

## **BAB IV**

### **HASIL DAN UJI COBA**

#### **IV.1. Tampilan Hasil**

Dalam bab ini akan dijelaskan dan ditampilkan bagaimana hasil dari pengujian rancangan alat yang dibuat beserta pembahasan tentang pergerakan, cara kerja pengukuran kaki dan perhitungan penentuan ukuran sepatu. Adapun hasil dari pengujian yang dilakukan adalah perangkat elektronik yang dibuat atau dirancang dan diprogram dengan menggunakan aplikasi Arduino IDE.

#### **IV.2. Pelaksanaan Pengujian Rangkaian**

Sebelum melakukan pengujian, beberapa hal yang harus diperhatikan dan dipersiapkan adalah sebagai berikut :

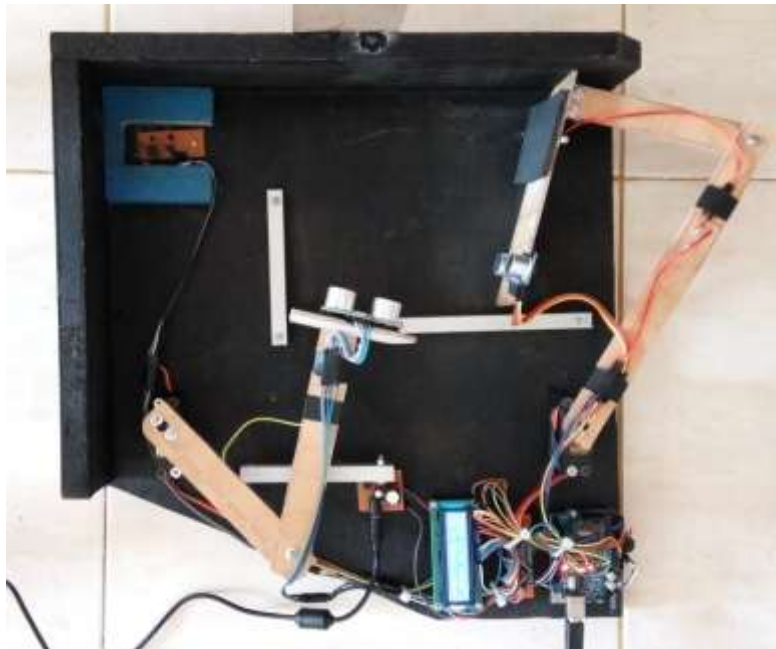
1. Perangkat pengukuran kaki dan berbasis mikrokontroler dalam keadaan siap diuji, tidak ada *trouble* pada saat pengujian.
2. Sebelum pengujian perangkat, hubungkan *power supply* (adaptor) 12VDC 1A pada sumber listrik.
3. Hasil pengujian dianalisa dan dibandingkan dengan perangkat pembanding, seperti menghitung jarak sensor dengan penggaris, menghitung tegangan menggunakan multimeter dan lain sebagainya.

4. Hasil pengujian dipaparkan dalam bentuk tabel dan grafik, dianalisa dan dijelaskan secara terperinci.
5. Pengukuran kaki menggunakan beberapa sampel random dengan ukuran kaki 20 cm – 30 cm.

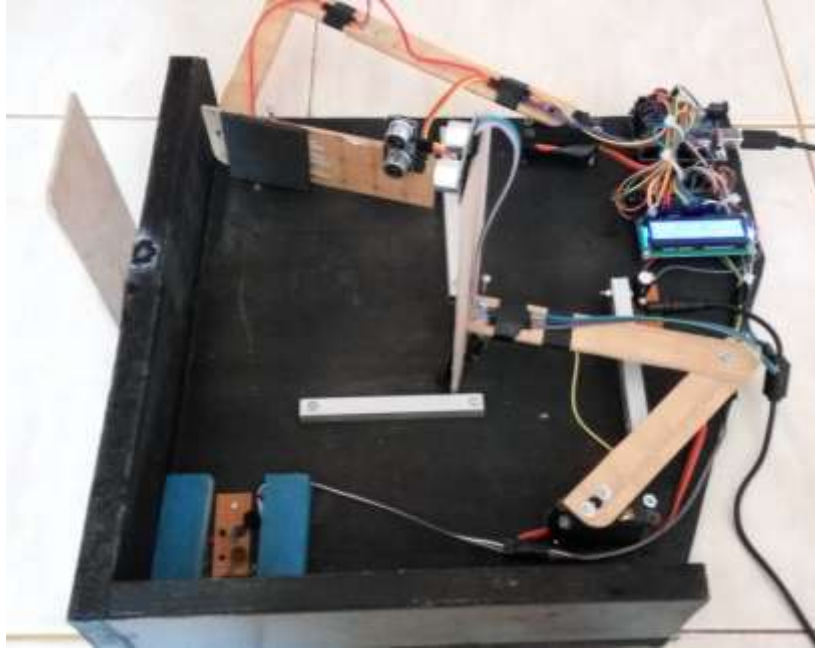
### **IV.3. Tampilan Hasil Perangkat**

Berikut adalah tampilan hasil perancangan pengukur kaki dan penentu ukuran sepatu, ditunjukkan oleh gambar di bawah ini :

