

**RANCANG BANGUN PERANGKAT LUNAK SISTEM MONITORING
DAN KONTROL AIR MINUM PADA KANDANG BURUNG PUYUH
MENGUNAKAN IOT BERBASIS ANDROID**

SKRIPSI



Oleh :
RIFQI RAHMADHANI
NIM. 16650037

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**RANCANG BANGUN PERANGKAT LUNAK SISTEM MONITORING
DAN KONTROL AIR MINUM PADA KANDANG BURUNG PUYUH
MENGUNAKAN IOT BERBASIS ANDROID**

SKRIPSI

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelas Sarjana Komputer (S.Kom)**

**Oleh :
RIFQI RAHMADHANI
NIM. 16650037**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN

RANCANG BANGUN PERANGKAT LUNAK SISTEM MONITORING DAN KONTROL AIR MINUM PADA KANDANG BURUNG PUYUH MENGUNAKAN IOT BERBASIS ANDROID

SKRIPSI

RIFQI RAHMADHANI
NIM. 16650037

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : Desember 2020

Dosen Pembimbing I

Dosen Pembimbing II

(Prof.Dr. Suhartono, M.Kom)
NIP. 19680519 200312 1 001

(Dr. Fachrul Kurniawan, M.MT)
NIP. 19771020 200912 1 001

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

RANCANG BANGUN PERANGKAT LUNAK SISTEM MONITORING DAN KONTROL AIR MINUM PADA KANDANG BURUNG PUYUH MENGUNAKAN IOT BERBASIS ANDROID

SKRIPSI

Oleh :
RIFQI RAHMADHANI
NIM. 16650037

Telah Dipertahankan Di depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal Desember 2020

Susunan Penguji

Tanda Tangan

Penguji I	:	<u>Roro Inda Melani, S.Kom., M.Sc</u> NIP. 197809252005012008	()
Penguji II	:	<u>Yunifa Miftachul Arif, M.T.</u> NIP. 19830616 201101 1 004	()
Pembimbing I	:	<u>Prof.Dr. Suhartono, M.Kom</u> NIP.19680519 200312 1 001	()
Pembimbing II	:	<u>Dr. Fachrul Kurniawan, M.MT.</u> NIP. 19771020 200912 1 001	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

**PERNYATAAN
KEASLIAN TULISAN**

Saya yang bertanda tangan di bawah ini:

Nama : Rifqi Rahmadhani

NIM : 16650037

Fakultas/Jurusan : Sains dan Teknologi/Teknik Informatika

Judul Skripsi : **Rancang Bangun Perangkat Lunak Sistem Monitoring Dan Kontrol Air Minum Pada Kandang Burung Puyuh Menggunakan IOT Berbasis Android**

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila kemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Desember 2020

Yang membuat pernyataan,



(Rifqi Rahmadhani)

NIM. 16650037

MOTTO

“Just do it and never give up”



HALAMAN PERSEMBAHAN

الحمد لله رب العالمين

Puji syukur atas kehadiran Allah SWT

Shalawat serta salam kepada Rasulullah SAW

Dengan segenap hati, penulis mempersembahkan sebuah karya ini kepada:

Kedua orang tua penulis tercinta, Bapak Mulyanto dan Ibu minarsih yang selalu dengan senang hati membimbing dan memotivasi diri penulis, yang selalu mendo'akan penulis, yang selalu mendukung dan memberi kasih sayang yang tak terhingga hingga kapan pun.

Dosen pembimbing penulis, Bapak Prof.Dr. Suhartono,M.Kom.,_dan Bapak Dr. Fachrul Kurniawan, M.MT yang telah dengan sabar membimbing jalannya penelitian skripsi ini dan selalu memberikan stimulus positif untuk tetap semangat menjalani setiap tahap ujian skripsi.

Dosen penguji, Ibu Roro Ina Melani, S.Kom., M.Sc dan Bapak Yunifa Miftachul Arif, M.T telah membimbing dan memberikan masukan kepada penulis sehingga tercapai hasil skripsi yang lebih baik.

Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, dan seluruh guru-guru penulis yang telah membimbing dan memberikan ilmunya yang sangat bermanfaat.

Keluarga Teknik Informatika UIN Maulana Malik Ibrahim Malang, terutama keluarga Andromeda (Angkatan 2016) yang telah memberikan semangat dan doanya.

Penulis ucapkan “jazakumullah khairan katsiiraan”. Semoga ukhuwah kita tetap terjaga dan selalu diridhoi Allah SWT. Aamiin Ya Rabbal ‘Alamiin

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur kami panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “ Rancang Bangun Perangkat Lunak Sistem Monitorin Dan Kontrol Air Minum Pada Kandang Burung Puyuh Dengan Menggunakan *Internet Of Things* Berbasis Android” ini dengan lancar dan baik. Dan tak lupa sholawat serta salam selalu kami haturkan kepada junjungan kita Nabi Muhammad SAW yang telah mengantarkan manusia dari zaman kegelapan ke zaman yang terang benderang ini. Tujuan dari penyusunan skripsi ini guna memenuhi salah satu syarat untuk bisa menempuh ujian sarjana komputer pada Fakultas Sains dan Teknologi (FSAINTEK) Program Studi Teknik Informatika di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Tentu saja selama proses pengerjaan skripsi ini penulis mendapat banyak sekali dukungan dan bantuan dari berbagai pihak. Oleh karena itu, di sini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. Abdul Haris, M.Ag selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crys dian, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Prof. Dr. Suhartono, M.Kom dan Dr. Fachrul Kurniawan, M.MT, selaku Dosen Pembimbing yang telah dengan sabar membimbing penulis, mem-

berikan masukan serta arahan sehingga penulis tidak hanya mampu menyelesaikan skripsi namun juga dapat memberikan nilai manfaat nantinya dari skripsi ini.

5. Roro Inda Melani, S.Kom., M.Sc dan Yunifa Miftachul Arif, M.T , selaku Dosen Penguji yang telah membimbing dan memberikan masukan kepada penulis sehingga tercapai hasil skripsi yang lebih baik.
6. Roro Inda Melani, S.Kom., M.Sc, selaku Dosen Wali yang senantiasa memberikan banyak motivasi dan saran untuk kebaikan penulis.
7. Orang tua tercinta yang tak pernah berhenti untuk memberikan do'a dan dukungan kepada penulis secara moril maupun materiil hingga skripsi ini dapat terselesaikan.
8. Anggota keluarga dan kerabat yang senantiasa memberikan do'a dan dukungan semangat kepada penulis.
9. Teman-teman Andromeda (Angkatan 2016) yang selalu memberikan semangat dan do'a kepada penulis.
10. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak bisa penulis sebutkan semuanya tanpa mengurangi rasa hormat dan terima kasih.
11. Dan tak lupa pada diri penulis sendiri yang sudah berusaha mengupayakan yang terbaik, yang tak pernah menyerah meski banyak rintangan yang dihadapi, yang selalu bersemangat dalam mengupayakan yang terbaik, dan yang selalu percaya bahwa dari setiap proses tak ada yang sia-sia.

