IMPLEMENTASI ARSITEKTUR PUBLISH AND SUBSCRIBE PADA ALAT MONITORING SUHU DAN KELEMBABAN KANDANG ULAR PYTHON REGIUS MENGGUNAKAN NODEMCU (ESP8266)

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Habib Zainal Sarif
NIM: 135150300111036

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

IMPLEMENTASI ARSITEKTUR PUBLISH AND SUBSCRIBE PADA ALAT MONITORING SUHU DAN KELEMBABAN KANDANG ULAR PYTHON REGIUS MENGGUNAKAN NODEMCU (ESP8266)

SKRIPSI
KEMINATAN TEKNIK KOMPUTER
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Habib Zainal Sarif
NIM: 135150300111036
Skripsi ini telah diuji dan dinyatakan lulus pada
2 Januari 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Mochammad Hannats Hanafi Ichsan, S.ST., M.T.	Rizal Maulana, S.T., M.T,. M.Sc
NIK: 201405 8812291001	NIK: 2016078910091001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T., Ph.D
NIP: 197105182003121001

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PERNYATAAN ORISINALITAS

Penulis menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan penulis, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, penulis bersedia skripsi ini digugurkan dan gelar akademik yang telah penulis peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Januari 2019

METERAI TEMPEL
8ECB9AFF345105984
6000
ENAM RIBU RUPIAH

Habib Zainal Sarif

NIM: 135150300111036

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena hanya dengan kasih dan anugerahnya penulis dapat menyelesaikan skripsi dengan judul Implementasi Arsitektur Publish And Subscribe Pada Alat Monitoring Suhu dan Kelembaban Kandang Ular Python Regius Menggunakan Nodemcu (Esp8266).

Melalui kesempatan ini, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan skripsi ini, diantaranya:

1. Allah yang Maha Esa yang selalu memberikan petunjuk dan hikmah dalam penulisan ini.
2. Orang Tua dan Keluarga atas nasehat, kasih sayang serta dukungan materil dan moril.
3. Bapak Moch. Hannats Hanafi I, S.ST., M.T. dan bapak Rizal Maulana, S.T., M.T., M.Sc. selaku dosen pembimbing skripsi yang telah memberikan masukan dan ilmu serta saran dalam menyelesaikan skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Dahnial Syauqi, S.T., M.T., M.Sc. selaku Ketua Program Studi Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Kepada REPTILOPHOBIA STORE yang telah membantu dalam proses pengujian penelitian ini.
7. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat secara langsung atau tidak langsung agar terselesaikan skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 2 Januari 2019

Penulis

habibsarif2006@gmail.com

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

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 khusunya 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

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

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 toosl 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*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL x
DAFTAR GAMBAR xi
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan masalah 2
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan masalah 3
1.6 Sistematika pembahasan 4
BAB 2 LANDASAN KEPUSTAKAAN 6
2.1 Kajian Pustaka 6
2.2 Landasan Teori 7
2.2.1 Publish and Subscribe 7
2.2.2 MQTT (*Message Queuing Telemetry Transport*) 8
2.2.3 NodeMCU (ESP8266) 8
2.2.4 JSON (*JavaScript Object Notation*) 9
2.2.5 Sensor *DHT11* 10
BAB 3 METODOLOGI 12
3.1 Studi Kepustakaan 12
3.2 Analisa Kebutuhan 13
3.2.1 Kebutuhan Perangkat Keras 13
3.2.2 Kebutuhan Perangkat Lunak 13
3.3 Perancangan Sistem 14
3.4 Implementasi 15
3.4.1 Implementasi Perangkat Keras 15

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3.4.2 Implementasi Perangkat Lunak 15
3.5 Pengujian dan Analisis 15
3.6 Kesimpulan 16
BAB 4 Rekayasa kebutuhan 17
4.1 Gambaran Umum Sistem 17
4.1.1 Prespektif Sistem 17
4.1.2 Ruang Lingkup 17
4.1.3 Karakterisitik Pengguna 17
4.1.4 Lingkungan Operasi Sistem 18
4.1.5 Batasan Perancangan Dan Implementasi 18
4.2 Rekayasa Kebutuhan 19
4.2.1 Kebutuhan Antarmuka 19
4.2.2 Kebutuhan Fungsional 20
4.2.3 Kebutuhan non Fungsional 21
BAB 5 PERANCANGAN DAN IMPLEMENTASI 22
5.1 Perancangan 22
5.1.1 Perancangan Perangkat Keras 22
5.1.2 Perancangan Perangkat lunak 26
5.2 Implementasi 32
5.2.1 Implementasi Perangkat Keras 32
5.2.2 Implementasi Perangkat Lunak 33
BAB 6 PENGUJIAN DAN ANALISIS 42
6.1 Pengujian Sensor *DHT11* 42
6.1.1 Tujuan Pengujian 42
6.1.2 Prosedur Pengujian 42
6.1.3 Hasil dan Analisis Pengujian 42
6.2 Pengujian Tampilan LCD 20X4 45
6.2.1 Tujuan Pengujian 45
6.2.2 Prosedur Pengujian 45
6.2.3 Hasil dan Analisis Pengujian 45
6.3 Pengujian Kinerja RTC DS3231 46
6.3.1 Tujuan pengujian 46

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.3.2 Prosedur Pengujian 47
6.3.3 Hasil Pengujian dan analisis 47
6.4 Pengujian Arsitektur Publish dan Subscribe 47
6.4.1 Tujuan pengujian 48
6.4.2 Prosedur Pengujian 48
6.4.3 Hasil dan Analisis Pengujian 48
6.5 Pengujian Protokol MQTT 49
6.5.1 Tujuan Pengujian 49
6.5.2 Prosedur Pengujian 49
6.5.3 Hasil dan Analisis Pengujian 50
6.6 Pengujian Aplikasi Android 51
6.6.1 Tujuan pengujian 51
6.6.2 Prosedur Pengujian 51
6.6.3 Hasil Pengujian dan analisis 51
BAB 7 Penutup 54
7.1 Kesimpulan 54
7.2 Saran 54
Daftar pustaka 56
LAMPIRAN 58

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

Tabel 2. 1 Spesifikasi NodeMCU *9*
Tabel 5. 1 Konfigurasi Pin DHT11 ..... *25*
Tabel 5. 2 Konfigurasi Pin RTC DS3231 ..... *26*
Tabel 5. 3 Konfigurasi Pin LCD 20x4 dengan I2C *26*
Tabel 5. 4 Potongan kode program sensor DHT11 ..... *33*
Tabel 5. 5 Potongan kode program RTC DS3231 ..... *35*
Tabel 5. 6 Potongan kode program LCD *37*
Tabel 5. 7 Potongan Kode Program Android *38*
Tabel 6. 1 Perhitungan Persentase Kesalahan Suhu DHT11 ..... *43*
Tabel 6. 2 Perhitungan Persentase Kesalahan Kelembaban DHT11 ..... *44*
Tabel 6. 3 Hasil Publish data pad sistem *48*
Tabel 6. 4 Hasil Subscribe data pad sistem *49*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2. 1 Diagram Sistem suatu protokol MQTT sederhana 8
Gambar 2. 2 Nilai Diagram Sintaks pada JSON..... 10
Gambar 3. 1 Diagram alir metode penelitian 12
Gambar 3. 2 Blok Diagram Sistem Monitoring 14
Gambar 5. 1 Peletakan Sistem Monitoring pad Kandang Ular..... 22
Gambar 5. 2 nodemcu dengan sensor DHT11 23
Gambar 5. 3 LCD dengan i2c pada nodemcu 23
Gambar 5. 4 RTC DS3231 dengan nodemcu 24
Gambar 5. 5 Rancangan Skematik Keseluruhan sistem 25
Gambar 5. 6 Flowchart dari sensor DHT11 27
Gambar 5. 7 Flowchat dari sistem RTC DS3231 28
Gambar 5. 8 Flowchart dari sistem LCD 28
Gambar 5. 9 Flowchart dari sistem Android 29
Gambar 5. 10 Perancangan Tampilan Aplikasi Android 30
Gambar 5. 11 Perancangan Protokol Komunikasi MQTT 31
Gambar 5. 12 Perancangan Format Teks JSON 31
Gambar 5. 13 Implementasi dari sistem 32
Gambar 5. 14 Implentasi sistem pada kandang ular Python Regius 33
Gambar 5. 15 Tampilan Utama Aplikasi Android 41
Gambar 6. 1 Tampilan LCD pada awal Proses 46
Gambar 6. 2 Tampilan Utama LCD saat sistem berjalan 46
Gambar 6. 3 Tampilan sistem pada Serial Monitor 47
Gambar 6. 4 Proses koneksi dengan Broker 50
Gambar 6. 5 Hasil pada MQTT client 50
Gambar 6. 6 Tampilan awal aplikasi Android 51
Gambar 6. 7 Tampilan saat subscribe 52
Gambar 6. 8 Tampilan saat Unsubscribe 52

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 1 PENDAHULUAN

1.1 Latar belakang

Saat ini kepemilikan satwa di Indonesia semakin meluas. Kalau di zaman dulu hanya orang-orang menengah keatas yang memilih hewan peliharaan sebagai hobi, untuk saat ini rakyat menengah kebawah pun mampu untuk memiliki satwa sebagai hobinya. Burung berkicau adalah salah satu satawa yang paling digemari oleh banyak kalangan orang di Indonesia. Selain untuk didengarkan suara khas kicaunya, kini lomba burung berkicau semakin marak. Akan tetapi tidak hanya burung berkicau yang menjadi satawa pilihan untuk dipelihara. Saat ini semakin banyak satwa jenis reptil yang menjadi satwa peliharaan salah satunya adalah ular. Mungkin banyak orang yang beranggapan semua ular itu membahayakan buat orang-orang sekitar karena memiliki racun pada gigitannya. Tapi ternyata tidak semua ular memiliki racun. Salah satunya adalah ular yang masih satu keluarga dengan *python.* Banyak sekali jenis python yang ada di Indonesia. Warna corak yang beragam membuat semakin banyak masyarakat yang ingin memilahara ular jenis ini. Selain karena perawatannya yang relatif mudah, pemberian makan pun hanya perlu dilakukan seminggu sekali saja. Namun selain ular python yang asli endemik Indonesia, ada satu jenis ular python yang menjadi primadona yakni *Regius Python* atau biasa masayarakat Indonesia sebut dengan nama *Ball Python.*

Regius Python atau yang biasa disebut *Ball Python* ini memiliki ukuran tubuh yang relatif lebih pendek daripada ular python endemik asli indoesia. Panjang tubuhnya ketika sudah dewasa hanya sampai 1,3 meter saja. Selain itu ular ini sangatlah pemalu, ular ini akan cenderung bersembunyi menutupi bagian kepalanya sampai *menyerupai* bola untuk menghindari ancaman. Oleh sebab itu ular ini sering kita kenal sebagai *Ball Python.* Kemudian yang paling menarik dari ular ini adalah jumlah corak yang sangat beragam sehingga sangat digemari oleh masyarakat. Menurut web *World of Ball Python* terdapat 6251 morps atau corak warna yang ada. Akan tetapi ular ini sangat rentan terhadap penyakit flu karena suhu tropis yang ada di Indonesia. 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 Pythons membutuhkan tingkat kelembaban sekitar 50-60%, jadi disarankan untuk membeli Hydrometer untuk memantau ini (Python, 2017). Oleh karena itu penulis ingin membuat sebuah alat yang mampu memonitoring tingkat suhu dan kelembaban yang nantinya akan diterapkan pada kandang ular ini. Yang nantinya data hasil pengamatan dapat diterima dengan klien sehingga dapat mengurangi resiko penyakit flu pada ular. Untuk itu perlu diterapkan suatu teknik terbaru yakni *IoT(Internet of Thing).* Terdapat banyak sistem yang dapat digunakan dalam *Iot* itu sendiri salah satunya adalah *Publish and Subscribe,* arsitektur ini dipilih karena sangat efektif untuk melakukan kinerja sebuah sitem yang sederhana. *Publish and subscribe* sendiri merupakan metode komunikasi berdasarkan kejadian dan mempunyai keunggulan dalam hal skalabilitas, time decoupling, space decoupling,

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

dan synchronization decoupling. 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).

Adapun penelitian sebelumnya yang memiliki keterkaitan dengan skripsi yang saya buat adalah penelitian yang dilakukan oleh Kasim dkk (2018) yang berjudul Implementasi Otomasi Kandang dalam Rangka Meminimalisisr *Heat Stress* pada Ayam Boiler dengan metode *Fuzzy Sugeno*", yaitu sebuah penelitian yang bertujuan untuk menstabilkan sebuah kandang hewan dengan mengukur suhu dan kelembabannya yang kemudian akan mengurangi tingkat kematian pada hewan itu sendiri. Selain itu penelitian yang masih berhubungan dengan penelitian ini adalah sebuah kinerja sistem *IoT* yang disusun oleh Taringan dkk (2014) yang berjudul "Pengukuran Kinerja Sitem *Publish/Subscribe* Menggunakan Protokol MQTT", dengan tujuan untuk mengukur kinerja dari arsitektur yang sama dengan yang penulis buat dengan menggunakan protokol MQTT dimana nantinya akan diketahui peforma dari arsitektur itu sendiri.

Pada penlitian kali ini penulis ingin membuat sebuah penelitian IoT yang menggunakan arsitektur *publish and subscribe* sebagai algoritma utama yang mendukungsistem 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 cerdas lainnya sehingga benda yang tadinnya konvensional akan menjadi sebuah benda yang lebih modern dan cerdas (Hussain, 2016). Perkembangan teknologi IoT ini sendiri semakin pesat dan tenar di kalangan masyarkat umum. Selain dari kalangan khusus seperti peneliti, teknologi ini pun sudah masuk dalah perkembangan industri sebagai contoh adalah pada tahun 2006 saja sudah terdapat 2 milliar lebih benda yang sudah terkoneksi dengan jaringan inrternet, berdasarkan salah satu perusahan teknologi terkemuka Intel terdapat hingga 200 milliar lebih objek yang akan mengalami perkembangan dari maafaat teknologi ini (Intel, 2015). Dan nantinya diharapkan bahwa tulisan ini akan bermanfaat untuk para pencinta reptil agar hewan kesayangannya dapat hidup dengan baik dan nyaman.

1.2 Rumusan masalah

Menurut uraian latar belakang yang telah dituliskan, maka penulis daapat menyimpulkan beberapa rumusan masalah berkaitan dengan latar belakang tersebut. Berikut rumusan masalahnya

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1. Bagaimana alat monitoring suhu dan kelembaban dengan arsitektur publish and subscribe dapat bekerja dengan baik pada kandang ular?
2. Bagaimana performa dari arsitektur publish and subscribe bekerja pada alat monitoring suhu dan kelebaban?
3. Bagaimana hasil yang dari sistem monitoring pada sebuah aplikasi android yang dibuat?

1.3 Tujuan

Dalam pembuatan laporan yang penulis buat, ada beberapa tujuan yang ingin dicapai pada sistem ini, yaitu :

1. Mampu mengetahui apakah alat monitoring suhu dan kelembaban dapat bekerja dengan baik menggunakan arsitektur publish and subscribe.
2. Mengetahui performa dari arsitektur publish and subscribe yang diterapkan pada alat monitoring suhu dan kelembaban.
3. Dapat mengurangi resiko kematian pada hewan karena ketidak nyamanan kandang.

1.4 Manfaat

Penlis beharap penelitian yang dilakukannya ini mampu memeberikan manfaat untuk banyak pihak. Adapun manfaat yang ingin dicapai oleh penulis adalah:

1. Bagi penulis
   Dapat mengimplementasikan arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular python regius. Serta memberikan pengetahuan dan pemahaman tantang kinerja dari alat monitoring suhu dan kelembaban setelah dimplementasikan arsitektur tersebut.
2. Bagi fakultas
   Memberikan gambaran teknologi baru yang semakin tidak terbatas karena semakin pesatnya teknologi baru ini.
3. Bagi masyarakat
   Dapat memberi pemahaman serta pengetahuanterhadapat sebuah arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular piton regius.

