

Sistem Otomasi Rumah Berbasis Internet of Things (IOT) Melalui Cloud Server dengan Pengendali Smartphone

Internet of Things (IOT) Based Home Automation System Via Cloud Server with Smartphone Controller

Nugroho Giri Jaladri¹, Dian Nova Kusuma Hardani^{2*}

^{1,2}Program Studi Teknik Elektro, Fakultas Teknik dan Sains

Universitas Muhammadiyah Purwokerto

Jl. Raya Dukuh Waluh, Kembaran 53182, Indonesia.

email: ¹nugroho.g.jaladri@gmail.com, ²diannova.kh@ump.ac.id

ABSTRAK

DOI;
10.30595/jrst.v5i2.8303

Histori Artikel:

Diajukan:
20/08/2020

Diterima:
16/08/2022

Diterbitkan:
26/08/2022

Rumah adalah kebutuhan primer sebagai tempat perlindungan agar kita bisa merasa aman dan nyaman di dalamnya. Kesibukan manusia sering kali membuat lupa untuk mematikan peralatan listrik. Hal ini akan membuat pemborosan konsumsi listrik dan menambah biaya. Rumah konvensional juga tidak mempunyai sistem keamanan untuk mencegah suatu bahaya. Maka dari itu diperlukan suatu inovasi untuk menyelesaikan permasalahan tersebut. Sistem otomasi rumah merupakan salah satu solusinya. Dalam penelitian ini dilakukan perancangan sistem otomasi rumah yang terhubung dengan internet melalui jaringan *wifi* menggunakan mikrokontroler Wemos D1 mini. *Cloud server* yang digunakan adalah *cloud server* dari goggle yaitu Firebase. Aplikasi *smartphone* android sebagai pengendalinya dirancang menggunakan App Inventor 2. Sistem ini dipadukan dengan modul Real Time Clock sebagai pewaktu untuk kendali otomatis berdasarkan waktu. Hasil penelitian dari sistem ini dapat mengendalikan lampu, stopkontak, penyiraman kebun, membuka dan menutup tirai, membuka kunci pintu dan sensor keamanan rumah. Sistem ini dapat mengawasi dan mengendalikan peralatan rumah dengan baik dan respons yang cepat dengan rata-rata respons 1,84 detik dari saat dikendalikan. Pengendalian secara otomatis berdasarkan waktu bekerja dengan cukup akurat sesuai waktu yang telah dijadwalkan dengan *error* 6,9 detik. Sistem otomasi rumah ini juga mempunyai sistem keamanan rumah. Alarm keamanan aktif dan muncul sebuah notifikasi di *smartphone* android jika terdeteksi suatu bahaya sehingga pengguna dapat melakukan tindakan yang tepat sebelum terjadi bahaya yang lebih besar.

Kata Kunci: Otomasi Rumah, *Internet of Things*, *Cloud Server*, *Smartphone*, Keamanan Rumah

ABSTRACT

The home is a primary need as a place of protection so that we can feel safe and comfortable in it. Busy people often forget to turn off electrical equipment. It will make electricity consumption waste and increase costs. Conventional homes also do not have a security system to prevent a hazard. Therefore innovation is needed to solve these problems. The home automation system is one solution. In this research, the design of a home automation system that is connected to the internet through a *wifi* network uses a Wemos D1 mini microcontroller. The cloud server used is the cloud server from google,

namely Firebase. Android smartphone application as a controller is designed using App Inventor 2. This system is integrated with the Real-Time Clock module as a timer for automatic control based on time. The results of this system can control lights, power outlets, watering gardens, opening, and closing curtains, door locks, and home security sensors. This system can monitor and control home appliances well and quickly, with an average response of 1.84 seconds when managed. Automatic control based on working time is accurate according to the scheduled time with an error of 6.9 seconds. This home automation system also has a home security system. A security alarm will activate, and a notification will appear on the android smartphone if a danger is detected so that the user can take appropriate action before a higher risk occurs.

Keywords: *Home Automation, Internet of Things, Cloud Server, Smartphone, Home Security*

1. PENDAHULUAN

Rumah merupakan kebutuhan primer bagi manusia. Rumah memiliki fungsi utama sebagai tempat tinggal. Gaya hidup masa kini menuntut desain arsitektur, desain interior dan mekanikal elektrikal (Arifiyanto dkk., 2013). Rumah adalah tempat perlindungan agar kita bisa merasa aman dan nyaman di dalamnya. Kegiatan manusia yang semakin sibuk membuat sering kali harus meninggalkan rumah dalam waktu yang lama.

Kelengahan pemilik rumah konvensional menjadi faktor utama banyaknya tindak kriminal, seperti lupa mengunci pintu atau meninggalkan rumah dalam keadaan lampu yang tidak menyala (Romoadhon & Anamisa, 2017). Hal ini menyebabkan pemborosan daya listrik yang menaikkan jumlah tagihan listrik yang kita bayar. Tidak adanya sistem pengawasan pada rumah menyebabkan tindak kriminal seperti pencurian sangat rentan terjadi.

Perangkat-perangkat listrik akan dapat bekerja secara otomatis sesuai dengan kebutuhan pengguna dengan menerapkan perangkat *Smart Home* di rumah atau perkantoran (Kurnianto dkk., 2016). Pengguna juga dapat memantau dan mengendalikan perangkat-perangkat listrik di dalam rumah dari jarak jauh melalui suatu saluran komunikasi seperti melalui jaringan internet, *Wi-Fi* atau *bluetooth*.

Pada penelitian sebelumnya, oleh Piyare (2013) sistem otomasi menggunakan mikrokontroler arduino dengan koneksi internet menggunakan *ethernet shield*. Otomasi rumah ini dapat dikendalikan dengan *smartphone*. Kelemahan dari penggunaan modul *ethernet shield* ini yaitu fleksibilitas yang kurang karena koneksi internet yang hanya bisa menggunakan kabel.