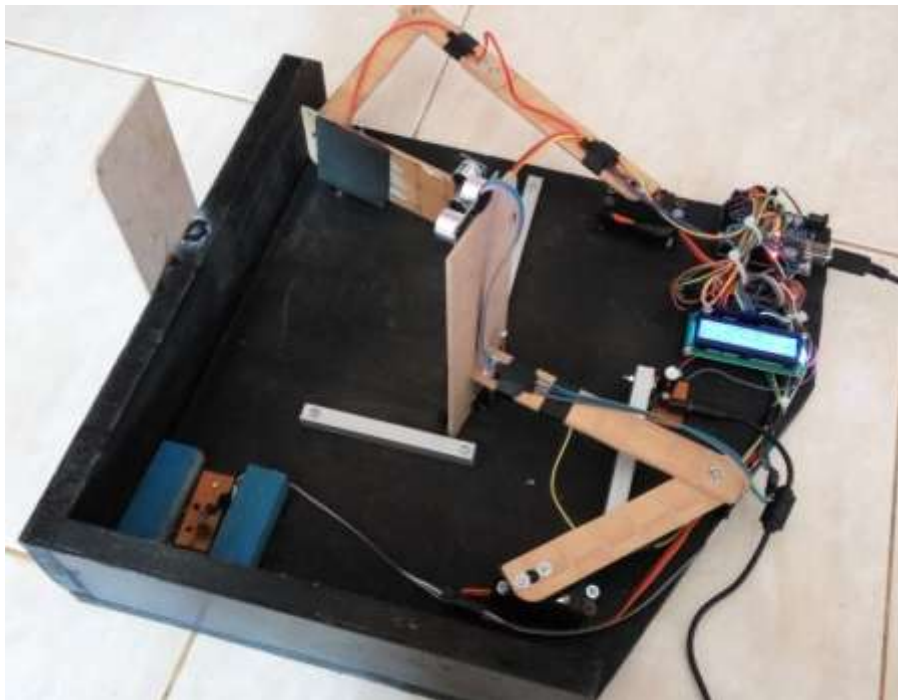
#### **A. Tampak Atas**



#### **B. Tampak Depan**



### C. Hasil Keseluruhan




**Gambar IV.1. Perangkat Keseluruhan**

### IV.4. Pengujian *Software*

Untuk mengetahui apakah rangkaian pada perangkat telah bekerja dengan baik, maka dilakukan pengujian dengan memberikan program perintah pada mikrokontroler dengan melakukan penginputan data dari komputer ke dalam mikrokontroler. Sebelum dilakukannya proses *download* program, hubungkan terlebih dahulu antara komputer melalui kabel USB dengan rangkaian mikrokontroler.

Dalam proses instalasi ini menggunakan aplikasi *Arduino 1.6.9*. Untuk melakukan instalasi ini dapat dilakukan dengan beberapa langkah antara lain :

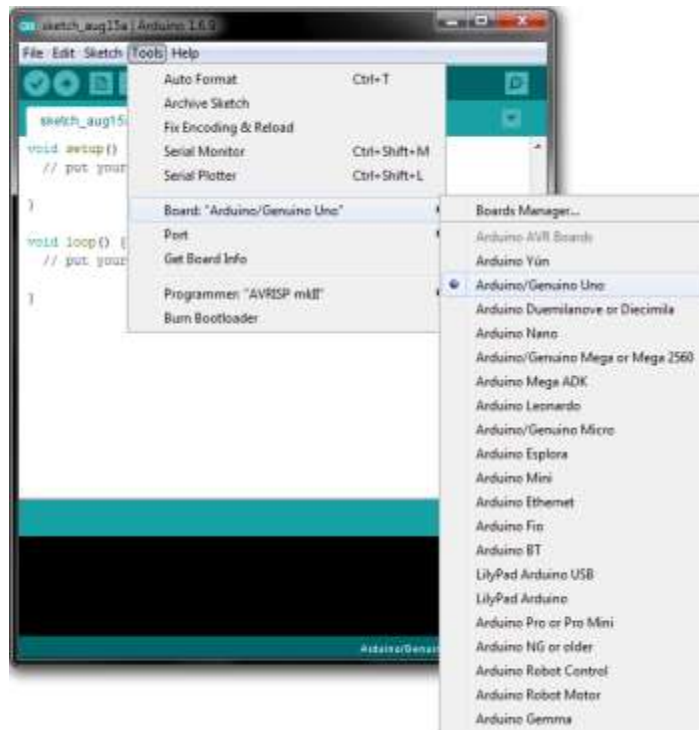
- a. Langkah pertama yang dilakukan adalah dengan mengklik *icon*  *Arduino*. Setelah program melakukan *load* maka akan terlihat bentuk tampilan seperti gambar IV.2 di bawah ini.



