

PEMBUATAN *MIDDLEWARE* UNTUK INTEGRASI DENGAN MIKROKONTROLER BERBASIS IOT

Gilang Aditya Mahesaputra ¹, Billy Montolalu, S.Kom.,M.Kom ²,

Dr. Helmy Widyantara, S.Kom.,M.Eng ^{3,*}

¹ Faculty of Electrical Technology and Smart Industry, Institut Teknologi Telkom Surabaya

gilang@student.ittelkom-sby.ac.id

² Faculty of Electrical Technology and Smart Industry, Institut Teknologi Telkom Surabaya

billy@ittelkom-sby.ac.id

³ Faculty of Information Technology and Business, Institut Teknologi Telkom Surabaya

helmywid@ittelkom-sby.ac.id

Abstrak: Di era Industri 4.0 semua alat yang terhubung dengan internet dan saling terintegrasi atau bisa di sebut *Internet of Things* (IoT). Di era Industri 4.0 sudah menggunakan teknologi kontrol mikrokontroler yang modern, namun beberapa mikrokontroler sulit dikontrol dan dimonitoring melalui berbagai *device* karena tidak adanya standarisasi. Penelitian mengenai pembuatan *Middleware* ini bertujuan untuk mengontrol mikrokontroler menggunakan standar khusus agar mikrokontroler dapat dikontrol menggunakan beberapa jenis *device*. *Middleware* sudah berfungsi cukup baik karena sudah berhasil menjalankan 2 *action* pada JSON *middleware*, yaitu untuk mengambil data dan menerima perintah. *Action* "on" "off" untuk memberikan perintah menyalakan dan mematikan pada modul switch lamp dan "get" untuk mendapatkan data sensor tanah dari *soil integrated sensor*. *Middleware* menggunakan mikrokontroler. Dengan adanya *middleware*, memudahkan user dalam mengontrol pompa air dan monitor kelembapan tanah dengan mengakses melalui *Website* dan *Android*, user juga tidak perlu takut apabila mikrokontroler bermasalah, karena satu mikrokontroler memiliki program yang sama dalam menjalin komunikasi dengan *middleware*, cukup dengan melakukan kompilasi ulang ketika mengganti mikrokontroler dan mengganti pin *output*. Hasil pengujian ini menentukan kelembapan tanah (RH) selalu terjaga di 400 hingga 700 dengan persentase *error* sebesar 4,9%, sedangkan untuk mengontrol pompa membutuhkan waktu akses sekitar 20 hingga 100 milisecond.

Kata Kunci: *Internet of Things*; *Middleware*; Mikrokontroler

DEVELOPMENT OF *MIDDLEWARE* FOR INTEGRATION WITH IOT BASED MICROCONTROLLER

Abstract: In the era of Industry 4.0, all tools are connected to the internet and integrated with each other or can be called the *Internet of Things* (IoT). In the Industrial 4.0 era, modern microcontroller control technology was used, but several microcontrollers were difficult to control and monitor through various devices because there was no standardization. Research on making *Middleware* aims to control microcontrollers using special standards so that microcontrollers can be controlled using several types of devices. The *middleware* is functioning quite well because it has succeeded in executing 2 actions on the JSON *middleware*, namely to retrieve data and receive commands. *Action* "on" "off" to give commands to turn on and off the switch lamp

module and "get" to get soil sensor data from the soil integrated sensor. Middleware uses a microcontroller. With middleware, it makes it easier for users to control water pumps and monitor soil moisture by accessing via the Website and Android, users also don't need to be afraid if the microcontroller has problems, because one microcontroller has the same program in establishing communication with middleware, just by recompiling when changing microcontroller and change the output pin. The results of this test determine that soil moisture (RH) is always maintained at 400 to 700 with an error percentage of 4.9%, while controlling the pump requires access time of around 20 to 100 milliseconds.

Keywords: *Internet of Things; Middleware; Microcontroller*

1. Pendahuluan

Saat ini kita sedang memasuki era revolusi industri 4.0. Revolusi industri 4.0 telah membawa perubahan yang fundamental dalam segala aspek kehidupan. Perubahan ini didorong oleh perkembangan internet yang luar biasa dan juga didukung oleh perkembangan teknologi digital[1]. Di era Industri 4.0 semua alat yang terhubung dengan *internet* dan saling terintegrasi atau bisa disebut *Internet of Things* (IoT). *Internet of Things* (IoT) berpengaruh dalam berbagai macam industri seperti manufaktur, logistik, kesehatan, tata kota, rumah, pertanian, bahkan industri otomotif. Fungsi utama IoT pada dasarnya sebagai data miner. IoT bekerja mencari dan mengumpulkan berbagai data dari lapangan yang nantinya akan diolah menjadi data yang lebih bermanfaat. Jadi Jaman Industri 4.0 dan *Internet of Things* (IoT) saling berhubungan[2].

Sistem IoT saat ini masih umum menggunakan mikrokontroler sedangkan pada bidang industri jenis mikrokontroler yang sering digunakan adalah PLC. Saat ini banyak industri yang menggunakan teknologi kontrol yang canggih seperti *Programmable Logic Controller* (PLC), *Arduino*, dll[3]. Dilihat dari bidangnya, Industri yang menggunakan teknologi kontrol mikrokontroler sulit dikontrol dan dimonitor menggunakan web atau perangkat karena standarisasi yang kurang. Ini mengharuskan setiap perangkat terintegrasi memiliki standar komunikasi yang khusus untuk mikrokontroler.

Penelitian ini fokus pada integrasi Vertical Farming di Rooftop Institut Teknologi Telkom Surabaya. Tujuannya adalah mengatasi kurangnya standarisasi mikrokontroler dengan mengembangkan *Middleware* yang memungkinkan komunikasi berbagai mikrokontroler dengan standar yang sama. *Middleware* ini memfasilitasi kendali pompa air dan pemantauan kelembapan melalui Website dan Android, mengurangi ketergantungan fisik. Keuntungan lainnya adalah keseragaman program komunikasi, memungkinkan pergantian mikrokontroler atau pin output tanpa masalah besar. Penelitian ini diharapkan menciptakan otomatisasi budidaya tanaman di Vertical Farming, menggantikan pendekatan manual yang digunakan saat ini.

2. Tinjauan Pustaka

2.1 *Middleware*

Middleware adalah sebuah perangkat lunak yang digunakan berkomunikasi dengan satu sama lain pada aplikasi yang berbeda. Atau lebih detailnya, *middleware* adalah perangkat lunak yang membuat suatu jaringan pertukaran informasi antar aplikasi komputer yang berbeda. Jaringan diimplementasi dengan penggunaan teknik pertukaran informasi yang sama pada semua aplikasi yang terimplikasi dengan bantuan perangkat lunak yang terkomposisi. *Middleware* sangat populer[4]. *Middleware* menyediakan fungsi untuk menghubungkan aplikasi secara cerdas dan efisien sehingga dapat berinovasi dengan lebih cepat. *Middleware* sebagai jembatan antara teknologi, alat, dan basis data yang beragam sehingga dapat melakukan integrasi tanpa batasan ke dalam suatu sistem tunggal. Sistem tunggal tersebut nantinya akan memberikan layanan terpadu kepada penggunanya.

2.2 Vertical Farming

Vertical Farming (VF) adalah teknik pertanian yang melibatkan produksi pangan skala besar di gedung-gedung tinggi yang memungkinkan pertumbuhan cepat dan produksi terencana dengan mengendalikan kondisi lingkungan dan larutan nutrisi untuk tanaman berbasis aeroponik, menggunakan metode dan teknologi rumah kaca tepi. Pertanian vertikal paling efektif di seluruh dunia di berbagai negara dengan evaluasi kelayakan VF untuk berbagai iklim/wilayah geografis berdasarkan jenis dan teknologinya. Karena tanaman akan ditanam di luar ruangan dengan teknik pertanian lingkungan yang terkendali. Jelas bahwa berbagai jumlah dan jenis produk yang ditentukan VF tidak terjadi begitu saja, tetapi berkembang di beberapa kota dengan karakteristik regional yang berbeda di seluruh dunia[5].

2.3 Mikrokontroler

Mikrokontroler adalah *single chip* komputer yang memiliki kemampuan untuk diprogram dan digunakan untuk tugas-tugas yang berorientasi kendali (*control*). Mikrokontroler muncul dengan dua alasan utama, yaitu kebutuhan pasar (*market need*) dan perkembangan teknologi baru (*expansion of technology*). Yang dimaksud dengan kebutuhan pasar adalah kebutuhan yang luas dari produk-produk elektronik akan perangkat pintar sebagai pengendali dan pemroses data. Sedangkan yang dimaksud dengan perkembangan teknologi baru adalah perkembangan teknologi semikonduktor yang memungkinkan pembuatan *chip* dengan kemampuan komputasi yang sangat cepat, bentuk yang semakin kecil, dan harga yang semakin murah (*smart, small, and cheap*)[6].

2.4 Arduino

Arduino Uno merupakan salah satu mikrokontroler *single-board* berbasis Arduino Uno yang bersifat yang dapat memudahkan pengguna untuk mengetahui dan mengembangkan cara kerja perangkat tersebut. Perangkat tersebut memiliki 14 *input/output digital* yang mana 3, 5, 6, 9, 10, dan 11 dapat digunakan sebagai *PWM output*, enam *input analog*, *power* (VIN, 5V, 3V3, dan GND), resonator keramik 16 MHz, *USB connector*, *power jack*, *ICSP header*, dan *reset button*. Sebagai contoh penggunaan Arduino Uno adalah sistem kontrol kekeruhan akuarium, pembacaan meteran air dimana Arduino Uno digunakan sebagai prosesor data dari node sensor, dan pembacaan frekuensi pada deteksi jalur pipa terpendam[7].

2.5 Arduino IDE

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino[8].

Program yang ditulis dengan menggunakan Arduino Software (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam *file* dengan ekstensi ".ino". *Tekt editor* pada *Arduino Software* memiliki fitur seperti *cutting/paste* dan *searching/replacing* sehingga memudahkan dalam menulis kode program[9].

Pada *Software* Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Dibagian bawah paling kanan *Software* Arduino IDE[10].

2.6 Sensor Tanah

Soil integrated sensor adalah sensor yang memiliki fungsi untuk mendeteksi tingkat kelembaban tanah dan juga dapat digunakan untuk menentukan apakah ada kandungan air di tanah/ sekitar sensor. Pada penelitian ini menggunakan *Soil Integrated Sensor* bertipe NPK 7 in 1[11]. Pada sensor ini terintegrasi dengan 7 fungsi item, yaitu:

1. Mengukur kelembapan tanah.
2. Konduktivitas.
3. Suhu.
4. Nitrogen.

5. Fosfor.
6. Kalium.
7. pH.

Sensor ini disebut kapasitif karena lima tembaga dalam sensor adalah lima plat kapasitor. Lima plat ini bisa ditarik menghadap satu sama lain, dengan ruang kosong di antaranya. Dengan menaruh material di antara plat, besaran muatan kapasitansi akan berubah dan mengubah tegangan. Bahan ini disebut dielektrik, dan banyaknya perubahan kapasitansi untuk material tertentu disebut konstanta dielektrik material. Tanah kering memiliki konstanta dielektrik yang berbeda dari tanah basah, yang berarti bahwa sensor di tanah basah akan memiliki kapasitansi yang berbeda dari tanah yang kering[12].

2.7 NodeJS

Node.js merupakan salah satu platform pengembang yang dapat digunakan untuk membuat aplikasi berbasis *Cloud*. Node.js dikembangkan dari *engine JavaScript* yang dibuat oleh Google untuk browser Chrome ditambah dengan *script library* serta beberapa pustaka lainnya. Node.js menggunakan *JavaScript* sebagai bahasa pemrograman dan *event-driven, non-blocking I/O (asynchronous)* model yang membuatnya ringan dan efisien. Node.js memiliki fitur *built-in HTTP server library* yang menjadikannya mampu menjadi sebuah *web server* tanpa bantuan *software* lainnya seperti *Apache* dan *Nginx*. Pada dasarnya, Node.js adalah sebuah *runtime environment* dan *script library* [13].

