

Sistem Pengaturan Suhu Air Menggunakan Kendali PID

Okky Janrizky*, Andy Panggabean*, Ahmad Riyad Firdaus*

* Politeknik Batam

Parkway Street, Batam Centre, Batam 29461, Indonesia

E-mail: 3210701024@students.polibatam.ac.id, 3210701012@students.polibatam.ac.id,
rifi@polibatam.ac.id

Abstrak

Makalah ini merupakan hasil penelitian mahasiswa tugas akhir program diploma tiga di Politeknik Batam [1] yang mengambil tema "Sistem Pengaturan Suhu Air Menggunakan Kendali PID". Penelitian ini merupakan studi implementasi konsep kendali PID (Proportional, Integral, Differential) dalam perangkat keras sebenarnya. Dalam konsep sebenarnya, kendali PID membutuhkan pemodelan sistem (penurunan model fisik menjadi model matematika) agar bisa dirancang sistem kendali yang optimal. Dalam penelitian ini, sistem pengaturan suhu air menggunakan kendali PID menggunakan metode "trial and error" dalam pemilihan konstanta pengendalinya (K_p , K_i , dan K_d), tidak melakukan permodelan sistem, dan implementasinya secara digital. Hasil pengujian sistem pengaturan ini memperlihatkan kinerja pengendali sebagai berikut : Rise time (s) : 0:18:58:17, Overhoot (%) : 4.88, Settling time (s) : 0:22:33:07, dan Steady State Error (%) : -0.23, Terlihat Rise time besar diakibatkan daya pemanas kurang besar untuk ukuran volume air yang dibuat.

Kata kunci : Kendali PID, Mikrokontroler, Zero Crossing Detector, dan Solid State Relay.

1. PENDAHULUAN

Penelitian ini dilatarbelakangi dengan maraknya penggunaan perangkat pemanas *water shower*, baik di hotel-hotel maupun di rumah tangga. Perangkat pemanas yang banyak di pasaran tidak dilengkapi dengan tampilan suhu air, dan pengaturan suhunya dilakukan secara manual, yang mengakibatkan suhu air yang diinginkan kadang terlalu panas atau kurang panas. Pada penelitian yang dilakukan pada tugas akhir mahasiswa ini, dicoba untuk membuat perangkat pengaturan suhu air yang dilengkapi dengan tampilan suhu air dan menggunakan metode kendali yang telah dipelajari selama perkuliahan yaitu kendali PID.

Sistem pengaturan suhu air diatur dan ditampilkan secara digital yang memanfaatkan kemampuan mikrokontroler Atmega8535 dalam akuisisi data dan mengambil keputusan. Keuntungan dari sistem ini adalah komponen rangkaian yang banyak dipasaran

yang harganya cukup terjangkau sehingga dalam membuat sistem pengaturan suhu ini biaya dapat dicapai, mudah dalam perawatan dan temperatur dapat terawasi secara langsung pada layar LCD, pengaturan suhu sesuai keinginan dengan memasukkan *setpoint* pada *push button*, kemudahan dalam pengoperasian. Sistem pengaturan suhu pada alat yang dirancang adalah menggunakan sistem kendali PID, yang diharapkan dapat memberikan kestabilan dalam pengaturan suhu air.

2. RUMUSAN MASALAH

Berdasarkan latar belakang yang telah diuraikan sebelumnya, penelitian pada Tugas Akhir ini bertujuan untuk merancang dan mengimplementasikan sistem pengaturan suhu air yang terpantau dan menggunakan kendali PID dalam perangkat keras secara digital. Dengan menggunakan metode kendali PID yang secara ilmiah bisa dibuktikan kestabilannya, dapat memberikan ketercapaian kestabilan pengaturan suhu air yang diinginkan.

