ATCSJ
Applied Technology and Computing Science Journal
Open Access **p-ISSN 2621-4458 | e-ISSN 2621-4474**

Available online to journal2.unusa.ac.id
UNUSA
S4-Accredited – SK No.200/M/KPT/2021
Journal Page is available to https://journal2.unusa.ac.id/index.php/ATCSJ

Moisture Control System in Mustard Plants Using the PID Method

Enny Indasyah[1], Robiq Abiyyu Sadid[2], Fauzi Imaduddin Adhim[3], Fivitria Istiqomah[4], Rizqi Putri Nourma Budiarti[5]

[1,2,3,4]*Institut Teknologi Sepuluh Nopember, Indonesia*
Raya ITS, Surabaya
[1*]enny_indasyah@its.ac.id, [3]fauzi@eea.its.ac.id

[5]*Universitas Nahdlatul Ulama Surabaya, Indonesia*
Raya Jemursari 51-57, Surabaya
[5]rizqi.putri.nb@unusa.ac.id

Article history:

Received 29 December 2022
Revised 29 December 2022
Accepted 29 December 2022
Available online 31 December 2022

Keywords:

Plant Growth,
Sensors,
Humidity Control,
Proportional Integral Derivative,
Automatic Irrigation.

Abstract

Irrigating paddy fields is a very important process for plant growth. When the process of irrigating rice fields is carried out by means of continuously flowing water and the soil moisture value is only based on estimates, there are many possibilities for the moisture value not to be measured where this causes the plants to take a long time to grow and the plants will rot or fail to harvest if the humidity value is too high. So a final project was designed with the title Humidity Control System in Mustard Plants Using the PID Method. The way this system works is by placing 5 humidity sensors at five different points to read soil moisture values and each sensor is connected to the Arduino Mega 2560 microcontroller. The controller used in this irrigation system is Proportional Integral Derivative (PID) to get humidity values according to the needs of the plant. The design and construction of an automatic irrigation system in this study was successfully carried out using the PID method, the corresponding Kp parameter values for the system were obtained, namely Kp = 4.328, Ki = 2.634, and Kd = 3.217. This automatic irrigation system can reach the moisture point of the mustard plant, which is 60% with a steady state error of 0.387% and a rise time 79 seconds. The test was carried out for 7 days and by comparing the average plant height in the manual watering area and the automatic watering area. The average growth of plant height in the manual watering area and the automatic watering area has a difference of 1.2 cm. Because this system only focuses on water irrigation where plants also need other nutrients for growth besides water.

1* Coressponding Author

I. INTRODUCTION

UD. Subur Jaya is a farmer group engaged in agriculture where most of its members grow staple crops and vegetable crops such as mustard greens and corn which are located in Doko Village, Doko District, Blitar Regency. The irrigation system for farming SMEs at UD Subur Jaya still uses manual methods or human labor. If the position of the river is higher than the rice fields, farmers must make their own waterways by hoeing the soil on the banks of the river so that water from the river can flow into the fields. The water will be allowed to continue flowing into the fields until the water has inundated the fields. Then the farmers must close the waterways that have been made with soil again. If the position of the river is lower than the rice fields, the farmer must use a diesel generator or generator to irrigation system[1]–[4]. With a watering system like this, soil moisture is not measured according to plant needs, but only based on farmers' estimates. If the water humidity is only based on farmers' estimates, there are many possibilities that the humidity value is not measured according to the needs of the plants, where this can cause the plants to take a long time to grow if there is a lack of water and the plants will rot or fail to harvest if there is too much water. Then a tool is needed to carry out automatic irrigation that can provide soil moisture values according to plant needs. On the other hand, UD Subur Jaya partners also need a remote-control feature on this tool.

The location of the paddy fields in this UMKM is quite far from residential areas, which is around 200-300 meters. Of course, with this distance, the rice fields cannot be reached by a power source. The condition of the paddy fields in UMKM is also quite difficult to get an internet signal, so the internet signal range is unstable. So, the project was designed with the title Humidity Control System in Mustard Plants Using the PID Method. The way this system works is by placing 5 humidity sensors at five different points to read soil moisture values and each sensor is connected to the Arduino Mega 2560 microcontroller. When the humidity value is below the plant's needs, namely 60%, there will be a warning from the system in the form of an SMS to farmers where this data is sent by the sim900a GSM module. Then user must provide confirmation in the form of an SMS reply to activate the automatic irrigation system, and user can ignore the message if he does not want to do irrigation. If the system receives a confirmation in the form of an SMS, the valve connected to the water reservoir will open where valve is connected to the stepper motor to control the discharge of water coming out of the reservoir. The valve will continue to open and slowly close when the humidity value approaches the plant's needs, namely 60%. And valve will be completely closed when the humidity value is right according to the needs of the plant. After valve is closed, there will be a notification from the system to the user that the irrigation system is no longer active[5], [6]

The controller used in this irrigation system is Proportional Integral Derivative (PID)[7]. By reducing the error signal that occurs when the system is working, as well as providing a control signal output that has a fast response, steady state error, and no overshoot. The smaller the error that occurs, the more accurate the humidity value according to plant needs so that plant growth can run well. So that farmers do not experience losses in the form of crop failure. This controller has a proportional value as the error value for the correction value[5], [8]–[12].

II. RELATED WORKS

A. Irrigation System

Irrigation is a system of irrigation to cultivated land. There are many types of irrigation systems. Irrigation or watering is important in agriculture. Without good irrigation, plants cannot grow optimally. This will also greatly affect the yield later. Knowing the importance of irrigation makes irrigation development a priority. According to data in the databox, the government through the Ministry of Public Works and Public Housing (PUPR) is targeting irrigation improvements of 90 percent in 2019. Improvements to the irrigation system for agricultural land have also been carried out in 2016 and 2017. In 2016 the Ministry of PUPR succeeded in repairing 286 thousand hectares irrigation and in 2017 an area of 325 thousand hectares. An understanding of irrigation is not only needed by farmers or the government. Knowledge of this irrigation system also needs to be known by many people. Irrigation is the systematic distribution of water for the purposes of plant growth into cultivated land[4], [7], [13]. Meanwhile, in the explanation in Government Regulation Number 20 of 2006, irrigation is an effort to provide, regulate, and dispose of irrigation water to support agriculture, the types of which include surface irrigation, swamp irrigation, underground water irrigation, pump irrigation, and pond irrigation. Irrigation is an activity related to efforts to obtain water for rice fields, fields, plantations, and other agricultural businesses[4].

