

Pemanfaatan Teknologi *IoT* Pada *Smart Farming Microgreen* dan Akuisisi Data

Rico Wahyu Laksana^{1*}, Bagus Setya Rintyarna¹, Aji Brahma Nugroho¹

¹ Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Jember
Jl. Karimata No.49, Jember
E-mail: wricco360@gmail.com

Naskah Masuk: 10 Agustus 2023; Diterima: 27 Agustus 2023; Terbit: 28 Agustus 2023

ABSTRAK

Abstrak – *Microgreen* adalah tanaman kecil dengan pertumbuhan lebih lama dan daun yang lebih besar dan hijau. Meskipun memiliki kandungan nutrisi yang tinggi dan pertumbuhan cepat, budidaya *microgreen* memerlukan pemantauan lingkungan yang tepat. Saat ini, banyak budidaya *microgreen* masih dilakukan secara manual dan rentan terhadap kesalahan serta ketidakteraturan perawatan. Berdasarkan hal tersebut maka perlu pemanfaatan *Internet of Things* (*IoT*) untuk mengatasi masalah ini. *IoT* memungkinkan pengiriman data tanpa interaksi manusia dan dapat digunakan untuk kontrol dan pemantauan tanaman secara jarak jauh. Pada penelitian ini, teknologi *IoT* pada *smart farming microgreen* dirancang dengan menggunakan mikrokontroler ESP32 untuk sistem monitor dan kontrol yang memuat parameter suhu ruangan, kelembaban udara, kelembaban media tanam yang dilengkapi dengan kendali *ON/OFF* pada *Cooling System*, LED, *speaker* dan *water pump*. Hasil pengujian yang telah dilakukan menunjukkan bahwa secara keseluruhan, sistem berjalan sesuai dengan yang diharapkan yaitu mempertahankan nilai suhu diantara 25-27°C. Selain itu, ketika mencapai angka 60% maka pompa akan aktif dan mulai membasahi media tanam hingga mencapai angka 40%, serta *scheduling* LED dan MP3 Player juga berjalan sesuai dengan yang diinginkan.

Kata kunci: *Internet of Things, Microgreen, Smart Farming.*

ABSTRACT

Abstract - *Microgreens* are small plants with a longer growth period and larger, greener leaves. Despite their high nutritional content and rapid growth, cultivating *microgreens* requires precise environmental monitoring. Currently, many *microgreen* cultivations are still carried out manually, making them susceptible to errors and irregular care. Based on this issue, the utilization of the *Internet of Things* (*IoT*) is necessary to address these challenges. *IoT* enables data transmission without human interaction and can be used for remote control and monitoring of plants. In this study, *IoT* technology is implemented in *smart farming* for *microgreens* using the ESP32 microcontroller. The system encompasses monitoring and control of room temperature, air humidity, and soil moisture, along with *ON/OFF* controls for the *Cooling System*, LED lights, *speaker*, and *water pump*. The conducted testing demonstrates that overall, the system operates as expected, maintaining the temperature within the range of 25-27°C. Additionally, when reaching 60% humidity, the pump becomes active and moistens the growing medium until reaching 40% humidity. The scheduled operations of the LED lights and MP3 Player also align with the desired outcomes.

Keywords: *Internet of Things, Microgreen, Smart Farming.*

Copyright © 2023 Jurnal Teknik Elektro dan Komputasi (ELKOM)

1. PENDAHULUAN

Microgreen merupakan tanaman kecil seperti kecambah tapi dengan pertumbuhan lebih lama dan memiliki daun lebih besar dan lebih hijau. *Microgreen* memiliki kepopuleran yang meningkat dalam beberapa tahun terakhir karena kandungan nutrisi yang tinggi dan pertumbuhan yang cepat [1]. Namun, budidaya *microgreen* memerlukan pemantauan dan pengendalian lingkungan yang tepat seperti suhu, kondisi kelembaban, dosis pengairan dan pencahayaan untuk mencapai hasil yang optimal. Pada saat ini, sebagian besar pembudidayaan *microgreen* masih dilakukan dengan cara lama atau manual dengan tenaga manusia, dan sering terjadi kesalahan dalam prosesnya seperti jadwal yang tidak tepat serta dosis yang tidak sesuai yang menjadikan pertumbuhan kurang efektif [2]. Disamping itu, faktor kelalaian manusia seperti

malas dan lupa dikarenakan aktivitas atau hal-hal yang lain menjadikan pemeliharannya tidak teratur, dan ketika pemilik sedang jauh dari tanaman juga menjadi hambatan yang menyebabkan pemeliharaan menjadi terabaikan. Berdasarkan pada masalah-masalah diatas, diperlukan sebuah sistem yang dapat diandalkan dalam mengatasi masalah tersebut, mengingat pemanfaatan teknologi saat ini dapat menjadi faktor penting dalam pengembangan di sektor pertanian saat ini yang mana akan berdampak langsung terhadap sektor pangan seperti sistem pemeliharaan tanaman otomatis dengan sistem kontrol dan monitoring pada tanaman dari jarak jauh dengan teknologi yang dikenal dengan *Internet of Things* (IoT).

Internet of Things (IoT) merupakan salah satu perkembangan teknologi yang sedang hangat saat ini. Dengan konsep dimana suatu objek dapat melakukan pengiriman data melalui jaringan internet, tanpa perlu interaksi antar manusia maupun manusia dengan komputer. Banyak sekali manfaat yang dapat diambil dari teknologi tersebut. Selain dapat berguna dalam kemudahan pemeliharannya, juga dapat mengurangi waktu panen dengan menyediakan parameter lingkungan sesuai untuk mengoptimalkan pertumbuhan tanaman. Sebagai sistem kontrol, dapat mengambil keputusan dan tindakan dalam menentukan kapan waktu yang tepat dalam melakukan pengairan, pencahayaan dan penyesuaian suhu ruang. Sebagai sistem *monitoring*, dapat memantau data dari parameter-parameter yang dibutuhkan.

Pada penelitian sebelumnya telah dirancang sistem monitoring dan kontrol batas parameter berbasis *IoT* yang dilakukan oleh Hilmy dkk [3]. Pada penelitiannya yang berjudul "*Rancang Bangun Smart Grow Box Hidroponik untuk Pertumbuhan Tanaman Microgreen Berbasis Internet of Things*" tersebut, telah dibuat suatu sistem yang menggunakan aktuator-aktuator seperti kipas angin dan pompa air yang dapat dikendalikan melalui aplikasi *Blynk* yang terpasang di *smartphone*. Dengan menggunakan metode set point batas parameter lingkungan, aktuator-aktuator tersebut dapat dimonitor dan dikendalikan dari jarak jauh melalui internet. Sistem tersebut dirancang untuk mengatur proses pencahayaan dan pengairan pada pertumbuhan tanaman *microgreen*. Pencahayaan akan berfungsi selama sekitar 12 jam sehari, sementara pengairan akan diatur sesuai dengan batas parameter yang telah ditentukan untuk media tanam.

Berdasarkan hasil kajian permasalahan tentang sistem monitor dan kontrol pertumbuhan tanaman *microgreen* yang telah dilakukan, maka pada penelitian ini dibuat sistem monitor dan kontrol pada *smart farming microgreen* yang memuat parameter suhu ruangan, kelembaban udara, kelembaban media tanam yang dilengkapi dengan kendali ON/OFF pada *Cooling System*, LED, *speaker* dan *water pump*. Aktuator-aktuator pada sistem ini juga dapat dikontrol melalui penjadwalan.

