

Monitoring dan Kontrolling PID pada Greenhouse Strawberry Berbasis Internet of Things (IoT)

Indra Maya¹, Harianto², Musayyanah^{3*} and Pauladie Susanto⁴

^{1, 2, 3*, 4} Universitas Dinamika; maya@dinamika.ac.id, hari@dinamika.ac.id, pauladie@dinamika.ac.id

Abstract: Buah Strawberry (*Fragaria sp.*) merupakan buah yang bernilai ekonomi tinggi. Strawberry tumbuh pada suhu dan kelembapan tertentu. Sehingga, perlu adanya Greenhouse untuk menanam Strawberry. Greenhouse dapat meningkatkan produktifitas hasil panen buah Strawberry pada lahan dataran rendah agar tanaman tidak terpengaruh oleh unsur dari luar. Untuk memenuhi kebutuhan tumbuh Strawberry, maka perlu memperhatikan kestabilan suhu dan kelembapan udara pada Greenhouse. Pengaturan tersebut dikendalikan dengan menggunakan sistem kontrol Proportional, Integral, dan Derivative (PID). Parameter yang dikontrol adalah kestabilan suhu dan kelembapan pada Greenhouse. Sistem kontrol PIDnya menggunakan metode Ziegler-Nicolas tipe 1 dengan penetapan nilai suhu 26,1°C dan kelembapan 80%. Hasil dari penelitian ini untuk parameter suhu mencapai nilai Kp 5,32; Ki 0,22 dan Kd 21,6. Parameter kelembapan mencapai nilai Kp 9,5; Ki 0,25 Kd 4,75. Overshoot suhu 1,15% dan kelembapan 5,88% dengan *rise time* 90 menit dan 40 menit. *Settling time* 190 menit dan 85 menit.

Keywords: *Strawberry, Greenhouse, suhu, kelembapan, PID, Ziegler Nichols, Open Loop*

1. Introduction

Tanaman Strawberry atau *Fragaria sp.* memiliki banyak manfaat untuk kesehatan dan menarik untuk dikembangkan, yaitu kesehatan jantung, mata, kulit dan menekan resiko mengidap kanker. Pembudidayaan Strawberry harus dilakukan di dataran tinggi dengan suhu iklim 17-26°C dan kelembapan 80-90%. Strawberry membutuhkan sinar matahari selama 8-10 jam per hari [1]-[2]. Pembudidayaan Strawberry dapat dilakukan di lahan yang sempit seperti perkotaan dengan menggunakan Greenhouse. Dengan Greenhouse pertumbuhan Strawberry dapat disesuaikan dengan kondisi suhu, cahaya, dan kelembapan yang dibutuhkan Strawberry. Selain itu, Greenhouse dapat melindungi tanaman dari bahaya hama dan penyakit tanaman yang lain [3].

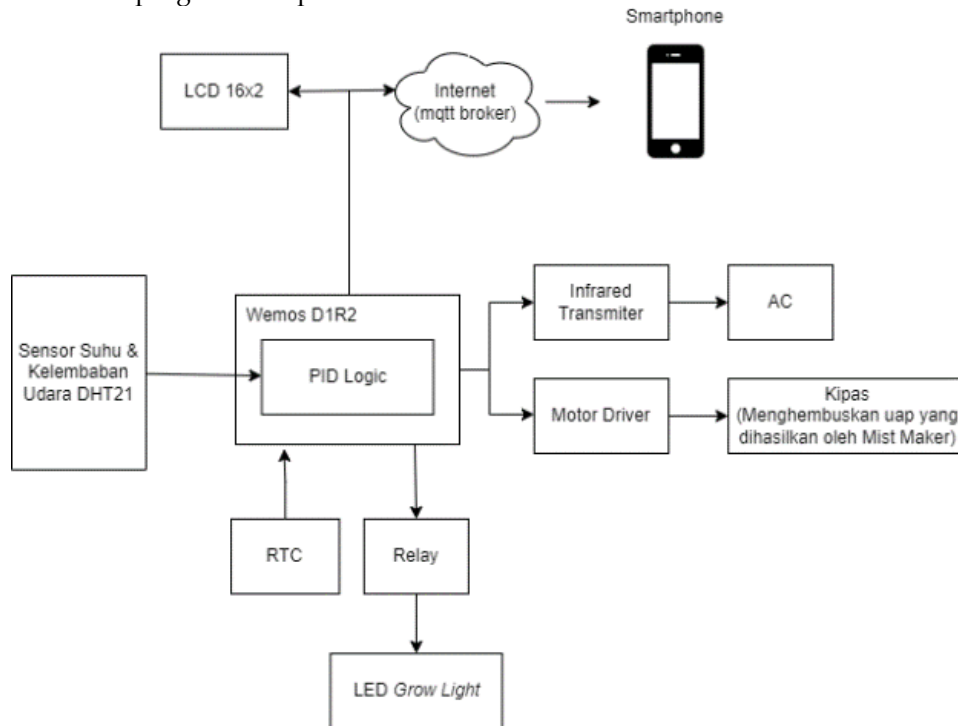
Penelitian mengenai Greenhouse telah banyak dilakukan, salah satunya adalah penelitian pada [4]-[5] menggunakan Fuzzy untuk kontrol suhu dan kelembapan. Namun Fuzzy dianggap kurang stabil dalam mengendalikan suhu dan kelembapan. Sedangkan untuk metode PID dianggap lebih stabil, dengan cara menggunakan nilai *error* dari umpan balik sistem, untuk menentukan nilai kendali berikutnya [6].

