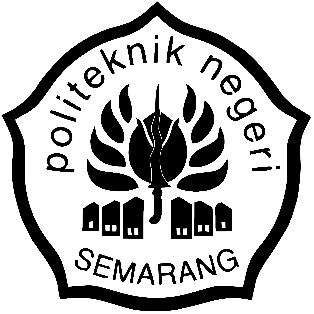
**KAPILAR : SISTEM KONTROL SUHU DAN PEMBERI PAKAN OTOMATIS KANDANG AYAM PETELUR BERBASIS *INTERNET OF THINGS***



Disusun Oleh :

Faradheza Haryundhina Larasati 3.34.16.1.09

Fathia Fidiniari 3.34.16.1.10

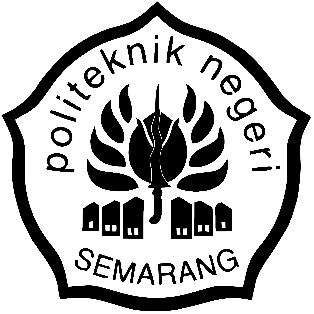
**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI SEMARANG**

**2019**

# KAPILAR : SISTEM KONTROL SUHU DAN PEMBERI PAKAN OTOMATIS KANDANG AYAM PETELUR BERBASIS INTERNET OF THINGS



Tugas akhir/skripsi ini disusun untuk melengkapi sebagian persyaratan menjadi Ahli Madya/Sarjana Terapan

Disusun Oleh :

Faradheza Haryundhina Larasati 3.34.16.1.09

Fathia Fidiniari 3.34.16.1.10

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI SEMARANG**

**2019**

# **PERNYATAAN KEASLIAN TUGAS AKHIR**

Menyatakan bahwa Tugas Akhir dengan judul **KAPILAR : SISTEM KONTROL SUHU DAN PEMBERI PAKAN OTOMATIS KANDANG AYAM PETELUR BERBASIS *INTERNET OF THINGS*** yang dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya pada Program Studi Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Semarang, bukan merupakan tiruan atau duplikasi dari Tugas Akhir yang sudah dipublikasikan dan pernah dipakai untuk mendapatkan gelar Ahli Madya di lingkungan Politeknik Negeri Semarang maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Semarang, Agustus 2019

Mahasiswa Mahasiswa

Faradheza Haryundhina Larasati Fathia Fidiniari

NIM. 3.34.16.1.09 NIM. 3.34.16.1.10

# **HALAMAN PERSETUJUAN**

Tugas akhir dengan judul **KAPILAR : SISTEM KONTROL SUHU DAN PEMBERI PAKAN OTOMATIS KANDANG AYAM PETELUR BERBASIS *INTERNET OF THINGS*** dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya pada Program Studi D3-Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Semarang dan disetujui untuk diajukan dalam sidang ujian tugas akhir.

Semarang, Agustus 2019

Pembimbing I, Pembimbing II,

Slamet Handoko, S.Kom.,M.Kom Liliek Triyono, S.T., M.Kom

NIP 197501302001121001 NIP 198404202015041003

Mengetahui,

Ketua Program Studi Teknik Informatika

Sukamto, S.Kom., M.T.

NIP 197101172003121001

# **HALAMAN PENGESAHAN**

Tugas akhir/skripsi dengan judul **KAPILAR : SISTEM KONTROL SUHU DAN PEMBERI PAKAN OTOMATIS KANDANG AYAM PETELUR BERBASIS *INTERNET OF THINGS*** Telah dipertahankan dalam ujian wawancara dan diterima sebagai syarat untuk menjadi Ahli Madya/Sarjana Terapan pada Program Studi D3 – Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang pada tanggal Agustus 2019.

Tim Penguji

Penguji I, Penguji II, Penguji III,

Nama Nama Nama

NIP. NIP. NIP.

Ketua Penguji, Sekretaris,

Slamet Handoko, S.Kom.,M.Kom Nama

NIP 197501302001121001 NIP.

Mengesahkan,

Ketua Jurusan Teknik Informatika

Dr.Amin Suharjono, S.T,M.T.

NIP 197210271999031002

# KATA PENGANTAR

Syukur Alhamdulillah penulis ucapkan kepada Allah SWT yang telah melimpahkan rahmat dan dihayah-Nya kepada penulis sehingga Tugas Akhir (TA) ini dapat terselesaikan dengan baik. TA ini dimaksudkan untuk memenuhi syarat kelulusan Program Diploma III Jurusan Teknik Elektro Program Studi Teknik Informatika Politeknik Negeri Semarang.

Selama menyelesaikan TA ini penulis banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Oleh karena itu penulis ingin menyampaikan ucapan terima kasih kepada :

1. Orang tua dan keluarga yang telah memberikan do’a, motivasi, bantuan, dan dukungan.
2. Bapak Ir. Supriyadi, M.T selaku Direktur Politeknik Negeri Semarang.
3. Bapak Dr. Amin Suharjono, S.T, M.T. selaku Ketua Jurusan Teknik Elektro.
4. Bapak Sukamto, S.Kom., M.T. selaku Ketua Program Studi Teknik Informatika.
5. Bapak Budi Suyanto, S.T., M.Eng. selaku Dosen Wali.
6. Bapak Wahyu Sulistiyo, S.T., M.Kom., selaku Pembimbing 1.
7. Bapak Slamet Handoko, S.Kom., M.Kom., selaku Pembimbing 2.
8. Teman-teman Program Studi Teknik Informatika angkatan tahun 2015 yang telah menempuh pendidikan bersama-sama selama 3 tahun.
9. Seluruh pihak yang telah membantu penulis dalam menyelesaikan TA.

Penulis menyadari bahwa dalam pembuatan TA ini masih jauh dari kata sempurna, oleh karena itu kritik dan saran yang membangun dari seluruh pihak sangat penulis harapkan. Semoga TA ini bermanfaat bagi banyak pihak.

Semarang, Agustus 2019

Penulis

# *Abstrak*

# DAFTAR ISI

Halaman

[JUDUL **Error! Bookmark not defined.**](#_Toc4137489)

[PERNYATAAN KEASLIAN TUGAS AKHIR ii](#_Toc4137490)

[HALAMAN PERSETUJUAN iii](#_Toc4137491)

[HALAMAN PENGESAHAN iv](#_Toc4137492)

[KATA PENGANTAR v](#_Toc4137493)

[*Abstrak* vi](#_Toc4137494)

[DAFTAR ISI vii](#_Toc4137495)

[DAFTAR GAMBAR ix](#_Toc4137496)

[DAFTAR TABEL x](#_Toc4137497)

[DAFTAR LAMPIRAN xi](#_Toc4137498)

[BAB I PENDAHULUAN 1](#_Toc4137499)

[1.1. Latar Belakang 1](#_Toc4137500)

[1.1.1. Perumusan Masalah 2](#_Toc4137501)

[1.1.2. Manfaat 2](#_Toc4137502)

[1.1.3. Tujuan 3](#_Toc4137503)

[1.2. Batasan Masalah 3](#_Toc4137504)

[1.3. Sistematika Penulisan 3](#_Toc4137505)

[BAB II TINJAUAN PUSTAKA 5](#_Toc4137506)

[BAB III PELAKSANAAN KEGIATAN 15](#_Toc4137507)

[BAB IV ANALISIS DAN PEMBAHASAN 51](#_Toc4137508)

[BAB V KESIMPULAN 52](#_Toc4137509)

[DAFTAR PUSTAKA 53](#_Toc4137510)

[LAMPIRAN 54](#_Toc4137511)

# DAFTAR GAMBAR

# DAFTAR TABEL

# DAFTAR LAMPIRAN

# BAB I PENDAHULUAN

## Latar Belakang

Peternakan ayam petelur merupakan salah satu peternakan yang berperan penting dalam memasok sumber protein hewani dan salah satu kebutuhan pokok dalam pemenuhan gizi masyarakat. Ada beberapa faktor yang harus dilakukan untuk mengelola peternakan ayam petelur yang mempengaruhi produksi telur diantaranya yaitu pemberian pakan dan air minum pada ayam, serta suhu dan kelembaban lingkungan. Setiap ayam petelur yang sedang dalam masa produksi dapat mengonsumsi pakan sekitar 100g – 200g per hari. Suhu dan kelembaban lingkungan pada kandang ayam juga perlu diperhatikan, suhu yang sesuai untuk ayam petelur adalah 21oC – 24oC. Apabila suhu kandang lebih tinggi dari suhu yang sesuai dapat menurunkan konsumsi pakan dan apabila suhu lebih rendah konsumsi pakan ayam akan meningkat (Setiawati, dkk, 2016). Sedangkan untuk kelembaban yang sesuai untuk ayam petelur berkisar antara 50 – 70%. Kelembaban tinggi akan meningkatkan kadar amoniak dan dapat menyebabkan gangguan pernapasan. Air minum juga berperan penting dalam produktivitas ayam petelur, kekurangan air minum pada ayam dapat menurunkan konsumsi pakan sehingga produktivitasnya akan menurun (Putri, Sukatana dan Partama, 2017). Dalam pengelolaannya para peternak biasa memberikan pakan satu kali sehari sebanyak 120g/ekor. Sedangkan untuk mengontrol suhu pada malam hari peternak menyalakan lampu. Namun, cara yang dilakukan peternak tersebut belum efektif.

Beberapa penelitian terkait efisiensi pengelolaan kandang sudah pernah dilakukan, seperti sistem pakan otomatis dan penerangan untuk kandang ayam petelur menggunakan mikrokontroler yang mampu memberi pakan otomatis dan kontrol lampu yang akan bekerja sesuai jadwal yang telah ditetapkan (Ridhamuttaqin, dkk, 2013; Warjono*,* dkk, 2018). Sistem kontrol pengatur suhu dan kelembaban otomatis pada kandang ayam broiler juga telah dibuat oleh Mansyur pada tahun 2018. Sistem ini menggunakan mikrokontroler Arduino Mega 2560 yang dikomunikasikan dengan Ethernet Shield. Masukan sistem ini adalah sensor suhu DHT11 dan menghasilkan keluaran yang ditampilkan secara realtime pada aplikasi android (Blynk). Keluaran juga dilengkapi dengan lampu dan kipas sebagai pengatur suhu dan kelembaban (Mansyur, 2018). Penelitian lain yang dilakukan oleh Laksono telah menggabungkan sistem pakan otomatis serta monitoring suhu dan kelembaban kandang. Sistem ini menggunakan mikrokontroler ATmega328, dan sensor photodioda untuk membaca keadaan tempat pakan ayam serta sensor suhu DHT11 sebagai masukan, sedangkan keluaran akan ditampilkan pada LCD (Laksono, 2017). Penelitian yang sudah dikembangkan tersebut masih belum menggunakan antarmuka yang dapat memantau suhu dan kelembaban kandang serta mengontrol pakan dari jarak jauh.

Laporan Tugas Akhir ini membahas tentang pembangunan Kandang Ayam Petelur Pintar “KAPILAR” yang mampu memberikan kendali jarak jauh berbasis *Internet of Things* yang terhubung ke *smartphone* android melalui jaringan internet *Wi-Fi*. Sistem ini akan mampu untuk me*monitoring* keadaan suhu dan kelembaban serta mampu memberi pakan otomatis dengan *setting* jadwal pada *smartphone* yang dilengkapi dengan sensor suhu dan kelembaban, ultrasonic, relay, kipas, lampu, dan motor servo dengan mikrokontroler NodeMCU. Sehingga peternakan dapat dikelola dengan efisien dan peternak ayam dapat mengontrol dan memonitoring kandang ayam dengan kendali jarak jauh secara online melalui smartphone android.

### Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada laporan TA ini adalah masih sulitnya peternak dalam memantau suhu, kelembaban kandang ayam serta pemberian pakan untuk ayam petelur secara otomatis dan dipantau secara real time melalui perangkat mobile.

### Manfaat

Manfaat pembuatan KAPILAR ini adalah:

1. Memudahkan peternak ayam petelur dalam melakukan pengontrolan dan memonitoring kandang ayam petelur dengan jarak jauh.
2. Memudahkan peternak ayam petelur dalam mencegah terjadinya ketidaktepatan waktu dalam mengontrol dan mengendalikan kandang ayam petelur.

### Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah merancang dan membuat alat sistem KAPILAR agar dapat meningkatkan efisiensi dalam mengatur pemberian pakan, memonitor dan mengendalikan suhu serta kelembaban pada kandang ayam dengan kendali jarak jauh secara online melalui smartphone android.

## Batasan Masalah

Dalam pembuatan Tugas Akhir ini, masalah yang akan dibahas terbatas pada:

1. Sistem yang dibangun merupakan sistem yang dapat mengontrol dan memonitoring kandang ayam petelur dengan Android.
2. Aplikasi Android KAPILAR hanya dibuat untuk peternak ayam petelur untuk mengontrol dan memonitoring kandang ayam petelur dari jarak jauh.
3. Sistem ini dikembangkan menggunakan bahasa pemrograman C dan bahasa pemrograman Java.
4. Sistem Monitoring pada KAPILAR mencakup monitoring suhu dan kelembaban kandang ayam petelur serta dilengkapi dengan pengipasan dan lampu menyala otomatis ketika suhu tidak sesuai.

## Sistematika Penulisan

Sistematika penulisan ini terdiri dari beberapa bab dan sub bab sebagai berikut:

BAB I PENDAHULUAN

Pendahuluan berisi latar belakang, batasan masalah dan sistematika penulisan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Berisi penjelasan tentang konsep dan prinsip dasar yang diperlukan untuk memecahkan masalah dalam Tugas Akhir dan untuk merumuskan hipotesis apabila memang diperlukan. Tinjauan pustaka dapat berbentuk uraian kualitatif, model matematis, atau persamaan-persamaan yang berkaitan dengan permasalahan.

BAB III PELAKSANAAN KEGIATAN

Pelaksanaan kegiatan menjelaskan tentang tahapan perancangan, pembuatan alat dan pengukuran alat serta hasil unjuk kerja dari alat yang dibuat.

BAB IV ANALISIS DAN PEMBAHASAN

Analisis dan pembahasan, membahas hasil dari penelitian atau hasil unjuk kerja rancang bangun, dengan analisis dan pembahasan yang sifatnya terpadu.

BAB V KESIMPULAN

Berisi tentang kesimpulan dan saran.

DAFTAR PUSTAKA

Daftar pustaka berisi link sumber dari berbagai makalah atau jurnal yang digunakan dalam laporan.

LAMPIRAN

Lampiran berisi dokumen terkait yang digunakan dalam laporan Tugas Akhir.

# BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai sistem informasi dan teknologi yang telah dipublikasikan pada beberapa jurnal tentang sistem kontrol suhu dan pemberi pakan otomatis yang telah dibuat sebelumnya, yang kemudian akan diulas dan dijadikan bahan untuk pembuatan KAPILAR. Terdapat 10 makalah yang digunakan sebagai rujukan dalam pembangunan KAPILAR. Teknologi yang digunakan yaitu berupa *hardware* maupun *software* untuk membangun sistem kontrol suhu dan pemberi pakan otomasi pada KAPILAR.