2. KAJIAN PUSTAKA

2.1. *Microgreen*

Microgreen adalah varietas baru produk sayuran yang semakin populer dari waktu ke waktu. Waktu panennya adalah 7-14 hari setelah penanaman yaitu hingga mulai tumbuhnya daun sejati pertama. Keunikan berbagai warna, tekstur, tampilan visual, aroma, dan rasa yang dimiliki oleh sayuran muda ini menjadi daya tarik utama. *Microgreen* memiliki beberapa manfaat yang baik untuk dikonsumsi, terkandung nutrisi yang lebih tinggi daripada sayuran dewasa [4]. *Microgreen* juga dapat dijual sebagai produk segar yang digunakan dalam salad, *sandwich*, atau sebagai *garnish* pada hidangan makanan [5]. Suhu yang diperlukan oleh *microgreen* agar dapat tumbuh dengan optimal yaitu di antara 24–29°C. Jika di luar rentang tersebut maka tanaman *microgreen* mengalami pertumbuhan terhenti hingga kerusakan. Kondisi kelembaban media tanam *microgreen* yang dibutuhkan berkisar 50%. Dari hasil percobaan jika melebihi kebutuhannya (70-80%) *microgreen* tidak akan tumbuh. Begitupun jika kurang (20- 30%) maka *microgreen* tidak akan tumbuh [6].

2.2. ESP32

Pin out pada ESP32 dapat dijadikan sebagai input atau output untuk mengontrol LED, kipas angin DC, atau bahkan menggerakkan motor DC. ESP32 adalah sistem dengan biaya rendah dan memiliki konsumsi daya yang sangat rendah berkat fitur-fitur hemat daya seperti *fine resolution clock gating*, *multiple power modes*, dan *dynamic power scaling* pada *chip* (SoC) dengan kemampuan Wi-Fi dan Bluetooth. ESP32 memiliki inti CPU yang lebih cepat dan kemampuan Wi-Fi yang lebih baik, serta dilengkapi dengan lebih banyak GPIO dan dukungan untuk *Bluetooth Low Energy*. Mikrokontroler ini dirancang untuk perangkat seluler, elektronik yang dapat dipakai, dan aplikasi *IoT* (*Internet of Things*). Meskipun ESP32 tidak menggantikan ESP8266, namun membawa peningkatan di berbagai aspek. CPU-nya mirip dengan ESP8266 yaitu Xtensa® LX6 32-bit, namun dengan inti ganda. ESP32 juga dilengkapi dengan ROM 128KB dan SRAM 416K, serta *Flash Memory* (untuk menyimpan program dan data) sebesar 64MB.

2.3. *Internet of Things*

Internet of Things (IoT) adalah suatu konsep dimana suatu objek memiliki kemampuan untuk mengirim data melalui jaringan internet, tidak memerlukan interaksi antar manusia atau manusia dengan komputer [7]. *IoT* dapat digunakan sebagai alat kontrol, *monitoring*, dan banyak lagi

kegunaannya di era modern ini. *IoT* merupakan kemajuan teknologi yang luar biasa karena mampu meningkatkan kualitas hidup melalui pemanfaatan sensor-sensor dan komponen-komponen yang dapat terhubung ke jaringan dan internet, jadi tidak memandang jarak. Prinsip kerja *IoT* adalah setiap objek harus memiliki alamat *Internet Protocol (IP)* yang berfungsi sebagai identitas dalam jaringan, memungkinkan objek tersebut saling berkomunikasi dengan objek lain dalam jaringan yang sama. Selanjutnya, alamat *IP* dalam setiap objek tersebut akan dihubungkan ke jaringan untuk memungkinkan koneksi dan pertukaran data [8]. Dalam artian sederhana perangkat-perangkat yang terkoneksi internet dan dapat dikontrol dengan mudah juga dapat di sebut sebagai *IoT*.

2.4. Android

Android adalah sebuah sistem operasi *mobile* yang dikembangkan oleh Google, yang merupakan hasil modifikasi dari versi kernel Linux. Sistem operasi ini dirancang khusus untuk perangkat layar sentuh seperti *smartphone* dan tablet. Antarmuka Android umumnya menggunakan manipulasi langsung dengan gerakan sentuh, seperti geser, ketuk, dan cubit, untuk memanipulasi objek di layar. Aplikasi Android, juga dikenal sebagai "*apps*", berfungsi untuk menambahkan fungsionalitas pada perangkat. Biasanya, aplikasi Android ditulis dalam bahasa pemrograman Java menggunakan *Android Software Development Kit (SDK)*. SDK menyediakan seperangkat alat pengembangan lengkap, termasuk *debugger*, pustaka perangkat lunak, *emulator handset*, dokumentasi, sampel kode, dan *tutorial*. Proses pengembangan aplikasi melibatkan pengelolaan kode, data, dan file untuk membentuk sebuah *Android Package (APK)*, yang merupakan file arsip dengan akhiran *.apk*. Satu file APK berisi semua komponen aplikasi Android dan digunakan oleh perangkat Android untuk menginstal aplikasi. Dalam penelitian ini, sebuah antarmuka untuk mengendalikan *Smart Farming Microgreen* dibuat dalam bentuk aplikasi (perangkat lunak) *Android Mobile Apps*. Untuk membuat antarmuka pengguna tersebut, diperlukan aplikasi IDE (*Integrated Development Environment*) Android Studio, yang dapat diunduh langsung pada website resminya.

2.5. Arduino IDE

Software Arduino IDE (Integrated Development Environment) adalah media pemrograman arduino, dimana *software* ini sering di sebut juga dengan nama "*Sketch*" sebagai pen-debug, pengkompilasi dan dapat menulis programnya. *Software Arduino IDE* dalam menuliskan kode programnya digunakan bahasa *C for Arduino* yaitu bahasa C yang telah ditingkatkan untuk memenuhi kebutuhan *Hardware Arduino*. Pada *software* ini dapat menulis program sesuai dengan yang diinginkan karena sudah dipermudah dengan *library* yang dapat dengan mudah diunduh pada menu bagian kiri dan banyak lagi fitur-fitur terbaru yang dapat dengan mudah diakses seiring dengan perkembangan *software* ini.

2.6. Capacitive Soil Moisture Sensor

Sensor merupakan perangkat yang digunakan untuk mendeteksi perubahan dalam lingkungan fisik atau kimia. Sensor mengubah variabel keluarannya menjadi sinyal listrik melalui suatu komponen yang disebut *transduser*. Saat ini, sensor-sensor ini telah dikembangkan dalam ukuran yang sangat kecil, bahkan mencapai orde nano meter. Ukuran yang sangat kecil ini memberikan kemudahan penggunaan dan juga menghemat energi. *Capacitive Soil Moisture Sensor* adalah sebuah perangkat sensor yang digunakan untuk mengukur kelembaban tanah. Sensor ini beroperasi berdasarkan prinsip kapasitansi, sirkuit tercetak yang terdapat pada sensor tersebut dilapisi dengan lapisan pernis untuk menghindari korosi. Sensor ini menghasilkan keluaran berupa tegangan analog dengan rentang nilai 1,2V hingga 2,5V. Penerapan *capacitive soil moisture sensor* untuk penyiraman otomatis sudah dapat dengan mudah dihubungkan dengan *microcontroller* [9].

