

## PENERAPAN SISTEM OTOMATIS MESIN PEMECAH KEDELAI BERBASIS ARDUINO NANO

**Moh. Ardin Baiquni**

D3 Teknik Mesin, Fakultas Teknik, Universitas Negeri Surabaya

Email: [moh.baiquni@mhs.unesa.ac.id](mailto:moh.baiquni@mhs.unesa.ac.id)

**Djoko Suwito**

Jurusan Teknik Mesin, Fakultas Teknik, Universitas Negeri Surabaya

Email: [djokosuwito@unesa.ac.id](mailto:djokosuwito@unesa.ac.id)

### Abstrak

Di zaman seperti sekarang ini, kehidupan manusia tidak terlepas dari piranti teknologi canggih baik berbentuk elektronik maupun teknologi lain. Salah satunya adalah sensor ultrasonik yang berfungsi memancarkan gelombang ultrasonik. Sesuai dengan fungsinya, maka dibuatlah sistem kontrol otomatis dengan Arduino sebagai inti sistem kontrol dan sensor sebagai input data. Kemudian dilakukan penelitian untuk menguji keakuratan sensor dalam pembacaannya. Penelitian dari hasil mesin pemecah kedelai ini memiliki kapasitas proses pemecahan kedelai 3,2 kg/menit, dengan ukuran rangka 850mm x 670mm x 468mm, daya tamping hopper maksimal 15 kg, ukuran volume hopper 23976333,33 mm<sup>3</sup>, dan menggunakan motor listrik ½ HP dengan kecepatan putaran 1420 rpm. Hasil pembacaan sensor ultrasonik yang belum terkalibrasi pada ketinggian hopper 1 sampai 18 cm sangat tinggi, jika tinggi persentase kesalahan semakin berkurang maka semakin akurat pula sensor bekerja. Berbeda dengan hasil sensor yang sudah terkalibrasi, tingkat keakuratannya hanya pada ketinggian 3 sampai 5 cm dan 26 sampai 37 cm menyebabkan persentase kesalahan dalam pembacaan berganti-ganti.

**Kata Kunci:** Arduino Nano, Penerapan Sistem Otomatis, Mesin Pemecah Kedelai.

### Abstract

In times like today, human life is inseparable from the tools of advanced technology in the form of electronics and other technologies. One of them is an ultrasonic sensor that functions to emit ultrasonic waves. In accordance with its function, an automatic control system with Arduino was made as the core control system and sensor as data input. Then a study was conducted to test the accuracy of the sensors in their readings. The results of this soybean breaking machine have a capacity of breaking soybeans 3.2 kg / minute, with a frame size of 850mm x 670mm x 468mm, a maximum tamping hopper power of 15 kg, hopper volume size 23976333, 33 mm<sup>3</sup>, and using an ½ HP electric motor with a rotation speed of 1420 rpm. The results of the ultrasonic sensor readings that have not been calibrated at the height of the hopper 1 to 18 cm are very high, if the higher the percentage of errors decreases, the more accurate the sensor works. In contrast to the results of the sensor that has been calibrated, the accuracy is only at a height of 3 to 5 cm and 26 to 37 cm causing the percentage of errors in the reading to change.

**Keywords:** Arduino Nano, Application of Automatic Systems, Soybean Breaking Machine.

### PENDAHULUAN

Di zaman seperti sekarang ini, kehidupan manusia tidak terlepas dari piranti teknologi canggih baik berbentuk elektronik maupun teknologi lain. Salah satu bentuk teknologi yang cukup memberikan suatu kemudahan bagi pemakainya adalah teknologi dengan fitur sensor ultrasonik. Prinsip kerjanya adalah transmitter mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek.

