Jurnal Teknik Elektro, Vol. 11, No. 2, September 2018, 60-65
ISSN 1411-870X
DOI: 10.9744/jte.11.2.60-65

Pembuatan Sistem *Home Automation* Berbasiskan *Internet of Things*

Geraldy Merkusi Limanta, Resmana Lim, Handry Khoswanto
Program Studi Teknik Elektro,Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya - 60236, Indonesia
Email: m23413016@john.petra.ac.id ; resmana@petra.ac.id ; handry@petra.ac.id

Abstrak--Perancangan sistem *home automation* ini dilakukan untuk membuat sebuah sistem yang dapat dikendalikan secara otomatis maupun manual. Sistem yang dibuat ini mengendalikan lampu LED, *horizontal blind*, dan AC. Pembuatan sistem *smart home* ini menggunakan *microcontroller* Raspberry Pi 3 model B. Proses pengiriman data dan kontrol menggunakan protokol internet. Sistem ini menggunakan *ubidots cloud*. *Ubidots* cloud bertugas untuk menyimpan data dan sekaligus sebagai panel kontrol untuk melakukan kendali sistem *smart* home. Sistem yang dibuat selain dapat dikendalikan dari jarak jauh juga dapat dikendalikan secara langsung di tempat. Pengendalian secara langsung di tempat dilakukan dengan menggunakan tombol. Teknik pengendalian lampu menggunakan PWM, pengaturan buka tutup dari *blind* menggunakan motor stepper, dan kontrol AC menggunakan LED *Infrared*. Hasilnya, proses otomatis dan manual dapat berjalan dengan baik. Proses kontrol menggunakan *cloud ubidots* juga berhasil. Proses pengambilan data dari *cloud ubidots* membutuhkan waktu kurang dari 1 detik. Sedangkan proses pengiriman data membutuhkan waktu paling lama 5 detik.

Kata kunci: *Smart Home*, Raspberry Pi, *Ubidots cloud*

I. PENDAHULUAN

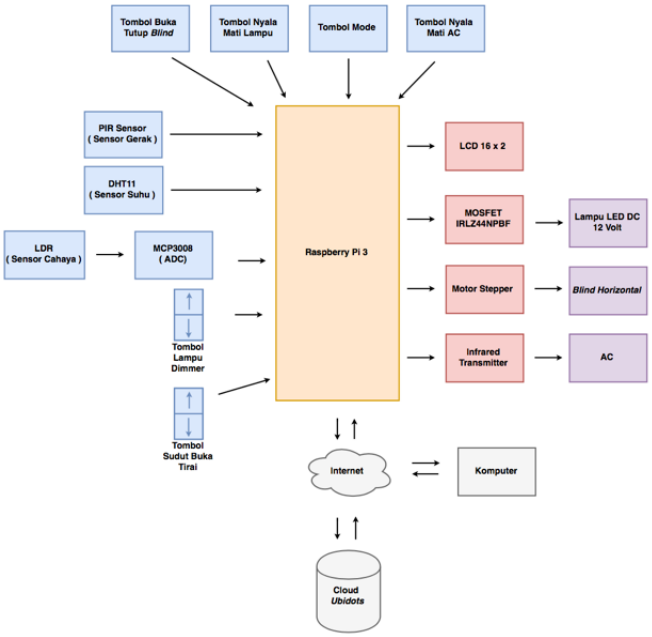
Pada perkembangan teknologi sekarang ini, kemudahan dan kepraktisan merupakan hal yang dicari manusia. Salah satu bentuk teknologi yang memudahkan manusia adalah *Smart Home*. *Smart Home* merupakan salah satu bentuk sistem *home automation* yang digabungkan ke dalam desain arsitektur sebuah rumah atau bangunan [1]. Prinsip dari *Smart Home* adalah merasakan, melihat, mengantisipasi, dan merespon setiap aktifitas yang terjadi di dalam sebuah rumah [2]. *Smart Home* menggunakan mikrokontroler yang secara otomatis memonitor kondisi rumah dan mengendalikan perangkat listrik tertentu.

