

SISTEM MONITORING DAN PENGONTROLAN TEMPERATUR PADA INKUBATOR PENETAS TELUR BERBASIS PID

Sofyan Shafiudin

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : Sofyanshafiudin@gmail.com

Nur Kholis

Dosen Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : kholisunesa@yahoo.com

Abstrak

Budidaya penetasan telur unggas ayam sangat perlu diperhatikan dari segi kestabilan temperatur terutama menggunakan inkubator penetas buatan yang masih menerapkan kontrol *On/Off* di pasaran. Pengontrolan yang masih bersifat *On/Off* menghasilkan respon waktu relatif lama untuk mencapai keadaan *steady state*. Ditambah lagi cara kerja sistem yang membuat komponen mudah aus akibat lampu yang mengalami padam-hidup secara berskala. Disamping itu inkubator penetas telur yang ada dipasaran kurang cocok digunakan pada temperatur lingkungan yang gampang berubah-ubah karena dapat mempengaruhi kestabilan temperatur *plant*. Tujuan penelitian ini adalah untuk merancang inkubator penetas telur otomatis yang dapat memperbaiki hasil respon temperatur *plant* inkubator agar tetap stabil sesuai nilai temperatur *setpoint* yang dikehendaki menggunakan kontroler PID. Metode pengontrolan PID dirancang dengan mengidentifikasi *plant* menggunakan ARX (*Auto Regressive eXogenous*) Matlab yang bersifat dinamis/nonlinier untuk mendapatkan model matematis serta nilai konstanta PID yang sesuai sistem. Perancangan *hardware* untuk inkubator penetas telur berbasis PID ini menggunakan Arduino Uno R3 sebagai pusat kontroler dengan memasukkan *source* PID dan PWM untuk menjaga kestabilan temperatur *plant* yang terintegrasi dengan aktuator lampu pijar dan sensor. Dengan sensor DHT11 sebagai pembaca kondisi temperatur serta kelembaban *plant*. Hasil penelitian menunjukkan nilai konstanta PID setiap *plant* berbeda untuk *plant* 15 Watt paralel $K_p = 3.9956$, $K_i = 0.361$, $K_d = 0$, sedangkan *plant* 25 Watt paralel nilai $K_p = 5.714$, $K_i = 0.351$, $K_d = 0$. Dengan nilai konstanta PID tersebut mampu menghasilkan respon sistem yang stabil sesuai *setpoint* dengan nilai error steady state berada di kisaran tidak lebih dari 5 %, yaitu 2.7%. Dengan persentase penetasan sebesar 70-80% berhasil menetas dilingkungan ber-AC (berubah-ubah).

Kata Kunci: Inkubator Penetas Telur, Pengontrolan Temperatur, Identifikasi ARX, Kontroler PID, Arduino Uno R3

Abstract

Poultry hatching cultivation is essential to be observed in terms of temperature stability by using artificial penetration incubator which applies On/Off control. The On/Off control produces relatively long response time to reach steady state conditions. Moreover, how the system works makes the component worn out because the lamp is on-off periodically. Besides, the cultivation in the market is less suitable to be used in an environment which has fluctuating temperature because it may influence plant's temperature stability. The study aims to design automatic poultry hatching cultivation that can repair the temperature's response of plant incubator to keep stable and in line with the intended set-point temperature value by using PID controller. The method used in PID controlling is designed to identify plant using ARX (Auto Regressive eXogenous) Matlab which is dynamic/non-linear to obtain mathematical model and PID constants value that is appropriate to system. The hardware design for PID-based egg incubator uses Arduino Uno R3, as the main controller that includes PID source, and PWM, to keep plant temperature stability, which is integrated with incandescent light actuators and sensors where DHTI 1 sensor as the reader as temperature condition and plant humidity. The result of the study showed that PID constants value of each plant is different. For parallel 15 Watt plant, $K_p = 3.9956$, $K_i = 0.361$, $K_d = 0$, while for parallel 25 Watt plant, the value of $K_p = 5.714$, $K_i = 0.351$, $K_d = 0$. The PID constants value were capable to produce stable system response which is based on set-point with steady state error's value is around 5%, that is 2.7%. With hatching percentage of 70-80%, the hatching process is successful in air-conditioned environment (changeable).

Keywords: Egg Incubator, Temperature Control, ARX Identification, PID Controller, Arduino Uno R3

PENDAHULUAN

Setiap hari kegiatan manusia memerlukan kondisi temperatur yang stabil baik untuk kenyamanan mereka dalam beraktivitas maupun untuk kelancaran pekerjaan mereka seperti di bidang peternakan unggas. Budidaya penetasan telur unggas ini perlu diperhatikan dalam segi keberhasilan terlebih lagi telur ayam lokal. Penetasan telur ada 2 cara yaitu 1) secara dierami oleh induk ayam langsung dan 2) melalui mesin penetas telur dengan sistem kerja mengontrol temperatur radiasi nyala lampu penghangat. Berdasarkan hasil survei di lapangan mengenai alat penetas telur didapatkan beberapa hal, antara lain: 1) temperatur pada saat pengeraman yang sesuai berada pada 37°C ; 2) rentang waktu masa pengeraman telur selama kurang lebih 21 hari; 3) serta karakteristik desain inkubator dan karakteristik telur. Tingkat kestabilan temperatur ruang penetasan harus selalu dijaga agar mendapat hasil yang maksimal (Angga, 2016)

Perubahan temperatur lingkungan yang berubah-ubah mampu mempengaruhi temperatur pada ruang penetasan (*forced air*), terlebih lagi ditambah cara kerja sistem masih bersifat *On/Off* dengan padamnya lampu penghangat yang berakibat komponen mudah mengalami aus. Disamping itu dengan sistem tersebut masih belum presisi, terdapat eror besar, mempunyai *settling time* yang relatif lama untuk mencapai *setpoint* selanjutnya akibat ketidakstabilan temperatur pancaran radiasi lampu. Oleh karena itu muncul ide untuk mengendalikan temperatur pada inkubator agar dapat mengurangi tingkat eror, aus pada komponen serta menjaga cara kerja sistem supaya temperatur lebih stabil (*steady state*) sesuai *setpoint* dengan metode kontrol PID. Dengan metode kontrol PID maka untuk menentukan masing-masing parameter dilakukan proses identifikasi *plant* untuk mendapatkan model matematis dengan ARX (*Auto Regressive eXogenous*).

