

Sistem Pengaturan Suhu Air Menggunakan Kendali PID berbasis LabVIEW

Water Temperature Control Using PID Control System Based on LabVIEW

Ari Ramadhani, Lery Sakti Ramba

Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Komputer
Universitas Komputer Indonesia Jl. Dipati ukur No 112, Bandung
Email : ariramadhani24@email.unikom.ac.id

Abstrak - Sistem otomasi sudah merambah ke segala bidang tak terkecuali sistem pemanas air. Proses memanaskan air secara tradisional dilakukan dengan memanfaatkan api sebagai media penghasil panas. Dan seiring berkembangnya teknologi pemanasan pun dapat dilakukan dengan bantuan energi listrik yang di konversi menjadi panas. Energi listrik tersebut dialirkan melalui sebuah batang besi yang kontak langsung dengan air sehingga menghasilkan panas yang dapat meningkatkan suhu air. Dalam beberapa kasus, ada beberapa proses yang membutuhkan suhu air yang tepat. Maka dari itu dibutuhkan suatu termometer yang dapat membaca suhu aktual yang dijadikan nilai acuan suhu yang diinginkan dan mengatur aliran listrik agar menghasilkan panas yang linear dengan besar aliran listriknya. Dengan bantuan sebuah mikrokontroler yang dapat menghasilkan keluaran PWM, hal ini dapat dilakukan. Selain itu, sebagai antarmuka dari sistem ini digunakan Labview untuk memantau dan menghasilkan keluaran yang telah di proses kedalam sistem kendali Proporsional, Integral, dan Derivatif (PID) untuk menghasilkan panas yang lebih akurat dan stabil. Berdasarkan hasil pengujian, sistem mampu memberikan respon yang cepat terhadap setiap perubahan yang terjadi, baik itu perubahan *set-point* maupun perubahan suhu air (nilai aktual). Pengujian lain yang dilakukan adalah dengan membandingkan nilai suhu yang dideteksi sensor suhu pada alat ini dengan sebuah *termometer digital external* yang ditempatkan pada sebuah wadah yang sama, dan dari beberapa pengujian tersebut maka nilai suhu yang dideteksi sensor suhu pada alat ini memiliki selisih sebesar ± 0.19 °C dengan termometer digital.

Kata kunci : Pemanas Air, Termometer, Mikrokontroler, LabView, PID.

Abstract - Automatic system have grown widespread across all sector so do water heater. Traditionally, heating water is done by utilizing fire as heat source. As the growing of technology, the heating process could be done by manipulating electrical energy by convert it to heat. Electrical energy is flown to a metal rod that contact directly with the water which increase the water temperature. On some case, appropriate water temperature is needed. Although, a thermometer is needed to read the actual temperature as a feedback value for the system and a system that can control the electricity current flow through the heater that the heat produced is linear to the current flow. With implementing microcontroller as a process node for generating PWM signal, this problem can be solved. Also, Labview is needed as an interface for monitoring and bursting an output which have been processed by Proportional, Integral, and Devivative (PID) controller to producing accurate and stable heat. Based on the results of testing, the system is able to provide a rapid response to any changes that occur, both changes in *set-point* and changes in water temperature (actual value). Another test is done by comparing the temperature value detected by the temperature sensor in this device with an external digital thermometer placed in the same place, and from some of the tests the temperature value detected by the temperature sensor in this device has a difference of ± 0.19 °C with a digital thermometer.

Keyword : Water Heater, Thermometer, Microcontroller, LabView, PID.

I. PENDAHULUAN

A. Latar Belakang

Sistem kendali pada jaman ini mengalami perkembangan pesat dikarenakan kebutuhan industri yang membutuhkan sistem dengan kecepatan dan akurasi yang tinggi. Seiring berjalannya waktu masyarakat umum juga semakin membutuhkan peralatan elektronik penunjang kehidupan sehari-hari yang memiliki kecepatan dan akurasi yang tinggi. Oleh karena itu diperlukan sebuah sistem kendali yang dapat bekerja yang dapat memenuhi tuntutan akan sebuah sistem yang dapat menghadirkan kecepatan dan akurasi yang tinggi. Terdapat beberapa metode sistem pengendali yang ada saat ini. Salah satu metode sistem pengendali yang dapat menghasilkan keluaran yang baik adalah sistem kendali PID. PID adalah sebuah mekanisme kontrol umpan balik yang membutuhkan kendali yang bekerja secara kontinu [1]. Sistem kendali PID merupakan *tool* standar yang digunakan pada otomasi industri. Fleksibilitas dari kontroler PID memungkinkannya untuk dapat digunakan di banyak aplikasi sistem kendali [2]. Terdapat banyak masalah dalam sistem kendali yang dapat ditangani dengan baik oleh sistem kendali PID. Pengontrol PID menghitung nilai galat $e(t)$ sebagai nilai perbandingan dari nilai acuan atau *setpoint* (SP), dan variabel proses (PV), yang akan di koreksi berdasarkan nilai proporsional, integral, dan derivatif (disingkat sebagai PID) sebagaimana nama dari sistem ini. PID ini akan sangat berguna sekali untuk mencapai keluaran yang diinginkan karena dapat mengatasi masalah-masalah perubahan nilai yang fluktuatif dan dengan menciptakan sistem yang responsif terhadap gangguan lainnya. Dengan sistem PID kita dapat mengatur respon seperti apa yang kita ingin gunakan dalam sistem yang akan di rancang. Sistem kendali PID dapat dengan efektif digunakan untuk sistem linier, tetapi biasanya sulit atau tidak dapat digunakan pada sistem tingkat tinggi dan nonlinier [3].

Sistem kendali PID juga akan digunakan pada pemanas air dalam penelitian ini. Sistem pengendalian suhu air merupakan salah satu aplikasi yang paling penting dan paling banyak digunakan pada sistem kendali *non-linear* [4] pada industri yang memproduksi berbagai minuman seperti minuman coklat, susu, dan berbagai produk minuman lainnya [5]. Mengendalikan suhu air dengan sistem kendali PID dilakukan dengan cara menentukan *set-point* suhu air yang diinginkan dan kemudian menjaga

agar suhu air (nilai aktual) tetap stabil dengan *error* yang minimal [5-6]. Keluaran dari PID akan di jumlahkan dengan nilai kontrol yang akan mengatur tegangan AC pada pemanas air. Nilai kontrol ini merupakan suatu nilai dalam bentuk persen (%) siklus kerja. Siklus kerja ini dimanfaatkan dalam pengontrolan dengan menggunakan modulasi lebar pulsa. Modulasi lebar pulsa (PWM) memanfaatkan nilai siklus kerja ini sebagai seberapa banyak sinyal keluaran yang bernilai tinggi dari satu gelombang. Nilai dari perbandingan antara seberapa lama sinyal bernilai tinggi dan seberapa lama sinyal bernilai rendah adalah nilai rata-ratanya. Keluaran PWM tersebut nantinya akan dimanfaatkan untuk mengontrol suplai tegangan AC pada pemanas dengan cara memutus sebagian aliran tegangan AC pada waktu yang tepat [7], sehingga dengan demikian suhu air akan tetap stabil dengan *error* yang minimal pada nilai *set-point* yang telah ditentukan. Untuk mengetahui suhu air (nilai aktual) pada penelitian ini, digunakan sensor DS18B20. DS18B20 adalah sensor suhu digital yang memiliki kemampuan mengukur suhu dari -55 °C sampai +125 °C dengan ketelitian ± 0.5 °C [8].