Pada makalah ini bertujuan membuat sebuah sistem kontrol PID tipe 1 (Open Loop) untuk mengontrol lingkungan area dalam Greenhouse sesuai dengan ketetapan yang ditentukan. PID dengan Ziegler Nichols pernah diterapkan pada [7], [8], [9], [10]. Parameter yang dikontrol adalah suhu dan kelembapan [11]. Instrumen yang digunakan adalah DHT 21 sebagai sensor suhu dan kelembapan area bagian dalam Greenhouse. Suhu didapatkan dari nyalanya AC dan kelembapan didapat dari Humidifier. Instrumen untuk waktu adalah Real Time Clock (RTC) sebagai penjadwalannya LED Growlight. Led ini digunakan sebagai pengganti fotosintesis tanaman Strawberry. Selain itu, makalah ini menerapkan sistem monitoring untuk mengetahui keadaan Greenhouse dan mengirimkan hasil pembacaan sensor secara *real time*, menggunakan Wemos D1 R2.

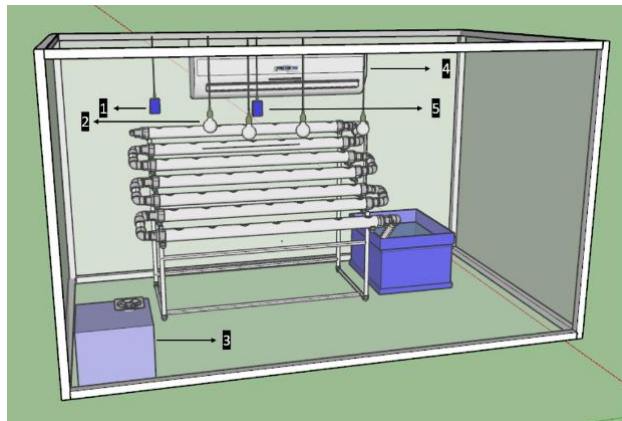
2. Materials and Methods

2.1. Metode Penelitian

Metode penelitian pada makalah ditunjukkan pada Gambar 1 yaitu model rancangan keseluruhan sistem. Sedangkan Gambar 2 adalah desain dari Greenhouse. Bagian ini membahas mengenai proses keseluruhan sistem program dan proses control PID.