2.7. Sensor DHT22

Pengukuran suhu (*temperature*) dan kelembaban udara (*humidity*) merupakan parameter yang sering digunakan dalam proses akuisisi data, terutama dalam bidang pertanian. Sensor memiliki peran penting dalam proses ini, karena bertugas mengubah kuantitas data dari alam menjadi bentuk digital yang dapat diproses oleh komputer. Akurasi hasil pengukuran sensor juga menentukan sejauh mana hasil yang diperoleh sesuai dengan pengukuran yang sebenarnya melalui instrumen ukur. Dalam proses akuisisi data, sensor dapat berbentuk komponen diskrit atau terintegrasi dalam satu rangkaian chip IC. Biasanya, untuk mengukur satu jenis parameter, digunakan satu jenis sensor. Meskipun demikian, ada juga sensor yang mampu mengukur dua atau lebih parameter secara simultan, seperti DHT22 sebagai contohnya. Pada sistem yang akan dibuat dalam penelitian ini menggunakan sensor DHT22, dimana yang mana sering menjadi pilihan dalam pengukuran suhu dan kelembaban udara dengan keluaran digital [10]. Dengan kualitas pembacaan yang baik, dapat terbukti dari respon proses akuisisi data yang cepat dan tepat dengan harga yang relatif murah dibanding alat *thermohyrometer*. Menggunakan kapasitor dan termistor untuk mengukur udara disekitarnya [11].

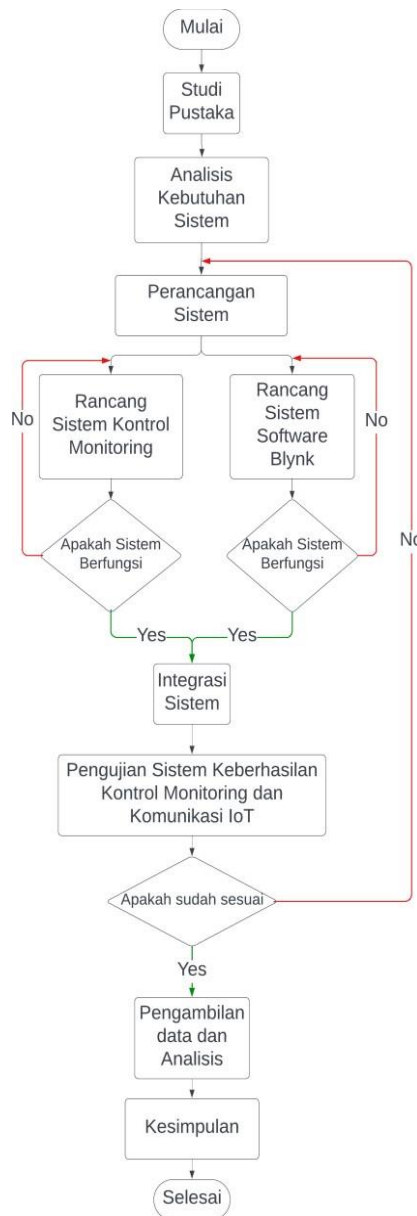
2.8. Blynk Apps

Blynk adalah sebuah aplikasi yang dirancang untuk digunakan pada berbagai perangkat. Aplikasi ini memiliki tampilan yang menarik untuk produk terhubung atau proyek *IoT*. *Blynk* adalah pembuat aplikasi tanpa perlu menulis kode yang sederhana namun kuat juga bisa dengan mudah membuat prototipe, menyebarkan, dan mengelola perangkat elektronik yang terhubung dalam berbagai skala, mulai dari proyek pribadi hingga implementasi jutaan produk. Jika menghubungkan *hardware* seperti ESP32, Arduino, Raspberry Pi, Seeed, Particle, SparkFun, Adafruit, TI, dan berbagai perangkat lainnya ke cloud menggunakan *Blynk*. Selain itu, dapat menggunakan *widget-widget* untuk membangun antarmuka pengguna. Antarmuka ini memungkinkan untuk memvisualisasikan data dari sensor dan mengontrol berbagai perangkat elektronik.

3. METODE PENELITIAN

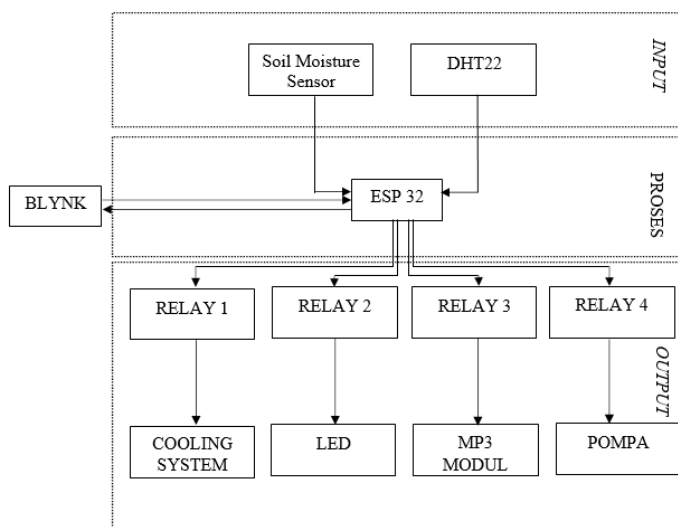
3.1. Alur Penelitian

Perancangan sistem pada penelitian ini terbagi menjadi dua bagian, yakni perancangan *hardware* dan perancangan *software* seperti yang diperlihatkan pada Gambar 1 berikut:



Gambar 1. Flowchart alur penelitian

3.2. Perancangan Hardware
3.2.1 Diagram Blok Sistem

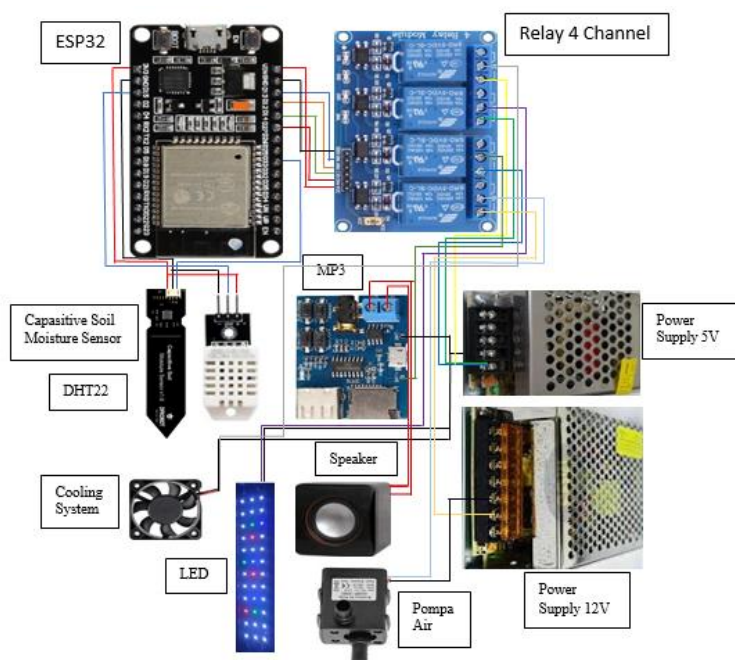


Gambar 2. Diagram blok sistem

Berdasarkan blok diagram sistem pada Gambar 2, sistem yang dirancang menggunakan perangkat berupa soil moisture sensor dan DHT22 sebagai Input, mikrokontroler ESP32 sebagai proses, aktuator- aktuator sebagai output untuk memanipulasi parameter lingkungan sesuai dengan yang dibutuhkan tanaman. Smartphone yang digunakan sebagai alat monitor dan kontrol parameter lingkungan melalui perantara Blynk. Sistem ini menggunakan aktuator-aktuator cooling system, LED, speaker, dan pompa air. Sistem ini menggunakan microkontroler ESP32 yang dapat dihubungkan ke internet karena memiliki modul wifi untuk mengakses Blynk agar dapat dengan mudah memonitor dan kontrol melalui *smartphone*.

