

## BAB V

### PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan

#### 5.1 Pengujian Sensor DS18B20

##### a. Tujuan

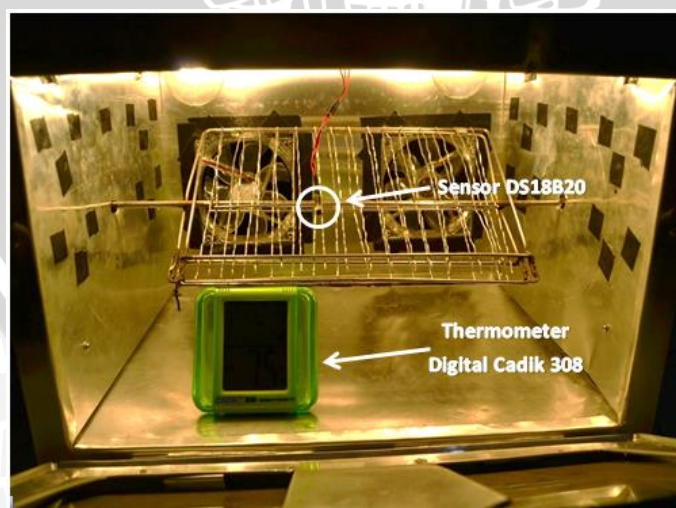
Mengetahui kemampuan pembacaan sensor DS18B20 terhadap perubahan suhu dengan melihat hasil pembacaan sensor yang ditampilkan..

##### b. Peralatan yang Digunakan

- Sensor DS18B20
- Lampu AC 100W
- Miniatur oven
- Thermometer Digital CADIK 308
- Komputer

##### c. Langkah Pengujian

1. Menyusun rangkaian seperti dalam Gambar 5.1



Gambar 5.1 Rangkaian Sensor DS18B20 dan Thermometer Cadik 308

2. Menghubungkan sensor DS18B20 dengan Arduino Mega 2560
3. Mengamati dan mencatat setiap perubahan suhu berdasarkan suhu thermometer acuan.

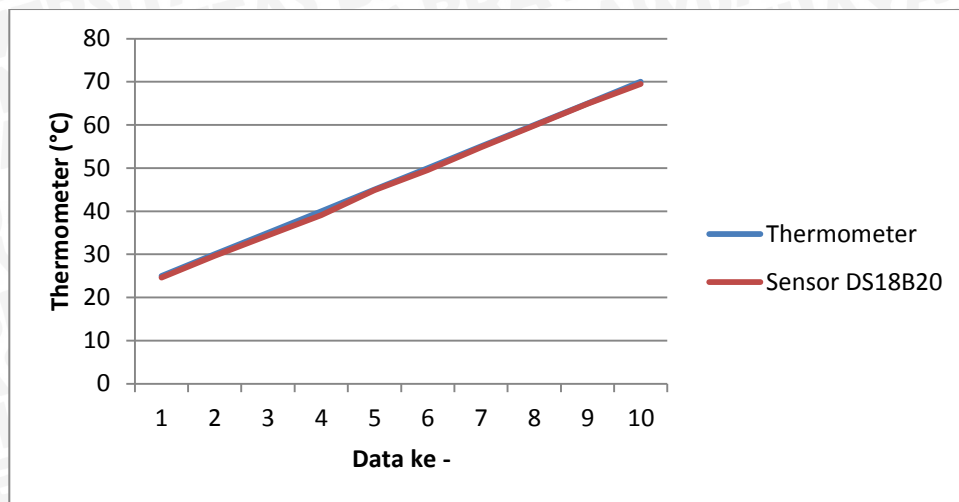
d. Hasil Pengujian dan Analisis

Hasil pengujian sensor DS18B20 ditunjukkan dalam Tabel 5.1. Didapatkan grafik linear antara suhu dengan thermometer seperti yang ditunjukkan dalam Gambar 5.2

Tabel 5.1 Perbandingan Pembacaan Suhu Sensor dengan Thermometer

No.	Suhu Thermometer (°C)	Suhu Pembacaan Sensor DS18B20	ERROR
1	25	24,6	0,4
2	30	29,65	0,35
3	35	34,42	0,58
4	40	39,10	0,90
5	45	44,85	0,15
6	50	49,54	0,46
7	55	54,79	0,21
8	60	59,83	0,17
9	65	64,91	0,09
10	70	69,49	0,51
<b>Error</b>			0,38

Berdasarkan hasil pengujian dalam Tabel 5.1, *error* antara suhu pembacaan thermometer dengan suhu pembacaan dari sensor DS18B20 adalah sekitar 0,38° C. Grafik perbandingan suhu hasil pembacaan thermometer acuan dengan suhu pembacaan dari sensor DS18B20 dapat dilihat dalam Gambar 5.2. Pada grafik tersebut terlihat bahwa selisih yang timbul antara kedua pembacaan suhu cenderung membentuk garis linear.