Gambar 1. Metode penelitian



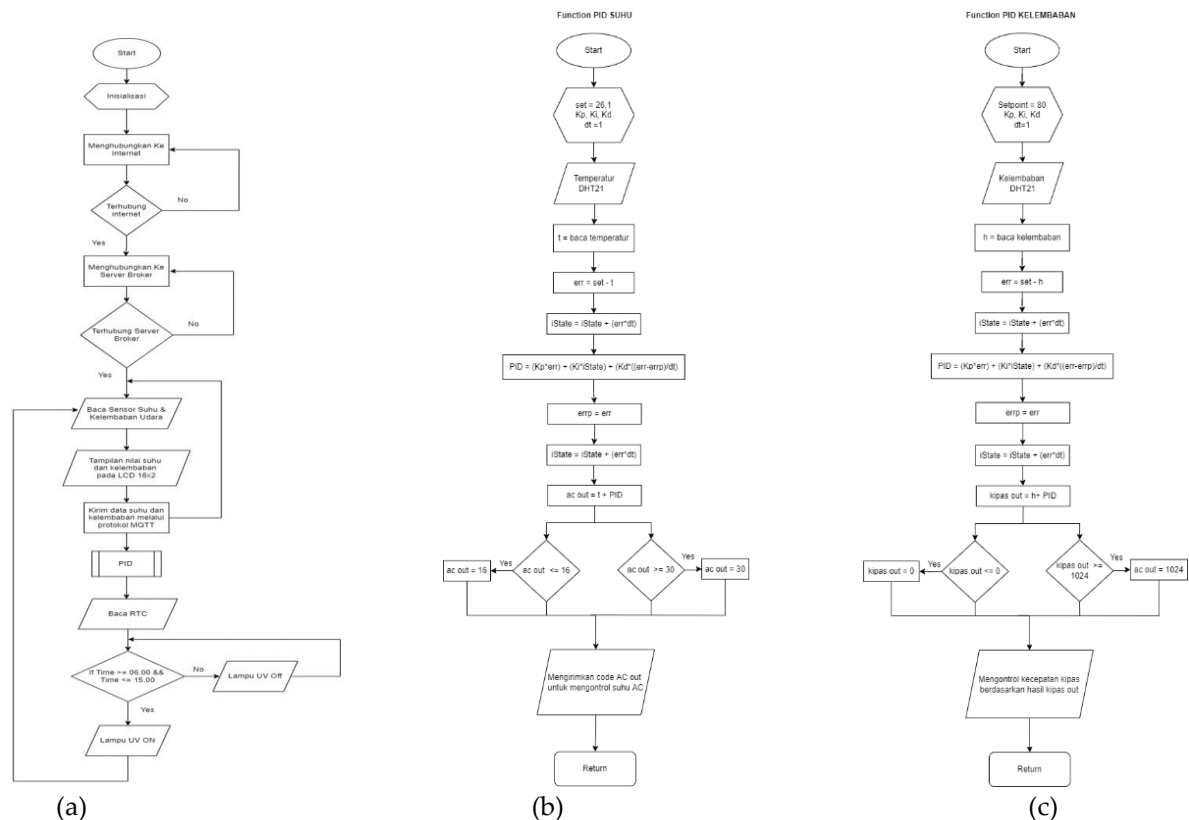
Gambar 2. Desain Greenhouse untuk Tanaman Strawberry

Berdasarkan Gambar 1 terdiri dari input, output dan proses yang dijelaskan sebagai berikut:

1. Input terdiri dari sensor DHT 21 dan RTC.
 - a. Sensor DHT21: Sensor ini berfungsi untuk mengukur suhu dan kelembaban pada ruangan Greenhouse.
 - b. RTC: komponen ini berfungsi mengatur waktu nyala UV.
2. Proses menggunakan mikrokontroler Wemos D1 R2 dilanjutkan dengan proses control PID.
 - a. Wemos D1 R2: Alat ini berfungsi untuk mengontrol input dan output pada sistem Greenhouse.

- b. Logika PID: Pengendali PID digunakan untuk menghitung perbandingan nilai dari input yang diterima dengan menggunakan parameter Kp, Ki, dan Kd yang telah ditentukan sehingga mengeluarkan output sesuai apa yang diinginkan.
3. Output terdiri beberapa komponen yang mengatur proses input serta pada output terdapat koneksi MQTT untuk monitoring IoT nya.
 - a. Relay: Komponen ini berfungsi sebagai saklar.
 - b. Infrared: Komponen ini untuk mengatur suhu AC.
 - c. AC: Alat ini berfungsi untuk mendinginkan ruangan.
 - d. Motor Driver: Sebagai pengatur PWM kecepatan kipas.
 - e. Kipas DC 12V: Mengeluarkan uap hasil dari Mistmaker.
 - f. LED Grow Light: berfungsi sebagai pengganti sinar matahari untuk proses fotosintesis pada tanaman.
 - g. Smartphone (MQTT): berfungsi untuk monitoring kondisi tanaman Strawberry dari jarak jauh.
 - h. LED 16x2: berfungsi untuk menampilkan kondisi tanaman Strawberry.

Berdasarkan komponen sebagai input dan output diletakkan di dalam Greenhouse, sesuai pada Gambar 2, Gambar tersebut terdapat posisi keberadaan sensor yang digunakan. Desain greenhouse digunakan untuk penanaman dan pembibitan tanaman hidroponik strawberry berbasis NFT. Selain itu, untuk menjaga suhu dari Greenhouse membutuhkan suhu ruang menggunakan AC. Sedangkan untuk kelembabanya menggunakan Humadifier. Kemudian, terdapat lampu UV sebagai pengganti cahaya untuk proses fotosintesis tanaman Strawberry. Peletakan sensor suhu di sebelah kanan kiri dari letak AC.



Gambar 3. (a) Alur keseluruhan sistem (b)-(c) Alur dari proses PID

Flowchart Gambar 3 (a) menunjukkan alur keseluruhan sistem yang diawali dengan inialisasi variabel. Kemudian dilanjutkan dengan pembacaan sensor DHT21, yang menghasilkan nilai suhu dan kelembaban udara. Nilai tersebut diproses menggunakan metode kontrol PID dan dapat mengatur AC dan kipas agar suhu dan kelembaban udara menjadi stabil. Setelah itu, RTC (Real Time Clock) diatur untuk jam 06.00 -15.00 lampu UV akan nyala. Ketika diluar jam nyala, maka lampu UV akan mati dan kembali ke pembacaan waktu.

Pada Gambar 3 (b)-(c) merupakan flowchart dari sistem PID suhu dan kelembaban udara, dimulai dari awal program melakukan inisialisasi variabel. Perhitungan PID suhu dan kelembaban diawali dengan nilai *error* dari pengurangan *setpoint* (variabel set) dengan pembacaan sensor suhu (variabel t). Kemudian nilai *error* tersebut dihitung menggunakan integral PID dengan proses diskrit seperti pada persamaan (1) menggunakan penamaan variabel *iState*. *iState* didapatkan dengan menjumlahkan *iState* sebelumnya dengan nilai *error* saat ini yang dikalikan dengan *dt*.

Proses perhitungan rumus PID dengan cara nilai *Kp* dikali dengan nilai *err*, ditambah nilai *Ki* dikali dengan nilai *iState* kemudian ditambah dengan nilai *Kd* dikali dengan selisih *error* saat ini (variabel *err*) dengan nilai *error* sebelumnya (variabel *errp*). Nilai variabel *errp* diawal diberi nilai 0 karna dianggap bahwa pada saat sistem dimulai tidak ada nilai *error*. Kemudian nilai *errp* diperbarui dengan nilai sama dengan *err*. Setelah mendapatkan nilai PID maka nilai suhu (variabel *ac out*) dan kelembaban (variabel *kipas out*) dapat diketahui dengan cara nilai suhu (variabel *t*) dan kelembaban (variabel *h*) saat ini di jumlahkan dengan nilai hasil PID.

Pada suhu, jika nilai *ac out* di bawah 16 °C maka nilai *ac out* sama dengan 16°C karna nilai AC minimum adalah 16°C, jika nilai *ac out* di atas 30°C maka nilai *ac out* sama dengan 30°C karna maksimum nilai AC yang digunakan adalah 30°C. Setelah mendapatkan nilai akhir dari *ac out*, Infrared Transmitter akan mengirimkan kode dari *ac out* untuk menyetel suhu dari AC pada ruangan Greenhouse. Pada kelembaban, setelah mendapatkan keputusan dari perhitungan PID, hasil tersebut kemudian menjadi nilai yang dapat mengatur kecepatan kipas dengan nilai PWM yang didapatkan dari hasil *kipas out* dimana putaran kipas paling rendah 0 rpm dan paling tinggi 1024 rpm, jika nilai *kipas out* lebih kecil dari 0 maka *kipas out* = 0 dan jika *kipas out* lebih besar dari 1024 maka nilai *kipas out* = 1024.

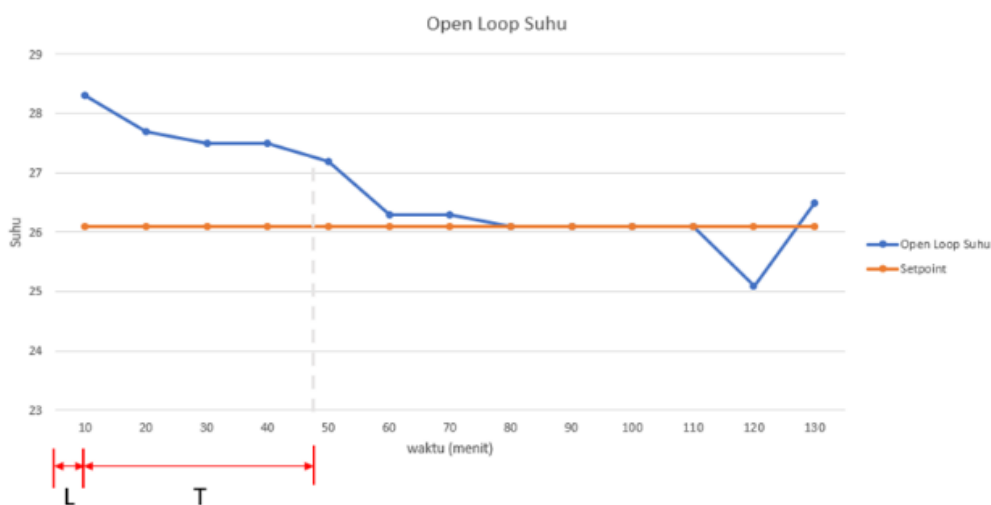
$$u(t_k) = K_p e(t_k) + K_i \Delta t \sum_{i=1}^k e(t_i) + \frac{K_d}{\Delta t} (e(t_k) - e(t_{k-1})) \quad (1)$$

Keterangan:

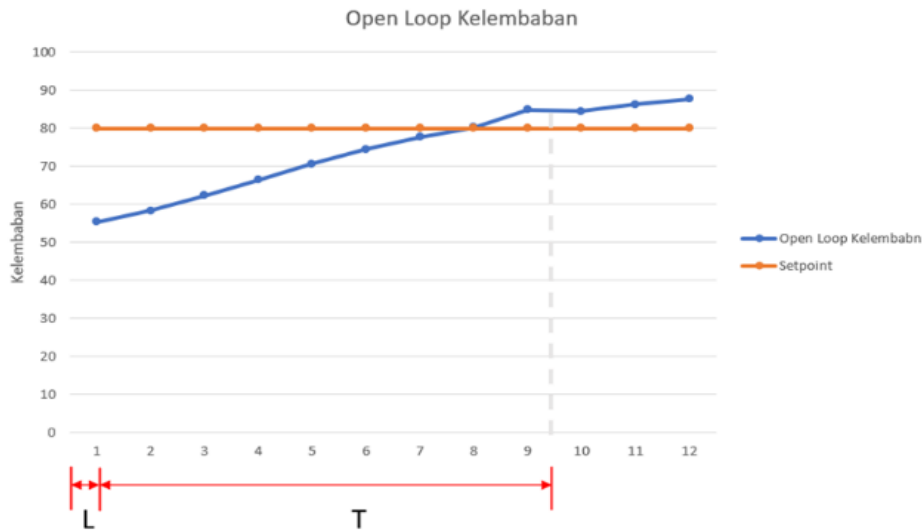
- $e(t_k)$: *error* sekarang.
- $\Delta t \sum_{i=1}^k e(t_i)$: jumlah *error*.
- $e(t_{k-1})$: *error* sebelumnya.

2.2. Metode Zieger-Nichols untuk menentukan nilai *Kp*, *Kd*, *Ki*

Metode Zieger-Nichols yang diterapkan adalah tipe 1 Open Loop. Sistem kontrol yang stabil bisa diperoleh, dengan cara mengetahui kestabilan sistem, dengan mendapatkan nilai penguatan. Gambar 4 dan Gambar 5 merupakan kurva uji open loop sistem terhadap input fungsi step dengan garis tangen.



Gambar 4. Uji Loop untuk suhu



Gambar 5. Uji Loop pada Kelembaban

Berdasarkan Gambar 4 dan Gambar 5 dihasilkan nilai parameter suhu $L = 10$ dan $T = 36,2$ dan kelembaban $L = 1$ & $T = 9,5$. Selanjutnya, nilai tersebut diproses ke dalam rumus Ziebler-Nichols tipe 1 dan akan memperoleh nilai-nilai parameter PID sesuai dengan perhitungan sebagai berikut:

- Perhitungan Parameter suhu

$$K_p = 1,2 \times T/L = 1,2 \times 36,2/10 = 4,32$$

$$T_i = 2 \times L = 2 \times 10 = 20$$

$$T_d = 0,5 \times L = 0,5 \times 10 = 5$$

$$K_i = K_p/T_i = 4,32/20 = 0,216$$

$$K_d = K_p \times T_d = 4,32 \times 5 = 21,6$$

- Perhitungan Parameter kelembaban

$$K_p = 1,2 \times T/L = 1,2 \times 9,5/1 = 9,5$$

$$T_i = 2 \times L = 2 \times 1 = 2$$

$$T_d = 0,5 \times L = 0,5 \times 1 = 0,5$$

$$K_i = K_p/T_i = 9,5/2 = 0,25$$

$$K_d = K_p \times T_d = 9,5 \times 0,5 = 4,75$$

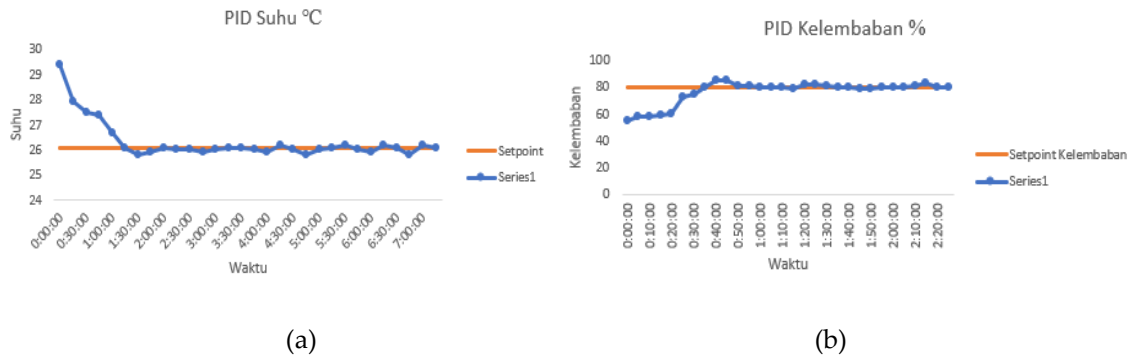
3. Hasil dan Pembahasan

Hasil dan pembahasan pada makalah penelitian meliputi hasil pengujian dengan PID dan hasil monitoring menggunakan MQTT, pengujian sensor, dan pengujian nyala lampu UV. Sistem kendali PID dapat berjalan baik dengan nilai K_p , K_d dan K_i tertentu untuk parameter suhu dan kelembaban. Begitu halnya dengan pengujian MQTT parameter yang dikirimkan lewat MQTT adalah suhu dan kelembaban. Pengujian MQTT hanya menguji keberhasilan pengiriman data. Pengujian sensor hanya keberhasilan pembacaan suhu, dan pengujian UV hanya pengujian nyala matinya lampu UV.

3.1. Pengujian Sistem Otomasi dengan PID

Pengujian ini bertujuan untuk mengetahui seluruh komponen berjalan dengan baik dan mengatur suhu, kelembaban, dan lampu UV. Pada pengujian kestabilan diterapkan medel Low Temperature Long Time. Pada model ini, membutuhkan suhu ruang $26,1^\circ\text{C}$ selama 2,5 jam dan kelembaban ruangan 80% selama 11 menit. Berdasarkan data dari nilai suhu dan kelembaban

dihasilkan respon dari AC dan kipas untuk mencapai titik stabil, perlu menerapkan metode Ziegler Nichols tipe 1 untuk nilai K_p , K_i , K_d . Untuk suhu didapatkan $K_p = 4,32$ $K_i = 0,216$ dan $K_d = 21,6$ dan untuk kelembaban didapatkan $K_p = 9,5$ $K_i = 0,25$ $K_d = 4,75$. Teknik pengujian dilakukan dengan cara mengatur nilai suhu dan kelembaban yang berbeda-beda terhadap *setpoint* suhu $6,1^\circ\text{C}$ dan *setpoint* kelembaban 85%.

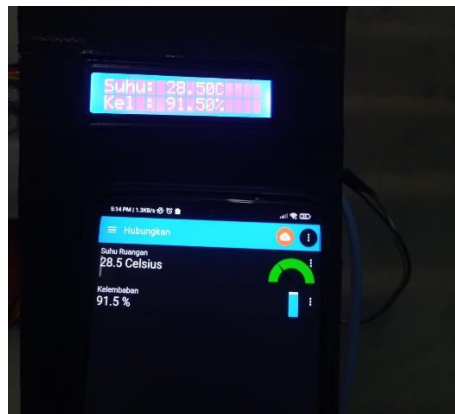


Gambar 6. Grafik PID (a) suhu (b) kelembaban

Berdasarkan hasil percobaan pada Gambar 6, pengujian PID dapat dikatakan stabil menggunakan nilai K_p 4,32 K_i 0,22 K_d 21,6 untuk suhu. K_p 9,5 K_i 0,25 K_d 4,75 untuk kelembaban. Sistem bekerja dengan baik pada suhu mencapai *rise time* 90 menit *overshoot* 1,52% dan *setting time* 180 menit. Sedangkan untuk kelembaban mencapai *rise time* 40 menit *overshoot* 1,77 dan *settling time* 85 menit.

3.2 Pengujian IoT dengan MQTT

Monitoring pada Greenhouse untuk Strawberry pada jarak jauh menggunakan MQTT. Data yang dikirim dan dipantau langsung adalah suhu dan kelembaban. Pada Tabel 1 ditunjukkan hasil berupa keberhasilan pengiriman data sensor ke media penyimpanan broker milik MQTT.



Gambar 7. Perbandingan broker MQTT dengan LCD

Pada Gambar 7 menunjukkan hasil pembacaan sensor dengan LCD dan data yang diterima pada broker MQTT. Tabel 1 memonitoring suhu dan kelembaban udara pada aplikasi broker MQTT, pengujian ini dilakukan selama 13 menit dengan rata-rata *error* suhu dan kelembaban sebesar 0%. Hasil percobaan ini dianggap monitoring menggunakan broker MQTT dapat digunakan karena menghasilkan *error* yang kecil.

Tabel 1. Hasil Pengujian MQTT

No	Suhu (°C)		Kelembaban (%)		Error	Error
	LCD	Smart phone	LCD	Smart phone	Suhu (%)	Kelembaban (%)
1	28,9	28,9	94,8	94,8	0	0
2	28,8	28,8	92,1	92,1	0	0
3	28,9	28,9	91,9	91,9	0	0
4	28,7	28,7	91,9	91,9	0	0
5	28,7	28,7	91,9	91,9	0	0
6	28,7	28,7	91,9	91,9	0	0
7	28,6	28,6	91,7	91,7	0	0
8	28,6	28,6	91,7	91,7	0	0
9	28,6	28,6	91,7	91,7	0	0
10	28,5	28,5	91,7	91,7	0	0
	Rata-Rata				0	0

3.2 Pengujian Sensor Suhu

Pengujian ini bertujuan untuk mengetahui perbedaan nilai suhu dan kelembaban dari alat pembanding Argometer dengan nilai suhu dan kelembaban yang dibaca oleh sensor DHT21. Pada pengujian sensor DHT21 diletakkan diatas tanaman dengan jarak 100 cm. Adapun hasil pengujian pada Tabel 2.

Tabel 2. Hasil Pengujian Sensor Suhu

No.	Suhu (°C)		Kelembaban (%)		Error	Error
	DHT21	Termometer	DHT21	Hygrometer	suhu (%)	Kelembaban (%)
1	28,9	29,3	90	84	1,4	7
2	29,3	29,3	90	84	0	7
3	28,1	28,6	92	86	1,7	7
4	29,1	28,6	90	86	1,7	5
5	28,2	28,7	90	86	1,7	5
6	28,4	28,7	89	86	1	3
7	28,4	28,8	89	86	1,4	3
8	29,3	28,8	89	85	1,7	5
9	28,5	28,8	88	85	1	4
10	29,4	28,8	88	85	2,1	4
	Rata - rata				1,0	6

Berdasarkan Tabel 2 terdapat perbedaan pembacaan sensor DHT21 dengan alat ukur Hydrometer. Hasil rata-rata error suhu sebesar 1% dan error kelembaban sebesar 6%. Perbedaan error ini masih dikatakan akurat, karena masih di bawah 10%.

3.3 Pengujian Lampu UV berdasarkan RTC

Sebelum melakukan pengujian lampu UV, maka perlu dipastikan sensor RTC terkalibrasi dengan waktu internet pada *Smartphones*. Gambar 8 menunjukkan waktu pada RTC pada LCD sama dengan waktu yang ada pada *Smartphones*. Pengujian kondisi lampu UV berdasarkan RTC bertujuan mengetahui nyala mati lampu UV sesuai dengan RTC. Skenario pengujian ini dilakukan dengan cara lampu UV akan mati pada angka menit genap dan akan menyala pada angka menit ganjil.



Gambar 8. Sinkronisasi waktu RTC dengan waktu internet.

Berdasarkan hasil percobaan pada Tabel 3 mengatur kondisi lampu UV berdasarkan RTC (*Real Time Clock*). Pengujian masing-masing dilakukan 1 menit, rata-rata pengujian menghasilkan keberhasilan 100%.

