

## Implementasi Arsitektur *Publish And Subscribe* Pada Alat Monitoring Suhu Dan Kelembaban Kandang Ular Python Regius Menggunakan NodeMCU (ESP8266)

Habib Zainal Sarif<sup>1</sup>, Mochammad Hannats Hanafi Ichsan<sup>2</sup>, Rizal Maulana<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>habibzarif2006@gmail.com, <sup>2</sup>hanas.hanafi@ub.ac.id, <sup>3</sup>rizal\_lana@ub.ac.id

### Abstrak

Kondisi iklim di kota Malang yang cukup dingin dan sedikit lembab membuat beberapa hewan peliharaan yang kita miliki menjadi rentan terhadap serangan penyakit, terutama pada hewan reptil khususnya ular piton regius atau yang biasa dikenal dengan sebutan piton bola. Selain dipengaruhi hormon pada ular piton dewasa, menurunnya nafsu makan ular juga dapat dipengaruhi oleh suhu dan kelembaban disekitar kandang ular tersebut. Untuk suhu dan kelembaban yang ideal untuk ular ini pada siang hari berkisar antara 27-29°C dan untuk malam harinya suhu idealnya adalah 23-24°C. Sedangkan untuk kelembaban yang baik untuk ular ini berkisar antara 50-60%. Walaupun sudah banyak dipasaran alat untuk mengukur suhu dan kelembaban suatu ruangan akan tetapi peternak masih membutuhkan suatu alat yang mampu dimobilisasikan sehingga mampu mengawasi keadaan kandang walaupun tidak berada di kandang. Berdasarkan masalah tersebut dibuatkan sistem monitoring kandang ular menggunakan arsitektur *Publish and Subscribe*. Dimana sistem ini didukung dengan sebuah protokol komunikasi *MQTT* dan akan ditampilkan pada sebuah aplikasi *Android*. Pada penelitian ini menggunakan 3 parameter yaitu suhu dan kelembaban dari pembacaan sensor *DHT11* dan waktu secara berkala dari *RTC DS3231*, sebagai hasil keluaran berupa tampilan pada LCD dan juga pada aplikasi *Android*. Untuk arsitektur *publish and subscribe* sendiri dipilih karena memiliki kecepatan dan penggunaan daya yang lebih rendah. Sehingga akan menciptakan suatu lingkungan yang baik untuk kehidupan ular itu sendiri.

**Kata kunci:** Reptil, Piton Regius, *DHT11*, *Publish and Subscribe*, *NodeMCU*, *MQTT* Suhu, Kelembaban

### Abstract

*The climatic conditions in the city of Malang, a pretty cool and slightly damp to make some of the pets we have become vulnerable to attack diseases in animals, especially reptiles, especially snakes Pythons regius or commonly known as ball python. In addition to the affected adult Pythons snakes on hormones, decreased appetite the snake can also be influenced by the surrounding temperature and humidity snake cages. For temperature and humidity are ideal for snakes at noon ranged between 27-29 ° C and for the evening temperature ideally is 23-24 ° C. As for the moisture is good for these snakes ranged from 50-60%. Although there have been many successful tool for measuring temperature and humidity of a room but the rancher still needs a tool capable to move so that it is able to keep an eye on the State of the cages despite not being on the enclosure. Based on the issue be made a snake enclosure monitoring system uses a Publish and Subscribe architecture. Where this system is supported by a communication protocol MQTT and will be shown on a Android applications. This research uses 3 parameters of readout of temperature and humidity from sensors DHT11 and time of DS3231 RTC periodically, as the output form of display on the LCD and also on Android applications. For publish and subscribe architecture or send it was chosen because it has the speed and lower power consumption. So that will create an environment that is good for the life of the snake itself.*

**Keywords:** Reptiles, Pythons Regius, *DHT11*, *Publish and Subscribe*, *NodeMCU*, *MQTT*, Temperature, Humidity

## 1. PENDAHULUAN

Perkembangan teknologi yang semakin pesat tidak hanya berpengaruh terhadap lingkungan yang berhubungan dengan teknologi informasi saja. Melainkan sudah mengubah gaya hidup dan hobi kebanyakan orang seperti pecinta reptil salah satunya. Dengan semakin pesatnya perkembangan teknologi memungkinkan seseorang mampu dengan mudah memiliki hewan peliharaannya seperti ular piton regius.