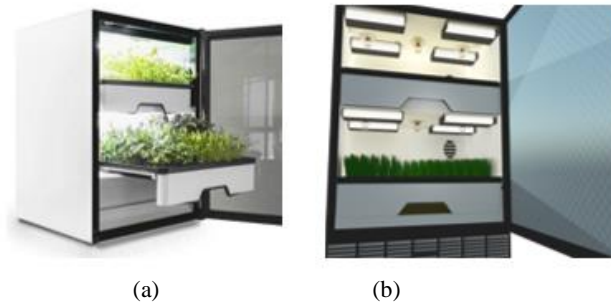
3.2.2 Desain Skematik Keseluruhan Sistem

Gambar 3 berikut menunjukkan desain skematik keseluruhan sistem yang dihasilkan pada penelitian ini:



Gambar 3. Desain skematik alat

3.2.3 Desain Alat

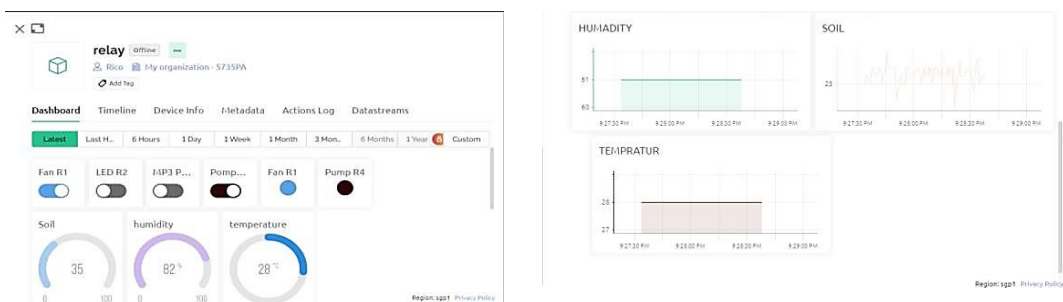


Gambar 4. Gambar alat (a) tampak samping, (b) tampak dalam

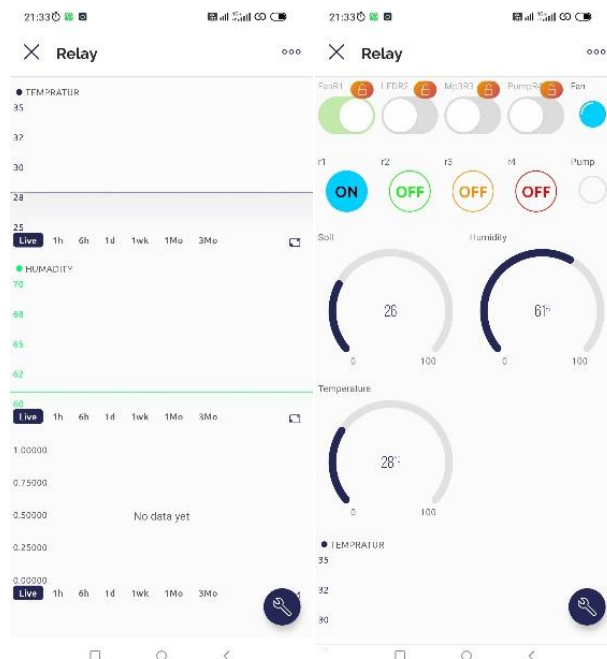
3.3. Perancangan Software

3.3.1 Desain Software

Desain *software* diatur sedemikian rupa agar dapat memudahkan pengguna dalam mengontrol dan *me-monitoring*, juga sudah tersedia widget *Superchart* yang dapat menyimpan hasil pembacaan data sensor dalam bentuk grafik. Grafik tersebut juga memungkinkan pengguna melihat hasil pembacaan sensor dalam kurun waktu tertentu. Desain *software* pada *Blynk Web* dapat dilihat pada Gambar 5 sedangkan pada aplikasi *Blynk Smartphone* dapat dilihat pada Gambar 6.



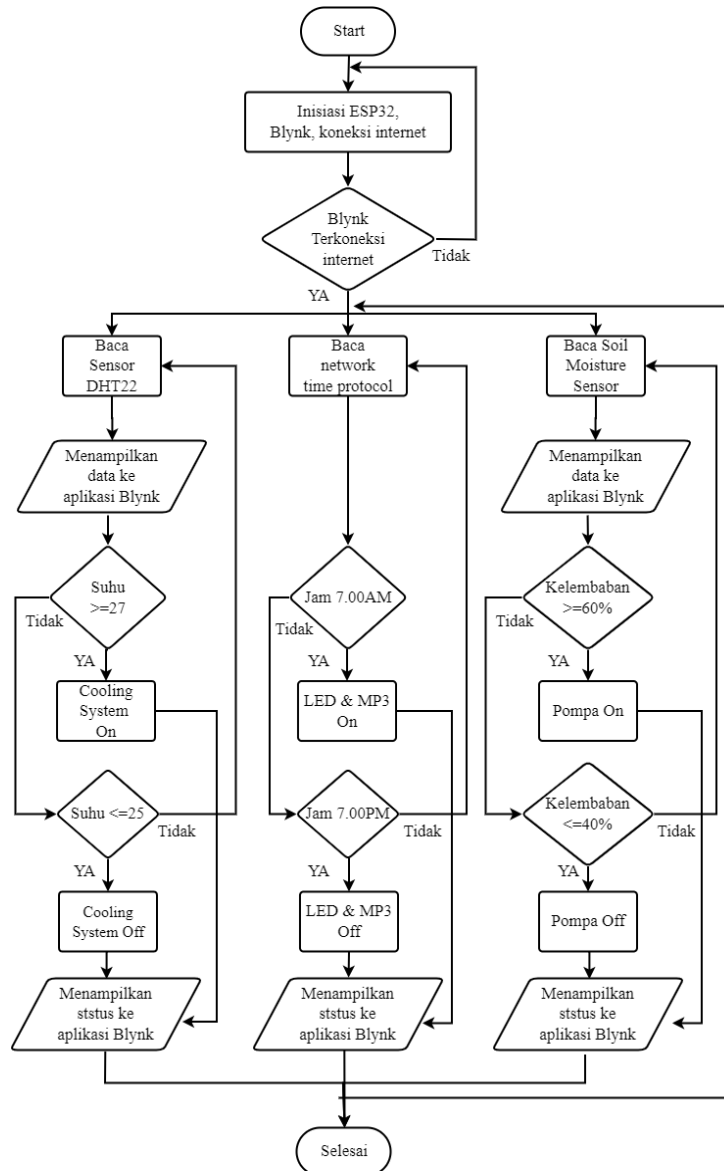
Gambar 5. Desain software pada *blynk web*



Gambar 6. Desain software pada *blynk smartphone*

3.3.2 Flowchart Sistem

Gambar 7 menunjukkan *flowchart* sistem alat yang dihasilkan dari penelitian ini. Pertama yakni inisiasi sistem atau ESP32 beserta koneksi internet. Jika sistem sudah terkoneksi ke internet dan terhubung ke server Blynk, ESP32 akan menerima data sensor yang diInputkan dari sensor suhu dan kelembaban untuk diproses dengan batasan-batasan kondisi nilai sensor seperti yang ada pada *flowchart* diatas, yaitu jika suhu (t) lebih atau sama dengan 27°C maka Fan Cooler dan Fan Airflow akan menyala, kemudian jika suhu sudah memenuhi batas yang telah ditentukan, yakni kurang dari atau sama dengan 25°C maka Fan Cooler dan Fan Airflow akan mati sampai batas yang ditentukan agar suhu tidak melebihi dari batas parameter yang dibutuhkan. Begitu pula dengan pompa, jika kelembaban media tanam lebih dari sama dengan 60% (kering) maka pompa akan menyala, setelah mencapai batas parameter yang ditentukan yakni kurang dari sama dengan 40% (basah) maka pompa akan mati. Untuk penggunaan LED dan MP3 Modul akan menyala secara manual atau sesuai dengan yang telah dijadwalkan.