Penelitian oleh Yan & Shi (2013) menggunakan modul *bluetooth* BF10-A untuk

menghubungkan mikrokontroler dan *smartphone* secara nirkabel, program aplikasi android diprogram menggunakan Eclipse IDE. Kelebihan pada sistem ini adalah tidak memerlukan koneksi internet sehingga tidak tergantung pada ISP, namun mempunyai kekurangan pengendalian yang terbatas pada jarak maksimal 8 meter dari modul *bluetooth*.

Penelitian oleh Romoadhon & Anamisa (2017) membuat sistem *smarthome* menggunakan mikrokontroler NodeMCU. Pada penelitian ini menggunakan aplikasi Blynk untuk akses kontrolnya. Kelebihan dari mikrokontroler NodeMCU ini adalah dapat mengakses koneksi internet melalui jaringan nirkabel *Wi-Fi* sehingga lebih fleksibel sumber internetnya. Kekurangan ada pada aplikasi Blynk karena tidak memprogram sendiri dari awal sehingga fasilitas dan desainnya terbatas.

Berdasarkan uraian tersebut, maka penelitian dilakukan untuk melengkapi metode otomasi rumah yang belum ada. Pada penelitian sebelumnya sistem otomasi menggunakan koneksi lokal saja sehingga mempunyai jarak yang terbatas, maka dari itu diperlukan metode untuk menambah jarak kendali melalui internet menggunakan *cloud server*. Menggunakan aplikasi android untuk mempermudah antarmuka pengguna dalam mengendalikan dan memonitoring sistem. Aplikasi android akan diprogram menggunakan MIT APP inventor 2. Output dari sistem yang tidak hanya *on-off* saja, tetapi juga sinyal PWM untuk mengendalikan motor *stepper*. Metode otomasi rumah juga ditambah dengan menggunakan *network time protocol* (NTP) agar dapat berjalan pada waktu yang ditentukan sebelumnya dan membuat suatu sistem keamanan rumah.

2. METODE PENELITIAN

Dalam melakukan penelitian ini perlu dibuat alur pelaksanaan kegiatan, agar dapat

memaksimalkan waktu supaya lebih efisien dan efektif. Alur penelitian dijelaskan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

2.1. Studi literatur

Tahapan ini adalah langkah awal dalam memulai penelitian yaitu mempelajari literatur yang berkaitan dengan perancangan sistem yang dibuat, baik yang bersumber dari buku, jurnal, penelitian tugas akhir mahasiswa, dan artikel-artikel di internet.

2.2. Perancangan perangkat keras

Pada proses perancangan perangkat keras agar dapat bekerja dengan baik harus dilakukan proses perencanaan yang matang dan didukung oleh perhitungan analisis sistem sebelumnya. Komponen atau bahan-bahan yang digunakan untuk membuat sistem adalah komponen yang berkualitas karena alat ini beroperasi secara terus menerus. Perancangan ini dimulai dari membuat blok sistem catu daya, mikrokontroler, sensor dan keluaran sistem berupa *relay*, kunci pintu solenoid, pompa air serta motor *stepper* dan hasil pengukuran yang datanya dikirim melalui koneksi *Wifi*.

2.3. Pemrograman perangkat keras

Dalam menerjemahkan diagram alir pada suatu sistem, dibutuhkan bahasa pemrograman, salah satunya bahasa C. Pemilihan bahasa pemrograman ini didasarkan pada kemudahan dalam memahami dan membuat suatu program, serta banyaknya literatur yang bisa digunakan.

Pembuatan program pada penelitian ini menggunakan IDE (*Integrated Development Environment*) Visual Studio Code dan PlatformIO sebagai *software compiler*.

2.4. Perancangan perangkat lunak

Perancangan perangkat lunak dilakukan dengan membuat sebuah *database* yang dapat menerima data sensor dari sistem *hardware* dan data perintah dari *smartphone* secara *realtime* yang telah dibuat serta dapat terkoneksi dengan jaringan internet. *Database realtime* dibuat dengan salah satu *platform database* bernama Firebase. Kemudian dibuat sebuah aplikasi android dengan sebuah *web app* yaitu MIT App Inventor 2. Aplikasi ini difungsikan sebagai pengendali aktuator dan penampil data dari sensor.

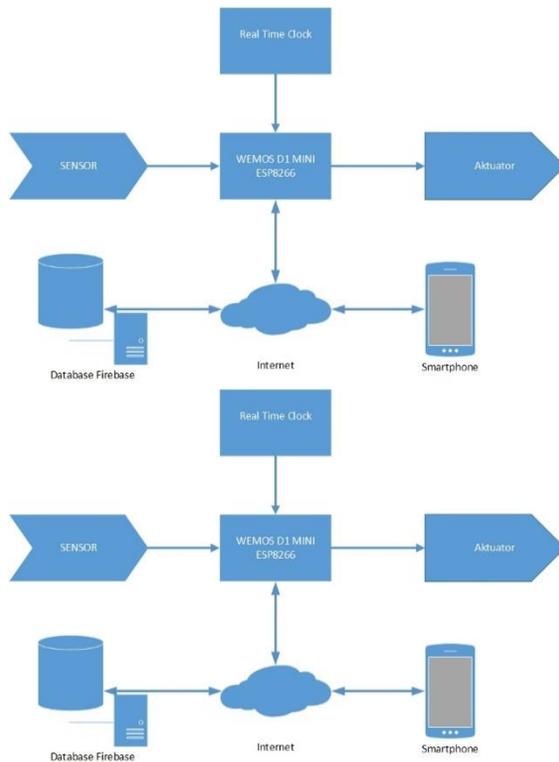
2.5. Pengujian alat

Setelah sistem telah selesai dirancang dan diprogram, maka langkah selanjutnya adalah menguji alat. Pengujian alat dilakukan dengan dengan cara menjalankan sistem secara berulang dan melakukan evaluasi hingga menghasilkan keluaran yang diharapkan.