Selama ini sistem *Smart Home* yang sudah ada masih sedikit yang sudah terhubung dengan jaringan internet. Padahal apabila sistem *Smart Home* dapat terintegrasi dengan jaringan internet, terdapat berbagai keuntungan yang bisa didapat. Salah satu bentuk sistem yang terhubung dengan internet adalah sistem yang berbasis *cloud IOT* (*Internet Of Things*). Keuntungan yang didapat dari sistem *cloud IOT* ini antara lain adalah sistem yang dibuat menjadi lebih aman, kemudahan untuk mengimplementasikan sistem di banyak perangkat dan kemudahan untuk mengimplementasikan sistem di banyak *platform*.

Pada kenyataannya penggunaan sistem *home automation* berbasis *cloud IOT* masih sedikit sekali pengaplikasiannya. Dengan melihat kondisi ini, maka perlu dilakukan pengembangan terhadap sistem *home automation* yang terhubung dengan *cloud IOT*. *IOT* disini dimanfaatkan sebagai basis jaringan sistem untuk tempat komunikasi antar perangkat yang tersambung melalui jaringan internet. Platform *IOT* yang dipilih adalah Ubidots [3].

II. PERANCANGAN SISTEM

Berikut adalah skema sistem secara keseluruhan.

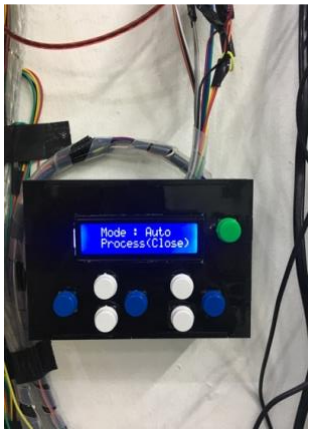


Gambar 1. Skema Sistem

Gambar 1 menunjukkan desain sistem yang akan dibuat. Pada sistem yang akan dibuat, *microcontroller* yang digunakan menggunakan Raspberry Pi 3 Model B. *Input* sensor yang digunakan menggunakan 1 buah sensor PIR sebagai sensor gerak, 1 buah sensor DHT11 sebagai sensor suhu, dan 1 buah sensor LDR sebagai sensor cahaya. Sedangkan *output* yang dikendalikan adalah 24 buah lampu LED DC 12 Volt, 1 buah *blind horizontal, dan* 1 buah AC. LCD berrfungsi sebagai *display*.

A. Desain Hardware

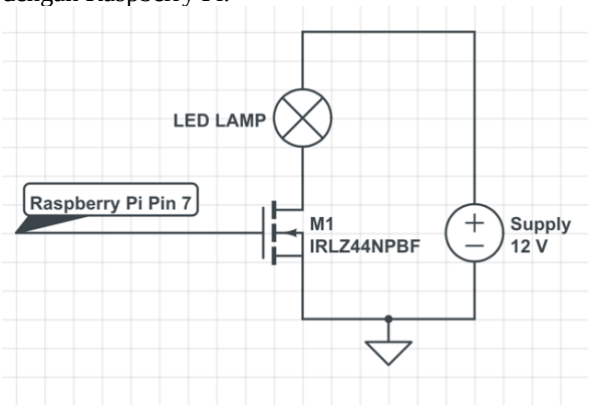
Desain *hardware* yang digunakan pada sistem ini terdiri dari 3 bagian. Bagian pertama yakni pembuatan kotak akrilik yang berisi LCD 16 x 2 dan 8 buah *tactile switch*. Bagian kedua yaitu sistem *home automation* itu sendiri yang terdiri dari Raspberry Pi dan rangkaian kontrol. Bagian ketiga yaitu tampilan panel kontrol pada *dashboard Ubitdots*.



