



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

RANCANG BANGUN INKUBATOR BERBASIS IoT

Andre Perwiratama
NRP 07111645000079

Dosen Pembimbing
Dr. Ir. Hendra Kusuma, M.Eng.Sc
Dr. Ir. Totok Mujiono, M.Kom

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TE 145561

RANCANG BANGUN INKUBATOR BERBASIS IoT

Andre Perwiratama
NRP 07111645000079

Dosen Pembimbing
Dr. Ir. Hendra Kusuma, M.Eng.Sc
Dr. Ir. Totok Mujiono, M.Kom.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 145561

DESIGN OF *IoT* BASED INCUBATOR

Andre Perwiratama
NRP 07111645000079

Advisor
Dr. Ir. Hendra Kusuma, M.Eng.Sc
Dr. Ir. Totok Mujiono, M.Kom.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Rancang Bangun Inkubator Berbasis IoT**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 2 Juli 2018



Andre Perwiratama
NRP 07111645000079

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN INKUBATOR BERBASIS IoT

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II



Dr. Ir. Hendra Kusuma, M.Eng.Sc
NIP: 196409021989031003



Dr. Ir. Totok Mujiono, M.Kom.
NIP: 196504221989031001



-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN INKUBATOR BERBASIS IoT

Nama : Andre Perwiratama
Pembimbing I : Dr. Ir. Hendra Kusuma, M.Eng.
Pembimbing II : Dr. Ir. Totok Mujiono, M.Kom.

ABSTRAK

Inkubator bayi saat ini pada umumnya hanya berfungsi sebagai tempat penumbuh bayi yang fungsinya hanya mengatur temperatur dan kelembaban udara di dalam inkubator tersebut. Selain itu inkubator komersial saat ini juga belum dapat dipantau dari jarak jauh sehingga menyebabkan perawat maupun dokter harus memantau kondisi bayi dengan mendatangi inkubator tersebut. Pada Tugas Akhir ini telah dibuat suatu inkubator yang dapat dikendalikan via Internet (IOT). Inkubator ini terintegrasi dengan sistem pengendali dan pemantauan suhu dan kelembaban di dalam inkubator. Dengan menggunakan mikrokontroler Arduino MEGA sebagai unit pemroses pusat, sistem inkubator ini juga mampu mengirim data-data yang terbaca pada sensor via internet (IoT). Prototipe inkubator hasil tugas akhir ini, mampu membaca parameter suhu dan kelembaban via Internet dengan konstan dan tepat serta mampu memberikan parameter setting point suhu dan kelembaban yang telah ditentukan.

Kata Kunci : Arduino MEGA, Inkubator, IoT

-----Halaman ini sengaja dikosongkan-----

DESIGN OF IoT BASED INCUBATOR

Name : Andre Perwiratama
Advisor I : Dr. Ir. Hendra Kusuma, M.Eng.
Advisor II : Dr. Ir. Totok Mujiono, M.Kom.

ABSTRACT

incubator for baby nowadays just have a function as a place for baby to grow that can only control the temperature and humidity inside that incubator. Also, the incubator cant be monitored from a far place making the doctor or nurse must go to the incubator in person to monitor the baby's condition manually. In this final project, proposed an incubator, a system that is integrated with a control system and monitoring of a temperature and humidity inside the incubator. beside that, with using microcontroller Arduino MEGA, this incubator system also can send the data that is read by the sensor to the internet (IoT) taht can be accessed from far away. the result that is obtained is that the incubator already can send the reading data to server through the internet and that the value of the set point can be set with the application from an android.

Keyword : Arduino MEGA, Incubator, IoT

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Strata-1 pada Bidang Studi Teknik Sistem Tenaga, Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN INKUBATOR BERBASIS IoT

Dengan selesainya Tugas Akhir ini penulis menyampaikan terimakasih sebesar – besarnya kepada:

1. Kedua orang tua atas limpahan doa bagi penulis
2. Bapak Dr. Ir. Hendra Kusuma, M.Eng. Sc dan bapak Dr. Ir. Totok Mujiono, M.Kom selaku dosen pembimbing pertama dan kedua yang telah meluangkan waktu dan tenangnya untuk dapat membimbing penulis dalam menyelesaikan Tugas Akhir ini.
3. Seluruh staff dan pengajar departemen Teknik Elektro FTE – ITS.
4. Semua pihak yang telah banyak membantu untuk menyelesaikan Tugas Akhir ini.
5. Harapan kami sebagai penulis adalah terselesaikannya.

Tugas Akhir ini dapat bermanfaat bagi kami serta pembaca. Sadar atas keterbatasan yang dimiliki oleh penulis karena hasil dari Tugas Akhir ini belum sempurna. Demikian penulis berusaha semaksimal mungkin dan penulis mengharapkan pintu maaf serta saran dan kritik yang membangun.

Surabaya, **2 Juli 2018**

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

HALAMAN JUDUL	i
HALAMAN JUDUL	ii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
RANCANG BANGUN INKUBATOR BERBASIS IoT.....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian.....	2
1.6 Relevansi.....	3
BAB II TEORI DASAR.....	5
2.1 <i>Internet of Things</i> (IoT).....	5
2.2 <i>Web server</i>	5
2.3 Kontrol Histerisis.....	6
2.3 Inkubator bayi.....	6
2.4 Arduino.....	8
2.5 Sensor kelembaban dan suhu, DHT11.....	8
2.6 humidifier.....	9
2.7 Elemen pemanas.....	10
2.8 Kipas DC.....	10
2.9 Arduino IDE.....	11
2.10 IFTTT.....	12
2.11 THINGSPEAK.....	13
BAB III TEORI DASAR.....	15
3.1 Blok Fungsional Sistem inkubator berbasis IoT.....	15

3.2	Perancangan Mekanik inkubator berbasis IoT	15
3.3	Perancangan Elektrik inkubator berbasis IoT	16
3.3.1	Perancangan Sensor DHT11	17
3.3.2	Perancangan sistem pengatur suhu.....	17
3.3.3	Perancangan sistem pengatur kelembaban	18
3.4	Perancangan <i>Software</i> inkubator berbasis IoT	19
3.4.1	Perancangan Program sensor dan aktuator suhu DHT11 .	19
3.4.2	Perancangan Program sensor dan aktuator kelembaban DHT11	20
3.4.3	Perancangan Program aplikasi pengaturan set point	21
3.5	Perancangan sistem pengatur suhu inkubator berbasis IoT	23
3.6	Diagram alir sistem inkubator berbasis IoT	24
3.7	Spesifikasi Inkubator berbasis IoT	25
BAB IV TEORI DASAR		27
4.1	Pengujian sensor DHT11	27
4.2	Pengujian sensor DHT11 dengan aktuator.....	29
4.2.1	Pengujian sensor dengan set point 38 derajat	29
4.2.2	Pengujian sensor dengan set point 40 derajat	30
4.3	Pengujian pembacaan sensor pada thingspeak	30
4.4	Pengujian notifikasi pada <i>Handphone</i>	32
4.5	Pengujian inkubator berbasis IoT secara keseluruhan	32
BAB V PENUTUP.....		39
DAFTAR PUSTAKA.....		41
LAMPIRAN A.....		43
A.1.	Program keseluruhan sensor DH11 dan aktuator	43
A.2	Datasheet DHT11.....	49
A.3	Datasheet ESP8266-01	51
RIWAYAT HIDUP.....		55

DAFTAR GAMBAR

Gambar 2.1 Konsep IoT	5
Gambar 2.2 Inkubator bayi	7
Gambar 2.3 Arduino MEGA	8
Gambar 2.4 bentuk fisik sensor DHT11	9
Gambar 2.5 bentuk fisik ultrasonik humidifier	9
Gambar 2.6 bentuk fisik elemen pemanas.....	10
Gambar 2.7 bentuk fisik kipas DC.....	10
Gambar 2.8 Tampilan Arduino IDE.	11
Gambar 2.9 Logo IFTTT.....	12
Gambar 2.10 Logo Thingspeak	123
Gambar 3.1 Diagram blok sistem.	15
Gambar 3.2 Desain inkubator tampak samping	16
Gambar 3.3 Desain inkubator tampak atas.....	16
Gambar 3.4 perancangan elektrik DHT11	17
Gambar 3.5 perancangan pengatur suhu DHT11	18
Gambar 3.6 perancangan pengatur kelembaban DHT11	19
Gambar 3.7 Program pengaturan suhu melalui DHT11	20
Gambar 3.8 Program pengaturan kelembaban	21
Gambar 3.9 program kontrol histerisis.....	24
Gambar 3.10 Diagram alir sistem	24
Gambar 4.1 Hasil pengambilan data selama 40 detik	274
Gambar 4.2 Grafik pembacaan sensor DHT11 dengan aktuator menggunakan set point 38 derajat	29
Gambar 4.3 Grafik pembacaan sensor DHT11 dengan aktuator menggunakan set point 40 derajat	30
Gambar 4.4 Grafik pembacaan suhu sensor DHT11 pada thingspeak ..	31
Gambar 4.5 Grafik pembacaan kelembaban sensor DHT11 pada thingspeak	31
Gambar 4.6 Gambar notifikasi pada handphone	32
Gambar 4.7 Pembacaan suhu saat alat pertama dinyalakan.....	33
Gambar 4.8 pembacaan kelembaban saat alat pertama dinyalakan.....	33
Gambar 4.9 relay menyala saat alat pertama dinyalakan	34
Gambar 4.10 Pembacaan suhu saat nilai pembacaan melebihi threshold.....	34
Gambar 4.11 Pembacaan kelembaban saat nilai suhu melebihi threshold.....	35
Gambar 4.12 Relay mati saat nilai suhu melebihi threshold	35