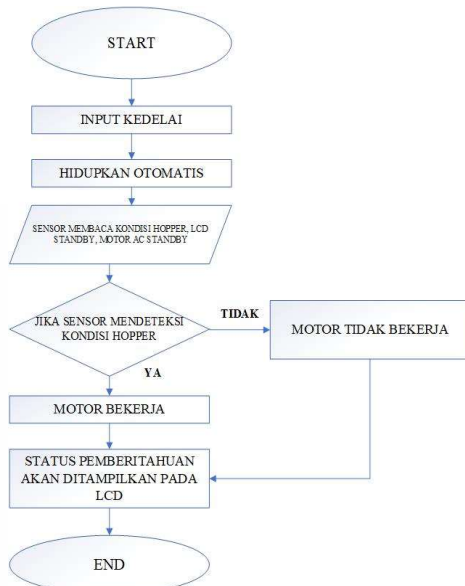
Proses pemecahan kedelai terdiri dari beberapa tahapan yaitu proses perebusan kedelai setengah matang, pemecahan kedelai, pemisahan kulit ari kedelai, perendaman, perebusan kembali hingga matang, peragian, dan pengemasan. Dalam hal ini, penulis mencari ide bagaimana agar mesin pembuat bahan pangan seperti kedelai untuk tempe bisa bekerja secara

otomatis tanpa harus manual. Misalnya, *hopper* dalam mesin jika kedelainya terisi maka semuanya akan bekerja secara otomatis. Dan sebaliknya jika kosong, maka berhenti secara otomatis juga. Dan untuk mengetahui *level hopper* pada mesin maka perlu menampilkannya pada sebuah LCD (*liquid crystal display*).

Melihat kondisi permasalahan yang ada serta dampak dari masalah yang ditimbulkannya, maka penulis merancang suatu alat yang bisa memecah kedelai secara otomatis berdasarkan daya tamping hopper sebesar 12kg sebagai tolak ukur sensor, serta menampilkannya pada sebuah LCD. Dengan begitu, bisa menghemat waktu, tenaga, dan biaya dengan memanfaatkan efisiensi ilmu dalam bidang teknologi guna membantu produsen dalam proses produksi pembuatan tempe untuk mencapai pemenuhan konsumen atas permintaan tempe dipasaran.

**METODE**

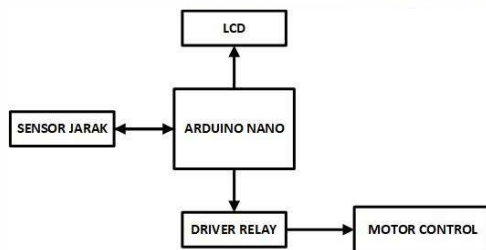
**Rencana Perancangan**



Gambar 1. Diagram Alur Sistem Kerja

Penyelesaian sistem otomatis untuk mesin pemecah kedelai dimulai dengan input kedelai kemudian hidupkan tombol mode otomatis. Untuk pembuatan rangkaian terpenting seperti sensor ultrasonik, LCD, dan motor mesin yang sudah ada. Jika sudah siap rangkaian tersebut, tahapan selanjutnya dilakukan pengujian berupa cara kerja dari sensor yang sudah terpasang pada hopper mesin. Hasil dari cara kerja sensor tersebut adalah motor bekerja saat penggilingan kedelai (YA) dan motor tidak bekerja saat penggilingan kedelai (TIDAK). Dari kedua hasil tersebut akan diinformasikan melalui LCD yang terdapat dalam rangkaian tersebut. Tahap selanjutnya adalah hasil dari proses penggunaan sensor pada hopper, yang selanjutnya akan dievaluasi.

