

## PERANCANGAN MODUL KONTROL TEKANAN RUANGAN BERBASIS ARDUINO PADA SISTEM VENTILASI RSG-GAS

### **DESIGN OF ROOM PRESSURE CONTROL MODULE BASED ARDUINO IN RSG-GAS VENTILATION SYSTEM**

Jazid Uchti Namir <sup>1</sup>, Kiswanto <sup>1</sup>, Cahyana <sup>1</sup>, Ferdy Triyuandika <sup>1</sup>  
<sup>1</sup> Pusat Reaktor Serba Guna-BATAN, Kawasan Puspipstek Serpong  
Email: [jazid-uchti@batan.go.id](mailto:jazid-uchti@batan.go.id).

Diterima: 21 Oktober 2021, diperbaiki : 28 Oktober 2021, disetujui : 28 Oktober 2021

#### ABSTRAK

**PERANCANGAN MODUL KONTROL TEKANAN RUANGAN BERBASIS ARDUINO PADA SISTEM VENTILASI RSG-GAS.** Salah satu fungsi dari sistem ventilasi RSG-GAS adalah mengatur tekanan negatif setiap ruang di dalam gedung RSG-GAS. Tekanan negatif diatur menggunakan modul kontrol tekanan dengan input sinyal dari transducer untuk menggerakkan aktuator damper. Tujuan dari tulisan ini adalah merancang modul kontrol tekanan ruangan agar dapat menggantikan modul kontrol tekanan lama yang masih memiliki tampilan analog dan barang diskontinu. Perancangan ini terdiri atas perancangan hardware dan software. Perancangan hardware menggunakan komponen-komponen yaitu modul HW 685, Arduino, Display OLED 0,96 inch, relay shield, potensiometer, switch auto/manual dan push button. Alur proses kendali tekanan yaitu sinyal dari transducer diterima modul HW 685. selanjutnya sinyal diteruskan ke arduino untuk diprogram. Sinyal hasil pembacaan ditampilkan pada display dan menggerakkan aktuator damper melalui relay shield. Perancangan software menggunakan software Arduino IDE. Perancangan software terdiri atas pembuatan program otomatis dan manual. Program otomatis beroperasi berdasarkan nilai Setting Point (SP) yang diberikan untuk mempertahankan nilai tekanan dan menggunakan perhitungan kontrol proposional, sedangkan program manual berdasarkan push button untuk menaikkan dan menurunkan tekanan. Perancangan kontrol tekanan digital akan digunakan pada sistem ventilasi RSG-GAS.

**Kata kunci :** Sistem ventilasi, kontrol tekanan, arduino

#### ABSTRACT

**DESIGN OF ROOM PRESSURE CONTROL MODULE BASED ARDUINO IN RSG-GAS VENTILATION SYSTEM.** One of the functions of the RSG-GAS ventilation system is to control the negative pressure of each room in the RSG-GAS building. The negative pressure is controlled using a pressure control module with a signal input from transducer to drive the damper actuator. The purpose of this paper is to design a room pressure control module in order to replace the old pressure control modules that still using analog display and discontinued items. This design consists of designing hardware and software. The hardware design uses components such as HW 685 module, Arduino, 0.96 inch OLED display, relay shield, potentiometer, auto/manual switch and push button. The flow of the pressure control process, that is the signal from the transducer is received by the HW 685 module. then the signal is forwarded to Arduino to be programmed. The readout signal is displayed on the display and drives the damper actuator through the relay shield. The design of the software uses the Arduino IDE software. Software design consists of automatic and manual programming. The automatic program operates based on the given Setting Point (SP) to maintain the pressure value and uses proportional control calculations, while the manual program is based on push button to increase and decrease pressure. The digital pressure control design will be used in the RSG-GAS ventilation system.

