

## BAB IV

### HASIL DAN PEMBAHASAN

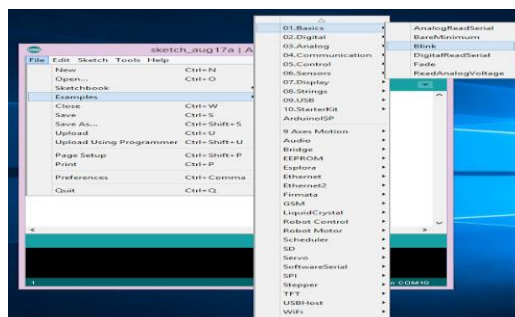
#### 4.1 Pembahasan dan Pengecekan Rangkaian

Setelah pembuatan seluruh rangkaian telah selesai, maka proses selanjutnya adalah melakukan pengujian alat dan membuat pembahasan tentang kinerja alat tersebut. Pengujian dilakukan tiap-tiap bagian rangkaian secara menyeluruh. Pengujian ini dilakukan dengan bertujuan untuk mengetahui bahwa alat bekerja dengan baik dan benar.

##### 4.1.1 Pengujian Mikrokontroler Arduino UNO

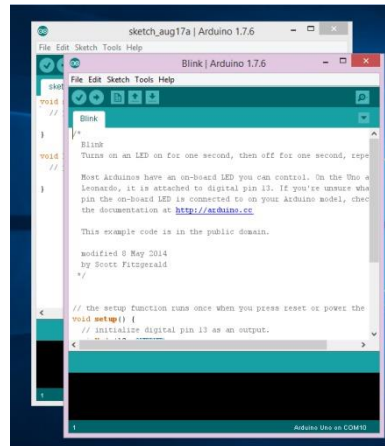
Pada pengujian mikrokontroler Arduino UNO dilakukan dengan menghubungkan kabel USB dengan mikrokontroler, kemudian pastikan dua lampu di *board* menyala. Lampu kuning menyala *nonstop* dan lampu *orange* berkedip berulang kali.

Langkah selanjutnya yaitu membuat program yang berfungsi untuk mengetahui apakah mikrokontroler dapat bekerja ketika memasukkan program ke dalam mikrokontroler tersebut menggunakan aplikasi Arduino IDE dengan cara masuk ke menu *file-examples-01.basic-blink*



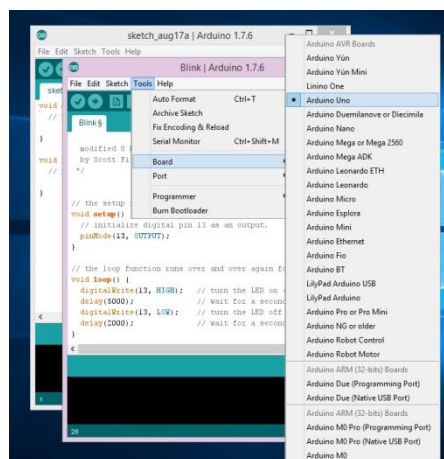
Gambar 4.1 halaman *blink*

Setelah itu akan muncul program *blink* yang sudah di setting secara *auto* untuk menjalankan perintah *blink*.



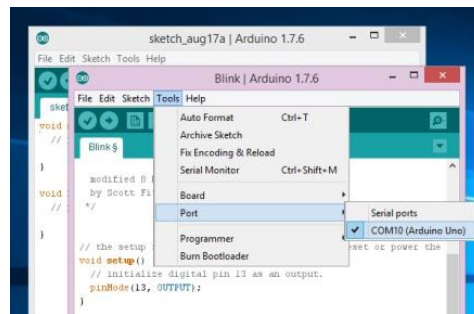
Gambar 4.2 Halaman Program *Blink*

Selanjutnya memilih port dan *board* yang akan kita hubungkan, pilih menu “*Tools-Board-Arduino uno*”



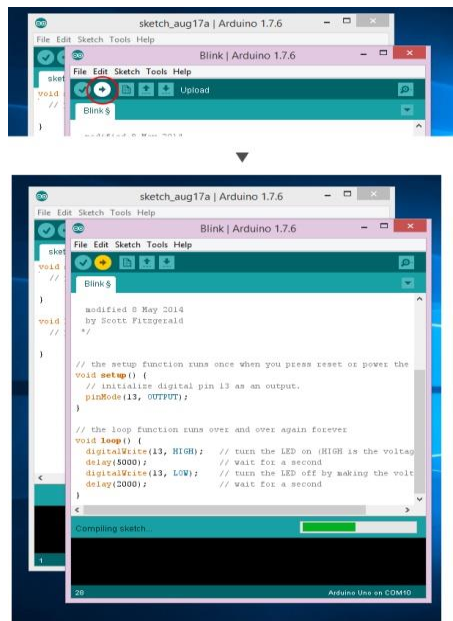
Gambar 4.3 Tampilan Board ke Arduino uno

Selanjutnya pastikan saluran *port* sesuai dengan *Board* pada mikrokontroler dengan mengeceknya, pilih menu “*Tools-Port-COM10(Arduino Uno)*”



Gambar 4.4 Tampilan Pemilihan *Port*

Selanjutnya jika sudah sesuai portnya, maka tinggal mengupload program yang sudah di pilih, klik *icon* panah untuk mengupload program kedalam mikrokontroler dan tunggu hingga selesai program terupload ke dalam mikrokontroler.