**Diagram Blok**

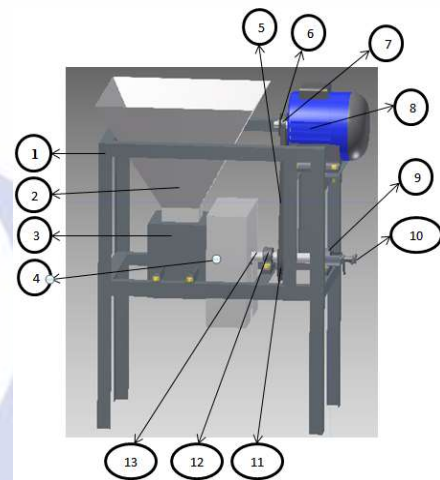


Gambar 2. Diagram Blok Perancangan

mekanisme alat tersebut sesuai blok diagram di atas adalah ketika sensor jarak (pada pin *Trigger Input*) akan membaca kondisi *hopper* dan kondisi tersebut akan dikirim ke Arduino nano. Selanjutnya Arduino nano akan mengolah informasi dari sensor tersebut dan akan ditampilkan melalui LCD. Jika hasil pengolahan dari

sensor (pin *Echo Output*) tersebut bernilai logika 1 (*hopper* penuh) maka arduino akan memberikan *trigger* ke *driver relay* untuk menjalankan *motor control* dan motor mesin pemecah kedelai akan menyala. Sebaliknya jika hasil bernilai 0 (*hopper* kosong) maka Arduino akan memberikan *trigger* ke *driver relay* untuk menghentikan *motor control*.

**Desain Mesin**

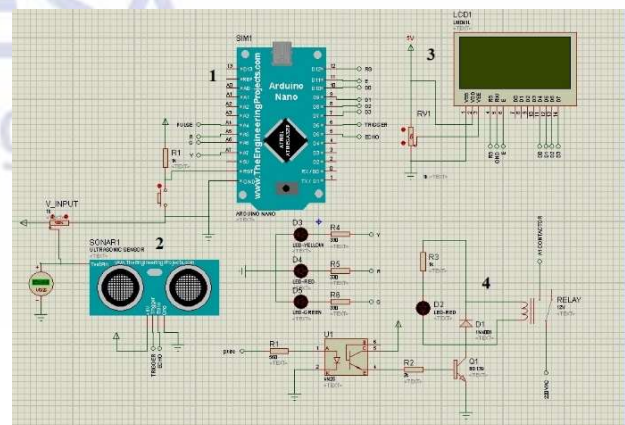


Gambar 3. Mesin Pemecah Kedelai Semiotomatis

**Keterangan:**

- 1. Rangka
- 2. Hopper
- 3. Rumah penggilas
- 4. Penutup tempat penggilas
- 5. V-belt
- 6. Pasak
- 7. Pulley penggerak
- 8. Motor listrik
- 9. Klem pengatur shaft
- 10. Pengatur shaft
- 11. Pulley yang digerakkan
- 12. Bantalan, bearing
- 13. Shaft.

**Rancangan Sistem Kontrol Otomatis**



Gambar 4. Rangkaian Sistem Kontrol Otomatis

**Keterangan:**

- 1. Arduino Nano
- 2. Sensor Ultrasonik
- 3. LCD
- 4. Driver Relay Circuit

**Sistem Pengaman**

Untuk sistem pengaman pada mesin pemecah kedelai otomatis terbagi menjadi 2 macam yaitu *emergency button* dan *overload*. *Emergency button* digunakan jika hopper mesin tersebut terdapat berbagai macam benda asing yang secara sengaja maupun tidak sengaja masuk. Sedangkan, *overload* digunakan jika *hopper* mesin kelebihan muatan maka akan mengaktifkan alarm yang dapat menghentikan motor dengan sendirinya.

