

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan dari pengujian dan analisis sistem adalah untuk menentukan apakah perancangan yang telah dilakukan pada bagian sebelumnya telah sesuai dengan tujuan perancangan. Pengujian pada sistem meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok dilakukan untuk menentukan letak kesalahan dan mempermudah analisis pada sistem keseluruhan apabila alat yang telah dirancang tidak bekerja sesuai dengan perancangan. Berikut ini dijabarkan langkah-langkah pengujian sistem.

5.1 Pengujian Sensor Temperatur DS18B20 Probe

A. Tujuan

Mengetahui kemampuan pembacaan sensor temperatur DS18B20 terhadap perubahan temperatur dengan melihat hasil pembacaan sensor yang ditampilkan dan menghitung *error* dari hasil pembacaan sensor temperatur DS18B20 dengan pembacaan termometer digital Krisbow KW06-308.

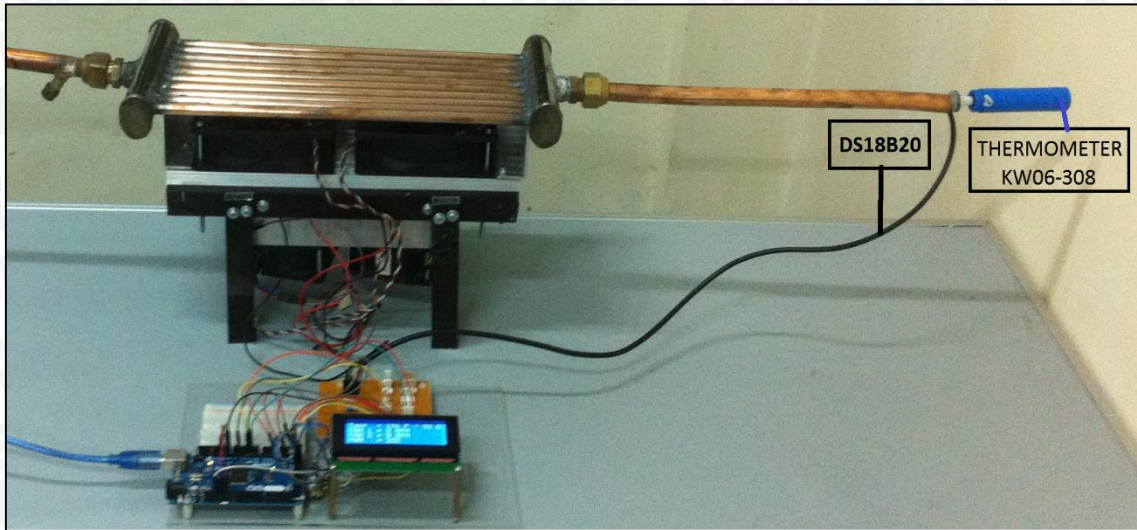
B. Peralatan Yang Digunakan

1. Sensor temperatur DS18B20 Probe.
2. Termometer digital Krisbow KW06-308.
3. Hot Jet WA-51.
4. Copper manifold.
5. Arduino Mega 2560 dan Komputer.

C. Langkah Pengujian

1. Merangkai peralatan seperti dalam Gambar 5.1.
2. Menghubungkan sensor temperatur DS18B20 dengan *pull-up* resistor 4,7K Ω dan Arduino Mega 2560 pada pin 2.
3. Mengaktifkan Hot Jet WA-51 dengan skala pemanas bernilai 1.
4. Merekam hasil pembacaan sensor DS18B20 terhadap hasil pembacaan termometer digital KW06-308.
5. Melakukan percobaan sebanyak 5 kali untuk menyimpulkan besaran *error* rata-rata pembacaan.





Gambar 5.1 Rangkaian Pengujian Sensor Temperatur DS18B20

D. Hasil Pengujian

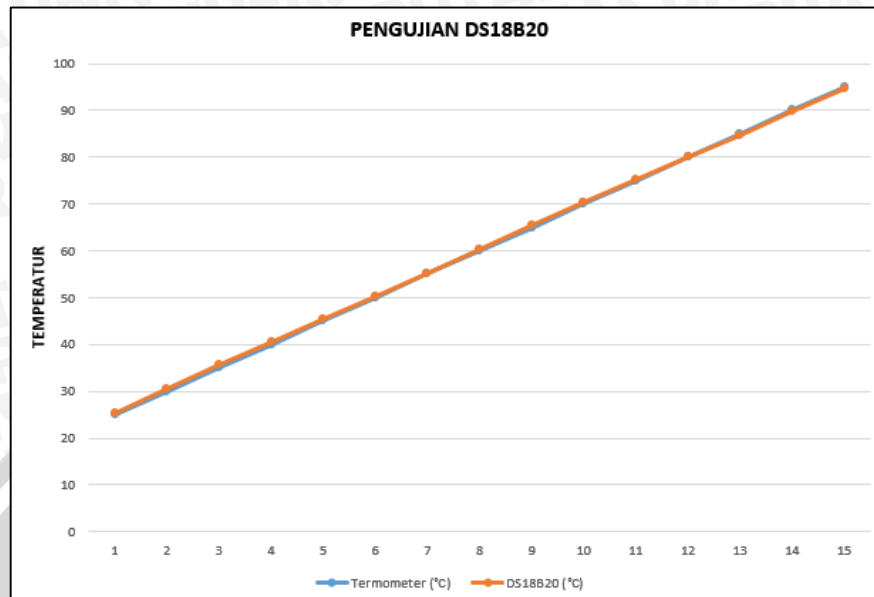
Data hasil pengujian sensor temperatur DS18B20 ditunjukkan dalam Tabel 5.1 berikut.