Gambar 4.13 Grafik suhu saat ditrigger dengan nilai 0 untuk suhu	36
Gambar 4.14 Notifikasi bila suhu dibawah nilai threshold bawah	36
Gambar 4.15 Notifikasi bila suhu dibawah nilai threshold atas	36
Gambar 4.16 tampilan pembacaan suhu kelembaban dan set point.....	37
Gambar 4.17 interface pengaturan set point pada handphone	37
Gambar 4.18 tampilan lcd setelah set point dirubah ke 40.....	38

DAFTAR TABEL

Tabel4.1 data pembacaan sensor DHT11 selama 40 detik.....	28
Tabel4.1 hasil masukan set point pada inkubator melalui <i>handphone</i> ..	38

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan meningkatnya kebutuhan masyarakat di jaman teknologi informasi mendorong kebutuhan suatu sistem yang mempermudah dan meningkatkan efektifitas dalam berbagai pekerjaan. Dengan teknologi di bidang elektronika dan komputer yang telah berkembang, maka banyak hal yang dapat dilakukan dengan cepat dan tepat untuk memenuhi kebutuhan manusia. Salah satu aplikasi dari teknologi tersebut adalah IoT (*Internet Of Things*) , yaitu sebuah konsep dimana suatu objek yang memiliki kemampuan untuk berkomunikasi data melalui jaringan internet tanpa memerlukan interaksi manusia.

inkubator merupakan sesuatu yang harus dimiliki oleh setiap rumah sakit, akan tetapi inkubator yang ada pada rumah sakit terkadang kurang praktis, karena perawat atau dokter harus mengecek inkubator secara berkala untuk memastikan bayi berada pada kondisi yang sesuai, selain itu, inkubator yang berkualitas bagus berharga mahal, hal ini menyebabkan terbatasnya ketersediaan inkubator pada beberapa rumah sakit, terutama daerah pedesaan.

Berdasarkan pertimbangan-pertimbangan diatas, untuk menyelesaikan masalah-masalah tersebut, dibuat inkubator yang dapat memantau keadaan bayi dari jarak jauh dengan harga yang terjangkau.

Inkubator ini berbasis IoT (*Internet of Things*) yang dapat menghubungkan inkubator dengan internet sehingga pembacaan sensor-sensor yang ada pada inkubator ini dapat dilihat melalui internet.

Diharapkan dengan adanya alat ini, dapat meringankan beban perawat maupun dokter pada rumah sakit yang masih menggunakan cara manual dengan mengecek keadaan bayi pada inkubator, dan juga membantu menyediakan inkubator yang terjangkau untuk rumah sakit yang belum bisa menyediakan inkubator berkualitas, terutama pada pedesaan.

1.2 Permasalahan

Permasalahan pada penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membuat struktur inkubator

sehingga mampu menghasilkan temperatur ruang yang stabil.

2. Bagaimana cara mengatur kelembaban yang stabil dan ideal untuk bayi di dalam inkubator.
3. Bagaimana cara menghubungkan inkubator ini ke internet.

1.3 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Inkubator yang dibuat merupakan simulasi/miniaturnya sesungguhnya.
2. Sistem kontrol inkubator menggunakan Arduino MEGA yang dilengkapi dengan sensor-sensor suhu dan kelembaban serta kadar O₂.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Membuat inkubator yang terdiri dari sensor suhu, kelembaban, serta *heater* dan *humidifier*.
2. Mengatur inkubator agar mempunyai suhu dan kelembaban ruang yang stabil dan ideal untuk bayi.
3. Dapat menghubungkan inkubator ini ke internet.

1.5 Metodologi Penelitian

Perancangan miniatur ranjang berbasis pengenalan suara terbagi menjadi empat tahapan, yaitu studi literatur, perancangan sistem, uji coba dan hasil pengujian, serta penyusunan laporan.

Pada tahap studi literatur, dilakukan pencarian literatur buku maupun kumpulan makalah dan jurnal yang mengarah pada topik yang dibahas. Tahapan ini dilakukan untuk mengumpulkan informasi tentang standar inkubator serta kondisi ideal untuk bayi pada inkubator.

Selanjutnya pada perancangan sistem, dibuat suatu program dengan perangkat Arduino IDE untuk pemrograman mikrokontroler A. Pada tahapan ini membahas tentang koneksi secara perangkat keras maupun secara perangkat lunak.

Setelah itu dilakukan pengujian baik perangkat keras maupun perangkat lunak. Pada tahap ini pengujian perangkat lunak dilakukan untuk menguji seberapa baik algoritma yang diterapkan pada pengenalan suara, sedangkan pengujian perangkat keras dilakukan agar mengetahui

sistem telah berjalan dengan baik atau belum. Pembahasan tugas akhir ini dibagi menjadi lima bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Pada bab pendahuluan, menjelaskan mengenai latar belakang pemilihan topik, perumusan masalah dan batasannya. Bab ini juga membahas mengenai tujuan penelitian, metodologi, sistematika laporan, dan relevansi dari penelitian yang dilakukan.

Bab II Teori Dasar

Penjelasan mengenai komponen *hardware* maupun *software* pendukung untuk Inkubator dibahas pada bab ini. Pokok bahasan pada bab ini diantaranya, yaitu komponen penyusun mesin, cara kerja, dan pengoperasian mesin tersebut.

Bab III Perancangan Alat

Pembahasan yang dilakukan pada bab ini, mengenai perancangan sistem secara keseluruhan serta prosedur koneksi *hardware* dan simulasi sistem yang dibuat.

Bab IV Pengujian *Hardware* dan Simulasi Sistem

Hasil dari uji coba program secara terpisah maupun secara keseluruhan terdapat pada bab ini.

Bab V Penutup

Pada bagian bab penutup, dibahas mengenai kesimpulan dan saran dari hasil pengujian.

1.6 Relevansi

Hasil yang diperoleh dari tugas akhir ini adalah kontribusi berupa perancangan dan implementasi sebuah inkubator yang dilengkapi berbagai fitur baru seperti IOT.

-----Halaman ini sengaja dikosongkan-----

BAB II TEORI DASAR

Pada bab ini akan dibahas mengenai tinjauan pustaka dari inkubator. Mulai dari mekanisme inkubator bayi, hardware dan software yang digunakan.

2.1 *Internet of Things (IoT)*

Internet of Thing (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer[1]. IoT telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems (MEMS)*, dan Internet. cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, yakni: Barang Fisik yang dilengkapi modul IoT, Perangkat Koneksi ke Internet seperti Modem dan Router Wireless, dan Cloud Data Center tempat untuk menyimpan aplikasi beserta data base.



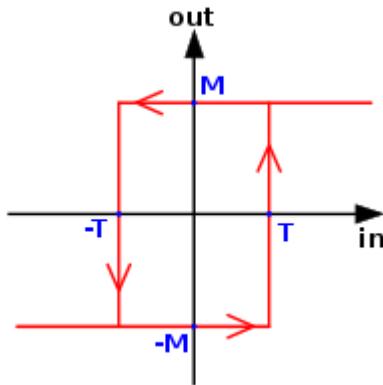
Gambar 2. 1 Konsep IoT

2.2 *Web server*

Webserver merupakan salah satu kebutuhan yang digunakan oleh user untuk website yang mempunyai kapasitas penyimpanan yang besar dan juga akses yang cepat untuk trafik yang besar dalam mencegah terjadinya down pada suatu website atau aplikasi[2]. Fungsi utama Server atau Web server adalah untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang di dalam berupa teks, video, gambar dan banyak lagi.

2.3 Kontrol Histerisis

Histerisis merupakan perilaku atau sifat dari sebuah sistem dimana sebuah sistem tersebut gagal untuk kembali ke keadaan semula atau sebelumnya, setelah penyebab dari perubahan-perubahan tersebut dihilangkan. Kontrol histerisis sendiri adalah salah satu jenis kontrol yang mengatur agar nilai yang dikontrol dapat stabil dalam satu nilai yang telah ditentukan, dengan mengatur batas atas dan batas bawah sedikit melebihi dan kurang dari set point yang diinginkan, hal ini dilakukan agar menghindari perpindahan saklar on/off yang cepat.



Gambar 2.2 kontrol histerisis

2.3 Inkubator bayi

Inkubator bayi adalah alat biomedis yang memberikan kehangatan dan kelembaban dimana seluruh lingkungannya terkontrol dan diperlukan oleh bayi yang baru lahir. Akan tetapi tidak semua bayi yang baru lahir memerlukan alat biomedis ini, ada beberapa kriteria bayi yang memerlukan kondisi terkontrol seperti dalam inkubator ini.

Inkubator ini dilengkapi dengan tombol pengatur suhu waktu untuk memudahkan pengaturan suhu yang dikehendaki.



Gambar 2.3 Inkubator bayi

Bayi prematur akan disimpan pada inkubator sampai sudah dianggap tidak memerlukan inkubator lagi.

Faktor-faktor yang menjadi pertimbangan bagi bayi untuk dikeluarkan dari inkubator adalah:

1. kemampuan bayi dalam menjaga temperatur tubuh.

faktor ini merupakan faktor kunci yang menjadi pertimbangan apakah bayi dapat dipindahkan dari dalam inkubator atau tidak. Pada bayi prematur, jaringan lemak / jaringan adiposa bayi serta lemak pada bagian subkutan kulitnya belum sempurna atau tidak seperti bayi normal pada umumnya, sehingga hal inilah yang menyebabkan bayi prematur tidak dapat mempertahankan suhu tubuhnya sendiri tanpa lingkungan yang hangat

2. berat badan bayi

bayi yang dapat dipindahkan dari inkubator adalah bayi yang beratnya sudah mencapai 1700 sampai 1800 gram[3].