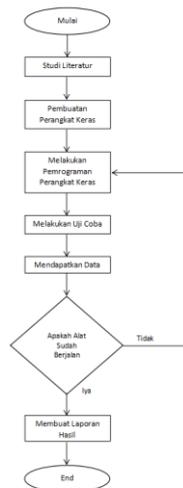
2.8 JSON

JavaScript Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, *Java*, *JavaScript*, *Perl*, *Python*, dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data[14].

3. Metode dan Pemodelan

3.1. Alur Penelitian

Pada Gambar 1 dijelaskan tahap-tahap dalam melakukan penelitian. Tahap awal dalam penelitian ini bertujuan untuk melakukan pencarian literatur dari penelitian sebelumnya, baik dalam bentuk buku, jurnal, maupun sumber-sumber internet. Tujuan dari langkah ini adalah untuk memahami teori yang relevan dan memastikan dasar yang kuat dalam penyusunan laporan serta meminimalkan potensi kesalahan dalam penelitian. Setelah literatur terkumpul, langkah selanjutnya adalah merancang alat yang akan dibuat berdasarkan hasil temuan dalam literatur. Proses ini melibatkan pemrograman alat yang direncanakan. Setelah pemrograman selesai, uji coba dilakukan terhadap alat tersebut. Jika hasil uji coba memenuhi harapan, langkah terakhir adalah menyusun laporan. Namun, jika alat tidak berhasil pada uji coba, akan dilakukan pemrograman ulang pada perangkat keras guna memperbaiki program sebelum melanjutkan pada tahap penyusunan laporan.



Gambar 1. Alur Penelitian

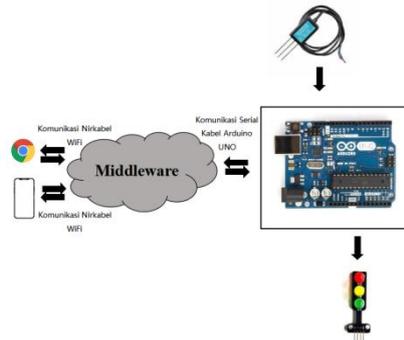
3.2. Metode Penelitian

Pada penelitian ini menggunakan jenis penelitian dan pengembangan atau *Research and Development* (R&D) dengan Metode *Waterfall*. Penelitian pengembangan menurut Endang Mulyatiningsih (2011: 145) merupakan penelitian yang bertujuan menghasilkan produk baru melalui proses pengembangan. Produk yang dikembangkan oleh peneliti adalah pembuatan *middleware* yang diintegrasikan dengan mikrokontroler berbasis IoT.

3.3. Skema Sistem

Pada Gambar 2 dijelaskan skema sistem dalam melakukan penelitian. Penelitian ini melibatkan penggunaan mikrokontroler dalam mengendalikan berbagai komponen seperti *Soil Integrated Sensor* dan Pompa Air. Mikrokontroler mengirimkan Action "get" ke *Soil Integrated Sensor* melalui kabel jumper yang terhubung pada pin "A0" dari client melalui *middleware*, kemudian data sensor dikirim kembali ke mikrokontroler. Selanjutnya, data dari *Soil Integrated Sensor* dikirim ke *middleware* menggunakan komunikasi serial melalui kabel Arduino Uno untuk dapat diakses oleh client melalui website atau android. Demikian pula, mikrokontroler mengirim Action "On" dan "Off" ke Pompa Air melalui kabel jumper yang terhubung pada pin "11, 12, 13". Mikrokontroler juga terhubung dengan Modul Lampu Switch sebagai indikator status Pompa Air. Data dari Modul Lampu Switch dikirim ke mikrokontroler dan kemudian diteruskan ke *middleware* untuk diakses oleh *client*.

