

Simulator *input-output* sistem kontrol menggunakan Raspberry Pi

Zainal Bachrudin, Catur Edi Widodo dan Kusworo Adi

Departemen Fisika, Fakultas Sains dan Matematika, Universitas Diponegoro, Semarang

E-mail: zainalbachrudin@st.fisika.undip.ac.id

ABSTRACT

In this research has been made I / O simulator which is a tool to simulate input and output of a control system using Raspberry Pi. Raspberry Pi has 26 GPIO (General Purpose Input and Output) pins that can be used to control inputs and outputs on the I / O simulator. The 26 GPIO pins are divided into two main systems, is 13 GPIO pins that are odd numbered as inputs and 13 other GPIO pins which are even numbered as outputs. The Raspberry Pi GPIO pins are ordered as inputs and outputs using Python programming languages. The command is done by reading the switch as input signal input, then Raspberry Pi process the input signal and send data as output signal with LED flame on the I / O Simulator. The I / O simulator can simulate logic gates, as AND, OR, NOT, and ADD, and can run mini distillation plant.

Keywords: *Simulation, Input-Output, Raspberry Pi, Python.*

ABSTRAK

Dalam penelitian ini telah dibuat simulator I/O yang merupakan alat untuk mensimulasikan input dan output suatu sistem kontrol menggunakan Raspberry Pi. Raspberry Pi memiliki 26 pin GPIO (General Purpose Input dan Output) yang dapat digunakan untuk mengendalikan input dan output pada Simulator I/O. 26 pin GPIO tersebut dibagi menjadi dua sistem utama, yaitu 13 pin GPIO yang bernomor ganjil sebagai input dan 13 pin GPIO lainnya yang bernomor genap sebagai output. Pin-pin GPIO Raspberry Pi tersebut diperintah sebagai input maupun output menggunakan Bahasa pemrograman Python. Perintah dilakukan dengan membaca saklar sebagai sinyal masukan pengganti input, kemudian Raspberry Pi mengolah sinyal input dan mengirim data berupa sinyal output yang ditandai dengan nyala LED pada Simulator I/O. Simulator I/O dapat mensimulasikan gerbang logika AND, OR, NOT, dan ADD, serta dapat menjalankan mini plant distilasi.

Kata kunci: *Simulasi, Input-Output, Raspberry Pi, Python.*

PENDAHULUAN

Ilmu pengetahuan dan teknologi yang berkembang pesat memberi peluang para praktisi, seperti: *programmer* dan *engineer* untuk selalu melakukan penelitian yang berguna, antara lain untuk membantu pekerjaan manusia dan menanggulangi permasalahan yang sedang dihadapi. Hal ini dapat dilihat dengan bermunculannya ide-ide kreatif dan inovatif dalam bidang ilmu pengetahuan dan teknologi yang sangat membantu kehidupan manusia agar menjadi lebih mudah. Sistem kontrol merupakan salah satu bidang ilmu dan teknologi yang saat ini banyak dikembangkan di perusahaan-

perusahaan untuk mengendalikan alat-alat instrumentasi elektronika secara otomatis [1].

Sistem kontrol adalah interkoneksi komponen membentuk konfigurasi sistem yang akan memberikan respon sistem yang diinginkan. Sehingga, komponen-komponen tersebut dapat dikendalikan oleh hubungan input-output yang merupakan hubungan sebab-akibat dari suatu proses, yang pada gilirannya merupakan pengolahan sinyal input untuk memberikan variabel sinyal output [2]. Kontrol otomatis berfungsi untuk membandingkan nilai aktual dari output *plant* dengan input referensi (nilai yang diinginkan) [3].

Raspberry Pi merupakan sebuah komputer yang berukuran kecil yang dapat

digunakan seperti sebuah *personal computer* (PC) [4]. Dengan memanfaatkan fitur GPIO (general purpose input output), Raspberry Pi dapat mengendalikan system operasi seperti halnya alat pengontrol PLC (*Programmable Logic Controller*). Raspberry Pi dapat diprogram dengan Bahasa pemrograman Python. Raspberry Pi dan python sebagai paket perangkat keras dan perangkat lunak yang diharapkan dapat menjadi solusi untuk menghadapi masalah teknik kontrol dalam dunia industri [5].