Dalam merancang dan mengimplementasikan sistem pengaturan suhu air ini, telah dirumuskan beberapa permasalahan yang harus diselesaikan, yaitu: bagaimana menurunkan konsep kendali PID ke dalam sistem digital, bagaimana merancang perangkat keras dalam menerapkan konsep kendali PID, bagaimana membuat algoritma kendali PID dan menerapkannya dalam perangkat kendali (*controller*) yang digunakan, dan bagaimana memilih konstanta kendali PID yang optimal.

3. METODOLOGI PENELITIAN

Untuk menyelesaikan tugas akhir ini, dilakukan langkah-langkah sebagai berikut:

1. Studi Pustaka/literature.

Studi ini dilakukan dengan cara mencari dan membaca *literature* serta karya-karya "Pengendali Suhu Kontrol PID" yang sudah di buat sebelumnya dan data-data yang berhubungan dengan alat yang akan dibuat.

2. Studi Lapangan.

Studi ini dilakukan untuk memperoleh alat dan bahan yang diperlukan untuk merancang dan membuat sistem pengaturan suhu air pada sebuah

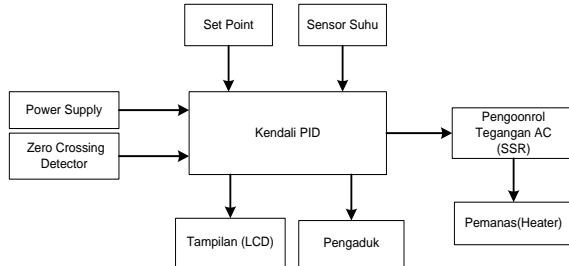
- wadah.
3. perancangan dan pengembangan sistem.
Pada tahap ini dilakukan perancangan dan pengembangan *hardware* maupun *software* untuk sistem pengaturan suhu air.
 4. Melakukan pengujian alat.
Studi ini meliputi pengujian sistem yang dirancang dengan menggunakan beberapa parameter pengujian sehingga diperoleh data hasil pengujian yang selanjutnya dapat dianalisis.

4. IKHTISAR SISTEM

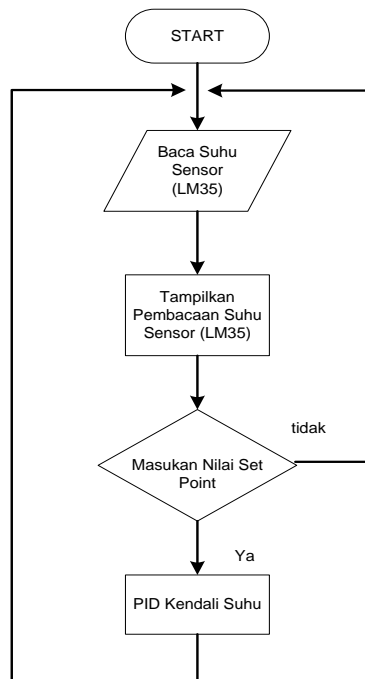
4.1 Deskripsi Umum

Perancangan dan pembuatan *sistem pengaturan suhu air* ini terdiri atas beberapa komponen penting yaitu *Power Supply*, Sensor Suhu, Tombol *Set Point*, Tampilan (*LCD*), Kendali *PID* yang menggunakan Mikrokontroler *ATMega8535*, Sinyal *Zero Crossing Detector*, Pengontrol Tegangan AC (*SSR*) dan Pemanas (*Heater*).

Blok diagram dari sistem alat ini seperti yang terlihat pada Gambar 1 di bawah ini:



Gambar 1. Blok Diagram Rangkaian Sistem Pengaturan Suhu Air



Gambar 2 Algoritma Sistem Pengaturan Suhu Air

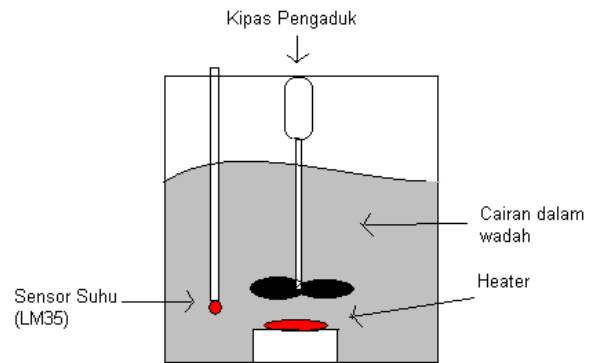
Dari Gambar 1 dapat dilihat, bahwa kendali *PID*