## Penilitian Terkait

Beberapa peneliti sudah banyak yang melakukan penelitian terkait pembuatan sistem kontrol suhu dan pemberi pakan otomatis yang diterapkan di berbagai obyek. Beberapa penelitian terkait TA sistem kontrol suhu dan pemberi pakan otomatis pada kandang ayam petelur meliputi sistem kontrol pengatur suhu dan kelembaban otomatis (Mansyur, 2018), sistem pengaturan jadwal pemberian pakan serta pengaturan jadwal penerangan pada kandang ayam (Warjono, dkk, 2018), sistem kendali pengaturan suhu dan kelembaban kandang ayam broiler (Budianto, dkk, 2017), sistem pemberi pakan otomatis berbasis *fuzzy* *logic* *control* (Ridhamuttaqin, dkk, 2013), sistem pemberi pakan ayam serta monitoring suhu, kelembaban berbasis ATmega328 (Laksono, 2017), sistem monitoring suhu dan kontrol air pada kandang burung puyuh (Mindriawan, dkk, 2018), sistem kandang burung otomatis berbasis arduino (Putra, 2018), sistem pengaturan suhu kandang ayam pedaging (Sebayang, dkk, 2016), sistem pemberian pakan dan minum otomatis (Yohanna, dkk, 2018), sistem kandang ayam broiler tipe *close* *house* (Syafar, 2018). Masing-masing memiliki fitur unggulan yang berbeda-beda.

Rancang bangun sistem kontrol pengatur suhu dan kelembaban otomatis pada kandang ayam broiler telah dibuat oleh Mansyur pada tahun 2018. Sistem ini menggunakan Arduino Mega 2560 sebagai mikrokontroler yang berkomunikasi dengan Ethernet Shield untuk mengirimkan data yang akan ditampilkan pada aplikasi android. Masukan sistem ini adalah sensor DHT11 untuk monitoring suhu dan kelembaban kandang. Menggunakan sistem ini, peternak dapat mengontrol secara otomatis suhu dan kelembaban kandang ayam memanfaatkan panas cahaya lampu dan kipas, serta monitoring suhu dan kelembaban secara realtime menggunakan aplikasi android (Blynk) (Mansyur, 2018). Namun, sistem ini masih memiliki kekurangan yaitu belum ada kontrol dan monitoring pakan ayam secara otomatis.

Warjono, dkk dari Program Studi Teknik Elektronika Politeknik Negeri Semarang telah menghasilkan rancang bangun kandang ayam yang mampu memberi pakan ayam sesuai jadwal yang telah diatur dan lampu yang akan menyala pada saat waktu yang telah ditentukan. Jadwal pemberian pakan ayam diatur melalui *keypad* dan akan ditampilkan dengan LCD. Sistem pemberian pakan ini juga dilengkapi dengan sistem peringatan apabila tempat pakan ayam sudah kosong dengan membunyikan *buzzer* dan indikator lampu LED yang menyala. Lampu akan dinyalakan pada malam hari sebagai penerangan di kandang agar ayam dapat memakan makanannya pada waktu malam (Warjono, dkk, 2018). Sistem ini masih memiliki kekurangan yaitu belum adanya antarmuka yang dapat mengendalikan pada jarak jauh.

Prototipe sistem kendali pengaturan suhu dan kelembaban kandang ayam boiler berbasis mikrokontroler atmega328 dengan masukan sensor DHT11 dan keluaran *buzzer*, LED serta LCD 16x2 sudah dibuat oleh Eko Wiji Setio Budianto, Ramadiani, Awang Harsa Kridalaksana (2017). Sistem ini bermanfaat untuk meciptakan suhu dan kelembaban lingkungan yang ideal terhadap ternak ayam boiler agar tidak terjadinya over heating pada ternak dengan pengaturan suhu dan kelembaban secara otomatis (Budianto, dkk, 2017). Kekurangan dari sistem ini yaitu peternak tidak bisa memonitoring suhu dan kelembaban dari jarak jauh.

Aji Ridhamuttaqin, Agus Trisanto, Emir Nasrullah (2013) telah membuat sebuah Rancang bangun model sistem pemberi pakan otomatis berbasis *fuzzy* *logic* *control*. Sistem ini menggunakan mikrokontroler Atmega 8535 sebagai pengendali utamanya yang berfungsi untuk mengatur beberapa bagian seperti katup 1, katup 2, dan konveyor. Kendali logika *fuzzy* digunakan agar pakan dapat terdistribusi secara merata disepanjang konveyor. Sistem ini dapat memberikan pakan ayam secara otomatis dengan timer RTC DS1307 dan strain gage sebagai timbangan berat pakan (Ridhamuttaqin, dkk, 2013). Kekurangan dari alat ini peternak tidak bisa memonitoring dan mengontrol pakan ayam dari jarak jauh.

Arief Budi Laksono (2017) telah membuat rancang bangun sistem pemberi pakan ayam serta monitoring suhu dan kelembaban kandang berbasis ATmega328. Sistem ini menggunakan ATmega328 sebagai mikrokontroler, motor servo sebagai pintu, sensor poto dioda yang dapat membaca wadah pakan ayam saat terisi penuh dan kosong, sensor DHT11 untuk membaca keadaan suhu dan kelembaban pada ruang kandang serta LCD 16x2 sebagai keluaran yang dapat menampilkan suhu dan kelembaban yang terbaca oleh sensor DHT11 dengan baik. Sistem ini telah mampu memberi pakan ayam secara otomatis dan menampilkan suhu dan kelembaban kandang pada LCD (Laksono, 2017). Namun, sistem ini belum dilengkapi pengendali jarak jauh.

Zulhan Mindriawan, I Wayan Agus Arimbawa, I Gede Pasek Suta Wijaya (2018) telah membuat sebuah implementasi *Internet of Things* pada sistem *monitoring* suhu dan kontrol air pada kandang burung puyuh petelur dengan menggunakan Protokol MQTT. Sistem ini merupakan salah satu solusi agar pengontrolan penampungan air dapat dikelola dengan efisien. Sistem yang telah dibangun oleh Zulhan Mindriawan dkk ini menggunakan mikrokontroler NodeMCU ESP8266, sensor DHT22 untuk mengukur suhu dan kelembaban udara, sensor ketinggian air sebagai pemicu jika air sudah penuh, sensor ultrasonik difungsikan untuk mengukur kapasitas air yang ada di dalam penampungan, dan motor servo untuk memutar keran (Mindriawan, dkk, 2018). Sayangnya sistem ini belum memliki pemberian pakan otomatis.



Gambar 2. Aplikasi Monitoring Suhu dan Kontrol Air pada Kandang Burung Puyuh Petelur

Pada tahun 2018 Bagas Pratama Putra telah membangun sebuah rancangan sistem kandang ternak burung otomatis berbasi arduino. Sistem ini dapat mempermudah pemeliharaan ternak burung dalam pemberian pakan dan minum serta monitoring suhu ideal kandang untuk burung. Sistem ini mengunakan Arduino Uno sebagai mikrokontroler yang dihubungkan dengan modul WiFi ESP8266 agar dapat terhubung dengan internet. Masukan dalam sistem ini adalah sensor ultrasonik yang bertugas untuk memantau keadaan pakan sehingga dapat mengaktifkan motor servo agar dapat membuka penutup penampung pakan, sensor water level sebagai indikator ketersediaan air dan sebagai pemicu bekerjanya pompa air, serta sensor suhu DHT11 yang akan mengaktifkan kipas atau pemanas sebagai pengontrol suhu kandang agar ideal. Hasil keluaran pada sistem ini akan ditampilkan melalui aplikasi berbasis web (Putra, 2018). Sistem ini masih memiliki kekurangan, yaitu belum adanya sistem pengendali dan pemantau jarak jauh.

Pada tahun 2016 telah dikembangkan suatu sistem pengaturan suhu kandang ayam pedaging oleh Sebayang, Zebua, dan Soedjarwanto. Sistem ini mampu mengurangi mengontrol suhu lingkungan sekitar kandang ayam pedaging agar tetap sesuai dengan suhu ideal. Sistem ini dilengkapi dengan sensor LM35 untuk mendeteksi suhu lingkungan yang keluarannya akan dibaca oleh mikrokontroler ATmega8535 dan ditampilkan melalui LCD. Sistem ini dapat mengontrol keadaan suhu dilingkungan agar tetap pada suhu ideal yang telah ditetapkan. Apabila suhu yang dibaca oleh sensor mengalami penurunan dari suhu yang telah ditentukan, lampu akan menyala sebagai pemanas. Dan apabila suhu mengalami kenaikan dari suhu yang telah ditentukan, kipas akan menyala (Sebayang, dkk, 2016). Pada sistem ini masih ada kekurangan, yaitu belum adanya kontrol pakan ayam secara otomatis dan belum adanya antarmuka yang dapat mengendalikan dan memantau dari jarak jauh.

Yohanna, dan Toruan (2018) telah membuat rancang bangun sistem pemberian pakan dan minum ayam secara otomatis. Sistem ini dapat mengetahui ketersediaan pakan dan air dan memberitahukan ke peternak melalui SMS. Sistem ini dilengkapi dengan sensor ultrasonik yang akan membaca keadaan wadah pakan dan air. Mikrokontroler yang digunakan pada sistem ini adalah Arduino ATmega328 yang akan mengirimkan data dari sensor ke modul GSM SIM900A. Kemudian status akan dikirimkan ke nomor seluler yang telah ditentukan dalam bentuk SMS (Yohanna, dkk, 2018). Pada sistem ini masih terdapat kekurangan yaitu belum ada controlling dan monitoring suhu dan kelembaban serta belum menggunakan android sebagai pengendali jarak jauh.

A. Muhammad Syafar (2018) telah membuat desain sistem kandang ayam broiler tipe close house berdasarkan parameter suhu dan kelembaban. Sistem ini menggunakan Arduino Mega 2560 sebagai mikrokontroler, sensor DHT 11 untuk mengukur suhu dan kelembaban, sensor LDR untuk mengatur ketersediaan pakan dalam wadah serta LCD yang digunakan untuk monitoring suhu dan kelembaban (Syafar, 2018). Namun sistem ini masih memiliki kekurangan karena belum dapat dikendalikan dari jarak jauh.

Dari beberapa penelitian diatas yang telah dilakukan oleh ahli, belum terdapat sebuah sistem yang mempunyai fitur untuk kontrol suhu dan pemberi pakan otomatis pada kandang ayam petelur yang dapat dikendalikan dengan jarak jauh. Oleh karena itu, TA sistem kontrol suhu dan pemberi pakan otomatis pada kandang ayam petelur ini akan mewujudkan sebuah sistem dengan fitur yang telah disebutkan sebelumnya. Selain itu, *database* yang digunakan adalah Firebase *database real time* yang dapat memperbaruidata tanpa harus dipulihkan setiap saat. Kontrol pakan, lampu dan kipas serta monitoring tangki pakan, tangki air minum, suhu dan kelembapan akan dilaporkan pada aplikasi android KAPILAR.

Tabel 2. Tabel Analisis Perbandingan Sistem Pakan Otomatis dan Monitoring Suhu

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Sistem | Fitur / Teknologi | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Rancangan Bangun Sistem Kontrol Otomatis Pengatur Suhu dan Kelembapan Kandang Ayam Broiler Menggunanakan Arduino | - | - | **-** | √ | √ | √ |
| 2 | Pengatur Pakan dan Penerangan Kandang Terprogram untuk Ayam Petelur | √ | √ | **-** | **-** | √ | √ |
| 3 | Prototipe Sistem Kendali Pengaturan Suhu dan Kelembaban Kandang Ayam Boiler Berbasis Mikrokontroler Atmega328 | **-** | **-** | **-** | **-** | √ | √ |
| 4 | Rancang Bangun Model Sistem Pemberi Pakan Otomatis Berbasis Fuzzy Logic Control | √ | √ | **-** | **-** | **-** | **-** |
| 5 | Rancang Bangun Sistem Pemberi Pakan Ayam Serta Monitoring Suhu dan Kelembaban Kandang Berbasis Atmega328 | √ | √ | **-** | **-** | √ | √ |
| 6 | Implementasi Internet of Things pada Sistem Monitoring Suhu dan Kontrol Air pada Kandang Burung Puyuh Petelur Dengan Menggunakan Protokol MQTT | **-** | **-** | √ | √ | **-** | √ |
| 7 | Rancang Bangun Kandang Ternak Burung Otomatis Berbasis Arduino | √ | √ | √ | **-** | √ | √ |
| 8 | Perancangan Sistem Pengaturan Suhu Kandang Ayam Berbasis Mikrokontroler | **-** | **-** | **-** | **-** | √ | √ |
| 9 | Rancang Bangun Sistem Pemberian Pakan dan Minum Ayam Secara Otomatis | **-** | √ | √ | **-** | **-** | **-** |
| 10 | Desain Sistem Kandang Ayam Broiler Tipe Close House Berdasarkan Parameter Suhu dan Kelembaban | √ | **-** | **-** | **-** | √ | √ |
| 11 | KAPILAR: Sistem Kontrol Suhu dan Pemberi Pakan Otomatis Kandang Ayam Petelur Berbasis Internet of Things | √ | √ | √ | √ | √ | √ |

Keterangan :

1. Kontrol Pakan Otomatis
2. Monitoring Pakan
3. Monitoring Tangki Air Minum
4. Android
5. Kontrol Suhu dan Kelembaban Otomatis
6. Monitoring Suhu dan Kelembaban

Berdasarkan Tabel 2.1 dapat dilihat bahwa dari beberapa sistem yang dibuat oleh peneliti dan berkaitan dengan TA ini belum ada yang membuat sistem kontrol suhu dan pemberi pakan otomatis yang dapat dikendalikan dengan jarak jauh menjadi satu sekaligus. Sistem KAPILAR yang dibuat memiliki fitur standar seperti sistem lain yang telah dibuat oleh peneliti diantaranya kontrol pakan otomatis, kontrol suhu dan kelembaban, monitoring Namun, kelebihan yang ada dari sistem KAPILAR ini adalah sistem ini dapat mengontrol melalui android serta memonitoring tangki pakan dan minum serta suhu dan kelembaban. Selain itu, data yang diterima oleh aplikasi sangat *update* dan data dikirimkan secara *real time* karena sistem ini menggunakan *database* Firebase.

## Teknologi yang digunakan

Pada pembuatan Sistem Kontrol Suhu Dan Pemberi Pakan Otomatis Kandang Ayam Petelur teknologi yang digunakan yaitu sebagai berikut:

### Android

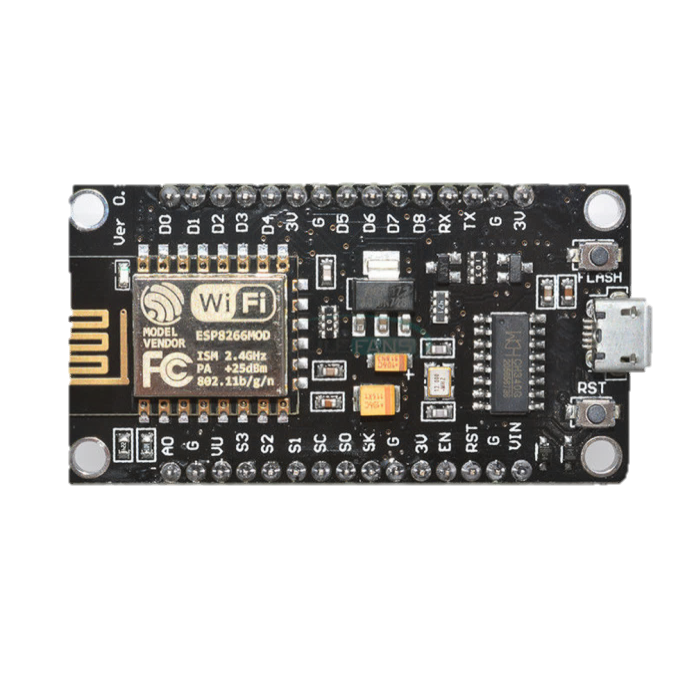
Android adalah salah satu sistem operasi untuk perangkat mobile yang bersifat open-source. Aplikasi berbasis android dikembangan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak android (SDK). Android menyediakan platform yang lengkap baik sistem operasinya, aplikasi maupun tool untuk mengembangkannya. Software yang digunakan untuk mengembangkan aplikasi android adalah Android Studio. Aplikasi android digunakan sebagai antarmuka user untuk memonitoring suhu dan kelembaban kandang, serta kondisi pakan dan tangki air minum secara realtime. Aplikasi ini juga digunakan untuk mengatur waktu pemberian pakan ayam dan waktu nyala lampu.

