

BAB III

METODE PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode yang digunakan dapat diuraikan pada skripsi ini sebagai berikut :

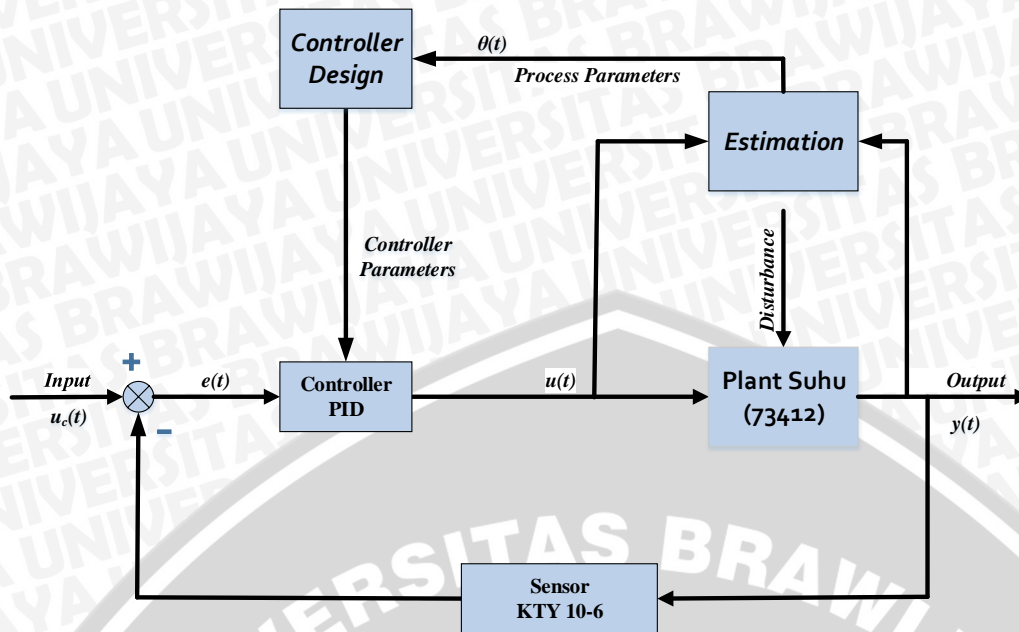
1. Perancangan blok diagram sistem
2. Spesifikasi desain
3. Karakterisasi setiap blok

Karakterisasi setiap blok dilakukan untuk mempermudah analisis sistem. Karakterisasi dibagi menjadi beberapa bagian, yaitu:

- a. Karakterisasi *plant* suhu 73412
 - b. Karakterisasi *driver* L298
 - c. Karakterisasi sensor KTY 10-6
 - d. Karakterisasi pengujian gangguan pada *plant* suhu 73412
4. Perancangan perangkat keras
 5. Penentuan nilai awal estimasi parameter control
 6. Perancangan dan pembuatan algoritma *self-tuning* PID *controller*

3.1 Perancangan Diagram Blok Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat bisa bekerja sesuai rencana. Blok diagram sistem untuk pengendalian suhu dapat kita lihat pada Gambar 3.1. *Input* sistem berupa suhu yang direpresentasikan dalam bentuk tegangan dimana 1V sama dengan 10°C. Kontroler yang digunakan dalam sistem ini adalah kontroler PID. *Plant* yang digunakan dalam sistem ini adalah *plant* suhu 73412 yang merupakan prototipe pengendalian suhu ruangan yang terdapat pada Laboratorium Sistem Kontrol Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang. *Disturbance* (gangguan) berupa perubahan nilai *input* pada sistem dan perubahan kecepatan kipas yang semula skala 2 menjadi skala 6. Pada sistem ini menggunakan sensor KTY 10-6 dengan range suhu -50°C – 150°C. Sedangkan untuk proses pencarian parameter estimasi menggunakan metode RLS (*Recursive Least Square*) dan pada desain kontroler menggunakan PID Dahlin serta *Output* pada sistem ini berupa suhu.



