

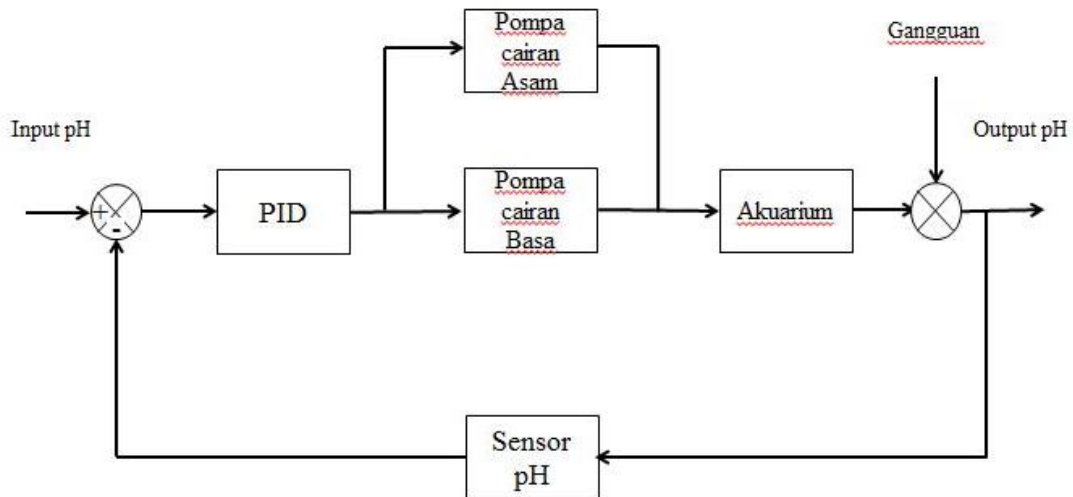
BAB III METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan informasi dengan tujuan dan manfaat tertentu. Metode penelitian pada skripsi ini meliputi:

- 3.1 Perancangan blok diagram sistem.
- 3.2 Spesifikasi desain.
- 3.3 Karakterisasi setiap blok.
 - 3.3.1 Karakterisasi driver motor.
 - 3.3.2 Karakterisasi motor DC 12 V
 - 3.3.3 Karakterisasi sensor pH SKUSEN0161.
- 3.4 Penentuan fungsi alih motor DC
- 3.5 Pembuatan perangkat keras.
 - 3.5.1 Desain media miniature tambak udang
 - 3.5.2 Prinsip kerja alat.
 - 3.5.3 Rangkaian catu daya.
 - 3.5.4 Konfigurasi pin mikrokontroler.
 - 3.5.5 Konfigurasi pin driver motor.
- 3.6 Perancangan algoritma.
 - 3.6.1 Parameter kontroler PID dengan metode root locus.
 - 3.6.2 Flowchart program.

3.1. Perancangan Blok Diagram Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana diagram blok sistem dapat dilihat pada gambar 3.1



Gambar 3. 1 Diagram blok sistem

Pada perancangan alat diperlukan perancangan diagram blok sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan desain yang diinginkan. Blok diagram sistem untuk pengontrolan pH dapat dilihat dalam Gambar 3.1. *Set point* sistem berupa pH 7 yang dikonversi dalam bentuk tegangan. Pusat pengendalian sistem menggunakan Arduino Mega 2560 yang memberikan sinyal keluaran berupa *pulse width modulation* (PWM) ke *driver* motor L298N. Pompa yang digunakan adalah pompa air dengan motor DC yang berfungsi sebagai aktuator. Sebagai *feed back* digunakan sensor pH SKU SEN0161 yang dihubungkan dengan rangkaian pengondisi sinyal karena tegangan yang dihasilkan sensor pH di luar cakupan tegangan masukan Arduino Mega 2560

3.2. Spesifikasi Desain

Desain yang diinginkan pada pengontrolan pH pada miniature tambak udang mempunyai spesifikasi yaitu:

1. Miniatur tambak udang menggunakan akuarium yang terbuat dari kaca yang berukuran 35cm x 18cm x 16cm.
2. Motor pompa yang digunakan untuk pengontrolan pH adalah motor DC dengan catu daya sebesar 12V. Motor DC ini digunakan karena catu daya yang digunakan adalah 12V DC.
3. Sensor pH yang digunakan adalah Sensor pH SKU SEN0161. Sensor ini digunakan karena dapat mengukur pH dari nilai 0 hingga 14.
4. pH sistem dikontrol pada nilai 7. Nilai ini digunakan karena pH ideal adalah 7 – 8.5.
5. Volume air yang digunakan sebanyak 4 liter

6. *Error steady state* kurang dari 5%. Diinginkan *error steady state* kurang dari 5% karena sistem yang baik memiliki *output* dengan batas nilai akhir 5% dari *setpoint*
7. Rasio redaman sama dengan 1. Diinginkan rasio peredaman sama dengan 1 agar *output* tidak berosilasi. Settling time kurang dari sama dengan 1 detik.

3.3. Karakterisasi Setiap Blok

3.3.1. Karakterisasi *Driver Motor*

a. Tujuan

Mengetahui kinerja dan respon rangkaian *driver motor* L298N dengan membandingkan *output* tegangan efektif *driver* dengan masukan *duty cycle* sinyal PWM yang diberikan oleh Arduino Mega 2560.

b. Peralatan yang digunakan

1. Komputer atau PC.
2. *Power Supply Unit*.
3. *Driver Motor* L298N.
4. Multimeter.
5. Arduino Mega 2560
6. Kabel penghubung.