Penulis sadar bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan masih jauh dari kata sempurna. Namun penulis berharap semoga skripsi ini dapat memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Malang, Desember 2020

Penulis

DAFTAR ISI

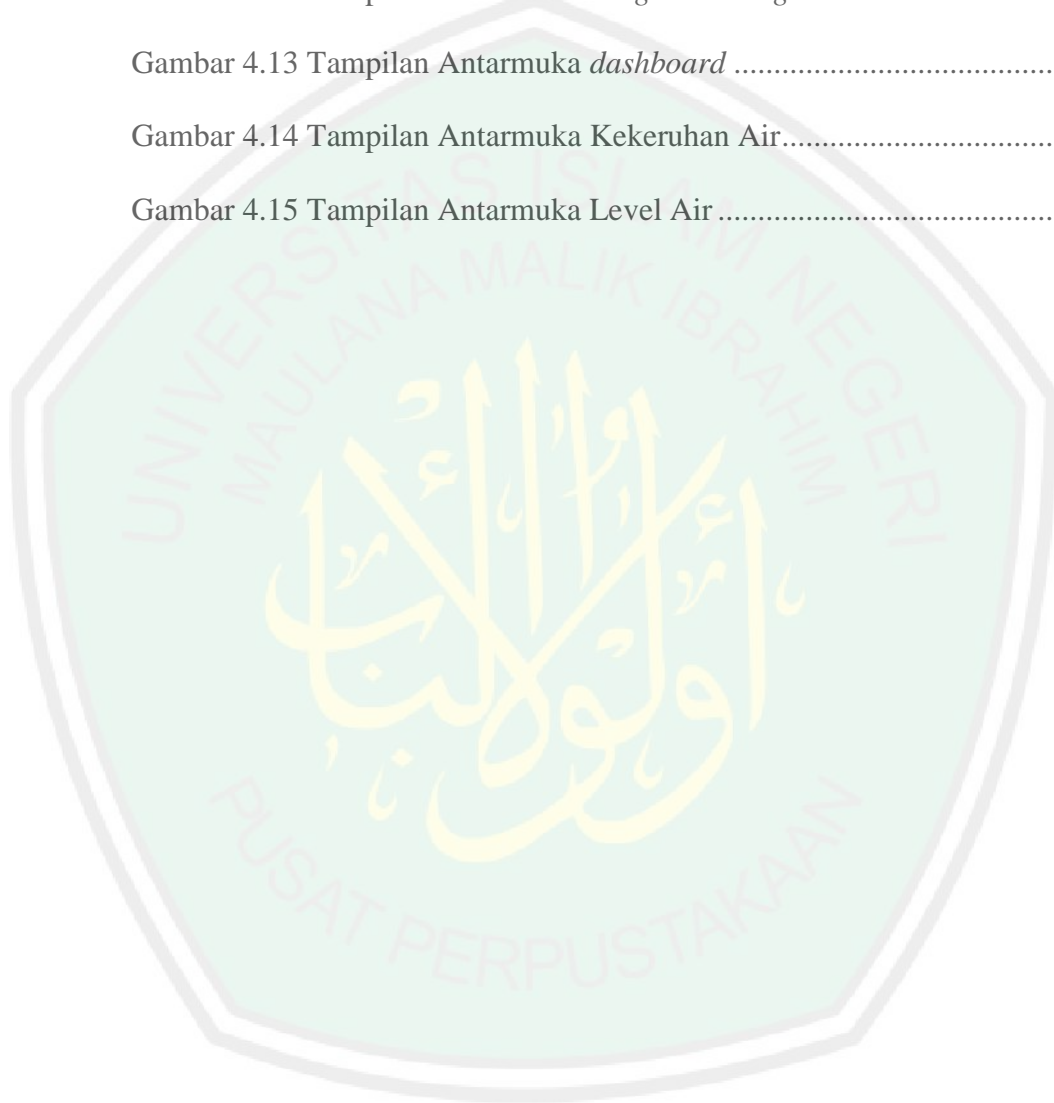
LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN KEASLIAN TULISAN.....	iii
MOTTO.....	iv
HALAMAN PERSEMBAHAN.....	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
ABSTRAK	xiii
ABSTRACT.....	xiv
المخلص	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah.....	4
1.5 Manfaat Penelitian	4
BAB II STUDI PUSTAKA.....	5
2.1 Penelitian Terkait	5
2.2 Landasan Teori.....	7
2.2.2. Android	7
2.2.3. Firebase	9
2.2.4 Internet Of Things	10
2.2.5 Mikrokontroler Node Mcu Esp8266	12
2.2.6 Model Proses Pembangunan Perangkat Lunak (<i>Waterfall</i>).....	14
BAB III METODE PENELITIAN.....	18
3.1 Desain Sistem.....	18
3.1.1 Perangkat Lunak Kontrol dan Monitor Air Minum Puyuh.....	18
3.1.1.1 Pembuatan Model Bisnis.....	18
3.1.1.2 Prototype Perangkat Lunak	23

3.1.1.3 Perancangan data.....	28
3.2 Pengujian Sistem.....	29
BAB IV HASIL DAN PEMBAHASAN	31
4.1 Implementasi Program	31
4.1.1 Kebutuhan implementasi.....	31
4.1.2 Struktru Pemograman dan kode program	32
4.1.3 Implementasi Jadwal Monitoring.....	38
4.1.4 Implementasi Kontroling Air Minum	46
4.2 Implementasi <i>Interface</i>	47
4.2.1 Implementasi Tampilan <i>Splash Screen</i>	47
4.2.2 Implementasi Antarmuka <i>Login dan Register</i>	48
4.2.3 Implementasi Antar muka <i>Dashboard</i>	49
4.2.4 Implementasi Antar muka Monitoring Kekeruhan	50
4.2.5 Implementasi Antar muka Monitoring Level Air	51
4.3 Pengujian Sistem.....	52
4.3.1 Metode Pengujian Sistem.....	52
4.3.2 Kasus dan Hasil Pengujian Fungsional	53
BAB V PENUTUP.....	59
5.1 Kesimpulan	59
5.2 Saran.....	60
DAFTAR PUSTAKA	61

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Android	9
Gambar 2.2. Sistem Kerja <i>Firebase</i>	10
Gambar 2.3. Cara Kerja IoT.....	12
Gambar 2.4. <i>NodeMCU</i> Esp8266.....	13
Gambar 2.5. Skematik posisi Pin <i>NodeMCU</i> Dev Kit v3	14
Gambar 2.6 <i>Waterfall</i>	16
Gambar 3.1 Diagram aktivitas Login dan Register.....	19
Gambar 3.2 Diagram Aktivitas Kontroling Air Minum Puyuh	20
Gambar 3.3 Activity Diagram Monitor Level Air	21
Gambar 3.4 Activity Diagram Monitor Keruh Air	22
Gambar 3.5 Use Case Diagram.....	23
Gambar 3.6 Tampilan Splash Screen.....	24
Gambar 3.7 Tampilan <i>Login</i> dan <i>Register</i>	25
Gambar 3.8 Tampilan <i>Dashboard</i>	26
Gambar 3.8 Tampilan Halaman Data Kekeruhan	27
Gambar 3.9 Tampilan Halaman Data Level	28
Gambar 3.10 Struktur kelas model.....	29
Gambar 4.1 Program Data Login.....	34
Gambar 4.2 Program Data Register	34
Gambar 4.3 Program Data Realtime	35
Gambar 4.4 Program Mendapatkan Data Sensor	35
Gambar 4.5 Program Kontrol Status Pompa.....	36
Gambar 4.6 Kode Program Monitoring Data keruh Perjam	37
Gambar 4.7 Program Menghitung Rata-Rata Data Perhari	38

