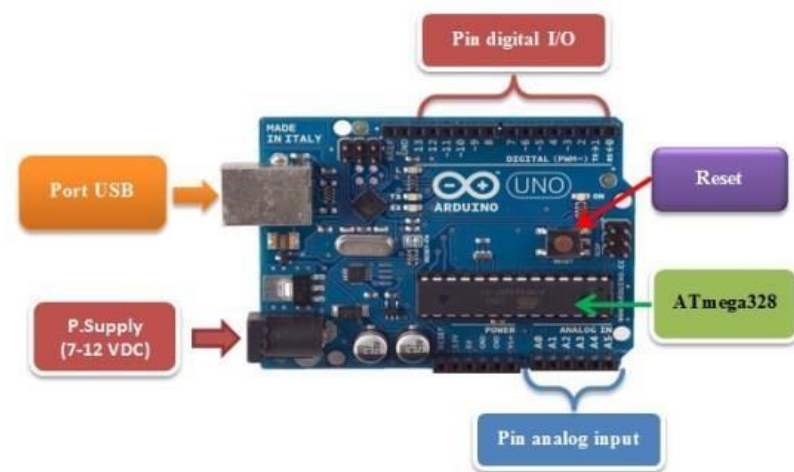
Gambar 2.1. Arduino Uno Board

Tabel 2.1. menunjukkan beberapa produk board yang diproduksi Arduino yang ada di pasaran :

Tabel 2.1 Produk Board Arduino

| No | Nama Board       | No | Nama Board                     |
|----|------------------|----|--------------------------------|
| 1  | Arduino Uno      | 14 | Lilypad Arduino                |
| 2  | Arduino Leonardo | 15 | Arduino NG (Nuova Generazione) |
| 3  | Arduino Due      | 16 | Arduino Extreme                |
| 4  | Arduino Yun      | 17 | Arduino Diecimila              |
| 5  | Arduino Tre      | 18 | Arduino Mega                   |
| 6  | Arduino Micro    | 19 | Arduino Mega 2560              |
| 7  | Arduino Robot    | 20 | Arduino Pro                    |
| 8  | Arduino Esplora  | 21 | Arduino Pro Mini               |
| 9  | Arduino Mega ADK | 22 | Arduino Ethernet               |
| 10 | Arduino Fio      | 23 | Arduino Duemilanove            |
| 11 | Arduino USB      | 24 | Arduino Shields                |
| 12 | Arduino Nano     | 25 | Arduino Singel Side Serial     |
| 13 | Arduino Serial   | 26 | Arduino Mini                   |

Gambar 2.2. memperlihatkan bagian utama dari papan Arduino uno, yakni terminal power supply, port USB, pin digital I/O, tombol reset, mikrokontroler ATmega328 dan pin analog input.



Gambar 2.2. Bagian Utama Arduino Uno Board

Berikut ini adalah penjelasan beberapa bagian utama dari papan Arduino uno, yaitu:

### 1. Power Supply.

Pada Arduino board, ada 2 (dua) pilihan sumber tegangan yang dapat digunakan, yakni dari port USB maupun dari power supply eksternal. Dengan menghubungkan port USB di komputer/laptop dengan Arduino maka secara otomatis *power supply* Arduino bersumber dari port USB. Untuk sumber tegangan eksternal (non-USB) yakni dengan menghubungkan Arduino board dengan sumber tegangan DC. Tegangan yang direkomendasikan adalah 7 sampai 12 V, jika kurang dari 7V akan menyebabkan ketidakstabilan tegangan, sedangkan jika lebih dari 12V akan menyebabkan panas dan akibat fatal berupa kerusakan pada board Arduino.

### 2. Input – Output.

Port Arduino berbeda penamaannya dengan sistem minimum mikrokontroler atau *microntroller development system*. Sebagai contoh pada system minimum ATmega8535 penamaan port adalah PORTA, PORTB, PORTC dan PORTD, untuk akses per-bit maka PORTA.0 s/d PORTA.7, contoh lain pada AT89S51 maka PORT0, PORT1 dan seterusnya. Sistem penamaan port pada Arduino merupakan urutan nomor port, mulai dari nomor nol (0), satu (1) dan seterusnya. Untuk digital I/O dengan nama pin 1, 2 sampai 13, sedangkan untuk analog input menggunakan nama A0, A1 sampai A5.

Pada Arduino uno terdapat 14 pin digital input – output. Secara umum berfungsi sebagai port input – output biasa, namun ada beberapa pin yang mempunyai fungsi alternatif. Sebagai contoh, pin 2 dan 3 juga sebagai pin interupsi eksternal. Kemudian pin 5,6,9,10 dan 11 dapat dipakai untuk PWM (*Pulse Width Modulation*) yang sering dipakai untuk kendali motor DC maupun motor servo. Tabel 4.2. menunjukkan nomor dan fungsi pin pada Arduino uno.

Tabel 2.2. Konfigurasi Pin Arduino Uno

| No Pin | Fungsi         | Fungsi alternatif         |
|--------|----------------|---------------------------|
| 0      | Digital I/O 0  | RX (serial – receiver)    |
| 1      | Digital I/O 1  | TX (serial – transmitter) |
| 2      | Digital I/O 2  | Interupsi external        |
| 3      | Digital I/O 3  | Interupsi external & PWM  |
| 4      | Digital I/O 4  | –                         |
| 5      | Digital I/O 5  | PWM                       |
| 6      | Digital I/O 6  | PWM                       |
| 7      | Digital I/O 7  | –                         |
| 8      | Digital I/O 8  | –                         |
| 9      | Digital I/O 9  | PWM                       |
| 10     | Digital I/O 10 | SPI – SS & PWM            |
| 11     | Digital I/O 11 | SPI – MOSI & PWM          |
| 12     | Digital I/O 12 | SPI – MISO                |
| 13     | Digital I/O 13 | SPI – SCK & LED           |

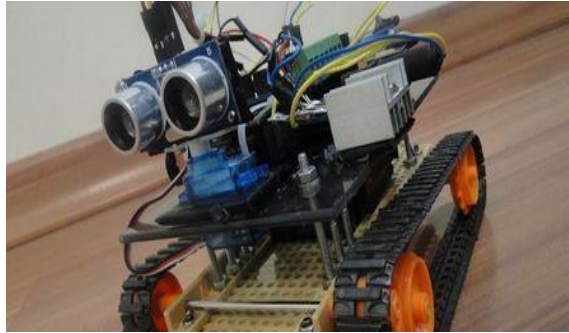
### 3. Analog Input.

Arduino memiliki 6 pin analog input, berfungsi membaca sinyal masukan analog seperti sensor analog. Meskipun demikian pin analog input dapat pula digunakan untuk keperluan digital I/O. Tabel 2.3. menunjukkan nomor dan fungsi pin input analog.

Tabel 2.3. Konfigurasi Pin Analog Input

| No pin | Fungsi         | Fungsi Alternatif |
|--------|----------------|-------------------|
| A0     | Analog Input 1 | –                 |
| A1     | Analog Input 2 | –                 |
| A2     | Analog Input 3 | –                 |
| A3     | Analog Input 4 | –                 |
| A4     | Analog Input 5 | TWI – SDA         |
| A5     | Analog Input 6 | TWI – SCL         |

Arduino memberikan kemudahan bagi penggunaanya untuk membuat berbagai proyek berbasis mikrokontroler. Contohnya yang dapat dibuat dengan Arduino antara lain, untuk membuat simulasi lampu, membuat robot, mengontrol motor dc, mengontrol motor stepper, pengatur suhu, display LCD, dan masih banyak yang lainnya. Gambar 2.3 adalah contoh produk alat menggunakan Arduino.



Gambar 2.3 Robot berbasis Arduino Uno

## B. Perangkat Lunak Arduino

Arduino dapat digunakan untuk mengembangkan objek interaktif, mengambil masukan dari berbagai *switch* atau sensor, dan mengendalikan berbagai lampu, motor, dan output fisik lainnya. Proyek Arduino dapat berdiri sendiri, atau berkomunikasi dengan perangkat lunak (*software*) yang berjalan pada komputer (misalnya: Flash, Pengolahan, MaxMSP, database, dsb). Board dapat dirakit sendiri atau dibeli; open-source IDE dapat didownload secara gratis.

Arduino adalah software open source yang memudahkan untuk menulis kode program dan meng-upload-nya ke board Arduino. Software Arduino dapat berjalan pada Windows, Mac OS X, dan Linux. Software ini ditulis dalam bentuk Java dan berbasis processing, avr-gcc, dan perangkat lunak open source lainnya.

Software Arduino yang ada dalam situs Arduino (<https://www.arduino.cc/>) telah memiliki versi 1.6.6, seperti terlihat pada Gambar 2.4. Software Arduino IDE (*Integrated Development Environment*) adalah sebuah perangkat lunak yang memudahkan dalam mengembangkan aplikasi mikrokontroler mulai dari menuliskan source program, kompilasi, upload hasil kompilasi, dan uji coba secara terminal serial. Arduino dapat dijalankan di komputer dengan berbagai macam *platform* karena didukung atau berbasis Java. Source program yang dibuat untuk aplikasi mikrokontroler adalah bahasa C/C++ dan dapat digabungkan dengan *assembly*.

Download the Arduino Software



Gambar 2.4. Tampilan Software Arduino 1.6.6

Di samping IDE Arduino sebagai jantungnya, *bootloader* adalah jantung dari Arduino lainnya yang berupa program kecil yang dieksekusi sesaat setelah mikrokontroler diberi catu daya. *Bootloader* ini berfungsi sebagai pemonitor aktifitas yang diinginkan oleh Arduino. Jika dalam IDE terdapat file hasil kompilasi yang akan di-*upload*, *bootloader* secara otomatis menyambutnya untuk disimpan dalam memori program. Jika pada saat awal mikrokontroler bekerja, *bootloader* akan mengeksekusi program aplikasi yang telah diupload sebelumnya. Jika IDE hendak mengupload program baru, *bootloader* seketika menghentikan eksekusi program berganti menerima data program untuk selanjutnya diprogramkan dalam memori program mikrokontroler.