3. usia kehamilan

bayi mulai memproduksi jaringan adiposa pada saat usia kandungan ibu sekitar 26-30 minggu[4]. jaringan adiposa pada bayi yang dilahirkan

secara normal sudah terbentuk sempurna sehingga bayi dapat menyesuaikan suhu tubuhnya dengan lingkungan sekitar. Jika bayi yang ada dalam inkubator sudah dapat beradaptasi dengan lingkungannya maka bayi dapat dikeluarkan dari inkubator.

2.4 Arduino

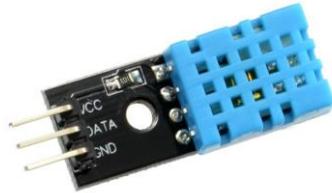
Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang[5]. Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis, namun ada individu/perusahaan yang membuat *clone* arduino dengan menggunakan mikrokontroler lain dan tetap kompatibel dengan arduino pada level hardware. Arduino juga adalah platform open source yang digunakan untuk membangun project elektronik. Arduino terdiri dari papan rangkaian yang dapat diprogram (hardware) dan perangkat lunak (software).



Gambar 2.4 Arduino MEGA

2.5 Sensor kelembaban dan suhu, DHT11

DHT11 adalah sensor yang dapat mengukur dua parameter lingkungan sekaligus yaitu suhu dan kelembaban udara. Dalam sensor ini terdapat sebuah thermistor tipe NTC (Negative Temperature Coefficient) untuk mengukur suhu, sebuah sensor kelembaban tipe resistif dan sebuah sensor kelembaban tipe resistif dan sebuah mikrokontroler 8 bit yang mengolah kedua sensor tersebut dan mengirim hasilnya ke pin output dengan format single-wire bi-directional (kabel tunggal dua arah). Pada sensor DHT 11, terdapat 3 kaki, yaitu VCC yang disambung pada tegangan 3.3 V, GND untuk ground dan DATA untuk pembacaan data yang akan tersambung pada kaki ADC arduino[6].



Gambar 2.5 bentuk fisik sensor DHT11

2.6 humidifier

Humidifier adalah sebuah aktuator untuk melembabkan udara, cara kerja dari humidifier adalah menyempotkan air yang ditampung ke udara dengan tujuan untuk melembabkan udara tetapi tidak dapat digunakan untuk membersihkan udara. *Ultrasonic mist make humidifier* ini bekerja pada tegangan 24 V.



Gambar 2.6 bentuk fisik ultrasonik humidifier

2.7 Elemen pemanas

Elemen pemanas berfungsi untuk memanaskan suhu pada inkubator agar suhu pada inkubator dapat sesuai dengan set point yang ditentukan, elemen pemanas pada inkubator ini berfungsi secara peniupan udara yang dilakukan dari salah satu sisi pemanas untuk mendapatkan udara yang panas dari sisi yang lain. Elemen pemanas ini bekerja pada tegangan AC 220V.



Gambar 2.7 bentuk fisik elemen pemanas.

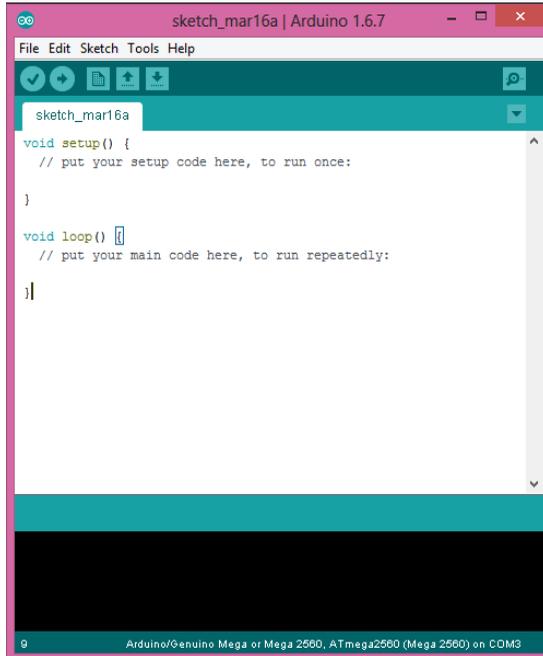
2.8 Kipas DC

Kipas DC merupakan kipas dengan tegangan input sebesar 12V, fungsi dari kipas DC ini adalah untuk meniupkan udara melalui elemen pemanas agar dapat menghasilkan udara panas yang digunakan untuk menghangatkan inkubator, pada rangkaian alat ini, kipas DC dikontrol oleh PWM yang besarnya bergantung pada data panas yang dibaca oleh sensor DHT11, semakin panas suhu yang dibaca, maka putaran dari kipas akan semakin lambat, begitu pula sebaliknya, bila suhu yang dibaca rendah, maka putaran kipas DC akan menjadi semakin cepat.



Gambar 2.8 bentuk fisik kipas DC.

2.9 Arduino IDE



Gambar 2.9 Tampilan Arduino IDE.

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan[7]. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai

penengah antara *compiler* Arduino dengan mikrokontroler. Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Pada Software Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan.

2.10 IFTTT

IFTTT adalah sebuah singkatan dari If This Then That. Sebuah layanan otomatisasi yang menggabungkan beberapa layanan internet menjadi satu. Layanan IFTTT secara garis besar lebih mudah digambarkan seperti ini, “Jika terjadi A, maka lakukan hal B”. Tidak hanya layanan web, IFTTT juga dapat digunakan untuk mengotomasi peralatan yang terhubung dengan internet seperti termostat atau lampu pintar.

Pada IFTTT, terdapat Applets atau semacam resep yang bisa kamu aktifkan untuk mengotomasi pekerjaan. Kamu bisa menggunakan Applets yang sudah dibuat oleh orang lain atau membuatnya sendiri berdasarkan kebutuhanmu. Jika kamu sudah pernah mendengar tentang IFTTT tetapi belum menggunakannya, berikut 6 alasan sekaligus Applets yang mungkin dapat memberikan gambaran betapa bergunanya IFTTT dalam keseharian.



Gambar 2.10 Logo IFTTT

2.11 THINGSPEAK

thingspeak adalah open source Internet of Things (IoT) dan API untuk menyimpan dan mengambil data menggunakan protokol HTTP menggunakan internet. Thingspeak memungkinkan pengguna untuk berkreasi dalam pembuatan aplikasi penulisan, tracking lokasi dan media sosial dengan update status.

thingspeak mendapatkan dukungan integrasi dari software MATLAB dan Mathworks, yang memungkinkan pengguna thingspeak untuk menganalisis dan memvisualisasikan data yang diupload menggunakan matlab tanpa perlu membeli lisensi matlab dari mathworks.



Gambar 2.11 Logo Thingspeak

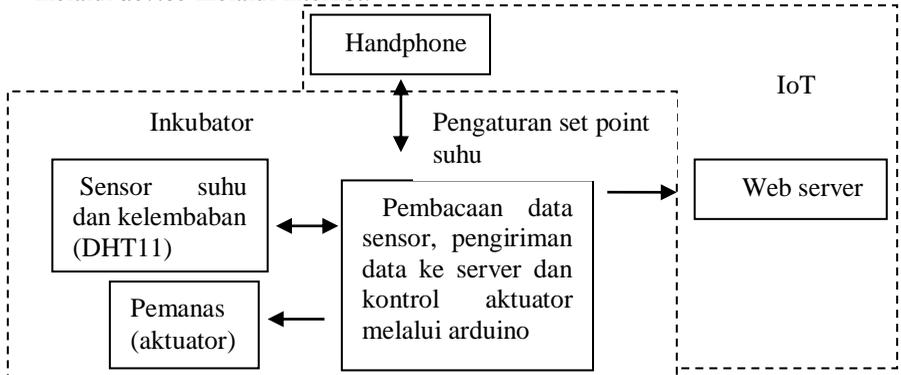
-----Halaman ini sengaja dikosongkan-----

BAB III PERANCANGAN ALAT

Pada bab ini akan dibahas mengenai perancangan dan pembuatan sistem perancangan inkubator yang meliputi blok fungsional sistem yang akan menjelaskan proses kerja alat dalam bentuk alir diagram, perancangan mekanik yang membahas tentang desain dan pembuatan mekanik yang mendukung cara kerja alat, dan perancangan perangkat elektrik.

3.1 Blok Fungsional Sistem inkubator berbasis ioT

Pada tugas akhir ini, diterapkan sistem inkubator yang dapat memonitor keadaan suhu dan kelembaban yang ada di dalam inkubator melalui internet. Seperti yang dapat dilihat pada gambar 3.1, sistem ini terdiri dari sensor suhu dan kelembaban serta aktuator yang berfungsi untuk mengontrol suhu yang ada di dalam inkubator agar keadaannya menjadi konstan, hasil dari pembacaan sensor dapat diakses melalui *device* melalui internet.



Gambar 3. 1 Blok fungsional sistem.

3.2 Perancangan Mekanik inkubator berbasis ioT

Perancangan mekanik dari inkubator ini meliputi perancangan *prototype* dari keseluruhan alat, mulai dari peletakan sensor, mikrokontroller, aktuator dan pemasangan per bagian secara keseluruhan.

Desain inkubator dibuat dengan ukuran 70cm x 50 cm x 15 cm, pada bagian samping terdapat lubang yang dapat mengalirkan panas dari pemanas yang telah dilengkapi dengan kipas DC, pada bagian samping yang lain terdapat humidifier yang telah dipisah dengan bagian akrilik sehingga asap hanya dapat masuk dari bagian bawah inkubator, dan pada bagian luar terdapat bagian kontrol yang mengatur jalannya inkubator.



Gambar 3.1 Desain inkubator tampak samping.



