

BAB I PENDAHULUAN

1.1 LATAR BELAKANG

Pada masa kini susu merupakan minuman yang telah dikonsumsi oleh banyak masyarakat dari berbagai usia. Mulai dari yang berusia bayi, hingga yang sudah berusia lanjut. Susu dapat diolah menjadi berbagai macam produk minuman yang menarik seperti yoghurt, kefir, yakult, dan sebagainya. Akan tetapi, susu merupakan media pertumbuhan yang sangat baik bagi bakteri dan dapat menjadi sarana potensial bagi penyebaran bakteri patogen yang mudah tercemar kapan saja dan dimana saja sepanjang penanganannya tidak memperhatikan faktor kebersihan. Pencemaran pada susu terjadi sejak proses pemerahan, yang berasal dari berbagai sumber seperti kulit sapi, kambing, air, tanah, debu, manusia, peralatan dan udara (Rombaut, 2005).

Dipandang dari segi kesehatan manusia, susu segar yang tidak dipasteurisasi merupakan bahan makanan yang membahayakan bila dikonsumsi langsung, karena susu merupakan media yang sempurna untuk pertumbuhan mikroba yang dapat menginfeksi manusia. Penyakit seperti TBC, typhus, disentri dapat ditularkan melalui susu mentah. Pasteurisasi yang dinamakan sesuai dengan penemunya (Louis Pasteur) adalah suatu proses pemanasan atau pemasakan suatu produk dalam hal ini yang berupa susu, dengan tujuan untuk membunuh semua mikroorganisme patogen. Selain membuat susu menjadi aman dikonsumsi manusia, pasteurisasi juga akan memperpanjang kualitas umur simpan dari susu karena sebagian bakteri perusak / pembusuk susu akan mati jika telah melewati proses pasteurisasi tersebut. Di zaman sekarang ini di mana teknologi yang semakin canggih seharusnya dapat mengolah susu dengan kualitas yang lebih baik.

Atas dasar permasalahan di atas, maka dalam skripsi ini penulis mengambil judul: “Sistem Pengendalian Suhu pada Proses Pasteurisasi Susu Berbasis Mikrokontroler Arduino Mega 2560”, dimana proses pengolahan susu yang digunakan dalam penelitian ini menggunakan metode *Low Temperature Low Time* (LTLT). Pasteurisasi dengan metode LTLT artinya, susu dipanaskan pada suhu 65⁰C selama 30 menit (Bakar A., 2001; 45–50). Penggunaan metode LTLT dimaksudkan agar susu dapat dikonsumsi dengan aman tanpa mengurangi nilai kandungan gizi susu itu sendiri. Dalam penelitian ini saya menggunakan

mikrokontroler arduino Mega 2560 sebagai perangkat pengontrol bejana pasteurisasi susu sedangkan untuk mengurangi error saya menggunakan kontroler *Propositional Integral* (PI).

1.2 RUMUSAN MASALAH

1. Bagaimana merancang dan membuat sistem pengontrolan suhu pasteurisasi susu yang sesuai?
2. Bagaimana merancang *software sistem* pengendalian suhu pasteurisasi susu?

1.3 BATASAN MASALAH

1. Perancangan dan pembuatan alat pengontrolan suhu pada proses pasteurisasi susu yang digunakan adalah *prototype* dengan desain sendiri.
2. Pembahasan ditekankan pada pengontrolan suhu pada sistem dengan menggunakan kontroler PI berbasis Mikrokontroler Arduino Mega 2560.
3. Metode yang digunakan untuk *tuning* parameter PI yang digunakan ada *hand tuning*.
4. Metode pasteurisasi yang digunakan adalah metode *Low Temperature Low Time* (LTLT).
5. Sensor suhu yang digunakan adalah *thermocouple* dengan rentang 0-1024° C.
6. Pembahasan mengenai bakteri, mekanik, model matematis, reaksi kimia, dan rangkaian elektronika tidak dibahas secara mendalam.

1.4 TUJUAN

Tujuan dari skripsi ini adalah untuk merancang sistem pengontrolan suhu pada sistem pasteurisasi susu dengan implementasi kontrol PI berbasis Mikrokontroler Arduino Mega 2560. Dalam hal ini penggunaan kontrol PI untuk menjaga suhu agar tetap konstan meskipun terdapat gangguan.

1.5 SISTEMATIKA PENULISAN

Skripsi ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

BAB I **Pendahuluan**

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika pembahasan.

BAB II **Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

BAB III Metodologi Penelitian

Membahas perancangan alat yang merupakan spesifikasi, perencanaan blok diagram, prinsip kerja dan pembuatan alat, serta memuat hasil pengujian terhadap alat yang telah dibuat.

BAB IV Hasil dan Pembahasan

Membahas hasil pengujian sistem yang sudah dibuat dan analisis hasil yang diperoleh.

BAB V Kesimpulan dan Saran

Membahas kesimpulan penelitian dan saran-saran yang diperlukan untuk pengembangan penelitian selanjutnya.





BAB II

TINJAUAN PUSTAKA

2.1 SUSU

Pada zaman dahulu, susu telah dipakai sebagai bahan pokok pangan manusia. Manusia mengambil susu dari hewan yang memiliki kelenjar susu, seperti sapi, kuda dan domba. Sapi dan domba mulai dijinakkan sejak 8000 SM untuk diambil daging, bulu dan susunya. Di Timur Tengah, susu bahkan terfermentasi menjadi keju oleh para pengembara gurun di sana. Diperkirakan susu mulai masuk ke dataran Eropa pada abad 5000 SM melewati daerah Anatolia. Sementara, susu mulai masuk ke Inggris pada periode Neolitik.

Susu adalah cairan bergizi berwarna putih yang dihasilkan oleh kelenjar susu mamalia, salah satunya manusia. Susu adalah sumber gizi utama bagi bayi sebelum mereka dapat mencerna makanan padat. Susu binatang (biasanya sapi) juga diolah menjadi berbagai produk seperti mentega, yogurt, es krim, keju, susu kental manis, susu bubuk dan lain-lainnya untuk konsumsi manusia.

Dewasa ini, susu memiliki banyak fungsi dan manfaat. Untuk umur produktif, susu membantu pertumbuhan mereka. Sementara itu, untuk orang lanjut usia, susu membantu menopang tulang agar tidak keropos. Susu secara alami mengandung nutrisi penting, seperti bermacam-macam vitamin, protein, kalsium, magnesium, fosfor, dan *zinc*, pendapat lain menambahkan bahwa susu mengandung mineral dan lemak. Oleh karena itu, setiap orang dianjurkan minum susu. Sekarang banyak susu yang dikemas dalam bentuk yang unik. Tujuan dari ini agar orang tertarik untuk membeli dan minum susu. Ada juga susu yang berbentuk fermentasi.

Penggunaan keju dan susu dari Timur Tengah lewat Turki mulai dikenal oleh bangsa Eropa pada zaman pertengahan. Kemudian, pada abad ke-15, para pelaut mulai membawa sapi perah untuk dipelihara dan ditenakkan di dataran Eropa untuk konsumsi susu. Susu sapi sendiri baru dikenal oleh bangsa Indonesia lewat penjajahan Hindia Belanda pada abad ke 18 (Discarara M., 2014).

2.2 Pasteurisasi

Pasteurisasi adalah pemanasan susu dengan suhu dengan waktu tertentu. Pemanasan dengan suhu pasteurisasi dimaksudkan untuk membunuh sebagian kuman *patogenik* yang ada pada susu, dengan seminimum mungkin kehilangan gizinya dan mempertahankan semaksimal mungkin sifat fisik dan cita rasa susu segar (Purnomo & Adiono, 1987).

Tes pasteurisasi pertama diselesaikan oleh Pasteur dan Claude Bernard pada 20 April 1862. Ada definisi lain yang menyebutkan, pasteurisasi adalah perlakuan panas yang diberikan pada bahan baku dengan suhu di bawah titik didih. Proses pasteurisasi merupakan proses pemanasan dengan suhu yang relatif cukup rendah (dibawah 100°C) dengan tujuan untuk menginaktivasi enzim dan membunuh mikroba pembusuk.

Pada suhu dan waktu tertentu, bakteri patogen akan mati. Pasteurisasi tidak berarti sterilisasi, tetapi mematikan semua bakteri patogen, ragi, jamur dan juga sebagian besar sel *vegetative* pada bakteri. Bakteri yang tahan hidup dapat diklasifikasikan sebagai organisme yang tahan panas atau *thermoduric*, diantaranya bakteri asam laktat seperti *Streptococcus thermophilus*, *Lactobacillus lactis*, *Lactobacillus thermophilus*, beberapa jenis *Micrococcus*, *Bacillus* dan *Clostridium*. Pasteurisasi menghancurkan 90-99% bakteri yang ada didalam susu dengan kemungkinan kerusakan yang sangat kecil bagi laktosa, casein dan unsur lemak, akan tetapi vitamin C dapat dirusak oleh cara ini.

Pasteurisasi digunakan untuk mengawetkan bahan pangan yang tidak tahan suhu tinggi, misalnya susu. Pasteurisasi tidak mematikan semua mikroorganisme, tetapi hanya yang bersifat patogen dan tidak membentuk spora. Oleh sebab itu, proses ini sering diikuti dengan teknik lain misalnya pendinginan atau pemberian gula (sukrosa) dengan konsentrasi tinggi. Produk hasil pasteurisasi bila disimpan pada suhu kamar hanya bertahan 1 sampai 2 hari sedang jika disimpan pada suhu rendah dapat tahan 1 minggu.

2.2.1 Jenis

Dilihat dari ketinggian suhu pada proses pasteurisasi dilakukan, dikenal beberapa teknik dalam melakukan pasteurisasi, yaitu:

2.2.1.1 Pasteurisasi model HTST

High Temperature Short Time (HTST) adalah suatu proses pemanasan dengan suhu tinggi dalam waktu singkat. Pemanasan pada model HTST ini dilakukan pada suhu 75°C selama 15 detik. Dalam proses pasteurisasi model HTST ini menggunakan alat yang disebut *heat plate exchanger* atau semacam perubah suhu tinggi (Bakar A., 2001; 45–50).

2.2.1.2 Pasteurisasi model UHT

Ultra High Temperature (UHT) adalah suatu proses pemanasan dengan suhu sangat tinggi dalam lebih singkat lagi. Pemanasan model UHT ini dilakukan dalam suhu 130°C selama hanya 0,5 detik saja. Pemanasan dilakukan dalam tekanan tinggi. Melalui proses ini seluruh mikroba yang terdapat dalam makanan dan minum mati, sehingga produk susu yang dipanaskan dengan UHT ini sering pula dikenal dengan nama susu steril (Bakar A., 2001; 45–50).

2.2.1.3 Pasteurisasi model LTLT

Low Temperature Long Time (LTLT) adalah metode pemanasan dengan suhu rendah dalam waktu cukup lama. LTLT dilakukan pada suhu rendah sekitar 60°C dalam waktu 30 menit. Perbedaan tinggi rendahnya suhu dalam pasteurisasi tersebut berbeda pula pada umur atau ketahanan makanan dan minum yang dipasteurisasi. Susu yang menggunakan pasteurisasi HTST misalnya, bisa tahan selama 1 minggu tanpa mengubah rasa. Sementara susu yang dipanaskan dengan sistem UHT bisa tahan sampai dengan 6 bulan (Bakar A., 2001; 45–50).

2.2.1.4 Kelebihan Pasteurisasi

Susu murni banyak digemari karena lebih alami, organik, asli, dan segar. Hanya saja harus dibangun kesadaran pada para konsumen susu bahwa susu murni mengandung banyak bakteri yang berpotensi menyebabkan penyakit pada manusia. Meskipun sebagian orang masih meyakini bahwa susu murni lebih baik karena lebih kaya akan vitamin, mineral, dan enzim, para ahli kesehatan sangat menyarankan susu melalui proses pasteurisasi terlebih dahulu sebelum dikonsumsi. Faktanya, susu yang sudah dipasteurisasi terbukti lebih baik dan lebih sehat. Meskipun terkesan menghilangkan zat-zat baik dalam susu, sebenarnya proses pasteurisasi sama sekali tidak mengurangi nilai nutrisi susu tersebut, bahkan tidak mengubah rasanya. Susu pasteurisasi lebih baik dikonsumsi daripada susu murni bagi mereka yang alergi laktosa. Hanya saja susu pasteurisasi dan susu murni sama-sama berpotensi menimbulkan alergi pada mereka yang sensitif terhadap protein susu.