DASAR TEORI

Raspberry Pi

Raspberry Pi adalah komputer kecil dengan dimensi 85,60 mm x 53,98 mm x 17 mm, dan beratnya hanya 45 g. Ukurannya yang kecil sangat cocok untuk otomasi rumahan, karena perangkat kecil dapat dengan mudah diletakkan dalam kotak dan dipasang di dalam kotak listrik, atau pengganti perangkat termostat yang ada di dinding [6]. Raspberry Pi merupakan komputer mikro *single board* yang dikembangkan dengan tujuan mengajarkan dasar-dasar ilmu komputer dan pemrograman. Raspberry Pi sangat berbeda dari mikrokontroler jenis lainnya, seperti Arduino, yang lebih populer untuk proyek-proyek prototyping, karena sesungguhnya Raspberry Pi adalah komputer. Raspberry Pi dapat digunakan untuk banyak hal seperti halnya komputer, misalnya: *spreadsheet*, *word processing*, pemrograman, dan lain-lain [7].

GPIO (General Purpose Input-Output)

GPIO (*General Purpose Input-Output*) adalah pin yang digunakan untuk membaca *input* dan mengontrol *output* berdasarkan kondisi yang berbeda sesuai dengan program yang telah dibuat pada Raspberry Pi [8]. Beberapa pin GPIO dapat digunakan sebagai *input/output* digital dan sebagai antarmuka untuk protocol. Selain itu, pin GPIO juga

dapat diakses untuk mengendalikan *output* perangkat keras, seperti LED, motor, dan relay. Adapun *input*, Raspberry Pi dapat membaca status tombol, *switch*, dan cepat, atau dapat membaca sensor, seperti suhu, cahaya, dan gerak [6]. Raspberry memiliki 40 pin yang terdiri dari: 2 pin sebagai sumber tegangan 5 V, 2 pin sumber tegangan 3,3 V, 8 pin *ground*, 26 pin GPIO dan 2 pin serial [9], seperti terlihat pada Gambar 1.

GPIO Numbers

Raspberry Pi B Rev 1 P1 GPIO Header			Raspberry Pi A/B Rev 2 P1 GPIO Header			Raspberry Pi B+ B+ J8 GPIO Header		
Pin No.	Pin No.	Pin No.	Pin No.	Pin No.	Pin No.	Pin No.	Pin No.	Pin No.
3.3V	1	2	3.3V	1	2	3.3V	1	2
GPIO0	3	4	GPIO2	3	4	GPIO2	3	4
GPIO1	5	6	GPIO3	5	6	GPIO3	5	6
GPIO4	7	8	GPIO4	7	8	GPIO4	7	8
GND	9	10	GND	9	10	GND	9	10
GPIO17	11	12	GPIO17	11	12	GPIO17	11	12
GPIO21	13	14	GPIO27	13	14	GPIO27	13	14
GPIO22	15	16	GPIO22	15	16	GPIO22	15	16
3.3V	17	18	3.3V	17	18	3.3V	17	18
GPIO10	19	20	GPIO10	19	20	GPIO10	19	20
GPIO8	21	22	GPIO8	21	22	GPIO8	21	22
GPIO11	23	24	GPIO11	23	24	GPIO11	23	24
GND	25	26	GND	25	26	GND	25	26
						DNC	27	DNC
						GPIO5	29	GND
						GPIO6	31	GPIO12
						GPIO13	33	GND
						GPIO19	35	GPIO15
						GPIO26	37	GPIO20
						GND	39	GPIO21

Key
 Power + UART
 GND SPI
 I²C GPIO

Gambar 1. Konektor GPIO Raspberry Pi [6]

Bahasa pemrograman Python

Python merupakan bahasa pemrograman yang berorientasi obyek dinamis yang paling mudah untuk dipelajari sehingga dapat digunakan untuk bermacam-macam pengembangan perangkat lunak. Python menyediakan dukungan yang kuat untuk integrasi dengan bahasa pemrograman lain dan alat-alat bantu lainnya. Python hadir dengan pustaka-pustaka standar yang dapat diperluas [10].