B. Tinjauan *State of Art*

Sistem kendali PID yang digunakan pada penelitian ini sama dengan sistem kendali PID yang umum digunakan. Namun pada penelitian ini terdapat beberapa software bantuan yang akan digunakan untuk membantu sistem kerja hardware. Pada penelitian ini akan dibuat sebuah aplikasi pada PC menggunakan LABView. Aplikasi ini akan secara realtime berkomunikasi dengan hardware sistem kendali dan berfungsi sebagai *interface* untuk sistem kendali PID [9]. Pada aplikasi inilah user dapat menentukan *set-point* yang diinginkan. Selain itu, seluruh proses kerja dari sistem kendali PID akan ditampilkan pada aplikasi ini, sehingga dengan demikian memudahkan user dalam memantau suhu air (nilai aktual), kestabilan, performa, dan proses kerja dari sistem kendali PID pada penelitian ini.

Dalam penelitian ini sebelum sistem kendali PID mulai bekerja, sistem pertama-tama akan membaca terlebih dahulu suhu air yang hendak dipanaskan. Suhu air inilah yang akan menjadi nilai aktual pada proses kerja sistem kendali PID. Untuk itu maka diperlukan sebuah sensor yang mampu membaca suhu air dengan akurat. Pada penelitian sebelumnya yang terkait dengan penelitian ini, proses pengukuran suhu air menggunakan sebuah sensor LM35 [10]. Sensor

LM35 adalah salah satu sensor suhu yang paling populer paling sering digunakan dalam membaca nilai suhu. Hal ini dikarenakan sensor ini sangat mudah untuk digunakan dan juga memiliki harga yang terjangkau. Namun *output* dari sensor LM35 masih dalam bentuk sinyal analog. Sehingga untuk dapat diproses oleh mikrokontroler perlu proses *Analog to Digital Converter* (ADC) terlebih dahulu. Sedangkan pada penelitian ini sensor yang digunakan untuk mendeteksi suhu air adalah sensor DS18B20. Kelebihan dari sensor DS18B20 adalah *output*-nya sudah berada dalam bentuk sinyal digital. Sehingga sinyal *output*-nya dapat langsung diproses oleh mikrokontroler tanpa memerlukan proses ADC.

Penelitian sebelumnya yang juga meneliti tentang pengendalian suhu air menggunakan sistem kendali PID, digunakan sebuah relay untuk menyalakan atau mematikan komponen pemanas elektrik, relay tersebut akan memutuskan dan menghubungkan arus tegangan tinggi ke komponen pemanas ketika mendapat *trigger* dari mikrokontroler [11]. Penggunaan relay merupakan salah satu opsi yang praktis untuk mengendalikan pemanas air, namun disatu sisi juga memiliki kelemahan, dimana relay tersebut akan terus berkerja untuk menghubungkan dan memutuskan arus tegangan tinggi ke beban. Sistem kerja tersebut akan membebani relay, dan tentu saja dapat membuat relay cepat rusak dikarenakan harus berkerja terus menerus. Oleh karena itu, pada penelitian ini digunakan sistem pengaturan tegangan menggunakan TRIAC yang terintegrasi dengan mikrokontroler dan rangkaian *zero crossing detector* untuk menentukan waktu mulai memberikan *trigger* pada TRIAC. Pemberian *trigger* pada TRIAC pada waktu yang tepat akan meningkatkan efektifitas dan efisiensi daya output dari pengendalian beban listrik AC. Sehingga dengan demikian sistem dapat membatasi seberapa besar daya yang akan diberikan ke pemanas air. Besar kecilnya daya yang diterima oleh pemanas air akan secara langsung menentukan besar kecilnya energi panas yang akan dihasilkan oleh pemanas air. Energi panas yang dihasilkan oleh pemanas kemudian akan di salurkan ke air yang berada disekitarnya.

Dalam proses perancangan sistem kendali PID dapat menggunakan alat bantu ataupun *library* sehingga dapat mempermudah proses perancangan sistem kendali PID, dikarenakan kita hanya perlu menggunakan tool-tool yang telah tersedia. Seperti pada penelitian lainnya yang masih berhubungan dengan penelitian ini, dimana digunakan sebuah toolbox FOMCON

(“*Fractional order Modelling and Control*”) yang merupakan sebuah toolbox matlab yang berorientasi pada *fractional-order calculus* [12]. Sedangkan pada penelitian ini, software yang digunakan untuk merancang algoritma sistem kendali adalah LabVIEW. LabVIEW merupakan sebuah aplikasi pemrograman berbasis visual, sehingga cenderung lebih mudah digunakan dari pada aplikasi lainnya seperti Matlab yang masih berbasis text. LabVIEW memiliki banyak *tool-tool* yang dapat memudahkan dalam merancang berbagai sistem, termasuk sistem kendali PID.

C. Tujuan

Tujuan dari penelitian ini adalah membuat sebuah sistem pemanas air yang akan memanaskan air sesuai dengan keinginan atau kebutuhan pengguna. Pengguna hanya perlu memasukkan nilai acuan (*set-point*) suhu air yang diinginkan. Dan setelah itu sistem akan berkerja untuk mencapai nilai *set-point* yang di-*input* oleh *user*. Dan ketika suhu air yang diinginkan (*set-point*) oleh *user* telah tercapai, maka sistem akan menjaga kestabilan suhu air, sehingga suhu air dapat tetap berada pada nilai suhu yang diinginkan oleh *user*.

II. METODOLOGI

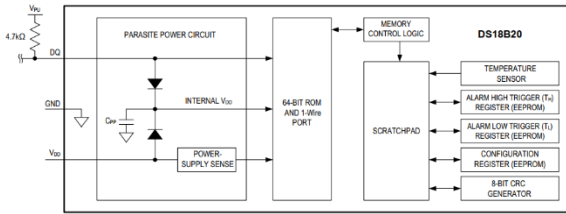
Penelitian ini merupakan jenis penelitian eksperimen yang dilakukan dengan percobaan pada serangkaian sistem baik *hardware* maupun *software* yang berkerja pada sistem kendali PID. Percobaan juga dilakukan untuk mengetahui seberapa baik *hardware* dan *software* saling terintegrasi satu sama lain pada penelitian ini.