Tabel 3. Pengujian UV berdasarkan RTC

No	RTC	Kondisi Sesungguhnya	Kondisi Lampu UV	Error (%)
1	0:00:00	Mati	Mati	0
2	0:01:00	Menyala	Menyala	0
3	0:02:00	Mati	Mati	0
4	0:03:00	Menyala	Menyala	0
5	0:04:00	Mati	Mati	0
6	0:05:00	Menyala	Menyala	0
7	0:06:00	Mati	Mati	0
8	0:07:00	Menyala	Menyala	0
9	0:08:00	Mati	Mati	0
10	0:09:00	Menyala	Menyala	0
Rata-rata				0

5. Kesimpulan

Berdasarkan analisis dan hasil percobaan maka dapat disimpulkan bahwa penerapan sistem kendali PID dengan metode Ziegler-Nichols tipe 1 mendapatkan nilai K_p , K_i , dan K_d untuk suhu K_p 4,32 K_i 0,22 K_d 21,6 kelembaban K_p 9,5 K_i 0,25 K_d 4,75. Pengujian sistem otomasi didapatkan nilai respon sistem terhadap pengaturan temperatur dan kelembaban dengan *overshoot* sebesar 1,15 % dan 5,88% , *rise time* dengan waktu 90 menit dan 40 menit, dan *settling time* selama 180 menit dan 85 menit. Pada pengujian sistem monitoring untuk suhu dan kelembaban udara menggunakan protokol

MQTT berhasil dengan hasil *error* sebesar 0% untuk suhu dan *error* kelembaban 0%. Pengujian kinerja sensor suhu mencapai 1% untuk deteksi suhu, 6% untuk deteksi kelembaban. Selain itu, kinerja dari RTC dan Lampu UV berkerja dengan baik dimana *error*-nya mencapai 0%.

6. Referensi

- [1] Allert Benedicto Ieuan Noya, "Jus Strawberry dan Manfaatnya untuk Kesehatan," *Alodokter*, 2019. <https://www.alodokter.com/jus-strawberry-dan-manfaatnya-untuk-tubuh>.
- [2] F. Giampieri, S. Tulipani, J. M. Alvarez-Suarez, J. L. Quiles, B. Mezzetti, and M. Battino, "The strawberry: Composition, nutritional quality, and impact on human health," *Nutrition*, vol. 28, no. 1, pp. 9–19, 2012, doi: 10.1016/j.nut.2011.08.009.
- [3] R. A. J. Pratama, "Tanaman Stroberi : Klasifikasi, Ciri Morfologi, Manfaat, dan Cara Budidaya," 2020. <https://dosenpertanian.com/tanaman-stroberi/>.
- [4] F. Lafont and J. F. Balmat, "Optimized fuzzy control of a greenhouse," *Fuzzy Sets Syst.*, vol. 128, no. 1, pp. 47–59, 2002, doi: 10.1016/S0165-0114(01)00182-8.
- [5] M. C. Subin, A. Singh, V. Kalaichelvi, R. Karthikeyan, and C. Periasamy, "Design and Robustness Analysis of Intelligent Controllers for Commercial Greenhouse," *Mech. Sci.*, vol. 11, no. 2, pp. 299–316, 2020, doi: 10.5194/ms-11-299-2020.
- [6] I. Puspasari, Y. Triwidyastuti, and H. Harianto, "Otomasi Sistem Hidroponik Wick Terintegrasi pada Pembibitan Tomat Ceri," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 1, 2018, doi: 10.22146/jnteti.v7i1.406.
- [7] K. J. Åström and T. Hägglund, "Revisiting the Ziegler-Nichols step response method for PID control," *J. Process Control*, vol. 14, no. 6, pp. 635–650, 2004, doi: 10.1016/j.jprocont.2004.01.002.
- [8] Handy Wicaksono and Josaphat Pramudijanto, "Kontrol PID Untuk Pengaturan Kecepatan Motor DC Dengan Metode Tuning Direct Synthesis," *J. Tek. Elektro*, vol. 4, no. 1, pp. 10–17, 2004, [Online]. Available: <http://puslit2.petra.ac.id/ejournal/index.php/elk/article/view/15881>.
- [9] Z. Jamal, "Implementasi Kendali Pid Penalaan Ziegler-Nichols Menggunakan Mikrokontroler," *J. Inform.*, vol. 15, no. 1, pp. 81–88, 2015.
- [10] F. Suryatini and A. Firasanti, "Kendali P, PI, dan PID analog pada pengaturan kecepatan motor dc dengan penalaan ziegler-nichols," *JREC J. Electr. Electron.*, vol. 6, no. 1, pp. 65–80, 2018.
- [11] N. Allu and S. Salu, "Aplikasi Penalaan Dengan Metode Ziegler Nichols di Perancangan Pengendali PID pada Putaran Motor DC," *Pros. Semin. Nas. Sinergitas Multidisiplin Ilmu Pengetah. dan Teknol.*, vol. 1, no. April, pp. 9–10, 2018, [Online]. Available: <https://jurnal.yapri.ac.id/index.php/semnassmipt/article/view/31>.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