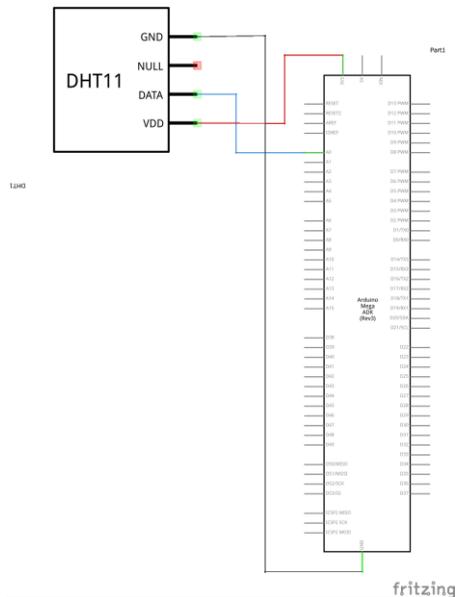
Gambar 3.2 Desain inkubator tampak atas.

3.3 Perancangan Elektrik inkubator berbasis IoT

Perancangan perangkat elektrik meliputi *wiring input* dan *output* dari kontroler serta perancangan *supply* tegangan.

3.3.1 Perancangan Sensor DHT11

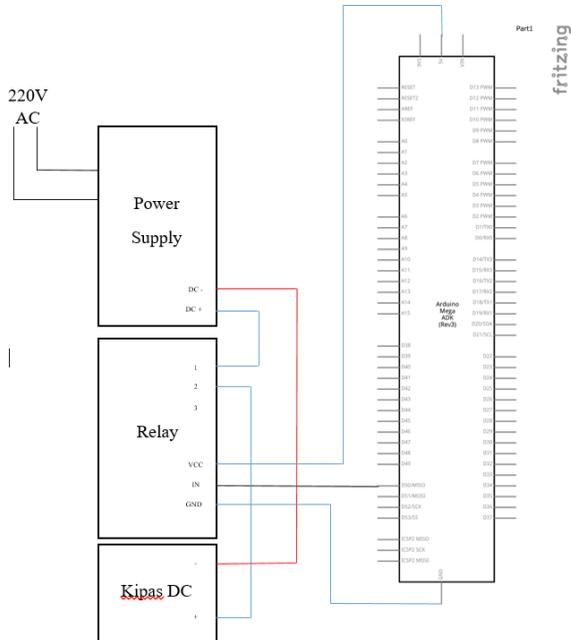
Pada perancangan sensor DHT 11, terdapat 3 pin yang harus dipasang, yaitu pin VCC yang disambungkan pada pin 3.3V arduino, GND untuk ground, dan data untuk pembacaan yang disambungkan ke pin ADC input A0



Gambar 3.3 perancangan elektrik DHT11

3.3.2 Perancangan sistem pengatur suhu

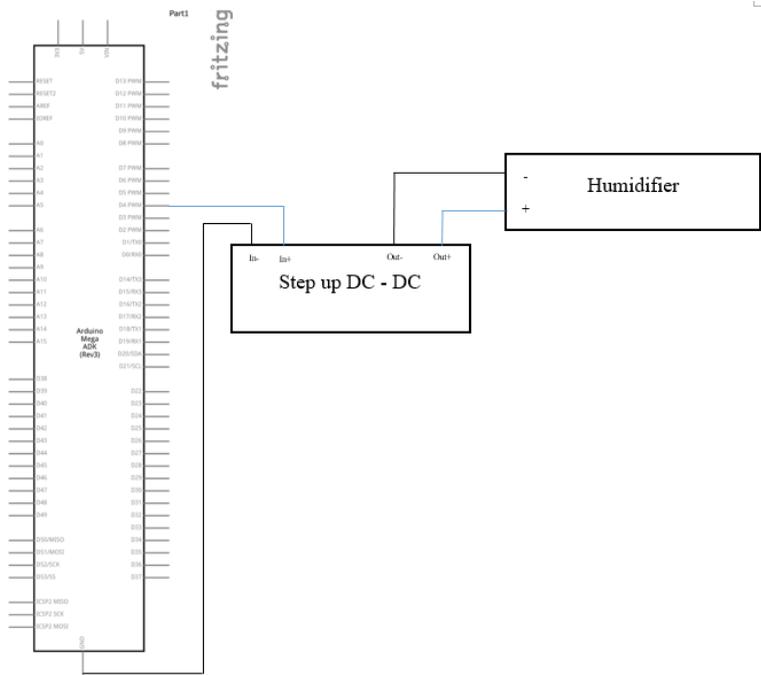
Pada perancangan sistem pengatur suhu, terdapat tiga komponen, yaitu mikrokontroler, kipas DC dan elemen pemanas. Kipas DC dihubungkan pada salah satu pin PWM, sedangkan elemen pemanas dihubungkan dengan tegangan 220V, kipas DC diatur sehingga pada saat suhu di dalam inkubator dingin, maka nilai PWM akan bertambah, dan sebaliknya jika suhu di dalam inkubator panas, nilai PWM akan berkurang.



Gambar 3.4 perancangan pengatur suhu DHT11

3.3.3 Perancangan sistem pengatur kelembaban

Pada perancangan sistem pengatur kelembaban, terdapat dua komponen yaitu *step-up DC to DC* dan aktuator yang berupa *humidifier*. *Humidifier* ini membutuhkan input tegangan DC sebesar 24V. Mikrokontroler dihubungkan dengan *step up* yang akan mengkonversi tegangan keluaran arduino sebesar 5V menjadi 24V agar dapat menjalankan *humidifier*.



Gambar 3.5 perancangan pengatur kelembaban DHT11

3.4 Perancangan *Software* inkubator berbasis IoT

Perancangan software meliputi daftar program untuk sensor dan aktuator dari sistem inkubator.

3.4.1 Perancangan Program sensor dan aktuator suhu DHT11

Program sensor suhu DHT11 meliputi pemasangan *library* dari sensor DHT11, pengaturan aktuator sesuai dengan pembacaan sensor dan penampilan data pembacaan ke serial monitor pada Arduino IDE. Arduino yang terhubung pada relay yang disambung ke aktuator yang berupa kipas DC diatur sehingga kipas akan menyala jika pembacaan suhu melebihi nilai maksimum yang ditentukan dan akan mati bila pembacaan suhu kurang dari nilai minimum yang ditentukan.

```

#include <dht11.h>
#include <dht.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#define dht_apin A0 // Analog Pin sensor is connected to
dht DHT;]
void setup(){
  pinMode(4, OUTPUT);
  pinMode(50, OUTPUT);
  Serial.begin(115200);
  delay(1000);//Wait before accessing Sensor
} //end "setup()"

void loop(){
  DHT.read11(dht_apin);
  float a = -1;
  Serial.print("% ");
  Serial.print("temperature = ");
  Serial.print(DHT.temperature);
  Serial.println("C ");
  Serial.println();
  if (DHT.temperature > 40){
    digitalWrite(50, HIGH);}
  else if (DHT.temperature <= 40 ){
    digitalWrite(50, LOW);}
  delay(2000);
} // end loop()

```

Gambar 3.6 Program pengaturan suhu melalui DHT11

Terdapat satu output pada program yaitu pin digital ke 50, pin ini tersambung ke input relay untuk mengatur agar relay menyambung power supply ke kipas DC jika pembacaan suhu kerang dari atau sama dengan 40 derajat dan akan memutus sambungan bila pembacaannya melebihi 40 derajat.

3.4.2 Perancangan Program sensor dan aktuator kelembaban DHT11

Program sensor kelembaban DHT11 pengaturan aktuator sesuai dengan pembacaan sensor dan penampilan data pembacaan ke serial monitor pada Arduino IDE. Arduino yang terhubung pada *humidifier* diatur sehingga *humidifier* akan menyala jika pembacaan kelembaban

melebihi nilai maksimum yang ditentukan dan akan mati bila pembacaan suhu kurang dari nilai minimum yang ditentukan.

Gambar 3.6 Program pengaturan kelembaban melalui DHT11

```
#include <dht11.h>
#include <dht.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#define dht_apin A0 // Analog Pin sensor is connected to
dht DHT;
void setup(){
  pinMode(4, OUTPUT);
  pinMode(50, OUTPUT);
  Serial.begin(115200);
  delay(1000); //Wait before accessing Sensor
} //end "setup()"

void loop(){
  if (DHT.humidity < 70){
    digitalWrite(4,HIGH);}
  else if (DHT.humidity > 70 ){
    digitalWrite(4,LOW);}
  Serial.print("Current humidity = ");
  Serial.print(DHT.humidity);
  Serial.print("% ");
  Serial.println();
  delay(2000);

  //Fastest should be once every two seconds.
} // end loop()
```

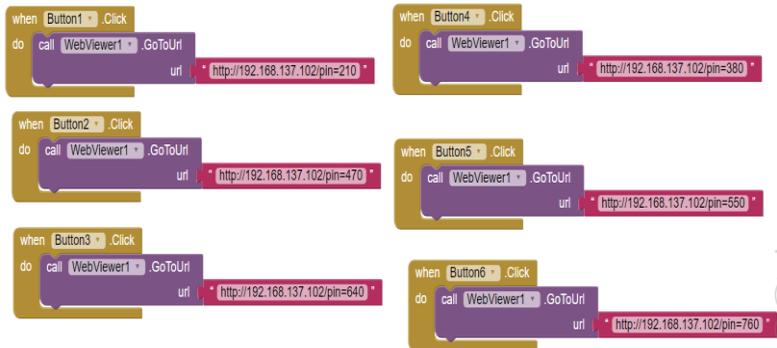
Gambar 3.7 Program pengaturan kelembaban

Terdapat satu output berupa pin digital nomor 4 yang berfungsi untuk mengeluarkan tegangan *high* atau *low* tergantung dari pembacaan kelembaban dari sensor DHT11, jika pembacaannya lebih dari 70, maka *humidifier* akan mati, dan jika pembacaannya kurang dari 70, maka *humidifier* akan menyala.