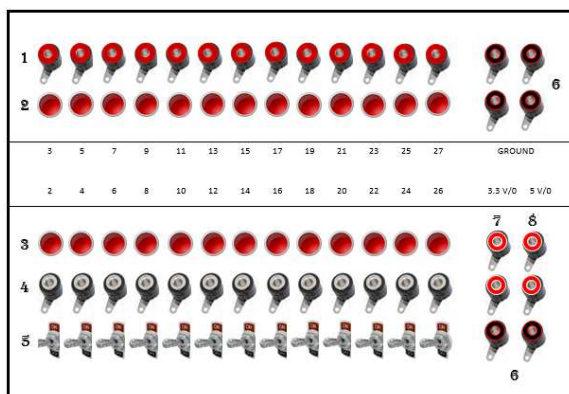
Beberapa fitur yang dapat dikatakan sebagai keunggulan bahasa pemrograman Python adalah :

1. Python sangat kuat dan cepat
2. Python dapat berjalan baik dengan bahasa pemrograman lain
3. Python dapat berjalan untuk banyak sistem operasi
4. Python mudah dipelajari
5. Python terbuka

RANCANGAN DAN REALISASI ALAT

Deskripsi dan rancangan simulator I/O

Simulator I/O terdiri dari beberapa komponen, yaitu Raspberry Pi B+, LED warna merah, saklar *toggle switch*, *socket*, dan kabel jumper. Komponen-komponen tersebut dirangkai pada box akrilik dan saling dihubungkan satu dengan lainnya seperti pada Gambar 2. Alat ini disebut simulator I/O karena secara umum terdiri dari dua sistem utama, yaitu sistem *input* dan sistem *output*. Simulator I/O memiliki *input* dan *output* masing-masing berjumlah 13 pin, sesuai dengan jumlah pin GPIO Raspberry Pi tipe B+, yaitu 26 pin GPIO. *Input* simulator I/O adalah pin GPIO Raspberry Pi bernomor ganjil, mulai dari pin GPIO 3 sampai pin GPIO 27, sedangkan *output* simulator I/O adalah pin GPIO Raspberry Pi bernomor genap, mulai dari pin GPIO 2 sampai pin GPIO 26.



Gambar 2. Rancangan simulator I/O

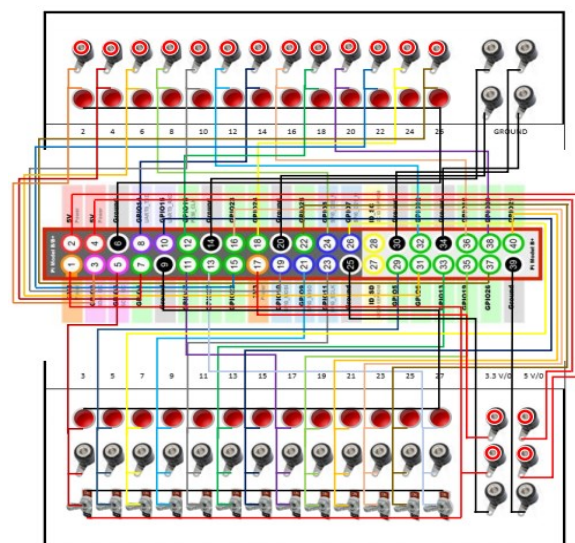
Keterangan:

1. Konektor dengan output *plant*
2. Indikator LED output
3. Indikator LED input
4. Konektor dengan input *plant*
5. Saklar (pengganti input)
6. Konektor ground
7. Konektor tegangan 3,3V
8. Konektor tegangan 5V

Simulator I/O terdiri dari 26 LED, yaitu: 13 LED sebagai indikator input dan 13 LED sebagai indikator output. Pada bagian input simulator I/O terhubung dengan saklar sebagai masukan kepada sistem I/O. Simulator I/O juga diberi *socket* yang terhubung ke *input* dan *output*, serta sumber tegangan dan *ground*.