Gambar 5.2 Grafik Hubungan Thermometer dengan pembacaan sensor DS18B20

## 5.2 Pengujian Sinyal Kontrol Motor DC Servo

### a. Tujuan

Pengujian sinyal kontrol motor DC *servo* ini bertujuan untuk melihat bagaimana bentuk sinyal saat berada pada posisi sudut yang telah ditentukan serta melihat tegangan yang dikeluarkan untuk setiap perubahan sudut motor DC *servo*.

### b. Peralatan yang digunakan

- Catu Daya 5V DC.
- Arduino mega 2560
- Komputer
- Konektor PC scope
- *Hardware* dan *software* PC Lab 2000SE.
- Program dan *software* Arduino.

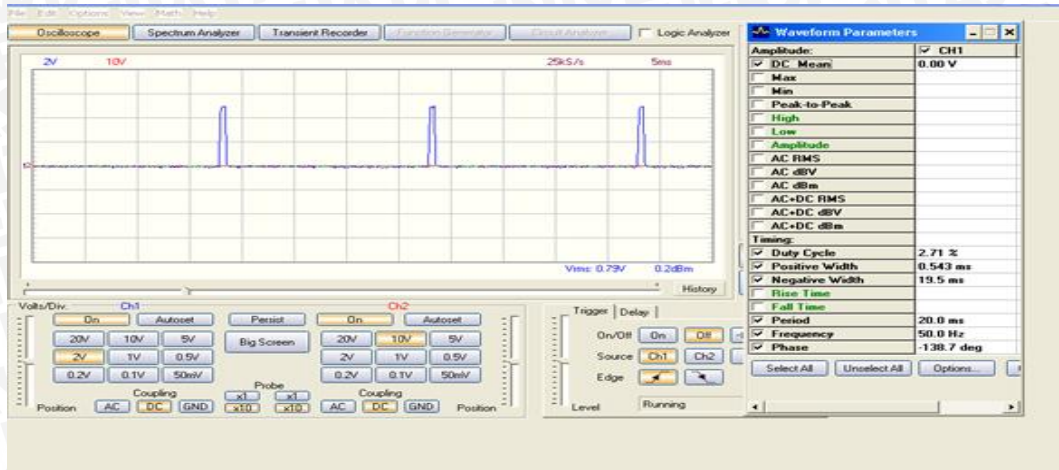
### c. Langkah Pengujian

1. Merangkai peralatan
2. Mengunduh program pengatur sudut *servo* dengan sudut 0, 60, 90, 160, 180 derajat pada *software* Arduino ERW 1.0.5.
3. Menjalankan *software* PC Lab 2000SE.
4. Memilih mode osiloskop lalu pilih *Run* untuk menjalankan osiloskop.

### d. Hasil Pengujian

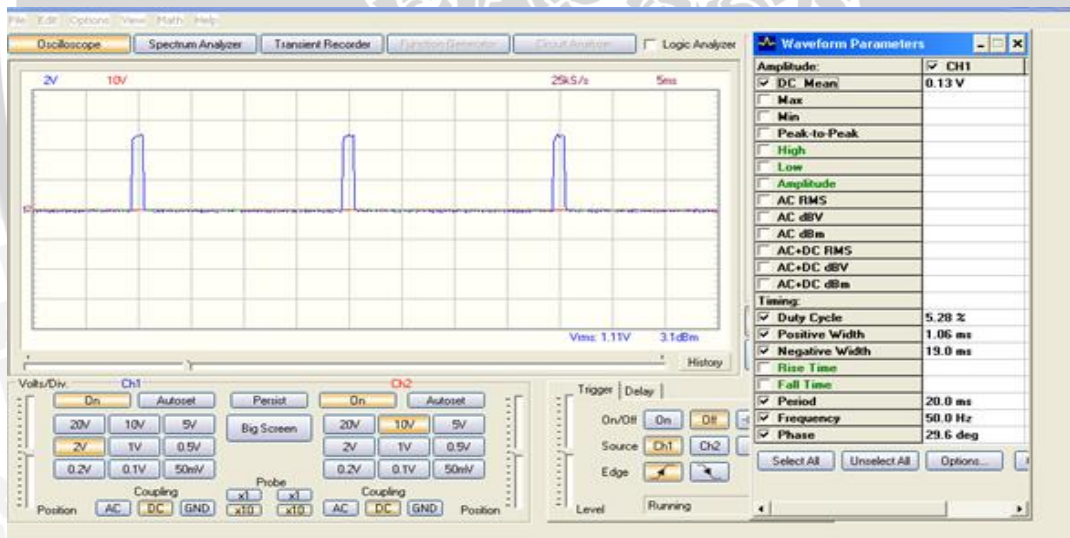
Setelah melakukan prosedur pengujian, didapatkan hasil sinyal dan tampilan

parameter. Parameter yang dihasilkan adalah amplitudo, *duty cycle*, dan lebar sinyal. Proses pengujian Sinyal Kontrol Motor DC *servo* sudut  $0^{\circ}$  ditunjukkan dalam Gambar 5.3.



