

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini 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 Driver Motor EMS H-Bridge 5 A dengan Beban Motor Pompa DC 12 V

a. Tujuan

Mengetahui *range* kerja dari motor pompa DC 12 V dengan input PWM yang berbeda-beda.

b. Peralatan yang digunakan

1. Motor pompa DC 12 V.
2. Multimeter.
3. Arduino Uno.
4. Driver EMS H-BRIDGE 5A.
5. Komputer.
6. Kabel jumper.
7. Catu daya 12 V.

c. Langkah pengujian

- I. Merangkai peralatan Catu daya, driver, Arduino dan motor pompa
- II. Memberikan PWM dengan kelipatan 10 melalui Arduino Uno.
- III. Mencatat data tegangan keluaran motor pompa DC dengan PWM yang diberikan untuk melihat kelinierannya.

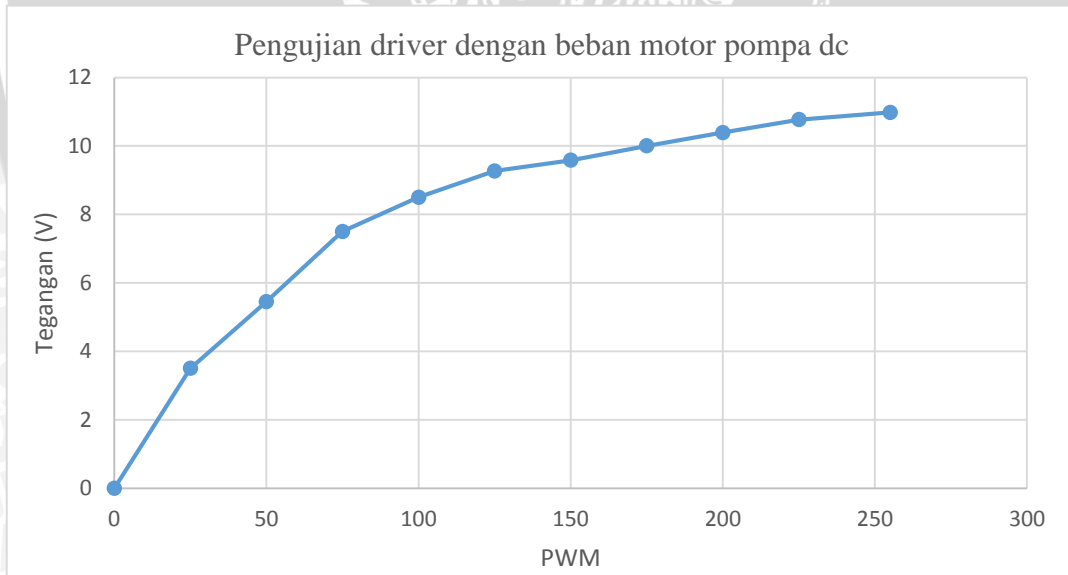
d. Hasil Pengujian

Data hasil pengujian motor pompa DC ditunjukkan dalam Tabel 5.1 berikut.

Tabel 5.1 Pengujian keluaran driver motor EMS H-Bridge 5 A dengan beban motor pompa DC 12 V

PWM	Keluaran Tegangan (V)
0	0
25	3.5
50	5.45
75	7.5
100	8.5
125	9.27
150	9.58
175	10
200	10.39
225	10.77
255	10.98

Grafik keluaran driver motor EMS H-Bridge 5 A dengan beban motor pompa DC 12 V dengan masukan PWM yang berbeda-beda dapat dilihat dalam Gambar 5.1



Gambar 5.1 Grafik Kinerja Motor Pompa DC

5.2 Pengujian Sinyal PWM Arduino UNO

a. Tujuan

Pengujian ini bertujuan untuk mengetahui kinerja sinyal PWM Arduino UNO.

b. Peralatan yang digunakan.

1. *Digital Oscilloscope* Vellemen PCSU1000
2. Arduino Uno.

c. Langkah pengujian

- I. Menghubungkan pin PWM Arduino UNO untuk *driver* motor dengan *Digital Oscilloscope* Vellemen PCSU1000.
- II. Arduino UNO memberikan instruksi sinyal masukan pada pin PWM *driver* motor EMS H-Bridge 5 A dengan PWM 0 - 255.
- III. Mencatat hasil pengujian.