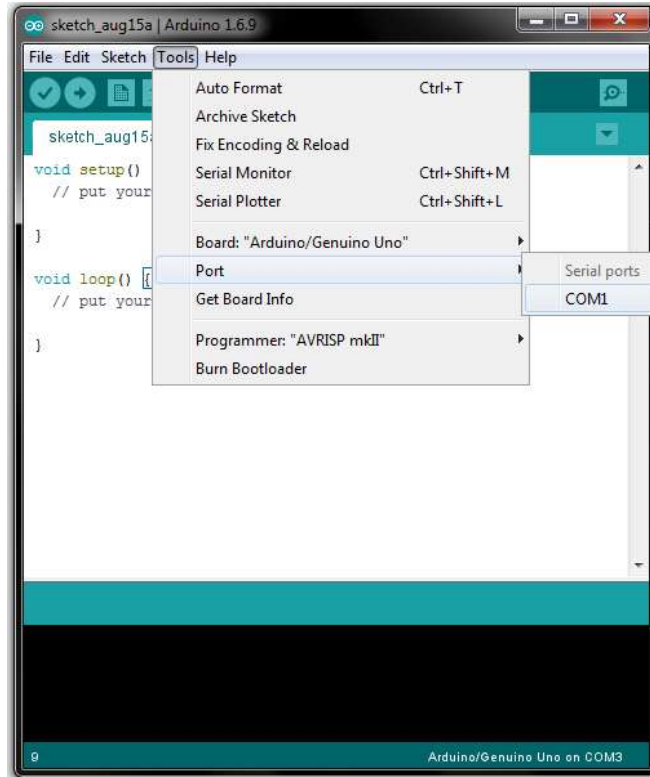
**Gambar IV.2. Tampilan Arduino 1.6.9**

- b. Selanjutnya yang dilakukan sebelum menginstal program terhadap mikrokontroler adalah melakukan pengaturan (*setting*) pada perangkat yang diperlukan dan menyetikkan

program sesuai dengan yang dibutuhkan. Pengaturan pertama adalah pemilihan *board* arduino yang digunakan pada *software* sesuai dengan perangkat yaitu Arduino UNO, seperti pada gambar IV.3. Pengaturan kedua adalah pemilihan *port USB* yang digunakan perangkat, seperti pada gambar IV.4. di bawah ini :



**Gambar IV.3. Pengaturan dan Pemilihan *Board* Arduino**



**Gambar IV.4. Pengaturan Port USB pada Software Arduino 1.6.9**

- c. Setelah pengaturan selesai, proses berikutnya adalah penulisan *listing* program. Berikut adalah *listing* program dari perancangan pengukur kaki dan penentu ukuran sepatu:

```

//*****

// include the library code:

#include <LiquidCrystal.h>

#include <Servo.h>

#define ECHOPIN A3      // Pin to receive echo pulse

#define TRIGPIN A2     // Pin to send trigger pulse

#define ECHOPIN2 10    // Pin to receive echo pulse

#define TRIGPIN2 11    // Pin to send trigger pulse

```

```
int distance;

int distance2;

int panggil = 0;

int jangandiulang = 0;

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

int analogkaki = 0;

int pospanjang = 0;

int poslebar = 0;

int limitpanjang=0;

int limitlebar=0;

int simpanpanjang = 0;

int simpanlebar = 0;

Servo servopanjang;

Servo servolebar;

void setup() {

  pinMode(ECHOPIN, INPUT);  pinMode(TRIGPIN, OUTPUT);

  pinMode(ECHOPIN2, INPUT); pinMode(TRIGPIN2, OUTPUT);

  pinMode(A5, INPUT);

  pinMode(A4, INPUT);
```

```
pinMode(A1, OUTPUT);

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

servopanjang.attach(8); // attaches the servo on pin 8 to the servo object
servolebar.attach(9); // attaches the servo on pin 9 to the servo object

servopanjang.write(0);
servolebar.write(0);

delay(500);

digitalWrite(A1, HIGH);

lcd.clear();

lcd.setCursor(0, 0); lcd.print("Pengukur Kaki ");
lcd.setCursor(0, 1); lcd.print("Pemilihan Sepatu");

delay(2000);

digitalWrite(A1, LOW);

}

void loop() {

  limitpanjang = digitalRead(A5);
  limitlebar = digitalRead(A4);

  lcd.clear();

  analogkaki = analogRead(A0) / 4;

  lcd.setCursor(0, 0); lcd.print("P:");
```

```
lcd.setCursor(2, 0); lcd.print(simpanpanjang);
```

```
lcd.setCursor(5, 0); lcd.print("L:");
```

```
lcd.setCursor(7, 0); lcd.print(simpanlebar);
```

```
lcd.setCursor(10, 0); lcd.print("|S=");
```

```
lcd.setCursor(0, 1); lcd.print(analogkaki);
```

```
  bacaSensor();
```

```
  lcd.setCursor(13, 0);
```

```
  if(simpanpanjang < 20) {lcd.print("Er"); }
```

```
  if(simpanpanjang >= 20 && simpanpanjang < 21){lcd.print("35"); }
```

```
  if(simpanpanjang >= 21 && simpanpanjang < 22){lcd.print("36"); }
```

```
  if(simpanpanjang >= 22 && simpanpanjang < 23){lcd.print("37"); }
```

```
  if(simpanpanjang >= 23 && simpanpanjang < 24){lcd.print("38"); }
```

```
  if(simpanpanjang >= 24 && simpanpanjang < 25){lcd.print("39"); }
```

```
  if(simpanpanjang >= 25 && simpanpanjang < 26){lcd.print("40"); }
```

```
  if(simpanpanjang >= 26 && simpanpanjang < 27){lcd.print("41"); }
```

```
  if(simpanpanjang >= 27 && simpanpanjang < 28){lcd.print("42"); }
```

```
  if(simpanpanjang >= 28 && simpanpanjang < 29){lcd.print("43"); }
```

```
  if(simpanpanjang >= 29 && simpanpanjang < 30){lcd.print("44"); }
```

```
  if(simpanpanjang > 30) {lcd.print("Er"); }
```

```
  if(analogkaki < 100){
```

```
    jangandiulang = 0;
```

```
    servopanjang.write(0);
```

```
    servolebar.write(0);
```

```
}
```

```
if(analogkaki > 240 && jangandiulang == 0 && panggil == 0)
```

```
{ delay(2000); servopanjang.write(40); bacasensor(); }
```

```
if (panggil==1)
```

```
{ panggil = 0; servolebar.write(50); bacasensor(); }
```

```
delay(100);
```

```
}
```

```
//************************************************************************
```

```
void bacasensor(){
```

```
digitalWrite(TRIGPIN, LOW); delayMicroseconds(2);
```

```
digitalWrite(TRIGPIN, HIGH); delayMicroseconds(10);
```

```
digitalWrite(TRIGPIN, LOW);
```

```
distance = pulseIn(ECHOPIN, HIGH);
```

```
distance= distance/58;
```

```
lcd.setCursor(3, 1); lcd.print("P:");
```

```
lcd.setCursor(5, 1); lcd.print(distance); lcd.setCursor(7, 1); lcd.print("cm");
```

```
if(limitpanjang == LOW)
```

```
{
```

```
delay(1000); simpanpanjang = distance - 3;
```

```
delay(500); digitalWrite(A1, HIGH);
```

```
servopanjang.write(0); panggil = 1; delay(500); digitalWrite(A1, LOW);
```

```
}
```

```
digitalWrite(TRIGPIN2, LOW); delayMicroseconds(2);
```

```
digitalWrite(TRIGPIN2, HIGH); delayMicroseconds(10);
```

```
digitalWrite(TRIGPIN2, LOW);
```

```
distance2 = pulseIn(ECHOPIN2, HIGH);
```

```
distance2= distance2/58;
```

```
lcd.setCursor(10, 1); lcd.print("L:");
```

```
lcd.setCursor(12, 1); lcd.print(distance2); lcd.setCursor(14, 1); lcd.print("cm");
```

```
if(limitlebar== LOW)
```

```
{
```

```
delay(1000); simpanlebar = distance2;
```

```
delay(500); digitalWrite(A1, HIGH);
```

```
servolebar.write(0); panggil = 0; delay(500); jangandiulang = 1; digitalWrite(A1, LOW);
```

