

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Dalam bab ini akan dibahas mengenai perancangan dan pembuatan sistem pengendali suhu dan kelembaban pada proses ekstraksi metode maserasi minyak atsiri bunga melati. Pada dasarnya perancangan alat meliputi perancangan perangkat keras dan perancangan perangkat lunak.

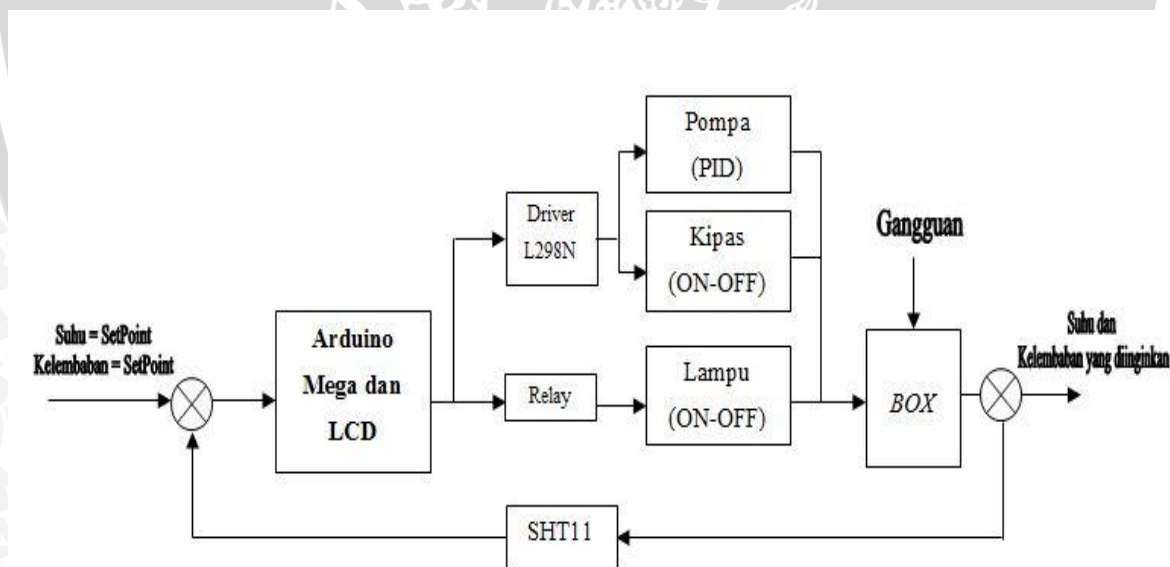
4.1 Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri atas:

1. Perancangan perangkat keras meliputi rangkaian *driver* motor DC L298N, dan Relay.
2. Pembuatan dan penyelesaian bahasa C++ pada *software Arduino Compiler* untuk Arduino Mega 2560.

4.2 Diagram Blok Sistem

Diagram blok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Balok Sistem Pengendalian Suhu dan Kelembaban (Perancangan)

Keterangan diagram blok pada Gambar 4.1 adalah sebagai berikut:

1. *Set point* sistem adalah suhu 27° Celcius dan Kelembaban 45% RH.

2. Pusat pengendalian sistem adalah Arduino Mega 2560 yang memberikan sinyal PWM (*Pulse Width Modulation*) pada *driver* motor DC L289N dan Motor DC Pompa.
3. Pompa DC, Lampu dan kipas ketiganya merupakan yang berfungsi sebagai aktuator
4. Sensor SHT11 digunakan sebagai *feedback* sistem dengan rentang tegangan masukan kontroler.
5. Keluaran dari sistem pengendalian suhu dan kelembaban merupakan masukan pada proses utama dalam ruang ekstraksi minyak atsiri.

4.3 Spesifikasi Alat

Spesifikasi sistem pengendalian suhu dan kelembaban pada ruang ekstraksi adalah sebagai berikut :

1. Rentang suhu yang dapat diproses adalah 0-100° Celcius.
2. Miniatur ruangan ekstraksi berupa sebuah *box* berbahan *plastic* dengan dilapisi *styrofoam* dan *aluminium foil* dengan lebar 28 cm , Panjang 35 cm dan tinggi 30 cm.
3. Bunga melati yang digunakan untuk ekstraksi adalah melati (*Jasminum sambac*) yaitu bagian bunganya saja daun dan batang telah dipisahkan.
4. Proses ekstraksi dilakukan dengan menggunakan 0,5 - 1 Kg bunga melati.
5. Sensor yang digunakan adalah sensor suhu dan kelembaban SHT11 pada rentang suhu -40-123,8°C dan rentang kelembaban 0-100 % RH.
6. Pengaturan suhu dan kelembaban miniatur ruangan ditentukan dengan motor DC dan motor DC pompa yang diatur kecepatan melalui *driver* L298N oleh Arduino mega 2560.
7. Pusat pengendali sistem adalah Arduino Mega 2560.