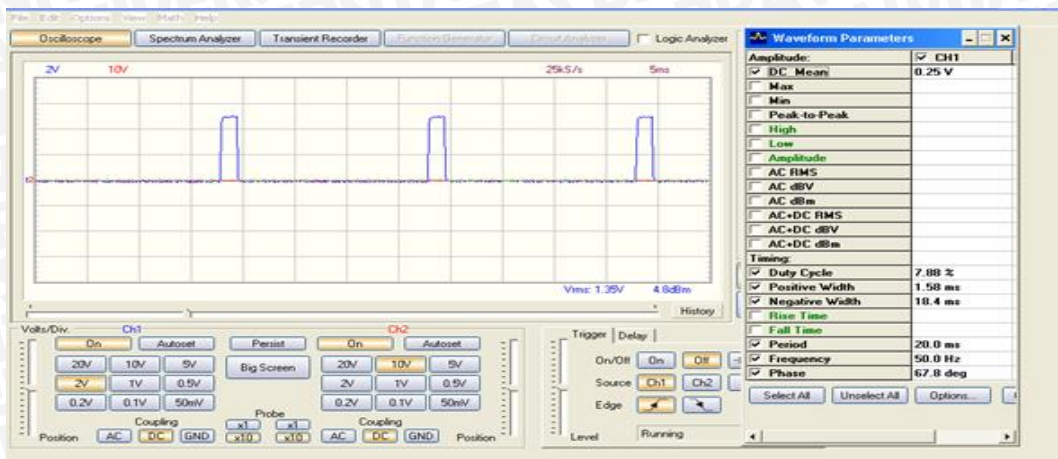
Gambar 5.3 Sinyal Kontrol dan Parameter Motor *Servo* Sudut  $0^{\circ}$

Sinyal Motor *servo* pada sudut  $60^{\circ}$  ditunjukkan dalam Gambar 5.4



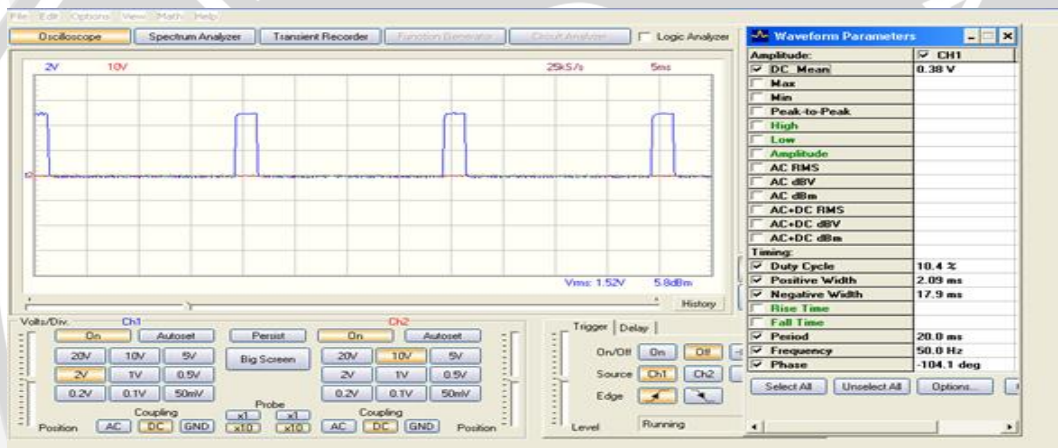
Gambar 5.4 Sinyal Kontrol dan Parameter Motor *Servo* Sudut  $60^{\circ}$

Sinyal motor DC *servo* pada sudut  $90^\circ$  ditunjukkan dalam Gambar 5.5



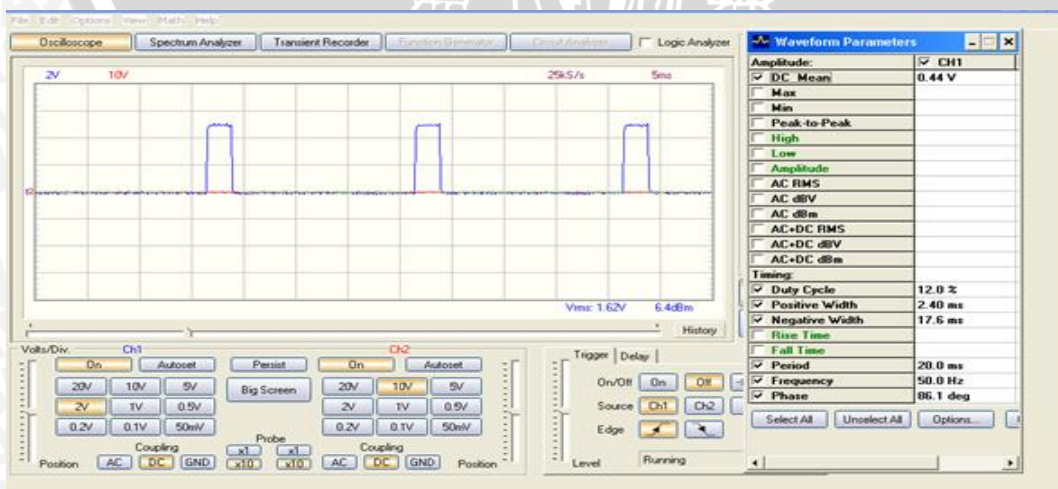
Gambar 5.5 Sinyal Kontrol dan Parameter Motor *Servo* Sudut  $90^\circ$