### Firebase

Firebase adalah sebuah *database real time* sehingga data yang dikirim ke *database* akan secara otomatis masuk tanpa harus di *refresh* setiap saatnya. Firebase mempunyai *host* di *cloud*. Data yang dikirimkan ke firebase adalah dalam bentuk JSON. Pada sistem ini, firebase digunakan sebagai *database real time* yang akan menerima data secara otomatis setiap data dikirimkan.

### NodeMCU

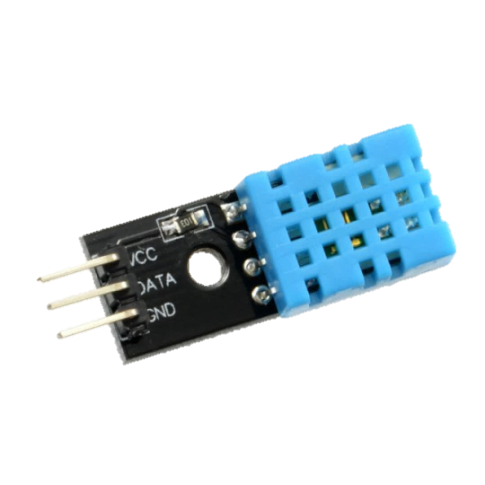
NodeMCU adalah mikrokontroler yang sudah dilengkapi dengan modul WiFi ESP8266 dan bersifat opensource. Dengan begitu board NodeMCU lebih cocok digunakan untuk memprogram sistem berbasis IoT (Internet of Things). Software yang digunakan untuk memprogram board NodeMCU adalah Arduino IDE dengan menginstall modul ESP8266 terlebih dahulu. NodeMCU memberikan akses komunikasi dengan serial agar dapat diprogram melalui komputer / laptop menggunakan kabel USB yang biasa untuk kabel charging smartphone Android. Board NodeMCU digunakan sebagai kontroler yang akan menerima perintah sehingga semua komponen dapat bekerja dengan semestinya dan mengirimkan data ke database. Perangkat mikrokontroler NodeMCU terlihat seperti pada Gambar 2. 2.



Gambar 2. NodeMCU

### Sensor DHT11

Sensor DHT11 adalah salah satu sensor yang dapat mengukur dua parameter lingkungan, yaitu suhu dan kelembaban. Sensor DHT11 memiliki sebuah thermistor tipe NTC (Negative Temeratue Coefficient) untuk mengukur suhu dan sebuah sensor kelembaban tipe resisitf. Tegangan yang dibutuhan untuk mengaktifkan sensor adalah sebesar 5V. Perbedaan nilai suhu dan kelembaban yang dihasilkan oleh sensor ini cukup akurat dengan rentang suhu 0 – 50oC dan memiliki kesalahan sebesar ± 2oC. Sedang untuk kelembaban adalah sebesar 20 – 90% RH dan memiliki kesalahan ± 5% RH. Sensor DHT11 digunakan untuk mengukur suhu dan kelembaban pada kandang ayam petelur, lalu data akan dikirimkan untuk ditampilkan hasil pengukurannya pada aplikasi android. Sensor DHT11 terlihat seperti pada Gambar 2. 3.



Gambar 2. Sensor DHT11

### Sensor Ultrasonic

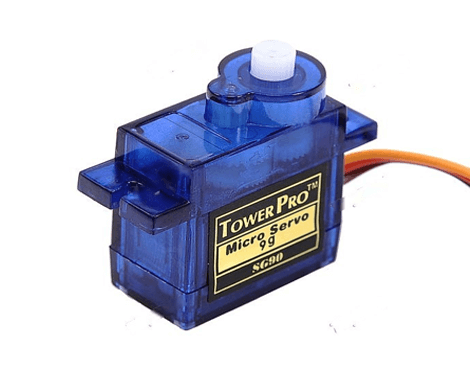
Ultrasonic seperti pada Gambar 2. 4 merupakan sebuah sensor yang biasa digunakan untuk mengukur jarak berbasis gelombang ultrasonic. Gelombang suara ultrasonic akan dipancarkan dan dipantulkan kembali ke receiver ultrasonic dengan frekuensi 40kHz dan diubah menjadi besaran listrik. Selisih waktu antara waktu pengiriman dan waktu penerimaan gelombang adalah representasi dari jarak objek. Jangkauan ukur sensor ini sejauh 3 – 300 cm dan membutuhkan tegangan kerja 5V DC. Di sistem ini sensor ultrasonic akan digunakan untuk memonitoring kondisi pakan dan air minum ayam petelur dengan mengukur ketinggian tangki pakan dan air minum, lalu data yang diterima dari sensor ultrasonic akan dikirimkan ke aplikasi KAPILAR.



Gambar 2. Sensor Ultrasonic

### Motor Servo

Motor servo seperti pada Gambar 2. 5 adalah sebuah actuator putar atau motor yang memiliki serangkaian gear internal, potensiometer dan rangkaian control elektronika dalam satu paket. Potensiometer dalam motor servo digunakan untuk menentukan batas sudut dari putaran motor servo. Motor servo merupakan sebuah motor DC yang dapat bekerja dua arah yaitu, searah jarum jam dan berlawanan dengan jarum jam. Untuk beroperasi, motor servo membutuhkan tegangan daya sebesar 5V DC. Motor servo akan bekerja dengan baik bila pin control menerima sinyal PWM dengan frekuensi 50 Hz dan mendapat masukan lebar pulsa yang tepat. Pada sistem ini motor servo digunakan sebagai motor penggerak katup pakan ayam petelur dan akan bergerak sesuai dengan waktu yeng telah diberikan sebelumnya.



Gambar 2. Motor Servo

# BAB III PELAKSANAAN KEGIATAN

Pada bab ini akan membahas mengenai pelaksanaan Tugas Akhir yang meliputi gambaran umum sistem, analisis sistem, perancangan sistem, antarmuka aplikasi serta perancangan alat KAPILAR.

## Gambaran Umum Sistem

Pada bagian ini akan digambarkan bagaimana sistem KAPILAR akan bekerja melalui gambaran umum serta alur kerja sistem, seperti pada Gambar 3.1:

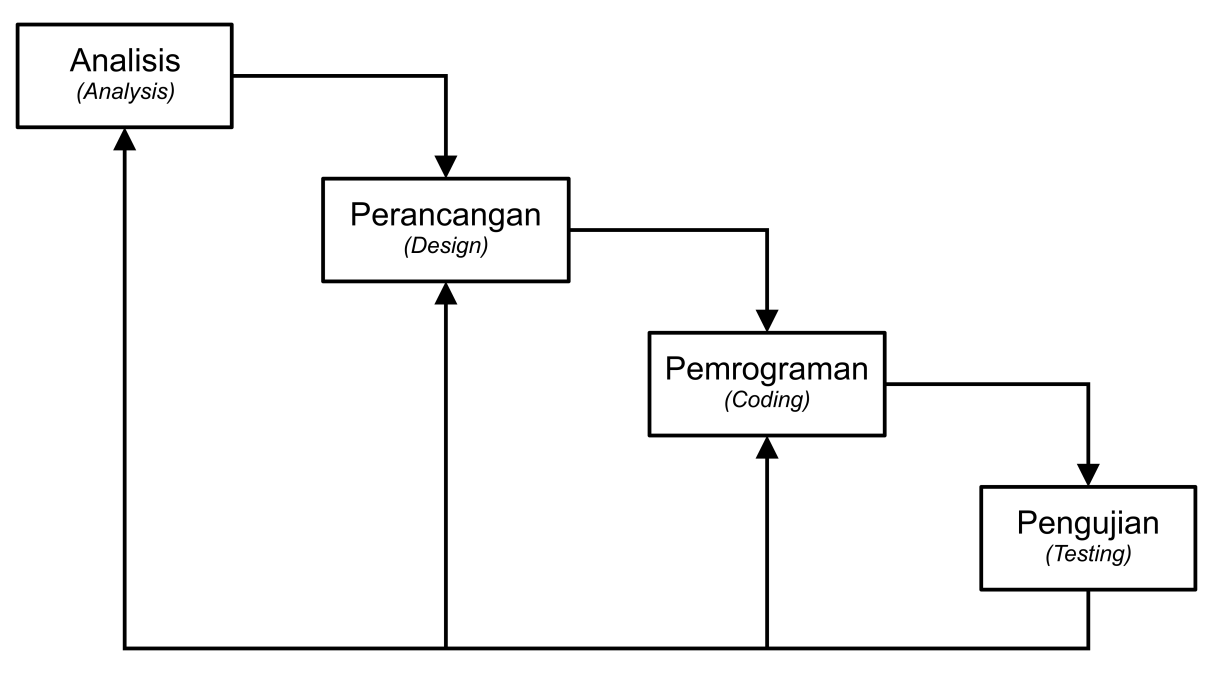


Gambar 3. Gambaran Umum Sistem KAPILAR

Berdasarkan Gambar 3.1 sistem KAPILAR terbagi menjadi dua sistem yaitu sistem *monitoring* dan sistem *controlling*. Sistem *monitoring* KAPILAR mendapat masukan dari sensor DHT11 untuk memantau suhu dan kelembaban serta sensor Ultrasonic untuk memantau kondisi tangki pakan ayam dan tangki air minum. Sistem *controlling* pada KAPILAR yaitu berupa pengendali lampu dan kipas otomatis berdasarkan kondisi suhu dan kelembaban serta pemroses NodeMCU yang tersambung dengan jaringan *wireless*. Sistem *monitoring* dan *controlling* tersebut terintegrasi dengan *real time database* Firebase, yang kemudian akan menghasilkan keluaran yang ditampilkan pada aplikasi android.

## Analisis Sistem

Dalam pelaksanaan Tugas Akhir ini, metode yang digunakan dalam pembuatan sistem KAPILAR adalah metode *waterfall*. Menurut Rosa dan Shalahuddin (2011) metode *waterfall* yang juga disebut *linear* *sequential* *model* atau *classic life of cycle* adalah metode pengembangan dengan cakupan aktivitas yang memiliki tahapan seperti pada Gambar 3.2:



Gambar 3. Tahapan Metode Waterfall Menurut Rosa dan Shalahuddin (2011)

Tahap pertama pada metode waterfall adalah tahap analisis yaitu mengumpulkan informasi kebutuhan untuk pengguna sehingga mendapat spesifikasi yang dibutuhkan dalam pengembangan sistem. Tahap perancangan dilakukan setelah mendapat informasi dan mengubahnya menjadi bentuk karakteristik yang akan dimengerti perangkat lunak sebelum melakukan penulisan program. Tahap pemrograman yaitu mengubah hasil perancangan dengan menuliskan kode program menggunakan bahasa pemrograman pada perangkat keras maupun perangkat lunak. Tahap terakhir yang dilakukan adalah pengujian pada sistem yang sudah selesai dirancang apakah terdapat kesalahan dan sudah sesuai dengan hasil yang diinginkan (Rosa & Shalahuddin, 2011).

Dalam pengembangannya, sistem KAPILAR menerapkan metode waterfall dengan tahapan sebagai berikut:

1. **Analisis**

Tahap analisis dilakukan untuk memahami kebutuhan yang diperlukan dalam pengembangan sistem KAPILAR sehingga dapat mengerti batasan perangkat lunak yang dikembangkan. Analisis dapat dilakukan setelah mengumpulkan informasi yang diperoleh dari wawancara kepada pihak yang ahli di bidang peternakan ayam petelur, mengumpulkan literatur yang berkaitan, dan survei langsung ke tempat peternakan ayam petelur. Informasi tersebut dianalisis sehingga mendapatkan data yang dibutuhkan.

1. **Perancangan**

Setelah hasil analisis kebutuhan didapat, tahap selanjutnya yaitu merancang sistem dalam bentuk karakteristik yang akan membantu dalam penulisan kode program.

1. **Pemrograman**

Penulisan kode program dilakukan pada perangkat keras serta sistem perangkat lunak yang sudah dirancang pada tahap sebelumnya.

1. **Pengujian**

Pengujian dilakukan pada sistem KAPILAR yang telah selesai dibuat apakah sudah berfungsi sesuai keinginan dan apakah terdapat kesalahan atau tidak yang kemudian dilakukan perbaikan dan pemeliharaan agar sistem KAPILAR bekerja sesuai fungsinya.

### Analisis Kebutuhan

Tahap awal untuk menganalisis kebutuhan yang diperlukan dalam pembuatan Tugas Akhir sistem KAPILAR adalah mengumpulkan informasi yang diperoleh dari wawancara kepada pengelola peternakan ayam petelur. Tempat yang dijadikan lokasi observasi yaitu usaha peternakan ayam petelur Lanu Farm di Desa Sodong, Kelurahan Genting, Sumowono. Serta peternakan ayam petelur milik pereorangan Bu Fatimah di Bubagan, Gunung Pati.

Hasil wawancara tersebut menghasilkan beberapa informasi terkait pengelolaan peternakan ayam petelur antara lain adalah: ayam petelur yang sedang dalam periode produksi telur merupakan ayam petelur yang berumur 30 minggu. Suhu ideal bagi ayam yang sedang dalam periode produksi berkisar antara suhu 20 – 30oC, sedangkan untuk kelembaban yang ideal berkisar antara 50 – 70%. Pengelola peternakan ini hanya memberi pakan pada ayam satu kali dalam sehari sebanyak 120g/ekor.

Informasi yang telah didapat dianalisis dan menghasilkan data yang dibutuhkan, antara lain kebutuhan perangkat keras (*hardware*) maupun kebutuhan perangkat lunak (*software*).

#### Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk membangun sistem KAPILAR adalah sebagai berikut:

1. NodeMCU
2. Sensor DHT11
3. Sensor Ultrasonic
4. Motor Servo
5. Kabel Power
6. Kabel Jumper
7. Lampu 220V dan Fitting
8. Kipas
9. Personal Komputer/ Laptop
10. Smartphone Android

#### Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membangun sistem KAPILAR adalah sebagai berikut:

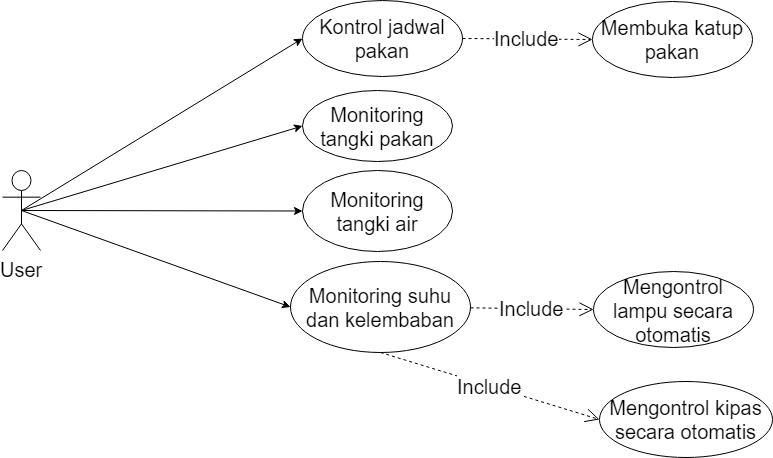
1. Android Studio
2. Arduino IDE
3. Firebase

## Perancangan Sistem

Pemodelan sistem yang digunakan dalam perancangan sistem KAPILAR adalah UML (*Unified Modeling Language*). Desain UML yang akan digunakan adalah *Use Case Diagram* dan *Activity Diagram*.