**Keywords :** ventilation system, pressure control, arduino

## PENDAHULUAN

Reaktor Serba Guna G.A. Siwabessy (RSG-GAS) memiliki sistem ventilasi yang berfungsi mengendalikan udara yang masuk dan keluar gedung reaktor. Sistem ventilasi reaktor juga berfungsi mempertahankan suhu, kelembaban dan tekanan ruang. Suhu dan kelembaban dipertahankan untuk menjaga integritas komponen di dalam reaktor dan kenyamanan manusia bekerja, sedangkan tekanan ruang dipertahankan untuk menjaga tekanan negatif di dalam gedung reaktor. Tekanan negatif di dalam gedung reaktor bervariasi nilainya dan ditetapkan berdasarkan potensi radiasi dan kontaminasi udara yang berasal dari ruang tersebut. Di dalam RSG-GAS, terdapat empat sistem udara resirkulasi ruang yang dipertahankan tekanannya yaitu sistem udara resirkulasi balai operasi (KLA31), sistem udara resirkulasi balai eksperimen (KLA32), sistem udara resirkulasi ruang sistem pendukung (KLA33) dan sistem udara resirkulasi *primary cell* (KLA34). Parameter persyaratan tekanan negatif tiap sistem yaitu KLA31 -1,5 mbar, KLA32 -0,5 mbar, KLA33 -1 mbar dan KLA34 -2 mbar.<sup>[1]</sup>

Tekanan negatif ruang didapatkan dari nilai pembacaan *transducer* yang membandingkan nilai tekanan ruang di dalam gedung reaktor dengan tekanan di luar gedung reaktor (tekanan atmosfer). Hasil pembacaan tersebut diteruskan pada modul kontrol tekanan untuk mengendalikan katup *dampner* pasokan udara segar (KLA10).

Modul kontrol tekanan pada sistem ventilasi RSG-GAS sudah mengalami penuaan dan merupakan barang diskontinu yang tidak tersedia lagi di pasaran. Modul tersebut masih menggunakan pembacaan analog dan tidak dapat dikembangkan sesuai kemajuan teknologi yang ada sekarang. Pada makalah ini, dilakukan perancangan modul kontrol tekanan digital untuk menggantikan modul kontrol tekanan lama yang masih analog. Modul kontrol tekanan digital ini diharapkan dapat mengendalikan tekanan ruang agar tetap pada nilai parameter yang dipersyaratkannya.

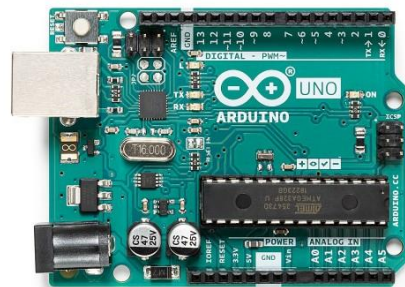
Perancangan modul kontrol tekanan menggunakan arduino sebagai pusat dari sistem kendalinya. Penggunaan arduino telah banyak dilakukan seperti untuk pengukuran konsentrasi CO<sub>2</sub>, suhu dan kelembaban pada penilaian pasokan udara luar sistem AC Split, aplikasi pemantauan kualitas udara ruang dan sistem kontrol aliran pendingin pada menara pendingin PLTN<sup>[2][3][4]</sup>. Pada makalah ini arduino digunakan untuk kontrol tekanan ruang dengan melakukan pemantauan tekanan ruang dari *transducer* dan mengendalikan buka tutup katup *dampner* pasokan udara segar.

Untuk menampilkan data digital, perancangan ini menggunakan *display OLED* dengan antarmuka *I2C* (*Inter Integrated Circuit*). Antarmuka *I2C* menggunakan 2 *wire* yaitu SDA dan SCL yang dapat diintegrasikan dengan pin arduino. Selanjutnya data yang diolah arduino dapat ditampilkan dalam bentuk digital pada *display*.<sup>[5]</sup>

## TEORI

### Arduino

Arduino merupakan perangkat elektronik *open-source* yang memiliki *input* dan *output* yang dapat diprogram.<sup>[6]</sup> Salah satu jenis papan arduino adalah arduino uno. Arduino uno merupakan sebuah *microcontroller* yang berbasis ATmega328P. *Microcontroller* ini memiliki 14 digital *input/output* pin dengan 6 pin diantaranya dapat digunakan sebagai PWM *output*, 6 analog *input*, *clock speed* 16 MHz dan koneksi USB. Tegangan minimum untuk arduino adalah 5 V, dan disarankan menggunakan tegangan sekitar 7 - 12 V.<sup>[7]</sup>



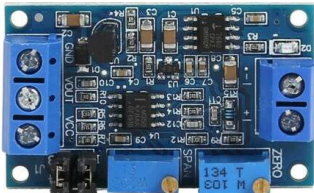
Gambar 1. Arduino Uno

## HW 685

HW 685 adalah modul elektronik yang berfungsi mengubah sinyal arus lemah (0/4 - 20 mA) menjadi tegangan DC 0 - 3,3/5/10 V. besar tegangan *output* yang diinginkan dapat dipilih dengan melakukan *jumper* pada pin J1.<sup>[8]</sup>