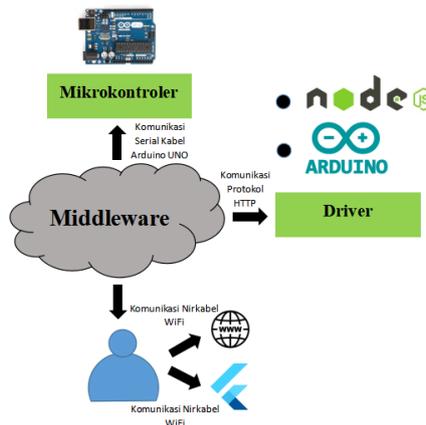
Proses komunikasi antara mikrokontroler dan *middleware* dilakukan melalui Program Arduino IDE yang menggunakan JSON. JSON digunakan sebagai penghubung antara mikrokontroler dan Platform NodeJS, yang nantinya akan terhubung ke dalam protokol HTTP untuk diakses oleh client melalui Website atau Android. Dalam penelitian ini, mikrokontroler yang digunakan adalah Arduino Uno. Selain itu, koneksi antara *middleware*, website, dan Android menggunakan jaringan nirkabel WiFi dengan alamat IP address yang sama sesuai dengan protokol HTTP yang telah dibuat. *Middleware* sendiri berada di perangkat laptop dan dijalankan pada platform NodeJS.



Gambar 2. Skema Sistem

3.4. Desain Sistem *Middleware*

Pada Gambar 3 dijelaskan desain sistem *middleware* dalam melakukan penelitian. Penelitian ini menggambarkan desain sistem *middleware* yang diintegrasikan dengan mikrokontroler berbasis IoT. Konsep desain ini memungkinkan penghubungan perangkat keras, seperti Arduino, dengan antarmuka pengguna (client) melalui website atau Android. *Middleware* berperan sebagai jembatan komunikasi antara perangkat keras dan perangkat lunak, mengatur pertukaran data dan instruksi di antara keduanya.



Gambar 3. Desain *Middleware*

Proses integrasi berlangsung sebagai berikut:

1. Mikrokontroler Arduino: Tahap dimulai dengan penggunaan mikrokontroler Arduino sebagai perangkat keras utama. Mikrokontroler ini dapat memiliki sensor dan aktuator yang terhubung, serta bertindak sebagai pengendali perangkat keras.
2. Koneksi ke *Middleware*: Mikrokontroler terhubung fisik ke *middleware* melalui komunikasi serial menggunakan kabel Arduino Uno atau melalui protokol komunikasi nirkabel seperti WiFi.
3. Peran *Middleware*: *Middleware* berada di dalam platform NodeJS pada laptop dan berfungsi sebagai pengatur pertukaran data antara mikrokontroler Arduino dan *client* melalui website atau Android.
4. Protokol Komunikasi: *Middleware* menggunakan protokol komunikasi, seperti HTTP, untuk berinteraksi dengan mikrokontroler Arduino.
5. Komunikasi dengan Mikrokontroler: *Middleware* mengirimkan permintaan dan instruksi ke mikrokontroler melalui protokol komunikasi yang ditentukan. Data hasil bacaan sensor atau konfirmasi instruksi dikirimkan kembali ke *middleware*.
6. Pengolahan Data: *Middleware* memproses data yang diterima dari mikrokontroler sesuai dengan logika bisnis yang diterapkan.
7. Koneksi dengan Client: *Middleware* menyediakan antarmuka yang dapat diakses oleh client melalui website atau Android menggunakan koneksi WiFi.

8. Permintaan Client - *Middleware*: Client mengirimkan permintaan melalui website atau Android, yang diteruskan oleh *middleware* untuk dieksekusi.
9. Pengiriman Instruksi ke Mikrokontroler: *Middleware* mengirimkan instruksi dari client ke mikrokontroler melalui protokol komunikasi yang digunakan.
10. Respons Time Mikrokontroler - *Middleware*: Mikrokontroler melakukan tindakan yang sesuai dan mengirimkan *Respons Times*nya kembali ke *middleware*.
11. Respons Time *Middleware* - Client: *Middleware* menerima *Respons Times* dari mikrokontroler dan mengirimkannya kembali ke client melalui website atau Android.

Dengan demikian, desain sistem *middleware* ini memungkinkan pengontrolan perangkat keras melalui antarmuka pengguna yang disediakan oleh *middleware*, memungkinkan pemantauan dan kontrol jarak jauh melalui internet.