**Langkah-langkah Proses Pemecah Kedelai**

Langkah-langkah proses pemecah kedelai secara otomatis antara lain:

- Persiapkan mesin pemecah kedelai (proses pengecekan awal sebelum operasi)
- Sebelum menggunakan mesin, masukkan kedelai yang sudah direbus kedalam hopper mesin (input).
- Hidupkan tombol mode otomatis pada mesin pemecah kedelai.
- Motor penggerak (motor listrik) dan sensor akan melakukan kinerja pada preoses pemecahan kedelai berlangsung. Sedangkan LCD akan menampilkan level kedelai dalam hopper mesin tersebut
- Proses pemecahan kedelai sama seperti mode semiotomatis sebelumnya.
- Jika level kedelai dalam hopper mesin sudah kosong maka secara otomatis pula mesin tersebut akan berhenti

**HASIL DAN PEMBAHASAN**

**Data Hasil Pengujian Tegangan Arduino**

Pengujian ini dilakukan dengan mengukur pin input yang tersambung dengan sumber tegangan volt.

Tabel 1. Tegangan Arduino Nano

No.	Pengukuran	Vinput	Vinput yang terbaca	Error (%)
1	Tanpa Beban	5	5	0
2	Dengan Beban	5	4.7	6

Dari Tabel 1. Menunjukkan bahwa tegangan Arduino yang terbaca dalam Avometer sebelum mendapatkan beban masih sama, sedangkan jika mendapatkan beban akan berubah nilai.

**Data Hasil Pengujian Sensor Ultrasonik HY-SRF05**

Pengujian ini dilakukan sebanyak 37 kali dengan kondisi sesuai warna LED. Maka hasil pengujian ini dapat dikelompokkan sebagai berikut

Tabel 2. Data Analisis Sensor di kondisi LED Hijau

No.	Hasil Pengukuran		Selisih	Error (%)	LED
	Sensor	Hopper			
1	2	1	1	100	hijau
2	3	2	1	50	
3	4	3	1	33.33333	
4	5	4	1	25	
5	6	5	1	20	
6	7	6	1	16.66667	
7	8	7	1	14.28571	
8	9	8	1	12.5	
9	10	9	1	11.11111	
10	11	10	1	10	
11	12	11	1	9.090909	
12	13	12	1	8.333333	

Dari Tabel 2. Menunjukkan bahwa pembacaan sensor pada tinggi *hopper* 1 sampai 12 cm hanya bertambah 1 cm setiap pengukurannya. Tetapi tinggi % *error* akan semakin berkurang.

Tabel 3. Data Analisis Sensor di Kondisi LED Kuning

No.	Hasil Pengukuran		Selisih	Error (%)	LED
	Sensor	Hopper			
13	14	13	1	7.692308	kuning
14	15	14	1	7.142857	
15	16	15	1	6.666667	
16	17	16	1	6.25	
17	18	17	1	5.882353	
18	19	18	1	5.555556	
19	19	19	0	0	
20	20	20	0	0	
21	21	21	0	0	
22	22	22	0	0	
23	23	23	0	0	
24	24	24	0	0	

Dari Tabel 3. Menunjukkan bahwa pembacaan sensor pada tinggi *hopper* 13 sampai 24 cm tetap sama, hanya saja % *error* yang tadinya kecil semakin berkurang dan akhirnya *null*.

Tabel 4. Data Analisis Sensor di Kondisi LED Merah

No.	Hasil Pengukuran		Selisih	Error (%)	LED
	Sensor	Hopper			
25	25	25	0	0	merah
26	26	26	0	0	
27	27	27	0	0	
28	28	28	0	0	
29	29	29	0	0	
30	30	30	0	0	
31	31	31	0	0	
32	32	32	0	0	
33	33	33	0	0	
34	34	34	0	0	
35	35	35	0	0	
36	36	36	0	0	
37	37	37	0	0	

Dari Tabel 4. Menunjukkan bahwa pembacaan sensor pada tinggi *hopper* 25 sampai 37 cm tetap sama, hanya saja % *error* sudah tidak ada atau akurat.