Tabel 1. Konfigurasi Pin J1 HW 685

Tegangan Output	Pin J1	
	1-2	3-4
0 - 3,3 V	<i>disconnected</i>	<i>disconnected</i>
0 - 5 V	<i>connected</i>	<i>connected</i>
0 - 10 V	<i>connected</i>	<i>disconnected</i>



Gambar 2. HW 685

### Display OLED 0.96 inch

Display OLED (Organic Light-Emitting Diode) merupakan suatu modul elektronik yang berfungsi untuk menampilkan huruf dan karakter (*alphanumeric*). Display OLED memiliki ukuran yang beragam, salah satunya adalah berukuran 0.96 inch dengan ketelitian *pixel* 128 x 64. Display ini dapat diprogram menggunakan *microcontroller* melalui komunikasi *I2C*. spesifikasi dari *display* ini adalah ukuran display 25 mm x 14 mm, tegangan kerja 5 VDC, arus 20mA dan komunikasi *I2C* 7-bit *address*.<sup>[9]</sup> Penelitian ini menggunakan *display OLED* sebagai penampil digital.



Gambar 3. Display OLED

### Relay Shield V3.0

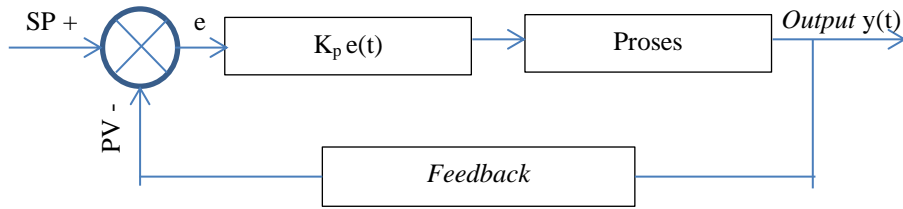
Relay shield V3.0 adalah sebuah modul *relay* generasi ke-3 yang kompatibel dengan arduino uno. *Relay shield* ini menyediakan 4 *relay* dengan masing-masing *relay* memiliki kontak *Normally Open* (NO) dan *Normally Close* (NC). Tegangan masukan untuk menggerakkan *relay* adalah 4,75-5,25 V dengan arus kerja 8-250 mA, dan kemampuan kontak NO/NC adalah tegangan *switching* 250VAC/30VDC dengan arus *switching* 15A/7A.<sup>[10]</sup>



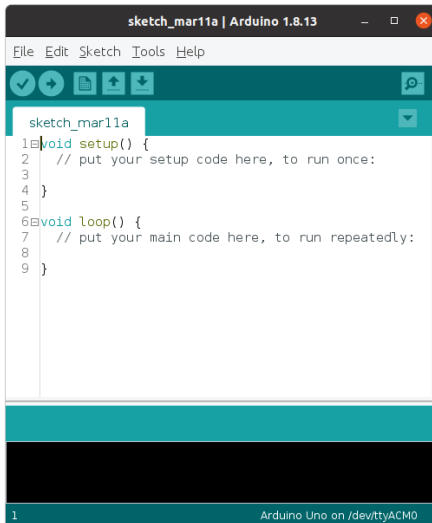
Gambar 4. Relay Shield V3.0

### Arduino IDE

Arduino IDE merupakan sebuah perangkat lunak menggunakan bahasa C yang bersifat *open-source* dan dapat menulis program yang dapat diunggah ke papan arduino.<sup>[11]</sup> Arduino IDE menyediakan berbagai jenis *library* program yang memudahkan pengguna untuk membuat program yang akan ditanamkan pada papan arduino.



Gambar 6. Close Loop Kontrol Proporsional



Gambar 5. Software Arduino IDE

### Kontrol Proporsional

Kontrol proporsional adalah kontrol yang mengendalikan nilai *error* dengan memberikan masukan pengali  $K_p$  (Konstanta proporsional). Nilai pengali  $K_p$  yang sangat besar menyebabkan sistem tidak stabil, dan nilai pengali  $K_p$  yang sangat kecil menyebabkan sistem menjadi lama untuk stabil.<sup>[12]</sup> Alur proses kontrol proporsional ditunjukkan pada Gambar 6.