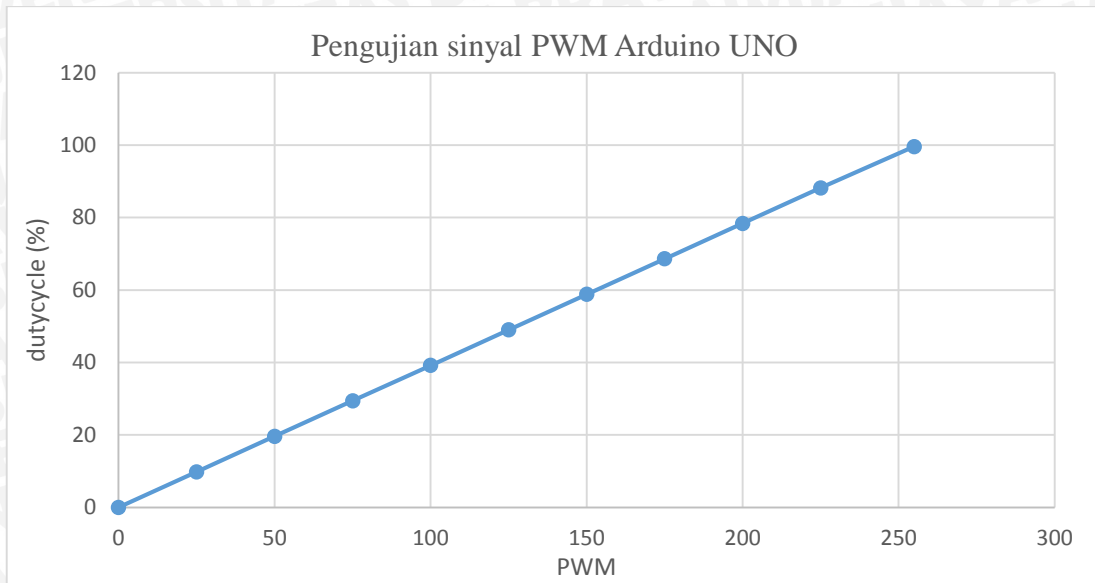
d. Hasil pengujian

Hasil pengujian sinyal PWM Arduino UNO dapat dilihat dalam Tabel 5.2.

Tabel 5.2 Hasil Pengujian PWM Arduino UNO

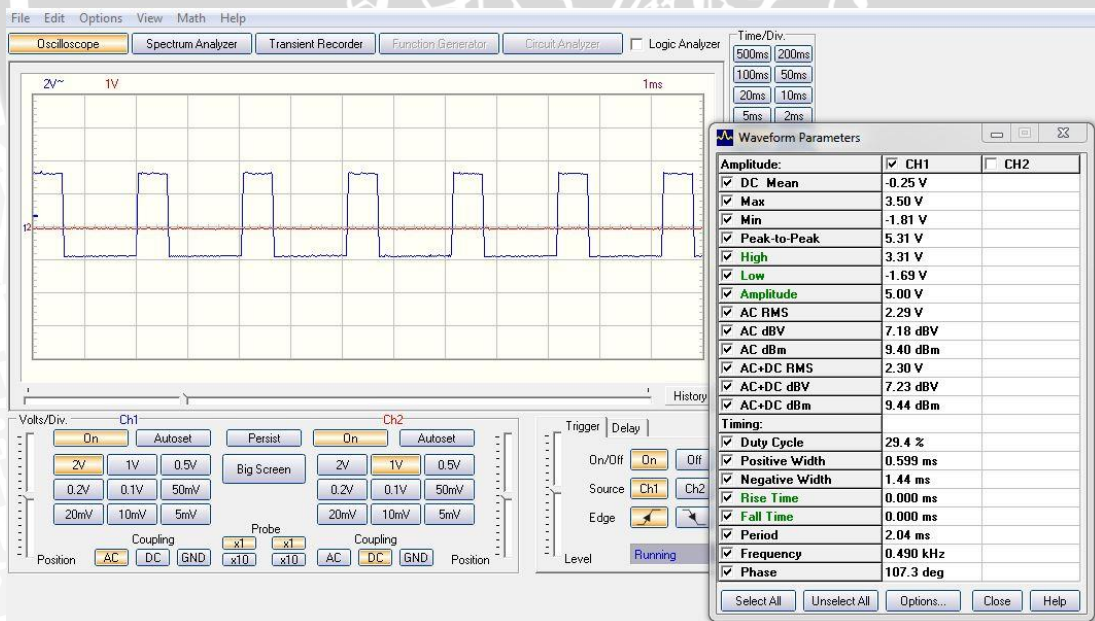
PWM	Duty Cycle (%)
0	0
25	9.8
50	19.6
75	29.4
100	39.2
125	49
150	58.8
175	68.6
200	78.4
225	88.2
255	100

Grafik hubungan antara PWM masukan dengan duty cycle keluaran Arduino UNO dapat dilihat dalam Gambar 5.2.