1.5 Batasan masalah

Penulis tidak ingin meluasnya pembahsan yang mungkin terjadi, oleh karena itu penulis memberikan pembatasan masalah antara lain:

1. Sistem hanya menggunakan data dari sensor.
2. Sistem hanya melakukan akusisi data menggunakan *Node*MCU sebagai mikrokontroller utama.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3. Sistem akan menggunakan arsitektur *publish and subscribe* yang akan didukung protokol MQTT sebagai protokol komunikasi utama.
4. Alat monitoring ini menggunakan jaringan sensor menggunakan Wi-Fi.
5. Sistem akan menampilkan hasil pengkuran melalui mobile device.

1.6 Sistematika pembahasan

Pada susb bab kali ini akan dipaparkan untuk memberikan sebuah gmabaran secara garis besar dari penulisan skripsi ini yang dituliskan dalam sebuah uraian yang mencakup bab-bab berikut ini:

Bab I : Pendahuluan

Pada bab ini penulis menyampaikan tentang pemikiran utama ditulisnya skripsi ini meliputi latar belakang pembuatan laporan skripsi, perrumusan masalah yang akan deselesaikan, pembatasan masalah yang diberikan, serta tujuan dari penulisan ini, kemudian manfaat dan sistematika penulisan. Dimana akan menjadi acuan agar mempermudah pembaca mengerti apa yang akan diteliti pada laporan yang ditulis.

Bab II : Tinjauan Pustaka dan Dasar Teori

Untuk bab dua ini penulis akan menjelaskan beberapa teori yang mendukung serta penelitian-penelitian yang telah dahulu sehingga dapat menguatkan dan menjadi dasar penelitian implementasi arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular python regius menggunakan nodemcu. Dimana acuan ini akan membantu penulis untuk menyelesaikan peneletian.

Bab III : Metodologi Penelitian

Pada bagian bab tiga ini penulis ingin membahas dan menguraikandaripada langkah kerja yang akan dilakukan dalam penulisan skripsi yang meliputi dari studi kepustakaan, analisis kebutuhan baik perangkat keras maupun lunak, perancangan sistem secara mendetail, kemedian pengimplementasian, serta melakukan pengujian dan analisis hasil, dan tak lupa untuk menarik kesimpulan. Dengan demikian penulis akan memberikan penjelasan secara rinci tentang tahap-tahap pengerjaan penelitan tentang arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular python regius menggunakan nodemcu.

BAB IV : Rekayasa Kebutuhan

Untuk bab kali ini akan dijelaskan secara menyeluruh serta mendetail tentang kebetuhan apa saja yang dibutuhkan oleh sistem yang meliputi kebutuhan perangkat keras maupun kebutuhan perangkat lunak, serta kebutuhan fungsional maupun non fungsional.

Bab V : Perancangan dan Implementasi

Pada bab kali ini akan dibahas mengenai perancangan utama sistem yang akan digunakan serta konfigurasi yang dilakukan untuk melakukan implementasi sistem pada arsitektur publish and subscribe pada alat

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

monitoring suhu dan kelembaban kandang ular python regius menggunakan nodemcu. Pada bab ini juga akan dijelaskan secara mendetail alur program yang akan dijalankan pada sistem tersebut.

Bab VI : Pengujian dan Analisis

Pada bab kali ini penulias ingin membahas tetntang pengujian serta analisis yang akan dilakukan pada sistem arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular python regius menggunakan nodemcu. Dimana akan dijelaskan beberapa pengujian yang meliputi pengujian sensor DHT11, pengujian LCD, pengujian aplikasi android serta pengujian pada arsitektur publish dan subscribe pada sistem.

Bab VII : Penutup

Untuk bab kali ini penulis akan menarik sebuah kesimpulan dari pengimplementasian serta pengujian perangkat keras maupun lunak pada sistem yang telah dikerjakan dalam skripsi ini dan juga berisikan saran untuk pengembangan lebih lanjut pada sistem arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular python regius menggunakan nodemcu.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab kali ini pembahasan difokuskan pada kajian pustaka yang mana adalah peninjauan kembali atas pustaka pendukung yang masih berhubungan dengan skripsi yang saya tulis ini. Selain itu dasar teori pada skripsi ini juga dibutuhkan untuk menjadi sebuah teori dasar agar pembaca dapat memahami serta mengerti permsoalaan yang akan dibahas oleh penulis pada penelitian kali ini.

2.1 Kajian Pustaka

Dalam kaajian pustaka kali ini akan membahas berbagai penelitian yang telah dilakukan terlebih dahulu sehingga menjadi alat pendukung untuk proses penelitian yang diajukan. Penelitian kali ini, terdapat beberapa kajian pustaka yang akan diambil dari beberapa penelitian terdahulu yang sudah ada. Yang pertama adalah Akbar, dkk (2016) yang berjudul "*Pervasive Device and Service Discovery Protocol In XBee Sensor Network"* adalah sebuah yang mampu untuk menerapkan sebuah protokol yang bertujuan untuk mengenali atau mengidentifikasi sebuah layanan baru pada suatu sistem yang didukung oleh sebuah sensor yang dilengkapi dengan Xbee. Berdasarkan hasil pengujian yang dilakukan terdapat beberapa nilai yang didapat terutama pada nilai waktu *round trip* dengan rata-rata untuk nilai dari *request* dan *respon* data dari gateway yakni sebesar 0.201 detik. Sedangkan untuk nilai waktu untuk mengambil data adalah sebesar 4,13 detik. Dimana penulis ini mengintegerasikan node klien dengan sebuah *transceiver.*

Untuk penelitian lainnya yang akan digunakan adalah Rochman, dkk (2016) yang berjudul "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada *Smarthome*" dimana penelitian kali ini dilakukan dengan menggunakan MQTT untuk protokol komunikasi utama yang memiliki kemapuan integeritas data hingga 100% dalam mengendalikan 2 buah lampu LED yang dilkukan untuk mengambil data intensitas cahaya dan juga data suhu. Disini penulis juga menggunakan *Wemos D1 R2* seabgai mikrokontroller utama sebagai node sensor maupun aktuator. Sedangakan untuk pengiriman datanya digunakan juga sebuah jaringang *Wi-Fi* yang juga terhubung dengan komputer klien yang ingin mengambil data dari sensor itu sendiri.

Berikutnya adalah penelitian dari Salsalina, dkk (2014) dengan judul "Pengukuran Kinerja Sistem *Publish/Subscribe* Menggunakan Protokol MQTT" yakni menjelaskan bahwa kompleksnya koneksi saat ini dikarenakan masiah bnyaknnya masyarakt awam yang masih menggunakan teknologi konvensiaonal yakni *point to point.* Dimana teknologi ini sudah semakin tidak efektif. Oleh karena itu adanya teknologi *publish and subscribe* ini diharapakan akan mampu menjadi solusi. Dimana sebuah *publisher* akan menerbitkan suatu data dengan topik yang telah ditentukan kemudian *subscriber* akan melakukan ketertarikan terhadap sebuah objek tadi. Sehingga akan menghemat penggunaan daya tersebut.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Berdasarkan peniltian yang telah diajelaskan diatas, dapat kita ketahui bahwa terdapat suatu metode yang digunakan untuk melakukan pengenalan perangkat terhadap suatu sensor telah dapat berhasil dengan angka keberhasilan mencapai 100%. Yang telah berhasil diterapkan terhadap suatu rumah cerdas yang memiliki keberagaman sensor sebagai pilihannya. Oleh karena itu fokus daripada penelitian kali ini bertujuan untuk mengimplementasikan arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular piton regius menggunakan nodemcu.

2.2 Landasan Teori

Berikut ini adalah beberapa teori yang digunakann dalam penelitian arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular piton regius menggunakan nodemcu. Adapun penjelasan-penjelasan mengenai teori tersbut dapat kita uraikan sebagai berikut :

2.2.1 Publish and Subscribe

Arsitektur *publish-subscribe* merupakan salah satu jenis arsitektur yang berada didalam komunikasi sistem terdistribusi (Dian, 2017). Mekanisme komunikasi yang terjadi didalam arsitektur ini merupakan komunikasi tidak langsung, sehingga perlu ada sesuatu yang menjembatani untuk melakukan suatu komunikasi antara pengirim dan penerima. Ada 3 entitas utama yang digunakan sebagai syarat komunikasinya yakni *broker* sebagai perantara, *publisher* sebagai pihak penyedia informasi serta *subscriber* sebagai pihak yang membutuhkan informasi.

Subscriber dan *publisher* akan menjalin komunikasi berdasarkan topik. Topik merupakan identitas tertentu yang memiliki pendekatan yakni ketertarikan atas suatu data yang telah diterbitkan. *Subscriber* hanya menerima data yang berisikan informasi dari broker sesuai dengan topik yang diinginkannya, sehingga *publisher* itu sendir tidak akan mengetahui *subscriber* yang sedang aktif pada topik yang sama. 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).

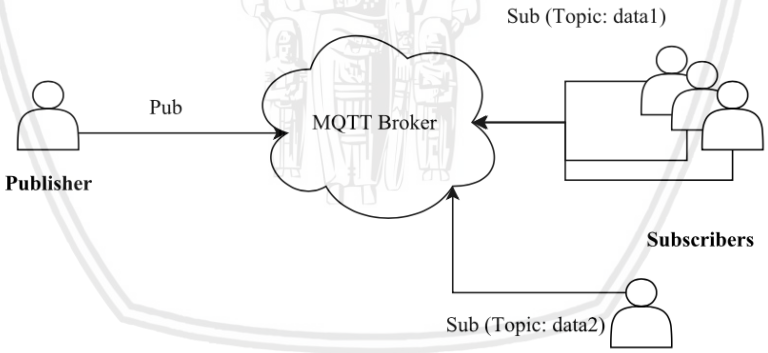
Pengunaan arsitektur ini pada sistem yang berbasiskan sistem pervasif memungkinkan untuk melakukan komunikasi secara efisien, hal ini dibutuhkan dikarenakan mengingat jumlah entitas yang tidak terbatas dan sistem yang cendurung kompleks sehingga dapat menghemat daya pada setiap perangkat. Selain itu, arsitektur ini cocok untuk diimplementasikan pada perangkat IoT yang memungkin perangkat aktuator untuk dapat langsung menerima data tanpa melakukan *request* terlebih dahulu seperti pada arsitektur *client-server*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.2.2 MQTT (*Message Queuing Telemetry Transport*)

MQTT adalah sebuahprotokol komuniaksi yang umum digunakan pada model *machine to machine* (m2m) dan juga mampu mendukung sebuah perangkat yang berbasikan *Internet of Things* (IoT). Protokol ini sendiri merupakan salah satu protokol yang biasa digunakan untuk melakukan sebuah arsitektur jaringan *publish and subscribe.* Sehingga memungkinkan sebuah komunikasi yang terjalin akan berjalan dengan sangat cepat dan efisien yang disebabkan arsitektur ini tidak menggunakan proses mekanisme *request-response* yang biasa kita temui pada protokol lain seperti pada protokol HTTP (*HyperText Transfer Protokol*) sehingga setiap pengguna diharapkan akan mendapatkan data maupun informasi secara *realtime*.

MQTT ini sendiri merupakan sebuah protokol yang mana didalamnya terdapat suatu *middleware* yakni *broker* yang berfungsi untuk pengendali utama yang berguna untuk menyimpan dan mendistribusikan seluruh data yang terkoneksi ke dan dari *broker* tersebut. Lebih jelasnya terlihat pada Gambar 2.1 merupakan sistem sederhana dari proses kerja protokol MQTT, dimana *subscriber* hanya akan menerima informasi yang sesuai pada suatu topik tertentu, kemudian dilanjutkan dengan *publisher* yang akan mengirimkan informasi ke suatu topik tertentu itu pula. Jadi selama *subscriber* dan *publisher* saling terhubung dan saling berlangganan maka topik yang sama maka komunikasi akan selalu terjalin (Tantitharanukul, 2016).



Gambar 2. 1 Diagram Sistem suatu protokol MQTT sederhana

(Tantitharanukul, 2016)

2.2.3 NodeMCU (ESP8266)

NodeMCU merupakan sebuah perangkat keras mikrokontroller yang berupa sebuah papan sirkuit yang memiliki basis utama ESP8266, yang mana papan mikrokontroller ini memiliki keunggulan koneksitivitas *WI-Fi* yang sudah ternaman atau biasa disebut on board dan memiliki kemampuan penyimpanan dan pengelolahan data yang lebih baik dibandingkan Arduino UNO. Harga papan mikrokontroller ini pun juga cukup terjangkau dengan memiliki daya yang cukup rendah serta spesifikasi yang komplit, sehingga alat ini sangat ramah lingkungan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

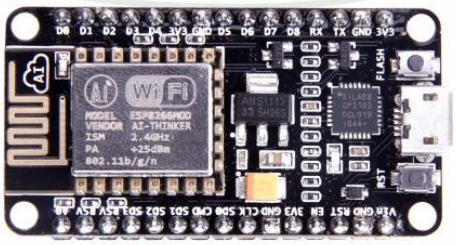
NodeMCU yang memiliki basis utama ESP8266 dirancang untuk mendukung perangkat yang memiliki fungsi secara mobile, perangkat *IoT* dan perangkat yang mudah digunakan lainnya serta perangkat yang memiliki daya rendah. Perangkat keras ini biasanya digunakan untuk perangkat-perangkat cerdas terbaru yang mampu membantu meringankan kinerja dari user penggunanya.

Kenggulan NodeMCU yang lain adalah banyaknya banyak bahasa pemrograman yang didukung, diantaranya adalah AT Command dan Lua yang meruopakan bahasa resmi yang didukung oleh *Node*MCU ini, Sealin itu NodeMCU ini juga dapat ditambhakan berbagia macam jenis *library* sehingga memungkinkan untuk dapat dilakukan pengkodingan dengan menggunakan bahas C++ melalui aplikasi editor Arduino IDE. Tabel 2.1 merupakan spesifikasi *Node*MCU sedangkan Gambar 2.2 adalah gambar Papan Sisrkuit *Node*MCU.