3.4.3 Perancangan Program aplikasi pengaturan set point

Pada sistem inkubator berbasis IoT ini, nilai set point dapat diatur melalui aplikasi pada android yang dapat merubah set point sesuai dengan input yang diberikan, nilai set point yang diberikan akan bersifat tetap sampai input set point yang selanjutnya diberikan, aplikasi ini

dibuat dengan bantuan app inventor milik MIT dengan blok program sebagai berikut.



Gambar 3.8 Tampilan blok program pada app inventor

Pengaturan set point inkubator

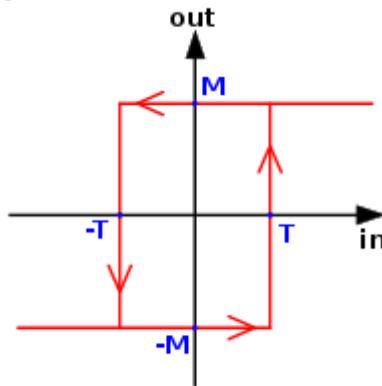
35	36
37	38
39	40

Gambar 3.9 Interface pada aplikasi android

Pada blok program diatas, dapat dilihat, pada setiap tombol diatur agar mengirim data yang nantinya akan dikonversikan menjadi nilai sesuai dengan tombol masing-masing.

3.5 Perancangan sistem pengatur suhu inkubator berbasis IoT

Sistem pengatur suhu pada inkubator berbasis IoT ini adalah dengan menggunakan kontrol histerisis, dimana nilai *setting point* yang sudah diatur akan menjadi titik tengah untuk kontrol histerisis, pada program, akan diatur pada batas atas sebesar nilai *setting point* ditambah dengan 2, sedangkan untuk batas bawahnya sebesar nilai *setting point* dikurangi dengan 2, hal ini agar menghindari sistem on/off pada relay tidak berganti secara cepat.



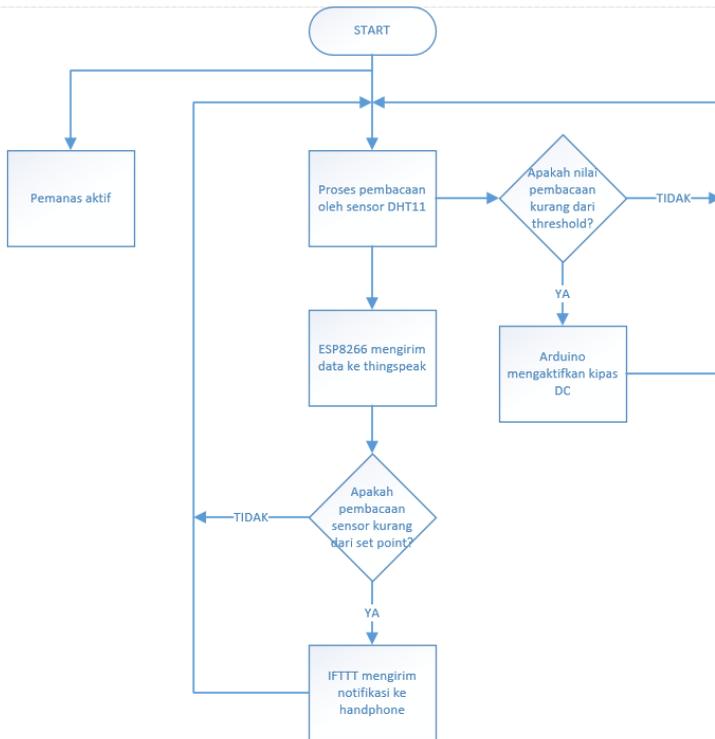
Gambar 3.10 kontrol histerisis

```
delay(500);  
  
if (temperature > (setpoint+2)){  
    digitalWrite(50, HIGH);}  
else if (temperature <= (setpoint-2)){  
    digitalWrite(50, LOW);};
```

Gambar 3.9 program kontrol histerisis

3.6 Diagram alir sistem inkubator berbasis IoT

Diagram alir sistem dimulai dari awal mengaktifkan alat, lalu sensor membaca data yang ada di dalam inkubator, jika data yang dibaca sudah sesuai, maka data akan tetap dibaca secara kontinyu, jika tidak, maka aktuator akan bekerja sehingga variabel yang dibaca dapat sesuai dengan data yang diinginkan, hasil pembacaan yang dilakukan secara kontinyu ini akan dikirim ke web server melalui koneksi internet agar dapat dibaca dari *device* orang lain menggunakan media seperti *handphone* atau komputer.



Gambar 3.10 Diagram alir sistem

3.7 Spesifikasi Inkubator berbasis IoT

Tegangan input	x 220 V AC x 12-24 V DC
Dimensi	70cm x 40cm x 15 cm
Sensor suhu dan kelembaban	DHT11
Aktuator kelembaban	Ultrasonic mist maker
Aktuator pemanas	Heater Tiup 220V 100w eleme pemanas udara
kontroler	Arduino MEGA
Tampilan layar	<i>Liquid Crystal Display 16 x 2</i>

-----Halaman ini sengaja dikosongkan-----

BAB IV PENGUJIAN

Pada bab ini dibahas mengenai pengujian inkubator berupa pengujian sensor DHT11, pengujian DHT11 yang telah disambung dengan aktuator, dan pengujian sistem secara keseluruhan.

4.1 Pengujian sensor DHT11

Pengujian sensor DHT11 dilakukan dengan meletakkan sensor pada ruangan dan sensor akan membaca suhu dan kelembaban pada ruangan tersebut setiap 2 detik.

```
Current humidity = 50.00%  temperature = 36.00C
Current humidity = 51.00%  temperature = 36.00C
```

Gambar 4.1 Hasil pengambilan data selama 40 detik

Tabel 4.1 data pembacaan sensor DHT11 selama 40 detik

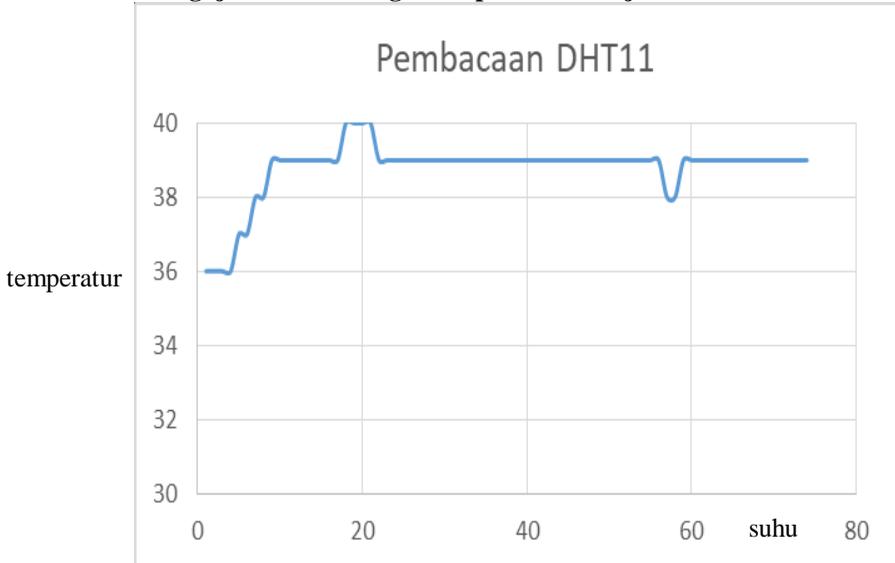
<i>Waktu</i>	<i>Kelembaban</i>	<i>Suhu</i>
1	50	36
2	50	36
3	50	36
4	50	36
5	50	36
6	51	36
7	51	36
8	51	36
9	51	36
10	51	36
11	51	36
12	51	36
13	51	36
14	51	36
15	51	36
16	51	36
17	51	36
18	51	36
19	51	36
20	51	36

Dari data tabel 4.1 dapat dilihat bahwa hasil pembacaan dari sensor dalam suhu ruangan selama 40 detik hampir konstan atau tidak berubah untuk kelembaban, dan untuk suhu, nilai pembacaannya tetap sebesar 36 derajat.

4.2 Pengujian sensor DHT11 dengan aktuator

Pengujian sensor DHT dengan aktuator yaitu menguji pembacaan sensor dari suhu ruangan dan dipanaskan, lalu set point diatur sebesar nilai tertentu untuk menjaga kestabilan suhu agar tetap pada set point yang ditentukan.

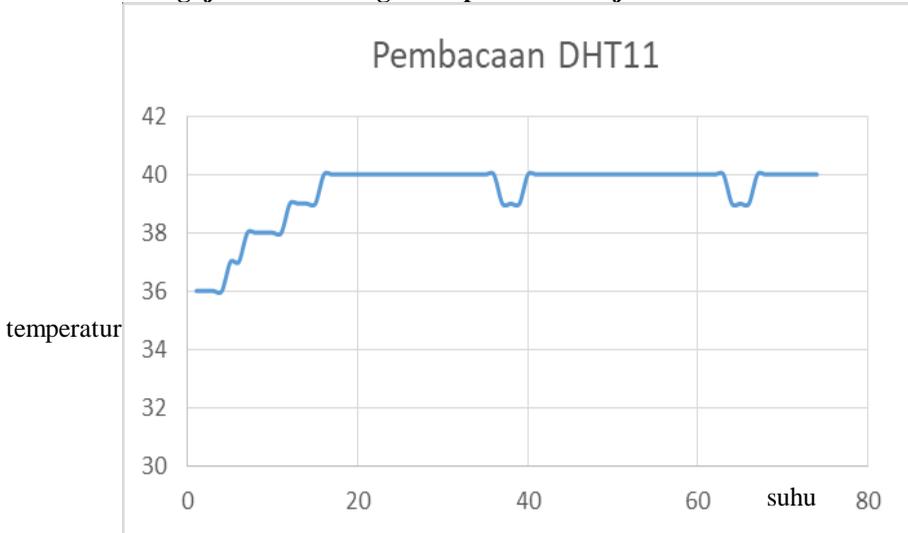
4.2.1 Pengujian sensor dengan set point 38 derajat



Gambar 4.2 Grafik pembacaan sensor DHT11 dengan aktuator menggunakan set point 38 derajat

Dari data grafik diatas, dapat dilihat jika set point diatur sebesar 38 derajat, pembacaan suhu yang mula-mula bernilai 36 derajat, akan naik hingga 40 derajat sebelum stabil di 39 derajat, jika suhu turun ke 38 derajat, maka kipas akan menyala dan menaikkan suhu menjadi 39 derajat kembali.