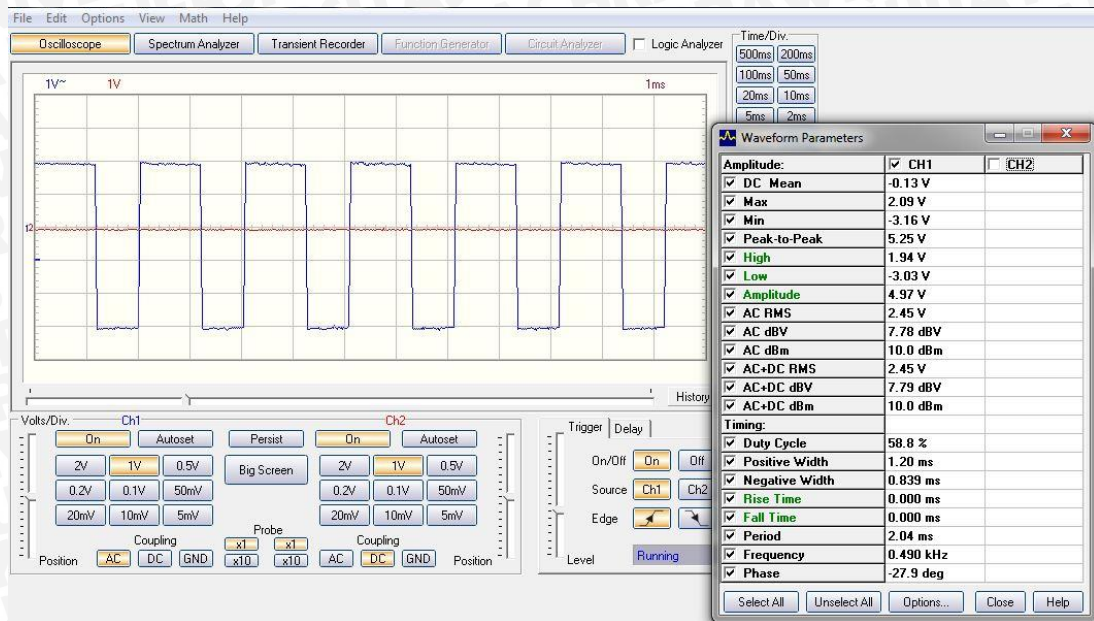


Gambar 5.2 Grafik sinyal PWM Arduino terhadap *duty cycle*

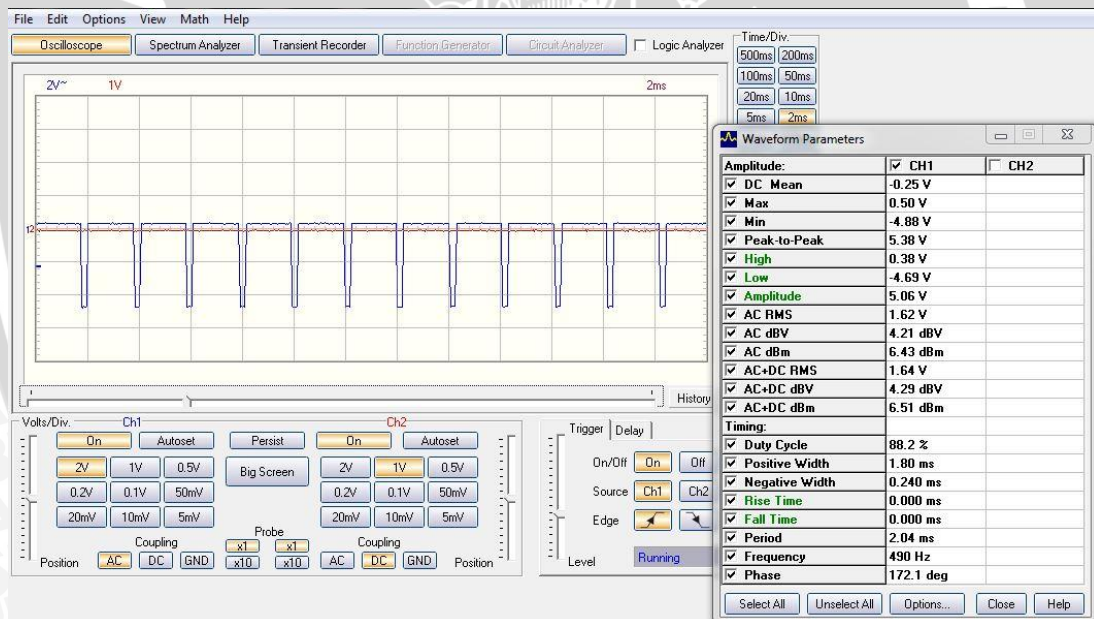
Hasil pengujian sinyal PWM Arduino Uno dengan sinyal input *duty cycle driver* motor dengan menggunakan Osiloskop Velleman PCSU1000 dan tampilan program *waveform* parameter dengan *duty cycle* 29,4%, 58,8% dan 88,2% ditunjukkan dalam Gambar 5.3, Gambar 5.4, Gambar 5.5.