Sinyal motor DC *servo* pada sudut  $160^\circ$  ditunjukkan dalam Gambar 5.6



Gambar 5.6 Sinyal Kontrol dan Parameter Motor *Servo* Sudut  $160^\circ$

Sinyal motor DC *servo* pada sudut  $180^\circ$  ditunjukkan dalam Gambar 5.7



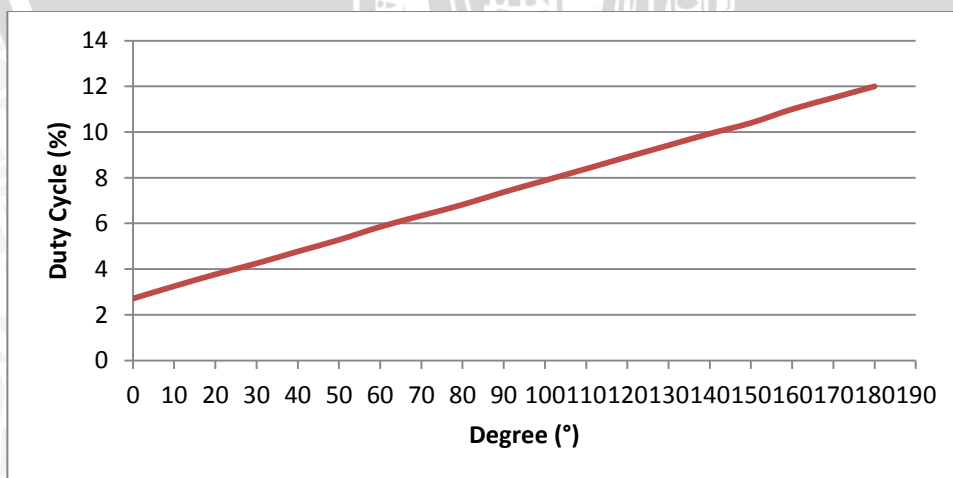
Gambar 5.7 Sinyal Kontrol dan Parameter Motor *Servo* Sudut  $180^\circ$

Hasil pengujian *duty cycle* ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hubungan Sudut dengan *Duty Cycle*

Sudut ( $^{\circ}$ )	Duty Cycle (%)
0	2,71
10	3,25
20	3,77
30	4,25
40	4,77
50	4,98
60	5,29
70	6,24
80	6,82
90	7,89
100	8,23
110	8,42
120	8,91
130	9,25
140	9,88
150	10,01
160	10,4
170	11,5
180	12

Grafik *duty cycle* motor DC servo ditunjukkan dalam Gambar 5.8.



Gambar 5.8 Grafik Hubungan Sudut dengan *Duty Cycle*

### 5.3 Pengujian Motor DC Servo terhadap Suhu

#### a. Tujuan

Pengujian ini dilakukan untuk mengetahui pengaruh perubahan suhu terhadap perubahan pergerakan motor DC servo.

#### b. Peralatan yang Digunakan

- Motor DC Servo
- Miniatur Oven
- Thermometer Digital CADIK 308
- Program dan software arduino compiler 1.5.7

#### c. Langkah Pengujian

1. Mengkondisikan miniatur oven
2. Memprogram pergerakan motor DC servo melalui arduino mulai dari sudut  $0^{\circ}$ .
3. Motor DC servo memutar sudut sesuai program yang diinginkan, putaran sudut memicu cahaya lampu untuk mengeringkan, setelah itu amati perubahan kenaikan suhu yang tertera pada thermometer.
4. Ulangi seperti langkah diatas hingga sudut  $160^{\circ}$ .
5. Mencatat kenaikan suhu dan membuat hubungan antara keduanya.

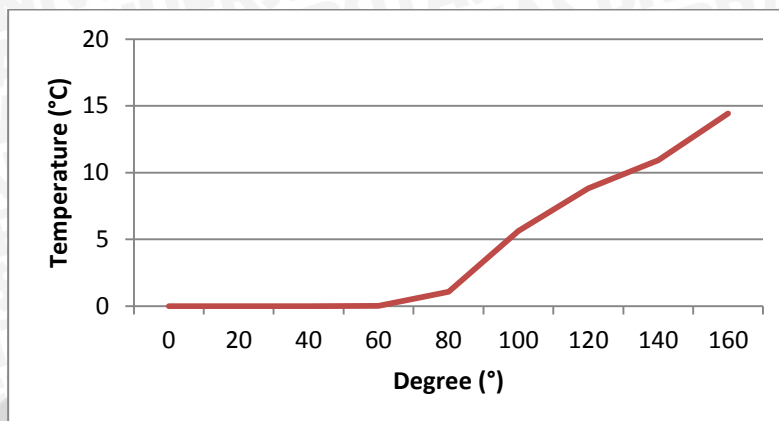
#### d. Hasil Pengujian

Setelah dilakukan sesuai dengan cara diatas didapatkan hasil seperti pada Tabel 5.3.