4.4 Prinsip Kerja Sistem

Cara kerja sistem adalah sebagai berikut :

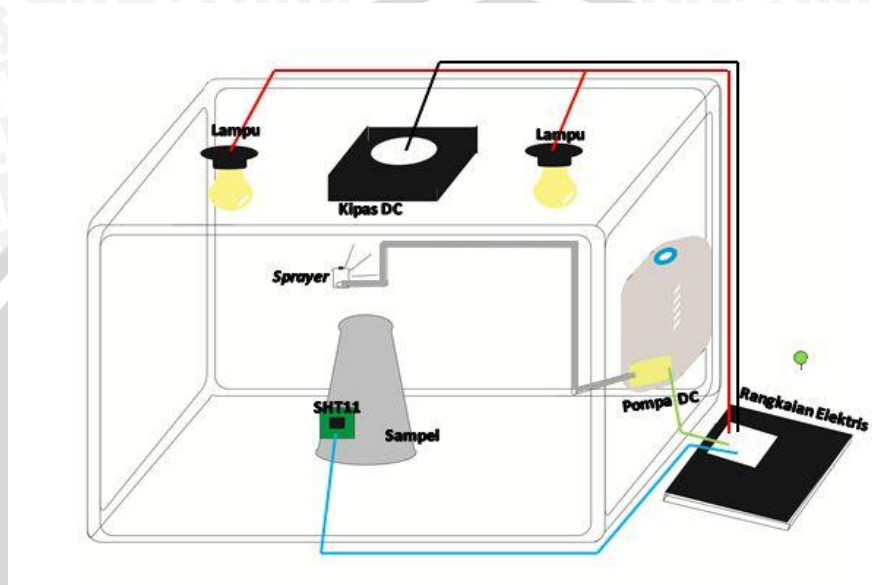
1. Catu daya 12 Volt digunakan sebagai catu motor DC pada pompa air.
2. *Port* komunikasi antara Arduino Mega dengan *Notebook* menggunakan perantara kabel USB dengan kecepatan transfer data sebesar 9600 bps.

3. Menggunakan sensor SHT11, sinyal-sinyal keluaran dari sensor yang sudah berupa sinyal digital, sesuai dengan sinyal yang dibutuhkan untuk dapat dibaca oleh Arduino Mega.
4. Sinyal digital yang masuk ke Arduino Mega kemudian diproses menjadi sinyal kontrol berupa PWM .
5. Sinyal kontrol dari Arduino Mega masuk ke *driver* L298N dan Relay. Driver L298N berfungsi menguatkan sinyal yang dihasilkan Arduino Mega dari 0-5 Volt menjadi 0-12 Volt sedangkan relay berfungsi sebagai saklar otomatis untuk menghidupi lampu, ketika dialiri tegangan 5V keluaran dari Arduino Mega.
6. Motor DC pada pompa air yang terhubung pada tangki berisi air akan berputar ketika pembacaan sensor menunjukkan bahwa keadaan miniatur *box* terlalu kering dari *setpoint* yaitu 45%RH.
7. Motor DC pada kipas yang terhubung pada tangki berisi air akan berputar ketika pembacaan sensor menunjukkan bahwa keadaan miniatur *box* terlalu panas dari *setpoint* yaitu 27°C.
8. Lampu akan menyala ketika pembacaan sensor menunjukkan bahwa keadaan miniatur *box* terlalu lembab dan terlalu dingin dari *setpoint* suhu 27°C dan kelembaban 45%RH
9. Dan keluaran sistem pengendalian suhu dan kelembaban dengan sensor SHT11 merupakan masukan bagi proses utama dalam ekstraksi metode maserasi minyak atsiri melati.

Spesifikasi sistem pengendalian suhu dan kelembaban pada alat miniatur *box* berbasis arduino Mega adalah sebagai berikut:

1. Rentang suhu yang dapat diproses adalah 26-30°C sedangkan rentang kelembaban yang dapat diproses adalah 20-60%RH.
2. Miniatur ruangan ekstraksi berupa sebuah *box* berbahan *plastic* dengan dilapisi *styrofoam* dan *aluminium foil* dengan lebar 28 cm , Panjang 35 cm dan tinggi 30 cm.
3. Bunga melati yang digunakan untuk ekstraksi adalah melati (*Jasminum sambac*) yaitu bagian bunganya saja daun dan batang telah dipisahkan.
4. Proses ekstraksi dilakukan dengan menggunakan 0,5 - 1 Kg bunga melati.
5. Sensor yang digunakan adalah sensor suhu dan kelembaban SHT11 pada rentang suhu -40-123,8° Celcius dan rentang kelembaban 0-100 % RH.

6. Pengaturan suhu dan kelembaban miniatur ruangan ditentukan dengan motor pompa, kipas DC dan Lampu yang diatur kecepatan melalui *driver* L298N dan mengatur sebagai saklar otomatis oleh relay dengan keluaran dari Arduino mega 2560.
 7. Pusat pengendali sistem adalah Arduino Mega 2560.
- Skema perancangan sistem dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Skema Keseluruhan Sistem (Perancang)

4.5 Perancangan Perangkat Keras (*Hardware*)

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan LCD Keypad shield, rangkaian catu daya, rangkaian *driver* motor beserta pompa air dan kipas DC, dan rangkaian relay lampu. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

4.5.1 Sensor SHT11

Sensor SHT11 pada perancangan ini berfungsi sebagai sensor pengendalian suhu dan kelembaban pada miniatur *box*. Keluaran dari sensor SHT11 ini digunakan sebagai masukan untuk perhitungan algoritma PID yang dirancang. Sensor SHT11 mampu melakukan pembacaan pada rentang suhu $-40-123,8^{\circ}\text{C}$ dan rentang kelembaban $0-100\%$ RH.