Hubungan komunikasi data antara IDE arduino dengan board Arduino digunakan komunikasi secara serial dengan protokol RS232. Jika board arduino sudah dilengkapi dengan komunikasi serial RS232 (biasanya USB), maka dapat langsung ditancapkan ke USB komputer. Piranti serial RS232 ini digunakan jika board arduino atau arduino buatan sendiri tidak dilengkapi dengan piranti serial 232.

### Prosedur Menggunakan Arduino Board

1. Menyiapkan Arduino Board dan Kabel USB (Gambar 2.5)



Gambar 2.5. Arduino dan Kabel USB

2. Men-download Software Arduino

Dapatkan versi terbaru dari halaman download yang tersedia di situs resmi Arduino, <https://www.arduino.cc/en/Main/Software>. Setelah download selesai, unzip file yang didownload (jika mendownload file tipe .zip). Pastikan untuk mempertahankan struktur folder. Klik dua kali pada folder untuk membukanya, dan akan ada beberapa file dan sub-folder di dalam.

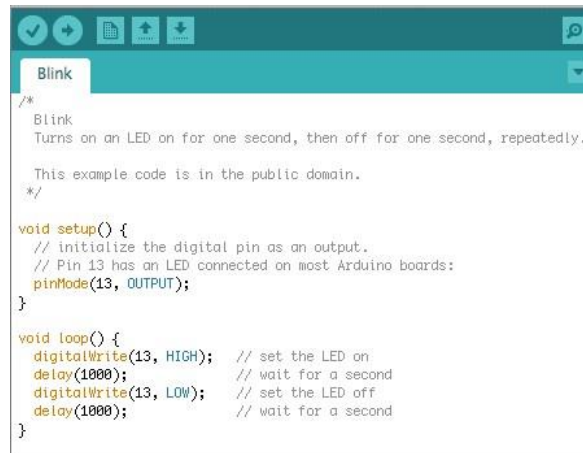
3. Menghubungkan Arduino Board ke Komputer

Arduino Uno, Mega Duemilanove dan Arduino Nano memerlukan sumber listrik dari salah satu koneksi USB komputer atau power supply eksternal. Sumber daya dipilih dengan jumper, plastik kecil yang terdapat antara USB dan jack listrik. Periksa apakah jumper diatur pada dua pin paling dekat dengan port USB atau tidak. Hubungkan papan Arduino ke komputer menggunakan kabel USB. LED indikator daya hijau (berlabel PWR) akan menyala.

#### 4. Membuka Software Arduino (Gambar 2.6)

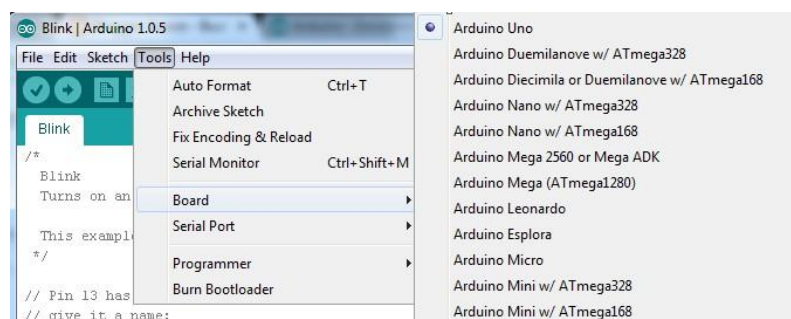
Klik dua kali aplikasi Arduino, lalu dapat dilakukan simulasi dengan hal-hal sebagai berikut :

- ☐ Buka contoh blink
- ☐ Buka LED berkedip contoh sketsa: File > Examples > 1.Basics > Blink.



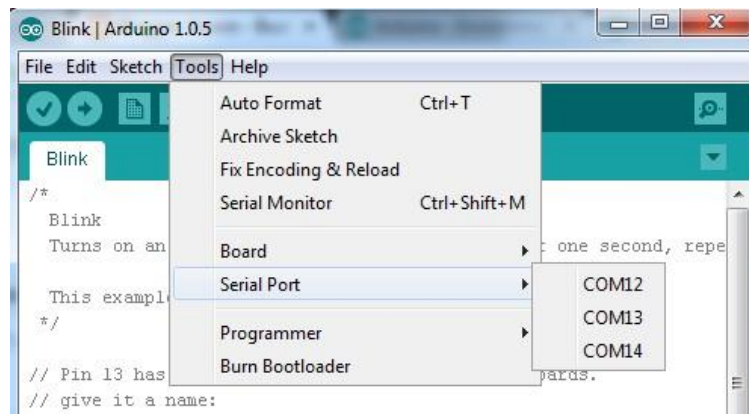
Gambar 2.6. Contoh Blink LED

- ☐ Pilih Board yang digunakan dengan memilih entri dalam menu Tools > Board yang sesuai dengan Arduino. Misalnya dipilih Arduino Uno (Gambar 2.7).



Gambar 2.7. Memilih Board

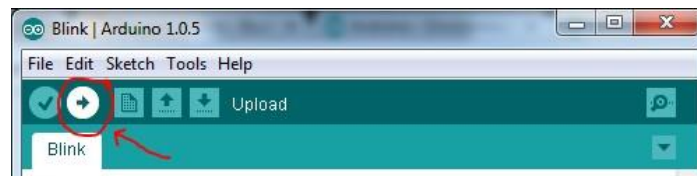
- ☐ Pilih port serial (Gambar 2.8)  
Pilih perangkat serial Arduino dari menu Tools | Serial Port. Biasanya Port akan otomatis terdeteksi.



Gambar 2.8. Memilih Port

- Upload Program (Gambar 2.9)

Dengan meng-klik tombol "*Upload*". Tunggu beberapa detik – akan terlihat akan berkedip LED RX dan TX pada Arduino Board.



Gambar 2.9. Upload Program

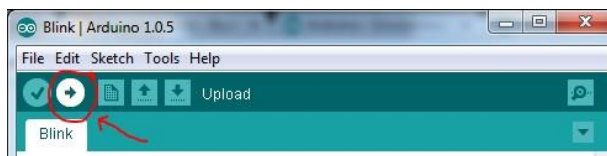


## BAB III

### PEMROGRAMAN ARDUINO

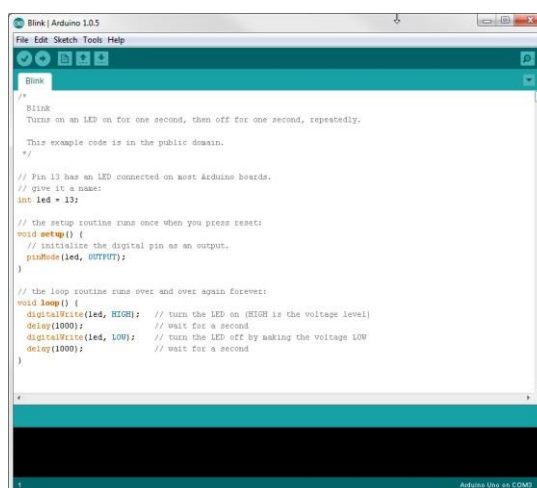
#### A. Konsep Dasar Pemrograman Arduino

Software/program yang ditulis menggunakan Arduino disebut sketsa (*sketch*). Sketsa ini ditulis dalam editor teks. Sketsa disimpan dengan ekstensi file .ino, yang memiliki fitur untuk meng-*cut*, meng-*copy*, mem-*paste*, mencari/ mengganti teks, dll. Area pesan (*console*) memberikan umpan balik, menyimpan dan mengeksport juga menampilkan kesalahan. Konsol menampilkan teks output dalam lingkungan Arduino termasuk detail pesan error dan informasi lainnya. Bagian sudut bawah sebelah kanan jendela menampilkan board dan port serial yang sedang dipakai. Tombol-tombol toolbar (Gambar 3.1.) memungkinkan untuk memverifikasi dan meng-*upload* program, membuat, membuka, dan menyimpan sketsa, juga membuka monitor serial.



Gambar 3.1. Toolbar dan Menu Arduino

Code Program Arduino biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. Program atau sketch yang sudah selesai ditulis di Arduino IDE dapat langsung di-*compile* dan di-*upload* ke Arduino Board.



Gambar 3.2. Sketch dalam Arduino IDE



Secara sederhana, *sketch* dalam Arduino dikelompokkan menjadi 3 blok (Gambar 3.2.), yakni : Header, Setup dan Loop. Untuk program yang lebih kompleks akan ada blok lain berupa fungsi-fungsi pendukung.

## Header

Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan library dan pendefinisian variable. Code dalam blok ini dijalankan hanya sekali pada waktu compile. Di bawah ini contoh code untuk mendeklarasikan variable led (integer) dan sekaligus di isi dengan angka 13

```
int led = 13;
```

## Setup

Di sinilah awal program Arduino berjalan, yaitu di saat awal, atau ketika power on Arduino board. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai input atau output, menggunakan perintah pinMode. Inisialisasi variable juga bisa dilakukan di blok ini

```
// the setup routine runs once when you press reset:
```

```
void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT);  
}
```

OUTPUT adalah suatu makro yang sudah didefinisikan Arduino yang berarti = 1. Jadi perintah di atas sama dengan pinMode(led, 1);

Suatu pin bisa difungsikan sebagai OUTPUT atau INPUT. Jika difungsikan sebagai output, dia siap mengirimkan arus listrik (maksimum 100 mA) kepada beban yang disambungkannya. Jika difungsikan sebagai INPUT, pin tersebut memiliki impedance yang tinggi dan siap menerima arus yang dikirimkan kepadanya.