2.6. Evaluasi dan kesimpulan

Tahapan ini merupakan tahapan dalam mengolah data yang telah didapat dari hasil pengujian. Data tersebut diolah sesuai dengan literatur dan teori-teori yang telah ada pada penelitian terdahulu yang kemudian disusun menjadi sebuah laporan. Untuk mengetahui bagaimana sistem bekerja maka diperlukan sebuah blok diagram sistem seperti yang ditunjukkan pada Gambar 2.

Mikrokontroler Wemos d1 mini ESP8266 adalah pusat kendali dari semua perangkat yang ada. Sensor yang digunakan yaitu sensor arus, *magnetic switch sensor*, sensor kelembaban tanah, sensor gerak, sensor suhu dan kelembaban, dan sensor asap. RTC digunakan untuk memberikan data waktu sebenarnya yang akan dibandingkan dengan jadwal yang telah dibuat sebagai perintah otomasi. Hasil pembacaan sensor dikirimkan melalui internet ke *database* Firebase dan dikirimkan kembali untuk ditampilkan di *smartphone*. *Smartphone* juga digunakan untuk memberi data perintah melalui internet dan *database* Firebase untuk selanjutnya dikirimkan ke Mikrokontroler. Aktuator pada sistem ini adalah lampu, stopkontak, kunci pintu solenoid, motor *stepper* pompa air dan *buzzer*.



Gambar 2. Diagram Blok Sistem

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dibagi menjadi dua bagian, yaitu perancangan dan pemrograman, dan pengujian. Perancangan dan pemrograman digunakan untuk merancang *layout* PCB, menentukan komponen yang digunakan serta membuat program mikrontroler dan aplikasi android. Pengujian dilakukan untuk melihat hasil dari perancangan yang telah dibuat.

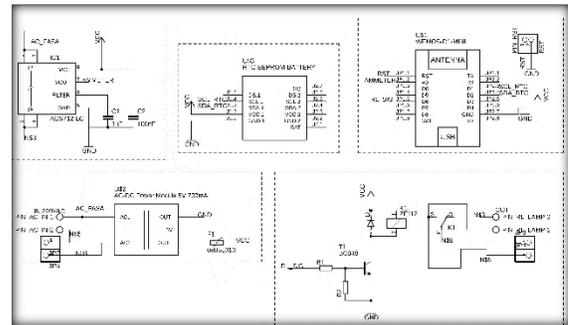
3.1. Perancangan dan Pemrograman

Perancangan dan pemrograman terbagi menjadi tiga, yaitu perancangan perangkat keras, pemrograman perangkat keras dan pemrograman android. Perancangan perangkat keras dibuat menggunakan *software* Eagle sementara pemrograman perangkat keras menggunakan Visual Code Studio, lalu pemrograman android menggunakan MIT App Inventor.

3.1.1. Perancangan Perangkat Keras

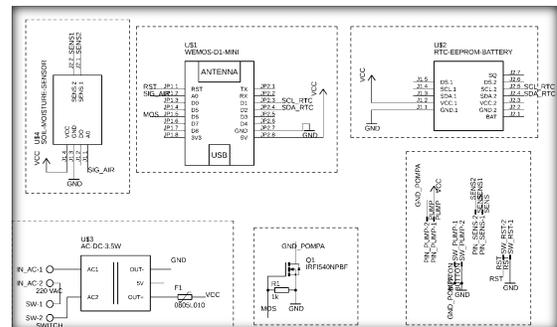
Perancangan perangkat keras ini dilakukan dengan menggunakan *software* Eagle. Perancangan ini dibagi menjadi lima bagian pada masing-masing alat, yaitu otomasi lampu 1, lampu 2 dan stopkontak yang skema

rangkaianya diperlihatkan pada Gambar 3, otomasi kebun yang skema rangkaianya diperlihatkan Gambar 4, otomasi tirai yang skema rangkaianya diperlihatkan pada Gambar 5, kunci pintu IOT yang skema rangkaianya diperlihatkan pada Gambar 6, dan sensor keamanan rumah yang skema rangkaianya diperlihatkan pada Gambar 7.



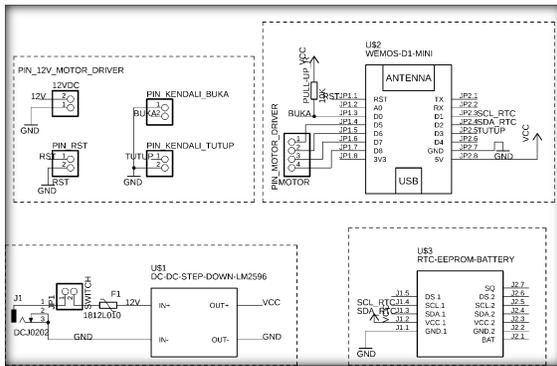
Gambar 3. Skema Rangkaian Otomasi Lampu 1, Lampu 2, dan Stopkontak

Pada Gambar 3, skema rangkaian ini menghubungkan Wemos D1 mini dengan *power supply* 5 volt DC, rangkaian relay, sensor arus ACS712 dan modul RTC DS1307.



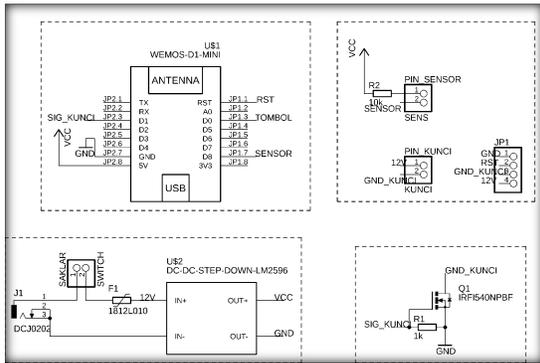
Gambar 4. Skema Rangkaian Otomasi Kebun

Pada Gambar 4, skema rangkaian ini menghubungkan Wemos D1 mini dengan *power supply* 5 volt DC, rangkaian mosfet IRF520, rangkaian sensor kelembaban tanah dan modul RTC DS1307.