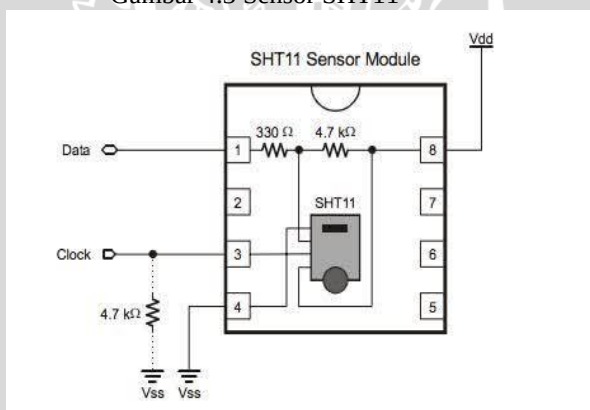
Sensor SHT11 adalah sebuah *single* chip sensor suhu dan kelembaban relatif dengan multi modul sensor yang outputnya telah dikalibrasi secara digital. Dibagian dalamnya terdapat kapasitas polimer sebagai elemen untuk sensor kelembaban relatif dan sebuah pita regangan yang digunakan sebagai sensor temperatur. *Output* kedua sensor

digabungkan dan dihubungkan pada ADC 14 bit dan sebuah *interface serial* pada satu *chip* yang sama.

Sensor SHT 11 diakses dengan metode *Two Wire Interface* yang artinya SHT11 membutuhkan dua jalur untuk berkomunikasi. Kedua jalur komunikasi *2-Wire Interface* dari sensor suhu-kelembaban SHT11 tersebut adalah *DATA line* dan *CLOCK line*. Pada *datasheet* SHT11 disebutkan bahwa diperlukan sebuah resistor *pull-up* sebesar 4.7 k Ω pada pin *CLOCK* yang berguna menaikkan level sinyal logika tinggi. Selama transmisi, garis *DATA* harus tetap stabil saat *CLOCK* pada kondisi *high*. Untuk menghindari kekacauan sinyal.



Gambar 4.3 Sensor SHT11



Gambar 4.4 Skema Modul Sensor SHT11

Tabel 4.1 Fungsi Masing-Masing Pin pada Sensor SHT11

Pin	Fungsi
1 (Data)	Digunakan untuk mengirim keluaran sensor menuju Arduino Mega
3 (Clock)	Digunakan untuk menghasilkan pulsa (clock)
4 (GND)	Digunakan untuk potensial 0 (netral)
8 (VDD)	Digunakan untuk potensial 5V

4.5.2 LCD Keypad Shield

Shield adalah istilah untuk modul tambahan pada Arduino, karena cara menggunakan modul tambahan pada Arduino adalah dengan cara menumpuk di bagian

atas Arduino, maka diberi istilah *shield* (tameng/perisai). Arduino *shield* yang digunakan pada penelitian ini berisi LCD.

LCD *Keypad Shield* dikembangkan untuk modul Arduino yang kompatibel, untuk menyediakan antarmuka *user-friendly* yang memungkinkan pengguna untuk pergi melalui menu, membuat pilihan dan lain-lain, terdiri dari 16x2 karakter putih LCD dengan latar belakang biru dan sebuah resistor variabel untuk mengatur kontras tampilan LCD. Tombol terdiri dari 5 tombol - reset, atas, kanan, bawah dan kiri. Untuk menyimpan pin I/O digital, antarmuka *keypad* hanya menggunakan satu saluran ADC. Nilai tegangan dari setiap *keypad* pada saat membaca adalah 5 V. Gambar LCD *Keypad Shield* dapat dilihat pada Gambar 4.5.



Gambar 4.5 LCD Keypad Shield

4.5.3 Driver L298N

Modul pengendali motor DC digunakan untuk mengendalikan putaran motor DC yang menjadi penggerak pompa dan kipas DC. *Driver* ini dihubungkan dengan Arduino Mega. *Driver* pengendali pada perancangan ini menggunakan *driver* L298N yaitu sebuah perangkat keras berupa modul yang berfungsi untuk menggerakkan berbagai macam motor DC. Gambar modul *driver* motor L298N ditunjukkan pada Gambar 4.6.



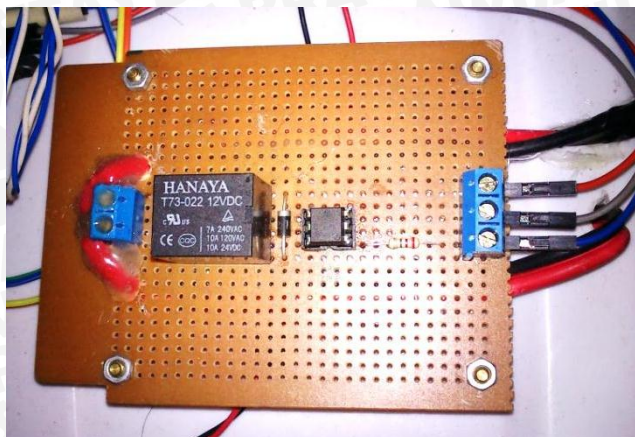
Gambar 4.6 Driver L298N

Tabel 4.2 Hasil Pengujian *Driver Motor*

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

4.5.4 *Driver Relay Lampu AC*

Modul relay digunakan untuk mengatur nyala dan matinya lampu AC. *Driver* ini dihubungkan dengan Arduino Mega. *Driver Relay* pada perancangan ini menggunakan *Relay SPDT (Single Pole Double Throw)* yaitu relay yang memiliki 5 terminal, 3 terminal untuk Saklar dan 2 terminalnya lagi untuk *Coil*. Relay berfungsi sebagai saklar otomatis yang bekerja pada saat dipicu tegangan keluaran dari Arduino Mega 5V dan tegangan sumber sebesar 12V. Gambar modul *driver Relay* ditunjukkan pada Gambar 4.7.