Tabel 2. 1 Spesifikasi *Node*MCU

Microprocessor	ESP8266
Certificate	FCC/CE/TELEC/SRRC
Wi-Fi Protocol	802.11 b/g/n
Wi-Fi Security	WPA/WPA2
Wi-Fi Encryption	WEP/TKIP/AES
Frequency Range	2.4 – 2.5G
Peripheral	UART, SDIO, SPI, I2C, I2S, Remote Control, GPIO, PWM
Operating Voltage	3.0 – 3.6V
Operating Current	80mA
Operating Temperature	-40 – 125 ^{O}C
Network Protocol	IPv4, TCP/UDP, HTTP, FTP

(Espressif, 2016)



Gambar 2.1 Papan Sirkuit NodeMCU

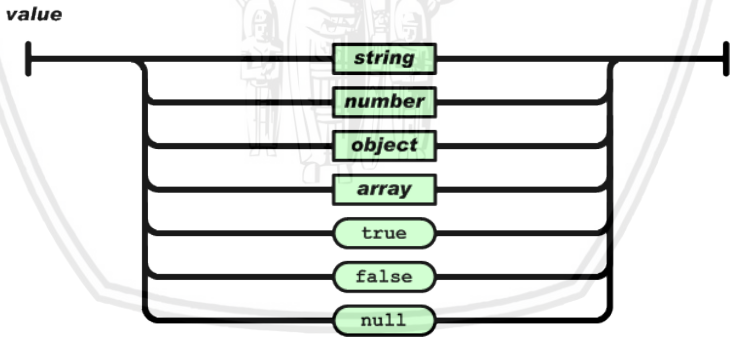
(Seedstudio, 2017)

2.2.4 JSON (*JavaScript Object Notation*)

JSON merupakan format teks yang dapat memfasilitasi pertukaran antar data yang terstruktur diantara semua bahasa pemorgraman (Ecma, 2013). JSON

merupakan sebuah format teks yang memiliki ukuran yang lumayan ringan, sehingga mudah untuk dibaca, ditulis dan dipahami oleh manusia serta mampu dengan mudah diterjemahkan oleh sebuah sistem komputer. JSON mampu didukung oleh hampir seluruh bahasa pemrogaman karena JSON ini sendiri adalah sebuah format teks yang tidak tergantung oleh bahasa pemrogaman apapun. 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 JSON sendiri terdapat salah satu bentuk pada format teks yakni *Object*. Pada format *JSON Object* ini tidak memiliki sepasang nama yang tidak berurutan selain itu format ini akan diawali dengan tanda { (kurung kurawal buka) dan diakhiri dengan tanda } (kurung kurawal tutup), serta setiap nama yang dideskripsikan akan diikuti dengan tanda : (titik dua) dan setiap pasangan nama dan nilai akan dipisahkan dengan tanda , (koma). Adapun bentuk lainnya yaitu *JSON Array*, dimana format ini berbeda dengan Object tadi dimana memiliki nilai yang saling berurutan dan kemudian diawali dengan index 0 (nol) dimana pada *JSON Array* ini akan dimulai dengan tanda [(kurung kotak terbuka) dan diakhiri dengan tanda] (kurung kotak tertutup). Dari kedua bentuk ini, setiap pasangan nilai atau namanya akan dipisahkan oleh tanda koma. Selain itu nilai yang dituliskan juga bisa seperti data *Boelean, String, Null,* Angka ataupun sebuah *Array* yang ditempatkan secara disusun bertingkat maupun sebuah *Object.* Pada Gambar 2.3 menunjukan nilai sintaks pada diagram JSON.



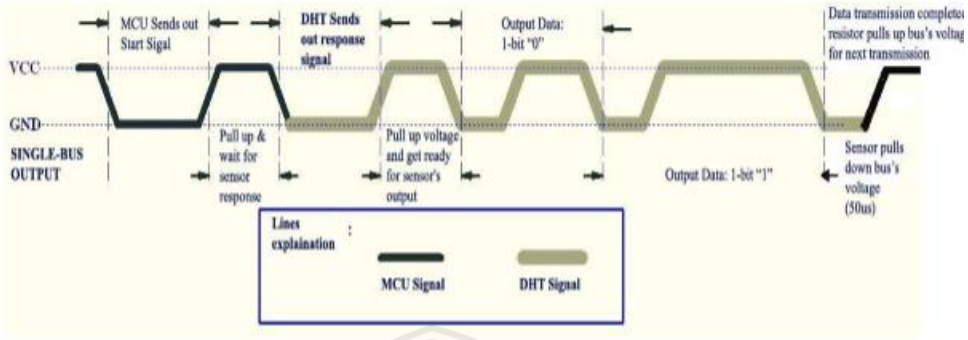
Gambar 2. 2 Nilai Diagram Sintaks pada JSON

(ECMA, 2013)

2.2.5 Sensor *DHT11*

Sensor *DHT11* seperti pada Gambar 2.1 merupakan sensor yang dapat digunakan untuk mengukur 2 parameter secara bersamaan yaitu parametr suhu dan kelembaban. Sensor ini memiliki *thermistor* tipe *NTC* (*Negative Temperature Coefficient*) yang berfungsi sebagai pengukur suhu dan sebuah sensor untuk mendeteksi kelembaban bertipe resisitif dan mikrokontroler *8-bit* yang berfungsi untuk mengolah data dari kedua sensor tersebut dan mengirimkan hasil tersebut

menuju *pin output* yang memiliki format *singe-wire bi-directional* (kabel tunggal dua arah).



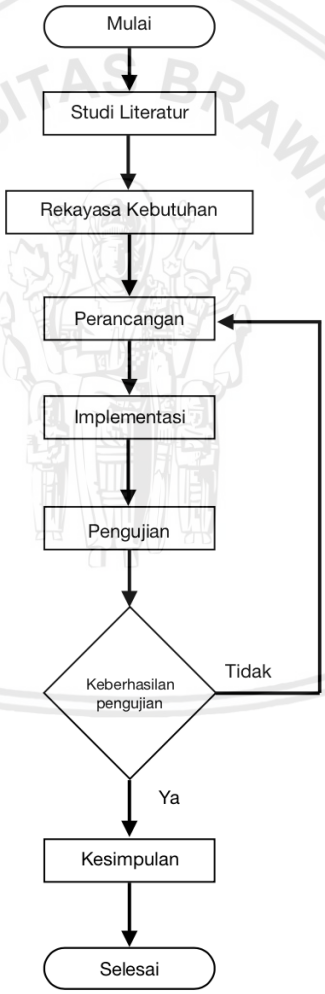
Gambar 2.3 Proses Komunikasi Sensor *DHT11*

(OSEPP Electronics, 2013)

Ketika *MCU* mengirim sinyal awal, *DHT11* berubah dari mode konsumsi daya rendah ke mode berjalan, kemudian *DHT11* menunggu *MCU* menyelesaikan sinyal awal. Setelah selesai, *DHT11* mengirimkan sinyal respon data 40-*bit* yang menyertakan informasi kelembaban dan suhu relatif ke *MCU*. Tanpa sinyal awal dari *MCU*, *DHT11* tidak akan memberikan sinyal respon kepada *MCU*. Setelah data dikumpulkan, *DHT11* akan berubah ke mode konsumsi daya rendah sampai menerima sinyal dari MCU lagi.

BAB 3 METODOLOGI

Pada bab kali ini akan dijelaskan tentang langkah-langkah atau beberapa metodologi yang akan mendukung pngerjaan penelitian ini. Penelitaian ini memiliki sifat implementatif yang berarti menerapkan suatu sistem layanan monitoring yang mengunakan perangkat sensor dan memiliki arsitektur *publish and subscribe* yang mana menggunkan protokol mqtt untuk saling berkomunikasi pada objek kandang ular piton regius. Untuk lebih jelasnya maka terlihat pada gambar 3.1 merupakan flowchart yang berisikan alur demi alur yang dilakukan dalam pembuatan system monitoring ini yang dimulai dengan studi kepustakaan, analisis kebutuhan, perancangan, implementasi, pengujian dan analisis hingga pengambilan sebuah keputusan.



Gambar 3. 1 Diagram alir metode penelitian

3.1 Studi Kepustakaan

Studi kepustakaan itu sendiri adalah sebuah pencarian sebuah kepustakkan yang berguna untuk melakukan pembuatan teori dasar serta refrensi yang akan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

mendukung penelitian kali ini. Proses studi kepustakaan kali ini dilkukan dengan memahami terhadap suatu tinjauan pustaka yang dapat kita peroleh dari sebuah jurnal, skripsi, buku, dan bahkan sebuah websait yang masih berhungan.adapun tinjauan pustaka dan dasar teori itu sendir meliputi :

- Arsitektur *Publish and Subscribe*
  Merupakan alah satu arsitektur terbaru yang saat ini banyak diminati dan dikembangkan untuk kebutuhan perangkat mobile maupun perangkat cerdas. Hal ini terjadi karena arsitektur ini memungkinkan untuk saling berkomunikasi antara banyak perangkat, serta mampu mempercepat kinerja suatu perangkat itu sendiri.
- MQTT
  Merupakan sebuah protokol komunikasi yang mampu mendukung kinerja sebuah perangkat *Internet of Things* dan didesain untuk kebutuhan komunikasi pada perangkat yang menggunakan arsitektur *publish and subscribe*.
- *Node*MCU
  Adalah sebuah mikrokontroler yang telah memiliki koneksitivitas *wireless* tersendiri atau biasa disebut dengan onboard untuk menggunakan sebuah perangkat *Wi-Fi* sehingga dapat digunakan untuk melakukan kegiatan *mempublish* data sensor yang digunakan pada sistem monitoring serta mampu bekerja lebih cepat dan efisien dibandingkan dengan Arduini UNO.
- JSON (*JavaScript Object Notation*)
  Merupakan sebuah format teks juga telah didukung hampir semua bahasa pemrograman. Dimana format teks ini sangat sering dipakai untuk melakukan kegiatan pertukaran data dan informasi baik di sisi perangkat maupun disisi *server.*

3.2 Analisa Kebutuhan

Analisis kebutuhan ini sendiri ditujukan agar penulis mampu menganalisa dan mengetahui kebutuhan apa saja yang akan dibutuhkan dan mendukung dalam proses penelitian ini. Adapun analisa kebutuhan itu sendiri diuraikan sebagai berikut:

3.2.1 Kebutuhan Perangkat Keras

1. Laptop
2. NodeMCU
3. Kabel Jumper
4. Sensor DHT11
5. RTC DS3231
6. LCD 20x4

3.2.2 Kebutuhan Perangkat Lunak

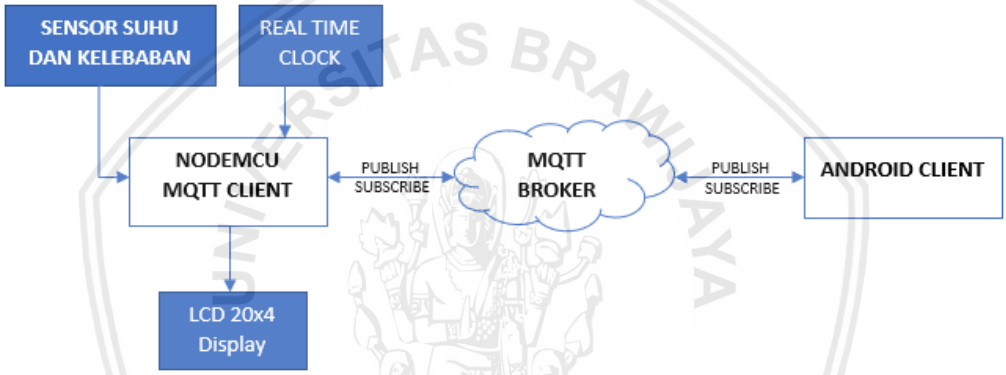
1. Arduino IDE

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2. Python
3. ArduinoJson *Library*
4. Mqtt *Library*
5. Mosquitto MQTT *Broker*
6. Android Studio

3.3 Perancangan Sistem

Untuk perancangan sistem itu sendiri dilakukan setelah melalui beberapa tahap yang telah terpenuhi terlebih dahulu yakni tahap analisis kebutuhan pada perangkat keras maupun perangkat lunak. Selain itu tujuan dari perancangan ini sendiri diharapakan agar proses penerapan sistem monitoring itu sendiri berjalan secara sistematis dan terstruktur. Adapun perancangan sistem itu sendiri digambarkan dalam sebuah pada diagram blok sistem pada Gambar 3.2.



Gambar 3. 2 Blok Diagram Sistem Monitoring

Dari ilustrasi Gambar 3.2 perancangan sistem dapat diuraikan pada langkah-lankah berikut dibawah ini :

1. Masing-masing perangkat keras pendukung akan saling tekoneksi pada papan mikrokontroler NodeMCU yang akan mengakuisisi data dari setiap perangkat keras seperti nilai suhu dan kelembaban dari sensor DHT11 kemudian nilai waktu dari RTC DS3231. Yang kemudian data tersebut akan dikirimkan dengan terbungkus kedalam format teks JSON dan berjalan dengan memanfaatkan aplikasi Mosquitto MQTT *Broker* menggunakan konektivitas *Wi-Fi* dengan menggunakan protokol komunikasi MQTT. Selain itu hasil dari perangkat keras itu juga akan ditampilkan pada layar LCD 20x4. Dimana pada proses ini NodeMCU bekerja sebagai data *publisher.*
2. Setelah data masuk kedalam broker kemudian data akan disusun secara urut berdasarkan topik dan subtopik dari yang telah ditetapkan oleh *publisher.* Kemudian sistem akan menunggu hingga terjadi sebuah koneksi baru dan ada permintaan *subscribe* dari user.
3. Untuk proses yang terkahir ini terjadi pada aplikasi android yang digunakan oleh user. Dimana user akan melakukan *subscribe* berdasarkan topik-topik yang disediakan. Sehingga user dapat dengan mudah mendapatkan informasi tentang suhu dan kelembaban dalam kandang ular tersebut.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3.4 Implementasi

Pada sub bab kali ini akan menjelaskan tentang tahap penimplementasian diamana akan merujuk pada hasil yang telah dirancang dan telah dilakukan. Adapun tahapan implementasi ini sendir memiliki 2 tahap yang mencakup pengimplementasian perangkat lunak maupun perangkat keras.

3.4.1 Implementasi Perangkat Keras

Untuk tahapan ini sendiri, implementasi perangkat keras memiliki beberapa alur tahapan yang meliputi implementasi perangkat keras pada sistem arsitektur publish and subscribe pada alat monitoring suhu dan kelembaban kandang ular python regius menggunakan nodemcu. Setelah itu keseluruhan tahapan harus dilakukan dan dibentuk agar dapat terhubung antar satu sama lain baik sensor dengan perangkat keras lainnya. Sehingga perangkat keras dapat berjalan sesuai kebutuhan sistem.

3.4.2 Implementasi Perangkat Lunak

Untuk taha implementasi perangkat lunak terdapat beberapa implementasi perangkat lunak pada arsitektur *publish and subscribe,* kemudian protokol MQTT dan tentunya apalikasi android. Berikut uraiannya.

1. Implementasi Arsitektur *publish and susbscribe*

   Pada penelitian ini dibuatlah sebuah arsitektur utama yang digunakan untuk melkaukan komunikasi yakni arsitektur *publish and subscribe.* Dimana pada arsitektur ini akan menjadi sistem utama untuk melakukan penyimpanan dan pengambilan data.

2. Implementasi *MQTT*

   Implementasi ini digunakan sebagai protokol utama komunikasi pada sistem. Sehingga arsitektur *publish and subscribe* dapat berjalan dengan baik. Implementasi ini meliputi implementasi MQTT pada perangkat keras maupun perangkat lunak pada sistem.

3. Implementasi aplikasi android

   Pada penelitian ini akan dibuatkan sebuah antarmuka yang berbasis android untuk melakukan pengamatan data sushu dan kelebaban yang diberikan oleh sistem. Aplikasi ini nantinya diharapkan mampu mempermudah peternak untuk melakukan pengamatan secara berkala pada kandang ular piton regius.

3.5 Pengujian dan Analisis

Pada tahap pengujian dan analisis ini akan dilakukan guna mengetahui kebenran hasil yang diberikan oleh sistem setelah proses pengimplementasian yang akan dibandingkan dengan spesifikasi sistem pada saat masih tahap perancangan. Adapun proses pengujian yang dilakukan yakni sebagai berikut:

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1. Pertama adalah pengujian data sensor *DHT11* dengan cara membandingkan menggunakan termometer dan higrometer untuk mengetahui niali kesusaian dan akurasi sensor tersebut.
2. Pengujian Aplikasi android apakah mampu melakukan proses subscribe data untuk melakukan monitoring suhu dan kelembaban.
3. Kemudian dilakukan pengujian sistem secara menyeluruh, yakni meliputi pengujian perangkat lunak yang didalammnya sudah mencakup arsitektur *publish and subscribe*, maupun perangkat keras pendukung lainnya.
4. Pengujian apakah kondisi kandang ular mengalami perubahan setelah sistem diimplementasikan pada kandang ular tersebut.

Setelah selesai melakukan beberapa tahap pengujian diatas kemudian akan dilakukan tahap analisis untuk mengetahui tingkat kesesuaian dan tingkat akurasi dengan perancangan dan spesifikasi yang telah dibuat sebelumnya.