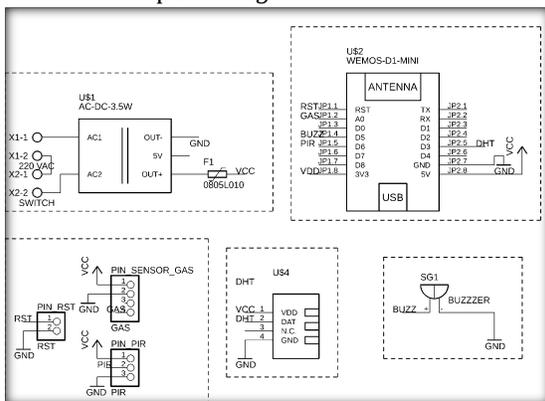
Gambar 5. Skema Rangkaian Otomasi Tirai

Pada Gambar 5, skema rangkaian ini menghubungkan Wemos D1 mini dengan modul *buck converter* LM2596, pin untuk *driver motor* L298N dan modul RTC DS1307.



Gambar 6. Skema Rangkaian Kunci Pintu IOT

Pada Gambar 6, skema rangkaian ini menghubungkan Wemos D1 mini dengan modul *buck converter* LM2596, rangkaian mosfet IRF520, pin untuk kunci pintu solenoid dan pin untuk sensor pintu *magnetic reed switch*.



Gambar 7. Skema Rangkaian Sensor Keamanan Rumah

Pada Gambar 7, skema rangkaian ini menghubungkan Wemos D1 mini dengan *power supply* 5 volt DC, PIN untuk sensor suhu DHT22, pin untuk sensor gas MQ-2, pin untuk sensor gerak *pyroelectric infrared*, dan rangkaian *buzzer*.

3.1.2. Pemrograman Perangkat Keras

Pemrograman pada sistem perangkat keras ini dilakukan dengan menggunakan Visual Studio Code dan PlatformIO, yang dilakukan adalah memberikan perintah kepada Wemos D1 Mini Pro agar dapat mengontrol sistem yang dibuat. Seperti untuk membaca hasil pembacaan sensor, menghubungkan ke jaringan wifi, mengirimkan dan menerima data dari *server*, serta menjalankan aktuator. *Flowchart* dari sistem perangkat keras ditunjukkan pada Gambar 8 berikut.



Gambar 8. Flowchart Sistem Perangkat Keras

Pada Gambar 8 dijelaskan bagaimana proses kerja dari sistem otomasi rumah bagian perangkat keras. Tahapan awal adalah inisialisasi sistem yaitu menentukan *library* yang digunakan, pengecekan alamat *input output*, dan lainnya. Bagian selanjutnya adalah menghubungkan mikrokontroler dengan jaringan internet agar semua sistem dapat bekerja. Setelah terhubung dengan internet dan Firebase, perangkat keras akan mengirim data sensor dan mengambil data perintah dari Firebase dan menuliskannya pada

EEPROM. Jika tidak terhubung maka mikrokontroler akan membaca data perintah sebelumnya yang ada pada EEPROM. Selanjutnya mikrokontroler akan menjalankan perintah yang sudah ada.

3.1.3. Pemrograman Perangkat Lunak Aplikasi Android

Pemrograman perangkat lunak terbagi menjadi 2, yaitu pembuatan *cloud server* atau *database* Firebase dan pembuatan aplikasi android. Pembuatan *cloud server* Firebase dengan cara mendaftarkan ke Firebase menggunakan akun Google untuk membuka database baru. Setelah membuat *database*, akan mendapat alamat *hosting project* dan kode autentifikasi. Alamat *hosting* dan kode autentifikasi rahasia ini yang digunakan untuk memberi akses database kepada perangkat keras dan aplikasi android.

Pembuatan aplikasi android dilakukan dengan menggunakan App Inventor 2. Aplikasi android ini adalah antar muka pengguna yang digunakan untuk memberi perintah ke perangkat keras otomasi rumah dan untuk pembacaan sensor melalui akses Firebase. *Flowchart* sistem pada bagian *smartphone* ditunjukkan pada Gambar 9 berikut.



Gambar 9. Flowchart Sistem Smartphone

Pada Gambar 9 dijelaskan bagaimana proses kerja pada *smartphone*. Proses awal adalah menghubungkan dengan jaringan

internet. Jika sudah terhubung dengan internet dan Firebase, *smartphone* akan menerima pembacaan sensor yang ada di Firebase dan mengirimkan data perintah ke Firebase.

3.2. Pengujian

Pengujian dilakukan untuk mengetahui apakah sistem sudah bekerja dengan baik. Pengujian yang dilakukan berupa pengujian fungsi alat dengan kendali *smartphone*, pengujian otomasi alat, dan pengujian sensor. Pengujian ini akan dijelaskan pada tiap masing-masing alat.

3.2.1. Otomasi Lampu 1, Lampu 2 dan Stopkontak

Pengujian ini mencoba menyalakan lampu 1, lampu 2, dan stopkontak melalui aplikasi android. Pengujian dilakukan sebanyak 10 kali dengan menekan tombol "Nyalakan" dan "Padamkan" pada aplikasi android. Hasilnya terlihat pada Tabel 1, Tabel 2, dan Tabel 3 berikut.