### *Use Case Diagram*

*Use Case Diagram* merupakan sebuah teknik yang digunakan dalam pengembangan perangkat lunak pada pemodelan sistem UML. *Use Case Diagram* digunakan untuk menjelaskan hubungan antara aktor dengan sistem yang ada. *Use Case Diagram* sistem KAPILAR terlihat seperti Gambar 3.3:



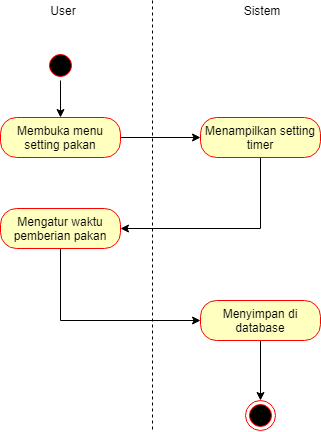
Gambar 3. *Use Case Diagram* Sistem KAPILAR

Berdasarkan Gambar 3.3 *user* dapat mengakses beberapa fitur yang ada pada aplikasi KAPILAR. Beberapa fitur yang dapat diakses oleh *user* antara lain, mengontrol jadwal pemberian pakan ayam, memantau tangki pakan dan air minum untuk ayam, serta memonitoring suhu dan kelembaban yang ada di lingkungan kandang ayam petelur. Pada kontrol jadwal pakan user dapat mensetting alarm pemberian pakan melalui android dari jarak berapapun dan dimanapun. Keterangan lebih lanjut mengenai use case ini dapat dilihat pada Tabel 3.1 rekap daftar pengujian aplikasi.

### *Activity Diagram*

1. ***Activity Diagram* Kontrol Jadwal Pakan**

*Activity Diagram* Kontrol Jadwal Pakan adalah seperti Gambar 3.4:

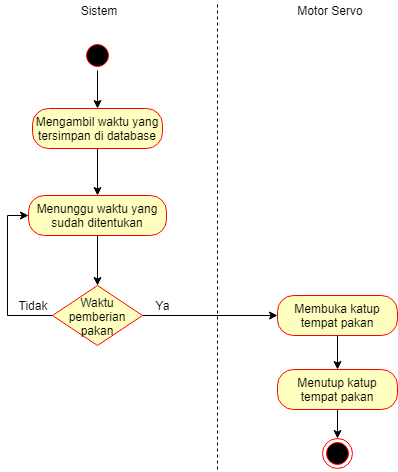


Gambar 3. *Activity Diagram* Kontrol Pakan Otomatis

Berdasarkan Gambar 3.4 *user* membuka menu pengaturan pakan lalu sistem akan menampilkan pengaturan waktu. Setelah user mengatur waktu pemberian pakan, data akan disimpan ke Firebase.

1. ***Activity Diagram* Membuka Katup Pakan**

*Activity Diagram* Membuka Katup Pakan adalah seperti Gambar 3.5:

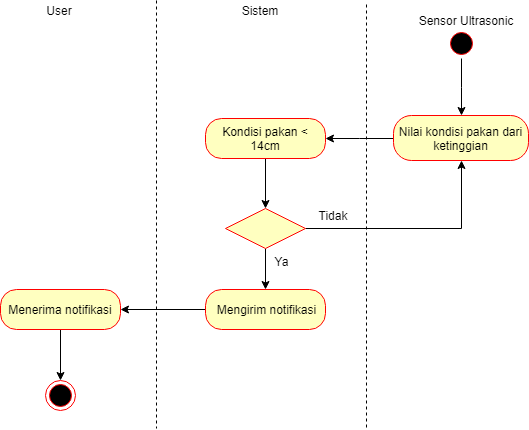


Gambar 3. *Activity Diagram* Membuka Katup Pakan

Berdasarkan Gambar 3.5 sistem mengambil waktu yang sudah di setting oleh user dan tersimpan di Firebase. Saat waktu pemberian pakan telah tiba, sistem mengirimkan perintah ke motor servo untuk membuka katup tempat pakan. Setelah beberapa saat motor servo akan menutup kembali katup tempat pakan.

1. ***Activity Diagram* *Monitoring* Kondisi Pakan**

*Activity Diagram* *Monitoring* Kondisi Pakan adalah seperti Gambar 3.6:

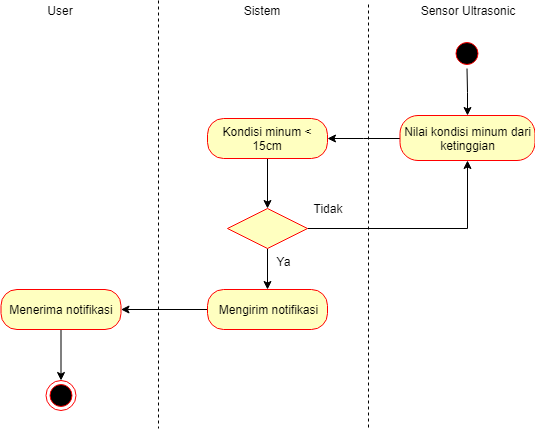


Gambar 3. *Activity Diagram* *Monitoring* Kondisi Pakan

Berdasarkan Gambar 3.6 sensor ultrasonic akan membaca kondisi pakan lalu mengirimkannya ke sistem dan di simpan ke *Real Time Database* Firebase. Sistem akan membaca nilai yang dikirim apabila ketinggian pakan pada tangki kurang dari 15 cm maka sistem akan mengirim notifikasi ke *user*. Dan apabila nilai yang dikirimkan masih belum mencapai batas sensor akan terus mengirimkan nilai kondisi pakan.

1. ***Activity Diagram* *Monitoring* Kondisi Air Minum**

*Activity Diagram* *Monitoring* Tempat Minum adalah seperti Gambar 3.7:

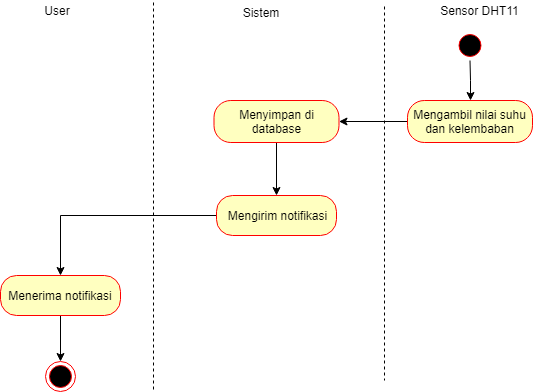


Gambar 3. *Activity Diagram* *Monitoring* Tempat Minum

Berdasarkan Gambar 3.7 sensor ultrasonic akan membaca kondisi air minum lalu mengirimkannya ke sistem dan di simpan ke *Real Time Database* Firebase. Sistem akan membaca nilai yang dikirim apabila ketinggian air pada tangki kurang dari 15 cm maka sistem akan mengirim notifikasi ke user. Dan apabila nilai yang dikirimkan masih belum mencapai batas sensor akan terus mengirimkan nilai kondisi air minum.

1. ***Activity Diagram* *Monitoring* Suhu & Kelembaban**

*Activity Diagram* *Monitoring* Suhu & Kelembaban adalah seperti Gambar 3.8:

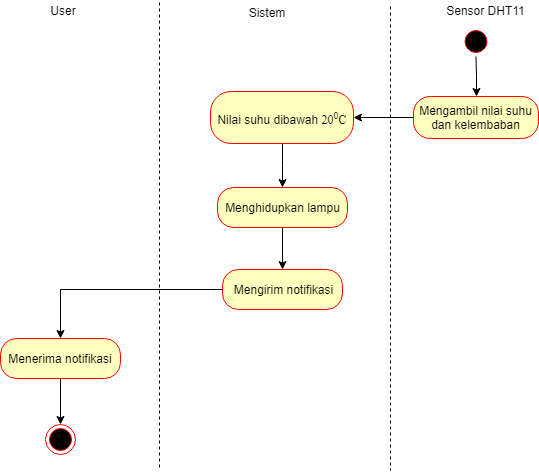


Gambar 3. *Activity Diagram* *Monitoring* Suhu & Kelembaban

Berdasarkan Gambar 3.8 sensor DHT11 akan membaca kondisi suhu dan kelembaban pada lingkungan kandang lalu mengirimkannya ke sistem dan di simpan ke *Real Time Database* Firebase. Sistem akan menampilkan nilai suhu dan kelembaban secara *real time* melalui aplikasi KAPILAR. Kipas atau lampu akan menyala apabila suhu dan kelembaban tidak sesuai dengan suhu dan kelembaban ideal yang sudah ditentukan.

1. ***Activity Diagram* MengontrolLampu Secara Otomatis**

*Activity Diagram* MengontrolLampu Secara Otomatis adalah seperti Gambar 3.9:

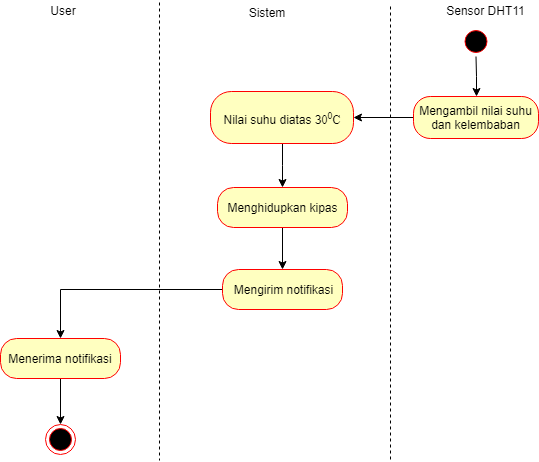


Gambar 3. *Activity Diagram* MengontrolLampu Secara Otomatis

Berdasarkan Gambar 3.9 sensor DHT11 akan membaca kondisi suhu dan kelembaban pada lingkungan kandang lalu mengirimkannya ke sistem dan disimpan di *Real Time Database* Firebase. Jika nilai suhu dibawah 200C maka sistem akan menyalakan lampu dan mengirimkan notifikasi ke *user* melalui aplikasi KAPILAR*.*

1. ***Activity Diagram* MengontrolKipas Secara Otomatis**

*Activity Diagram* MengontrolKipas Secara Otomatis adalah seperti Gambar 3.10:



Gambar 3. *Activity Diagram* MengontrolKipas Secara Otomatis

Berdasarkan Gambar 3.10 sensor DHT11 akan membaca kondisi suhu dan kelembaban pada lingkungan kandang lalu mengirimkannya ke sistem dan disimpan di *Real Time Database* Firebase. Jika nilai suhu diatas 300C maka sistem akan menyalakan kipas dan mengirimkan notifikasi ke *user* melalui aplikasi KAPILAR*.*

### Perancangan Struktur *Database* Firebase

Semua data pada *realtime* *database* firebase disimpan sebagai objek JSON. Ketika data yang sudah didapat disimpan ke JSON *Tree*, data tersebut akan menjadi node di struktur JSON yang ada dengan nama *child* atau *field* yang diberikan saat akan menambahkan data pada *database*. Perbedaan struktur database pada firebase dengan database SQL adalah database pada firebase tidak memiliki tabel atau catatan. Meskipun database ini menggunakan JSON *Tree*, data yang tersimpan di database dapat digambarkan sebagai jenis bawaan tertentu yang berkaitan dengan jenis JSON yang sudah ada, agar penulisan kode menjadi lebih mudah dikelola.

Untuk menyimpan masukan data ke *database* firebase yang didapat dari sensor-sensor, penulisan kode program NodeMCU pada proyek Arduino IDE dituliskan dengan format perintah seperti berikut:

Firebase.set<Tipe\_data>(“/nama\_child”, <namaVariabel>);

Sehingga penulisan kode program untuk menyimpan data aplikasi KAPILAR adalah sebagai berikut:

Firebase.setString("/suhu\_kelembaban/nilai\_kelembaban",fireHumid);

Firebase.setString("/suhu\_kelembaban/nilai\_suhu", fireTemp);

Firebase.setString("/tinggi\_pakan", fireUltra);

Firebase.setString("/tinggi\_air", fireUltraAir);

Firebase.setBool("/status\_servo", fireServo);

Firebase.setBool("/status\_lampu", fireLampu);

Firebase.setBool("/status\_kipas", fireKipas);

Sedangkan untuk membaca atau menulis data Firebase pada Android Studio, data firebase ditulis ke referensi FirebaseDatabase dan diambil dengan memasang *listener* asinkron ke refensi. *Listener* terpicu sekali untuk status awal data dan terpicu lagi setiap kali data berubah. *Instance* DatabaseReference dibutuhkan untuk membaca atau menulis data dari database, sehingga penulisan kode program pada aplikasi KAPILAR untuk nama *child* waktu adalah sebagai berikut:

**private** FirebaseDatabase **firebaseDatabase** = FirebaseDatabase.*getInstance*();  
**private** DatabaseReference **mRootRef** = **firebaseDatabase**.getReference();  
**private** DatabaseReference **mWaktuRef** = **mRootRef**.child(**"waktu"**);

Untuk menambahkan data pada *child* waktu digunakan perintah setValue seperti berikut:

String timeText = DateFormat .*getTimeInstance* (DateFormat.***SHORT***). format(c.getTime());  
**mWaktuRef**.setValue(timeText);}

Untuk mendeteksi perubahan dari data yang disimpan, digunakan metode addValueEventListener() pada child waktu, sehingga penulisan kode programnya seperti berikut:

@Override  
 **protected void** onStart() {

**mWaktuRef**.addValueEventListener(**this**);}

@Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {

**if**(dataSnapshot.getValue(String.**class**)!=**null**){

String key = dataSnapshot.getKey();

**if** (key.equals(**"waktu"**)){

String waktu = dataSnapshot.getValue(String.**class**);

**if**(!waktu.equals(**"0"**)){

**mTextView**.setText(waktu);}}}}

@Override

**public void** onCancelled(@NonNull DatabaseError databaseError) {}}

Sehingga struktur data sistem KAPILAR pada *realtime* *database* firebase terlihat seperti berikut:

{ kapilarapp

“status\_kipas”: true

“status\_lampu”: false

“status\_servo”: false

“suhu\_kelembaban”: {

“nilai\_kelembaban”: “78.00”,

“nilai\_suhu”: “31.20”

“tinggi\_air”: “8”

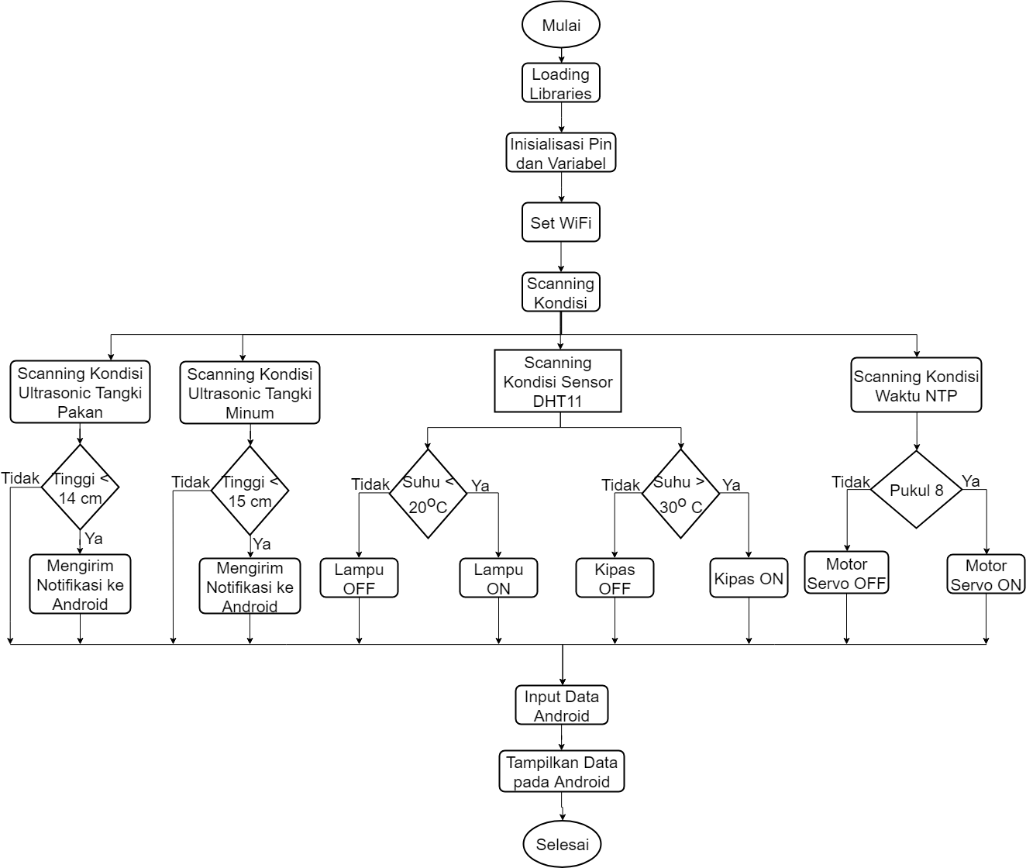
“tinggi\_pakan”: “5”