Pada penelitian ini menggunakan Arduino Uno R3, sensor DHT11 dan *driver* lampu sebagai *driver* pengendali PWM (*Pulse Width Modulation*) lampu pijar 15 Watt/25 Watt Paralel serta *driver* kipas untuk saklar kipas DC 12 Volt. Disamping itu untuk memonitoring grafik temperatur dalam sistem menggunakan GUI Matlab. Inkubator penetas telur ini menggunakan penggerak telur secara manual setiap 6 jam sekali yang bertujuan agar saat hasil penetasan embrio tidak cacat melekat pada kulit telur. Permasalahan yang didapat pada penelitian ini antara lain ialah merealisasikan inkubator penetas telur otomatis dengan metode kontrol PID, sehingga temperatur didalam sistem dapat stabil (*steady state*) dikondisi lingkungan temperatur berubah-ubah agar kualitas dan kuantitas telur yang ditetaskan optimal.

KAJIAN PUSTAKA

Inkubator Penetas Telur

Usaha budidaya ayam perlu didukung dengan sarana yang memadai. Sarana yang termasuk pendukung dalam proses penetasan telur adalah dengan menggunakan mesin penetas telur buatan. Mesin penetas telur adalah alat yang digunakan untuk meringankan beban induk ayam dalam proses mengerami telur dengan prinsip kerja mengontrol temperatur dari pancaran lampu penghangat.

Mesin penetas buatan mempunyai beberapa poin yang baik dalam sarana penetasan telur, yaitu:

1. Temperatur
2. Kelembaban Udara (*Humidity*)
3. Ventilasi
4. Pemutaran Rak Telur
5. Kebersihan
6. Bahan Desain Mesin (Jaya, 2016).

Sensor DHT11

Sensor DHT11 adalah sensor temperatur dan kelembaban, yang memiliki keluaran sinyal digital dikalibrasi dengan sensor suhu dan kelembaban yang kompleks. Sensor DHT11 mempunyai akurasi $\pm 2^{\circ}\text{C}$ dan $\pm 5\%$ RH. Berikut Gambar 1. Sensor DHT11 secara fisik.



Gambar 1. Sensor DHT11
(Sumber: Datasheet DHT11)

Pembacaan data keluaran sensor DHT11 ditampilkan pada grafik GUI Matlab yang terhubung dengan Arduino Uno R3.

Arduino Uno R3

Arduino Uno R3 adalah *prototyping platform* sebuah paket berupa papan (*board*) elektronik (*hardware*) dan lingkungan pengembangan (*software*) yang memanfaatkan kemampuan mikrokontroler jenis tertentu. Mikrokontroler yang digunakan pada Arduino Uno adalah jenis Atmel seri ATmega 328 (Wardana, 2015). Berikut Gambar 2. Arduino Uno R3.

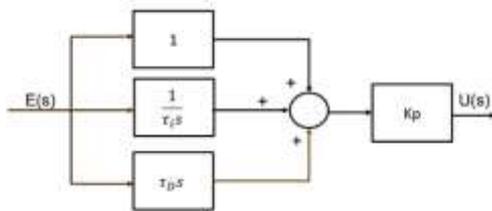


Gambar 2. Arduino UNO
(Sumber: www.arduino.cc)

Prototyping platform ini berkomunikasi secara serial dengan perangkat kabel USB yang digunakan untuk *upload* program dari *software* IDE Arduino. Penelitian ini menggunakan pin-pin data *analog* dan digital. Untuk keperluan PWM (*Pulse Width Modulation*) terdapat pada pin 3,5,6,9,10 dan 11.

Kontroler

Dalam suatu sistem kendali otomatis dikenal adanya beberapa aksi pengendalian, diantaranya aksi kendali *On/Off*, aksi kendali proporsional (P), aksi kendali integral (I) dan aksi kendali differensial (D). Masing-masing aksi kendali tersebut mempunyai keunggulan tertentu. Aksi kendali *On/Off* mempunyai keunggulan waktu naik yang cepat, aksi kendali proporsional mempunyai keunggulan waktu naik yang cepat dan stabil, aksi kendali integral mempunyai keunggulan untuk memperkecil kesalahan, dan aksi kendali differensial memiliki keunggulan meredam kekurangan tanggapan atau kelebihan tanggapan. Untuk mendapatkan suatu sistem kendali dengan hasil pengendalian yang memiliki waktu naik yang cepat, kesalahan yang kecil dan kestabilan yang baik, dapat dilakukan dengan menggabungkan ketiga aksi kendali tersebut menjadi aksi kendali PID (Gunterus, 1994). Berikut Gambar 3. Diagram blok kontroler PID.



Gambar 3. Diagram Blok Kontroler PID (Sumber : Ogata, 1985)

Berikut ini merupakan penjelasan singkat dari parameter-parameter yang ada pada kontrol PID:

- a) Kontrol Proporsional

Proporsional adalah hasil dari perkalian antara konstanta proporsional dengan nilai error nya. Perubahan yang terjadi pada nilai *input* akan menyebabkan sinyal *Output* sebesar konstanta pengalinya. Pengaruh kontrol Proporsional pada suatu sistem, yaitu:

 - a. Menambah atau mengurangi kestabilan.
 - b. Memperbaiki *transient respon* khususnya: *rise time* dan *settling time*.
 - c. Mengurangi (bukan menghilangkan) *error steady state*.
- b) Kontrol Integral

Kontrol Integral memiliki karakteristik mengurangi waktu naik, menambah *overshoot*

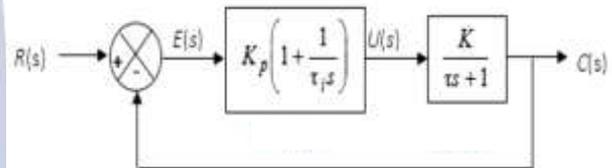
dan waktu turun, serta menghilangkan kesalahan keadaan tunak. . Kontrol integral membantu menaikkan respon sehingga menghasilkan *output* yang diinginkan

c) Kontrol Derivatif

Kontrol derivatif tidak akan pernah digunakan sendirian, karena kontroler ini hanya akan aktif pada periode peralihan. Pada periode peralihan, kontrol derivatif menyebabkan adanya redaman pada sistem sehingga lebih memperkecil lonjakan. Seperti pada kontrol proporsional, kontrol derivatif juga tidak dapat menghilangkan *offset* (Ogata, 1985).