Gambar 3.1 Diagram Blok Sistem

Keterangan:

- $u_c(t)$: *setpoint* (nilai masukan sistem yang diinginkan pada keluaran sistem).
- $e(t)$: nilai selisih antara *setpoint* dengan *output*.
- $u(t)$: nilai keluaran dari kontroler PID (sinyal kontrol).
- $y(t)$: keluaran sebenarnya dari sistem.
- $\hat{\theta}(t)$: parameter estimasi yang dihasilkan dari metode RLS.

Kontroler parameter: nilai yang menghasilkan parameter kontroler yang baru untuk kontroler PID.

3.2 Spesifikasi Desain

Desain yang diinginkan pada perancangan pengontrolan sistem pengendalian suhu dengan *self-tuning* PID controller mempunyai spesifikasi yaitu:

1. *Settling time* < 120 detik

Settling time < 120 detik, karena pada sistem diharapkan dengan menggunakan teknik adaptif dengan skema *self-tuning controller* pada tuning kontroler PID mampu mempercepat *settling time* sistem kurang dari 120 detik.

2. Tidak memiliki *Overshoot*

Sistem mempunyai nilai *maximum overshoot* 0%.

3. Error Steady State < 5%

Error Steady State < 5%, karena sistem dapat dikatakan baik ketika memiliki *output* dengan batas nilai akhir 5% dari *setpoint*.

3.3 Karakterisasi Plant Suhu 73412

Untuk mengetahui karakteristik *plant* suhu 73412 pada perubahan *input* tegangan *plant* suhu 73412 terhadap *output plant* suhu 73412, maka dilakukan pengujian karakterisasi *plant* 73412. Pengujian dilakukan dengan cara membandingkan tegangan *input plant* suhu 73412 terhadap *output plant* suhu 73412 dengan menggunakan voltmeter yang dipasang pada *output plant* suhu 73412 (dimana 1 volt sama dengan 10°C) sebagai pembaca tegangan *output plant* suhu 73412.

Peralatan yang digunakan, yaitu:

1. Power supply 0-12V
2. Voltmeter
3. Plant suhu 73412

Langkah-langkah pengujian sebagai berikut:

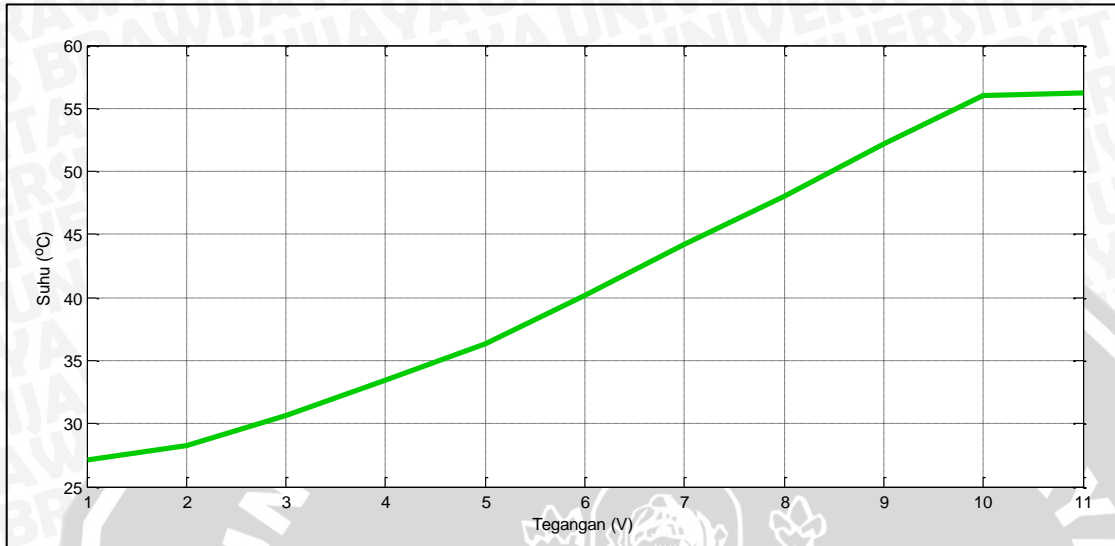
1. Menghubungkan *output power supply* ke *input plant* suhu 73412.
2. Menghubungkan voltmeter dengan *output plant* suhu 73412 dengan memilih skala 1V = 10°C.
3. Naikkan tegangan pada *power supply* dengan kenaikan 1V ketika sudah *steady*, dalam rentang 1-11V.
4. Ketika sudah *steady state* maka catat tegangan yang tertera pada *display* voltmeter.