Tabel 5.1 Hasil Pengujian Sensor Temperatur DS18B20

No	Termometer (°C)	Pengujian Sensor Temperatur DS18B20 (°C)						
		1	2	3	4	5	Rata-rata	Error
1	25	25.25	25.44	25.37	25.37	25.44	25.374	0,374
2	30	30.75	30.44	30.37	30.44	30.25	30.45	0.45
3	35	35.31	35.94	35.75	35.5	35.31	35.562	0.562
4	40	40.25	40.48	40.56	40.38	40.38	40.41	0.42
5	45	45.06	45.38	45.56	45.38	45.25	45.326	0.326
6	50	50.31	50.38	50.19	50.44	50.31	50.326	0.326
7	55	55.13	55.19	55.25	55.38	55.31	55.252	0.252
8	60	60.13	60.31	60.69	60.13	60.25	60.302	0.302
9	65	65.06	65.19	65.37	65.81	65.5	65.386	0.386
10	70	70.19	70.12	70.5	70.62	70.31	70.348	0.348
11	75	75	75.06	75.25	75.06	75.19	75.112	0.112
12	80	79.69	80.12	80.13	80.12	80.37	80.086	0.086
13	85	84.28	84.6	84.82	84.52	85.82	84.684	0.316
14	90	89.12	89.8	89.24	89.08	90.32	89.758	0.242
15	95	94.13	94.65	94.96	95.13	95.24	94.822	0.187
Rata rata error (C⁰)								0.3013

Berdasarkan hasil pengujian dalam Tabel 5.1, *error* antara temperatur pembacaan termometer KW06-308 dengan temperatur rata-rata hasil pembacaan sensor temperatur DS18B20 yang dihubungkan dengan Arduino Mega 2560 adalah 0,3013°C.

Grafik perbandingan temperatur hasil pembacaan termometer KW06-308 dengan temperatur rata-rata sensor temperatur DS18B20 ditampilkan pada Gambar 5.2.



Gambar 5.2 Grafik Pengujian Laju Pertambahan Temperatur Terhadap Waktu

5.2 Pengujian Hot Jet WA-51

A. Tujuan

Mengetahui waktu yang dibutuhkan bagi Hot Jet WA-51 untuk mencapai temperatur masukan yang telah ditentukan yaitu 192°F.

B. Peralatan Yang Digunakan

1. Hot Jet WA-51.
2. Sensor Temperatur DS18B20 Probe.
3. Arduino Mega 2560 dan Komputer.
4. Timer.

C. Langkah Pengujian

1. Merangkai peralatan seperti dalam Gambar 5.3.
2. Menghubungkan Hot Jet WA-51 dengan sumber 220 VAC.
3. Menghubungkan sensor temperatur DS18B20 dengan Arduino Mega 2560.
4. Mengaktifkan Hot Jet WA-51 dan Arduino Mega 2560.
5. Merekam laju pergerakan perubahan temperatur dalam satuan sekon.



Gambar 5.3 Rangkaian Pengujian Hot Jet WA-51

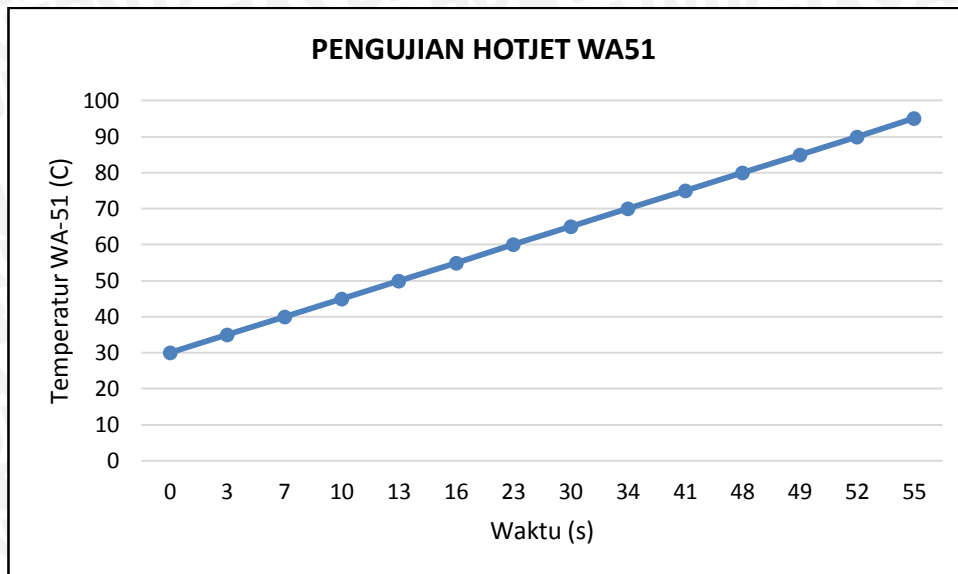
D. Hasil Pengujian

Data hasil pengujian *hot jet* WA-51 ditunjukkan dalam Tabel 5.2 berikut.

Tabel 5.2 Hasil Pengujian Hot Jet WA-51

No	Temperature (°C)	Waktu (s)
1	25	0
2	30	3,23
3	35	6,6
4	40	10,12
5	45	13,04
6	50	16,45
7	55	23,02
8	60	29,56
9	65	34,31
10	70	41,28
11	75	47,87
12	80	48,8
13	85	51,52
14	90	55,35
15	95	58,71

Grafik laju pergerakan temperatur keluaran Hot Jet WA-51 terhadap waktu dalam sekon ditunjukkan pada Gambar 5.4.



Gambar 5.4 Grafik Laju Pergerakan Temperatur Hot Jet WA-51 Terhadap Waktu

Berdasarkan hasil pengujian dalam Tabel 5.2 dapat diketahui bahwa Hot Jet WA-51 dapat menghasilkan udara yang sudah terkompresi dengan suhu sesuai *setpoint* 192°F (89°C) dalam waktu sekitar 54 sekon yang selanjutnya digunakan sebagai *input* pada *manifold*.