3.6 Kesimpulan

Untuk proses pengambilan kesimpulan itu sendiri dilakukan setelah semua proses dan tahapan sudah berjalan dengan baik yang dimulai dari tahap perancangan, implementasi, hingga pengujian dan analisis. Kesimpulan itu disusun berdasarkan nilai hasil daripada pengujian yang telah dilakukan dan analisis sistem yang dibuat berdasarkan hasil pengujian tersebut. Sehingga nantinya isi dari kesimpulan ini diharapkan mampu menjadi acuan pada penelitian lain untuk mengembangkan sistem monitoring kandang ular piton regius. Kesimpulan dan saran diharapkan dapat dimanfaatkan semaksimal mungkin sebagai bahan acuan dalam pengembangan sistem monitoring kandang ular piton yang lebih lanjut sehingga penelitian ini tidak akan berhenti ketika penelitian ini selesai dilakukan saja.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini memberikan penjelasan mengenal persyaratan-persyaratan yang harus dipenuhi serta persyaratan lainnya yang bersifat opsional sehingga setiap fungsi dari sistem akan berjalan dengan baik serta sesuai dengan tujuan penelitian ini.

4.1 Gambaran Umum Sistem

Pada alat monitoring suhu dan kelembaban ini alan diimplementasikan sebuah arsitektur *publish and subscribe* menggunakan metode mqtt sebagai protokol dalam pengerjaannya. Selain itu alat ini menggunakan nodemcu ESP8266 sebagai mikrocontroller yang akan melakukan publish data kepada broker yang kemedian akan disubscribe oleh pengguna melalui aplikasi android. User akan melakukan permintaan *subscribe* berdasarkan topik yang telah ditentukan. Alat ini diharapkan mampu membantu mempermudah pengawasan suhu dan kelembaban pada perawatan ular regius python yang akan mengurangi resiko ular tersebut terhadap penyakit.

4.1.1 Prespektif Sistem

Sistem dapat dikatakan berhasil apabila sistem dapat menampilan data suhu, dan kelembaban pada aplikasi android dengan seakurat mungkin. Selain itu sistem ini harus dapat melakukan proses komunikasi dengan baik menggunkan metode mqtt dengan nilai loss serta delay yang seminimal mungkin.

4.1.2 Ruang Lingkup

Sistem ini menggunakan aplikasi android untuk memonitoring kondisi kandang. Sebagai parameter nilai dari kelembaban sistem ini akan diberikan nilai acak yang akan ditampilkan sebagai nilai dar suhu dan kelembaban itu sendiri. Sistem ini dikhususkan untuk memonitoring suhu dan kelembaban pada kandang ular trgiud python.

4.1.3 Karakterisitik Pengguna

Karakteristik pengguna diperuntukan pada pemelihara ular regius python, yang bertujuan agar sistem dapat membantu pemelihara ular regius python dalam meminimalisir kemungkinan terkena penyakit yang menjadi kendala utama dalam pemliharaan ular regius python dan dapat memonitoring kandang jarak jauh dengan menggunakan aplikasi android.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.1.4 Lingkungan Operasi Sistem

Adapun persyaratan lingkungan oprasi sistem yang mendukung kebutuhan sistem adalah sebagai berikut:

1. Kondisi suhu lingkungan kandang yang dimonitoring yaitu 20°-31° C atau sesuai dengan nilai di dapat dari sensor suhu DHT11.
2. Kondisi kelembaban lingkungan kandang yang dimonitoring adalah mulai dari 60% - 90% atau sesuai dengan nilai di dapat dari sensor suhu DHT11.
3. Pada arsitektur ini sendiri akan menggunakan protokol komunikasi MQTT dimana untuk QOSnya sendiri digunakan dengan QOS0.

4.1.5 Batasan Perancangan Dan Implementasi

Pada tahap ini bertujuan memberikan batasan pada sistem supaya penelitian bisa terarah dan berjalan sesuai yang diinginkan, adapaun batasan masalah sebagai berikut :

1. Sistem monitoring kandang untuk mencegah kemungkinan terkena penyakit pada ular regius python menggunakan data yang didapat dari sensor DHT11 yang telah dideklarasikan terlebih dahulu.
2. Sistem monitoring kandang ini dapat menampilkan pembacaan suhu dan kelembaban berupa tampilan pada aplikasi android sebagai media monitoring.

Output:

Outpun metode ini akan menggunakan mqtt dimana hasil analisanya dapat dinilai melalui qos 0.

Beberapa asumsi dan ketergantungan persyaratan sistem sebagai berikut:

1. Data suhu dan kelembaban yang telah diambil dari sensor dapat bekerja dengan baik serta mampu dipublish sengan sempurna pada broker.
2. Untuk dapat memonitoring suhu dan kelembaban berdasarkan data dari sensor yang telah didklarasikan yaitu melalui aplikasi android yang mampu melakukan subscribe berdasarkan topik yang telah ditentukan.
3. Arsitektur *publish and subscribe* ini dapat bekerja dengan baik pada metode mqtt dan dapat menampilkan data dengan semaksimal mungkin.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.2 Rekayasa Kebutuhan

Pada sub bab ini menjelaskan seluruh kebutuhan dari sistem supaya dapat bekerja sesuai dengan tujuan dari penelitian ini. Pada sub bab ini menjelaskan kebutuhan antarmuka sistem, kebutuhan komunikasi, kebutuhan fungsional sistem dan kebutuhan lainnya. Berikut penjelasan dari rekayasa kebutuhan sistem.

4.2.1 Kebutuhan Antarmuka

4.2.1.1 Antarmuka Pengguna

Antarmuka pengguna merupakan kebutuhan yang bertujuan untuk menghubungkan interaksi antara sistem dengan penggunanya. Kebutuhan antarmuka yaitu penggunaan aplikasi android untuk menampilan informasi suhu dan kelembaban berdasarkan waktu secara realtime.

4.2.1.2 Antarmuka Perangkat Keras

Kebutuhan antarmuka perangkat keras sistem terdiri dari beberapa komponen yaitu:

1. NodeMCU

   Pada penlitian menggunakan *NodeMCU* dikarekan *mikrokontroler* ini sudah dilengkapi dengan modul *wi-fi* yang akan digunakan untuk pengiriman data agar pengguna tetap dapat memonitoring keadaan kandang walaupun tidak berada di lingkungan kandang.

2. Sensor DHT11

   Sensor *DHT11* dapat mendeteksi suhu dan kelembaban udara pada 1 modul dimana *output* yang dihasilkan berupa sinyal digital yang sudah terkalibrasi.

3. RTC DS3231

   Modul RTC DS3231 ini sendiri merupakan suatu modul yang dberfungsi sebagi sebuah *Real Time Clock.* RTC ini digunakan oleh peneliti untuk memberikan nilai waktu secara real time. Yang akan dipublish juga datanya pada mqtt broker.

4. LCD 20x4

   Digunakan unruk menampilkan data yang dipublish oleh sistem kepada mqtt broker. Selain itu dengan adanya LCD ini juga dapat memudahkan client mengamati data suhu dan kelmbaban pada sistem ini.

5. Smartphone android

   Digunakan untuk menjalankan aplikasi android yang akan memudahkan client untuk mengawasi atau mengetahui nilai suhu dan kelembaban dari kejauhan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.2.1.3 Antarmuka Perangkat Lunak

Kebutuhan antarmuka perangkat lunak sistem terdiri dari beberapa perangkat yaitu :

1. Arduino IDE

   Pada penelitian ini *Arduino IDE* digunakan untuk memprogram *NodeMCU* yang menggunakan bahasa pemrograman *C/C++*.

2. Andoid studio

   Pada penelitian kali ini digunakan Android studio untuk melakukan pemrogaman untuk membuat aplikasi android yang nantinya akan mempermudah klien melihat hasil penelitian.

3. Broker.shiftr.io

   Merupakan broker tempagt data yang akan tersimpan. Data dari sensor akan dipublish ke broker ini yang kemudin akan ditunggu untuk di subscribe oleh pengguna sistem.

4.2.2 Kebutuhan Fungsional

Pada kebutuhan fungsional pada sistem ini yaitu menjelaskan tentang apa saja yang diperlukan agar dapat berjalan sesuai yang diharapkan. Berikut adalah beberapa sistem yang harus ada penelitian ini :

4.2.2.1 Fungsi Melakukan koneksi terhadap jaringan internet

Fungsi ini mengharuskan sistem dapat melakukan sebuah koneksi terhadap jaringan internet yang telah tersedia. Karena padda penelitian kali ini saya akan menggunkan broker yang tersedia di internet secara gratis. Oleh sebab itu fungsi ini sangat penting untuk memuai sebuah sistem yang akan diterapkan.

4.2.2.2 Fungsi melakukan koneksi terhadap broker

Fungsi ini merupakan tujuan dari sistem yang harus dicapai dimana sistem harus dapat melakukan koneksi dengan broker guna melakukan publish data random yang telah dideklarasikan terlebih dahulu.

4.2.2.3 Fungsi publish data

Fungsi ini meerupakan fungsi yang dilakukan untuk melakukan publish data kepada broker. Setelah malakukan koneksi kepada broker data kemudian akan di publish dan akan menunggu untuk di subscribe oleh user.

4.2.1.1 Fungsi menampilkan data kandang

Fungsi ini dapat memberikan informasi dari kondisi kandang yang didapatkan dari pembacaan nilai pada setiap sensor pada fungsi ini memungkinkan pengguna memonitoring kondisi kandang dalam jarak jauh melalui aplikasi *Android* yang terhubung langsung pada *realtime database*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.2.3 Kebutuhan non Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang bersifat opsional agar sistem dapat berjalan lebih baik, namun apabila tidak dijalankan sistem tetep berjalan sesuai dengan fungsionalnya. Pada bagian ini menjelaskan terkait dengan kebutuhan keamanan pada sistem ini. Bagaian ini meliputi keamanan dari sisi koneksivitasnya serta sisi komunikasi terhadap *broker*nya.

1. Umum :
   a. Parameter yang dideteksi yaitu berupa suhu dan kelembaban pada udara.
   b. Konsenstrasi pembacaan suhu 0 - 50 ^{o}C dan kelembaban 20-90% RH.
2. Khusus:
   a. Sumber daya harus terhubung terus menerus dengan sistem.
   b. Semua komponen dapat bekerja dengan baik.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

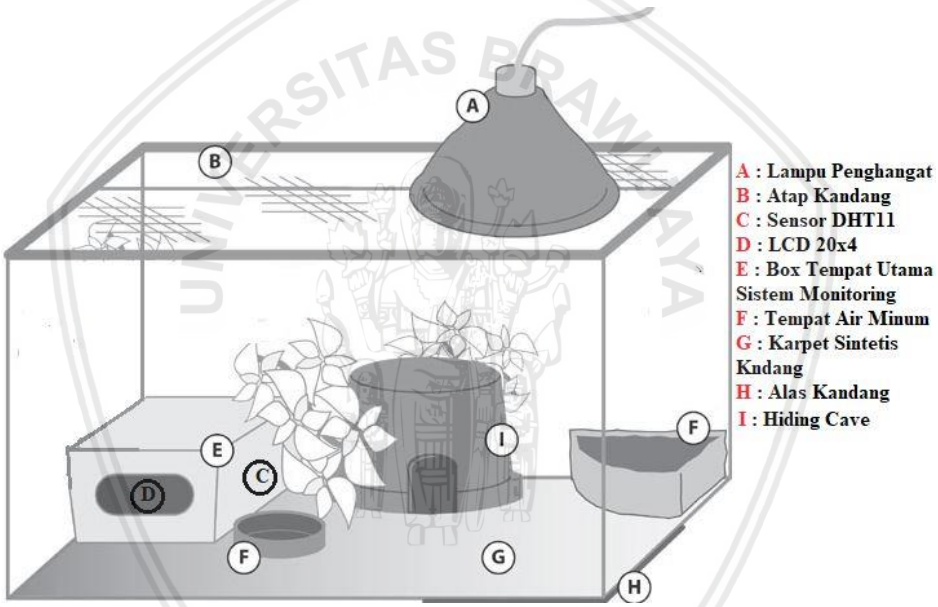
BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Pada sub bab ini peneliti akan menjelaskan perancangan dari sistem yang akan diimplementasikan yang meliputi perancangan perangkat keras dan perancangan perangkat lunak.

5.1.1 Perancangan Perangkat Keras

Pada perancangan skema lingkungan yang 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 5.1 berikut.



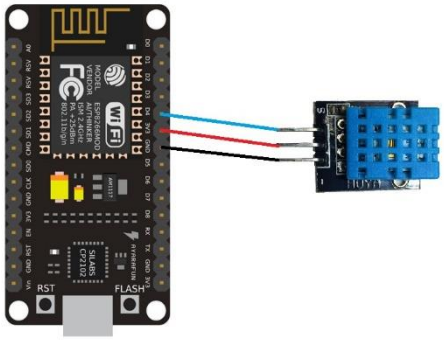
Gambar 5. 1 Peletakan Sistem Monitoring pad Kandang Ular

Pada gambar 5.1 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 agar dapat membaca data.

Perancangan perangkat keras ini juga akan mencakup penjelasan melalui gambaran skematik dari sistem yang akan penulis buat mulai dari perancangan skematik pada sensor DHT11, LCD, dan RTC DS3231 yang akan penulis terangkan seperti berikut.

Perancangan sesnsor dengan DHT11 dengan nodemcu. Sistem ini akan menjelaskan bagaimana proses perancangan skematik dari pada nodemcu dengan sensor DHT11.

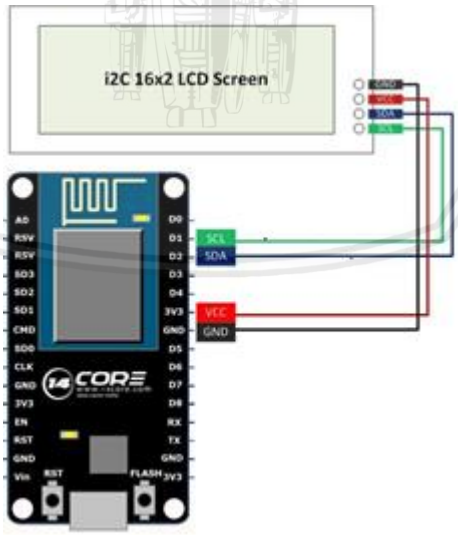
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 5. 2 nodemcu dengan sensor DHT11

Pada gambar 5.2 dijelaskan bahwa yang penulis gunakan adalah tipe sensor DHT11 dengan pin yang berjumlah 3. Yang nantinya 2 pin pertama akan dihubungkan pada VCC dan GND pada nodemcu yang berguna untuk memberikan tegangan serta hambtan pada sensor DHT11 sehingga dapat bekerja dengan baik. Lalu 1 pin lagi yang tersisa akan dihubungkan pada pin D4 pada nodemcu. Pin ini yang digunakan oleh sensor DHT11 untuk mengirimkan data suhu dan kelembaban pada sistem.

Kemudian yang dijelaskan adalah perancngan LCD display dengan nodemcu. LCD ini akan menggunakan modul i2c untuk meminimalkan penggunaan pin pada nodemcu itu sendiri. Sehingga output dari LCD display sendiri hanya menjadi 4 pin saja. Nantinya LCD ini akan menampilkan nilai dari *temperature, humadity,* dan *time sent.* Berikut alur skematik perancangan dari LCD tersebut.