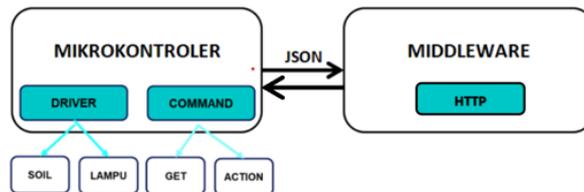
3.5. Desain *Framework*

Pada Gambar 4 dijelaskan desain *framework* dalam melakukan penelitian. Pada aplikasi ini, pengguna akses *Middleware* melalui HTTP. Untuk mendapatkan data dari sensor, pengguna kirim "get" melalui HTTP ke *Middleware*. *Middleware* teruskan ke Mikrokontroler melalui komunikasi serial dengan format JSON. Mikrokontroler proses "get" dengan akses driver sesuai untuk baca sensor.

Data dari sensor dikirim ke *Middleware* dalam format JSON melalui serial. *Middleware* teruskan data ke pengguna via HTTP. Untuk aksi ke Mikrokontroler, pengguna kirim "action" via HTTP ke *Middleware*. *Middleware* kirim "action" ke Mikrokontroler melalui serial dengan JSON.

Mikrokontroler olah "action" via driver sesuai untuk kendalikan aktuator, contohnya lampu. Mikrokontroler jalankan aksi seperti nyalakan/matikan lampu. *Middleware* juga kirim *Respons Times* ke pengguna mengenai hasil dari aksi.

Dengan alur ini, Mikrokontroler berinteraksi dengan pengguna via *Middleware* dengan HTTP, format JSON untuk tukar data dan perintah.



Gambar 4. Desain *Framework*

4. Hasil dan Analisa

4.1. Hasil Pengujian Pengiriman Instruksi melalui Website

Pada Tabel 1 merupakan pengujian *Respons Time* website yang dilakukan sebanyak 12 kali percobaan dan menghasilkan 12 data yang memiliki waktu milisecond yang berbeda. Hal ini terjadi karena factor jaringan internet yang tidak stabil. Pada Pengujian *Respons Time* memiliki *Respons Time* yang berbeda, karena memiliki *delay* pada saat pengambilan data. Hal ini terjadi karena untuk mendapatkan data sensor membutuhkan waktu respon untuk diolah oleh Sensor NPK.

Tabel 1. Pengujian *Respons Time* Website

Pengujian <i>Respons Time</i> Website			
Percobaan Ke	Command	Object	<i>Respons Time</i> (ms)
1	On	Modul Lampu Switch 1	47
2	On	Modul Lampu Switch 1	40
3	On	Modul Lampu Switch 2	41
4	On	Modul Lampu Switch 3	40

5	Off	Modul Lampu Switch 1	41
6	Off	Modul Lampu Switch 1	38
7	Off	Modul Lampu Switch 2	39
8	Off	Modul Lampu Switch 3	35
9	Get	Soil Integrated Sensor	5020
10	Get	Soil Integrated Sensor	5026
11	Get	Soil Integrated Sensor	5032
12	Get	Soil Integrated Sensor	5047

4.2. Hasil Pengujian Pengiriman Instruksi melalui Android

Pada Tabel 2 merupakan pengujian *Respons Time* website yang dilakukan sebanyak 12 kali percobaan dan menghasilkan 12 data yang memiliki waktu milisecond yang berbeda. Hal ini terjadi karena factor jaringan internet yang tidak stabil. Pada Pengujian *Respons Time* memiliki *Respons Time* yang berbeda, karena memiliki *delay* pada saat pengambilan data. Hal ini terjadi karena untuk mendapatkan data sensor membutuhkan waktu respon untuk diolah oleh Sensor NPK.