Kontroler PI

Sebuah kontroler PI untuk kondisi *plant* orde satu membentuk sistem umpan balik dengan masukan R(s) dan keluaran C(s), diagram blok kontroler PI *plant* orde satu dapat digambarkan Pada Gambar 4. sebagai berikut.



Gambar 4. Diagram Blok Kontroler PI *Plant* Orde Satu (Sumber : Ogata, 1985)

Dari Gambar 4 diatas dapat dirumuskan untuk perbandingan C(s) dan R(s) sebagai berikut :

$$\frac{C(s)}{R(s)} = \frac{Kp(1 + \frac{1}{\tau_i s})(\frac{K}{\tau_s s + 1})}{(Kp(1 + \frac{1}{\tau_i s})(\frac{K}{\tau_s s + 1}) + 1)} \tag{1}$$

Dari persamaan sistem, terdapat beberapa pemilihan nilai τ_i , jika nilai $\tau_i = \tau$ maka hasil desain adalah orde satu. Sedangkan jika nilai $\tau_i \neq \tau$ maka hasil desain adalah orde dua.

$$\tau_i = \tau \tag{2}$$

$$\frac{C(s)}{R(s)} = \frac{KpK}{\tau_i s + KpK} = \frac{1}{\frac{\tau_i}{KpK}s + 1} \tag{3}$$

$$\tau * = \frac{\tau_i}{KpK} \tag{4}$$

$$PI = Kp(1 + \frac{1}{\tau_i s}) \tag{5}$$

Keterangan :

C(s) : Keluaran

R(s) : Masukan

K : *Gain Overall*

τ_i : Konstanta Waktu Tertentu

τ_s : Konstanta Waktu

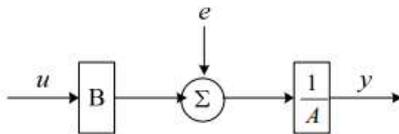
PI : Proporsional Integral (Ogata, 1985)

Pengontrolan PI dilakukan apabila hasil respon *plant* tidak mempunyai *overshoot* yang tergolong pada sistem orde satu (*zero off-set* %Ess = 0%). Realisasi parameter

kontroler masing-masing akan diprogram di mikrokontroler Arduino Uno R3 sebagai pusat kendali sistem.

Identifikasi Sistem ARX

Identifikasi sistem adalah metodologi untuk membangun model matematika dari suatu sistem dinamis berdasarkan perhitungan dari sinyal *input* dan *output* sistem. Secara konsep, identifikasi sistem merupakan pemodelan sistem dinamis dari data yang dihasilkan dalam eksperimen (Ljung L., 2011). Berikut Gambar 5. Diagram blok struktur model ARX



Gambar 5. Diagram Blok Struktur Model ARX
(Sumber : Ljung L., 2011)

AR-Auto Regressive yaitu *output* yang dihasilkan saat ini (current) berhubungan dengan nilai *output* sebelumnya (previous). X-eXogenous *Input* yaitu sistem tidak hanya bergantung pada input saat ini (current) tetapi juga berdasarkan rekam jejak (history) *input*.

Bentuk umum model ARX :

$$A(q)y(t) = B(q)u(t) + e(t) \quad (6)$$

Keterangan :

A(q) : Polinomial A

B(q) : Polinomial B

y(t) : *Output*

u(t) : *Input*

e(t) : eror

PWM (Pulse Width Modulation)

PWM adalah singkatan dari *Pulse Width Modulation*, yaitu teknik yang biasa digunakan untuk mengontrol daya ke perangkat listrik, dibuat praktis dengan *switch* daya elektronik. Metode PWM merupakan metode untuk pengaturan pemanas dengan cara mengatur persentase lebar pulsa *high* terhadap periode dari suatu sinyal persegi dalam bentuk tegangan periodik yang diberikan ke *driver* sebagai pemanas.

METODE PENELITIAN

Pendekatan Penelitian

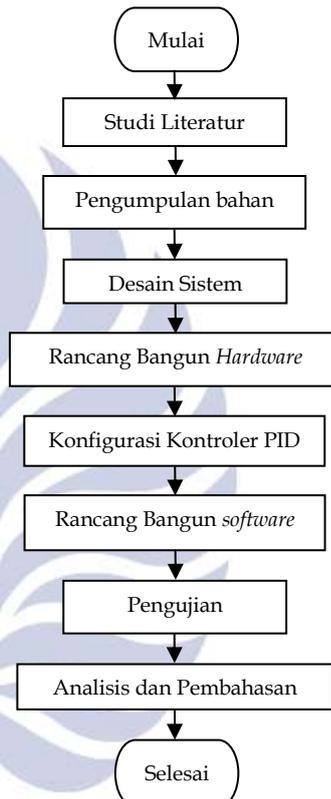
Pada penelitian ini menggunakan *software* Arduino IDE dan Matlab 2015a untuk simulasi serta menjalankan sistem PID yang akan digunakan dan merancang bangun *plant* yang akan menghasilkan respon temperatur di dalam inkubator penetas telur.

Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Laboratorium Fisika Teknik Jurusan Teknik Elektro Universitas Negeri Surabaya dan waktu pelaksanaannya dilakukan pada semester genap 2016/2017.

Rancangan Penelitian

Rancangan penelitian ini bertujuan untuk menjawab permasalahan dalam rangka merumuskan kesimpulan, seperti dijelaskan pada Gambar 6. diagram alir sebagai berikut:



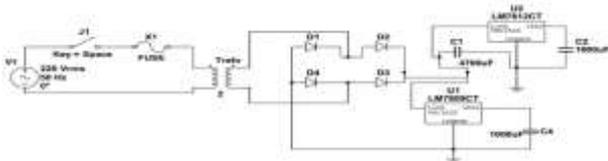
Gambar 6. Diagram Alir Tahapan Penelitian

Rancang Bangun Hardware

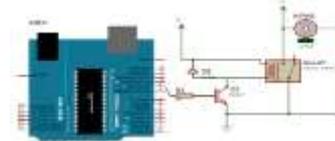
Rancang bangun *hardware* meliputi beberapa rancangan blok yaitu; blok catu daya, blok sensor, blok *driver* lampu, blok *driver* kipas, dan blok LCD serta blok *plant* inkubator penetas telur.