Wiring simulator I/O

Wiring simulator I/O ke pin Raspberry Pi terdapat 13 pin *input* dan 13 pin *output* yang digunakan untuk mensimulasikan suatu sistem kontrol ditunjukkan pada Gambar 3. Simulator *input* dihubungkan dengan pin GPIO bernomor ganjil, sedangkan simulator *output* dihubungkan dengan pin GPIO bernomor genap. Semua pin GPIO dihubungkan dengan kaki positif LED, sedangkan kaki negatif LED dihubungkan dengan *ground*. Untuk simulator *input*, pin GPIO dihubungkan dengan terminal sentral pada saklar *toggle switch*, sedangkan terminal ON dihubungkan dengan sumber tegangan. Selain itu, semua pin Raspberry Pi, baik pin GPIO, voltase, maupun *ground* juga dihubungkan dengan *socket* sebagai sarana penghubung menuju terminal *output*, seperti sensor dan alat instrumentasi elektronika lainnya.



Gambar 3. Wiring simulator I/O

Program I/O

Langkah-langkah dalam pembuatan program *input* dan *output* simulator I/O adalah:

1. Mengaktifkan pin GPIO Raspberry Pi dengan mengimpor modul "RPi.GPIO" ke Raspberry Pi. Perintah yang digunakan adalah "import RPi.GPIO as GPIO".
2. Mendefinisikan pin GPIO Raspberry Pi sebagai *Broadcom*. *Broadcom* merupakan penomoran pin GPIO berdasarkan nomor urut pin GPIO Raspberry Pi B+, yaitu pin GPIO 2 sampai pin GPIO 27. Program pada simulator I/O menggunakan perintah "GPIO.setmode(GPIO.BCM)". Urutan penomoran pin GPIO pada Raspberry Pi B+ dapat dilihat pada Gambar 4.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I2C)		DC Power 5v	04
05	GPIO03 (SCL1, I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Gambar 4. Nomor pin dan port GPIO Raspberry Pi B+ [8]

3. Mendefinisikan pin GPIO sebagai *input/output*. Pada simulator I/O, pin GPIO hanya difungsikan sebagai *input* dan *output*. Pin GPIO bernomor ganjil didefinisikan sebagai *input* menggunakan perintah "GPIO.setup(pin_GPIO,GPIO.IN)", sedangkan pin GPIO bernomor genap

didefinisikan sebagai *output* menggunakan perintah

"GPIO.setup(pin_GPIO,GPIO.OUT)".

4. Mengaktifkan dan menonaktifkan pin GPIO. Untuk mengaktifkan pin GPIO digunakan perintah "HIGH", "TRUE" atau dengan logika angka "1". Sedangkan untuk menonaktifkan pin GPIO adalah kebalikannya, yaitu menggunakan perintah "LOW", "FALSE" atau menggunakan logika angka "0". Perintah program untuk mengaktifkan dan menonaktifkan pin GPIO sebagai *output* adalah "GPIO.output(pin_GPIO,1)" dan "GPIO.output(pin_GPIO,0)". Sedangkan *input* pada program I/O didefinisikan sebagai variable, misalnya dengan perintah "a=GPIO.input(pin_GPIO)". Dengan demikian, pin GPIO pada input dapat diaktifkan atau dinonaktifkan dengan mendefinisikan variable *a* sebagai logika angka "1" atau "0".

Contoh: $a = 1$ (mengaktifkan input)

$a = 0$ (menonaktifkan input)

PENGUJIAN ALAT

Pengujian Simulator I/O

Pada simulator I/O dibuat program simulasi *input-output* gerbang logika sebagai berikut:

- a. Gerbang logika AND

Gerbang logika AND merupakan suatu kondisi yang hanya akan menghasilkan *output* bernilai 1 bila semua *input* bernilai 1. Apabila salah satu saja input bernilai 0, maka *output* juga akan bernilai 0. Hasil simulasi ditunjukkan pada Tabel 1. Perintah yang digunakan untuk mensimulasikan gerbang logika AND pada simulator I/O menggunakan bahasa pemrograman python sebagai berikut:

```
a=GPIO.input(5)
b=GPIO.input(7)
c=GPIO.input(9)
m=a and b and c
```

```
if m==1:
    GPIO.output(6,GPIO.HIGH)
else:
    GPIO.output(6,GPIO.LOW)
```