“waktu”: “08:00:00”

}

## Flowchart

Flowchart menggambarkan alur kerja dari sistem KAPILAR yang terdiri dari sistem *monitoring* dan sistem *controlling*. Sistem *monitoring* dalam sistem KAPILAR meliputi *monitoring* tempat pakan dan tempat air minum yang memanfaatkan kinerja sensor ultransonic. Sedangkan sistem *controlling* dibuat secara otomatis untuk membuka katup tangki pakan sesuai waktu yang telah ditetapkan. Flowchart sistem KAPILAR terlihat seperti Gambar 3.11

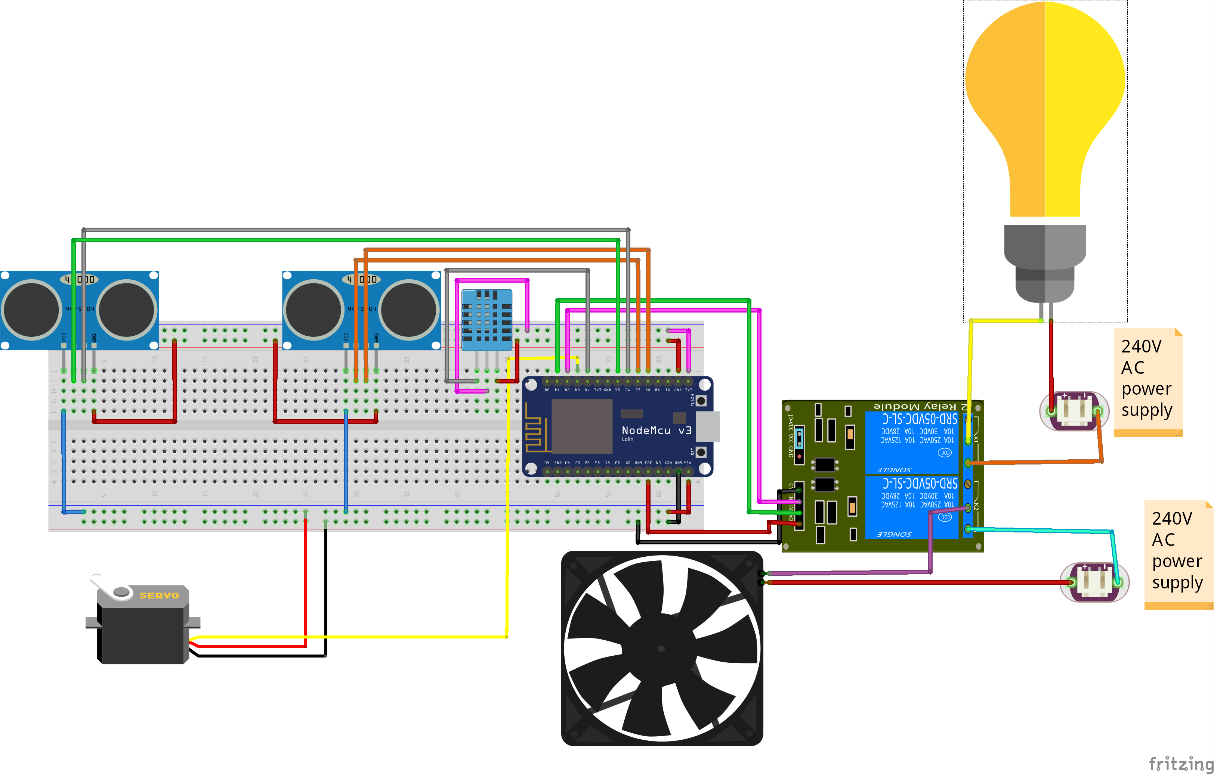


Gambar 3. Flowchart sistem KAPILAR

Pada Gambar 3.11 sistem monitoring tempat pakan dan tempat air minum dimulai dari pemindaian kondisi oleh sensor ultrasonic. Sistem akan mengirim notifikasi ke Android apabila ketinggian pakan yang diukur dari sensor kurang dari 14 cm dan ketinggian air yang diukur dari sensor kurang dari 15 cm. Untuk monitoring dan kontrol suhu kelembaban dimulai dari pemindaian oleh sensor DHT11. Apabila suhu kurang dari 20oC lampu akan menyala, dan apabila suhu lebih dari 30oC kipas akan menyala. Sedangkan untuk pemberian pakan otomatis apabila sudah tiba waktu yang telah diatur dalam sistem, maka motor servo akan membuka katup tempat pakan ayam.

## Wiring Diagram

Wiring diagram sistem KAPILAR akan ditampilkan pada Gambar 3.12



Gambar 3. Wiring Diagram Sistem KAPILAR

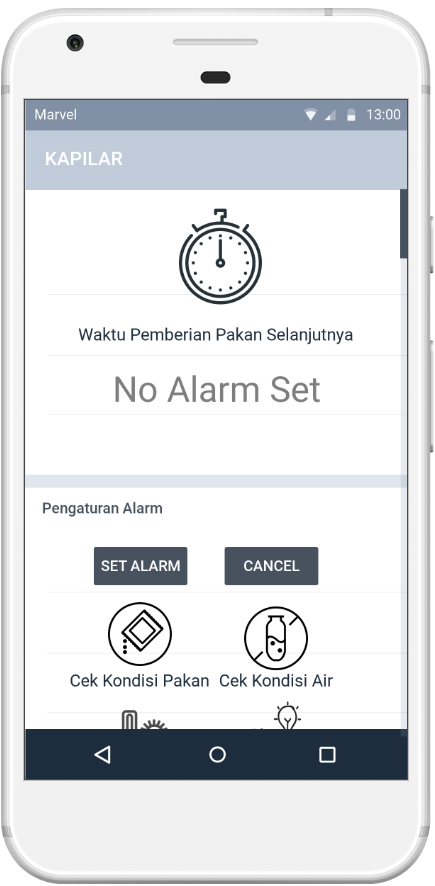
Pada Gambar 3.12 terdapat sebuah lampu dan kipas yang dihubungkan ke relay dan NodeMCU. Lampu dan kipas tersebut akan menyala dan mati secara otomatis sesuai dengan kondisi suhu sekitar yang diukur menggunakan sensor DHT11. Jika sensor DHT11 membaca suhu kandang 20oC maka NodeMCU akan memberikan perintah untuk menyalakan lampu secara otomatis, bila sensor DHT11 membaca suhu kandang 30oC maka NodeMCU akan memberikan perintah untuk menyalakan kipas secara otomatis. Apilkasi juga akan menerima data suhu dan kelembaban untuk memonitoringnya. Motor servo akan bergerak secara otomatis sesuai dengan jadwal yang telah di set melalui aplikasi. Sensor ultrasonic akan mengukur kondisi pakan dan tangki minum berdasarkan dari ketinggiannya, dan bila air minum atau pakan akan habis maka NodeMCU akan mengirimkan notifikasi peringatan ke aplikasi.

## Antarmuka Aplikasi

Pada bagian ini akan ditampilkan antarmuka berbasis grafis untuk pengguna aplikasi KAPILAR.

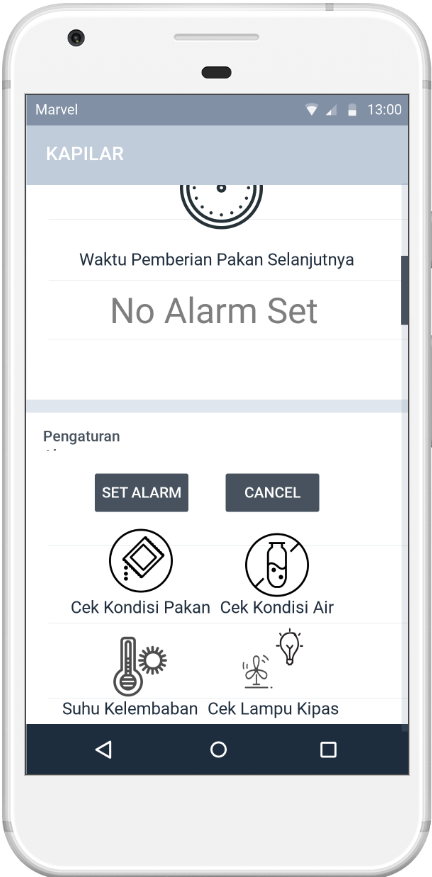
### Tampilan Menu Utama Aplikasi

Tampilan desain menu utama aplikasi akan ditampilkan pada Gambar 3.13



Gambar 3. Tampilan Menu Utama Aplikasi (1)

Apabila halaman menu utama digulir ke bawah akan muncul tampilan yang menampilkan beberapa menu aplikasi KAPILAR seperti Gambar 3.14.

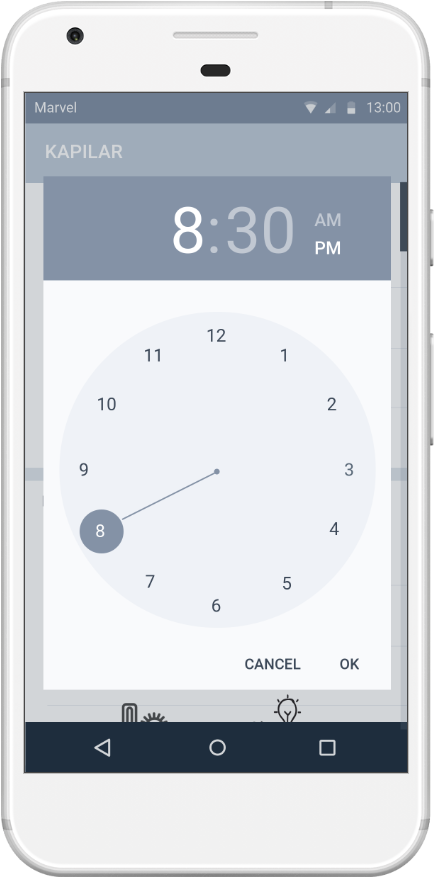


Gambar 3. Tampilan Menu Utama Aplikasi (2)

Dalam halaman menu utama aplikasi kapilar akan ditunjukkan waktu pemberian pakan selanjutnya yang apabila belum diatur waktunya akan muncul tulisan “No Alarm Set”, tombol untuk mengatur dan membatalkan waktu pemberian pakan, serta tersedia menu-menu yang ada pada aplikasi kapilar seperti menu cek kondisi pakan, menu cek kondisi air, menu suhu kelembaban, dan menu cek lampu kipas.

### Tampilan Pengaturan Alarm Pemberian Pakan

Tampilan pengaturan alarm pemberian pakan akan ditampilkan pada Gambar 3.15



Gambar 3. Tampilan Pengaturan Alarm Pemberian Pakan

Tampilan seperti pada Gambar 3.15 akan muncul apabila pengguna memilih tombol “SET ALARM” yang ada pada menu utama. Dalam menu ini pengguna akan dapat mengatur waktu pemberian pakan untuk ayam yang berupa pengaturan jam dan menit. Setelah mengatur waktu pemberian pakan, pengguna dapat memilih “OK” untuk menyimpan waktu pemberian pakan atau dapat memilih “CANCEL” apabila batal untuk mengatur alarm pemberian pakan.

### Tampilan Menu Utama Kondisi Alarm *On*

Tampilan menu utama saat kondisi alarm *on* akan ditampilkan pada Gambar 3.16

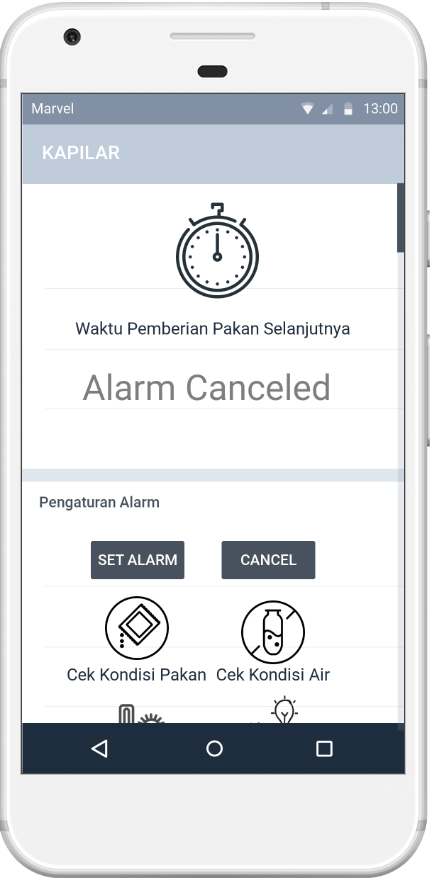


Gambar 3. Tampilan Menu Utama Kondisi Alarm *On*

Tampilan halaman menu utama aplikasi KAPILAR akan terlihat seperti Gambar 3.16 saat alarm telah diatur oleh pengguna. Tulisan yang awalnya “No Alarm Set” akan berganti dengan waktu yang telah ditetapkan.

### Tampilan Menu Utama Kondisi Alarm Dibatalkan

Tampilan halaman menu utama saat kondisi alarm dibatalkan akan ditampilkan pada Gambar 3.17

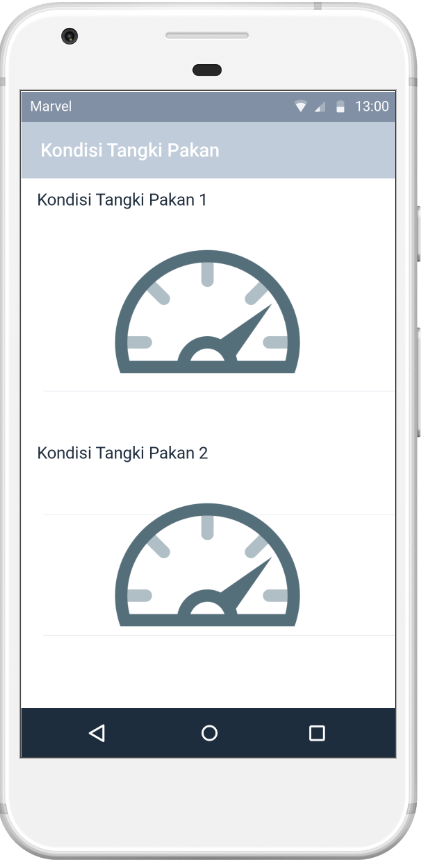


Gambar 3. Tampilan Menu Utama Kondisi Alarm Dibatalkan

Saat waktu alarm yang sudah diatur oleh pengguna sebelumnya dibatalkan, maka tampilan halaman menu utama aplikasi KAPILAR akan terlihat berubah menjadi seperti Gambar 3.17. Tulisan waktu alarm pemberian pakan yang telah ditetapkan akan berganti menjadi tulisan “Alarm Canceled”.