Gambar 4.8 Kode Program Tampil Dataperhari	38
Gambar 4.9 Prototype Alat Uji Coba.....	41
Gambar 4.10 Grafik Level Air pada Wadah Air Minum Burung Puyuh.....	44
Gambar 4.11 Grafik Keruh Air pada Wadah Air Minum Burung Puyuh.....	46
Gambar 4.12 Tampilan Antar Muka <i>Login and Register</i>	48
Gambar 4.13 Tampilan Antarmuka <i>dashboard</i>	50
Gambar 4.14 Tampilan Antarmuka Kekerusuhan Air.....	51
Gambar 4.15 Tampilan Antarmuka Level Air	52



DAFTAR TABEL

Tabel 3.1 Perancangan data.....	28
Tabel 3.2 Pengujian Fungsional Perangkat Lunak.....	30
Tabel 4.1 Spesifikasi Software dan Hardware	31
Tabel 4.2 <i>File Sourcode</i> Pembangunan Halaman Monitoring Level Air	38
Tabel 4.3 Data Kondisi Wadah Air Minum Puyuh dengan “Status Level”	40
Tabel 4.4 Data Kondisi Wadah Air Minum Puyuh dengan “Status Keruh”	43
Tabel 4.5 Aturan Kualifikasi Kekeruhan Air Minum	46
Tabel 4.7 Data Pengujian	52
Tabel 4.8 Hasil Pengujian Fungsionalitas Splash Screen	53
Tabel 4.9 Hasil Pengujian Fungsionalitas <i>Login</i> dan <i>Register</i>	54
Tabel 4.10 Hasil Pengujian Fungsionalitas <i>Dashboard</i>	55
Tabel 4.11 Hasil Pengujian Fungsionalitas Monitor Level.....	56
Tabel 4.12 Hasil Pengujian Fungsionalitas Monitor Level.....	57



ABSTRAK

Rahmadhani, Rifqi. 2020. **Rancang Bangun Perangkat Lunak Sistem Monitorin Dan Kontrol Air Minum Pada Kandang Burung Puyuh Dengan Menggunakan *Internet Of Things* Berbasis Android.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, M.Kom (II) Dr. Fachrul Kurniawan, M.T.

Kata Kunci : Air Minum Burung Puyuh, Perangkat Lunak, NodeMCU, *Internet of Thing*.

Usaha peternakan puyuh merupakan usaha yang menjajinkan. Banyak masyarakat Indonesia yang mengeluti dalam usaha tersebut. Namun dalam pengelolaan peternak puyuh ternyata tidak mudah, tentu banyak factor-faktor yang perlu diperhatikan dalam perawatan peternak puyuh. Salah satunya adalah air minum burung puyuh, air minum burung puyuh mempunyai peran yang signifikan terhadap Kesehatan dan produktifitas burung puyuh. Oleh karena itu peternak memerlukan pemantaun khusus terhadap kondisi air minum burung puyuh. monitoring dan kontrol air minum burung puyuh menggunakan sistem *Internet Of Things* berbasis Android adalah suatu alat yang digunakan untuk memantau dan mengontrol kondisi air pada wadah air minum burung puyuh. didalam perangkat lunak terdapat beberapa fitur yang dapat membantu peternak dalam memantau kondisi air minum burung puyuh, antara lain fitur monitoring kekeruhan dan level air, yang mana peternak dapat memantau kondisi air minum burung puyuh secara realtime dan berkala perjam, perhari, perminggu dengan data-data sensor yang berada pada perangkat keras yang dikirimkan oleh NodeMcu ke *database*. Dalam perangkat lunak ini juga terdapat fitur kontrol pompa pengisian dan pengurasan air minum burung puyuh yang dapat digunakan peternak dalam mengisi dan menguras pada wadah penampungan air minum burung puyuh. peneliti melakukan uji coba monitoring selama 1 minggu yang dilakukan dengan menggunakan 4 ekor burung puyuh sebagai bahan uji coba dan wadah air berukuran 2 liter didapatkan hasil untuk konsumsi air minum burung puyuh setiap harinya yaitu 0.6 liter air atau 600 mililiter. Untuk kekeruhan air minum burung puyuh dalam 1 minggu didapatkan hasil rata-rata 10 Ntu yang artinya masih tergolong jernih.

ABSTRACT

Rahmadhani, Rifqi. 2020. **Designing Software for Monitoring and Control of Drinking Water in Quail Cages Using the Internet of Things based on Android.** Department of Informatics, Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University of Malang. Advisor: (I) Prof. Dr. Suhartono, M.Kom (II) Dr. Fachrul Kurniawan, M.T.

Keywords : Quail Drinking Water, Software, NodeMCU, *Internet of Thing*.

The quail farming business is a promising business. Many Indonesian people are involved in this business. However, the management of quail breeders is not easy, Of course there are many factors that need to be considered in the care of quail breeders. One of them is quail drinking water, quail drinking water has a significant role in the health and productivity of quails. Therefore, breeders need special monitoring of the condition of quail drinking water. monitoring and controlling quail drinking water using an Android-based Internet Of Things sistem is a tool used to monitor and control water conditions in quail drinking water containers. in the software there are several features that can help breeders in monitoring the condition of quail drinking water, including the monitoring feature of turbidity and water level, where farmers can monitor the condition of quail drinking water in real time and periodically hourly, daily, per week with data. sensors residing on the hardware sent by NodeMcu to the database. In this software there is also a control feature for the filling and draining of quail drinking water that breeders can use to fill and drain the quail drinking water reservoir.

المخلص

رحمذاني ورفقي. 2020. تصميم برنامج لرصد مياه الشرب والتحكم فيها في أقباص السمان باستخدام إنترنت الأشياء استنادًا إلى نظام Android. مقال. قسم المعلوماتية ، كلية العلوم والتكنولوجيا ، جامعة الدولة الإسلامية (UIN) مولانا مالك إبراهيم مالانج. المستشار: (I) أ.د. دكتور. سوهارتو نو ، كوم (الثاني) د. فخر كورنياوان ، م.

مياه شرب السمان ، البرمجيات ، NodeMCU ، إنترنت الأشياء.