Hasil dari pengujian tersebut dapat dilihat pada Tabel 3.1 dan dari Tabel 3.1 didapatkan kurva hubungan tegangan *input plant* suhu 73412 terhadap suhu *output plant* suhu 73412 seperti pada Gambar 3.2.

Tabel 3.1 Hasil Pengujian Plant Suhu 73412

Input plant suhu 73412 (V)	Output plant suhu 73412 (V)	Output plant suhu 73412 (°C)
1	2,71	27,1
2	2,83	28,3
3	3,06	30,6
4	3,34	33,4
5	3,63	36,3
6	4,02	40,2
7	4,42	44,2
8	4,8	48

<i>Input plant suhu 73412 (V)</i>	<i>Output plant suhu 73412 (V)</i>	<i>Output plant suhu 73412 (°C)</i>
9	5,22	52,2
10	5,6	56
11	5,62	56,2



Gambar 3.2 Hubungan Tegangan *Input Plant Suhu 73412* terhadap Suhu *Output Plant Suhu 73412*

3.4 Karakterisasi *Driver L298*

Untuk mengetahui karakteristik, kinerja, dan *output* dari *driver L298* terhadap perubahan *duty cycle* dari *microcontroller STM32F4 Discovery*, maka dilakukan pengujian karakterisasi *driver L298*. Pengujian dilakukan dengan cara membandingkan *input* berupa sinyal PWM (*Pulse Width Modulation*) yang diberikan oleh mikrokontroler *STM32F4 Discovery* terhadap tegangan *output driver* dengan menggunakan voltmeter yang dipasang pin *output driver*.

Peralatan yang digunakan, yaitu:

1. *STM32F4 Discovery*
2. *Driver L298*
3. *Power Supply 12V*
4. *Power Supply 5V*
5. Voltmeter

Langkah-langkah pengujian sebagai berikut:

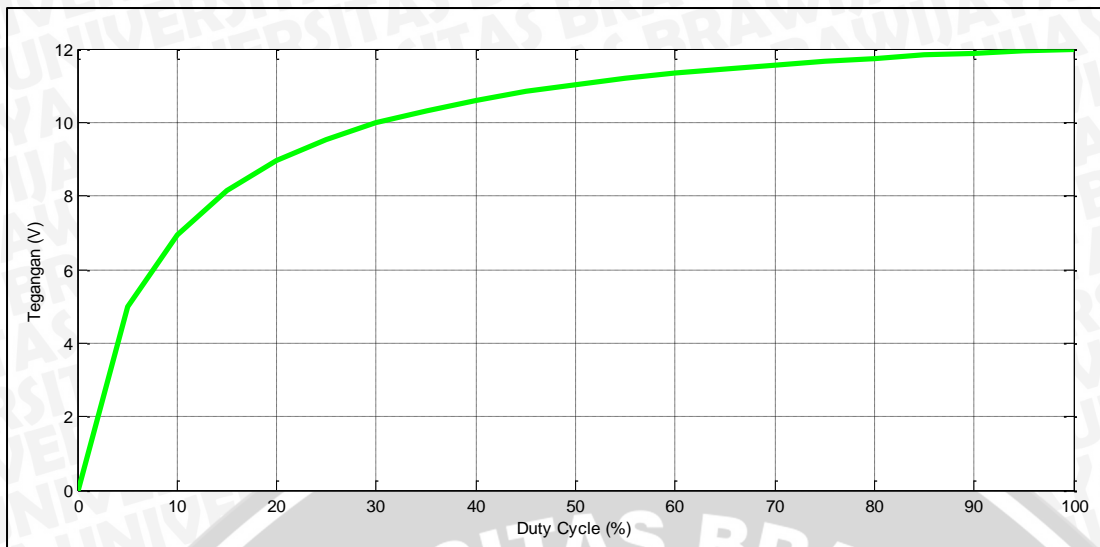
1. Menghubungkan *power supply 12V* dengan Pin *Vs driver L298*.
2. Menghubungkan *power supply 5V* dengan Pin *Vin driver L298*.
3. Menghubungkan Pin *D1-D2 STM32F4 Discovery* dengan Pin *INPUT_1* dan Pin *INPUT_2 driver L298*.