### Tampilan Halaman Monitoring Kondisi Pakan

Tampilan halaman monitoring kondisi pakan akan tampil apabila pengguna memilih menu Cek Kondisi Pakan pada halaman menu utama. Tampilan halaman monitoring kondisi pakan akan ditunjukkan pada Gambar 3.18

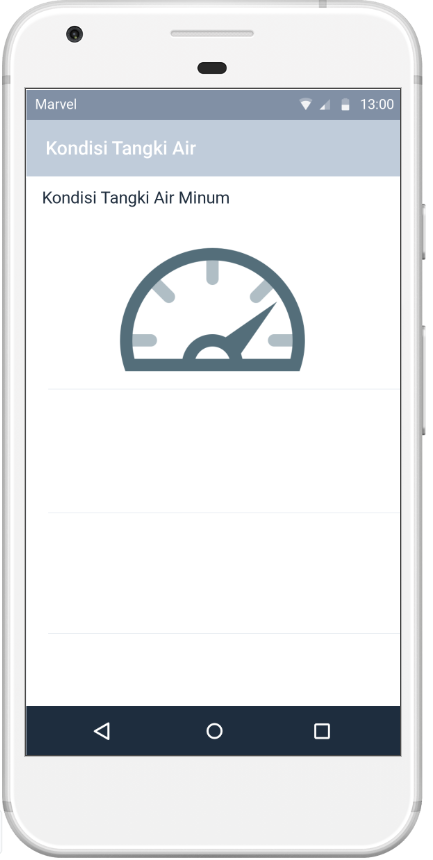


Gambar 3. Tampilan Halaman Monitoring Kondisi Pakan

Pada Gambar 3.18 dapat dilihat bahwa kondisi tempat pakan ayam akan ditunjukkan menggunakan grafik yang datanya diambil dari *realtime* *database* di Firebase. Grafik tersebut akan menampilkan kondisi pakan dalam persen serta menunjukkannya lewat indikator dalam grafik.

### Tampilan Halaman Monitoring Kondisi Air

Tampilan halaman monitoring kondisi air minum akan tampil apabila pengguna memilih menu Cek Kondisi Air pada halaman menu utama. Tampilan halaman monitoring kondisi air akan ditunjukkan pada Gambar 3.19

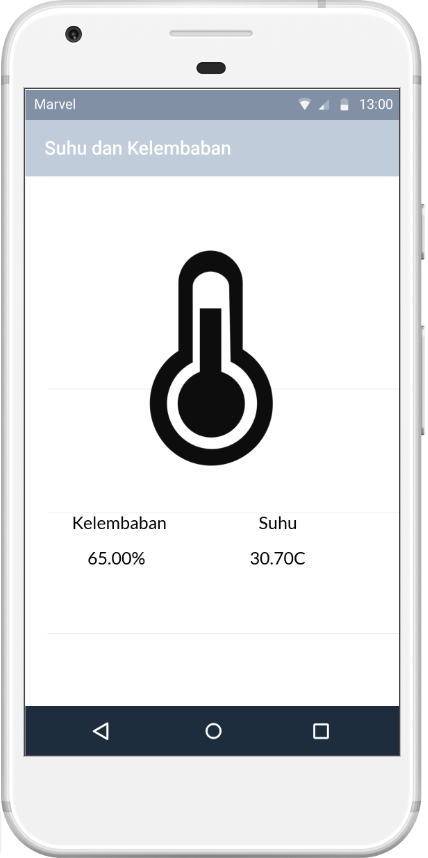


Gambar 3. Tampilan Halaman Monitoring Kondisi Pakan

Pada Gambar 3.19 dapat dilihat bahwa kondisi tempat air minum ayam akan ditunjukkan menggunakan grafik yang datanya diambil dari *realtime* *database* di Firebase. Grafik tersebut akan menampilkan kondisi pakan dalam persen serta menunjukkannya lewat indikator dalam grafik.

### Tampilan Halaman *Monitoring* Suhu dan Kelembaban

Tampilan halaman *monitoring* suhu dan kelembaban akan muncul apabila pengguna memilih menu Suhu Kelembaban yang ada pada halaman menu utama. Tampilan halaman monitoring suhu dan kelembaban akan ditampilkan pada Gambar 3.20

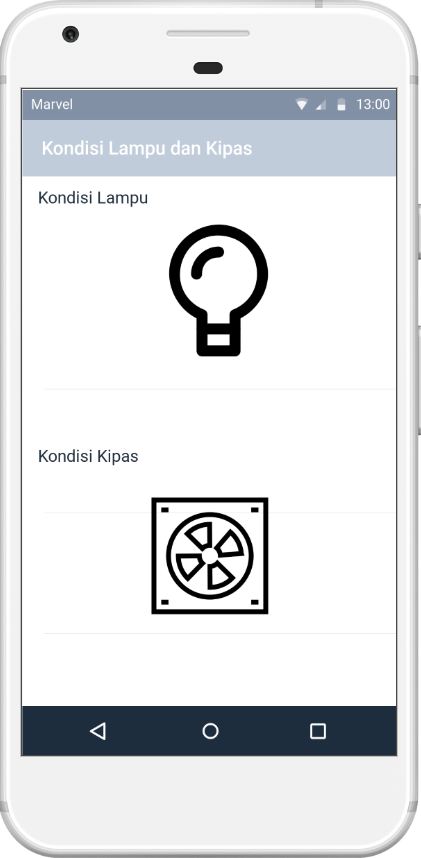


Gambar 3. Tampilan Halaman *Monitoring* Suhu dan Kelembaban

Kondisi suhu akan ditampilkan dengan grafik secara *real time*. Panah indikator pada grafik akan menunjukkan suhu lingkungan sekitar kandang yang didapat dari masukan sensor suhu DHT11 dan disimpan pada *realtime* *database* di Firebase. Data suhu dan kelembaban diambil dari *realtime* *database* di Firebase yang juga akan ditampilkan berupa keterangan tulisan yang terdapat dibawah grafik.

### Tampilan Menu Kondisi Lampu dan Kipas

Tampilan halaman *monitoring* kondisi lampu dan kipas akan muncul apabila pengguna memilih menu Cek Lampu Kipas yang ada pada halaman menu utama. Tampilan halaman *monitoring* kondisi lampu dan kipas akan ditampilkan pada Gambar 3.21

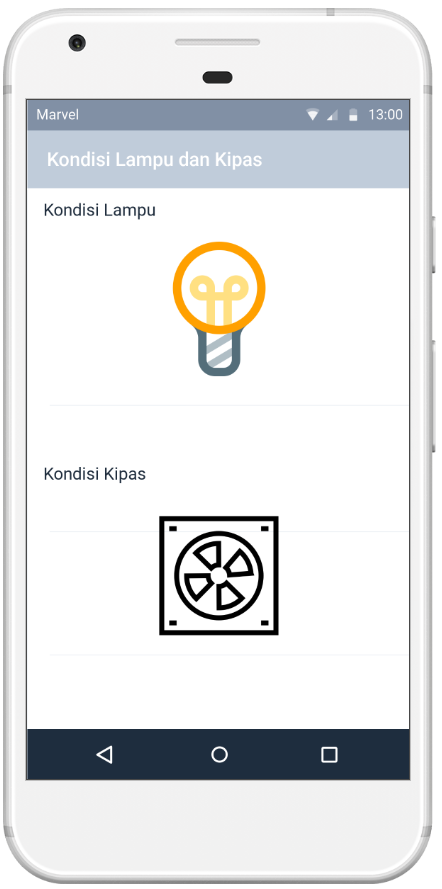


Gambar 3. Tampilan Menu Kondisi Lampu dan Kipas

Terlihat pada Gambar 3.21 halaman *monitoring* kondisi lampu dan kipas akan ditunjukkan dengan gambar lampu dan kipas yang tidak berwarna.

### Tampilan Menu Kondisi Lampu dan Kipas Saat Lampu Menyala

Tampilan halaman menu kondisi lampu dan kipas saat kondisi lampu menyala akan ditampilkan pada Gambar 3.22

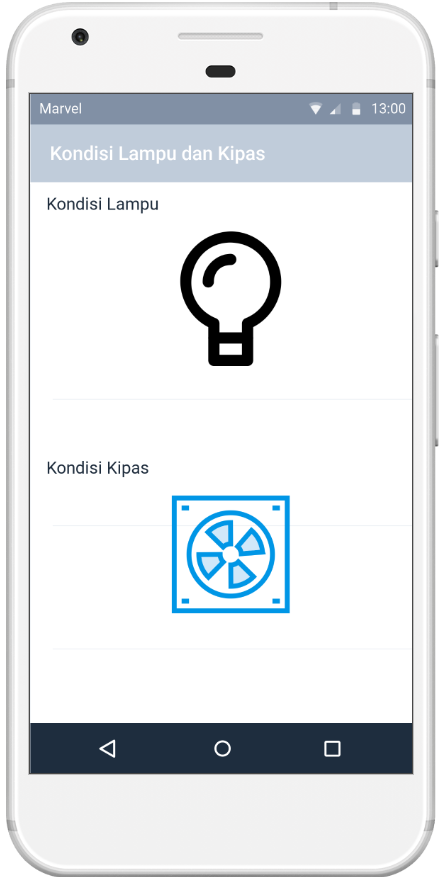


Gambar 3. Tampilan Halaman Menu Kondisi Lampu dan Kipas Saat Lampu Menyala

Pada Gambar 3.22 menunjukkan indikator kondisi lampu saat menyala pada halaman menu lampu dan kipas akan ditunjukan dengan gambar lampu yang sebelumnya tidak berwarna berubah menjadi berwarna.

### Tampilan Menu Kondisi Lampu dan Kipas Saat Kipas Menyala

Tampilan halaman menu kondisi lampu dan kipas saat kondisi kipas menyala akan ditampilkan pada Gambar 3.23



Gambar 3. Tampilan Halaman Menu Kondisi Lampu dan Kipas Saat Kipas Menyala

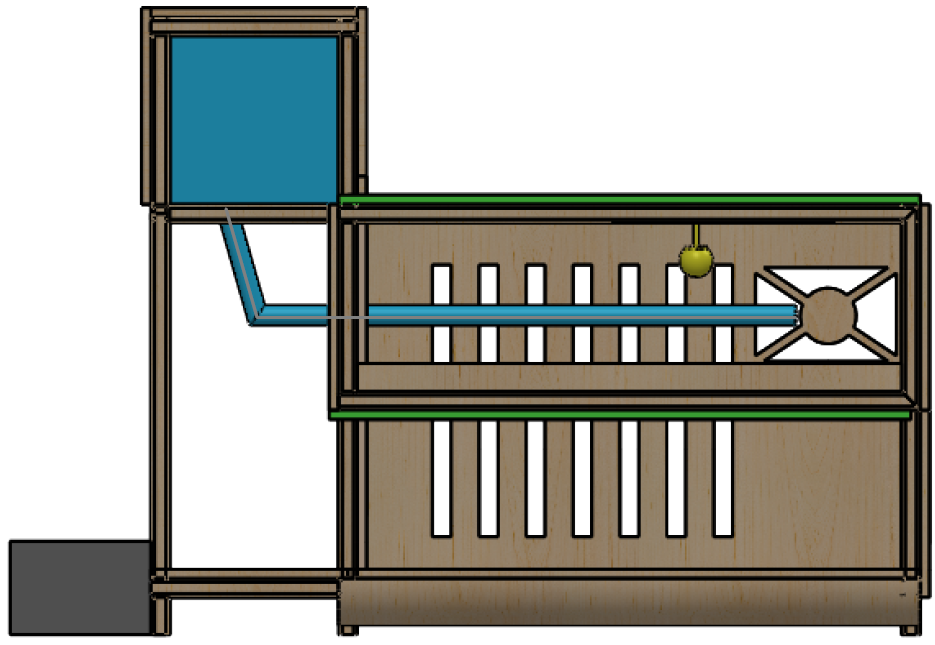
Pada Gambar 3.23 indikator kondisi kipas saat menyala pada halaman menu lampu dan kipas akan ditunjukan dengan gambar kipas yang sebelumnya tidak berwarna berubah menjadi berwarna.

## Perancangan Alat KAPILAR

Pada bagian ini akan ditampilkan rancangan alat KAPILAR yang meliputi bagian depan, atas, dan samping. Pada rancangan alat KAPILAR ini juga akan ditampilkan perangkat-perangkat yang digunakan seperti lampu, kipas, tempat pakan, tempat makan, tangki air minum dan kotak panel yang berisi NodeMCU dan perangkat lainnya.

### Tampilan Bagian Depan

Tampilan bagian depan dari rancangan alat KAPILAR akan ditampilkan pada Gambar 3.24

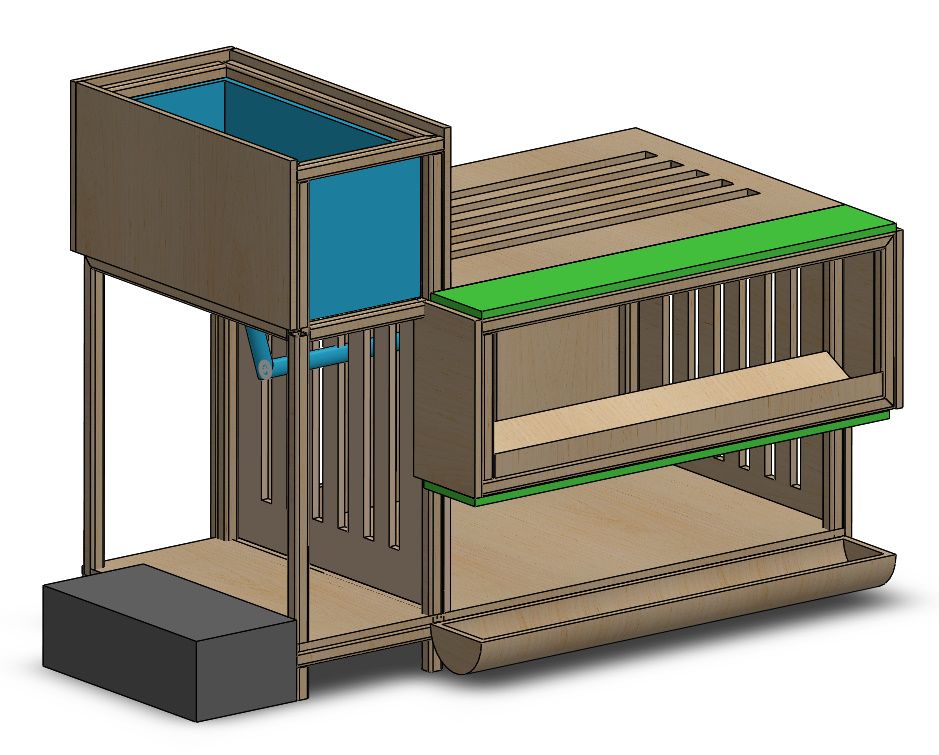


Gambar 3. Tampilan Bagian Depan

Pada Gambar 3.24 terlihat ada sebuah kipas, lampu dan pipa untuk minum ayam di dalamnya.