A. Perancangan pendeteksi suhu air

Pada penelitian ini, sistem pertama tama akan mendeteksi besar suhu air. Sensor yang digunakan untuk mendeteksi suhu air adalah DS18B20. Sensor yang digunakan sudah terisolasi dan dimasukkan kedalam suatu wadah khusus anti air dengan campuran bahan alumunium. Berikut adalah bentuk dari sensor DS18B20 yang ditunjukkan pada **Gambar 1** dan blok diagram dari sensor DS18B20 pada **Gambar 2**.



Gambar 1. Sensor DS18B20



Gambar 2. Blok diagram sensor DS18B20

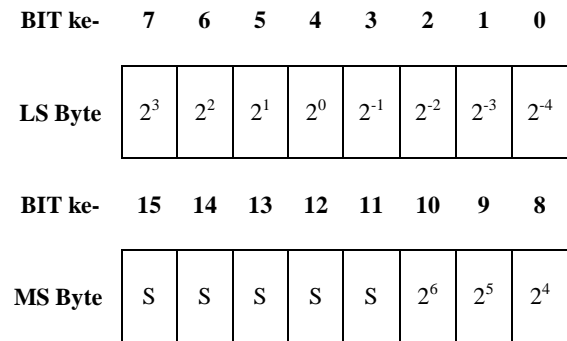
Dapat dilihat pada Gambar 2, ROM 64-bit dapat menyimpan kode serial unik dari perangkat. Memory acak yang menyimpan temperatur terdiri dari register 2-byte yang menyimpan nilai digital dari sensor temperatur. Memori acak ini dapat menyediakan akses ke 1 upper-byte dan register lower alarm trigger. Konfigurasi dari register memungkinkan pengguna untuk mengatur resolusi dari suhu ke konversi digital dari 9, 10, 11, atau 12 bits. Nilai T_H dan T_L dan konfigurasi register terdapat pada memori non-volatile (EEPROM), Setiap sensor DS18B20 memiliki serial kode 64bit yang unik, yang memungkinkan untuk memanfaatkan sensor ini pada bus 1-wire yang sama. Maka dari itu sensor ini lebih mudah dibaca dengan hanya menggunakan satu mikroprosesor untuk membaca banyak masukan. Aplikasi ini dapat bermanfaat untuk mengukur lingkungan, sistem temperatur dalam bangunan, peralatan, atau mesin dan untuk proses monitoring dari sistem kontrol.

Kelebihan dari sensor ini adalah outputnya sudah dalam bentuk sinyal digital digital, sehingga tidak lagi memerlukan proses Analog to Digital Converter (ADC) [13]. Sensor suhu DS18B20 yang digunakan pada penelitian ini dapat menghasilkan nilai pengukuran sebanyak 9-bit sampai 12-bit sesuai kebutuhan dengan suhu pembacaan dari -55°C sampai 125°C dan akurasi $\pm 0.5^{\circ}\text{C}$ dari -10°C sampai $+85^{\circ}\text{C}$. Semakin sedikit bit yang digunakan untuk tranfer data, maka semakin cepat pengkonversian nilai suhu ke bentuk digital. Berikut kecepatan konversi pada kondisi setiap bit ditunjukkan pada Tabel 1.

Tabel 1. Waktu terhadap kecepatan konversi

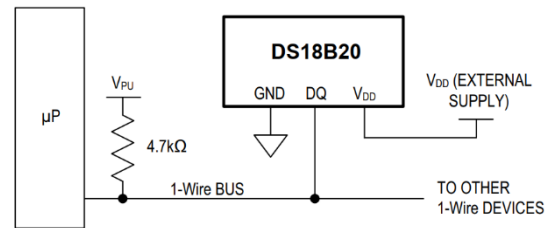
Resolusi bit	Max	Unit
9-bit	93.75	ms
10-bit	187.5	
11-bit	375	
12-bit	750	

Data yang dikirimkan disimpan dalam sebuah register 16-bit dengan 5 buah bit sign. Bit sign mengindikasikan nilai yang terbaca positif apabila $S=0$ atau negatif apabila $S=1$. Apabila menggunakan resolusi 12-bit, semua baris akan berisi data yang valid. Apabila menggunakan 11-bit, maka bit 0 tidak terdefinisi. Apabila menggunakan resolusi 10-bit maka bit 1 dan bit 0 tidak terdefinisi. Apabila menggunakan resolusi 9-bit, maka bit 2, bit 1, dan bit 0 tidak terdefinisi. Pada Gambar 3 terdapat definisi dari setiap bit.



Gambar 3. Definisi setiap bit pada komunikasi DS18B20.

Berikut rangkaian dasar untuk menggunakan sensor DS18B20 pada Gambar 4.



Gambar 4. Rangkaian dasar DS18B20

Untuk membaca nilai masukan dari sensor ini, program mikrokontroler membutuhkan sebuah library bernama one-wire. Sistem komunikasi menggunakan menggunakan one-wire dapat menyediakan transmisi data rendah (16.3kbps), sinyal, atau daya melalui satu buah konduktor penghantar. Untuk wiring dari sistem pembacaan suhu ini dapat dilihat pada Gambar 5.



Gambar 5. Wiring dari pembacaan suhu

Pada sistem pembacaan suhu digunakan Arduino Mega sebagai mikrokontroler. Untuk sistem pembacaan suhu, data dari sensor dimasukkan ke pin 4 dari Arduino Mega. Berikut program yang tersimpan dalam Arduino Mega untuk pembacaan suhu pada **Gambar 6**.

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into pin 4 on the Arduino
#define ONE_WIRE_BUS 4

// Setup a oneWire instance to communicate with any OneWire devices
// (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup(void)
{
  // start serial port
  Serial.begin(9600);
  // Serial.println("Dallas Temperature IC Control Library Demo");

  // Start up the library
  sensors.begin();
}

void loop(void)
{
  // call sensors.requestTemperatures() to issue a global temperature
  // request to all devices on the bus
  sensors.requestTemperatures(); // Send the command to get temperatures

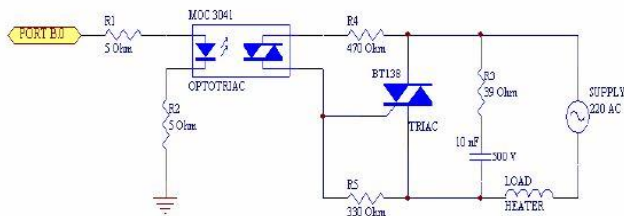
  Serial.println(sensors.getTempCByIndex(0)); // Why "byIndex"?
  // You can have more than one IC on the same bus.
  // 0 refers to the first IC on the wire
  delay(200);
}
```

Gambar 6. Kode program arduino untuk pembacaan suhu

Pada program pembacaan suhu dibutuhkan library OneWire dan Dallas Temperature. Onewire adalah sebuah library untuk pembacaan data digital hanya menggunakan satu kabel saja, sedangkan Dallas temperature adalah library untuk menerjemahkan masukan digital dari DS18B20 menjadi besaran Celcius.

B. Sistem pengaturan tegangan AC

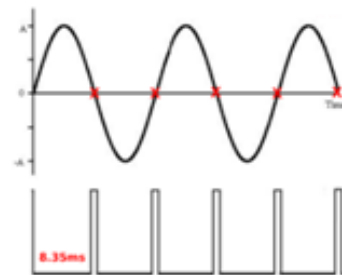
Sebagai pengatur tegangan AC, digunakan rangkaian *Solid State Relay* (SSR) yang pada prinsipnya adalah penggabungan antara rangkaian *optocoupler* yang menggunakan *Zero Crossing Circuit* di MOC 3041 dan Triac, seperti pada rangkaian **Gambar 7** berikut ini:



Gambar 7. Rangkaian Pengatur Tegangan

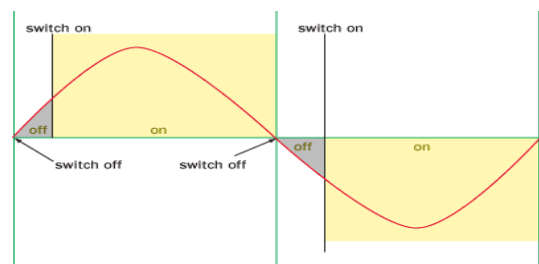
Zero Crossing detector pada penelitian ini memiliki peranan yang sangat vital. *Zero Crossing detector* digunakan untuk mendeteksi

gelombang sinus AC saat melewati titik tegangan nol [5]. Titik tegangan nol yang dimaksud pada penelitian ini adalah peralihan dari positif menuju negatif dan peralihan dari negatif menuju positif. *Zero crossing detector* diperlukan untuk memotong sinyal AC sehingga dapat menghasilkan sinyal keluaran AC yang variatif. Berikut adalah ilustrasi *zero crossing detector* pada **Gambar 8**.



Zero crossing pulse at 0, 180, and 360 degrees.
 $F = 60\text{Hz}; P = 1/60 = 16.7\text{mSec}; \text{Half-Cycle} = 8.35\text{mSec}$
Gambar 8. Ilustrasi zero crossing detector

Pemotongan sinyal ini hampir sama seperti sinyal PWM, dimana tegangan rata-rata didapatkan dengan memotong sinyal menjadi sinyal diskrit yang terdapat nilai On dan Off. Semakin lama sinyal tersebut dalam kondisi On, maka semakin maksimal tegangan yang dihasilkan. Perbandingan antara kondisi On dan Off ini disebut dengan *Duty Cycle*. PWM digunakan pada sinyal DC yang tidak memiliki frekuensi, maka sinyal ini dapat di potong di waktu kapan saja. Namun pada sinyal AC waktu pemotongan sinyal tersebut harus presisi agar dapat menghasilkan nilai yang sesuai. Berikut gambar sinyal ac yang terpotong pada **Gambar 9**.



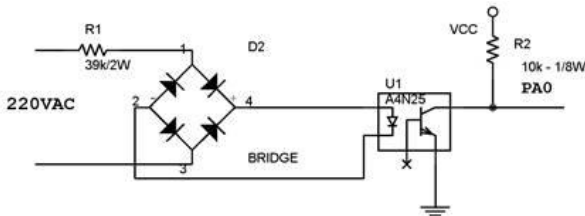
Gambar 9. Gambar sinyal AC yang terpotong berdasarkan *zero crossing*

C. Water Heater/Pemanas Air

Untuk menaikkan suhu air, dibutuhkan sebuah pemanas dengan tegangan AC. Pemanas terdiri dari sebuah elemen pemanas yang dialiri oleh listrik tegangan AC yang akan menghasilkan panas. Dengan mengatur daya yang diberikan pada elemen pemanas maka panas yang dikeluarkan dari elemen pun akan berbeda beda. Untuk dapat mengatur daya, maka dibutuhkan

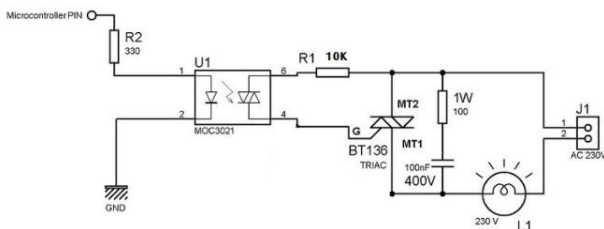
sebuah rangkaian yang dapat memotong tegangan AC untuk merubah daya yang diberikan pada elemen, rangkaian tersebut adalah AC dimmer. Dengan AC dimmer, tegangan yang dihasilkan dapat variatif dan akan memberikan keluaran panas yang linear dengan tegangan masukannya [14].

Untuk memotong tegangan AC harus dilakukan dengan presisi. Pemotongan sinyal harus dimulai dari titik *zero crossing*, yaitu suatu kondisi peralihan dimana nilai tegangan sama dengan 0. Maka dari itu dibutuhkan sebuah rangkaian *zero crossing detector* untuk mendeteksi transisi tegangan AC agar dapat memotong dengan sesuai dan kondisi tersebut dikirimkan ke Arduino [15]. Berikut gambar rangkaian *Zero Crossing* pada **Gambar 10**.



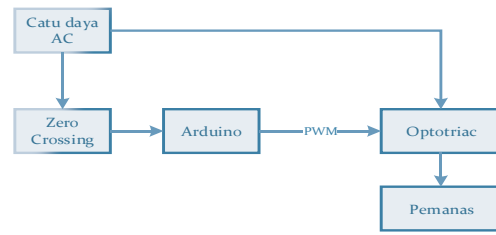
Gambar 10. Rangkaian Zero Crossing

Keluaran dari Arduino merupakan gelombang PWM yang dapat menghasilkan tegangan dari 0 sampai 5V, maka dari itu dibutuhkan sebuah rangkaian yang dapat menterjemahkan tegangan keluaran dari arduino tersebut menjadi sebuah gelombang yang memotong sinyal AC. Arduino terhubung dengan kontrol tegangan AC melalui sebuah komponen Optocoupler Triac atau Optotriac berbentuk sebuah IC dengan 6 kaki. Optotriac berfungsi untuk menghubungkan tegangan DC ke sebuah pin Gate pada Triac internal didalam IC. Berikut rangkaian dari pengatur tegangan ke pemanas pada **Gambar 11**.



Gambar 11. Rangkaian AC Dimmer.

Sederhananya, sistem pengaturan suhu ini dapat digambarkan kedalam blok diagram pada **Gambar 12** berikut.



Gambar 12. Blok diagram pemanas.

Pada perancangan AC dimmer, digunakan sebuah Arduino Uno dengan mikroprosesor ATmega328. Arduino Uno tersebut sudah terinstall sebuah program seperti pada **Gambar 13** berikut.