**Data Hasil Pengujian Sistem Kerja Alat**

Pengujian ini dilakukan untuk membandingkan bagaimana kesalahan dalam pembacaan sensor sesudah terkalibrasi



Tabel 5. Data Analisis Sensor Terkalibrasi di LED Hijau

No.	Pembacaan Sensor Sudah Terkalibrasi	Tinggi Hopper	LED	Error (%)
1	0.774570089	1	Hijau	-22.543
2	1.816779573	2		-9.16102
3	2.858989057	3		-4.70036
4	3.901198541	4		-2.47004
5	4.943408025	5		-1.13184
6	5.985617509	6		-0.23971
7	7.027826993	7		0.397528
8	8.070036477	8		0.875456
9	9.112245961	9		1.247177
10	10.15445545	10		1.544554
11	11.19666493	11		1.787863
12	12.23887441	12		1.99062

Dari Tabel 5. Menunjukkan bahwa sensor yang sudah terkalibrasi mendapatkan hasil % error yang bervariasi pada pembacaan tinggi hopper 1 sampai 12 cm. di lihat dari tingkat akuratnya, pada tinggi hopper 4 sampai 8 cm-lah yang sedikit akurat sebelum berubah.

Tabel 6. Data Analisis Sensor Terkalibrasi di LED Kuning

No.	Pembacaan Sensor Sudah Terkalibrasi	Tinggi Hopper	LED	Error (%)
13	13.2810839	13	Kuning	2.162184
14	14.32329338	14		2.309238
15	15.36550287	15		2.436686
16	16.40771235	16		2.548202
17	17.44992183	17		2.646599
18	18.49213132	18		2.734063
19	18.49213132	19		-2.67299
20	19.5343408	20		-2.3283
21	20.57655029	21		-2.01643
22	21.61875977	22		-1.73291
23	22.66096925	23		-1.47405
24	23.70317874	24		-1.23676

Dari Tabel 6. Menunjukkan bahwa pengukuran dan hasil % error tetap sama pada tinggi hopper 13 sampai 24 cm.

Tabel 7. Data Analisis Sensor Terkalibrasi di LED Merah

No.	Pembacaan Sensor Sudah Terkalibrasi	Tinggi Hopper	LED	Error (%)
25	24.74538822	25	Merah	-1.01845
26	25.78759771	26		-0.81693
27	26.82980719	27		-0.63034
28	27.87201668	28		-0.45708
29	28.91422616	29		-0.29577
30	29.95643564	30		-0.14521
31	30.99864513	31		-0.00437
32	32.04085461	32		0.127671
33	33.0830641	33		0.251709
34	34.12527358	34		0.368452
35	35.16748306	35		0.478523
36	36.20969255	36		0.582479
37	37.25190203	37		0.680816

Dari Tabel 7. Menunjukkan bahwa sensor pada tinggi hopper 26 sampai 37 cm akan menghasilkan pembacaan yang akurat meskipun nilai % error-nya berbeda.

**Analisis Pengujian tegangan Arduino**

Arduino nano memiliki tegangan sekitar 5 volt, untuk tegangan masukkan sekitar 7-12 volt, dan batas tegangan masukkan yang dimilikinya sekitar 6-20 volt. Seperti saat masuk dalam Arduino dengan kondisi tanpa beban, tegangan tetap stabil sekitar 5 volt. Sedangkan pada saat pin I/O telah terhubung dengan beban tegangan menjadi 4.7 volt.

Sedangkan hasil pengukuran dari semua pengujian yang dilakukan tidak ada yang sama dengan pengukuran aslinya, maka didapatkan perhitungan nilai error sebagai berikut:

$$Error = \frac{\text{Hasil Pengukuran} - \text{Pengukuran asli}}{\text{Pengukuran asli}} \times 100\% \quad (1)$$

**Analisis Pengujian Sensor Ultrasonik**

Ultrasonik digunakan sebagai sensor pendeteksi ukuran hopper serta banyaknya kedelai yang akan dikeluarkan. Sensor ini diberi tegangan 5volt dari output Arduino dan sudah berjalan dengan baik. Pengujian terhadap sensor ini dilakukan sebanyak 37 kali, terdapat pengujian yang tidak sesuai pada saat pengukuran yaitu pengukuran hopper dan pembacaan sensor. Untuk pengukuran hopper yang terbaca 1 cm tetapi dalam pembacaan sensor 2 cm.