c. Langkah pengujian

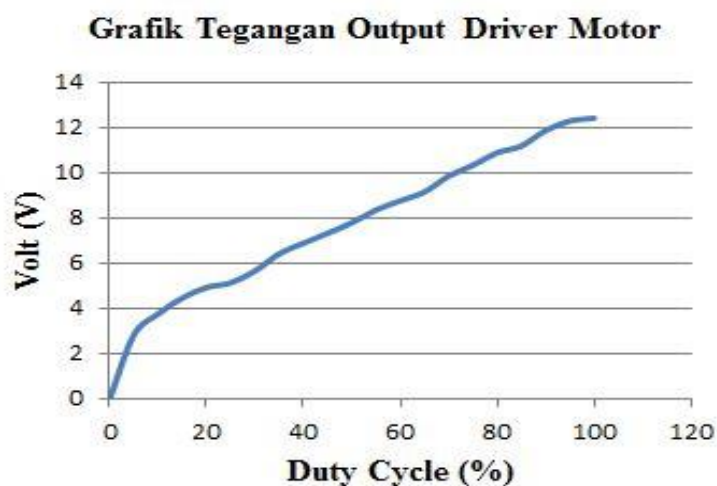
1. Menghubungkan *output* tegangan *Power Supply Unit* (PSU) pada *input* tegangan referensi *driver motor* LN298.
2. Menghubungkan *input* tegangan *driver motor* LN298 dengan pin *output* PWM di Arduino Mega 2560.
3. Menghubungkan *output* tegangan *driver motor* LN298 dengan multimeter.
4. Mengatur *duty cycle* sinyal PWM pada Arduino Nano dengan nilai 0%-100%.
5. Mengamati dan mencatat hasil pembacaan multimeter disetiap kenaikan 5%.

Data pengujian *driver motor* LN298 ditunjukkan dalam Tabel 3.1. Nilai tegangan keluaran *driver motor* dapat dilihat pada tabel di bawah ini.

Tabel 3. 1 Perubahan nilai tegangan terhadap *duty cycle*

Duty Cycle (%)	Vout (0-12V)
0	0
5	2,82
10	3,74
15	4,45
20	4,91
25	5,12
30	5,64
35	6,40
40	6,87
45	7,31
50	7,78
55	8,34
60	8,75
65	9,18
70	9,84
75	10,32
80	10,87
85	11,17
90	11,85
95	12,27
100	12,38

Hubungan tegangan keluaran driver dengan masukan driver yang berupa sinyal PWM pada tabel 3.1 diatas dapat direpresentasikan melalui gambar 3.2 dibawah ini.

Gambar 3. 2 Grafik hubungan tegangan terhadap *duty cycle*

3.3.2. Karakterisasi Motor DC

a. Tujuan

Mengetahui karakter atau gain motor DC pengontrolan pH. Hal tersebut diperoleh dengan mengamati kecepatan motor DC terhadap perubahan tegangan input motor DC.

b. Peralatan yang digunakan

1. *Power Supply*.
2. Motor DC pengontrolan pH.
3. Arduino Mega 2560.
4. *Driver* motor L298N .
5. Sensor IR LED dan *Photodiode*
6. Kabel penghubung

c. Langkah Pengujian

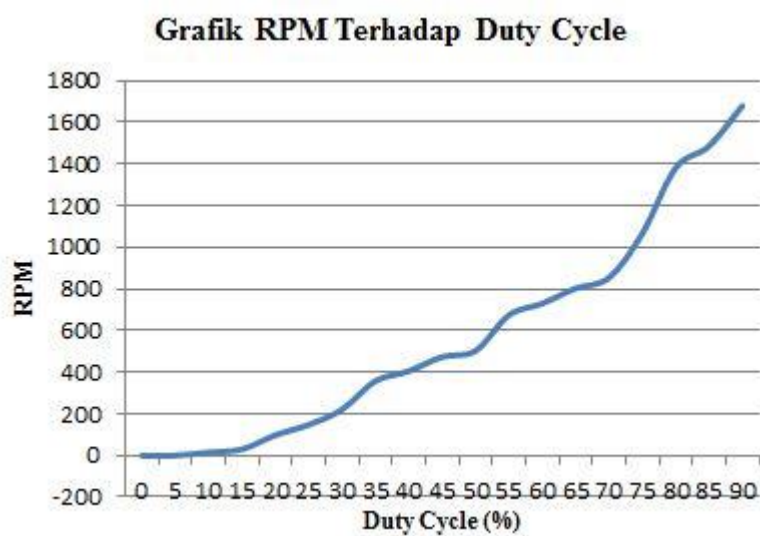
1. Menghubungkan output tegangan *Power Supply* pada *input* tegangan referensi *driver* motor L298N.
2. Menghubungkan *input* tegangan *driver* motor L298N dengan pin *output* PWM di Arduino Mega 256.
3. Menghubungkan output tegangan *driver* motor dengan motor DC 12V.
4. Merangkai sensor IR LED dan *Photodiode*.
5. Mengatur *duty cycle* sinyal PWM pada arduino mega 2560 dengan nilai 0%-90%.
6. Mengamati dan mencatat hasil pembacaan putaran motor setiap kenaikan 5%.

Data hasil pengujian *duty cycle* terhadap kecepatan motor DC dalam *rotation per minute* (rpm) untuk pengontrolan pH ditunjukkan dalam Tabel 3.2

Tabel 3. 2 Pengujian kecepatan motor DC terhadap *duty cycle*

Duty Cycle (%)	RPM
0	0
5	0
10	15
15	30
20	98
25	148
30	221
35	358
40	405
45	474
50	503
55	674
60	730
65	803
70	854
75	1067
80	1382
85	1487
90	1680

Hubungan rpm terhadap *duty cycle* pada tabel diatas dapat direpresentasikan melalui gambar 3.3 dibawah ini.