## Loop

Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol power Arduino di matikan. Di sinilah fungsi utama program Arduino kita berada.

```
void loop() {  
    digitalWrite(led, HIGH); // nyalakan LED delay(1000); //  
    tunggu 1000 milidetik digitalWrite(led, LOW); // matikan  
    LED
```

```

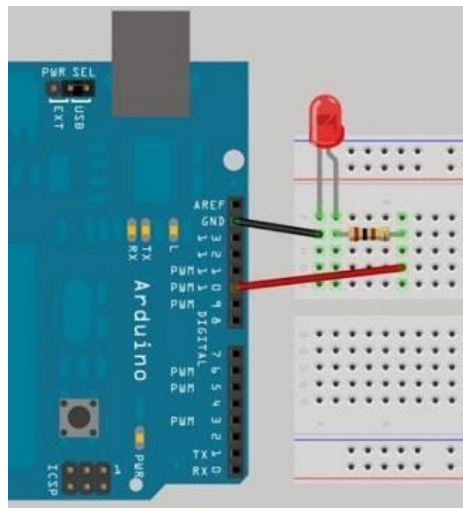
    delay(1000); // tunggu 1000 milidetik
}

```

Perintah `digitalWrite(pinNumber,nilai)` akan memerintahkan arduino untuk menyalakan atau mematikan tegangan di `pinNumber` tergantung nilainya. Jadi perintah di atas `digitalWrite(led,HIGH)` akan membuat pin nomor 13 (karena di header dideklarasikan `led = 13`) memiliki tegangan = 5V (HIGH). Hanya ada dua kemungkinan nilai `digitalWrite` yaitu HIGH atau LOW yang sebetulnya adalah nilai integer 1 atau 0.

## B. Program Sederhana Arduino

Program sederhana adalah *Led Blink*, program ini akan mengakses pin 10 dan memerintahkan Arduino untuk mengulang *blink led*, Gambar 3.3.



Gambar 3.3. Wiring LED blinking

### Sketch

```

// Project 1 - LED Flasher int ledPin =
10;

void setup() {
    pinMode(ledPin, OUTPUT);
}

void loop() { digitalWrite(ledPin, HIGH);
    delay(1000); digitalWrite(ledPin,
    LOW);
    delay(1000);
}

```

## Pembahasan Sketch

```
// Project 1 - LED Flasher
```

Ini adalah komentar baris yang berguna untuk dokumentasi program, kompiler akan mengabaikan bagian ini. Baris komentar berguna bagi programmer agar bisa mengerti maksud program.

```
int ledPin = 10;
```

Inisialisasi variable, dalam hal ini inisialisasi variable bernama ledPin dengan type data integer dan nilai 10.

```
void setup() {  
    pinMode(ledPin, OUTPUT);  
}
```

Setiap sketch Arduino wajib memiliki fungsi setup() dan loop(). Fungsi setup() dipanggil hanya sekali saat pertama kali program berjalan. Fungsi setup() biasanya tempat untuk men-setup hal-hal umum agar program siap dijalankan, seperti setup pin modes, setting serial baud rates, dan lainnya. Pada sketch Led Blink, fungsi setup hanya memiliki 1 baris perintah yaitu

```
pinMode(ledPin, OUTPUT);
```

pinMode fungsi yang berguna untuk memberitahu arduino bahwa pin pada board akan digunakan sebagai input atau output. Dalam baris program di atas, memberitahu arduino untuk menset pin 10 (nilai ledPin adalah 10) sebagai Output.

```
void loop() { digitalWrite(ledPin, HIGH);  
    delay(1000); digitalWrite(ledPin,  
    LOW);  
    delay(1000);  
}
```

Fungsi loop() function adalah program utama yang dipanggil secara continue selama Arduino menyala. Setiap perintah dalam fungsi loop() akan dipanggil satu persatu sampai perintah terakhir dalam blok loop dicapai, lalu Arduino akan kembali ke awal perintah di blok fungsi loop(), sampai Arduino dimatikan atau tombol reset ditekan.

# MATERI PERCOBAAN

## I. ANALOG I/O

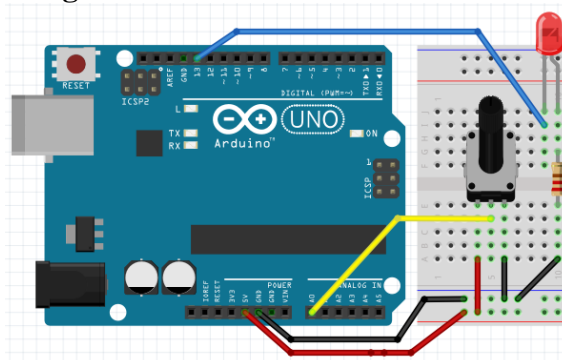
### A. Teori

Pin analog pada Arduino (dan mikrokontroler lain pada umumnya) dapat digunakan untuk input dan output digital. Hanya saja pin analog memiliki fitur untuk dapat mengubah sinyal analog yang masuk menjadi nilai digital yang mudah diukur. Pin digital hanya dapat mengenali sinyal 0 volt sebagai nilai LOW dan 5 volt sebagai nilai HIGH. Sedangkan Pin analog dapat mengenali sinyal pada rentang nilai voltase tersebut. Hal ini sangat berguna ketika kita hendak mengukur sesuatu dari sensor dan menggunakan nilai masukan tersebut untuk keperluan lain. Pin analog ini terhubung dengan converter pada mikrokontroler yang dikenal dengan istilah analog-to-digital converter (disingkat ADC atau A/D). Converter ini mengubah nilai analog berbentuk sinyal voltase ke dalam bentuk digital/angka supaya nilai analog ini dapat digunakan dengan lebih mudah dan aplikatif. Pada Arduino (mikrokontroler ATmega) converter ini memiliki resolusi 10 bit, artinya nilai hasil konversi berkisar dari 0 hingga 1023. Pada Arduino UNO, pin analog ditandai dengan label A0 sampai A5. Pada board lainnya, pin-pin yang diberi tanda A, Analog, ADC adalah pin analog.

Fungsi yang kita gunakan untuk membaca nilai analog pada Arduino adalah `analogRead([nomorPin])`. Pada contoh kasus berikut, kita akan menghubungkan potensiometer pada pin analog A5 dan hasil konversi nilai analog akan kita lihat pada jendela Serial Monitor pada Arduino IDE.

### B. Percobaan ADC dan Pengatur Kecerahan LED

- **Tujuan Percobaan**
  - Mengetahui proses ADC yang berlangsung pada input analog
  - Menggunakan I/O analog untuk mengatur kecerahan LED
- **Alat dan Bahan**
  - Board Arduino
  - Potensiometer 10K, Resistor 330K $\Omega$
  - LED
  - Bread-board
  - Kabel jumper secukupnya
- **Rangkaian Percobaan**



- **Program Percobaan**

```
int led = 13; // Memilih pin digital untuk lampu LED
int pinPot = 0; // Memilih pin analog untuk VR
int potVal = 0; // Tempat untuk menerima nilai tegangan dari VR

void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT); // Mengatur lampu LED sebagai output
}

void loop() {
  potVal = analogRead(pinPot); // terima masukan nilai dari VR
  Serial.println(potVal);

  // ubah nilai (0-1023) jadi (0-255)
  potVal = map(potVal, 0, 1023, 0, 255);

  // ubah nilai VR untuk mengatur kecerahan
  analogWrite(led, potVal);
}
```

### C. Percobaan LDR (Light Dependent Resistor)

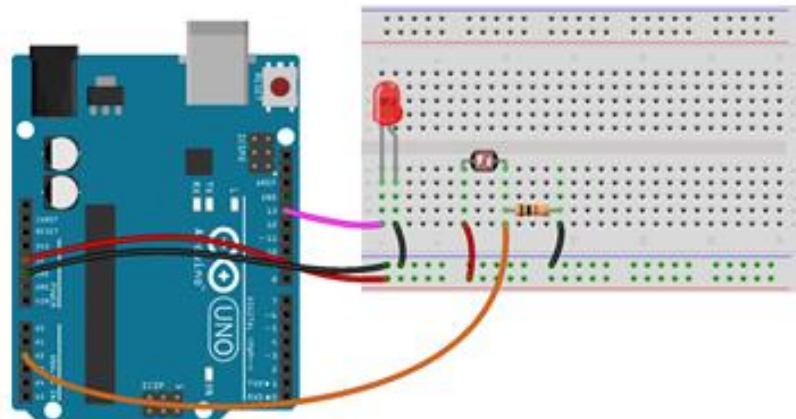
- **Tujuan Percobaan**

- Menggunakan LDR sebagai input analog
- Menggunakan LDR sebagai sensor untuk menghidupkan LED

- **Alat dan Bahan**

- Board Arduino
- LDR
- LED
- Resistor 330Ω
- Resistor 10KΩ
- Bread-board
- Kabel jumper secukupnya

- **Rangkaian Percobaan**



- **Program Percobaan**

```
int LDR = A2;
int LED = 13;
int nilaiLDR = 0;
```

```

void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  nilaiLDR = analogRead(LDR);
  Serial.print("nilaiLDR = ");
  Serial.println(nilaiLDR);
  if (nilaiLDR < 500){
    digitalWrite(LED, HIGH);
  }
  else {
    digitalWrite(LED, LOW);
  }
}

```

## II. DIGITAL I/O

### A. Teori

Semua pin pada Arduino, baik pin digital maupun pin analog, dapat digunakan sebagai pin digital. Digital berarti sinyal yang dikirimkan/diterima bernilai 1 atau 0, on atau off, HIGH atau LOW, ada atau tidak ada sinyal. Berbeda dengan sinyal analog yang nilainya bersifat kontinyu, yakni nilai antara 0 dan 1 dipertimbangkan. Pin digital berarti pin dapat menerima/mengirim sinyal digital. Pada dasarnya semua pin yang ada pada Arduino (ATMega) berada pada mode input secara default. Jadi ketika kita ingin menggunakan suatu pin sebagai input, maka kita tidak mesti menuliskan `pinMode(nomorPin, INPUT);`.