B. PID controller

The proportional or gain controller acts as an amplifier that can change the output of the system proportionally without giving a dynamic effect on the performance of the controller as shown in Figure 1 [7]. The response of the proportional controller can be expressed by equation 1 as follows:

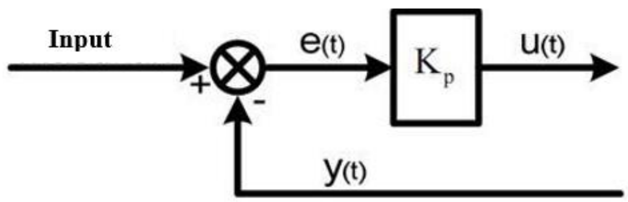

Figure 1. Proportional Control

$$\boldsymbol{u(t) = Ki \int_0 e(t)d(t)} \quad (1)$$

Information:
Kp = gain proportional
e = error
u = output value relative to time (t).

Integral or reset controllers are controllers that can improve the steady state response of the system, so that these controllers are able to minimize system errors as shown in Figure 2. The response of the integral controller can be expressed in equation 2 as follows:

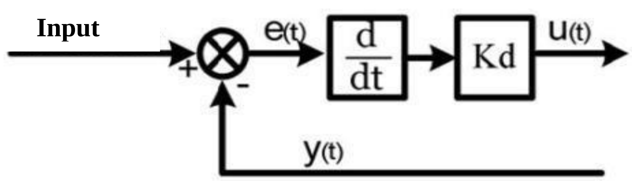

Figure 2. Integral Control

$$\mathbf{u(t)} = \boldsymbol{Kd \frac{de(t)}{dt}} \quad (2)$$

Description:
Ki = gain integral
e = error
u = output value relative to time (t).

The large output of the derivative controller has properties like differential operations in general. Derivative controllers use the rate of change of the error signal as a control parameter. If there is no change in the error signal, the output of the derivative control will not change as shown in Figure 3. The response of the derivative controller can be expressed in equation 3 as follows:

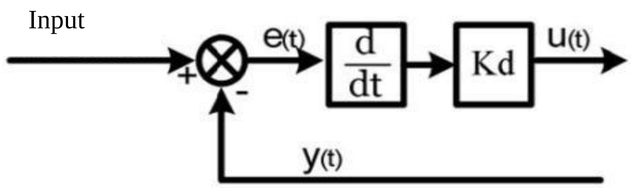

Figure 3. Derivative control

$$\mathbf{u(t)} = \boldsymbol{Kd}\ \frac{\boldsymbol{de(t)}}{\boldsymbol{dt}} \tag{3}$$

Note:
Kd = gain derivative
e = error
u = output value relative to time (t).

The PID controller is the result of a combination of three components, namely proportional, integral, and derivative which can be used simultaneously or individually depending on the desired response in a system or plant as shown in Figure 4. From each controller that has been described, so that it can be expressed in the equation 4 of the PID controller is drawn as follows:

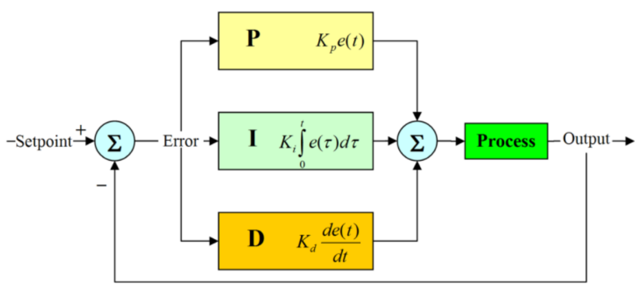

Figure 4. PID Control Block Diagram

$$\boldsymbol{u(t)} = \boldsymbol{Kp\ e(t)} + \boldsymbol{Ki}\ \int_0^t \boldsymbol{e(t)d(t)} + \boldsymbol{Kd}\ \frac{\boldsymbol{de(t)}}{\boldsymbol{dt}} \tag{4}$$

Description:
u(t) = PID controller output signal
Kp = proportional constant
Ki = integral constant
Kd = derivative constant
e(t) = error signal

C. Determining PID Parameters PID

controller parameter tuning is always based on a review of the set characteristics (Plant). To determine the parameters of the PID controller, a method is used, namely the Ziegler Nichols method[7].

D. The 1st method The *Ziegler-Nichols*

The method The *Ziegler-Nichols* is a way or method to determine the PID value. This method has been developed by 2 figures namely John G Ziegler and Nathaniel B Nichols. *Ziegler-Nichols* first introduced his method in 1942. This method has two ways, the oscillation method, and the reaction curve. Both methods are intended to produce a system response with a maximum spike of 25%. Figure 5 shows a curve with a 25% spike[7].