Gambar 3. 3 Grafik hubungan RPM terhadap *duty cycle*

3.3.3. Karakterisasi Sensor pH SKU SEN0161

a. Tujuan

Mengetahui kemampuan pembacaan sensor pH SKU SEN0161 terhadap perubahan pH.

b. Peralatan yang digunakan

1. Sensor pH SKU SEN0161.
2. pH meter.
3. Larutan pH *buffer*.
4. Arduino Mega 2560.
5. Kabel penghubung.

c. Langkah Pengujian

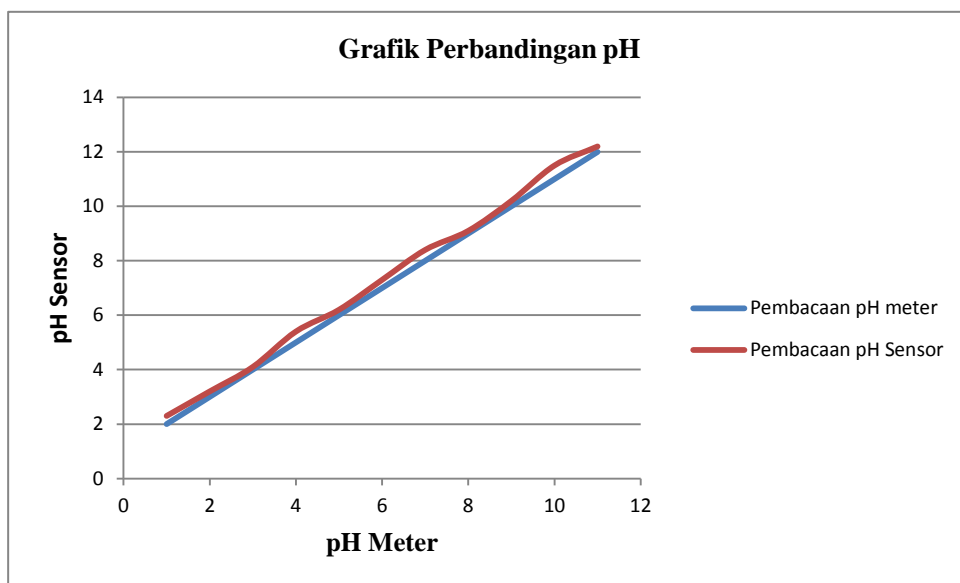
1. Menghubungkan *output* sensor pH ke pin analog arduino mega 2560.
2. Meletakkan larutan pH *buffer* pada wadah.
3. Mencelupkan ujung sensor pH dan pH meter pada larutan *buffer*.
4. Mencatat data pembacaan antara pH meter dan sensor pH.

Hasil pengujian sensor pH SKU SEN0161 ditunjukkan dalam Tabel 3.3. Pada Tabel tersebut terdapat pembacaan pH meter sebagai acuan dan pembacaan sensor pH. Pada pembacaan antara pH meter dan sensor pH terdapat error yang dinyatakan dalam persen (%).

Tabel 3. 3 Hasil Pengujian Sensor pH SKU SEN0161

NO	pH Pembacaan pH Meter	pH Pembacaan pH Sensor	Error (%)
1.	2	2,3	0,15
2.	3	3,2	0,06
3.	4	4,1	0,02
4.	5	5,4	0,08
5.	6	6,2	0,03
6.	7	7,3	0,04
7.	8	8,4	0,05
8.	9	9,1	0,01
9.	10	10,2	0,02
10.	11	11,5	0,04
11.	12	12,2	0,01
Rata-rata			0,51

Berdasarkan hasil pengujian dalam Tabel 3.3, error antara pembacaan pH meter dengan pembacaan pH sensor pada serial monitor yaitu 0,51%. Grafik perbandingan pH hasil pembacaan pH meter acuan dengan pH sensor pada serial monitor dapat dilihat dalam Gambar 3.4. Pada grafik tersebut terlihat bahwa selisih yang timbul antara kedua pembacaan pH cenderung membentuk garis linear, Sehingga untuk mengurangi error pembacaan pH dapat menggunakan cara kalibrasi. Perbandingan pengukuran pH pada Sensor dengan pH pada pH meter pada tabel diatas dapat direpresentasikan melalui gambar 3.4 dibawah ini.

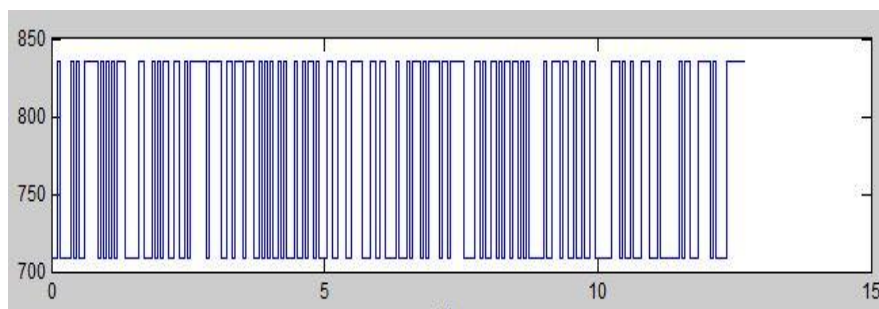


Gambar 3. 4 Grafik perbandingan pH

3.4. Penentuan Fungsi Alih Motor DC

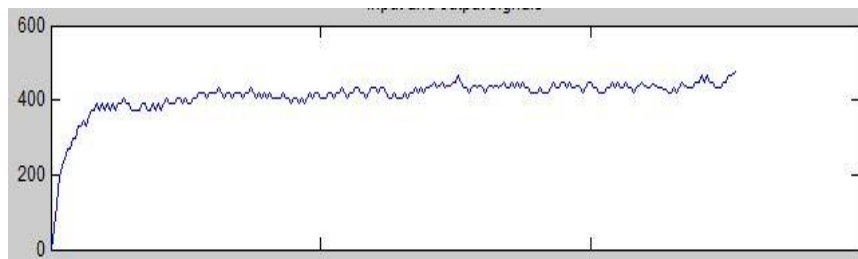
Dalam pengontrolan pH pada miniatur tambak udang dibutuhkan sistem yang mampu mengontrol motor pompa. Pengontrolan kecepatan motor pompa DC menggunakan Arduino Mega 2560 sebagai pengolah data. Motor pompa DC yang digunakan pada perancangan ini tidak diketahui karakteristiknya, sehingga yang perlu dilakukan adalah melakukan pengujian dengan menggunakan sensor IR LED dan IR *photodiode*. Karakteristik motor pompa DC pada perancangan ini didapatkan dengan cara memberikan masukan unit step. Fungsi alih motor pompa didapatkan dari pemodelan dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS). Langkah yang dilakukan untuk membangkitkan sinyal PRBS adalah sebagai berikut :