Ular yang memiliki nama latin *Regius Python* atau kebanyakan orang menyebutnya dengan *Ball Python* ini merupakan sebuah ular piton yang tidak memiliki tubuh besar seperti ular-ular pinton yang umum kita jumpai. Ular ini umumnya memiliki ukuran yang tidak mencapai 1,5 meter untuk ukuran indukan. Dengan ukuran tubuh yang relatif pendek, ular ini juga memiliki sifat yang cenderung jinak dan pemalu. Hal ini terlihat ketika ular ini merasa terancam maka ular ini akan membentuk posisi seperti sebuah bola untuk melindungi kepalanya daripada melakukan serangan pada ancaman itu sendiri. Peminat ular ini juga tidak lepas dari corak warna tubuhnya yang sangat beragam dan menarik. Menurut web *World of Ball Python* terdapat 6251 morps atau corak warna yang ada. Namun terdapat beberapa perhatian khusus untuk memelihara ular endemik asli benua Afrika ini, terutama suhu dan kelembaban kandang ular tersebut. Zona nyaman ular ini berkisar antara 27 - 29 derajat celcius pada siang hari, dengan tempat berjemur sekitar 32 derajat celcius. Sedangkan suhu waktu malam bisa turun menjadi sekitar 23 -24 derajat celcius (MCLEOD, 2017). Untuk kelembaban *Regius Python*s membutuhkan tingkat kelembaban sekitar 50-60%, jadi disarankan untuk membeli Hydrometer untuk memantau ini (Python, 2017).

Berdasarkan permasalahan diatas penulis ingin membuat sebuah sistem yang mampu mengawasi keadaan dari kandang ular itu sendiri. Yaitu sebuah sistem monitoring yang berbasis teknologi terbaru yakni *IoT(Internet of Thing)*. Salah satu arsitektur yang dapat dikembangkan pada teknologi *IoT* itu sendiri adalah arsitektur *Publish and Subscribe*. Arsitektur ini memiliki beberapa keunggulan diantaranya *space decoupling*, *time decoupling*, *synchronization decoupling* dan juga skalabilitas

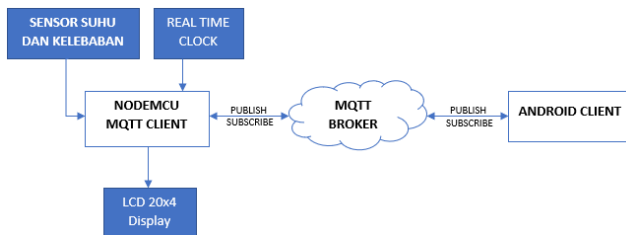
Pada penelitian kali ini penulis ingin membuat sebuah penelitian *IoT* yang menggunakan arsitektur *publish and subscribe* sebagai

algoritma utama yang mendukung sistem monitoring kandang ular tersebut. *Internet of Things (IoT)* itu sendiri merupakan suatu teknologi terbaru yang membuat suatu objek dapat berkomunikasi dengan objek lainnya yang saling terhubung melalui sebuah jaringan internet. Teknologi ini memberikan sebuah gambaran besar tentang sumberdaya suatu objek mampu saling bertukar informasi dan bahkan mampu mendengar, melihat dan tentunya berpikir untuk mengambil sebuah keputusan yang mampu merubah gaya hidup masyarakat di era perkembangan teknologi yang semakin pesat dengan mamaksimalkan teknologi embeded sistem dengan sensor terhadap protokol *cdrras* lainnya sehingga benda yang tadinya konvensional akan menjadi sebuah benda yang lebih modern dan cerdas (Hussain, 2016).