Blok Catu Daya

Catu Daya (*Power Supply*) pada sistem ini memiliki peranan penting sebagai sumber tegangan DC pada sistem. Dengan *input* tegangan jala-jala PLN sebagai *input Power Supply* memiliki 2 buah *output* yang masing-masing 9 Volt dan 12 Volt. Tegangan 9 Volt digunakan untuk Arduino Uno R3 sedangkan tegangan 12 Volt untuk kebutuhan *supply driver* kipas. Berikut Gambar 7. Skema rangkaian catu daya.



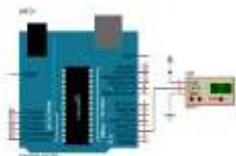
Gambar 7. Catu Daya



Gambar 10. Driver Kipas

Blok Sensor DHT11

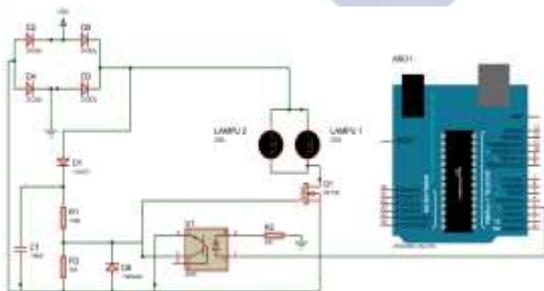
Sensor DHT 11 adalah sebuah sensor yang sudah terkalibrasi dengan tingkat akurasi untuk temperatur ± 2 °C dan kelembaban ± 5 % RH yang dihubungkan ke pin 2 Arduino Uno dengan kabel 40 cm untuk diolah dan diproses sebagai masukan data. Berikut Gambar 8. skema sensor DHT11.



Gambar 8. Sensor DHT11

Blok Driver Lampu

Driver lampu mempunyai prinsip kerja menggunakan masukan tegangan AC 220 Volt yang disearahkan melewati beberapa dioda yang disusun secara jembatan *wheatstone* yang menghasilkan tegangan DC dengan daya serta arus yang besar. Terdapat komponen Optocoupler sebagai saklar penghubung dengan MOSFET. Driver lampu mempunyai peranan sebagai PWM aktuator yang mendapat sinyal dari Arduino. Berikut Gambar 9. skema rangkaian driver lampu.



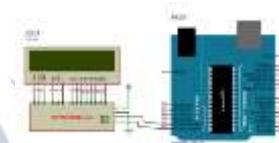
Gambar 9. Driver Lampu

Blok Driver Kipas

Driver kipas ini mempunyai peranan untuk menjalankan kipas pada saat proses penetasan telur agar temperatur didalam inkubator merata serta membuang udara panas apabila temperatur melebihi nilai *setpoint*. Berikut Gambar 10. Skema driver kipas.

Blok LCD 16X2

Tampilan LCD difungsikan untuk menampilkan nilai temperatur dan kelembaban didalam inkubator penetas telur. LCD 16X2 ini menggunakan modul I2C LCD untuk meminimalisir jumlah kabel yang terhubung pada pin Arduino. Berikut Gambar 11. Skema LCD 16X2.



Gambar 11. Skema LCD 16X2

Blok Plant Inkubator Penetas Telur

Pada skripsi ini menggunakan *plant* terbuat dari kayu jenis *Triplek*. Jenis kayu ini ideal digunakan sebagai bahan baku inkubator penetas telur. Dimensi inkubator 30 x 40 x 32 cm dengan ketinggian 30 cm dari tanah. Berikut Gambar 12. Desain *plant* inkubator penetas telur.



Gambar 12. Plant Inkubator Penetas Telur

Konfigurasi Kontrol PID

Sebelum mendesain konfigurasi kontrol PID yang sesuai dengan *plant*, langkah pertama adalah merancang sistem keseluruhan dari inkubator penetas telur kemudian mengidentifikasi sistem dengan metode ARX (AR-Auto Regressive X-eXogenous) yang berhubungan dengan nilai masukan (*input*) tangga PWM dan keluaran (*output*) berupa temperatur yang diukur untuk mendapatkan model matematis suatu sistem dinamis. Objek data diciptakan menggunakan Matlab *system identification toolbox* dengan Simulink ARX. Berikut Gambar 13. Simulink yang digunakan untuk identifikasi sistem untuk masing-masing *plant*.



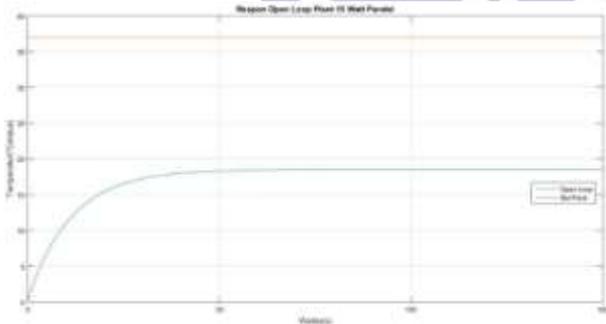
Gambar 13. Simulink Identifikasi Sistem Lampu 15 Watt Paralel dan 25 Watt Paralel

Hasil *transfer function* yang diperoleh dari Simulink tersebut masih berupa *transfer function* diskrit. Untuk mendapatkan *transfer function continuous* (*Transformasi Laplace*) menggunakan syntax `sysc = d2c(sysd)`. Berikut masing-masing *Transfer Function continuous plant* 15 Watt Paralel dan 25 Watt Paralel:

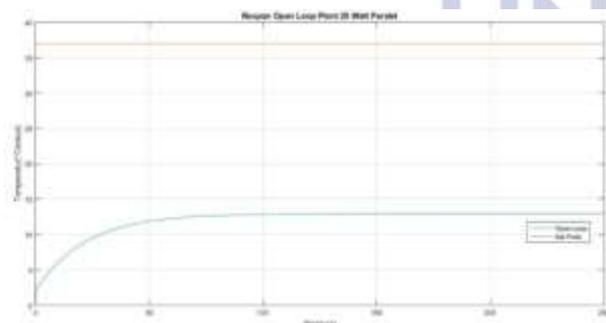
$$\text{Plant 15 Watt} = \frac{0.496s+1.658}{s^2+37.6s+3.313}$$

$$\text{Plant 25 Watt} = \frac{1.711s+0.4965}{s^2+31.02s+1.419}$$

Pengujian respon *open loop* model matematis *plant* menggunakan sinyal uji *step* dengan nilai *step* 37 sesuai temperatur yang diinginkan. Berikut Gambar 14. Dan Gambar 15. hasil respon *open loop* masing-masing *plant*:



Gambar 14. Respon *Open Loop Plant* 15 Watt Paralel



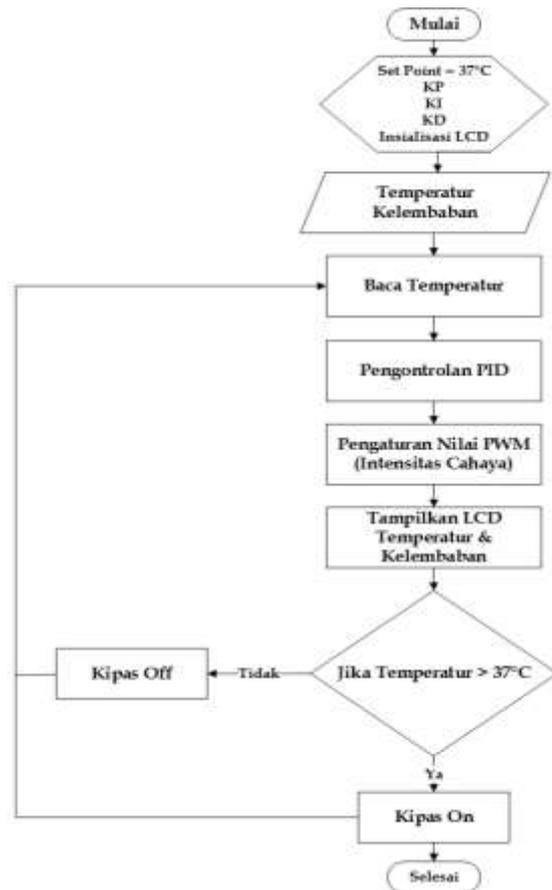
Gambar 15 Respon *Open Loop Plant* 25 Watt Paralel

Hasil dari Gambar 14. dan Gambar 15. adalah karakteristik grafik respon sistem orde satu. Nilai keluaran *steady state* dari masing-masing sistem masih belum mencapai nilai *setpoint* 37 yaitu 18.52°C untuk *plant* 15 Watt dan 12.95°C untuk 25 Watt. Oleh sebab itu dibutuhkan sebuah kontrol PID dengan metode pendekatan orde satu untuk mencapai nilai *steady state*

yang diinginkan. Dengan metode pendekatan sistem orde satu maka ditentukan permisalan nilai $\tau = \tau_i$ untuk menentukan nilai K_p, K_i, K_d .

Rancang Bangun Software

Software dalam penelitian ini menggunakan IDE Arduino 1.6.12 dan Matlab 2015a. Perancangan program yang digunakan untuk sistem penetas telur PID dimulai dengan membuat *flowchart* program. *Flowchart* program keseluruhan dapat dilihat pada Gambar 16. berikut.



Gambar 16. *Flowchart* Program Sistem

HASIL DAN PEMBAHASAN

Pengujian dan analisis penelitian meliputi; 1) analisis dan pengujian rancangan rangkaian sistem kontrol *plant* untuk PID, 2) Pengaplikasian konfigurasi kontrol PID pada *plant*, 3) Hasil pengujian kontrol PID.

1) Pengujian Rancangan Rangkaian

Pengujian rancangan rangkaian meliputi catu daya, sensor DHT11, *driver* lampu, serta *driver* kipas. Berikut tiap blok pengujian rangkaian:

Catu Daya

Catu Daya (*Power Supply*) pada sistem ini memiliki peranan penting sebagai sumber tegangan DC pada sistem. Berdasarkan alat pengukuran *input Power Supply* sebesar 224,6 Volt AC. Untuk

tegangan *output* 1 yang terukur yaitu 9,06 Volt sebagai sumber Arduino, sedangkan tegangan *output* 2 yang terukur sebesar 12,20 Volt sebagai sumber kipas DC.

Sensor DHT11

Pada penelitian Skripsi ini membandingkan data pembacaan temperatur DHT 11 dengan instrumentasi temperatur lain yang memiliki tingkat akurasi temperatur $\pm 3\%$ dengan resolusi 1°C yaitu *Clamp Meter* dan *Digital Thermometer* dengan akurasi $\pm 1^\circ\text{C}$. Pengujian dengan membandingkan pembacaan sensor temperatur jenis lain bertujuan untuk melihat tingkat akurasi pada sensor DHT11. Berikut Tabel Perbandingan akurasi sensor:

Tabel 1. Pengujian Sensor DHT 11 Dan Clamp Meter

No.	Waktu	Sensor DHT 11		Clamp Meter	Selisih
		$^\circ\text{C}$	%RH	$^\circ\text{C}$	
1.	Pagi (07.00-08.00)	22	61	24	2
2.	Siang(13.00-14.00)	24	54	26	2
3.	Malam(22.00-23.00)	21	62	22	1
Rata-rata					1.7

Tabel 2. Pengujian Sensor DHT 11 Dan Digital Thermometer

No.	Waktu	Sensor DHT 11		Digital Thermometer	Selisih
		$^\circ\text{C}$	%RH	$^\circ\text{C}$	
1.	Pagi (07.00-08.00)	22	61	25.3	3.3
2.	Siang(13.00-14.00)	24	54	27.9	3.9
3.	Malam(22.00-23.00)	21	62	24.7	3.7
Rata-rata					3.6

Driver Lampu

Driver lampu ini mempunyai prinsip kerja PWM untuk memberikan *supply* tegangan ke lampu pijar. Dengan indikator nilai PWM 0-255 dari kontrol Arduino yang akan memberikan respon tingkat pencahayaan lampu. Analisis perhitungan spesifikasi komponen *driver* lampu:

Komponen Dioda Zener yang digunakan adalah 15 Volt 1 Watt.

$$P = Vzener \cdot I \quad (7)$$

$$I = \frac{P}{Vzener} \quad (8)$$

$$I = \frac{1}{15} = 0.07 \text{ A}$$

Tegangan setelah penyearah jembatan *wheatstone* :

$$Vm = VAC \times \sqrt{2} \quad (9)$$

$$Vm = 224.6 \times \sqrt{2} = 317.63 \text{ Volt}$$

$$VDC = \frac{2Vm}{\pi} \quad (10)$$

$$VDC = \frac{2 \times 317.63}{3.14} = 202.31 \text{ Volt}$$

Arus yang mengalir di R1

$$VRI = Vm - Vzener \quad (11)$$

$$VRI = 317.63 - 15 = 312.63 \text{ Volt}$$

$$IRI = \frac{VRI}{R1}$$

$$IRI = \frac{312.63}{100K} = 3.1263 \text{ mA}$$

Daya yang dihasilkan adalah

$$P = V \times I \quad (12)$$

$$P = 312.63 \times 3.126 \times 10^{-3} = 0.977 \text{ Watt} \approx 1 \text{ Watt}$$