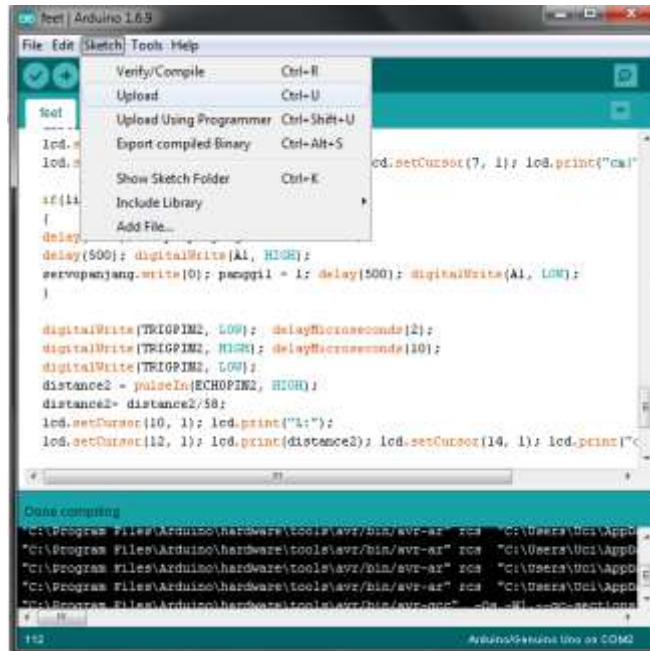
```
delay(200); digitalWrite(A1, HIGH); delay(500); digitalWrite(A1, LOW);
```

```
}
```

```
}
```

```
//*********************************************************************
```

- d. Proses berikutnya adalah melakukan *Verify/Compile* program dan *Upload* program, dengan memilih menu *Sketch -> Upload* pada *software* Arduino 1.6.9, seperti pada gambar di bawah berikut ini :



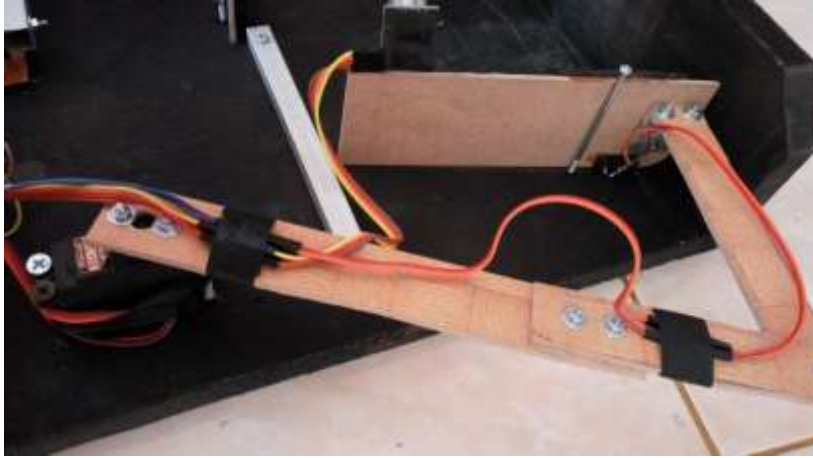
**Gambar IV.5. Proses Upload Program Software Arduino 1.6.9**

- e. Setelah proses *upload* program selesai terhadap rangkaian mikrokontroler, maka dapat dilihat kinerja dari perangkat berjalan sesuai dengan program yang diperintahkan dengan melakukan pengujian perangkat secara *hardware*.