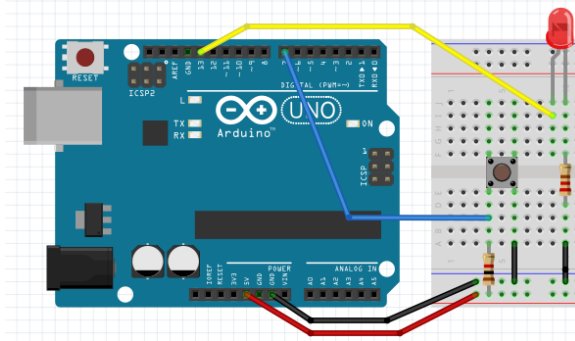
Untuk menerima input digital yang masuk ke pin, kita gunakan fungsi `digitalRead(nomorPin)`. Fungsi ini menerima satu parameter, yaitu nomor pin mana yang akan dibaca nilai inputnya. Fungsi ini akan mengembalikan nilai 1 dan 0, atau HIGH dan LOW (HIGH adalah konstanta dengan nilai 1 dan LOW adalah konstanta dengan nilai 0). Supaya lebih paham, silakan coba contoh berikut. Skenarionya adalah kita akan menghubungkan pushbutton ke pin 2 dan menyalakan LED yang ada pada board Arduino Uno ketika pushbutton tersebut ditekan. Sebenarnya pada bagian pembahasan Digital Input di atas, kita sudah mempraktekkan digital output. Untuk mengirimkan sinyal digital, kita gunakan fungsi `digitalWrite(nomorPin, nilaiDigital)`. Fungsi ini dapat digunakan pada pin yang sebelumnya sudah diset ke mode OUTPUT. Parameter kedua adalah set nilai HIGH atau LOW. Apabila pin diset dengan nilai HIGH, maka voltase pin tersebut akan diset ke 5V (atau 3.3V pada board bertipe 3.3V) dan bila pin diset ke LOW, maka voltase pin tersebut akan diset ke 0V.

### B. Percobaan Input Digital dengan Push-Button

- **Tujuan Percobaan**
  - Menggunakan push-button sebagai input digital
- **Alat dan Bahan**
  - Board Arduino
  - Push-button
  - LED
  - Resistor 1K $\Omega$  (2)
  - Bread-board

- Kabel jumper secukupnya

- **Rangkaian Percobaan**



- **Program Percobaan**

```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7;   // pushbutton connected to digital pin 7
int val = 0;     // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output
  pinMode(inPin, INPUT);   // sets the digital pin 7 as input
}

void loop() {
  val = digitalRead(inPin); // read the input pin
  digitalWrite(ledPin, val); // sets the LED to the button's value
}
```

## C. Percobaan LCD (Liquid Crystal Display)

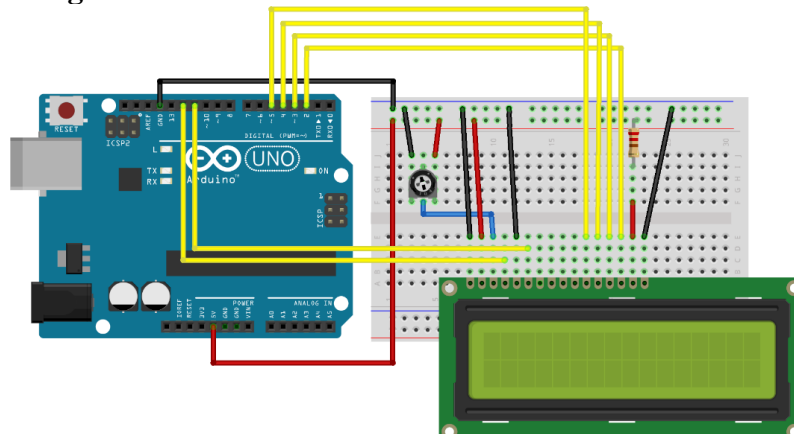
- **Tujuan Percobaan**

- Menggunakan LCD sebagai output digital

- **Alat dan Bahan**

- Board Arduino
- LCD 1602
- Potensiometer 10K
- Bread-board
- Kabel jumper secukupnya

- **Rangkaian Percobaan**



- **Program Percobaan**

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // Turn off the display:
  lcd.noDisplay();
  delay(500);
  // Turn on the display:
  lcd.display();
  delay(500);
}
```

## D. Percobaan Motor Servo

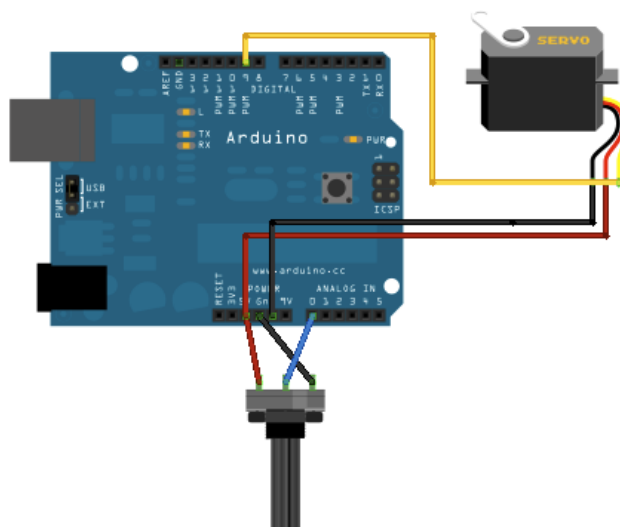
- **Tujuan Percobaan**

- Menggerakkan motor servo sebagai aktuator dengan i/o digital

- **Alat dan Bahan**

- Board Arduino
- Motor Servo
- Potentiometer 10K
- Bread-board
- Kabel jumper secukupnya

- **Rangkaian Percobaan**





- **Program Percobaan**

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo
  object
}

void loop() {
  // reads the value of the potentiometer (value between 0 and
  1023)
  val = analogRead(potpin);

  // scale it to use it with the servo (value between 0 and 180)
  val = map(val, 0, 1023, 0, 180);

  // sets the servo position according to the scaled value
  myservo.write(val);
  delay(15); // waits for the servo to get there
}
```

## E. Percobaan Sensor Ultrasonik HC-SR04

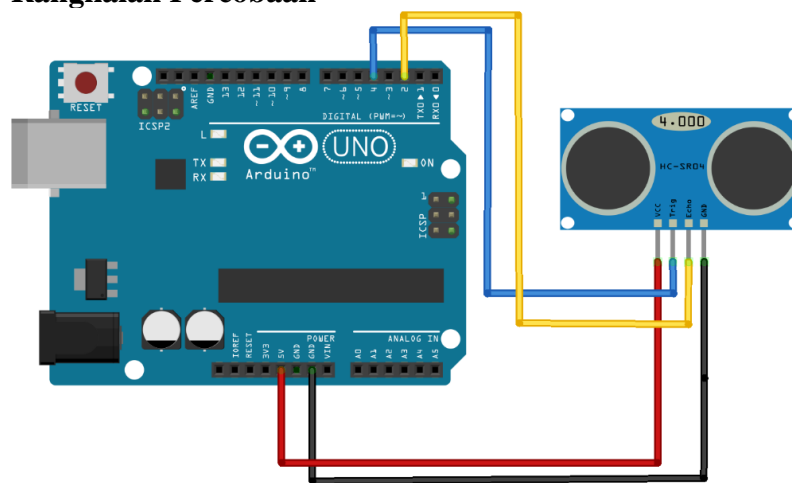
- **Tujuan Percobaan**

- Menggunakan sensor ultrasonic HC-SR04 sebagai I/O digital

- **Alat dan Bahan**

- Board Arduino
- HC-SR04
- Bread-board
- Kabel jumper secukupnya

- **Rangkaian Percobaan**



- **Program Percobaan**

```
#define trigger 4 //mendefinisikan trigger pada pin 4
#define echo 2 //mendeklarasikan echo pada pin 2
void setup() {
```

```

Serial.begin(115200); //memulai serial
pinMode (triger, OUTPUT); //trigger sebagai output
pinMode (echo, INPUT); //echo sebagai input
}

void loop() {
  digitalWrite (triger, HIGH); //mengirim suara
  delayMicroseconds(10); //selama 10 mikro detik
  digitalWrite (triger, LOW); //berhenti mengirim suara

  //membaca data dan di masukkan ke variabel jarak
  float jarak = pulseIn(echo, HIGH);
  jarak=jarak/1000000; //konversi mikro detik ke detik
  jarak=jarak*330/2; //data mentah di ubah ke dalam meter
  jarak=jarak*100; //mengubah data ke dalam centi meter

  Serial.println(jarak); //menampilkan nilai jarak pada serial

  delay(500); //delay 500ms
}

```

### III. MEMORY (EEPROM)

#### A. Teori

EEPROM merupakan suatu memori yang dapat kita gunakan untuk menyimpan suatu data, data tersebut tidak akan terhapus walaupun Arduino dimatikan atau di reset, dan hanya akan terhapus melalui perintah (program) yang sudah ditentukan, perintah tersebut adalah EEPROM.clear. EEPROM dapat ditulis per-blok dimana dalam setiap bloknya mampu menyimpan data sekitar 0 sampai 255.

Kapasitas dari EEPROM sendiri berbeda-beda sesuai dengan mikrokontroler yang digunakan, dalam percobaan ini saya menggunakan board Arduino Pro yang menggunakan mikrokontroler atmega328p, chip ini sama dengan yang digunakan oleh Arduino uno R3, kapasitas EEPROM yang dimiliki oleh atmega328P adalah 1 KB, yang berarti mempunyai 1023 blok penyimpanan EEPROM. Untuk menggunakan EEPROM, kita harus menggunakan library EEPROM.h yang sudah disediakan pada Arduino IDE.