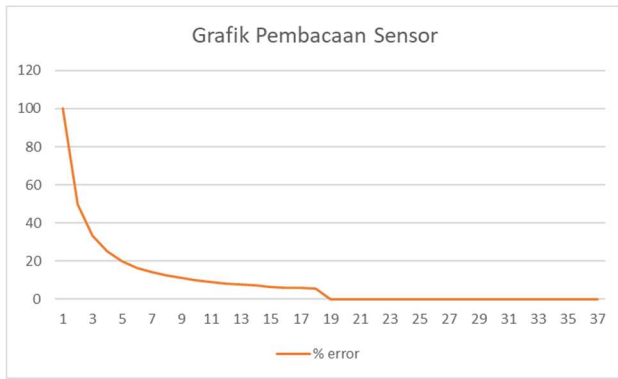
Gambar 5. LCD Memuat nilai Baca Sensor

**Analisis Pengujian Sistem Kerja Alat**



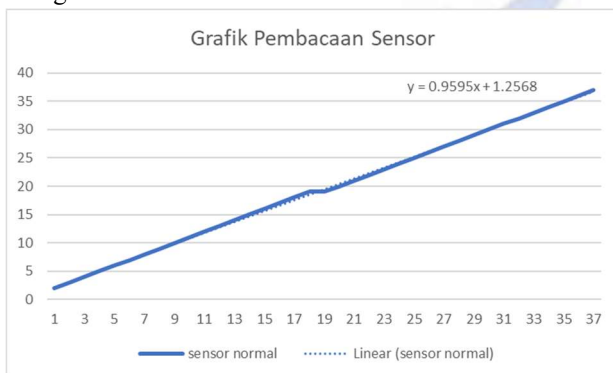
Gambar 6. Sensor Membaca Kondisi Hopper

Saat pembacaan sensor yang belum terkalibrasi, tingkat kesalahannya sangat tinggi mulai dari ketinggian hopper 1 cm sampai 18 cm. jika tinggi kesalahan dalam pembacaan sensor berkurang, maka semakin akurat pula sensor akan bekerja



Gambar 7. Grafik Nilai % Error

Untuk sumbu X sebagai pengukuran hopper dan sumbu Y sebagai pembacaan sensor maka grafiknya adalah sebagai berikut:

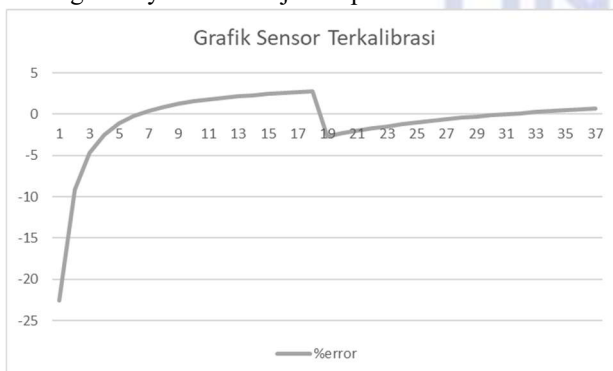


Gambar 8. Grafik Pembacaan Sensor

Diperoleh persamaan  $y = 0,9595x + 1,2568$  (2) persamaan ini yang akan ditanamkan pada program Arduino dengan mengambil nilai x sebagai pembacaan sensor yang telah terkalibrasi. Persamaan tersebut antara lain:

$$x = \frac{(y-1,2568)}{0,9595} \quad (3)$$

maka grafiknya akan menjadi seperti ini



Dari hasil grafik ke-2, pembacaan sensor bekerja optimal di ketinggian hopper 4 sampai 8 cm dan 26 sampai 37 cm. untuk pembacaan sensor di ketinggian hopper yang lain selalu berganti-ganti ukuran. Jadi, dua pengujian ini

dilakukan untuk mengetahui seberapa besar % error tersebut akan berkurang.