Gambar 2. Desain Pembuatan Kotak Akrilik

Kotak rangkaian kontrol manual ini memiliki fungsi pengaturan kendali sistem *smart home* secara langsung di tempat melalui penekanan tombol yang disediakan. LCD berfungsi sebagai *display* supaya pengguna dapat mengetahui mode yang sedang dijalankan oleh sistem. Dari 8 tombol yang ada, tombol warna hijau digunakan untuk melakukan pemilihan mode. Sedangkan 3 tombol berwarna biru digunakan untuk menyalakan atau mematikan AC, membuka atau menutup *blind*, serta menyalakan dan mematikan lampu. Terakhir 4 tombol berwarna putih digunakan untuk mengatur tingkat kecerahan dari lampu dan mengatur sudut buka dari *blind*

Komunikasi LCD yang digunakan menggunakan mode 4 bit dimana pada mode ini hanya membutuhkan total 6 pin yang terhubung ke Raspberry Pi. MOSFET yang digunakan adalah MOSFET IRLZ44NPBF. Pemilihan MOSFET IRLZ44NPBF berdasarkan arus beban lampu LED dan nilai tegangan *gate threshold* untuk mencapai saturasi. Kebutuhan arus beban untuk menyalakan 1 buah rangkaian lampu yang terdiri dari 12 lampu LED sebesar 6 amper, sedangkan MOSFET IRLZ44NPBF dapat dialiri arus hingga 47 amper [4]. Gambar 3 menunjukkan rangkaian kontrol yang dibuat antara MOSFET dengan Raspberry Pi.



Gambar 3. Rangkaian kontrol Raspberry Pi pada MOSFET

Motor stepper yang digunakan dikendalikan oleh sebuah *module driver* ULN2003. Pemilihan *driver* ULN2003 ini berdasarkan arus yang mengalir pada motor stepper. Kebutuhan arus pada setiap pin motor stepper sekitar 100 mA. Berdasarkan *datasheet*, ULN2003 ini dapat dialiri arus maksimal pada setiap pin nya sebesar 500 mA [5]. Untuk sensor PIR dan DHT11, pemasangan dilakukan menggunakan kabel yang terhubung secara langsung menuju Raspberry Pi. Untuk sensor LDR, pemasangan dilakukan pada IC ADC MCP3008 pada salah satu *channel*-nya. Terdapat 4 pin dari MCP3008 yang menuju ke Raspberry Pi yaitu adalah MISO, MOSI, CLOCK, dan CS0. Protokol yang digunakan untuk pengiriman data dari MCP3008 ke Raspberry Pi menggunakan protocol SPI (*Serial Peripheral Interface*).

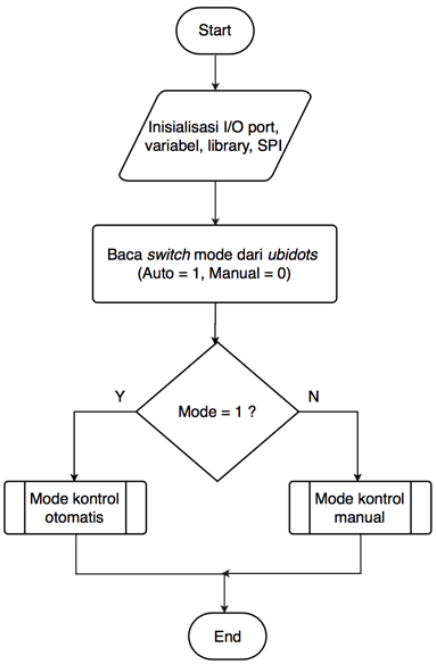
Switch yang digunakan adalah tactile switch. Pemasangan *switch* ini menggunakan rangkaian *internal pull up resistor* dari Raspberry Pi. *Switch* ini akan dibaca sebagai aktif *LOW* ketika ditekan. Konfigurasi yang dilakukan adalah menyambungkan salah satu pin *switch* pada *ground* Raspberry Pi dan menyambungkan pin *switch* yang lain pada pin Raspberry Pi sesuai yang tertera pada table 3.1. Pada setiap *switch*, diberi penambahan kapasitor keramik sebesar 82pf untuk menjaga *switch* dari gangguan *noise*.