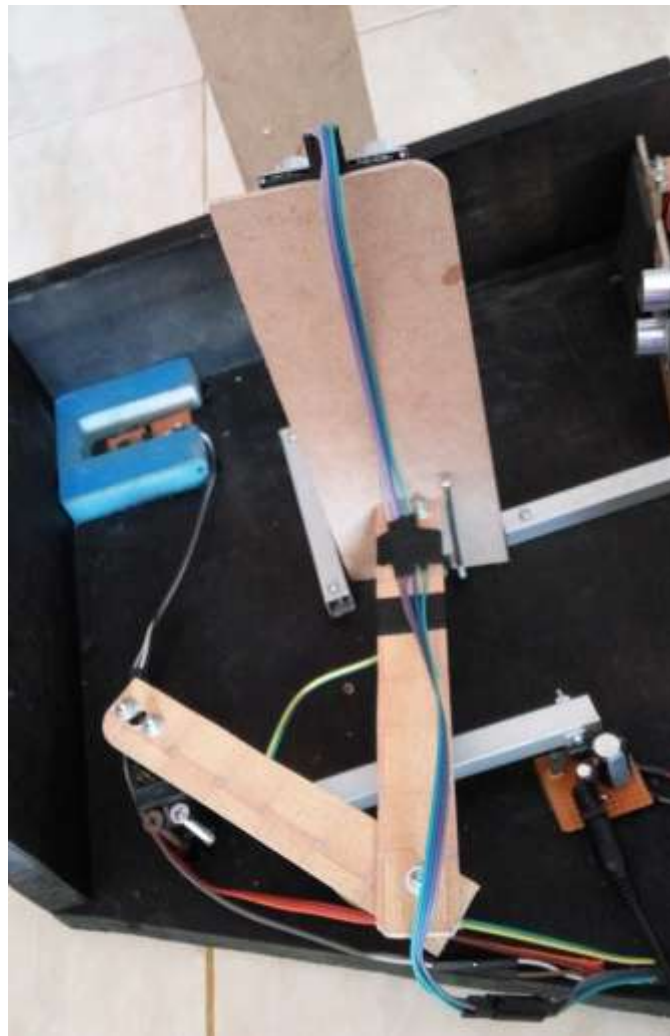
## IV.5. Pengujian *Hardware*

Setelah semua rangkaian yang telah selesai dirancang pada perancangan pengukur kaki dan penentu ukuran sepatu, kemudian dilakukan penyatuan semua rangkaian yang telah selesai dengan mekanik. Berikut adalah gambar hasil dari perancangan mekanik dan eletronik ditunjukkan oleh gambar IV.6 di bawah ini :

### A. Mekanik Pengukur Panjang Kaki



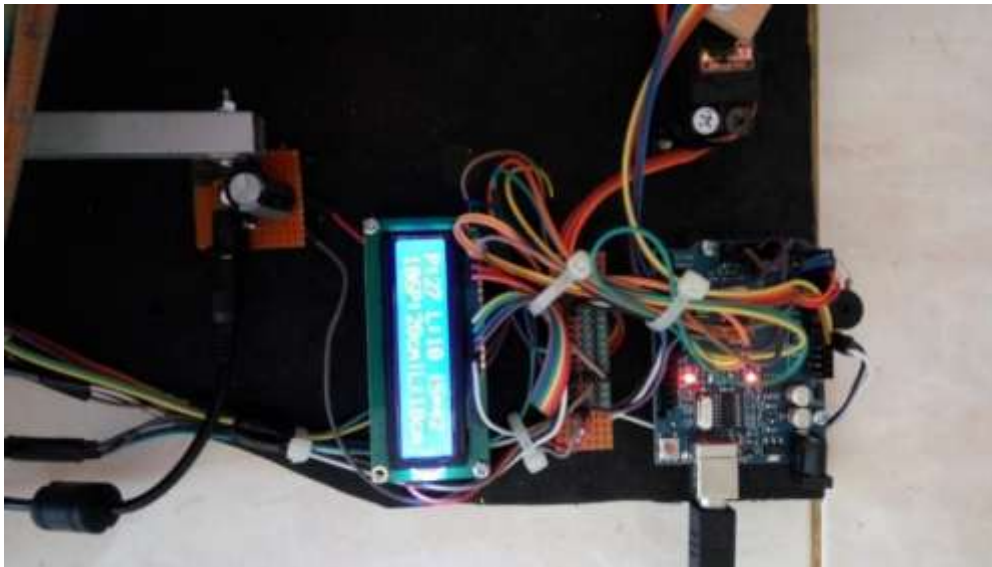
**B. Mekanik Pengukur Lebar Kaki**



**C. Posisi Peletakan Sensor Infrared dan Photodioda**



#### **D. Posisi Peletakan Komponen Elektronik**



**Gambar IV.6. Hardware Mekanik dan Elektronik**

#### **IV.6. Uji Coba Perangkat**

Setelah semua komponen terpasang dan program selesai disusun, maka langkah berikutnya adalah melakukan pengujian alat. Pengujian ini dilakukan secara bertahap dari rangkaian ke rangkaian berikutnya.

### IV.6.1. Pengujian Rangkaian Mikrokontroler Arduino Uno

Untuk mengetahui apakah rangkaian mikrokontroler Arduino Uno telah bekerja dengan baik, maka dilakukan pengujian. Pengujian bagian ini dilakukan dengan memberikan program pada mikrokontroler Arduino Uno. Program sederhana yang digunakan adalah *blink led* pada Pin 13 Arduino. Berikut adalah listing program dari *blink led* :

```
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off  
  delay(1000);           // wait for a second  
}
```

Jika setelah *upload* program dilakukan dan led berkedip setiap 1000 milisekon (1 detik) maka arduino dalam keadaan baik.