Tabel 1. Hasil Simulasi I/O gerbang logika AND

No.	Input			Output M
	a	b	c	
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0
5	1	1	0	0
6	1	0	1	0
7	0	1	1	0
8	1	1	1	1

b. Gerbang logika OR

Berbeda dengan gerbang logika AND, gerbang logika OR akan bernilai 1 bila salah satu/lebih dari *input* bernilai 1. Jadi, *output* gerbang logika OR hanya akan bernilai 0 bila semua *input* bernilai 0 seperti pada Tabel 2 dengan perintah sebagai berikut:

```
p=GPIO.input(5)
q=GPIO.input(7)
r=GPIO.input(9)
n=p or q or r

if n==1:
    GPIO.output(6,GPIO.HIGH)
else:
    GPIO.output(6,GPIO.LOW)
```

Tabel 2. Hasil Simulasi I/O gerbang logika OR

No.	Input			Output N
	p	q	r	
1	0	0	0	0
2	1	0	0	1
3	0	1	0	1
4	0	0	1	1
5	1	1	0	1
6	1	0	1	1
7	0	1	1	1
8	1	1	1	1

c. Gerbang logika NOT

Gerbang logika NOT merupakan suatu kondisi yang akan menghasilkan output berupa kebalikan dari *input*. Apabila *input* bernilai 0, maka *output* akan bernilai 1. Sebaliknya bila *input* bernilai 1, maka *output* akan bernilai 0. Hasil simulasi I/O gerbang logika NOT dapat dilihat pada Tabel 3 dengan perintah gerbang logika NOT sebagai berikut:

```
j=GPIO.input(5)
z=(j==1)

if j==1:
    GPIO.output(4,0)
else:
    GPIO.output(4,1)
```

Tabel 3. Hasil Simulasi I/O gerbang logika NOT

No.	Input		Output z
	j		
1	1		0
2	0		1

d. Gerbang logika ADD

Hasil simulasi I/O gerbang logika ADD dapat dilihat pada Tabel 4. Gerbang logika ADD memiliki dua *output* yang berbeda, yaitu gerbang logika NOR dan gerbang logika AND. Gerbang logika NOR merupakan keadaan yang akan menghasilkan *output* bernilai 1 bila salah satu atau sebagian dari *input* bernilai satu, sedangkan bila semua *input* bernilai 1 maka akan menghasilkan *output* AND. Perintah yang digunakan untuk mensimulasikan gerbang logika ADD adalah:

```
x=GPIO.input(5)
y=GPIO.input(7)
s=(x==1 and y==0) or (x==0 and y==1)
c=x and y

if s==1:
    GPIO.output(4,GPIO.HIGH)
else:
    GPIO.output(4,GPIO.LOW)
```



```

if c==1:
    GPIO.output(6,GPIO.HIGH)
else:
    GPIO.output(6,GPIO.LOW)

```

Tabel 4. Hasil Simulasi I/O gerbang logika ADD

No.	Input		Output	
	x	y	s	c
1	0	0	0	0
2	1	0	1	0
3	0	1	1	0
4	0	0	1	0
5	1	1	1	0
6	1	0	1	0
7	0	1	1	0
8	1	1	0	1

Pengujian simulator I/O terhadap *mini plant* distilasi

Mini plant distilasi seperti ditunjukkan Gambar 5 merupakan serangkaian prototipe yang terdiri dari: pompa air, *photodiode* dan *infrared* sebagai sensor ketinggian cairan, *heater*, sensor suhu termokopel tipe-K, dan 4 *valve* sebagai jalan masing-masing cairan. Sistem distilasi digunakan untuk memisahkan air (C) dengan metanol (B) dan etanol (A). Metode yang digunakan untuk memisahkan ketiga cairan tersebut adalah dengan memanfaatkan suhu untuk menguapkan metanol pada suhu 64,7⁰C dan etanol pada suhu 78,29⁰C.