4. Menghubungkan Pin A0 STM32F4 *Discovery* dengan Pin ENABLE_A *driver* L298.
5. Dalam pengujian pertama *duty cycle* PWM yang keluar dari Pin A0 STM32F4 *Discovery* akan dinaikan setiap 5% dalam rentang 0-100%.
6. Kemudian catat masing-masing tegangan yang ditampilkan pada *display* voltmeter.

Hasil dari pengujian tersebut dapat dilihat pada Tabel 3.2 dan dari Tabel 3.2 didapatkan kurva hubungan *input duty cycle* sinyal PWM terhadap tegangan *output driver* L298 seperti pada Gambar 3.3.

Tabel 3.2 Hasil Pengujian Driver L298

<i>Duty Cycle (%)</i>	<i>Output Driver (V)</i>
0	0
5	5
10	6,94
15	8,14
20	8,95
25	9,54
30	9,98
35	10,33
40	10,61
45	10,84
50	11,03
55	11,2
60	11,34
65	11,46
70	11,57
75	11,67
80	11,75
85	11,83
90	11,89
95	11,96
100	12



Gambar 3.3 Hubungan Tegangan *Output Driver L298* terhadap *Duty Cycle*

3.5 Karakterisasi Sensor KTY 10-6

Untuk mengetahui kemampuan pembacaan dari sensor suhu KTY 10-6 terhadap perubahan suhu, maka dilakukan pengujian karakterisasi sensor KTY 10-6. Pengujian dilakukan dengan cara membandingkan sensor KTY 10-6 dengan sensor suhu *infrared*.

Peralatan yang digunakan, yaitu:

1. Sensor Suhu *Infrared*
2. Sensor KTY 10-6
3. *Plant* Suhu 73412
4. *Power Supply* 12V
5. Voltmeter

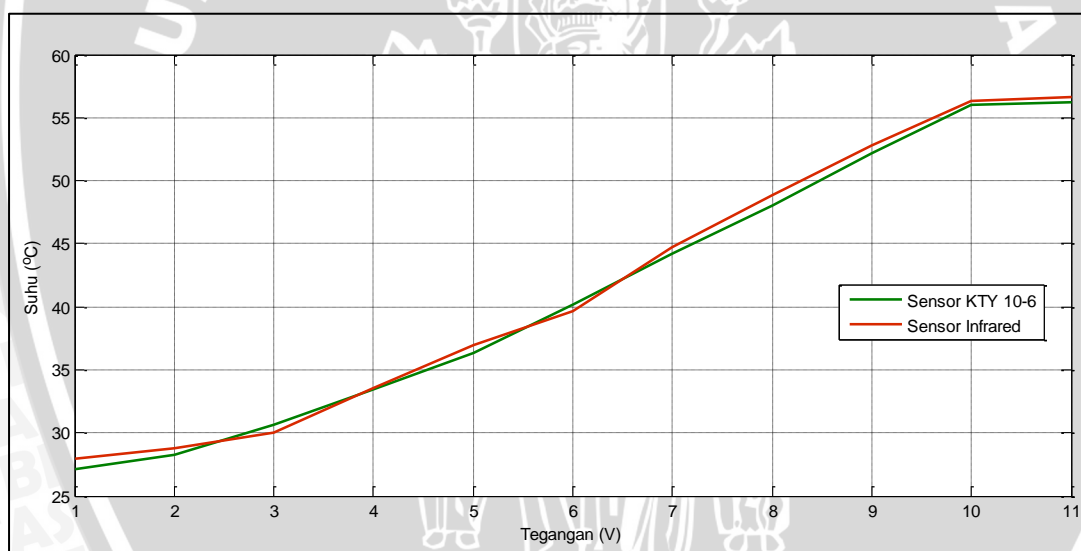
Langkah-langkah pengujian sebagai berikut:

1. Menghubungkan *power supply* 12V dengan *input plant* 73412.
2. *Input plant* 73412 dinaikkan 1V ketika *steady*, dalam rentang 1V-11V.
3. Ketika sudah *steady state* maka catat tegangan yang tertera pada *display* voltmeter.
4. Membandingkan hasil *output* sensor KTY 10-6 dengan sensor suhu *infrared*.

Hasil dari pengujian tersebut dapat dilihat pada Tabel 3.3 dan dari Tabel 3.3 didapatkan kurva perbandingan dari *output* sensor suhu KTY 10-6 dengan *output* sensor suhu *infrared* seperti pada Gambar 3.4.

Tabel 3.3 Hasil Pengujian Sensor Suhu KTY 10-6

<i>Input plant suhu</i> 73412 (V)	<i>Output sensor suhu</i> KTY 10-6	<i>Output sensor suhu</i> <i>infrared</i>	<i>Error pembacaan</i> (%)
1	27,1	27,9	2,86
2	28,3	28,8	1,74
3	30,6	30	2
4	33,4	33,5	0,3
5	36,3	37	1,89
6	40,2	39,7	1,28
7	44,2	44,7	1,12
8	48	48,9	1,84
9	52,2	52,8	1,14
10	56	56,3	0,53
11	56,2	56,6	0,71
<i>Error rata-rata</i>			1,4

Gambar 3.4 Perbandingan Sensor KTY 10-6 dengan Sensor *Infrared*

Dapat kita lihat dari Tabel 3.3 dan Gambar 3.4, sensor suhu KTY 10-6 memiliki kemampuan yang cukup baik dalam melakukan pembacaan perubahan suhu. Hal itu dibuktikan dengan *error* rata-rata yang didapatkan dengan membandingkannya dengan sensor *infrared* sebesar 1,4%.

3.6 Karakterisasi Pengujian Gangguan pada *Plant* Suhu 73412

Untuk mengetahui pengaruh perubahan *output plant* suhu 73412 saat diberikan gangguan. Oleh Karena itu, gangguan yang diberikan berupa perubahan kecepatan putaran kipas dari skala 2 menjadi skala 6.