Gambar 7. *Flowchart* sistem

4. HASIL DAN PEMBAHASAN

Hasil pengujian alat pada penelitian ini terbagi menjadi beberapa bagian, mulai pengujian sensor yang digunakan hingga pengujian alat keseluruhan. Dari hasil pengujian-pengujian ini dapat dianalisa performa-performa dari tiap-tiap rangkaian yang saling terhubung hingga menjadi *box Smart Farming Microgreen*.

Pengujian keseluruhan berguna untuk mengetahui kinerja sistem secara menyeluruh apakah sistem sudah berjalan sesuai dengan baik seperti yang diharapkan.

4.1. Pengujian Power Supply 12V dan 5V

Pengujian *power supply* atau catu daya ini bertujuan untuk mengetahui kinerja dari *power supply* apakah sesuai dengan spesifikasinya dengan mengukur *input* dan *output* dari *power supply* sendiri dan juga dengan menggunakan avometer digital. Hasil pengujian ini ditunjukkan pada Tabel 1 dan Tabel 2.

Tabel 1. Pengukuran *power supply* 12V

No.	Parameter	Tanpa beban		Dengan beban		ΔUkur		Ket
		V	I(mA)	V	I	V	I	
1	Input	236,1	7,10	235	7,09	1,1	0,01	Berhasil
2		236,2	7,07	235,4	7,10	0,8	-0,03	Berhasil
3		236,1	7,08	235,2	7,09	0,9	-0,01	Berhasil
4		236,1	7,09	235,2	7,10	0,9	-0,01	Berhasil
5		236,1	7,06	235,3	7,11	0,8	-0,05	Berhasil
Σ Input		1180,6	35,4	1176,1	35,49	4,5	0,11	
Rata-Rata Input		236,12	7,08	235,22	7,098	0,9	0,022	
1	Output	12,15	0,98	12	0,94	0,15	0,04	Berhasil
2		12,15	0,99	11,9	0,95	0,25	0,04	Berhasil
3		12,15	1,00	12,02	0,96	0,13	0,04	Berhasil
4		12,15	1,01	11,88	0,97	0,27	0,04	Berhasil
5		12,15	1,00	12,02	0,97	0,13	0,03	Berhasil
Σ Output		60,75	4,98	59,82	4,79	0,93	0,19	
Rata-Rata Output		12,15	0,996	11,964	0,958	0,186	0,038	

Error pada *Power Supply* 12V:

$$\text{Error V Input} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,9}{236,12} 100\% = 0,38\%$$

$$\text{Error I Input} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,022}{7,08} 100\% = 0,31\%$$

$$\text{Error V output} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,186}{12,15} 100\% = 1,5\%$$

$$\text{Error V output} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,038}{0,996} 100\% = 3,81\%$$

Tabel 1. Tabel uji *power supply* 5V

No.	Parameter	Tanpa beban		Dengan beban		ΔUkur		Ket
		V	I	V	I	V	I	
1	Input	231,7	7,11	231,2	7,08	0,5	0,02	Berhasil
2		231,6	7,10	230,9	7,09	0,7	0,01	Berhasil
3		231	7,09	230,6	7,08	0,4	0,01	Berhasil
4		230,6	7,09	230,7	7,09	0,1	0	Berhasil
5		231	7,09	230,7	7,08	0,3	0,01	Berhasil
Σ Input		1155,9	35,48	1154,1	35,42	2	0,05	Berhasil
Rata2 Input		231,18	7,096	230,82	7,084	0,4	0,01	Berhasil
1	Output	5,39	0,33	5,38	0,20	0,01	0,13	Berhasil
2		5,40	0,33	5,38	0,25	0,02	0,07	Berhasil
3		5,40	0,33	5,38	0,28	0,02	0,05	Berhasil
4		5,39	0,33	5,38	0,16	0,01	0,17	Berhasil
5		5,39	0,33	5,38	0,19	0,01	0,14	Berhasil
Σ Output		26,97	1,65	26,9	1,08	0,07	0,56	
Rata2 Output		5,394	0,33	5,38	0,216	0,014	0,112	

Error pada *Power Supply* 5V:

$$\text{Error V Input} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,4}{231,18} 100\% = 0,17\%$$

$$\text{Error I Input} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,01}{7,096} 100\% = 0,14\%$$

$$\text{Error V output} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,014}{5,394} 100\% = 0,25\%$$

$$\text{Error V output} = \frac{\Sigma \Delta U_{kur}}{n} 100\% = \frac{0,112}{0,33} 100\% = 0,33\%$$

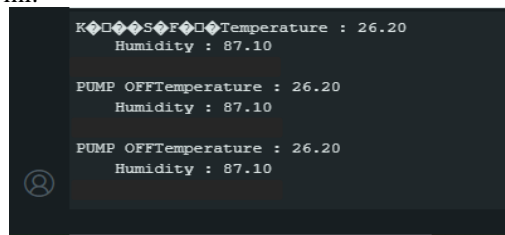
Dari data hasil uji diatas, didapat nilai error yang sangat kecil, dapat diambil kesimpulan bahwa kinerja dari *power supply* tersebut normal dan dapat bekerja dengan baik.

4.2. Pengujian dan running sensor DHT22

Pada pengujian dan *running* sensor DHT22 dilakukan beberapa tahapan untuk memastikan akurasi pembacaan sensor sesuai jika dibandingkan dengan alat ukur standart yaitu seperti dibawah ini:

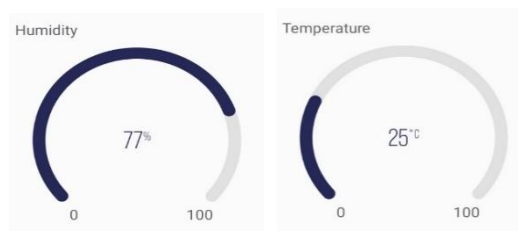
- a. *Running* sensor DHT22 bertujuan untuk mengetahui apakah sensor DHT22 sudah dapat mengukur suhu dengan baik atau tidak, dan mengetahui apakah sensor DHT22 sudah dapat mengirim data dengan benar ke ESP32 yang kemudian akan ditampilkan pada *Blynk*.
- b. *Output running* sensor DHT22

Didapat tampilan *output Running* sensor DHT22 pada “Serial Monitor” *Software* Arduino IDE seperti dibawah ini.



Gambar 8. Tampilan *output* pada serial monitor Arduino IDE

Tampilan *output running* sensor DHT22 pada aplikasi *Blynk* seperti dibawah ini.



Gambar 9. Tampilan *output* pada *blynk*

Gambar-gambar diatas merupakan *output running* sensor DHT22 yang merupakan hasil dari *uploade coding* program ke mikrokontrol ESP32, yang mana dapat diatur sesuai dengan keinginan pada program di *software* Arduino IDE.