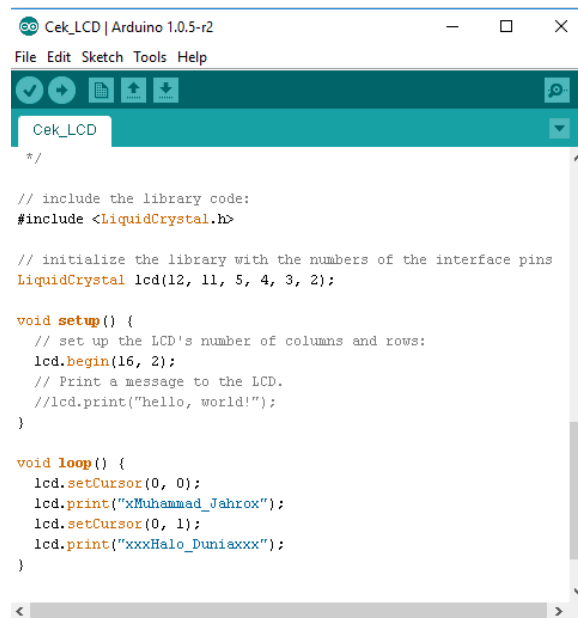


Gambar 4.5 Perintah Upload Program kedalam Mikrokontroler

Jika lampu berkedip secara bergantian dengan selang waktu 1000 ms berarti dapat di pastikan bahwa mikrokontroler tersebut berfungsi dengan baik.

### 4.1.2 Pengujian LCD (Liquid Crystal Display)

Pengujian LCD dilakukan agar dapat mengetahui bahwa dapat bekerja dengan baik. Pengujian LCD dilakukan dengan cara memasukkan teks di program Arduino UNO yang nantinya akan di tampilkan pada LCD tersebut. Berikut program yang di tampilkan pada LCD baris pertama “xMuhammad\_Jahrox” dan baris kedua “xxxHalo\_Duniaxxx” dan seperti ini yang tertampil pada LCD.



```
Cek_LCD | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Cek_LCD
*/
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

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

void loop() {
  lcd.setCursor(0, 0);
  lcd.print("xMuhammad_Jahrox");
  lcd.setCursor(0, 1);
  lcd.print("xxxHalo_Duniaxxx");
}
```

Gambar 4.6 Program LCD



Gambar 4.7 Tampilan pada LCD

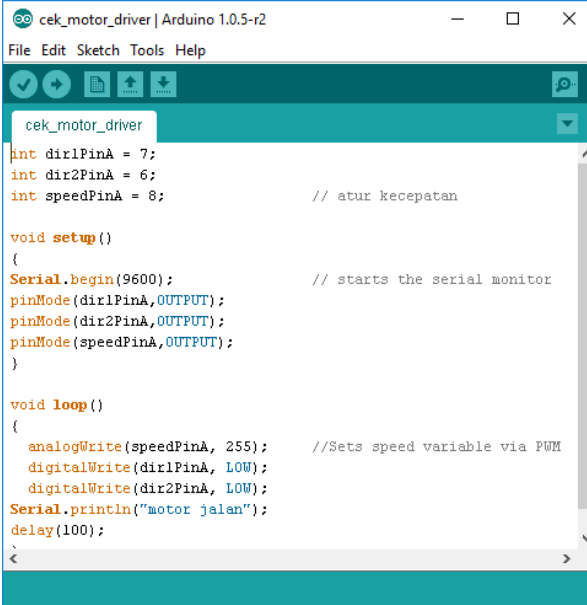
Tabel 4.1 Uji Karakter pada Tampilan *LCD 2x16*.

Posisi	Karakter	Keterangan	Posisi	Karakter	Keterangan
(1,0)	x	Tertampil	(1,1)	x	Tertampil
(2,0)	M	Tertampil	(2,1)	x	Tertampil
(3,0)	u	Tertampil	(3,1)	x	Tertampil
(4,0)	h	Tertampil	(4,1)	H	Tertampil
(5,0)	a	Tertampil	(5,1)	e	Tertampil
(6,0)	m	Tertampil	(6,1)	l	Tertampil
(7,0)	m	Tertampil	(7,1)	o	Tertampil
(8,0)	a	Tertampil	(8,1)	_	Tertampil
(9,0)	d	Tertampil	(9,1)	W	Tertampil
(10,0)	_	Tertampil	(10,1)	o	Tertampil
(11,0)	J	Tertampil	(11,1)	l	Tertampil
(12,0)	a	Tertampil	(12,1)	r	Tertampil
(13,0)	h	Tertampil	(13,1)	d	Tertampil
(14,0)	r	Tertampil	(14,1)	x	Tertampil
(15,0)	o	Tertampil	(15,1)	x	Tertampil
(16,0)	x	Tertampil	(16,1)	x	Tertampil

Dari tabel tersebut dapat di simpulkan bahwa seluruh *pixel* pada LCD dapat berfungsi dengan baik dan tidak ada *dot pixel* pada LCD.

#### 4.1.3 Pengujian Driver Motor DC (Motor Shield L298)

Pengujian pada driver motor dapat dilakukan dengan cara menghubungkan Pin *Enable driver* motor ke Pin PWM arduino UNO, kemudian hubungkan catu daya pada *driver* motor dengan regulator 12 Volt, untuk outputnya dapat dilihat serial monitor pada Arduino IDE.



```
cek_motor_driver | Arduino 1.0.5-r2
File Edit Sketch Tools Help

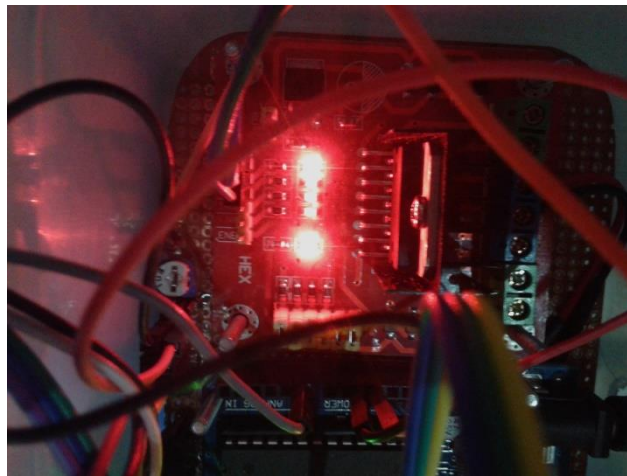
cek_motor_driver

int dir1PinA = 7;
int dir2PinA = 6;
int speedPinA = 8;           // atur kecepatan

void setup()
{
  Serial.begin(9600);        // starts the serial monitor
  pinMode(dir1PinA, OUTPUT);
  pinMode(dir2PinA, OUTPUT);
  pinMode(speedPinA, OUTPUT);
}

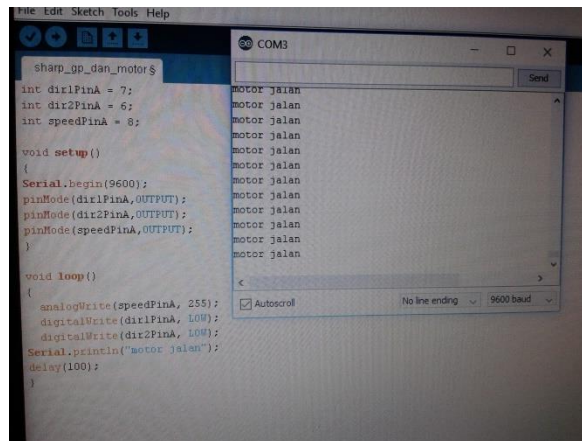
void loop()
{
  analogWrite(speedPinA, 255); //Sets speed variable via PWM
  digitalWrite(dir1PinA, LOW);
  digitalWrite(dir2PinA, LOW);
  Serial.println("motor jalan");
  delay(100);
}
```