### Tampilan Bagian Atas

Tampilan bagian atas dari rancangan alat KAPILAR akan ditampilkan pada Gambar 3.25

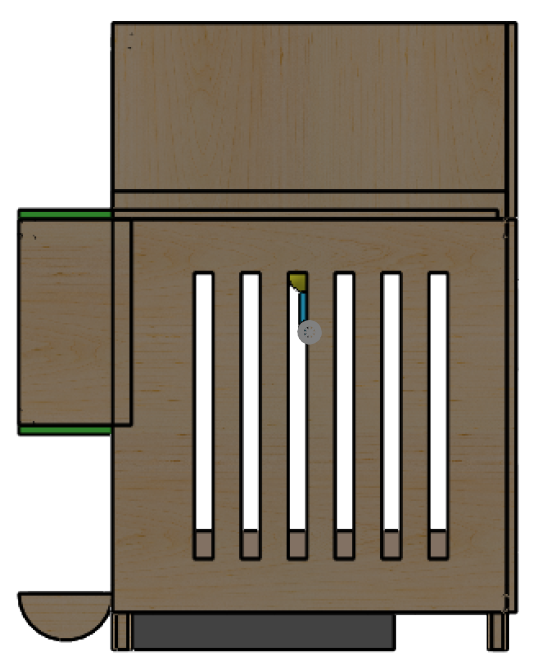


Gambar 3. Tampilan Bagian Atas

Pada Gambar 3.25 terlihattangki air minum ayam, tempat pakan dan tempat makan ayam. Terdapat pula *panel box* pada bagian bawah disamping rancangan alat. *Panel box* ini nantinya akan terisi NodeMCU yang digunakan untuk mengontrol Sistem KAPILAR.

### Tampilan Bagian Samping

Tampilan bagian samping dari rancangan alat KAPILAR akan ditampilkan pada Gambar 3.26.



Gambar 3. Tampilan Bagian Samping

Pada Gambar 3.26 terlihat bentuk tempat pakan dan tempat makan ayam serta sirkulasi dari kandang ayam*.*

## Penulisan Kode Program

Penulisan kode program sistem dan aplikasi KAPILAR dibagi menjadi dua penulisan kode program. Kode program pertama digunakan untuk memprogram NodeMCU sebagai mikrokontroler sistem KAPILAR. Kode program kedua digunakan untuk memprogram aplikasi KAPILAR berbasis android.

### Kode Program NodeMCU

Bahasa yang digunakan dalam membuat kode program Node MCU adalah Bahasa C dan diprogram menggunakan aplikasi Arduino IDE. Dalam memprogram sistem KAPILAR membutuhkan beberapa library yang didefinisikan pada Arduino IDE, berikut kode programnya:

#include <ESP8266WiFi.h> //library NodeMCU

#include <FirebaseArduino.h> //library Firebase

#include <DHT.h> //library DHT

#include <NewPing.h> //library Ultrasonic

#include <Wire.h>

#include <NTPClient.h> //library NTP

#include <WiFiUdp.h>

#include <Servo.h> //library motor servo

NodeMCU harus terhubung dengan WiFi agar dapat mengirim dan menerima data dari firebase, berikut kode programnya :

void testKoneksi\_wifi(){

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

Serial.print("Connecting to ");

Serial.print(WIFI\_SSID);

while (WiFi.status() != WL\_CONNECTED) {

Serial.print(".");

delay(500);}

Setelah NodeMCU terkoneksi dengan WiFi selanjutnya hubungkan dengan firebase yang merupakan *real-time* database yang digunakan untuk menyimpan data.

#define FIREBASE\_HOST "kapilarapp.firebaseio.com"

#define FIREBASE\_AUTH "LvegGjBZHWGleB5fACdae2388sHF0wj4lPwc6PzX"

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);

Untuk bisa mengambil waktu NodeMCU menggunakan *library NTP client* dimana waktu tersebut diambil dari server pool NTP. Berikut kode programnya :

timeClient.update();

hour = timeClient.getHours();

minute = timeClient.getMinutes();

second = timeClient.getSeconds();

Untuk menggerakkan motor servo NodeMCU membutuhkan waktu yang sudah diatur user melalui android, data waktu tersebut akan tersimpan di firebase lalu data tersebut diambil oleh NodeMCU dengan program berikut :

void ambil\_waktuFB(){

String path ="/";

FirebaseObject object = Firebase.get(path);

waktu = object.getString("waktu");}

Berikut kode program menjalankan motor servo dan mengirimkan status servo ke firebase pada void loop():

ambil\_waktuFB();

String jam\_sekarang = String (timeClient.getHours()) +":"+ String(timeClient.getMinutes());

bool fireServo = 0;

if(jam\_sekarang == waktu){

myservo.write(val);

Serial.println("katup terbuka");

fireServo = 1; //mengirim status\_servo pada firebase}

else {

digitalWrite(LEDpin2, 1);

fireServo = 0; //mengirim status\_servo pada firebase}

Berikut adalah kode program untuk monitoring serta pengontrolan suhu dan kelembaban otomatis serta mengirimkan data yang didapat ke firebase

float h = dht.readHumidity();

float t = dht.readTemperature();

bool fireLampu = 0;

bool fireKipas = 0;

if (t >= 30.00){

digitalWrite (LEDpin2, HIGH);

digitalWrite (LEDpin1, LOW);

fireKipas = 1;

fireLampu = 0;}

else if (t <= 20.00){

digitalWrite (LEDpin1, HIGH);

digitalWrite (LEDpin2,LOW);

fireKipas = 0;

fireLampu = 1;}

Firebase.setString("/suhu\_kelembaban/nilai\_kelembaban",fireHumid);

Firebase.setString("/suhu\_kelembaban/nilai\_suhu", fireTemp);

Firebase.setBool("/status\_lampu", fireLampu);

Firebase.setBool("/status\_kipas", fireKipas);

Berikut adalah kode program untuk memonitoring tangki pakan dan tangki air minum, data yang telah diterima dari sensor ultrasonic akan dikirim ke firebase.

jarak = sonar.ping\_cm();

jarakair = sonar.ping\_cm();

if(jarak>=10){

Serial.println("PAKAN AYAM HAMPIR HABIS");

delay(1000);}

if(jarakair>=10){

Serial.println("AIR MINUM HAMPIR HABIS");

delay(1000);}

Firebase.setString("/tinggi\_pakan", fireUltra);

Firebase.setString("/tinggi\_air", fireUltraAir);

### Kode Program Android

Kode program android dituliskan melalui Android Studio. Dalam memprogram tampilan antarmuka, penulisan kodenya menggunakan bahasa xml. Dan untuk menginisialisasi fungsi dan prosedur yang tersedia pada tampilan antarmuka, penulisan kodenya menggunakan bahasa java. Berikut merupakan kode program untuk membuat tampilan antarmuka dan inisialisasi fungsi aplikasi KAPILAR.

Dalam membuat tampilan grafik seperti pada grafik kondisi pakan dan air serta grafik kondisi suhu dan kelembaban perlu menambahkan beberapa *dependencies* pada file build.gradle (Module: app), berikut adalah kode programnya:

implementation **'com.android.support:cardview-v7:28.0.0'**implementation **'com.android.support:design:28.0.0'**implementation **'com.github.anastr:speedviewlib:1.3.1'**

Berikut kode program membuat tampilan grafik kondisi tangki pakan dan air:

<**android.support.v7.widget.CardView  
 android:id="@+id/cvpakan"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"**>  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Kondisi Tangki Pakan 1"  
 android:textSize="16sp"  
 android:id="@+id/tvpakan"**/>  
 <**com.github.anastr.speedviewlib.TubeSpeedometer  
 android:id="@+id/speedview"  
 android:layout\_below="@+id/tvpakan"  
 android:layout\_centerHorizontal="true"  
 android:layout\_width="180dp"  
 android:layout\_height="wrap\_content"  
 app:sv\_unit="%"  
 app:sv\_withTremble="false"** />  
 </**RelativeLayout**>  
</**android.support.v7.widget.CardView**>

Berikut adalah kode program membuat tampilan grafik kondisi suhu kelembaban:

<**com.github.anastr.speedviewlib.AwesomeSpeedometer  
 android:id="@+id/speedviewsuhu"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 app:sv\_unit="˚C"  
 app:sv\_withTremble="false"  
 app:sv\_minSpeed="10"  
 app:sv\_maxSpeed="40"  
 app:sv\_speedTextFormat="FLOAT"** />

Berikut adalah kode program yang digunakan untuk membuat pengaturan atur waktu, serta mengirimkan data waktu ke realtime database firebase:

Untuk mengirimkan suatu data ke realtime database firebase perlu ditambahkannya dependencies pada file build.gradle (Module: app):

implementation **'com.google.firebase:firebase-database:16.0.4'**

Kode program pada TimePickerFragment.java digunakan untuk membuat tampilan fragment pilih waktu, seperti berikut:

**package** com.example.kapilarapp;  
**import** android.app.Dialog;  
**import** android.app.TimePickerDialog;  
**import** android.os.Bundle;  
**import** android.support.annotation.NonNull;  
**import** android.support.v4.app.DialogFragment;  
**import** android.text.format.DateFormat;  
**import** java.util.Calendar;  
**public class** TimePickerFragment **extends** DialogFragment {  
 @NonNull  
 @Override  
 **public** Dialog onCreateDialog(Bundle savedInstanceState){  
 Calendar c = Calendar.*getInstance*();  
 **int** hour = c.get(Calendar.***HOUR\_OF\_DAY***);  
 **int** minute = c.get(Calendar.***MINUTE***);  
 **return new** TimePickerDialog(getActivity(),(TimePickerDialog.OnTimeSetListener)getActivity(),hour,minute,DateFormat.*is24HourFormat*(getActivity()));}}

Kode program pada MainActivity.java

**package** com.example.kapilarapp;  
**import** android.support.annotation.NonNull;**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.app.AlarmManager;  
**import** android.app.PendingIntent;  
**import** android.app.TimePickerDialog;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.os.Build;  
**import** android.support.v4.app.DialogFragment;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** android.widget.TimePicker;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**import** java.text.DateFormat;  
**import** java.util.Calendar;  
**public class** MainActivity **extends** AppCompatActivity **implements** TimePickerDialog.OnTimeSetListener, ValueEventListener {  
 **private** TextView **mTextView**;  
 **private** FirebaseDatabase **firebaseDatabase** = FirebaseDatabase.*getInstance*();  
 **private** DatabaseReference **mRootRef** = **firebaseDatabase**.getReference();  
 **private** DatabaseReference **mWaktuRef** = **mRootRef**.child(**"waktu"**);  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 **mWaktuRef**.addValueEventListener(**this**);}  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **mTextView** = findViewById(R.id.***alarm***);  
 Button buttonTimePicker = findViewById(R.id.***setalarm***);  
 buttonTimePicker.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 DialogFragment timePicker = **new** TimePickerFragment();  
 timePicker.show(getSupportFragmentManager(),**"Time Picker"**);}});  
 Button buttonCancelAlarm = findViewById(R.id.***cancelalarm***);  
 buttonCancelAlarm.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 cancelAlarm();}});}  
 **private void** cancelAlarm() {  
 AlarmManager alarmManager = (AlarmManager) getSystemService(Context.***ALARM\_SERVICE***);  
 Intent intent = **new** Intent(**this**, AlertReceiver.**class**);  
 PendingIntent pendingIntent = PendingIntent.*getBroadcast*(**this**, 1, intent,0);  
 alarmManager.cancel(pendingIntent);  
 **mWaktuRef**.setValue(**"0"**);  
 **mTextView**.setText(**"Alarm Canceled"**);}  
 @Override  
 **public void** onTimeSet(TimePicker view, **int** hourOfDay, **int** minute){  
 Calendar c = Calendar.*getInstance*();  
 c.set(Calendar.***HOUR\_OF\_DAY***, hourOfDay);  
 c.set(Calendar.***MINUTE***, minute);  
 c.set(Calendar.***SECOND***, 0);  
 updateTimeText(c);  
 startAlarm(c);}  
 **private void** startAlarm(Calendar c) {  
 AlarmManager alarmManager = (AlarmManager) getSystemService(Context.***ALARM\_SERVICE***);  
 PendingIntent pendingIntent = PendingIntent.*getBroadcast*(**this**, 1, intent, 0);  
 **if** (c.before(Calendar.*getInstance*())) {  
 c.add(Calendar.***DATE***, 1);}  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***KITKAT***) {  
 alarmManager.setExact(AlarmManager.***RTC\_WAKEUP***,c.getTimeInMillis(),pendingIntent);}}  
 **private void** updateTimeText(Calendar c) {  
 String timeText = DateFormat.*getTimeInstance*(DateFormat.***SHORT***).format(c.getTime());  
 **mWaktuRef**.setValue(timeText);}  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **if**(dataSnapshot.getValue(String.**class**)!=**null**){  
 String key = dataSnapshot.getKey();  
 **if** (key.equals(**"waktu"**)){  
 String waktu = dataSnapshot.getValue(String.**class**);  
 **if**(!waktu.equals(**"0"**)){  
 **mTextView**.setText(waktu);}}}}  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {}}

Berikut adalah kode program untuk membaca data dari firebase dan menampilkannya pada grafik

**package** com.example.kapilarapp;  
**import** android.support.annotation.NonNull;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** com.github.anastr.speedviewlib.TubeSpeedometer;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
**public class** KondisiPakanActivity **extends** AppCompatActivity {  
 **private** FirebaseDatabase **firebaseDatabase** = FirebaseDatabase.*getInstance*();  
 **private** DatabaseReference **mRootRef** = **firebaseDatabase**.getReference();  
 String **tinggi\_pakan**;  
 **double sisa\_pakan**;  
 **double rumus**;  
 TubeSpeedometer **pointerSpeedometer**;  
 TubeSpeedometer **pointerSpeedometer2**;  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 **mRootRef**.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **tinggi\_pakan** =dataSnapshot.child(**"tinggi\_pakan"**).getValue(String.**class**);  
 **sisa\_pakan** = 30 - Double.*valueOf*(**tinggi\_pakan**);  
 **rumus** = Double.*valueOf*(**sisa\_pakan**)/30\*100;  
 **pointerSpeedometer**.speedTo((**float**) **rumus**);  
 **pointerSpeedometer2**.speedTo((**float**) **rumus**);}  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {}});}  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_kondisi\_pakan***);  
 **pointerSpeedometer** = findViewById(R.id.***speedview***);  
 **pointerSpeedometer2** = findViewById(R.id.***speedview2***);  
 **pointerSpeedometer**.setMaxSpeed(100);  
 **pointerSpeedometer2**.setMaxSpeed(100);}}

## Perancangan Pengujian Sistem

Perancangan pengujian sistem KAPILAR terbagi menjadi dua perancangan pengujian, yaitu perancangan uji alat dan perancangan pengujian user.