Gambar 5.3 Sinyal PWM Arduino UNO dengan Sinyal *Duty Cycle* 29,4%



Gambar 5.4 Sinyal PWM Arduino UNO dengan Sinyal *Duty Cycle* 58,8%



Gambar 5.5 Sinyal PWM Arduino UNO dengan Sinyal *Duty Cycle* 88,2%

5.3 Pengujian *Driver* Motor EMS H-Bridge 5 A tanpa beban

a. Tujuan

Tujuannya adalah untuk mengetahui output dari driver motor apabila diberi input yang PWM berbeda-beda.

b. Peralatan yang digunakan

1. *Driver* EMS H-Bridge 5A.
2. Arduino Uno.

c. Langkah pengujian

- I. Menghubungkan masukan *driver* EMS H-Bridge 5 A ke Arduino Uno.
- II. Arduino Uno memberikan instruksi sinyal masukan pada pin PWM *driver* motor EMS H-Bridge 5 A dengan PWM 0 - 255.
- III. Mencatat hasil pengujian.

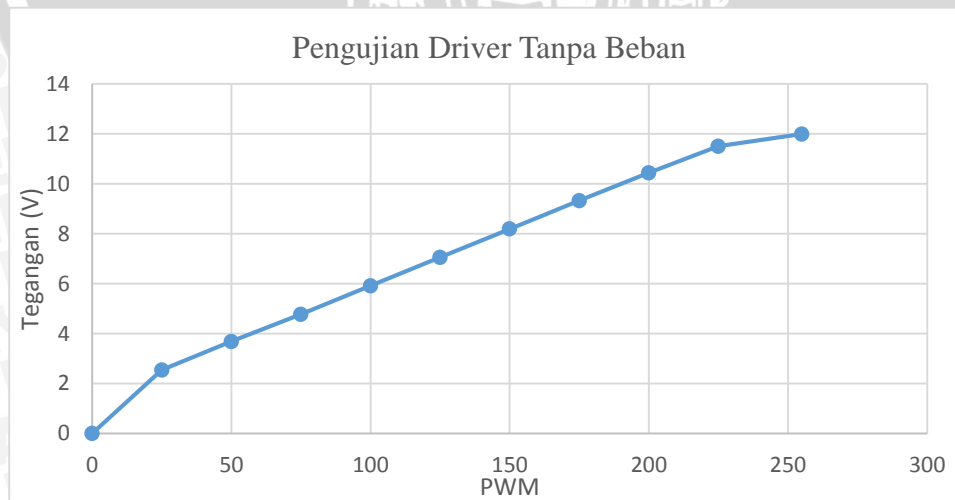
d. Hasil pengujian

Hasil pengujian *driver* dengan masukan PWM dapat dilihat dalam Tabel 5.3.

Tabel 5.3 Hasil Pengujian *Driver* EMS H-Bridge 5A

PWM	Tegangan Driver (volt)
0	0
25	2.54
50	3.68
75	4.77
100	5.91
125	7.05
150	8.19
175	9.32
200	10.44
225	11.5
255	11.99

Grafik hubungan antara PWM masukan dengan Tegangan keluaran driver EMS H-Bridge 5A dapat dilihat dalam Gambar 5.2.