1. Mencari nilai yang linear dari hasil kecepatan motor terhadap *duty cycle* PWM.
2. Memasukkan nilai batas atas dan batas bawah berdasarkan nilai yang linear untuk membangkitkan sinyal PRBS.
3. Sinyal PRBS yang telah dibangkitkan kemudian digunakan sebagai masukan motor pompa DC.
4. Setelah didapatkan data sinyal PRBS dan data kecepatan motor pompa DC, selanjutnya adalah melakukan identifikasi dengan menggunakan software Matlab. Grafik input PRBS dapat dilihat dalam Gambar 3.5



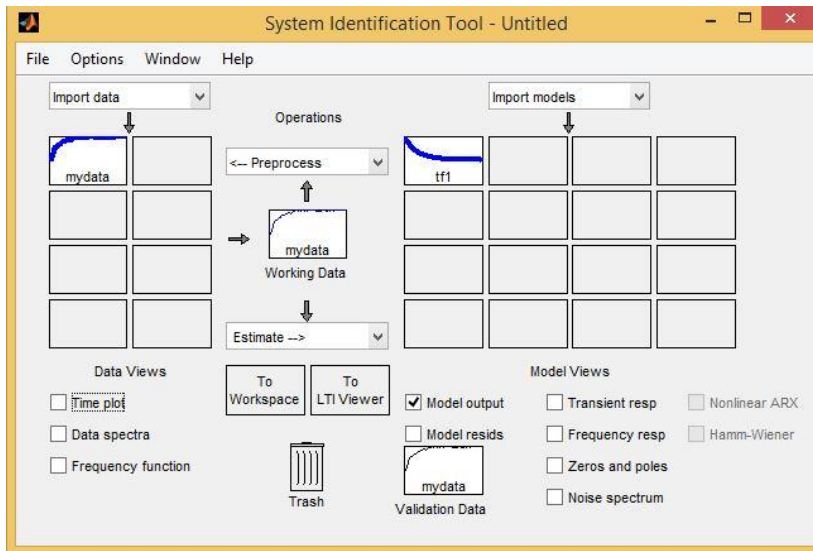
Gambar 3. 5 Input PRBS

Grafik output PRBS dapat dilihat dalam Gambar 3.6



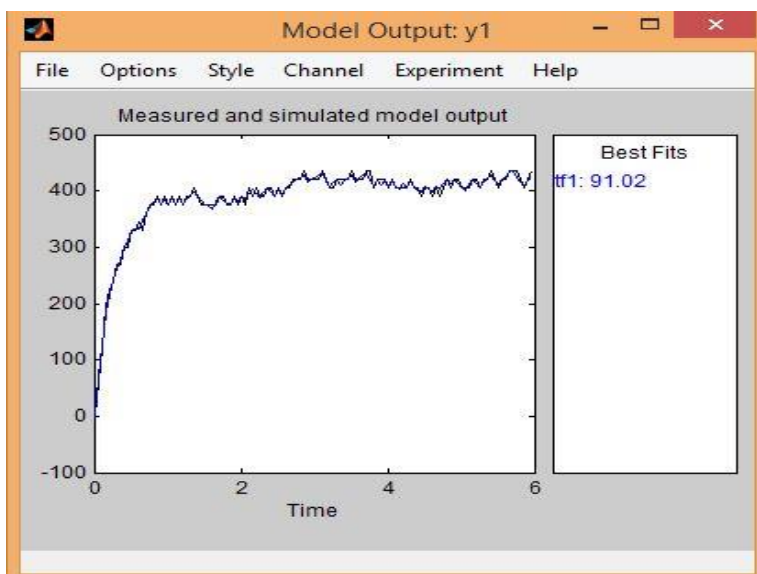
Gambar 3. 6 Output PRBS

5. Data kecepatan motor yang telah disimpan kemudian di import pada blok System Identification Toolbox seperti dalam Gambar 3.7



Gambar 3. 7 Tampilan aplikasi ident pada software matlab

6. Setelah melakukan beberapa estimasi model berdasarkan data yang telah di import didapatkan fungsi alih dari motor dengan bestfit sebesar 91.02 seperti dalam, Gambar 3.8



Gambar 3. 8 Tampilan Best Fits

7. Dari hasil identifikasi, fungsi alih motor yang didapat adalah

$$G(s) = \frac{2,573s + 1,298}{s^2 + 5,466s + 2,372}$$

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

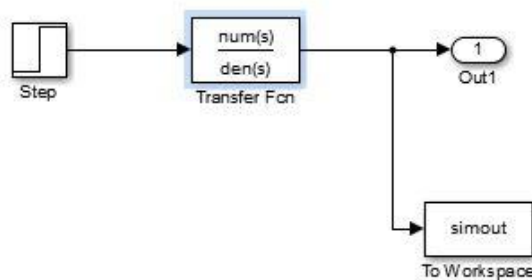
$$2\xi\omega_n = 5,466$$

$$\omega_n = \sqrt{2,372} = 1,540$$

$$\xi = \frac{5.466}{2 \times 1,540} = 1,774$$

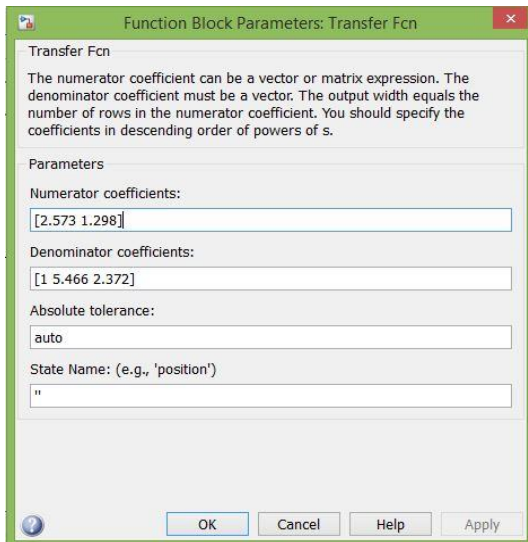
8. Setelah didapatkan fungsi alih maka fungsi alih harus divalidasi karena suatu fungsi alih dapat dikatakan telah mewakili *plant*. Proses validasi, langkah-langkah proses validasi adalah :