تعتبر زراعة السمان من الأعمال التجارية الواعدة. يشارك العديد من الإندونيسيين في هذا العمل. ومع ذلك ، فإن إدارة مربي السمان ليست سهلة ، بالطبع هناك العديد من العوامل التي يجب مراعاتها في رعاية مربي السمان. إحداها مياه شرب السمان ، فمياه شرب السمان لها دور هام في صحة وإنتاجية السمان لذلك ، يحتاج المربون إلى مراقبة خاصة لحالة مياه شرب السمان. مراقبة مياه شرب السمان والتحكم فيها باستخدام نظام إنترنت الأشياء القائم على نظام Android هو أداة تستخدم لمراقبة ظروف المياه والتحكم فيها في حاويات مياه شرب السمان. يوجد في البرنامج العديد من الميزات التي يمكن أن تساعد المربين في مراقبة حالة مياه شرب السمان ، بما في ذلك ميزة مراقبة التعكر ومستوى المياه ، حيث يمكن للمزارعين مراقبة حالة مياه شرب السمان في الوقت الحقيقي وبشكل دوري كل ساعة ، يوميًا ، أسبوعيًا مع البيانات. أجهزة الاستشعار الموجودة على الأجهزة التي أرسلتها NodeMcu إلى قاعدة البيانات. في هذا البرنامج ، توجد أيضًا ميزة تحكم لملء وتصريف مياه شرب السمان التي يمكن للمربين استخدامها لملء وتصريف خزان مياه شرب السمان.

BAB I PENDAHULUAN

1.1 Latar Belakang

Peternakan merupakan bagian dari pembangunan nasional yang bertujuan untuk menyediakan pangan hewani berupa daging, susu serta telur yang bernilai gizi tinggi, meningkatkan pendapatan peternak serta menambah devisa dan memperluas kesempatan kerja. Pada masa yang akan datang diharapkan pembangunan perekonomian bangsa (Sudrajat, 2018). Binatang ternak dalam Al-Qur`an banyak sekali dijelaskan sebagai binatang yang mempunyai banyak manfaat. Contoh saja dalam Al-Qur`an surat Al-Mu`minun ayat 21-22 sebagai berikut :

وَالْأَنْعَامَ خَلَقَهَا لَكُمْ فِيهَا دِفْءٌ وَمَنْفَعٌ وَمِمَّا تَأْكُلُونَ

Artinya : Dan Dia telah menciptakan binatang ternak untuk kamu; padanya ada (bulu) yang menghangatkan dan berbagai-bagai manfaat, dan sebahagiannya kamu makan. (An-Nahl 16:5).

Berdasarkan ayat diatas dapat diambil kesimpulan bahwasanya Allah menciptakan binatang ternak untuk manusia dan diambil segala manfaat yang ada padanya untuk mencukupi kebutuhan manusia. Bulu dan kulit dapat dimanfaatkan sebagai pakaian ataupun daging dan telur untuk dikonsumsi sebagai makanan.

Usaha peternakan burung puyuh merupakan usaha yang banyak diminati, karena usaha ternak puyuh dapat menyediakan bahan pangan sebagai sumber protein hewani yang dihasilkan oleh telur. peternakan burung puyuh merupakan salah satu sektor peternakan yang paling efisien dalam menyediakan daging dan telur serta merupakan bahan makanan sumber hewani yang bergizi tinggi. Telur puyuh yang dihasilkan melalui peternakan burung puyuh, menambah variasi pilihan

bagi masyarakat dalam pemenuhan bahan pangan sumber protein hewani selain telur ayam ras dan ayam kampung yang umumnya dikenal masyarakat (Wangefela, Santa, Salendu, & Kalangi, 2018).

Namun dalam pengelolaan peternakan burung puyuh tidak mudah, banyak faktor yang perlu diperhatikan seperti kandang, pakan ternak ataupun air minum ternak. Untuk pemberian air minum pada burung puyuh merupakan hal yang penting dalam perawatan burung puyuh, karena air minum sangat mempengaruhi kondisi pada burung puyuh. Pemberian air minum harus dilakukan setiap saat dengan cara mengalirkan air dari penampungan air ke tempat minum puyuh yang ada pada setiap kandang puyuh melalui pipa air. Pengecekan kondisi air harus dilakukan setiap saat, karena jika air dalam kondisi kotor dapat mempengaruhi kesehatan dan produktifitas pada puyuh dan juga dapat merugikan peternak itu sendiri.

Penampungan air pada kandang puyuh mempunyai fungsi yang sangat banyak terhadap puyuh selain sebagai tempat penyaluran air, namun juga berfungsi sebagai tempat mencampurnya vitamin pada unggas agar puyuh tidak mudah sakit. Ketersediaan dan kebersihan air pada penampungan ini sangat penting karena jika air dalam kondisi kosong ataupun kotor puyuh tidak mendapat minum atau kehausan ini berdampak pada produktifitas telur nantinya. Pengontrolan air minum menjadi penting karna harus diisi setiap kali air pada penampungan habis dan mengguras air jika air dalam kondisi kotor. Hal ini kemudian menimbulkan masalah yaitu, peternak harus melakukan pengontrolan pada penampungan air setiap saat.

Dengan penerapan bidang teknologi merupakan salah satu solusi untuk mempermudah pengontrolan penampungan air agar dapat dikelola secara efisien dan peternak dapat mendapatkan informasi tentang kondisi pada penampungan air minum. Salah satu teknologi yang dapat digunakan untuk mendapatkan informasi kondisi pada penampungan air minum puyuh yaitu dengan memanfaatkan teknologi *Internet of Things*. *Internet of Things* (IoT) merupakan kumpulan benda benda (*things*), berupa perangkat fisik (*hardware*) yang mampu bertukar informasi antara sumber informasi, operator layanan ataupun perangkat lainnya yang terhubung kedalam sistem sehingga dapat memberikan kemanfaatan yang lebih besar (M. I. Mahali, 2016). Sistem kontroling suhu ataupun penampungan air ini akan di implementasikan dengan IoT menggunakan mikrokontroler dan sensor-sensor yang akan digunakan untuk mendapatkan data tentang kondisi pada penampungan air. Kemudian data-data tersebut akan dikirim menggunakan *mikrokontroler* Node MCU melalui *Wifi* yang akan memberikan informasi kepada peternak melalui perangkat *Mobile*. Informasi-informasi tersebut berupa jarak kedalaman air dan kekeruhan air, peternak dapat memantau kondisi air minum burung puyuh secara *realtime* maupun berkala setiap jam, hari dan minggu dengan data-data yang telah dikirimkan oleh perangkat keras dan ditampilkan dalam perangkat lunak berupa data grafik yang mudah dipahami oleh peternak. Dengan informasi tersebut diharapkan peternak dapat dengan mudah memonitoring dan mengontrol kondisi pada penampungan air minum puyuh melalui perangkat *Mobile*.

1.2 Pernyataan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan sebelumnya, pertanyaan penelitian yang akan diangkat adalah bagaimana membangun sistem

perangkat lunak yang dapat membantu peternak dalam mengontrol dan memonitoring kondisi penampungan air minum burung puyuh berdasarkan status level dan kekeruhan air berbasis *Mobile Android*?

1.3 Tujuan Penelitian

Dalam penelitian yang dilakukan, adapun tujuan yang ingin dicapai dari penelitian ini adalah untuk mengetahui cara membangun *software* sebagai monitoring dan kontrol air minum burung puyuh berdasarkan status level dan kekeruhan air menggunakan sistem *Internet Of Things* berbasis *mobile android*.