- c. Tabel pengujian akurasi suhu dengan DHT22 dan Thermometer Digital

Tabel 2. Tabel uji banding akurasi suhu

No.	Parameter	Input t (°C)	Output sensor	Thermometer	ΔUkur	Ket
1	DHT22	26,7°C	27,1°C	26,7°C	0,4	Berhasil
2		28,3°C	28,5°C	28,3°C	0,2	Berhasil
3		24,7°C	24,9°C	24,7°C	0,2	Berhasil
4		25,4°C	25,3°C	25,4°C	0,1	Berhasil
5		25,2°C	25,2°C	25,2°C	0	Berhasil
Jumlah		130,3	131	130,3	0,9	

Rata-rata	26,06	26,2	26,06	0,18
-----------	-------	------	-------	------

$$\text{Error} = \frac{\sum \Delta U_{kur}}{n} 100\% = \frac{0,18}{26,06} 100\% = 0,6\%$$

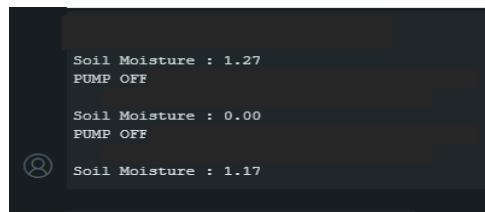
Dari data yang diperoleh dari pengujian perbandingan nilai suhu DHT22 dan *Thermometer Digital* didapat data seperti tabel dan grafik diatas, diperoleh nilai error yang kecil dengan selisih angka yang tidak begitu jauh atau bisa ditoleransi.

4.3. Pengujian dan *running soil moisture sensor*

Pada pengujian dan *running soil moisture sensor* ini dilakukan beberapa tahapan dan tujuan yaitu seperti dibawah ini:

- a. Tujuan pengujian dan *running soil moisture sensor* ini dilakukan untuk mengetahui apakah sensor bekerja dengan baik dan dapat menampilkan data sesuai yang diinginkan.
- b. *Output Running Soil Moisture Sensor*

Didapat tampilan *output running Soil Moisture Sensor* pada Serial Monitor Software Arduino IDE seperti dibawah ini.



Gambar 10. Tampilan *output* pada serial monitor Arduino IDE

Tampilan *output running Soil Moisture Sensor* pada aplikasi *Blynk* seperti dibawah ini.



Gambar 11. Tampilan output pada *blynk*

Gambar-gambar diatas merupakan *output running Soil Moisture Sensor* yang sesuai dengan program *coding* pada *software* Arduino IDE yang sudah diupload.

- c. Berikut tabel pengujian akurasi kelembaban tanah dengan *Soil Moisture Sensor* dan *Moisture Sensor*

Tabel 4. Pengujian akurasi kelembaban tanah

No.	Parameter	Input % (Hg)	Output Sensor	Moisture Sensor	ΔUkur	Ket
1.	<i>Soil Moisture Sensor</i>	35	36,51	35	1,51	Berhasil
2.		39,8	39,54	39,8	0,26	Berhasil
3.		40,1	40,02	40,1	0,08	Berhasil
4.		43	42,59	43	0,41	Berhasil
5.		44,5	45,29	44,5	0,79	Berhasil
	Jumlah	202,4	203,95	202,4	3,05	
	Rata-rata	40,48	40,79	40,48	0,31	

$$\text{Error} = \frac{\sum \Delta U_{kur}}{n} 100\% = \frac{0,31}{40,48} 100\% = 0,76\%$$

Dari data hasil uji pada tabel dan grafik diatas, didapat hasil yang cukup memuaskan, selisih nilai yang terbaca oleh sensor dengan *moisture sensor* tidak jauh dan nilai error yang didapat juga cukup kecil.

4.4. Pengujian dan running mikrokontrol

Pengujian mikrokontrol ini bertujuan untuk mengecek apakah mikrokontrol yang digunakan ada error atau semacamnya yang akan mengganggu kinerja sistem secara keseluruhan. Berikut merupakan beberapa komponen yang termasuk dalam pengujian ini.

4.4.1 Cooling System

Pengujian cooling system ini bertujuan untuk memastikan apakah relay yang terhubung pada cooling system dapat bekerja dengan baik dengan beberapa kondisi sesuai batas parameter pada nilai yang terbaca oleh sensor DHT22 yang sudah diatur, dibawah ini tahap pengujian cooling system tersebut. Didapat tampilan *output running Cooling System* pada Serial Monitor *Software* Arduino IDE seperti dibawah ini.

```
FAN ON
PUMP OFF
Temperature : 27.50
Humidity : 75.20
Soil Moisture : 25.67
FAN ON
PUMP OFF
Temperature : 27.50
Humidity : 75.20
Soil Moisture : 29.23
FAN ON
PUMP OFF
Temperature : 27.50
```

Gambar 12. Tampilan *output* pada serial monitor Arduino IDE

Tampilan *output Running Cooling System* pada aplikasi *Blynk* seperti dibawah ini.



Gambar 13. Tampilan *output* pada *blynk*

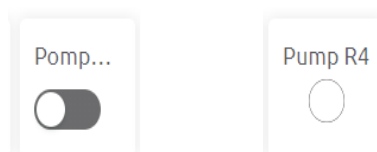
4.4.2 Pompa Air

Pada pengujian ini, dimaksudkan untuk mengetahui apakah pompa dapat berjalan sesuai yang diharapkan. Pompa disini akan diberi beberapa batas set parameter untuk ON dan OFF-nya, yakni sesuai dengan nilai yang terbaca oleh *soil moisture sensor*. Jika nilai sensor mencapai batas bawah kelembaban media tanam, maka pompa akan aktif, dan jika telah mencapai batas atasnya pompa akan mati, begitu juga seterusnya. Didapat tampilan *output running Pompa Air* pada “Serial Monitor” *Software* Arduino IDE seperti dibawah ini.

```
FAN ON
PUMP OFF
Temperature : 27.50
Humidity : 75.30
Soil Moisture : 29.43
FAN ON
PUMP OFF
Temperature : 27.50
```

Gambar 24. Tampilan *output* pada serial monitor Arduino IDE

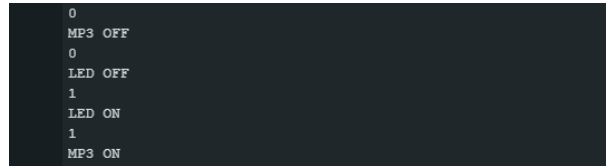
Tampilan *output running Pompa Air* pada aplikasi *Blynk* seperti dibawah ini.



Gambar 15. Tampilan *output* pada *blynk*

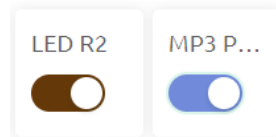
4.4.3 LED dan MP3 Module

Pengujian LED dan MP3 ini bertujuan untuk mengetahui apakah keduanya dapat bekerja dengan sebagaimana mestinya. Kedua aktuator ini bekerja sesuai dengan *schedule* yang diatur pada “Automation” pada menu *Blynk*. *Scheduling* bertujuan untuk mengatur waktu hidup dan mati pada LED dan *MP3 Module*. Pengujian ini dilakukan untuk mengetahui apakah *scheduling* ini dapat berjalan dengan baik pada waktu yang sudah diatur. Didapat tampilan *output running* LED dan *MP3 Module* pada “Serial Monitor” *Software* Arduino IDE seperti dibawah ini.



Gambar 16. Tampilan *output* pada serial monitor Arduino IDE

Tampilan *output running* LED dan *MP3 Module* pada aplikasi *Blynk* seperti dibawah ini.