Tabel 5.3 Hubungan Sudut Servo dengan Kenaikan Suhu

Sudut Servo	Kenaikan Suhu dalam 1 Menit
0	0
20	0
40	0
60	0.03
80	1.09
100	5.63
120	8.83
140	10.92
160	14.43

Grafik hubungan antara derajat motor *servo* dengan kenaikan suhu digambarkan dalam Gambar 5.9.



Gambar 5.9 Grafik Hubungan Derajat Motor *Servo* dengan Suhu.

Dari hasil pengujian yang dilakukan, dalam grafik dapat dilihat bahwa semakin besar perubahan derajat pergerakan *servo* yang diberikan, maka kenaikan suhu juga semakin besar dikarenakan pemasangan mekanik membutuhkan putaran *servo* yang berbanding lurus.

#### 5.4 Pengujian *Driver* L298N

##### a. Tujuan

Pengujian ini bertujuan untuk mengetahui kinerja dan respon dari rangkaian *driver* pengendali motor *driver* L298N dengan membandingkan dan menguji tegangan keluaran Arduino Mega dengan tegangan keluaran *driver* L298N.

##### b. Peralatan yang digunakan

1. *Driver* L298N
2. Multimeter
3. Arduino Mega 2560

##### c. Langkah pengujian

1. Menghubungkan masukan *driver* L298N ke Arduino Mega.
2. Menghubungkan keluaran *driver* L298N ke *Digital Multimeter*
3. Arduino Mega memberikan instruksi sinyal masukan pada pin PWM *driver* L298N dengan sinyal PWM 0-255
4. Pengujian terhadap tegangan keluaran Arduino Mega dan *driver* L298N dilakukan secara bergantian



5. Mencatat hasil perbandingan antara tegangan keluaran Arduino Mega dan *driver* L298N

d. Hasil pengujian

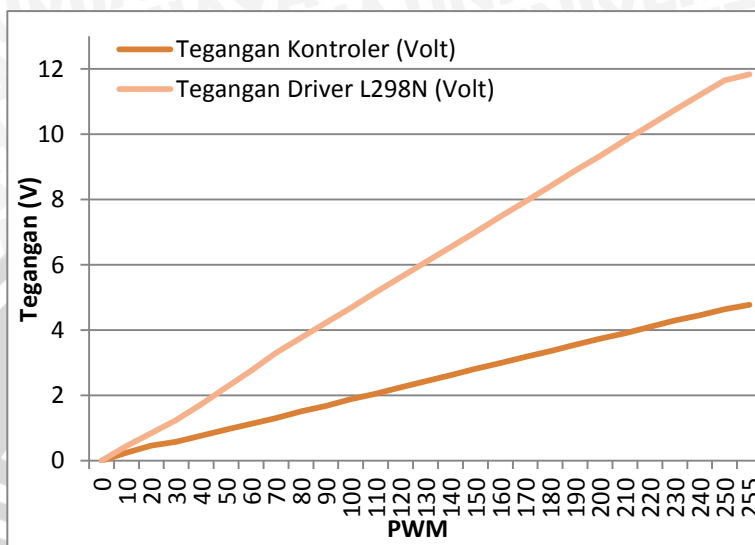
Besar perbandingan antara tegangan keluaran kontroler dengan tegangan keluaran *driver* L298N dapat dilihat dalam Tabel 5.4.

Tabel 5.4 Hasil Pengujian *Driver* L298N

No	PWM	Tegangan Kontroler (Volt)	Tegangan <i>Driver</i> L298N (Volt)
1	0	0	0
2	10	0.24	0.45
3	20	0.46	0.84
4	30	0.58	1.24
5	40	0.76	1.72
6	50	0.95	2.24
7	60	1.13	2.75
8	70	1.30	3.30
9	80	1.51	3.76
10	90	1.67	4.22
11	100	1.88	4.67
12	110	2.05	5.16
13	120	2.24	5.62
14	130	2.43	6.08
15	140	2.62	6.54
16	150	2.81	7.00
17	160	2.99	7.48
18	170	3.17	7.93
19	180	3.35	8.40
20	190	3.55	8.88
21	200	3.73	9.33
22	210	3.90	9.81
23	220	4.10	10.27
24	230	4.29	10.74
25	240	4.45	11.19
26	250	4.64	11.65
27	255	4.77	11.83

Berdasarkan Tabel 5.4 pengujian ini menunjukkan bahwa keluaran tegangan yang dihasilkan oleh rangkaian *driver* motor L298N sesuai dengan yang dibutuhkan oleh kipas DC. Sehingga dapat disimpulkan *driver* motor L298N dapat bekerja dengan baik.