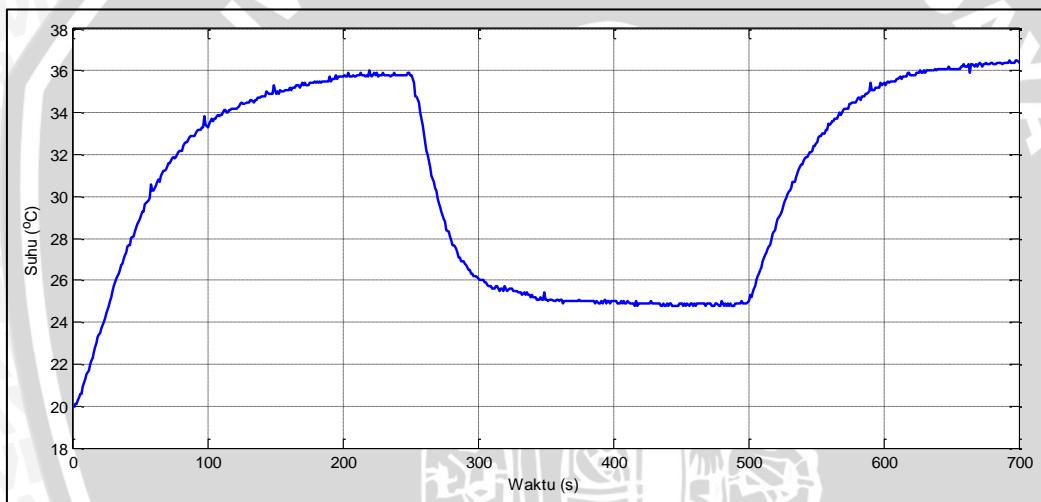
Peralatan yang digunakan, yaitu:

1. *Personal Computer (PC)*
2. *Power Supply 12V*
3. *Plant 73412*

Langkah-langkah pengujian sebagai berikut:

1. Menghubungkan *power supply* 12V dengan *input plant* suhu 73412.
2. *Power supply* diatur pada tegangan 6V.
3. Skala kecepatan kipas diatur pada skala 2.
4. Ketika respon *steady* maka skala kecepatan kipas diubah menjadi 6.
5. Ketika respon *steady* maka skala kecepatan kipas di kembalikan ke posisi 2.

Hasil dari pengujian tersebut dapat dilihat pada lampiran Tabel 4.1 dan dari Tabel 4.1 didapatkan kurva pengujian gangguan pada *plant* suhu 73412 seperti pada Gambar 3.5



Gambar 3.5 Output Plant Suhu 73412 dengan Gangguan

Berdasarkan Gambar 3.5, pengaruh gangguan yang diujikan dapat menurunkan suhu hingga 16°C atau 44,57% dari suhu *steady state*.

3.7 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal tersebut dimaksudkan agar sistem pengendali suhu *plant* (73412) dapat berjalan sesuai dengan yang telah direncanakan. Perancangan perangkat keras ini meliputi perancangan rangkaian *Driver* L298 dan rangkaian *STM32F4 Discovery*.

1. Rangkaian *Driver* L298

Rangkaian *driver* menggunakan jenis IC L298. *Driver* ini digunakan untuk mengontrol aktuator berupa lampu halogen dengan metode mengatur lebar modulasi sinyal PWM (*Pulse Width Modulation*) yang dikeluarkan dari *output*

mikrokontroler STM32F4 *Discovery* yang masuk ke *driver*. Catu daya pada lampu halogen yang digunakan adalah catu daya eksternal dengan tegangan sebesar 12V. Berikut konfigurasi pin *driver* yang digunakan pada sistem ini ditunjukkan Tabel 3.4.

Tabel 3.4 Konfigurasi Pin Driver L298

Pin	Fungsi
Power 5 V	Catu daya rangkaian driver
GND	Com rangkaian
Power Vs	Catu daya aktuator (lampu halogen)
<i>OUT_1</i>	Keluaran ke aktuator (lampu halogen)
ENABLE_A	Input PWM dari STM32F4 <i>Discovery</i>
<i>INPUT_1</i>	Pin pengendali <i>driver</i>
<i>INPUT_2</i>	Pin pengendali <i>driver</i>

2. Rangkaian STM32F4 *Discovery*

Perangkat keras kontroler sistem pengendalian suhu pada *plant* suhu 73412 ini menggunakan mikrokontroler STM32F4 *Discovery* sebagai pemroses kontrol utama. Konfigurasi pin *Input/ Output* (I/O) dari STM32F4 *Discovery* yang digunakan ditunjukkan pada Tabel 3.5.