Berdasarkan Gambar 6, sinyal dari kontrol proporsional dapat dinyatakan sebagai berikut:

$$e(t) = SP - PV \quad (1)$$

$$y(t) = K_p \cdot e(t) \quad (2)$$

Dimana  $y(t)$  adalah sinyal kontrol, dan  $e(t)$  sinyal *error*. Nilai  $e(t)$  didapatkan dari selisih  $SP$  (*Set Point*) dan  $PV$  (*Present Value*).

## METODOLOGI

### Identifikasi

Pada sistem ventilasi RSG-GAS, memiliki berbagai jenis nilai tekanan negatif pada setiap ruangan. Perbedaan nilai tekanan negatif tiap ruang menyebabkan beragamnya nilai jangkauan transducer tiap ruang. Hal ini perlu diidentifikasi guna menentukan skala pembacaan sensor pada modul kontrol tekanan. Sistem ventilasi RSG-GAS memiliki *transducer* dengan besar jangkauan pembacaan tekanan yang berbeda-beda yang ditunjukkan pada Tabel 2. Nilai maksimum dan minimum dari pembacaan *transducer* dijadikan nilai batas pembacaan sensor pada input analog modul kontrol tekanan.

Tabel 2. Nilai Jangkauan *Transducer*

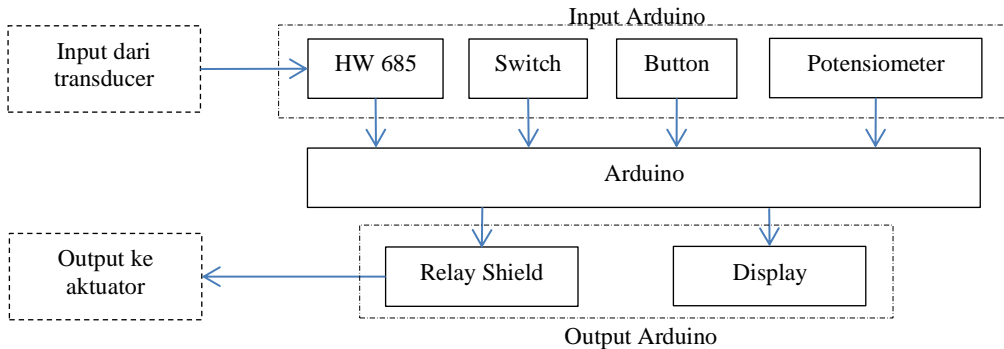
Sistem	<i>Transducer</i> (mbar)	
	Minimum	Maksimum
KLA31	-5	0
KLA32	-3	+2
KLA33	-3	+2
KLA34	-5	0

Pembacaan nilai tekanan ruang dilakukan konversi nilai sebanyak 4 kali. Pertama yaitu dari tekanan mbar dikonversi menjadi arus lemah 0-20 mA. Pada tahap ini terjadi di *transducer*. Kedua yaitu dari arus 0-20 mA dikonversi menjadi tegangan 0-5 VDC. Konversi ini terjadi pada modul HW 685. Ketiga yaitu dari tegangan 0-5 VDC dikonversi menjadi skala ADC (*Analog Digital Converter*) 0-1023. tahap terakhir yaitu dari skala ADC 0-1023 dikonversi menjadi nilai tekanan mbar untuk ditampilkan di *display*. Tahap ketiga dan keempat terjadi di dalam program arduino.

### Perancangan Hardware

Perancangan *hardware* pada modul kontrol tekanan digital digunakan untuk mengetahui pengkabelan antar tiap komponen. Perancangan ini dimaksudkan juga untuk menentukan pin analog *input* dan digital *input output* yang digunakan pada

papan arduino. Penentuan pin arduino ini selanjutnya digunakan untuk inisialisasi *input* dan *output* pada perancangan *software*. Blok diagram *hardware* ditunjukkan pada Gambar 7 dan tabel 3 menunjukkan konfigurasi *input output* arduino.



