

## BAB 2

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 1.1 Tinjauan Pustaka

Bermain merupakan kegiatan yang dilakukan anak secara berulang-ulang demi kesenangan tanpa adanya tujuan dan sasaran yang hendak dicapai<sup>[1]</sup>. Untuk mencapai kesenangan tersebut, sebuah taman bermain atau wahana permainan harus memiliki keamanan yang ketat agar bermain dapat mencapai kesenangan tanpa terganggu oleh sesuatu apapun. Dalam suatu wahana permainan, pasti memiliki daya maksimum penumpangnya. Untuk mencapai keamanan yang ketat tersebut, salah satu langkah yang ditempuh yaitu menyortir pengunjung berdasarkan aspek tertentu. Disini Penyusun akan menyortir pengunjung berdasarkan tinggi dan berat badan pengunjungnya. Menurut Barry L. Johnson yang dikutip oleh Murtiantmo wibowo adi berpendapat bahwa tinggi badan merupakan ukuran posisi tubuh berdiri (vertical) dengan kaki menempel pada lantai, posisi kepala dan leher tegak, pandangan rata-rata air, dada dibusungkan, perut datar dan tarik nafas beberapa saat<sup>[2]</sup>. Sedangkan menurut Cipto Surono mengatakan bahwa berat badan adalah ukuran tubuh dalam sisi beratnya yang ditimbang dalam keadaan berpakaian *minimal* tanpa perlengkapan apapun<sup>[2]</sup>.

Sebelumnya, sudah diciptakan alat tentang penyeleksi tinggi badan otomatis pada wahana permainan berbasis *ATmega 328*<sup>[3]</sup>. Namun, menurut kami, alat tersebut belum sempurna karena alat tersebut belum mencantumkan berat badan pengunjung sebagai kriteria memasuki suatu wahana permainan. Lalu telah

ditemukan juga alat pengukur tinggi badan dan berat badan berbasis *Arduino*<sup>[4]</sup> dan alat sistem buka tutup pintu wahana safari menggunakan kamera untuk mencegah binatang keluar<sup>[5]</sup>. Dari alat – alat di atas tadi, penyusun menggabungkan dan meringkasnya menjadi satu yaitu Rancang Bangun Pintu Wahana Otomatis Menggunakan Sensor Ultrasonik sebagai Pengukur Tinggi Badan dan Sensor *Loadcell* sebagai Pengukur Berat Badan berbasis *Arduino*.

## **1.2 Sensor**

Sensor adalah komponen elektronika yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besaran listrik berupa tegangan, resistansi dan arus listrik<sup>[2]</sup>. Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian. Berdasarkan sinyal keluarannya, sensor diklasifikasikan sebagai analog atau digital. Pada sensor analog perubahan keluaran secara kontinyu pada tingkat makroskopik. Informasi ini biasanya diperoleh dari amplitudo, meskipun sensor dengan keluaran dalam domain waktu biasanya dianggap sebagai analog.

Dalam tugas akhir ini menggunakan 2 jenis sensor. Sensor yang digunakan adalah sensor ultrasonik dan sensor *Load Cell*, berikut penjelasan umumnya.

### **1.2.1 Sensor Ultrasonik HC-SR04**

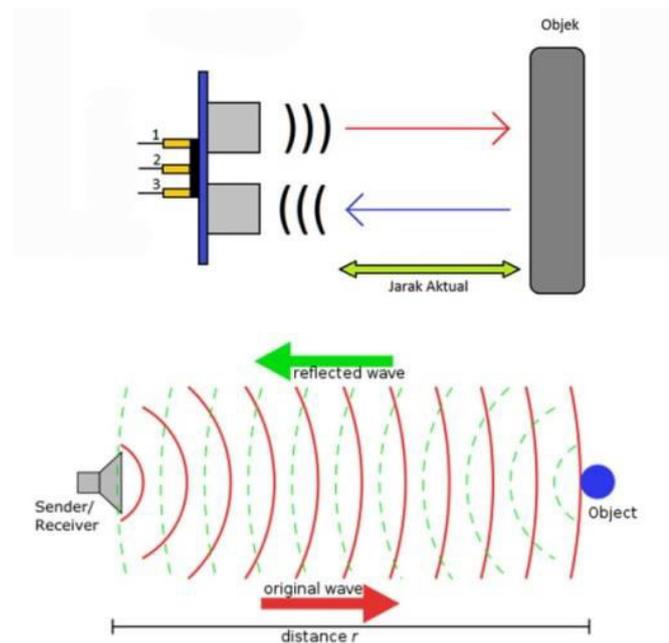
Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya<sup>[6]</sup>. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan 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  $20.000 \text{ Hz}$ <sup>[6]</sup>. 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.

#### **1.2.1.1 Cara Kerja Sensor Ultrasonik**

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi  $40 \text{ kHz}$ ) 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<sup>[6]</sup>.



**Gambar 0.1** Cara kerja sensor ultrasonik dengan *transmitter* dan *receiver*<sup>[6]</sup>

Secara detail, cara kerja sensor ultrasonik seperti pada gambar 2.1 adalah sebagai berikut:

- Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas  $20kHz$ . Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah  $40kHz$ .
- Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar  $340\text{ m/s}$ . Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
- Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus<sup>[6]</sup> :

$$S = 340.t/2$$

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*.

### 1.2.1.2 Aplikasi Sensor Ultrasonik

Dalam bidang kesehatan, gelombang ultrasonik bisa digunakan untuk melihat organ-organ dalam tubuh manusia seperti untuk mendeteksi tumor, liver, otak dan menghancurkan batu ginjal. Gelombang ultrasonik juga dimanfaatkan pada alat USG (ultrasonografi) yang biasa digunakan oleh dokter kandungan<sup>[6]</sup>.

Dalam bidang industri, gelombang ultrasonik digunakan untuk mendeteksi keretakan pada logam, meratakan campuran besi dan timah, meratakan campuran susu agar homogen, mensterilkan makanan yang diawetkan dalam kaleng, dan membersihkan benda benda yang sangat halus. Gelombang ultrasonik juga bisa digunakan untuk mendeteksi keberadaan mineral maupun minyak bumi yang tersimpan di dalam perut bumi<sup>[6]</sup>.

Dalam bidang pertahanan, gelombang ultrasonik digunakan sebagai radar atau navigasi, di darat maupun di dalam air. Gelombang ultrasonik digunakan oleh kapal pemburu untuk mengetahui keberadaan kapal selam, dipasang pada kapal selam untuk mengetahui keberadaan kapal yang berada di atas permukaan air, mengukur kedalaman palung laut, mendeteksi ranjau, dan menentukan posisi sekelompok ikan<sup>[6]</sup>.

### 1.2.1.3 Rangkaian Sensor Ultrasonik

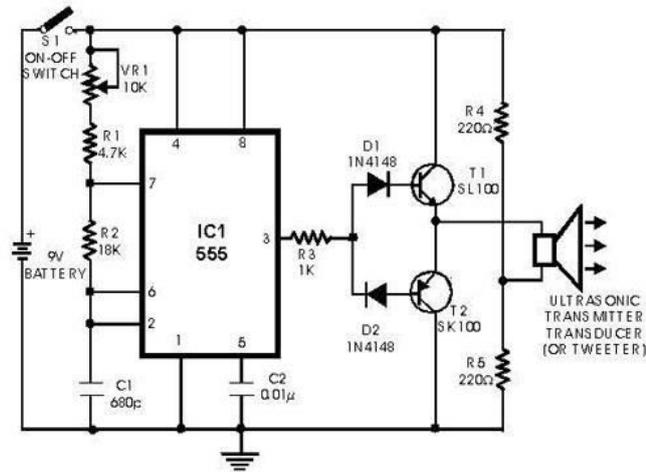
#### 1. Piezoelektrik

Piezoelektrik berfungsi untuk mengubah energi listrik menjadi energi mekanik. Bahan piezoelektrik adalah material yang memproduksi medan listrik ketika dikenai regangan atau tekanan mekanis. Sebaliknya, jika medan listrik diterapkan, maka material tersebut akan mengalami regangan atau tekanan mekanis. Jika rangkaian pengukur beroperasi pada mode pulsa elemen piezoelektrik yang sama, maka dapat digunakan sebagai *transmitter* dan *receiver*<sup>[6]</sup>. Frekuensi yang ditimbulkan tergantung pada osilatornya yang disesuaikan frekuensi kerja dari masing-masing transduser. Karena kelebihan ini maka transduser piezoelektrik lebih sesuai digunakan untuk sensor ultrasonik.

#### 2. Transmitter

*Transmitter* adalah sebuah alat yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi tertentu (misal, sebesar 40 *KHz*) yang dibangkitkan dari sebuah osilator<sup>[6]</sup>. Untuk menghasilkan frekuensi 40 *KHz*, harus di buat sebuah rangkaian osilator dan keluaran dari osilator dilanjutkan menuju penguat sinyal. Besarnya frekuensi ditentukan oleh komponen *RLC* / kristal tergantung dari disain osilator yang digunakan. Penguat sinyal akan memberikan sebuah sinyal listrik yang diumpankan ke piezoelektrik dan terjadi reaksi mekanik sehingga bergetar dan memancarkan gelombang yang sesuai dengan besar frekuensi pada osilator.

Gambar 2.2 adalah rangkaian dasar dari *transmitter* ultrasonik *HC-SR04*.

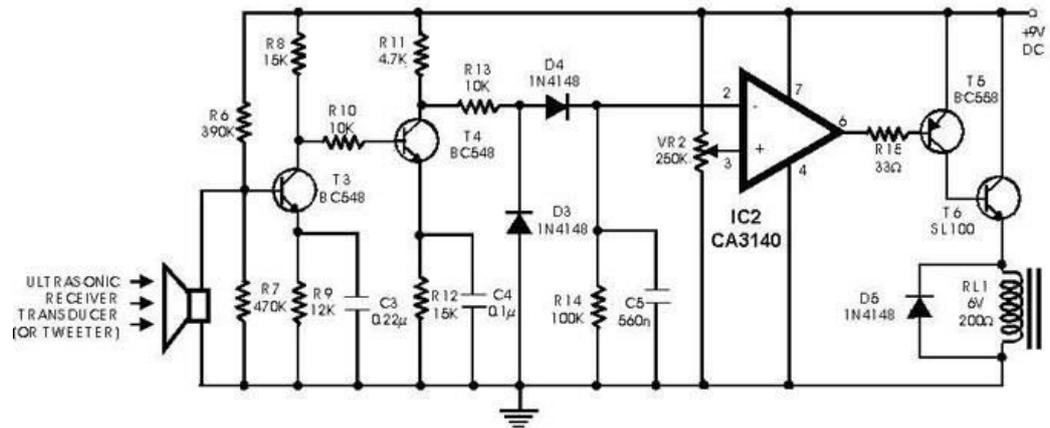


**Gambar 0.2** Rangkaian dasar dari transmitter ultrasonik<sup>[6]</sup>

### 3. Receiver

*Receiver* terdiri dari transduser ultrasonik menggunakan bahan piezoelektrik, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari *transmitter* yang dikenakan pada permukaan suatu benda atau gelombang langsung *LOS (Line of Sight)* dari *transmitter*<sup>[6]</sup>. Oleh karena bahan piezoelektrik memiliki reaksi yang reversible, elemen keramik akan membangkitkan tegangan listrik pada saat gelombang datang dengan frekuensi yang resonan dan akan menggetarkan bahan piezoelektrik tersebut.