Tabel 1. Pengujian Lampu 1 Terhadap Kendali Smartphone

No.	Kendali	Respons	Delay (detik)
1	Nyalakan	Menyala	1,80
2	Padamkan	Padam	1,95
3	Nyalakan	Menyala	1,53
4	Padamkan	Padam	1,71
5	Nyalakan	Menyala	1,87
6	Padamkan	Padam	0,76
7	Nyalakan	Menyala	0,63
8	Padamkan	Padam	1,44
9	Nyalakan	Menyala	0,84
10	Padamkan	Padam	0,99
Rata-rata delay (detik)			1,53

Tabel 2. Pengujian Lampu 2 Terhadap Kendali Smartphone

No.	Kendali	Respons	Delay (detik)
1	Nyalakan	Menyala	2,59
2	Padamkan	Padam	2,52
3	Nyalakan	Menyala	2,51
4	Padamkan	Padam	1,45
5	Nyalakan	Menyala	1,26

6	Padamkan	Padam	1,35
7	Nyalakan	Menyala	2,43
8	Padamkan	Padam	1,26
9	Nyalakan	Menyala	2,07
10	Padamkan	Padam	2,34
Rata-rata <i>delay</i> (detik)			1,98

Tabel 3. Pengujian Stopkontak Terhadap Kendali *Smartphone*

No.	Kendali	Respons	<i>Delay</i> (detik)
1	Nyalakan	Menyala	0,99
2	Padamkan	Padam	2,79
3	Nyalakan	Menyala	2,79
4	Padamkan	Padam	2,88
5	Nyalakan	Menyala	2,16
6	Padamkan	Padam	2,25
7	Nyalakan	Menyala	2,29
8	Padamkan	Padam	2,25
9	Nyalakan	Menyala	2,79
10	Padamkan	Padam	0,63
Rata-rata <i>delay</i> (detik)			2,18

Hasil pengujian yang terlihat pada Tabel 1 menyatakan bahwa waktu respons menyalanya lampu 1 mempunyai rata-rata *delay* 1,53 detik dari saat tombol ditekan dari *smartphone*, pada Tabel 2 terlihat lampu 2 mempunyai rata-rata *delay* 1,98 dan pada Tabel 3 terlihat stopkontak mempunyai rata-rata *delay* 2,18 detik.

Pengujian selanjutnya menguji otomasi lampu menggunakan penjadwalan waktu. Pengujian dilakukan dengan membandingkan waktu lampu menyala dan padam dengan waktu yang sebenarnya. Pengujian dilakukan sebanyak 5 kali dengan mengubah-ubah jadwal. Hasil pengujian terlihat pada Tabel 4, Tabel 5, Tabel 6, Tabel 7, Tabel 8, dan Tabel 9 berikut.

Tabel 4. Pengujian Otomasi Lampu 1 Menyala Berdasarkan Jadwal

No	Jadwal Waktu Menyala	Respons Waktu Menyala	<i>Error</i> (detik)
1	06:40:00	06:40:07	7
2	06:42:00	06:42:08	8

3	06:44:00	06:44:07	7
4	06:46:00	06:46:09	9
5	06:48:00	06:48:05	5
Rata-rata <i>error</i> (detik)			7,2

Tabel 5. Pengujian Otomasi Lampu 1 Padam Berdasarkan Jadwal

No	Jadwal Waktu Padam	Respons Waktu Padam	<i>Error</i> (detik)
1	06:41:00	06:41:08	8
2	06:43:00	06:43:09	9
3	06:45:00	06:45:06	6
4	06:47:00	06:47:07	7
5	06:49:00	06:49:08	8
Rata-rata <i>error</i> (detik)			7,6

Hasil pengujian pada Tabel 4 dan Tabel 5 terlihat bahwa sistem bekerja dengan baik. Sistem dapat menyalakan dan mematikan lampu 1 sesuai jadwal. Namun terdapat *error* selisih waktu rata-rata selama 7,4 detik.

Tabel 6. Pengujian Otomasi Lampu 2 Menyala Berdasarkan Jadwal

No.	Jadwal Waktu Menyala	Respons Waktu Menyala	<i>Error</i> (detik)
1	05:48:00	05:48:06	6
2	05:50:00	05:50:06	6
3	05:52:00	05:52:05	5
4	05:54:00	05:54:07	7
5	05:56:00	05:56:06	6
Rata-rata <i>error</i> (detik)			6

Tabel 7. Pengujian Otomasi Lampu 2 Padam Berdasarkan Jadwal

No.	Jadwal Waktu Padam	Respons Waktu Padam	<i>Error</i> (detik)
1	05:49:00	05:49:08	8
2	05:51:00	05:51:08	8
3	05:53:00	05:53:06	6
4	05:55:00	05:55:13	13
5	05:57:00	05:57:08	8
Rata-rata <i>error</i> (detik)			8,6

Hasil pengujian pada Tabel 6 dan Tabel 7 terlihat bahwa sistem bekerja dengan baik. Sistem dapat menyalakan dan mematikan lampu 2 sesuai jadwal. Namun terdapat *error* selisih waktu rata-rata selama 7,3 detik.

Tabel 8. Pengujian Otomasi Stopkontak Menyala Berdasarkan Jadwal

No.	Jadwal Waktu Menyala	Respons Waktu Menyala	Error (detik)
	Waktu	Waktu	
1	06:12:00	06:12:08	8
2	06:14:00	06:14:06	6
3	06:16:00	06:16:09	6
4	06:18:00	06:18:09	9
5	06:20:00	06:20:06	6
Rata-rata error (detik)			7

Tabel 9. Pengujian Otomasi Stopkontak Padam Berdasarkan Jadwal

No.	Jadwal Waktu Padam	Respons Waktu Padam	Error (detik)
	Waktu	Waktu	
1	06:13:00	06:13:06	6
2	06:15:00	06:15:08	8
3	06:17:00	06:17:07	7
4	06:19:00	06:19:06	6
5	06:21:00	06:21:07	7
Rata-rata error (detik)			6,8

Hasil pengujian pada Tabel 8 dan Tabel 9 terlihat bahwa sistem bekerja dengan baik. Sistem dapat menyalakan dan mematikan stopkontak sesuai jadwal. Namun terdapat *error* selisih waktu rata-rata selama 6,9 detik.