Gambar 7. Blok Diagram Hardware

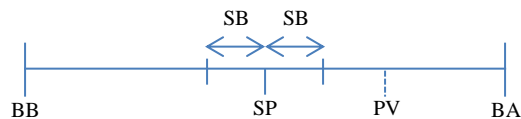
Tabel 3. Konfigurasi *input* dan *output* arduino

Komponen		Arduino
Display OLED 0,96 inch	GND	GND
	VCC	+5V
	SCL	SCL/A5
	SDA	SDA/A4
HW 685	GND	GND
	VCC	VIN
	VOUT	A0
Potensiometer	1	+5V
	2	A1
	3	GND
Button +	1	+5V
	2	D8
Button -	1	+5V
	2	D9
Switch A/M	1	+5V
	2	D10
Relay Shield	D4	D4
	D5	D5

### Perancangan Software

Perancangan *software* ini dilakukan untuk pembuatan program kendali tekanan otomatis dan manual menggunakan *software* Arduino IDE. Perancangan ini terdiri atas pembuatan beberapa sub program diantara-

nya sub program otomatis dan sub program manual. Rancangan program otomatis ditunjukkan pada Gambar 8.



Gambar 8. Rancangan Program Otomatis

Pada Gambar 8, BB dan BA merupakan batasan jangkauan pengukuran *transducer*. SP merupakan nilai tekanan yang ingin dicapai dan dipertahankan. PV merupakan nilai tekanan terkini hasil pembacaan dari *transducer*. SB yaitu nilai batasan dari SP dimana jika PV berada pada daerah SB maka program tidak memberikan perintah ke aktuator untuk melakukan kompensasi nilai tekanan. Jika PV berada di luar daerah SB, maka program akan memerintahkan aktuator untuk membuka/ menutup sampai PV=SP atau PV berada pada daerah SB. Tabel 4 menunjukkan arah gerak aktuator berdasarkan nilai PV, SP dan SB.



Tabel 4. Arah gerak aktuator berdasarkan nilai PV, SP dan SB

Kondisi	Aktuator
$PV < (SP - SB)$	Membuka
$PV > (SP + SB)$	Menutup
$(SP - SB) \leq PV \leq (SP + SB)$	Diam

Proses PV menuju SP menggunakan perhitungan kontrol proposional yang ditunjukkan pada persamaan (1) dan (2) yaitu:

$$Error = SP - PV \quad (3)$$

$$Time Error = |Error| \times Kp \quad (4)$$

*Time Error* merupakan nilai waktu program memberikan perintah ke aktuator untuk beroperasi (buka/tutup). Aktuator *dampner* akan bergerak menutup jika nilai *Error* positif dan aktuator *dampner* akan bergerak membuka jika nilai *Error* negatif. Lamanya waktu buka dan tutup aktuator bergantung pada nilai *Time Error*. Nilai *Time Error* diatur melalui besar nilai *Kp*. Semakin besar nilai *Kp* maka semakin cepat nilai PV menuju SP dan sebaliknya, semakin kecil nilai *Kp* maka semakin lambat nilai PV menuju SP.

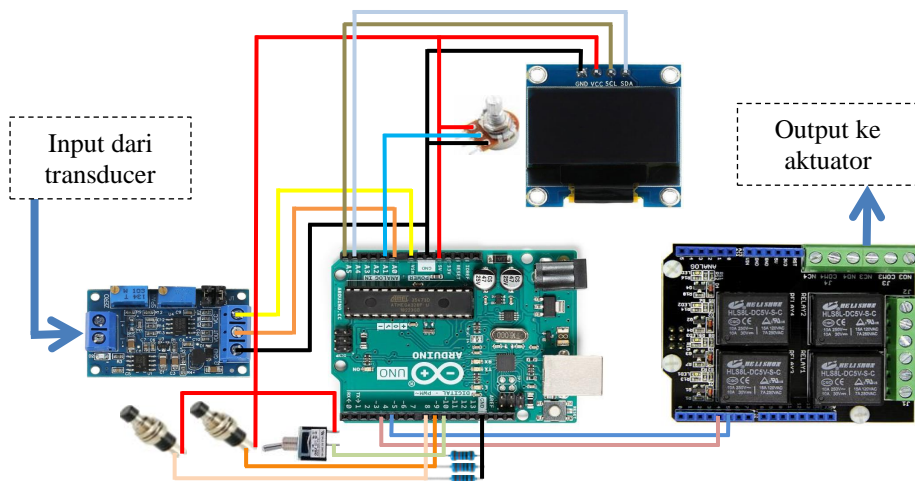
Rancangan program manual menggunakan *push button* sebagai perintah untuk menggerakkan aktuator. Jika *push button +*

ditekan, maka aktuator akan bergerak membuka, sedangkan jika *push button -* ditekan, maka aktuator akan bergerak menutup.