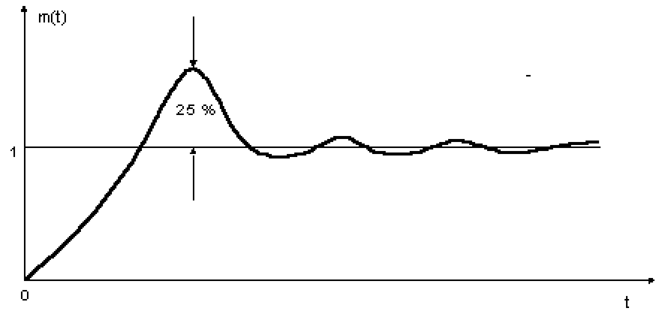

Figure 5. Unit Ladder Response Curve Showing 25% Maximum Surge

III. METHODS

A. System Design

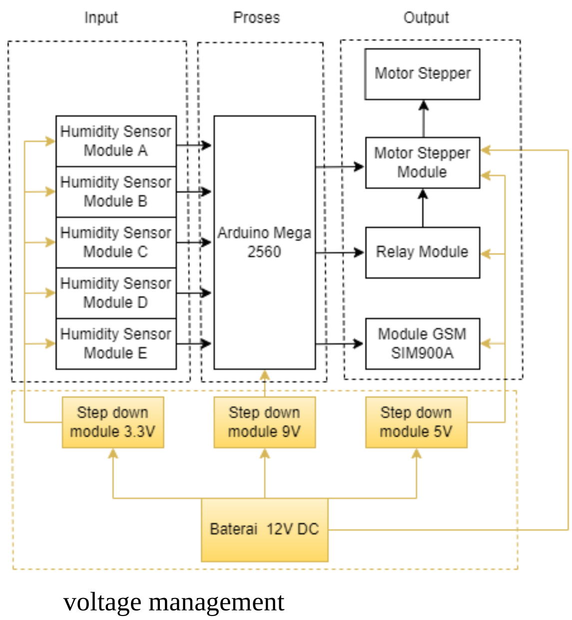

Figure 6. System Block Diagram

In Figure 6, it is explained that this irrigation system is built with a control panel that functions as a system controller and several other sensors. The design of this tool system consists of several interconnected components, starting from the battery as a voltage source. This battery will be lowered in advance according to the needs of each component. The humidity sensor requires a voltage of 3.3 volts, while for the relay module, the GSM module requires a voltage of 5 volts and the Arduino Mega requires a voltage between 7-12 volts according to the *datasheet*. Data from each sensor will be processed on Arduino Mega. The stepper motor module requires a voltage of 5 volts for the stepper motor driver voltage and 12 volts for the stepper motor. The humidity sensor data will later be sent to the user using the GSM module through an intermediary BTS tower, where the GSM module is also connected to the Arduino Mega. Apart from being a data sender, the GSM module functions as a data receiver from the user in the form of commands to send humidity data. Of course, the network used is the SMS and telephone network or the *second generation* (2D) network.

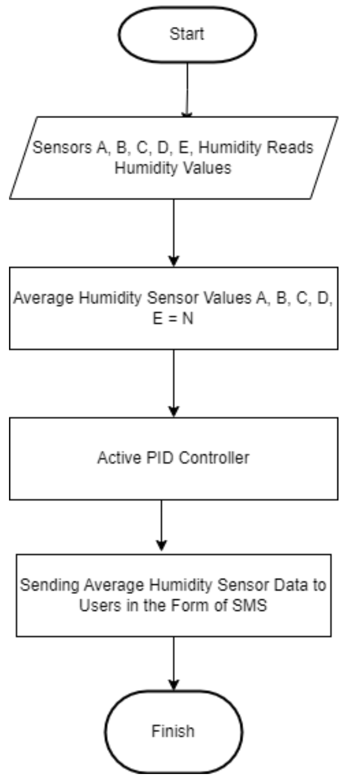

Figure 7. System Flowchart

In Figure 7, the system flowchart shows the sensor reading soil moisture whether the soil moisture value is below 60%. If it is below 60%, then the *valve* connected to the stepper motor will open according to the needs controlled by the PID controller. With the confirmation given by user, the valve will continue to open until the soil moisture value reaches 60%. The valve will slowly close automatically when the humidity value continues to increase close to 60% and will automatically stop when the humidity value is exactly 60%. The reading of the humidity value is carried out by the soil moisture which is connected to the microcontroller.

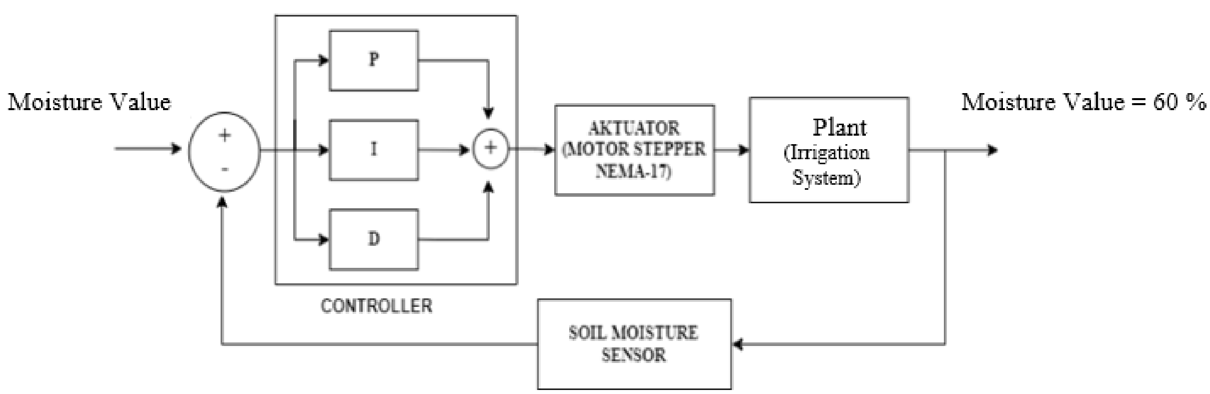

Figure 8. System Control Block Diagram

Figure 8 describes the work process flow of the irrigation system for humidity control. The working system of this tool starts when the humidity sensor reads the soil moisture value. This humidity sensor is placed at five different points and each sensor is connected to a microcontroller as a data receiver. When the soil moisture value is below the set point value, the *valve* will open waiting for commands from the *user*. The *valve* will automatically slowly close until the humidity value reaches the set point value. Closing rotation *valve* is determined by the moisture value in the soil that has been given the PID controller according to the calculation. The system will automatically stop when the soil moisture value has reached a relative set point of 60%. Data will be sent as a warning to the user whenever soil moisture is below a predetermined value or set point, namely for mustard greens the relative humidity is 60%. Then a message will appear confirming that the *valve* has been closed or the humidity value has reached the set point and the system will stop.

B. Mechanical Design

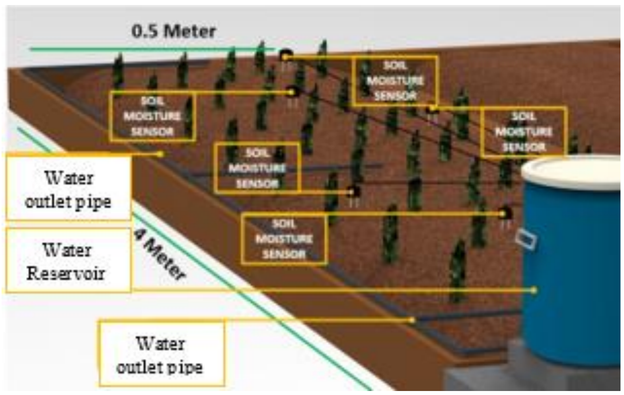

Figure 9. Mechanical Design

Figure 9 illustrates the location of the humidity sensor placed in the paddy fields. There are five humidity sensors placed at five different points in the hope that the sensor readings can be evenly distributed to get maximum sensor readings. The cable length for each humidity sensor is 3 meters by plugging the sensor into the ground with a soil depth of 5 cm.

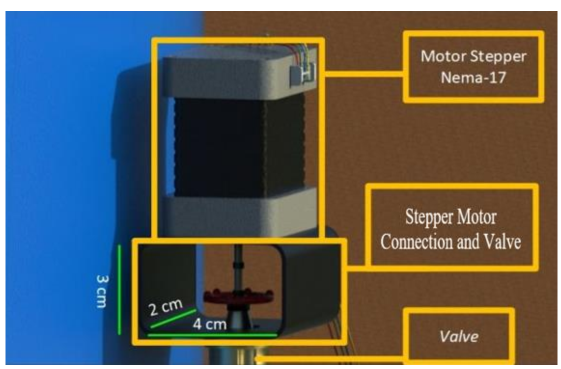

Figure 10. Placement of Stepper Motor with *Valve*

In the reservoir there is a *valve* with a diameter of 1.27cm and it is connected to the stepper motor just above the *valve*. For the connection between the stepper motor and the *valve*, use a square iron with a length of 3 cm, a width of 2 cm and a height of 4 cm as shown in Figure 10. *valve* is also connected to a pipe measuring 1.27 cm in diameter which has been directed around the rice fields at several points. The aim of directing the pipe around the rice fields at several points is that the water can flow evenly to all rice fields so that irrigation can be carried out evenly.

E-ISSN:*2621-4474*, *ISSN:2621-4458.*

C. Electronic Design

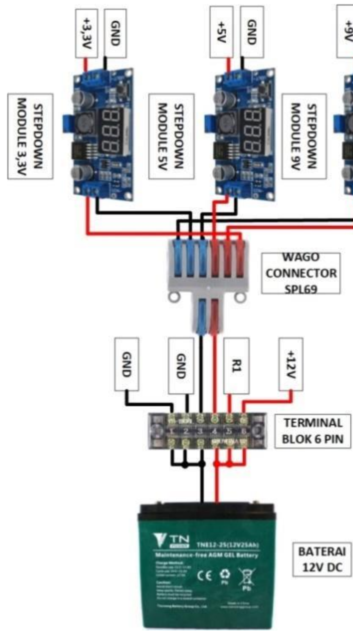

Figure 11. *Wiring* Power Circuit

In Figure 11 are *wiring*, terminal blocks and *stepdown*. The battery used in this system is a 12 Volt 10Ah battery. This is chosen for the durability of the system as well as adjusting to the size of the sensor and several other components related to the power supply. Starting from the battery which is connected to the 6-pin terminal block using an AWG 18 0.75mm cable, then each pin is connected to each other. Then the 2 outputs from the terminal block are connected to the *wago connector* SPL62 terminal block. The output of the *wago connector* modules *stepdown* with a voltage of 3.3 Volts, 5 Volts and 9 Volts respectively. This was chosen because it adapts to the voltage requirements of each component used.

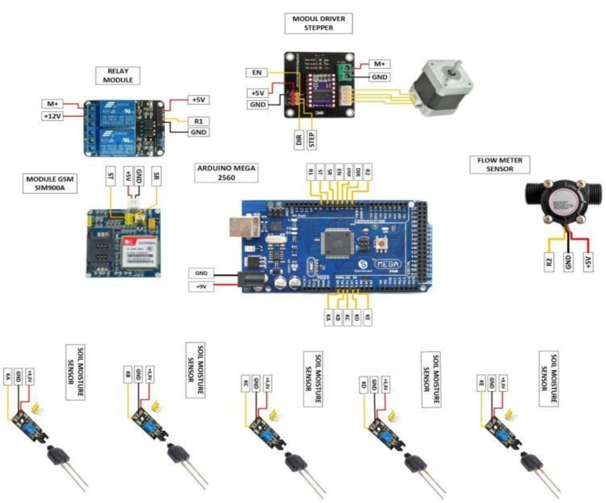

Figure 12. *Wiring* Control Circuit

In Figure 12 is a front panel design where there is a controller used in this system, namely by using Arduino Mega 2560. This is used with the hope that the system will have high durability. The use of *pinout*

A0, A1, A2, A3, A4 is used to connect with the humidity sensor. For D2 and D8 connected to *the relay* used to control *the relay*. *Pinouts* D3, D4, and D5 are connected to the stepper motor module which has *driver* as a control *valve*. *Pinout* D6 and D7 are connected to the *GSM Module SIM900A* to transmit sensor data. And the last is *the power jack* which is connected to a power source (9 Volts) and *ground*). Then there is also a *relay* in this system that is used as a substitute for a switch, which disconnects the voltage from the battery to the stepper motor. This is used in the hope of reducing *overheating* of stepper motor drivers and stepper motors. To connect *the relay module* controller's *pinout* GND and VCC *the relay module*. For sending data in the form of voltage (voltage) obtained from the microcontroller, pins IN1 and IN2 are connected to *pinouts* D2 and D8 on the controller. Furthermore, there is a GSM Module in this system that functions as communication between the system and the *user.* The GSM module used is the *GSM Module SIM900A.* To operate this GSM module requires a 5V voltage source. The VCC and GND pins on the GSM module are each connected to a 5 Volt voltage source. While pins 5VR and 5VT on the GSM module are connected to pins D7 and D8 on the controller for data transmission. Then there is a stepper motor in this system which functions as a regulator for opening and closing the *valve* so that it meets the plant's water needs. The stepper motor used is the nema-17 stepper motor. The stepper motor rotation is regulated by the DRV8825 stepper motor driver which is connected to the *stepper driver module.* To operate *stepper driver module*, pins V and G are connected to a 5 Volt voltage source and *ground* to supply the stepper motor driver voltage. For the VIN and GND pins, each is connected to a 12 Volt voltage source and *ground* to supply the stepper motor voltage. For data communication, EN, STEP, DIR pins are connected to pins D5, D4, D3 on the controller. For pins 1A, 1B, 2A, 2B connected to the stepper motor. Furthermore, there *soil moisture sensor* where to turn on the *soil moisture sensor* a voltage source is needed, namely 3.3 Volts. On *the soil moisture sensor* pin + and - respectively connected to a 3.3 Volt voltage source and *ground.* Then the AO pin is connected to the analogue pin on the microcontroller. Then for the next electrical component, namely the *flow sensor,* to turn on the YF-S201 flow sensor, the required voltage source is needed. Where the required sensor requirement is 5 Volts which is connected by a *stepdown module* that outputs 5 Volts and Ground voltage. Then, it is connected to pinout D2 on the Arduino Mega 2560 so that the pulse on the flow sensor can be read by the controller.

D. Design of PID

Control PID in this system aims to control soil moisture in mustard plants. Where when the humidity of the water is only based on farmers' estimates, there are many possible moisture values that are not measured according to the needs of the plants where this can cause the plants to take a long time to grow if there is a lack of water and the plants will rot or fail to harvest if there is too much water. So that the desired PID control is to be able to control humidity by providing water discharge according to plant needs, namely the soil moisture value is 60%, the soil moisture value does not experience an *overshoot* of 16.67% and does not experience a *steady state error* for 120 seconds. If *steady state error* occurs for 120 seconds or more and *overshoot* of 16.67% will result in rotting of the plants so that the plants will die, and farmers will experience crop failure. Then the time to go to the set point or rise time (Tr) is no more than 90 seconds. This is because the time needed by farmers to water plants with a paddy field size of 2 m^2 is no more than 90 seconds. This is because the time needed by farmers to water plants with a paddy field size of 2 m^2 is no more than 90 seconds. So that with PID control in this irrigation system it can provide moisture values for mustard plants according to plant needs and can also replace the role of farmers in terms of watering manually.

According to the 1st Ziegler Nichols method, before tuning the PID, a response curve from *plant* which aims to obtain *delay* and time constant results. The correlation between the step motor, water discharge and humidity are obtained from *valve* which is regulated by the stepper motor. *The valve* is connected to the *flow* sensor and then the water discharge value is obtained at each step. PID testing starts with running the system using an *open loop* system control system to look for a system response that matches the humidity value on the sensor. The humidity value in question refers to the workflow of the system. So, the authors tried testing by giving 625 steps to the stepper motor, namely with a water discharge value of 368 L/H with a *set point* humidity. The system response can be seen in Figure 13.

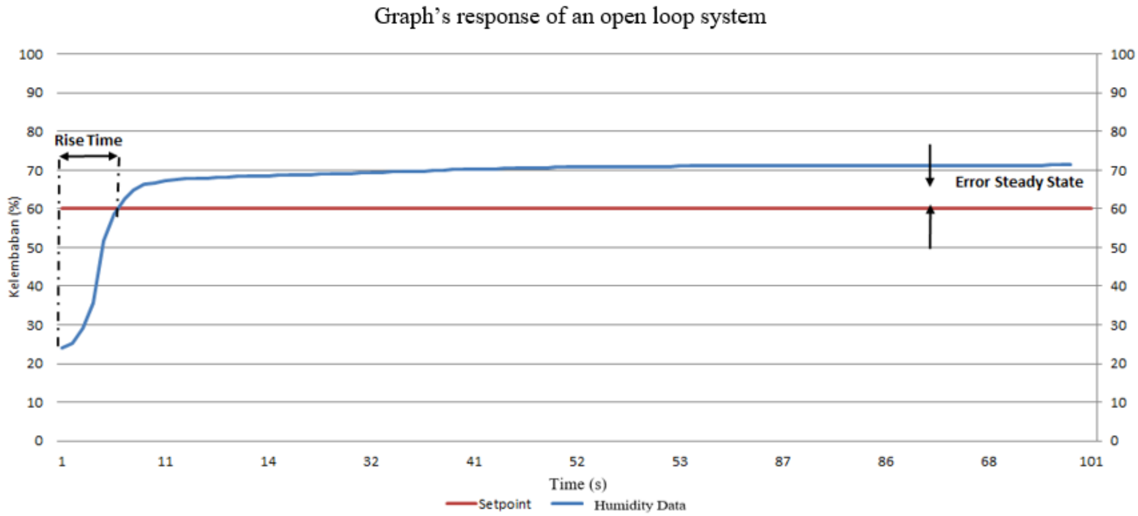

Figure 13. Graphs's Response of Open-loop System.

In graphic 13 you can see the blue line as the response curve from the humidity sensor and the red line is the *set point value.* The results of the graph above will later become the basic parameters of the *Ziegler Nichols* 1 method to obtain time delay and constant time. After obtaining the response graph of the *open loop* then determining the system modelling or *transfer function* by means of the data that has been obtained entered the *software,* namely the MATLAB *system identification toolbox* as shown in Figure 14. The result of the value of the *transfer function* of *software* is the overall system modelling.

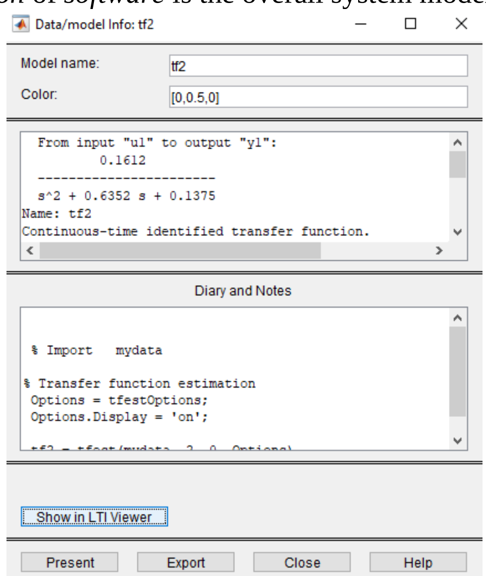

Figure 14. System Transfer Function Value

The value of *transfer function* above refers to the response of the system using *an open loop system.* Then from the value of the *transfer function* a simulation is carried out to obtain a tuning graph of the *transfer function* as shown in Figure 15.

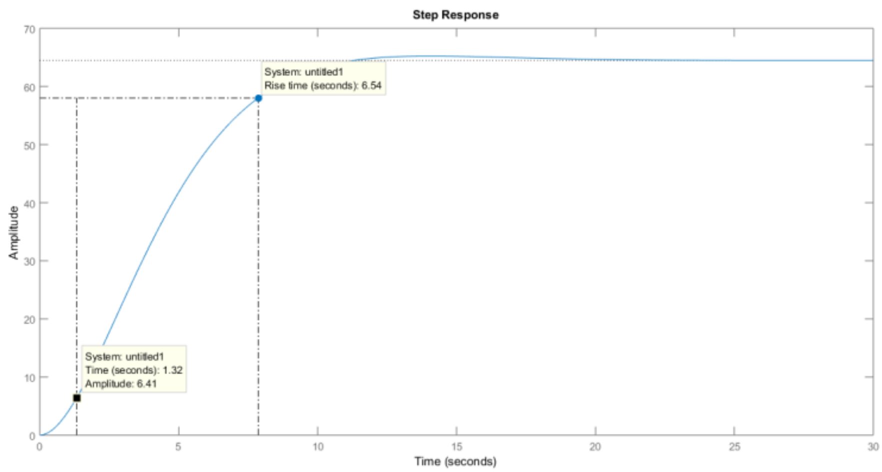

Figure 15. System Transfer Function Response Graph

In Figure 15 the value is obtained with L = 1.34 and the value T = 5.2, this can be seen from the deviation of the blue line to determine the value of kp, ki and kd with the calculations in Table 1. To determine the parameters P, I and D based on the *Ziegler Nichols* 1 method, the first is to find the value of a by looking at the resulting delay time (L) and constant time value (T) using Equation 5.

$$a = {}^{T}/_{L} \quad (5)$$

$L = 1{,}34$ s

$T = 6{,}54\ s - L$

$T = 5{,}2\ s$

So, $a = \frac{5{,}2}{1.34} = 3{,}881\ s$

From these calculations, the values for Kp, Ki, Kd, Ti, and Td are obtained using the Ziegler Nichols 1 tuning formula as shown in Table 1.

Table 1. 1st Ziegler Nichols tuning

controller	Kp	Ti	Td
P	1*(T/L)	-	-
PI	0.9*(T/L)	3L	-
PID	1.2*(T/L)	2L	L/2

In Table 1 is the 1st Ziegler Nichols tuning method which produces Kp, Ti and Td values. The results of Ti and Td values are used to find Ki and Kd values using the following mathematical model.

$$\text{Ki} = \text{Kp}.\frac{1}{\text{Ti}} \quad (6)$$

Table 2. Calculation Results with Ziegler Nichols Tuning -1

	Kp	Ki	Kd	Ti	Td	T	L
P	3.881	-	-	-	-	-	-
PI	3.4925	0.8717	-	3.9	-	-	-
PID	4.6257	1.7260	3.0992	2.68	0.67	5.2	1, 34

$$Kd = Kp.Td \quad (7)$$

To get the Ki and Kd values as in Table 3.2 use the formula as in equation 3.2 and formula 3.3. From the calculation results in Table 3.2, the Kp, Ki, and Kd values are 0.05432, 0.02926, 0.01168. After obtaining these parameters, the values will be entered when testing the PID on the *plant* and analysed the graph of the *plant* using the controller. For the time needed for water to seep into the ground to a depth of 5 cm it takes 3 seconds. So that in the PID program used during the test the authors provide time for 3 seconds to activate the PID control until the humidity value reaches *the set point* for plant needs, namely 60%. They should also make adequate reference to accepted methods and identify differences.

IV. Results And Discussions

A. Testing with the On-Off Method

The testing currently, aims to find the appropriate system response to the soil moisture value. The soil moisture value in question refers to the system process flow. So the authors tried testing at 25 steps, 50 steps, 75 steps, 100 steps, 125 steps, 150 steps, 175 steps, 200 steps, 225 steps, 250 steps, 275 steps with a *set point* soil moisture and what discharge is the response of the system according to the desired soil moisture value. System response data at predetermined steps can be seen in Table 3.

Table 3. Experimental Data of On-Off Method

No.	Step	Debit (L/second)	Average *Error* Humidity	*Rise Time(s)*
1.	25	0.002	0.31	377
2.	50	0.004	0.60	366
3.	75	0.008	0.27	265
4.	100	0.013	1, 09	306
5.	125	0.017	1.13	226
6.	150	0.022	3.23	130
7.	175	0.0267	2.92	103
8.	200	0.031	2.82	69
9.	225	0.035	2.77	645
10.	250	3.0	_ 41	68
11.	275	0.044	3.69	52

Results from Table 3. System response data is obtained to reach *set point*, namely by providing a water discharge of 0.044 L/second, *rise time* of 52 seconds is obtained. Meanwhile, for the smallest average *error*, by providing a water discharge of 0.008 L/second, an average *error* of 0.27% is obtained. From this data, the authors can determine the water discharge that produces the *rise time* average *error* smallest This is because the greater the water discharge, the faster the humidity value reaches *the set point* but will experience a high average *error*. And the smaller the water discharge, the longer the humidity value reaches *the set point* and has a low average *error* value. Of the 11 experiments above, the authors used a water discharge of 0.008 L/second to do a comparison of whether automatic irrigation systems get better results using the PID method without or using the PID method.

B. The PID test

In Figure 16 is the result of assigning a PID value using the results from, namely Kp, Ki, Kd of 4.6257, 1.7260, 3.0992. From the graph, the average *error* 0.435%, the rise *time* (Tr) is 96 seconds, and the *overshoot* is 1.11667%. Where this has not reached *rise time* the desired for *overshoot* in the PID test using the value from the results of the Ziegler Nichols 1 tuning is in accordance with what the author wants where this is explained in chapter 3 of the PID design sub-chapter.

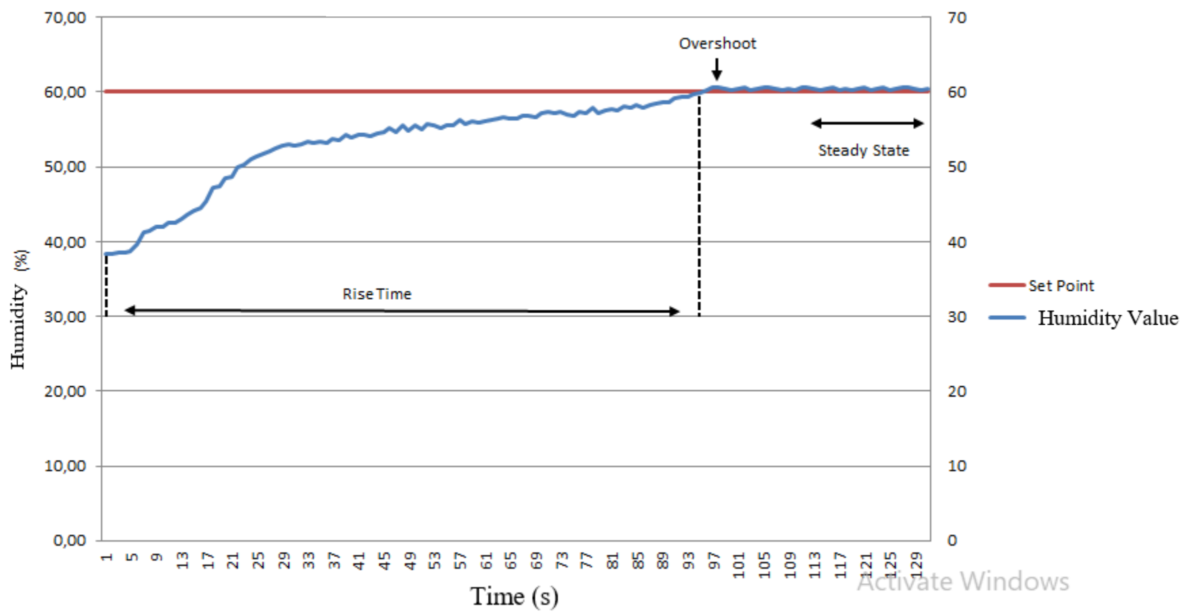

Figure 16. Tuning Results on

C. Manual Tuning

Tuning is determining the Kp, Ki and Kd values by trying various values until a system response is obtained with a *rise time* of no more than 90 seconds and *overshoot* occurring no more than 120 seconds. In Figure 17, 4 attempts were made to assign different values of Kp, Ki and Kd.

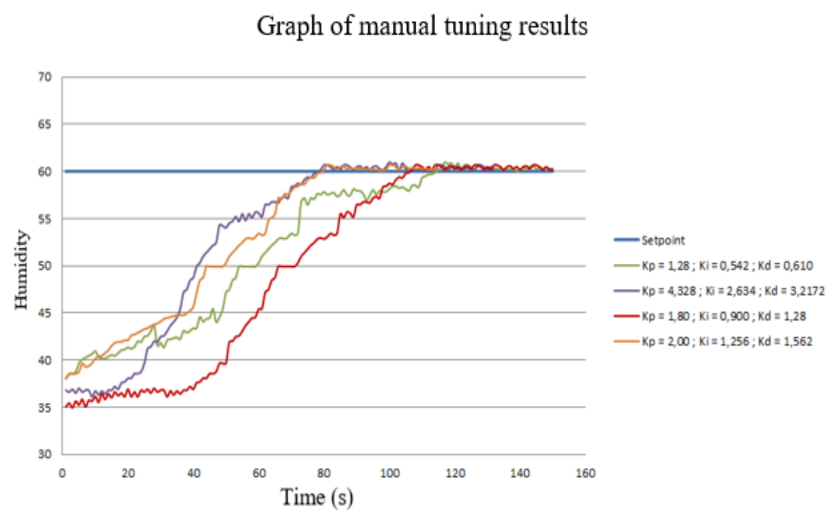

Figure 17. Results of Manual Tuning

Table 4. Kp, Ki, Kd obtained

No	Value of Kp, Ki, Kd	Overshoot (%)	Rise Time (s)	Steady State Error (%)
1	1.28; 0.542; 0.610	-	115	0.332778
2	1.80; 0.900; 1.28	-	107	0.4350
3	2.00; 1.256; 1.562	1.11	80	0.404259
4.	4.328; 2,634; 3.2172	1.61	79	0.387778

D. Plant Growth

Testing the system on plants was carried out for 6 hours in 7 consecutive days to observe the growth of mustard plants. In this case the author tries to observe whether the system that has been created can help relieve farmers in irrigation activities by comparing watering plants using a system with PID control and watering plants manually. To observe plant growth, the authors observed on the first day and the seventh or last day the authors tested the automatic irrigation system. So the authors measured 5 plant heights randomly in the automatic watering area and 5 plant heights randomly in the manual watering area. Then the data obtained in Table 5 and Table 6.

Table 5. Height of First Day Plants

No	Area Watering	Plant Height (cm)					Average (cm)
		Mustard 1	Mustard 2	Mustard 3	Mustard 4	Mustard 5	
1.	Manual Area	14	19.5	9	11	10	12.7
2.	Automatic Area	13.5	6	14	12	15	12.1

Table 6. Plant Height on the Seventh Day

No	Area Watering	Plant Height (cm)					Average (cm)
		Mustard Greens 1	Mustard Greens 2	Mustard Greens 3	Mustard Greens 4	Mustard 5	
1.	Manual Area	28	20.5	12.5	26	24	22.2
2.	Automatic Area	20	26	13	18	27	20.8

From Table 5 and Table 6, the average difference in height of 5 plants in the manual watering area is 9.5 cm and an automatic watering area of 8.7 cm. The effect of automatic irrigation systems to control plant moisture using the PID method does not significantly affect the growth of mustard plants. This can be seen in the difference in plant height on the first day and the seventh day as shown in Table 5 and Table 6. The average growth of plant height in the manual watering area and the automatic watering area has a difference of 1.2 cm. Because this system only focuses on water irrigation where plants also need other nutrients for growth besides water such as organic fertilizers, and other substances and nutrients to support plant growth.

V. CONCLUSIONS AND RECOMMENDATIONS

After testing the entire system in this study, the following conclusions can be drawn into four conclusions. First, Test results using the on-off method by providing several different water discharges obtain a rise time 265 seconds and a steady state error of 0.27% at a water discharge of 0.008 L /second. Second, Using the Ziegler Nichols 1 method steady state error of 0.435%, Rise Time (Ts) of 96 seconds and overshoot of 1.11667% is obtained. Third, by using the manual tuning method overshoot of 1.61% was obtained, a steady state error of 0.387% and a rise time (Tr) of 79 with Kp = 4.328, Ki = 2.634, and Kd = 3.217. The last, Automatic irrigation systems for controlling plant moisture using the PID method do not affected on plant growth rate compared to manual watering method. Because this system only focuses on water irrigation where plants also need other nutrients for growth.

VI. REFERENCES

[1] J. IRIGASI, “Standar Perencanaan Irigasi,” 2013.

[2] M. Barakat, B. Cheviron, and R. Angulo-Jaramillo, “Influence of the irrigation technique and strategies on the nitrogen cycle and budget: A review,” *Agric Water Manag*, vol. 178, pp. 225–238, 2016.

[3] R. K. Saputro, “Peningkatan Efisiensi Irigasi (Studi Kasus: Aset Irigasi di Kabupaten Trenggalek),” Institut Teknologi Sepuluh Nopember, 2017.

[4] N. S. Aeni, “Mengenal Irigasi dari Pengertian sampai Jenisnya,” *Banyumas*, 2021.

[5] M. A. Martono, “DESAIN SISTEM IRIGASI TANAMAN HIDROPONIK PADA UPT PERTANIAN DAN PETERNAKAN TERPADU SMART GREENHOUSE POLITEKNIK NEGERI JEMBER,” 2020.

[6] E. R. Irawan *et al.*, “OTOMATISASI SISTEM IRIGASI PADA TANAMAN CABAI BERBASIS ARDUINO DENGAN PARAMETER KELEMBABAN TANAH,” in *Prosiding Seminar Nasional Terapan Riset Inovatif (SENTRINOV)*, 2021, vol. 7, no. 3, pp. 301–308.

[7] M. I. Ma’arif, F. I. Adhim, and F. Istiqomah, “Implementasi Metode PID untuk Mengontrol Posisi Motor Servo pada Sistem Sortir Berat Adonan,” *Jurnal Teknik ITS*, vol. 10, no. 2, pp. A352–A359, 2021.

[8] R. Salazar, J. C. Rangel, C. Pinzón, and A. Rodr\'\iguez, “Irrigation system through intelligent agents implemented with arduino technology,” 2013.

[9] P. Jain, P. Kumar, and D. K. Palwalia, “Irrigation management system with micro-controller application,” in *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, 2017, pp. 1–6.

[10] S. Thakare and P. H. Bhagat, “Arduino-based smart irrigation using sensors and ESP8266 WiFi module,” in *2018 Second International Conference on intelligent computing and control systems (ICICCS)*, 2018, pp. 1–5.

[11] K. Taneja and S. Bhatia, “Automatic irrigation system using Arduino UNO,” in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017, pp. 132–135.

[12] U. Albab, D. Sucipto, and others, “PEMANFAATAN TEKNOLOGI IRIGASI OTOMATIS BERTENAGA SURYA DI KELOMPOK TANI KEDONDONG KECAMATAN PAGERBARANG KABUPATEN TEGAL,” *Jurnal Pengabdian Mandiri*, vol. 1, no. 2, pp. 175–180, 2022.

[13] K. Ogata and others, *Modern control engineering*, vol. 5. Prentice hall Upper Saddle River, NJ, 2010.