Gambar 4.7 Driver Relay

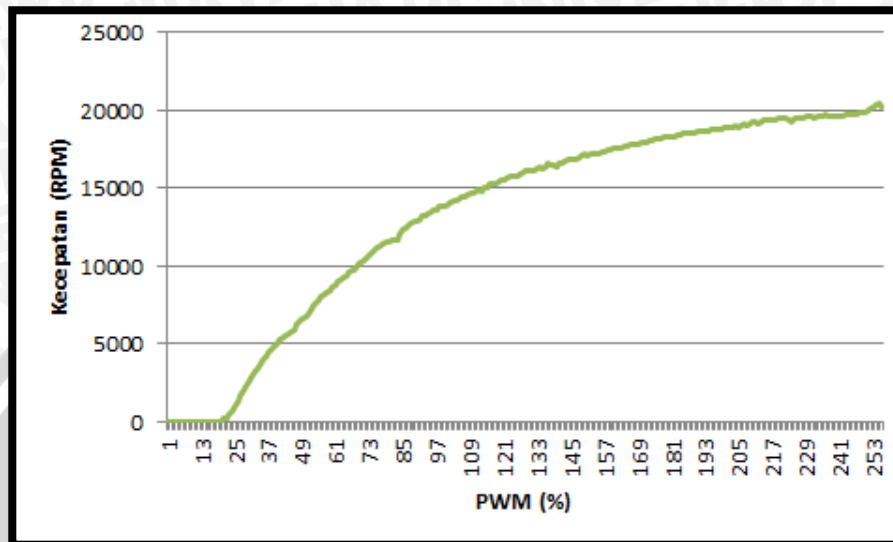
4.5.5 Motor DC

Pengendalian motor DC menggunakan rangkaian Arduino Mega sebagai pengolah dan memberikan data berupa *Pulse width Modulation* (PWM) agar motor dapat bergerak. Motor DC yang digunakan pada perancangan ini tidak diketahui karakteristiknya, sehingga yang perlu dilakukan adalah melakukan pengujian dengan menggunakan rangkaian *optocoupler*. Karakteristik motor DC pada perancangan ini didapatkan dengan cara memberi masukan unit *step*. Hasil kecepatan motor terhadap PWM ditunjukkan pada Gambar 4.8.

Tabel 4.3 Respon Kecepatan Motor Terhadap Sinyal PWM

No	PWM	Kecepatan (Rpm)
1	0	0
2	10	0
3	20	0
4	30	2880
5	40	5220
6	50	6900
7	60	8760
8	70	10320
9	80	11700
10	90	13020
11	100	13980
12	110	14760
13	120	15540
14	130	16140
15	140	16560
16	150	17100
17	160	17520
18	170	17880
19	180	18300
20	190	18600
21	200	18900
22	210	19260
23	220	19500

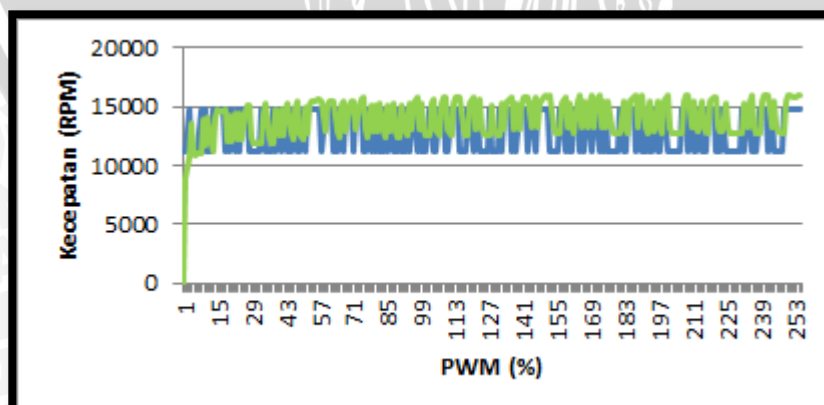
24	230	19620
25	240	19560
26	250	19980
27	255	20220



Gambar 4.8 Grafik Karakteristik *Steady State* pada Motor DC (Perancangan)

Dari hasil pengujian yang ditunjukkan dalam Tabel 4.3 dan Gambar 4.8 dapat disimpulkan bahwa semakin besar nilai PWM maka putaran motor semakin cepat pula.

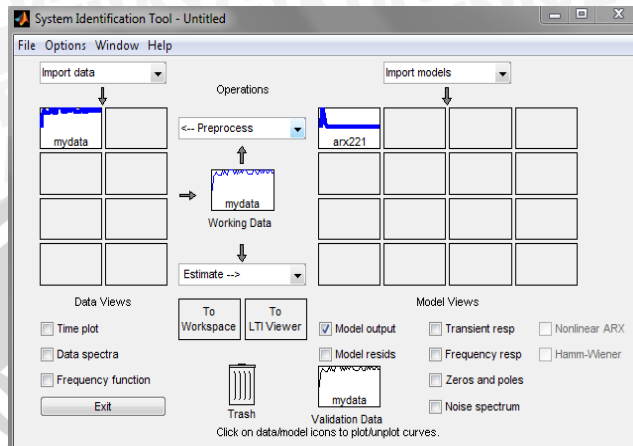
Setelah didapat nilai kecepatan motor terhadap sinyal PWM, selanjutnya dilakukan pengujian respon kecepatan motor terhadap sinyal *Pseudo Random Binary Sequences* (PRBS) dengan menggunakan nilai PWM *Low* = 76 dan nilai PWM *High* = 110 didapat respon kecepatan motor terhadap sinyal PRBS seperti yang ditunjukkan dalam Gambar 4.9.



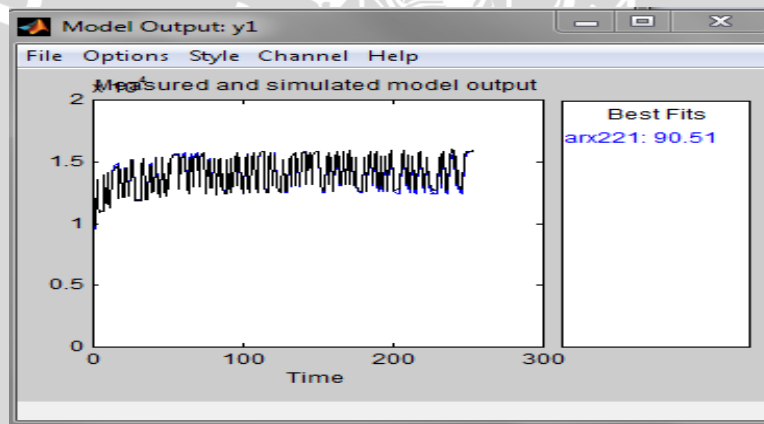
Gambar 4.9 Grafik Kecepatan Motor DC Terhadap Sinyal PRBS (Perancangan)

- Setelah didapatkan sinyal PRBS selanjutnya adalah melakukan identifikasi dengan menggunakan MATLAB 2010.

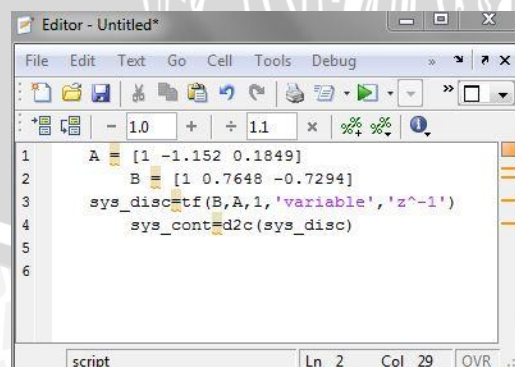
- Digunakan ident yaitu sistem identifikasi yang ada pada MATLAB 2010. Data sinyal PRBS yang tadi telah disimpan kemudian diimport pada blok *system identification tool* sistem model yang digunakan adalah *Auto Regresive with Exogenous Input* (ARX) dan estimasi parameter yang sesuai adalah 2 2 1 dengan *best fit* sebesar 90.51 % yang ditunjukkan pada Gambar 4.10



Gambar 4.10 . Identifikasi dengan ident pada MATLAB 2010 (Perancangan)



Gambar 4.11 Grafik Simulasi *Model Output* dengan *Best Fit* 90.51% (Perancangan)



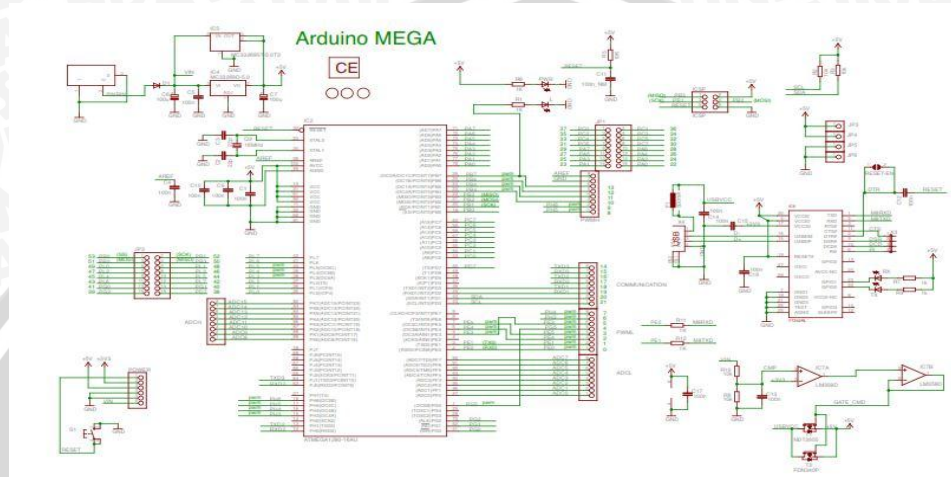
Gambar 4.12 Proses Mendapatkan Nilai Fungsi Alih (Perancangan)

- Dari hasil identifikasi, fungsi alih motor $\frac{Y(s)}{U(s)}$ yang didapat adalah

$$\frac{1.554 s + 0.07371}{s^2 + 1.688 s + 0.0685} \quad (4.1)$$

4.5.6 Konfigurasi I/O Arduino Mega

Pada alat ini digunakan Arduino Mega sebagai pusat pengolah utama dalam melakukan proses pengendalian. Konfigurasi I/O dari Arduino Mega ditunjukkan pada Gambar 4.13.



Gambar 4.13 Skematik Arduino Mega

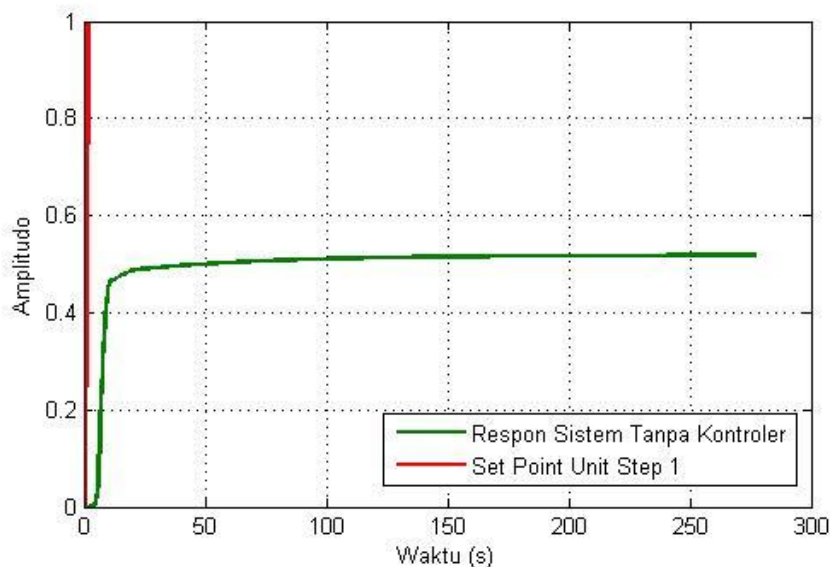
Tabel fungsi masing masing pin Arduino Mega dapat dilihat dalam Tabel 4.4.