1.4 Batasan Masalah

Untuk menghindari pembahasan yang keluar dari materi, maka ditentukan batasan masalah. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Data yang didapatkan adalah level dan kekeruhan air yang berada pada penampungan air minum burung puyuh.
2. Sistem komunikasi menggunakan konsep *Internet of Things*

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut :

1. Mengetahui bagaimana cara membuat sistem monitoring dan kontrol air pada penampungan air minum burung puyuh menggunakan *software Mobile* berbasis IoT.
2. Menghasilkan sistem kontrol dan monitoring air pada penampungan air minum burung puyuh yang nantinya dapat mempermudah peternak dalam memantau kondisi pada kandang burung puyuh melalui perangkat *Mobile*.

BAB II STUDI PUSTAKA

Pada bab kajian pustaka akan dibahas mengenai penelitian terkait dan landasan teori yang digunakan oleh peneliti sebagai acuan melakukan penelitian.

2.1 Penelitian Terkait

Pada bab ini dibahas penelitian-penelitian yang sudah dilakukan sebagai perbandingan dan acuan untuk penelitian ini. Berikut merupakan pemaparan dari beberapa peneliti yang pernah melakukan penelitian yang berkaitan dengan penelitian ini.

Zulfan Mindriawan, dkk, melakukan penelitian dengan judul Implementasi *Internet of Things* Pada Sistem Monitoring Suhu dan Kontrol Air Pada Kandang Burung Puyuh Petelur dengan Menggunakan Protokol MQTT, dalam penelitian ini penulis ingin membuat sebuah alat untuk mengontrol kondisi penampungan air dengan memanfaatkan *Wireless sensor Network* berbasis *Internet of Thing* menggunakan protokol MQTT (*Message Queuing Telemetry Transport*). Sensor yang d hubungkan pada penampungan air adalah sensor ketinggian air, sensor jarak, sensor suhu yang kemudian data-data dari sensorsensor tersebut dikirimkan ke *Broker* melalui *Wi-Fi*, lalu pada keran penampungan air akan digunakan dua servo yang dihubungkan dengan internet sehingga pengontrolan air dapat dilakukan secara online dan peternak juga mendapat notifikasi waktu air pada penampungan penuh atau habis (Mindriawan, Arimbawa, & Wijaya, 2018).

Achmad Fauzi juga telah melakukan penelitian yang berhubungan dengan Iot dengan judul Sistem Kontrol Suhu Ruangan pada *Inkubator* Anak Ayam Menggunakan *ESP Wemos* Di Berbasis IoT. Dalam penelitian ini peneliti

menggunakan mikrokontroler *ESP Wemos D1*, Sensor suhu (*DHT11*). Pengontrolan suhu hangat dilakukan dengan menggunakan *dimmer* untuk mengatur panas pada 2 buah lampu dan pengontrolan suhu dingin menggunakan kipas *blower* 1 buah yang terhubung ke *relay* sebagai saklar otomatis semua *output* alat dan kegiatan alat dapat di lihat melalui *smarthphone* Android atau iPhone yang sedang digunakan (Fauzi, 2017).

Ari,dkk, melakukan penelitian tentang monitoring peternakan ayam boiler berbasis IoT, penelitian ini, menjelaskan sistem alat pendeteksi suhu, kelembaban dan kadar gas berbahaya pada kandang ayam. Sistem ini bekerja dengan mengirimkan informasi kondisi kualitas kadar gas amonia, gas metana, suhu dan kelembaban pada kandang ayam. Sistem ini memberikan hasil dari monitoring data gas, suhu dan kelembaban yang tidak sesuai kepada pekerja di peternakan ayam broiler. Pada sistem ini dapat dimonitoring secara *Platform*. Secara umum sistem ini terdiri dari sensor *DHT11* dan sensor *MQ135*. Sensor *DHT11* untuk mendeteksi suhu dan kelembaban pada kandang ayam dan sensor *MQ135* untuk mendeteksi kadar gas amonia pada kandang ayam. Sistem ini menggunakan modul *ESP8266* dan mikrokontroler *Arduino Mega 2560*. Dari cara kerjanya untuk mengakses data sensor ke web *Firebase* yang kemudian ditampilkan dalam bentuk grafik sebagai sarana untuk memonitoring kandang ayam. Terdapat sistem peringatan apabila suhu tidak sesuai dengan suhu ideal yang sudah ditentukan. Data dapat diakses secara *real time* menggunakan aplikasi *smartphone*. (Masriwilaga, Jabar, Subagja, & Septiana, 2019).

2.2 Landasan Teori

Teori umum yang digunakan meliputi teori pengembangan perangkat lunak kontroling minum dan beberapa teori lainnya yang mendasari penyusunan tugas akhir ini.

2.2.1 Perangkat Lunak

Menurut Ian Sommerville perangkat lunak tidak hanya mencakup program komputer saja, akan tetapi juga termasuk semua dokumentasi dan konfigurasi data yang berhubungan yang diperlukan untuk membuat program beroperasi dengan benar (Setiyani, 2018).

Sedangkan menurut Pressman mengemukakan bahwa perangkat lunak adalah instruksi – instruksi dalam hal ini adalah program komputer yang ketika dijalankan menyediakan fitur – fitur , fungsi – fungsi dan kinerja – kinerja yang dikehendaki pengguna, dalam perangkat lunak terdapat struktur data yang memungkinkan program -program dapat memanipulasi informasi, dan informasi tersebut tercetak dalam bentuk maya yang menggambarkan pengoperasian dan penggunaan program - program tersebut.

2.2.1.1. Fungsi Perangkat Lunak (*Software*)

fungsi perangkat lunak tentunya menyediakan manfaat langsung pada para pengguna akhir dan juga menyediakan dukungan internal untuk fitur - fitur yang tampak oleh pengguna (Setiyani, 2018).

2.2.2. Android

Definisi Android menurut Nazrudin Safaat H (2011) adalah sebuah sistem operasi untuk perangkat *Mobile* berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang

berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri.

Keuntungan dari Android adalah memberikan kemudahan pada pengembangan Android. *Developer* hanya perlu mengembangkan aplikasi pada *platform Android*, dan aplikasi tersebut akan dapat berjalan pada berbagai perangkat, yang tentunya mempunyai sistem operasi Android. Selain itu Android merupakan *platform* yang lengkap, baik dari sistem operasi, aplikasi serta *tools* untuk mengembangkan aplikasi.

2.2.2.1 Arsitektur Android

Menurut (Lee, 2012) Android terbagi menjadi lima bagian dari empat lapisan utama yaitu :

1. *Linux Kernel*

Linux kernel merupakan dasar dari Android. Lapisan ini semua *low level device driver* untuk berbagai komponen perangkat keras dari perangkat Android.

2. *Libraries*

Lapisan ini mengandung seluruh kode yang menyediakan fitur utama dari sistem operasi Android. Sebagai contoh, *SQLite library* menyediakan dukungan *Database* sehingga aplikasi dapat menggunakannya sebagai tempat penyimpanan.