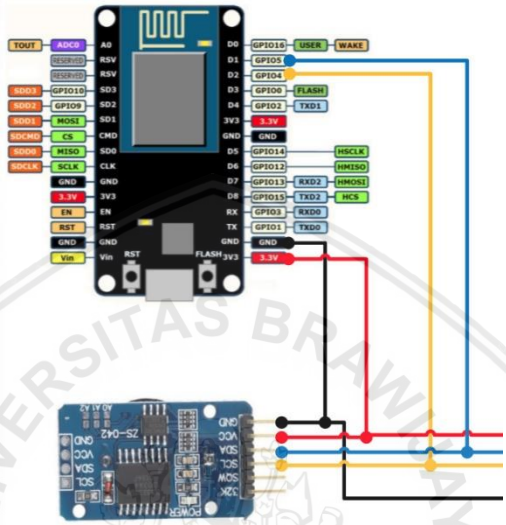
Gambar 5. 3 LCD dengan i2c pada nodemcu

Pada gambar 5.3 memperlihatkan penggunaan pin dari LCd dengan modul i2c dengan sistem nodemcu. Terlihat bahwa penggunaannya meliputi 4 pin dengan rincian 2 pin dihubungkan dengan VCC dan GND pada nodemcu yang digunakan untuk memberikan tengangan dan hambatan pada LCD sehingga mampu menyala dan bekerja dengan baik. Kemudian 2 pin berikutnya akan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

dihubungkan dengan pin D1 dan D2 pada nodemcu. Dua pin ini yang akan dihungkan dengan SDA dan SCL sebgai inputan analog dari LCD sehingga LCD mampu menampilkan data output yang diberikan oleh sistem.

Kemudian selanjutnya adalah perancangan RTC DS3231 dengan nodemcu. Perancangan itu sendiri meliputi penempatan pin pin yang akan digunakan pada nodemcu terhadap RTC itu sendiri. Sehingga rtc dapat berjalan dengan baik.

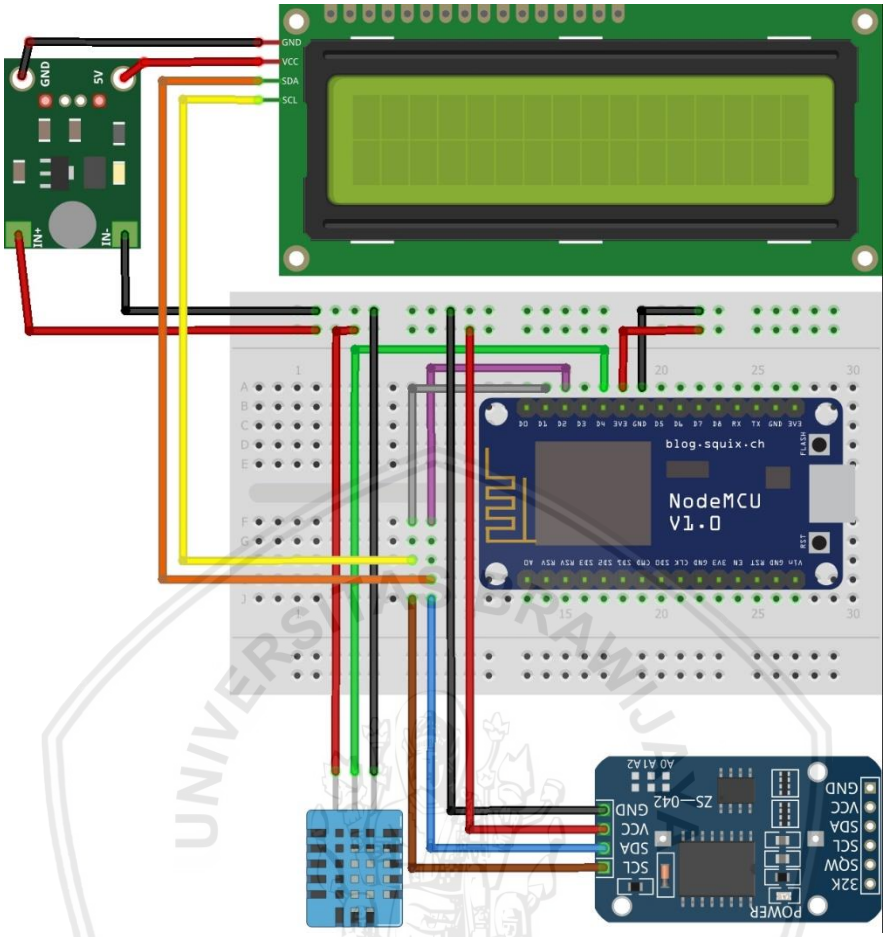


Gambar 5. 4 RTC DS3231 dengan nodemcu

Pada gambar 5.4 terlihat bahwa penggunaan pin dari RTC ini meliputi 4 pin yakni VCC, GND, SDA, dan SCL atau bisa dibilang sama persis dengan LCD display. Hal ini terjadi karena RTC DS3231 ini sendiri memiliki modul i2c didalamnya sehingga untuk memberikan nilai waktu pada sistem memerlukan 4 pin yang sama dengan yang digunakan oleh LCD.

Untuk lebih jelasnya berikut penulis akan menujukkan perancangan keseluruhan dari sistem yang penulis buat.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 5. 5 Rancangan Skematik Keseluruhan sistem

Pada gambar 5.5 menunjukan rancangan skematik secara keseluruha dari sistem. Terdapat satu tambahan modul yang digunakan yakni step up. Modul ini digunakan oleh penulis untuk meningkatkan tegangan dari nodemcu dengan LCD sehingga LCD dapat menyala dengan sempurna. Hal ini dikarenakan nodemcu hanya memiliki nilai output sebesar 3V sedangkan untuk menyalakan LCD dengan sempurna dibutuhkan tegangan sebesar 5V. Selain itu pada gambar diatas juga terlihat bahwa ada 2 modul yang dipasang secara parrarel yakni LCD dan RTC DS3231. Hal ini dilakukan penulis karena kebutuhan pin dari kedua modul tersebut sama.

Tabel 5. 1 Konfigurasi Pin *DHT11*

Sensor *DHT11*	*NodeMCU*
+	3V
Data	D4
-	Gnd

Pada tabel 5.1 merupakan konfigurasi pada sensor *DHT11* dengan *NodeMCU,* dimana *DHT11* memiliki 3 buah pin yaitu pin + (VCC) yang terhubung pada pin 3V, pin data terhubung pada pin D4, dan pin – (GND) tehubung pada pin GND pada *NodeMCU.*

Tabel 5. 2 Konfigurasi Pin RTC DS3231

Sensor *RTC DS3231*	*NodeMCU*
VCC	*3V*
SDA	*D2*
SCL	*D3*
GND	*GND*

Pada tabel 5.2 menunjukkan konfigurasi RTC DS3231 dengan *NodeMCU.* RTC DS3231 memiliki 4 pin yaitu pin VCC yang terhubung pada pin 3V, pin SDA yang mana dihubungkan dengan pin D2 pada *NodeMCU*, pin SCL yang terhubung dengan pin D3, dan pin GND yang terhubung pada pin GND.

Tabel 5. 3 Konfigurasi Pin LCD 20x4 dengan I2C

Sensor *LCD*	*NodeMCU*
VCC	*3V*
SDA	*D2*
SCL	*D3*
GND	*GND*

Pada tabel 5.2 menunjukkan konfigurasi LCD 20x4 mengunakan I2C dengan *NodeMCU.* LCD 20x4 mengunakan I2C memiliki 4 pin yaitu pin VCC yang terhubung pada pin 3V, pin SDA yang mana dihubungkan dengan pin D2 pada *NodeMCU*, pin SCL yang terhubung dengan pin D3, dan pin GND yang terhubung pada pin GND.

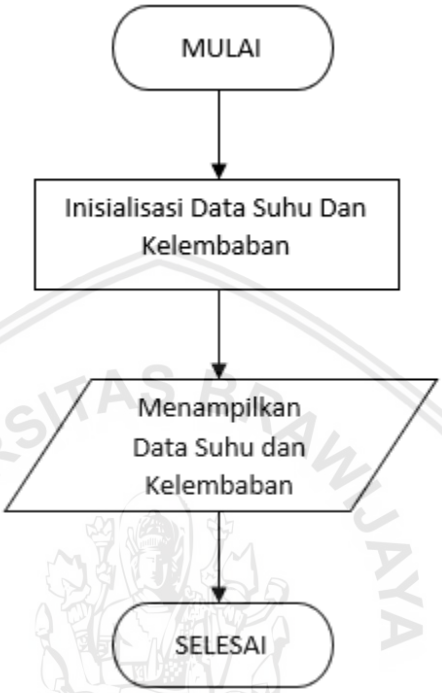
5.1.2 Perancangan Perangkat lunak

Pada sub bab ini penulis akan mejelaskan perancangan dari sistem melalui flowchart.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.1.2.1 Nodemcu – DHT

Perancangan sistem ini akan menjelaskan bagaimana proses kerja dari sensor DHT11 dengan menggunakan sebuah Flowchart. Berikut adalah folwchat dari sistem sensor DHT11.



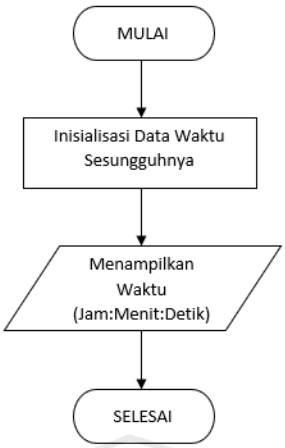
Gambar 5. 6 Flowchart dari sensor DHT11

Pada gambar 5.6 bahwa sensor akan memulai kerja dari sistem itu sendiri yang kemudian akan menunggu perintah untuk mengambil data suhu dan kelembaban. Setelah itu data akan dikirim ke nodemcu dimana akan diproses untuk ditmapilkan pada LCD display. Selain itu data yang dikirim oleh sensor DHT11 itu sendiri akan dipublish ke jaringan broker, sehingga nantinya hasil dari data yang diberikan oleh nodemcu dapat dilihat dengan cara di *subscribe* oleh user dimanapun user berada. Dan kemedian selesai.

5.1.2.2 Nodemcu – RTC

Perancangan itu sendiri meliputi penempatan pin pin yang akan digunakan pada nodemcu terhadap RTC itu sendiri. Sehingga rtc dapat berjalan dengan baik. Selain itu dijelaskan pula alur flowchart dari perancangan RTC.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

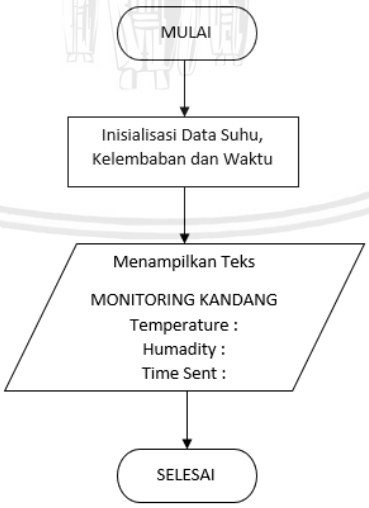


Gambar 5. 7 Flowchat dari sistem RTC DS3231

Pada gambar 5.7 dijelaskan alur kerja dari sistem RTC DS3231 yang digunakan oleh sistem untuk menampilkan waktu dari pengiriman data pada sistem. Data yang diambil dari RTC DS3231 ini sendiri sesuai perintah yang kemudian akan dikirimkan secara terus menerus kepada nodemcu berdasarkan dari waktu yang sesungguhnya. Dan kemudian akan diproses oleh nodemcu untuk meemberilan data waktu pengiriman pada masing-masing data suhu dan kelembaban yang akan dikirim ke broker.

5.1.2.3 Nodemcu – LCD

Nantinya LCD ini akan menampilkan nilai dari *temperature, humadity,* dan *time sent.* Berikut alur skematik perancangan dari LCD tersebut.



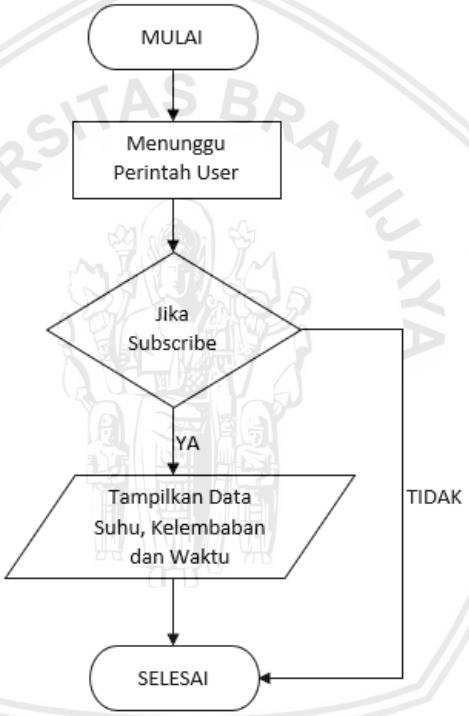
Gambar 5. 8 Flowchart dari sistem LCD

Pada gambar 5.8 dijelaskan bahwa LCD akan menampilkan text *connecting to wifi* ini berarti sistem masih menunggu jaringan internet untuk tersambung jika iya maka akan mucul tulisan *connected to wifi* yang kemudian diteruskan untuk melanjutkan proses sistem itu sendiri yakni memulai koneksi dengan broker

dengan adanya indikasi tulisan *connecting to broker* dan dilanjutkan dengan *connected to broker* yang bersrti sistem sudah terhubung dengan baik dan tinggal menunggu perintah atau inputan data dari sistem itu sendiri. Dan apabila data sudah diberikan oleh sistem maka LCd akan menampilkan data tersebut yang berisikan *temperature, humadity*, dan *time sent*. Hal ini akan mempermudah monitoring sistem dari dekat. Sealin nantinya akan dapat dimonitoring dari kejauhan dari aplikasi android.

5.1.2.4 Android

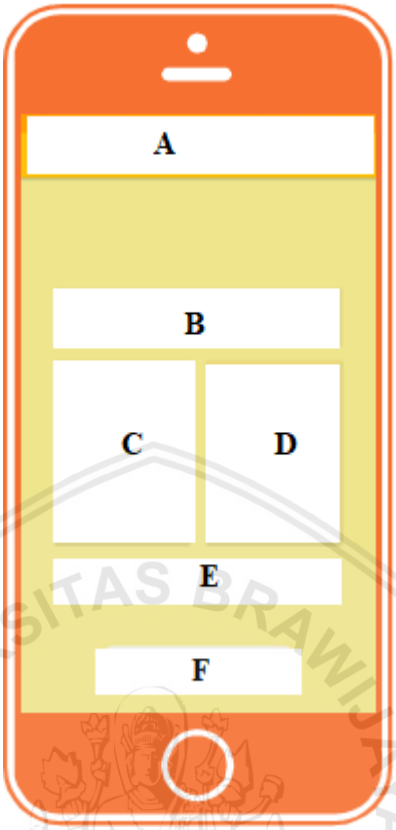
Pada perancangan ini dijelaskan tentang alur kerja dari aplikasi android yang digunakan untuk melihat data dengan cara melakukan subscribe sesuai topik pada broker.



Gambar 5. 9 Flowchart dari sistem Android

Pada flowchart diatas dijelaskan bahwa aplikasi android ini digunakan untuk melihat data dasi sistem dengan cara melakukan subscribe berdasarkan topik pada mqtt broker. Proses akan dimulai dengan menunggu perintah user. Dimana akan ditampilkan layar *idle.* Kemudian apabila ditekan tombol subscribe maka akan ditampilkan data Suhu, Kelembaban dan Waktu secara real time.

Selain itu pada perancangan sistem android ini akan dilakukan perancangan tampilan dari aplikasi android itu sendiri.



Gambar 5. 10 Perancangan Tampilan Aplikasi Android

Pada Gambar 5.10 diatas merupakan perancangan yang nantinya akan ditampilkan pada aplikasi android itu sendiri. Mulai dari bagian "A" yaitu judul aplikasi android itu sendiri. Kemudian pada bagian "B" digunakan untuk menampilkan nilai waktu terkirim dan diterima pada sistem. Kemudian pada bagian "C" digunakan untuk menampilkan nilai suhu sednagkan pada bagian "B" untuk menampilkan nilai dari tingkat kelembaban. Kemudian pada bagian "E" akan ditempakan tombol *Subscribe* untuk menjalankan sistem monitoring itu sendiri. Dan yang terakhir adalah bagian "F" akan digunakan untuk menampilkan teks komentar statur koneksi dari user dengan sistem.