Tabel 2. Pengujian *Respons Time* Android

Pengujian <i>Respons Time</i> Android			
Percobaan Ke	Command	Object	<i>Respons Time</i> (ms)
1	On	Modul Lampu Switch 1	737
2	On	Modul Lampu Switch 1	103
3	On	Modul Lampu Switch 2	667
4	On	Modul Lampu Switch 3	1106
5	Off	Modul Lampu Switch 1	1006
6	Off	Modul Lampu Switch 1	201
7	Off	Modul Lampu Switch 2	1231
8	Off	Modul Lampu Switch 3	840
9	Get	Soil Integrated Sensor	5449
10	Get	Soil Integrated Sensor	5616
11	Get	Soil Integrated Sensor	6872
12	Get	Soil Integrated Sensor	5578

4.3. Hasil Pengujian Alat Sensor NPK dengan 3 in 1 Soil Meter

Pada Tabel 3 merupakan pengujian perbandingan hasil pengukuran antara *soil integrated sensor* di website dan Android dengan 3 in 1 Soil Meter. Pengujian alat sensor tanah berhasil menampilkan data pada website dan Android, berfungsi sebagai sumber informasi kelembapan tanah yang diperlukan oleh klien. Perbandingan data menunjukkan perbedaan di 5 tanaman selama pengujian, mungkin disebabkan oleh *error* sensor saat pengambilan data.

Tabel 3. Pengujian Kalibrasi Soil Integrated Sensor dengan 3 in 1 Soil Meter

Pengujian Kalibrasi Alat Soil Integrated Sensor dengan 3 in 1 Soil Meter				
Percobaan Ke	Soil Integrated Sensor	3 in 1 Soil Meter	Selisih	Persentase Error (%)
1	650 RH	600 RH	50	8,3%

2	722 RH	700 RH	22	3,2%
3	724 RH	680 RH	44	6,5%
4	588 RH	590 RH	2	0,3%
5	448 RH	460 RH	28	6,1%
Rata Rata <i>error</i> (%)				4,9%

Perbandingan hasil pengukuran antara *soil integrated sensor* di website dan Android dengan 3 in 1 Soil Meter menghasilkan persentase *error*, dihitung dari selisih pembacaan dibagi nilai 3 in 1 Soil Meter, kemudian dikalikan 100%.

$$Error = \frac{\text{Selisih nilai pembacaan}}{\text{Nilai 3 in 1 Soil Meter}} \times 100\%$$

Berdasarkan rumus di atas, hasil perhitungan yang diperoleh adalah sebagai berikut:

$$Error = \frac{50}{600} \times 100\%$$

$$= 0,083 \times 100\%$$

$$= 8,3\%$$

Adapun rata rata persentase *error* yang di dapat dari jumlah hasil persentase *error* kemudian dibagi dengan banyaknya percobaan.

$$\text{Rata Rata Error (\%)} = \frac{\text{Jumlah persentase error (\%)}}{\text{Banyaknya percobaan}}$$

$$= \frac{24,5(\%)}{5}$$

$$= 4,9\%$$

4.3. Format JSON Monitor Data Sensor NPK dan Kontrol Modul Lampu Switch

Pada Gambar 5 merupakan format JSON untuk memberikan perintah dalam hal Menyalakan dan Mematikan Pompa 1, 2, dan 3, serta memberikan perintah untuk mendapatkan data sensor tanah. Format JSON ini terdapat pada mikrokontroler Arduino untuk memberikan perintah kepada pompa air dan sensor tanah.

```
//adapun format JSON untuk menjalankan Action mendapatkan data dan mengirim data
{
  "command":"on", "pin":"11",
  "command":"on", "pin":"12",
  "command":"on", "pin":"13",
  "command":"off", "pin":"11",
  "command":"off", "pin":"12",
  "command":"off", "pin":"13",
  "command":"get", "pin":"9"
}
```

Gambar 5 Format JSON

Adapun penjelasan dari format JSON adalah sebagai berikut :

- *Command* = Sebuah perintah kepada mikrokontroler untuk melakukan sesuatu.
- *On* = Sebuah perintah untuk menyalakan pompa air dari mikrokontroler.
- *Off* = Sebuah perintah untuk mematikan pompa air dari mikrokontroler.
- *Get* = Sebuah perintah untuk mendapatkan data dari sensor tanah dari mikrokontroler.
- *Device* = Sebuah media perintah yang menghubungkan dari perintah *command* ke sebuah *object lamp* atau *soil* agar dapat diperintah.
- *Lamp* = Sebuah lampu indikator yang menandakan bahwa pompa air sedang posisi mati atau hidup.
- *Soil* = Sebuah sensor tanah yang digunakan untuk mengukur kelembapan.