3. *Android Runtime*

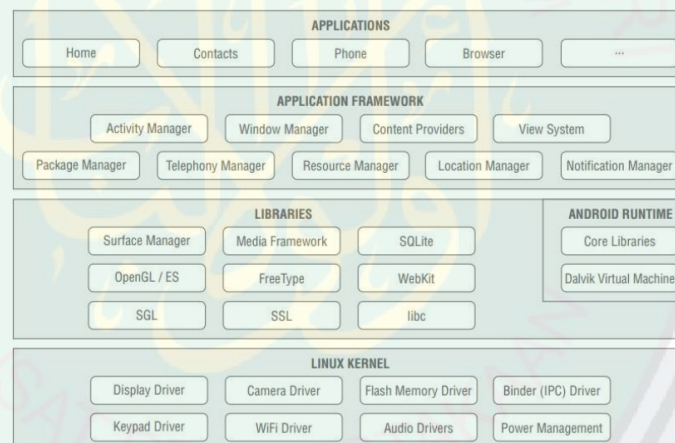
Android Runtime terdapat pada lapisan yang sama dengan *libraries*, yang menyediakan kumpulan *libraries* utama yang dapat digunakan pengembang untuk membuat aplikasi pada Android menggunakan bahasa pemrograman *Java*. *Android Runtime* juga termasuk *Dalvik virtual machine*, yang memungkinkan aplikasi Android berjalan dengan proses tersendiri.

4. Application Framework

Memperlihatkan berbagai kemampuan dari sistem operasi Android untuk pengembang aplikasi Android sehingga mereka dapat menggunakannya dalam membuat aplikasi.

5. Applications

Applications merupakan lapisan paling atas, yang memungkinkan para pengembang untuk menemukan aplikasi bawaan yang terdapat pada perangkat Android. Contohnya saja adalah *Phone*, *Contacts*, *Browser* dan lainnya. Selain itu juga aplikasi yang dapat di unduh dan diinstall dari *Android Market*. Setiap aplikasi yang dibuat terdapat pada lapisan ini. Adapun gambaran arsitektur Android seperti dibawah ini



Gambar 2.1 Arsitektur Android

2.2.3. Firebase

Firebase adalah penyedia layanan realtime *Database* dan backend sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda – beda dan disimpan pada cloud-nya *Firebase*. *Firebase* memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, Ios, *Javacript*, *Java*, *Objective-C*

dan Node.JS . *Database Firebase* juga bersifat bisa diakses lewat *REST API*. *REST API* tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima push notification dari server. Pengembang menggunakan REST API untuk post data yang selanjutnya *Firebase client library* yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data secara realtime. Gambaran *sistem kerja Firebase* seperti pada gambar 2.2 dibawah ini :



Gambar 2.2. Sistem Kerja *Firebase*

Pengembang juga dapat menggunakan *Database* ini untuk mengamankan data menggunakan server *Firebase* dengan rules yang ada. Untuk hosting file *Firebase* menyediakan hosting untuk static file dengan fasilitas CDN dan SNL(Khanifa, 2016).

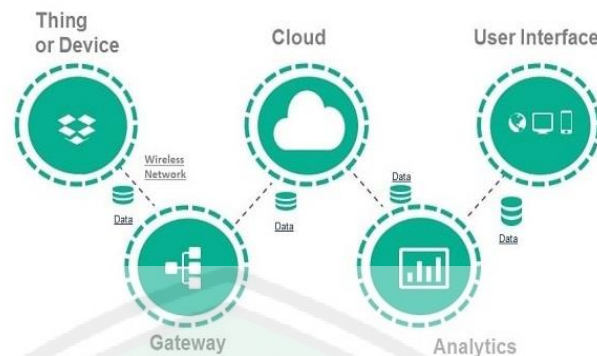
2.2.4 Internet Of Things

Internet of Things (IoT) adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer (Hafsa, T., Ayesha, K., & Junaid, M., 2016.)

IoT sudah berkembang pesat mulai dari penggabungan teknologi *nirkabel*, *MicroElectromechanical Systems* (MEMS) dan juga Internet. IoT menggunakan beberapa teknologi yang secara garis besar digabungkan menjadi satu kesatuan diantaranya sensor sebagai pembaca data, koneksi internet dengan beberapa macam topologi jaringan, *Radio Frequency Identification* (RFID), *wireless sensor network* dan teknologi yang terus akan bertambah sesuai dengan kebutuhan (C. Wang et al., 2013). IoT juga bisa mencakup teknologi-teknologi sensor lainnya, seperti teknologi nirkabel maupun kode QR yang sering kita temukan di sekitar kita, contoh penerapannya dalam benda yang ada di dunia nyata adalah untuk pengolahan bahan pangan, elektronik, dan berbagai mesin atau teknologi lainnya yang semuanya tersambung ke jaringan lokal maupun global lewat sensor yang tertanam dan selalu menyala aktif. IoT ini mengacu pada mesin atau alat yang bisa diidentifikasi sebagai representasi virtual dalam strukturnya yang berbasis Internet.

2.2.4.1 Cara Kerja IoT

Cara Kerja *Internet of Things* yaitu dengan memanfaatkan sebuah instruksi pemrograman yang dimana tiap-tiap perintah argumennya itu dapat menghasilkan sebuah interaksi antara sesama perangkat yang saling terhubung satu sama lainnya secara otomatis tanpa campur tangan manusia. Bahkan dalam jarak yang jauh sekalipun. Internet dapat menjadi penghubung diantara kedua interaksi perangkat tersebut. Sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung. Adapaun gambaran cara kerja Iot seperti pada gambar dibawah ini :



Gambar 2.3. Cara Kerja IoT

2.2.5 Mikrokontroler Node Mcu Esp8266

NodeMCU ESP8266 adalah chip terintegrasi yang dirancang untuk menghubungkan mikrokontroler dengan internet melalui *Wi-Fi*. Ia menawarkan solusi jaringan *Wi-Fi* yang lengkap dan mandiri, yang memungkinkan untuk menjadi host ataupun sebagai *Wi-Fi client*. ESP8266 memiliki kemampuan pengolahan dan penyimpanan on-board yang kuat, yang memungkinkannya untuk diintegrasikan dengan sensor dan aplikasi perangkat khusus lain melalui GPIO dengan pengembangan yang mudah serta waktu *loading* yang minimal. Tingkat integrasinya yang tinggi memungkinkan untuk meminimalkan kebutuhan sirkuit eksternal, termasuk modul *front-end*, dirancang untuk mengisi daerah PCB yang minimal (Marvin, 2017). *NodeMCU* menggunakan bahasa pemrograman Lua yang merupakan package dari esp8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan c hanya berbeda *syntax*. Jika menggunakan bahasa Lua maka dapat menggunakan *tool Lua loader* maupun *Lua uploader*. Selain dengan bahasa Lua *NodeMCU* juga *support* dengan *software Arduino IDE* dengan melakukan sedikit perubahan *board manager* pada *Arduino IDE*.