Jadi untuk pemakaian komponen R1 adalah $100K\Omega$ dengan daya 2 Watt agar lebih aman. Spesifikasi diode zener Vzener 15 Volt, untuk mendapat nilai R3 :

$$R3 = \frac{Vzener}{IR1}$$

$$R3 = \frac{15}{3.1263} = 4.798 \Omega \approx 4k7 \Omega$$

Jadi untuk pemakaian komponen R3 adalah $6K8 \Omega$ dengan daya 1 Watt agar lebih aman.

Driver Kipas

Cara kerja dari rangkaian ini ketika ada sinyal *input* berlogika 1 dari Arduino akan mengaktifkan transistor kondisi *On* yang membuat relay aktif *normally open* menjadi *normally close*. Dengan aktifnya relay *normally close* maka kipas akan aktif sampai dengan batas temperatur yang telah ditentukan. *Driver* kipas ini akan *Off* apabila mendapat input berlogika 0 saat temperatur terpenuhi yang membuat transistor akan posisi *Off*. Disamping itu relay akan menjadi *Off* ke posisi *normally open*.

2) Pengaplikasian Konfigurasi Kontrol PID

Berdasarkan hasil dari respon *open loop* masing-masing *plant* mengindikasikan jenis respon termasuk karakteristik sistem orde satu karena tidak ada *Overshoot*. Sehingga metode pengontrolan menggunakan kontroler PI (Proporsional Integral) dengan nilai $Kd = 0$, $\tau = \tau_i$ agar nilai respon keluaran optimal sesuai dengan nilai masukan.

Untuk *Plant* 15 Watt Paralel:

$$K = \frac{Y_{ss}}{X_{ss}} \quad (13)$$

$$K = \frac{18.52}{37} = 0.50054$$

$$\tau = 11.046 \text{ s dipercepat } 2x$$

$$\tau^* = 5.523 \text{ s}$$

Untuk mendapatkan nilai K_p mengacu pada persamaan (4).

$$K_p = \frac{\tau_i}{K\tau^*} = \frac{11.046}{0.50054 \times 5.523} = 3.9956$$

$$K_i = \frac{K_p}{\tau_i} = \frac{3.9956}{11.046} = 0.361$$

Validasi kontrol PI 15 Watt untuk mengetahui hasil respon setelah dikontrol.

$$PI = K_p \left(1 + \frac{1}{\tau_i s} \right)$$

$$PI = 3.9956 \left(1 + \frac{1}{11.046}\right) = 4.357$$

Untuk *Plant* 25 Watt Paralel:

$$K = \frac{12.95}{37} = 0.35$$

$$\tau = 18.1140 \text{ s dipercepat } 2x$$

$$\tau^* = 9.057 \text{ s}$$

Untuk mendapatkan nilai K_p mengacu pada persamaan (4).

$$K_p = \frac{\tau_i}{K\tau^*} = \frac{18.1140}{0.35 \times 9.057} = 5.714$$

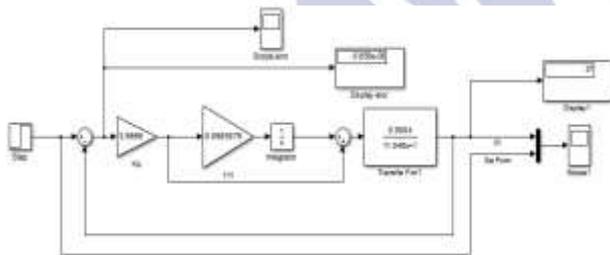
$$K_i = \frac{K_p}{\tau_i} = \frac{5.714}{18.1140} = 0.315$$

Validasi kontrol PI 25 Watt untuk mengetahui hasil respon setelah dikontrol.

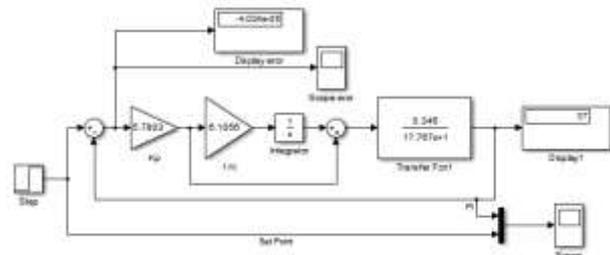
$$PI = K_p \left(1 + \frac{1}{\tau_i s}\right)$$

$$PI = 5.714 \left(1 + \frac{1}{18.1140}\right) = 6.0294$$

Berikut Gambar 17. dan Gambar 18. simulasi masing-masing *plant* menggunakan kontroler PI.



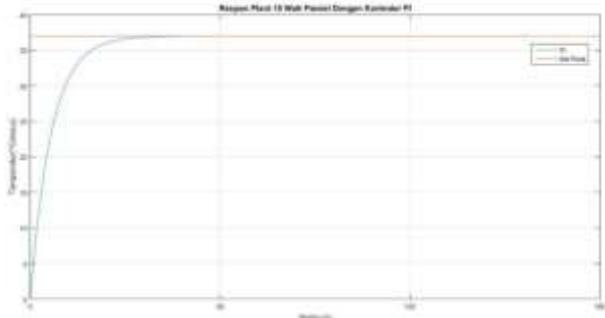
Gambar 17. Simulasi Kontroler PI pada *Plant* 15 Watt



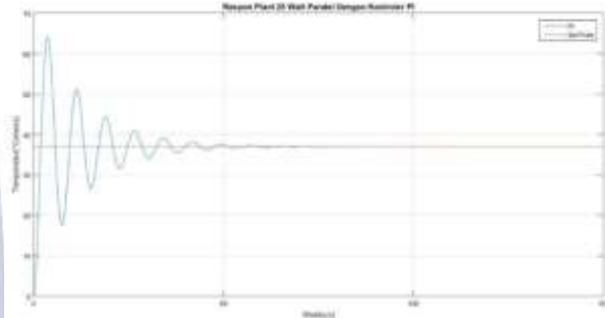
Gambar 18. Simulasi Kontroler PI pada *Plant* 25 Watt