- Menentukan masukan step, pada proses validasi ini dipilih masukan step PWM 70 yaitu rpm sebesar 854.
- Menguji plant motor DC untuk didapatkan nilai keluaran dengan masukan PWM 70 untuk dibandingkan dengan keluaran dari model fungsi alih yang didapatkan dari identifikasi di Matlab.
- Membuat simulink keluaran model fungsi alih yang sudah didapatkan dengan identifikasi di matlab di tool simulink seperti pada gambar 3.9 dibawah ini.



Gambar 3. 9 Simulink Keluaran Model Fungsi Alih

- Memasukan nilai step 1, dengan nilai awal bernilai 0 dan nilai akhir yang dicapai adalah sesuai dengan *set point* pwm yang ditentukan yaitu 854.
- Memasukan nilai model fungsi alih yang sudah di dapat ke dalam *function block parameters transfer function* dalam simulink.



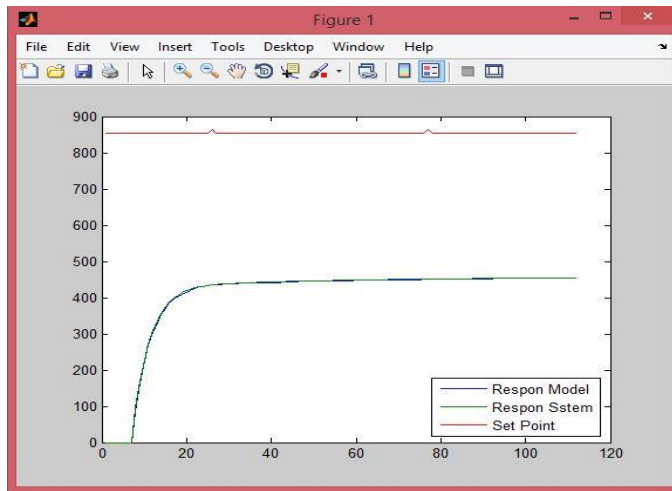
Gambar 3. 10 Function blok parameter transfer function

- Kemudian adalah “run” program untuk mendapatkan nilai keluaran dari model fungsi alih.

	1	2	3
84	452	452.2387	854
85	452	452.3488	854
86	452	452.4563	854
87	452	452.5613	854
88	452	452.6638	854
89	452	452.7639	854
90	452	452.8617	854
91	452	452.9572	854
92	452	453.0504	854
93	453	453.1414	854
94	453	453.2303	854
95	453	453.3171	854

Gambar 3. 11 Nilai keluaran fungsi alih dan motor DC

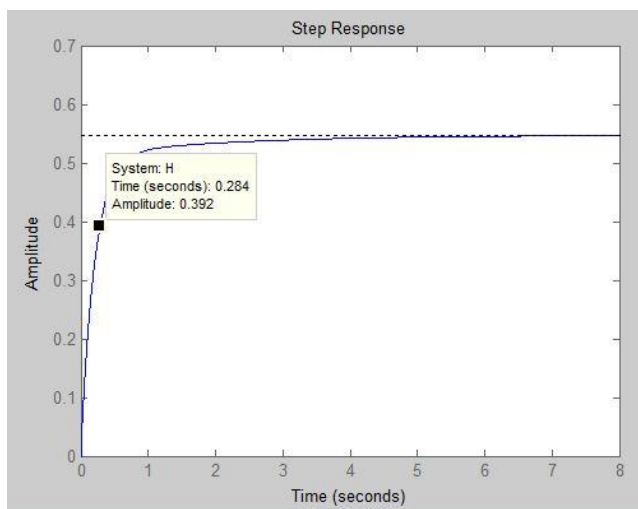
- Setelah di “Run” maka akan keluar nilai di *workspace* seperti gambar diatas. Kolom paling kiri adalah nilai keluaran model fungsi alih yang didapatkan dengan simulink. Kolom yang ditengah adalah nilai keluaran yang didapatkan dengan menguji langsung plant motor DC di arduino. Sementara kolom yang paling kanan adalah nilai masukan yang diinginkan
- Membandingkan nilai keluaran model fungsi alih dengan nilai keluaran langsung dari motor DC dengan cara plot di Matlab



Gambar 3. 12 Hasil plot validasi fungsi alih

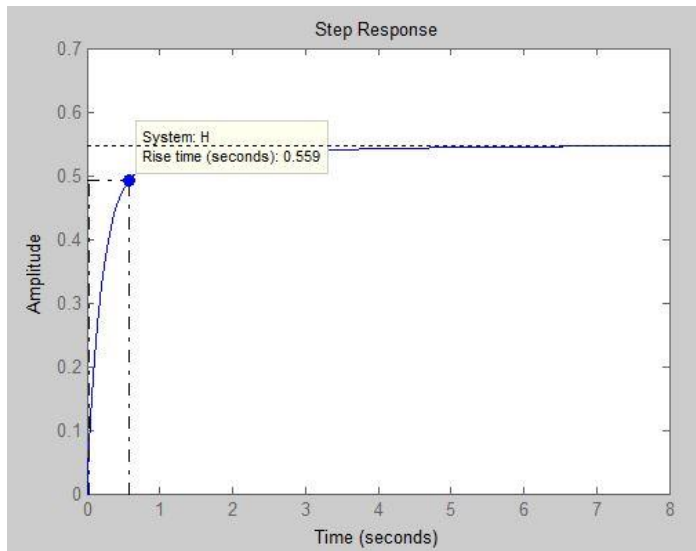
Dari gambar 3.12 diatas dapat dilihat bahwa model keluaran fungsi alih sudah mewakili model motor DC aslinya sehingga fungsi alih yang didapat sudah sesuai.