Gambar 4.8 Program Driver Motor



Gambar 4.9 Lampu Indikator Pada Driver Motor

Lampu indikator menyala menandakan bahwa driver motor sedang bekerja.

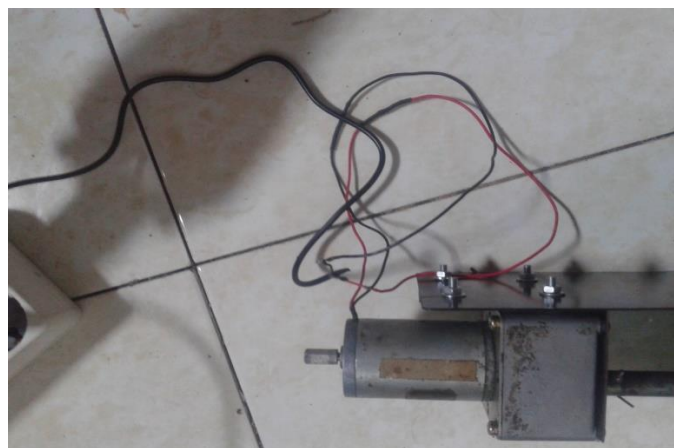


Gambar 4.10 Output Driver Motor pada Serial Monitor

Tampilan serial monitor pada saat motor *driver* bekerja yang di tampilkan melalui serial monitor.

#### 4.1.4 Pengujian Motor DC

Pengujian motor DC yaitu dengan cara menghubungkan adaptor secara langsung, jika berputar dengan lancar dapat dipastikan motor tersebut dalam kondisi baik.

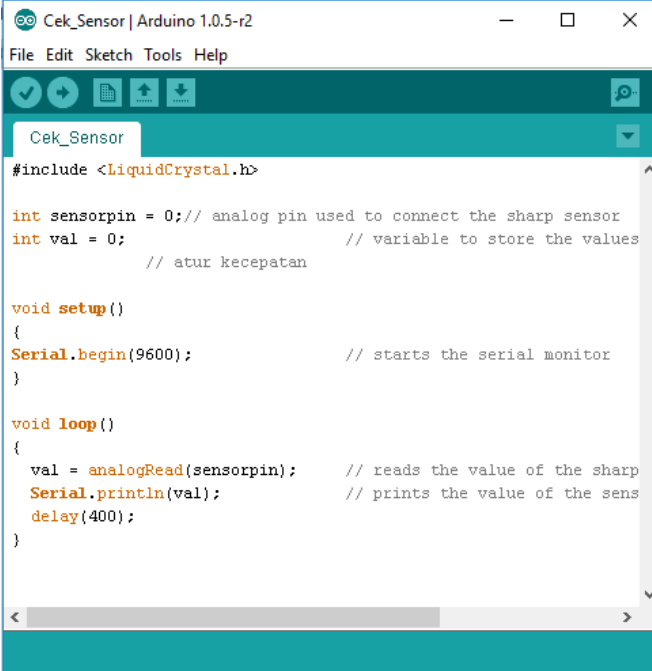


Gambar 4.11 Menguji coba motor DC

Motor DC berjalan ketika diberi tegangan melalui adaptor 12 Volt dan perputarannya lancar tidak tersendat.

### 4.1.5 Pengujian Sensor Sharp GP

Pengujian sensor sharp GP ini dilakukan dengan menghubungkan ke mikrokontroler arduino UNO, kemudian di tampilkan ke LCD sebagai outputnya agar bisa terlihat nilai sensor yang terbaca.



```
Cek_Sensor | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Cek_Sensor
#include <LiquidCrystal.h>

int sensorpin = 0; // analog pin used to connect the sharp sensor
int val = 0; // variable to store the values
// atur kecepatan

void setup()
{
  Serial.begin(9600); // starts the serial monitor
}

void loop()
{
  val = analogRead(sensorpin); // reads the value of the sharp
  Serial.println(val); // prints the value of the sens
  delay(400);
}
```

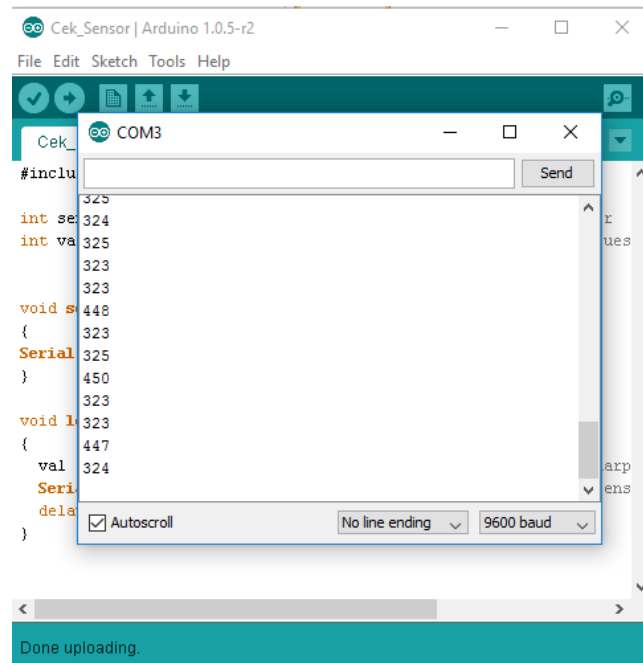
Gambar 4.12 Program Sensor Sharp GP



Gambar 4.13 Pengujian Sensor Sharp GP



Ketika diberi halangan maka sensor akan membaca jarak dari titik sensor ke titik benda yang menghalanginya.