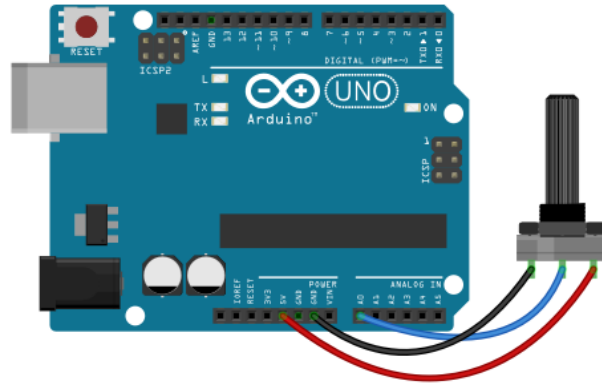
- EEPROM.write (addr, val) → untuk menulis data ke EEPROM. Parameter 'addr' adalah alamat EEPROM mulai dari 0 s/d 1023, sedangkan parameter 'val' adalah nilai/data yang berada dalam alamat EEPROM tersebut. Karena lebar data hanya 8 bit maka data yang ditampung hanya dapat bernilai 0 s/d 255. Saya akan menjelaskan lebih detail mengenai hal ini di bawah.
- EEPROM.read(address) → untuk membaca/mengambil data dari EEPROM. Parameter 'address' merupakan alamat EEPROM yang akan dibaca.

#### B. Percobaan Read/Write Data pada EEPROM

- **Tujuan Percobaan**
  - Melakukan operasi baca / tulis data pada EEPROM

- **Alat dan Bahan**
  - Board Arduino
  - Potensiometer
  - Bread-board
  - Kabel jumper secukupnya

- **Rangkaian Percobaan**



- **Program Percobaan**

```
#include <EEPROM.h>

int address = 0;
byte value;
int addr = 0;

void setup(){
  Serial.begin(9600);
}

void loop(){
  int val = analogRead(0) / 4;
  EEPROM.write(addr, val);

  address=addr;
  value = EEPROM.read(address);

  Serial.print(address);
  Serial.print("\t");
  Serial.print(value, DEC);
  Serial.println();

  address = address + 1;
  if (address == 512)
    address = 0;

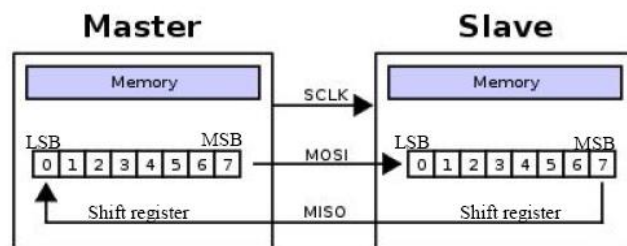
  addr = addr + 1;
  if (addr == 512)
    addr = 0;

  delay(500);
}
```

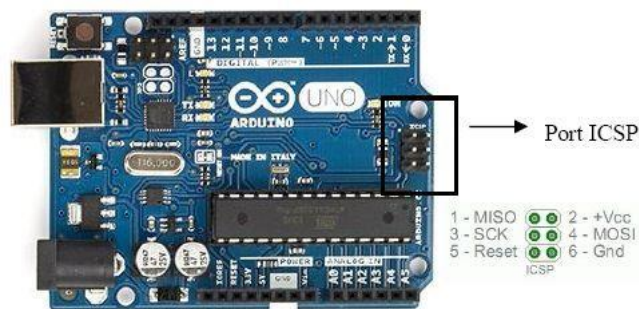
## IV. KOMUNIKASI SPI (Serial Peripheral Interface)

### A. Teori

Library SPI.h adalah library yang khusus bertugas menangani komunikasi serial sinkron SPI (Serial Peripheral Interface) di Arduino. Serial sinkron adalah protokol komunikasi data secara serial namun membutuhkan jalur clock untuk sinkronisasi antara transmitter dan receiver-nya. Sedangkan secara khusus istilah ‘serial sinkron SPI’ ditujukan untuk tipe protokol komunikasi serial sinkron yang memiliki 3 jalur kabel yakni MISO (Master In Slave Out), MOSI (Master Out Slave In) dan SCLK (Serial Clock). MOSI merupakan jalur pengiriman data dari master ke slave, sedangkan MISO merupakan kebalikannya. Biasanya ada satu tambahan pin yang digunakan untuk mengaktifkan/menonaktifkan perangkat slave SPI yang dinamakan CS (Chip Select) atau SS (Slave Select). Pin CS/SS bersifat spesifik untuk tiap perangkat slave yang menggunakan komunikasi SPI, jadi bisa berbeda-beda untuk masing-masing perangkat. Ilustrasi cara kerja protokol SPI ini ditunjukkan pada gambar di bawah :



Beberapa tipe Arduino memiliki konfigurasi yang berbeda untuk SPI ini. Contohnya, Arduino memiliki pin MOSI di pin 11, MISO di pin 12, SCLK di pin 13 dan pin CS/SS di pin 10 . Sedangkan untuk Arduino Mega/Mega ADK, berturut-turut untuk pin MISO, MOSI, SCLK, CS/SS adalah di pin 50, 51, 52 dan 53. Pin-pin tersebut juga terhubung langsung dengan port ICSP di papan Arduino. Letak port ICSP dapat Anda lihat pada gambar di bawah.



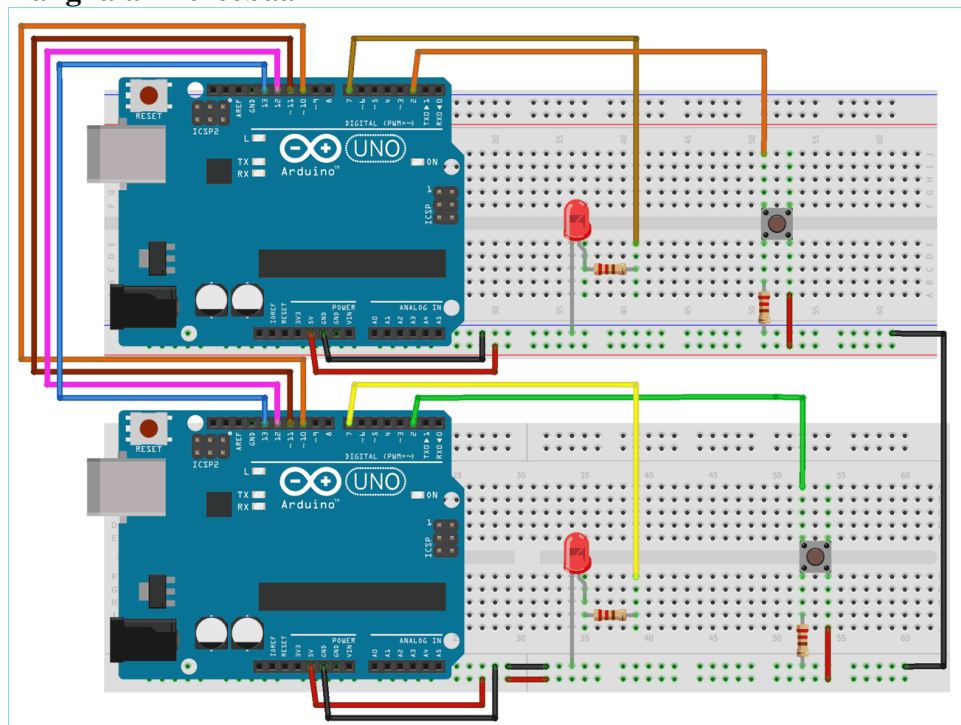
Dari sisi software, Arduino sudah menyediakan library internal khusus untuk menangani komunikasi serial sinkron dengan SPI ini. Seperti yang sudah disebutkan di atas, nama library tersebut adalah ‘SPI.h’. Library ini bisa langsung digunakan dalam sketch Arduino utama atau sebagai pendukung library yang lain. Sebagai contoh, library ‘Ethernet.h’ atau ‘SD.h’ harus menyertakan ‘SPI.h’ dalam sketch utama karena secara hardware, kedua library tersebut menggunakan protokol komunikasi SPI.

Dalam implementasinya, SPI banyak digunakan sebagai alternatif untuk berkomunikasi dengan perangkat lain misalnya EEPROM (SPI EEPROM), sensor (barometer, tekanan, dll), komponen elektronika (SPI digital

potensiometer) atau controller lain (Arduino, AVR, MCS51, ARM dll). Alternatif komunikasi lainnya adalah menggunakan serial asinkron UART atau serial sinkron IIC/I2C/TWI yang sudah saya bahas di artikel yang lalu. Dengan menggunakan alternatif jalur komunikasi tersebut, setidaknya Anda dapat menghemat port digital Anda.