Gambar 2.3 adalah gambar rangkaian dari *receiver* sensor ultrasonik *HC-SR04*.



Gambar 0.3 Rangkaian dasar receiver sensor ultrasonik<sup>[6]</sup>

#### 1.2.1.4 Cara Kerja Sensor Ultrasonik *HC-SR04*

Sensor ini 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. 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.

Gambar 2.4 adalah gambar bentuk fisik dari sensor ultrasonik *HC-SR04*.

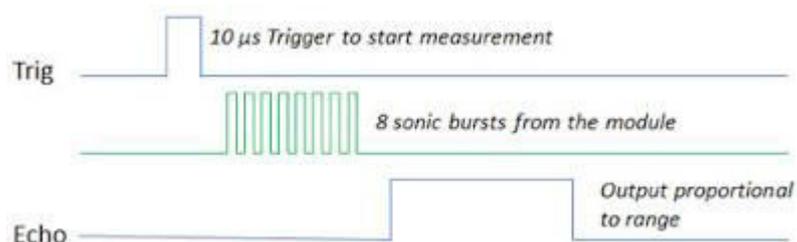


Gambar 0.4 Sensor Ultrasonik *HC-SR04*<sup>[6]</sup>

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, maka selisih waktu ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut. Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus (2-1) :

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*.

Gambar 2.5 adalah visualisasi dari sinyal yang dikirimkan oleh sensor *HC-SR04*



**Gambar 0.5** Sistem pewaktu pada sensor *HC-SR04*<sup>[6]</sup>

Adapun spesifikasi dari sensor HCSR – 04 ditunjukkan pada tabel 2.1

**Tabel 0.1** Spesifikasi Sensor *HC-SR04*<sup>[6]</sup>

No	Spesifikasi	Keterangan
1.	<b>Jangkauan deteksi</b>	<b>2cm s/d 400 -500cm</b>
2.	<b>Sudut deteksi terbaik</b>	<b>15 derajat</b>
3.	<b>Tegangan kerja</b>	<b>5V DC</b>
4.	<b>Resolusi</b>	<b>1cm</b>
5.	<b>Frekuensi Ultrasonik</b>	<b>40 kHz</b>
6.	<b>Dapat dihubungkan langsung ke kaki mikrokontroler</b>	<b>4 kaki pin</b>

Fungsi dari masing – masing PIN sensor Ultrasonik ini adalah sebagai berikut :

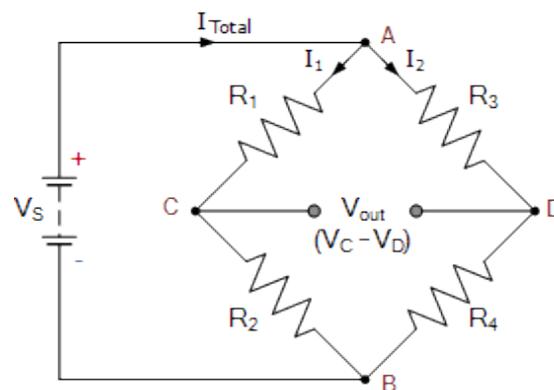
- 1) *VCC* = 5V Power Supply. Pin sumber tegangan positif sensor.
- 2) Trig = *Trigger*/Penyulut. Pin ini yang digunakan untuk membangkitkan sinyal ultrasonik.
- 3) Echo = Receive/Indikator. Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik.
- 4) *GND* = *Ground*/0V Power Supply. Pin sumber tegangan negatif sensor.

### 1.2.2 Jembatan Wheatstone

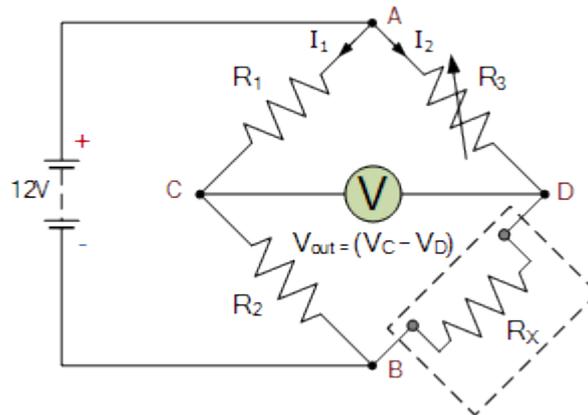
Jembatan *Wheatstone* awalnya dikembangkan oleh Charles *Wheatstone* untuk mengukur nilai-nilai perlawanan yang tidak diketahui dan sebagai sarana untuk mengkalibrasi mengukur instrumen, voltmeters, ammeters, dll, dengan menggunakan kawat panjang *resistif slide*<sup>[7]</sup>.

Jembatan *Wheatstone* (atau jembatan perlawanan) sirkuit dapat digunakan dalam sejumlah aplikasi dan hari ini, dengan amplifier operasional modern kita dapat menggunakan sirkuit *Wheatstone* jembatan untuk antarmuka berbagai transduser dan sensor.

Sirkuit *Wheatstone* jembatan ini tidak lebih dari dua pengaturan seri-paralel sederhana resistansi terhubung antara tegangan suplai terminal dan tanah yang memproduksi nol tegangan perbedaan antara dua cabang paralel ketika seimbang<sup>[7]</sup>. Rangkaian *Wheatstone* bridge memiliki dua terminal input dan dua output terminal yang terdiri dari empat resistor dikonfigurasi dalam pengaturan seperti pada gambar 2.6



**Gambar 0.6** Rangkaian Jembatan *Wheatstone*<sup>[7]</sup>



**Gambar 0.7** Rangkain Jembatan *Wheatstone*<sup>[7]</sup>

Pada gambar 2.7 dengan mengganti R4 di dengan ketahanan nilai yang dikenal atau tidak dikenal di lengan penginderaan jembatan *Wheatstone* sesuai dengan Rx dan menyesuaikan resistor lawan, R3 untuk "keseimbangan" jembatan jaringan, akan menghasilkan output tegangan yang nol. Kemudian kita dapat melihat bahwa keseimbangan terjadi saat<sup>[7]</sup> :

$$\frac{R_1}{R_2} = \frac{R_3}{R_X} = 1 \text{ (balance)}$$

Persamaan *Wheatstone* jembatan yang diperlukan untuk memberikan nilai perlawanan tidak diketahui,  $R_X$  di balance yang diberikan<sup>[7]</sup>:

$$V_{out} = (V_C - V_D) = (V_{R2} - V_{R4}) = 0$$

$$R_C = \frac{R_2}{R_1 + R_2} \text{ Dan } R_D = \frac{R_4}{R_3 + R_4}$$

Di saat seimbang maka  $R_C = R_D$  menjadi  $\frac{R_2}{R_1 + R_2} = \frac{R_4}{R_3 + R_4}$  maka :

$$R_2(R_3 + R_4) = R_4(R_1 + R_2)$$

$$R_2R_3 + R_2R_4 = R_1R_4 + R_2R_4$$

$$R_4 = \frac{R_2R_3}{R_1} = R_X$$

Dengan berdasarkan persamaan diatas, untuk menghitung Vout yang ada pada jembatan *Wheatstone* adalah<sup>[7]</sup> :

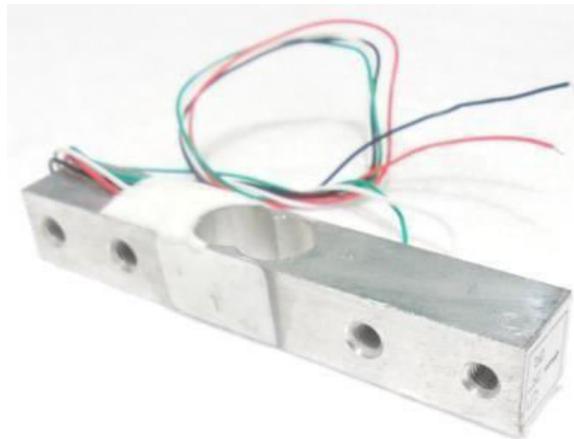
$$V_o = \left( V_{In} \times \left( \frac{R_2}{R_1 + R_2} \right) \right) - \left( V_{In} \times \left( \frac{R_4}{R_3 + R_4} \right) \right)$$

### 1.2.3 Sensor *Load Cell*

*Load Cell* adalah sebuah alat uji perangkat listrik yang dapat mengubah suatu energi menjadi energi lainnya yang biasa digunakan untuk mengubah suatu gaya menjadi sinyal listrik<sup>[8]</sup>.

Sensor *Load Cell* merupakan sensor yang dirancang untuk mendeteksi tekanan atau berat sebuah beban<sup>[8]</sup>, sensor *Load Cell* umumnya digunakan sebagai komponen utama pada sistem timbangan digital dan dapat diaplikasikan pada jembatan timbangan yang berfungsi untuk menimbang berat dari truk pengangkut bahan baku, pengukuran yang dilakukan oleh *Load Cell* menggunakan prinsip tekanan.

Gambar 2.8 adalah gambar dari bentuk fisik sensor *Load Cell*



**Gambar 0.8** Sensor *Load Cell*<sup>[4]</sup>

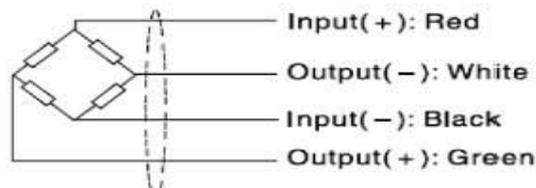
Keterangan gambar :

Kabel merah adalah input tegangan sensor.

Kabel hitam adalah input *ground* sensor.

Kabel hijau adalah output positif sensor.

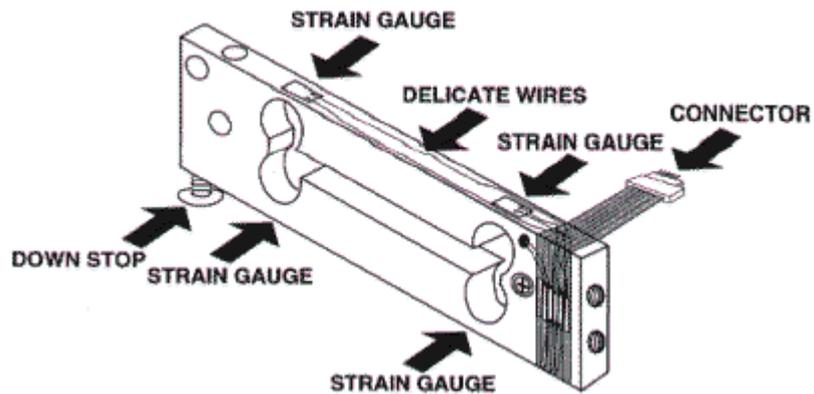
Kabel putih adalah output negatif sensor.