Gambar 4.14 Tampilan Serial Monitor

Tampilan serial monitor pada saat pengujian sensor yang dihalangi oleh objek, dan dapat dipastikan sensor bekerja dengan baik.



Gambar 4.15 Pengukuran Jarak minimal sensor sharp GP

Pengukuran jarak minimal sensor sharp GP yaitu 6 cm.



Gambar 4.16 Tampilan Nilai Sensor Minimal di LCD.

Tampilan jarak minimal sensor sharp GP adalah 651 pada LCD.



Gambar 4.17 Pengukuran Jarak Maksimal Sensor Sharp GP

Dari pengujian sensor ini jarak maksimal yang diperoleh dari sensor sharp GP adalah 80 cm, karena spesifikasi sensor tersebut jarak maksimal adalah 80 cm.



Gambar 4.18 Tampilan maksimal Nilai sensor Sharp GP di LCD

Pengujian jarak maksimal sensor sharp GP yang di tampilkan pada LCD adalah 105.

#### 4.1.6 Pengujian Catu Daya

Catu daya yang digunakan pada komponen alat ini adalah regulator *12Volt 1 Ampere AC-DC* yang akan mensuplay seluruh komponen elektronik diantaranya : *Arduino UNO, Driver Motor, sensor Sharp GP, LCD, Push Button,* dan *Motor DC*.



Gambar 4.19 Regulator 12 Volt

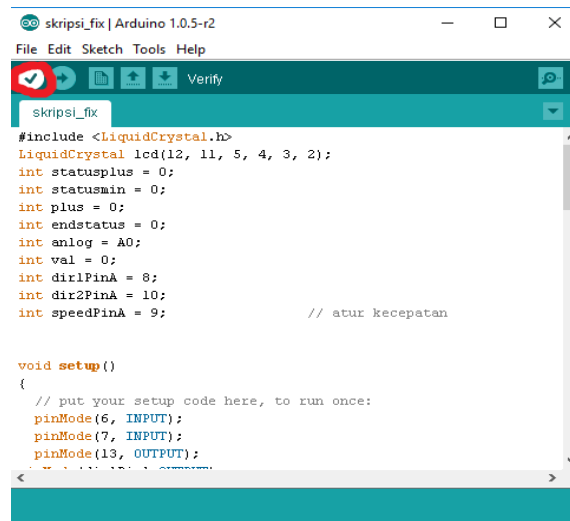
Tabel 4.2 Catu Daya Perkomponen

Komponen	Voltase
Sensor Sharp GP	5 Volt
Arduino UNO	5 Volt
Driver Motor	12 Volt
Push Button	5 Volt
Motor DC	12 Volt
LCD	5 Volt

#### 4.2 Pengujian Pemograman

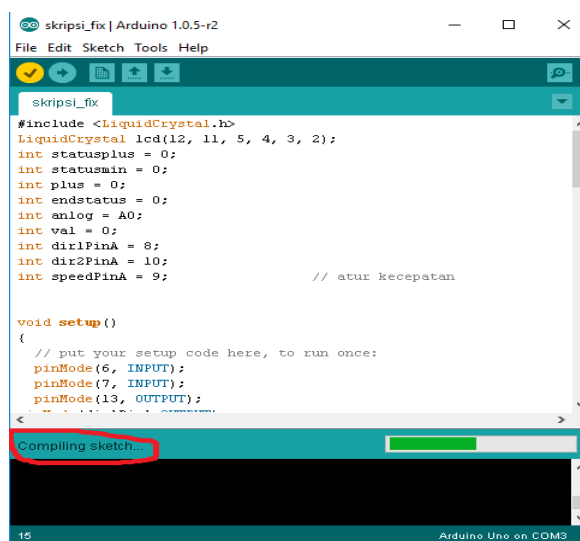
Pengujian pemograman dilakukan setelah perangkat hardware selesai dibuat, dan apakah perangkat berjalan sesuai dengan yang diinginkan. Kemudian masukan program assambly yang sudah dibuat pada aplikasi Arduino IDE pada mikrokontroler arduino UNO dengan cara, sebagai berikut :

1. Tancapkan *port* USB ke laptop dan soket pada Arduino UNO.
2. Hubungkan adaptor 12 Volt DC pada Arduino UNO dan *jumper* tegangan 12 Volt ke *driver* motor yang membutuhkan tegangan buat memutar motor DC.
3. *File* program yang sudah di ketik pada program Arduino IDE dan melakukan *verifikasi* pada program tersebut.