5.3 Pengujian *Driver* DC Fan Delta AFC1212DE

A. Tujuan

Pengujian ini bertujuan untuk mengetahui kinerja dan respon dari rangkaian *driver* DC Fan Delta AFC1212DE dengan membandingkan dan menguji sinyal respon dari *driver* motor terhadap sinyal masukan PWM yang diberikan oleh mikrokontroler Arduino Mega 2560.

B. Peralatan Yang Digunakan

1. DC Fan Delta AFC1212DE.
2. Arduino Mega 2560.
3. Multimeter DT-830B.

C. Langkah Pengujian

1. Menghubungkan masukan *driver* DC Fan Delta AFC1212DE menuju pin PWM Arduino Mega 2560.
2. Memprogram Arduino Mega 2560 agar memberikan sinyal PWM mulai dari 0 hingga 255.
3. Mencatat besaran tegangan masukan dan keluaran *driver* DC Fan Delta AFC1212DE.

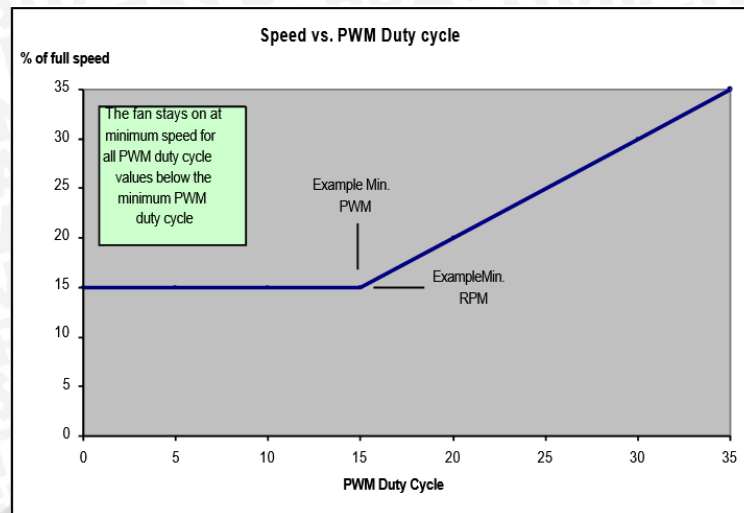
D. Hasil Pengujian

Besar perbandingan antara tegangan masukan *driver* dengan tegangan keluaran *driver* dapat dilihat dalam Tabel 5.3.

Tabel 5.3 Hasil Pengujian *Driver* DC Fan Delta AFC1212DE

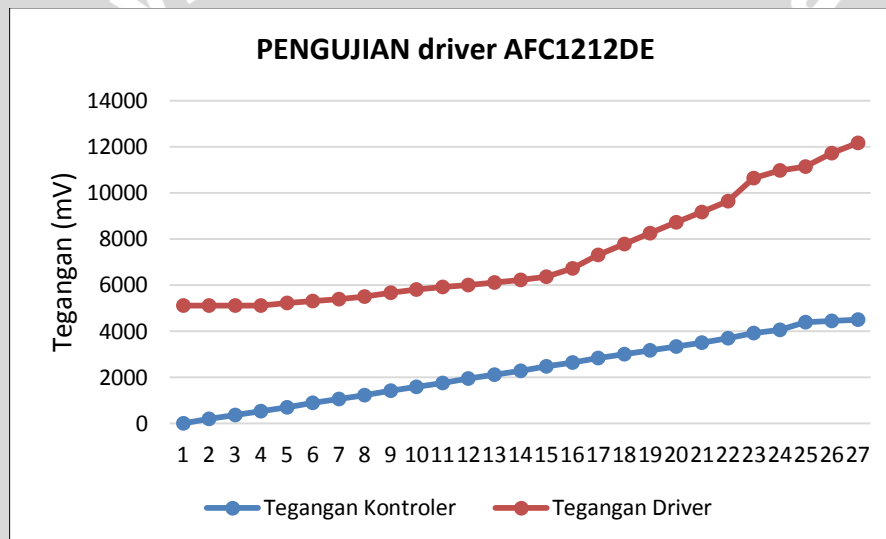
No	PWM	Tegangan Arduino (mV)	Tegangan Driver (mV)
1	0	0	5120
2	10	192.6	5120
3	20	367.4	5120
4	30	530	5120
5	40	707	5210
6	50	880	5300
7	60	1059	5400
8	70	1231	5490
9	80	1405	5670
10	90	1584	5800
11	100	1756	5910
12	110	1936	6010
13	120	2111	6120
14	130	2283	6230
15	140	2464	6370
16	150	2637	6720
17	160	2825	7310
18	170	2991	7760
19	180	3171	8240
20	190	3342	8720
21	200	3510	9170
22	210	3703	9640
23	220	3912	10630
24	230	4055	10960
25	240	4380	11130
26	250	4452	11710
27	255	4489	12150

Berdasarkan Tabel 5.3 dapat disimpulkan *driver* DC Fan Delta AFC1212DE sudah memberikan tegangan sebesar 5,12 V ketika nilai PWM masukan *driver* dibawah 40 Hal ini merupakan karakteristik khusus yang hanya dimiliki oleh DC Fan yang mempunyai masukan pin PWM seperti yang ditunjukkan pada Gambar 5.5. Grafik hubungan antara tegangan masukan dengan tegangan keluaran *driver* Delta AFC1212DE dapat dilihat dalam Gambar 5.6.



Gambar 5.5 Grafik Karakteristik PWM Driver Delta AFC1212DE

Sumber: 4 Wire PWM Controlled Fan Rev.1.2



Gambar 5.6 Grafik Hubungan Tegangan Kontroler Dan Driver Delta AFC1212DE

5.4 Pengujian DC Fan Delta AFC1212DE

A. Tujuan

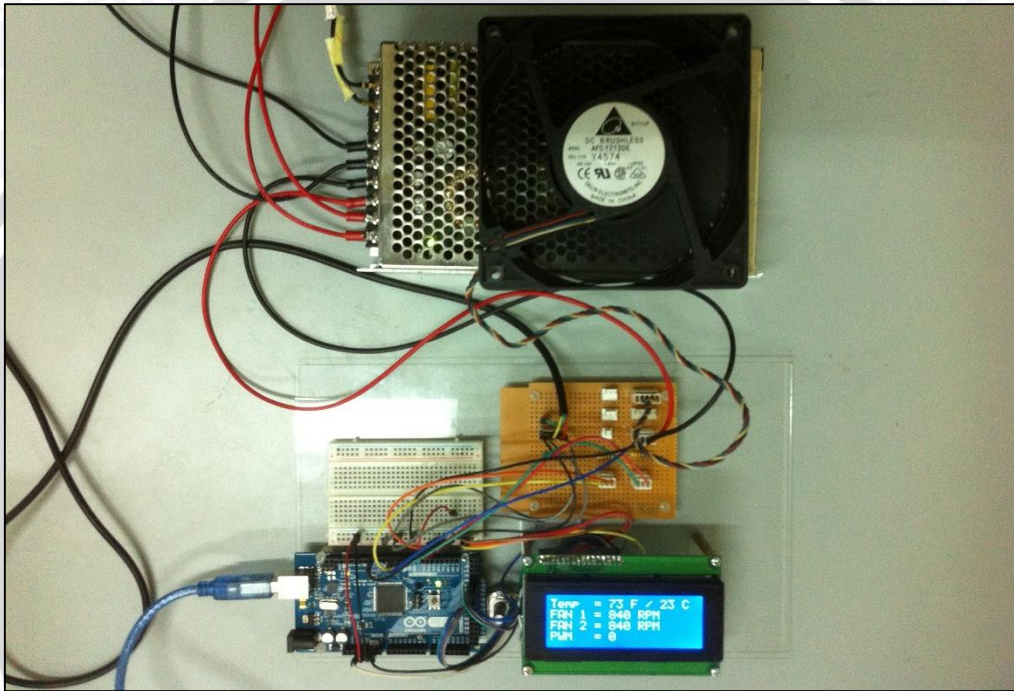
Pengujian ini bertujuan untuk mengetahui besar nilai kecepatan rotasi permenit (RPM) terhadap nilai PWM sebagai masukan pada DC Fan Delta AFC1212DE.

B. Peralatan Yang Digunakan

1. DC Fan Delta AFC1212DE.
2. Arduino Mega 2560 dan Komputer.
3. Rangkaian *pull-up* resistor bagi keluaran *tachometer* menuju Arduino Mega 2560.

C. Langkah Pengujian

1. Merangkai peralatan seperti pada Gambar 5.7.
2. Menghubungkan DC Fan Delta AFC1212DE menuju catu daya 12VDC
3. Menjalankan program PWM pada program Arduino 1.0.6 untuk mengaktifkan *driver* Delta AFC1212DE agar bekerja.
4. Mencatat besar perubahan kecepatan rotasi permenit (RPM) terhadap nilai PWM.



Gambar 5.7 Rangkaian Membaca RPM Terhadap PWM

D. Hasil Pengujian

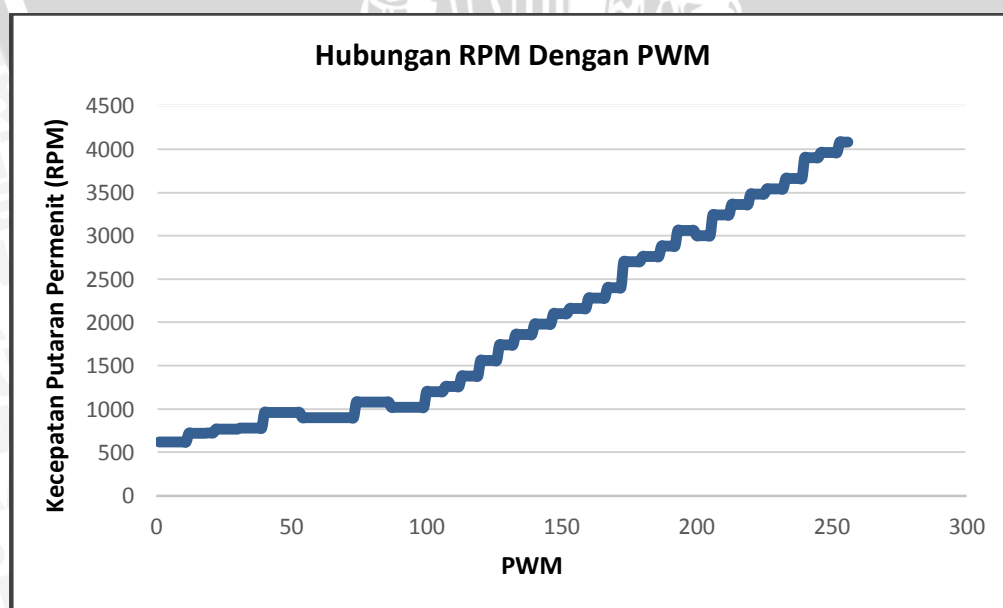
Data hasil pengujian DC Fan Delta AFC1212DE ditunjukkan dalam Tabel 5.4 berikut

Tabel 5.4 Perbandingan Nilai PWM Terhadap RPM DC Fan Delta AFC1212DE

No	PWM	Kecepatan (RPM)
1	0	780
2	10	780
3	20	780
4	30	780
5	40	910
6	50	930
7	60	960
8	70	995

9	80	1030
10	90	1080
11	100	1160
12	110	1260
13	120	1560
14	130	1740
15	140	1980
16	150	2100
17	160	2280
18	170	2400
19	180	2760
20	190	2880
21	200	3000
22	210	3240
23	220	3480
24	230	3540
25	240	3780
26	250	3930
27	255	4020

Karakteristik khusus yang dimiliki oleh DC Fan Delta AFC1212DE adalah bahwa DC Fan sudah bekerja walau masukan sinyal PWM bernilai 0. Hal ini dijelaskan pada bagian 5.3 yang membahas mengenai *driver* DC Fan AFC1212DE, dimana pada *duty cycle* dibawah 15%, maka DC Fan sudah berputar dengan kecepatan sebesar 15% dari kecepatan maksimal. Grafik hubungan antara nilai PWM sebagai masukan *driver* DC Fan AFC1212DE terhadap kecepatan rotasi permenit (RPM) ditunjukkan pada Gambar 5.8.



Gambar 5.8 Grafik Hubungan Nilai PWM Dengan RPM DC Fan Delta AFC1212DE

5.5 Pengujian Tanpa Kontroler

A. Tujuan

Pengujian tanpa kontroler bertujuan untuk mengetahui bagaimana perbedaan respon sistem terhadap *setpoint* ketika tanpa kontroler dibandingkan dengan menggunakan kontroler PID, sehingga dapat ditentukan bahwa sistem memerlukan sebuah sistem pengontrolan.

B. Peralatan Yang Digunakan

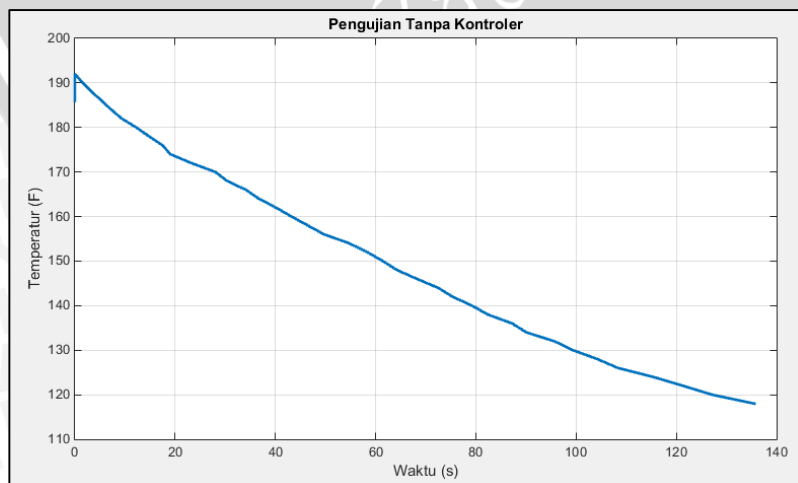
1. Hot Jet WA-51.
2. DC Fan AFC1212DE.
3. Sensor temperatur DS18B20
4. Arduino Mega 2560 Dan Komputer.
5. Rangkaian elektrik yang terdiri dari *pull up* resistor 10 K Ω untuk *tachometer* dan *pull up* resistor 4,7K Ω untuk sensor temperatur DS18B20.
6. Catu daya 220VAC, 12VDC, Dan 5VDC.

C. Langkah Pengujian

1. Merangkai peralatan seperti pada Gambar 5.9.
2. Mengaktifkan Hot Jet WA-51 dengan skala 1.
3. Ketika temperatur keluaran Hot Jet WA-51 mencapai *setpoint* kerja, yaitu 192°F, maka DC Fan Delta AFC1212DE bekerja menurunkan temperatur.
4. Mencatat laju penurunan temperatur terhadap waktu (sekon).

D. Hasil Pengujian

Pada pengujian tanpa kontroler PID, sistem dapat menurunkan temperatur menjadi 118°F selama kurun waktu 136 sekon. Dibutuhkan sebuah kontroler PID untuk dapat menjaga temperatur sesuai *setpoint*, yaitu 125°F.



Gambar 5.9 Grafik Pengujian Tanpa Kontroler

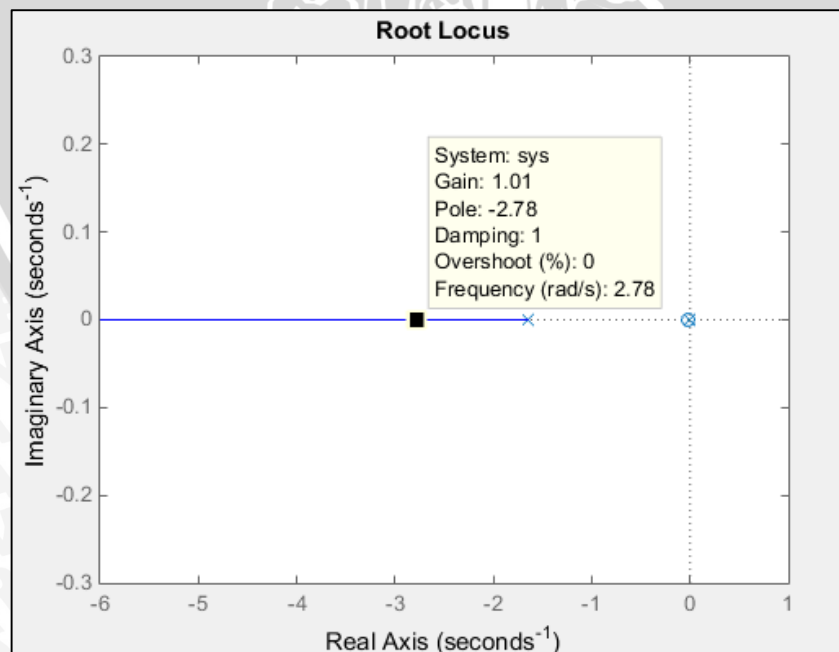
5.6 Pengujian Keseluruhan Sistem

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.