Adapun protokol komunikasi yang digunakan adalah MQTT. Protokol dalam pengiriman pesan yang dapat mendukung metode dari *publish and subscribe* ialah MQTT dan XMPP. Protokol MQTT memiliki karakteristik sederhana, ringan dan mudah untuk diimplementasikan (Taringan, 2014). Dan nantinya diharapkan bahwa penelitian ini bertujuan untuk menstabilkan sebuah kandang hewan dengan mengukur suhu dan kelembabannya yang kemudian akan mengurangi tingkat kematian pada hewan itu sendiri. Serta bertujuan untuk mengukur kinerja dari arsitektur yang sama dengan yang penulis buat dengan menggunakan protokol MQTT dimana nantinya akan diketahui performa dari arsitektur itu sendiri.

## 2. PERANCANGAN ALUR KERJA SISTEM

Perancangan alur kerja sistem ini merupakan penjelasan secara menyeluruh tahap-tahap yang akan dilakukan pada sistem ini mulai dari awal sistem dijalankan hingga sistem menghasilkan *output* yang diharapkan oleh peneliti. Selain itu pada bagian ini penulis akan menjelaskan gambaran secara umum daripada sistem monitoring kandang ular piton *Regius* dengan menggunakan sebuah arsitektur jaringan terkini yaitu arsitektur *publish and subscribe*. Seperti dijelaskan pada box diagram berikut ini pada Gambar 1.



Gambar 1 Diagram Blok dari Sistem Monitoring

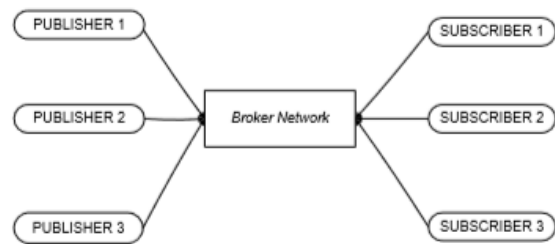
### 2.1 Publish and Subscribe

Arsitektur ini merupakan sebuah proses komunikasi yang terjadi secara tidak langsung. Karena untuk menjalin sebuah komunikasi pada arsitektur ini dibutuhkan pihak ketiga yang akan menjembatani antara penyedia informasi dan user. Adapun pada arsitektur ini terdapat 3 bagian utama yang akan saling menjalin sebuah komunikasi yakni *publisher/server* sebagai penyedia informasi, *subscriber/client* yang membutuhkan informasi, serta *broker* yang akan menjembatani proses pertukaran data antara *publisher* dan *subscriber*.

Pertama-tama *subscriber* dan *publisher* akan membangun sebuah komunikasi berdasarkan topik yang diinginkan. Dalam hal ini *publisher* akan menyediakan topik yang berisikan data dan informasi. Dan topik tersebut yang akan diminati oleh *subscriber*. Topik itu sendiri merupakan sebuah identitas dari sebuah data dan informasi yang telah diterbitkan. Pada proses ini *publisher* tidak perlu tahu user peminat yang aktif karena data akan dikirim secara terus menerus, sedangkan *subscriber* hanya akan mendapatkan informasi berdasarkan topik yang diminatinya saja. Mekanisme yang ditawarkan pada arsitektur ini untuk sektor teknologi IoT memiliki banyak keuntungannya dibanding arsitektur sistem terdistribusi lainnya yakni informasi dapat dikirimkan secara efisien karena *subscriber* hanya menerima pesan sesuai dengan topik yang diminatinya (Kasyful, 2016).

Penggunaan arsitektur ini pada sistem monitoring ini akan didukung oleh protokol komunikasi MQTT dan sebuah format teks JSON. Karena pada sistem ini yang digunakan adalah format teks serta protokol komunikasi yang cukup ringan, maka akan diharapkan sistem dapat berjalan dengan lebih efisien dan maksimal. Karena pada sistem ini data akan dikirim secara terus menerus selama user masih berlangganan pada topik tersebut. Selain itu arsitektur ini sangat cocok diterapkan pada sebuah perangkat *IoT* karena memungkinkan user dapat mendapatkan lebih dari satu data

pada satu topik tertentu tersebut Pada Gambar 2 berikut adalah gambaran sederhana dari sebuah arsitektur *publish and subscribe*.