Sistem otomasi dapat bekerja dengan normal. Terdapat *error* dengan rata-rata *error* 7,3 detik saat lampu dan stopkontak menyala atau padam dari yang telah dijadwalkan karena program yang harus membaca nilai yang tersimpan pada EEPROM dan melakukan perhitungan serta perbandingan dengan waktu sebenarnya terlebih dahulu untuk menjalankan perintah. Selain itu juga karena program lain yang masih bekerja dan harus dieksekusi oleh program.

Pengujian berikutnya adalah menguji saklar manual yang ada pada perangkat keras. Saklar manual ini dipasang dengan tujuan agar lampu dapat dikendalikan secara konvensional. Sehingga jika terjadi kerusakan pada sistem otomasi, lampu masih tetap bisa dinyalakan dan dipadamkan secara manual. Saklar ini dipasang secara paralel dengan sistem sehingga tidak terpengaruh dengan keadaan sistem otomasi. Hasilnya saklar bekerja dengan baik, lampu dapat dinyalakan dan dipadamkan menggunakan saklar manual ini.

3.2.2. Otomasi Kebun

Pengujian selanjutnya adalah menghidupkan dan mematikan pompa menggunakan *smartphone* dengan cara menekan tombol "Siram" dan tombol "hentikan". Pengujian dilakukan sebanyak 10 kali. Hasil dari pengujian terlihat pada Tabel 10 berikut.

Tabel 10. Respons Pompa Terhadap Pengendalian *Smartphone*

No	Kendali	Respons Pompa	Delay (detik)
1	Siram	Hidup	2,61
2	Hentikan	Mati	2,07
3	Siram	Hidup	4,59
4	Hentikan	Mati	3,24
5	Siram	Hidup	2,79
6	Hentikan	Mati	4,41
7	Siram	Hidup	4,50
8	Hentikan	Mati	2,79
9	Siram	Hidup	3,51
10	Hentikan	Mati	3,06
Rata-rata delay (detik)			3,36

Terlihat bahwa respons dari pompa sudah sesuai dengan perintah meskipun ada *delay*. Rata-rata *delay* adalah 3,36 detik. *Delay* dipengaruhi oleh kecepatan koneksi internet dan program lain yang berjalan.

Pengujian selanjutnya adalah menguji otomasi dari perangkat keras. Otomasi ini berdasarkan jadwal waktu penyiraman dan kelembaban tanah. Batas atas kelembaban tanah sudah diatur dalam program. Menurut Sawidin dkk., (2016), kelembaban tanah yang baik untuk tanaman bunga yaitu 90%. Jadwal diatur untuk penyiraman pagi hari pukul 07.00 WIB dan penyiraman sore hari pukul 17.00 WIB. Pengujian dilakukan selama 5 hari mulai dari tanggal 1 Juli 2020 s/d 5 Juli 2020. Hasil dari pengujian ini ditunjukkan pada Tabel 11 berikut.

Tabel 11. Pengujian Pompa Berdasarkan Jadwal dan Humidity (Kelembaban) Tanah

Tanggal	Waktu (WIB)	Humidity Tanah	Humidity Tanah
		Sebelum Disiram (%)	Setelah Disiram (%)
1 Juli 2020	07.00	87	90
	17.00	87	90
	07.00	86	90

2 Juli 2020	17.00	84	90
3 Juli 2020	07.00 17.00	86 85	90 90
4 Juli 2020	07.00 17.00	86 84	90 90
5 Juli 2020	07.00 17.00	86 86	90 90

Hasil pengujian ini menunjukkan bahwa otomasi kebun sudah berjalan sesuai dengan rencana. Sistem dapat berkerja menghidupkan pompa sesuai jadwal dan mengatur kelembaban tanah mencapai 90% sesuai kelembaban yang telah ditentukan.

Pengujian berikutnya adalah menghidupkan pompa menggunakan tombol pada perangkat keras. Tombol ini dipasang untuk langsung menghantarkan tegangan 5 volt ke pompa. Hasil pengujian ini berjalan sesuai rencana. Saat tombol ditekan pompa berjalan dengan normal.

3.2.3. Otomasi Tirai

Pengujian dilakukan dengan menguji respons motor *stepper* terhadap perintah melalui *smartphone*. Alat ini dipasang pada miniatur tirai dengan lebar tirai dalam posisi terbuka penuh adalah 11,8 cm. Pengujian dilakukan dengan cara menggeser-geser *slider* pada menu Tirai. Persentase kondisi tirai diperoleh dari perhitungan sebagai berikut.

$$Kondisi\ tirai\ \% = \frac{Panjang\ tirai\ terbuka\ (cm)}{11,8\ (cm)} \times 100\% \quad (1)$$

Hasil pengujian ditunjukkan pada Tabel 12 berikut.

Tabel 12. Pengujian Kendali Tirai Berdasarkan Kendali *Smartphone*

No	Nilai Slider (%)	Respons Lebar Tirai (cm)	Kondisi Tirai Terbuka (%)	Error (%)
1	100	11,8	100	0
2	80	9,5	80,5	0,5
3	60	7	59,3	0,7
4	40	4,5	38,1	1,9
5	20	2,2	18,6	1,4
6	0	0	0	0
Rata-rata error (%)				0,75

Hasil dari pengujian ini adalah sistem otomasi tirai berjalan dengan baik. Tirai dapat menutup dan membuka sesuai dengan kendali *slider* dari *smartphone*. Terdapat sedikit *error* dengan rata-rata *error* sebesar 0,75 %.

Pengujian selanjutnya adalah menutup dan membuka tirai sesuai jadwal. Jadwal diatur melalui aplikasi android. Hasil dari pengujian ini ditunjukkan pada Tabel 13 dan Tabel 14 berikut.