Dengan memberikan input unit step pada program Matlab didapatkan output pada fungsi alih *closed loop* dengan nilai *time constant* yang merupakan waktu yang dibutuhkan untuk mencapai 63,2% dari nilai steady state, yaitu 0,284 detik seperti dalam Gambar 3.13



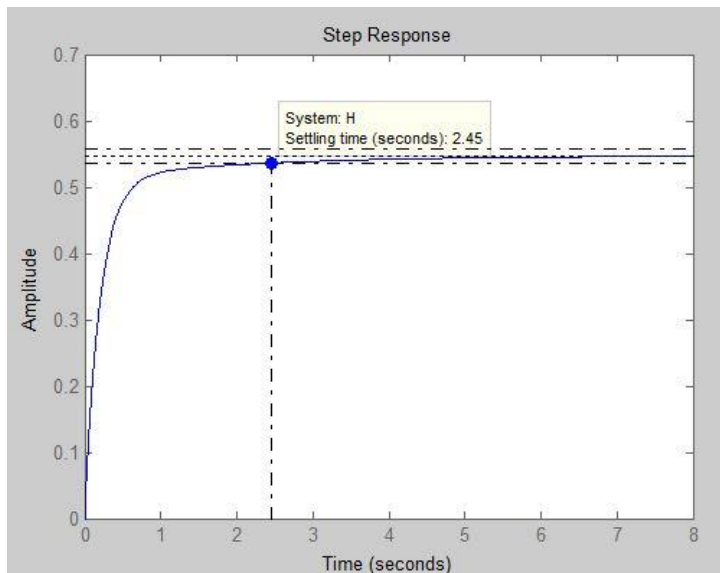
Gambar 3. 13 Nilai *time constant* output motor DC dengan *input unit step*

Dengan memberikan *input* unit step pada program Matlab didapatkan pula nilai *rise time* adalah 0,559 detik seperti dalam Gambar 3.14



Gambar 3. 14 Nilai *rise time output* Motor DC dengan *input unit step*

Dengan memberikan *input unit step* pada program Matlab didapatkan nilai *settling time* adalah 2.45 detik seperti dalam Gambar 3.15



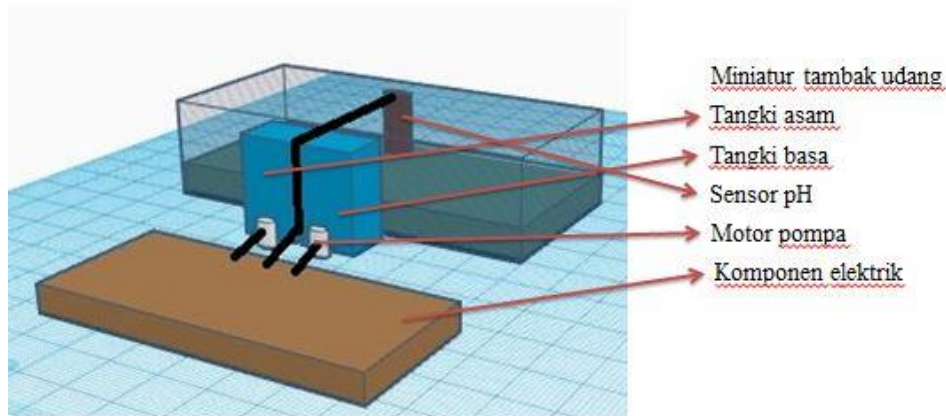
Gambar 3. 15 Nilai *settling time output* motor DC dengan *input unit step*

3.5. Perancangan dan Pembuatan Perangkat Keras

Perancangan dan pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya, hal ini bertujuan supaya *output* pH sesuai dengan masukan yang diinginkan dan sistem dapat bekerja dengan baik sesuai yang direncanakan.

3.5.1. Desain Media Miniatur Tambak Udang

Desain miniature tambak udang menggunakan box yang terbuat dari bahan kaca dengan ukuran 35 cm x 18 cm x 16 cm. Box diisi air dengan volume 4 liter. Desain media hidroponik dapat dilihat dalam Gambar 3.16



Gambar 3. 16 Desain media miniatur tambak udang

3.5.2. Prinsip Kerja Alat

1. Catu daya sebesar ± 5 Volt digunakan sebagai catu rangkaian pengondisi sinyal, dan *enable driver* motor. Catu daya 12 Volt digunakan sebagai catu motor DC pada pompa air.
2. Port komunikasi antara arduino mega 2560 dan PC menggunakan perantara kabel USB.
3. Menggunakan sensor pH SKU SEN0161, sinyal-sinyal keluaran dari sensor yang masih berupa sinyal analog diproses terlebih dahulu melalui rangkaian pengondisi sinyal agar sesuai dengan sinyal yang dibutuhkan untuk dapat dibaca oleh Arduino Mega 2560.
4. Motor DC pada pompa air yang terhubung pada tangki berisi cairan asam akan berputar ketika pembacaan sensor menunjukkan bahwa keadaan air terlalu basa atau lebih dari setpoint yaitu pH 7.
5. Motor DC pada pompa air yang terhubung pada tangki berisi cairan basa akan berputar ketika pembacaan sensor menunjukkan bahwa keadaan air terlalu asam atau kurang dari *setpoint* yaitu pH 7.
6. Motor DC pada kedua pompa air berhenti berputar saat pH air mencapai *setpoint* yang diinginkan.