Grafik hubungan antara tegangan keluaran Arduino Mega dan *driver* L298N dapat dilihat dalam Gambar 5.10.



Gambar 5.10 Grafik Perbandingan tegangan keluaran Arduino Mega dan *driver* L298N

## 5.5 Pengujian Keseluruhan Sistem

### a. Tujuan

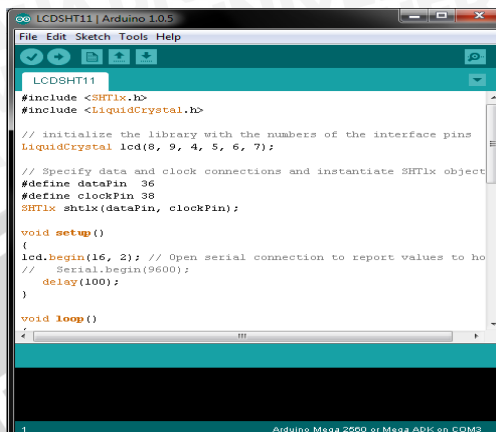
Pengujian ini bertujuan untuk mengetahui bagaimana kinerja sistem secara keseluruhan dan mengamati respons kontroler yang terbaik terhadap *setpoint*.

### b. Peralatan yang Digunakan

- Miniatur *oven*
- Motor DC *servo*
- 2 Lampu AC 100W
- *Driver* L298N
- Arduino mega 2560
- Kipas DC 12V
- Catu daya 12V
- Sensor DS18B20
- LCD *Keypad shield*
- Komputer
- Program dan *software* arduino mega 2560

### c. Langkah Pengujian

1. Membuat program bahasa C berdasarkan *library* pada *Arduino Compiler 1.5.7* untuk di masukkan ke dalam ATmega 2560 pada *Arduino Mega* ditunjukkan dalam Gambar 5.11.



```

LCDSHT11
File Edit Sketch Tools Help
#include <SHT1x.h>
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

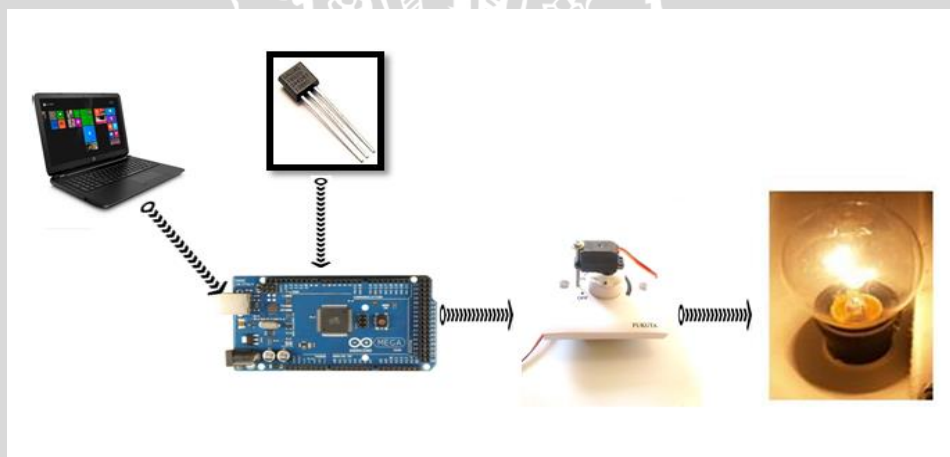
// Specify data and clock connections and instantiate SHT1x object
#define dataPin 36
#define clockPin 38
SHT1x sht1x(dataPin, clockPin);

void setup()
{
  lcd.begin(16, 2); // Open serial connection to report values to host
  // Serial.begin(9600);
  delay(100);
}

void loop()
  
```

Gambar 5.11 Program Bahasa C pada *Arduino Compiler 1.5.7*

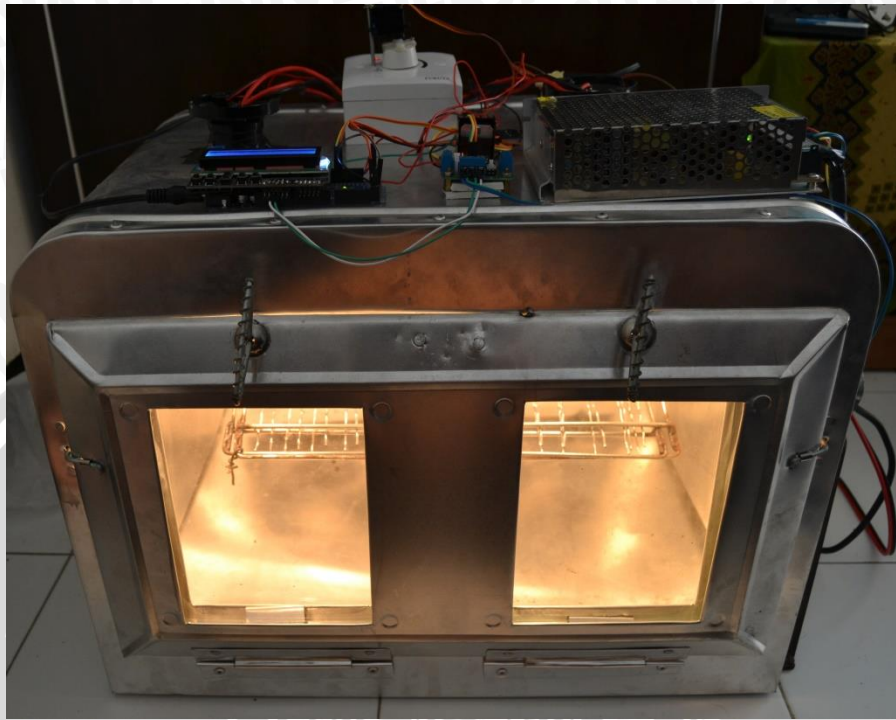
2. Menyusun rangkaian *Arduino mega* seperti dalam Gambar 5,12.