**Gambar 0.9** Konfigurasi Kabel Sensor *Load Cell*<sup>[4]</sup>

Gambar 2.9 adalah konfigurasi kabel dari sensor *Load Cell*. yang terdiri dari kabel berwarna merah, hitam, biru, dan putih. Kabel merah merupakan input tegangan sensor, kabel hitam merupakan input *ground* pada sensor, kabel warna biru / hijau merupakan output positif dari sensor dan kabel putih adalah output *ground* dari sensor. Nilai tegangan output dari sensor ini sekitar 1,2 mV<sup>[4]</sup>. Gambar 2.7 adalah gambaran dari penjelasan tentang konfigurasi kabel pada sensor *Load Cell*.

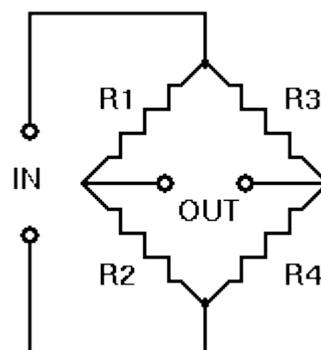
Selama proses penimbangan akan mengakibatkan reaksi terhadap elemen logam pada *Load Cell* yang mengakibatkan gaya secara elastis. Gaya yang ditimbulkan oleh regangan ini dikonversikan kedalam sinyal elektrik oleh *strain gauge* (pengukur regangan) yang terpasang pada *Load Cell*<sup>[9]</sup>. Pada gambar 2.8 akan ditunjukkan tentang posisi *strain gauge* pada sensor *Load Cell* tersebut.



**Gambar 0.10** Posisi *Strain Gauge*<sup>[9]</sup>

Pada gambar 2.10 telah dijelaskan posisi dari *strain gauge*. Dari posisi *strain gauge* yang seperti pada gambar 2.10, maka perhitungan *strain gauge* ini menggunakan prinsip Jembatan *Wheatstone*. *Strain gauge* adalah perangkat yang mengukur perubahan hambatan listrik dalam menanggapi, dan proporsional, ketegangan yang diterapkan ke perangkat..

Prinsip kerja *Load Cell* berdasarkan rangkaian Jembatan *Wheatstone* dapat dilihat pada gambar 2.11.



**Gambar 0.11** Rangkaian Jembatan *Wheatstone* Pada Sensor *Load Cell*<sup>[9]</sup>

Jika rangkaian jembatan *Wheatstone* diberi beban, maka nilai R pada rangkaian akan berubah, nilai  $R_1 = R_4$  dan  $R_2 = R_3$ . Sehingga membuat sensor

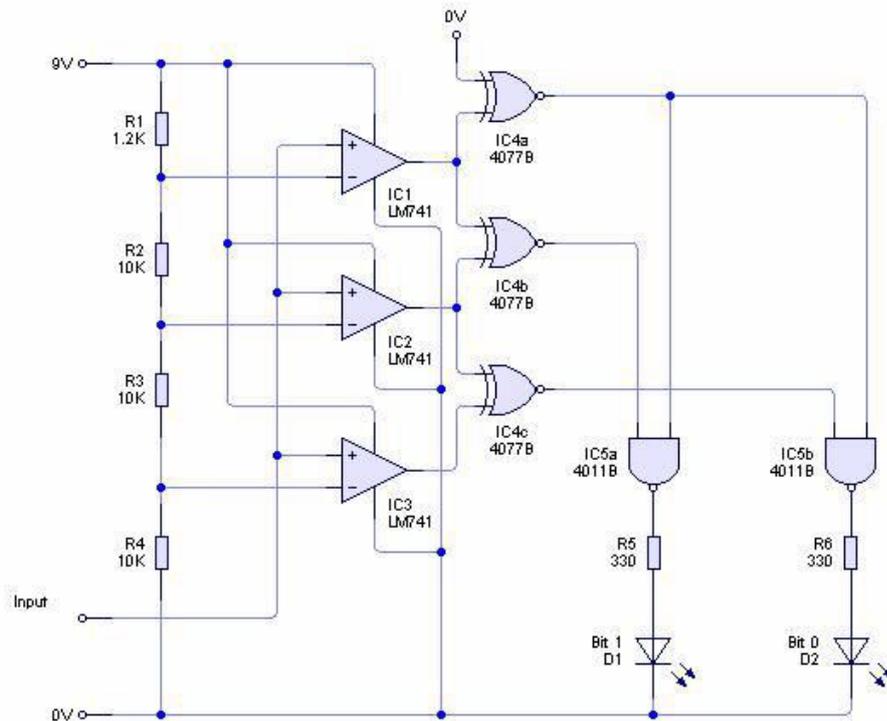
*Load Cell* tidak dalam kondisi yang seimbang dan membuat beda potensial. Beda potensial inilah yang menjadi outputnya. Untuk menghitung  $V_{out}$  atau  $A$  seperti pada gambar, maka rumus yang digunakan adalah sebagai berikut<sup>[4]</sup> :

$$V_o = \left( V_{In} \times \left( \frac{R_2}{R_1 + R_2} \right) \right) - \left( V_{In} \times \left( \frac{R_4}{R_3 + R_4} \right) \right)$$

Secara teori, prinsip kerja *Load Cell* berdasarkan pada jembatan *Wheatstone* dimana saat *Load Cell* diberi beban terjadi perubahan pada nilai resistansi, nilai resistansi  $R_1$  dan  $R_3$  akan turun sedangkan nilai resistansi  $R_2$  dan  $R_4$  akan naik ketika posisi setimbang,  $V_{out}$  *Load Cell* = 0 volt, namun ketika nilai resistansi  $R_1$  dan  $R_3$  naik maka akan terjadi perubahan  $V_{out}$  pada *Load Cell*. Pada *Load Cell* output data (+) dipengaruhi oleh perubahan resistansi pada  $R_1$ , sedangkan output (-) dipengaruhi oleh perubahan resistansi  $R_3$ . Karena keluaran tegangannya yang bernilai Volt, susah untuk dideteksi oleh *Arduino*, oleh karena itu butuh ditambahkan penguatan berupa *Programmable Gain Amplifier* (PGA) yang berada pada ADC (*Analog Digital Converter*).

### 1.3 ADC (*Analog Digital Converter*)

**ADC** adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinyu) menjadi data digital. Perangkat ADC (*Analog To Digital Conversion*) dapat berbentuk suatu modul atau rangkaian elektronika maupun suatu chip IC. ADC (*Analog To Digital Converter*) berfungsi untuk menjembatani pemrosesan sinyal analog oleh sistem digital<sup>[10]</sup>. Berikut Gambar 2.7 adalah gambar rangkaian dari ADC secara umum.



**Gambar 2.7** Rangkaian ADC<sup>[10]</sup>

### 1.3.1 Prinsip Kerja Rangkaian ADC

Rangkaian adc pada gambar 2.7 memanfaatkan rangkaian pembanding op-amp sebagai rangkaian dasar. Dimana perbedaan yang sedikit pada kedua terminal *input* op-amp akan menghasilkan tegangan sebesar Vdd atau Vcc op-amp. Jika tegangan pada terminal positif *input* lebih besar dari pada terminal *negative input* maka keluaran adalah 9 volt (sesuai dengan Vdd), sedangkan jika tegangan pada terminal *negative input* lebih besar maka tegangan keluarannya adalah 0 volt (sesuai dengan Vcc). Menggunakan 3 (tiga) buah op-amp dengan tujuan setiap satu op-amp mewakili satu jangkah pembagian tegangan input. Pada masing-masing terminal *negative input* op-amp mendapatkan tegangan referensi (penentuan) yang ditentukan oleh pembagian tegangan antara R1, R2, R3 dan R4.

R2, R3 dan R4 sengaja dibuat dengan nilai yang sama dengan maksud supaya tegangan pada terminal negative (referensi) masing-masing op-amp membentuk jangkah atau range yang teratur. Masing-masing terminal positif input op-amp digabung dan digunakan sebagai jalur input sinyal analog. Hal ini sengaja diatur supaya posisi sinyal input analog tersebut bisa dibaca oleh masing-masing op-amp yang mana pada masing-masing terminal negative input op-amp tersebut sudah dipasang tegangan penentu.

IC3 mewakili range tegangan terendah, kemudian dilanjutkan oleh IC2, IC1 mewakili range tertinggi.

Tegangan pada terminal negative input IC3 adalah

$$\left( \frac{R_4}{(R_1 + R_2 + R_3 + R_4)} \right) \times 9 \text{ Volt} = \left( \frac{10K}{31,2K} \right) \times 9 \text{ Volt} = 2,89 \text{ Volt}$$

Tegangan pada terminal negative input IC2 adalah

$$\left( \frac{R_3 + R_4}{(R_1 + R_2 + R_3 + R_4)} \right) \times 9 \text{ Volt} = \left( \frac{20K}{31,2K} \right) \times 9 \text{ Volt} = 5,77 \text{ Volt}$$

Tegangan pada terminal negative input IC1 adalah

$$\left( \frac{R_2 + R_3 + R_4}{(R_1 + R_2 + R_3 + R_4)} \right) \times 9 \text{ Volt} = \left( \frac{30K}{31,2K} \right) \times 9 \text{ Volt} = 8,65 \text{ Volt}$$

Jadi dari perhitungan tegangan referensi pada terminal negative input ketiga op-amp tersebut adalah mempunyai delta atau jangkah tegangan 2.88 volt. Tegangan 2,88 volt ini yang saya sebut sebagai jangkah tegangan referensi atau penentu. Jadi bisa disimpulkan bahwa rangkaian diatas akan membaca sinyal input analog :

- 0 sd 2,88 volt sebagai angka 0

- > 2,88 volt sd 5,77 volt sebagai angka 1

- > 5,77 volt sd 8,65 volt sebagai angka 2

- > 8,65 volt sebagai angka 3

*Analog To Digital Converter* (ADC) adalah pengubah input analog menjadi kode – kode digital. ADC banyak digunakan sebagai pengatur proses industri, komunikasi digital dan rangkaian pengukuran/ pengujian. Umumnya ADC digunakan sebagai perantara antara sensor yang kebanyakan analog dengan sistim komputer seperti sensor suhu, cahaya, tekanan/ berat, aliran dan sebagainya kemudian diukur dengan menggunakan sistim digital (komputer). ADC (*Analog to Digital Converter*) memiliki 2 karakter prinsip, yaitu kecepatan sampling dan resolusi

ADC (*Analog to Digital Converter*) memiliki 2 karakter prinsip, yaitu kecepatan sampling dan resolusi.

### 1. Kecepatan Sampling ADC

Kecepatan sampling suatu ADC menyatakan “seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu”. Kecepatan sampling biasanya dinyatakan dalam sample per second (SPS).

### 2. Resolusi ADC

Resolusi ADC menentukan “ketelitian nilai hasil konversi ADC”. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 ( $2^n - 1$ ) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit. ADC 24 bit akan memiliki output 24 bit data digital yang berarti sinyal input dapat dinyatakan dalam 16777215 nilai diskrit. Dari contoh diatas ADC 24

bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 12 bit dan 8 bit.