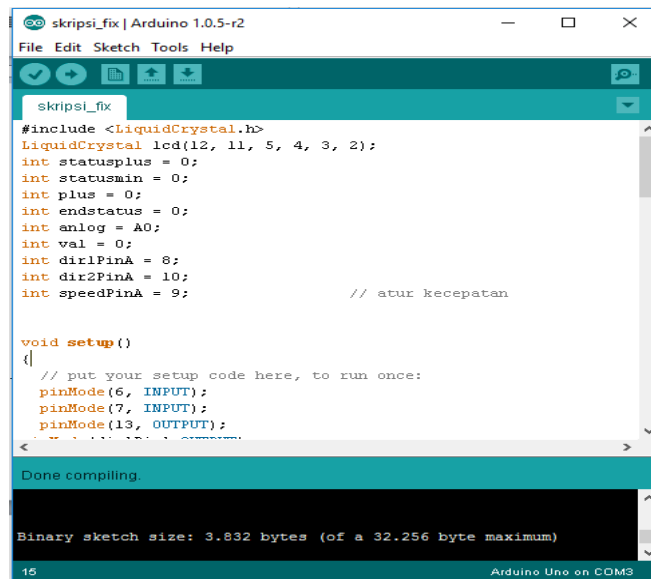


Gambar 4.20 Verifikasi program pada Arduino UNO

4. Proses melakukan pengecekan *verifikasi* pada program, apakah terdapat kesalahan atau tidak.



Gambar 4.21 Proses Verifikasi.



```

skripsi_fix | Arduino 1.0.5-r2
File Edit Sketch Tools Help
skripsi_fix
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int statusplus = 0;
int statusmin = 0;
int plus = 0;
int endstatus = 0;
int analog = A0;
int val = 0;
int dir1PinA = 8;
int dir2PinA = 10;
int speedPinA = 9; // atur kecepatan

void setup()
{
  // put your setup code here, to run once:
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  pinMode(13, OUTPUT);
}

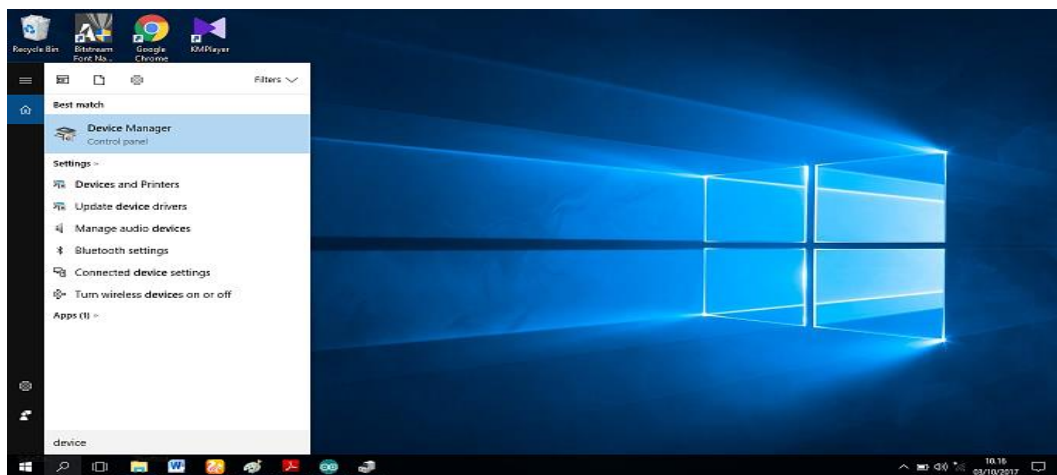
Done compiling.

Binary sketch size: 3.832 bytes (of a 32.256 byte maximum)
15 Arduino Uno on COM3

```

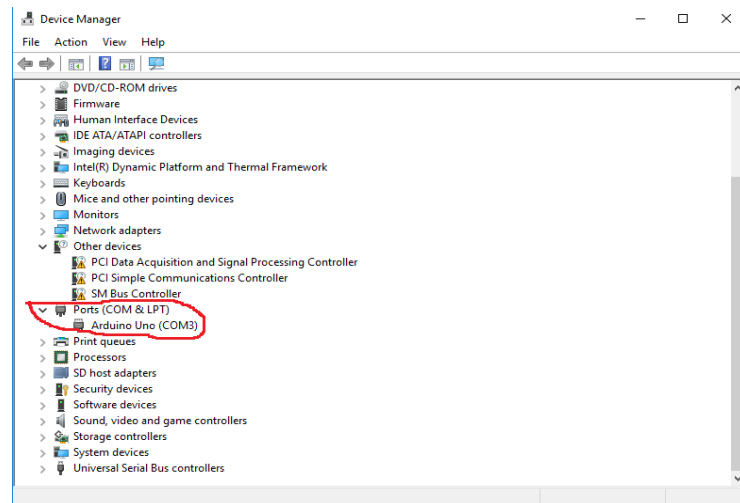
Gambar 4.22 Verifikasi selesai.

5. Jika selama *verifikasi* tidak terjadi kesalahan, maka selanjutnya akan di lakukan pengecekan *port* USB yang terbaca pada laptop dengan pengecekan *Device Meneger* pada laptop.



Gambar 4.23 Search *Device Meneger* pada laptop.

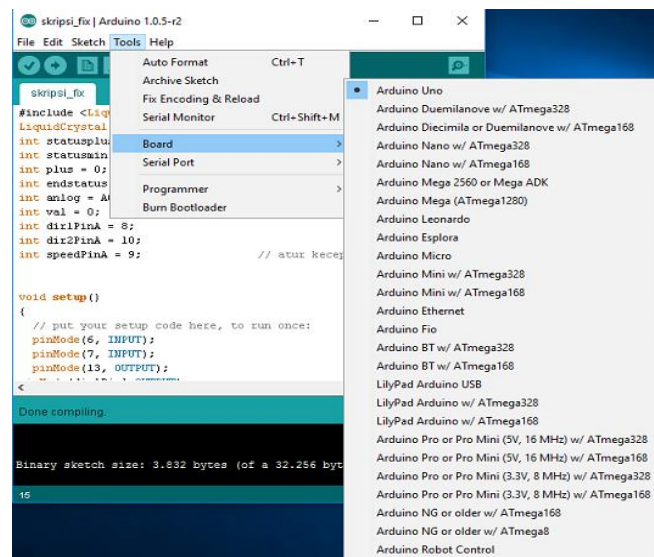
Ketik *Device Meneger* pada *icon Search Windows* maka akan muncul menu *Device Meneger* tersebut.