Susu hasil pasteurisasi tidak begitu saja bisa tahan lama. Dalam jangka waktu tertentu susu pasteurisasi pun tetap akan kedaluwarsa, apalagi jika disimpan di suhu ruangan. Namun, susu pasteurisasi memiliki kelebihan yang tidak dimiliki susu murni, yakni ketiadaan bakteri-bakteri berbahaya yang berpotensi menularkan penyakit. Oleh karena itu, sebaiknya konsumsi susu pasteurisasi daripada susu murni. Bagi beberapa jenis orang, susu murni bahkan tidak disarankan untuk dikonsumsi, seperti bagi ibu hamil, bayi dan balita, lansia, orang dengan HIV AIDS, orang dengan kanker, dan orang yang telah menjalani transplantasi organ tubuh. Selain kelebihan di atas, proses pasteurisasi juga memiliki kelebihan lain antara lain:

1. Proses Pasteurisasi dapat membunuh bakteri patogen, yaitu bakteri yang berbahaya karena dapat menimbulkan penyakit pada manusia. Bakteri pada susu yang bersifat patogen misalnya *Mycobacterium tuberculosis* dan *Coxiella burnetii* dan mengurangi populasi bakteri.
2. Proses Pasteurisasi dapat memperpanjang daya simpan bahan atau produk

3. Proses Pasteurisasi dapat menimbulkan citarasa yang lebih baik pada produk
4. Pada susu proses ini dapat menginaktifkan enzim *fosfatase* dan *katalase* yaitu enzim yang membuat susu cepat rusak.

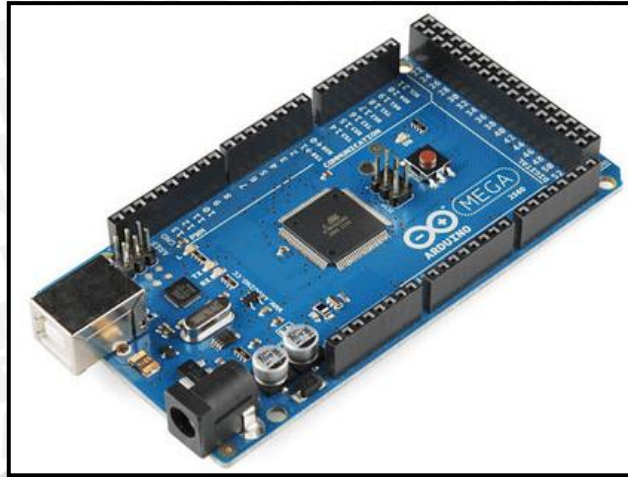
2.2.1.5 Kekurangan Pasteurisasi

Proses pasteurisasi dengan penanganan suhu yang tidak tepat dapat mengakibatkan *loss nutrition*, yaitu hilangnya nutrisi-nutrisi penting yang terkandung dalam susu. Penanganan suhu yang salah juga dapat mengakibatkan bakteri patogen yang tetap hidup di dalam susu, sehingga mengakibatkan ketahanan susu menjadi berkurang, serta beresiko menyebarkan bakteri ke dalam tubuh manusia.(Listia H.,2013)

2.3 Arduino Mega

Arduino Mega 2560 adalah *board* mikrokontroler berbasis ATmega 2560. Arduino Mega memiliki 54 pin *Input/Output* digital (dimana 15 pin diantaranya dapat digunakan sebagai *output* PWM), 16 *input* analog, 4 UARTs (port serial perangkat keras), 16 MHz osilator Kristal, koneksi USB, *power jack*, *ICSP header*, dan tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Arduino Mega 2560 ke komputer dengan menggunakan kabel USB atau dengan memberi daya dengan adaptor AC-to-DC atau baterai agar dapat bekerja. Arduino Mega 2560 juga cocok menggunakan sebagian besar *shield* yang didesain untuk Arduino *Duemilanove*.

Arduino Mega 2560 berbeda dengan *board* Arduino sebelumnya dimana *board* sebelumnya menggunakan FTDI USB-to-serial *driver chip*. Pada Arduino Mega 2560 terdapat ATmega 16U2 (sebagai ganti dari ATmega8U2 pada board sebelumnya) yang diprogram sebagai USB-to-serial *converter*. Arduino Mega 2560 juga memiliki *RESET* sirkuit yang lebih kuat dibandingkan *board* sebelumnya. Fitur baru yang terdapat pada Arduino Mega 2560 yaitu penambahan pin SDA dan SCL yang berdekatan dengan pin AREF dan dua pin baru lainnya diletakkan dekat pin RESET, pin pertama yaitu IOREF yang memperbolehkan *shield* untuk beradaptasi dengan tegangan yang disediakan oleh *board*. Kedepannya, *shield* akan bekerja sangat baik dengan *board* yang menggunakan AVR yang dapat beroperasi dengan 5V dan dengan 3,3V. Pin kedua adalah pin yang tidak terhubung yang disediakan untuk tujuan yang akan datang. Gambar 2.1 merupakan tampak depan dari Arduino Mega 2560-R3.



Gambar 2. 1 Arduino Mega2560-R3

Sumber: (arduino.cc)

2.3.1 Daya

Arduino Mega 2560 dapat diaktifkan melalui koneksi *Universal Serial Bus* (USB) atau dengan menggunakan catu daya eksternal. Daya eksternal (non-USB) dapat berasal dari adaptor AC-to-DC atau baterai. Adaptor dapat dihubungkan dengan menancapkan *plug jack* dengan pusat-positif sebesar 2,1 mm pada *power jack* pada *board*. Untuk baterai, ujung kepala baterai dapat dimasukkan pada pin Gnd dan Vin pada *header* di bagian konektor *power*. Board Arduino Mega 2560 ini dapat beroperasi dengan catu daya eksternal antara 6 sampai dengan 20 volt. Jika diberi daya kurang dari 7V, kemungkinan pin 5V tetap dapat beroperasi tetapi tidak stabil. Jika menggunakan daya lebih dari 12V, maka regulator tegangan akan panas dan dapat merusak *board* Arduino Mega 2560. *Range* daya yang direkomendasikan antara 7 sampai dengan 20 volt. Pin *power* pada *board* Arduino Mega 2560 diantaranya adalah:

1. **VIN**, Tegangan masukan untuk *board* Arduino ketika menggunakan catu daya eksternal (berbeda dengan 5V yang berasal dari konektor USB atau sumber tegangan yang telah disesuaikan).
2. **5V**, Pin *output* ini mengeluarkan *output* sebesar 5V yang telah disesuaikan menggunakan regulator yang berasal dari *board* Arduino. *Board* Arduino dapat dicatu dengan daya yang berasal dari *power jack* DC (7-12V), konektor USB (5V), atau pin VIN yang terdapat pada *board* (7-12V). Mencatu daya pada pin 5V dan 3,3V akan merusak regulator dan *board* Arduino.
3. **3,3V**, Merupakan catu daya sebesar 3,3V yang dihasilkan oleh regulator pada *board* Arduino.
4. **GND**, Merupakan pin *ground*.

5. **IOREF**, Pada board *Arduino* pin ini menyediakan tegangan referensi yang dioperasikan oleh mikrokontroler. *Shield* yang telah dikonfigurasi dengan baik dapat membaca tegangan pin IOREF dan dapat memilih catu daya yang sesuai atau dapat mengaktifkan tegangan translasi pada *output* yang bekerja pada 5V atau 3,3V.

2.3.2 Memori

ATmega2560 memiliki *flash memory* sebesar 256KB untuk penyimpanan kode (dimana 8KB digunakan sebagai bootloader), 8KB untuk SRAM dan 4KB untuk EEPROM (yang dapat dibaca dan ditulis dengan *library* EEPROM).

2.3.3 Input dan Output

Setiap pin dari 54 pin digital pada *Arduino Mega 2560* dapat digunakan sebagai *input* dan *output* menggunakan fungsi *pinMode*, *digitalWrite*, dan *digitalRead*. Setiap pin beroperasi pada tegangan 5V. Masing-masing pin dapat menyediakan atau menerima arus maksimal sebesar 40 mA dan memiliki resistor *pull-up* internal dengan 20-50k Ω . Selain itu, beberapa pin memiliki fungsi special seperti:

1. Serial 0:0 (RX) dan 1 (TX); Serial 1:19 (RX) dan 18 (TX); Serial 2:17 (RX) dan 16 (TX); Serial 3:15 (RX) dan 14 (TX). Pin (RX) digunakan untuk menerima dan pin (TX) untuk mentransmisikan data TTL serial. Pin 0 dan 1 juga terhubung dengan pin koresponden dari ATmega16U2 USB-to-TTL serial *chip*.
2. Eksternal Interrupt: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), dan 21 (interrupt 2). Pin tersebut dapat dikonfigurasi untuk memicu interrupt pada kondisi *LOW*, tepi naik atau turun, dan pada kondisi *HIGH*. Untuk lebih jelasnya dapat dilihat pada fungsi *attachInterrupt*.
3. PWM: 2-13 dan 44-46. Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite*.
4. SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin tersebut mendukung komunikasi SPI dengan menggunakan *library* SPI. Pin SPI juga terdapat pada ICSP header yang secara fisik dapat bekerja dengan baik untuk Uno, Duemilanove, Diecimila.
5. LED: 13. Merupakan LED yang terpasang pada board dan terhubung dengan pin *digital* 13. LED akan menyala saat nilai *HIGH* dan akan mati ketika pin bernilai *LOW*.
6. TWI: 20 (SDA) dan 21 (SCL). Mendukung komunikasi TWI menggunakan *Wire library*. Pin tersebut tidak terletak pada posisi yang sama dengan pin TWI pada Duemilanove atau Diecimila.

Arduino Mega 2560 memiliki 16 analog *input* yang masing-masing menyediakan resolusi 10 bit (1024 yang memiliki nilai berbeda). Pada dasarnya nilai yang terukur dari

ground hingga 5V, yang titik tertingginya dapat diubah menggunakan pin AREF dan fungsi *analogReference*. Berikut adalah beberapa pin *analog* yang terdapat pada board:

1. AREF, Merupakan tegangan referensi untuk *input analog* digunakan dengan *analogReference*.
2. Reset, Membawa garis *LOW* untuk me-reset mikrokontroler. Pada umumnya digunakan untuk menambahkan tombol reset untuk membatasi tombol reset yang berada pada *board*. (Arisandi R.,2014)

2.3.4 Komunikasi

Arduino Mega 2560 memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, Arduino yang lain, atau mikrokontroler lain. ATmega2560 menyediakan empat perangkat UART untuk komunikasi *serial TTL* (5V). ATmega16U2 pada *board* merupakan saluran untuk USB dan menyediakan *port com* virtual untuk *software* pada komputer. Pada Windows diperlukan sebuah file dengan tipe .inf; tetapi tidak pada OSX atau Linux, kedua *Operating Sistem* ini akan mengidentifikasi atau mengenali secara otomatis *board* Arduino sebagai *port COM*. *Software* Arduino menyediakan serial monitor yang memperbolehkan data tekstual untuk dikirimkan baik ke *board* ataupun dari *board*. LED RX dan TX pada *board* akan berkedip saat data dikirimkan melalui chip ATmega8U2/ ATmega16U2 dan koneksi USB ke konektor (namun tidak untuk komunikasi serial pada pin 0 dan 1).

SoftwareSerial library memperbolehkan komunikasi serial antara beberapa pin *digital* pada Arduino Mega2560. ATmega2560 juga mendukung komunikasi TWI dan SPI. *Software* Arduino menyediakan *Wire library* untuk memudahkan bus TWI dan komunikasi SPI. (Arisandi R.,2014)

2.4 Kontroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolok ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespon *input*, dan bagaimana peredaman terhadap adanya lonjakan (*overshoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *outputnya* berosilasi terus menerus ketika

dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respon transien berupa suatu osilasi teredam sebelum mencapai *steady state*. Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran plan dengan nilai referensi. Kemudian kontroler menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata K., 1997).

2.4.1 Kontroler Proposional

Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

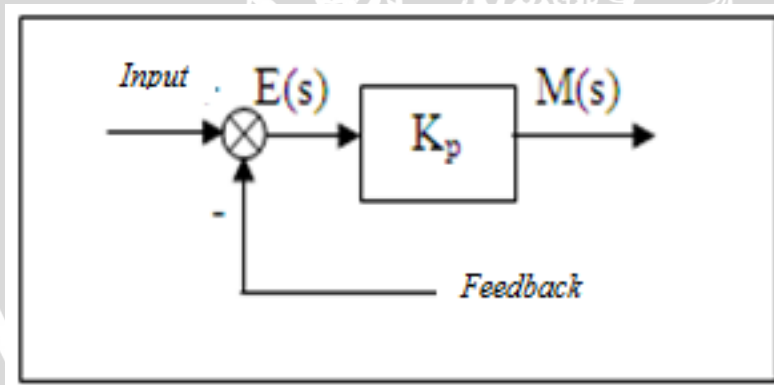
$$m(t) = K_p e(t) \dots \dots \dots (2.1)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots \dots \dots (2.2)$$

di mana K_p adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerak, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997). Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.2.



Gambar 2. 2 Diagram Blok Kontroler Proposional
Sumber: (Ogata K., 1997)

2.4.2 Kontroler Integral

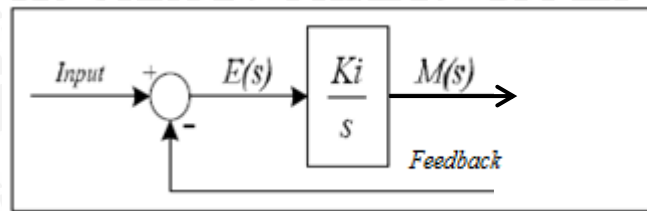
Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$.

Jadi,

$$\frac{dm(t)}{dt} = K_i e(t) \dots \dots \dots (2.3)$$

dengan K_i adalah konstanta integral. Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga

$m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* yang digunakan untuk menghilangkan *error steady state* (Ogata K.,1997). Diagram blok kontroler integral ditunjukkan dalam Gambar 2.3.



Gambar 2.3 Diagram Blok Kontroler Integral
Sumber: (Ogata K., 1997)

2.4.3 Kontroler Proporsional Integral

Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler.

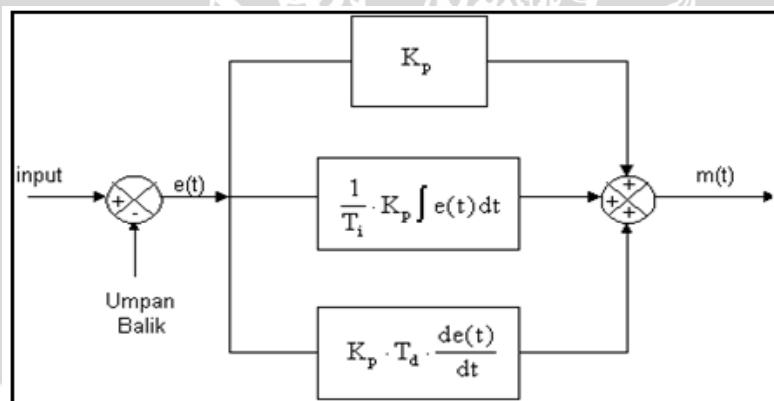
Persamaan kontroler PID ini dapat dinyatakan sebagai berikut :

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \dots \dots \dots (2.5)$$

Dalam transformasi Laplace dinyatakan sebagai berikut :

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \dots \dots \dots (2.6)$$

dengan T_i adalah waktu integral, dan T_d adalah waktu derivatif. Gambar 2.4 menunjukkan diagram blok kontroler PID



Gambar 2. 4 Diagram Blok Kontroler PID
Sumber: (Ogata K., 1997)

2.5 Motor DC Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat diatur untuk menentukan dan memastikan posisi sudut dari poros *output* motor. Motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian *gear*, rangkaian kontrol dan potensiometer. Serangkaian *gear* yang melekat pada poros motor DC akan memperlambat

putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo.

Penggunaan sistem kontrol loop tertutup pada motor servo berguna untuk mengontrol gerakan dan posisi akhir dari poros motor servo. Penjelasan sederhananya begini, posisi poros *output* akan di-*feedback* untuk mengetahui posisi poros sudah tepat seperti yang diinginkan atau belum, dan jika belum, maka kontrol *input* akan mengirim sinyal kendali untuk membuat posisi poros tersebut tepat pada posisi yang diinginkan. Untuk lebih jelasnya mengenai sistem kontrol loop tertutup, perhatikan contoh sederhana beberapa aplikasi lain dari sistem kontrol loop tertutup, seperti penyetelan suhu pada AC, kulkas, setrika dan lain sebagainya.

Motor servo biasa digunakan dalam aplikasi-aplikasi di industri, selain itu juga digunakan dalam berbagai aplikasi lain seperti pada mobil mainan radio kontrol, robot, pesawat, dan lain sebagainya. Berikut bentuk fisik motor DC servo SG-5010 dapat dilihat pada Gambar 2.5. (Arisandi R.,2014)



Gambar 2.5 Motor Servo

Sumber: (continentale.com)

Ada dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan, motor servo DC biasanya lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil dan bila dibedakan menurut rotasinya, umumnya terdapat dua jenis motor servo yang terdapat di pasaran, yaitu motor *servo rotation* 180⁰ dan *servo rotation continuous*.

Motor servo standard (*servo rotation* 180⁰) adalah jenis yang paling umum dari motor servo, dimana putaran poros *output*nya terbatas hanya 90⁰ kearah kanan dan 90⁰ kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180⁰.

Motor servo rotation continuous merupakan jenis motor servo yang sebenarnya sama dengan jenis servo standard, hanya saja perputaran porosnya tanpa batasan atau dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (Pulse Wide Modulation / PWM) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros motor servo ke posisi sudut 90. Bila pulsa lebih pendek dari 1,5 ms maka akan berputar ke arah posisi 0 atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar ke arah posisi 180 atau ke kanan (searah jarum jam). (Arisandi R.,2014)

2.6 Sensor *Thermocouple*

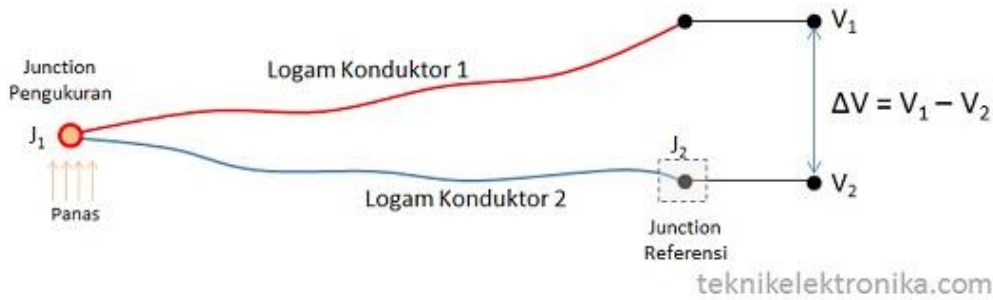
Thermocouple adalah jenis sensor suhu yang digunakan untuk mendeteksi atau mengukur suhu melalui dua jenis logam konduktor berbeda yang digabung pada ujungnya sehingga menimbulkan Efek *Thermoelectric*. Efek *Thermoelectric* pada *thermocouple* ini ditemukan oleh seorang fisikawan Estonia bernama Thomas Johann Seebeck pada Tahun 1821, dimana sebuah logam konduktor yang diberi perbedaan panas secara gradient akan menghasilkan tegangan listrik. Perbedaan Tegangan listrik diantara dua persimpangan (*junction*) ini dinamakan dengan Efek *Seeback*.

Thermocouple merupakan salah satu jenis sensor suhu yang paling populer dan sering digunakan dalam berbagai rangkaian ataupun peralatan listrik dan Elektronika yang berkaitan dengan Suhu . Beberapa kelebihan *thermocouple* yang membuatnya menjadi populer adalah responnya yang cepat terhadap perubahan suhu dan juga rentang suhu operasionalnya yang luas yaitu berkisar diantara -200°C hingga 2000°C . Selain respon yang cepat dan rentang suhu yang luas, *Thermocouple* juga tahan terhadap guncangan/getaran dan mudah digunakan.

2.6.1. Prinsip Kerja *Thermocouple*

Prinsip kerja *thermocouple* cukup mudah dan sederhana. Pada dasarnya *thermocouple* hanya terdiri dari dua kawat logam konduktor yang berbeda jenis dan digabungkan ujungnya. Satu jenis logam konduktor yang terdapat pada *thermocouple* akan berfungsi sebagai referensi dengan suhu konstan (tetap) sedangkan yang satunya lagi sebagai logam konduktor yang mendeteksi suhu panas. Berikut ini dijelaskan prinsip kerja sensor *thermocouple* seperti pada gambar

2.6



Gambar 2.6 Prinsip kerja sensor *thermocouple*

Sumber : (teknikelektronika.com)

Berdasarkan Gambar 2.6, ketika kedua persimpangan atau *Junction* memiliki suhu yang sama, maka beda potensial atau tegangan listrik yang melalui dua persimpangan tersebut adalah “NOL” atau $V_1 = V_2$. Akan tetapi, ketika persimpangan yang terhubung dalam rangkaian diberikan suhu panas atau dihubungkan ke obyek pengukuran, maka akan terjadi perbedaan suhu diantara dua persimpangan tersebut yang kemudian menghasilkan tegangan listrik yang nilainya sebanding dengan suhu panas yang diterimanya atau $V_1 - V_2$. Tegangan Listrik yang ditimbulkan ini pada umumnya sekitar $1 \mu\text{V} - 70 \mu\text{V}$ pada tiap derajat Celcius. Tegangan tersebut kemudian dikonversikan sesuai dengan Tabel referensi yang telah ditetapkan sehingga menghasilkan pengukuran yang dapat dimengerti oleh kita. (Siregar I.,2009)

2.6.2. Jenis-jenis Sensor *Thermocouple*

Thermocouple tersedia dalam berbagai ragam rentang suhu dan jenis bahan. Pada dasarnya, gabungan jenis-jenis logam konduktor yang berbeda akan menghasilkan rentang suhu operasional. Berikut ini adalah penampang sensor *thermocouple* seperti pada gambar 2.7.



Gambar 2.7 Bentuk penampang sensor *thermocouple*

Sumber : (teknikelektronika.com)

Dibawah ini dijelaskan beberapa macam tipe-tipe *thermocouple* beserta bahan penyusunnya dan rentang suhu sensor ini dapat bekerja dengan baik.

1. Thermocouple Tipe E

Bahan Logam Konduktor Positif : *Nickel-Chromium*

Bahan Logam Konduktor Negatif : *Constantan*

Rentang Suhu : $-200^{\circ}\text{C} - 900^{\circ}\text{C}$

2. Thermocouple Tipe J

Bahan Logam Konduktor Positif : *Iron*

Bahan Logam Konduktor Negatif : *Constantan*

Rentang Suhu : $0^{\circ}\text{C} - 750^{\circ}\text{C}$

3. Thermocouple Tipe K

Bahan Logam Konduktor Positif : *Nickel-Chromium*

Bahan Logam Konduktor Negatif : *Nickel-Aluminium*

Rentang Suhu : $-200^{\circ}\text{C} - 1250^{\circ}\text{C}$

4. Thermocouple Tipe N

Bahan Logam Konduktor Positif : *Nicrosil*

Bahan Logam Konduktor Negatif : *Nisil*

Rentang Suhu : $0^{\circ}\text{C} - 1250^{\circ}\text{C}$

5. Thermocouple Tipe T

Bahan Logam Konduktor Positif : *Copper*

Bahan Logam Konduktor Negatif : *Constantan*

Rentang Suhu : $-200^{\circ}\text{C} - 350^{\circ}\text{C}$

6. Thermocouple Tipe U (kompensasi Tipe S dan Tipe R)

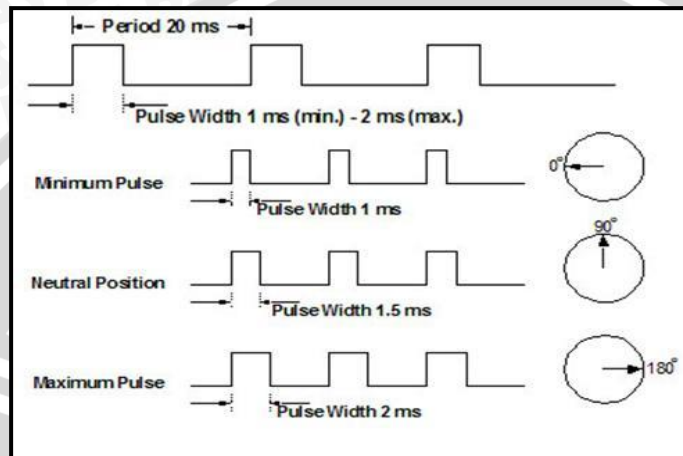
Bahan Logam Konduktor Positif : *Copper*

Bahan Logam Konduktor Negatif : *Copper-Nickel*

Rentang Suhu : $0^{\circ}\text{C} - 1450^{\circ}\text{C}$ (Siregar I.,2009)

2.7 PWM (Pulse Width Modulation)

PWM (Pulse Width Modulation) memiliki fungsi untuk mengatur sudut putaran motor servo. Teknik PWM (*pulse width modulation*) untuk mengatur sudut motor servo dapat dilihat dalam Gambar 2.8.



Gambar 2.8 Pengaturan Sudut Motor Servo
Sumber: (electroons.com)

Berdasarkan Gambar 2.6, dapat diasumsikan bahwa saat diberikan sinyal periodik dengan lebar sebesar 1 ms maka motor servo akan bergerak dengan sudut 0° . Apabila diberikan sinyal sebesar 1.5 ms maka motor servo akan bergerak dengan sudut 90° . Begitu pula jika diberi sinyal 2 ms, maka motor servo akan bergerak dengan sudut 180° . Perhitungan rumus motor servo akan ditunjukkan Persamaan 2-1 sebagai berikut:

$$S = D \left(\frac{1000}{180} \right) + 1000 \mu\text{s} \quad (2-1)$$

Dimana:

S = Lebar Pulsa dalam μs

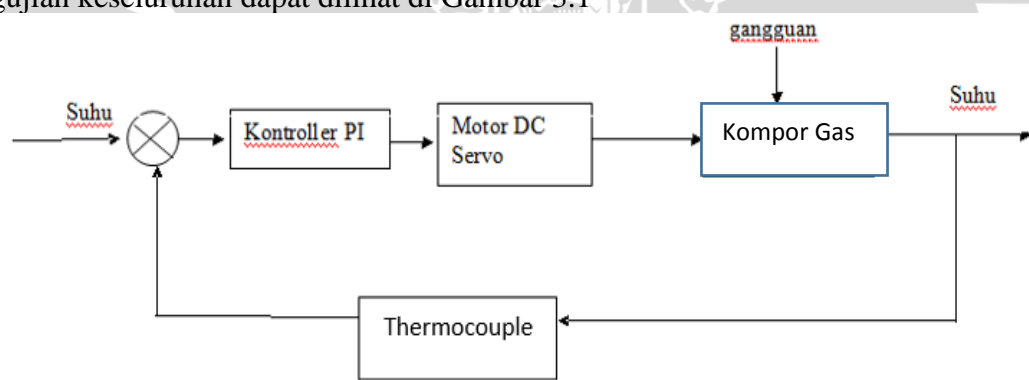
D = Sudut putar servo dalam derajat.

BAB III METODE PENELITIAN

Penyusunan skripsi ini didasarkan dalam masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiian sistem agar dapat bekerja sesuai dengan yang direncanakan. Langkah-langkah yang perlu dilakukan untuk merealisasiikan sistem yang dirancang adalah sebagai berikut:

3.1. Perancangan Blok Diagram Sistem

Perancangan diagram blok sistem pengaturan suhu bertujuan untuk mempermudah dalam peracacangan perangkat keras sistem maupun perancangan perangkat lunak sistem yang berhubungan dengan pengaturan suhu. Diagram blok yang dirancang untuk sistem pengujian keseluruhan dapat dilihat di Gambar 3.1



Gambar 3.1 Blok Diagram Sistem

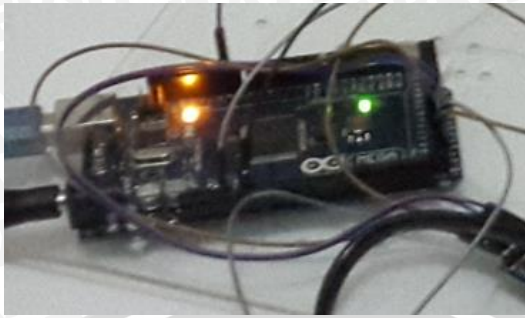
Keterangan :

1. *Setpoint* sistem berupa nilai suhu sebesar 65°C.
2. Kontroler yang digunakan adalah kontroler Proportional Integral (PI) menggunakan perangkat keras Arduino Mega 2560.
4. *Plant* yang digunakan adalah tabung pasteurisasi susu.
5. Sensor *Thermocouple* digunakan sebagai pembaca suhu tabung pasteurisasi susu.
6. Gangguan pada sistem berupa penambahan susu pada tabung pasteurisasi susu.

3.2. Spesifikasi Alat

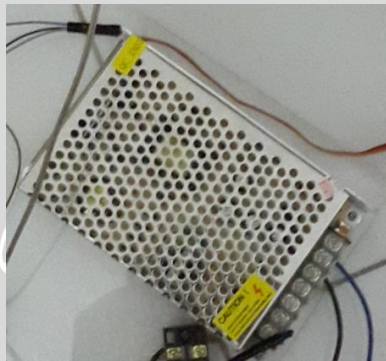
Spesifikasi alat yang digunakan pada skripsi ini adalah sebagai berikut :

1. Arduino Mega 2560 dipergunakan sebagai perangkat pengontrolan *plant* , seperti dalam Gambar 3.2.



Gambar 3.2 Arduino Mega 2560.

2. *Power Supply Switching* yang digunakan memiliki *range* 0 - 12V. *Power Supply* berfungsi sebagai sumber tegangan catu daya Arduino dimana tegangan *output* diatur sebesar 12V, seperti dalam Gambar 3.3.



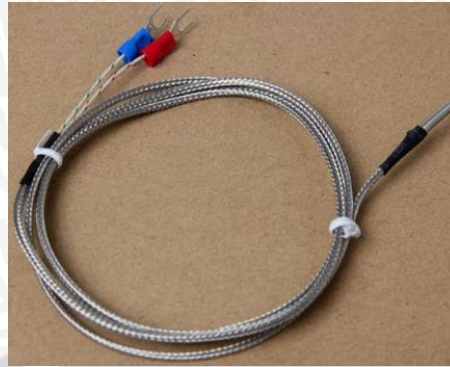
Gambar 3.3 Power Supply 12V.

3. Kompor gas *portable* digunakan sebagai alat pemanas dari tabung pasteurisasi susu, seperti yang ditunjukkan dalam Gambar 3.4.



Gambar 3.4 Kompor Portable.

4. *Thermocouple* digunakan sebagai perangkat pengukur suhu dalam proses pasteurisasi. *Thermocouple* ditunjukkan dalam Gambar 3.5.



Gambar 3.5 Thermocouple type K

5. Motor servo digunakan sebagai penggerak putaran kompor gas. Motor servo dapat dilihat dalam Gambar 3.6.



Gambar 3.6 Motor Servo SG-5010

3.3. Spesifikasi Desain

Desain yang diinginkan pada perancangan pengendalian sudut putaran kompor gas mempunyai spesifikasi yaitu

1. *Steady State* < 5%:

Error Steady State < 5%, karena sistem yang baik memiliki *output* dengan batas nilai akhir 5% dari *setpoint*.

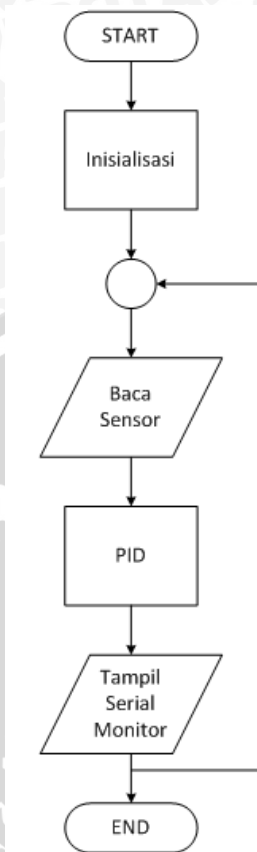
2. *Settling time* < 120 detik

Settling time < 120 detik, karena diharapkan penggunaan kontroler PI mampu mempercepat *settling time* sistem kurang dari 120 detik.

3. *Output* tanpa *overshoot*

Penggunaan kontroler PI diharapkan mampu meminimalkan *overshoot*.

3.4. Flowchart system



Gambar 3.7 Gambar *flowchart* sistem secara keseluruhan

Pengujian dan analisis dalam penelitian ini dilakukan untuk mengetahui apakah seluruh sistem bekerja sesuai dengan perancangan. Pengujian dilakukan pada masing-masing blok dalam perancangan *hardware* serta pengujian keseluruhan untuk mengetahui *software* dapat bekerja dengan baik atau tidak. Untuk bentuk *flowchart* keseluruhan system dapat dilihat di atas pada Gambar 3.7. Adapun serangkaian pengujian yang dilakukan pada sistem adalah sebagai berikut:

1. Pengujian sensor suhu *thermocouple type K*
2. Pengujian motor servo DC TowerPro SG5010
3. Pengujian dimmer listrik
4. Pengujian sistem secara keseluruhan

3.5. Pengujian sensor suhu *thermocouple type K*

Pengujian sensor suhu bertujuan untuk mengetahui seberapa besar *error* sensor yang digunakan dengan cara membandingkannya dengan suhu yang tertera pada thermometer digital.

3.5.1. Peralatan yang digunakan

Alat yang digunakan pada pengujian sensor suhu:

1. Sensor suhu *thermocouple type K*
2. Arduino Mega 2560
3. *Thermometer digital*

3.5.2. Langkah Pengujian

Langkah – langkah yang dilakukan untuk mendapatkan hasil yang diinginkan diantaranya :

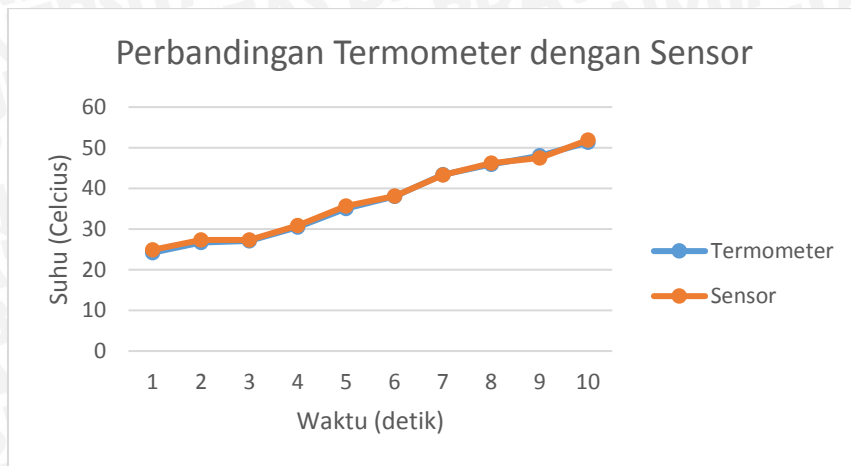
1. Sediakan air secukupnya dalam panci
2. Celupkan sensor dan thermometer kedalam panci
3. Buka *software* Arduino pada PC
4. *Compile listing* program dalam *software* Arduino
5. Buka serial monitor pada Arduino
6. Amati perubahan suhu pada serial monitor dan thermometer

3.5.3. Hasil Pengujian dan Analisa

Hasil pengujian pada sensor suhu *thermocouple type K* ditunjukkan dalam Tabel 3.1 dan grafik ditunjukkan dalam Gambar 3.8.

Tabel 3. 1 pengujian sensor suhu terhadap thermometer

Termometer °C	Sensor °C	Error(%)
24.2	24.9	2.89
26.7	27.3	2.24
27.1	27.3	0.73
30.5	30.9	1.31
35	35.7	0.57
38	38.1	0.26
43.4	43.3	0.23
45.9	46.2	0.65
48	47.5	1.05
51.3	51.9	1.16



Gambar 3. 8 pengujian sensor suhu *thermocouple* type k

3.6. Perancangan Kontroler

Pada penelitian kali ini untuk memenuhi tujuan performansi loop yang diinginkan, maka perlu ditambahkan kontroler dalam sistem. Kontrol yang dipilih adalah kontrol proporsional integral. Kontroler proporsional integral dipilih karena karakteristik *plant* tersebut tidak membutuhkan kontrol yang terlalu cepat tetapi meminimalkan *overshoot*.

Pencarian parameter kontroler PI dapat dicari menggunakan beberapa cara, antara lain *ziegler-nichols tuning*, *loop shaping*, metode analitis, optimasi, *pole placement*, *autotuning* dan *hand tuning* (Smith,1997; Astrom & Hagglund,2004). Dengan menggunakan metode *hand tuning*, nilai parameter PI yaitu K_p dan K_i perlu diubah-ubah secara *trial* dan *error* agar respon yang diperoleh sesuai dengan yang diinginkan.

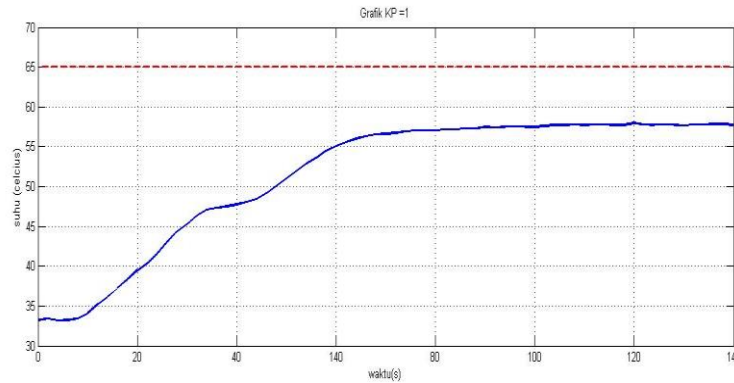
Metode *hand tuning* digunakan karena menghasilkan nilai sesuai dengan set point yaitu, nilai k_p , k_i dan k_d yang sesuai dengan yang diharapkan. Meskipun menggunakan metode *hand tuning* adapun beberapa prosedur yang harus dilakukan untuk mendapatkan hasil yang sesuai diantaranya :

1. Atur nilai K_p dan hilangkan nilai K_i dengan memasukkan nilai $K_i = 0$ hingga sistem mencapai keadaan *setpoint* seperti pada Tabel 3.2

Tabel 3.2 Tabel Penentuan parameter K_p

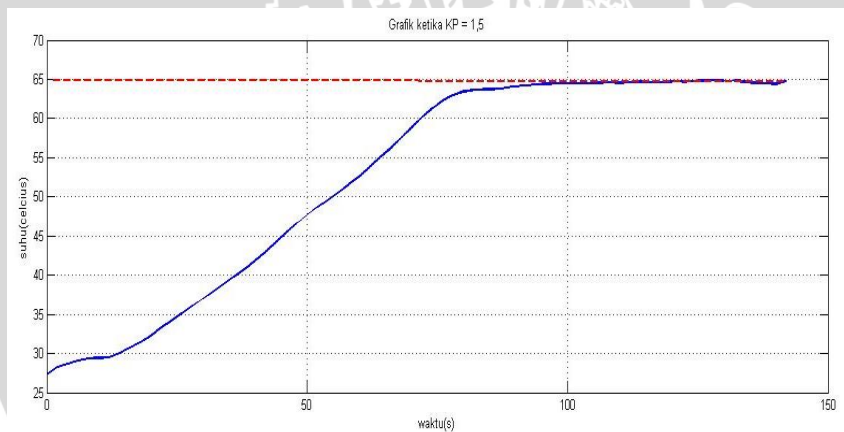
Pengujian Ke-	K_p	K_i	Settling Time (s)	Keterangan
1.	1	0	-	-
2.	1,5	0	108	Mampu mencapai <i>setpoint</i>
3.	2	0	80	Overshoot 1,92%

Berdasarkan Tabel 3.2 dapat dilihat dimana ketika nilai $K_p=1$ dan nilai $K_i=0$ sistem tidak mampu mencapai *setpoint* yang diinginkan seperti yang ditunjukkan dalam Gambar 3.9.



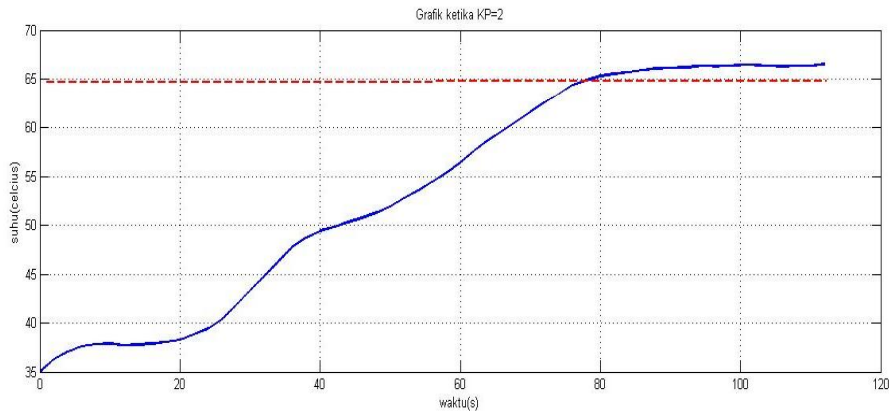
Gambar 3. 9 Gambar ketika $K_p=1$, $K_i=0$

Sedangkan bila K_p diberi nilai sebesar 1,5 sistem mampu mencapai *setpoint* yang diinginkan meskipun terdapat sedikit *error* seperti yang ditunjukkan dalam Gambar 3.10.



Gambar 3. 10 Gambar ketika $K_p=1,5$ dan $K_i =0$

Tetapi bila nilai K_p diberikan nilai sebesar 2, sistem mampu melewati *setpoint* tetapi terdapat *error* yang cukup besar bila dibandingkan dengan nilai K_p sebesar 1,5 seperti yang ditunjukkan dalam Gambar 3.11



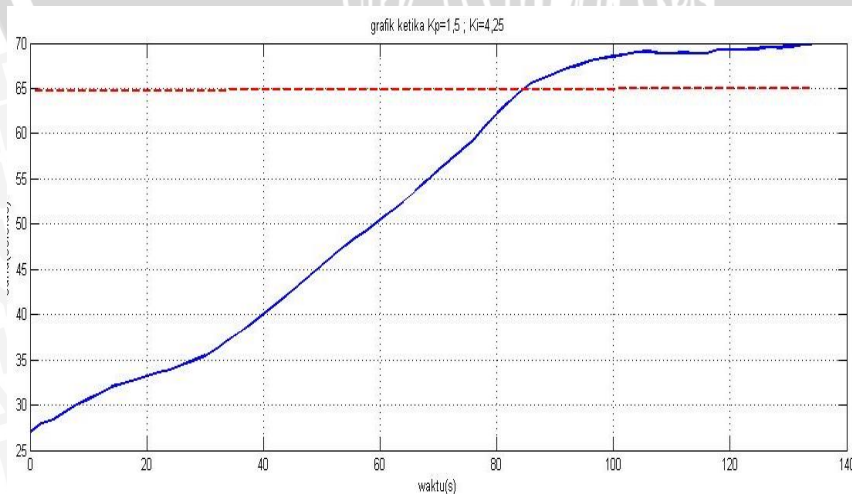
Gambar 3. 11 Gambar ketika $K_p=2, K_i =0$

- Atur nilai K_i untuk mengurangi *error* hingga mendapatkan respon yang diinginkan. Setelah mengatur beberapa nilai K_i , didapatkan data seperti yang ditunjukkan dalam Tabel 3.3.

Tabel 3. 3 Pengujian beberapa nilai K_i pada sistem

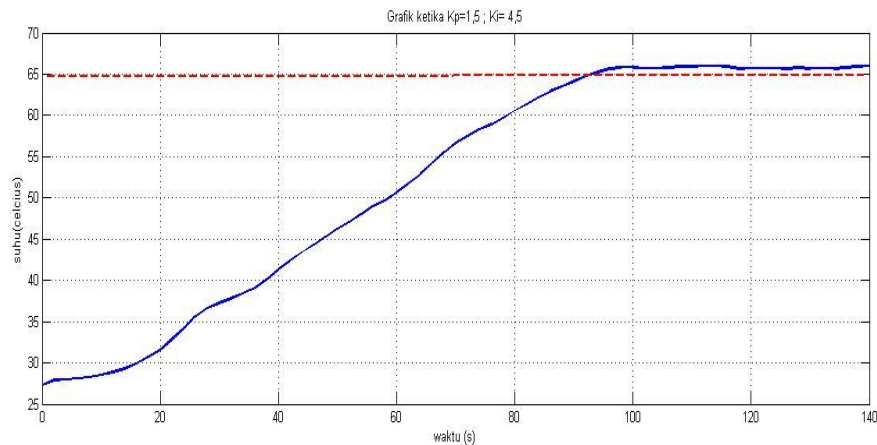
Pengujian ke-	K_p	K_i	Settling Time (s)	Mencapai <i>Setpoint</i>	Keterangan
1.	1,5	4,25	86	Ya	Ess(%)= 7,31 %
2.	1,5	4,5	94	Ya	Ess(%)= 1,53 %
3.	1,5	4,75	80	Ya	Ess(%)= 0,38 %

Berdasarkan Tabel 3.3, bila sistem diberikan nilai K_i sebesar 4,25 sistem mampu mencapai *setpoint* pada detik ke 86, tetapi terdapat *error steady state* sebesar 7,31% seperti yang ditunjukkan dalam Gambar 3.12.



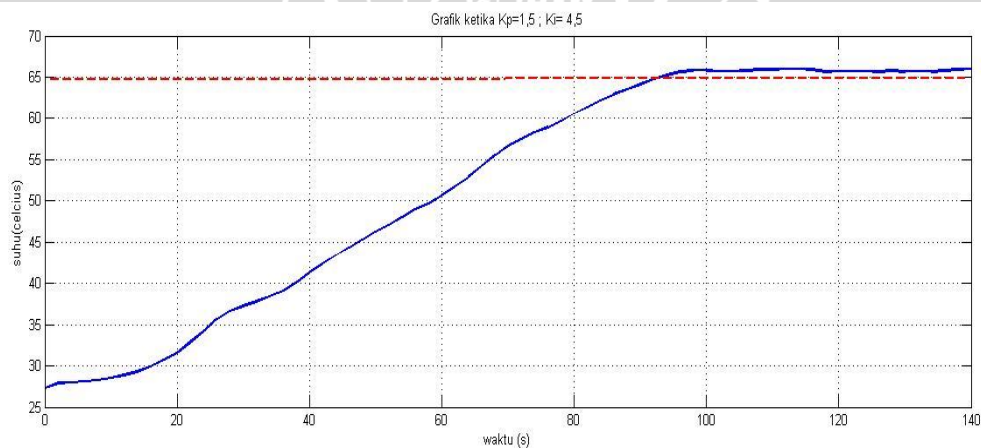
Gambar 3.12 Gambar ketika $K_p=1,5 K_i=4,25$

Berdasarkan Tabel 3.3, bila sistem diberikan nilai K_i sebesar 4,5 sistem mampu mencapai *setpoint* pada detik ke 94, tetapi terdapat *error steady state* sebesar 1,53% seperti yang ditunjukkan dalam Gambar 3.13.



Gambar 3.13 Gambar ketika $K_p=1,5$ $K_i=4,5$

Berdasarkan Tabel 3.3, bila sistem diberikan nilai K_i sebesar 4,75 sistem mampu mencapai *setpoint* pada detik ke 80, tetapi terdapat *error steady state* sebesar 0,38% seperti yang ditunjukkan dalam Gambar 3.14.



Gambar 3.14 Gambar ketika $K_p=1,5$, $K_i=4,75$

3.8 Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan data yang didapat dari hasil pengujian sistem secara keseluruhan. Apabila hasil yang didapatkan sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan dapat dikembangkan untuk penelitian selanjutnya untuk disempurnakan.



BAB IV

PENGUJIAN DAN ANALISIS SISTEM

4.1 Nilai penguatan (Gain)

a. Tujuan

Pengujian ini bertujuan untuk mencari nilai gain yang berfungsi sebagai penguat.

b. Peralatan yang digunakan

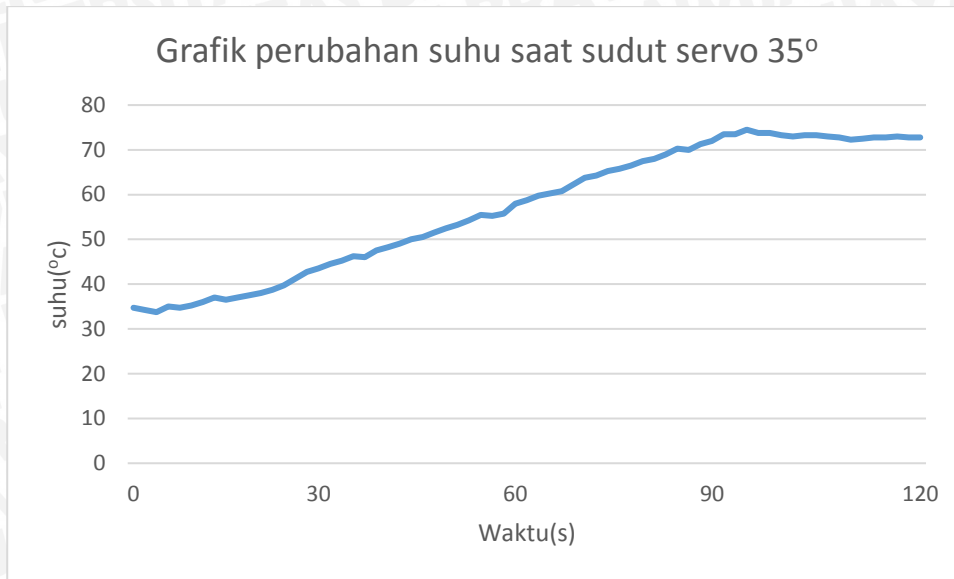
1. Kompor gas *portable*
2. Motor DC servo
3. Arduino Mega
4. Susu segar
5. Catu daya 12 VDC
6. Sensor suhu *thermocouple* type K
7. Tabung pasteurisasi
8. Laptop
9. Program dan software Arduino 1.6.9

c. Langkah pengujian

1. Merakit seperti dalam Gambar 4.8 dan 4.9
2. Buka software Arduino 1.6.9 dan jalankan program
3. Buka serial monitor pada Arduino
4. *Copy* data pada serial monitor ke *Microsoft Excel* kemudian masukkan data pada *chart*

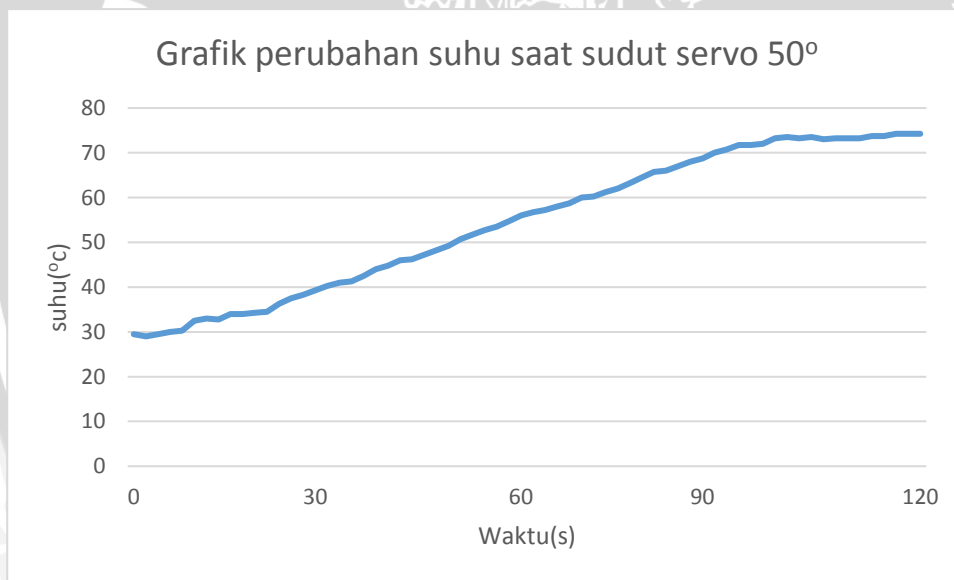
d. Hasil Pengujian

- Grafik hasil pengujian selama 2 menit ketika servo diatur dengan sudut 35° ditunjukkan dalam Gambar 4.1.



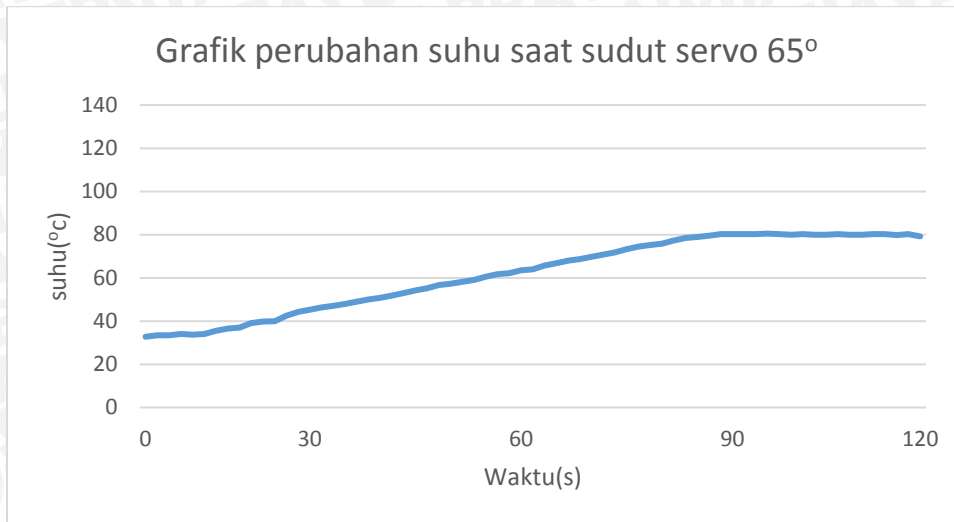
Gambar 4. 1 Grafik perubahan suhu saat sudut servo 35°

- Grafik hasil pengujian selama 2 menit ketika servo diatur dengan sudut 50° ditunjukkan dalam Gambar 4.2.



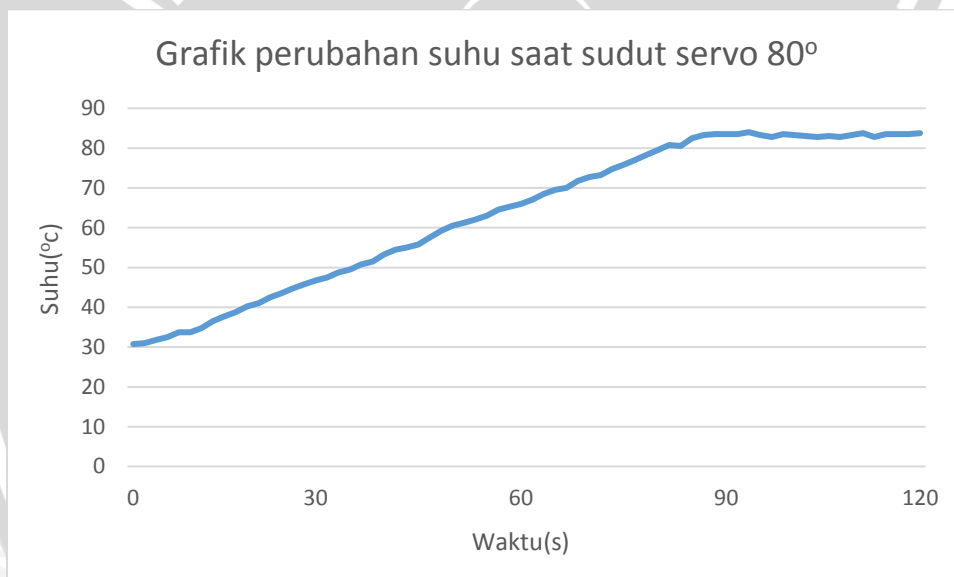
Gambar 4. 2 Grafik perubahan suhu saat sudut servo 50°

- Grafik hasil pengujian selama 2 menit ketika servo diatur dengan sudut 65° ditunjukkan dalam Gambar 4.3.



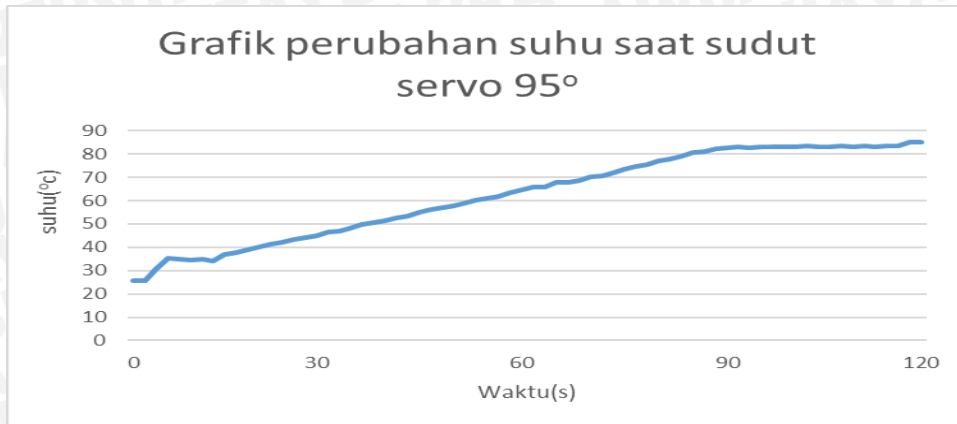
Gambar 4. 3 Grafik perubahan suhu saat sudut servo 65°

- Grafik hasil pengujian selama 2 menit ketika servo diatur dengan sudut 80° ditunjukkan dalam Gambar 4.4.



Gambar 4. 4 Grafik perubahan suhu saat sudut servo 80°

- Grafik hasil pengujian selama 2 menit ketika servo diatur dengan sudut 95° ditunjukkan dalam Gambar 4.5.



Gambar 4. 5 Grafik perubahan suhu saat sudut servo 95°

- Grafik hasil pengujian selama 2 menit ketika servo diatur dengan sudut 125° ditunjukkan dalam Gambar 4.6.



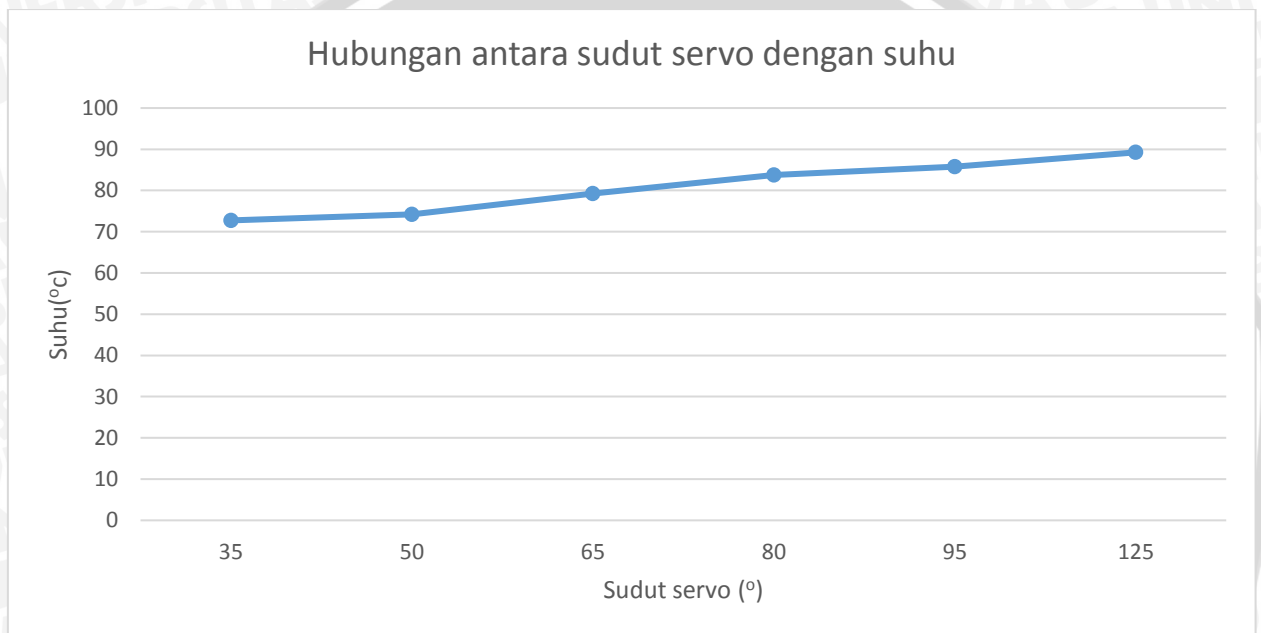
Gambar 4. 6 Grafik perubahan suhu saat sudut servo 125°

- Nilai *steady state* gain yang didapatkan dari hasil perhitungan nilai akhir suhu pada masing-masing nilai bukaan kompor yang telah diatur oleh sudut-sudut servo yang telah ditentukan seperti beberapa grafik diatas ditunjukkan pada Tabel 4.1.dan Gambar 4.7.

Tabel 4. 1 Grafik perubahan suhu terhadap perubahan sudut servo

Sudut servo(°)	Suhu(°C)
35	72.75

50	74.25
65	79.25
80	83.75
95	85.75
125	89.25



Gambar 4.7 Grafik hubungan antara perubahan suhu dan perubahan sudut servo

Dari perhitungan sesuai dengan grafik dan tabel diatas maka di dapatkan nilai gain(K) sebesar 0.166666667.

4.2 Pengujian Keseluruhan Sistem

a. Tujuan

Pengujian ini bertujuan untuk mengetahui kinerja sistem dengan mengamati respon sistem terhadap *setpoint*.

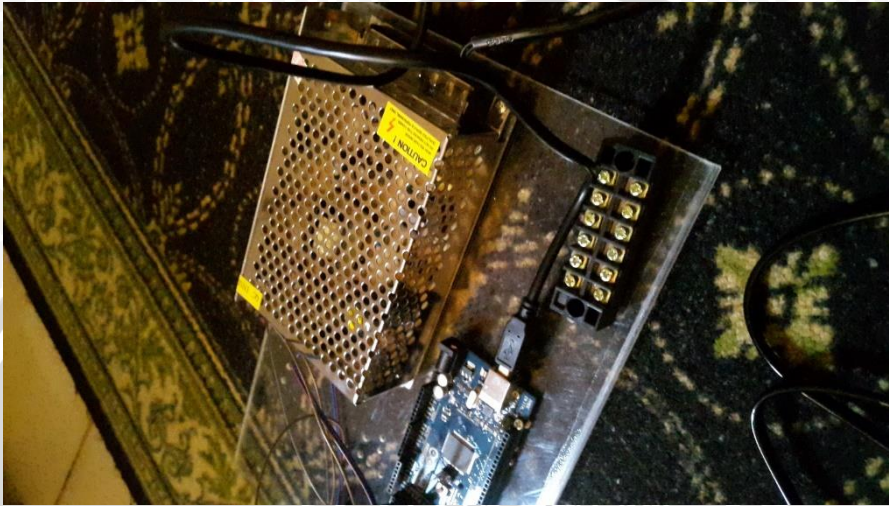
b. Peralatan yang Digunakan

- Kompor gas *portable*
- Motor DC *servo*
- Arduino Mega
- Susu segar
- Catu daya 12 VDC

- Sensor suhu *thermocouple type K*
- Tabung Pasteurisasi
- Laptop
- Program dan *software* Arduino

c. Langkah Pengujian

1. Merakit seperti dalam Gambar 4.8 dan 4.9



Gambar 4. 8 Rangkaian Elektrik Keseluruhan Sistem

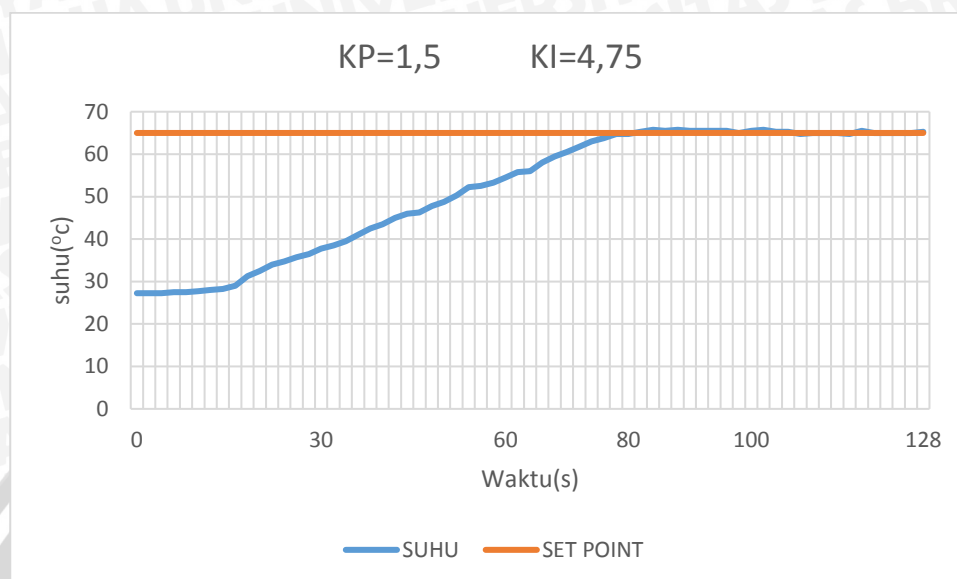


Gambar 4. 9 Tabung Pasteurisasi

2. Mengunduh program dengan nilai $K_p = 1,5$, $K_i = 4,75$
3. Melihat hasil keluaran nilai suhu dan menjadikan grafik terhadap waktu.

d. Hasil Pengujian

Setelah melaksanakan prosedur pengujian didapatkan respon sistem dari *plant*. Grafik hasil respon sistem seperti pada Gambar 4.10.



Gambar 4.10 Respon Sistem dengan Menggunakan $K_p = 1,5$ dan $K_i = 4,75$

Untuk nilai $K_p = 1,5$ dan $K_i = 4,75$, sistem mencapai *setpoint* dengan *settling time* 80 detik. Suhu minimum pada saat *steady* = $64,25^{\circ}\text{C}$ dan suhu maksimum saat *steady* = $65,75^{\circ}\text{C}$. Dari respon sistem di atas, tidak terjadi *overshoot*.

Hasil *tuning* parameter kontroler diperoleh nilai $K_p = 1,5$ dan $K_i = 4,75$, dimana didapatkan respon sistem yang tidak terdapat *overshoot*. Dikarenakan respon sistem sudah tidak terdapat *overshoot*, maka tidak perlu melakukan pencarian nilai parameter K_d atau nilai K_d dianggap = 0.

Berdasarkan hasil pengujian dalam Gambar 4.10, diperoleh kinerja sistem antara lain:

1. t_u (waktu tunda) yaitu waktu ketika suhu belum naik karena pemanas masih dalam proses pemanasan. t_u berdasarkan pengujian adalah 42 detik.
2. t_s (*settling time*) yaitu waktu yang diperlukan sistem untuk mencapai nilai akhir ketika *steady*. t_s berdasarkan pengujian adalah 80 detik. *Settling time* didapat ketika suhu telah mencapai $65,25^{\circ}\text{C}$.
3. Perhitungan nilai *error steady state* pada sistem didapatkan nilai sebesar 0,38%

$$\%Ess = \frac{\text{rata" suhusteady} - \text{SETPOINT}}{\text{SETPOINT}} \times 100\%$$

$$\%Ess = \frac{65,25 - 65}{65} \times 100\%$$

$$\%Ess = 0,38\%$$

4.3 Pengujian Keseluruhan Sistem dengan Gangguan

a. Tujuan

Pengujian ini bertujuan untuk mengetahui kinerja sistem dengan mengamati respon sistem terhadap *setpoint* ketika mendapatkan gangguan berupa perubahan suhu yang turun drastis.

b. Peralatan yang Digunakan

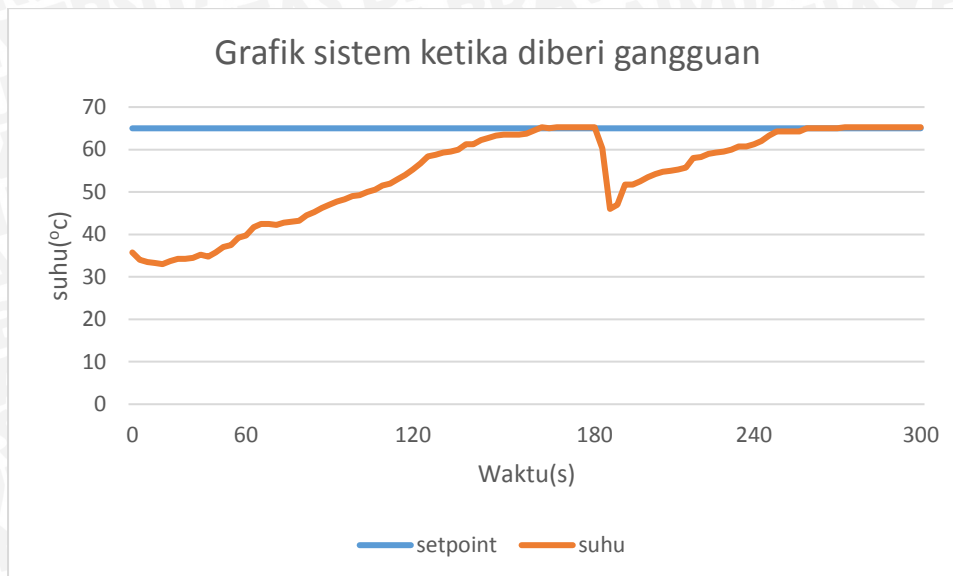
- Kompor gas *portable*
- Motor DC *servo*
- Arduino Mega
- Susu
- Catu daya 12VDC
- *Thermocouple type K*
- Tabung Pasteurisasi
- Laptop
- Program dan *software* Arduino

c. Langkah Pengujian

1. Merakit seperti dalam Gambar 4.8 dan 4.9
2. Menggunakan kontrol PID
3. Meletakkan panci di dalam air dingin kemudian meletakkan kembali ke kompor gas *portable*
4. Melihat hasil keluaran nilai suhu dan menjadikan grafik terhadap waktu.

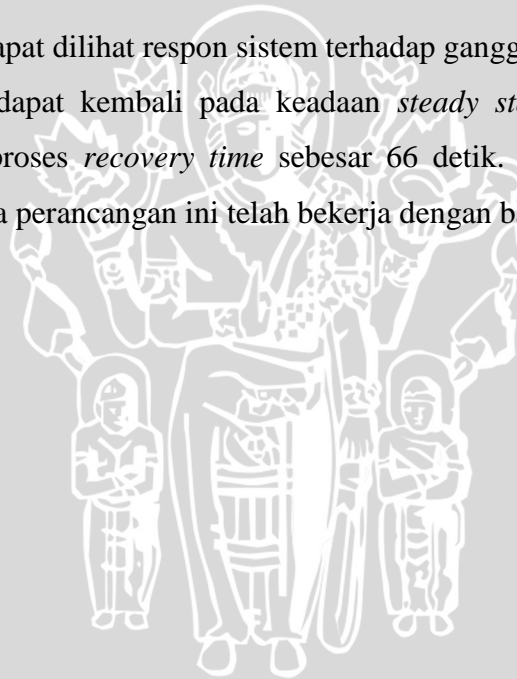
d. Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil respon sistem setelah diberi gangguan yang dapat dilihat dalam Gambar 4.11.



Gambar 4. 11 Respon sistem yang telah diberi gangguan

Dari Gambar 4.11, dapat dilihat respon sistem terhadap gangguan berupa perubahan suhu yang dingin. Sistem dapat kembali pada keadaan *steady state* setelah terjadinya gangguan dan mengalami proses *recovery time* sebesar 66 detik. Dengan begitu, dapat dikatakan respon sistem pada perancangan ini telah bekerja dengan baik.





BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada penelitian sistem pengendalian suhu komposter maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan data respon sistem yang diperoleh dari pengujian dengan metode *hand tuning*, maka parameter kontroler PID dapat ditentukan dengan *gain* $K=0,1666667$, $K_p = 1,5$, $K_i = 4,75$, dan $K_d = 0$.
2. Hasil pengujian ini menunjukkan bahwa Arduino Mega dengan metode kontrol PID menghasilkan respon sesuai dengan yang diharapkan dan mampu diaplikasikan pada sistem pengendalian suhu karena *error* yang didapatkan sebesar 0,38% dengan *settling time* berdasarkan pengujian adalah 80 detik.

5.2 Saran

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada penelitian sistem pengendalian suhu pasteurisasi ada beberapa saran yang dapat dilakukan agar penelitian dapat dilanjutkan dan memberikan hasil yang lebih sempurna sebagai berikut :

1. Pada saat pengujian dilakukan diharapkan menggunakan sensor dengan keakuratan dan ketelitian lebih baik, agar hasil yang didapatkan sensor lebih baik dan *error*-nya sedikit.
2. Peletakkan sensor diharapkan dapat lebih mendekati dasar agar hasil yang didapatkan lebih mendekati hasil yang *real-time* yang sebenarnya, tetapi sensor jangan sampai mengenai dinding sensor ataupun mengenai dasar sensor.



Daftar Pustaka

- Arisandi, R. 2014 Sistem Pengendalian Suhu Pada Proses Pasteurisasi Kuning Telur Menggunakan Kontroler Logika *Fuzzy* Berbasis Mikrokontroler Arduino Mega 2560. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Balai Penelitian Ternak 2000. Pengaruh Suhu Dan Waktu Pasteurisasi Terhadap Mutu Susu Selama Penyimpanan. Bogor: Badan Penelitian Ternak.
- Destriwanta F. 2014 Sistem Pengendalian Suhu Komposter Limbah Padat Tempe dan Biotama-3 Menggunakan Kontroler PID Berbasis Arduino Mega. *Skripsi* . Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Discarara, M. 2014 Sistem Pengendalian Suhu Pada Proses Fermentasi Susu Menggunakan Kontroler Logika *Fuzzy* Berbasis Mikrokontroler Arduino Mega 2560. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Ogata, K. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Terjemahan Edi Leksono, Jakarta : Erlangga.
- Ogata, Katsuhiko. 1994, *Teknik Kontrol Automatik Jilid 1*, Erlangga: Jakarta.
- Purnomo S. 2013 Perancangan Sistem Keamanan Rumah Berbasis SMS Gateway Menggunakan Mikrokontroler Arduino ATmega 2560. *FT UMRAH* :4-5
- Siregar, I. 2009 Teknik Kalibrasi *Thermocouple Type- K* di PT Inalum Kuala Tanjung. *Skripsi*. Tidak dipublikasikan. Medan: Universitas Sumatera Utara.
- Wibowo, W. 1989. Pemantauan bakteri susu kotak pada berbagai umur simpan. *Skripsi* Fakultas Teknologi Pangan Universitas Gadjah Mada, Yogyakarta.
- Yuli , Ratna .2008. Tingkat keamanan susu berlabel pasteurisasi di wilayah Surabaya selama masa penyimpanan pada suhu refrigerator. *Skripsi* Fakultas Teknologi Pangan UPN “Veteran” Jatim

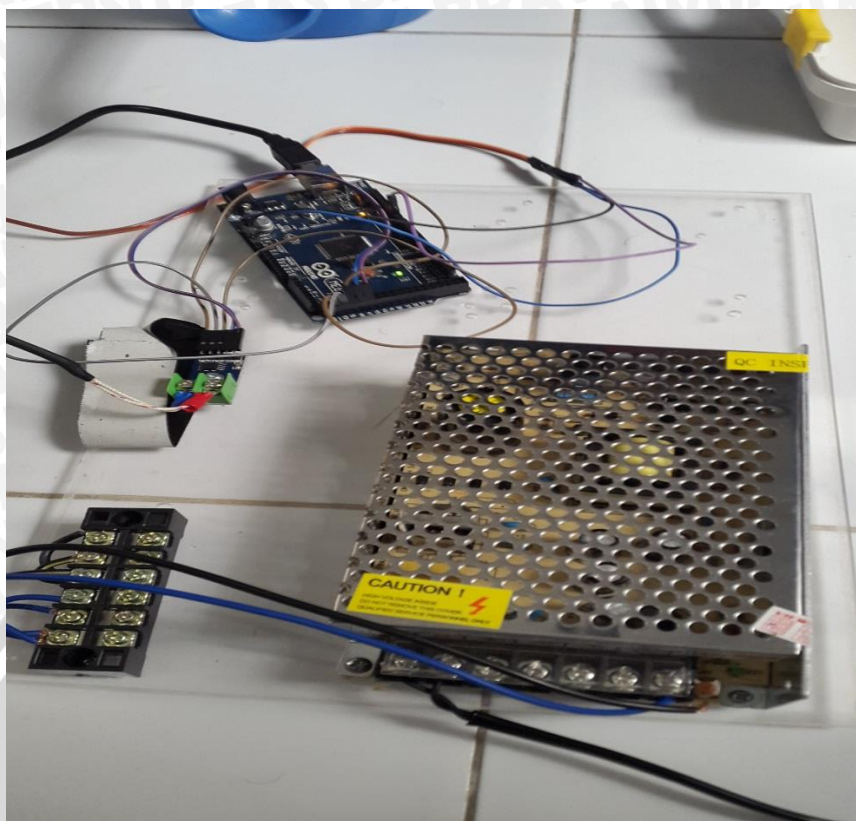


UNIVERSITAS BRAWIJAYA

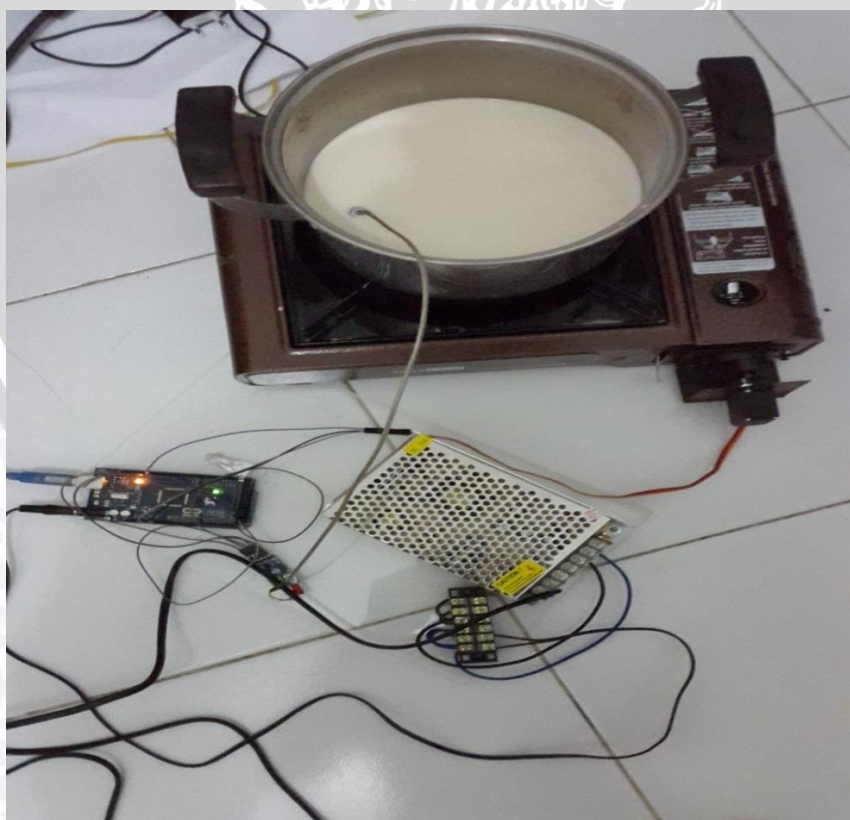


Lampiran I

Foto Alat



Gambar rangkaian elektrik



Gambar alat tampak atas



Gambar alat tampak depan





UNIVERSITAS BRAWIJAYA



Lampiran II

Listing Program

```

#include <PID_v1.h>
#include <max6675.h>
#include <Servo.h>
Servo myservo;
float sudutfix, sudut;
int thermo_gnd_pin = 45;
int thermo_vcc_pin = 47;
int thermo_so_pin = 49;
int thermo_cs_pin = 51;
int thermo_sck_pin = 53;

MAX6675 thermocouple(thermo_sck_pin, thermo_cs_pin, thermo_so_pin);
int in1=5;
int in2=4;
int in3=3;
int in4=2;
int enA=6;

unsigned long time;
double Setpoint, Input, Output; // these are just variables for storing values
// Tuning parameters
float Kp=1.5; //Initial Proportional Gain
float Ki=4.75; //Initial Integral Gain
float Kd=0; //Initial Differential Gain
float suhu;
float K=0.17; //steady state gain
PID myPID(&Input, &Output, &Setpoint,Kp,Ki,Kd, DIRECT);

// Timestamp
unsigned long serialTime; //this will help us know when to talk with processing
const int sampleRate = 1; // Variable that determines how fast our PID loop runs

```



```
const long serialPing = 500; //This determines how often we ping our loop
unsigned long now = 0; //This variable is used to keep track of time
unsigned long lastMessage = 0; //This keeps track of when our loop last spoke to
serial
```

```
void setup() {
  myservo.attach(9);
  myservo.write(180);
  delay(5000);
  Serial.begin(9600);

  pinMode(thermo_vcc_pin, OUTPUT);
  pinMode(thermo_gnd_pin, OUTPUT);
  digitalWrite(thermo_vcc_pin, HIGH);
  digitalWrite(thermo_gnd_pin, LOW);
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  pinMode(enA,OUTPUT);
  pinMode(8,OUTPUT);
  digitalWrite(8, HIGH);
  suhu=thermocouple.readCelsius();

  Input = suhu; // Read in light level
  Setpoint = 65; // Read the SetPoint value from the potentiometer
  myPID.SetMode(AUTOMATIC); // Turn on the PID loop as automatic control
  myPID.SetOutputLimits(0, 1300);
  myPID.SetSampleTime(sampleRate); // Sets the sample rate
  lastMessage = millis(); // timestamp
}

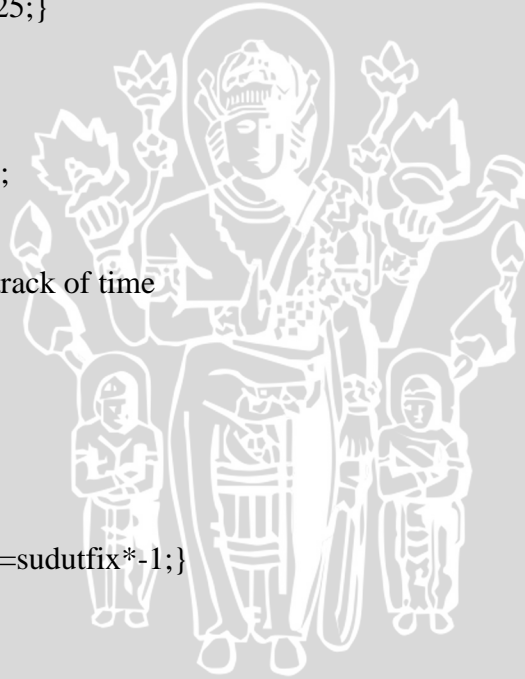
void loop() {
  time=millis();
```

```
suhu=thermocouple.readCelsius();  
delay(500);  
Input = suhu; // Read in light level  
Setpoint = 65; // Read the SetPoint value from the potentiometer  
myPID.Compute(); // Run the PID loop
```

```
digitalWrite(8, HIGH);  
Serial.print(Output);  
Serial.print("\t");  
Serial.println(thermocouple.readCelsius());
```

```
delay(500);  
sudut=Output/K;  
if (sudut>125){sudut=125;}  
if (sudut<0){sudut=0;}  
konversisudut();  
myservo.write(sudutfix);  
delay(1000);  
now = millis(); // Keep track of time  
}
```

```
void konversisudut(){  
sudutfix=sudut-126;  
if (sudutfix<0){sudutfix=sudutfix*-1;}  
}
```



UNIVERSITAS BRAWIJAYA



Lampiran III

Datasheet