Gambar 5.6 Grafik Tegangan Keluaran *Driver* tanpa beban

5.4 Pengujian Sensor Jarak HC-SR04

a. Tujuan

Mengetahui kemampuan sensor jarak dalam membaca jarak dan perubahan jarak.

b. Peralatan yang digunakan

1. Sensor Jarak HC-SR04.
2. Kabel Penghubung.
3. Arduino.
4. Komputer.
5. Penggaris.

c. Langkah pengujian

I. Merangkai peralatan sensor jarak, Arduino dan Komputer.

II. Memberikan penghalang yang berubah-ubah jaraknya.

III. Bandingkan dengan penggaris.

IV. Pencatatan.

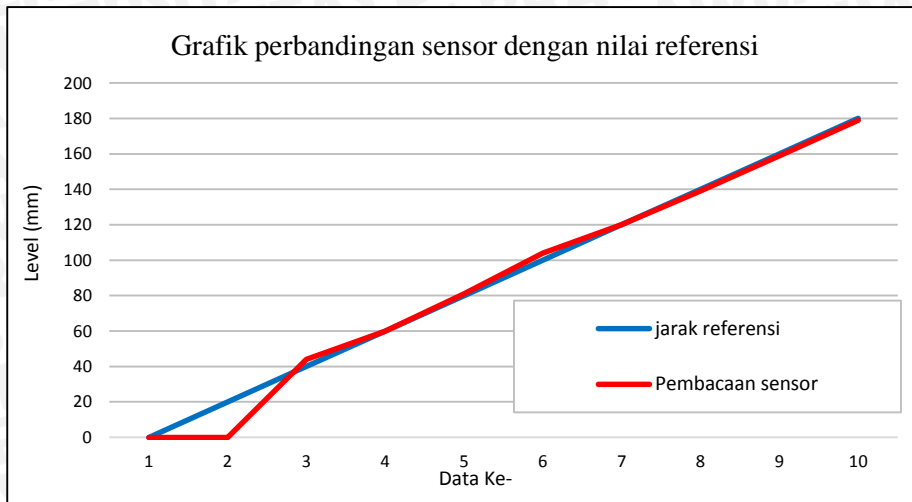
d. Hasil pengujian

Hasil pengujian *driver* dengan masukan PWM dapat dilihat dalam Tabel 5.3.

Tabel 5.4 Pengujian Sensor Jarak HC-SR04

Jarak Referensi (mm)	Pengukuran Sensor (mm)
0	0
20	0
40	44
60	60
80	81
100	104
120	120
140	139
160	159
180	179

Dari hasil pengujian yang dilakukan pada sensor jarak didapatkan grafik seperti pada Gambar 5.7.



Gambar 5.7 Grafik Pembacaan sensor dan nilai jarak referensi.

Perhitungan kesalahan:

$$\% \text{ kesalahan} = \frac{(\sum \text{Hasil Pengukuran} - \sum \text{Referensi})}{\sum \text{Referensi}} \times 100\%$$

$$\% \text{ kesalahan} = \frac{(886 - 900)}{900} \times 100\% = 1.55556\%$$

Dari hasil pengujian yang dilakukan pada sensor jarak dapat dilihat bahwa hasil pembacaan sensor jarak yang digunakan mampu mendeteksi perubahan jarak.

5.4 Pengujian Keseluruhan Sistem

a. Tujuan

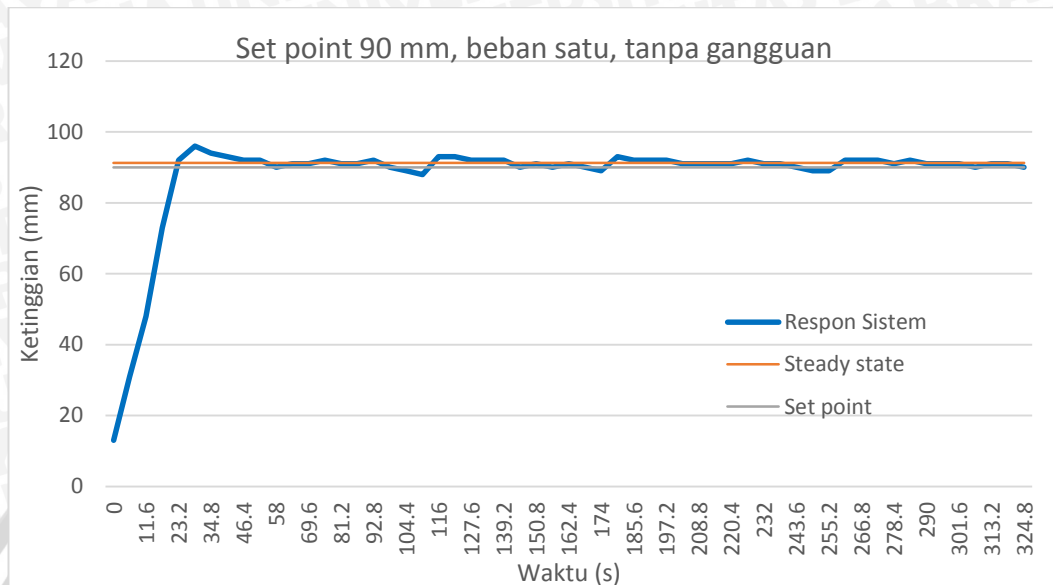
Pengujian keseluruhan sistem dilakukan untuk menguji apakah parameter yang sudah ditentukan dapat diaplikasikan pada alat dan sudah sesuai dengan *setpoint* yang diinginkan. Pada pengujian ini semua blok rangkaian dihubungkan setelah itu memasukkan nilai parameter yang telah ditentukan, serta mengamati dan menganalisa hasil kinerja alat.