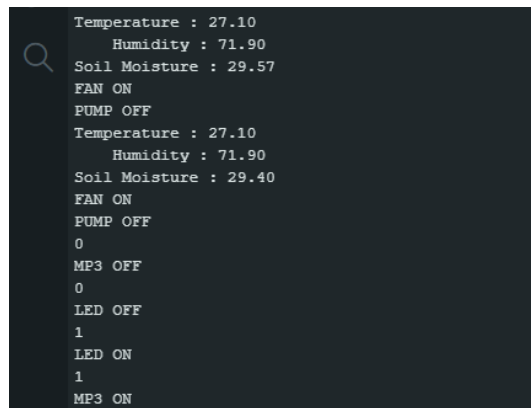
Gambar 173. Tampilan output pada *blynk*

4.4.4 Hasil pengujian mikrokontrol dan IoT

Dalam pengujian ini mencakup keseluruhan sistem yang digunakan, seberapa bagus performa kinerja sistem dapat bekerja

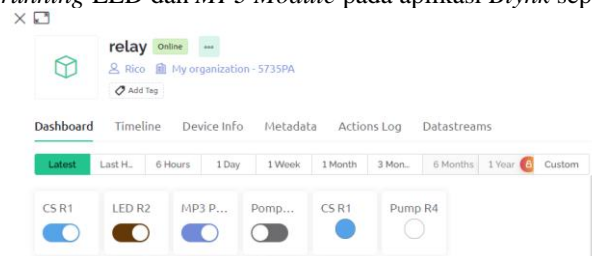
- a. *Output running* mikrokontrol dan IoT

Didapat tampilan *output running* LED dan *MP3 Module* pada Serial Monitor *Software* Arduino IDE seperti dibawah ini.



Gambar 18. Tampilan *output* keseluruhan pada serial monitor Arduiuno IDE

Tampilan *output running* LED dan *MP3 Module* pada aplikasi *Blynk* seperti dibawah ini.



Gambar 19. Tampilan *output* mikrokontrol pada *blynk*

b. Hasil pengujian mikrokontrol dan IoT

Tabel 5. Tabel uji mikrokontrol

No	Parameter	Input					Output								Ket		
							Drvr		CS		Led		Mp3			Pump	
							ON	OFF	ON	OFF	ON	OFF	ON	OFF		ON	OFF
1	ESP32	0	0	0	0	0	√		√		√		√		√	Brhsl	
2	ESP32	0	0	0	0	1	√		√		√		√		√	Brhsl	
3	ESP32	0	0	0	1	0	√		√		√		√		√	Brhsl	
4	ESP32	0	0	0	1	1	√		√		√		√		√	Brhsl	
5	ESP32	0	0	1	0	0	√		√		√		√		√	Brhsl	
6	ESP32	0	0	1	0	1	√		√		√		√		√	Brhsl	
7	ESP32	0	0	1	1	0	√		√		√		√		√	Brhsl	
8	ESP32	0	0	1	1	1	√		√		√		√		√	Brhsl	
9	ESP32	0	1	0	0	0	√		√		√		√		√	Brhsl	
10	ESP32	0	1	0	0	1	√		√		√		√		√	Brhsl	
11	ESP32	0	1	0	1	0	√		√		√		√		√	Brhsl	
12	ESP32	0	1	0	1	1	√		√		√		√		√	Brhsl	
13	ESP32	0	1	1	0	0	√		√		√		√		√	Brhsl	
14	ESP32	0	1	1	0	1	√		√		√		√		√	Brhsl	
15	ESP32	0	1	1	1	0	√		√		√		√		√	Brhsl	
16	ESP32	0	1	1	1	1	√		√		√		√		√	Brhsl	
17	ESP32	1	0	0	0	0	√		√		√		√		√	Brhsl	
18	ESP32	1	0	0	0	1	√		√		√		√		√	Brhsl	
19	ESP32	1	0	0	1	0	√		√		√		√		√	Brhsl	
20	ESP32	1	0	0	1	1	√		√		√		√		√	Brhsl	
21	ESP32	1	0	1	0	0	√		√		√		√		√	Brhsl	
22	ESP32	1	0	1	0	1	√		√		√		√		√	Brhsl	
23	ESP32	1	0	1	1	0	√		√		√		√		√	Brhsl	
24	ESP32	1	0	1	1	1	√		√		√		√		√	Brhsl	
25	ESP32	1	1	0	0	0	√		√		√		√		√	Brhsl	
26	ESP32	1	1	0	0	1	√		√		√		√		√	Brhsl	
27	ESP32	1	1	0	1	0	√		√		√		√		√	Brhsl	
28	ESP32	1	1	0	1	1	√		√		√		√		√	Brhsl	
29	ESP32	1	1	1	0	0	√		√		√		√		√	Brhsl	
30	ESP32	1	1	1	0	1	√		√		√		√		√	Brhsl	
31	ESP32	1	1	1	1	0	√		√		√		√		√	Brhsl	
32	ESP32	1	1	1	1	1	√		√		√		√		√	Brhsl	

Tabel 6. Tabel uji komunikasi IoT

No	Parameter	Input		Output		ΔUkur		Ket
		t (s)		t (s)		t (s)		
		Log in	Log in	Active	Log out	Active	Log out	
1.		30:15.75	30:15.75	30:16.35	30:16.71	00:00.60	00:00.96	Brhsl
2.		32:01.42	32:01.42	32:03.23	32:03.45	00:00.80	00:02.03	Brhsl
3.	CS	34:37.55	34:37.55	34:38.25	34:39.24	00:00.70	00:01.69	Brhsl
4.		37:24.71	37:24.71	37:25.38	37:26.27	00:00.67	00:01.56	Brhsl
5.		39:47.56	39:47.56	39:48.26	39:48.53	00:00.70	00:00.97	Brhsl
Σ						00:03.47	00:07.21	
Rata-Rata						00:00.69	00:01.44	
1		43:01.95	43:01.95	43:04.65	43:05.03	00:00.56	00:03.08	Brhsl
2		45:19.86	45:19.86	45:20.42	45:24.44	00:00.56	00:04.58	Brhsl
3	LED	48:52.52	48:52.52	48:53.56	48:57.65	00:01.04	00:05.13	Brhsl
4		49:39.78	49:39.78	49:40.21	49:44.98	00:00.43	00:05.20	Brhsl
5		51:21.34	51:21.34	51:22.07	51:26.83	00:00.73	00:05.49	Brhsl
Σ						00:03.32	00:23.48	
Rata-Rata						00:00.66	00:04.69	
1		54:47.40	54:47.40	54:47.92	54:49.51	00:00.52	00:02.11	Brhsl
2		55:57.64	55:57.64	55:57.95	55:59.81	00:00.31	00:02.17	Brhsl
3	Mp3	57:20.31	57:20.31	57:20.63	57:21.94	00:00.32	00:01.63	Brhsl
4		58:46.66	58:46.66	58:47.31	58:47.47	00:00.65	00:00.81	Brhsl
5		59:53.47	59:53.47	59:54.60	59:54.89	00:01.20	00:01.42	Brhsl
Σ						00:03.00	00:08.14	Brhsl
Rata-Rata						00:00.60	00:01.62	Brhsl