Tabel 13. Pengujian Kendali Tirai Menutup Berdasarkan Jadwal

No	Pengaturan Jadwal Tutup Tirai	Waktu Respons Tirai Menutup	Error (detik)
1	13:05:00	13:05:05	5
2	13:07:00	13:07:07	7
3	13:09:00	13:09:05	5
4	13:11:00	13:11:07	7
5	13:13:00	13:13:07	7
Rata-rata error (detik)			6,2

Tabel 14. Pengujian Kendali Tirai Membuka Berdasarkan Jadwal

No	Pengaturan Jadwal Buka Tirai	Waktu Respons Tirai Membuka	Error (detik)
1	13:06:00	13:06:04	4
2	13:08:00	13:08:04	4
3	13:10:00	13:10:05	5
4	13:12:00	13:12:04	4
5	13:14:00	13:14:05	5
Rata-rata error (detik)			5,3

Hasil dari pengujian ini adalah tirai bekerja dengan baik secara otomasi sesuai jadwal. Terdapat *error* rata-rata selama 5,3 detik. *Error* ini terjadi karena program yang harus mengambil data dari EEPROM, selanjutnya melakukan perhitungan dan juga ada program lain yang harus dieksekusi terlebih dahulu.

Pengujian berikutnya adalah mencoba menggerakkan tirai menggunakan tombol fisik yang ada pada perangkat keras otomasi tirai ini. Terdapat dua tombol kendali fisik yaitu tombol untuk membuka dan tombol untuk menutup. Hasilnya tirai bergerak sesuai dengan tombol yang ditekan.

3.2.4. Kunci Pintu IOT

Pengujian berikutnya adalah melihat respons kunci saat tombol “Buka Kunci” pada aplikasi android ditekan. Pengujian dilakukan sebanyak 5 kali. Hasilnya ditunjukkan pada Tabel 15 berikut.

Tabel 15. Pengujian Respons Kunci Terhadap Kendali *Smartphone*

No	Respons Kunci	Delay (detik)
1	Terbuka	1,53
2	Terbuka	1,62
3	Terbuka	2,16
4	Terbuka	1,35
5	Terbuka	1,80
Rata-rata delay		1,69

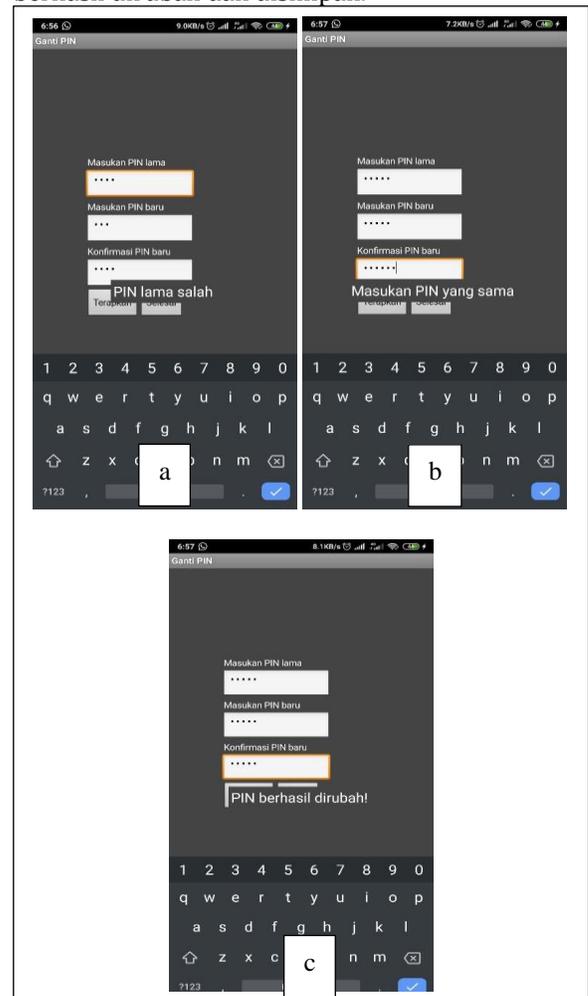
Hasil dari dari pengujian ini adalah kunci merespons membuka kunci saat tombol “Buka Kunci” ditekan. Terdapat rata-rata delay selama 1,69 detik. Delay dipengaruhi oleh kecepatan koneksi internet.

Pengujian selanjutnya adalah menguji penggunaan PIN untuk membuka kunci. Pengujian dilakukan 2 kali, pertama untuk PIN yang benar dan kedua untuk PIN yang salah. Respons saat PIN yang dimasukkan benar adalah kunci langsung terbuka. Sedangkan respons Saat PIN yg dimasukkan salah adalah muncul *alert notification* bahwa pin yang dimasukkan salah seperti terlihat pada Gambar 10 dan kunci tidak terbuka.



Gambar 10. Alert Notification PIN Salah

Pengujian berikutnya adalah mengubah PIN lama menjadi PIN yang baru. Pada pengujian ini PIN yang lama adalah “12345”. Kemudian PIN akan diubah menjadi “54321”. Percobaan pertama adalah memasukkan PIN yang salah. Hasilnya adalah muncul *alert notification* “PIN lama salah” ditunjukkan Gambar 11 (a) dan PIN tidak berhasil dirubah. Percobaan kedua adalah memasukkan PIN yang benar tetapi memasukkan PIN baru yang berbeda dengan PIN konfirmasinya. Hasilnya adalah muncul *alert notification* “Masukkan PIN yang sama” ditunjukkan Gambar 11 (b) dan PIN tidak berhasil dirubah. Percobaan ketiga adalah memasukkan PIN lama yang benar dan memasukkan PIN baru serta PIN konfirmasinya dengan sama. Hasilnya adalah muncul *alert notification* “PIN berhasil dirubah” ditunjukkan Gambar 11 (c) dan PIN berhasil dirubah dan disimpan.