5.1.2.5 Arsitektur *Publish and Subscribe*

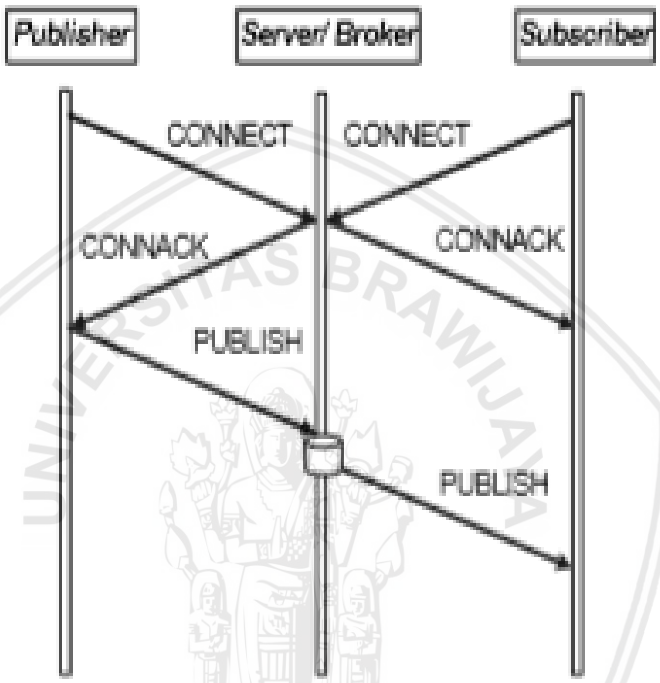
Pada sub bab ini penulis akan menjelaskan tentang proses perancangan daripada arsitektur *publish and subscribe* yang didalamnya berisi perancangan protokol komunikasi MQTT dan juga perancangan format JSON.

Perancangan Protokol MQTT

Pada protokol MQTT ini terdapat suatu *middleware* yakni *broker* yang berfungsi sebagai pengendali pusat untuk mendistribusikan seluruh data yang terkoneksi ke dan dari *broker* tersebut. Pada Gambar 5.14 digambarkan proses perancangan sebuah komunikasi yang melibatkan daripada sistem *publisher* yang digunakan sebagai penyedia informasi. Dalam penelitian ini *publisher* adalah

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

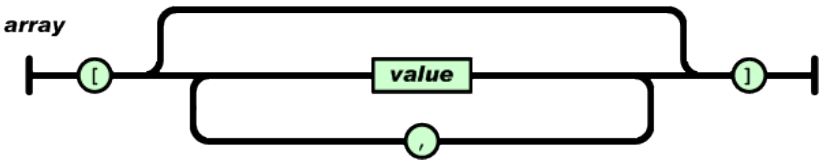
sistem monitoring yang ada pada kandang yang telah dilengkapi oleh sensor suhu. Sedangkan broker yang digunakan sebagai server tempat penyimpanan data yang kemudian akan di minta oleh user dengan cara melakukan sebuah proses *subscribe.* Proses diawali ketika *publisher* dan *subscriber* melakukan komunikasi yang kemudian akan dibalas oleh *broker* dengan *CONNACK* yang berarti keduannya sudah slaing terhunbung dengan *broker* dan siap untuk melakukan proses pertukaran data.



Gambar 5. 11 Perancangan Protokol Komunikasi MQTT

Perancangan JSON

Format teks yang digunakan adalah *JSON Array*, memiliki nilai yang berurutan yang diawali dengan index 0 (nol) dimana pada bentuk ini memiliki karakteristik yang dimulai dengan tanda [(kurung kotak terbuka) dan diakhiri dengan tanda] (kurung kotak tertutup). Pada Gambar 5.15 menunjukkan diagram sintaks untuk tipe *JSON Array*.



Gambar 5. 12 Perancangan Format Teks JSON

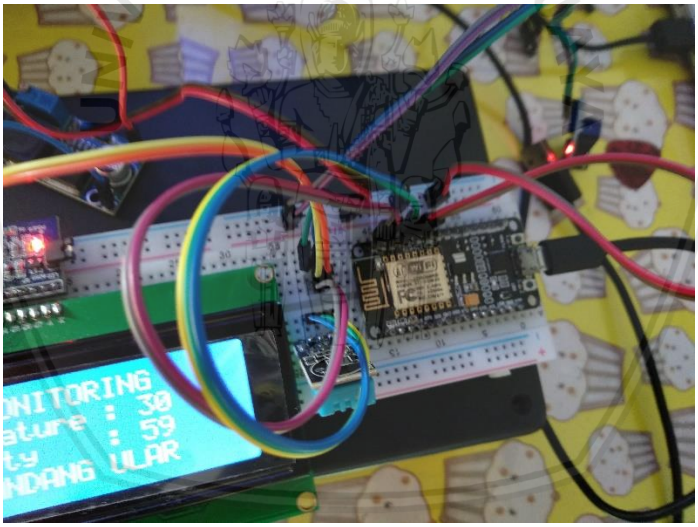
Pada bentuk ini setiap pasangan nama/nilainya dipisahkan oleh tanda "," (koma) dan memiliki nilai yang dapat berupa data *String*, Angka, *Boolean*, *Null* atau sebuah *Array* yang dapat disusun secara bertingkat.

5.2 Implementasi

Implementasi sistem merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan semua perancangan yang telah dilakukan sebelumnya. Pada sub bab ini peneliti akan menjelaskan implementasi dari sistem yang akan diterapkan yang meliputi perancangan perangkat keras dan perancangan perangkat lunak.

5.2.1 Implementasi Perangkat Keras

Pada sub bab ini penulis akan menjelaskan proses implementasi perangkat keras yang akan diterapkan untuk melakukan monitoring kandang ular. Dalam hal ini si ular akan ditempatkan pada sebuah aquarium yang berukuran 60x30x30 dimana didalamnya akan ditempatkan box yang sudah berisi alat monitoring kandang yang dibuat oleh penulis. Berikut hasil foto yang telah dilakukan oleh penulis.



Gambar 5. 13 Implementasi dari sistem

Gambar 5.10 diatas merupakan penerapan perangkat keras seperti LCD, RTC dan sensor DHT11 pada nodemcu. Semua sistem tersebut saling dihungkan dengan menggunakan project board sesuai dari sistem skematik yang telah dirancang sebelumnya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 5. 14 Implentasi sistem pada kandang ular Python Regius

Gambar 5.11 diatas merupakan gambar pengimplementasian sistem pada kandang ular. Sistem akan menampilkan nilai suhu, kelembaban serta waktu pengiriman data. Dengan adanya sistem tersebut diharapkan bahwa sistem akan membantu mempermudah monitoring suhu dan kelembaban dalam kandang ular itu sendiri.

5.2.2 Implementasi Perangkat Lunak

Pada sub bab ini penulis akan menjelaskan proses pengimplementasian perangkat lunak pada sistem yang meliputi penjelelasan kode program pada masing-masing bagian berikut.

5.2.2.1 Nodemcu – DHT

Pada sub bab ini penulis akan mejelaskan proses implementasi yang dari fungsi yang akan digunakan untuk mengimplemtasikan sensor DHT11 pada sistem. Dalam hal ini penulis akan menuliskan potongan dari kode program untuk menjalankan perangkat sensor DHT11 pada nodemcu.

Tabel 5. 4 Potongan kode program sensor DHT11

Baris	Kode program
1	`#include "DHT.h"`
2	`#define DHTTYPE DHT11`
3	`dht.begin(); //`
4	`#define dht_dpin 2`
5	`DHT dht(dht_dpin, DHTTYPE);`
6	`void readSensorData() {`

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
7        temperature = dht.readTemperature();
8        humidity = dht.readHumidity();
9      }
10
11     void publishSensorData(int qos) {
12       if (temperature <= 100 ) {
13         String payload = String(temperature) + ',' +
14     String(humidity) + ',' + timestamp;
15         client.publish("/node/sensor/dht", payload,
16     false, qos);
17
18       }
     }
```

Pada Tabel 5.4 dijelaskan beberapa potongan dari kode program untuk menjalankan modul sensor DHT11 pada sistem nodemcu. Pada baris pertama merupakan deklarasi untuk menggunakan sistem DHT11 kemudian dilanjutkan pada baris ke empat yaitu `dht.begin();` yang berarti sistem dari DHT11 mulai dijalankan. Kemudian pada baris ke8 terdapat fungsi `void readSensorData()` yang digunakan untuk mendeklarasikan 2 variabel yang digunakan dalam sistem yakni temperature dan humadity. Kemudian pada baris ke11 dilanjutkan dengan fungsi `void publishSensorData(int qos)`. Fungsi ini digunakan untuk melakukan proses publish data yang nantinya data akan dipublish kepada broker dengan qos 0 dan kemudian dapat diakses oleh user dengan mealkukan subscribe pada node yang telah dideklarasikan yakni `/node/sensor/dht` sehingga nantinya user akan mendapatkan nilai suhu dan kelembaban dari data yang telah dikirim oleh sensor DHT11. Adapun nilai parameter 100 pada baris ke12 diatas adalah untuk membatasi data yang akan diterima oleh sensor, sehingga apabila nilainnya diatas angka 100 maka sensor tersebut mengalami masalah.

5.2.2.2 Nodemcu – RTC

Pada sub bab ini penulis akan menjelaskan pengimplementasian perangkat RTC yang digunakan sebagai penunjuk waktu pada sistem sehingga memepermudah penulis saat akan melakukan pengujian terhadap sistem. Berikut potongan kode program untuk melakukan implementasi RTC pada nodemcu.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5. 5 Potongan kode program RTC DS3231

Baris	Kode program
1	`#include <RtcDS3231.h>`
2	`RtcDS3231<TwoWire> Rtc(Wire);`
3	
4	`Rtc.Begin();`
5	`RtcDateTime compiled = RtcDateTime(__DATE__,`
6	`__TIME__);`
7	
8	`if (!Rtc.IsDateTimeValid()) {`
9	`Serial.println("RTC lost confidence in the`
10	`DateTime!");`
11	`Rtc.SetDateTime(compiled);`
12	`}`
	`if (!Rtc.GetIsRunning()) {`
13	`Serial.println("RTC was not actively running,`
14	`starting now");`
15	`Rtc.SetIsRunning(true);`
16	`}`
17	`RtcDateTime now = Rtc.GetDateTime();`
18	`if (now < compiled) {`
	`Serial.println("RTC is older than compile time!`
19	`(Updating DateTime)");`
20	`Rtc.SetDateTime(compiled);`
	`}`
	`else if (now > compiled) {`
21	`Serial.println("RTC is newer than compile time.`
22	`(this is expected)");`
	`}`
	`else if (now == compiled) {`
23	`Serial.println("RTC is the same as compile time!`
24	`(not expected but all is fine)");`
	`}`
	`Rtc.Enable32kHzPin(false);`
25	`Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);`
26	

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
27        loopForever = false;
28      }
29
30      void dateTimeFormater(const RtcDateTime& dt) {
31        char datestring[9];
32
33        snprintf_P(datestring,
34                   countof(datestring),
35                   PSTR("%02u:%02u:%02u"),
36                   dt.Hour(),
37                   dt.Minute(),
38                   dt.Second() );
39        timestamp = datestring;
40        Serial.println(datestring);
41        Serial.println();
        }
```

Pada tabel 5.5 dijelaskan potongan dari kode program untuk menjalankan sistem RTC DS3231 pada nodemcu. Dimulai pada baris pertama yaitu deklarasi penggunaan modul RTC DS3231 pada sistem dengan dilanjutkan penjelasan penggunaan pin pada RTC pada baris berikutnya. Kemudian pada baris ke 4 yakni `Rtc.Begin()` yang berarti sistem RTC DS3231 mulai dijalankan. Stelah itu dilanjutkan pada baris ke 5 yang merupakan deklarasi dari sistem untuk menggunnakan RTC sebagai penunjuk waktu yang di dalamnya terdapat beberapa kondisi antara lain. Apabila RTC gagal memberikan nilai yang sebenarnya maka pada serial monitor akan ditampilkan tulisan `RTC lost confidence in the DateTime!`, dan jika RTC dapat berjalan dengan baik maka sistem akan mulai dijalnnkan dengan memanggil variabel `RtcDateTime` sebagai variabel yang digunakan untuk menunjukan waktu yang sesungguhnya. Kemudian dilanjutkan pada baris ke 28 dimana terdapat sebuah fungsi `void dateTimeFormater(const RtcDateTime& dt)` yang digunakan untuk meberikan fortmat penulisan dari waktu yang diberikan oleh RTC pada sistem. Pada fungsi ini digunakan variabel baru yakni `datestring` yang memiliki nilai karakter sebanyak 9. Variabel ini yang nantinya digunakan untuk format penulisan jam yang akan nantinya akan menampilkan jam, menit, dan detik. Ketiganya akan dipisahkan dengan tanda titik dua ":".

5.2.2.3 Nodemcu – LCD

Pada sub bab ini penulis akan menjelaskan pengimplementasian lcd pada sistem nodemcu yang meliputi pengalamatan tata letak penulisan pada lcd tersebut. Berikut adalah potongan kode programnya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5. 6 Potongan kode program LCD

Baris	Kode program
1	`#include <LiquidCrystal_I2C.h>`
2	
3	`LiquidCrystal_I2C lcd(0x3F, 20, 4);`
4	
5	`void lcdSetView(int temperature, int humidity) {`
6	`  if (temperature <= 100 ) {`
7	`  lcd.setCursor(1, 0);`
8	`  lcd.print("MONITORING KANDANG");`
9	
10	`  lcd.setCursor(0, 1);`
11	`  lcd.print("Temperature : ");`
12	`  lcd.setCursor(14, 1);`
13	`  lcd.print(temperature);`
14	
15	`  lcd.setCursor(0, 2);`
16	`  lcd.print("Humidity  : ");`
17	`  lcd.setCursor(14, 2);`
18	`  lcd.print(humidity);`
19	
20	`  lcd.setCursor(0, 3);`
21	`  lcd.print("Time Sent : " + timestamp);`
22	`  }`
23	`}`

Pada tabel 5.6 dijelaskan beberapa potongan kode program dari penggunaan LCD dengan i2c pada nodemcu. Dimulai pada beris pertama yakni `#include <LiquidCrystal_I2C.h>` yang berarti sistem ini menggunakan library LCD dengan menggunakan i2c. Dilanjutkan pada baris ke 3 yakni `LiquidCrystal_I2C lcd(0x3F, 20, 4);` yang merupakan alamat yang digunakan oleh nodemcu untuk melakukan pengalamatan tampilan pada lcd yang berati lcd tersebut memiliki jumlah panjang 20 karakter dan tinggi 4 karakter, sedangkan untuk alamat 0x3F sudah merupakan bawaan dari hardware LCD itu sendiri yang didapat saat pertama kali melakukan LCD test maka akan didapatkan alamt tersebut. Kemudian dilanjutkan pada baris ke 5 terdapat sebuah fungsi `void lcdSetView(int temperature, int humidity)` yang mengatur pengalamatan dari pada teks pada setiap barisnya. Pada fungsi ini pula masih terhubung dengan nilai suhu dan kelembaban yang diberikan oleh sensor DHT11. Kemudian pada baris ke 7 merupakan kode program untuk melakukan set kursor pada LCD baris

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

pertama yang akan muncul teks “MONITORING KANDANG” kemudian pada baris ke 10 merupakan kode program untuk melakukan set kursor pada baris kedua pada LCD yang akan ditampilkan teks “Temperature : “ dan kemudian dikuti nilat suhu yang didapat dari sensor DHT11. Dilanjutkan pada baris ke 15 yang digunakan untuk mealkukan set kursor baris ketiga pada LCD dimana akan ditampilkan teks “Humadity : “ yang kemudian diikuti nilai dari kelembaban yang didapat dari sensor DHT11. Kemudian pada baris ke 20 yang merupakan kode program untuk melakukan set kursor pada baris keempat pada LCD yang nantinya akan ditampilkan teks “Time Sent : “ yang kemudian dilanjutkan dengan nilai waktu yang di dapat dari RTC DS3231.