Tabel 3.5 Konfigurasi Pin STM32F4 *Discovery*

Pin	Fungsi
Power 5 V	Catu daya STM32F4 <i>Discovery</i>
GND	Com Rangkaian
Pin A0	Keluaran sinyal kontrol (PWM)
Pin A1	Input sensor KTY 10-6
Pin D1	Pengendali <i>driver</i> L298
Pin D2	Pengendali <i>driver</i> L298
Pin D5	Tx USART2 STM32F4 <i>Discovery</i>
Pin D6	Rx USART2 STM32F4 <i>Discovery</i>

3.8 Prinsip Kerja Sistem

Prinsip kerja sistem pengendalian suhu pada *plant* suhu 73412 dengan menggunakan *self-tuning* PID *controller* adalah sebagai berikut:

1. Suhu *output* dari *plant* dibaca oleh sensor KTY 10-6 yang merupakan sensor suhu yang bersifat resistif, dimana jika suhu meningkat maka resistansi pada sensor KTY 10-6 juga akan berubah.
2. Selanjutnya hasil pembacaan dari sensor KTY 10-6 masuk ke rangkaian pengkondisi sinyal.

3. Tegangan keluaran rangkaian pengondisi sinyal berfungsi sebagai masukan analog bagi rangkaian ADC. Rangkaian ADC tersebut terdapat pada STM32F4 *Discovery* sebagai pemroses utama ADC.
4. Kemudian sinyal keluaran ADC tersebut kemudian diproses dengan menggunakan *self-tuning PID controller* sehingga menghasilkan keluaran berupa sinyal PWM pada pin A0 di mikrokontroler STM32F4 *Discovery*.
5. Keluaran sinyal PWM dari STM32F4 *Discovery* digunakan sebagai sinyal kontrol bagi aktuator berupa lampu halogen yang kemudian masuk kedalam *driver* L298.
6. Tegangan keluaran dari *driver* L298 merupakan sinyal kontrol PWM yang telah dikuatkan.
7. Lampu halogen akan memanaskan suhu dalam *plant* suhu 73412 yang diatur sesuai dengan besar *duty cycle* PWM yang dikeluarkan oleh mikrokontroler STM32F4 *Discovery*.
8. Kemudian hasil dari pembacaan sensor akan di tampilkan melalui GUI (*Grafical User Interface*) yang dibuat dengan *software* Processing 3.2.3 secara *real-time*.

3.9 Perancangan dan Pembuatan Program *Self-Tuning PID Controller*

Perancangan algoritma pada skripsi ini meliputi:

1. Perancangan algoritma untuk mencari estimasi parameter dari *plant*.
2. Perancangan algoritma untuk mendapatkan parameter kontroler dengan metode Kontroler Dahlin PID.
3. Perancangan algoritma untuk menentukan estimasi parameter awal.
4. Pembuatan *flowchart* untuk pembuatan program di perangkat lunak.
5. Implementasi pada alat.

3.10 Penentuan Nilai Awal Estimasi Parameter Kontrol

Untuk menghindari osilasi pada *output plant* (proses) maka disarankan untuk memilih estimasi parameter kontrol awal menggunakan persamaan berikut. Pada dasarnya persamaan ini merupakan *invers* dari persamaan (2-35), (2-36) dan (2-37).

$$C = 1 + \frac{T_0}{T_I} + \frac{T_D}{T_0} \quad (3-1)$$

$$\hat{a}_1 = -\frac{1 + 2\frac{T_D}{T_0}}{C} \quad (3-2)$$

$$\hat{a}_2 = \frac{T_D}{T_0 C} \quad (3-3)$$

$$\hat{b}_1 = \frac{Q}{K_p C} \quad (3-4)$$

Oleh karena itu, ketika memulai merancang algoritma kita harus menghindari memilih nilai nol untuk parameter \hat{b}_1 .