### IV.6.3. Pengujian Rangkaian Regulator Tegangan

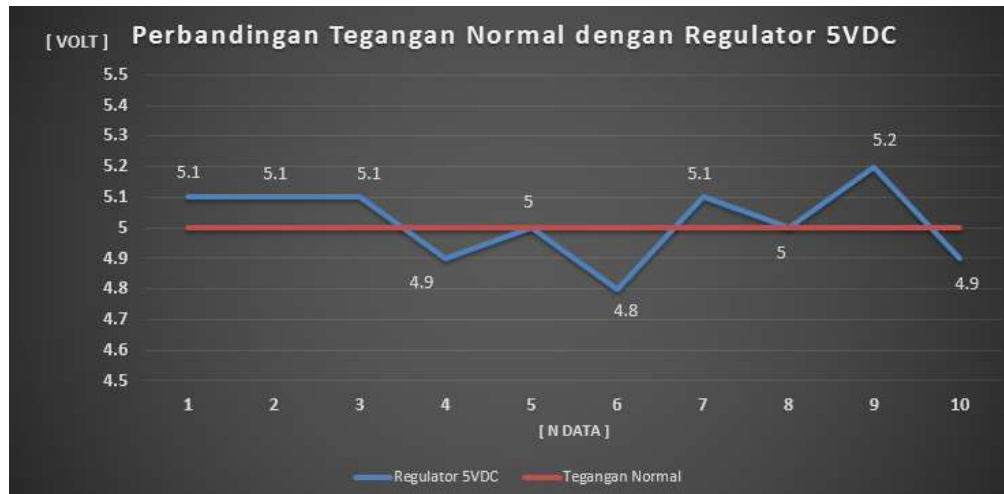
Pengujian ini dilakukan untuk mengukur tegangan yang dihasilkan dari regulator tegangan LM7805. Regulator mengubah tegangan 12VDC dari Adaptor menjadi 5VDC untuk tegangan kerja perangkat keseluruhan. Pengukuran dilakukan menggunakan multimeter atau *voltmeter*. Berikut adalah hasil dari pengukuran tegangan, ditunjukkan pada tabel IV.1 :

**Tabel IV.1. Hasil Pengujian Regulator Tegangan 5VDC**

<b>No. Pengujian</b>	<b>Hasil Pengukuran (Volt)</b>	<b>Error (Volt)</b>
1	5,1	0,1
2	5,1	0,1
3	5,1	0,1
4	4,9	0,1
5	5	0
6	4,8	0,2
7	5,1	0,1
8	5	0
9	5,2	0,2
10	4,9	0,1
<b><math>\Sigma</math> Error</b>		<b>1,0</b>
<b>Rata – Rata Error</b>		<b>0,1</b>

Berdasarkan data dari tabel di atas, disimpulkan bahwa *error* dari tegangan normal dengan tegangan regulator 5VDC memiliki total selisih *error*  $\pm 01$  Volt pada 10 kali pengujian

(n) atau rata – rata *error* sebesar 0,1 Volt. Berdasarkan tabel diatas, dapat digambarkan pada grafik di bawah ini :



**Gambar IV.7. Grafik Hasil Perbandingan Tegangan Normal dengan Regulator 5VDC**

Terlihat hasil kedua pengujian hampir sama dengan selisih nilai pengukuran yang kecil. Kesimpulan dari pengujian ini adalah regulator tegangan 5VDC dapat dinyatakan berkerja dengan baik dan sesuai jika dibandingkan dengan tegangan kerja normal yaitu 5VDC.

### **IV.6.3. Pengujian Rangkaian Servo**

Pengujian rangkaian servo dilakukan untuk mengetahui apakah motor servo dapat berkerja menggerakkan mekanik pengukur panjang kaki dan lebar kaki dengan baik atau tidak. Pengujian ini dilakukan dengan mengakses setiap motor servo. Motor servo diberikan nilai minimum dan maksimum dari nilai servo setiap 1000 milisekon (1 detik). Berikut adalah gambar *code* untuk melakukan pengujian servo :

```
Sweep | Arduino 1.6.9
File Edit Sketch Tools Help
Sweep
// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position

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

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable
    delay(15); // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos); // tell servo to go to position in variable
    delay(15); // waits 15ms for the servo to reach the position
  }
}
```

**Gambar IV.8. Listing Program Pengujian Motor Servo**

#### **IV.6.4. Pengujian Rangkaian Sensor Jarak SRF04**

SRF04 sebagai sensor jarak yang mengirimkan data jarak terhadap objek. Pengujian ini dilakukan dengan mencoba mengirimkan data jarak ke mikrokontroler. Data pembandingan yang digunakan adalah penggaris dengan pengambilan data 10 cm – 40 cm.