5.2.2.4 Android

Pada sub bab ini penulis akan sedikit menjelaskan tentang pemrogaman yang dilakukan pada aplikasi android sehingga aplikasi mampu melakukan subscribe data berdasarkan topik yang telah ada pada broker.

Tabel 5. 7 Potongan Kode Program Android

Baris	Kode program
1	`protected void onCreate(Bundle savedInstanceState) {`
2	`super.onCreate(savedInstanceState);`
3	`setContentView(R.layout.activity_home);`
4	`mqttHelper = new MqttHelper();`
5	`qos = 0;`
6	
7	`temperature = (TextView)`
8	`findViewById(R.id.temperature);`
9	`humidity = (TextView)`
10	`findViewById(R.id.humidity);`
11	`timeSent = (TextView)`
12	`findViewById(R.id.timeSent);`
13	`timeReceive = (TextView)`
14	`findViewById(R.id.timeReceive);`
15	`subscribeButton = (CardView)`
16	`findViewById(R.id.subscribeButton);`
17	`unSubscribeButton = (CardView)`
18	`findViewById(R.id.unSubscribeButton);`
19	`qosSpinner = (Spinner)`
20	`findViewById(R.id.qosSpinner);`
21	
	`ArrayAdapter<String> adapter = new`
	`ArrayAdapter<>(this,`
	`android.R.layout.simple_spinner_dropdown_item, items);`

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
22
23              qosSpinner.setAdapter(adapter);
24
25              subscribeButton.setOnClickListener(this);
26              unSubscribeButton.setOnClickListener(this);
27              qosSpinner.setOnItemSelectedListener(this);
28
29              client =
30      mqttHelper.getMqttClient(getApplicationContext(),
31      MQTT_BROKER_URL, CLIENT_ID);
32
33              client.setCallback(new MqttCallbackExtended()
34      {
35                  @Override
36                  public void connectComplete(boolean b,
37      String s) {
38
39                  }
40
41                  @Override
42                  public void connectionLost(Throwable
43      throwable) {
44
45                  }
46
47                  @Override
48                  public void messageArrived(String s,
49      MqttMessage mqttMessage) throws Exception {
50                      DateFormat df = new
51      SimpleDateFormat("HH:mm:ss");
52                      String timeNow =
53      df.format(Calendar.getInstance().getTime());
54                      String[] dataReceive =
55      mqttMessage.toString().split(",");
56                      Log.d(TAG, mqttMessage.toString());
57                      temperature.setText(dataReceive[0] +
58      " °C");
                        humidity.setText(dataReceive[1] + "
      %");
                        timeSent.setText("Time Sent        :
      " + dataReceive[2]);
```

```
                    timeReceive.setText("Time Receive  :
" + timeNow);
                }

                @Override
                public void
deliveryComplete(IMqttDeliveryToken iMqttDeliveryToken)
{