adalah suatu pengolahan data yang utama dalam sistem. Pembacaan suhu menggunakan *LM35* yaitu sensor pendeteksi suhu yang dapat mengubah besaran suhu menjadi besaran elektrik berupa tegangan. Pemanasan air menggunakan elemen pemanas yang dikendalikan oleh Pengontrol Tegangan AC (*SSR*), dan Mikrokontroler sebagai pengendali utama. Hasil pengolahan data ditampilkan pada *LCD* (*Liquid Crystal Display*). Secara umum algoritma sistem pengaturan suhu air ini dapat dilihat pada Gambar 2.

4.2 Perancangan Sistem

Sistem Pengaturan Suhu

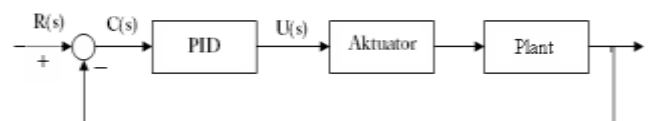
Air yang berada di dalam wadah dipanaskan menggunakan sebuah pemanas (*heater*) listrik. air diaduk sedemikian rupa sehingga seluruh cairan yang berada dalam wadah dapat dianggap memiliki temperatur yang sama.



Gambar 3. Tata letak komponen sistem

Dalam kasus sistem pengaturan, suhu adalah variabel yang akan diatur, dan nilai suhu inilah yang diinginkan sebagai *output*. Kemudian *input* kendali adalah *output* dari pemanas listrik (*electric heater*). Besarnya kalor sebagai *input* kendali selalu diatur dengan mengatur tegangan yang diberikan ke pemanas. Namun telah ditunjukkan bahwa besarnya temperatur dapat diatur dengan mengatur besarnya tegangan yang diberikan ke pemanas.

Kendali *PID* merupakan salah satu jenis pengatur yang banyak digunakan. Selain itu sistem ini mudah digabungkan dengan metoda pengaturan yang lain seperti *Fuzzy* dan *Robust* [3]. Sehingga akan menjadi suatu sistem pengatur yang semakin baik. Tulisan ini dibatasi pada sistem dengan *Unity Feedback System* seperti terlihat pada Gambar 4 [3].



Gambar 4. Blok diagram untuk *Unity Feedback Systems*

Kendali *PID* Sebenarnya terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu *P* (*Proportional*) *C* (*Derivative*) *C* (*Controller*), *D* (*Derivative*) *C* (*Controller*),

dan I (Integral) Controller [3],[10],[11]. Masing-masing memiliki parameter tertentu yang harus diset untuk dapat beroperasi dengan baik, yang disebut sebagai konstanta. Setiap jenis, memiliki kelebihan dan kekurangan masing-masing, hal ini dapat dilihat pada tabel di bawah ini :

Tabel 1. Respon PID Controller Terhadap Perubahan Konstanta

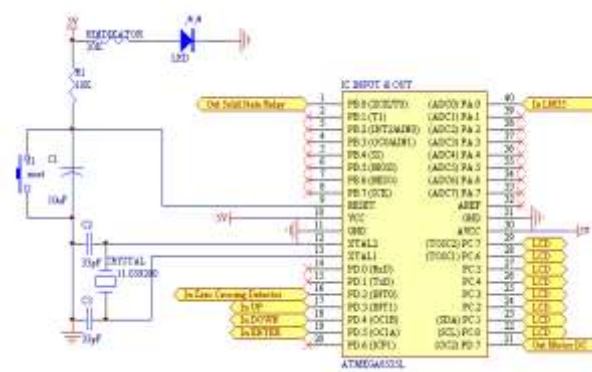
Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady State Error
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small change	Decrease	Decrease	Small change

Parameter-parameter tersebut, tidak bersifat berdiri sendiri, sehingga pada saat salah satu nilai konstantanya diubah, maka mungkin sistem tidak akan bereaksi seperti yang diinginkan [3],[10],[11]. Tabel 1 di atas hanya dipergunakan sebagai pedoman jika akan melakukan perubahan konstanta. Untuk merancang suatu kendali PID , biasanya dipergunakan metoda *trial & error*. Sehingga perancang harus mencoba kombinasi pengatur beserta konstantanya untuk mendapatkan hasil terbaik yang paling sederhana.

Mikrokontroler AVR ATmega8535

Mikrokontroler adalah sebuah sistem yang dapat mengontrol peralatan elektronik[2]. Sebuah mikrokontroler umumnya berisi seluruh memori dan antarmuka I/O yang dikemas dalam sebuah chip.

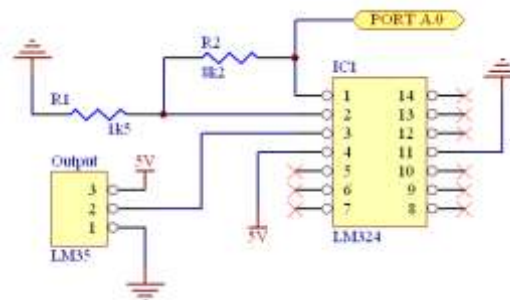
Mikrokontroler AVR memiliki arsitektur RISC (Reduced Instruction Set Computing) 8 bit yang instruksinya dikemas dalam kode 16 bit dan dieksekusi dalam satu siklus clock. Mikrokontroler Atmega 8535 memiliki fitur sebagai berikut: Frekuensi *clock* maksimum 16 MHz, Jalur I/O 32 buah, yang terbagi dalam PortA, PortB, PortC dan PortD, *Analog to Digital Converter* 10 bit sebanyak 8 *input*, *Timer/Counter* sebanyak 3 buah, CPU 8 bit yang terdiri dari 32 *register*, *Watchdog Timer* dengan *osilator internal*, SRAM sebesar 512 byte, *Memori Flash* sebesar 8 Kbyte dengan kemampuan *read while write*, *Interrupt internal* maupun *eksternal*, *EEPROM* sebesar 512 byte yang dapat diprogram saat operasi, *Analog Comparator*, Komunikasi *serial USART* dengan kecepatan maksimal 2,5 Mbps.



Gambar 5. Rangkaian Mikrokontroler

Sensor Suhu LM35

LM35 sebagai sensor suhu yang teliti dan terkemas dalam bentuk Integrated Circuit (IC), dimana output tegangan keluaran sangat linear tergantung dengan perubahan suhu. Sensor ini berfungsi sebagai pengubah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar $10 \text{ mV} / ^\circ\text{C}$ yang berarti bahwa kenaikan suhu 1°C maka akan terjadi kenaikan tegangan sebesar 10 mV [5].

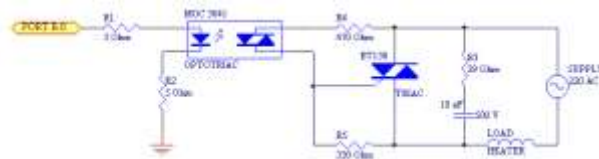


Gambar 6. Rangkaian Sensor Suhu LM35

Pengatur Tegangan AC

Sebagai pengatur tegangan, digunakan rangkaian *Solid State Relay* (SSR) yang pada prinsipnya adalah penggabungan antara rangkaian *optocoupler* yang menggunakan *Zero Crossing Circuit* di dalam *MOC 3041* dan *TRIAC* [6],[7],[9]. Gambar 7 menunjukkan gambar rangkaian SSR, sedangkan cara kerja rangkaian ini adalah sebagai berikut [7],[9] : Jika ada arus melewati MOC kaki 1 dan 2, Hal ini akan mengakibatkan dioda yang berada didalam *MOC 3041* aktif dan transistor yang berada di dalam *MOC 3041* juga aktif. Keadaan ini akan mengakibatkan arus dari jala-jala 220VAC mengalir ke kaki *Gate TRIAC* dan akan memicu *TRIAC* tersebut. Pemicuan ini mengakibatkan kaki MT1 dan MT2 akan terhubung dan jala-jala 220VAC akan mengalir melalui beban. Hal ini berakibat beban ON. C $10\text{nF}/500\text{V}$ pada rangkaian ini berfungsi untuk mengurangi arus yang sangat besar saat beban mulai ON. Jika tidak ada arus melewati MOC kaki 1 dan 2, Hal ini mengakibatkan dioda dan transistor yang berada di dalam *MOC 3041* tidak aktif dan tidak akan ada pemicuan pada *TRIAC* sehingga beban tidak terhubung ke jala-jala 220VAC atau dengan kata lain beban OFF. Pemberian logika 0

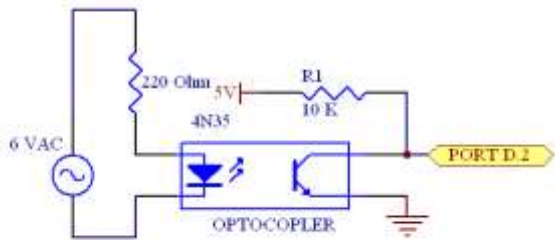
ataupun logika 1 tersebut dikendalikan oleh mikrokontroler.



Gambar 7. Rangkaian Solid State Relay

Zero Crossing Detector

Zero Crossing Detector seperti yang bisa dilihat pada Gambar 8 adalah rangkaian yang digunakan untuk mendeteksi gelombang sinus AC 220 volt saat melewati titik tegangan nol. Seberangan titik nol yang dideteksi adalah peralihan dari positif menuju negatif dan peralihan dari negatif menuju positif [8].



Gambar 8. Rangkaian Zero Crossing Detector

Perangkat Lunak dan Algoritma Kendali

Perancangan perangkat lunak secara garis besar bertujuan untuk mengatur kerja sistem, seperti pembacaan hasil sensor, proses pengaturan temperatur sistem dengan menggunakan algoritma kontrol PID digital, serta menghasilkan sinyal pengatur tegangan AC yang akan menggerakkan aktuator. Dengan demikian perancangan perangkat lunak ini meliputi program utama serta beberapa fungsi-fungsi pendukung. Program utama berperan sebagai jantung perangkat lunak yang akan mengatur keseluruhan operasi yang melibatkan fungsi-fungsi pendukung. Sedangkan fungsi-fungsi pendukung akan melakukan kerja khusus sesuai kebutuhan dari program utama

Dalam mengimplementasikan konsep kendali ke dalam mikrokontroler ATmega 8535, digunakan bahasa C dengan algoritma sebagai berikut:

```
void PID()
{
    float Deriv,Integ,error,error_akhir;

    error=i-suhu;
    Deriv=error-error_akhir;
    Integ = Integ + error;

    SuhuKendali=(int) ((KP*error)+(KI*Integ)+(KD*Deriv));
    if (SuhuKendali>MaxSuhu) SuhuKendali=MaxSuhu;
    if (SuhuKendali<MinSuhu) SuhuKendali=MinSuhu;
    pwm=255-SuhuKendali;
    error_akhir=error;
}
```

5. PENGUKURAN DAN PENGUJIAN SISTEM

Pengujian alat secara keseluruhan juga dilakukan dengan proses menempatkan sensor suhu dan *thermometer digital* dalam alat dengan suhu yang sama, kemudian membandingkan antara suhu pengukuran dan suhu yang tertampil pada LCD.

Berikut adalah data hasil pengukuran pada pemanas air seperti yang dapat dilihat pada Tabel 2.

Tabel 2. Data Pengukuran Dan Pengujian Alat Secara Keseluruhan

Tampilan Suhu Pada LCD(°C)	Tampilan Suhu Pada Thermometer (°C)	Selisih
30	30.5	0.5
33	33.2	0.2
37	36.8	0.2
40	39.7	0.3
43	43.3	0.3
45	45.7	0.7
48	47.6	0.4
51	51.1	0.1
54	54.5	0.5
57	56.6	0.4
60	60.4	0.4

$$\% \text{ Error Suhu} = \text{Jumlah Selisih} / \text{jumlah percobaan}$$

$$\% \text{ Error Suhu} = 4 / 11 = 0.36 \text{ } ^\circ \text{C}$$

Untuk pemberian nilai parameter Kp, Kd dan Ki untuk memperoleh respon sistem yang optimal dilakukan dengan menggunakan metode manual (*trial and error*). Adapun langkah-langkah penalaan dilakukan sebagai berikut :

- Suhu air dinaikan hingga mencapai suhu 43°C.
- Memberikan nilai Kp, Ki dan Kd hingga kontroler dengan mengamati lampu indikator dan suhu yang tertampil di *thermometer*.

Tabel 3 memperlihatkan data hasil pengukuran kinerja pengendali dengan nilai konstanta kendali sebagai berikut:

Tabel 3. Data Pengukuran kinerja pengendali

Konstanta Kendali			Respon			
Kp	Ki	Kd	Rise Time (s)	Overshoot (%)	Settling Time (s)	Steady State Error (%)
60		10	0:22:20:41	0.93	0:22:42:14	-4.18
60		5	0:21:44:19	2.55	0:23:03:10	-4.65
60		3	0:21:30:54	3.95	0:23:34:19	-5.34
80		10	0:19:05:46	5.11	0:21:15:05	2.32
80		5	0:19:20:31	5.34	0:21:21:11	0.69
80		3	0:18:47:21	7.91	0:21:25:04	-0.93
100		10	0:17:43:09	9.76	0:20:31:05	8.37
100		5	0:17:55:16	10.4	0:20:39:08	6.27
100		3	0:16:20:02	13.02	0:20:52:03	3.72
60	0.1	5	0:21:35:14	1.86	0:23:43:24	-7.91
80	0.1	5	0:18:58:17	4.88	0:22:33:07	-0.23
100	0.1	5	0:17:20:12	7.44	0:21:40:01	5.81

- $\% \text{ Overshoot} = ((\text{SuhuOut} - \text{SuhuTeori}) / \text{SuhuTeori}) * 100\%$
- $\% \text{ Steady State Error} = ((\text{SuhuOut} - \text{SuhuTeori}) / \text{SuhuTeori}) * 100\%$

Berdasarkan hasil percobaan untuk *set point* 43°C, diperoleh data tanggapan waktu paling optimal dengan *rise time* selama 18 menit 58 detik, persen *overshoot* dan *steady state error* terkecil. Kinerja pengendali ini diperoleh dengan nilai konstanta kendali $K_p=80$, $K_i=0.1$, dan $K_d=5$.

6. ANALISA SISTEM

Dari Tabel 3 memperlihatkan kinerja sistem pengaturan suhu air ini belum optimal untuk tanggapan waktunya, karena dari data tersebut waktu yang dibutuhkan sistem untuk mencapai nilai *setpoint* masih cukup lama sekitar 18 menit, sedangkan untuk % *overshoot* dan % *steady state error* memperlihatkan nilai yang cukup kecil, artinya kinerja pengendali ini cukup baik.

Dari pengujian kinerja sistem kendali, pemberian nilai K_p digunakan untuk memperbaiki *rise time*. Dari pengujian pemberian nilai K_p dibawah 80 mengakibatkan waktu *rise time* menjadi lebih lama, sedangkan % *overshoot* semakin kecil, *settling time*-nya akan semakin besar dan *steady state error*-nya akan semakin besar dan sebaliknya pemberian nilai K_p lebih besar akan mengakibatkan waktu *rise time* semakin cepat, *overshoot* semakin besar, *settling time* semakin cepat, dan *steady state error*-nya akan semakin besar

Nilai K_d digunakan untuk memperbaiki *overshoot*. Pemberian nilai K_d dibawah 5 mengakibatkan *rise time* semakin cepat, *overshoot* semakin besar, *settling time* semakin lama, dan *steady state error*-nya akan semakin besar, dan sebaliknya pemberian nilai K_d diatas 5 akan mengakibatkan *rise time*-nya semakin lama, *overshoot* semakin kecil, *settling time* semakin cepat dan *steady state error*-nya akan semakin besar.