Tabel 4.4 Fungsi Pin Arduino Mega

No.	Pin	Fungsi
1	P2	Digunakan sebagai <i>enable driver</i> motor A (Pompa DC)
2	P3	Digunakan sebagai <i>enable driver</i> motor B (Kipas DC)
3	D52	Digunakan sebagai antar muka data dari sensor SHT11
4	D53	Digunakan sebagai antar muka <i>clock</i> dari sensor SHT11
5	5V	Digunakan sebagai catu tegangan sensor SHT11
6	GND	Digunakan sebagai <i>ground</i> sensor SHT11
7	COM15	Digunakan sebagai antar muka <i>driver</i> motor 1
8	COM16	Digunakan sebagai antar muka <i>driver</i> motor 1
9	COM17	Digunakan sebagai antar muka <i>driver</i> motor 2
10	COM18	Digunakan sebagai antar muka <i>driver</i> motor 2
11	COM19	Digunakan sebagai antar muka <i>driver</i> relay lampu

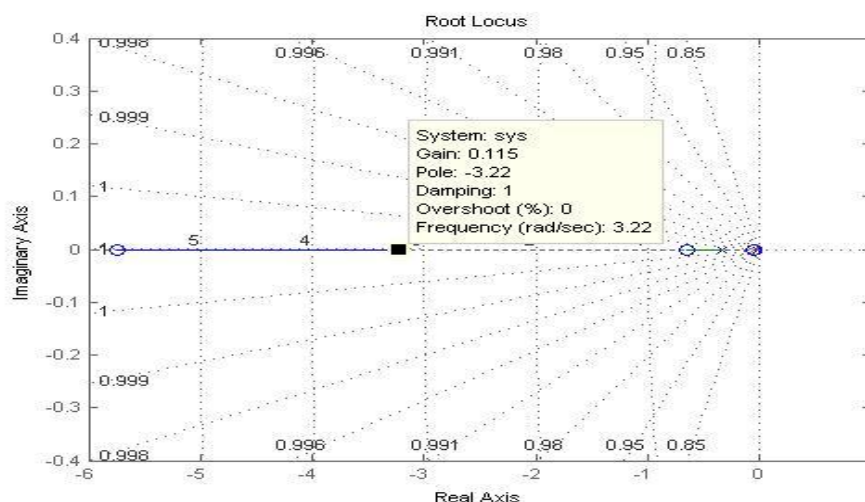
4.5.7 Perancangan Kontroler Proporsional Integral Differensial (PID)

Memenuhi tujuan performansi loop yang diinginkan, maka perlu ditambahkan kontroler pada sistem tersebut. Kontroler yang dipilih adalah kontroler Proporsional Integral Differensial (PID) dengan metode *root locus* setelah didapatkan fungsi alih *plant* yaitu $F(s) = \frac{1.554s + 0.07371}{s^2 + 1.688s + 0.0685}$. Respon sistem terhadap masukan fungsi unit *step* 1 dapat dilihat pada gambar 4.14.



Gambar 4.14 Grafik Simulasi Respon Sistem Tanpa Kontroler (Perancangan)

Dari respon yang didapat, sistem memiliki *error steady state* yang besar sehingga diperlukan kontroler untuk memperbaiki sistem. Berdasarkan fungsi alih *loop* tertutup dapat diketahui sistem berorde dua. Nilai parameter PID ditentukan oleh pemilihan *pole* pada diagram *root locus* sebagaimana pada sub bab 2.9. Pada penelitian ini digunakan $s_1 = -3.1975$. Penentuan letak *pole* pada diagram *root locus* terlihat dalam Gambar 4.15



Gambar 4.15 Letak Pole Pada Diagram Root Locus (Perancangan)

Setelah didapatkan letak *pole* yang diinginkan kemudian dengan mensubstitusikan nilai $s_1 = -3.1975$ dan nilai fungsi alih sistem pada persamaan 2.17 dan memvariasikan nilai K_i akan didapatkan parameter PID pada Tabel 4.5. Pencarian parameter K_p , K_i , dan K_d pada persamaan 2.17 dengan menggunakan MATLAB 2010 ditunjukkan pada listing program berikut :

%nilai pole yang ditentukan dari gambar root locus

```
s1=-3.1975
```

```
KI=[10 7 4 0.1 0.25]
```

```
plant_num=[0 1.554 0.07371];  
plant_den=[1 3.242 0.14221];
```

```
s1mag = abs(s1)  
beta = angle(s1)
```

```
plant_al = polyval(plant_num,s1)/polyval(plant_den,s1);
```

```
plants1mag = abs(plant_al)  
psi = angle(plant_al)  
t=0:1:20:300;
```

```
for k =1:5
```

```
KP = -sin(beta+psi)/(plants1mag*sin(beta))-2*KI(k)*cos(beta)/s1mag  
nilai_KI = KI(k)  
KD = sin(psi)/(s1mag*plants1mag*sin(beta))+KI(k)/s1mag^2
```

```
Gcnum = [KD KP KI(k)];  
Gcden = [0 1 0];
```

```
Tnum = conv(plant_num,Gcnum);  
Tden = conv(plant_den,Gcden)+conv(plant_num,Gcnum);
```

```
r = roots(Tden)
```

```
step (Tnum,Tden,t)  
hold on  
end
```

```
hold off  
figure, rlocus(Tnum,Tden)
```

Hasil pencarian parameter Kp, Ki dan Kd dari perhitungan pada program diatas ditunjukkan dalam Tabel 4.5.

