

ALAT UKUR SUHU MENGGUNAKAN SENSOR THERMISTOR NTC BERBASIS MIKROKONTROLER ARDUINO UNO

Untuk memenuhi tugas *Research Based Learning* (RBL) mata kuliah FI2271 Sistem Instrumentasi



Anggi Maulana Adi S.	10217040
Ahmad Al Ghiffari	10217047
Efraim Partogi Nahotasi	10217048

Program Studi Fisika
Fakultas Matematika dan Ilmu Pengetahuan Alam Institut
Teknologi Bandung
2018/2019

Abstrak

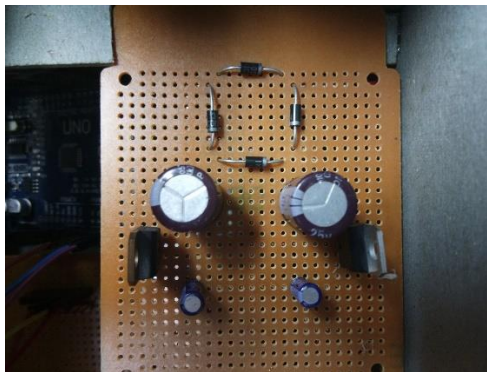
Laporan Research Based Learning ini disusun dengan tujuan untuk menentukan rangkaian elektronika sistem kendali pemanas air beserta komponen yang menyusunnya. Metode yang digunakan dalam RBL ini memanfaatkan sensitifitas sensor thermistor NTC 10k terhadap perubahan suhu lingkungan disekitarnya. Selanjutnya didesain pengondisi sinyal sehingga menghasilkan keluaran 0-5V lalu dikonversi ke dalam sistem digital oleh mikrokontroler Arduino Uno. Sistem kontrol suhu menggunakan thermistor sebagai umpan balik dan heater sebagai plant system tersebut. Hasil dari percobaan RBL ini didapatkannya rangkaian yang bekerja dengan menyalakan heater sesuai dengan masukan suhu yang diberikan.

Kata Kunci: Heater, Mikrokontroler, Pengondisi sinyal, PID, Termometer, Thermistor

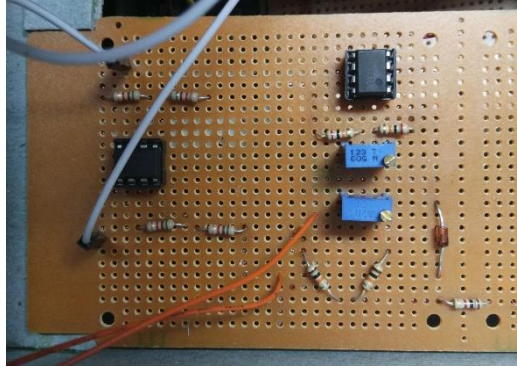
RANCANGAN BANGUN RANGKAIAN PENGONDISI SINYAL

Akan didesain dan dibangun alat ukur suhu dengan rentang suhu diantara 30 - 60 °C dan diperlukan sinyal tegangan keluaran dengan rentang 0 - 5V untuk dikonversi menjadi sinyal digital lalu diolah oleh mikrokontroler Arduino Uno. Untuk mendeteksi perubahan suhu lingkungan digunakan sensor thermistor NTC 10k yang dirangkai menjadi rangkaian jembatan dengan hambatan lainnya dengan nilai tertentu. Supply ke rangkaian jembatan ini diberikan oleh rangkaian pembagi tegangan antara sebuah resistor dan dioda zener 5.1V. Nilai resistor pada rangkaian jembatan dibuat sedemikian sehingga dihasilkan beda tegangan keluaran rangkaian jembatan sebesar 0 V pada suhu 30 °C. Untuk menguatkan sinyal keluaran dari rangkaian jembatan, digunakan penguat instrumentasi yang masuk ke mikrokontroler.

Pengolahan digital dilakukan oleh Arduino Uno. Dilakukan konversi sinyal analog ke sinyal digital (ADC) yang kemudian diolah untuk ditampilkan ke LCD; kontrol PID. PID diperlukan untuk mengatur daya pada *plant* pemanas (*heater*) sehingga suhu lingkungan (dalam hal ini air) dapat dipertahankan pada nilai titik acu (*set point*). Dalam kontrol suhu ini, mikrokontroler mendapatkan umpan balik (*feedback*) dari sensor.



Gambar 1. Rangkaian Power Supply CT



Gambar 2. Rangkaian Pengondisi Sinyal

MODEL MATEMATIS RANGKAIAN SIGNAL CONDITIONING

Dalam rangkaian signal conditioning digunakan rangkaian jembatan dan penguat instrumentasi. Persamaan matematis pada rangkaian jembatan sebagai berikut.

$$V_b = \frac{RPOTE}{RPOTE + R_2} V_{ref} \quad (1)$$

$$V_a = \frac{R_T}{R_T + R_3} V_{ref} \quad (2)$$

Dengan $R_2 = R_3$ dan mensubstitusikan persamaan 1 dan 2, diperoleh

$$V_{ab} = V_a - V_b = \left(\frac{R_T}{R_T + R_2} - \frac{RPOTE}{RPOTE + R_2} \right) V_{ref} \quad (a)$$

Pada penguat Instrumentasi, persamaan matematis seperti berikut.

$$I_1 = \frac{V_1 - V_a}{R_4} \quad (3)$$

$$I_2 = \frac{V_a - V_b}{RPOT} \quad (4)$$

$$I_3 = \frac{V_b - V_2}{R_5} \quad (5)$$

Dari persamaan tersebut diperoleh $I_1 = I_2 = I_3$, sehingga

$$I_1 = I_2 = I_3 = I = \frac{V_a - V_b}{RPOT} \quad (6)$$

Karena tidak ada arus yg masuk ke Op-Amp dan $R_4 = R_5$, maka

$$I = \frac{V_1 - V_2}{2R_4 + RPOT} \quad (7)$$

Substitusikan persamaan 6 dan 7, maka diperoleh

$$V_1 - V_2 = \frac{(2R_4 + RPOT)}{RPOT} (V_a - V_b)$$

$$V_1 - V_2 = \left(\frac{2R_4}{RPOT} + 1 \right) (V_a - V_b) \quad (8)$$

Karena $R4 = R5$, maka dapat diperoleh

$$V_o = I (2R4 + RPOT) \quad (9)$$

Dengan melakukan substitusi pada persamaan 3 dan 4, diperoleh

$$V_o = \frac{V_{ab}}{RPOT} (2R4 + RPOT) \quad (10)$$

Kemudian $R6 = R7 = R8 = R9$ dan pembagi tegangan, sehingga

$$(V+) = (V-) = \frac{R6}{R6 + R6} V2 = \frac{V2}{2} \quad (11)$$

$$I4 = \frac{V1 - (V-)}{R6} = \frac{V1}{R6} - \frac{V2}{2R6} \quad (12)$$

$$I5 = \frac{(V-) - V_o}{R6} \quad (13)$$

Substitusikan persamaan 13 dan 11, diperoleh

$$I5 = \frac{V2}{2R6} - \frac{V_o}{R6} \quad (14)$$

Kemudian karena $I4 = I5$, maka

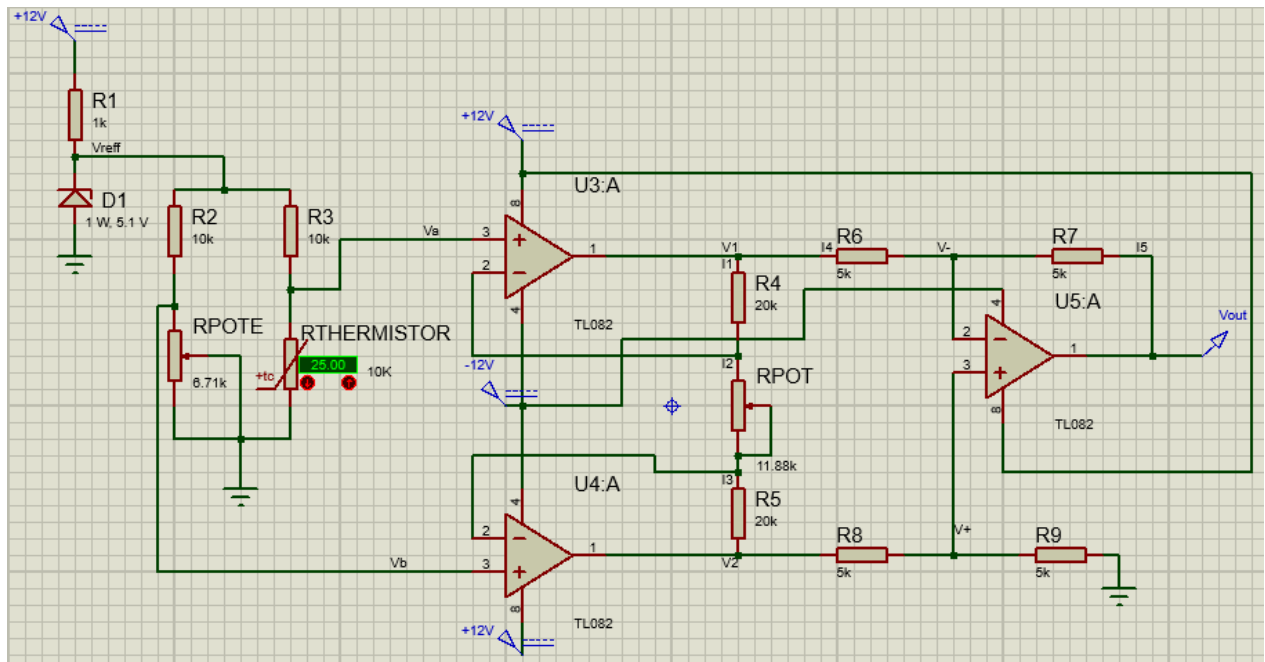
$$\frac{V1}{R6} - \frac{V2}{2R6} = \frac{V2}{2R6} - \frac{V_o}{R6} \quad (15)$$