### Pembahasan Progam

Rangkaian-rangkaian diatas tidak dapat berjalan sistematis tanpa bantuan program pengendali. Dengan memanfaatkan ATmega 328p sebagai komponen pengendali seluruh rangkaian, serta software dan Bahasa Arduino menjadikan komponen-komponen tersebut dapat saling berjalan berkesinambungan

```
#include <LiquidCrystal.h> //Load Liquid Crystal Library
//LiquidCrystal LCD (12, 11, 5, 4, 3, 2); //Create Liquid Crystal Object called LCD
LiquidCrystal LCD (7, 8, 9, 10, 11, 12);
```

```
int trigPin=4; //Sensor Trip pin connected to Arduino pin 9
int echoPin=5; //Sensor Echo pin connected to Arduino pin 7
int myCounter=0; //declare your variable myCounter and set to 0
int servoControlPin=2; //Servo control line is connected to pin 6
long pingTime, targetDistance; //time for ping to travel from the sensor to the target and return
//Distance to Target in Centimeters
//float speedOfSound=343; //Speed of sound in miles per hour
```

```
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(servoControlPin, OUTPUT);
  LCD.begin(16,4); //Tell Arduino to start your 16x2 LCD
  LCD.setCursor(0,0); //Set LCD cursor to upper left corner, column 0, row 0
  LCD.print("Distance:"); //Print Message on First Row
}
```

```
void loop() {
  digitalWrite (trigPin, LOW); //Set trigger pin low
  delayMicroseconds (2000); //Let signal settle
  digitalWrite (trigPin, HIGH); //Set trigPin high
  delayMicroseconds (15); //Delay in high state
  digitalWrite (trigPin, LOW); //ping has now been sent
  delayMicroseconds (10); //Delay in high state
```

```
pingTime = pulseIn (echoPin, HIGH); //pingTime in
microseconds
targetDistance = (pingTime/2)/29.1;
```

```
LCD.setCursor(0,1); //Set the cursor to the first column
of the second row
LCD.print(" "); //Print blanks to clear the row
LCD.setCursor(0,1); //Set Cursor again to the first
column of the second row
LCD.print(targetDistance); //Print measured distance
LCD.print("cm"); //Print your units delay (500); //Pause
to let things settle
```

```
if (targetDistance >= 10 & targetDistance <= 10)
{
LCD.setCursor(0,2);
LCD.print("hopper penuh");
//digitalWrite(Buzzer,LOW);
digitalWrite(servoControlPin,LOW);
}
else {
LCD.setCursor(0,2);
LCD.print("hopper kosong");
//tone(servoControlPin,400); // play tone of 400Hz
for 500 ms
digitalWrite(servoControlPin,HIGH);
}
}
```

## PENUTUP

### Simpulan

Setelah dilakukan analisis pengujian dapat disimpulkan:

- Perancangan sistem kontrol otomatis terdiri dari 2 tahap yaitu perancangan *hardware* dan perancangan *software*. Perancangan *hardware* tersusun dari beberapa bagian, antara lain: 1) Sensor Jarak, 2) Arduino Nano, 3) *Liquid Crystal Display* (LCD), 4) *Driver Relay Circuit*, 5) *Motor Control*. Untuk perancangan *software* menggunakan Arduino IDE sebagai cara kerja dari perancangan *hardware*.
- Cara kerjanya ketika sensor jarak (pada pin *Trigger Input*) akan membaca kondisi *hopper* dan kondisi tersebut akan dikirim ke Arduino nano. Selanjutnya Arduino nano akan mengolah informasi dari sensor tersebut dan akan ditampilkan melalui LCD. Jika hasil pengolahan dari sensor (pin *Echo Output*) tersebut bernilai logika 1 (hopper penuh) maka arduino akan memberikan *trigger* ke *driver relay* untuk menjalankan *motor control* dan motor mesin pemecah kedelai akan menyala. Sebaliknya jika hasil berlogika 0 (hopper kosong) maka Arduino akan memberikan

*trigger* ke *driver relay* untuk menghentikan *motor control*.