Tabel 4.5 Parameter PID dengan $s_1 = -3.1975$

No.	Kp	Ki	Kd	Pole 1	Pole 2	Pole 3
1	6.5259	10	1.0205	-3.1975	-1.9618	-0.0454
2	4.6494	7	0.7270	-3.1975	-1.6976	-0.0446
3	2.7730	4	0.4336	-3.1975	-1.2902	-0.0427
4	0.3336	0.1	0.0522	-3.1975	-0.2768	-0.0077
5	0.4274	0.25	0.0668	-3.1975	-0.3298	-0.0158

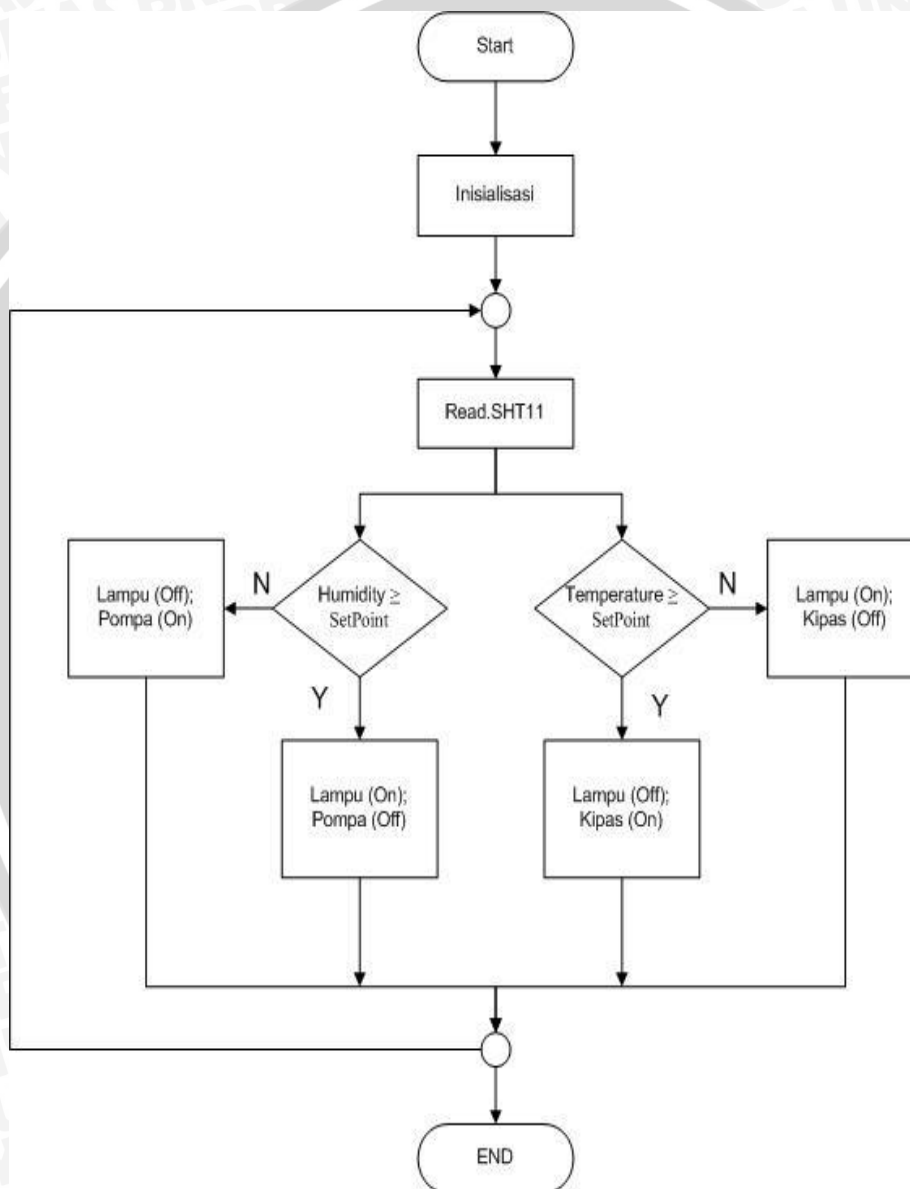
Setelah didapatkan nilai Kp, Ki dan Kd hasil perhitungan, kemudian dilakukan pengujian terhadap sistem, dan parameter yang sesuai dengan sistem.

4.6 Perancangan Perangkat Lunak (Software)

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat lunak meliputi perancangan bahasa C++ pada Arduino *Compiler* 1.5.7.

4.6.1 Flowchart Sistem Keseluruhan

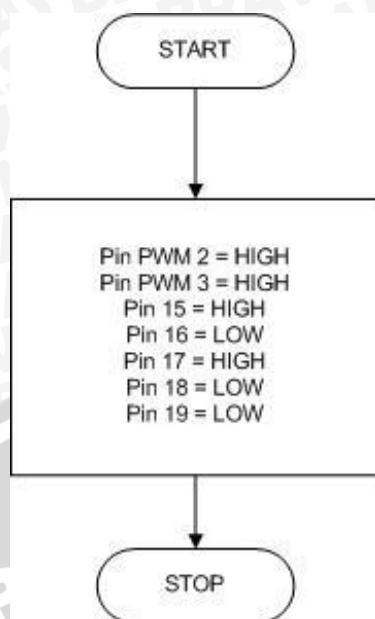
Flowchart keseluruhan sistem ditunjukkan pada Gambar 4.16.