Langkah – langkah pengujian keseluruhan adalah sebagai berikut :

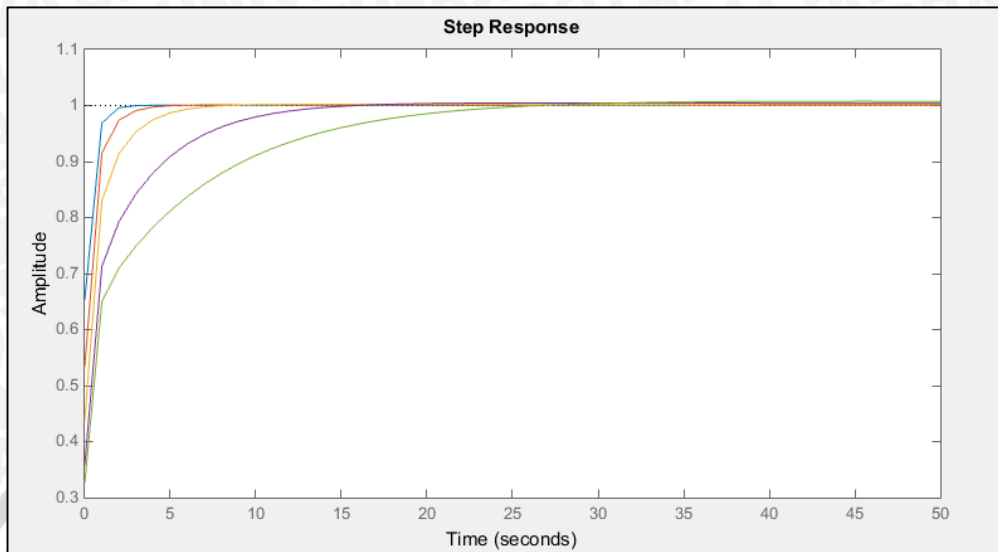
1. Memasukkan nilai parameter K_p , K_i , dan K_d dari hasil perhitungan.
2. Memasukkan *setpoint* yang diinginkan.
3. Mengamati dan menganalisa hasil dari kinerja sistem.

Penentuan pole dari fungsi alih sistem *loop* terbuka dilakukan dengan verifikasi model sistem. Verifikasi dilakukan untuk mengetahui apakah model sistem yang telah disusun menghasilkan respon sistem yang memenuhi kriteria kestabilan mutlak. Dalam hal ini model sistem akan diuji dengan memberikan masukan berupa unit *step*, selain itu akar-akar *loop* dari sistem ditampilkan dalam *root locus* diagram. Akar – akar *loop* terbuka sistem ditunjukkan dalam diagram *root locus* seperti terlihat dalam Gambar 5.10. Dapat dilihat letak *pole – pole* sistem dalam Gambar 5.10 ada pada sebelah kiri bidang s sehingga memenuhi syarat kestabilan. Langkah selanjutnya adalah penentuan parameter PID dengan memasukkan s_1 dalam Persamaan 2.2 dengan memodifikasi nilai K_i yang diinginkan dihasilkan respon dalam Gambar 5.11.



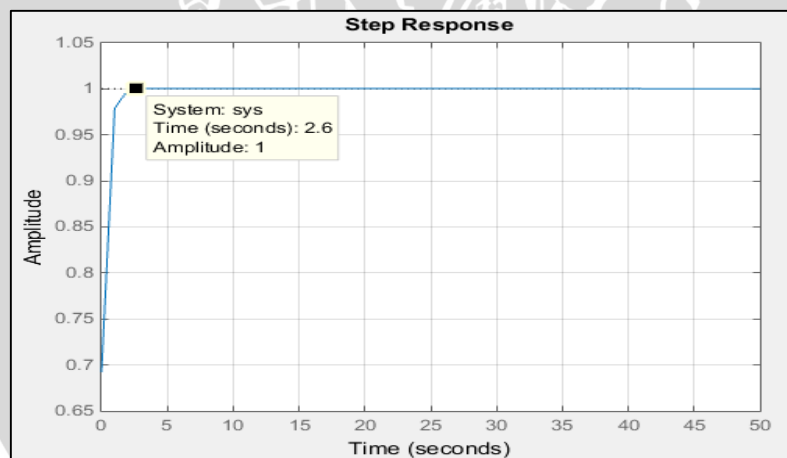
Gambar 5.10 Diagram *Root Locus* Tanpa Kontroler

Berdasarkan Gambar 5.11 respon sistem dengan $K_p = 14.19$ $K_i = 20$ dan $K_d = 2.59$ memiliki *error* terkecil dengan *ts* tercepat.



Gambar 5.11 Respon Sistem Berdasarkan Parameter PID pada Tabel 4.3

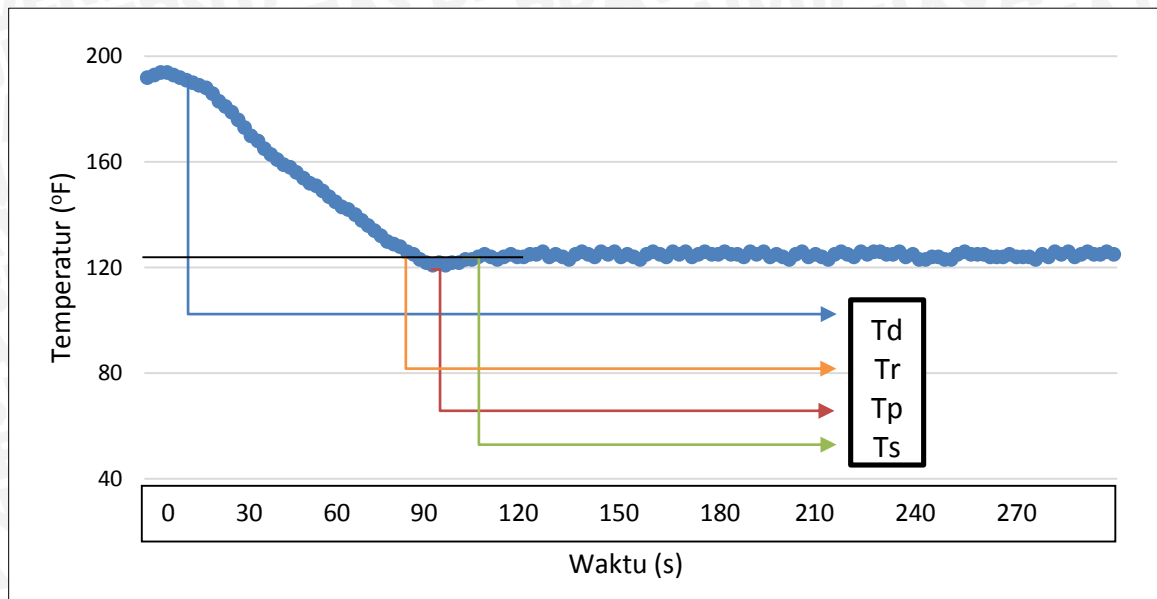
Gambar 5.12 menunjukkan keluaran respon dengan kontroler menggunakan simulasi Matlab R2014b. Masukan berupa unit *step*. Dari hasil simulasi respon yang dihasilkan sesuai dengan *setpoint* yang diinginkan.



Gambar 5.12 Grafik Respon Sistem dengan $K_p = 14.19$ $K_i = 20$ dan $K_d = 2.59$

5.6.1 Pengujian Pada Temperatur Ambient 25°C

Pada pengujian dengan temperatur *ambient* sebesar 25°C (77°F) sebagai simulasi temperatur pada malam hari, parameter Kontroler PID hasil perhitungan pada bab 4 kemudian diimplementasikan pada sistem dan didapatkan pembacaan temperatur terhadap waktu seperti yang ditunjukkan pada pada Gambar 5.13.



Gambar 5.13 Grafik Pengujian Tanpa *Disturbance* Pada Temperatur Ambient 25°C

$$F(s) = \frac{1.136 s + 0.01513}{s^2 + 1.65 s + 0.01431} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

$$2\xi\omega_n = 1.65$$

$$\omega_n = \sqrt{0.01431} = 0.11962$$

$$\xi = \frac{1.65}{2 \times 0.11962} = 6.8965$$

Delay Time (Td) = 9 s

Rise Time (Tr) = 126 s

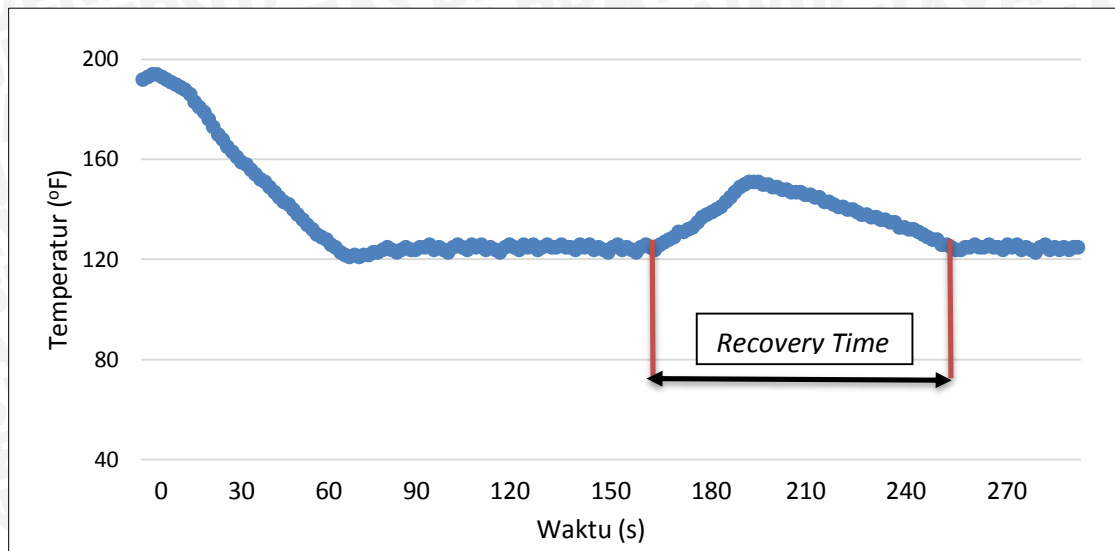
Peak Time (Tp) = 130 s

Settling Time (Ts) = 140 s

$$\text{Maximum Overshoot (Mp)} = \left| \frac{122 - 125}{125} \right| \times 100\%$$

$$= 0.024 \times 100\%$$

$$= 2.4 \%$$

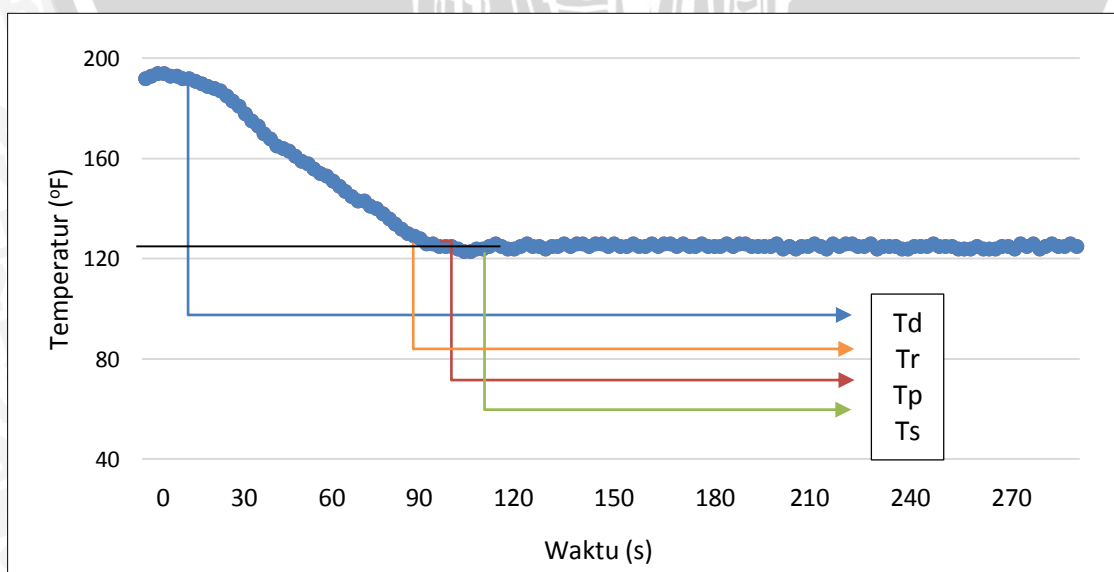


Gambar 5.14. Grafik Pengujian Dengan *Disturbance* 15°F Pada Temperatur *Ambient* 25°C

Pada Gambar 5.14 menunjukkan kinerja sistem pada temperatur *ambient* sebesar 25°C saat diberi gangguan berupa peningkatan temperatur sebesar 15°F. Sistem dengan nilai parameter $K_p = 14.19$ $K_i = 20$ dan $K_d = 2.59$ dapat mempertahankan temperatur sistem setelah mengalami *recovery time* selama 63 detik.

5.6.2 Pengujian Pada Temperatur *Ambient* 35°C

Sama dengan pengujian sebelumnya, pada pengujian dengan temperatur *ambient* sebesar 35°C (95°F) sebagai simulasi temperatur pada siang hari, parameter Kontroler PID hasil perhitungan pada bab 4 kemudian diimplementasikan pada sistem dan didapatkan pembacaan temperatur terhadap waktu seperti yang ditunjukkan pada pada Gambar 5.15.



Gambar 5.15 Grafik Pengujian Tanpa *Disturbance* Pada Temperatur *Ambient* 35°C

$$F(s) = \frac{1.136 s + 0.01513}{s^2 + 1.65 s + 0.01431} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

$$2\xi\omega_n = 1.65$$

$$\omega_n = \sqrt{0.01431} = 0.11962$$

$$\xi = \frac{1.65}{2 \times 0.11962} = 6.8965$$

$$\text{Delay Time (Td)} = 15 \text{ s}$$

$$\text{Rise Time (Tr)} = 133 \text{ s}$$

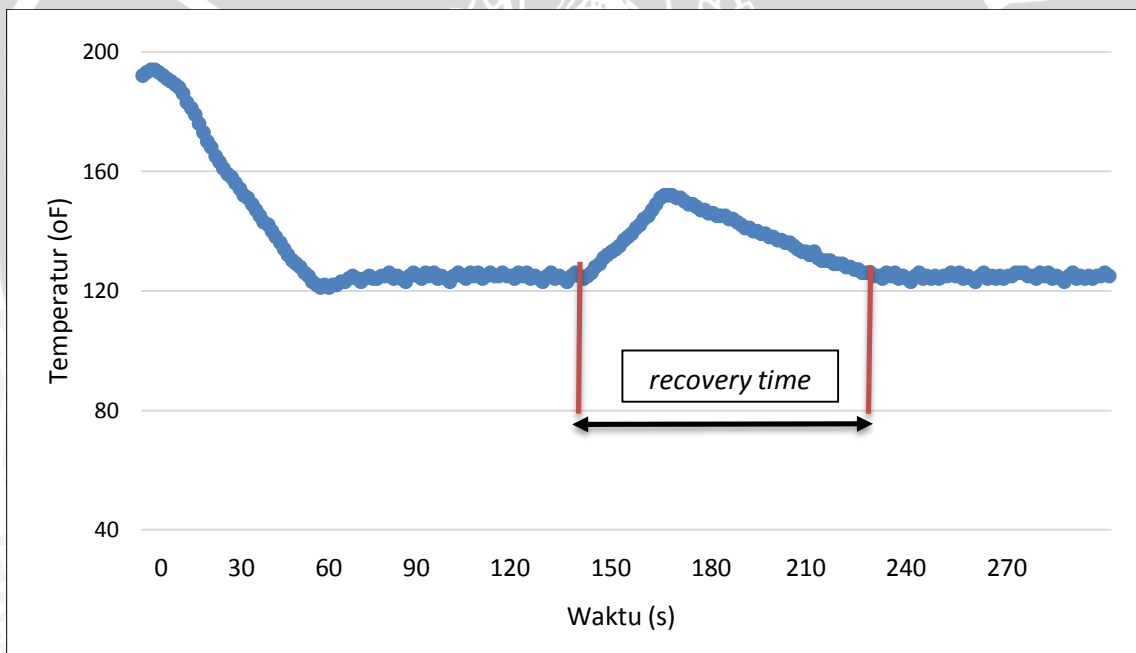
$$\text{Peak Time (Tp)} = 138 \text{ s}$$

$$\text{Settling Time (Ts)} = 144 \text{ s}$$

$$\text{Maximum Overshoot (Mp)} = \left| \frac{123 - 125}{125} \right| \times 100\%$$

$$= 0.016 \times 100\%$$

$$= 1.6 \%$$



Gambar 5.16 Grafik Pengujian Dengan *Disturbance* 15°F Pada Temperatur Ambient 35°C

Pada Gambar 5.16 menunjukkan kinerja sistem pada temperatur *ambient* 35°C saat diberi gangguan berupa peningkatan temperatur sebesar 15°F. Sistem dengan nilai parameter $K_p = 14.19$ $K_i = 20$ dan $K_d = 2.59$ dapat mempertahankan temperatur sistem setelah mengalami *recovery time* selama 75 sekond.