Gambar 4. Tampilan *dasboard ubidots*

Gambar 4 menunjukkan tampilan panel kontrol pada *dashboard Ubidots* yang dibuat. *Switch mode* digunakan untuk pemilihan *mode* yang akan digunakan. Apabila *switch mode* bernilai *high*, maka sistem akan berada dalam kondisi otomatis. Sebaliknya apabila bernilai *low*, maka sistem akan berjalan dalam kondisi manual.

B. Desain Software

Bagian ini merupakan bagian yang paling penting dalam membentuk sistem *home automation* ini.



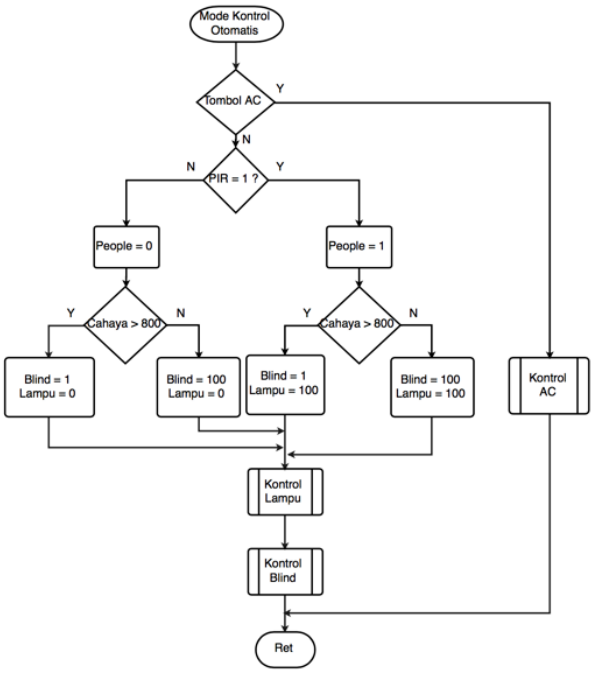
Gambar 5. Flowchart pemilihan mode kontrol

Gambar 5 menunjukkan *flowchart* pemilihan mode kontrol. Hal pertama ketika program mulai berjalan, program akan menanyakan mode kontrol yang dipilih. Cara memilih mode ini dapat dilakukan dengan menekan *switch* mode yang

terpasang pada kotak akrilik atau melalui dashboard *ubidots* dengan menekan *switch* mode.

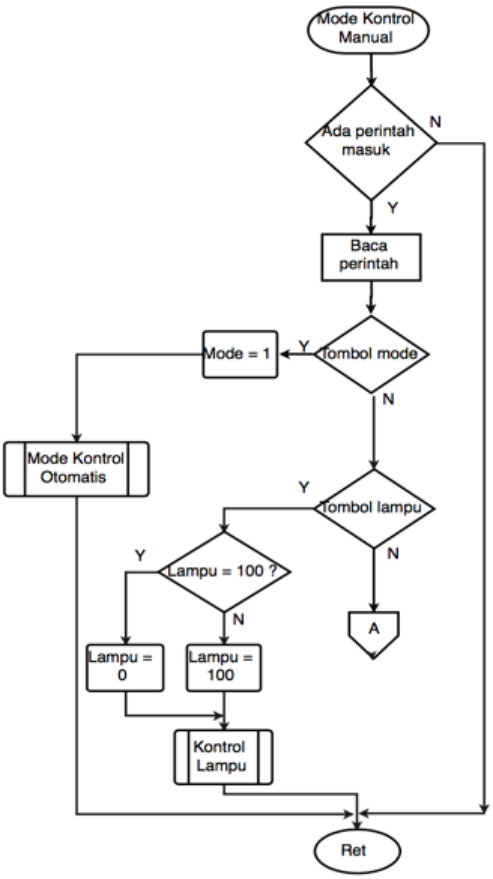
Apabila mode yang dipilih adalah otomatis, maka LCD akan menampilkan mode otomatis. Sebaliknya apabila mode yang dipilih adalah manual, maka LCD akan menampilkan mode manual. Setelah menentukan mode yang dipilih, Raspberry Pi akan mengambil nilai dari *indicator tirai* dan mengambil nilai dari *pwm* tirai dari *cloud ubidots*. Setelah mengambil nilai *pwm* tirai, Raspberry Pi akan menyimpan nilai pwm tirai yang diambil tadi pada sebuah file dengan format teks. Tujuan dari pengambilan nilai *indicator* tirai ini untuk mengetahui kondisi terakhir dari tirai apakah dalam keadaan terbuka atau tertutup. Sedangkan tujuan pengambilan nilai dari pwm tirai untuk mengetahui kondisi tirai terakhir berada di sudut buka berapa.