5. Kesimpulan

1. Akurasi Sensor Kelembapan Tanah:

Pengujian dilakukan terhadap sensor kelembapan tanah Soil Integrated Sensor dengan membandingkannya dengan 3 in 1 Soil Meter. Hasil pengujian menunjukkan bahwa sensor Soil Integrated memiliki rata-rata persentasi error sebesar 4,9% dalam kisaran nilai kelembapan 400 hingga 700. Hal ini menunjukkan bahwa sensor memiliki akurasi yang kurang tepat, kemungkinan disebabkan oleh adanya error pada sensor elektronik itu sendiri saat mengambil data.

2. Respons Time Modul Lampu Switch:

Pengujian Respons Time pada modul Lampu Switch menunjukkan bahwa nilai Respons Time berada dalam rentang 20 hingga 100 milidetik. Sementara itu, Respons Time dalam pengambilan data dari Soil Integrated Sensor memiliki rentang waktu 5000 hingga 6800 milidetik karena sensor memerlukan waktu tunggu respon sebelum dikirim kepada client. Hasil ini mengindikasikan adanya perbedaan waktu Respons Time dikarenakan faktor ketidakstabilan jaringan internet.

Dengan demikian, kesimpulan yang dapat diambil adalah bahwa sensor kelembapan tanah memiliki akurasi yang kurang tepat, mungkin akibat adanya error pada sensor elektronik saat pengambilan data. Selain itu, perbedaan Respons Time dalam modul Lampu Switch dan pengambilan data sensor juga dipengaruhi oleh faktor ketidakstabilan jaringan internet.

Referensi

- [1] Rojko A. 2017. "Industry 4.0 concept: Background and overview. *International Journal of Interactive Mobile Technologies*". 77-90. 11(5).
- [2] Prasetyo Teguh Utomo. 2019. "Potensi Implementasi Internet of Things Untuk Perpustakaan". 1-18. 2(1).
- [3] IEEE. 2021. "Programmable Logic Controllers in the Context of Industry 4.0". 17(5).
- [4] Surya. 2018. "Pengembangan Middleware Berbasis Event-Driven Untuk Sinkronisasi Database RFID Book Drop dan Slims". 3(3).
- [5] *Journal of Landscape Ecology*. 2018. "Opportunities And Challenges In Sustainability Of Vertical Farming: A Review". 1-16. 35(1).
- [6] Fiqhi Achmad Ibadillah, Alfita Riza. 2017. "Mikrokontroler dan Aplikasinya". 1-165.
- [7] Virginia. 2018. "Rancang Bangun Sistem Kontrol Lampu Berbasis Arduino Mega 2560". 5(2).
- [8] Kusnanto Hadi, Pardono. 2018. "Rancang Bangun Alat Uji Buckling Portable Berbasis Arduino Uno".
- [9] Yurkha Akfi Kusuma, Pratikno Heri, Puspasari Ira. 2020. "Rancang Bangun Alat Pelipat Baju Otomatis Menggunakan Arduino Uno". 9(2).
- [10] *Jurnal Sistem Informasi dan Telematika*. 2019. "Model Smart Room Dengan Menggunakan Mikrokontroler Arduino Untuk Efisiensi Sumber Daya". 10(1).
- [11] Husdi. 2018. "Monitoring Kelembapan Tanah Pertanian Menggunakan Soil Moisture Sensor FC-28 Dan Arduino Uno". 10(2).
- [12] Eka Joni Candra, Maulana Algifanri. 2019. "Penerapan Soil Moisture Sensor Untuk Desain System Penyiraman Tanaman Otomatis.
- [13] Kurniawan, Humaira, & Rozi, F. 2020. REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, 1(4), 127 - 132.
- [14] Sudirman. 2016. "Analisis Komunikasi Data Dengan XML dan JSON Pada Webservice". 1(2).