### 3. Prinsip Kerja ADC

Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%.

Jadi, jika menggunakan ADC 24 bit dengan skala maksimum 16777215, akan didapatkan sinyal digital sebesar  $60\% \times 16777215 = 10066329$  (bentuk decimal) atau 999999 dalam bentuk hexadecimal.

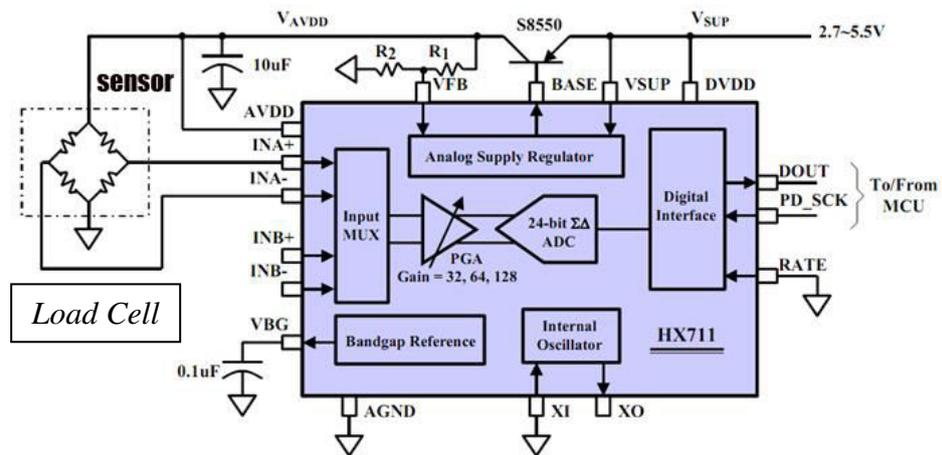
$$\begin{aligned} \text{signal} &= (\text{sample}/\text{max\_value}) * \text{reference\_voltage} \\ &= (10066329/16777215) * 5 \\ &= 3 \text{ volt} \end{aligned}$$

#### 1.3.2 ADC Modul HX711

HX711 adalah sebuah komponen modul presisi 24-bit analog to digital converter (ADC) yang didesain untuk sensor timbangan digital dan industrial control aplikasi yang terkoneksi sensor jembatan<sup>[4]</sup>.

HX711 adalah modul timbangan, yang memiliki prinsip kerja mengkonversi perubahan yang terukur dalam perubahan resistansi dan mengkonversinya ke dalam besaran tegangan melalui rangkaian yang ada<sup>[4]</sup>. Modul melakukan komunikasi dengan computer/mikrokontroler melalui TTL232. Struktur yang sederhana, mudah dalam penggunaan, hasil yang stabil dan *reliable*, memiliki

sensitivitas tinggi, dan mampu mengukur perubahan dengan cepat. Pada gambar 2.12 digambarkan tentang diagram blok kerja dari modul ADC HX711 terhadap sensor *Load Cell*.

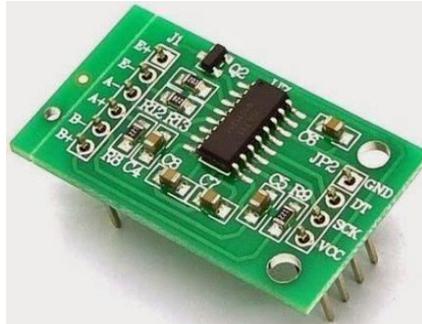


**Gambar 0.12** Blok Diagram *HX711*<sup>[11]</sup>

Pada gambar 2.12 bahwa sensor *Load Cell* membutuhkan daya sebesar 5 volt dari ADC HX711 dengan media kabel berwarna merah yang tersambung pada AVDD. Sensor *Load Cell* juga membutuhkan ground pada kabel hitam yang nantinya akan disambungkan pada ground (pada gambar 2.13 pada port 2). Untuk menerima data dari *Load Cell*, ADC HX711 menerima tegangan output dari kabel hitam untuk negatif yang disambungkan pada INA- dan kabel hijau untuk positif yang disambungkan pada INA+ (pada gambar 2.13 kabel hitam ke port 3 dan kabel hijau ke port 4). Setelah data diterima oleh ADC HX711, data dari INA+ dan INA- akan di kirim ke Multiplexer yang nanti outputan dari proses multiplexer akan di perkuat oleh PGA (Programmable Gain Amplifier). Setelah dikuatkan, maka akan dirubah dari sinyal yang awalnya analog menjadi digital pada 24 bit ADC. Setelah itu keluaran data yang diproses tadi dari port DOUT dan



Gambar 2.14 adalah bentuk fisik dari modul *ADC HX711*



**Gambar 0.14** Modul *ADC HX711*<sup>[4]</sup>

Untuk Mengukur besaran *ADC* pada modul *ADC HX711* menggunakan rumus<sup>[9]</sup> :

$$\text{signal} = (\text{sample}/\text{max\_value}) * \text{reference\_voltage}$$

Keterangan :

Signal = Tegangan Output *ADC* pada *DOUT*

Sample = Output tegangan analog dari sensor *Load Cell* pada *AVDD*

reference\_voltage = Tegangan referensi dari modul *HX711* yaitu 5V

Maksimal Data = Output 24 bit data digital yaitu  $2^{24} - 1 = 16777215$

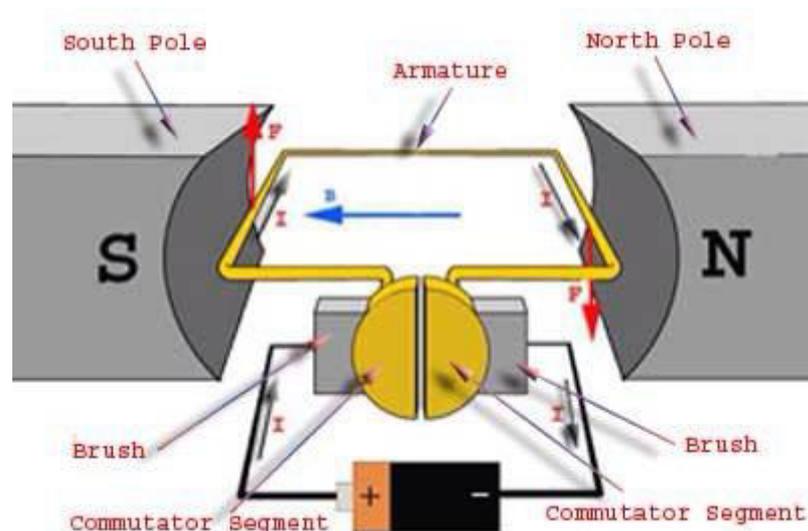
Jika menggunakan *ADC* 24 bit dengan skala maksimum 16777215, akan didapatkan sinyal digital sebesar  $60\% \times 16777215 = 10066329$  (bentuk decimal) atau 999999 dalam bentuk hexadecimal.

$$\begin{aligned} \text{signal} &= (\text{sample}/\text{max\_value}) * \text{reference\_voltage} \\ &= (10066329/16777215) * 5 \\ &= 3 \text{ volt} \end{aligned}$$

## 1.4 Motor

Motor listrik merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya, memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dan lain sebagainya. Motor listrik digunakan juga di rumah (mixer, bor listrik, fan atau kipas angin) dan di industri. Motor listrik dalam dunia industri seringkali disebut dengan istilah “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industry <sup>[12]</sup>.

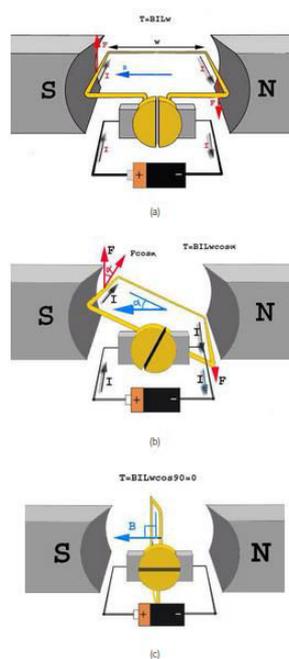
Motor DC merupakan jenis motor yang paling sering digunakan di dalam dunia robotika. Salah satu alasannya adalah arah putaran motor DC, baik searah jarum jam maupun berlawanan arah jarum jam, dapat dengan mudah dikendalikan.



**Gambar 0.15** Bagian Bagian Motor Listrik<sup>[12]</sup>

Pada gambar 2.15 ditunjukkan tentang gambar skema motor listrik DC. Pada skema gambar 2.15, rotor motor diskemakan dengan sebuah kawat angker

penghantar listrik (armature) yang membentuk persegi panjang. Pada kedua ujung kawat angker terpasang komutator berbentuk lingkaran yang terbelah di tengahnya, komponen ini sering kita dengar dengan sebutan cincin belah. Cincin belah termasuk bagian dari rotor, sehingga ia ikut berputar dengan rotor. Sedangkan stator motor tersusun atas dua magnet dengan kutub berbeda yang saling berhadapan. Pada bagian yang kontak langsung dengan cincin belah, stator dilengkapi dengan sikat karbon yang berfungsi untuk menghubungkan arus listrik dari sumber tegangan ke kumparan rotor. Sumber tegangan DC diilustrasikan dengan gambar baterai pada skema motor DC di atas. Masing-masing kutub baterai terhubung dengan sikat karbon, sehingga tercipta arus listrik DC dengan arah arus dari kutub positif ke negatif melewati sikat karbon, satu bagian cincin belah, kawat angker (armature), kembali ke cincin belah, sikat karbon dan ke kutub negatif baterai.



**Gambar 0.16** Proses Berputarnya Rotor Motor Listrik DC<sup>[12]</sup>

Gambar 2.16 adalah gambaran dari proses berputarnya Rotor Motor Listrik DC. Pada gambar 2.16 (a), garis medan magnet mengarah ke kiri yang disimbolkan dengan garis biru dan huruf (B). Untuk arah arus listrik ditunjukkan dengan garis berwarna hitam dan huruf (I). Jika kita mencoba menggunakan kaidah tangan kiri kita pada sisi kiri kawat angker, maka akan kita dapatkan bahwa gaya dorong (F) akan mengarah ke atas. Sedangkan untuk sisi kanan kawat angker, kaidah tangan kiri akan menunjukkan bahwa gaya dorong akan mengarah ke bawah. Gaya dorong yang tegak lurus langsung terhadap kawat angker kanan dan kiri ini menghasilkan torsi yang paling besar pada rotor motor. Gaya torsi inilah yang akan memutar rotor motor.

Pada posisi rotor seperti Gambar 2.16 (b), masing-masing cincin belah masih terhubung dengan sikat karbon sehingga arah arus listrik tidak berubah. Dengan cara yang sama menggunakan kaidah tangan kiri, arah gaya dorong juga mengarah ke atas untuk kawat angker kiri dan ke bawah untuk kawat angker kanan. Namun besar gaya torsi yang terjadi adalah lebih kecil sebesar  $\cos \alpha$  daripada gaya F. Gaya torsi ini masih akan membuat rotor motor berputar searah jarum jam.