Nilai K_i digunakan untuk memperbaiki *steady state error*. Pemberian nilai K_i sebesar 0.1 mengakibatkan *rise time* akan semakin cepat, *overshoot* semakin kecil, *settling time* semakin lama, namun *steady state error*-nya lebih baik dari pada jika hanya menerapkan kendali PD yang nilai konstantanya $K_p=80$ dan $K_d=5$.

7. KESIMPULAN DAN SARAN

Kesimpulan

Dari hasil analisa data diatas dapat diambil beberapa kesimpulan diantaranya:

1. Sistem pengaturan suhu air ini mempunyai keakuratan pengukuran suhu yang mendekati dengan thermometer.
2. Alat ini bekerja dengan baik dan mencapai tujuan yang diinginkan, walaupun tanggapan waktu yang

diperoleh masih cukup besar, yaitu 18 menit dan 58 detik.

3. Alat ini dapat memanaskan air sesuai dengan suhu air yang dibutuhkan kisaran suhu 30°C-60°C.
4. Menggunakan daya heater yang lebih kecil bila di bandingkan volume air mengakibatkan *rise time* (tanggapan waktu) atau waktu mencapai *setpoint* akan semakin lama.

Saran

Untuk sistem yang lebih baik kedepannya perlu ditambahkan beberapa hal diantaranya :

1. Untuk mendapatkan pembacaan suhu air yang lebih baik lagi, lebih baik menggunakan sensor suhu yang khusus untuk pembacaan suhu dalam air.
2. Dalam pembuatan sistem pengaturan suhu air ini dapat dibuat permodelan sistem alat agar dapat memperoleh nilai konstanta K_p, K_d dan K_i yang lebih baik.
3. Membuat aplikasi sistem monitoring kinerja pengendali berbasis komputer.
4. Implementasi pada aplikasi *watershower* sesungguhnya.

8. REFERENSI

- [1]. Okky Janrizky, Andy Panggabean, "Sistem Pengaturan Suhu Air Menggunakan Kendali PID", Tugas Akhir Diploma III Program Studi Teknik Elektronika Politeknik Batam, 2010
- [2]. Budihartono, Widodo. 2005. *Perancangan Sistem dan Aplikasi Mikrokontroler*. Jakarta: PT Elex Media Komputindo.
- [3]. Pitowarno, Endra, 2006. *Robotika*. Yogyakarta : CV Andi Offset.
- [4]. Rashid, Mehammad H. 1999. *Elektronika Daya*. Jakarta: PT Prenhallindo.
- [5]. <http://www.alldatasheet.com/datasheet-pdf/pdf/8866/NSC/LM35.html> [08-04-2010 16:23]
- [6]. <http://www.scribd.com/doc/33212330/SCR-DIAC-TRIAC>
- [7]. <http://www.avrku.com/2009/09/driver-beban-ac-tanpa-relay.html> [12-04-2010 20:22]
- [8]. <http://www.alldatasheet.com/datasheet-pdf/pdf/2848/MOTOROLA/4N35.html> [15-04-2010 17:05]
- [9]. <http://www.alldatasheet.com/datasheet-pdf/pdf/5039/MOTOROLA/MOC3041.html> [15-04-2010 17:05]
- [10]. <http://id.wikipedia.org/wiki/PID> [02-05-2010 22:31]
- [11]. <http://meriwardana.blogspot.com/2010/06/mengenal-pengendali-pid-pid-controller.html> [02-05-2010 22:40]