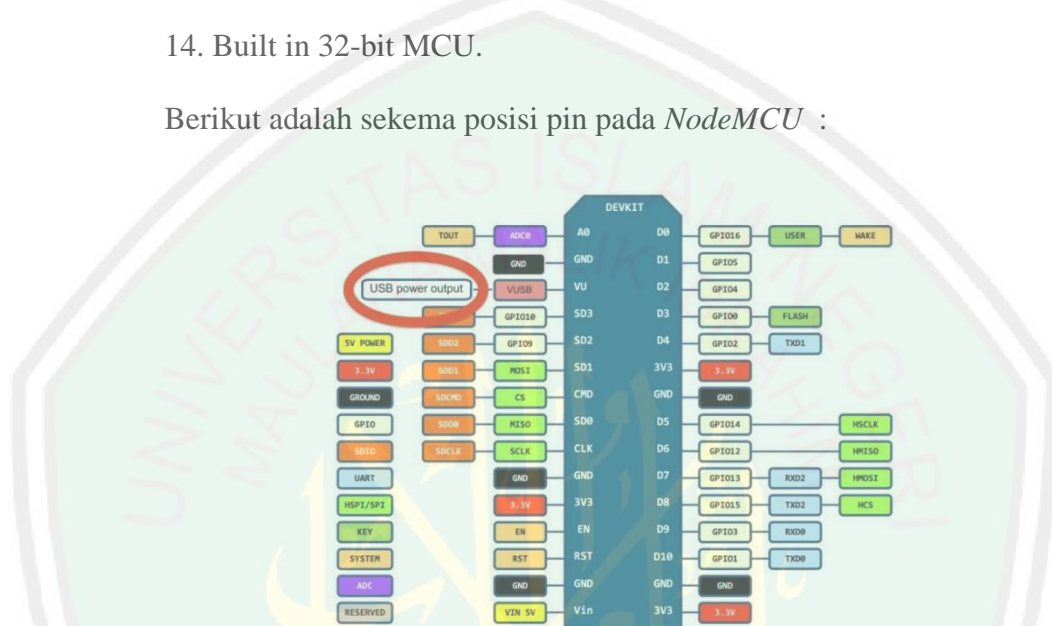
Gambar 2.4. *NodeMCU* Esp8266

Spesifikasi yang dimiliki oleh *NodeMCU* seperti pada gambar di atas sebagai berikut :

1. *Board* ini berbasis ESP8266 serial WiFi SoC (*Single on Chip*) dengan onboard USB to TTL. Wireless yang digunakan adalah IEEE 802.11b/g/n.
2. 2 *tantalum capacitor* 100 *micro farad* dan 10 *micro farad*.
3. 3.3v LDO regulator.
4. *Blue led* sebagai indikator.
5. Cp2102 usb to UART *bridge*.
6. Tombol reset, port usb, dan tombol flash.
7. Terdapat 9 GPIO yang di dalamnya ada 3 pin PWM, 1 x ADC Channel, dan pin RX TX
8. 3 pin ground.
9. S3 dan S2 sebagai pin GPIO
10. S1 MOSI (Master Output Slave Input) yaitu jalur data dari master dan masuk ke dalam slave, sc cmd/sc.

11. S0 MISO (Master Input Slave Input) yaitu jalur data keluar dari slave dan masuk ke dalam master.
12. SK yang merupakan SCLK dari master ke slave yang berfungsi sebagai clock.
13. Pin Vin sebagai masukan tegangan.
14. Built in 32-bit MCU.

Berikut adalah sekema posisi pin pada *NodeMCU* :



Gambar 2.5. Skematik posisi Pin *NodeMCU* Dev Kit v3

2.2.6 Model Proses Pembangunan Perangkat Lunak (*Waterfall*)

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian (Tristanto, 2018).

Metode *Waterfall* memiliki tahapan-tahapan sebagai berikut (Sommerville, 2011):

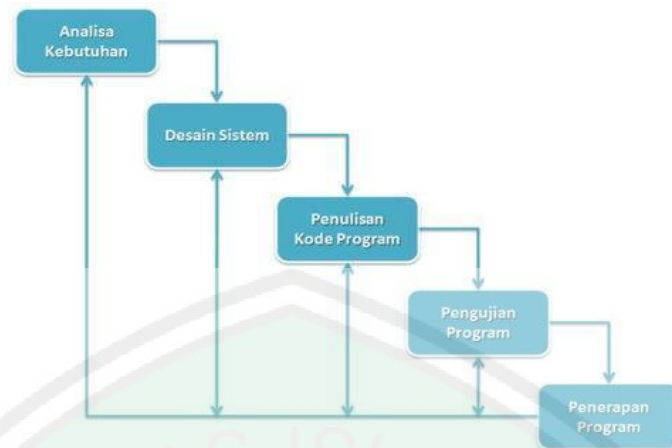
- 1) *Requirements analysis and definition* Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2) *Sistem and software design* Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3) *Implementation and unit testing* Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4) *Integration and sistem testing* Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

5) *Operation and maintenance* Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. Maintenance melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru. Model *waterfall* bila digambarkan dalam bentuk diagram akan seperti pada gambar dibawah ini :



Gambar 2.6 *Waterfall*

Adapun Kelebihan dan kelemahan dari metode *Waterfall* menurut (Prayudita,2018) :

2.2.6.1 Kelebihan Metode *Waterfall*

1. Dari sisi *user* juga lebih menguntungkan , karena dapat merencanakan dan menyiapkan kebutuhan data dan proses yang diperlukan sejak awal.
2. Penjadwalan juga menjadi lebih menentu , karena jadwal setiap proses dapat ditentukan secara pasti . sehingga dapat dilihat jelas target penyelesaian pengembangan program.
3. Dengan adanya urutan yang pasti , dapat dilihat pula perkembangan untuk setiap tahap secara pasti.
4. Model ini merupakan jenis model yang bersifat dokumen lengkap sehingga proses pemeliharaan dapat dilakukan dengan mudah. Prosesnya lebih terstruktur , hal ini membuat kualitas *software* baik dan tetap terjaga.

2.2.6.2 Kekurangan Metode *Waterfall*

1. Membutuhkan daftar kebutuhan yang lengkap sejak awal . Untuk menghindari pengulangan tahap dari awal , *user* harus memberikan seluruh prosedur , data dan laporan yang diinginkan mulai dari tahap awal pengembangan .
2. *User* tidak dapat mencoba sistem sebelum benar - benar selesai.
3. Kinerja personil menjadi kurang optimal karena terdapat proses menunggu suatu tahap selesai terlebih dahulu . oleh karena itu , seringkali diperlukan personil yang "*multi-skilled*" sehingga minimal dapat membantu pengerjaan untuk tahapan berikutnya.
4. Bersifat Kaku , sehingga sulit melakukan perubahan di tengah proses. Jika terdapat kekurangan proses atau prosedur dari tahap sebelumnya , maka tahapan pengembangan harus dimulai dari awal lagi . Hal tersebut akan memakan waktu yang lebih lama. Jika proses sebelumnya belum selesai sampai akhir maka proses selanjutnya juga tidak dapat berjalan. Oleh karena itu , jika terdapat kekurangan dalam permintaan *user* maka proses pengembangan harus dimulai dari awal lagi.

BAB III METODE PENELITIAN

3.1 Desain Sistem

Desain sistem perancangan sistem monitoring dan kontroling air minum burung puyuh, dengan Arduino dan NodeMcu sebagai mikrokontroler dan juga sensor sebagai input data berikut merupakan tahapan perakitan perangkat lunak.