Setelah itu Raspberry Pi akan menentukan ke arah mana program akan berjalan, menuju ke kontrol manual atau ke kontrol otomatis. Hal tersebut ditentukan oleh pembacaan *switch* mode dari *dashboard ubidots*. Apabila *switch mode* bernilai 0, maka Raspberry Pi akan menjalankan program kontrol manual. Sebaliknya apabila bernilai 1, maka Raspberry Pi akan menjalankan program kontrol otomatis.

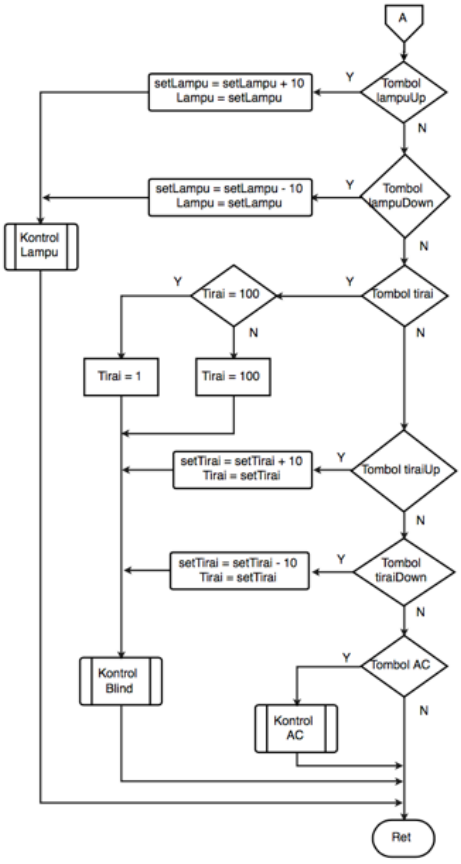


Gambar 6. Flowchart mode otomatis

Gambar 6 menunjukkan flowchart mode otomatis. Kontrol otomatis akan berjalan terus menerus sampai *switch mode* pada *ubidots* berubah menjadi kontrol manual. Untuk mengubah mode dapat dilakukan dengan 2 cara, yaitu dengan menekan *switch mode* di *dashboard ubidots* atau dengan menekan *switch mode* di kotak akrilik. Pada mode otomatis, peranan sensor menjadi sangat penting. Semua kendali dilakukan oleh sensor. Dapat dilihat pada *flowchart* mode kontrol otomatis, sensor PIR digunakan sebagai *trigger* untuk menyatakan apakah ada pergerakan orang atau tidak. Setelah pembacaan sensor PIR, sensor LDR digunakan sebagai *trigger* untuk mengetahui kondisi cahaya pada saat sistem berjalan. Apabila terdapat orang dalam ruangan dan kurang pencahayaan dari luar, maka lampu ruangan akan menyala dan tirai akan menutup. Untuk logika kendali lainnya dapat dilihat pada *flowchart* kontrol otomatis.



Gambar 7. Flowchart mode manual



Gambar 8. Flowchart mode manual (Sambungan)

Gambar 7 dan ganbar 8 menunjukkan *flowchart* mode manual. Pada mode manual, segala jenis pengaturan digantikan oleh peran pengguna. Pada mode ini, semua nilai variabel yang ada pada kontrol otomatis berpindah pada kontrol manual. Dengan demikian, nilai terakhir hasil proses kontrol otomatis akan tetap disimpan dan digunakan untuk proses kontrol manual. Apabila pada mode otomatis kondisi terakhir yang sedang berjalan adalah lampu LED menyala dan tirai menutup, maka lampu LED akan tetap menyala dan tirai akan tetap menutup ketika berada di mode manual.