## HASIL DAN PEMBAHASAN

### Hasil Perancangan *Hardware*

Hasil perancangan *hardware* ditunjukkan pada Gambar 9. HW 685 menerima sinyal dari *transducer* untuk diteruskan ke arduino melalui ADC menggunakan pin A0. Sinyal yang masuk ke arduino diolah dan ditampilkan pada display melalui antarmuka I2C menggunakan pin SDA dan SCL. Hasil olah sinyal di dalam arduino digunakan untuk mengatur operasi aktuator melalui *relay shield* pada pin D4 dan D5. Nilai SP yang digunakan pada pengoperasian otomatis dikendalikan menggunakan potensiometer yang dihubungkan pada A1 pin arduino. Selanjutnya pada pengoperasian manual terdapat *push button +*, *push button -* dan *switch A/M* yang masing-masing menggunakan pin arduino D8, D9 dan D10.



Gambar 9. Hasil Perancangan *Hardware*

## Hasil Perancangan Software

Hasil perancangan *software* terdiri atas beberapa sub program, yaitu sub program inialisasi, sub program manual dan sub program otomatis.

### Sub program inialisasi

Hasil perancangan sub program inialisasi ditunjukkan pada Gambar 10.

```
void inialisasi(void) {  
    while(digitalRead(buttonPos)==HIGH){  
        //nothing  
    }  
    while (digitalRead(switchAM) == LOW) {  
        display.clearDisplay();  
        display.setTextSize(2);  
        display.setTextColor(SSD1306_WHITE);  
        display.setCursor(0, 0);  
        display.print(F("BA = "));  
        display.print(BA);  
        display.display();  
        delay(300);  
        if (digitalRead(buttonPos) == HIGH)BA++;  
        if (BA > 500)BA = 500;  
        if (digitalRead(buttonNeg) == HIGH)BA--;  
        if (BA < -500)BA = -500;  
    }  
    delay(1000);  
    while (digitalRead(switchAM) == HIGH) {  
        display.clearDisplay();  
        display.setTextSize(2);  
        display.setTextColor(SSD1306_WHITE);  
        display.setCursor(0, 0);  
        display.print(F("BB = "));  
        display.print(BB);  
        display.display();  
        delay(300);  
        if (digitalRead(buttonPos) == HIGH)BB++;  
        if (BB > 500)BB = 500;  
        if (digitalRead(buttonNeg) == HIGH)BB--;  
        if (BB < -500)BB = -500;  
    }  
    delay(1000);  
    while (digitalRead(switchAM) == LOW) {  
        display.clearDisplay();  
        display.setTextSize(2);  
        display.setTextColor(SSD1306_WHITE);  
        display.setCursor(0, 0);  
        display.print(F("SB = "));  
        display.print(SB);  
        display.display();  
        delay(300);  
        if (digitalRead(buttonPos) == HIGH)SB++;  
        if (digitalRead(buttonNeg) == HIGH)SB--;  
        if (SB < 0)SB = 0;  
    }  
    delay(1000);  
    while (digitalRead(switchAM) == HIGH) {  
        display.clearDisplay();  
        display.setTextSize(2);  
        display.setTextColor(SSD1306_WHITE);  
        display.setCursor(0, 0);  
        display.print(F("KP = "));  
        display.print(KP);  
        display.display();  
        delay(300);  
        if (digitalRead(buttonPos) == HIGH)KP++;  
        if (digitalRead(buttonNeg) == HIGH)KP--;  
        if (KP < 1)KP = 1;  
    }  
    delay(1000);  
}
```

Gambar 10. Potongan Source Code Sub Program Inialisasi

Sub program ini dieksekusi ketika modul hidup pertama kali, *Switch A/M* pada posisi manual dan *push button +* ditekan. Sub

program ini dibuat untuk menentukan nilai BA, BB, SB dan Kp. Ketika masuk ke sub program ini, diminta untuk mengatur nilai BA dengan menekan *push button +* dan *push button -*. Untuk berpindah ke pengaturan selanjutnya, *switch A/M* diposisikan otomatis dan program beralih untuk pengaturan nilai BB. Selanjutnya, *Switch A/M* diposisikan manual dan program beralih ke pengaturan nilai SB. Pada tahap terakhir dari sub program ini adalah pengaturan nilai Kp dengan memposisikan *switch A/M* ke otomatis. Untuk keluar dari sub program, *switch A/M* diposisikan manual dan program akan beralih ke sub program manual.