4.2.2 Pengujian sensor dengan set point 40 derajat



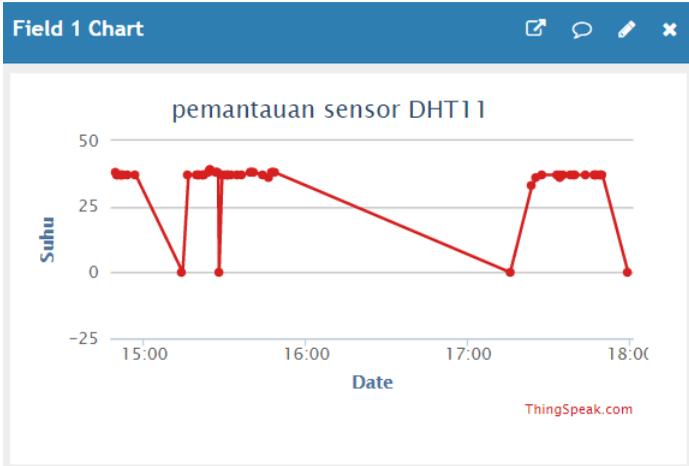
Gambar 4.3 Grafik pembacaan sensor DHT11 dengan aktuator menggunakan set point 40 derajat

Dari data grafik diatas, dapat dilihat jika set point diatur sebesar 40 derajat, pembacaan suhu yang mula-mula bernilai 36 derajat, akan naik hingga 40 derajat sebelum stabil, jika suhu turun ke 39 derajat, maka kipas akan menyala dan menaikkan suhu menjadi 40 derajat kembali.

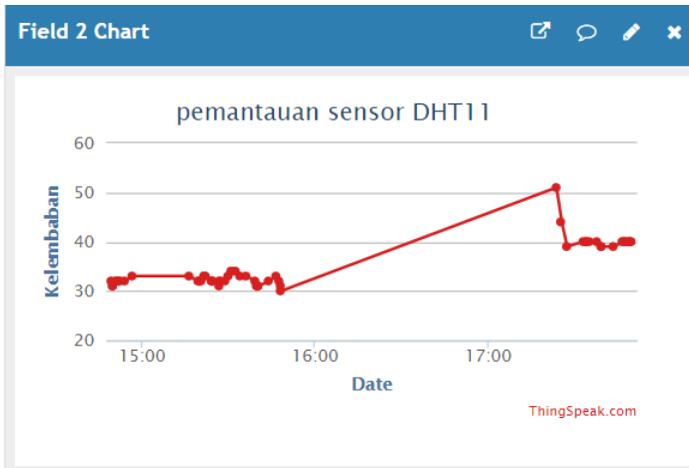
4.3 Pengujian pembacaan sensor pada thingspeak

Pada saat sensor sudah membaca suhu dan kelembaban, data pembacaan sensor akan dikirimkan ke web thingspeak yang menyediakan *database* untuk pengguna agar data dapat dipantau, pada thingspeak akan ditampilkan besar data dan waktu pengiriman, yang diupdate secara berkala.

Pada thingspeak yang dipakai pada tugas akhir ini, ditampilkan hasil pembacaan dari sensor DHT11 berupa suhu dan kelembaban, yang dapat dilihat oleh semua orang yang mengakses tautan dari *channel* yang sudah diberikan.



Gambar 4.4 Grafik pembacaan suhu sensor DHT11 pada thingspeak



Gambar 4.5 Grafik pembacaan kelembaban sensor DHT11 pada thingspeak

4.4 Pengujian notifikasi pada *Handphone*

Jika pembacaan suhu pada thingspeak besarnya lebih kecil dari set point yang ditentukan, maka notifikasi dan sms akan dikirimkan pada handphone yang terintegrasi dengan thingspeak melalui aplikasi IFTTT, notifikasi ini akan dikirim setiap ada data pengiriman untuk suhu yang nilainya dibawah set point yang ditentukan.

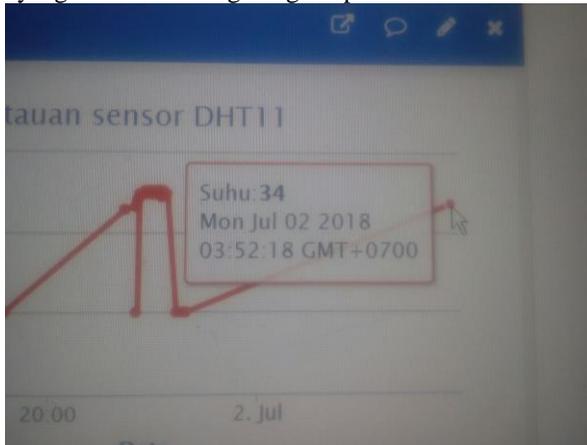


Gambar 4.6 Gambar notifikasi pada handphone

4.5 Pengujian inkubator berbasis IoT secara keseluruhan

Pengujian secara keseluruhan dilakukan dengan mengamati suhu ruangan terlebih dahulu untuk melihat suhu awal yang terukur, lalu alat dinyalakan, setelah alat dinyalakan, relay akan menyala dan karena suhu masih dibawah threshold yang diatur, kipas akan menyala hingga suhu melebihi threshold yang ada, dan mematikan kipas. Hasil pembacaan dari suhu dan kelembaban akan dikirim ke web thingspeak, jika nilai dari suhu kurang atau lebih dari set point yang ditentukan (30 untuk batas bawah, dan 45 untuk batas atas) maka thingspeak akan mentrigger

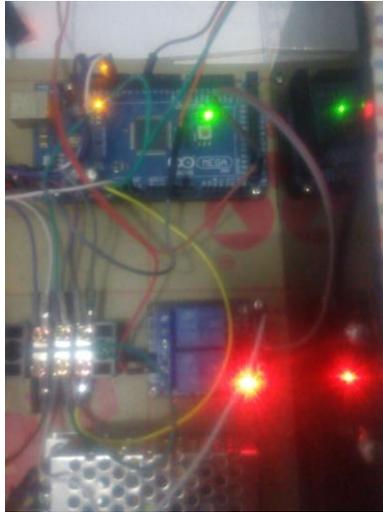
web yang telah diintegrasikan dengan IFTTT yang akan mengirim notifikasi ke HP yang telah terhubung dengan aplikasi IFTTT.



Gambar 4.7 Pembacaan suhu saat alat pertama dinyalakan



Gambar 4.8 pembacaan kelembaban saat alat pertama dinyalakan

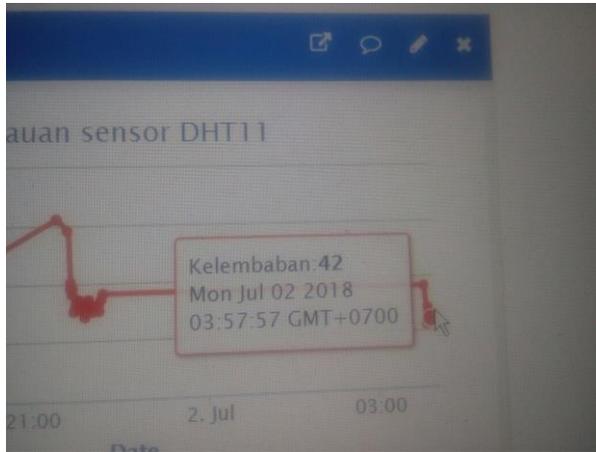


Gambar 4.9 relay menyala saat alat pertama dinyalakan

Setelah dilihat data suhu dan kelembaban awal, maka alat dibiarkan selama beberapa saat hingga relay terbuka dan kipas DC berhenti berfungsi menandakan suhu yang dibaca sudah melebihi threshold yang diberikan.



Gambar 4.10 Pembacaan suhu saat nilai pembacaan melebihi threshold



Gambar 4.11 Pembacaan kelembaban saat nilai suhu melebihi threshold

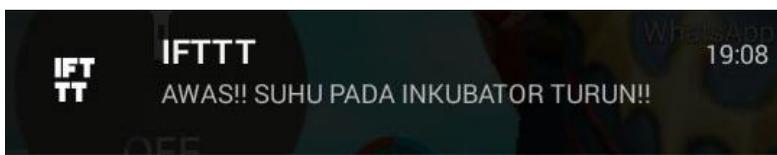


Gambar 4.12 Relay mati saat nilai suhu melebihi threshold

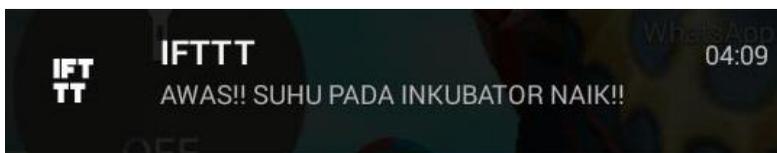
Setelah pengujian selesai, nilai dari data pada thingspeak ditrigger agar dapat bernilai 0 atau 100 untuk menguji sistem notifikasi yang ada pada inkubator, karena jika hasil pembacaan kurang atau lebih dari threshold yang telah diberikan, maka notifikasi akan muncul padad handphone pengguna.