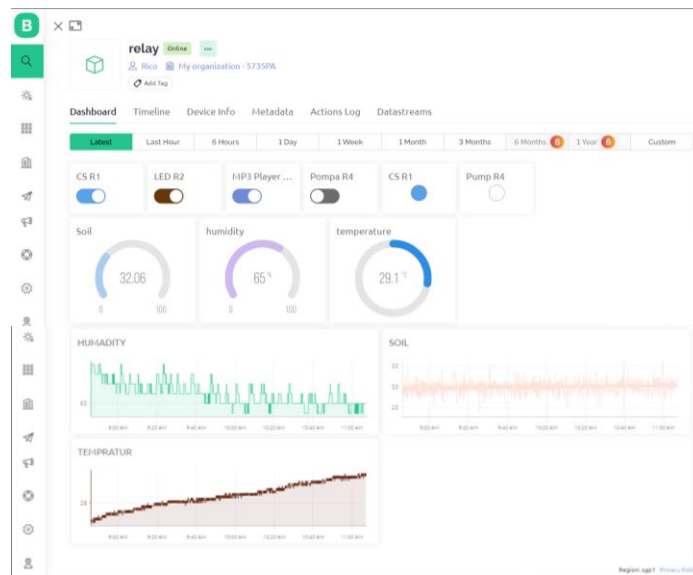
No	Parameter	Input t (s)		Output t (s)		ΔUkur t (s)		Ket
		Log in	Log in	Active	Log out	Active	Log out	
1	Pump	01:06.46	01:06.46	01:06.73	01:07.08	00:00.27	00:00.62	Brhsl
2		03:23.56	03:23.56	03:24.64	03:24.86	00:01.08	00:01.30	Brhsl
3		04:53.13	04:53.13	04:54.98	04:55.37	00:00.85	00:02.24	Brhsl
4		06:52.67	06:52.67	06:53.18	06:55.45	00:00.51	00:00.78	Brhsl
5		08:01.16	08:01.16	08:03.00	08:03.50	00:00.84	00:02.34	Brhsl
Σ						00:03.55	00:07.28	
Rata-Rata						00:00.71	00:01.45	

Pengujian diatas tidak mencantumkan jam, hanya mencantumkan menit, detik dan mili detik, dan dilakukan mulai jam 12.30 secara bergiliran. Dari tabel diatas dapat disimpulkan bahwa respon waktu komunikasi *IoT* sudah tergolong baik. Penggunaan cocopeat juga memberi pengaruh yang baik pada sistem tersebut, yang mana daya simpan airnya yang besar dan juga tahan lama, jadi pompa tidak mudah menyala dalam waktu yang singkat.

Kelemahan penggunaan *Automation* pada aplikasi *Blynk* yaitu jika suatu saat koneksi sistem ke internet terputus akan mempengaruhi pada kinerja sistem seperti halnya salah satu aktuator yang diatur menyala pada waktu tertentu namun pada waktu menyala koneksi internet tiba-tiba terputus, maka aktuator tersebut akan melewati *schedule* tersebut dan tidak aktif saat sistem kembali terkoneksi ke internet. Begitupun saat koneksi terputus juga salah satu aktuator yang seharusnya sudah menyala tiba-tiba tidak merespon batas set parameternya, seperti *cooling system* yang tidak menyala saat suhu mencapai titik 24°C tanggal 23 Juli pada 01.30AM.

4.5. Running keseluruhan sistem

Pada pengujian kali ini bertujuan untuk mengetahui apakah sistem secara keseluruhan dapat berjalan dengan baik atau tidak, mulai dari pembacaan sensor suhu, kontrol manual pada aktuator, begitupula otomatisasi pada aktuator. Tampilan *output* dan pengujian keseluruhan sistem pada *Blynk*. Tampilan *output* keseluruhan sistem pada *Blynk* dapat dilihat pada Gambar 20 berikut:



Gambar 20. Tampilan output keseluruhan sistem pada *blynk*

5. KESIMPULAN

Secara keseluruhan, sistem berjalan sesuai dengan yang diinginkan. Ketika suhu mencapai angka 27°C maka *Cooling System* akan menyala dan akan mati setelah mencapai angka 25°C, sehingga *box smart farming microgreen* dapat mempertahankan nilai suhunya diantara 25-27°C. Begitu pula pada kelembaban media tanam, ketika mencapai angka 60% maka pompa akan aktif dan mulai membasahi media tanam hingga mencapai angka 40%. Scheduling LED dan MP3 Player juga berjalan sesuai dengan yang diinginkan dengan waktu menyala yakni 12 jam mulai pukul 7.00AM sampai pukul 7.00PM. Jika dilihat pada pertumbuhan *microgreen*, didapat hasil yang cukup memuaskan dimana *microgreen* dapat dipanen 7

hari setelah tanam dengan batang yang tebal, daun yang hijau segar dan akar yang kuat. Untuk pengembangan penelitian ini, pada wadah kecil di bagian bawah *microgreen* dapat ditambahkan *mist maker*.

REFERENSI

- [1] H. Valupi, Rosmaiti, dan Iswahyudi, "Pertumbuhan Dan Hasil Microgreens Beberapa Varietas Pakcoy (*Brassica rapa* . L) Pada Media Tanam Yang Berbeda," in *Seminar Nasional Fakultas Pertanian Universitas Samudra Ke-VI*, 2021, hal. 1–13.
- [2] M. Andrianto, "Penerapan Iot Pada Perawatan Tanaman Di Dalam Rumah," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 3, no. 1, hal. 173–180, 2019.
- [3] R. H. Hilmy, R. Susana, dan F. Hadiatna, "Rancang Bangun Smart Grow Box Hidroponik Untuk Pertumbuhan Tanaman Microgreen Berbasis Internet of Things," *Power Elektron. J. Orang Elektro*, vol. 10, no. 2, hal. 41–47, 2021.
- [4] I. Maulidiyah, M. W. Lestari, dan S. A. Mardiyani, "Pengaruh Aplikasi Perendaman Berbagai Jenis Media Tanam dengan Beberapa Pupuk Cair Terhadap Kualitas dan Tingkat Kesukaan Konsumen Microgreen Wheatgrass (*Triticum aestivum* L) Indah," *J. Folium*, vol. 6, no. 2, hal. 118–126, 2022.
- [5] C. Kaiser dan M. Ernst, "Babyveggies," *Cent. Crop Diversif.*, vol. 1, no. 1, hal. 1–3, 2017.
- [6] S. Saputra, A. Jaenul, dan A. Olivia, "Prototype Sistem Monitoring Dan Controlling Budidaya Microgreen Dengan Menggunakan Website Berbasis Internet Of Things (IoT)," *J. Media Elektro*, vol. XI, no. 2, hal. 178–188, 2022.
- [7] M. . Fahlevi dan H. Gunawan, "Perancangan Sistem Pendeteksi Banjir Berbasis Internet of Things," *It (Informatic Tech. J.*, vol. 8, no. 1, hal. 23–29, 2021.
- [8] Wilianto dan A. Kurniawan, "Sejarah , Cara Kerja Dan Manfaat Internet Of Things," *J. MATRIX*, vol. 8, no. 2, hal. 36–41, 2009.
- [9] A. A. Wicaksana, A. T. Mulyani, N. Suranti, dan M. Sukmah, "Penerapan Teknologi Tepat Guna Penyiraman Otomatis Menggunakan Capacitive Soil Moisture Sensor pada Taman Tanaman Obat Keluarga (TOGA) Desa Gedangan," *J. Pengabd. Masy. Indones.*, vol. 2, no. 2, hal. 150–163, 2023.
- [10] A. H. Saptadi, "Perbandingan Akurasi Pengukuran Suhu dan Kelembaban Antara Sensor DHT11 dan DHT22," *J. INFOTEL - Inform. Telekomun. Elektron.*, vol. 6, no. 2, hal. 49–56, 2014,.
- [11] P. Puspasari, T. P. Satya, I. Oktawati, Unan Yusmaniar Fahrurrozi, dan H. Prisyanti, "Analisis Akurasi Sistem Sensor DHT22 berbasis Arduino terhadap Thermohygrometer Standar," *J. Fis. dan Apl.*, vol. 16, no. 1, hal. 40–45, 2020.