Gambar 4.24 Tampilan *Device Meneger*

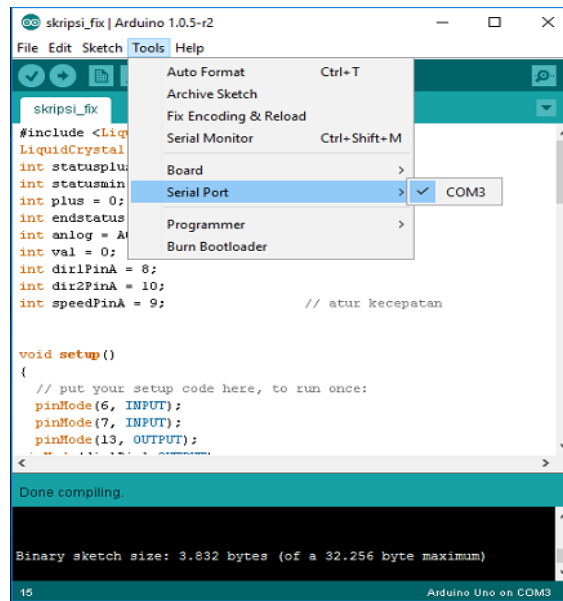
Ketika sudah di menu *Device meneger* kemudian klik menu *Ports (COM & LPT)* dan terdapat menu *Arduino UNO (COM3)*, maka *Port USB Arduino UNO* pada laptop membaca Port tersebut berada di *COM 3*.

6. Selanjutnya melakukan pengecekan pada Board dan Serial Port.



Gambar 4.25 Tampilan menu *Tools-Board-arduino UNO*

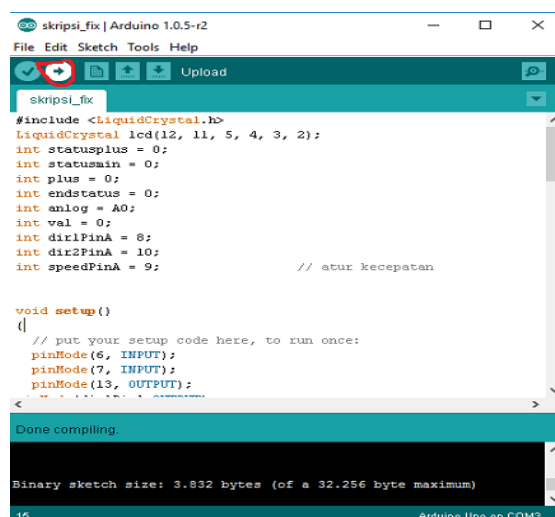
Pastikan terlebih dahulu bahwa *Board* tersebut ialah *Arduino UNO*.



Gambar 4.26 Tampilan Serial Port-COM 3

Sebelum mentransfer program tersebut, pastikan *Board*, *Serial Port* sudah sesuai dengan jenis mikrokontroler Arduino UNO dan *port* USB yang terbaca oleh laptop yaitu COM 3.

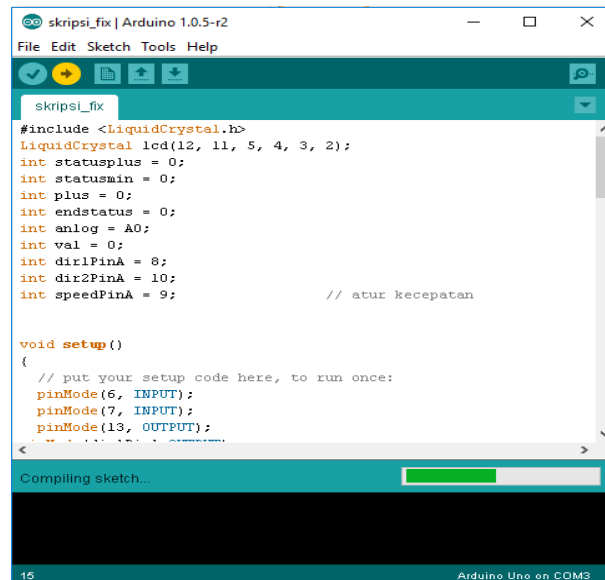
7. Jika semua serial *port* dan *board* sudah sesuai, maka tahap selanjutnya mengupload program pada mikrokontroler Arduino UNO.



Gambar 4.27 Tampilan Icon Upload.

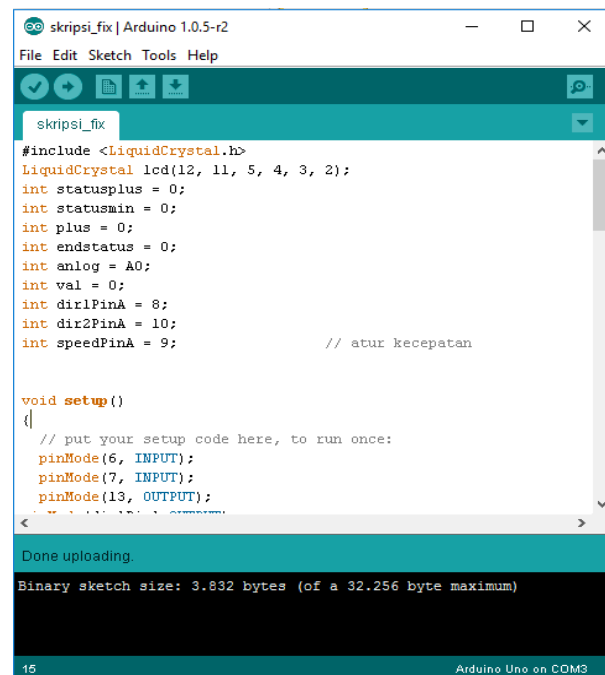


Klik tombol *Icon Upload* untuk mentransfer data ke mikrokontroler Arduino UNO.



Gambar 4.28 Proses Upload Program Pada Arduino UNO.

Pastikan ketika *Upload* data tidak terjadi error pada program tersebut.

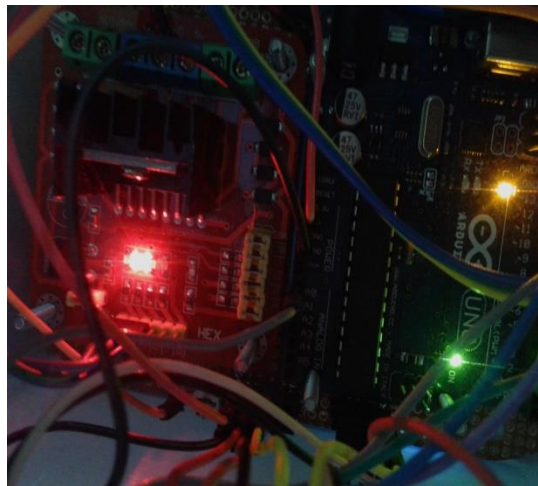


Gambar 4.29 Proses Upload telah selesai.