Setiap masing-masing *plant* mempunyai nilai konstanta PID yang berbeda untuk mencapai nilai stabil. Pada *plant* dengan lampu 15 Watt paralel telah diidentifikasi respon sistem sesuai dengan sistem orde satu yang pada umumnya menggunakan kontroler PID nilai konstanta $K_p = 3.9956$, $K_i = 0.361$, $K_d = 0$. Sedangkan pada *plant* dengan lampu 25 Watt paralel telah diidentifikasi respon sistem nilai konstanta $K_p = 5.714$, $K_i = 0.315$, $K_d = 0$. Setelah itu nilai PID diimplementasikan ke dalam program Arduino. Pengujian masing-masing *plant* dengan kontroler PI memiliki hasil respon temperatur yang sesuai nilai masukan *step setpoint* 37°C. Berikut Gambar

19. dan Gambar 20. Respon *Plant* masing-masing dengan kontroler PI.



Gambar 19. Respon *Plant* 15 Watt dengan Kontroler PI



Gambar 20. Respon *Plant* 25 Watt dengan Kontroler PI

3) Hasil Pengujian Kontrol PID

Monitoring *Train Grafik Plant*

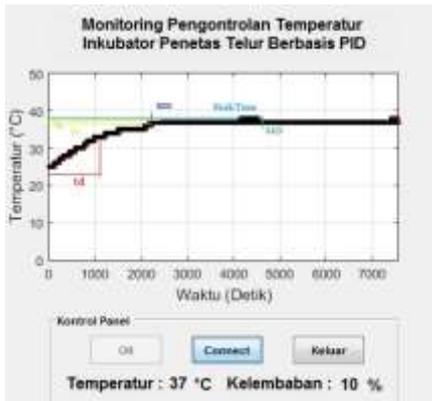
Dalam proses monitoring *train* grafik kontrol PID untuk masing-masing *plant* mempunyai respon yang berbeda sesuai dengan hasil nilai konstanta PID yang telah dianalisis. Pengujian dilakukan dengan temperatur awal menyesuaikan temperatur lingkungan terhadap sistem antara 25-26° C dengan memberikan nilai *setpoint* 37°C.

Untuk *plant* 15 Watt Paralel pada inkubator penetas telur mempunyai nilai konstanta $K_p = 3.9956$, $K_i = 0.361$, $K_d = 0$ menghasilkan respon waktu naik (*rise time*) sebesar 2178 detik untuk mencapai *steady state* (*settling time*) pertama kali serta waktu tunda (*delay time*) sebesar 1089 detik. Pada pengujian kontrol PID *plant* 15 Watt ini mempunyai nilai *error steady state* sebesar 0% yang mengindikasikan pengontrolan yang berhasil dari toleransi 5%. Dalam pengujian ini mendapat hasil nilai persentasi *Overshoot Maximum* 2.7 % dengan nilai puncak 38°C. Waktu untuk mencapai nilai puncak dari respon (*peak time*) sebesar 4100 detik.

Plant 25 Watt paralel pada inkubator penetas telur mempunyai nilai konstanta $K_p = 5.714$, $K_i = 0.351$, $K_d = 0$ menghasilkan respon waktu naik (*rise time*) sebesar 1244 detik untuk mencapai nilai *setpoint* pertama kali serta waktu tunda (*delay time*) sebesar 622 detik. Dalam pengujian ini mendapat hasil nilai persentasi *Overshoot Maximum* 2.7 % dengan nilai puncak 38°C. Waktu

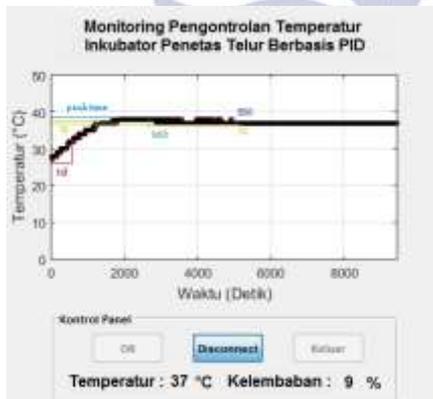
untuk mencapai nilai puncak dari respon (*peak time*) sebesar 1740 detik. Untuk mencapai nilai *steady state* setelah nilai puncak mempunyai waktu tunak (*settling time*) sebesar 5010 detik. Pada pengujian kontrol PID *plant 25 Watt* ini mempunyai nilai *error steady state* sebesar 2.7% yang mengindikasikan pengontrolan yang berhasil dari toleransi 5%.

Berikut Gambar 21. analisis grafik respon pada tampilan GUI *plant 15 Watt* menggunakan kontroler PID.



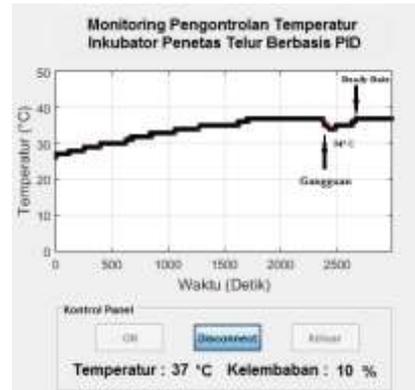
Gambar 21. Analisis Grafik Respon Pada Tampilan GUI *plant 15 Watt* Menggunakan Kontroler PID

Berikut Gambar 22. Analisis grafik respon pada tampilan GUI *plant 25 Watt* Menggunakan Kontroler PID.



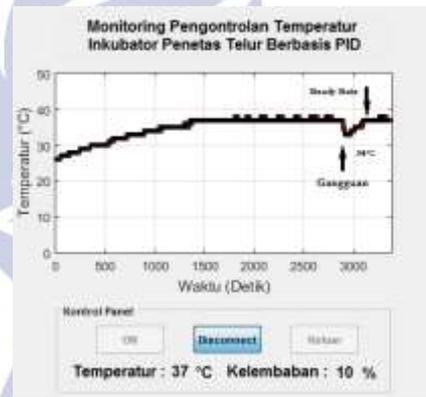
Gambar 22. Analisis Grafik Respon Pada Tampilan GUI *Plant 25 Watt* Menggunakan Kontroler PID

Untuk Pengujian *plant 15 watt* terhadap gangguan diberikan pada waktu 2368 detik. Ketika diberikan gangguan berupa terbukanya pintu inkubator selama 20 detik, grafik respon menunjukkan sistem mengalami ketidakstabilan, temperatur menurun 3°C dari *setpoint*. Dibutuhkan waktu tunak (*settling time*) 215 detik untuk mencapai *setpoint* bertahan pada kondisi *steady state*. Hal tersebut mengindikasikan cukup baik respon sistem dalam menanggapi gangguan. Berikut Gambar 23. Analisis grafik respon *plant 15 Watt* terhadap gangguan.