Gambar 11. Alert Notification PIN Lama Salah (a), PIN Konfirmasi Salah (b) dan PIN berhasil dirubah (c)

3.2.5. Sensor Keamanan Rumah

Pengujian berikutnya adalah melakukan monitoring suhu dan kelembaban ruangan. Pengujian dilakukan sebanyak 5 kali pada waktu yang berbeda. Hasil dari pengujian ini ditunjukkan pada Tabel 16 berikut.

Tabel 16. Pengujian Monitoring Suhu dan Kelembaban Ruangan

No	Waktu (WIB)	Suhu (°C)	Kelembaban (%)
1	04.00	29,6	69,6
2	08.00	31,7	61,9
3	12.00	32,9	57,3
4	16.00	33,5	59,4
5	20.00	33,1	56,4

Pengujian selanjutnya adalah pengujian sistem keamanan berdasarkan suhu. Pengujian ini dilakukan dengan cara mendekatkan solder yang menyala dengan sensor suhu DHT22. Hasil dari pengujian ditunjukkan pada Tabel 17 berikut.

Tabel 17. Pengujian Sistem Keamanan Berdasarkan Suhu

No	Menit ke	Suhu (°C)	Alarm	Notifikasi Android
1	0	29,8	Tidak aktif	Tidak ada
2	3	31,9	Tidak aktif	Tidak ada
3	6	36,9	Tidak aktif	Tidak ada
4	9	41	Aktif	Ada
5	12	43,9	Aktif	Ada

Pengujian selanjutnya adalah menguji sistem keamanan berdasarkan sensor gas. Pengujian dilakukan sebanyak 5 kali dengan cara memaparkan gas korek api secara langsung ke sensor MQ-2. Hasil dari pengujian ditunjukkan pada Tabel 18 berikut.

Tabel 18. Pengujian Sistem Keamanan Berdasarkan Sensor Gas

No	Paparan Gas (ppm)	Alarm	Notifikasi Android
1	302	Aktif	Muncul
2	883	Aktif	Muncul
3	1023	Aktif	Muncul
4	1554	Aktif	Muncul
5	1698	Aktif	Muncul

Pengujian selanjutnya adalah menguji sistem keamanan berdasarkan sensor gerak saat mode rumah kosong diaktifkan. Sensor dipasang pada tiang pintu dengan jarak 2,3 meter dari lantai. Pengujian dilakukan sebanyak 5 kali dengan cara menggerak-gerakan badan didepan sensor PIR. Hasil dari pengujian ditunjukkan pada Tabel 19 berikut.

Tabel 19. Pengujian Sistem Keamanan Berdasarkan Sensor Gerak

Jarak Sensor Dengan Objek (cm)	Deteksi Gerakan	Alarm	Notifikasi Android
210	Tidak	Tidak aktif	Tidak muncul
180	Terdeteksi	Aktif	Muncul
150	Terdeteksi	Aktif	Muncul
120	Terdeteksi	Aktif	Muncul
90	Terdeteksi	Aktif	Muncul

Dari hasil pengujian sensor gerak di atas, sensor gerak dapat mendeteksi gerakan hingga 180 cm atau 1,8 meter di depan sensor. Jika mode rumah kosong diaktifkan dan sensor mendeteksi gerakan, alarm pada alat akan aktif dan notifikasi pada *smartphone* android juga muncul sebagai peringatan bahwa ada gerakan di dalam rumah.

4. KESIMPULAN

Hasil dari penelitian ini adalah semua perangkat sistem otomasi rumah dapat bekerja dengan baik. Sistem dapat mematikan dan menghidupkan lampu dan stopkontak, menyiram tanaman secara otomatis, membuka tutup tirai, membuka kunci pintu melalui *smartphone*, dan menambah tingkat keamanan menggunakan sensor.

Terdapat sedikit *delay* (jeda waktu) dengan rata-rata 1,84 detik saat mengendalikan sistem. Sedikit *error* untuk sistem jadwal juga terjadi dengan rata-rata *error* 6,9 detik. Jeda waktu yang terjadi dapat dikarenakan karena 2 hal, yaitu koneksi internet yang lambat dan mikrokontroler yang dalam pembacaan programnya harus menyelesaikan loop terlebih dahulu sebelum datanya ter-*update*. Sementara *error* pada sistem jadwal dapat dikarenakan *server* NTP (*Network Time Protocol*) yang berbeda antara *server* NTP yang digunakan oleh

mikrokontroler dengan server NTP yang digunakan *smartphone*.

DAFTAR PUSTAKA

Arifiyanto, F., Syafei, W. A., & Somantri, M. (2013). Perancangan Prototype Web-Based Online Smart Home Controlled By Smartphone. *Transient*, 2(3), 1-8.

Kurnianto, D., Hadi, A. M., & Wahyudi, E. (2016). Perancangan Sistem Kendali Otomatis pada Smart Home menggunakan Modul Arduino Uno. *Jurnal Nasional Teknik Elektro*, 5(2), 260.
<https://doi.org/10.25077/jnte.v5n2.276.2016>

Piyare, R. (2013). Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone. *International Journal of Internet of Things*, 2013(1), 5-11.
<https://doi.org/10.5923/j.ijit.20130201.02>

Romoadhon, A. S., & Anamisa, D. R. (2017). Sistem Kontrol Peralatan Listrik pada Smart Home Menggunakan Android. *Rekayasa*, 10(2), 116.
<https://doi.org/10.21107/rekayasa.v10i2.3613>

Sawidin, S., Engelin Melo, O., & Marsela, T. (2016). Monitoring Kontrol Greenhouse untuk Budidaya Tanaman Bunga Krisan dengan LabView. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 4(4).
<https://doi.org/10.22146/jnteti.v4i4.169>

Yan, M., & Shi, H. (2013). Smart Living Using Bluetooth-Based Android Smartphone. *International Journal of Wireless & Mobile Networks*, 5(1), 65-72.
<https://doi.org/10.5121/ijwmn.2013.5105>