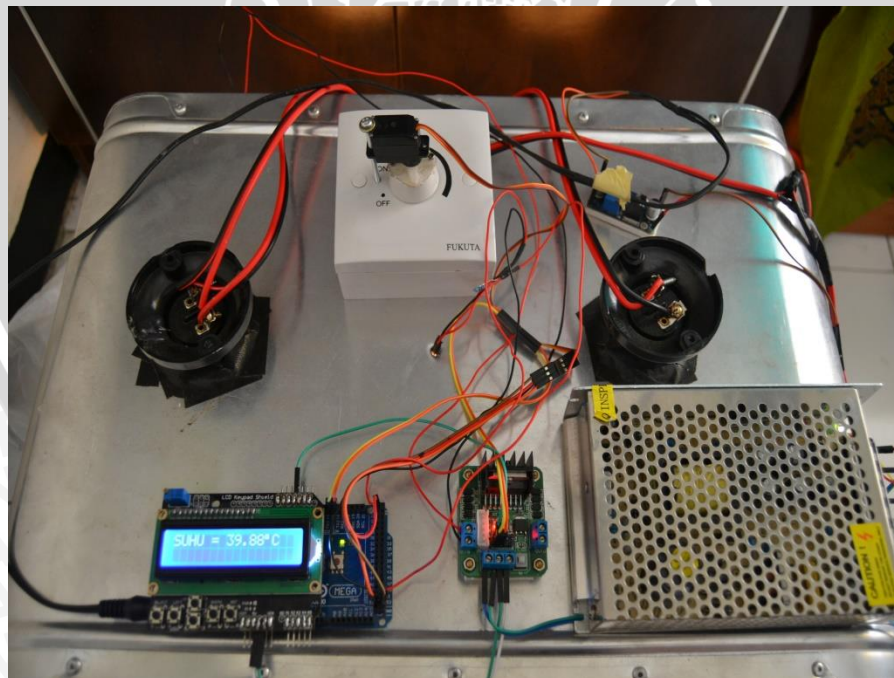
Gambar 5.12 Rangkaian secara keseluruhan sistem

3. Menghubungkan *power supply* sebagai sumber catu daya untuk *Arduino mega* dan aktuator.
4. Setelah membuat program bahasa C pada *Arduino Compiler 1.5.7*, kemudian di *Verify* untuk memeriksa kembali program apakah sudah benar atau masih mempunyai *error*.
5. Sebelum menjalankan program dalam *Arduino Compiler 1.5.7*, sensor suhu DS18B20 yang akan menjadi *input* pada *Arduino Mega* dihubungkan melalui port digital *Arduino Mega*.

6. Menghubungkan keluaran PWM dari port 01 dan 03 dengan *driver* motor L298N. Dan kemudian *upload* bahasa C pada *Arduino Compiler 1.5.7*.



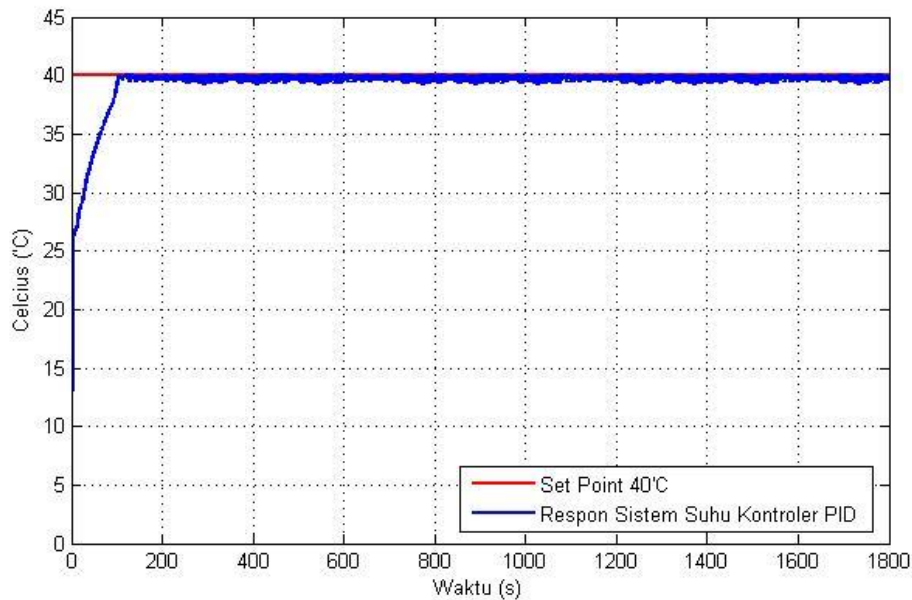
Gambar 5.13 Pengujian Keseluruhan Sistem