Gambar 2.3 . Analisis Grafik Respon *Plant 15 Watt* Terhadap Gangguan

Sedangkan untuk *plant 25 watt* diberikan gangguan pada waktu 2891 detik. Ketika diberi gangguan dengan terbukanya pintu inkubator selama 20 detik grafik respon mengalami ketidakstabilan, temperatur menurun hingga 3°C mencapai 34°C. Dibutuhkan waktu tunak (*settling time*) 150 detik untuk mencapai nilai *setpoint* bertahan pada *steady state*. Hal tersebut mengindikasikan bahwa respon *plant 25 watt* sistem lebih cepat mencapai *steady state* daripada respon sistem *plant 15 watt* dalam menanggapi gangguan. Berikut Gambar 24. Analisis grafik respon *plant 25 Watt* terhadap gangguan.



Gambar 24. Analisis Grafik Respon *Plant 15 Watt* Terhadap Gangguan

Hasil Penetasan

Berdasarkan penelitian yang telah dilaksanakan tanggal 7 April sampai 2 Mei 2017 bahwa perlakuan kestabilan temperature berpengaruh nyata terhadap hasil tetas. Dengan pengontrolan temperatur yang stabil mempunyai tingkat waktu temperatur lebih cepat dalam hal penetasan. Hasil dari *plant* lampu 15 Watt dengan kontroler PID yang mempunyai kestabilan temperatur terjaga sesuai *setpoint* mampu menetas telur dengan rentang waktu 19 hari. Sedangkan hasil dari *plant 25 Watt* yang mempunyai kestabilan temperatur naik turun dari *setpoint* mampu menetas telur dengan rentang waktu 21 hari. Berikut Tabel 2 dan Tabel 3 Hasil pengujian penetasan masing-masing *plant*.

Tabel 2 Hasil Pengujian Penetasan *Plant* 15 Watt

<i>Plant</i> 15 Watt Paralel (10 telur)			
Telur Menetas			Telur Tidak Menetas
70%			3 Telur (30%)
Normal	Cacat	Mati	
7	0	0	

Tabel 3. Hasil Pengujian Penetasan *Plant* 25 Watt

<i>Plant</i> 25 Watt Paralel (10 telur)			
Telur Menetas			Telur Tidak Menetas
80%			2 Telur (20%)
Normal	Cacat	Mati	
7	0	1	

PENUTUP

Simpulan

Berdasarkan penelitian yang dilakukan, maka didapatkan simpulan bahwa sistem monitoring dan pengontrolan temperatur pada inkubator penetas telur untuk kontroler PID berhasil dibuat terintegrasi secara otomatis dengan metode PWM pada lampu pijar. Untuk mengaplikasikan kontrol PID pada *plant* inkubator penetas telur adalah dengan metode identifikasi sistem ARX dengan nilai konstanta $K_p = 3.9956$, $K_i = 0.361$, $K_d = 0$ untuk *plant* 15 Watt paralel dan nilai konstanta $K_p = 5.714$, $K_i = 0.351$, $K_d = 0$ untuk *plant* 25 Watt paralel. Dengan konstanta tersebut respon sistem kontrol mampu menstabilkan temperatur inkubator penetasan telur. Pada pengujian *plant* dengan kontrol PID, *plant* 15 Watt menghasilkan respon lebih lambat untuk respon waktu naik (*rise time*) $\tau_r = 2178$ detik, waktu tunda (*delay time*) $\tau_d = 1089$ detik. Sedangkan respon *plant* 25 Watt waktu naik (*rise time*) $\tau_r = 1244$ detik, waktu tunda (*delay time*) $\tau_d = 622$ detik. Tetapi untuk respon waktu tunak (stabil) *plant* 15 Watt mampu lebih cepat $\tau_s = 2178$ detik daripada waktu tunak *plant* 25 Watt $\tau_s = 5010$ detik. Nilai *error steady state* yang didapat sebesar 2.7% yang mengindikasikan pengontrolan temperatur berjalan baik di kondisi lingkungan berubah-ubah dengan persentase menetas 70-80% telur.

Saran

Berdasarkan simpulan diatas ada beberapa saran yang dapat dilakukan untuk pengembangan sistem agar hasil lebih maksimal yaitu menggunakan sensor yang lebih akurat dan presisi seperti tipe SHT 11, menerapkan sistem pengontrolan kelembaban udara pada *plant*, menerapkan sistem rak putar secara otomatis, menerapkan penggunaan UPS sebagai *backup* daya

listrik apabila listrik PLN padam, menerapkan sistem monitoring secara online atau Internet Of Thing (IoT), serta metode pengontrolan temperatur masih dapat dikembangkan dengan mengubah kontroler jenis kontrol adaptif, fuzzy logic, dan jaringan saraf tiruan (JST) agar dapat mengurangi nilai *error steady state*.

DAFTAR PUSTAKA

- Gunterus, Frans.1994. *Falsafah Dasar : Sistem Pengendalian Proses*. Jakarta: PT. Elex Media Komputindo
- Hastono, Djoko Tri. 2009. *Sistem Pengendali dan Pengukur Suhu pada Mesin Penetas Telur Berbasis Mikrokontroler AT89S51*. Semarang: Universitas Negeri Semarang.
- Jaya, Mitra. 2016. *Buku Panduan Praktis C-500 & C-1000*. Malang: Mitra Jaya Company
- Ljung L., 2011. *System Identification Toolbox™ User's Guide*. Natick: The MathWorks, Inc.
- Ogata, Katsuhiko. 1985. *Teknik Kontrol Automatik jilid 1*. Terjemahan Edi Laksono. Jakarta: Erlangga.
- Wardana, I Nyoman Kusuma. 2015. *Teknik Antarmuka MATLAB dan Arduino*. Denpasar: Vaikutha International Publication.
- Web.<https://www.arduino.cc> Diakses 24 Januari 2017.