Saklar, *heater* dan *valve-4*

Saklar merupakan kontak yang berfungsi untuk menjalankan dan mematikan seluruh perintah dalam program python, seperti perintah berikut:

```

switch = GPIO.input(5)
if switch==0:
    GPIO.output(4,0) #switch off
    GPIO.output(6,0) #pump off
    GPIO.output(8,0) #valve 1 close
    GPIO.output(10,0) #heater off

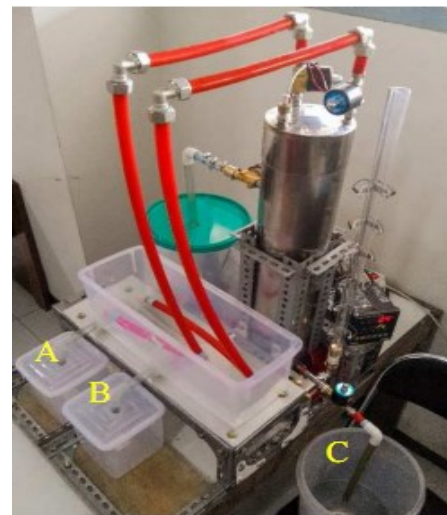
```

```

GPIO.output(12,0) #valve 2 close
GPIO.output(14,0) #valve 3 close
GPIO.output(16,1) #valve 4 open
else:
    GPIO.output(4,1) #switch on
    GPIO.output(10,1) #heater on
    GPIO.output(16,0) #valve 4 close

```

Berdasarkan perintah tersebut, bagian dari *mini plant* distilasi yang kerjanya berhubungan langsung dengan saklar adalah heater dan *valve-4*. Proses pemanasan larutan menggunakan heater berlangsung sejak saklar dinyalakan hingga saklar dimatikan. Sebaliknya, ketika saklar mati maka akan membuka *valve-4* sebagai jalan keluar air dan *valve-4* akan menutup ketika saklar hidup.



Gambar 5. *Mini plant* distilasi

Sensor *high level*, pompa dan *valve-1*

Level ketinggian cairan dideteksi menggunakan sensor inframerah dan fotodiode. Sensor *high level* cairan ini bekerja dengan metode gerbang logika NOT karena pada awalnya semua sensor potodiode akan menerima cahaya dari inframerah sehingga terbaca oleh Raspberry Pi sebagai logika 1. Sensor tersebut akan berlogika 0 apabila pelampung sebagai batas ketinggian cairan menghalangi cahaya inframerah ke fotodiode.

yang nantinya akan berfungsi untuk menentukan level ketinggian cairan. Sebagai contoh perintah yang digunakan untuk mengisi tabung distilasi dengan cairan hingga *high level 1* adalah:

```
#filling of the distillation tube
empty = GPIO.input(5)==1 and
GPIO.input(7)==1 and temp<30
full = GPIO.input(5)==1 and
GPIO.input(7)==0

if switch and empty:
    GPIO.output(6,1) #pump on
    GPIO.output(8,1) #valve 1 open
    GPIO.output(12,1) #valve 2 open
    GPIO.output(14,1) #valve 3 open

if switch and full:
    GPIO.output(6,0) #pump off
    GPIO.output(8,0) #valve 1 close
    GPIO.output(12,0) #valve 2 close
    GPIO.output(14,0) #valve 3 close
```

Perintah tersebut sekaligus akan menyalakan pompa untuk memindahkan cairan dari tempat penampungan menuju tabung distilasi dan membuka *valve-1* sebagai jalan cairan masuk ke dalam tabung distilasi.

Pembacaan skala suhu termokopel

Pembacaan sensor suhu termokopel tipe-K menggunakan max6675 untuk mengubah data bit suhu dari analog ke digital maupun digital ke analog. Max6675 akan mengubah data analog dari termokopel menjadi data digital. Data digital yang diterima oleh Raspberry Pi, kemudian diubah dan ditampilkan dalam bentuk data analog berupa derajat suhu dalam skala Celcius. Perintah yang digunakan untuk membaca sensor suhu termokopel tipe-K dengan max6675 adalah:

```
CLK=23
CS=19
SO=21
sensor=MAX6675.MAX6675 (CLK, CS, SO)
```

```
temp=sensor.readTempC()
print 'temperature:
{0:0.3F}*C'.format(temp)
time.sleep(5)
```