Gambar 2 Arsitektur Publish and Subscribe

### 2.2 MQTT

Pada tahun 1990 Alen Nipper dari Eurotech melakukan sebuah kolaborasi dengan Andy Stanford-Clark dari IBM untuk menciptakan suatu protokol komunikasi baru yakni *MQTT*. Yang merupakan sebuah protokol yang mendukung kinerja dari sebuah konektivitas *machine to machine (M2M)* dan juga sebuah perangkat terbaru yang sering disebut yaitu *Internet of Things*. *MQTT* ini sendiri juga sering digunakan sebagai sebuah broker pada arsitektur *Publish and Subscribe* karena protokol ini sangatlah ringan dan mudah untuk diterapkan. Hal ini menyebabkan protokol *MQTT* ini sangat idel untuk diimplementasikan pada suatu lingkungan jaringan yang terbatas dan juga pada suatu sistem *embeded* yang memiliki prosesor dan memori yang terbatas. Adapun *MQTT* memiliki 14 jenis pesan yang berbeda seperti pada disebutkan pada Tabel 1.

Adapun cara kerja dari prokol *MQTT* adalah sebagai berikut:

1. *Connection Setup*

Merupakan bagian awal yang dilakukan oleh *publisher*, *broker*, dan juga *subscriber* untuk saling terhubung dan membangun sebuah jaringan komunikasi.

2. *Disconnecting*

Proses ini terjadi pada saat user pengguna berhenti berlangganan pada sustu topik yang telah dipublikasikan oleh *server (unsubscribe)*.

3. *Heart Beat*

Adalah sebuah tahap antara *publisher*, *broker*, dan *subscriber* pada saat *disconnecting*.

4. *Subscribing*

Merupakan sebuah proses dimana user pengguna akan berlangganan pada beberapa topik yang diminati dari server yang tersedia.

5. Publishing

Adalah sebuah tahap dimana server akan mengirimkan data sesuai topik yang diminati oleh user pengguna.

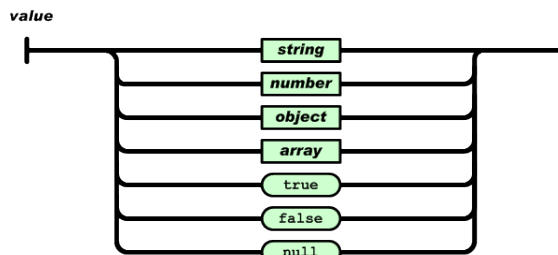
Tabel 1 Jenis Pesan pada MQTT

Mnemonic	No.	Description
CONNECT	1	Klien request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received-assured delivery part 1
PUBREL	6	Publish Release-assured delivery part 2
PUBCOMP	7	Publish Complete-assured delivery part 3
SUBSCRIBE	8	Klien Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Klien Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Klien is Disconnecting

2.3 JSON

JSON merupakan format teks yang dapat memfasilitasi pertukaran antar data yang terstruktur diantara semua bahasa pemrograman (Ecma, 2013). JSON sendiri adalah sebuah format teks penulisan kode program yang memiliki ukuran data yang lumayan ringan, oleh karena format teks ini sangat mudah untuk dimengerti dan dipahami oleh manusia serta ditulis dan dibaca oleh sistem komputer. Format teks ini juga mampu diterapkan pada banyak bahasa pemrograman. Karena JSON juga merupakan sebuah format teks yang tidak tergantung pada sebuah bahasa pemrograman tertentu. Adapun bahasa yang dapat didukung cukup lengkap yakni meliputi Javascript, Java, C++, C, Python dan lain sebagainya. Sehingga hal ini membuat JSON yang sangat baik untuk digunakan untuk pertukaran data.

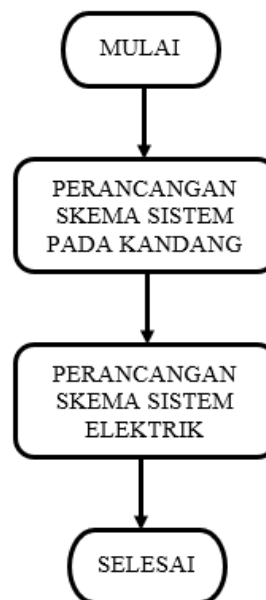
Dalam format teks JSON ini terdapat beberapa bentuk pada format teks yakni Object dan JSON Array. Dimana kedua bentuk format teks ini dapat disusun secara bertingkat. Seperti diilustrasikan pada Gambar 3 berikut.



Gambar 3 Nilai diagram sintak pada JSON

2.4 Perancangan Perangkat Keras

Pada tahapan ini akan dijelaskan tentang proses perancangan perangkat keras dimana akan dibagi dalam bagian yakni perancangan skema sistem monitoring pada lingkungan kandang dan perancangan rangkaian elektronik. Seperti dijelaskan pada diagram alir pada Gambar 4.

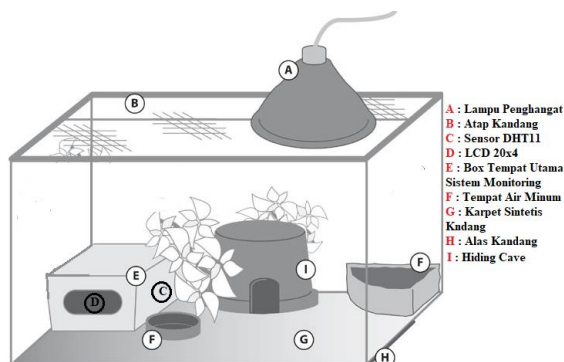


Gambar 4 Diagram Alir Perancangan Perangkat Keras

2.5 Perancangan Skema Lingkungan

Pada perancangan ini sistem akan ditempatkan pada suatu kandang ular yang memiliki panjang 50 cm, lebar 25 cm dan tinggi 28 cm. Kandang ini biasa diperuntukan untuk 1 ekor ular saja berukuran Juvenile atau remaja. Perancangan sistem akan dapat dilihat pada Gambar 5.



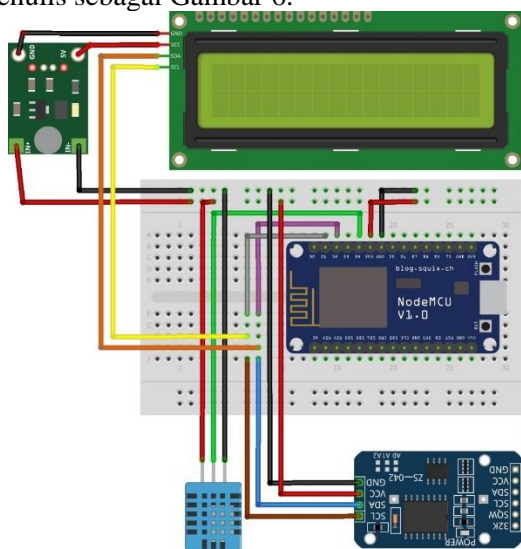


Gambar 5 Peletakan Alat Monitoring

Pada gambar diatas terdapat beberapa perlengkapan pendukung untuk merawat sebuah ular dari mulai lampu penghangat, tempat minum hingga *hiding cave* atau tempat sembunyi ular. Untuk utama sistem diletakkan pada bagian bawah belakang kandang. Sedangkan untuk sensor DHT11 berada di bagian samping dari box yang terletak dalam kandang sehingga dapat membaca data suhu dan kelembaban di dalam kandang ular tersebut.

### 2.6 Perancangan Elektrik Sistem

Pada tahap kali ini akan dijelaskan tentang perancangan pada suatu rangkaian elektronik perangkat keras yang mendukung kinerja dari sistem monitoring itu sendiri dimana nantinya rangkaian elektrik ini diharapkan mampu bekerja sesuai hasil yang ingin dicapai oleh penulis sebagai Gambar 6.



Gambar 6 Perancangan Rangkaian Elektrik

## 3. PENGUJIAN DAN ANALISIS

### 3.1 Pengujian Sensor DHT11

Sensor DHT11 merupakan sensor utama yang digunakan pada sistem monitoring ini

untuk dapat memberikan data suhu dan kelembaban. Pada penelitian kali ini sensor DHT11 digunakan oleh peneliti untuk melakukan pembacaan pada suhu dan kelembaban pada kandang ular. Berikut ini adalah nilai presentase kesalahan dari hasil pembacaan sensor *DHT11* yang dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2 Presentase Kesalahan Pembacaan Suhu

NO	Hasil pembacaan sensor <i>DHT11</i> dengan <i>HTC-1</i>		
	<i>DHT11</i> (°C)	<i>HTC-1</i> (°C)	Presentase Kesalahan (%)
1	28	28,6	2,06 %
2	28	29,4	1,33 %
3	28	28	0 %
4	25	26,2	0,76 %
5	23	23,4	1,73 %
6	24	25,5	6,25 %
7	34	35,8	2,28 %
8	35	37,5	1,35 %
9	31	31,5	1,61 %
10	32	34,7	2,05 %
Rata-Rata			1,94 %

Pada Tabel 2 diatas adalah hasil dari proses pengujian yang dilakukan dengan untuk mengukur presentase kesalahan pada pengujian suhu dengan membandingkan antara sensor *DHT11* dengan *HTC-1* untuk mengukur presentase kesalahan pada pengujian suhu dimana rata-rata kesalahan dari pengujian adalah 1,94 %.

Tabel 3 Presentase Kesalahan Pembacaan Kelembaban

NO	Hasil pembacaan sensor <i>DHT11</i> dengan <i>HTC-1</i>		
	<i>DHT11</i> (%)	<i>HTC-1</i> (%)	Presentase Kesalahan (%)
1	40	43	7,5 %
2	50	53	6 %
3	65	67	3,07 %
4	39	40	2,56 %
5	37	40	8,10 %
6	60	62	2,5 %
7	63	65	2,40 %

8	50	52	4 %
9	60	64	6,66 %
10	69	70	1,44 %
Rata-Rata			4,42 %

Pada Tabel 3 diatas adalah hasil dari proses pengujian yang dilakukan untuk menilai dan mengukur tingkat presentase kesalahan pada pengujian suhu dengan cara membandingkan antara sensor DHT11 dengan HTC-1. Dari hasil pengujian didapatkan rata-rata kesalahan dari pengujian adalah 4,42 %.

### 3.2 Pengujian LCD 20x4

Pada pengujian kali ini ditujukan untuk menguji hasil output daripada LCD 20x4. Dimana LCD ini akan membantu mempermudah proses monitoring sistem.



Gambar 7 Tampilan LCD awal



Gambar 8 Tampilan LCD ketika sistem berjalan

Berdasarkan hasil dari pada pengujian yang telah dilakukan seperti pada Gambar 7 dan 8 tampilan antara LCD dengan serial monitor adalah sama, sehingga mampu disimpulkan bahwa LCD mampu berjalan dengan baik sesuai dengan yang harapan dari perneliti.

### 3.3 Pengujian RTC DS3231

Pada pengujian RTC DS3231 ini bertujuan untuk mengetahui aapakah perangkat keras dari RTC DS3231 ini dapat bekerja dengan baik atau tidak. Karena RTC ini adalah komponen yang ditampilkan untuk mengetahui waktu secara real

time pada sistem. Tabel 4 nilai hasil pengujian RTC.

Tabel 4 Hasil Pengujian RTC DS3231

NO	Waktu
1	08:07:53
2	08:08:03
3	08:08:14
4	08:08:24
5	08:08:35
6	08:08:45
7	08:08:55
8	08:09:06
9	08:09:16
10	08:09:27

Tabel diatas didapat dari pengujian dengan mengirimkan data waktu pada sistem sebanyak 10 kali dan dengan jeda pengiriman sebesar 10 detik. Dengan demikian RTC DS3231 itu sendiri sudah mampu keluar dengan baik dan benar pada serial monitor maupun pada LCD yang terpasang pada perangkat keras sistem.

### 3.4 Pengujian Publish and Subscribe

Dilakukannya pengujian ini adalah untuk mengetahui apakah arsitektur publish and subscribe ini dapat berjalan dengan baik dan benar pada sistem. Sehingga data dapat sampai pada pengguna. Untuk melakukan pengujian terhadap arsitektur publish and subscribe itu sendiri akan dilakukan dengan cara mengamati data yang di dipublish oleh perangkat keras maupun data ya diterima oleh user dengan cara melakukan subscribe pada topik yang telah ditentukan. Pada pengujian ini masing-masing data akan dikirim dengan jeda waktu sebesar 10 detik. Dan akan diambil 10 data sample acak untuk membuat hasil dan analisa dari kegiatan pengujian ini sendiri. Berikut ini hasil pengujian yang dituliskan pada Tabel 5 dan Tabel 6.