## B. Percobaan Komunikasi 2 Arduino dengan Protokol SPI

- **Tujuan Percobaan**
  - Memahami protocol komunikasi sinkron SPI pada mikrokontroler
  - Melakukan komunikasi antara 2 arduino dengan protocol SPI
- **Alat dan Bahan**
  - Board Arduino (2)
  - LED (2)
  - Push-button (2)
  - Resistor 10K $\Omega$  (2)
  - Resistor 2.2K $\Omega$  (2)
  - Bread-board
  - Kabel jumper secukupnya
- **Rangkaian Percobaan**



- **Program Percobaan**  
Master Arduino Code:

```
#include<SPI.h> //Library for SPI
#define LED 7
#define ipbutton 2
int buttonvalue;
int x;
void setup (void)
```

```

{
  Serial.begin(115200); //Starts Serial Communication at Baud Rate
115200

  pinMode(ipbutton,INPUT); //Sets pin 2 as input
  pinMode(LED,OUTPUT);      //Sets pin 7 as Output

  SPI.begin(); //Begins the SPI commnuication
  SPI.setClockDivider(SPI_CLOCK_DIV8); //Sets clock for SPI
communication at 8 (16/8=2Mhz)
  digitalWrite(SS,HIGH); // Setting SlaveSelect as HIGH (So master
doesnt connect with slave)
}

void loop(void)
{
  byte Mastersend,Mastereceive;

  buttonvalue = digitalRead(ipbutton); //Reads the status of the
pin 2

  if(buttonvalue == HIGH) //Logic for Setting x value (To be sent
to slave) depending upon input from pin 2
  {
    x = 1;
  }
  else
  {
    x = 0;
  }

  digitalWrite(SS, LOW); //Starts communication with Slave
connected to master

  Mastersend = x;
  Mastereceive=SPI.transfer(Mastersend); //Send the mastersend
value to slave also receives value from slave

  if(Mastereceive == 1) //Logic for setting the LED output
depending upon value received from slave
  {
    digitalWrite(LED,HIGH); //Sets pin 7 HIGH
    Serial.println("Master LED ON");
  }
  else
  {
    digitalWrite(LED,LOW); //Sets pin 7 LOW
    Serial.println("Master LED OFF");
  }
  delay(1000);
}

```

Slave Arduino Code:

```

#include<SPI.h>
#define LEDpin 7
#define buttonpin 2
volatile boolean received;
volatile byte Slavereceived,Slavesend;
int buttonvalue;

```

```

int x;
void setup()

{
    Serial.begin(115200);

    pinMode(buttonpin,INPUT); // Setting pin 2 as INPUT
    pinMode(LEDpin,OUTPUT);   // Setting pin 7 as OUTPUT
    pinMode(MISO,OUTPUT);     //Sets MISO as OUTPUT (Have to Send data
to Master IN

    SPCR |= _BV(SPE);        //Turn on SPI in Slave Mode
    received = false;

    SPI.attachInterrupt(); //Interuupt ON is set for SPI commnucation
}

ISR (SPI_STC_vect) //Inerrrrput routine function
{
    Slavereceived = SPDR; // Value received from master if store in
variable slavereceived
    received = true;      //Sets received as True
}

void loop()
{ if(received) //Logic to SET LED ON OR OFF depending upon the
value recerived from master
    {
        if (Slavereceived==1)
        {
            digitalWrite(LEDpin,HIGH);    //Sets pin 7 as HIGH LED ON
            Serial.println("Slave LED ON");
        }else
        {
            digitalWrite(LEDpin,LOW);      //Sets pin 7 as LOW LED OFF
            Serial.println("Slave LED OFF");
        }

        buttonvalue = digitalRead(buttonpin); // Reads the status of
the pin 2

        if (buttonvalue == HIGH) //Logic to set the value of x to
send to master
        {
            x=1;

        }else
        {
            x=0;
        }

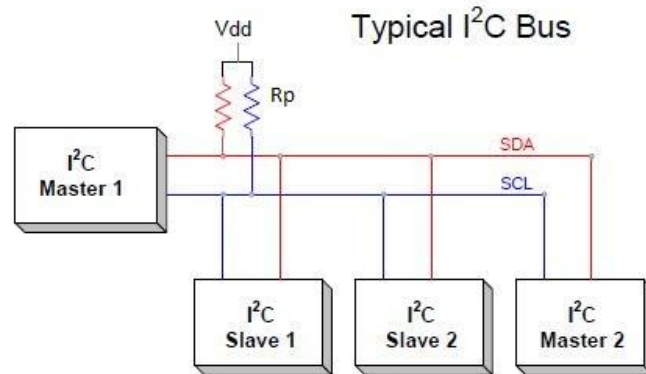
        Slavesend=x;
        SPDR = Slavesend; //Sends the x value to master via SPDR
        delay(1000);
    }
}

```

## V. KOMUNIKASI I<sup>2</sup>C (Inter-Integrated Circuit)

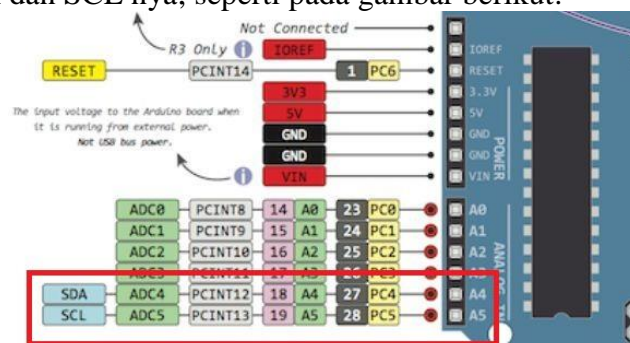
### A. Teori

Arduino sendiri sudah mendukung protokol I<sup>2</sup>C/IIC. Di papan Arduino Uno, port I<sup>2</sup>C terletak pada pin A4 untuk jalur SDA (Serial Data) dan pin A5 untuk jalur SCL (Serial Clock). Jangan lupa untuk menghubungkan jalur kabel Ground antara Arduino dengan perangkat I<sup>2</sup>C client. Untuk sisi software, Arduino sudah cukup membantu kita bekerja dengan protokol ini melalui library 'Wire.h'. Tipikal komunikasi I<sup>2</sup>C dapat dilihat pada gambar di bawah ini :



Sebagai contoh, untuk mengakses Arduino dimana ada dua buah Arduino yang saling berhubungan atau Berkomunikasi antar keduanya dapat menggunakan Komunikasi I<sup>2</sup>C (Inter Integrated Circuit) dimana bisa mengatur Arduino tersebut menjadi slave atau master, slave atau master keduanya bisa dijadikan untuk mengirim data atau sebagai penerima data tergantung kita memprogramnya.

Pada komunikasi I<sup>2</sup>C kita menggunakan komunikasi sinkron yang artinya pengiriman clock secara bersamaan dan juga kita dapat menggunakan beberapa perangkat sekaligus dengan maksimal perangkat 112, dan jika kita menggunakan banyak perangkat maka kita wajib menggunakan resistor pullup pada jalur SDA dan SCL nya, seperti pada gambar berikut.



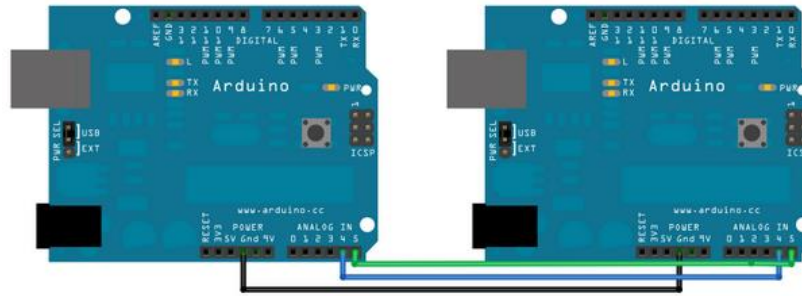
### B. Percobaan Komunikasi 2 Arduino dengan Protokol I<sup>2</sup>C

- **Tujuan Percobaan**
  - Memahami protocol komunikasi sinkron I<sup>2</sup>C pada mikrokontroler
  - Melakukan komunikasi antara 2 arduino dengan protocol I<sup>2</sup>C
- **Alat dan Bahan**
  - Board Arduino (2)



➤ Kabel jumper secukupnya

- **Rangkaian Percobaan**



- **Program Percobaan**

Master Code :

String data;

```
#include <Wire.h>
```

```
void setup() {  
  Wire.begin();  
  Serial.begin(9600);  
}
```

```
void loop() {  
  Wire.requestFrom(7, 6); // 7 adalah alamat untuk perangkat yang  
    akan terhubung dengan master, 6 adalah banyaknya data dalam bytes  
    yang diterima
```

```
    while (Wire.available()) {  
      char c = Wire.read();  
      Serial.print(c);  
      if (c==' '){  
        Serial.println(c);  
      }  
    }  
    delay(500);  
}
```

Slave Code :

```
#include <Wire.h>
```

```
void setup() {  
  Wire.begin(7);  
  Wire.onRequest(requestEvent);  
}
```

```
void loop() {  
  delay(100);  
}
```

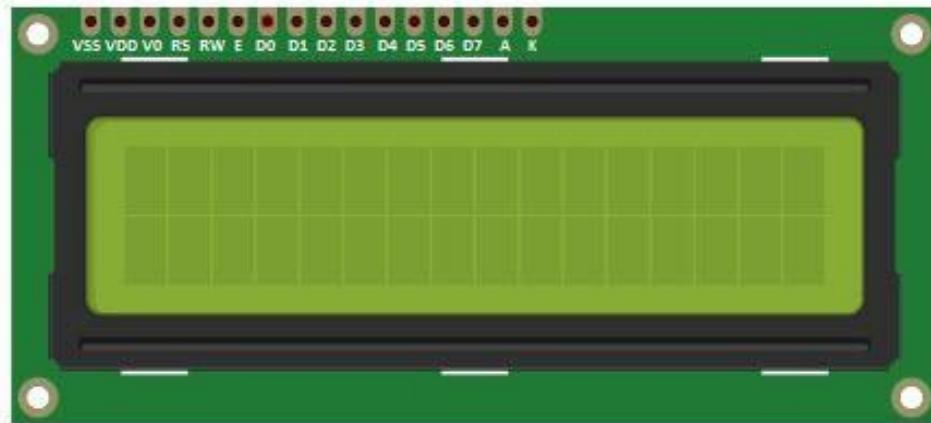
```
void requestEvent() {  
  Wire.write("hello ");  
}
```

## LAMPIRAN

### 1. LCD (Liquid Crystal Display)

LCD merupakan singkatan dari Liquid Crystal Display, atau umumnya disebut dengan LCD atau display saja. Di pasaran beragam jenis LCD dan berbagai ukuran yang bisa Anda gunakan. LCD bisa digunakan untuk menampilkan huruf dan angka, bahkan ada yang bisa untuk menampilkan gambar.

LCD yang biasa dipakai dikenal juga dengan LCD 1602 dengan beberapa varian seperti 1602A, dll. LCD ini bisa bekerja pada 5 volt, sehingga Anda bisa menyambungkannya secara langsung ke pin VCC pada board Arduino. Perlu diperhatikan, jika Anda menggunakan LCD jenis lainnya, ada juga LCD yang bekerja pada voltase yang berbeda. Sehingga kesalahan pemasangan sumber tegangan bisa membuat LCD rusak.



Gambar A.1 LCD 1602

LCD 1602 memiliki 16 pin dengan fungsi-fungsi sebagai berikut:

| Simbol | Value   | Fungsi                              |
|--------|---------|-------------------------------------|
| VSS    | 0V      | Ground                              |
| VDD    | +5V     | Power Supply / VCC                  |
| V0     | -       | Pengaturan kontras backlight        |
| RS     | H/L     | H = data, L = command               |
| R/W    | H/L     | H = read, L = write                 |
| E      | H.H - L | Enable Signal                       |
| D1-D3  | H/L     | Jalur untuk transfer 8 bit data     |
| D4-D7  | H/L     | Jalur untuk transfer 4 & 8 bit data |
| A      | +5V     | VCC untuk backlight                 |
| K      | 0V      | GND untuk backlight                 |

Berdasarkan karakteristik tersebut, maka semua pin akan digunakan kecuali pin D1 – D3 sebab kita akan menggunakan jalur data untuk transfer 4 bit atau 8 bit.

Penjelasan singkat tentang RS, R/W, dan E:

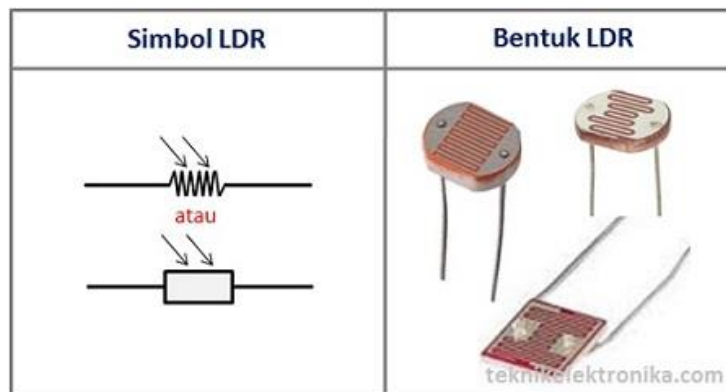
- RS merupakan kependekan dari Register Selector, pin ini berfungsi untuk memilih register control atau register data. Register control digunakan untuk mengkonfigurasi LCD, sedangkan register data digunakan untuk menuliskan data berupa karakter untuk ditampilkan di LCD.
- R/W atau Read/Write, digunakan untuk memilih aliran data mikrokontroler akan membaca data yang ada di LCD atau menuliskan data ke LCD. Jika LCD hanya

digunakan untuk menulis / menampilkan data, maka pin ini bisa langsung disambungkan ke GND sehingga logika bisa diset menjadi L (Low).

- E atau Enable, digunakan untuk mengaktifkan LCD ketika proses penulisan data ke register control dan register data.

## 2. LDR (Light Dependent Resistor)

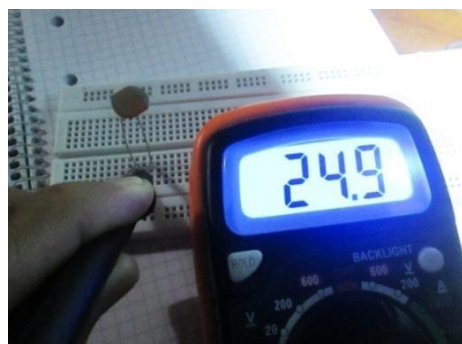
LDR disebut juga sebagai photoresistor sebab alat ini akan memiliki resistansi yang akan berubah seiring dengan perubahan intensitas cahaya yang mengenainya. Dalam kondisi gelap, resistansi LDR bisa mencapai 10 M ohm, tapi dalam kondisi terang, resistansi LDR turun hingga 1 K ohm bahkan bisa kecil lagi (Gambar A.3 dan Gambar A.4). Sifat inilah yang membuat LDR bisa dimanfaatkan sebagai sensor cahaya.



Gambar A.2 Simbol dan bentuk LDR



Gambar A.3 Resistansi LDR saat terkena cahaya



Gambar A.4 Resistansi LDR saat cahaya dihalangi

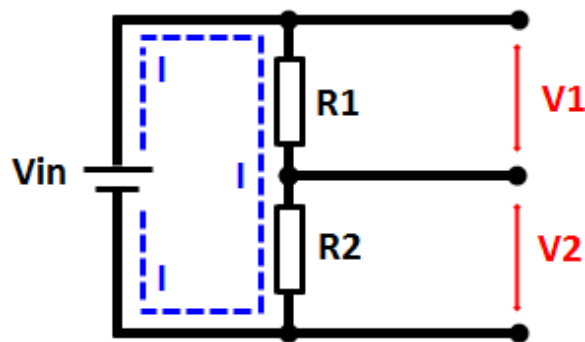
LDR terbuat dari sebuah cakram semikonduktor seperti kadmium sulfida dengan dua buah elektroda pada permukaannya. Pada saat intensitas cahaya yang mengenai LDR sedikit, bahan dari cakram LDR tersebut menghasilkan elektron bebas dengan jumlah

yang relatif kecil. Sehingga hanya ada sedikit elektron untuk mengangkut muatan elektrik. Artinya saat intensitas cahaya yang mengenai LDR sedikit maka LDR akan memiliki resistansi yang besar.

Sedangkan pada saat kondisi terang, maka intensitas yang mengenai LDR banyak. Maka energi cahaya yang diserap akan membuat elektron bergerak cepat sehingga lepas dari atom bahan semikonduktor tersebut. Dengan banyaknya elektron bebas, maka muatan listrik lebih mudah untuk dialirkan. Artinya saat intensitas cahaya yang mengenai LDR banyak maka LDR akan memiliki resistansi yang kecil dan menjadi konduktor yang baik.

Gambar di atas adalah resistansi pada LDR dalam kondisi terang dan kondisi gelap. Dalam kondisi terang, resistansi masih kisaran 1Kohm, dan ketika cahaya sedikit terhalangi sehingga agak gelap, maka resistansi meningkat hingga puluhan kilo ohm. Karakteristik inilah yang bisa kita manfaatkan untuk mengaktifkan relay dan menhidupkan lampu.

Ketika ingin menjadikan LDR sebagai sensor, maka kita bisa mengacu pada rangkaian resistor sebagai pembagi tegangan (lihat Gambar A.5). Dengan menggabungkan antara LDR dengan resistor (atau potensiometer), maka kita bisa mendapatkan variasi tegangan (pada V1 atau V2) yang nantinya menjadi inputan pada pin analog Arduino.



Gambar A.5 Rangkaian pembagi tegangan

Tegangan pada V1 atau V2 dapat dihitung berdasarkan hukum ohm dan aturannya pada rangkaian seri. Pada rangkaian tersebut, arus pada semua titik dalam rangkaian tersebut nilainya sama sehingga kita bisa menghitung V1 atau V2 tanpa mengetahui arus yang mengalir. Lalu bagaimana cara menghitung V1 dan V2?

Pada rangkaian, ada 3 titik yang memiliki tegangan berbeda. Tegangan  $V_{in}$ , tegangan pada R1, dan tegangan pada R2. Berdasarkan hukum ohm,  $V_{in}$ , V1, dan V2 bisa dihitung dengan cara:

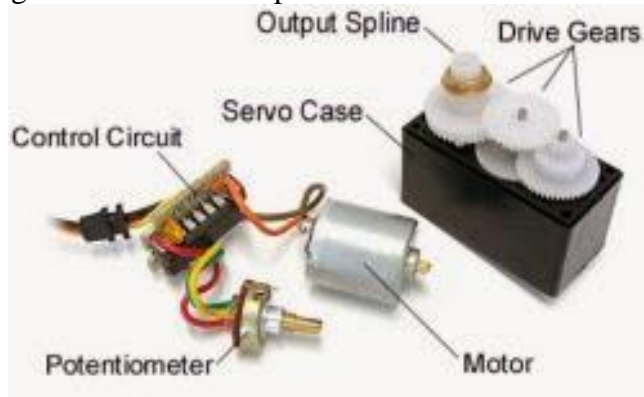
$$V_{in} = I (R1 + R2) \quad (1.1)$$

$$V1 = \frac{R1}{(R1 + R2)} V_{in} \quad (1.2)$$

$$V2 = \frac{R2}{(R1 + R2)} V_{in} \quad (1.3)$$

### 3. MOTOR SERVO

Motor servo adalah jenis motor DC dengan sistem umpan balik tertutup yang terdiri dari sebuah motor DC, serangkaian gear, rangkaian kontrol, dan juga potensiometer. Jadi motor servo sebenarnya tak berdiri sendiri, melainkan didukung oleh komponen-komponen lain yang berada dalam satu paket.



Gambar A.6 Konstruksi motor servo

Sedangkan fungsi potensiometer dalam motor servo adalah untuk menentukan batas sudut dari putaran servo. Sementara sudut sumbu motor servo dapat diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel servo itu sendiri. Oleh karena itu motor servo dapat berputar searah dan berlawanan arah jarum jam.

Motor servo dapat menampilkan gerakan 0 derajat, 90 derajat, 180 derajat, hingga 360 derajat. Tak heran jika motor ini banyak diaplikasikan untuk penggerak kaki dan juga lengan robot. Selain itu motor servo juga memiliki torsi yang besar sehingga mampu menopang beban cukup berat. Berikut bagian-bagian dari motor servo.