Gambar 4.13 Grafik suhu saat diittrigger dengan nilai 0 untuk suhu.



Gambar 4.14 Notifikasi bila suhu dibawah nilai threshold bawah



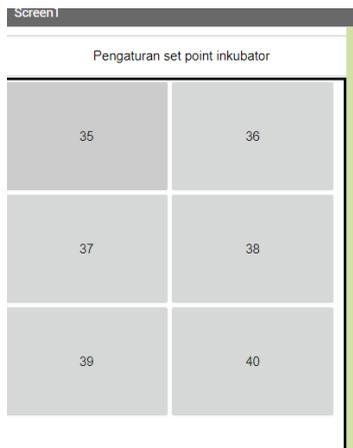
Gambar 4.15 Notifikasi bila suhu diatas nilai threshold atas

Hasil pembacaan suhu dan kelembaban dari sensor akan ditampilkan pada lcd beserta dengan set point yang berlaku, jika pembacaan berubah, maka tampilan pada lcd akan turut berubah, begitu pula jika set point dirubah, maka tampilan lcd juga akan berubah.



Gambar 4.16 tampilan pembacaan suhu kelembaban dan set point

Set point pada inkubator dapat diatur melalui aplikasi pada handphone yang terdiri dari pilihan set point yang bisa ditekan untuk mengatur set point suhu yang diaplikasikan pada inkubator.



Gambar 4.17 interface pengaturan set point pada handphone

Jika salah satu tombol ditekan pada saat lcd bertuliskan input, maka set point pada inkubator akan berubah sesuai dengan input yang diberikan, set point ini akan terus aktif hingga pergantian set point yang selanjutnya.



Gambar 4.18 tampilan lcd setelah set point dirubah ke 40

Tabel 4.2 hasil masukan set point pada inkubator melalui *handphone*

Input Set Point	Hasil pembacaan LCD
35	35
36	36
37	37
38	38
39	39
40	40
39	40
38	38
37	37
36	36

Dari hasil masukan setting point melalui aplikasi android pada *handphone* ke inkubator yang diterima oleh esp8266, dari 9 percobaan sebanyak 10 kali, didapatkan keberhasilan sebanyak 9 kali atau 90% dari total percobaan yang dilakukan.

BAB V

PENUTUP

Hasil dari pengujian sensor suhu untuk inkubator berbasis IoT menggunakan arduino mega dan esp8266, didapatkan bahwa inkubator telah berfungsi dan mencapai target yang diinginkan, yaitu stabil pada suhu atau set point yang ditentukan, dan telah dapat mengirim data hasil pembacaan sensor ke web server secara berkala. Nilai dari set point juga bisa diatur melalui aplikasi pada android dengan hasil pembacaan dan nilai set point ditampilkan pada lcd dengan keberhasilan sebesar 90%.

Saran untuk penelitian selanjutnya adalah menambahkan jumlah sensor dan aktuator dari yang telah ada, untuk mendapatkan hasil yang lebih maksimal.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] “Mari Mengenal Apa itu Internet of Thing (IoT) | IDCloudHost.” [Online]. Available: <https://idcloudhost.com/mari-mengenal-apa-itu-internet-thing-iot/>. [Accessed: 03-Mar-2018].
- [2] “Pengertian Web Server dan Fungsinya | IDCloudHost.” [Online]. Available: <https://idcloudhost.com/pengertian-web-server-dan-fungsinya/>. [Accessed: 03-Jun-2018].
- [3] S. Sahoo, B. Champaty, K. Pal, S. S. Ray, and D. N. Tibarewala, “Wireless transmission of alarm signals from baby incubators to neonatal nursing station,” in *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*, 2014, pp. 1–5.
- [4] “Inkubator Bayi Mendukung Kelangsungan Hidup Bayi | Medicalogy.” [Online]. Available: <https://www.medicalogy.com/blog/inkubator-bayi-mendukung-kelangsungan-hidup-bayi/>. [Accessed: 08-Mar-2018].
- [5] “Pengertian Arduino Adalah,” *Saptaji.com*, 11-Nov-2016. [Online]. Available: <http://saptaji.com/2016/11/11/pengertian-arduino-adalah/>. [Accessed: 12-Apr-2018].
- [6] “Mengukur Suhu dan Kelembaban Udara dengan Sensor DHT11 dan Arduino,” *Saptaji.com*, 10-Aug-2016. [Online]. Available: <http://saptaji.com/2016/08/10/mengukur-suhu-dan-kelembaban-udara-dengan-sensor-dht11-dan-arduino/>. [Accessed: 01-Mar-2018].
- [7] “Mengenal Arduino Software (IDE) – SinauArduino.”
- [8] “Pengertian dan Keunggulan Framework Laravel | IDCloudHost.” [Online]. Available: <https://idcloudhost.com/pengertian-dan-keunggulan-framework-laravel/>. [Accessed: 03-Mar-2018].

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1. Program keseluruhan sensor DH11 dan aktuator

```
#include "DHT.h"
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#define DEBUG true
SoftwareSerial wifi(10,11); // RX, TX
const int rs = 8, en = 7, d4 = 6, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define DHTPIN 5
#define DHTTYPE DHT11
int spoint;
DHT dht(DHTPIN, DHTTYPE);
String apiKey = "1J38BQFUAWFRPAW1";
String Host_Name = "Connectify-me";
String Password = "jalansakura";

void setup() {
  Serial.begin(115200);
  delay(500);
  wifi.begin(115200);
  delay(500);
  pinMode(50, OUTPUT);
  pinMode(52, OUTPUT);
  // pinMode(10,OUTPUT);
  // digitalWrite(10,HIGH);
  // pinMode(11,OUTPUT);
  // digitalWrite(11,HIGH);
  pinMode(12,OUTPUT);
  digitalWrite(12,HIGH);
  pinMode(13,OUTPUT);
  digitalWrite(13,HIGH);
  dht.begin();
  lcd.begin(16, 2);
  /*  wifi.println("AT+RST");
  wifi.println("AT+CWMODE=3");
  char inv ="";
```

```

String cmd = "AT+CWJAP";
    cmd+= "=";
    cmd+= inv;
    cmd+= Host_Name;
    cmd+= inv;
    cmd+= ",";
    cmd+= inv;
    cmd+= Password;
    cmd+= inv;
wifi.println(cmd);
wifi.println("AT+CIFSR");
wifi.println("AT+CIPMUX=1"); */
/* // Reset modul
    sendCommand("AT+RST\r\n",2000,DEBUG);
    // Konfigurasi sebagai Akses poin
    sendCommand("AT+CWMODE=3\r\n",1000,DEBUG);
    // Sesuaikan dengan SSID dan Password
    sendCommand("AT+CWJAP=\"Connectify-
me\", \"jalansakura\" \r\n",10000,DEBUG);
    // Konfigurasi untuk multiple connection
    sendCommand("AT+CIPMUX=1\r\n",1000,DEBUG);
    // Aktifkan server pada port 80
    sendCommand("AT+CIPSERVER=1,80\r\n",1000,DEBUG);
    Serial.println("Server sudah siap!");
    delay(1000);*/ }

void loop() {
    // Reset modul
    sendCommand("AT+RST\r\n",2000,DEBUG);
    // Konfigurasi sebagai Akses poin
    sendCommand("AT+CWMODE=3\r\n",1000,DEBUG);
    // Sesuaikan dengan SSID dan Password
    sendCommand("AT+CWJAP=\"Connectify-
me\", \"jalansakura\" \r\n",10000,DEBUG);
    // Konfigurasi untuk multiple connection
    sendCommand("AT+CIPMUX=1\r\n",1000,DEBUG);

    // Aktifkan server pada port 80
    sendCommand("AT+CIPSERVER=1,80\r\n",1000,DEBUG);

```

```

delay(2000);
    lcd.clear();
    lcd.print("input");
    delay(2000);
    if (wifi.find("pin=")){
    int pinNumber = (wifi.read());
    //pinNumber += (wifi.read());
    Serial.println();
    Serial.println("tes2");
    Serial.println(pinNumber);
    spoint=pinNumber;
    Serial.println();
    }
    Serial.println("tes3");
    Serial.println(spoint);
delay(2000);
spoint=spoint;

//sendCommand("AT+RST\r\n",2000,DEBUG);
// SoftwareSerial wifi(10,11); // RX, TX
// wifi.begin(115200);
// delay(500);
    wifi.println("AT+RST");
    wifi.println("AT+CWMODE=3");
    char inv = "";
String cmd = "AT+CWJAP";
    cmd+= "=";
    cmd+= inv;
    cmd+= Host_Name;
    cmd+= inv;
    cmd+= ",";
    cmd+= inv;
    cmd+= Password;
    cmd+= inv;
wifi.println(cmd);
wifi.println("AT+CIFSR");
wifi.println("AT+CIPMUX=1");
delay(3000);
int humidity = dht.readHumidity();

```

```

int temperature = dht.readTemperature();
String state1=String(temperature);
String state2=String(humidity);
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("s");
lcd.setCursor(5, 0);
lcd.print("k");
lcd.setCursor(9, 0);
lcd.print("s.point");
lcd.setCursor(1, 1);
lcd.print(temperature);
lcd.setCursor(5, 1);
lcd.print(humidity);
lcd.setCursor(10, 1);
lcd.print(spoint-15);
delay(2000);