Tabel 5 Hasil Publish Data

NO	DATA DITERIMA		WAKTU
	Suhu	Kelembaban	Dikirim
1	YA	YA	08:07:53
2	YA	YA	08:08:03
3	YA	YA	08:08:14
4	YA	YA	08:08:24
5	YA	YA	08:08:35
6	YA	YA	08:08:45
7	YA	YA	08:08:55
8	YA	YA	08:09:06

9	YA	YA	08:09:16
10	YA	YA	08:09:27

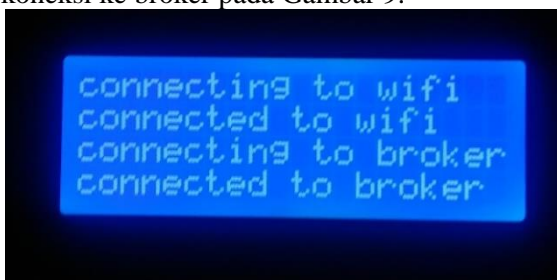
Tabel 6 Hasil *Subscribe* Data

NO	DATA DITERIMA		Waktu
	Suhu	Kelembaban	Diterima
1	YA	YA	08:07:59
2	YA	YA	08:08:09
3	YA	YA	08:08:19
4	YA	YA	08:08:30
5	YA	YA	08:08:40
6	YA	YA	08:08:51
7	YA	YA	08:09:01
8	YA	YA	08:09:11
9	YA	YA	08:09:22
10	YA	YA	08:09:32

Berdasarkan hasil pengujian dan pengamatan yang dilakukan pada sistem monitoring ini dengan melakukan 10 kali pengiriman data, nilai dari data suhu, kelembaban dan waktu mampu dikirim dan diterima dengan baik pada arsitektur *publish and subscribe* oleh user pengguna sistem ini.

### 3.5 Pengujian MQTT

Pengujian ini dilakukan untuk mengetahui apakah sistem mampu melakukan komunikasi dengan baik dengan broker. Sehingga data dapat *publish* dengan baik dan benar. Berikut proses koneksi ke broker pada Gambar 9.



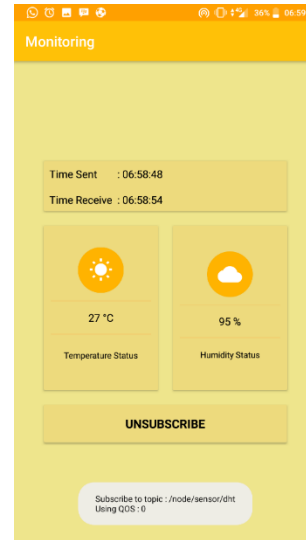
Gambar 9 Tampilan LCD ketika proses koneksi dengan broker

Pada gambar diatas terlihat pada tampilan LCD pada saat sistem memulai proses koneksi pada broker hingga terhubung pada broker. Sehingga nantinya data yang didapat dari sensor maupun *RTC* akan *publish* pada broker menunggu data akan *unsubscribe* oleh pengguna.

### 3.6 Pengujian Aplikasi Android

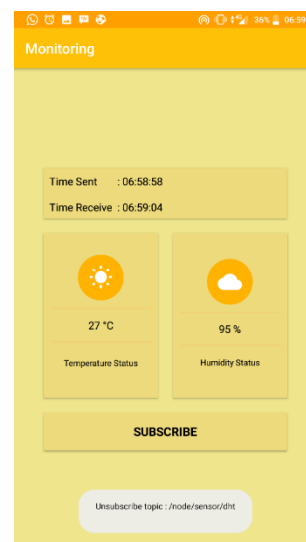
Pengujian aplikasi android ini bertujuan untuk mengetahui apakah aplikasi mampu berjalan dengan baik dan benar. Mulai data

temperatur, kelembaban, dan waktu. Selain itu apakah mampu *subscribe* dengan baik dan benar. Sehingga dapat diketahui perbedaan waktu pengiriman data dari alat monitoring hingga waktu diterima pada aplikasi android. Gambar 10 dan 11 adalah hasil tampilan aplikasi android.