Valve-2, valve-3, dan kestabilan suhu

Valve-2 dan *valve-3* masing-masing digunakan untuk jalan keluar uap metanol pada suhu 64,7⁰C dan etanol pada suhu 78,29⁰C. *Valve-2* dan *valve-3* akan membuka dan menutup dengan perintah:

```
if GPIO.input(27)==0:
    setpoint1 = 65 #SP metanol

    if temp>=62 and temp<=72:
        GPIO.output(8,0)
        GPIO.output(12,1)

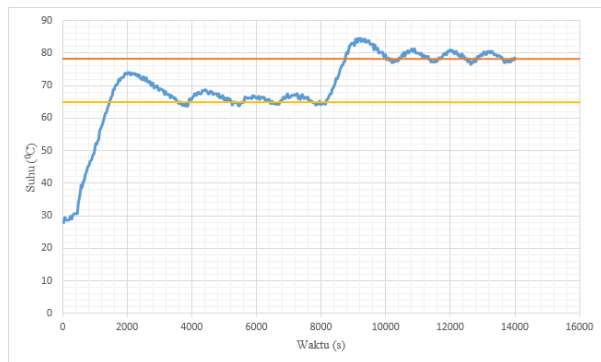
        for n in range(0,1):
            if temp>=setpoint1:
                GPIO.output(10,0)
            if temp<=setpoint1:
                GPIO.output(10,1)

    else:
        setpoint2 = 78 #SP etanol

        if temp>=75 and temp<=85:
            GPIO.output(8,0)
            GPIO.output(14,1)

            for n in range (0,1):
                if temp>=setpoint2:
                    GPIO.output(10,0)
                if temp<=setpoint2:
                    GPIO.output(10,1)
```

Hasil pengujian *mini plant* distilasi menggunakan simulator I/O dapat dilihat pada Gambar 6. Grafik 6 menggambarkan kestabilan suhu terhadap *set point* dengan menggunakan kendali on-off terhadap *heater* agar suhunya bisa stabil pada *set point*. *Heater* diatur pada 2 kali *set point*, yaitu suhu 65⁰C dan 78⁰C yang merupakan titik didih metanol dan etanol



Gambar 6. Grafik kestabilan suhu terhadap set point

KESIMPULAN

Penelitian ini memberikan kesimpulan bahwa Raspberry Pi dapat digunakan sebagai simulator I/O. Raspberry Pi dapat menjalankan simulasi input-output pada simulator I/O dan mini plant distilasi menggunakan Bahasa pemrograman Python.

DAFTAR PUSTAKA

- [1] Andrianto dan Susanto, A. (2015) *Aplikasi Pengontrol Jarak Jauh pada Lampu Rumah Berbasis Android*, Prosiding SNATIF Volume 2.
- [2] Dorf, R.C. dan Bishop, R.H. (2010) *Modern Control System*, Pearson Education, Inc., New Jersey.
- [3] Ogata, K. (2010) *Modern Control Engineering*, Pearson Education, Inc., New Jersey.
- [4] Prabowo, I.P.H., Nugroho, S. dan Utomo, D. (2014) *Penggunaan Raspberry Pi sebagai Web Server pada Rumah untuk Sistem Pengendali Lampu Jarak Jauh dan Pemantauan Suhu*. Jurnal Ilmiah Elektroteknika, Vol. 13 No. 1.
- [5] Hoyo, A., Guzman, J.L., Moreno, J.C., dan Berenguel, M. (2015) *Teaching Control Engineering Concepts using Open Source tools on a Raspberry Pi board*, International Federation of Automatic Control, Vol. 48 No. 29.
- [6] Vujovic, V. dan Maksimovic, M. (2015) *Raspberry Pi as a Sensor Web node for home automation*. Computers and Electrical Engineers, Vol. 44.
- [7] Alee, R. (2013) *Reading Data From a Digital Multimeter Using a Raspberry Pi*, Tesis, Turku University of Applied Science, Istanbul.
- [8] Pchan (2015) *Raspberry Pi 3 Model B GPIO 40 Pin Block Pinout*, <https://www.element14.com/community/docs/DOC-73950/l/raspberry-pi-3->, diakses tanggal 24 Okt 2016 jam 11:12.
- [9] Asadi, A. (2014) *Raspberry Pi for Beginners Second Revised Edition*, Imagine Publishing Ltd, Bournemouth.
- [10] Santoso, B.I. (2010) *Bahasa Pemrograman Python Di Platform GNU/LINUX*, Universitas Multimedia Nusantara, Tangerang.