### Sub program manual

Hasil perancangan sub program manual ditunjukkan pada Gambar 11.

```
//Mode Manual  
if (digitalRead(switchAM) == LOW) {  
    if (digitalRead(buttonPos) == HIGH) {  
        digitalWrite(relayPosition, HIGH);  
        digitalWrite(relayActive, HIGH);  
        displayPos();  
        delay(300);  
    }  
    else if (digitalRead(buttonNeg) == HIGH) {  
        digitalWrite(relayPosition, LOW);  
        digitalWrite(relayActive, HIGH);  
        displayNeg();  
        delay(300);  
    }  
    else {  
        digitalWrite(relayPosition, LOW);  
        digitalWrite(relayActive, LOW);  
        displayStandby();  
        delay(50);  
    }  
}
```

Gambar 11. Potongan Source Code Sub Program Manual

Sub program manual dieksekusi ketika *switch A/M* diposisikan manual. Pada sub program ini tekanan dikendalikan secara manual. Tekanan akan turun ketika *push button -* ditekan sehingga program akan memerintahkan aktuator untuk bergerak menutup. Selanjutnya tekanan akan naik ketika *push button +* ditekan sehingga program akan memerintahkan aktuator untuk bergerak membuka.

### Sub program otomatis

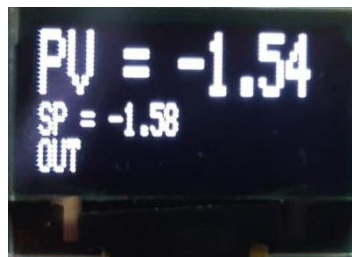
Hasil perancangan sub program otomatis ditunjukkan pada Gambar 12.

```
//Mode Otomatis
if (digitalRead(switchAM) == HIGH) {
  sensor1 = analogRead(A0);
  sensor2 = analogRead(A1);
  PV = map(sensor1, 0, 1023, BB, BA);
  SP1 = map(sensor2, 0, 1023, BB, BA);
  error = SP1 - PV;
  if (error > SB) {
    Yt = abs(error)*KP;
    TimeError = Yt;
    if(TimeError > 250)TimeError=250;
    digitalWrite(relayPosition, HIGH);
    digitalWrite(relayActive, HIGH);
    for (int i = 0; i < TimeError; i++) {
      delay(100);
      displayPos();
      if (PV >= SP1)break;
    }
    digitalWrite(relayPosition, LOW);
    digitalWrite(relayActive, LOW);
    error = 0;
  }
  if (error < -SB) {
    Yt = abs(error)*KP;
    TimeError = Yt;
    if(TimeError > 250)TimeError=250;
    digitalWrite(relayPosition, LOW);
    digitalWrite(relayActive, HIGH);
    for (int i = 0; i < TimeError; i++) {
      delay(100);
      displayNeg();
      if (PV <= SP1)break;
    }
    digitalWrite(relayPosition, LOW);
    digitalWrite(relayActive, LOW);
    error = 0;
  }
  displayStandby();
  delay(50);
}
```

Gambar 12. Potongan Source Code Sub Program Otomatis

Sub program otomatis dieksekusi ketika *switch* A/M diposisikan otomatis. Sub program ini bekerja berdasarkan nilai SP yang diatur melalui potensiometer. Pada awal sub program, nilai PV dan SP dikonversi dari satuan ADC ke satuan mbar dengan nilai maksimum BA dan nilai minimum BB. Selanjutnya, dilakukan perhitungan nilai *Error* yang didapatkan dari selisih antara PV dan SP. Setelah perhitungan nilai *Error*, dilakukan perhitungan nilai *Time Error* dari perkalian antara nilai absolut *Error* dengan Kp. Jika nilai absolut *Error* lebih besar dari nilai SB, maka program akan memerintahkan aktuatur untuk bergerak. Nilai *Error* positif akan memerintahkan aktuatur bergerak membuka selama (*Time Error* x 100) ms. Nilai *Error* negatif akan memerintahkan aktuatur bergerak

menutup selama (*Time Error* x 100) ms. Jika nilai absolut *Error* kurang atau sama dengan nilai SB, maka aktuatur tidak beroperasi. Nilai PV, SP dan indikator buka/tutup ditampilkan ke *display* sebagai pembacaan digital. Tampilan *display* ditunjukkan pada Gambar 13.