Gambar 4.16 Flowchart Keseluruhan Sistem (Perancangan)

4.6.2 Flowchart Driver Enable

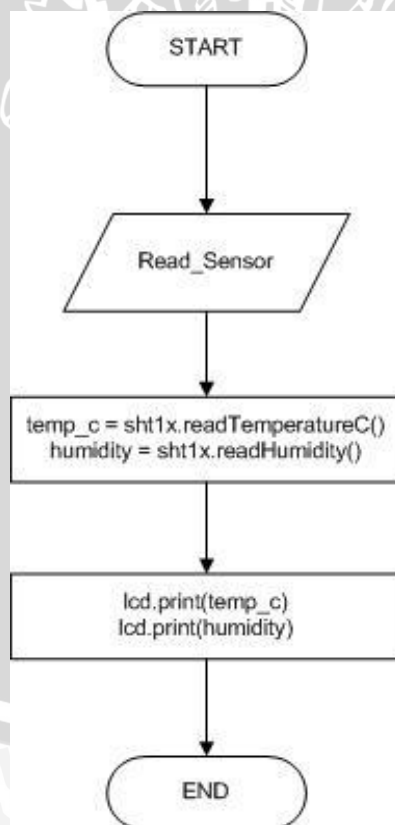
Flowchart driver enable ditunjukkan pada Gambar 4.17.



Gambar 4.17 Flowchart Driver Enable (Perancangan)

4.6.3 Flowchart Pembacaan Sensor SHT11

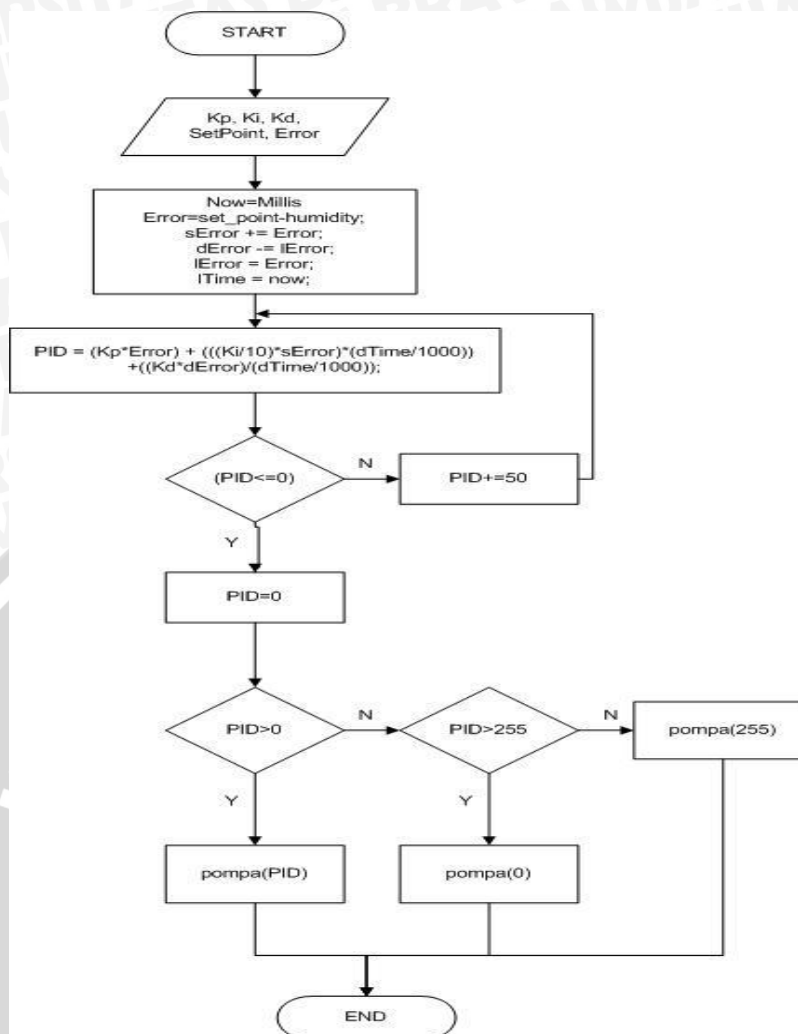
Flowchart pembacaan sensor SHT11 ditunjukkan pada Gambar 4.18.



Gambar 4.18 Flowchart Pembacaan Sensor SHT11 (Perancangan)

4.6.4 Flowchart Perhitungan Parameter PID

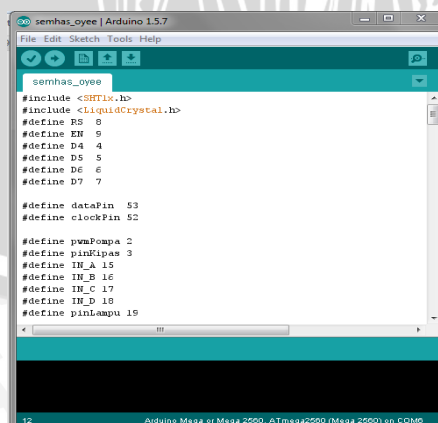
Flowchart perhitungan parameter PID ditunjukkan pada Gambar 4.19.



Gambar 4.19 Flowchart Perhitungan Parameter PID (Perancangan)

4.6.5 Membuat program C++ dengan Arduino Compiler

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 4.20.



Gambar 4.20 Program Bahasa C++ pada *Arduino Compiler* 1.5.7