### Gambar IV.9. Pengujian Rangkaian Sensor Jarak SRF04

Berikut *listing* program untuk menampilkan data sensor jarak SRF04 dalam cm :

```
#define ECHOPIN 2    // Pin to receive
```

```
echo pulse
```

```
#define TRIGPIN 3    // Pin to send
```

```
trigger pulse
```

```
void setup(){
```

```
  Serial.begin(9600);
```

```
  pinMode(ECHOPIN, INPUT);
```

```
  pinMode(TRIGPIN, OUTPUT);
```

```
}
```

```
void loop(){
```

```
  digitalWrite(TRIGPIN, LOW);
```

```
  //Set the trigger pin to low for 2uS
```

```
  delayMicroseconds(2);
```

```
  // Send a 10uS high to trigger ranging
```

```
  digitalWrite(TRIGPIN, HIGH);
```

```
  delayMicroseconds(10);
```

```
  // Send pin low again
```

```
digitalWrite(TRIGPIN, LOW);  
  
// Read in times pulse  
int distance = pulseIn(ECHOPIN, HIGH);  
  
// Calculate distance from time of pulse  
distance= distance/58;  
Serial.println(distance);  
delay(50);  
}
```

```

SRF | Arduino 1.6.9
File Edit Sketch Tools Help
SRF
#define ECHOPIN 2 // Pin to receive
echo pulse
#define TRIGPIN 3 // Pin to send
trigger pulse

void setup() {
  Serial.begin(9600);
  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
}

void loop() {
  digitalWrite(TRIGPIN, LOW);
  //Set the trigger pin to low for 2uS
  delayMicroseconds(2);
  // Send a 10uS high to trigger ranging
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(10);
  // Send pin low again
  digitalWrite(TRIGPIN, LOW);
  // Read in times pulse
  int distance = pulseIn(ECHOPIN, HIGH);

  // Calculate distance from time of pulse
  distance= distance/58;
  Serial.println(distance);
  delay(50);
}

Done Saving.

18 Arduino/Genuino Uno on COM1

```

**Gambar IV.10. Listing Program Pengujian Motor Servo**

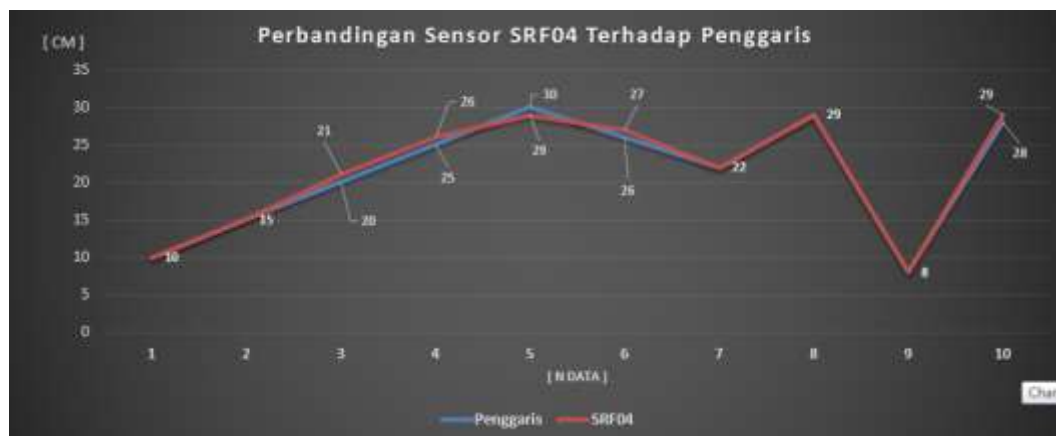
Berikut adalah hasil dari pengukuran sensor jarak terhadap penggaris, ditunjukkan pada tabel IV.2 :

**Tabel IV.2. Hasil Pengujian Sensor Jarak Terhadap Penggaris**

No. Pengujian	Hasil Pengukuran (cm)		Error (cm)
	Penggaris	SRF04	

1	10	10	0
2	15	15	0
3	20	21	1
4	25	26	1
5	30	29	1
6	26	27	1
7	22	22	0
8	29	29	0
9	8	8	0
10	28	29	1
<b><math>\Sigma</math> Error</b>			<b>0,5</b>
<b>Rata – Rata Error</b>			<b>0,05</b>

Berdasarkan data dari tabel di atas, disimpulkan bahwa *error* dari perhitungan jarak penggaris dengan SRF04 memiliki total selisih *error*  $\pm 0,5$  cm pada 10 kali pengujian (n) atau rata – rata *error* sebesar 0,05 cm. Berdasarkan tabel diatas, dapat digambarkan pada grafik di bawah ini :



#### **Gambar IV.11. Grafik Hasil Perbandingan Sensor Jarak SRF04 Terhadap Penggaris**

#### **IV.6.5. Pengujian Analisa Perangkat Keseluruhan**

Pengujian ini dilakukan untuk mengetahui apakah perancangan pengukur kaki dan penentu ukuran sepatu bekerja sesuai dengan logika program. Sebelum pengujian dilakukan, perangkat dalam telah menyala program tampilan awal keadaan normal. Setiap proses akan menghidupkan buzzer selama beberapa saat. Berikut adalah gambar dari kondisi awal perancangan pengukur kaki dan penentu ukuran sepatu:



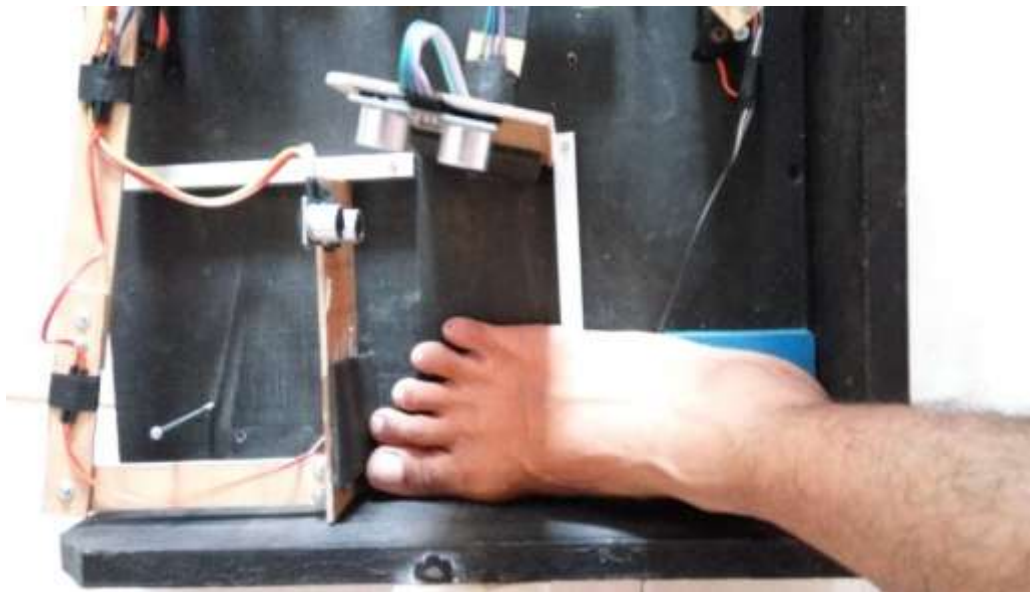
**Gambar IV.12. Tampilan Awal**

Proses berikutnya adalah pengguna meletakkan kaki pada posisi yang sudah ditentukan. Sensor infrared dan photodiode akan mendeteksi kaki dan akan menggerakkan motor servo. Proses tersebut ditunjukkan pada gambar di bawah ini :



**Gambar IV.13. Tampilan Posisi Meletakkan Kaki Awal**

Ketika mekanik pengukur panjang tapak kaki mendapatkan posisi menyentuh ujung kaki pengguna, sensor jarak SRF04 melakukan pengukuran panjang tapak kaki. Proses tersebut ditunjukkan pada gambar di bawah ini :



**Gambar IV.14. Tampilan Pengukuran Panjang Tapak Kaki**

Ketika mekanik pengukur panjang tapak kaki selesai melakukan proses, perangkat akan menggerakkan mekanik pengukur lebar tapak kaki dan sensor jarak SRF04 melakukan pengukuran lebar tapak kaki. Proses tersebut ditunjukkan pada gambar di bawah ini :



**Gambar IV.15. Tampilan Pengukuran Lebar Tapak Kaki**

Setelah proses pengukuran panjang dan lebar tapak kaki selesai dilakukan. Variabel panjang dan lebar tapak kaki yang telah didapatkan kemudian diproses dan diolah untuk mendapatkan ukuran sepatu yang sesuai dengan ukuran kaki. Perangkat akan kembali ke posisi normal dan siap untuk melakukan pengukuran kembali. Hasil dari pengukuran ditunjukkan pada LCD 16x2, seperti pada gambar di bawah ini :



**Gambar IV.16. Tampilan Hasil Pengukuran**

Berikut adalah hasil dari pengujian pengukuran kaki dan penentu ukuran sepatu, ditunjukkan pada tabel di bawah ini :

**Tabel IV.3. Hasil Pengukuran Kaki dan Penentuan Ukuran Sepatu**

No. Pengujian	Hasil Pengukuran (cm)		Ukuran Sepatu
	Panjang	Lebar	
1	27	10	42
2	28	10	43
3	29	11	44
4	26	10	41
5	25	10	40
6	27	10	42
7	25	11	40
8	26	10	41
9	25	10	40
10	26	10	41

#### **IV.7. Kelebihan dan Kekurangan**

Pada perancangan pengukur kaki dan penentu ukuran sepatu ini masih kurang sempurna. Perakitan dan pembuatan perangkat ini masih memiliki beberapa kelebihan dan kekurangan, diantaranya:

##### **a. Kelebihan**

Adapun beberapa kelebihan yang dimiliki, antara lain :

1. Dengan adanya perangkat ini, maka kita dapat mengukur panjang dan lebar kaki serta pembantu sebagai penentu ukuran sepatu sesuai dengan ukuran kaki pengguna yang telah diukur.
2. Kecepatan respon sensor infrared dan photodiode sangat baik dalam mendeteksi adanya kaki yang akan diukur.
3. Perangkat pengukur kaki dan penentu ukuran sepatu bekerja menggunakan *power supply* 12 VDC 1A dan regulator tegangan 5VDC.

##### **b. Kekurangan**

Adapun beberapa kekurangan yang dimiliki, antara lain:

1. Perancangan pengukur kaki dan penentu ukuran sepatu menggunakan 2 buah motor servo dengan rincian 1 motor servo untuk mengukur panjang dan 1 unit motor servo untuk mengukur lebar kaki, sehingga konsumsi tegangan menjadi besar.
2. Pengguna harus menyesuaikan posisi kaki pada perangkat untuk pengukuran yang tepat.
3. Hasil penentuan ukuran sepatu yang digunakan adalah ukuran sepatu untuk pria dewasa dengan ukuran sepatu berkisar pada ukuran 35 – 44.

4. Sensor jarak hanya dapat mengukur panjang kaki antara 20 cm – 30 cm dikarenakan keterbatasan mekanik perangkat.
5. Sumber daya yang dibutuhkan untuk mengendalikan banyak servo cukup besar dan terjadi pemanasan berlebih pada regulator tegangan (membutuhkan *heatsink* / pendingin berbahan aluminium untuk mendinginkan regulator tegangan)