b. Langkah Pengujian

- I. Memasukkan nilai parameter K_p , K_i dan K_d dari hasil perhitungan.
- II. Memasukkan *setpoint* yang diinginkan.
- III. Atur pembukaan katub keluaran air sebagai beban dan beban lebih selama 25 detik sebagai gangguan.
- IV. Mengamati dan menganalisa hasil dari kinerja sistem.

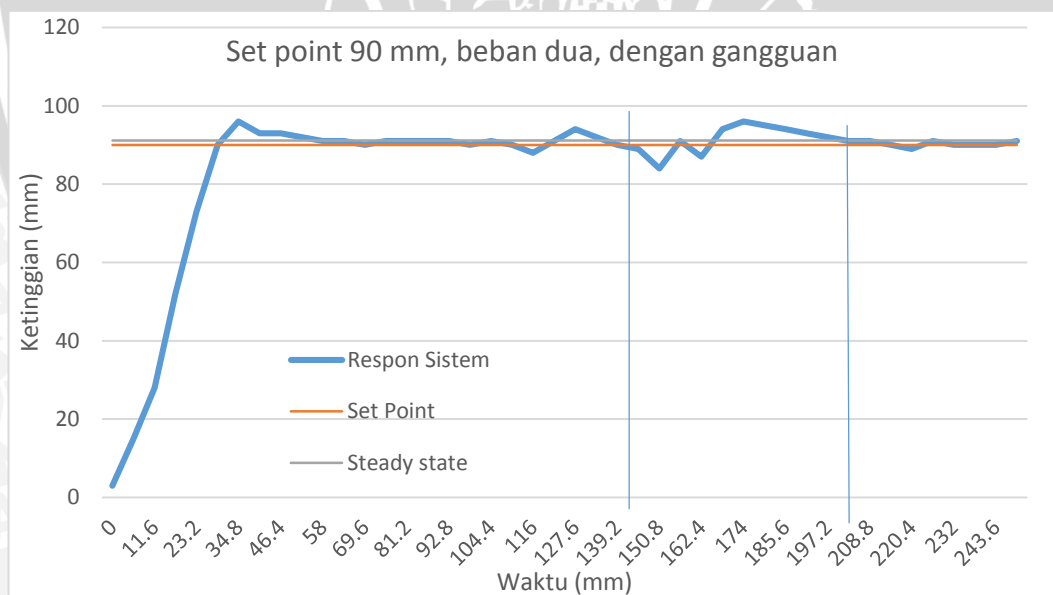
c. Hasil Pengujian

- Pengujian pada setpoint 90 mm beban satu, tanpa gangguan dalam Gambar 5.8 dan beban satu, dengan gangguan pada Gambar 5.9.



Gambar 5.8 Grafik *output plant* setpoint 90 mm, beban satu, tanpa gangguan.