```

```

Serial.println("=====
=====");
    Serial.println("");
    String cmd2 = "AT+CIPSTART=\"TCP\", \"";
    cmd2 += "184.106.153.149";
    cmd2 += "\",80";
    wifi.println(cmd2);
    Serial.println(cmd2);
    Serial.println(temperature);
    Serial.println(humidity);
    Serial.println(spoint);

    if(wifi.find("Error")){
        Serial.println("AT+CIPSTART error");
        return;
    }

    String getStr = "GET /update?api_key=";
    getStr += apiKey;
    getStr += "&field1=";
    getStr += String(temperature);

```

```

getStr += "&field2=";
getStr += String(humidity);
getStr += "\r\n\r\n";

cmd2 = "AT+CIPSEND=";
cmd2 += String(getStr.length());
wifi.println(cmd2);
lcd.clear();
lcd.print("sending");
Serial.println(cmd2);
if(wifi.find(">")){
  wifi.println(getStr);
  Serial.println(getStr);
  wifi.println("AT+CIPCLOSE=5");
  // alert user
  Serial.println("AT+CIPCLOSE");

Serial.println("=====
=====");
  Serial.println("");
  //lcd.clear();
}
else{
  wifi.println("AT+CIPCLOSE=5");
  // alert user
  Serial.println("AT+CIPCLOSE");

Serial.println("=====
=====");
  Serial.println("");
}
delay(500);

if (temperature > 36){
  digitalWrite(50, HIGH);}
else if (temperature <= 35 ){
  digitalWrite(50, LOW);}
if (humidity > 70){
  digitalWrite(52, LOW);}

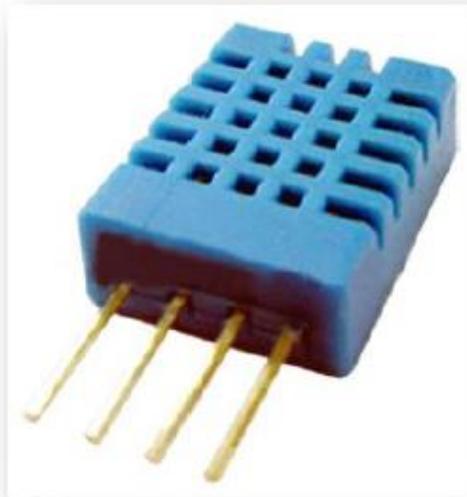
```

```
else if (humidity <= 70 ){  
    digitalWrite(52, HIGH);}  
delay(1000);
```

```
}
```

```
String sendCommand(String command, const int timeout, boolean  
debug)  
{  
    String response = "";  
    wifi.print(command);  
    long int time = millis();  
  
    while( (time+timeout) > millis())  
    {  
        while(wifi.available())  
        {  
            char c = wifi.read();  
            response+=c;  
        }  
    }  
  
    if(debug) {  
        Serial.print(response);  
    }  
    return response;  
}
```

A.2 Datasheet DHT11



2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	±5%RH	±2°C	1	4 Pin Single Row

Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			± 1%RH	
Accuracy	25°C		± 4%RH	
	0-50°C			± 5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
Temperature				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			± 1°C	
Accuracy		± 1°C		± 2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

3. Typical Application (Figure 1)

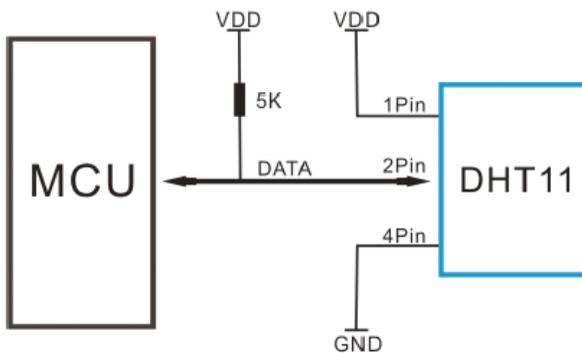
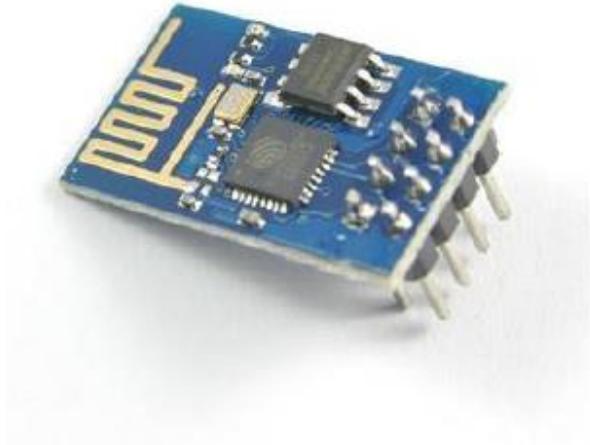


Figure 1 Typical Application

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

A.3 Datasheet ESP8266-01



Com mand s	Descri ption	Ty pe	Set/Execute	Inquir y	test	Param eters	Examples
AT+RST	restart the module	basic	-	-	-	-	
AT+CWMODE	wifi mode	wifi	AT+CWMODE=<mode>	AT+CWMODE?	AT+CWMODE=?	1= Sta, 2= AP, 3=both	
AT+CWJAP	join the AP	wifi	AT+ CWJAP =<ssid>,<pwd >	AT+ CWJAP?	-	ssid = ssid, pwd = wifi	

Mode	Min	Typical	Max	Units
802.11b, CCK 1Mbps, POUT=+19.5dBm		215		mA
802.11b, CCK 11Mbps, POUT=+18.5dBm		197		mA
802.11g, OFDM 54Mbps, POUT=+16dBm		145		mA
802.11n, MCS7, POUT=+14dBm		135		mA
802.11b, packet size of 1024 bytes, -80dBm		60		mA
802.11b, packet size of 1024 bytes, -70dBm		60		mA
802.11b, packet size of 1024 bytes, -65dBm		62		mA
Standby		0.9		uA
Deep sleep		10		mA
Saving mode DTIM 1		1.2		mA
Saving mode DTIM 3		0.86		mA
Shutdown		0.5		uA

RF specifications

The following data is at room temperature, the voltage of 3.3V and 1.1V, respectively, when measured

Description	Min	Typical	Max	Units
Input Frequency	2412		2484	MHz
Input resistance		50		Ω
Input reflection			-10	dB
At 72.2Mbps, PA output power	14	15	16	dBm
11b mode, PA output power	17.5	18.5	19.5	dBm
Sensitivity				
CCK, 1Mbps $\frac{1}{2}$ BPSK		-98		dBm
CCK, 11Mbps $\frac{1}{2}$ BPSK		-91		dBm
6Mbps (1/2 BPSK) $\frac{3}{4}$ DSSS		-93		dBm
54Mbps (3/4 64-QAM) $\frac{3}{4}$ DSSS		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps) $\frac{3}{4}$ DSSS		-71		dBm
Adjacent suppression				
OFDM, 6Mbps		37		dB

Command	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
						password	
AT+CWLAP	list the AP	wifi	AT+CWLAP				
AT+CWQAP	quit the AP	wifi	AT+CWQAP	-	AT+CWQAP=?		
AT+CWSAP	set the parameters of AP	wifi	AT+CWSAP=<ssid>,<pwd>,<chl>,<ecn>	AT+CWSAP?		ssid, pwd, chl = channel, ecn = encryption	Connect to your router: : AT+CWJAP="YOURSSID", "helloworld"; and check if connected: AT+CWJAP?
AT+CIPSTATUS	get the connection status	TCP/IP	AT+CIPSTATUS				
AT+CIPSTART	set up TCP or UDP connection	TCP/IP	1)single connection (+CIPMUX=0) AT+CIPSTART=<type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART=<id><type>,<addr>,<port>	-	AT+CIPSTART=?	id = 0-4, type = TCP/UDP , addr = IP address, port= port	Connect to another TCP server, set multiple connection first: AT+CIPMUX=1; connect: AT+CIPSTART=4,"TCP","X1.X2.X3.X4",9999
AT+CIPSEND	send data	TCP/IP	1)single connection(+CIPMUX=0) AT+CIPSEND=<length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND=		AT+CIPSEND=?		send data: AT+CIPSEND=4,15 and then enter the data

Command	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
AT+CIPCLOSE	close TCP or UDP connection	TCP/IP	<id>,<length> AT+CIPCLOSE= <id> or AT+CIPCLOSE			AT+CIPCLOSE=?	
AT+CIFSR	Get IP address	TCP/IP	AT+CIFSR			AT+CIFSR=?	
AT+CIPMUX	set multiple connection	TCP/IP	AT+CIPMUX=<mode>	AT+CIPMUX?			0 for single connection 1 for multiple connection
AT+CIPSERVER	set as server	TCP/IP	AT+CIPSERVER= <mode>[,<port>]				mode 0 to close server mode, mode 1 to open; port = port turn on as a TCP server: AT+CIPSERVER=1,8888, check the self server IP address: AT+CIFSR=?
+IPD	received data						



RIWAYAT HIDUP

Andre Perwiratama lahir di Kediri pada tanggal 26 Maret 1996. Menyelesaikan pendidikannya di SDN Pare I (2001-2007), SMP Negeri 2 Pare (2007-2010), SMA Negeri 2 Pare (2010-2013) dan D3 Teknik Elektro ITS Surabaya (2013-2016). Pada tahun 2016 melanjutkan pendidikan lintas jalur di Institut Teknologi Sepuluh Nopember (ITS), Departemen Teknik Elektro. Di Departemen Teknik Elektro ini, kemudian memilih bidang elektronika sebagai bidang studi yang ditekuni. Pada

bulan Juni 2018, mengikuti seminar tugas akhir dan pada Juli 2018 mengikuti ujian tugas akhir sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik.

-----Halaman ini sengaja dikosongkan-----