

**Nama: Deva Maulid Dini**






**Nim: 22106020017**

**LAPORAN**  
**PROSES PEMANASAN**

**1. ALAT DAN BAHAN**

Adapun Alat dan Bahan yang digunakan dalam percobaan kali ini:

Tabel 1. Alat dan Bahan

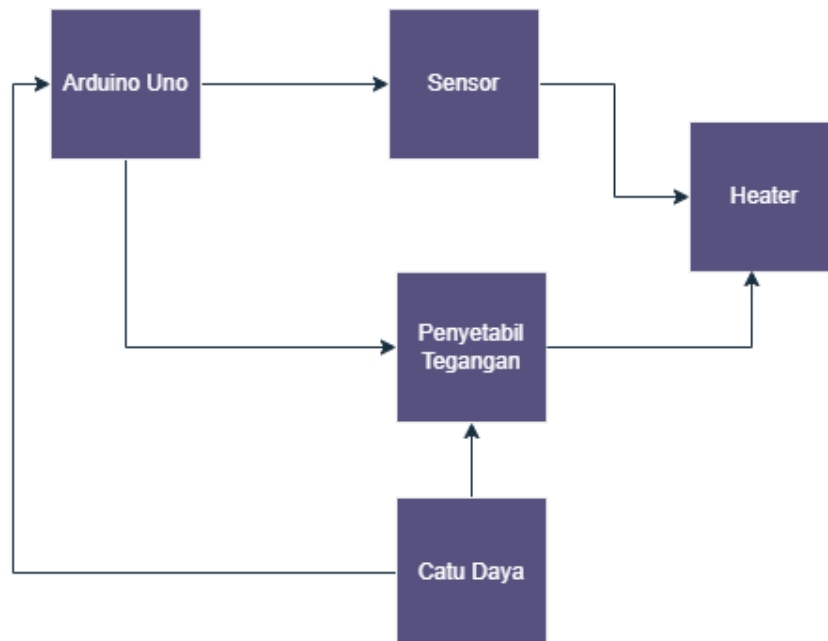
No	Alat & Bahan	Gambar
1.	Arduino Uno	
2.	Jumper	
3.	Resistor	
4.	Heater & Sensor	
6.	Kabel Penghubung	

6.	Penyearah Tegangan	
8.	Kabel Konektor	
9.	Breadboard	

## 2. PERANCANGAN

Adapun Perancangan sebagai berikut:

### a. Diagram Block



Gambar 1 Diagram Block

### 3. PEMBUATAN

Berikut adalah langkah-langkah untuk perakitan komponen:

- **Langkah 1:** Hubungkan salah satu kabel dari Termistor NTC 100k 3950 ke pin analog Arduino (A0). Rangkaian ini harus dirancang dalam bentuk rangkaian seri dengan resistor untuk pembacaan data suhu.
- **Langkah 2:** Hubungkan kabel dari Termistor ke pin 3.3V pada Arduino untuk memberikan tegangan pada rangkaian tersebut. Tegangan 3.3V ini penting untuk mendapatkan pembacaan sensor yang akurat dan sesuai dengan spesifikasi Termistor.
- **Langkah 3:** Hubungkan kabel M- dan M+ dari modul BTS 7960 ke heater PWM. Heater PWM ini akan dikendalikan oleh BTS 7960 untuk mengatur daya yang diberikan berdasarkan suhu yang dibaca oleh termistor.
- **Langkah 4:** Hubungkan B- ke Ground dari power supply, dan B+ ke VCC power supply. Ground dan VCC dari power supply diperlukan untuk memberikan arus yang stabil ke modul driver.
- **Langkah 5:** Hubungkan pin Ground dari BTS 7960 ke pin Ground di Arduino Uno untuk menyamakan referensi ground antara driver dan Arduino. Selanjutnya, sambungkan pin VCC dari BTS 7960 ke pin Aref Arduino untuk referensi tegangan.
- **Langkah 6:** Sambungkan pin RPWM dari BTS 7960 ke pin Digital-9 di Arduino. Pin ini akan digunakan sebagai input PWM yang mengontrol keluaran daya ke heater PWM berdasarkan sinyal yang dihasilkan oleh Arduino.

### 4. PENGUJIAN

Adapun pengujian yang dilakukan adalah:

1. Pengujian dengan beberapa variasi yang berbeda

Tabel 4 Hasil Pengujian

<i>Percobaan Kp</i>		<i>Ki</i>	<i>Kd</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Steadystate error</i>
1	1	0	0	-19,41	30,81	
2	1	0.008	0	22,25	123,26	
3	1	0.008	0.001	21,99	126,74	

kita dapat menganalisis pengaruh parameter kontrol Proportional ( $K_p$ ), Integral ( $K_i$ ), dan Derivative ( $K_d$ ) terhadap kinerja sistem, khususnya pada rise time, overshoot, dan steady-state error.

- Percobaan 1 ( $K_p=1$ ,  $K_i=0$ ,  $K_d=0$ )

Rise Time: -19,41

Nilai negatif pada rise time ini menunjukkan adanya ketidaksesuaian atau kemungkinan kesalahan pengukuran. Idealnya, rise time menunjukkan waktu yang dibutuhkan sistem untuk mencapai 90% dari nilai akhir.

Overshoot: 30,81

Sistem mengalami overshoot sebesar 30,81, yang artinya respons melebihi nilai target sebelum akhirnya stabil. Nilai ini menunjukkan sistem tidak stabil karena parameter  $K_i$  dan  $K_d$  bernilai nol.

- Percobaan 2 ( $K_p=1$ ,  $K_i=0.008$ ,  $K_d=0$ )

Rise Time: 22,25

Rise time positif menunjukkan sistem mencapai 90% dari nilai akhir dalam waktu tertentu, lebih lama dibandingkan percobaan 1. Penambahan komponen  $K_i$  menambah waktu respon karena aksi integral bertujuan memperbaiki kesalahan kumulatif yang menambah lambatnya sistem mencapai target.

Overshoot: 123,26

Overshoot meningkat drastis menjadi 123,26, yang menunjukkan bahwa aksi integral berlebihan dan menyebabkan sistem melampaui target secara signifikan. Ini biasanya terjadi jika nilai  $K_i$  terlalu besar.

- Percobaan 3 ( $K_p=1$ ,  $K_i=0.008$ ,  $K_d=0.001$ )

Rise Time: 21,99

Rise time sedikit berkurang dibanding percobaan 2. Penambahan komponen derivatif ( $K_d$ ) memberikan respons korektif yang mempercepat proses stabilisasi awal.

Overshoot: 126,74

Overshoot masih tinggi dan meningkat menjadi 126,74. Penambahan  $K_d$  membantu sedikit memperbaiki respons awal, tetapi tidak cukup signifikan untuk mengurangi overshoot yang tinggi akibat  $K_i$ .

Dari ketiga percobaan ini, terlihat bahwa Pengaruh  $K_p$  menentukan sensitivitas sistem terhadap error semakin besar nilai  $K_p$ , semakin agresif respons sistem.

Pengaruh  $K_i$  untuk mengurangi steady-state error, tetapi jika terlalu besar dapat menyebabkan overshoot berlebih dan memperpanjang rise time.

Pengaruh  $K_d$  memberikan efek redaman untuk mengurangi osilasi yang timbul, tetapi efeknya belum terlihat signifikan di sini. Untuk mencapai hasil optimal, parameter  $K_p$ ,  $K_i$ , dan  $K_d$  perlu diatur dengan tepat agar rise time dan overshoot minimal, serta steady-state error mendekati nol.