3.5.3. Rangkaian Catu Daya

Rangkaian catu daya yang digunakan adalah adaptor yang memiliki tegangan keluaran sebesar 12V. Adaptor merupakan rangkaian elektronika yang bekerja mengubah tegangan AC yang tinggi menjadi tegangan DC yang rendah. Catu Daya 12 Volt untuk valve diambil dari *switching power supply* 12V dengan arus maksimal sebesar 3 ampere. Adaptor ditunjukkan pada gambar 3.17



Gambar 3. 17 Adaptor

3.5.4. Konfigurasi Pin Mikrokontroller Arduino Mega 2560

Berikut adalah konfigurasi pin mikrokontroler Arduino Mega 2560 ditunjukkan dalam Tabel 3.4

Tabel 3. 4 Konfigurasi pin arduino mega 2560

NO	Pin	Fungsi
1.	D3	Pump A PWM
2.	D4	Pump A Direction A
3.	D5	Pump A Direction B
4.	D6	Pump B Direction A
5.	D7	Pump B Direction B
6.	D8	Pump B PWM
7.	D13	Sensor pH
8.	GND	Ground
9.	Vcc	Catu daya 5 volt

3.5.5. Konfigurasi Pin Driver Motor

Rangkaian *driver* motor yang digunakan adalah *driver* motor L298N. *Driver* motor ini digunakan untuk mengendalikan 2 motor DC. Motor DC yang digunakan adalah motor DC untuk pompa asam dan motor DC untuk pompa basa. Mode pengontrolan kecepatan motor yang digunakan adalah mode sinyal pulse width modulation (PWM). Catu daya motor yang digunakan adalah catu daya eksternal 12V. Konfigurasi pin driver motor ini ditunjukkan dalam Tabel 3.5 berikut:

Tabel 3. 5 Konfigurasi pin *driver* motor

NO	Pin	Fungsi
1.	D3	Pump A PWM
2.	D4	Pump A Direction A
3.	D5	Pump A Direction B
4.	D6	Pump B Direction A
5.	D7	Pump B Direction B
6.	D8	Pump B PWM
7.	Vcc	Catu daya 5 volt
8.	GND	Ground

3.6. Perancangan Algoritma

3.6.1. Penentuan Parameter Kontroler PID dengan Metode Root Locus

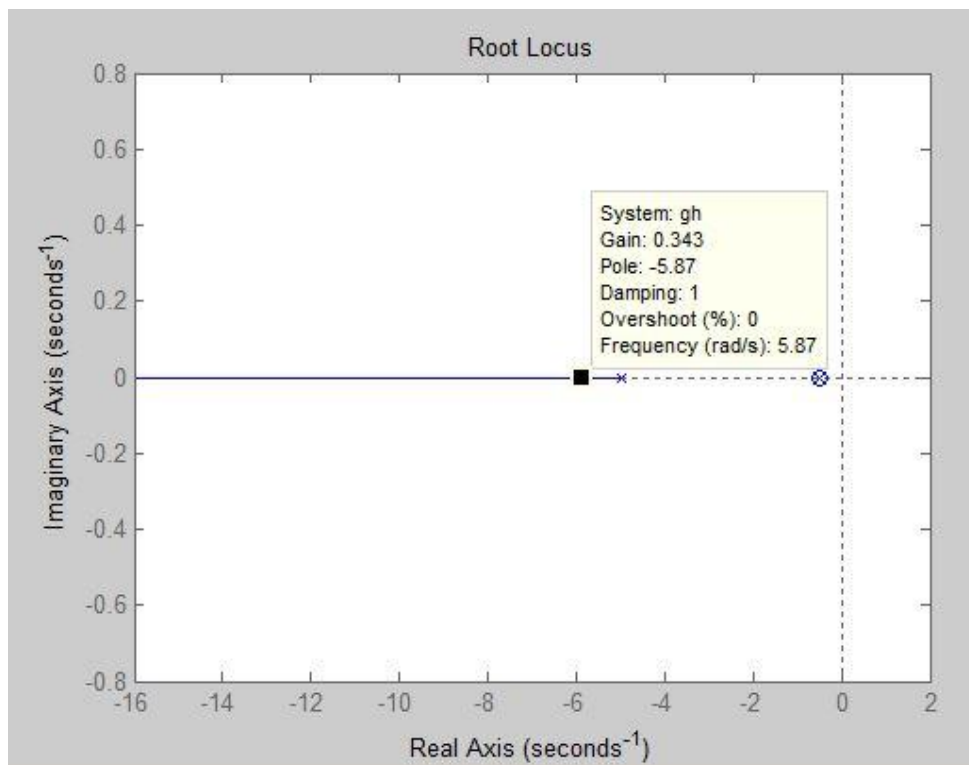
Metode *Root Locus* atautetak kedudukan akar digunakan untuk meneliti perilaku sistem dengan parameter sistem berubah pada lingkup tertentu. Rancangan dimaksudkan agar letak *pole* dan *zero* dari fungsi alih loop tertutup terletak pada daerah yang ditentukan. Agar sistem stabil, pole dan zero harus terletak pada bidang *s* sebelah kiri sumbu imajiner. Kontroler yang dipilih ialah Kontroler *Proporsional Integral Diffeensial* (PID).

- Mencari fungsi alih Motor DC tanpa Kontroler terlebih dahulu dengan memberikan sinyal *Pseudo Random Binary Sequence* (PRBS).
- Setelah didapatkan fungsi alih, maka di periksa kestabilannya dengan *root locus* yang ada di Matlab. Fungsi Alih yang stabil adalah fungsi alih yang nilai akar *s* nya sudah berada di sumbu kiri imajiner.
- Menentukan letak poles s_1 sesuai spesifikasi desain pada penelitian ini.
- Setelah itu mencari nilai parameter K_p , K_i dan K_d di Matlab dengan mensubstitusi nilai s_1 dan nilai fungsi alih sistem.
- Untuk mendapatkan Nilai Parameter K_p , K_i dan K_d , maka variasikan nilai parameter K_i terlebih dahulu kemudian akan didapatkan nilai parameter PID yang dilakukan di program Matlab.