Sebelum kita dapat menentukan nilai dari $\hat{a}_1, \hat{a}_2, \hat{b}_1$ kita harus mendapatkan nilai K_p, T_I, T_D . Menurut (Dimas, 2015: 19-20) untuk mendapatkan nilai dari K_p, T_I, T_D dapat dengan menggunakan metode Ziegler-Nichols 1, dimana *plant* di rancang secara *open loop* dan diberikan tegangan masukan ke *plant* dan di lihat bagaimana responnya. Dari metode tersebut didapatkan nilai $K_p = 55.4, K_I = 44.2$, dan $K_D = 37.25$.

Untuk waktu sampling $T_0 = 1$ detik dan nilai *adjustment factor* menurut Bobál *et al*, $B = 1 \times 10^{-6}$. Sehingga dengan diketahuinya beberapa nilai yang di butuhkan, kita dapat menentukan nilai awal untuk estimasi parameter kontrol, dengan cara dibawah ini:

$$Q = 1 - e^{-\frac{T_0}{B}} = 1 - e^{-\frac{1}{10^{-6}}} = 1 - 0 = 1$$

$$T_I = \frac{K_p}{K_I} = \frac{55.4}{44.2} = 1.253393665$$

$$T_D = \frac{K_D}{K_p} = \frac{37.25}{55.4} = 0.672382671$$

$$C = 1 + \frac{T_0}{T_I} + \frac{T_D}{T_0} = 1 + \frac{1}{1.253393665} + \frac{0.672382671}{1} = 2.470216606$$

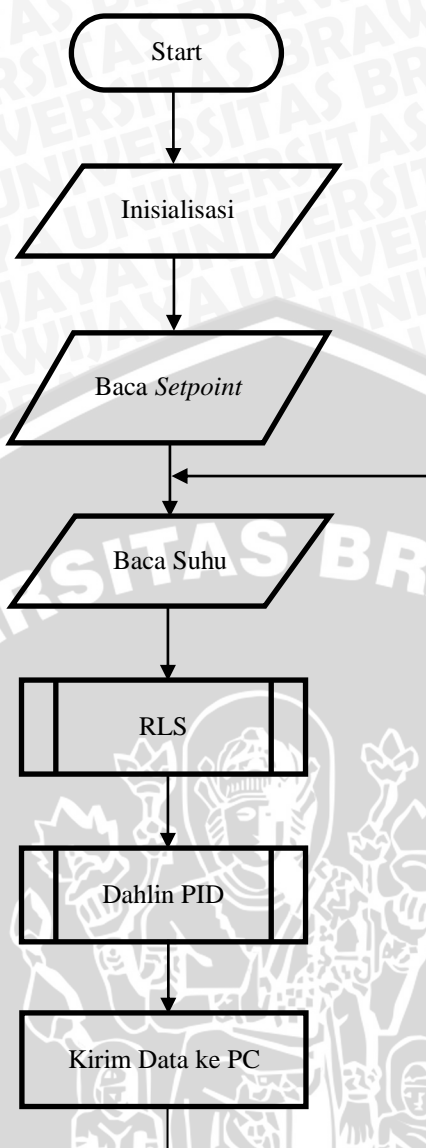
$$\hat{a}_1 = -\frac{1 + 2\frac{T_D}{T_0}}{C} = -\frac{1 + 2\frac{0.672382671}{1}}{2.470216606} = -0.949214468$$

$$\hat{a}_2 = \frac{T_D}{T_0 C} = \frac{0.672382671}{1 \times 2.470216606} = 0.272195834$$

$$\hat{b}_1 = \frac{Q}{K_p C} = \frac{1}{55.4 \times 2.470216606} = 0.007307270736$$

3.11 Flowchart Program Utama

Perancangan perangkat lunak berkaitan dengan *flowchart* program yang akan diimplementasikan ke *microcontroller* STM32F4 *Discovery*. Gambar 3.6 adalah *flowchart* sistem keseluruhan dan *flowchart* *Self-Tuning* PID Controller.



Gambar 3.6 *Flowchart* Sistem Keseluruhan