Torsi rotor akan menjadi nol pada saat kawat angker berposisi seperti pada gambar 2.16 (c). Sesuai dengan kaidah tangan kiri, jika pada kawat angker terdapat arus listrik, maka arah gaya dorong kawat juga ke atas atau pun ke bawah. Namun karena gaya tersebut segaris dengan titik poros rotor, atau dapat pula dikatakan tegak lurus dengan arah putaran rotor, maka tidak akan timbul gaya torsi pada kawat angker. Sudut  $\alpha$  yang sebesar  $90^\circ$  menjelaskan pula tidak

akan timbul gaya torsi pada saat posisi kawat angker demikian, karena nilai dari  $\cos 90^\circ$  adalah nol. Nilai torsi nol ini tidak akan membuat rotor motor berhenti berputar, karena sifat kelembaman rotor maka rotor akan terus berputar selama masih ada arus listrik yang mengalir pada kawat angker. Setelah kawat angker melewati fase tegak lurus dan membentuk sudut  $-\alpha$ , arah arus listrik akan mengalir dengan arah yang sama seperti pada saat kawat angker bersudut  $+\alpha$  (gambar b). Komponen komutator yang selalu ikut berputar dengan rotor dan sikat karbon yang selalu diam, menjadi komponen yang akan menjaga arah arus listrik untuk selalu tetap yakni --sesuai gambar skema-- mengalir dari sisi kiri kawat angker ke kanan. Arah arus listrik yang selalu tetap di setiap setengah putaran rotor inilah yang akan membuat rotor motor listrik selalu berputar selama masih ada arus listrik yang mengalir ke kawat angker.

#### **1.4.1 Motor Servo DC**

Motor *servo* adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (*servo*), sehingga dapat di *set-up* atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor<sup>[12]</sup>. Motor *servo* merupakan perangkat yang terdiri dari motor *DC*, serangkaian gear, rangkaian kontrol dan potensiometer. Serangkaian gear yang melekat pada poros motor *DC* akan memperlambat putaran poros dan meningkatkan torsi motor *servo*, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran

poros motor *servo*. Dalam alat ini, penyusun menggunakan motor *servo* tipe *FeeTech FS5109M* dengan kecepatan 0,18 detik/60 derajat pada tegangan 4,8 volt.

Gambar 2.17 adalah bentuk fisik dari motor servo jenis *FeeTech FS5109M* yang penyusun gunakan.

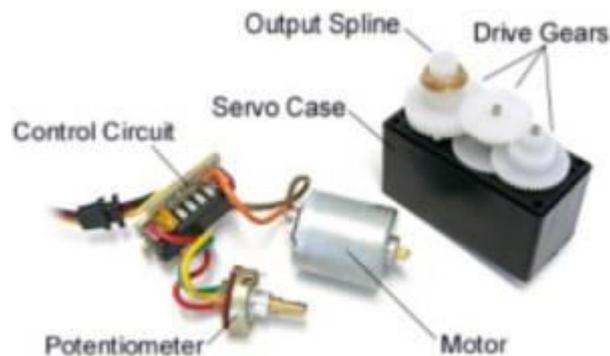


**Gambar 0.17** Motor *Servo FeeTech FS5109M*<sup>[13]</sup>

#### 1.4.1.1 Komponen Penyusun Motor *Servo*

Motor *servo* pada dasarnya dibuat menggunakan motor *DC* yang dilengkapi dengan controler dan sensor posisi sehingga dapat memiliki gerakan 0°, 90°, 180° atau 360°. Berikut adalah komponen internal sebuah motor *servo* 180°.

Pada Gambar 2.18 dijelaskan tentang bagian bagian internal motor servo.



**Gambar 0.18** Komponen Internal Motor *Servo*<sup>[12]</sup>

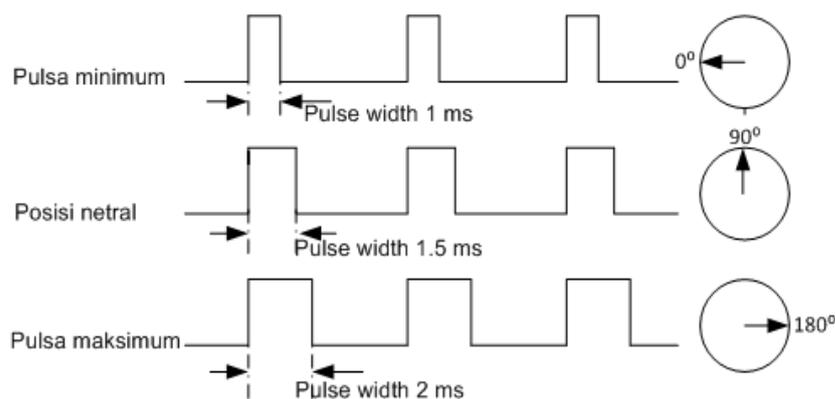
Tiap komponen pada motor *servo* diatas masing-masing memiliki fungsi sebagai controler, driver, sensor, girbox dan aktuator. Pada gambar diatas terlihat beberapa bagian komponen motor *servo*. Motor pada sebuah motor *servo* adalah motor *DC* yang dikendalikan oleh bagian *controler*, kemudian komponen yang berfungsi sebagai sensor adalah potensiometer yang terhubung pada sistem girbox pada motor *servo*<sup>[11]</sup>.

#### 1.4.1.2 Prinsip Kerja Motor *Servo*

Motor *servo* dikendalikan dengan memberikan sinyal modulasi lebar pulsa (Pulse Wide Modulation / PWM) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor *servo*. Ton duty cycle Motor *Servo* akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi Ton duty cycle 1 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0°/ netral).

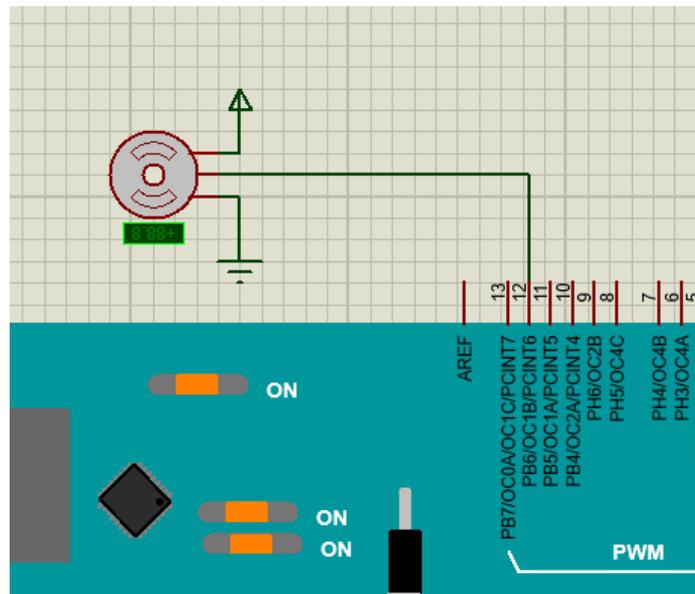
Pada saat Ton duty cycle dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (Counter Clock wise, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya Ton duty cycle, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika Ton duty cycle dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier pula terhadap besarnya Ton duty cycle, dan bertahan diposisi tersebut. Jika Ton duty cycle dari sinyal yang diberikan dengan waktu 1,5 ms (mili detik) akan memutar poros

motor *servo* ke posisi sudut  $90^{\circ}$ . Jika Ton duty cycle dari sinyal yang diberikan dengan waktu 2 ms (mili detik) akan memutar poros motor *servo* ke posisi sudut  $180^{\circ}$ . Pada gambar 2.19 ditunjukkan ilustrasi dari penjelasan tentang prinsip kerja motor servo.



**Gambar 0.19** Sinyal *PWM* Motor *Servo*<sup>[12]</sup>

Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (rating torsi servo). Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya. Gambar 2.20 adalah gambar dari rangkaian Motor Servo yang akan disusun gunakan.

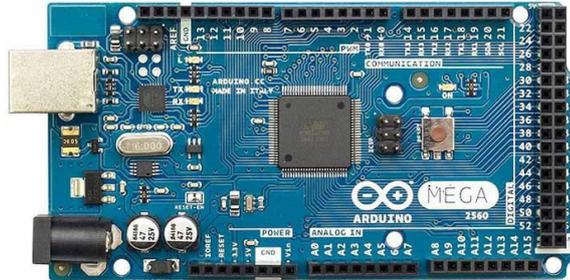


**Gambar 0.20** Rangkaian Motor Servo

### 1.5 *Arduino Mega 2560*

*Arduino Mega 2560* adalah piranti mikrokontroler menggunakan *ATmega2560*. Modul ini memiliki 54 digital input atau output. Dimana 14 pindigunakan untuk PWM output dan 16 pin digunakan sebagai analog input, 4 pin untuk UART, 16 MHz osilator kristal, koneksi USB, power jack ICSP header, dan tombol reset. Modul ini memiliki segala yang dibutuhkan untuk memprogram mikrokontroler seperti kabel USB dan catu daya melalui adaptor atau baterai. Semua ini diberikan untuk mendukung pemakain mikrokontroler *Arduino*, hanya terhubung ke komputer dengan kabel USB atau listrik dengan adaptor dari AC ke DC atau baterai untuk memulai pemakaian. *Arduino Mega* kompatibel dengan shiled yang dirancang untuk *Arduino Duemilanove*, *Decimila* maupun UNO<sup>[12]</sup>.

Gambar 2.21 adalah foto dari bentuk fisik dari *Arduino Mega 2560* yang penyusun gunakan.



**Gambar 0.21** *Arduino Mega 2560*<sup>[14]</sup>

Pada tabel 2.2, akan di uraikan spesifikasi dari *Arduino Mega 2560* yang penyusun gunakan.

**Tabel 0.2** Spesifikasi *Arduino Mega 2560*<sup>[14]</sup>

No	Spesifikasi	Keterangan
1	Mikrokontroler	ATmega 2560
2	<i>Operating Voltage</i>	5V
3	<i>Input Voltage</i> (recommended)	7 – 12 V
4	<i>Input Voltage</i> (limits)	6 – 20 V
5	Digital I/O Pins	54 (15 PWM output)
6	Analog Input Pins	16
7	<i>DC current</i> for I/O pin	40 mA
8	<i>DC current</i> for 3.3 V pin	50 mA
9	Flash Memory	256 KB (8 KB digunakan untuk bootloader)

No	Spesifikasi	Keterangan
10	SRAM	8 KB
11	EEPROM	4 KB
12	<i>Clock Speed</i>	16 MHz
13	Dimensi	101.5 mm x 53.4 mm
14	Berat	37g

### 1.5.1 *Proteksi Arduino Mega 2560*

Development board *Arduino Mega 2560 R3* telah dilengkapi dengan polyfuse yang dapat direset untuk melindungi port USB komputer/laptop anda dari korsleting atau arus berlebih. Meskipun kebanyakan komputer telah memiliki perlindungan port tersebut didalamnya namun sikring pelindung pada *Arduino Uno* memberikan lapisan perlindungan tambahan yang membuat anda bisa dengan tenang menghubungkan *Arduino* ke komputer anda. Jika lebih dari 500mA ditarik pada port USB tersebut, sirkuit proteksi akan secara otomatis memutuskan hubungan, dan akan menyambung kembali ketika batasan aman telah kembali<sup>[14]</sup>.