Setelah didapatkan fungsi alih motor DC yaitu :

$$\frac{Y(s)}{U(s)} = \frac{2,573s + 1,298}{s^2 + 5,466s + 2,372}$$

maka selanjutnya dapat menentukan letak poles tertutup. Penentuan letak pole pada *Root Locus* harus sesuai dengan spesifikasi desain kontrol yang telah ditentukan. Pada penelitian ini spesifikasi desain yang ditentukan adalah *overshoot* kurang dari 5%, rasio redaman=1 serta *settling time* kurang dari sama dengan 1. *Settling time* adalah ukuran waktu yang menyatakan respon telah masuk 5% atau 2% atau 0.5% dari keadaan *steady state*. Pada kurva *Root Locus* penelitian ini digunakan poles $s_1 = -5,87$ dikarenakan pada letak poles $s_1 = -5,87$ didapatkan nilai rasio redaman = 1. Penentuan letak poles tersebut sudah sesuai dengan spesifikasi desain kontrol pada penelitian ini. Penentuan letak poles *root locus* dapat dilihat pada Gambar 3.18



Gambar 3. 18 Letak pole pada diagram *root locus*

Setelah ditentukan letak pole yang diinginkan kemudian dengan menstutitisi nilai s_1 pada nilai fungsi alih sistem dan memvariasikan nilai K_i akan didapatkan parameter PID dalam Tabel 3.6. Pencarian parameter K_p , K_i , dan K_d dengan menggunakan MATLAB 2014a ditunjukkan pada listing program berikut :

`%nilai pole yang ditentukan dari Gambar root locus`

```

s1=-5.87

KI=[0.01 0.1 0.5 1 2]

plant_num=[0 2.573 1.298];
plant_den=[1 5.466 2.372];

slmag = abs(s1)

beta = angle(s1)

plant_a1 = polyval(plant_num,s1)/polyval(plant_den,s1);

plantslmag = abs(plant_a1)

psi = angle(plant_a1)

t=0:1:20:300;

for k =1:5

KP          =          -sin(beta+psi)/(plantslmag*sin(beta))-
2*KI(k)*cos(beta)/slmag
nilai_KI = KI(k)
KD = sin(psi)/(slmag*plantslmag*sin(beta))+KI(k)/slmag^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)

```

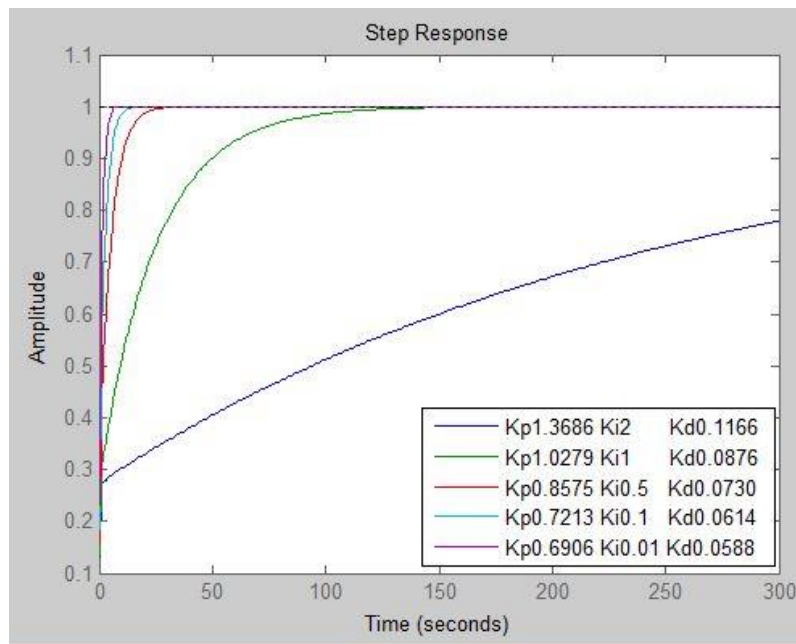
Hasil pencarian parameter Kp, Ki dan Kd dari perhitungan pada program diatas ditunjukkan dalam Tabel 3.6

Tabel 3. 6 Parameter PID dengan $s=-5,31$

NO	Kp	Ki	Kd
1.	0,6906	0,01	0,0588
2.	0,7213	0,1	0,0614
3.	0,8575	0,5	0,0730
4.	1,0279	1	0,0876
5.	1,3686	2	0,1166

Setelah didapatkan nilai K_p , K_i dan K_d hasil perhitungan, kemudian dilakukan pengujian terhadap sistem, dan parameter yang sesuai dengan sistem.

Pada Gambar 3.19 berikut adalah grafik output motor DC pengontrolan pH dengan menggunakan kontroler dan memvariasikan nilai K_p , K_i dan K_d seperti dalam Tabel 3.6

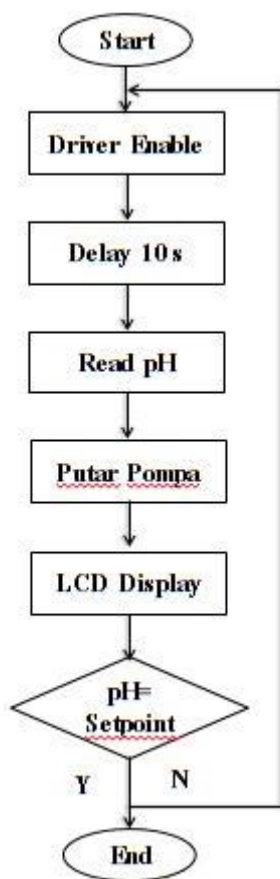


Gambar 3. 19 Output sistem pengontrolan pH dengan kontroler PID

Dari 5 jenis parameter PID yang didapat dipilih nilai parameter PID yang memiliki respon terbaik yaitu nilai $K_p = 0,6906$, nilai $K_i = 0,01$ dan nilai $K_d=0,0588$.

3.6.2. Flowchart

Flowchart program merupakan gambaran alur proses program yang dilakukan oleh kontroler pada saat implementasi. *Flowchart* program dalam skripsi ini dapat dilihat dalam Gambar 3.20



Gambar 3. 20 Flowchart