III. PENGUJIAN SISTEM

Pada bagian ini terdapat 4 macam pengujian yaitu :

1. Pengujian intensitas pencahayaan ruang
2. Pengujian sensor PIR
3. Pengujian sensor suhu DHT11
4. Pengujian waktu pengambilan dan pengiriman data ke *ubidots*.

Berikut ini adalah hasil dari masing-masing pengujian.

A. Pengujian Intensitas Pencahayaan Ruang

Pengujian ini bertujuan untuk mengetahui seberapa besar lux yang dihasilkan ketika lampu menyala dan untuk mengetahui apakah teknik PWM yang dilakukan terhadap lampu berjalan. Pengujian dilakukan dengan prosedur sebagai berikut :

1. Variasi set pencahayaan lampu dilakukan mulai dari 0 hingga 100 %
2. Pengujian dilakukan pada pukul 13:00. Dengan *range* waktu setengah jam dari waktu pengujian dimulai. Pengujian yang dilakukan mencatat nilai lux, arus, dan daya yang dihasilkan.
3. Pengujian dilakukan pada 2 kondisi, yaitu saat tirai dalam keadaan tertutup dan terbuka.

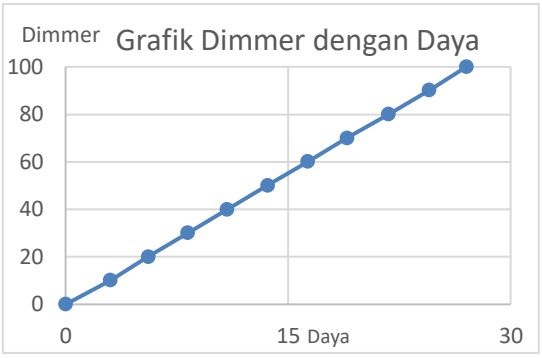
Berdasarkan hasil pengujian pada tabel 1 dan tabel 2, dapat dilihat bahwa nilai lux meningkat seiring peningkatan nilai *dimmer*. Selain itu jika melihat nilai arus dan daya yang dihasilkan, terjadi peningkatan seiring peningkatan nilai *dimmer*. Gambar 9 menunjukkan grafik antara *dimmer* dengan daya yang dihasilkan. Apabila melihat grafik yang dihasilkan, bentuk grafik adalah linier. Dapat dikatakan bahwa fungsi *dimmer* berjalan dengan baik.

Tabel 1. Pengujian Pencahayaan Tirai Terbuka

DIM MER	LUX 13:00	ARUS (Ampere)	DAYA (Watt)	Kondisi Tirai
0	21	0	0	Terbuka
10	32	0.251	3.012	
20	39	0.465	5.58	
30	46	0.686	8.232	
40	51	0.909	10.908	
50	55	1.135	13.62	
60	66	1.359	16.308	
70	76	1.58	18.96	
80	86	1.814	21.768	
90	91	2.042	24.504	
100	99	2.251	27.012	

Tabel 2. Pengujian Pencahayaan Tirai Tertutup

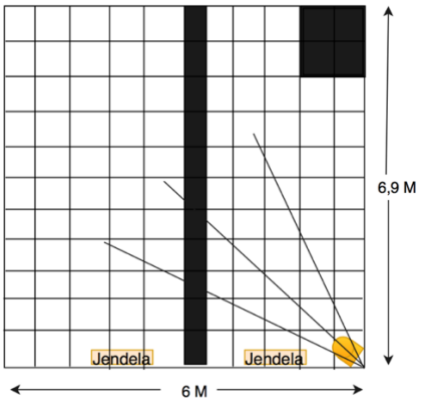
DIM MER	LUX 13:00	ARUS (Ampere)	DAYA (Watt)	Kondisi Tirai
0	11	0	0	Terbuka
10	22	0.251	3.012	
20	30	0.465	5.58	
30	39	0.686	8.232	
40	46	0.909	10.908	
50	49	1.135	13.62	
60	57	1.359	16.308	
70	59	1.58	18.96	
80	74	1.814	21.768	
90	84	2.042	24.504	
100	88	2.251	27.012	