Selanjutnya substitusikan persamaan 15 dan 8, maka diperoleh

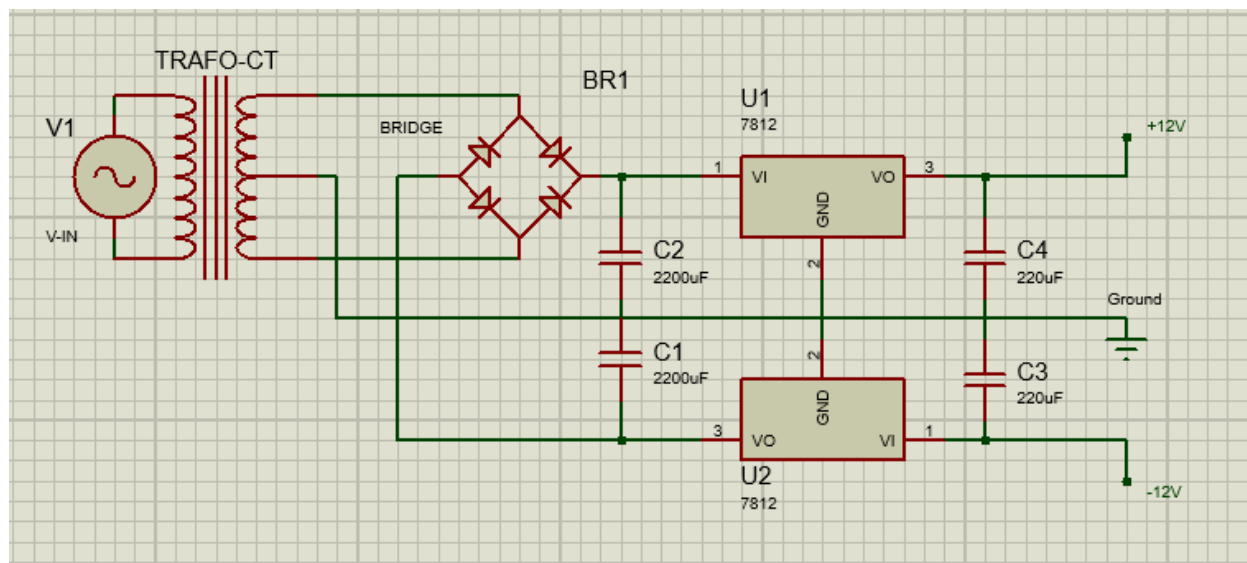
$$V_o = \left(\frac{2R4}{RPOT} + 1 \right) (Vb - Va) \quad (b)$$

SKEMATIK RANGKAIAN

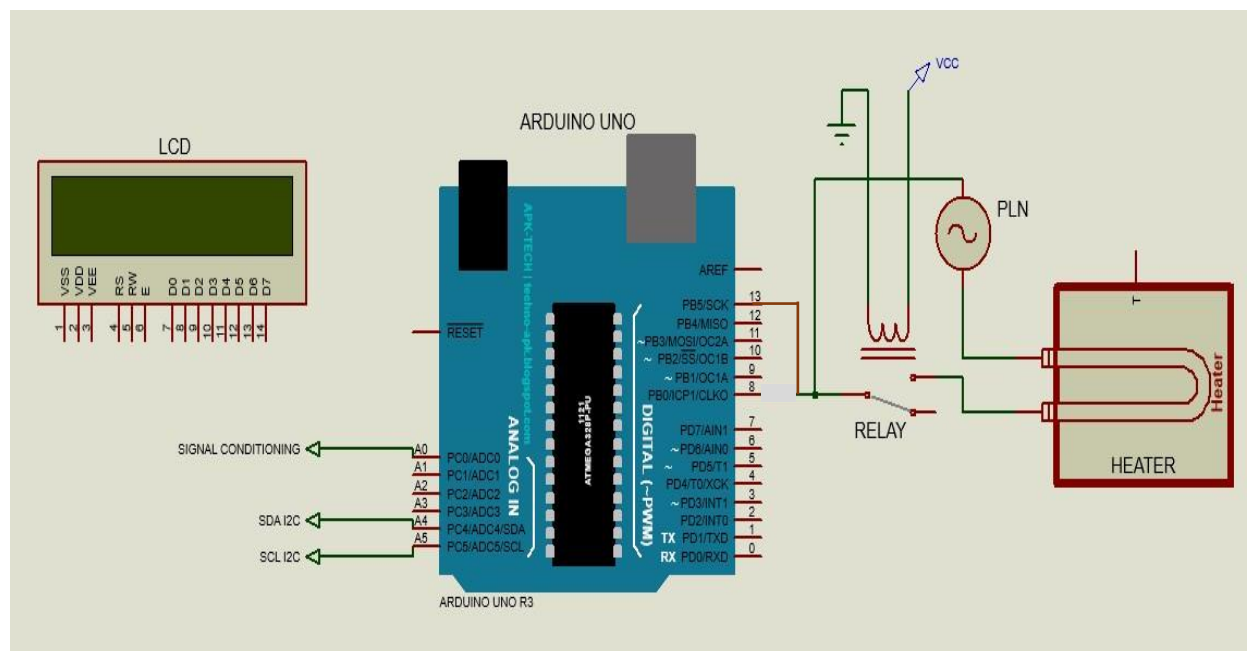
Penguat Instrumentasi yang digunakan yaitu TL082 dengan nilai V_{cc} masukan sebesar ± 12 V. Heater dikontrol menggunakan PID dari Arduino Uno. Fitur hardware yang digunakan yaitu tampilan LCD dan pengatur suhu set (keypad).



Gambar 3. Rangkaian menggunakan Thermistor NTC



Gambar 4. Rangkaian Power Supply CT



Gambar 5. Rangkaian Pin pada LCD I2C dan Plant dengan Arduino Uno.

HASIL KALIBRASI SENSOR SUHU

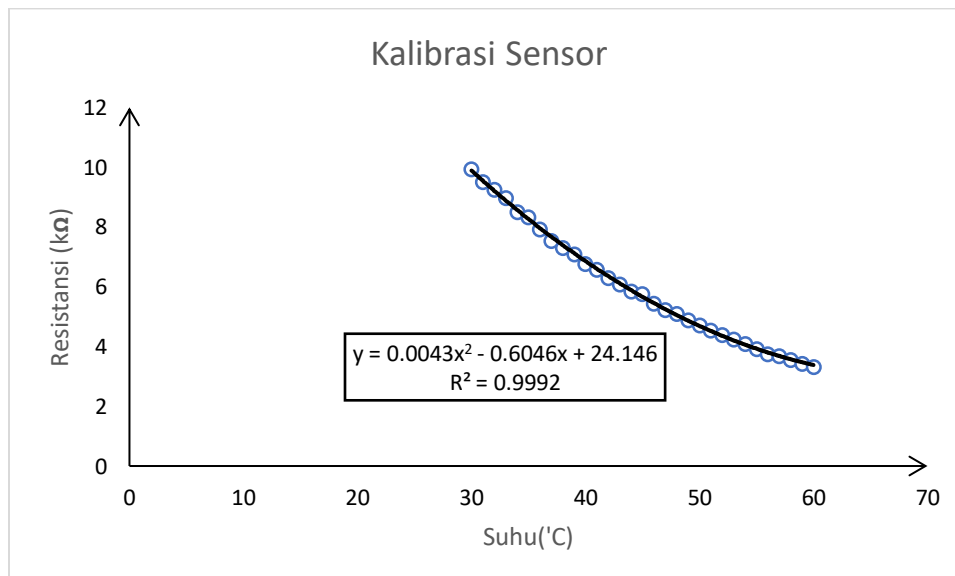
Sensor thermistor NTC 10k dilakukan kalibrasi antara suhu terhadap nilai resistansi.

Tabel 1. Nilai Kalibrasi Suhu terhadap Resistansi pada Sensor NTC 10k

T (°C)	R (kΩ)	T (°C)	R (kΩ)	T (°C)	R (kΩ)
30	9.95	40	6.77	50	4.72
31	9.52	41	6.59	51	4.55
32	9.26	42	6.31	52	4.4
33	8.98	43	6.09	53	4.24
34	8.5	44	5.85	54	4.1
35	8.33	45	5.77	55	3.92
36	7.94	46	5.44	56	3.75
37	7.55	47	5.24	57	3.7
38	7.31	48	5.11	58	3.564
39	7.09	49	4.9	59	3.43
				60	3.318

Selanjutnya dengan melakukan plot pada tabel 1, diperoleh persamaan hasil regresi nilai resistansi terhadap suhu sebagai berikut:

$$T = 0.0043(R)^2 - 0.6046(R) + 24.146$$



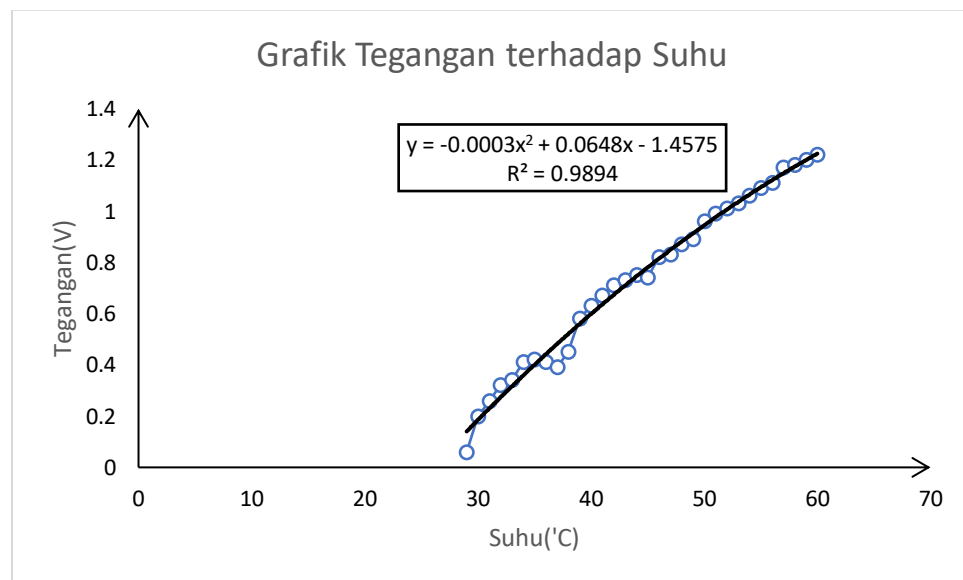
Gambar 6. Grafik Kalibrasi Resistansi terhadap Suhu pada Sensor suhu NTC 10k

HASIL PENGUKURAN KELUARAN

Setelah sensor suhu telah terkalibrasi, kemudian dilakukan kalibrasi pada rangkaian bridge dan pengondisi sinyal. Dengan menentukan nilai resistansi sedemikian rupa pada rangkaian bridge dan pengondisi sinyal, diperoleh nilai tegangan keluaran terhadap suhu pada PCB sebagai berikut.

Tabel 2. Nilai Tegangan Keluaran pada Rangkaian Bridge di PCB

Suhu	Nilai	Suhu	Nilai	Suhu	Nilai
T (°C)	Vout (V)	T (°C)	Vout (V)	T (°C)	Vout (V)
29	0.06	40	0.63	51	0.99
30	0.2	41	0.67	52	1.01
31	0.26	42	0.71	53	1.03
32	0.32	43	0.73	54	1.06
33	0.34	44	0.75	55	1.09
34	0.41	45	0.74	56	1.11
35	0.42	46	0.82	57	1.17
36	0.41	47	0.83	58	1.18
37	0.39	48	0.87	59	1.2
38	0.45	49	0.89	60	1.22
39	0.58	50	0.96		



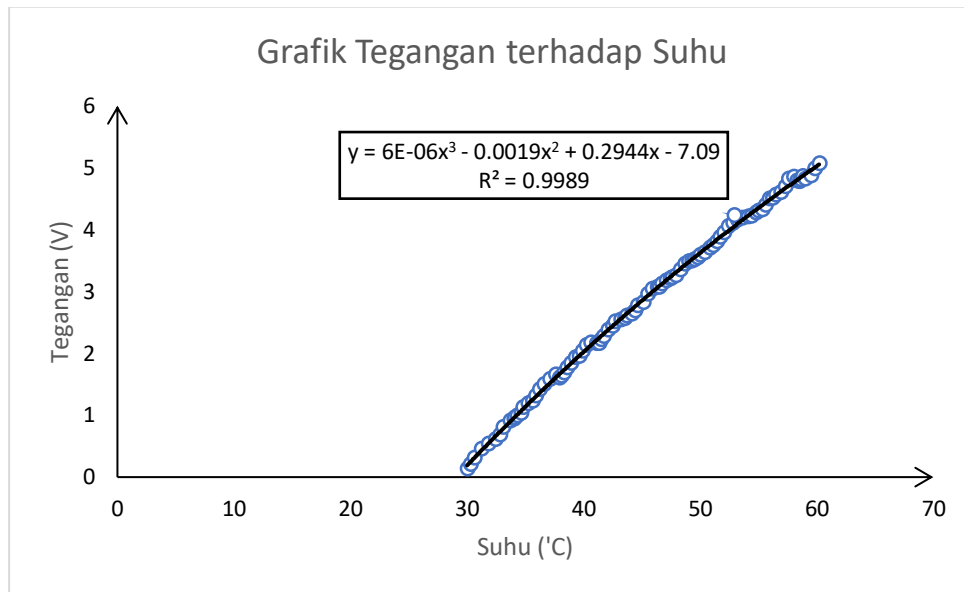
Gambar 7. Grafik Tegangan Terhadap Suhu pada rangkaian bridge

Setelah tegangan keluaran dari rangkain resistor *bridge* diatur sedemikian rupa, kemudian dilakukan variasi resistansi pada rangkaian pengondisi sinyal dengan mengatur penguatannya sehingga diperoleh data sebagai berikut.

Tabel 3. Nilai Tegangan Keluaran pada Rangkaian Pengondisi sinyal di PCB

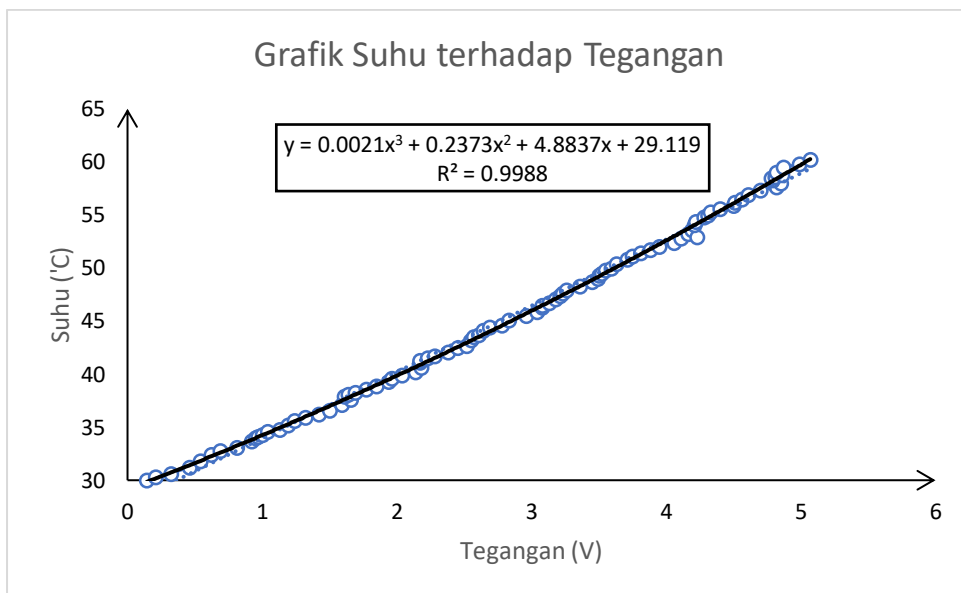
Suhu	Nilai	Suhu	Nilai	Suhu	Nilai
T (°C)	Vout (V)	T (°C)	Vout (V)	T (°C)	Vout (V)
30	0.14	41.1	2.17	50.8	3.71
30.3	0.21	41.3	2.17	51.1	3.75
30.6	0.32	41.5	2.23	51.4	3.81
31.2	0.46	41.7	2.28	51.7	3.88
31.8	0.54	42.1	2.38	52	3.95
32.4	0.62	42.5	2.45	52.4	4.06
32.8	0.69	42.7	2.52	52.8	4.11
33.1	0.81	43.2	2.55	53.2	4.16
33.7	0.92	43.5	2.57	53.6	4.19
34	0.95	43.7	2.61	52.9	4.23
34.1	0.97	44.1	2.64	54.1	4.21
34.3	1	44.4	2.69	54.4	4.22
34.6	1.04	44.6	2.78	54.8	4.28
34.8	1.13	45.1	2.83	55	4.31
35.2	1.19	45.5	2.96	55.3	4.33
35.6	1.24	45.9	3.04	55.6	4.4
35.9	1.32	46.3	3.08	55.9	4.5
36.2	1.42	46.5	3.08	56.2	4.51
36.6	1.5	46.7	3.13	56.5	4.56
37.1	1.59	47.1	3.18	56.9	4.61
37.6	1.66	47.4	3.21	57.3	4.7
37.9	1.61	47.6	3.23	57.6	4.82
38.1	1.64	47.9	3.26	58	4.85
38.3	1.69	48.3	3.36	58.3	4.79
38.6	1.77	48.7	3.45	58.5	4.78
38.9	1.85	49	3.49	58.6	4.81
39.3	1.94	49.3	3.5	58.8	4.86
39.6	1.96	49.5	3.52	59	4.82
39.9	2.04	49.8	3.55	59.5	4.87
40.2	2.14	50	3.59	59.8	4.99
40.6	2.18	50.4	3.63	60.2	5.07

Selanjutnya dengan melakukan plot pada tabel 3, diperoleh persamaan tegangan keluaran terhadap suhu sebagai berikut:



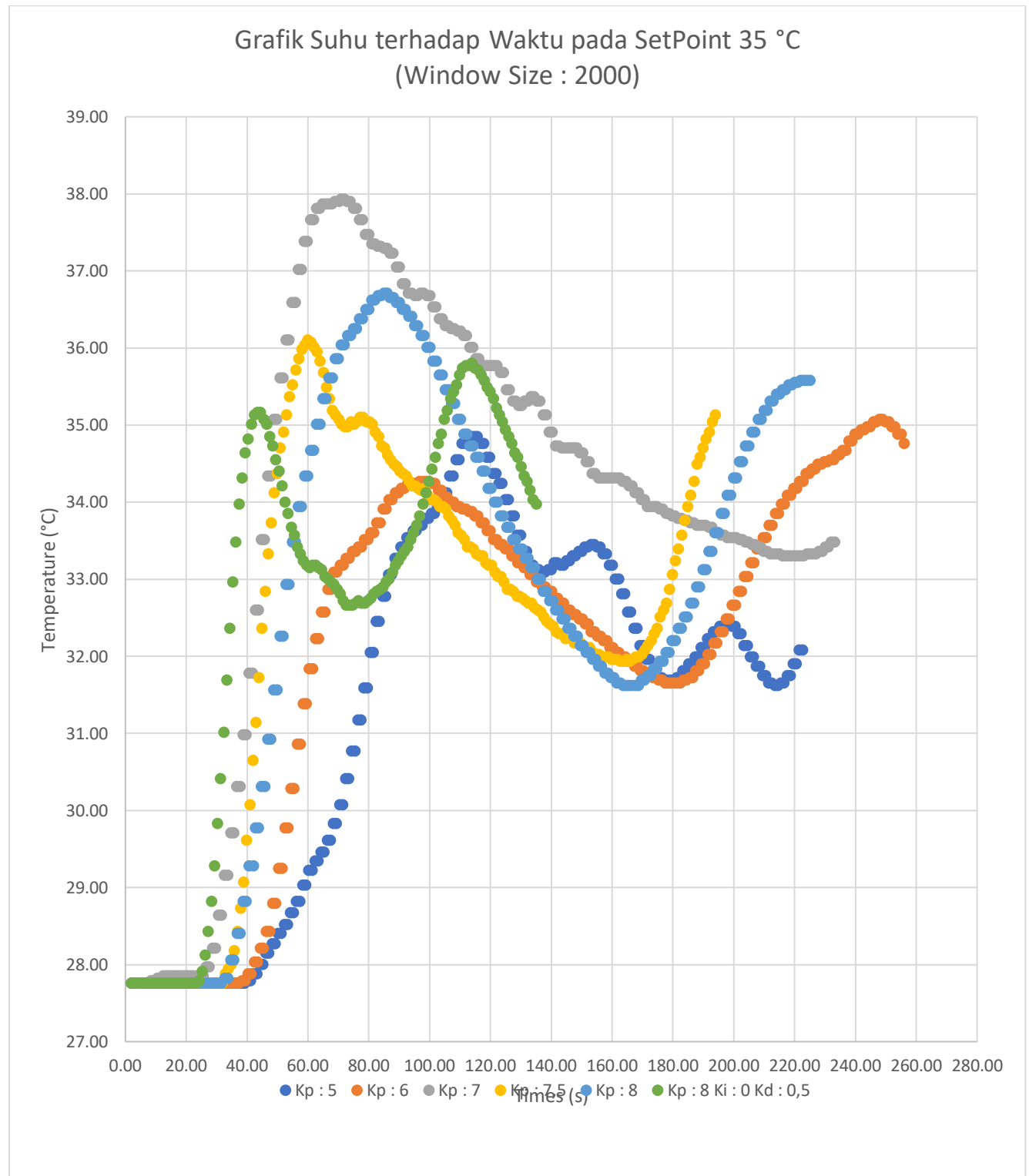
Gambar 8. Grafik Tegangan Keluaran Terhadap Suhu

Untuk masukan persamaan pada *coding* Arduino Uno, digunakan persamaan dari plot suhu terhadap tegangan keluaran sebagai berikut.



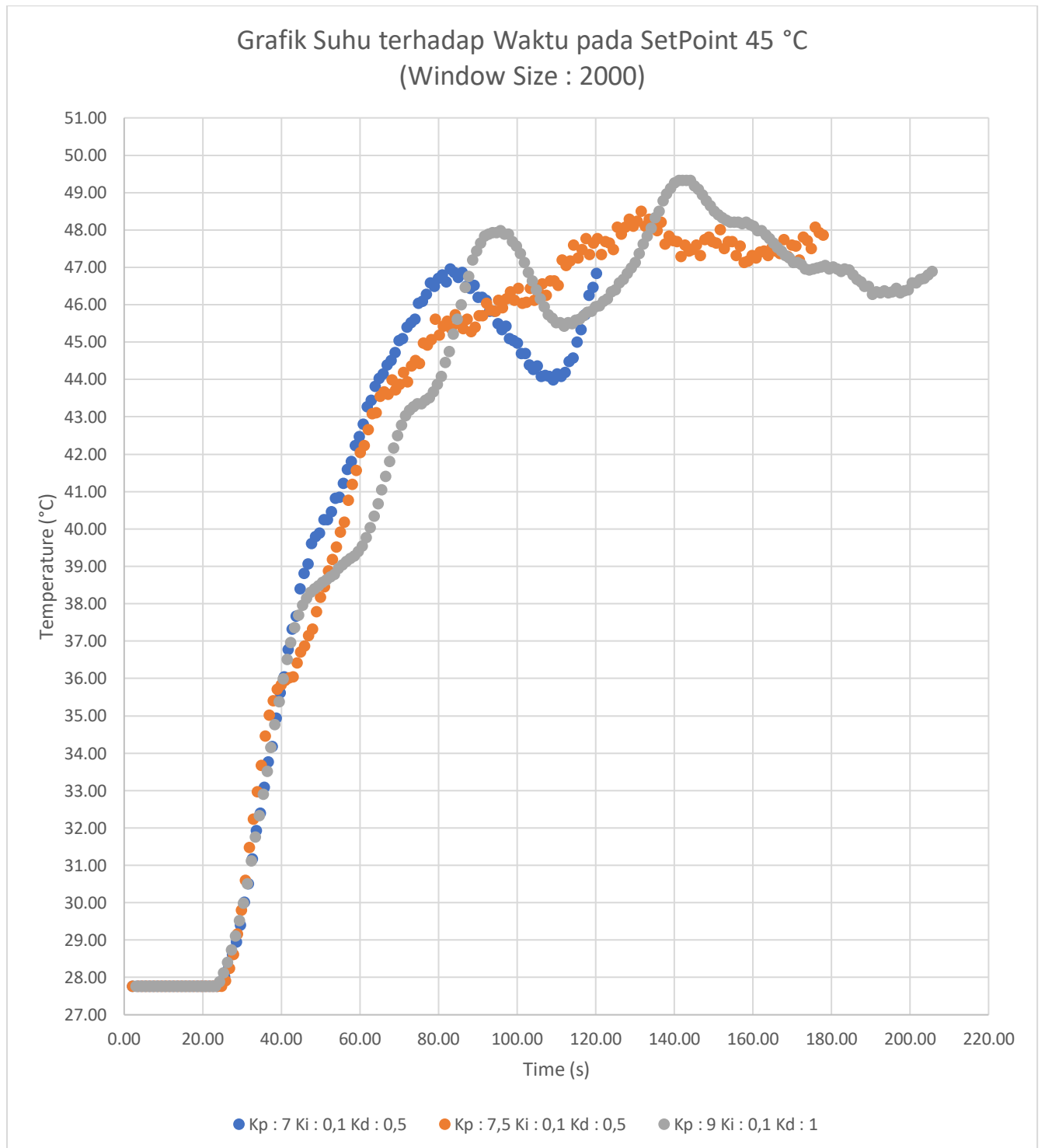
Gambar 9. Grafik Suhu Terhadap Tegangan Keluaran pada rangkaian pengondisi sinyal

Plot kenaikan suhu terhadap waktu menggunakan PID dengan *set point* 35°C



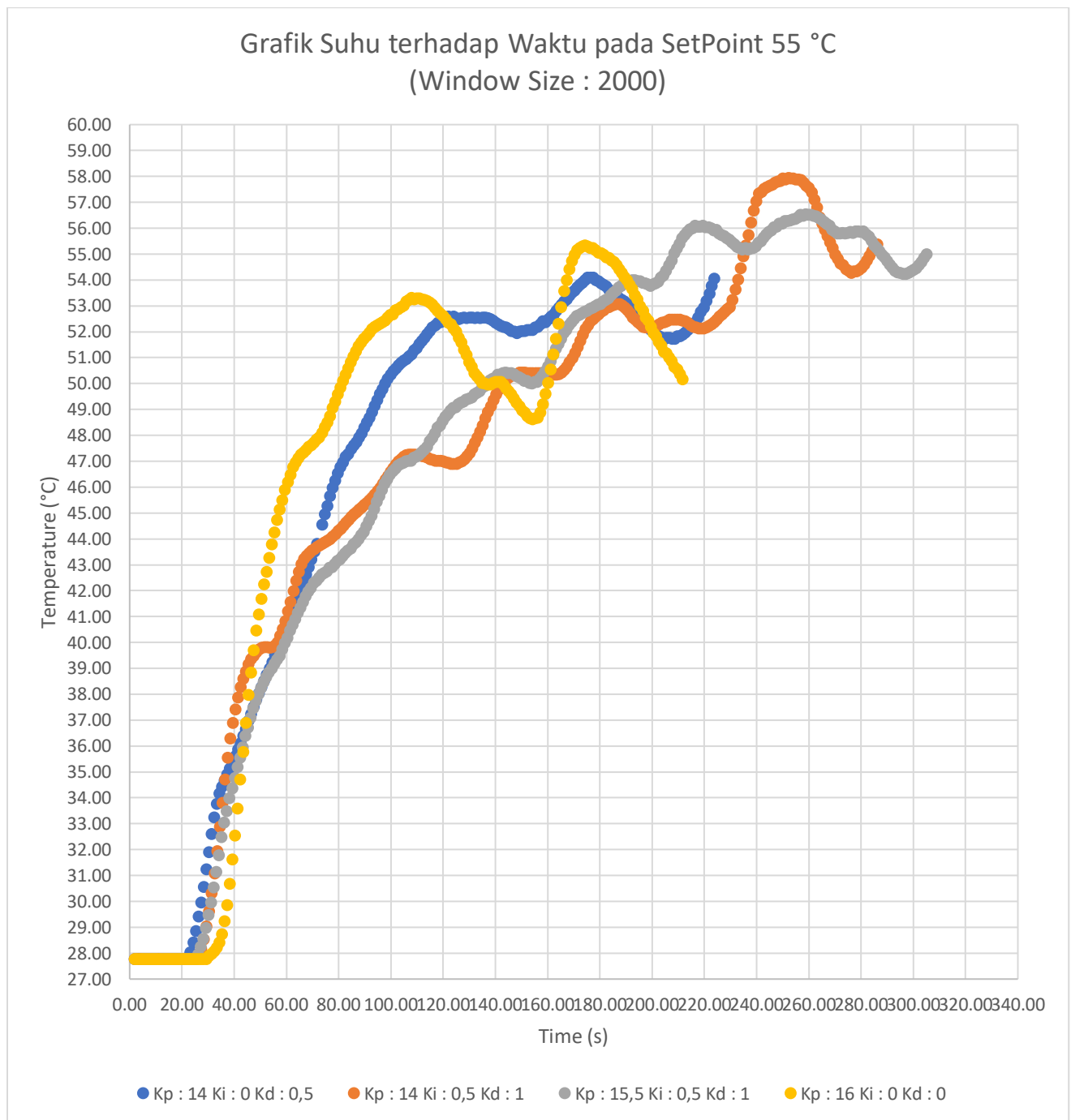
Gambar 9. Grafik Temperatur terhadap waktu pada setpoint 35°C

Plot kenaikan suhu terhadap waktu menggunakan PID dengan *set point* 45°C



Gambar 9. Grafik Temperatur terhadap waktu pada setpoint 45°C

Plot kenaikan suhu terhadap waktu menggunakan PID dengan *set point* 55°C



Gambar 9. Grafik Temperatur terhadap waktu pada setpoint 55°C

Penjelasan dari grafik kenaikan suhu terhadap waktu pada *set point* 35°C, 45°C, dan 55°C dijelaskan lebih lengkap pada bagian analisis.

ANALISIS

Nilai akurasi suhu yang ada di tampilan LCD dengan tampilan kalibrator berbeda $\pm 0.5^{\circ}\text{C}$ (rentang ini merupakan selisih nilai alat ukur dengan kalibrator; disebabkan karena ketelitian kalibrator hanya 1°C) dengan nilai di tampilan LCD lebih kecil dibandingkan dengan di tampilan kalibrator. Kesamaan nilai tampilan alat ukur dengan kalibrator berkurang saat suhu mendekati titik acu yang digunakan.

Analisis saat eksperimen PID dengan berbagai nilai K_p , K_i dan K_d dengan nilai window size yaitu sama sebesar 2000. Untuk yang pertama yaitu membuat setpoint pada 35 derajat celcius dan harus membuat *error steady state* nya dibawah 5% atau ± 1.75 derajat celcius dari hasil percobaan yang didapatkan nilai *error steady state* paling baik saat nilai $K_p = 8$ dan $K_d = 0.5$ yaitu sebesar 3.7% atau ± 1.3 derajat celcius. Untuk nilai $K_p = 8$ tanpa K_d didapatkan *overshot* begitu besar hingga berbeda 3 derajat celcius dari nilai setpoint, saat nilai $K_p = 6$ tanpa K_d didapatkan tidak pernah mencapai *overshot*. Untuk yang kedua yaitu setpoint pada 45 derajat celcius dan harus membuat *error steady state* nya dibawah 5% atau ± 2.25 derajat celcius dari hasil percobaan yang didapatkan nilai *error steady state* paling baik saat nilai $K_p = 7$, $K_i = 0.1$ dan $K_d = 0.5$ yaitu sebesar 3.3% atau ± 1.485 derajat celcius. Saat nilai $K_p = 9$, $K_i = 0.1$ dan $K_d = 1$ akan terjadi *overshot* sebesar 3 derajat yang berarti sudah diluar batas *error steady state* dan saat nilai $K_p = 7.5$ $K_i = 0.1$ dan $K_d = 0.5$ nilai temperatur akan diam pada nilai 47. Untuk yang ketiga yaitu setpoint pada 55 derajat celcius dan harus membuat *error steady state* nya dibawah 5% atau ± 2.75 derajat celcius, dari hasil percobaan didapatkan nilai paling *error steady state* paling baik saat $K_p = 15.5$ $K_i = 0.5$ dan $K_d = 1$ dengan nilai *error steady state* sebesar 1.69% atau ± 0.93 derajat celcius. Untuk variasi nilai lain $K_p = 14$ $K_i = 0.5$ dan $K_d = 1$ didapatkan nilai *overshot* sebesar 3 derajat celcius dengan waktu *rise time* yang cukup lama, lalu dengan percobaan yang lain dengan $K_p = 16$ terdapat fluktuasi nilai temperatur yang sangat tinggi dan juga tidak mencapai nilai setpoint.

KESIMPULAN

Telah dibuat alat ukur suhu menggunakan sensor thermistor NTC 10 k Ω , alat ini bekerja pada rentang suhu sebesar 30°C - 60°C . Dari hasil percobaan pada variasi nilai yang digunakan pada PID untuk nilai setpoint kecil seperti rentang 30 hingga 35 cukup susah untuk mencapai nilai *steady state error* dibawah 5% dikarenakan nilai skala yang cukup kecil yaitu hanya sebesar 5 derajat celcius dari hasil eksperimen didapatkan sebesar 3.7% , untuk set point 45 didapatkan nilai *steady state error* sebesar 3.3% , dan yang terakhir untuk setpoint 55 derajat celcius dengan skala yang cukup besar maka bisa didapatkan nilai *error steady state* semakin baik dari hasil eksperimen didapatkan sebesar 1.69%.

REFERENSI

- [1] Anonim. TT. Thermistor. <https://www.thermistor.com/thermistors>. Diakses pada 16 April 2019 jam 09.28 WIB.
- [2] Anonim. TT. Thermistors. <https://www.electronics-tutorials.ws/io/thermistors.html>. Diakses pada 16 April 2019 jam 09.30 WIB.
- [3] Justin Bauer. 2015. *Design of an Instrumentation Amplifier*. [pdf]. <https://www.electronicstutorials.ws/io/thermistors.html>. Diunduh pada 15 April 2019 jam 09.40 WIB.
- [4] Prof. Katherine Chandler. 2012. *Operational Amplifiers (Op Amps)*. [pdf]. www.eng.hmc.edu/NewE80/PDFs/Lecture%205%20-%20Op%20Amps.pdf. Diakses pada 12 April 2018.
- [5] Anonim. TT. Arduino Uno. <http://www.tuxti.com.br/wiki/index.php?title=Arquivo:Circuito-Arduino-i2c-display-lcd-16x2.png>. Diakses pada 16 April 2018 jam 14.24 WIB.

Daftar Simbol

$$R_1 = 1 \text{ k}\Omega$$

$$R_2 = R_3 = 2 \text{ k}\Omega$$

$$R_{POTE} = 6,71 \text{ k}\Omega$$

$$R_4 = R_5 = 20 \text{ k}\Omega$$

$$R_{POT} = 11,88 \text{ k}\Omega$$

$$R_6 = R_7 = R_8 = R_9 = 5 \text{ k}\Omega$$

$$R_T = \text{Resistansi pada Thermistor NTC 10K } (\Omega)$$

$$V_a = \text{Tegangan yang ada di titik antara } R_T \text{ dan } R_3 \text{ (V)}$$

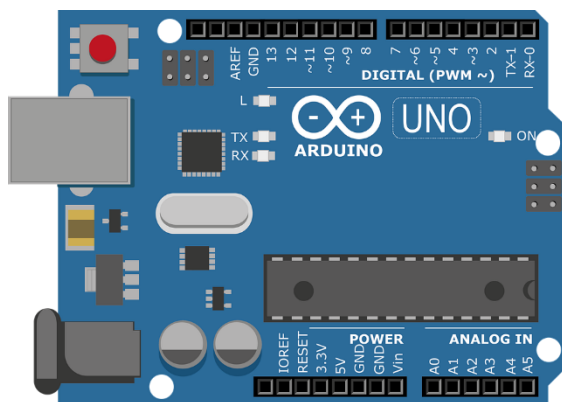
$$V_b = \text{Tegangan yang ada di titik antara } R_{POT} \text{ dan } R_2 \text{ (V)}$$

$$V_1 = \text{Tegangan yang ada di titik antara } R_4 \text{ dan } R_6 \text{ (V)}$$

$$V_2 = \text{Tegangan yang ada di titik antara } R_5 \text{ dan } R_8 \text{ (V)}$$

$$V_o = \text{Tegangan keluaran (V)}$$

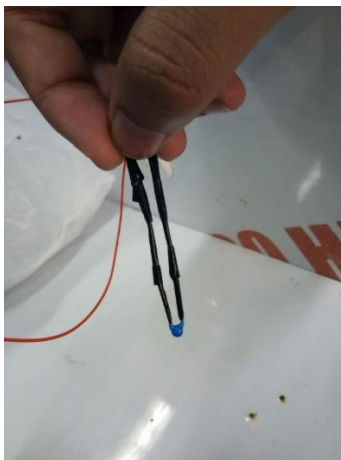
LAMPIRAN



Gambar 11. Arduino Uno



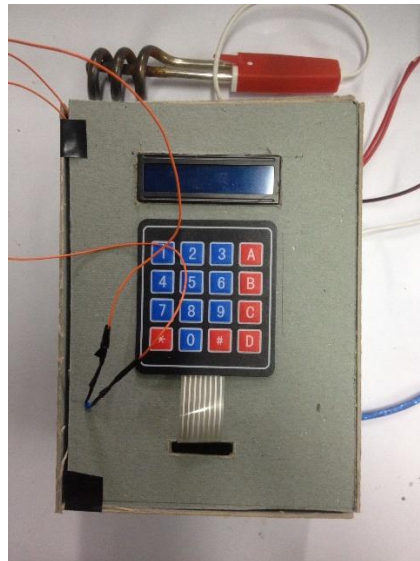
Gambar 12. Termometer



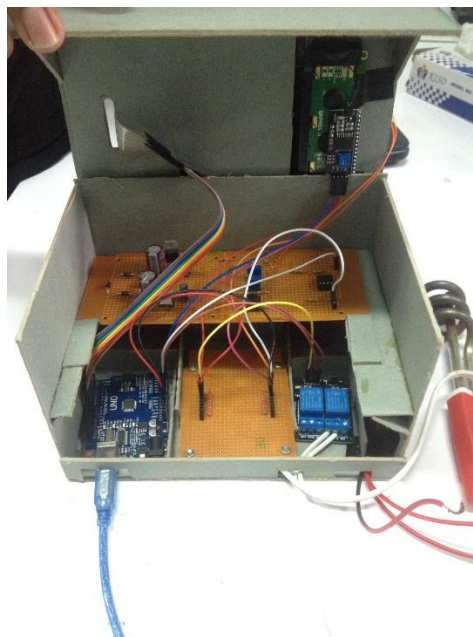
Gambar 13. Thermistor NTC 10k



Gambar 14. Trafo CT



Gambar 15. Heater, LCD, keypad yang digunakan



Gambar 16. Komponen yang terdapat didalam kotak

Source Code Arduino Uno

```
// Deklarasi Library yang digunakan
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <avr/pgmspace.h>
#include <PID_v1.h>

// Inisiasi variabel yang digunakan
const int sensorthermi = A0;
double resultPID;
double tempc;
double temps;
float Ton, Toff;
char InTemp[2];

unsigned long previousMillis1 = 0;
unsigned long previousMillis2 = 0;
unsigned long previousMillis3 = 0;
unsigned long previousMillis4 = 0;
const long interval1 = 2000;
const long interval2 = 2000;
const long interval3 = 500;
const long interval4 = 1000;

unsigned long PTexcel, PTdisplay, PTmeasure = 0;
unsigned long CTexcel, CTdisplay, CTmeasure = 0;
unsigned long ITexcel = 1000;
unsigned long ITdisplay = 666;
unsigned long ITmeasure = 100;

// SMOOTHING
const int numReadings = 10;
int readings[numReadings]; // the readings from the analog input
int readIndex = 0;         // the index of the current reading
int total = 0;
int average = 0;           // the average

// Menginisiasi nilai lookup tabel dengan skala 0-5 V secara teoritik
const float ADC_TO_TEMP[1024] PROGMEM = {27.7555, 27.7859, 27.8164, 27.8469, 27.8773, 27.9078,
27.9382, 27.9687, 27.9992, 28.0296, 28.0601, 28.0906, 28.121, 28.1515, 28.1819, 28.2124, 28.2429, 28.2733,
28.3038, 28.3343, 28.3647, 28.3952, 28.4257, 28.4561, 28.4866, 28.517, 28.5475, 28.578, 28.6084, 28.6389,
28.6694, 28.6998, 28.7303, 28.7607, 28.7912, 28.8217, 28.8521, 28.8826, 28.9131, 28.9435, 28.974, 29.0044,
29.0349, 29.0654, 29.0958, 29.1263, 29.1568, 29.1872, 29.2177, 29.2482, 29.2786, 29.3091, 29.3395, 29.37,
29.4005, 29.4309, 29.4614, 29.4919, 29.5223, 29.5528, 29.5832, 29.6137, 29.6442, 29.6746, 29.7051, 29.7356,
29.766, 29.7965, 29.827, 29.8574, 29.8879, 29.9183, 29.9488, 29.9793, 30.0097, 30.0402, 30.0707, 30.1011,
30.1316, 30.162, 30.1925, 30.223, 30.2534, 30.2839, 30.3144, 30.3448, 30.3753, 30.4058, 30.4362, 30.4667,
30.4971, 30.5276, 30.5581, 30.5885, 30.619, 30.6495, 30.6799, 30.7104, 30.7408, 30.7713, 30.8018, 30.8322,
30.8627, 30.8932, 30.9236, 30.9541, 30.9845, 31.015, 31.0455, 31.0759, 31.1064, 31.1369, 31.1673, 31.1978,
31.2283, 31.2587, 31.2892, 31.3196, 31.3501, 31.3806, 31.411, 31.4415, 31.472, 31.5024, 31.5329, 31.5633,
31.5938, 31.6243, 31.6547, 31.6852, 31.7157, 31.7461, 31.7766, 31.807, 31.8375, 31.868, 31.8984, 31.9289,
31.9594, 31.9898, 32.0203, 32.0508, 32.0812, 32.1117, 32.1421, 32.1726, 32.2031, 32.2335, 32.264, 32.2945,
```

32.3249, 32.3554, 32.3858, 32.4163, 32.4468, 32.4772, 32.5077, 32.5382, 32.5686, 32.5991, 32.6296, 32.66,
32.6905, 32.7209, 32.7514, 32.7819, 32.8123, 32.8428, 32.8733, 32.9037, 32.9342, 32.9646, 32.9951, 33.0256,
33.056, 33.0865, 33.117, 33.1474, 33.1779, 33.2083, 33.2388, 33.2693, 33.2997, 33.3302, 33.3607, 33.3911,
33.4216, 33.4521, 33.4825, 33.513, 33.5434, 33.5739, 33.6044, 33.6348, 33.6653, 33.6958, 33.7262, 33.7567,
33.7871, 33.8176, 33.8481, 33.8785, 33.909, 33.9395, 33.9699, 34.0004, 34.0309, 34.0613, 34.0918, 34.1222,
34.1527, 34.1832, 34.2136, 34.2441, 34.2746, 34.305, 34.3355, 34.3659, 34.3964, 34.4269, 34.4573, 34.4878,
34.5183, 34.5487, 34.5792, 34.6097, 34.6401, 34.6706, 34.701, 34.7315, 34.762, 34.7924, 34.8229, 34.8534,
34.8838, 34.9143, 34.9448, 34.9752, 35.0057, 35.0361, 35.0666, 35.0971, 35.1275, 35.158, 35.1885, 35.2189,
35.2494, 35.2799, 35.3103, 35.3408, 35.3712, 35.4017, 35.4322, 35.4626, 35.4931, 35.5236, 35.554, 35.5845,
35.6149, 35.6454, 35.6759, 35.7063, 35.7368, 35.7673, 35.7977, 35.8282, 35.8587, 35.8891, 35.9196, 35.95,
35.9805, 36.011, 36.0414, 36.0719, 36.1024, 36.1328, 36.1633, 36.1938, 36.2242, 36.2547, 36.2851, 36.3156,
36.3461, 36.3765, 36.407, 36.4375, 36.4679, 36.4984, 36.5289, 36.5593, 36.5898, 36.6202, 36.6507, 36.6812,
36.7116, 36.7421, 36.7726, 36.803, 36.8335, 36.8639, 36.8944, 36.9249, 36.9553, 36.9858, 37.0163, 37.0467,
37.0772, 37.1077, 37.1381, 37.1686, 37.199, 37.2295, 37.26, 37.2904, 37.3209, 37.3514, 37.3818, 37.4123,
37.4428, 37.4732, 37.5037, 37.5341, 37.5646, 37.5951, 37.6255, 37.656, 37.6865, 37.7169, 37.7474, 37.7779,
37.8083, 37.8388, 37.8692, 37.8997, 37.9302, 37.9606, 37.9911, 38.0216, 38.052, 38.0825, 38.1129, 38.1434,
38.1739, 38.2043, 38.2348, 38.2653, 38.2957, 38.3262, 38.3567, 38.3871, 38.4176, 38.448, 38.4785, 38.509,
38.5394, 38.5699, 38.6004, 38.6308, 38.6613, 38.6918, 38.7222, 38.7527, 38.7831, 38.8136, 38.8441, 38.8745,
38.905, 38.9355, 38.9659, 38.9964, 39.0269, 39.0573, 39.0878, 39.1182, 39.1487, 39.1792, 39.2096, 39.2401,
39.2706, 39.301, 39.3315, 39.3619, 39.3924, 39.4229, 39.4533, 39.4838, 39.5143, 39.5447, 39.5752, 39.6057,
39.6361, 39.6666, 39.697, 39.7275, 39.758, 39.7884, 39.8189, 39.8494, 39.8798, 39.9103, 39.9408, 39.9712,
40.0017, 40.0321, 40.0626, 40.0931, 40.1235, 40.154, 40.1845, 40.2149, 40.2454, 40.2758, 40.3063, 40.3368,
40.3672, 40.3977, 40.4282, 40.4586, 40.4891, 40.5196, 40.55, 40.5805, 40.6109, 40.6414, 40.6719, 40.7023,
40.7328, 40.7633, 40.7937, 40.8242, 40.8547, 40.8851, 40.9156, 40.946, 40.9765, 41.007, 41.0374, 41.0679,
41.0984, 41.1288, 41.1593, 41.1898, 41.2202, 41.2507, 41.2811, 41.3116, 41.3421, 41.3725, 41.403, 41.4335,
41.4639, 41.4944, 41.5248, 41.5553, 41.5858, 41.6162, 41.6467, 41.6772, 41.7076, 41.7381, 41.7686, 41.799,
41.8295, 41.8599, 41.8904, 41.9209, 41.9513, 41.9818, 42.0123, 42.0427, 42.0732, 42.1037, 42.1341, 42.1646,
42.195, 42.2255, 42.256, 42.2864, 42.3169, 42.3474, 42.3778, 42.4083, 42.4388, 42.4692, 42.4997, 42.5301,
42.5606, 42.5911, 42.6215, 42.652, 42.6825, 42.7129, 42.7434, 42.7738, 42.8043, 42.8348, 42.8652, 42.8957,
42.9262, 42.9566, 42.9871, 43.0176, 43.048, 43.0785, 43.1089, 43.1394, 43.1699, 43.2003, 43.2308, 43.2613,
43.2917, 43.3222, 43.3527, 43.3831, 43.4136, 43.444, 43.4745, 43.505, 43.5354, 43.5659, 43.5964, 43.6268,
43.6573, 43.6877, 43.7182, 43.7487, 43.7791, 43.8096, 43.8401, 43.8705, 43.901, 43.9315, 43.9619, 43.9924,
44.0228, 44.0533, 44.0838, 44.1142, 44.1447, 44.1752, 44.2056, 44.2361, 44.2666, 44.297, 44.3275, 44.3579,
44.3884, 44.4189, 44.4493, 44.4798, 44.5103, 44.5407, 44.5712, 44.6017, 44.6321, 44.6626, 44.693, 44.7235,
44.754, 44.7844, 44.8149, 44.8454, 44.8758, 44.9063, 44.9367, 44.9672, 44.9977, 45.0281, 45.0586, 45.0891,
45.1195, 45.15, 45.1805, 45.2109, 45.2414, 45.2718, 45.3023, 45.3328, 45.3632, 45.3937, 45.4242, 45.4546,
45.4851, 45.5156, 45.546, 45.5765, 45.6069, 45.6374, 45.6679, 45.6983, 45.7288, 45.7593, 45.7897, 45.8202,
45.8507, 45.8811, 45.9116, 45.942, 45.9725, 46.003, 46.0334, 46.0639, 46.0944, 46.1248, 46.1553, 46.1857,
46.2162, 46.2467, 46.2771, 46.3076, 46.3381, 46.3685, 46.399, 46.4295, 46.4599, 46.4904, 46.5208, 46.5513,
46.5818, 46.6122, 46.6427, 46.6732, 46.7036, 46.7341, 46.7646, 46.795, 46.8255, 46.8559, 46.8864, 46.9169,
46.9473, 46.9778, 47.0083, 47.0387, 47.0692, 47.0997, 47.1301, 47.1606, 47.191, 47.2215, 47.252, 47.2824,
47.3129, 47.3434, 47.3738, 47.4043, 47.4347, 47.4652, 47.4957, 47.5261, 47.5566, 47.5871, 47.6175, 47.648,
47.6785, 47.7089, 47.7394, 47.7698, 47.8003, 47.8308, 47.8612, 47.8917, 47.9222, 47.9526, 47.9831, 48.0136,
48.044, 48.0745, 48.1049, 48.1354, 48.1659, 48.1963, 48.2268, 48.2573, 48.2877, 48.3182, 48.3486, 48.3791,
48.4096, 48.44, 48.4705, 48.501, 48.5314, 48.5619, 48.5924, 48.6228, 48.6533, 48.6837, 48.7142, 48.7447,
48.7751, 48.8056, 48.8361, 48.8665, 48.897, 48.9275, 48.9579, 48.9884, 49.0188, 49.0493, 49.0798, 49.1102,
49.1407, 49.1712, 49.2016, 49.2321, 49.2626, 49.293, 49.3235, 49.3539, 49.3844, 49.4149, 49.4453, 49.4758,
49.5063, 49.5367, 49.5672, 49.5976, 49.6281, 49.6586, 49.689, 49.7195, 49.75, 49.7804, 49.8109, 49.8414,
49.8718, 49.9023, 49.9327, 49.9632, 49.9937, 50.0241, 50.0546, 50.0851, 50.1155, 50.146, 50.1765, 50.2069,
50.2374, 50.2678, 50.2983, 50.3288, 50.3592, 50.3897, 50.4202, 50.4506, 50.4811, 50.5116, 50.542, 50.5725,
50.6029, 50.6334, 50.6639, 50.6943, 50.7248, 50.7553, 50.7857, 50.8162, 50.8466, 50.8771, 50.9076, 50.938,
50.9685, 50.999, 51.0294, 51.0599, 51.0904, 51.1208, 51.1513, 51.1817, 51.2122, 51.2427, 51.2731, 51.3036,
51.3341, 51.3645, 51.395, 51.4255, 51.4559, 51.4864, 51.5168, 51.5473, 51.5778, 51.6082, 51.6387, 51.6692,
51.6996, 51.7301, 51.7606, 51.791, 51.8215, 51.8519, 51.8824, 51.9129, 51.9433, 51.9738, 52.0043, 52.0347,

```

52.0652, 52.0956, 52.1261, 52.1566, 52.187, 52.2175, 52.248, 52.2784, 52.3089, 52.3394, 52.3698, 52.4003,
52.4307, 52.4612, 52.4917, 52.5221, 52.5526, 52.5831, 52.6135, 52.644, 52.6745, 52.7049, 52.7354, 52.7658,
52.7963, 52.8268, 52.8572, 52.8877, 52.9181, 52.9486, 52.9791, 53.0095, 53.04, 53.0705, 53.1009, 53.1314,
53.1618, 53.1923, 53.2228, 53.2532, 53.2837, 53.3142, 53.3446, 53.3751, 53.4055, 53.436, 53.4665, 53.4969,
53.5274, 53.5578, 53.5883, 53.6188, 53.6492, 53.6797, 53.7102, 53.7406, 53.7711, 53.8015, 53.832, 53.8625,
53.8929, 53.9234, 53.9539, 53.9843, 54.0148, 54.0452, 54.0757, 54.1062, 54.1366, 54.1671, 54.1975, 54.228,
54.2585, 54.2889, 54.3194, 54.3499, 54.3803, 54.4108, 54.4412, 54.4717, 54.5022, 54.5326, 54.5631, 54.5936,
54.624, 54.6545, 54.6849, 54.7154, 54.7459, 54.7763, 54.8068, 54.8372, 54.8677, 54.8982, 54.9286, 54.9591,
54.9896, 55.02, 55.0505, 55.0809, 55.1114, 55.1419, 55.1723, 55.2028, 55.2333, 55.2637, 55.2942, 55.3246,
55.3551, 55.3856, 55.416, 55.4465, 55.4769, 55.5074, 55.5379, 55.5683, 55.5988, 55.6293, 55.6597, 55.6902,
55.7206, 55.7511, 55.7816, 55.812, 55.8425, 55.873, 55.9034, 55.9339, 55.9643, 55.9948, 56.0253, 56.0557,
56.0862, 56.1166, 56.1471, 56.1776, 56.208, 56.2385, 56.269, 56.2994, 56.3299, 56.3603, 56.3908, 56.4213,
56.4517, 56.4822, 56.5127, 56.5431, 56.5736, 56.604, 56.6345, 56.665, 56.6954, 56.7259, 56.7563, 56.7868,
56.8173, 56.8477, 56.8782, 56.9087, 56.9391, 56.9696, 57, 57.0305, 57.061, 57.0914, 57.1219, 57.1524, 57.1828,
57.2133, 57.2437, 57.2742, 57.3047, 57.3351, 57.3656, 57.396, 57.4265, 57.457, 57.4874, 57.5179, 57.5484,
57.5788, 57.6093, 57.6397, 57.6702, 57.7007, 57.7311, 57.7616, 57.7921, 57.8225, 57.853, 57.8834, 57.9139,
57.9444, 57.9748, 58.0053, 58.0358, 58.0662, 58.0967, 58.1271, 58.1576, 58.1881, 58.2185, 58.249, 58.2794,
58.3099, 58.3404, 58.3708, 58.4013, 58.4318, 58.4622, 58.4927, 58.5231, 58.5536, 58.5841, 58.6145, 58.645,
58.6754, 58.7059, 58.7364, 58.7668, 58.7973, 58.8278, 58.8582, 58.8887, 58.9191}

```

```

;
// Alamat LCD
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE);

// Menginisiasi nilai kolom dan baris yang digunakan
const byte ROWS = 4;          //Define amount of Rows
const byte COLS = 4;          //Define amount of Columns
char hexaKeys[ROWS][COLS] = { //Define Keypad layout
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; //Pins for Keypad Input
byte colPins[COLS] = {5, 4, 3, 2};
Keypad myKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

```

```

// Setup for PID
double Setpoint, Input, Output;
double Kp=5.0, Ki=3.0, Kd=3.0;
const int Relay = A1;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT); // Kpnnya dulu dimainkan agar mendapatkan
rise time yang paling cepat
//PID myPID(&Input, &output, &setpoint, Kp, Ki, Kd, Direction);
int WindowSize = 2000;
unsigned long windowStartTime;

```

```

void setup() {
    Serial.begin(9600);

    // Setup untuk PID
    windowStartTime = millis();
    myPID.SetOutputLimits(0, WindowSize);
    myPID.SetMode(AUTOMATIC);
}

```

```

// Untuk smoothing
for (int thisReading = 0; thisReading < numReadings; thisReading++) {
  readings[thisReading] = 0;}

// Menginisiasi LCD
  lcd.begin(16,2);
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0,0);
pinMode(A0, INPUT);
pinMode(A1, OUTPUT);

// Persiapan
  lcd.print("READY");
}

void loop() {
  unsigned long currentMillis1 = millis(); // Syarat agar keypad selalu dapat menerima nilai masukan
  if (currentMillis1 - previousMillis1 >= interval1) { previousMillis1 = currentMillis1;

    // Mencari nilai rata-rata masukan ADC
    // subtract the last reading:
    total = total - readings[readIndex];
    // read from the sensor:
    readings[readIndex] = analogRead(sensorthermi);
    // add the reading to the total:
    total = total + readings[readIndex];
    // advance to the next position in the array:
    readIndex = readIndex + 1;

    // if we're at the end of the array...
    if (readIndex >= numReadings) {
      // ...wrap around to the beginning:
      readIndex = 0;
    }

    // calculate the average:
    average = total / numReadings;
    // mengkonversi tegangan yang dibaca untuk ditampilkan pada serial monitor
    float voltage = average * 5.0;
    voltage /= 1024.0; }

    // memunculkan nilai tegangan pada serial monitor
    //Serial.print(voltage); Serial.println(" volts"); }

// PENGAMBILAN
  // melakukan fungsi lookup table dengan data lookup tabel yang sudah dinisiasikan
  unsigned long currentMillis2 = millis(); // Syarat agar keypad selalu dapat menerima nilai masukan
  if (currentMillis2 - previousMillis2 >= interval2) { previousMillis2 = currentMillis2;
    tempc = pgm_read_float_near (ADC_TO_TEMP + average);
    Serial.println(tempc); //Serial.println(" derajat celcius"); // menulis nilai keluaran
    //Serial.print("nilai bit rata-rata : "); Serial.println(average); // dimunculkan pada serial monitor
  }
}

```

```

//Serial.println(); Serial.println();
//Serial.print(" temps : ");Serial.println(temps);

// MENAMPILKAN NILAI CURRENT TEMPERATURE DAN SET TEMPERATURE PADA DISPLAY
lcd.setCursor(0,0); // we start writing from the first row first column
lcd.print("TCur : ");lcd.print(tempc);lcd.print((char)223);lcd.print("C"); // Mengeluarkan nilai temperatur
saat ini pada LCD
lcd.setCursor(0,1);
lcd.print("TSet : ");lcd.print(temps); lcd.print((char)223);lcd.print("C"); // Mengeluarkan nilai temperatur
yang akan di set
}

// MENGAMBIL MASUKAN Temps DARI KEYPAD, belum dimasukkan nilai temps nya!
unsigned long currentMillis3 = millis(); // Syarat agar keypad selalu dapat menerima nilai masukan
if (currentMillis3 - previousMillis3 >= interval3) { previousMillis3 = currentMillis3;
char Key = myKeypad.getKey();
if (Key == 'A'){ //When "Enter" button is pushed
lcd.clear();
lcd.setCursor(0,0);lcd.print("Target Temp : "); //Output LCD top bar
lcd.setCursor(0,1);lcd.print(temps); //Output LCD bottom bar

KeyInput: //Goto start of KeyInput
lcd.setCursor(0,1);lcd.print(" "); //Clear bottom line
int data_count = 0; //Reset data_count
do{
Key = myKeypad.getKey(); //Get Key from Keypad
if (isdigit(Key)){ //Check for input
InTemp[data_count] = Key; //Get input into char array
lcd.setCursor(data_count,1);lcd.print(InTemp[data_count]); //Output Key on ICD
data_count++;
}
}while(data_count < 2);

temps = atoi(InTemp); //Convert Char array to Integer
if (temps > 60 || temps < 30) { //Check for invalid input
lcd.setCursor(0,1);lcd.print("ERROR");
;
goto KeyInput;
} ; }; char Key2 = myKeypad.getKey(); if (Key == 'B') {lcd.clear();};
}

CTexcel = millis();
if (CTexcel - PTexcel >= ITexcel) {
Serial.print("DATA, ");
Serial.print(tempc);
Serial.print(",");
Serial.print("TIMER");
Serial.print(",");
Serial.println();

// SETUP PID sesuai dengan library
myPID.Compute();
Input = (int)(((tempc - 27.75555)/31.1636)*1024);

```

```
Setpoint = (int)((((temps - 27.75555)/31.1636)*1024);

if (millis() - windowStartTime > WindowSize)
{ //time to shift the Relay Window
  windowStartTime += WindowSize;
}

if ( Output < millis() - windowStartTime) digitalWrite(Relay, HIGH);
else digitalWrite(Relay, LOW); Serial.println(Output);
}

}
```