### 1.5.2 *Power Supply*

Board *Arduino Mega 2560* dapat ditenagai dengan power yang diperoleh dari koneksi kabel USB, atau via power supply eksternal. Pilihan power yang digunakan akan dilakukan secara otomatis.

External power supply dapat diperoleh dari adaptor AC-DC atau bahkan baterai, melalui jack DC yang tersedia, atau menghubungkan langsung GND dan

pin Vin yang ada di board. Board dapat beroperasi dengan power dari external power supply yang memiliki tegangan antara 6V hingga 20V. Namun ada beberapa hal yang harus anda perhatikan dalam rentang tegangan ini. Jika diberi tegangan kurang dari 7V, pin 5V tidak akan memberikan nilai murni 5V, yang mungkin akan membuat rangkaian bekerja dengan tidak sempurna. Jika diberi tegangan lebih dari 12V, regulator tegangan bisa over heat yang pada akhirnya bisa merusak pcb. Dengan demikian, tegangan yang di rekomendasikan adalah 7V hingga 12V<sup>[14]</sup>.

Beberapa pin power pada *Arduino* Uno :

- **GND**. Ini adalah *ground* atau negatif.
- **Vin**. Ini adalah pin yang digunakan jika anda ingin memberikan power langsung ke board *Arduino* dengan rentang tegangan yang disarankan 7V - 12V
- **Pin 5V**. Ini adalah pin output dimana pada pin tersebut mengalir tegangan 5V yang telah melalui regulator
- **3V3**. Ini adalah pin output dimana pada pin tersebut disediakan tegangan 3.3V yang telah melalui regulator
- **IREF**. Ini adalah pin yang menyediakan referensi tegangan mikrokontroler. Biasanya digunakan pada board shield untuk memperoleh tegangan yang sesuai, apakah 5V atau 3.3V<sup>[14]</sup>.

### 1.5.3 *Input/Output (I/O)*

*Arduino Mega 2560* memiliki jumlah pin terbanyak dari semua papan pengembangan *Arduino*. *Mega 2560* memiliki 54 buah digital pin yang dapat digunakan sebagai input atau output, dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus sebesar 20mA, dan memiliki tahanan pull-up sekitar 20-50k ohm (secara default dalam posisi disconnect). Nilai maximum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan chip mikrokontroler<sup>[14]</sup>.

Beberapa pin memiliki fungsi khusus :

- **Serial**, memiliki 4 serial yang masing-masing terdiri dari 2 pin. Serial 0 : pin 0 (RX) dan pin 1 (TX). Serial 1 : pin 19 (RX) dan pin 18 (TX). Serial 2 : pin 17 (RX) dan pin 16 (TX). Serial 3 : pin 15 (RX) dan pin 14 (TX). RX digunakan untuk menerima dan TX untuk transmit data serial TTL. Pin 0 dan pin 1 adalah pin yang digunakan oleh chip USB-to-TTL ATmega16U2
- **External Interrupts**, yaitu pin 2 (untuk interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Dengan demikian *Arduino Mega 2560* memiliki jumlah interrupt yang cukup melimpah : 6 buah. Gunakan fungsi `attachInterrupt()` untuk mengatur interrupt tersebut.
- **PWM**: Pin 2 hingga 13 dan 44 hingga 46, yang menyediakan output PWM 8-bit dengan menggunakan fungsi `analogWrite()`

- **SPI** : Pin 50 (MISO), 51 (MOSI), 52 (SCK), dan 53 (SS) mendukung komunikasi SPI dengan menggunakan SPI Library
- **LED** : Pin 13. Pada pin 13 terhubung built-in led yang dikendalikan oleh digital pin no 13. Set HIGH untuk menyalakan led, LOW untuk memadamkan nya.
- **TWI** : Pin 20 (SDA) dan pin 21 (SCL) yang mendukung komunikasi TWI dengan menggunakan Wire Library.

*Arduino Mega 2560 R3* memiliki 16 buah input analog. Masing-masing pin analog tersebut memiliki resolusi 10 bits (jadi bisa memiliki 1024 nilai). Secara default, pin-pin tersebut diukur dari *ground* ke 5V, namun bisa juga menggunakan pin AREF dengan menggunakan fungsi `analogReference()`. Beberapa in lainnya pada board ini adalah :

- **AREF**. Sebagai referensi tegangan untuk input analog.
- **Reset**. Hubungkan ke LOW untuk melakukan reset terhadap mikrokontroler. Sama dengan penggunaan tombol reset yang tersedia<sup>[12]</sup>.

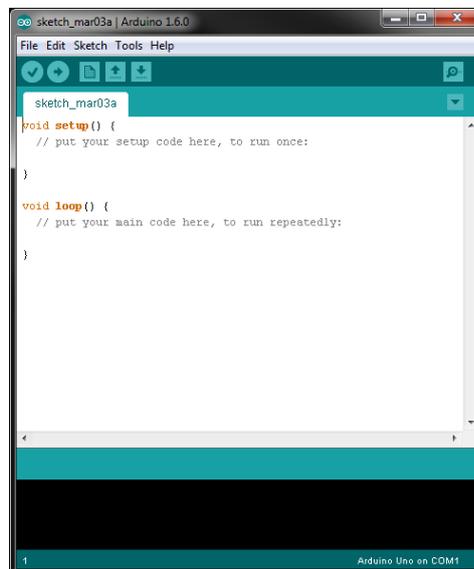
#### 1.5.4 *Software Arduino*

*Arduino* diciptakan untuk pemula bahkan yang tidak memiliki basic bahasa pemrograman sama sekali, karena untuk pemrograman *Arduino* menggunakan bahasa C *Arduino* yang telah dipermudah melalui library (*Arduino.cc*)<sup>[14]</sup>. *Arduino* menggunakan software processing yang digunakan untuk menulis program kedalam *Arduino*. Processing sendiri merupakan penggabungan antara bahasa C++ dan java software. Software *Arduino* ini dapat di-instal diberbagai

*operating system* (OS) seperti: LINUX, Mac OS, dan Windows. Software (Integrated Development Environment) IDE *Arduino* terdiri dari tiga bagian yaitu :

1. Editor program, untuk menulis dan mengedit program dalam bahasa processing. Listing program pada *Arduino* disebut sketch.
2. Compiler, modul yang berfungsi mengubah bahasa processing (kode program) kedalam kode biner karena kode biner adalah satu- satunya bahasa program yang dipahami oleh *Arduino*.
3. Uploader, modul yang berfungsi memasukan kode biner kedalam memori *Arduino*<sup>[14]</sup>.

Berikut adalah tampilan gambar 2.22 tentang software *Arduino* IDE yang penyusun gunakan untuk memprogram *Arduino Mega 2560*.



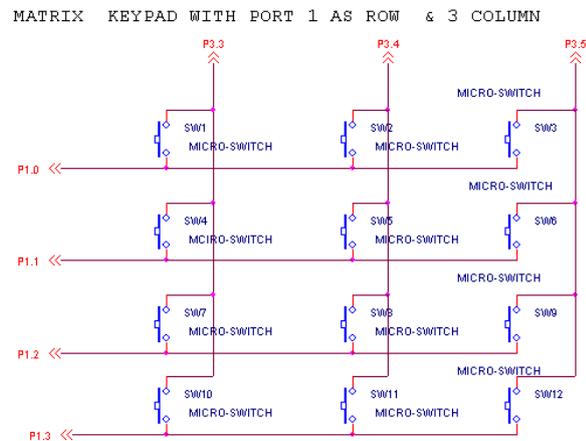
**Gambar 0.22** Tampilan *Arduino software*<sup>[14]</sup>

Struktur perintah *Arduino* secara garis besar terdiri dari dua bagian yaitu void setup dan void loop. Void setup berisi perintah yang akan dieksekusi hanya satu kali sejak *Arduino* dihidupkan sedangkan void loop berisi perintah yang akan dieksekusi berulang-ulang selama *Arduino* dinyalakan

### **1.6 Keypad 3 x 4 membran**

*Keypad* adalah bagian penting dari suatu perangkat elektronika yang membutuhkan interaksi manusia. *Keypad* berfungsi sebagai interface antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah HMI (Human Machine Interface). Matrix *keypad* 3×4 merupakan salah satu contoh *keypad* yang dapat digunakan untuk berkomunikasi antara manusia dengan mikrokontroler. Matrix *keypad* 3×4 memiliki konstruksi atau susunan yang simple dan hemat dalam penggunaan port mikrokontroler. Konfigurasi *keypad* dengan susunan bentuk matrix ini bertujuan untuk penghematan port mikrokontroler karena jumlah key (tombol) yang dibutuhkan banyak pada suatu sistem dengan mikrokontroler<sup>[15]</sup>.

Konstruksi matrix *keypad* 3×4 untuk mikrokontroler dapat dibuat seperti pada gambar 2.23 berikut ini.



**Gambar 0.23** Rangkain *keypad* matriks 3 x 4<sup>[15]</sup>

Konstruksi matrix *keypad* 3x4 diatas cukup sederhana, yaitu terdiri dari 4 baris dan 3 kolom dengan *keypad* berupa saklar push buton yang diletakan disetiap persilangan kolom dan barisnya. Rangkaian matrix *keypad* diatas terdiri dari 12 saklar push buton dengan konfigurasi 4 baris dan 3 kolom. 7 line yang terdiri dari 3 baris dan 4 kolom tersebut dihubungkan dengan port mikrokontroler 8 bit. Sisi baris dari matrix *keypad* ditandai dengan nama P1.0, P1.1, P1.2 dan P1.3 kemudian sisi kolom ditandai dengan nama P3.3, P3.4, P3.5. Sisi input atau output dari matrix *keypad* 4x4 ini tidak mengikat, dapat dikonfigurasi kolom sebagai input dan baris sebagai output atau sebaliknya tergantung programernya.