Gambar 9.Grafik Dimmer dengan Daya

B. Pengujian Sensor PIR

Pengujian ini bertujuan untuk mengetahui daerah kerja sensor PIR dan daerah mana saja yang tidak termasuk dalam cakupan sensor PIR. Pengujian ini dilakukan dengan melihat *output* sensor PIR pada tampilan terminal di Raspberry Pi. Apabila terdapat pergerakan, maka akan muncul pada tampilan terminal di Raspbery Pi. Hasil pengujian sensor PIR dapat dilihat pada gambar 10. Daerah kerja yang dapat dibaca oleh sensor PIR ditandai dengan garis.



Gambar 10.Hasil pengujian sensor PIR

C. Pengujian DHT11

Pada pengujian ini, *output* suhu dari sensor suhu DHT11 diambil melalui tampilan terminal di Raspberry Pi dan dibandingkan dengan menggunakan termometer raksa. Percobaan ini dilakukan sebanyak 20 kali. Tujuan dari pengujian ini untuk mengetahui seberapa besar tingkat kesalahan dari pembacaan suhu menggunakan sensor DHT11. Hasil pengujian DHT11 dapat dilihat pada tabel 3.

Tabel 3. Hasil Pengujian DHT11

Data	Tanggal	Waktu	DHT 11 (ºC)	Termometer Raksa (ºC)	Selisih (ºC)
1	6 Juli 2017	13:57:47	27	25.5	1.5
2		14:12:47	27	25.5	1.5
3		14:27:47	27	25.5	1.5
4		14:42:47	27	25.5	1.5
5		14:57:47	27	25.5	1.5
6		15:12:47	27	25.5	1.5
7		15:27:47	27	25.5	1.5
8		15:42:47	27	25.5	1.5
9		15:57:47	27	25.5	1.5
10		16:12:47	27	25.5	1.5
11		16:27:47	27	25.5	1.5
12		16:42:47	27	25.5	1.5
13		16:57:47	27	25.5	1.5
14		17:12:47	27	25.5	1.5
15		17:27:47	27	25.5	1.5
16		17:42:47	27	25.5	1.5
17		17:57:47	27	25.5	1.5
18		18:12:47	27	25.5	1.5
19		18:27:47	27	25.5	1.5
20		18:42:47	27	25	2
Rata - Rata (ºC)					1.525

Berdasarkan data diatas, apabila nilai selisihnya di rata-rata, maka nilai selisihnya mejadi 1,525 °C. Kesimpulan dari pengujian ini adalah sensor DHT11 memiliki tingkat kesalahan kurang lebih sebesar 1,525 °C.

D. Pengujian Waktu Pengambilan Dan Pengiriman Data ke *ubidots*.

Pengujian ini dilakukan untuk mengetahui seberapa besar respon waktu pengiriman data dari Raspberry Pi ke *Cloud Ubidots* dan respon waktu pengambilan data dari *Cloud Ubidots* ke Raspberry Pi. Tabel 4 merupakan salah satu tabel pengambilan data dari *Cloud Ubidots* menuju Raspberry Pi. Data yang diambil adalah data ketika sistem pertama kali dijalankan dan sistem membaca mode yang sedang berjalan adalah manual. Teknik yang dilakukan adalah mencatat waktu ketika Raspberry Pi meminta data dari *Cloud Ubidots* hingga data sampai pada Raspberry Pi. Kedua nilai itu lalu diselisihkan dan kemudian di ambil rata-ratanya.