### Perancangan Uji Alat

1. Rancangan Uji Alat Sistem KAPILAR
2. Motor Servo (Membuka dan menutup katup pakan sesuai jadwal)

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Sebagai pembuka dan penutup katup pakan ayam sesuai dengan jadwal yang telah diatur melalui Android. |
| b. | Cara Menguji: | Menghubungkan motor servo dengan vcc, ground, dan pin *output*  pada NodeMCU. Lalu tuliskan kode program yang berisi inisialisasi pin-pin motor servo. Kemudian *upload* kode program yang berisi kondisi yang memungkinkan motor servo untuk membuka dan menutup ke NodeMCU. |

1. Sensor DHT11 (Pengukuran suhu dan kelembaban kandang ayam)

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Sebagai pengukur dan pengontrol otomatis suhu serta kelembaban pada kandang ayam petelur. |
| b. | Cara Menguji: | Menghubungkansensor DHT11 dengan vcc, ground, dan pin *output* pada NodeMCU. Lalu upload kode program yang berisi inisialisasi pin – pin serta kode program untuk membaca suhu dan kelembaban ke NodeMCU. Apabila nilai suhu kurang dari 20oC maka NodeMCU akan memberi perintah untuk menyalakan lampu secara otomatis dan bila suhu lebih dari 30oC maka NodeMCU akan memberi perintah untuk menyalakan kipas secara otomatis. |

1. Sensor Ultrasonic (Pengukuran kondisi pakan dan minum berdasarkan ketinggian)

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Sebagai pengukur kondisi pakan dan minum dengan membaca ketinggian tangki pakan dan minum. |
| b. | Cara Menguji: | Menghubungkan sensor ultrasonic dengan NodeMCU dan *upload* kode program yang berisi kondisi apabila sensor bernilaikurang dari 10 cm maka NodeMCU akan mengirimkan pemberitahuan ke Android. |

1. Rancangan Uji Aplikasi KAPILAR
2. Pengujian *Setting* Alarm Pakan

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Untuk mengontrol jadwal pemberian pakan ayam petelur melalui android dari jarak berapapun dan dimanapun. |
| b. | Cara Menguji: | Memasukkan waktu pemberian pakan melalui aplikasi KAPILAR di *smartphone* Android. |

1. Pengujian Menu Monitoring Tempat Pakan 1 dan 2

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Mengetahui kondisi pakan ayam berdasarkan ketingian |
| b. | Cara Menguji: | Memilih menu Cek Kondisi Pakan yang terdapat pada halaman menu utama |

1. Pengujian Menu Monitoring Tangki Air Minum

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Mengetahui kondisi air minum ayam berdasarkan ketingian |
| b. | Cara Menguji: | Memilih menu Cek Kondisi Air yang terdapat pada halaman menu utama |

1. Pengujian Menu Monitoring Suhu dan Kelembaban

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Mengetahui kondisi suhu dan kelembaban kandang ayam |
| b. | Cara Menguji: | Memilih menu Suhu Kelembaban yang terdapat pada halaman menu utama |

1. Pengujian Menu Monitoring Kondisi Lampu dan Kipas

|  |  |  |
| --- | --- | --- |
| a. | Tujuan: | Mengetahui kondisi lampu dan kipas yang berada pada kandang ayam |
| b. | Cara Menguji: | Memilih menu Cek Lampu Kipas yang terdapat pada halaman menu utama |

Berikut ini terdapat daftar pengujian dalam bentuk tabel uji pengujian alat dan aplikasi seperti yang terlihat pada Tabel 3.1 dan Tabel 3.2 berikut :

Tabel 3. Rekap Daftar Pengujian Alat

|  |  |  |  |
| --- | --- | --- | --- |
| No | Pengujian | Tujuan | Indikator |
| 1 | Pengujian Motor Servo | Sebagai pembuka dan penutup katup pakan ayam sesuai dengan jadwal yang telah diatur melalui Android | Indikasi pengujian ini berhasil apabila motor servo dapat membuka dan menutup katup tempat pakan sesuai dengan jadwal yang telah diberikan. |
| 2 | PengujianSensor DHT11 | Sebagai pengukur dan pengontrol otomatis suhu serta kelembaban pada kandang ayam petelur. | Indikasi pengujian ini berhasil apabila sensor DHT11 dapat membaca nilai suhu dan kelembaban kandang ayam serta jika suhu kurang dari 20oC maka lampu akan menyala secara otomatis dan jika suhu lebih dari 30oC maka kipas akan menyala secara otomatis. |
| 3 | Pengujian Sensor Ultrasonic | Sebagai pengukur kondisi pakan dan minum dengan membaca ketinggian tangki pakan dan minum. | Indikasi pengujian ini berhasil apabila sensor ultrasonic menampilkan nilai kondisi tangki pakan dan air minum secara *real-time* pada Android. |

Tabel 3. Rekap Daftar Pengujian Aplikasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Pengujian | Tujuan | Indikator |
| 1 | Pengujian *Setting* Alarm Pakan | Untuk mengontrol jadwal pemberian pakan ayam petelur melalui android dari jarak berapapun dan dimanapun. | Indikasi pengujian ini berhasil apabila dapat memasukkan jadwal waktu pemberian pakan ayam. |
| 2 | Pengujian Monitoring Tempat Pakan 1 dan 2 | Mengetahui kondisi pakan ayam berdasarkan ketingian | Indikasi pengujian ini berhasil apabila Android dapat menampilkan kondisi tangki pakan secara *real-time.* |
| 3 | Pengujian Monitoring Tangki Air Minum | Mengetahui kondisi air minum ayam berdasarkan ketingian | Indikasi pengujian ini berhasil apabila Android dapat menampilkan kondisi tangki air minum secara *real-time.* |
| 4 | Pengujian Monitoring Suhu dan Kelembaban | Mengetahui kondisi suhu dan kelembaban kandang ayam | Indikasi pengujian ini berhasil apabila Android dapat menampilkan kondisi suhu dan kelembaban kandang ayam secara *real-time.* |
| 5 | Pengujian Menu Monitoring Kondisi Lampu dan Kipas | Mengetahui kondisi lampu dan kipas yang berada pada kandang ayam | Indikasi pengujian ini berhasil apabila sistem aplikasi Android dapat menampilkan kondisi lampu dan kipas yang berada pada kandang ayam |

### Perancangan Pengujian User

Pada tahapan ini dilakukan proses pengujian sistem yang sudah jadi kepada *user*. Dalam hal ini penulis menggunakan sampel 10 orang untuk melakukan pengujian sistem. Dalam pengujian sistem ini didapatkan data dengan cara memberikan beberapa poin pertanyaan seputar sistem ini serta mengharapkan kritik dan saran demi perkembangan selanjutnya.

Dalam pengujian alat Sistem KAPILAR ini diberikan beberapa pertanyaan untuk mendapatkan hasil perhitungan yang digunakan sebagai acuan untuk mengetahui tingkat kepuasan pengguna dalam penggunaannya. Untuk daftar pertanyan dapat dilihat pada Tabel 3.3, sedangkan untuk penilaian pengukuran tingkat kepuasan pengguna didasarkan pada Tabel 3.3 dan Tabel 3.4.

Tabel 3. Daftar Pertanyaan

|  |  |
| --- | --- |
| **No** | **Kriteria Pertanyaan** |
| 1 | Apakah anda setuju bahwa Sistem KAPILAR mempermudah peternak ayam petelur dalam melakukan pengontrolan kandang ayam petelur dengan jarak jauh? |
| 2 | Apakah anda setuju bahwa Sistem KAPILAR mempermudah peternak ayam petelur dalam melakukan memonitoring kandang ayam petelur dengan jarak jauh? |
| 3 | Apakah anda setuju bahwa Aplikasi KAPILAR memiliki antarmuka yang mudah dipahami? |
| 4 | Apakah anda setuju bahwa katup pakan ayam terbuka sesuai dengan jadwal yang telah diberikan? |
| 5 | Apakah anda setuju bahwa Aplikasi KAPILAR memonitoring tangki pakan dengan akurat? |
| 6 | Apakah anda setuju bahwa Aplikasi KAPILAR memonitoring tangki air minum dengan akurat? |
| 7 | Apakah anda setuju bahwa Aplikasi KAPILAR memonitoring suhu dan kelembaban dengan akurat? |
| 8 | Apakah anda setuju bahwa sistem KAPILAR mengontrol suhu dan kelembaban secara otomatis? |
| 9 | Apakah anda setuju bahwa sistem KAPILAR mengontrol lampu dengan akurat? |
| 10 | Apakah anda setuju bahwa sistem KAPILAR mengontrol kipas dengan akurat? |

Tabel 3. Indikator Penilaian

|  |  |
| --- | --- |
| **Predikat** | **Nilai** |
| Sangat Baik | 5 |
| Baik | 4 |
| Cukup | 3 |
| Kurang | 2 |
| Buruk | 1 |

Tabel 3. Indikator Kepuasan Pengguna Aplikasi

|  |  |
| --- | --- |
| **Kategori** | **Presentase** |
| Sangat Memuaskan | 81% - 100% |
| Memuaskan | 61% - 80% |
| Cukup Memuaskan | 41% - 60% |
| Tidak Memuaskan | 21% - 40% |
| Sangat Tidak Memuaskan | 0-20 % |

Untuk menghitung tingkat kepuasan pengguna menggunakan rumus sebagai berikut:

Presentase kepuasan pengguna (%) =

# BAB IV ANALISIS DAN PEMBAHASAN

Pada bab ini akan membahas mengenai analisis alat dan aplikasi yang telah dibuat serta pengujian alat dan aplikasi secara langsung kepada pengguna yaitu peternak ayam petelur.

## Analisis Sistem

Pada bagian ini berisi mengenai analisis Sistem KAPILAR berbasis NodeMCU sebagai sistem pengontrol pakan ayam dan suhu serta memonitoring tangki pakan dan tangki air minum secara *real time* menggunakan aplikasi android.

### Analisis Alat

Untuk membuat sistem KAPILAR komponen – komponen utama yang dibutuhkan yaitu NodeMCU sebagai mikrokontroler, motor servo, sensor DHT11, dan sensor ultrasonic. Selain itu juga terdapat beberapa komponen pendukung seperti relay, kipas dan lampu sebagai *output*. Motor servo ini berjumlah 1 buah di setiap tempat pakan, lalu akan membuka dan menutup katup pakan ayam sesuai dengan jadwal yang telah diatur melalui Android. Motor servo bergerak 1800 untuk membuka katup pakan dan bergerak selama beberapa kali dalam 1 menit. Jadwal pemberian pakan diatur oleh *user* melalui Android dan datanya akan disimpan di *Firebase.* Data waktu tersebut akan dibaca oleh NodeMCU dan akan membuka katup pakan sesuai dengan data yang telah diberikan.

Padasensor DHT11, setiap detik suhu dan kelembaban di sekitar kandang ayam akan diukur dan datanya akan disimpan ke Firebase. Nilai suhu dan kelembaban tersebut dapat mengaktifkan kipas atau lampu secara otomatis bila NodeMCU membaca suhu disekitar kandang kurang dari 200C maka lampu akan menyala otomatis, dan bila suhu disekitar kandang lebih dari 300C maka akan menyalakan kipas. Data suhu dan kelembaban yang tersimpan pada Firebase akan ditampilkan di Android sehingga user dapat memonitoring suhu dan kelembaban kandang ayam. User juga dapat memonitoring kondisi lampu dan kipas apakah menyala atau tidak melalui Android.

Sensor ultrasonic berfungsi sebagai pengukur kondisi pakan dan minum dengan membaca ketinggian tangki pakan dan minum. Sensor ini bekerja dengan memancarkan gelombang ultrasonic yang kemudian akan memantul kembali ke sensor. Pada sistem KAPILAR sensor ultrasonic akan mengukur jarak berdasarkan ketinggian permukaan pakan dan air minum, lalu nilai tersebut akan disimpan di firebase dan ditampilkan pada Android sehingga user dapat memonitoring kondisi tangki pakan dan tangki air minum ayam petelur. Bila ketinggian pakan ayam lebih dari 14 cm maka Android akan mengirimkan notifikasi pada user agar user dapat mengisi kembali tangki pakan ayam. Bila ketinggian air minum lebih dari 15 cm maka Android juga akan mengirimkan notifikasi pada user sehingga user dapta mengisi kembali tangki air minum.

### Analisis Aplikasi Android

Aplikasi android KAPILAR digunakan untuk mengontrol serta memonitoring pakan ayam dan suhu pada kandang ayam petelur. Aplikasi KAPILAR terdiri dari beberapa fitur yaitu, pengaturan alarm, status servo, cek kondisi pakan, cek kondisi air, suhu dan kelembaban, dan cek lampu kipas.

Kontrol pakan ayam menggunakan alarm berjadwal yang di inputkan oleh user melalui Android. User harus mensetting waktu pemberian pakan ayam yang diinginkan melalui fitur pengaturan alarm, lalu data waktu tersebut akan tersimpan di Firebase. Selanjutnya data tersebut akan diambil oleh NodeMCU lalu bila waktu pemberian pakan telah tiba maka NodeMCU akan menggerakkan servo untuk membuka katup pakan ayam. Node MCU juga mengambil kondisi servo dan mengirimkannya ke Firebase untuk mengubah status servo yang ada di fitur aplikasi KAPILAR. Status tersebut akan berubah menjadi “ON” dan akan berubah menjadi “OFF” bila servo menutup katup pakan kembali. User juga dapat membatalkan jadwal pakan sehingga waktu yang telah tersimpan di firebase akan terhapus. Waktu yang telah di setting juga akan ditampilkan pada Android dan user akan menerima notifikasi bila jadwal pemberian pakan telah tiba.

Fitur cek kondisi pakan terdiri dari kondisi tangki-tangki pakan yang ada di kandang ayam. Dengan fitur ini user dapat memonitoring kondisi tangki pakan ayam apakah masih banyak atau tinggal sedikit. Fitur ini menampilkan kondisi pakan pada android dengan tampilan speedview dan berupa persentasi dari nilai ketinggian permukaan pakan. Persentasi tersebut didapat dengan menggunakan rumus yang diletakkan pada kode program Android. Nilai keetinggian pakan diambil dari firebase yang dikirimkan Node MCU setelah ultrasonic membaca kondisi ketinggian pakan ayam. Bila ketinggian pakan lebih dari 14 cm maka Node MCU akan mengirimkan notifikasi pada user. Pada fitur cek kondisi air juga memiliki cara kerja dan tampilan yang sama dengan fitur cek kondisi pakan perbedaannya terdapat pada material yang diukur dan indikator tinggi minimal untuk mengirimkan notifikasi. Pada fitur cek kondisi air tinggi minimalnya adalah 15 cm, bila lebih dari itu maka Node MCU akan mengirimkan notifikasi pada user.

Pada fitur suhu dan kelembaban akan menampilkan nilai kondisi suhu dan kelembaban pada kandang ayam. Nilai tersebut diambil dari sensor DHT11 yang dibaca oleh Node MCU lalu mengirimkan nilainya ke firebase lalu android mengambil nilai itu dan menampilkannya ke user. Nilai suhu yang ditampilkan menggunakan derajat Celcius sedangkan untuk kelembaban menggunakan persentase.

Terakhir, ada fitur lampu dan kipas yang menampilkan kondisi lampu dan kipas yang ada pada kandang ayam. Lampu akan menyala secara otomatis bila suhu disekitar kandang ayam kurang dari 200C sedangkan kipas akan menyala secara otomatis bila suhu disekitar kandang ayam lebih dari 300C.

## Pembahasan Sistem

### Implementasi Hardware

Komponen – komponen utama yang digunakan pada *hardware* Sistem KAPILAR adalah sebagai berikut :

1. NodeMCU merupakan komponen utama dalam Sistem KAPILAR. NodeMCU adalah mikrokontroler yang akan memberikan perintah dalam hal kontrol pakan dan suhu serta kelembaban, mengirim data ke Firebase, dan mengambil data dari Firebase untuk membuka dan menutup katup pakan ayam sesuaidengan jadwal yang telah user berikan melalui android. NodeMCU memiliki 13 pin GPIO (*General Purpose Input Output*), 4 Mb memori dan *Wi*-*Fi*.
2. Motor servo digunakan sebagai motor penggerak yang akan membuka dan menutup katup ayam. Motor servo bergerak sesuai dengan perintah dari NodeMCU. Motor servo akan bergerak jika waktu yang dimiliki NodeMCU sama dengan nilai waktu yang diambil dari firebase. Motor servo akan terbuka sebesar 1800 dan kembali menutup dengan kondisi awal 00.
3. Sensor DHT11 digunakan sebagai pengukur dan pengontrol otomatis suhu serta kelembaban pada kandang ayam petelur. Nilai suhu dan kelembaban yang telah diukur oleh sensor ini akan dikirimkan ke firebase oleh NodeMCU lalu ditampilkan pada Android user. Bila sensor DHT11 membaca suhu kurang dari 200C maka NodeMCU akan memberikan kondisi 1 pada lampu sehingga lampu akan menyala. Bila sensor membaca suhu lebih dari 300C maka NodeMCU akan memberikan kondisi 1 pada kipas sehingga kipas menyala.
4. Sensor ultrasonic
5. Relay merupakan saklar yang dikontrol oleh NodeMCU untuk mematikan dan menghidupkan lampu dan kipas pada system KAPILAR.

# BAB V KESIMPULAN

# DAFTAR PUSTAKA

# LAMPIRAN