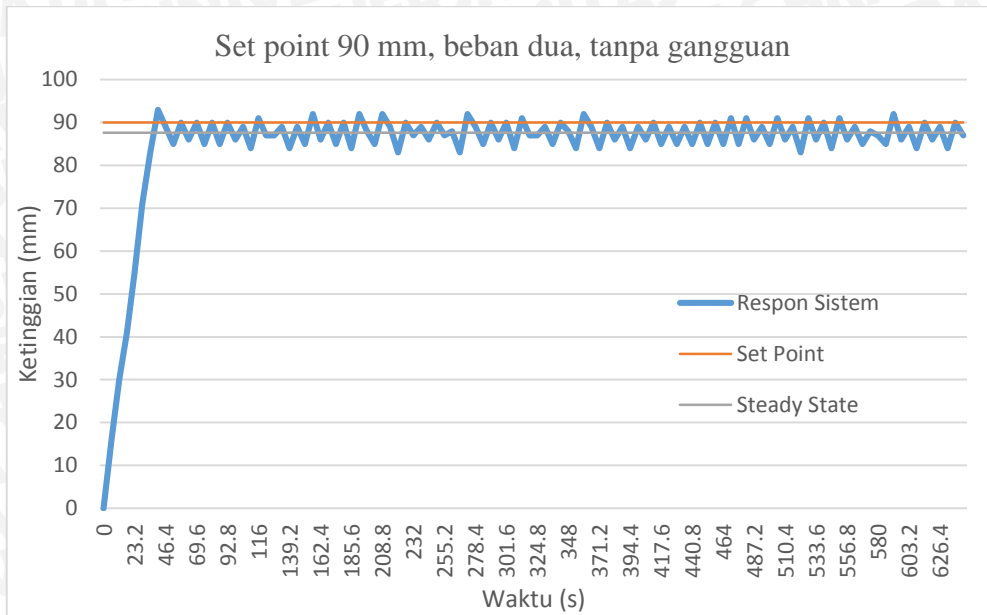
Dari grafik *output plant* dalam Gambar 5.8 dapat disimpulkan t_s selama 46,4 detik, $error\ steady\ state$ 1,6 %, t_r selama 23,2 detik, t_d selama 11,6 detik, t_p selama 29 detik dan M_p sebesar 6,07 %



Gambar 5.9 Grafik *output plant* setpoint 90 mm, beban satu, dengan gangguan.

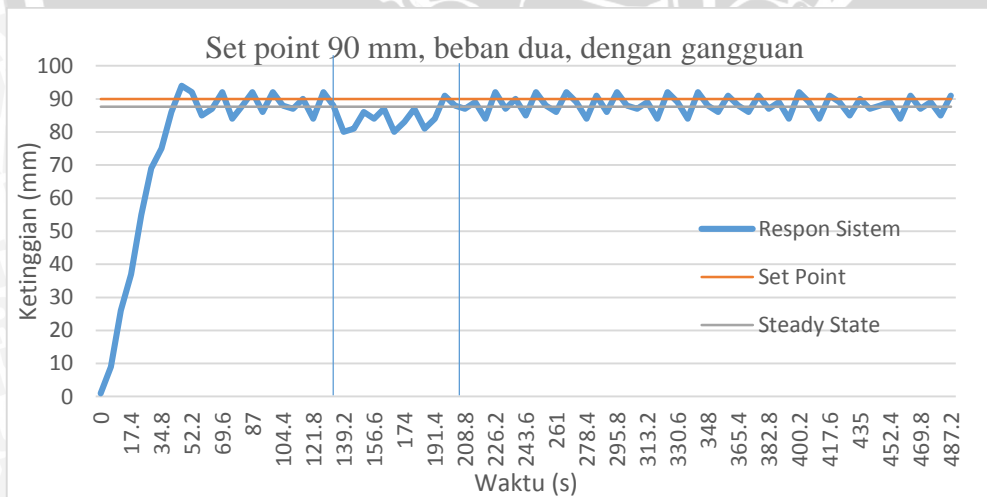
Dari grafik *output plant* dalam Gambar 5.9 *recovery timenya* saat ada gangguan yaitu 63,8 detik.

- Pengujian dengan setpoint 90 mm dengan beban dua tanpa gangguan dalam Gambar 5.10 dan dengan gangguan pada Gambar 5.11.



Gambar 5.10 Grafik *output plant* setpoint 90 mm, beban dua, tanpa gangguan.