- Hasil analisa sensor yang belum mendapatkan kalibrasi menunjukkan tingkat % error atau kesalahannya sangat tinggi mulai dari ketinggian hopper 1 cm sampai 18 cm. jika tinggi kesalahan dalam pembacaan sensor berkurang, maka semakin akurat sensor bekerja. Berbeda dengan hasil analisa sensor yang sudah mendapat kalibrasi, pembacaan sensor tersebut akurat hanya di ketinggian hopper 3 sampai 5 cm dan 26 sampai 37 cm. karena mengikuti ketinggian hopper, hasil % error-nya sering berganti-ganti.

### Saran

Saran dari penulis yang dapat diberikan dari hasil laporan Tugas Akhir ini adalah sebagai berikut:

- Desain keseluruhan dan mekanik dibuat lebih sistematis agar lebih fleksibel lagi jika digunakan.
- Perlu adanya pengembangan lagi tentang kalibrasi ulang sensor.

### DAFTAR PUSTAKA

- Archtz, (2015). Mengenal Arduino Nano. <http://archtz.wordpress.com/2015/05/12/mengenal-arduino-nano>. Diakses pada tanggal 20 November 2018.
- Badan Standarisasi Nasional, (2012). Tempe Persembahan Indonesia Untuk Dunia. Badan Standarisasi Nasional, Jakarta
- Bureau of Energy Efficiency (BEE), Ministry of Power, (2004). *Energy Efficiency in Electrical Utilities*. (Book 3). India: Ministry of Power.
- Chanshue, (2010). Cara Membaca Gelang Warna resistor. <http://chanshue.wordpress.com/2010/04/26/cara-membaca-gelang-warna-resistor/>. Diakses pada tanggal 5 April 2019.
- Djuandi, Feri, (2011). Pengenalan Arduino. Elexmedia, Jakarta.
- Dwi Septian Suyadhi, Taufiq., (2010). Buku Pintar Robotika. ANDI, Yogyakarta.
- Dermanto, Trikueni., (2013). Sistem Desain. <http://trikueni-desain-sistem.blogspot.com/2013/09/menghitung-arusmotor-Ac.html/>. Diunduh pada tanggal 25 Maret 2019.
- Ihsan, (2016). Berkenalan Dengan Arduino Nano. <http://ecadio.com/mengenal-dan-belajar-arduino-nano>. Diakses pada tanggal 20 Januari 2019
- Koswara, S, (1992). Teknologi Pengolahan Kedelai Menjadikan Makanan Bermutu. Pustaka Sinar Harapan, Jakarta.

- Parekh, R., (2003). *AC Induction Motors Fundamentals*, AN887. Microchip Technology Inc. [ww1.microchip.com/downloads/en/AppNotes/00887a.pdf](http://ww1.microchip.com/downloads/en/AppNotes/00887a.pdf). Diunduh pada tanggal 12 Maret 2019.
- Santoso, H.B., (1993). Pembuatan tempe dan tahu kedelai. Kanisius, Yogyakarta.
- Sarwono, B., (2002). Membuat Tempe dan Oncom. Penebar Swadaya, Jakarta.
- Sumanto, M.A., (1993). Motor Listrik Arus Bolak-Balik. Andi Offset, Yogyakarta
- Yohannes, Christoforus., (2011). Sistem Penghitung Jumlah Barang Otomatis Dengan Sensor Ultrasonik. <http://digilib.unhas.ac.id/>. Universitas Hasanuddin. Diunduh pada tanggal 25 Maret 2019.