Gambar 5.14 Rangkaian Keseluruhan Sistem

#### d. Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil respon yang terbaik dengan *settling time* paling kecil dan waktu untuk mencapai *steady state* paling cepat yang dapat dilihat dalam Gambar 5.15.



Gambar 5.15 Grafik Respon Sistem Suhu Kontroler PID dengan *Set Point* 40°C

Pengujian sistem secara keseluruhan ini dilakukan 30 menit hingga 4 jam pada suhu 40°C, untuk mengetahui seberapa besar nilai kontroler yang dibutuhkan agar sistem bekerja sesuai dengan *set point* 40° C dengan *settling time* 105 s.

## 5.6 Pengujian Keseluruhan Sistem dengan Gangguan

### a. Tujuan

Pengujian ini bertujuan untuk mengetahui bagaimana kinerja sistem secara keseluruhan dan mengamati respon sistem terhadap *setpoint* ketika mendapatkan gangguan berupa perubahan aliran suhu yang menurun.

### b. Peralatan yang Digunakan

- Miniatur oven
- Motor DC servo
- 2 Lampu AC 100W
- Driver L298N
- Arduino mega 2560
- Kipas DC 12V

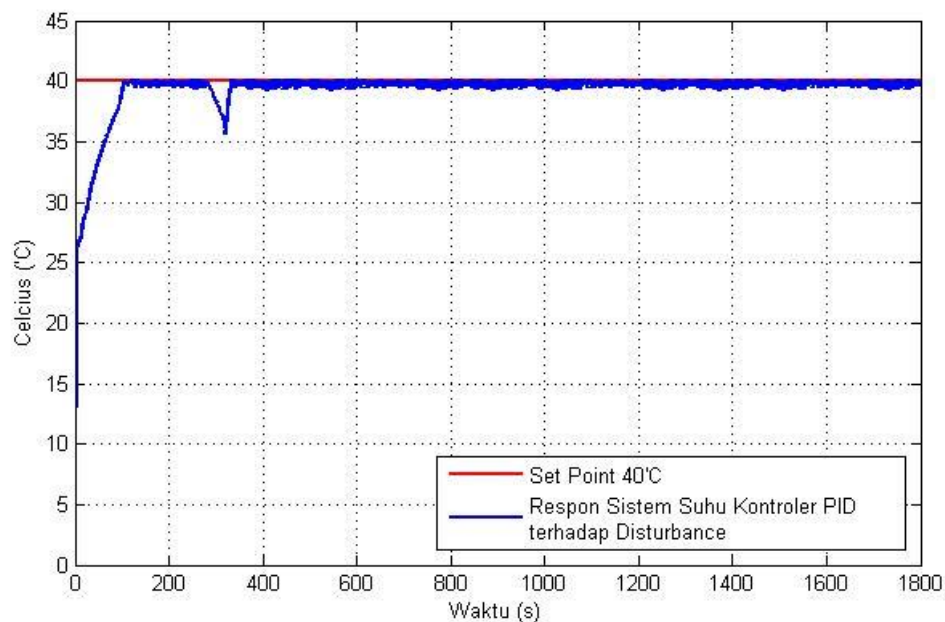
- Catu daya
- Sensor Suhu DS18B20
- LCD Keypad shield
- Komputer
- Program dan *software* Arduino mega 2560

c. Langkah Pengujian

1. Menggunakan kontrol PID
2. Memberikan gangguan mengalirkan udara luar ketika membuka tutup *oven*
3. Melihat hasil keluaran nilai suhu dan menjadikan grafik terhadap waktu.

d. Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil respon setelah diberi *disturbance* (gangguan) yang dapat dilihat dalam Gambar 5.16.



Gambar 5.16 Grafik Respon Sistem Suhu Kontroler PID terhadap *Disturbance*

Dari grafik pada Gambar 5.16 yang berupa hasil pengujian dapat dilihat respon sistem terhadap *disturbance* berupa perubahan suhu yang menurun. Sistem dapat kembali pada keadaan *steady state* setelah terjadinya *disturbance* dan mengalami proses *recovery* (pemulihan) dalam waktu sebesar 55 detik (s). Oleh karena itu dapat disimpulkan perancangan sistem pengendalian pada perancangan alat ini telah bekerja dengan baik