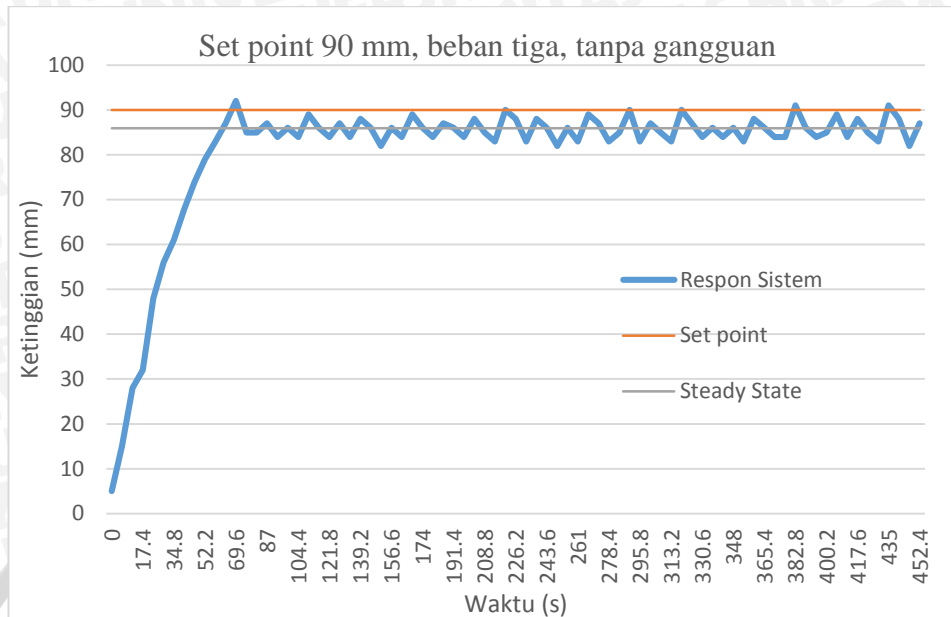
Dari grafik *output plant* dalam Gambar 5.10 dapat disimpulkan t_s selama 58 detik, $error\ steady\ state$ 2,6 %, t_r selama 34,8 detik, t_d selama 17,4 detik, t_p selama 40,6 detik dan M_p sebesar 6.09 %



Gambar 5.11 Grafik *output plant* setpoint 90 mm, beban dua, dengan gangguan.

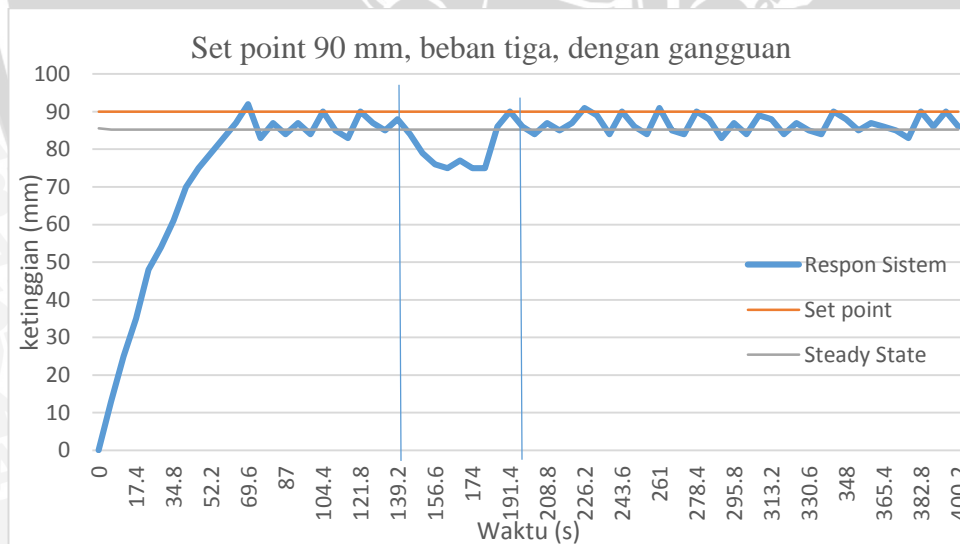
Dari grafik *output plant* dalam Gambar 5.11 *recovery timenya* saat ada gangguan yaitu 69,6 detik.

- Pengujian dengan setpoint 90 mm dengan beban tiga tanpa gangguan dalam Gambar 5.12 dan dengan gangguan pada Gambar 5.13.



Gambar 5.12 Grafik *output plant* setpoint 90 mm, beban tiga, tanpa gangguan

Dari grafik *output plant* dalam Gambar 5.12 dapat disimpulkan t_s selama 75,4 detik, $error\ steady\ state$ 4,8 %, t_r selama 63,8 detik, t_d selama 31,9 detik, t_p selama 69,6 detik dan M_p sebesar 7,10 %



Gambar 5.13 Grafik *output plant* setpoint 90 mm, beban tiga, dengan gangguan

Dari grafik *output plant* dalam Gambar 5.13 *recovery timenya* saat ada gangguan yaitu 58 detik.