Tabel 4. Pengambilan Data dari *Cloud Ubidots* ke Raspberry Pi

Percobaan	Start - Manual		
	Waktu Ambil (s)	Waktu Selesai (s)	Selisih (s)
1	41.55	42.161	0.611
2	14.783	15.712	0.929
3	33.342	34.271	0.929
4	4.075	5.329	1.254
5	43.557	44.364	0.807
6	3.343	4.284	0.941
7	26.647	28.204	1.557
8	48.062	48.831	0.769
9	6.941	7.655	0.714
10	25.822	26.5	0.678
Rata-Rata (s)			0.8578

Tabel 5. Pengiriman Data dari Raspbery Pi ke *Cloud Ubidots*

Percobaan	Lampu Continue – Request Indikator Lampu		
	Waktu Ambil (s)	Waktu Selesai (s)	Selisih (s)
1	34	40.583	6.583
2	38	40.397	2.397
3	40	50.664	10.664
4	15	20.713	5.713
5	23	26.246	3.246
6	37	42.649	5.649
7	53	59.187	6.187
8	4	11.925	7.925
9	26	29.424	3.424
10	41	47.39	6.39
Rata-Rata (s)			5.1595

Pengujian yang dilakukan selanjutnya adalah mengirim data dari Raspberry Pi menuju *Cloud Ubidots*. Pengujian dapat dilihat pada tabel 5. Data yang dikirim adalah data penekanan tombol lampu, dimana *output* yang dihasilkan adalah indikator lampu. Teknik yang dilakukan untuk pengiriman data sedikit berbeda dengan pengambilan data. Teknik yang dilakukan adalah mencatat waktu ketika *Cloud Ubidots* sudah mengubah data hingga data sampai pada Raspberry Pi.

Berdasarkan hasil pengujian yang dilakukan, dapat dilihat bahwa waktu untuk mengirim data dari Raspberry menuju ke *Cloud Ubidots* lebih lama dibandingkan waktu pengambilan dari *Cloud Ubidots* menuju Raspberry Pi. Rata-rata waktu yang didapat untuk pengambilan dari dari *Cloud Ubidots* kurang dari 1 detik. Sedangkan waktu untuk pengiriman data dari Raspberry Pi menuju *Cloud Ubidots* membutuhkan waktu kurang lebih 5 detik.

IV. KESIMPULAN

Kesimpulan yang dapat diambil dari perancangan sistem dan pengujian pada pembuatan sistem *smart home* ini adalah :

- Pembacaan sensor suhu DHT11 memiliki tingkat kesalahan sebesar 1,525 °C.
- Area cakupan sensor PIR ini belum menjangkau seluruh bagian ruangan.
- *Dimmer* lampu yang diinginkan berhasil dengan baik.
- Pengambilan data dari *Cloud Ubidots* membutuhkan waktu kurang dari 1 detik.
- Proses pengiriman data menuju *Cloud Ubidots* membutuhkan waktu lebih lama dibandingkan proses pengambilan data. Waktu yang dibutuhkan kurang lebih 5 detik.

DAFTAR PUSTAKA

[1] Wikipedia, “Home Automation,” 2016. [Online]. Available: https://en.wikipedia. org/wiki/ Home_automation. [Accessed: 11-Okt-2016].

[2] Dey. K, Zimmerman. J, Yiu. C, Lee. K, and Davidoff. S, “*Principles of Smart Home Control*, ” Ubicomp 2016. LNCS, vol. 4206, pp. 19-34, 2016.

[3] Ubidots, “Ubidots, ” 2016. [Online]. Available: https://ubidots.com/. [Accessed: 5-Okt-2016].

[4] Infineon, “IRLZ44NPBF,” 2017. [Online]. Available: https://www.infineon.com/dgd l/irfz44npbf.pdf?fileId=5546d462533600a40153563b 3a9f220d. [Accessed: 10-Jun-2017].

[5] Texas Instrument, “ULN2003,” 2017. [Online]. Available: www.ti.com/lit/ds/sym link/ulq2003a.pdf. [Accessed: 1-Jun-2017].