```
#include <Time.h>
#include <RBDDimmer.h>

//define USE_SERIAL SerialUSB //Serial for boards with USB serial port
#define USE_SERIAL Serial
#define outputPin 3
#define zerocross 2 // for boards with CHANGEABLE input pins

//dimmerLamp dimmer(outputPin, zerocross);
//initialase port for dimmer for ESP8266, ESP32, Arduino due boards
dimmerLamp dimmer(outputPin);
//initialase port for dimmer for MEGA, Leonardo, UNO,
//Arduino M0, Arduino Zero

int outVal = 0;

void setup() {
  USE_SERIAL.begin(9600);
  dimmer.begin(NORMAL_MODE, ON);
  //dimmer initialisation: name.begin(MODE, STATE)
}

void loop()
{
  if (Serial.available()){
    outVal = Serial.readString().toInt();
    USE_SERIAL.println(outVal);
    dimmer.setPower(outVal); // name.setPower(0%-100%)
  }
}
```

Gambar 13. kode program Arduino untuk AC dimmer

Pada program diatas terdapat library RBDDimmer, yaitu sebuah library khusus untuk mengontrol modul AC dimmer. Seperti yang dibahas sebelumnya, untuk mengatur tegangan AC dibutuhkan suatu cara untuk memotong sinyal AC pada waktu yang tepat yaitu dengan *zero crossing detector* dan PWM yang frekuensi nya sama dengan frekuensi jala-jala listrik. Pada modul AC dimmer ini sudah terdapat rangkaian *Zero Crossing* dan sebuah IC Optotriac untuk menghubungkan tegangan PWM dari Arduino ke tegangan AC dari jala-jala listrik.

Untuk instalasi modul AC dimmer ini terdapat beberapa aturan pin Arduino yang dapat digunakan untuk mengatur tegangan AC menggunakan Arduino yang dapat dilihat pada **Gambar 14** berikut ini sedangkan gambar dari Arduino UNO, modul AC dimmer dan definisi pin dapat dilihat pada **Gambar 15**.

Untuk arduino UNO, pin Zero Crossing dihubungkan pada pin 2. Digunakan Pin 2 karena Pin ini adalah sebuah pin khusus untuk

mengaktifkan fungsi interrupt, maka dari itu dapat dilihat pada definisi pin Arduino diatas kebanyakan pin zero crossing nya tidak dapat dirubah (*Not Changeable*). Sedangkan Pin digitalnya dapat terhubung ke pin D0-D1 dan D3-D20.