Setelah proses pengujian program selesai maka tinggal mengecek apakah program tersebut dapat berjalan dengan baik sesuai perintah yang diinginkan.



Gambar 4.30 Tampilan data sensor Sharp GP.



Gambar 4.31 Lampu indikator Arduino UNO dan Driver Motor.

Pengujian tersebut dapat di simpulkan bahwa program dapat berjalan dengan baik dengan indikasi nilai sensor dan set sensor dapat tertampil pada LCD, *push button UP* dan *DOWN* dapat berfungsi dengan baik serta lampu indikator merah pada *Driver* Motor dan Orange pada Mikrokontroler Arduino UNO yang menandakan komponen sedang bekerja.

### 4.3 Pengujian Alat

Setelah selesai merangkai dan komponen elektronik terisi program, langkah selanjutnya adalah melakukan uji coba alat. Pada alat tersebut terdapat adaptor 12 Volt yang berfungsi sebagai catu daya seluruh komponen elektronik dan motor. Tombol push button Up dan Down yang berfungsi sebagai masukan nilai sensor Sharp GP. Pengujian cara kerja sensor Sharp GP yaitu jika nilai sensor melebihi dari nilai seting sensor maka motor akan berhenti dengan otomatis, dan jika nilai sensor lebih kecil dari nilai setting sensor maka motor akan berjalan sampai nilai sensor sama dengan nilai setting sensor.

Pengujian alat selanjutnya dilakukan dengan uji coba pengangkutan pakan ayam dari titik wadah pengisian sampai titik wadah pakan terakhir hingga terisi penuh, nilai sensor sebelum wadah pakan terakhir terisi penuh adalah 464, nilai setting sensor jika terisi penuh adalah 600 dan massa pakan masing-masing sebanyak 1 kg yang terdiri dari pakan *mesh*, *crumble*, dan *pellet*.

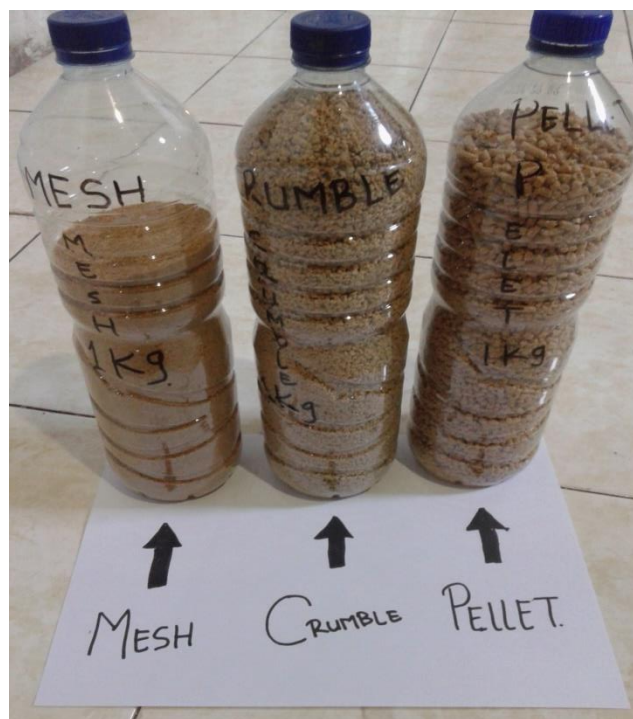


Gambar 4.32 Nilai sensor dalam keadaan wadah pakan kosong.



Gambar 4.33 Nilai Setting sensor jika wadah pakan terisi penuh.

Tampilan nilai sensor ketika pakan sudah terisi penuh pada wadah pakan tersebut.



Gambar 4.34 Bentuk dan Jenis Pakan.

Jenis pakan yang digunakan pada budidaya ayam pedaging. Pakan tersebut memiliki massa jumlah yang sama akan tetapi memiliki volume yang berbeda.



Gambar 4.35 Total pakan yang digunakan.

Uji coba pada alat ini dengan cara menghitung kecepatan motor dalam mengisi wadah pakan sebanyak 2 buah dengan jenis pakan yang berbeda-beda.

Tabel 4.3 Data Pengisian Pakan

Jenis Pakan	Durasi Pengisian	Satuan Pakan
Pakan Mesh	0,7:40 menit	1 kg
Pakan Crumble	0,4:20 menit	1 kg
Pakan Pellet	0,5:40 menit	1 kg

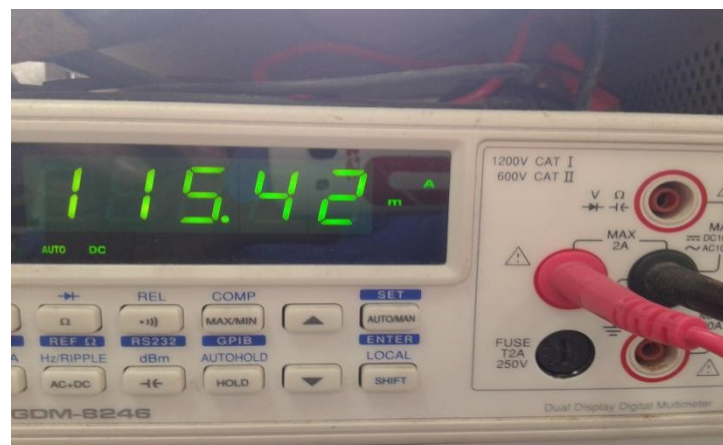
Dari data yang di peroleh, dapat disimpulkan bahwa perbedaan jenis dan bentuk pakan berpengaruh pada lama tidaknya pakan tersebut di distribusikan oleh motor dan spiral. Selain itu juga jenis spiral dan motor DC yang digunakan juga berperan dalam cepat tidaknya pakan yang didistribusikan. Sebab spiral yang digunakan tidak pipih melainkan bulat, sehingga mempersulit pakan jenis *Mesh*