                }
            });
        }
```

Pada tabel 5.7 diatas dijelaskan beberapa potongan dari kode program pada aplikasi android yang digunakan oleh user untuk melakukan subscribe pada broker untuk mengambil data yang telah dipublish. Pada baris ke1 merupakan deklarasi dari fungsi untuk memulai kerja dari kode program itu sendiri. Kemudian dilanjutkan pada baris ke3 yakni `setContentView(R.layout.activity_home);` ini digunakan oleh sistem untuk memanggil fungsi tampilan pada kode program activity_home dimana pada baris ini akan menghasilkan tampilan yang telah ditetapkan terlebih dahulu. Kemudian pada baris ke4 yakni `mqttHelper = new MqttHelper();` digunakan untuk mendeklarasikan funsi mqtt helper yang akan digunakan untuk komunikasi utama pada sistem. Dimana mqtthelper ini akan menghubungkan sistem android dengan broker tempat sistem menyimpan data yang telah di publish. Kemudian pada baris ke 7 hingga 11 dideklarasikan bagian-bagian yang akan ditampilkan pada aplikasi android. Mulai dari temperaturu, kelembaban, waktu pengiriman, hingga tombol subscribe yang akan berubah menjadi unsubsribe ketika selasai ditekan. Kemudian dilanjutkan pada baris ke 43 hingga 56 diamana disana dlakukan pengambilan niali yang didapat dari broker dengan tidak lupa melakukan pencocokan penempatan pada masing-masing data yang didapat ketika dilakukan subscribe.

Pada Gambar 5.16 di bawah ini merupakan tampilan utam dari aplikasi android yang nantinya akan digunakan user untuk menjalankan sistem monitoring kandang ular tersebut. Dimana pada gambar di bawah ini terlihat mulai dari judul aplikasi sudah tampil sesuai dengan perancangan sebelumnya yang telah dibuat. Kemudian ada bagian untuk menampilkan informasi waktu data dikirim dari sistem dan waktu pada saat data diterima oleh aplikasi android. Kemudian terdapat bagian yang akan menampilkan nilai suhu yang terdapat animasi matahari dan data kelembaban yang ditandai dengan animasi awan. Dimana semua data tersebut akan ditampilkan ketika user menekan tombol *"SUBSCRIBE"* pada aplikasi tersebut. Berikut ini adalah hasil dari implementasi sistem aplikasi android yang digunakan pada sistem monitoring kandang ular.

Gambar 5. 15 Tampilan Utama Aplikasi Android

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini membahas proses pengujian dan menganalisa pengujian yang dilakukan berdasarkan sistem yang telah dibuat. Manfaat dilakukannya pengujian adalah untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenuhi oleh sistem. Pada proses pengujian yang dilakukan yakni berupa pengujian fungsional, pengujian akurasi dan pengujian kecepatan pemrosesan sistem, dimana pengujian fungsional yakni menguji fungsi dari perangkat keras dalam hal ini berupa sensor dan rtc yang digunakan serta LCD 20x4 apakah dapat bekerja spesifikasinya, pengujian akurasi yaitu menguji seberapa akurat pada sistem yang telah dirancang dan diimplementasikan dibandingkan nilai atau hasil yang sebenarnya. Berikut dijelaskan beberapa skenario pengujian yang dilakukan untuk menguji sistem.

6.1 Pengujian Sensor *DHT11*

Sensor *DHT11* merupakan sensor yang dapat mendeteksi suhu dan kelembaban. Pada penelitian sensor *DHT11* merupakan komponen utama pada sistem ini untuk melakukan pembacaan nilai suhu dan kelembaban pada kandang.

6.1.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui respon suhu dan kelembaban yang diberikan oleh sensor dan keakuratan sensor dengan alat *HTC-1. HTC-1* yang merupakan sensor suhu dan kelembaban yang sudah dilengkapi dengan monitor dan mikrokontroler sehingga memudahkan penggunaan.

6.1.2 Prosedur Pengujian

1. Menghubungkan sistem otomasi dengan komputer.
2. Upload program NodeMCU
3. Amati pembacaan suhu dan kelembaban pada serial monitor kemudian bandingkan dengan alat *HTC-1*

6.1.3 Hasil dan Analisis Pengujian

Hasil dan pengujian keakurasian sensor *DHT11* dalam mendeteksi suhu ditampilkan pada Tabel 6.1.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 6. 1 Perhitungan Persentase Kesalahan Suhu *DHT11*

No	Pembacaan Suhu pada sensor *DHT11* dan HTC-1		
	DHT11 **(°C)**	**HTC-1 (°C)**	**Presentase Kesalahan (%)**
1	28	29	2,06 %
2	28	29	1,33 %
3	28	28	0 %
4	25	26	0,76 %
5	23	23	1,73 %
6	24	25	6,25 %
7	34	36	2,28 %
8	35	37	1,35 %
9	31	31	1,61 %
10	32	35	2,05 %
	Rata- Rata		1,94 %

Tabel 6.1 merupakan tabel hasil pengujian keakurasian sensor *DHT11* dalam mendeteksi suhu. Presentase akurasi diperoleh dari perhitungan sebagai berikut :

$$\text{Kesalahan } \% = \frac{\text{Nilai Suhu HTC1} - \text{nilai sensor DHT11}}{\text{Nilai Suhu HTC1}} \times 100\%$$

$$\text{Kesalahan } \% = \frac{23{,}4 - 23}{23{,}4} \times 100\%$$

$$\text{Kesalahan } \% = 1{,}73\ \%$$

Setelah diperoleh nilai kesalahan pada masing-masing nilai suhu yang dideteksi akan diperoleh rata-rata dengan perhitungan sebagai berikut :

$$\text{Kesalahan} = \frac{\text{Total kesalahan } \%}{\text{Jumlah data}}$$

$$\text{Kesalahan} = \frac{19{,}42\ \%}{10}$$

$$\text{Kesalahan} = 1{,}94\ \%$$

Setelah diperoleh rata - rata kesalahan, maka diperoleh keakurasian sensor dengan persamaan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

$$\text{Akurasi} = 100\% - \text{rata presentase kesalahan } \%$$

$$\text{Akurasi} = 100\% - 1{,}94\ \%$$

$$\text{Akurasi} = 98{,}06\ \%$$

Sehingga pada penelitian ini akurasi sensor adalah 98,06 % untuk pembacaan suhu.

Hasil dan pengujian keakurasian sensor *DHT11* dalam mendeteksi Kelembaban ditampilkan pada Tabel 6.2

Tabel 6. 2 Perhitungan Persentase Kesalahan Kelembaban *DHT11*

No	Pembacaan Kelembaban pada sensor *DHT11* dan HTC-1		
	***DHT11* (%)**	**HTC-1(%)**	**Persentase 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 %

Tabel 6.1 merupakan tabel hasil pengujian keakurasian sensor *DHT11* dalam mendeteksi suhu. Presentase kesalahan diperoleh dengan perhtungan sebagai berikut :

$$\text{Kesalahan } \% = \frac{\text{Nilai Kelembaban HTC1} - \text{ Nilai DHT11}}{\text{Nilai Kelembaban HTC1}} \times 100\%$$

$$\text{Kesalahan } \% = \frac{52 - 50}{52} \times 100\%$$

$$\text{Kesalahan } \% = 4\ \%$$

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Setelah diperoleh nilai kesalahan pada masing-masing nilai suhu yang dideteksi akan diperoleh rata-rata dengan perhitungan sebagai berikut :

$$\text{Kesalahan} = \frac{\text{Total kesalahan \%}}{\text{Jumlah data}}$$

$$\text{Kesalahan} = \frac{44{,}23\ \%}{10}$$

$$\text{Kesalahan} = 4{,}42\ \%$$

Setelah diperoleh rata - rata kesalahan, maka diperoleh keakurasian sensor dengan persamaan.

$$\text{Akurasi} = 100\% - \text{rata presentase kesalahan \%}$$

$$\text{Akurasi} = 100\% - 4{,}42\ \%$$

$$\text{Akurasi} = 95{,}58\ \%$$

Sehingga pada penelitian ini akurasi sensor adalah 95,58 % untuk pembacaan kelembaban.

6.2 Pengujian Tampilan LCD 20X4

LCD 20x4 merupakan komponen untuk menampilkan karakter pada layar berukuran 20 kolom dan 4 baris. Karakter yang ditampilkan pada laya LCD pada umumnya berdasarkan apa yang telah ditentukan pada mikrokontroler, oleh karena ini pengujian ini dilakukan dengan melihat kesesuaian antara tampilan dengan Program yang ada.

6.2.1 Tujuan Pengujian

Pengujian Tampilan LCD 20x4 ini bertujuan untuk mengetahui apakah hasil pembacaan nilai sensor dan hasil klasifikasi yang diharapkan pada saat ditampilkan pada layar LCD sesuai dengan kode program yang telah dirancang dan diimplentasikan.

6.2.2 Prosedur Pengujian

Untuk melakukan pengujian pada LCD 20x4 yaitu dengan prosedur sebagai berikut :

1. Menghubungkan sistem monitoring dengan Komputer
2. Upload kode program NodeMCU
3. Mengamati tampilan LCD dengan serial Monitor Pada Arduino IDE dengan memberikan beberapa kondisi yang berbeda pada saat pengujian apakah sama atau tidak tampilan pada serial monitor dan LCD pada sistem.

6.2.3 Hasil dan Analisis Pengujian

Hasil dari penelitian LCD akan ditampilkan pada gambar-gambar berikut ini.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 6. 1 Tampilan LCD pada awal Proses

Pada gambar diatas merupakan tampilan awal LCD pada saat pertama kali melakukan proses sistem monitoring. Dimana proses tersebut ditandai dengan proses menyambungkan perangkat pada layanan *Wi-Fi* yang telah tersedia.

Gambar 6. 2 Tampilan Utama LCD saat sistem berjalan

Gambar diatas merupakan tampilan utama dari sistem monitoring yang peneliti buat dimana data *temperature, humidity, time sent* dapat ditampilkan dengan baik dan benar.

Berdasarkan hasil pengujian diatas tampilan pada LCD 20x4 bahwa LCD sesuai dengan yang diharapkan peneliti. Mulai dari proses awal melakukan koneksi hingga menampilkan semua data yang diinginkan oleh penulis.

6.3 Pengujian Kinerja RTC DS3231

RTC DS3231 pada sistem ini digunakan untuk memberikan nilai waktu secara real time pada sistem. Sehingga waktu saat pengiriman data pada broker dapat tercatat.

6.3.1 Tujuan pengujian

Pada pengujian RTC DS3231 ini bertujuan untuk mengetahui aapakah perangkat keras dari RTC DS3231 ini dapat bekerja dengan baik atau tidak. Karena

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

RTC ini adalah komponen yang ditampilkan untuk mengetahui waktu secara *real time* pada sistem.

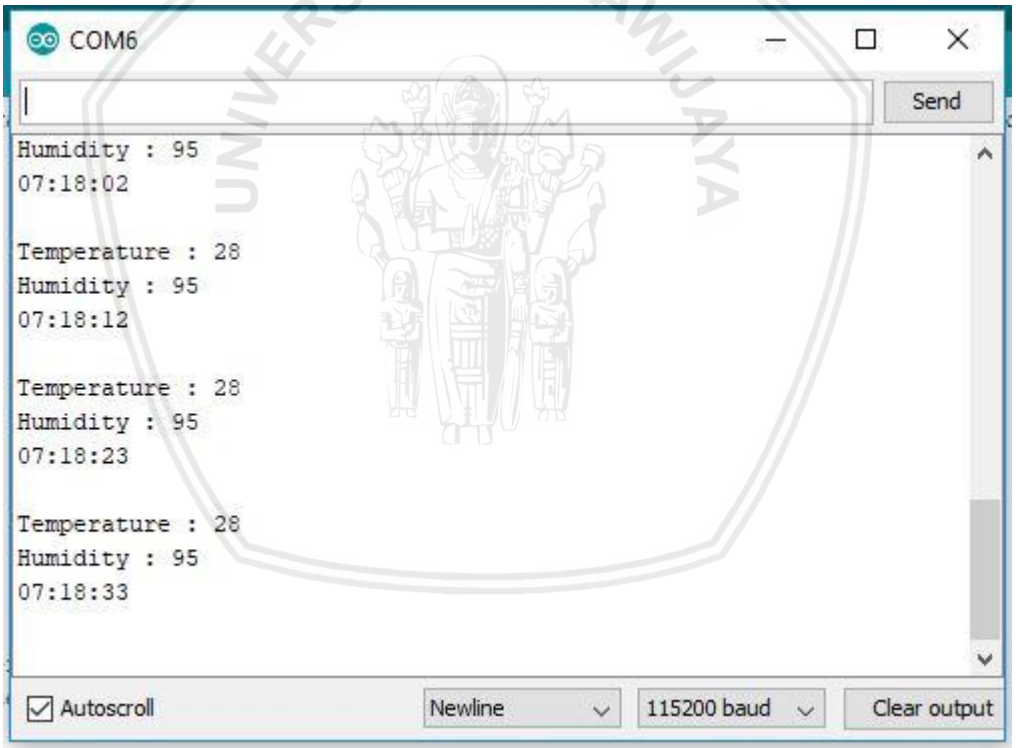
6.3.2 Prosedur Pengujian

Untuk melakukan pengujian pada RTC DS3231 itu sendiri dilakukan dengan prosedur sebagai berikut :

1. Menghubungkan Sistem dengan Komputer
2. Upload kode program RTCSetManually pada NodeMCU
3. Lakukan langakah-langkah memasukkan data atau format waktu dan tanggal yang diingingkan.
4. Amati hasil percobaan pada *serial monitor.*

6.3.3 Hasil Pengujian dan analisis

Hasil dan pengujian RTC DS3231 ditampilkan pada gambar dibawah ini.



Gambar 6. 3 Tampilan sistem pada *Serial Monitor*

Dari gambar diatas dapat kita lihat bahwa nilai dari RTC DS3231 itu sendiri sudah mampu keluar dengan baik dan benar pada serial monitor dengan memiliki jedah waktu pengiriman sebesar 10 detik.

6.4 Pengujian Arsitektur Publish dan Subscribe

Arsitektur ini merupakan arsitektur utama yang akan digunakan pada sistem monitoring ini. Dimana sistem akan mengirimkan data atau mem*publish* data pada

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

broker sesuai dengan topik yang telah ditentukan dan kemudian akan menunggu untuk di *subscribe* oleh user.

6.4.1 Tujuan pengujian

Tujuan dari 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.

6.4.2 Prosedur Pengujian

Untuk melakukan pengujian terhadap arsitektur *publish and subscribe* itu sendiri akan dilakukan dengan cara mengamati data yang di di*publish* 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.

6.4.3 Hasil dan Analisis Pengujian

Hasil dan pengujian arsitektur *publish and subscribe* pada sistem monitoring akan ditampilkan pada tabel berikut ini.

Tabel 6. 3 Hasil *Publish* data pad sistem

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

Pada tabel diatas terliahat bahwa semua data yang dikirim oleh perangkat keras pada broker dapat dikirim dengan baik dan benar. Adapun selisih waktu atau delay yang terjadi pada saat dua kali hingga tiga kali sebesar 1 detik itu terjadi karena pada sisi RTC DS3231 tidak dapat menampilkan nilai hingga mili detik. Sehingga delay teras seperti diakumulasikan pada pengiriman data ketiga dan keempat.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 6. 4 Hasil *Subscribe* data pada sistem

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

Tabel diatas merupakan tabel hasil pengujian pada sisi user pengguna android. Dimana data juga dapat deterima dengan baik dan benar dengan permasalahn delay yang sama. Adapun perbedaan waktu yang signifikan ini terjadi karena ketidak cocokan pada jam di *smartphone* pengguna dengan jam pada perangkat keras. Dengan demikian pengujian pada arsitektur *publish and subscribe* ini dapat dikatakan mampu berjalan dengan baik dan benar karena semua data dapat dikirim dan diterima dengan baik dan benar.

6.5 Pengujian Protokol MQTT

Merupakan salah satu protokol yang menerapkan arsitektur *publish-subscribe* dan didesain untuk kebutuhan komunikasi pada perangkat *Internet of Things*. Protokol komunikasi ini memungkinkan sistem melakukan komunikasi dengan broker dan menyimpan data yang kemudian akan menunggu untuk disubscribe.

6.5.1 Tujuan Pengujian

Untuk mengetahui apakah sistem mampu melakukan komunikasi dengan baik dengan broker. Sehingga data dapat di*publish* dengan baik dan benar.

6.5.2 Prosedur Pengujian

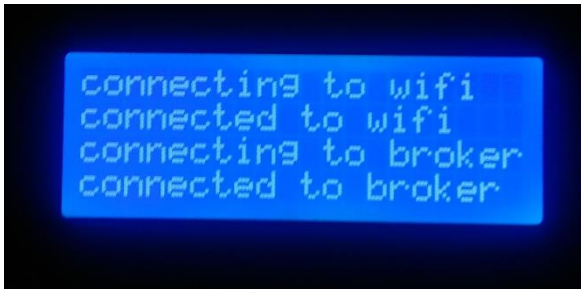
Untuk melakukan pengujian pada LCD 20x4 yaitu dengan prosedur sebagai berikut :

1. Menghubungkan sistem monitoring dengan Komputer
2. Upload kode program ke NodeMCU
3. Mengamati tampilan LCD apakah mampu melakukan proses *connecting to broker* dengan baik dan benar.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

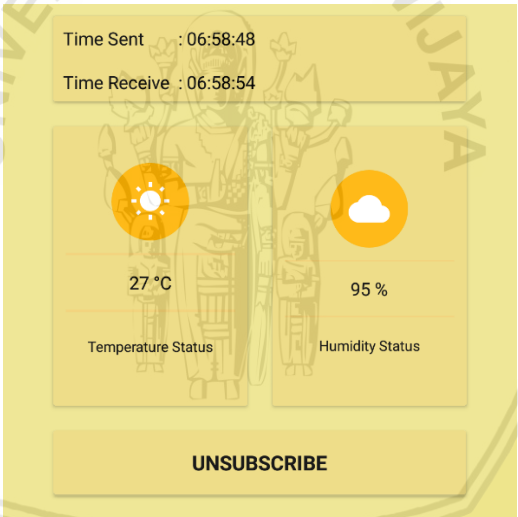
6.5.3 Hasil dan Analisis Pengujian

Untuk mengamati hasil dari percobaan mqtt itu sendiri dapat kita lihat pada hasil yang diberikan pada LCD berikut ini.



Gambar 6. 4 Proses koneksi dengan *Broker*

Pada gambar diatas menunjukan proses awal perangkat keras untuk melakukan sebuah proses *publish* data pada broker. Setelah melakukan koneksi dengan jaringan internet yang tersedia, kemudian sistem akan melakukan koneksi dengan broker yang telah ditentukan.



Gambar 6. 5 Hasil pada *MQTT client*

Tampilan diatas merupakan tampilan pada sisi pengguna dimana ini dilakukan pada proses *subscribe* dari pengguna sistem monitoring.

Berdasarkan hasil pengujian membuktikan bahwa sistem sudah mampu untuk melakukan proses koneksi pada broker sampai pada proses tersambung pada broker. Serta protokol mqtt ini juga sudah dapat memberikan nilai pada user pengguna dengan baik. Dengan demikian bisa dikatakan bahwa protokol mqtt dapat berjalan baik dan benar

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.6 Pengujian Aplikasi Android

Aplikasi android ini diharapkan mempermudah pengguna untuk mengetahui data yang telah di *publish* ke broker. Selain itu aplikasi android ini adalah komponen utama untuk melakukan *subscribe* pada sistem monitoring ini.

6.6.1 Tujuan pengujian

Pada 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 men*subscribe* dengan baik dan benar. Dengan demikian dapat diketahui perbedaan waktu pengiriman data dari alat monitoring hingga waktu diterima pada aplikasi android.

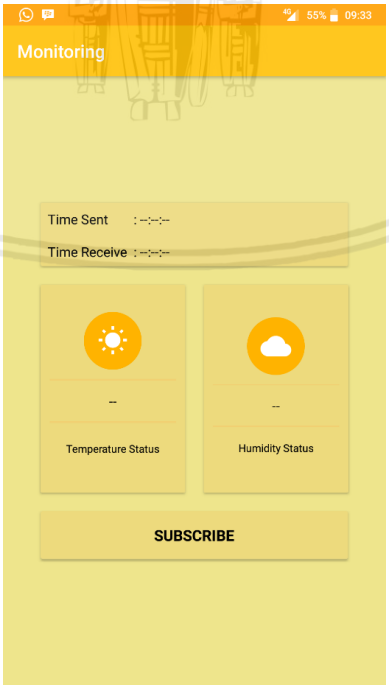
6.6.2 Prosedur Pengujian

Untuk melakukan pengujian pada kipas yaitu dengan prosedur sebagai berikut :

1. Menyalakan Sistem monitoring.
2. Buka aplikasi android yang sudah terpasang pada perangkat android.
3. Tekan tombol subscribe yang ada pada aplikasi android.

6.6.3 Hasil Pengujian dan analisis

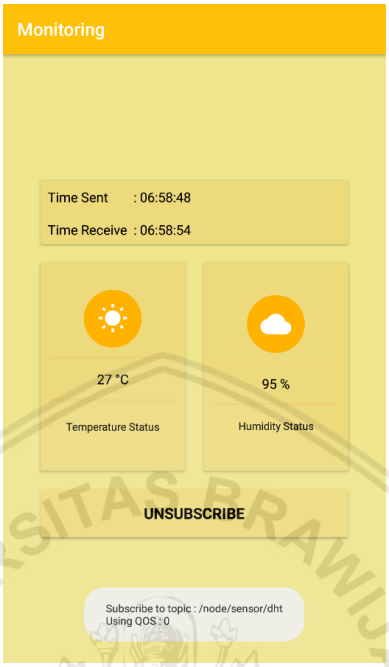
Hasil dan pengujian aplikasi android dapat kita lihat pada gambar dibawah ini.



Gambar 6. 6 Tampilan awal aplikasi *Android*

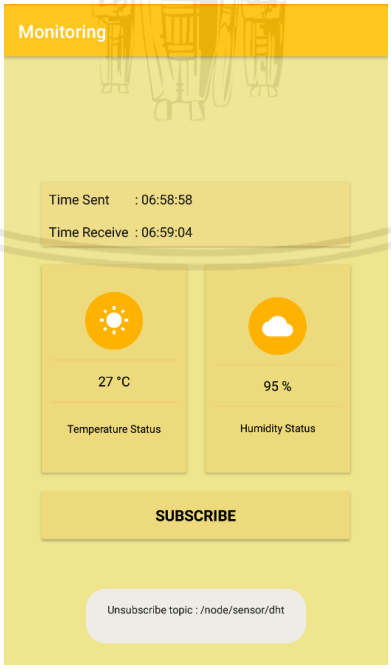
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar diatas merupakan gambar dimana kondisi *idle*. Dimana kondisi tersebut akan menunggu si pengguna akan melakukan proses *subscribe.*



Gambar 6. 7 Tampilan saat *subscribe*

Pada proses yang ditampilkan pada gmbar diatas merupakan proses *subscribe* pada sistem diamana terdapat teks komentar yang muncul berisikan penjelasan topik yang di *subscribe*.



Gambar 6. 8 Tampilan saat *Unsubscribe*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Dan yang terakhir adalah proses *unsubscribe* dimana user akan berhenti berlangganan pada topik tertentu tersebut. Pada proses ini juga akan muncul teks komentar pada bagian bawah aplikasi yang menjelaskan bahwa user telah berhenti berlangganan pada topik tersebut.

Dengan demikian bisa dikatakan bahwa 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 itu sendiri.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan perancangan dan hasil pengujian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Pada penelititan 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.
2. 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. Hal ini ditandai dengan data mampu di *publish* pada broker serta mampu di *subscribe* oleh pengguna pada aplikasi android dengan tanda munculnya teks koment yang menyatakan ”Subscribe to topic : /node/sensor/dht Using QOS : 0” pada aplikasi android. Dengan demikian disimpulkan bahw data mampu diterima dengan baik dan sesuai oleh use pengguna.
3. 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 kanadang 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 dengan menggunakan aplikasi android yang penulis buat. Dengan aplikasi yang mudah untuk dijlankan dan tampilan yang mudah dimengerti oleh pemilik ular, maka dengan demikian sistem akan mampu meningkatkan kualitas kandang itu sendiri.

7.2 Saran

Terdapat beberapa saran yang ditujukan peneliti untuk pengembangan sistem berikutnya sebagai berikut :

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1. Untuk penelitian selanjutnya diharapkan dapat melakukan otomasi pada sistem untuk menetralisirkan suhu dan kelembaban sesuai parameter yang telah ditetapkan.
2. Pada penelitian berikutnya diharapkan melakukan penelitian mendalam tentang kemungkinan apa saja yang menyebabkan ular piton regius terkana penyakit. Sehingga alat otomasi yang akan dibuat mampu mengatasi dengan baik dan benar.
3. Pada penelitian berikutnya diharapkan menggunakan perangkat RTC yang dapat melakukan update jam serta memberikan nilai hingga mili detik.
4. Untuk penelitian selanjutnya diharapkan aplikasi android memiliki sistem notifikasi sehingga pengguna dapat lebih waspada.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Ajie, S. (2016, Agustus 10). *Mengukur Suhu dan Kelembaban Udara Dengan Sensor DHT11 dan Arduino*. Diambil kembali dari saptaji.com: http://saptaji.com/2016/08/10/mengukur-suhu-dan-kelembaban-udara-dengan-sensor-dht11-dan-arduino/

Akbar, S, R., Kurniawan, W., Ichsan, M, H, H., Arwani, I., Handono, M, T., 2016. *Pervasive Device and Service Discovery Protocol In XBee Sensor Network*. Malang

Amron, K., Sakti, E., Data, M., 2016. Pemodelan dan Analisis *Wireless Mesh Network* dengan Arsitektur *Publish-Subscribe* dan Protokol MQTT. Malang

CNXSOFT. (2015, April 18). *NodeMCU is both a Breadboard-Friendly ESP8266 Wi-Fi Board and a LUA based Firmware*. Diambil kembali dari cnxsoft: https://www.cnx-software.com/2015/04/18/nodemcu-is-both-a-breadboard-friendly-esp8266-wi-fi-board-and-a-lua-based-firmware/

DFRobot. (2017, September 14). *DHT11 Temperature and Humidity Sensor (SKU: DFR0067)*. Dipetik Juli 5, 2018, dari DFRobot: https://www.dfrobot.com/wiki/index.php/DHT11_Temperature_and_Humidity_Sensor_(SKU:_DFR0067)

ECMA International., 2013. *The JSON Data Interchange Format* . Jenewa

Espressif Systems IOT Team. (2015). *ESP8266EX Datasheet Version 4.3.* bbs.espressif.

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].

Kamilaris, A., Pitsillides, A., 2016. *Mobile Phone Computing and the Internet of Things: A Survey.*

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.

OSEPP Electronics. (2013). DHT11 Technical Data Sheet. Dalam *DHT11 Humidity & Temperature Sensor* (hal. 6). Mouser Electronics.

Putra, C. G. (2017). *IMPLEMENTASI OTOMASI KANDANG DALAM RANGKA MEMINIMALISIR HEAT STRESS PADA AYAM BROILER DENGAN METODE FUZZY SUGENO .* Malang: JPTIIK UNIVERSITAS BRAWIJAYA.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

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

Rochman, H, A., Primananda, R., Nurwasito, H., 2016. *Sistem Kendali Berbasis Mikrokontroller Menggunakan Protokol MQTT pada Smarthome.* Malang

Suprayogi, D, A., Akbar, S, R., Ichsan, M, H, H., 2016. *Implementasi Pervasive Service Discovery Protocol pada Rumah Cerdas Berbasis NRF24L01.* Malang

Taringan, S. O. F. (2014). Pengukuran Kinerja Sistem *Publish/Subscribe* Menggunakan Protokol MQTT *(Message Queuing Telemetry Transport).* Bandung

Wiley, 2010. *Fundamentals of Wireless Sensor Networks. United Kingdom*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA