Implementation and Analysis of PID Control System Visually GeoGebra-Based

Yudha Firizky Dwitama
e-mail: yudhaflede12@gmail.com
Solly Aryza Lubis
e-mail: sollyaryzalubis@pancabudi.ac.id
Rahmaniar
e-mail: rahmaniar@dosen.pancabudi.ac.id

Universitas Pembangunan Panca Budi

Abstract
PID (Proportional-Integral-Derivative) control systems are a control method that is widely used in various engineering applications because of its ability to maintain stability and improve system performance. However, understanding the working mechanism of PID is often challenging, especially for learners or users without a strong control background. This study aims to implement and visually analyze the PID control system using GeoGebra software. Through interactive modeling, each controlling component (proportional, integral, and derivative) is visualized separately to show the influence of each on the system's response. In addition, the GeoGebra construction protocol is leveraged to construct dynamic simulations that allow users to adjust PID parameters directly. The results of the study show that GeoGebra is effectively used as a learning and exploration medium for the concept of the PID control system, as well as facilitating the process of visually tuning parameters. Thus, this approach can be an intuitive educational alternative to understanding and analyzing the PID control system.

Keywords: ***PID, Geogebra, Control System, Visualization, Parameter Tuning***

Introduction

Control systems are an important element in various fields of engineering and industry, especially in maintaining the stability and performance of a process or dynamic system. One of the most widely used methods in control systems is the PID (Proportional-Integral-Derivative) controller. This method utilizes a combination of three control actions—proportional, integral, and differential—to minimize the error between the desired value (set point) and the actual output value of the system. PID controllers have proven to be effective in handling various types of systems, both linear and non-linear, and are able to provide stability and good response times in a wide range of operating conditions. However, the process of designing and tuning PID parameters is not a simple matter. This process requires an in-depth understanding of system dynamics as well as skills in parameter analysis and optimization. For students, students, or practitioners who do not have a strong background in systems control, the concept of PID and the system's response to parameter changes is often difficult to understand through mathematical or textual approaches alone.

This is where the role of interactive visualization becomes very important. GeoGebra, a widely recognized open-source software in the field of mathematics, provides a visual and interactive environment that supports real-time exploration of functions, graphs, and simulations. With the ability to integrate dynamic graphics and parameter control, GeoGebra has great potential for use in the simulation and analysis of PID control systems, particularly as a medium for intuitive learning and concept understanding.

This study aims to visually implement the PID control system using GeoGebra, as well as analyze the system's performance based on changes in PID parameters. This process includes simulating a graph of the system's response to interference or changes in the reference value, visually optimizing the PID parameters, and comparison with the system without control or with other control methods. With this approach, it is hoped that users—both academics and practitioners—can gain a better understanding of how PID works and be able to design control systems more effectively and interactively.

Through the use of GeoGebra, this research not only emphasizes the technical aspects of PID control, but also on simplifying concepts through visual representation, so that control systems are no longer a difficult topic to understand, but can be accessed and widely applied.

Problem Formulation

The formulation of the problem in this study:

1. How to implement the PID (Proportional-Integral-Derivative) method in a control system using GeoGebra?
2. How to visually analyze the performance of the PID control system through interactive simulations in GeoGebra?
3. How to optimize PID parameters so that the control system provides a stable and efficient response using GeoGebra?

Leterature Review

1. PID Control System

PID (Proportional-Integral-Derivative) controllers are one of the most widely used control methods in the engineering and industrial world. According to Ogata (2010), PID controllers work by using three main components: proportional (P), integral (I), and derivative (D) to set the error signal between the set point value and the process value (PV). The proportional component functions to respond to the magnitude of the current error, the integral component accumulates the error over time, and the derivative component estimates the direction of the error change. With the combination of these three elements, the system can achieve stability and a good response speed.

2. **PID Parameter Optimization**
   In order for the PID system to work effectively, the parameters Kp (proportional gain), Ki (integral gain), and Kd (derivative gain) need to be adjusted to the characteristics of the system. Seborg et al. (2017) explain that proper tuning of PID parameters will improve system performance, speed up recovery time, and reduce overshoot. Some of the known tuning methods include Ziegler-Nichols, Cohen-Coon, and manual experiments. However, this tuning process can be complex if it is not accompanied by visual aids.

3. **Control System Visualization**
   Understanding the concepts of control systems, especially PID, can be difficult for learners or non-technical users. For this reason, learning media that supports interactive visualization is needed. Ali (2004) showed that visualization using software such as MATLAB can help students understand the system's response to changes in PID parameters. However, the use of MATLAB is quite technical and requires programming.

4. **GeoGebra as a Visualization Tool**
   GeoGebra is an open-source software that is widely used in mathematics and engineering learning due to its ability to create visual and interactive simulations. Suwarno (2021) mentioned that GeoGebra can be used to visualize control phenomena such as signal waves, function graphs, and other interactive processes in real-time. In the context of PID, GeoGebra is able to model the control blocks, respond to error signals, and show the results changes directly when the PID parameters are changed.

Research Methods

The PID control block diagram is generally shown in the following Figure:

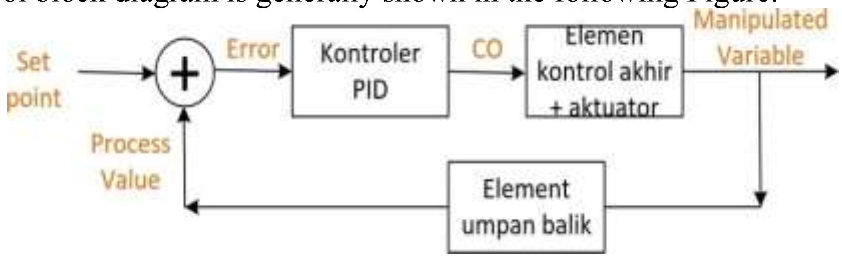

Figure 1. PID Controller block diagram

Description of PID Control Block Diagram:

1. Set Point: The value that the system wants to achieve, such as a certain temperature or speed.
2. Error: The difference between the set point and the current system output value.
3. PID Controller: The part that organizes the system to keep errors as small as possible. PID consists of three parts: P (Proportional): Responds to the magnitude of the error. I (Integral): Addresses persistent errors. D (Derivative): Prevents too fast error changes.
4. System (Plant): A controlled tool or process, such as a motor or heater.
5. Output: The output of the system, e.g. temperature, position, or speed.
6. Feedback: The output is returned to be compared again with the set point so that the system can continue to adjust.

The block diagram of the PID control system consists of two main parts, namely signal and block. The signals used include Set Point (SP) as the desired value, Process Value (PV) as the system output, Error as the difference between SP and PV, Controller Output (CO), and Manipulated Variable (MV). The blocks involved include an enumeration block to calculate errors, a PID controller block, a block of end control elements and actuators, and a feedback block to return the PV value to the beginning of the system.

Each block in this system will be visualized using GeoGebra. Visualization starts from an error signal with two types of actions, namely direct action and reverse action. On the final control block and actuator, it shows how the signal is converted into action against the system. The PID controller block is visualized in stages: the proportional part shows the influence of gain values and bandwidth, the integral part shows the accumulation of errors over time, and the derivative part shows the changes due to SP and PV shifts. With the help of GeoGebra, the entire process in the PID system can be visualized interactively, making it easier to understand and analyze, especially for users who are new to the control system.

Results And Discussion

1. **Error Calculation and SP–PV Relationship in PID System**

In a PID control system, the calculation of errors is a very important first step. Error is calculated from the difference between the Set Point (SP), which is the desired value, and the Process Value (PV), which is the actual output value of the system. There are two types of actions in error calculation, namely direct action and reverse action. In live action, the error is calculated by subtracting the PV value from SP. On the other hand, in the opposite action, the error is calculated by subtracting the SP from the PV. The selection of this type of action depends on the characteristics of the system and the desired direction of control.

The relationship between SP and PV can be visualized to understand the behavior of the system more clearly. SP serves as a fixed reference, while PV will move up and down according to the system's response to the control signal. Both SP and PV are usually expressed in percentages (%), as serving in percent makes it easier for the operator to set and read the values, as well as ensure that the values do not exceed the predetermined limits. In real implementation, minimum and maximum values on SP and PV should be specified to prevent system malfunctions or irregularities. PV values can be overshoot (exceeding the upper limit) or undershoot (falling past the lower limit), which can affect the stability and overall performance of the system. Therefore, setting error values and selecting appropriate control actions are key in keeping the system within safe and responsive limits.

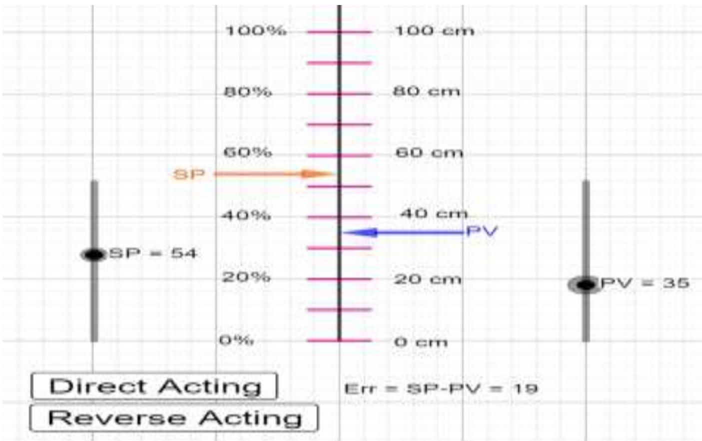

Figure 2. Control Panel for SP and PV

2. **Final Control Elements and Control Signal Mapping**

The final control element is an important part of the control system that functions to translate the output signal from the controller into physical actions that affect the process. This element is usually connected directly to an actuator, such as a valve, motor, or other regulating device, which is in charge of executing commands from the control system.

The output signal from the controller, or commonly called the Controller Output (CO), needs to be mapped into a form that the actuator can understand and execute. This process is referred to as mapping the control signal into manipulated variables (MV). In practice, the actuator setting unit is generally expressed in the form of a percentage (%), for example opening the valve by 75% or regulating the motor speed by 50%. This makes it easy to calibrate and read, and keeps actuator action within safe limits and proportionate to the needs of the system. This mapping must be done precisely so that the actions given by the actuator truly reflect the commands from the PID controller. Errors in mapping can cause the system to be unresponsive or even unstable. Therefore, the end control elements and actuators must be properly calibrated as well as adapted to the characteristics of the controlled system.

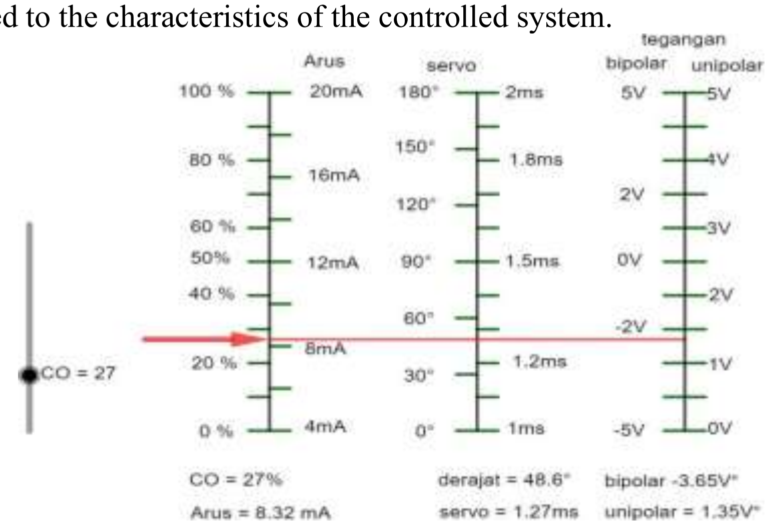

Figure 3. Mapping for CO to be external

3. Mapping of Output Controller (CO) and Proportional Controller

In a PID control system, the output signal from the controller (Controller Output or CO) needs to be mapped to a physical shape that corresponds to the actuator or end device. This mapping depends on the type of actuator used. Some examples of forms of CO mapping include the following: CO in percent × 16 + 4 (e.g. for 4–20 mA signals), CO × 180° (for angle actuators), CO × 1 + 1 (for scales 1–2), CO × 10 – 5 (for a scale of –5 to +5), and CO × 5 (for a scale of 0–5). This mapping ensures that abstract control signals (percentages) can be converted into real action by devices in the field.

In the proportional controller (P) section, the relationship between proportional gain (Kp), error (error), and CO is described in the form: CO = Kp × Error. In this context, CO and error are calculated in percent for ease of visualization. For example, if the error ranges from –10% to +10% and the maximum CO is 100%, then the Kp value can be calculated as the difference of CO divided by the difference of the error, i.e. Kp = ΔCO / ΔError = 100% / 20% = 5. The Proportional Band (PB) value is the opposite of Kp, i.e. PB = 1 / Kp = 1/5 = 20%.

This means that the smaller the PB value, the greater the Kp gain. A narrow PB causes a small error change is enough to produce a large change in the control signal. However, if it is too large, the system can become unstable or oversensitive.

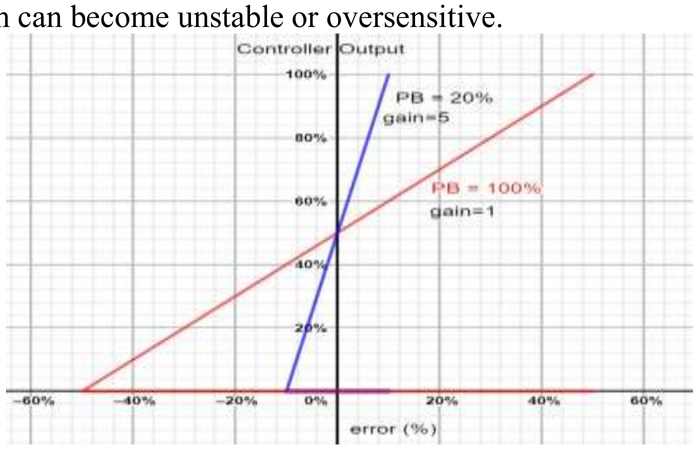

Figure 4. Proportional Band, external error

4. Integral Controller Parts

In the integral part of the PID controller, the controller's output signal is affected by the accumulation of errors over time. The longer the error occurs, the greater the contribution of the integral part to the output. The relationship between the gain integral (Ki), the time integral (Ti), and the output of the controller is explained through the process of integrating into errors. The value of integral time is determined by the formula Ti=1/KiTi = 1/KiTi=1/Ki, so that Ki and Ti have an inverse relationship.

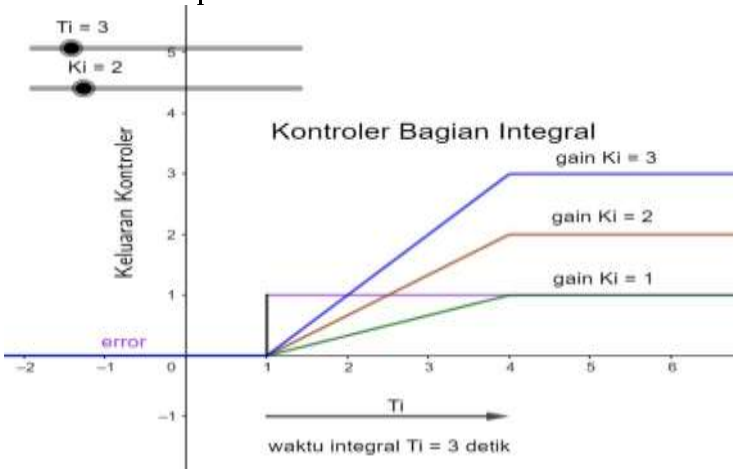

Figure 5. Influence Integral part controller

If the gain integral (Ki) value is large, then the system will respond faster because the errors that occur immediately accumulate and affect the output significantly. Conversely, if the value of Ki is small (or Ti is large), then the contribution of integral parts becomes slower, so the system reacts more slowly to persistent errors. To disable or eliminate the influence of the integral part, the integral time value can be made very large, even towards infinity (∞), so that the accumulation effect does not occur.

Integral units of time generally use the unit of repeat per time. For example, an integral time of 1 minute per repeat means that in one minute, the system will respond once to the error that occurred. In another example, if the integral time Ti is set to 3 seconds and Ki = 1, then in 3 seconds the controller output will be equivalent to the input error. If Ki = 3, then the system will respond 3 times per second or called 3 repeats per second.

However, it is important to note that if the integral part is not restricted, then the controller's output can quickly reach the saturation (saturation) point. Saturation occurs when the addition of errors no longer results in changes to the controller's output. This can cause the system to become unstable or difficult to control, especially in a system that is sensitive to sudden major changes.

5. Derivatives Controller Parts

The derivatives controller functions to provide a quick response to error changes in a short period of time. In contrast to proportional controllers that only respond to large errors and integral controllers that respond to accumulated errors, derivative controllers react to the speed at which errors change over time.

In general, derivative control works based on the first derivative of the error, which is the change of the error per unit of time. If an error changes quickly, then the output of the derivatives section will also provide a great signal to anticipate the change. This is especially useful for damping oscillations and accelerating system stabilization.

For example, if the Set Point (SP) suddenly changes drastically, as in the form of a step function, then the derived value of SP over time (dSP/dt) will form an impulse function—a sharp spike that occurs in a very short period of time. Figure 7 shows the derivative controller's response to this condition, where the output signal immediately spikes in reaction to a sudden change.

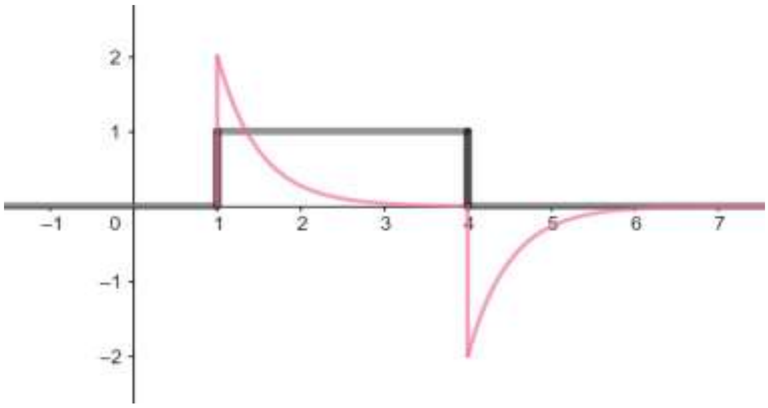

Figure 6. Shock (impulse function) on error changes

Derivative controllers are very sensitive to rapid changes, they can also amplify minor interference or noise, so they should be used with caution. In practice, the derivative part is usually not used alone, but is combined with the proportional and integral part in the PID system to keep the system stable and responsive.

6. Enhanced PID and GeoGebra Controllers

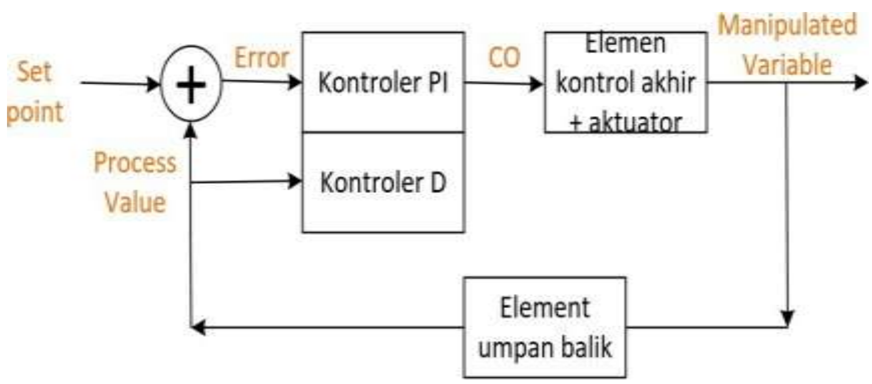

Figure 7. Enhanced PID controller

In a PID control system, a sudden change in the Set Point value can cause a signal spike in the derivative part, called *a derivative kickoff*. To avoid this, the PID controller is refined by differentiating error calculations. The proportional and integral parts still use the error of the difference between SP and PV, while the derivative part only sees changes from PV only, not SP. This makes the system more stable and not easily disrupted by sudden changes.

In addition, offsets are added to match the output signal with devices such as actuators, so that the controls can work correctly.

In visualization in GeoGebra, all of these processes can be described step by step through the Construction Protocol feature. For example, in Figure 8, a sequence of steps is arranged to create an integral controller graph. With GeoGebra, users can see how parameter changes affect the system directly and interactively, making it easier to understand.

	Name	Description	Value
1	Function f		f(t) = If(1 ≤ t ≤ 4, 1, 0)
2	Function g		g(t) = If(1 < t < 4, 2ℯ^(-2 (t - 1))
3	Function h		h(t) = If(4 < t, -2 ℯ^(-2 (t - 4)))
4	Segment i	Segment (1, 0), (1, 1)	i = 1
5	Segment j	Segment (4, 0), (4, 1)	j = 1
6	Segment k	Segment (1, 0), (1, 2)	k = 2
7	Segment l	Segment (4, 0), (4, -2)	l = 2

Figure 8. GeoGebra Construction Protocol

This research has several limitations. First, the implementation of the PID controller is not covered in the form of a programming script. This is because PID scripts are already packaged in the form of libraries [8], so users cannot easily understand the internal processes that occur in the controller. In addition, the visualization used does not support dynamic input changes, so it does not display the system in real-time. However, the construction protocol feature in GeoGebra still allows users to change certain parameters and observe how those changes affect each block in the PID control system.

Conclusion

This study aims to implement and visually analyze the PID control system using GeoGebra. Based on the results of exploration and simulation, the following can be concluded:

Results The implementation of the PID method in GeoGebra can be done by building a block diagram that represents the relationship between Set Point (SP), Process Variable (PV), error, and proportional, integral, and derivative control components. GeoGebra allows the visualization of each component separately, making the control process easier to understand.

Visual analysis of the performance of the PID control system can be done through the dynamic slider feature and construction protocol on GeoGebra. With this feature, changes in error signals, system response, and the effects of each parameter can be observed interactively, helping users to understand the influence of each PID component on system stability.

PID parameter optimization can be done by manually setting proportional (Kp), integral (Ki), and derivative (Kd) gain values via a slider in GeoGebra. This change in parameters can be directly observed to have an impact on the system's response, both in the form of error graphs, system output, and stability and speed of response. This makes it easier to tuning the tuning process to get a stable and efficient PID setup.

References

[1] Anisah, S., Tharo, Z., Hamdani, H., & Butar, A. K. B. (2023). Optimization Analysis Of Solar And Wind Power Hybrid Power Plant Systems. *Prosiding Universitas Dharmawangsa*, *3*(1), 614-624.

[2] Hamdani, H., Sastra, A., & Firmansyah, D. (2023). Kajian Pembangunan Lift Barang Pintar Kapasitas 50 Kg Dengan Pembangkit Listrik Tenaga Surya (PLTS). *INTECOMS: Journal of Information Technology and Computer Science*, *6*(1), 429-433.

[3] Hamdani, H., Tharo, Z., Anisah, S., & Lubis, S. A. (2020, September). Rancang Bangun Inverter Gelombang Sinus Termodifikasi Pada Pembangkit Listrik Tenaga Surya Untuk Rumah Tinggal. In *Prosiding Seminar Nasional Teknik UISU (SEMNASTEK)* (Vol. 3, No. 1, pp. 156-162).

[4] Tharo, Z., Hamdani, H., & Andriana, M. (2019, May). Pembangkit listrik hybrid tenaga surya dan angin sebagai sumber alternatif menghadapi krisis energi fosil di sumatera. In *Prosiding Seminar Nasional Teknik UISU (SEMNASTEK)* (Vol. 2, No. 1, pp. 141-144).

[5] Tharo, Z., Hamdani, H., Andriana, M., & Yusar, J. H. (2022). Implementasi genset ramah lingkungan berbasis panel surya di Desa Tomuan Holbung. *Jurnal Derma Pengabdian Dosen Perguruan Tinggi (Jurnal Deputi)*, *2*(2), 98-101.

[6] Tharo, Z., Syahputra, M. R., Hamdani, H., & Sugino, B. (2020). Analisis Sistem Proteksi Jaringan Tegangan Menengah Menggunakan Aplikasi Etap Di Bandar Udara Internasional Kualanamu. *Journal Of Electrical And System Control Engineering*, *4*(1), 33-42.

[7] Wibowo, P., Lubis, S. A., & Hamdani, Z. T. (2017). Smart home security system design sensor based on pir and microcontroller. International Journal of Global Sustainability, 1(1), 67-73.

[8] Yusup, M. (2022). Teknologi Radio Frequency Identification (RFID) sebagai tools system pembuka pintu outomatis pada smart house. *Jurnal Media Infotama*, *18*(2), 367-373.
[9] Ogata, K. (2010). Modern Control Engineering (5th ed.). New Jersey: Prentice Hall.
[10] Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & Doyle, F. J. (2017). Process Dynamics and Control (4th ed.). Hoboken, NJ: John Wiley & Sons.
[11] Nise, N. S. (2010). Control System Engineering (6th ed.). New Jersey: John Wiley & Sons.
[12] Albertos, P., & Mareels, I. (2010). Feedback and Control for Everyone. Berlin: Springer.
[13] Setiawan, I. (2013). Kontrol PID untuk Proses Industri. Jakarta: Elex Media Komputindo.
[14] Rosada, K. (2017). Sistem Kontrol Pompa Air Menggunakan Kontroler PID Berbasis Raspberry Pi (Skripsi). Institut Teknologi Sepuluh Nopember.
[15] Ali, M. (2004). Pembelajaran perancangan sistem kontrol PID dengan software MATLAB. Jurnal Edukasi@ Elektro, 1(1).
[16] Visioli, A. (2006). Practical PID Control. Berlin: Springer Science & Business Media.
[17] Bista, D. (2016). Understanding and Design of an Arduino-based PID Controller. [Unpublished report].
[18] Suwarno, D. U. (2021). Visualization of Rectified Sine Waves and Triggering Angles on Thyristor Using GeoGebra. Proceeding International Conference on Mathematics and Learning Research.