dan *Pellet*. Dikarenakan pakan tersebut tidak terbawa secara keseluruhan melainkan terbawa sebagian, lain halnya dengan pakan yang berjenis *Crumble* yang notabennya terdiri dari pecahan pakan jenis *pellet* dan *mesh* sehingga lebih mudah didistribusikan. Kemudian jenis motor DC yang digunakan yaitu yang memiliki torsi. Torsi yang digunakan pada motor DC tersebut adalah perbandingan *gear rasio* 1:200, maka secara otomatis pergerakan motor DC memutar spiral juga berpengaruh dengan kecepatan dalam mengisi tempat pakan ayam tersebut.

#### 4.4 Pengukuran Konsumsi Arus

Setelah selesai menguji pendistribusian pakan ayam tersebut kemudian dilakukan pengukuran konsumsi arus rangkaian pada saat menjalankan alat tersebut.

Pada saat alat dinyalakan dan motor DC belum berjalan maka arus yang mengalir pada rangkaian sebesar 115,42 mA.



Gambar 4.36 Arus yang mengalir pada saat dinyalakan dengan kondisi motor DC tidak menyala.

Kemudian pada saat motor DC menyala maka arus yang dihasilkan lebih besar yaitu 362.07 mA.

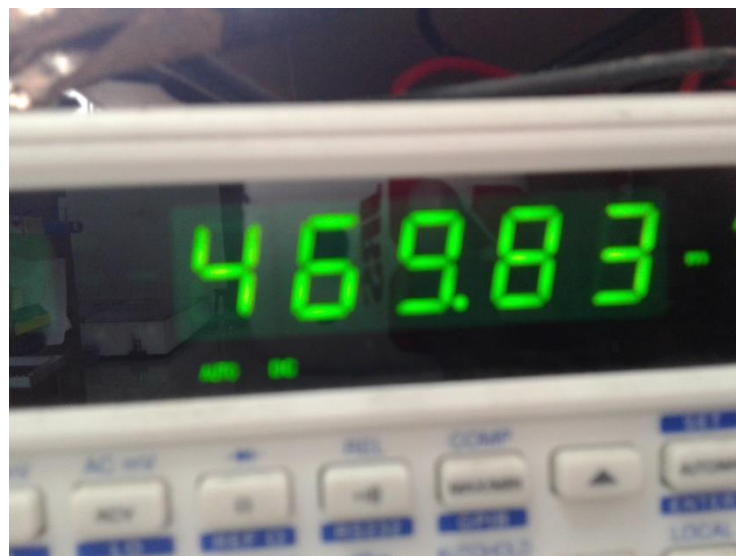


Gambar 4.37 Arus yang dihasilkan ketika motor DC menyala.

Pada saat motor DC diberi beban berupa pakan ayam maka arus yang dihasilkan berbeda berdasarkan jenis dan bentuk pakan.

a. Jenis Pakan *Crumble*

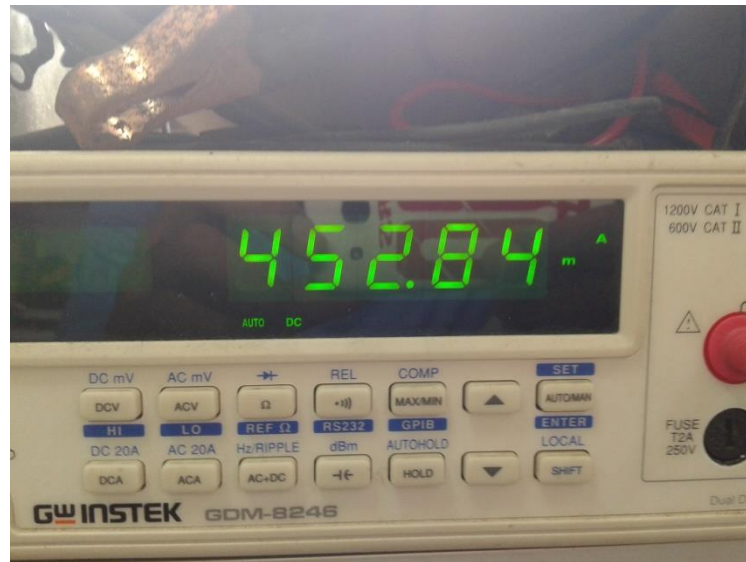
Konsumsi arus jenis pakan *crumble* sebesar 469,83 mA



Gambar 4.38 Konsumsi arus pada pakan ayam jenis *Crumble*.

b. Jenis Pakan Pellet

Konsumsi arus yang dihasilkan pada saat mendistribusikan pakan jenis pellet ini sebesar 452,84 mA.



Gambar 4.39 konsumsi arus pada pakan ayam jenis *pellet*.

c. Jenis Pakan *Mesh*

Konsumsi arus yang dihasilkan pada pakan ayam jenis mesh ini sama dengan pakan jenis pellet yaitu 452,84 mA.



Gambar 4.40 Konsumsi arus pada pakan ayam jenis *mesh*.



Dari data yang diperoleh maka dapat disimpulkan bahwa konsumsi arus pada saat dinyalakan tanpa adanya beban motor DC yang berputar yaitu sebesar 115,42 mA. Pada saat motor DC menyala terjadi kenaikan konsumsi arus yaitu sebesar 362,07 mA. Ketika motor DC diberi beban pakan ayam kenaikan arus cukup bervariasi antara lain : pakan jenis *Crumble* sebesar 469,83 mA, pakan jenis pellet dan *mesh* arus yang dihasilkan sama yaitu 452,84 mA. Penyebab konsumsi arus pada pakan jenis *crumble* adalah daya angkut yang bawa oleh *screw* banyak sehingga menyebabkan arus yang dihasilkan juga besar serta daya pengisian tempat pakannya juga cepat.