Proses scanning untuk membaca penekanan tombol pada matrix *keypad* 3x4 untuk mikrokontroler diatas dilakukan secara bertahap kolom demi kolom dari kolom pertama sampai kolom ke 3 dan baris pertama hingga baris ke 4. Program untuk scanning matrix *keypad* 3x4 dapat bermacam-macam, tapi pada intinya sama. Misal kita asumsikan keypad aktif LOW (semua line kolom dan baris dipasang resistor pull-up) dan dihubungkan ke port mikrokontroler dengan jalur

kolom adalah jalur input dan jalur baris adalah jalur output maka proses scanning matrix *keypad* 3×4 diatas dapat dituliskan sebagai berikut :

Pertama, mengirimkan logika Low untuk kolom 1 (P3.3) dan logika HIGH untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (P1.0) akan LOW sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW4 maka data pada baris ke 2 akan LOW sehingga data yang terbaca 1011, atau tombol SW7 yang ditekan sehingga data yang terbaca 1101, atau tombol SW10 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

Kedua, mengirimkan logika Low untuk kolom 2 (P3.4) dan logika HIGH untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (P1.0) akan LOW sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW4 maka data pada baris ke 2 akan LOW sehingga data yang terbaca 1011, atau tombol SW7 yang ditekan sehingga data yang terbaca 1101, atau tombol SW10 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

Ketiga, mengirimkan logika Low untuk kolom 3 (P3.5) dan logika HIGH untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (P1.0) akan LOW sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan LOW sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga

data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

Kemudian data pembacaan baris ini diolah sebagai pembacaan data penekanan tombol *keypad*. Sehingga tiap tombol pada matrix *keypad* 4×4 diatas dengan teknik scanning tersebut akan menghasilkan data penekanan tiap-tiap tombol sebagai berikut :

SW1 = 011 0111	SW9 = 011 1101
SW2 = 101 0111	SW9 = 110 1101
SW3 = 110 0111	SW10 = 011 1110
SW4 = 011 1011	SW11 = 101 1110
SW5 = 101 1011	SW12 = 110 1110
SW6 = 110 1011	
SW7 = 011 1101	
SW8 = 101 1101	

Data port mikrokontroler, misalkan pada SW2 = 101 0111 tersebut terbagi dalam nible atas dan nible bawah dimana data nible atas (1011) merupakan data yang kita kirimkan sedangkan data nible bawah (0111) adalah data hasil pembacaan penekanan tombol *keypad* SW2 pada proses scanning matrix *keypad* 3×4 diatas.

Penggunaan *keypad* ini adalah bertujuan untuk mengatur standard / landasan berat badan dan tinggi badan pengunjung yang masuk. Dengan adanya *keypad* ini, pengguna dapat mengatur sesuai kebutuhannya.

### 1.7 LCD 20 x 4

Display *LCD* 20x4 berfungsi sebagai penampil karakter yang di input melalui *keypad*. *LCD* yang digunakan pada alat ini mempunyai lebar display 4 baris 20 kolom atau biasa disebut sebagai *LCD* Character 20 x 4, dengan 16 pin konektor, yang didefinisikan sebagai berikut<sup>[16]</sup>:

*LCD* (Liquid Crystal Display) atau karakter, adalah *LCD* yang tampilannya terbatas pada tampilan karakter, khususnya karakter ASCII (seperti karakter-karakter yang tercetak pada keyboard komputer).

Sedangkan *LCD* Graphics atau grafik, adalah *LCD* yang tampilannya tidak terbatas, bahkan dapat menampilkan foto. *LCD* grafik inilah yang terus berkembang seperti layar *LCD* yang bisa dilihat di notebook/laptop. Dalam pembahasan kali ini akan dikonsentrasikan pada *LCD* karakter.

Jenis *LCD* karakter yang beredar di pasaran biasa dituliskan dengan bilangan matriks dari jumlah karakter yang dapat dituliskan pada *LCD* tersebut, yaitu jumlah kolom karakter di kali jumlah baris karakter. Sebagai contoh, *LCD* 20 x 4, artinya terdapat 20 kolom dalam 4 baris ruang karakter, yang berarti total karakter yang dapat dituliskan adalah 80 karakter

*LCD* karakter dalam pengendaliannya cenderung lebih mudah dibandingkan dengan *LCD* grafik. Namun ada kesamaan diantara keduanya, yaitu inisialisasi. Inisialisasi adalah prosedur awal yang perlu dilakukan dan dikondisikan kepada *LCD* agar *LCD* dapat bekerja dengan baik. Hal yang sangat penting yang ditentukan dalam proses inisialisasi adalah jenis interface (antarmuka) antara *LCD* dengan controller (pengendali). Pada umumnya terdapat

dua jenis antarmuka yang dapat digunakan dalam pengendalian *LCD* karakter 4-bit dan 8-bit.

Untuk dapat mengendalikan *LCD* karakter dengan baik, tentu perlu koneksi yang benar. Dan koneksi yang benar dapat diwujudkan dengan cara mengetahui pin-pin antarmuka yang dimiliki oleh *LCD* karakter tersebut. *LCD* karakter yang beredar di pasaran memiliki 16 pin antarmuka dan pin-pin antarmuka *LCD*<sup>[16]</sup>.

Dibawah ini adalah gambar 2.24 tentang tampilan pada *LCD* 20x4.



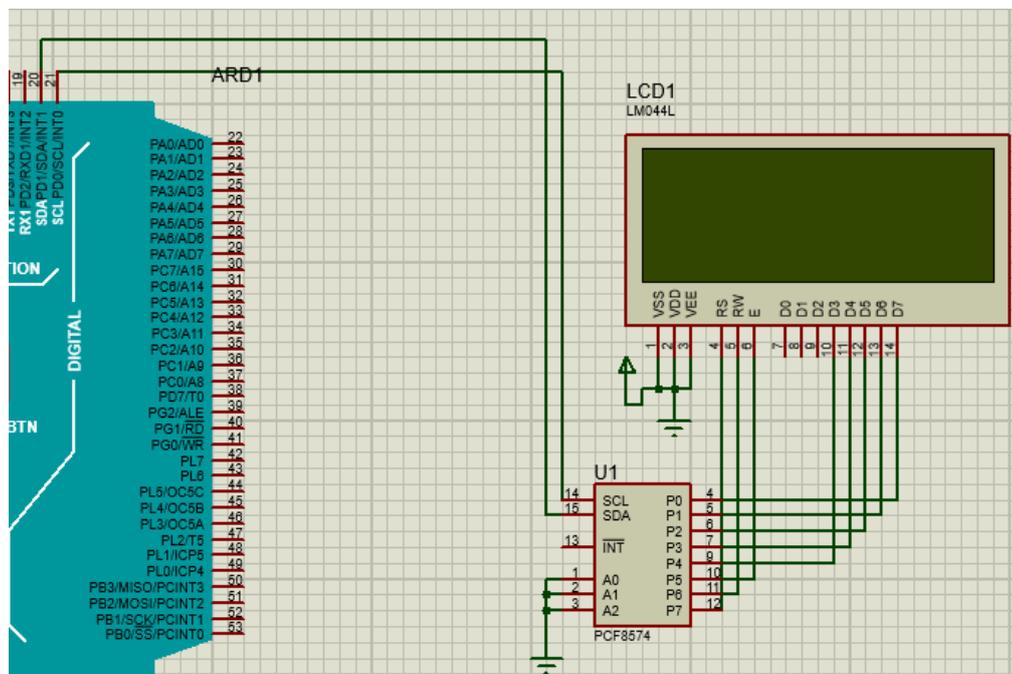
**Gambar 0.24** *LCD* 20 x 4<sup>[16]</sup>

Keterangan:

1. VSS : *Ground*
2. VCC : Input Positif 5 Volt
3. Vo : Tegangan untuk mengatur kontras dari tampilan karakter
4. RS : *Instruction/Register Select*
5. R/W : *Read/Write LCD Register*
6. E : *Enable Clock*
7. DB0 : Data bit 0
8. DB1 : Data bit 1
9. DB2 : Data bit 2
10. DB3 : Data bit 3

11. DB4 : Data bit 4
12. DB5 : Data bit 5
13. DB6 : Data bit 6
14. DB7 : Data bit 7
15. Anoda LED *Backlight LCD*
16. Katoda LED *Backlight LCD*<sup>[16]</sup>

### 1.7.1 Cara Kerja LCD 20 x 4



**Gambar 0.25** Fungsi Pin *LCD*

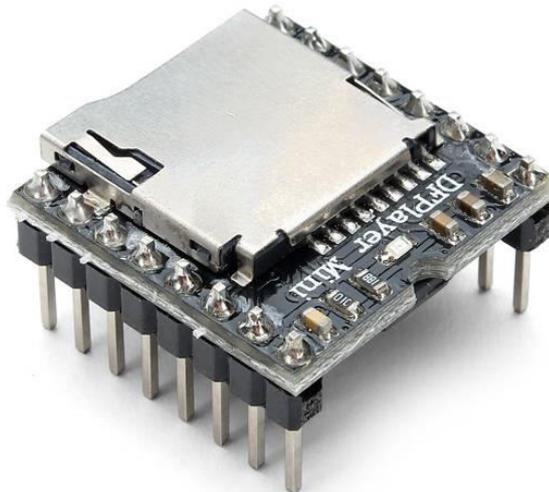
Gambar 2.25 menunjukkan pin yang terdapat pada *LCD*. Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Interface *LCD* merupakan sebuah parallel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke *LCD*. Kode ASCII yang

ditampilkan sepanjang 8-bit dikirim ke *LCD* secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa clock EN setiap nibblenya). Jalur kontrol EN digunakan untuk memberitahu *LCD* bahwa mikrokontroller mengirimkan data ke *LCD*. Untuk mengirim data ke *LCD* program harus menset EN ke kondisi high “1” dan kemudian menset dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.

Saat jalur lainnya sudah siap, EN harus diset ke “0” dan tunggu beberapa saat (tergantung padadatasheet *LCD*), dan set EN kembali ke high “1”. Ketika jalur RS berada dalam kondisi low “0”, data yang dikirimkan ke *LCD* dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi cursor dll). Ketika RS dalam kondisi high atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diset ke “1”. Jalur kontrol R/W harus berada dalam kondisi low (0) saat informasi pada data bus akan dituliskan ke *LCD*. Apabila R/W berada dalam kondisi high “1”, maka program akan melakukan query (pembacaan) data dari *LCD*. Instruksi pembacaan hanya satu, yaitu Get *LCD* status (membaca status *LCD*), lainnya merupakan instruksi uan. Jadi hampir setiap aplikasi yang menggunakan *LCD*, R/W selalu diset ke “0”. Jalur data dapat terdiri 4 jalur (tergantung mode yang dipilih pengguna), DB4, DB5, DB6 dan DB7. Mengirim data secara parallel baik 4-bit merupakan 2 mode operasi primer. Untuk membuat sebuah aplikasi interface *LCD*, menentukan mode operasi merupakan hal yang paling penting.

Mode 4 bit *minimal* hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroler dan *LCD*. Jika bit ini di set ( $RS = 1$ ), maka byte pada posisi kursor *LCD* saat itu dapat dibaca atau ditulis. Jika bit ini di reset ( $RS = 0$ ), merupakan instruksi yang dikirim ke *LCD* atau status eksekusi dari instruksi terakhir yang dibaca.

### 1.8 Modul Mp3 Player

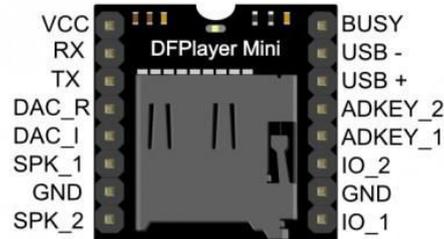


**Gambar 0.26** *DFPlayer mini*<sup>[17]</sup>

Gambar di atas yaitu gambar 2.26 adalah bentuk dari *DFPlayer mini*. *DFPlayer Mini* adalah modul Sound/music Player yang mendukung beberapa file salah satunya adalah file .mp3 yang umum kita gunakan sebagai format sound file. *DFPlayer mini* mempunyai 16 pin interface berupa standar DIP pin header pada kesua sisinya<sup>[17]</sup>.

Berikut nama dan fungsi masing-masing pin pada modul *DFPlayer mini* digambarkan pada gambar 2.27 dan 2.28.

## PinOut



**Gambar 0.27** PinOut DFPlayer mini<sup>[17]</sup>

Number	Name	Description	Note
1	VCC	Input Voltage	DC 3.2-5.0V; Typical: DC4.2
2	RX	UART serial input	
3	TX	UART serial output	
4	DAC_R	Audio output right channel	Drive earphone and amplifier
5	DAC_L	Audio output left channel	Drive earphone and amplifier
6	SPK2	Speaker	Drive speaker less than 3W
7	GND	Ground	Power Ground
8	SPK1	Speaker	Drive speaker less than 3W
9	IO1	Trigger port 1	Short pree to play previous(long press to decrease volume)
10	GND	Ground	Power Ground
11	IO2	Trigger port 2	Short pree to play next(long press to increase volume)
12	ADKEY1	AD port 1	Trigger play first segment
13	ADKEY2	AD port 2	Trigger play fifth segment
14	USB+	USB+ DP	USB Port
15	USB-	USB- DM	USB Port
16	Busy	Playing Status	Low means playing\High means no

**Gambar 0.28** Fungsi Masing – Masing Pin pada DFPlayer mini<sup>[17]</sup>

DFPlayer mini dapat bekerja sendiri secara standalone ataupun bekerja bersama dengan *Arduino* melalui koneksi serial.

DFPlayer memiliki koneksi serial yang dapat menerima instruksi dari controller lain seperti *Arduino*.

Berikut adalah daftar perintah yang dapat digunakan untuk kontrol Modul *DFPlayer mini* melalui koneksi serial dijelaskan pada gambar 2.29 dan 2.30

#### Serial Mode

Support for asynchronous serial communication mode via PC serial sending commands

##### • Instruction Description

Commands	Function Description	Parameters(16 bit)
0x01	Next	
0x02	Previous	
0x03	Special tracking(NUM)	0-2999
0x04	Increase volume	
0x05	Decrease volume	
0x06	Specify volume	0-30
0x07	Specify EQ 0/1/2/3/4/5	Normal/Pop/Rock/Jazz/Classic/Bass
0x08	Specify playback mode(0/1/2/3)	repeat/folder repeat/single repeat/random
0x09	Specify playback source( 0/1/2/3/4)	U/TF/AUX/SLEEP/FLASH
0x0A	Enter into standby-low power loss	
0x0B	Normal working	
0x0C	Reset module	
0x0D	Playback	
0x0E	Pause	
0x0F	Specify folder to playback	1-10(need to set by user)
0x10	Volume adjust set	[DH=1:Open volume adjust][DL:set volume gain 0-31]
0x11	Repeat play	[1:start repeat play][0:stop play]
0x12	Specify MP3 tracks folder	0-9999
0x13	Commercials	0-9999
0x14	Support 15 folder	See detailed description below
0x15	Stop playback, play background	
0x16	Stop playback	

**Gambar 0.29** Kontrol Modul *DFPlayer mini* melalui koneksi serial *Arduino*<sup>[17]</sup>

##### • Serial Query Cmd

Commands	Function Description	Parameters(16bit)
0x3C	STAY	
0x3D	STAY	
0x3E	STAY	
0x3F	Send initialization parameters	0-0x0F( each bit represent one device of the low-four bits)
0x40	Returns an error, request retransmission	
0x41	Reply	
0x42	Query the current status	
0x43	Query the current volume	
0x44	Query the current status EQ	
0x45	Query the current palyback mode	This version retains this feature
0x46	Query the current software version	This version retains this feature
0x47	Query the total number of TF card files	
0x48	Query the total number of U-disk files	
0x49	Query the total number of FLASH card files	
0x4A	keep on	
0x4B	Queries the current track of TF card	
0x4C	Queries the current track of U-disk	
0x4D	Queries the current track of Flash	

**Gambar 0.30** Kontrol Modul *DFPlayer mini* melalui koneksi serial *Arduino*<sup>[17]</sup>

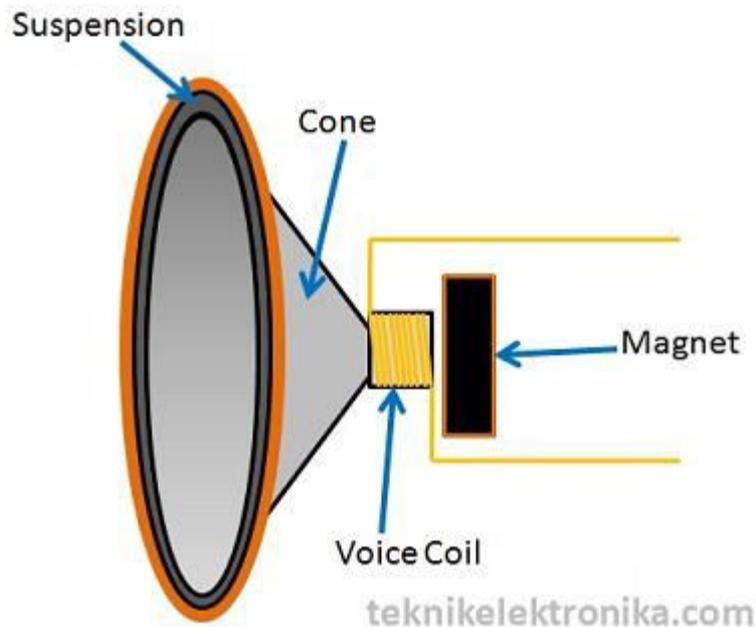
Dengan menggunakan modul *DFPlayer mini* tersebut, maka akan menghasilkan keluaran berupa suara “Silahkan Masuk” yang akan dikirim ke

*speaker*, ketika pengunjung wahana yang sebelumnya diukur berat badan dan tingginya memenuhi standar yang telah ditetapkan pada alat pengukur yang penyusun buat.

### 1.9 Speaker DC

Kita dapat mendengarkan musik radio, mendengarkan suara dari drama televisi ataupun suara dari lawan bicara kita di ponsel, semua ini karena adanya komponen Elektronika yang bernama *Loudspeaker* yang dalam bahasa Indonesia disebut dengan Pengeras Suara. *Loudspeaker* atau lebih sering disingkat dengan *Speaker* adalah Transduser yang dapat mengubah sinyal listrik menjadi Frekuensi Audio (sinyal suara) yang dapat didengar oleh telinga manusia dengan cara mengetarkan komponen membran pada *Speaker* tersebut sehingga terjadilah gelombang suara. Yang dimaksud dengan “Suara” sebenarnya adalah Frekuensi yang dapat didengar oleh Telinga Manusia yaitu Frekuensi yang berkisar di antara  $20\text{Hz} - 20.000\text{Hz}$ . Timbulnya suara dikarenakan adanya fluktuasi tekanan udara yang disebabkan oleh gerakan atau getaran suatu obyek tertentu. Ketika Obyek tersebut bergerak atau bergetar, Obyek tersebut akan mengirimkan Energi Kinetik untuk partikel udara disekitarnya. Hal ini dapat di-analogi-kan seperti terjadinya gelombang pada air. Sedangkan yang dimaksud dengan Frekuensi adalah jumlah getaran yang terjadi dalam kurun waktu satu detik. Frekuensi dipengaruhi oleh kecepatan getaran pada obyek yang menimbulkan suara, semakin cepat getarannya makin tinggi pula frekuensinya<sup>[18]</sup>.

Dibawah ini pada gambar 2.31 adalah gambar bagian bagian dari *speaker*.



**Gambar 0.31** Bagian – bagian dari *Speaker*<sup>[18]</sup>

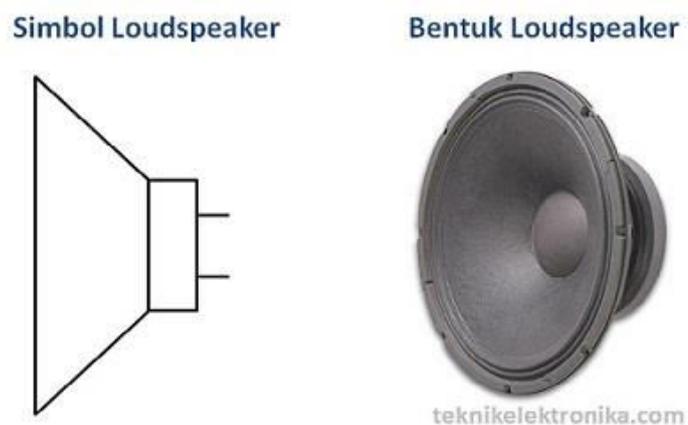
Pada gambar diatas, dapat kita lihat bahwa pada dasarnya *Speaker* terdiri dari beberapa komponen utama yaitu Cone, Suspension, Magnet Permanen, Voice Coil dan juga Kerangka *Speaker*.

Dalam rangka menterjemahkan sinyal listrik menjadi suara yang dapat didengar, *Speaker* memiliki komponen Elektromagnetik yang terdiri dari Kumparan yang disebut dengan Voice Coil untuk membangkitkan medan magnet dan berinteraksi dengan Magnet Permanen sehingga menggerakkan Cone *Speaker* maju dan mundur. Voice Coil adalah bagian yang bergerak sedangkan Magnet Permanen adalah bagian *Speaker* yang tetap pada posisinya. Sinyal listrik yang melewati Voice Coil akan menyebabkan arah medan magnet berubah secara cepat sehingga terjadi gerakan “tarik” dan “tolak” dengan Magnet Permanen. Dengan demikian, terjadilah getaran yang maju dan mundur pada Cone *Speaker*.

Cone adalah komponen utama *Speaker* yang bergerak. Pada prinsipnya, semakin besarnya Cone semakin besar pula permukaan yang dapat menggerakkan udara sehingga suara yang dihasilkan *Speaker* juga akan semakin besar.

Suspension yang terdapat dalam *Speaker* berfungsi untuk menarik Cone ke posisi semula setelah bergerak maju dan mundur. Suspension juga berfungsi sebagai pemegang Cone dan Voice Coil. Kekakuan (rigidity), komposisi dan desain Suspension sangat mempengaruhi kualitas suara *Speaker* itu sendiri.

Berikut ini adalah symbol dan bentuk *loudspeaker* pada gambar 2.32



**Gambar 0.32** Simbol dan bentuk *Loudspeaker*<sup>[18]</sup>

*Speaker* ini berguna untuk menyalurkan outputan dari *DFPlayer mini* yang di beri perintah oleh *Arduino* untuk mengeluarkan suara berupa kata – kata “Selamat Datang” ketika pengunjung wahana lolos masuk sesuai standar yang telah di tentukan setelah selesai mengukur berat badan dan tinggi badannya pada alat yang penyusun buat.