Gambar 10 Tampilan Aplikasi *Android* saat melakukan *subscribe*

Berdasarkan 2 gambar ini aplikasi android dapat berjalan dengan baik dan benar dari mulai proses awal menunggu, kemudian melakukan *subscribe* pada topik tertentu, hingga proses *unsubscribe* pada topik tertentu.



Gambar 11 Tampilan Aplikasi *Android* saat melakukan *unsubscribe*

## 4. Kesimpulan

Pada penelitian ini telah dibuat sistem alat monitoring suhu dan kelembaban kandang ular python regius menggunakan nodemcu sebagai

pendeteksi suhu dan kelembaban yang berhasil diolah oleh mikrokontroler NodeMCU, Serta memiliki memiliki presentase kesalahan sebesar 1,94% dan keakurasian sebesar 98,06% pada pembacaan suhu. Untuk pembacaan kelembaban memiliki presentase kesalahan sebesar 4,42 % dan keakurasian sebesar 95,58%, dibandingkan nilai suhu dan kelembaban yang dideteksi dengan HTC-1.

Pada penelitian ini menggunakan arsitektur *publish and subscribe* yang mampu dijalankan oleh sistem dengan baik dan benar. Mulai dari proses *publish* data dari sistem perangkat keras hingga proses *subscribe* data dengan aplikasi android. Dengan demikian disimpulkan bahwa data mampu diterima dengan baik dan sesuai oleh user pengguna.

Berdasarkan pengujian yang dilakukan pada salah satu peternak ular regius piton di kota malang, dapat disimpulkan bahwa sistem monitoring ini berhasil membantu pemilik peternakan untuk mengawasi kandang ularnya dengan mudah. Karena sebelum menggunakan sistem ini pemilik harus melakukan pengecekan secara berkala untuk mengetahui kondisi suhu dan kelembaban pada kandang ular. Sedangkan sekarang pemilik kandang hanya perlu melakukan pengecekan kandang. Sehingga mampu meningkatkan kualitas kandang itu sendiri.

Adapun saran untuk perkembangan penelitian ini adalah dengan melakukan implementasi sistem otomasi pada sistem monitoring ini sehingga semakin mempermudah user pengguna dan penelitian ini tidak cuma berhenti ketika penelitian ini berakhir.

## DAFTAR PUSTAKA

- Amron, K., Sakti, E., Data, M., 2016. Pemodelan dan Analisis *Wireless Mesh Network* dengan Arsitektur *Publish-Subscribe* dan Protokol MQTT. Malang
- ECMA International., 2013. *The JSON Data Interchange Format* . Jenewa
- Hussain, S., Yaqoob, I., 2016. *A Survey: Internet of Things (IOT) Technologies, Applications and Challenges*. Pakistan
- Intel, 2015. *A Guide to the Internet of Things Infographic*. [online] Tersedia di: <<http://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html>> [Diakses 11 Januari 2018].
- Kasim, A. A. (2018). Implementasi Otomasi Kandang dalam Rangka Meminimalisir *Heat Stress* pada Ayam Broiler dengan metode *Fuzzy Sugeno*. Malang: JPTIIK UNIVERSITAS BRAWIJAYA.
- MCLEOD, L. (2017). Do Ball Pythons Make Good Pets? . *Do Ball Pythons Make Good Pets?* , 5.
- Python, T. R. (2017, 1 26). *Royal Python Care Sheet*. Diambil kembali dari The Royal Python.co.uk:  
[http://www.theroyalpython.co.uk/royal\\_python\\_care\\_sheet.htm](http://www.theroyalpython.co.uk/royal_python_care_sheet.htm)
- Taringan, S. O. F. (2014). Pengukuran Kinerja Sistem Publish/Subscribe Menggunakan Protokol *MQTT (Message Queuing Telemetry Transport)*. Bandung