```

*
* ----- OUTPUT & INPUT Pin table -----
*
* | Board | INPUT Pin | OUTPUT Pin |
* |-----|-----|-----|
* | Leonardo | D7 (NOT CHANGABLE) | D0-D6, D8-D13 |
* |-----|-----|-----|
* | Mega | D2 (NOT CHANGABLE) | D0-D1, D3-D70 |
* |-----|-----|-----|
* | Uno | D2 (NOT CHANGABLE) | D0-D1, D3-D20 |
* |-----|-----|-----|
* | ESP8266 | D1 (IO5), D2 (IO4), D5 (IO14), D6 (IO12), D7 (IO13) | D0 (IO16), D1 (IO5), D2 (IO4), D5 (IO14), D6 (IO12), D7 (IO13), D8 (IO15) |
* |-----|-----|-----|
* | ESP32 | 4 (GPIO36), 5 (GPIO39), 8 (GPIO32), 10 (GPIO25), 11 (GPIO26), 12 (GPIO27), 13 (GPIO14), 14 (GPIO25), 16 (GPIO26), 23 (GPIO15), 24 (GPIO2), 25 (GPIO13), 26 (GPIO4), 27 (GPIO16), 28 (GPIO17), 29 (GPIO5), 30 (GPIO18), 31 (GPIO19), 33 (GPIO21), 34 (GPIO3), 35 (GPIO1), 36 (GPIO22), 37 (GPIO23) | 6 (GPIO34), 7 (GPIO35), 9 (GPIO33), 12 (GPIO27), 13 (GPIO14), 14 (GPIO25), 11 (GPIO26), 16 (GPIO26), 14 (GPIO12), 16 (GPIO13), 12 (GPIO12), 13 (GPIO14), 23 (GPIO15), 24 (GPIO2), 25 (GPIO13), 26 (GPIO4), 21 (GPIO7), 23 (GPIO15), 27 (GPIO16), 28 (GPIO17), 24 (GPIO2), 25 (GPIO0), 29 (GPIO5), 30 (GPIO18), 26 (GPIO4), 27 (GPIO16), 31 (GPIO19), 33 (GPIO21), 28 (GPIO17), 29 (GPIO5), 34 (GPIO3), 35 (GPIO1), 30 (GPIO18), 31 (GPIO19), 36 (GPIO22), 37 (GPIO23), 33 (GPIO21), 35 (GPIO1), 36 (GPIO22), 37 (GPIO23) |
* |-----|-----|-----|
* | Arduino M0 | D7 (NOT CHANGABLE) | D0-D6, D8-D13 |
* | Arduino Zero | | |
* |-----|-----|-----|
* | Arduino Due | D0-D53 | D0-D53 |
* |-----|-----|-----|
* | STM32 | PA0-PA15, PB0-PB15 | PA0-PA15, PB0-PB15 |
* | Black Pill | PC13-PC15 | PC13-PC15 |
* | BluePill | | |
* | Etc... | | |
*
    
```

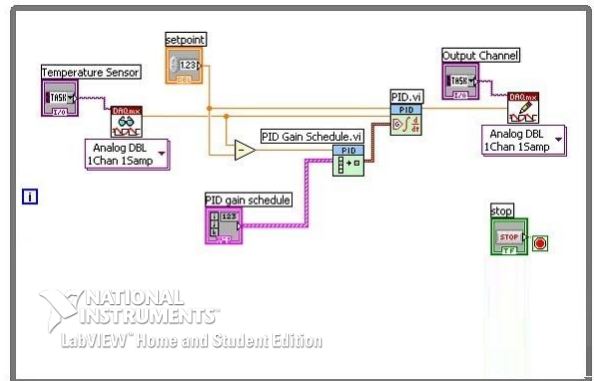
Gambar 14 Definisi pin yang dapat digunakan untuk modul AC dimmer.



Gambar 15. Wiring dari Arduino ke Modul AC Dimmer

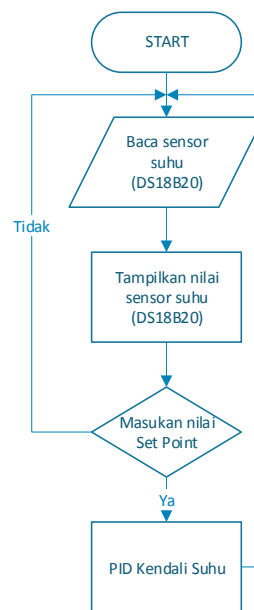
D. Perancangan Software

Pada penelitian ini, akan digunakan algoritma/program PID berbasis Labview. Dimana akan terdapat *Graphic User Interface* yang akan digunakan *user/pengguna* untuk memasukkan nilai suhu yang diinginkan (*set-point*). Nilai *set-point* ini nantinya akan dibandingkan dengan nilai suhu yang terbaca oleh sensor (nilai aktual) untuk mengetahui nilai *error* antara *set-point* dan nilai aktual. Sistem kendali PID adalah sebuah tool standar yang digunakan pada otomasi industri. Sistem kendali PID menghitung nilai galat $e(t)$ sebagai nilai perbandingan dari nilai acuan atau *set-point* (SP) dan variabel proses (PV). Sistem PID ini dapat di aplikasikan kedalam aplikasi Labview. Berikut program PID pada Labview pada Gambar 16.



Gambar 16. Program PID Labview

Sederhananya program PID di labview dapat di jabarkan seperti yang terdapat pada *flowchart* Gambar 17 berikut ini.



Gambar 17. Flowchart Sistem PID

III. HASIL DAN PEMBAHASAN

A. Pengujian output Sensor Suhu DS18B20

Seperti yang diketahui, sensor suhu DS18B20 memiliki keluaran digital dengan *One-Wire* dan tidak dapat di baca oleh ADC, karena apabila datanya dibaca dengan ADC biasa masukan yang diterima hanya akan berbentuk nilai 1 atau 0(on/off). Maka dari itu dibutuhkan sebuah *driver One-Wire Utilities* agar dapat membaca masukan dari sensor. Driver ini sudah tersedia di situs resmi *National Instrument*. Pada **Tabel 2** dapat dilihat hubungan antara temperatur dengan nilai bit.

Tabel 2. Hubungan antara temperature dan data keluaran

Temperature	Digital Output (12 Bit Binary)	Digital Output (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C	0000 0101 0101 0000	0550h
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FF6Fh
-55°C	1111 1100 1001 0000	FC90h

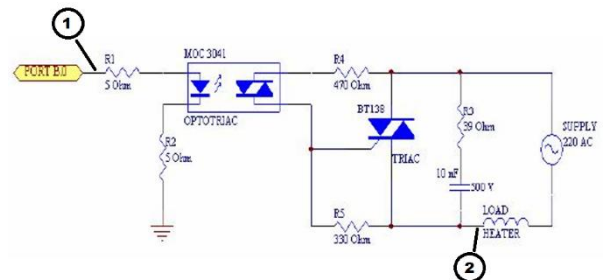
Dapat dilihat pada *output* digital, terdapat maksimal 12-bit. 4-bit pertama merupakan sign bit, yaitu sebagai tanda, apabila bernilai satu, maka temperaturnya negatif. 8 bit berikutnya adalah nilai temperatur dalam bilangan bulat. Walaupun nilai maksimal dari 8-bit adalah 255, komponen pengukur suhu DS18B20 ini tidak dapat mengukur lebih dari 150°C atau dalam biner 1001 0110. 4-bit terakhir adalah yang menyatakan nilai dalam desimal. Ketelitian dari sensor ini sampai 0.0625°C.

B. Pengukuran pengontrol tegangan AC (Rangkaian AC Dimmer)

Pengukuran rangkaian AC Dimmer pada tahap ini dilakukan supaya mengetahui perubahan

tegangan AC ke beban yang dipengaruhi oleh tegangan DC yang melewati komponen optotriac pada rangkaian AC Dimmer. Pengukuran akan dilakukan pada dua titik, seperti pada **Gambar 18**.

Pada titik 1 dari **Gambar 18**, pengukuran dilakukan untuk mengetahui *output* tegangan DC dari mikrokontroler, sedangkan pada titik 2 dari **Gambar 18**, pengukuran dilakukan untuk mengetahui besar *output* tegangan AC yang akan disalurkan ke beban. Berikut pada **Tabel 3** dapat dilihat pengaruh tegangan DC terhadap *output* tegangan AC pada rangkaian AC dimmer.



Gambar 18. Pengukurang Rangkaian AC Dimmer

Tabel 3. Perubahan tegangan AC terhadap tegangan DC

Tegangan DC (mV)	Tegangan AC (V)
2102	220
2050	221
2000	220
1948	219
1902	218
1850	216
1805	210
1770	207
1710	201
1660	193
1608	185
1570	179
1520	172
1472	166
1422	170
1378	152
1330	149
1285	145
1230	152
1284	147
1232	143
1188	140
1140	133
1084	130
1045	128
995	125
950	123
825	122

Tegangan DC (mV)	Tegangan AC (V)
789	121
750	120
705	119
655	114
604	107
556	101
505	91
455	82
415	72
360	60
310	50
365	38
218	30
160	16
120	14
60	7
14	0.8
2	0.330
0	0.102

Dari data yang dapat diamati pada **Tabel 3** diatas, maka dapat disimpulkan bahwa nilai tegangan DC yang diterima oleh optocoupler dari mikrokontroler, akan berpengaruh secara langsung terhadap *output* tegangan AC dari rangkaian AC Dimmer. Semakin besar tegangan DC yang diterima optocoupler, maka output tegangan AC rangkaian AC Dimmer akan semakin besar, sebaliknya semakin kecil tegangan DC yang diterima optocoupler, maka output tegangan AC rangkaian AC Dimmer akan semakin kecil pula.

C. Pengukuran dan pengujian Alat Secara Keseluruhan

Pengujian alat secara keseluruhan dilakukan dengan menempat sensor DS18B20 dan *thermometer* dalam air yang hendak dipanaskan oleh alat ini. Nilai suhu yang dideteksi oleh termometer kemudian akan dibandingkan dengan nilai suhu yang dideteksi sensor DS18B20 yang tampil pada aplikasi LABView. Berikut ini pada **Tabel 4** merupakan hasil pengujian yang telah dilakukan.

$$\begin{aligned} \text{Error Suhu} &= \text{Jumlah selisih} / \text{jumlah percobaan} \\ \text{Error Suhu} &= 5.6/29 \\ &= 0.19 \text{ } ^\circ\text{C} \end{aligned}$$

Tabel 4. Pegujian dan pengukuran alat secara keseluruhan

Tampilan Suhu pada LABView ($^{\circ}\text{C}$)	Tampilan Suhu Pada Thermometer ($^{\circ}\text{C}$)	Selisih ($^{\circ}\text{C}$)
30	30.4	0.4
33	33.3	0.3
35	35.2	0.2
38	38.1	0.1
40	40.5	0.5
42	42	0
43	42.8	0.2
45	45	0
48	48.2	0.2
50	50.3	0.3
53	53.1	0.1
55	55.4	0.4
57	57	0
59	59.2	0.2
60	59.8	0.2
63	62.9	0.1
65	65.5	0.5
67	67.2	0.2
68	68	0
70	70	0
72	72.1	0.1
75	75.2	0.2
78	78.2	0.2
80	80.2	0.2
82	82	0
85	85.4	0.4
87	87.3	0.3
89	88.8	0.2
90	90.1	0.1

Nilai *error* suhu yang berada pada nilai ± 0.19 $^{\circ}\text{C}$ bisa terjadi karena pada penelitian ini kami menggunakan sensor suhu DS18B20, dimana seperti yang telah dijelaskan sebelumnya bahwa sensor suhu digital yang memiliki kemampuan mengukur suhu dari -55 $^{\circ}\text{C}$ sampai $+125$ $^{\circ}\text{C}$ dengan ketelitian ± 0.5 $^{\circ}\text{C}$ [8]. Sehingga nilai *error* ± 0.26 $^{\circ}\text{C}$ masih berada pada nilai toleransi.

Untuk pemberian parameter dari nilai K_p , K_d , dan K_i pada penelitian ini dilakukan dengan cara manual. Percobaan pada tahap ini bertujuan untuk memperoleh nilai yang paling tepat untuk parameter K_p , K_d , dan K_i sehingga dapat diperoleh sebuah sistem kendali dengan respon sistem yang optimal. Berikut adalah data yang diperoleh dari percobaan nilai konstanta kendali.

- Untuk nilai $K_p=60$, $K_i=0$, dan $K_d=12$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - Rise Time* (s) = 4:32

- *Overshoot (%)* = 0.5
 - *Settling Time (s)* = 4:36
 - *Steady State Error (%)* = -4.2
2. Untuk nilai $K_p=60$, $K_i=0$, dan $K_d=8$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 4:25
 - *Overshoot (%)* = 0.7
 - *Settling Time (s)* = 4:28
 - *Steady State Error (%)* = -4.5
 3. Untuk nilai $K_p=60$, $K_i=0$, dan $K_d=4$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 4:15
 - *Overshoot (%)* = 1.1
 - *Settling Time (s)* = 4.18
 - *Steady State Error (%)* = -4.8
 4. Untuk nilai $K_p=100$, $K_i=0$, dan $K_d=12$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 4:08
 - *Overshoot (%)* = 1.3
 - *Settling Time (s)* = 4:10
 - *Steady State Error (%)* = 1.18
 5. Untuk nilai $K_p=100$, $K_i=0$, dan $K_d=8$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 4:02
 - *Overshoot (%)* = 1.5
 - *Settling Time (s)* = 4:05
 - *Steady State Error (%)* = 0.5
 6. Untuk nilai $K_p=100$, $K_i=0$, dan $K_d=4$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 3:56
 - *Overshoot (%)* = 1.8
 - *Settling Time (s)* = 4:00
 - *Steady State Error (%)* = -0.3
 7. Untuk nilai $K_p=60$, $K_i=0.1$, dan $K_d=0$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 5:50
 - *Overshoot (%)* = 0.1
 - *Settling Time (s)* = 5:58
 - *Steady State Error (%)* = 0
 8. Untuk nilai $K_p=60$, $K_i=0.2$, dan $K_d=0$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 5:45
 - *Overshoot (%)* = 0.2
 - *Settling Time (s)* = 5:53
 - *Steady State Error (%)* = 0
 9. Untuk nilai $K_p=60$, $K_i=0.3$, dan $K_d=0$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 5:30
 - *Overshoot (%)* = 0.3
 - *Settling Time (s)* = 5:40
 - *Steady State Error (%)* = 0
 10. Untuk nilai $K_p=100$, $K_i=0.1$, dan $K_d=0$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 5:15
 - *Overshoot (%)* = 0.5
 - *Settling Time (s)* = 5:23
 - *Steady State Error (%)* = 0
 11. Untuk nilai $K_p=100$, $K_i=0.2$, dan $K_d=0$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 5:08
 - *Overshoot (%)* = 0.7
 - *Settling Time (s)* = 5:20
 - *Steady State Error (%)* = 0
 12. Untuk nilai $K_p=100$, $K_i=0.3$, dan $K_d=0$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 5:02
 - *Overshoot (%)* = 0.9
 - *Settling Time (s)* = 5:15
 - *Steady State Error (%)* = 0
 13. Untuk nilai $K_p=60$, $K_i=0.1$, dan $K_d=4$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 3.05
 - *Overshoot (%)* = 0.8
 - *Settling Time (s)* = 3.10
 - *Steady State Error (%)* = -0.2
 14. Untuk nilai $K_p=80$, $K_i=0.1$, dan $K_d=4$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
 - *Rise Time (s)* = 2:58
 - *Overshoot (%)* = 1.1
 - *Settling Time (s)* = 3:10
 - *Steady State Error (%)* = 0.2

15. Untuk nilai $K_p=100$, $K_i=0.1$, dan $K_d=4$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
- *Rise Time* (s) =2:50
 - *Overshoot* (%) =1.5
 - *Settling Time* (s) =3.02
 - *Steady State Error* (%) =0.5
16. Untuk nilai $K_p=60$, $K_i=0.1$, dan $K_d=8$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
- *Rise Time* (s) =2:30
 - *Overshoot* (%) =2.5
 - *Settling Time* (s) =2:35
 - *Steady State Error* (%) =0.8
17. Untuk nilai $K_p=80$, $K_i=0.1$, dan $K_d=8$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
- *Rise Time* (s) =2:15
 - *Overshoot* (%) =2.8
 - *Settling Time* (s) =2.20
 - *Steady State Error* (%) =1
18. Untuk nilai $K_p=100$, $K_i=0.1$, dan $K_d=8$ dengan *set-point* 60°C diperoleh nilai sebagai berikut.
- *Rise Time* (s) =2.05
 - *Overshoot* (%) =3.8
 - *Settling Time* (s) =2.11
 - *Steady State Error* (%) =1.3

Dari data hasil pengukuran diatas dapat diamati bahwa perubahan konstanta K_p , K_i , dan K_d akan memberikan respon sistem yang berbeda-beda. Pemberian nilai K_p akan memperbaiki *Rise Time* dari sistem kendali PID, semakin tinggi konstanta nilai K_p , maka *Rise Time* dan *Settling Time* juga menjadi semakin kecil, namun disatu sisi meningkatkan nilai *overshoot* dari sistem kendali. Sebaliknya ketika nilai K_p dikurangi, maka nilai *Rise Time* dan *Settling Time* akan semakin besar, namun dengan *overshoot* yang semakin kecil.

Untuk nilai K_d , digunakan untuk memperbaiki nilai *overshoot*. Ketika sistem diberi konstanta K_d 4, maka mengakibatkan rise time semakin cepat, *overshoot* yang semakin besar, *Settling Time* semakin lama, dan dengan persentase *Steady state error* yang semakin besar. Namun ketika diberi nilai K_d diatas 4, dalam hal ini adalah 8 dan 12, maka rise time akan semakin lama, dengan nilai *overshoot* yang semakin kecil, settling time

semakin cepat, dan dengan persentase *steady state error* yang semakin besar.

Pemberian nilai K_i pada sistem kendali PID pada penelitian ini akan mempengaruhi dari segi *steady state error*. Semakin besar nilai K_i yang diberikan maka nilai *steady state error* akan semakin kecil, dan pada percobaan memberikan nilai K_p senilai 100, nilai K_i yang berubah ubah, dan nilai K_d di set ke angka 0 yang berarti sistem di set menjadi kendali PI. Maka didapatkan bahwa *steady state error*-nya dieliminasi atau bernilai 0. Namun akan meningkatkan settling time dari sistem kendali.

Dari pengamatan yang dilakukan, nilai yang paling optimal untuk sistem pada project ini adalah $K_p=80$, $K_i=0.1$, dan $K_d=8$. Nilai-nilai konstanta tersebut menjadikan sistem menjadi seimbang di banyak aspek sekaligus.

Pada penelitian Jun-Hong Yang (2010) sistem PID diterapkan pada sebuah sistem industri, agrikultur, dan pemanas rumahan dengan kendali on/off menggunakan relay. Pada percobaan nya untuk menaikkan suhu dari 40 sampai ke 60 derajat celsius, *settling time* yang dibutuhkan adalah 450 detik dengan eror statis 0.2 derajat celsius dengan overshoot maksimal 0.5 derajat celsius. Dibandingkan dengan penelitian tersebut, nilai *settling time* yang dihasilkan pada penelitian ini lebih baik. Waktu *settling time* lebih cepat pada penelitian ini dengan menggunakan nilai $K_p = 80$, $K_i = 0.1$, $K_d = 8$. Namun untuk tingkat akurasi hasil akhir dari pengaturan suhu, penelitian ini belum mencapai nilai yang akurat dengan nilai overshoot paling besar 2.5% atau $60 \times 2.5\% = 1.5$ derajat celsius. Nilai *maximum overshoot* ini jauh lebih besar dibandingkan dengan penelitian Jun-Hong Yang dengan nilai *maximum overshoot* 0.5 derajat celsius.

Pada penelitian ini terdapat beberapa dokumentasi berbentuk video dan program yang dapat mempermudah memahami isi dari penelitian ini. Video tersebut dapat dilihat melalui laman berikut ini <https://www.youtube.com/watch?v=Zq5WOGTAKtY>.

IV. KESIMPULAN

Kesimpulan yang didapat dari hasil pengujian adalah sebagai berikut.

1. Sistem pada penelitian ini dapat bekerja dengan baik memenuhi tujuan dari penelitian ini.

2. Sistem pengaturan suhu air dapat bekerja dengan akurasi, dan memiliki nilai yang mendekati dengan hasil ukur menggunakan termometer dengan selisih ± 0.19 °C
3. Semakin banyak volume air yang hendak dipanaskan, maka sistem akan bekerja lebih lama untuk mencapai *set-point*.
4. Tegangan Dc yang diterima rangkaian Ac Dimmer dari mikrokontroler akan secara langsung berpengaruh terhadap output tegangan Ac rangkaian Ac dimmer. Semakin besar tegangan Dc yang diberikan maka semakin besar pula output tegangan Ac pada rangkaian Ac dimmer, sebaliknya semakin kecil tegangan Dc yang diberikan maka semakin kecil pula output tegangan Ac pada rangkaian Ac dimmer.
5. Dapat dilihat dari setiap parameter kontrol. *Rise time* dan *settling time* akan berkurang saat nilai K_p semakin tinggi, namun nilai *Overshoot* akan bertambah. Kemudian nilai *rise time* akan berkurang dan *overshoot* dengan *settling time* akan bertambah apabila nilai K_i semakin besar. Kemudian *overshoot* dan *settling time* akan berkurang apabila nilai K_d semakin besar

Pada penelitian ini, percobaan dilakukan dengan menggunakan air dalam volume yang sedikit, dikarenakan pemanas air yang digunakan pada project ini memiliki kemampuan yang terbatas dalam memanaskan air. Oleh karena itu, pengembangan selanjutnya dapat dilakukan dengan meningkatkan kemampuan dari sistem pemanas air khususnya pemanas air, sehingga dapat memanaskan air dalam jumlah yang banyak sekalipun.

DAFTAR PUSTAKA

- [1] I. H. Akyuz, E. Yolacan, H. M. Ertunc and Z. Bingul, "PID and state feedback control of a single-link flexible joint robot manipulator," *International Conference on Mechatronics*, pp. 409-414, 2011.
- [2] W. D. Chang, "PID control for chaotic synchronization using particle swarm optimization," *Chaos, Solutions and Fractals*, pp. 910-917, 2009.
- [3] A. B. Ptil and A. V. Salunkhe, "Adaptive Neuro Fuzzy Controller for Process Control System," *IEEE Region 10 and the Third International Conference on Industrial and Information Systems*, pp. 1-5, December 2008.
- [4] N. Jha, U. Singh, T. K. Saxena and A. Kapoor, "Online Adaptive Control for Non-Linear Processes Under Influence of External Disturbance," *IJAE*, vol. 2, no. 2, pp. 36-46, 2011.
- [5] O. P. Verma, R. Singla and R. Kumar, "Intelligent Temperature Controller for Water-Bath System," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 9, pp. 1178-1184, 2012.
- [6] T. P. Mote and S. D. Lokhande, "Temperature Control System Using ANFIS," *International Journal of Soft Computing and Engineering*, vol. 2, no. 1, pp. 156-161, 2012.
- [7] H. K. Jung, J. H. Kim, B. K. Lee, and D. W. Yoo, "A New PWM Dimmer Using Two Active Switches for AC LED Lamp," *The 2010 International Power Electronics Conference*, pp. 1547-1551, 2010.
- [8] L. Pengfei, L. Jiakun and J. Junfeng, "Wireless temperature monitoring system based on the ZigBee Technology," *2010 2nd International Conference on Computer Engineering and Technology*, vol. 1, pp. 160-163, 2010.
- [9] M. Aria, "PID Control of a Three-Degrees-Of-Freedom Model Helicopter," *Majalah Ilmiah Unikom*, vol. 9, no. 2, pp. 207-213, 2011.
- [10] M. R. Hasan, K. Arifin, A. Rahman and A. Azad, "Design, Implementation And Performance Of A Controller For Uninterruptible Solar Hot Water System," *IEEE*, pp. 584-588, 2011.
- [11] J. H. Yang and X. Y. Bi, "High-precision Temperature Control System Based on PID Algorithm," *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, vol. 12, pp. 568-571, 2010.
- [12] T. Aleksei, P. Eduard and B. Juri, "A Flexible MATLAB Tool for Optimal Fractional-order PID Controller Design Subject to Specifications," *Proceedings of the 31st Chinese Control Conference*, pp. 4698-4703, 2012.
- [13] H. L. Li and A. P. Hu, "A Direct AC-AC Converter for Inductive Power-Transfer Systems," *IEEE TRANSACTIONS ON POWER ELECTRONICS*, vol. 27, no. 2, pp. 661-668, February 2012.
- [14] P. N. Enjeti and S. Choi, "An approach to realize higher power PWM AC controller. In Proceedings Eighth Annual Applied Power Electronics Conference and Exposition," *IEEE*, pp. 323-327, March 1993.
- [15] O. Vainio and S. J. Ovaska, "Digital filtering for robust 50/60 Hz zero-crossing detectors," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 426-430, 1996.