Gambar 13. Tampilan Display

## KESIMPULAN

Perancangan kontrol tekanan ruang digital diharapkan dapat mengganti modul kontrol tekanan yang lama dengan fungsi kerja sama tetapi dengan tampilan yang sudah digital. Perancangan ini menggunakan arduino sebagai pusat kendali sehingga mudah dikembangkan lebih lanjut ke arah IoT (*Internet of Things*) dengan menambahkan komponen lain seperti *ethernet shield* dan sebagainya. Rancangan ini diharapkan dapat diaplikasikan pada sistem ventilasi RSG-GAS seperti pada sistem KLA31, KLA32, KLA33 dan KLA34 sebagai pengatur tekanan ruang.

## UCAPAN TERIMAKASIH

Dengan selesainya penelitian ini, kami mengucapkan terimakasih kepada PRSG-BATAN yang telah memfasilitasi penelitian ini. Kami mengucapkan terimakasih juga kepada Ir. Yusi Eko Yulianto selaku Kepala PRSG-BATAN dan Ir. Edison, M.Si selaku Koordinator Pemeliharaan Reaktor.

## DAFTAR PUSTAKA

- [1] LAK. RSG-GAS Rev.10.1. PRSG – BATAN.
- [2] A. Bimaridi, K. D. Putra, E. Djunaedy, and M. R. Kirom, "Assasment of Outside Air Supply for Split AC system - Part A:



- Affordable Instrument,” *Procedia Eng.*, vol. 170, pp. 248–254, 2017.
- [3] S. Abraham and X. Li, “A cost-effective wireless sensor network system for indoor air quality monitoring applications,” *Procedia Comput. Sci.*, vol. 34, pp. 165–171, 2014
- [4] A. Hossain, S. Islam, Z. Rahman, A. Z. M. Salahuddin, and A. S. Mollah, “An intelligent flow control system of coolant for a water reactor based cooling tower,” *Energy Procedia*, vol. 160, pp. 566–573, 2019
- [5] A. OO, “Design, simulation and implementation of an Arduino microcontroller based automatic water level controller with I2C LCD display,” *Int. J. Adv. Appl. Sci.*, vol. 9, no. 2, p. 77, 2020
- [6] Anonim, “What is Arduino?”, Arduino, 5 Februari 2018, [Online]. Tersedia: <https://www.arduino.cc/en/Guide/Introduction> [Diakses 5 Januari 2021]
- [7] Electronic Component’s Datasheet, [Online]. Tersedia: <https://www.alldatasheet.com/view.jsp?Searchword=ATMEGA328P> [Diakses 10 Januari 2021]
- [8] Anonim, “HW-685 4-20mA Current Sensing”, Turais Tech, 1 Januari 2021, [Online]. Tersedia: <https://docs.turais.de/docs/modules/hw685/> [Diakses 15 Januari 2021]
- [9] Purbakawaca, Rady, “Monochrome 0.96” 128x64 OLED Graphic Display”, *Academia*, 24 Februari 2019, [Online]. Tersedia: [https://www.academia.edu/38428432/Monochrome\\_0\\_96\\_128x32\\_OLED\\_graphic\\_display\\_pdf](https://www.academia.edu/38428432/Monochrome_0_96_128x32_OLED_graphic_display_pdf) [Diakses 4 Februari 2021]
- [10] Anonim, “Relay Shield v3.0”, Seeed, 2016, [Online]. Tersedia: [https://seeeddoc.github.io/Relay\\_Shield\\_v3.0/](https://seeeddoc.github.io/Relay_Shield_v3.0/) [Diakses 4 Februari 2021]
- [11] Anonim, “Arduino IDE 1.8.13”, Arduino, [Online]. Tersedia: <https://www.arduino.cc/en/software> [Diakses 4 Februari 2021]
- [12] A. R. Al Tahtawi, Y. Somantri, and E. Haritman, “Design and Implementation of PID Control-based FSM Algorithm on Line Following Robot,” *JTERA*, vol. 1, pp. 23-30, 2016