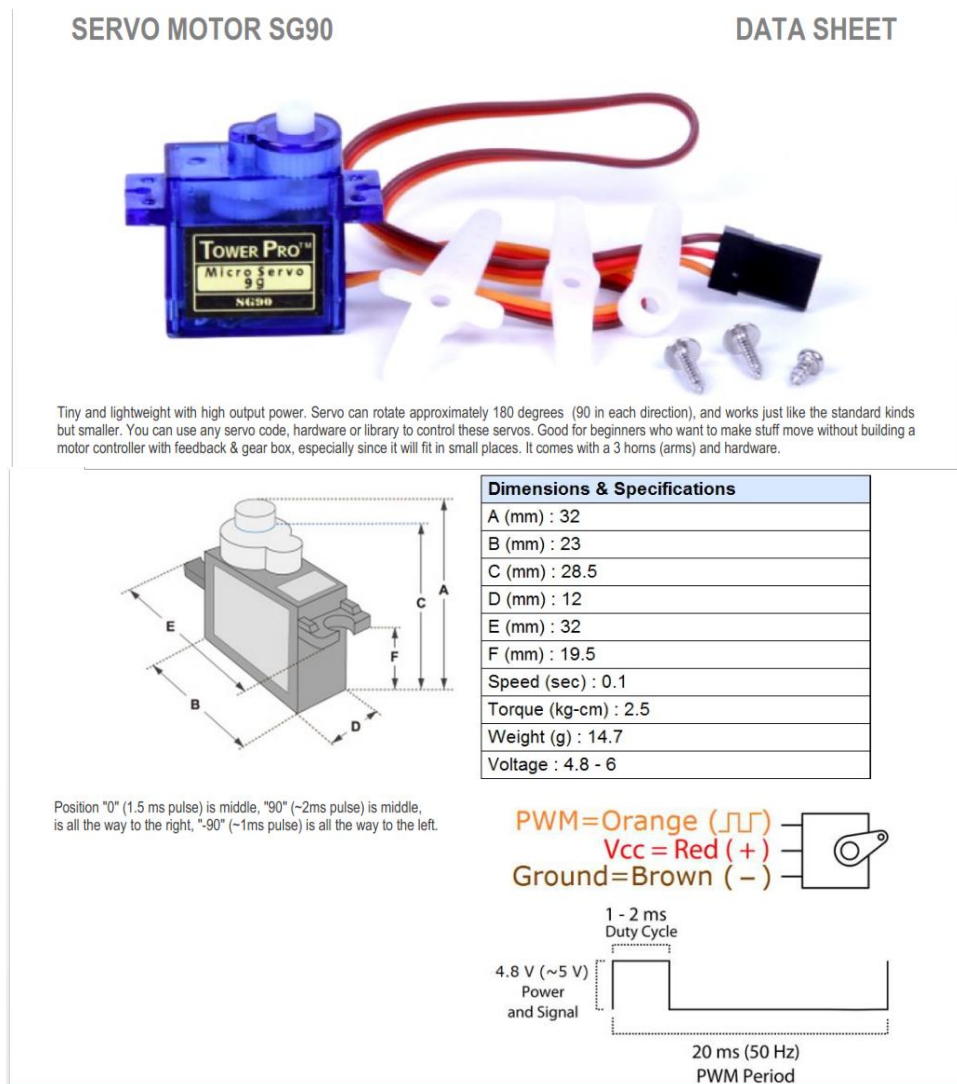
Sebenarnya prinsip kerja motor servo tidak jauh berbeda dengan jenis motor DC yang lainnya, hanya saja prinsip paling utama yang dimiliki dari motor Servo adalah mampu berputar searah ataupun berlawanan jarum jam.

#### Prinsip Kerja Motor Servo

- **Bekerja Searah ataupun Berlawanan Jarum Jam**  
Berbeda dengan prinsip kerja yang dimiliki motor DC, jika pada motor DC hanya dapat bekerja menghasilkan putaran searah jarum jam saja maka pada prinsip kerja motor servo yaitu dapat bekerja searah jarum jam maupun arah sebaliknya atau berlawanan dengan arah jarum jam.
- **Motor Servo Dapat Dikontrol**  
Motor servo memiliki prinsip kerja yang dapat dikontrol dengan mudah, sistem kontrol dari motor servo sendiri disebut dengan loop tertutup, sehingga mudah diatur untuk memastikan atau menentukan posisi sudut dari poros output motor. Penggunaan loop pada kontrol motor servo juga berfungsi untuk mengontrol posisi akhir dan gerakan dari poros motor servo.

Contoh dari penggunaan sistem umpan balik loop sendiri tidak hanya dapat kita temukan di motor servo saja. Kita juga dapat menemukan sistem umpan balik loop pada perangkat rumah tangga seperti kulkas, AC, Setrika, dan masih banyak lagi perangkat rumah tangga yang menggunakan sistem umpan balik loop.

- **Bekerja Dengan sinyal PWM**  
Motor servo bekerja menggunakan sinyal PWM (Pulse Wide Modulation) atau dalam bahasa Indonesia motor servo bekerja dengan sinyal modulasi lebar pulsa melalui kabel kontrol. Dengan menggunakan sinyal tersebut, akan berfungsi menentukan posisi sudut putaran dari poros motor servo.
- **Poros Motor Bertahan pada Posisinya**  
Agar poros motor servo tetap bertahan pada posisinya kendali sinyal lebar pulsa harus diatur atau diulang setiap 20 ms (mili detik). Jika tidak dilakukan hal yang demikian motor servo tidak akan bertahan pada posisinya untuk selamanya.
- **Bagian-bagian dari Motor Servo**  
Sebagai sebuah motor, motor servo juga terdiri dari beberapa bagian yaitu, motor DC, gear, potensiometer, dan rangkaian sistem kontrol. Fungsi dari komponen gear pada motor servo yaitu berfungsi untuk memperlambat putaran utama kemudian juga dapat meningkatkan torsi putaran pada motor servo. Kemudian fungsi potensiometer yaitu sebagai penentu batas putaran utama pada motor servo dan untuk merubah hambatan atau resistansi.



Gambar A.7 Datasheet motor servo TOWER PRO SG90



#### 4. HC-SR04

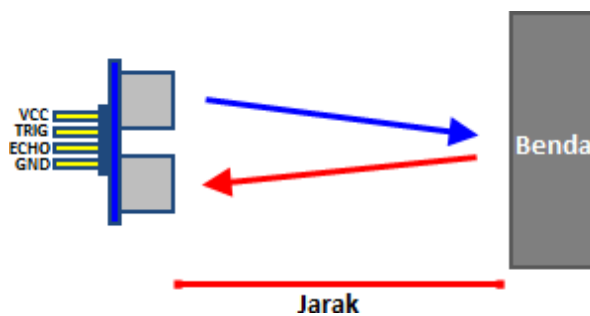
Sensor ultrasonic (Gambar A.8) adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik).

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa.



Gambar A.8 Sensor Ultrasonik tipe HC-SR04

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima (Gambar A.9).



Gambar A.9 Cara kerja sensor HC-SR04

Karena kecepatan bunyi adalah 340 m/s, maka rumus untuk mencari jarak berdasarkan ultrasonik adalah :

$$S = \frac{340 \times t}{2}$$

dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh transmitter dan waktu ketika gelombang pantul diterima receiver.

HC-SR04 merupakan sensor ultrasonik siap pakai, satu alat yang berfungsi sebagai pengirim, penerima, dan pengontrol gelombang ultrasonik. Alat ini bisa digunakan untuk mengukur jarak benda dari 2cm - 4m dengan akurasi 3mm. Dengan demikian, untuk menghitung jarak yang hanya maksimal 4 m maka rumus di atas harus dimodifikasi atau disesuaikan satuannya. Mikrokontroller bisa bekerja pada order mikrosekon ( $1s = 1.000.000 \mu s$ ) dan satuan jarak bisa kita ubah ke satuan cm ( $1m = 100 \text{ cm}$ ). Oleh sebab itu, rumus di atas bisa diupdate menjadi:

$$S = \frac{340 \left( \frac{100}{1000000} \right) t}{2}$$

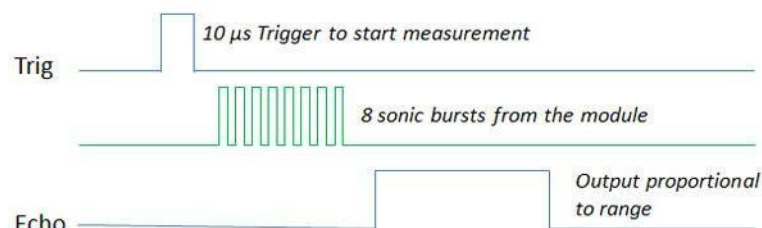
$$S = \frac{0.034 t}{2}$$

Alat ini memiliki 4 pin, pin Vcc, Gnd, Trigger, dan Echo. Pin Vcc untuk listrik positif dan Gnd untuk ground-nya. Pin Trigger untuk trigger keluarnya sinyal dari sensor dan pin Echo untuk menangkap sinyal pantul dari benda.

Cara menggunakan alat ini yaitu:

- Ketika kita memberikan tegangan positif pada pin Trigger selama  $10\mu s$ , maka sensor akan mengirimkan 8 step sinyal ultrasonik dengan frekuensi 40kHz
- Selanjutnya, sinyal akan diterima pada pin Echo
- Untuk mengukur jarak benda yang memantulkan sinyal tersebut, selisih waktu ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut
- Rumus untuk menghitung jaraknya adalah  $S = (0.034 * t) / 2 \text{ cm}$ .

Berikut adalah visualisasi dari sinyal yang dikirimkan oleh sensor HC-SR04 :



Gambar A.10 Timing HC-SR04



## Konfigurasi pin sensor HC-SR04

| Pin Number | Pin Name | Description  |
|------------|----------|--|
| 1          | Vcc      | The Vcc pin powers the sensor, typically with +5V  |
| 2          | Trigger  | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.                                       |
| 3          | Echo     | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4          | Ground   | This pin is connected to the Ground of the system.   |

### HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered:  $<15^\circ$
- Operating Current:  $<15\text{mA}$
- Operating Frequency: 40Hz