

**SKRIPSI**

**SISTEM PENGATUR SUHU INKUBATOR OTOMATIS**



**Enrico Findley**

**NPM: 2013730008**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2020**

**UNDERGRADUATE THESIS**

**AUTOMATIC INCUBATOR TEMPERATURE CONTROL  
SYSTEM**



**Enrico Findley**

**NPM: 2013730008**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2020**

# LEMBAR PENGESAHAN

## SISTEM PENGATUR SUHU INKUBATOR OTOMATIS

Enrico Findley

NPM: 2013730008

Bandung, 3 Januari 2020

Menyetujui,

Pembimbing

Chandra Wijaya, M.T.

Ketua Tim Penguji

Anggota Tim Penguji

Elisati Hulu, M.T.

Vania Natali, M.T.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

## **PERNYATAAN**

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### **SISTEM PENGATUR SUHU INKUBATOR OTOMATIS**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal 3 Januari 2020

Meterai Rp. 6000
---------------------

Enrico Findley  
NPM: 2013730008

## ABSTRAK

Metode yang digunakan untuk menetas telur masih menggunakan metode manual yaitu dengan metode pengeraman telur menggunakan induk ayam atau dengan menggunakan lampu dan bak air sebagai pengatur suhu dan kelembapan, metode tersebut membutuhkan tenaga dan waktu manusia dalam prosesnya. Metode tersebut belum cukup optimal untuk menetas telur dalam jumlah yang sangat banyak, karena biaya dan tenaga yang dikeluarkan lumayan banyak. Teknologi *Internet of Things (IoT)* memungkinkan suhu dapat dideteksi dengan sensor dan dapat dilihat oleh pengguna lalu suhu dalam inkubator dapat diatur dengan menggunakan bantuan *microcontroller* dan *relay*. Untuk itu, diperlukan sebuah sistem yang dapat mengatur suhu dalam inkubator agar telur dapat menetas secara otomatis sehingga efisiensi waktu dan biaya dapat ditingkatkan.

Untuk menjawab masalah tersebut, maka dibuat sistem penetas telur otomatis berbasis Arduino. Sistem ini menggunakan perangkat Arduino sebagai perangkat utamanya. Perangkat Arduino dibagi menjadi dua, yaitu Arduino *master* dan Arduino *slave* lalu akan dibuat tampilan berbasis *web*. Arduino *master* berfungsi sebagai pengirim dan penerima data dari basis data dan dari *slave*, sedangkan *slave* berfungsi untuk mendeteksi suhu ruangan dengan menggunakan sensor suhu DHT11, mengirimkan data tersebut ke *master*, lalu *slave* akan mematikan dan menyalakan kipas atau lampu dengan bantuan *relay* agar batas suhu dapat diatur dengan baik. Dengan menggunakan koneksi *Wi-Fi* dan protokol *Message Queueing Telemetry Transport (MQTT)*, pengaturan batas suhu dan pemantauan suhu ruangan oleh perangkat-perangkat tersebut dapat dilakukan dari jarak jauh.

Untuk menguji sistem yang dibangun, dilakukan analisa terhadap fungsi-fungsi dan tingkat ketahanan sistem jika terjadi kendala, lalu dilakukan pengujian fungsional untuk menguji fitur-fitur yang ada pada sistem dan pengujian eksperimental untuk menguji kehandalan sistem. Dari hasil pengujian dalam penelitian ini, sistem yang dibangun telah dapat menjalankan fungsi-fungsi yang dibutuhkan dengan baik.

Berdasarkan hasil pengujian maka dapat diambil kesimpulan bahwa sistem penetas telur otomatis berbasis Arduino telah berhasil dibangun. Kesimpulan yang didapat adalah suhu dalam inkubator dapat terjaga, batas suhu dapat diatur sesuai keinginan pengguna, kipas atau lampu dapat dimatikan secara otomatis oleh Arduino, protokol *MQTT* sudah berjalan dengan baik, dan data suhu dapat ditampilkan pada *web browser*.

**Kata-kata kunci:** pendeteksian suhu, *MQTT*, Arduino *master*, Arduino *slave*, *Wi-Fi*, *web*

## ABSTRACT

The method used to hatch eggs is still using the manual method, namely the egg incubation method using the hen or using lights and water tanks as regulators of temperature and humidity, this method requires human energy and time in the process. The method is not considered optimal enough for hatching egg in large amount, because this method needed a significant amount of time and energy. The Internet of Things (IoT) technology allows temperatures to be detected by sensors and can be seen by users and the temperature in the incubator can be adjusted using the help of microcontroller and relay. Therefore, we need a system that can adjust the temperature in the incubator so that eggs can hatch automatically so that time and cost efficiency can be increased.

To solve this problem, an Arduino-based automatic egg hatcher system was created. This system uses the Arduino device as its main device. Arduino devices are divided into two, namely master Arduino, slave Arduino, and a web-based user interface will be created. Master Arduino used as the sender and receiver of data from the database and from slave, while slave used to detect room temperature using DHT11 temperature sensor, send the data to master, then slave will turn off and turn on the fan or lamp with the help of relay so that the temperature limit can be managed properly. By using the Wi-Fi connection and the Message Queuing Telemetry Transport (MQTT) protocol, temperature control and room temperature monitoring by these devices can be done remotely.

To test the system being built, an analysis of the functions and the level of system resilience in the event of a problem is performed, then a functional test is performed to test the features of the system and an experimental test to test the system's reliability. From the test results in this study, the system built has been able to perform the functions needed properly.

Based on the test results, it can be concluded that the Arduino-based automatic egg hatcher system has been successfully built. The conclusion is that the temperature in the incubator can be maintained, the temperature limit can be adjusted according to the user's wishes, the fan or the lamp can be turned off automatically by Arduino, the MQTT protocol is running well, and the temperature data can be displayed on web browser.

**Keywords:** temperature detection, MQTT, master Arduino, slave Arduino, Wi-Fi, web

*Dipersembahkan untuk kedua orang tua, pembimbing, dan seluruh rekan yang telah membantu dalam pembuatan skripsi ini*

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya penulis berhasil menyusun skripsi dengan judul "Sistem Pengatur Suhu Inkubator Otomatis". Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

- Kedua orang tua dan kakak yang selalu memberikan dukungan untuk menyelesaikan skripsi ini.
- Dosen pembimbing, Pak Chandra yang memberikan bimbingan, masukan, dan tambahan wawasan selama proses pembuatan skripsi ini sehingga selesai dengan baik.
- Teman-teman saya terutama grup "Bacot Family" yang sudah menemani saya selama kuliah.
- Seluruh dosen dan pekaya Universitas Katholik Parahyangan Bandung, terutama Jurusan Teknik Informatika yang telah memberikan ilmu serta motivasi kepada penulis.
- Seluruh teman-teman lainnya yang tidak dapat disebutkan satu per satu yang memberikan motivasi bagi penulis dalam proses pembuatan skripsi ini.

Bandung, Januari 2020

Penulis



# DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xix</b>
<b>DAFTAR TABEL</b>	<b>xxi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	3
1.3 Tujuan . . . . .	3
1.4 Batasan Masalah . . . . .	3
1.5 Metodologi . . . . .	4
1.6 Sistematika Pembahasan . . . . .	4
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Mesin Penetas Telur . . . . .	5
2.2 <i>Internet Of Things (IoT)</i> . . . . .	6
2.3 Komponen Pendukung <i>IOT</i> . . . . .	7
2.3.1 <i>Microcontroller</i> . . . . .	7
2.3.2 <i>Relay</i> . . . . .	11
2.3.3 <i>Sensor</i> . . . . .	13
2.4 <i>Message Queuing Telemetry Transport (MQTT)</i> . . . . .	16
2.4.1 <i>Arsitektur MQTT</i> . . . . .	17
2.4.2 <i>Kelebihan MQTT</i> . . . . .	18
2.4.3 <i>Sinyal Kontrol</i> . . . . .	18
2.4.4 <i>Topik dan Quality of Service(Qos)</i> . . . . .	19
<b>3 PERCOBAAN DAN ANALISIS</b>	<b>21</b>
3.1 Percobaan . . . . .	21
3.1.1 Pengaturan <i>Microcontroller</i> di Arduino IDE . . . . .	21
3.1.2 Percobaan Pendeteksian Suhu Menggunakan Modul Pendeteksi Suhu dan Kelembapan DHT11 . . . . .	23
3.1.3 Percobaan Pengiriman Pesan Antar <i>Microcontroller</i> . . . . .	24
3.1.4 Percobaan Menyalakan Lampu dan Kipas Dengan Menggunakan <i>Relay</i> . . . . .	28
3.1.5 Percobaan Mengirim Data dari <i>Microcontroller</i> ke Basis Data . . . . .	30
3.2 Analisis . . . . .	32
3.2.1 Analisis Struktur Sederhana Sistem Pengatur Suhu Otomatis . . . . .	32
3.2.2 Analisis Perangkat Keras . . . . .	33
3.2.3 Analisis Perangkat Lunak . . . . .	34
3.2.4 Analisis Diagram <i>ER</i> . . . . .	35
3.2.5 Analisis Diagram <i>Use Case</i> . . . . .	36

3.2.6	Analisis Diagram <i>Data Flow</i> . . . . .	38
3.2.7	Analisis Diagram Alir . . . . .	40
<b>4</b>	<b>PERANCANGAN</b>	<b>45</b>
4.1	Perancangan Perangkat Keras . . . . .	45
4.1.1	Perancangan <i>Master</i> . . . . .	46
4.1.2	Perancangan <i>Slave</i> . . . . .	46
4.1.3	Perancangan Inkubator . . . . .	47
4.2	Perancangan Modul . . . . .	47
4.2.1	Modul <i>Web</i> . . . . .	47
4.2.2	Modul <i>Master</i> . . . . .	50
4.2.3	Modul <i>Slave</i> . . . . .	55
4.3	Perancangan Komunikasi Antar <i>Master</i> dan <i>Slave</i> . . . . .	57
4.4	Perancangan Basis Data . . . . .	58
4.4.1	Diagram <i>Relasional Model</i> . . . . .	59
4.4.2	Perancangan Basis Data Fisik . . . . .	59
4.5	Perancangan Antar Muka . . . . .	60
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>63</b>
5.1	Implementasi Perangkat Keras . . . . .	63
5.1.1	Perangkat Keras <i>Master</i> . . . . .	63
5.1.2	Perangkat Keras <i>Slave</i> . . . . .	63
5.1.3	Implementasi Kotak Inkubator . . . . .	64
5.2	Implementasi Perangkat Lunak . . . . .	65
5.2.1	Lingkungan Implementasi Perangkat Lunak . . . . .	65
5.2.2	Implementasi <i>MQTT Broker</i> . . . . .	66
5.2.3	Implementasi Antar Muka . . . . .	66
5.2.4	Implementasi Basis Data . . . . .	67
5.2.5	Kode Program Hasil Implementasi Perangkat Lunak . . . . .	68
5.3	Pengujian . . . . .	68
5.3.1	Rancangan Perangkat Pengujian . . . . .	68
5.3.2	Pengujian Fungsional . . . . .	69
5.3.3	Pengujian Eksperimental . . . . .	76
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>83</b>
6.1	Kesimpulan . . . . .	83
6.2	Saran . . . . .	83
	<b>DAFTAR REFERENSI</b>	<b>85</b>
	<b>A KODE PROGRAM MASTER</b>	<b>87</b>
	<b>B KODE PROGRAM SLAVE</b>	<b>89</b>
	<b>C KODE PROGRAM WEB SERVER</b>	<b>93</b>

## DAFTAR GAMBAR

1.1	Contoh <i>Device IoT</i> yaitu OpenSprinkler . . . . .	1
1.2	Contoh <i>microcontroller</i> . . . . .	2
1.3	Contoh Gambaran Cara Kerja <i>MQTT</i> . . . . .	2
1.4	Gambar Inkubator Konvensional . . . . .	3
2.1	Contoh Gambar Penetas Telur dengan Pemanas Lampu Minyak Tanah [1] . . . . .	6
2.2	Contoh Gambar Mobil Tanpa Pengemudi . . . . .	6
2.3	Contoh Perangkat Arduino . . . . .	8
2.4	Wemos D1 . . . . .	8
2.5	Wemos D1 Mini . . . . .	8
2.6	NodeMcu V3 . . . . .	9
2.7	Contoh Gambaran Arduino <i>IDE</i> . . . . .	9
2.8	Contoh Tampilan <i>Serial Monitor</i> pada Arduino <i>IDE</i> . . . . .	10
2.9	<i>Relay</i> (Dickson, 2016) . . . . .	11
2.10	Ilustrasi Proses Pendeteksian Sensor . . . . .	13
2.11	Contoh Gambaran Sensor Mekanik . . . . .	14
2.12	Contoh Gambaran Sensor Optik . . . . .	14
2.13	Contoh Gambaran Sensor <i>Semiconductor</i> . . . . .	14
2.14	Contoh Gambaran Sensor <i>Electrochemical</i> . . . . .	15
2.15	Contoh Gambaran <i>Biosensor</i> . . . . .	15
2.16	Contoh Gambaran Sensor DHT11 . . . . .	16
2.17	Contoh <i>MQTT Broker Mosquitto</i> . . . . .	17
2.18	Contoh Perintah <i>Subscribe</i> pada Mosquitto . . . . .	18
2.19	Contoh Perintah <i>Publish</i> pada Mosquitto . . . . .	18
2.20	Contoh Gambaran Cara Kerja <i>MQTT</i> . . . . .	18
2.21	Contoh Gambaran Cara Kerja <i>QoS</i> . . . . .	20
3.1	Menu <i>Preferences</i> . . . . .	22
3.2	Menu <i>Boards Manager</i> . . . . .	22
3.3	Skema Pemasangan Modul DHT11 pada <i>Microcontroller</i> . . . . .	23
3.4	Hasil Pendeteksian Suhu Ruangan Menggunakan DHT11 . . . . .	24
3.5	Hasil Percobaan Interaksi Antar <i>Microcontroller</i> pada Modul <i>Master</i> . . . . .	25
3.6	Hasil Percobaan Interaksi Antar <i>microcontroller</i> pada Modul <i>Slave</i> . . . . .	26
3.7	Hasil Percobaan Interaksi Dua ke Satu Antar <i>Microcontroller</i> pada Modul <i>Master</i> . . . . .	27
3.8	Hasil Percobaan Interaksi Antar <i>Microcontroller</i> pada Modul <i>Slave 1</i> . . . . .	28
3.9	Hasil Percobaan Interaksi Antar <i>Microcontroller</i> pada Modul <i>Slave 2</i> . . . . .	28
3.10	Hasil Percobaan Menyalakan Kipas Dengan <i>Relay</i> . . . . .	29
3.11	Hasil Percobaan Menyalakan Lampu <i>Relay</i> . . . . .	29
3.12	Hasil Percobaan Mengirim Data dari <i>Microcontroller</i> ke Basis Data . . . . .	32
3.13	Ilustrasi Sederhana Struktur Sistem Pengatur Suhu Otomatis . . . . .	33
3.14	Diagram <i>ER</i> . . . . .	35
3.15	<i>Use Case Diagram</i> Perancangan Perangkat Lunak . . . . .	36
3.16	Data Context Diagram . . . . .	39

3.17	Data Flow Diagram	39
3.18	Diagram Alir <i>Master</i>	42
3.19	Diagram Alir <i>Slave</i>	43
4.1	Gambaran Sederhana Rancangan Perangkat Keras	45
4.2	Gambaran Perancangan <i>Slave</i>	46
4.3	Gambaran Perancangan inkubator	47
4.4	Contoh Pesan Komunikasi Yang Dikirim Oleh Inkubator 1 Pada <i>Serial Monitor Arduino IDE</i>	58
4.5	Contoh Pesan Komunikasi Yang Dikirim Oleh Inkubator 1 pada <i>Command Prompt</i>	58
4.6	Contoh Pesan Komunikasi Yang Diterima Oleh <i>Master</i> Pada <i>Serial Monitor Arduino IDE</i>	58
4.7	Contoh Pesan Komunikasi Yang Dikirim Oleh <i>Master</i> pada <i>Command Prompt</i>	58
4.8	Rancangan Basis Data	59
4.9	Rancangan Antar Muka halaman <i>login</i> pada <i>Web Browser</i>	60
4.10	Rancangan Antar Muka halaman utama pada <i>Web Browser</i>	61
5.1	Hasil Implementasi Kotak Inkubator Bagian Dalam	64
5.2	Hasil Implementasi Kotak Inkubator Bagian Luar	64
5.3	Tampilan Antar Muka	66
5.4	Tampilan Antar Muka	67
5.5	Struktur tabel <i>login</i> pada basis data	67
5.6	Struktur tabel <i>batas</i> pada basis data	67
5.7	Struktur tabel <i>incubator</i> pada basis data	68
5.8	Topologi Pengujian	69
5.9	Hasil Pengujian Komunikasi Antara <i>Master</i> dan <i>Slave</i>	70
5.10	Halaman <i>Login</i> Dapat Diakses	71
5.11	Tampilan <i>Pop-Up Alert</i> Jika Salah Memasukkan <i>Username</i> dan <i>Password</i>	71
5.12	Tampilan Halaman <i>Login</i> Jika Dilakukan Langkah 3	72
5.13	Pengguna Akan Diarahkan Ke Halaman Data Monitoring Jika Berhasil <i>Login</i>	72
5.14	Tampilan Halaman <i>Data Monitoring</i> Pada Bagian Batas Suhu Jika Dilakukan Langkah 1	73
5.15	Tampilan <i>Pop-Up Alert</i> Jika Memasukkan Batas Bawah Lebih Besar Dari Batas Atas	73
5.16	Tampilan <i>Pop-Up Alert</i> Jika Memasukkan Batas Bawah Sama Dengan Batas Atas	73
5.17	Tampilan Berhasil Mengubah Batas Suhu	74
5.18	Hasil Grafik Pada <i>Data Monitoring</i>	74
5.19	Hasil Tabel Pada <i>Data Monitoring</i>	74
5.20	Grafik Saat Listrik Dimatikan	76
5.21	Data Suhu Pada Inkubator 1 Ketika Terkena Hembusan Angin Dari Hairdryer	77
5.22	Tampilan Sistem Ketika <i>Access Point</i> Dimatikan	77
5.23	Tampilan <i>Error</i> Jika Basis Data Dimatikan	78
5.24	Data Inkubator 1 Ketika Tutup Inkubator Dibuka	78
5.25	Data Inkubator 2 Ketika Tutup Inkubator Dibuka	79
5.26	Data Inkubator 1 Ketika <i>Master</i> Dimatikan	80
5.27	Data Inkubator 2 Ketika <i>Master</i> Dimatikan	80
5.28	Data Inkubator 1 Ketika <i>Slave</i> 1 Dimatikan	80
5.29	Data Inkubator 2 Ketika <i>Slave</i> 1 Dimatikan	81

## DAFTAR TABEL

3.1	Penjelasan Entitas <i>Incubator</i> . . . . .	35
3.2	Penjelasan Entitas <i>Login</i> . . . . .	35
3.3	Penjelasan Entitas Batas . . . . .	36
4.1	Tabel <i>Incubator</i> . . . . .	59
4.2	Tabel Batas . . . . .	60
4.3	Tabel <i>Login</i> . . . . .	60
5.1	Tabel Data Hasil Pengujian Pengaturan Suhu <i>Incubator 1</i> . . . . .	75
5.2	Tabel Data Hasil Pengujian Pengaturan Suhu <i>Incubator 2</i> . . . . .	75

# BAB 1

## PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian dan sistematika pembahasan.

### 1.1 Latar Belakang

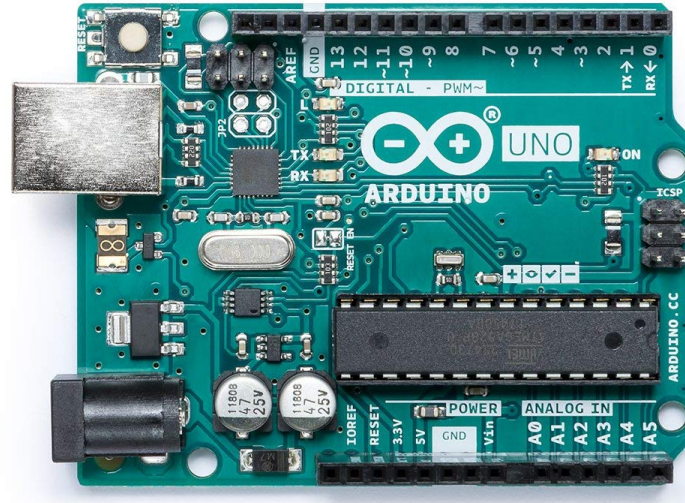
Di era perkembangan teknologi dan internet yang sangat pesat sehingga membantu dalam peningkatan kualitas dan kesejahteraan hidup manusia. Namun, terdapat beberapa teknologi yang belum efektif dalam membantu manusia menjalankan aktivitasnya. Penerapan sistem-sistem yang belum efektif tersebut dapat dibuat lebih efektif dengan bantuan teknologi *Internet Of Things(IoT)*. *IoT* adalah sebuah jaringan dari benda-benda yang awalnya tidak terhubung dengan internet tetapi kemudian ditanam dengan benda-benda elektronik yang memungkinkan benda-benda yang tidak dapat berkomunikasi tanpa internet tersebut menjadi dapat saling terhubung dan saling menukar data. Tujuan dari *IoT* adalah agar dapat mengurangi interaksi manusia pada benda-benda tersebut dengan memanfaatkan internet. Gambar 1.1 adalah contoh *device IoT* yaitu OpenSprinkler yang dapat menyiram tanaman secara otomatis sesuai jadwal yang telah ditentukan. OpenSprinkler adalah sebuah pengontrol *sprinkler* / irigasi yang memiliki *User Interface (UI)*, dapat diakses secara *remote*, dan dapat mengatur waktu penyiraman dengan data dari sensor temperatur dan kelembapan.



Gambar 1.1: Contoh *Device IoT* yaitu OpenSprinkler

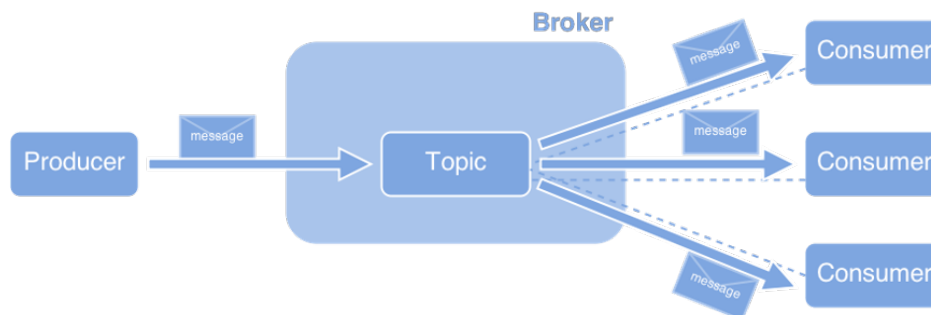
*Microcontroller* adalah sebuah komputer kecil pada sebuah sirkuit terintegrasi yang memiliki sebuah prosesor inti, memori, dan perangkat *input/output* yang dapat diprogram. *Microcontroller* digunakan untuk produk yang dapat dikendalikan secara otomatis atau di era sekarang sering disebut juga sebagai "smart device". *Microcontroller* berkomunikasi dengan sinyal elektronik atau disebut juga sebagai *control signal*. Setelah itu digunakan protokol yang memungkinkan beberapa *microcontroller* dapat saling berkomunikasi dan saling mengendalikan, untuk bentuk dari interaksi antar *microcontroller* tergantung dari contoh kasus yang diberikan, misalnya menggunakan

*microcontroller* pada ruang A untuk mengirim sinyal matikan lampu ke *microcontroller* pada ruang B. Untuk mendukung kegunaan tersebut maka *microcontroller* dapat dihubungkan dengan perangkat-perangkat eksternal lainnya seperti modul *speaker*, modul lampu LED, modul pendeteksi suhu dan kelembapan ruangan, modul frekuensi radio, serta banyak modul lainnya. Gambar 1.2 merupakan contoh *microcontroller* Arduino Uno.



Gambar 1.2: Contoh *microcontroller*

*Message Queuing Telemetry Transport (MQTT)* adalah sebuah protokol komunikasi yang sangat ringan karena pesan yang dikirim berukuran kecil dan koneksi untuk protokol tersebut dapat digunakan berulang kali. Pada umumnya *MQTT* digunakan untuk lokasi yang sulit dijangkau, dan memiliki keterbatasan *bandwidth*. Cara kerja *MQTT* adalah dengan mengirimkan pesan ke seluruh entitas yang telah bergabung dengan *broker* sehingga seluruh entitas tersebut dapat berkomunikasi dengan entitas lainnya. Melalui *MQTT*, performa *device IoT* dapat bertambah dibandingkan dengan protokol *HTTP* (protokol umum pada internet) karena tidak tepatnya penggunaan model interaksi di protokol tersebut, *MQTT* menggunakan model *publish/subscribe* tetapi *HTTP* menggunakan model *request/response*. Gambar 1.3 adalah contoh gambaran cara kerja *MQTT*.



Gambar 1.3: Contoh Gambaran Cara Kerja *MQTT*

Pada skripsi ini, telah dibuat sebuah sistem pengatur suhu inkubator otomatis dengan menggunakan *microcontroller*, karena inkubator untuk menetas telur yang konvensional memiliki beberapa masalah seperti membutuhkan pengecekan ke inkubator oleh manusia setiap periode tertentu untuk mengecek suhu dan variabel-variabel lainnya yang mempengaruhi proses penetasan telur seperti kelembapan, dan masih ada kemungkinan terjadi kejadian *human error* seperti lupa mematikan pemanas pada inkubator yang mengakibatkan suhu inkubator menjadi terlalu tinggi

sehingga telur gagal menetas. Oleh karena itu, dengan berkembangnya teknologi *IoT* hal-hal seperti *human-error* dan campur tangan manusia pada pengaturan suhu pada inkubator dapat dikendalikan oleh *microcontroller*.

Sistem pengatur suhu inkubator otomatis menghubungkan *microcontroller* agar dapat saling berkomunikasi dengan *MQTT* melalui perantara sebuah *MQTT broker* yaitu Mosquitto. Mosquitto merupakan sebuah *MQTT broker* yang *open source*. Lalu dilakukan proses implementasi *MQTT* untuk menghubungkan *microcontroller* untuk mengatur suhu dan kelembapan dalam inkubator dengan mengendalikan kipas sebagai pendingin lalu ada lampu pijar yang berfungsi untuk menghasilkan panas untuk menaikkan suhu pada inkubator. Gambar 1.4 merupakan contoh gambar inkubator yang konvensional.



Gambar 1.4: Gambar Inkubator Konvensional

## 1.2 Rumusan Masalah

- Bagaimana cara membangun *MQTT broker* Mosquitto?
- Bagaimana cara komunikasi antar *microcontroller* dengan menggunakan *MQTT*?
- Bagaimana cara membangun protokol *MQTT* yang dapat dimanfaatkan oleh *IoT* untuk mematikan dan menyalakan perangkat listrik?
- Bagaimana cara membangun sistem pengatur suhu inkubator otomatis dengan menggunakan *microcontroller*?

## 1.3 Tujuan

- Membangun *MQTT broker* dengan Mosquitto.
- Membuat protokol agar beberapa *microcontroller* dapat berkomunikasi dan saling mengendalikan.
- Membuat perangkat lunak agar *microcontroller* dapat dimanfaatkan *IoT* untuk mematikan atau menyalakan perangkat listrik.
- Membangun sistem pengatur suhu inkubator otomatis dengan menggunakan *microcontroller*.

## 1.4 Batasan Masalah

1. Parameter yang akan dimonitor adalah suhu.
2. *Broker* yang digunakan adalah Eclipse Mosquitto versi 1.5.4.



## 1.5 Metodologi

1. Melakukan Studi Literatur dari buku, *paper*, dan media lainnya untuk mengetahui teori-teori yang berkaitan dengan perangkat lunak dan alat yang akan digunakan yaitu tentang *microcontroller*, sensor, *relay*, *MQTT*, dan mesin penetas telur.
2. Melakukan analisis terhadap lingkungan yang dibutuhkan perangkat lunak dan perangkat keras agar dapat sistem pengatur suhu inkubator otomatis dapat berfungsi dengan baik.
3. Melakukan perancangan perangkat lunak yang akan dibuat berdasarkan hasil studi literatur, dan analisis yang telah dilakukan.
4. Melakukan implementasi sistem pengatur suhu inkubator otomatis dari hasil perancangan yang telah dibuat sebelumnya.
5. Melakukan pengujian terhadap sistem pengatur suhu inkubator otomatis yang telah dibangun.

## 1.6 Sistematika Pembahasan

1. Bab Pendahuluan  
Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan permasalahan, sistematika pembahasan, dan metodologi penelitian.
2. Bab Landasan Teori  
Bab ini berisi teori-teori mengenai IoT, *microcontroller*, mesin penetas telur, *relay*, sensor, *MQTT*, dan *MQTT broker*.
3. Bab Analisis  
Bab ini berisi percobaan tentang peralatan dan pengiriman data, lalu juga berisi analisis mengenai struktur sistem, perangkat keras, perangkat lunak, *microcontroller* di Arduino IDE, diagram *ER*, diagram *use case*, diagram *data flow*, dan diagram alir .
4. Bab Perancangan  
Bab ini berisi rancangan-rancangan perangkat keras, antar muka, basis data, komunikasi antar *master slave*, dan modul.
5. Bab Implementasi dan Pengujian  
Bab ini berisi hasil implementasi perangkat keras, perangkat lunak, dan pengujian.
6. Bab Kesimpulan dan Saran  
Bab ini berisi tentang kesimpulan yang diambil dari hasil pengujian dan saran.