3.1.1 Perangkat Lunak Kontrol dan Monitor Air Minum Puyuh

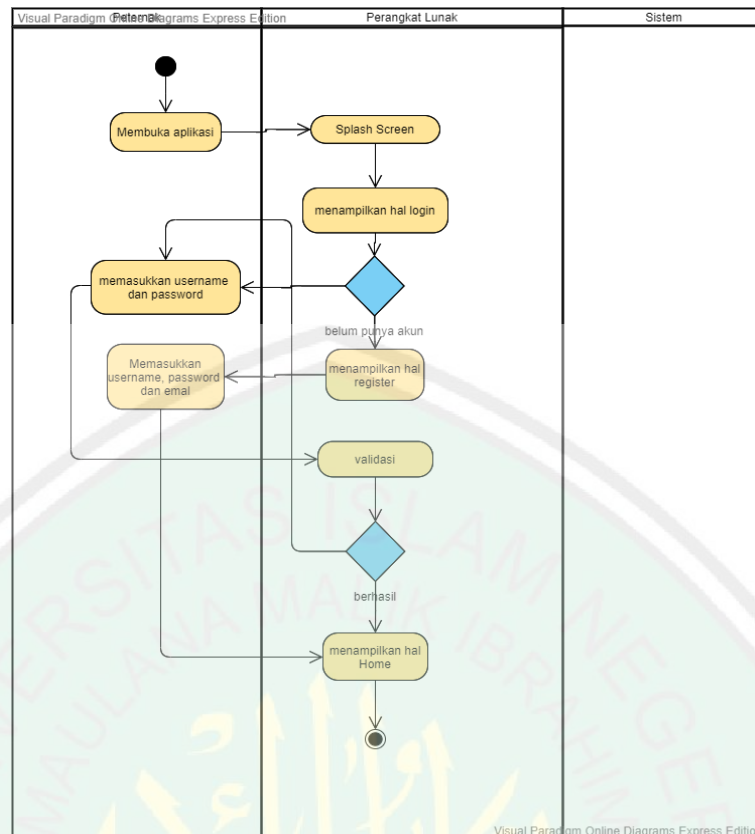
Perangkat lunak yang digunakan berbasis *android*, dimana *android* adalah operasi sistem yang banyak digunakan dalam perangkat *mobile* yang hampir setiap hari digunakan. Hal ini dapat mempermudah *user* dalam mengakses perangkat lunak kapanpun. Agar dapat mempermudah dalam memonitoring dan mengontrol kondisi air minum burung puyuh. Dalam perancangan *android* tersebut melalui beberapa tahapan antara lain:

3.1.1.1 Pembuatan Model Bisnis

Pembuatan Model bisnis merupakan tahapan untuk menentukan ruang lingkup perangkat lunak yang akan dibangun. Dari ruang lingkup itu sendiri nantinya akan didapatkan sebagai referensi dalam pembagunan perangkat lunak.

1. *Activity Diagram*

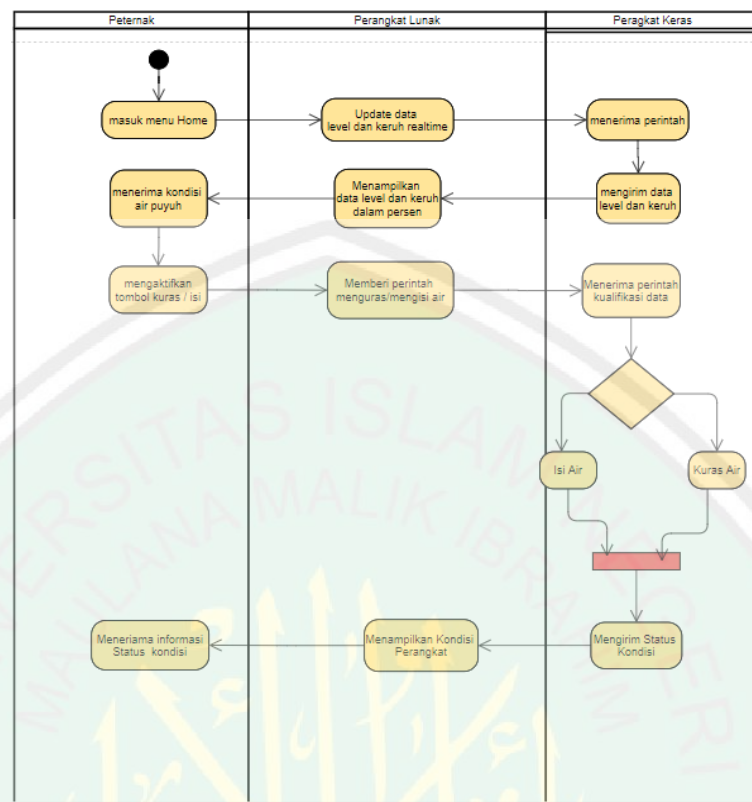
Pembuatan *activity diagram* yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir.. Berikut ini merupakan *activity diagram login* dan *register* akun pada gambar 3.4.



Gambar 3.1 Diagram aktivitas Login dan Register

Dari diagram aktivitas yang tertera di atas menggambarkan alur yang terjadi untuk login dan registrasi akun. Dimana tampilan awal saat aplikasi dibuka maka akan muncul tampilan *Splash Screen* dan dilanjutkan ke halaman login, yang di mana user harus memasukan username dan password yang telah terdaftar. Jikalau belum user diharuskan untuk mendaftar akun dengan memasukan username, password dan email.

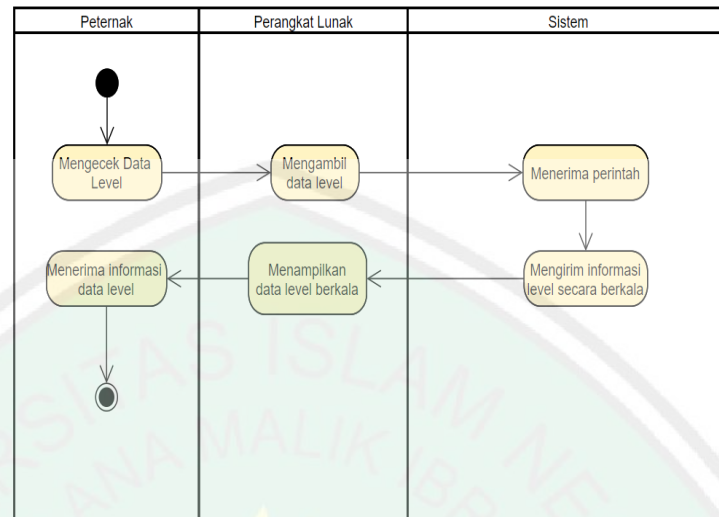
Kemudian berikut adalah *activity diagram* Mengontrol air minum puyuh



Gambar 3.2 Diagram Aktivitas Kontroling Air Minum Puyuh

Dari diagram aktivitas yang tertera di atas telah menggambarkan alur yang terjadi bagaimana perangkat kontroling air minum puyuh bekerja . menguras dan mengisi air yang dihubungkan menggunakan perangkat lunak yang dilakukan secara otomatis, sehingga pompa air yang diganti menggunakan tombol yang terletak pada perangkat lunak dapat dilakukan para peternak seolah dapat mengisi dan menguras air minum puyuh. Dimulainya tahapan dari aktivitas peniraman ini yaitu dari peternak tanaman melakukan pengoperasian pengisian atau pengurasan menggunakan perangkat lunak kemudian dari sistem akan memberi perintah pengoperasian kepada perangkat keras yang dimana pompa pada perangkat keras akan menyala.

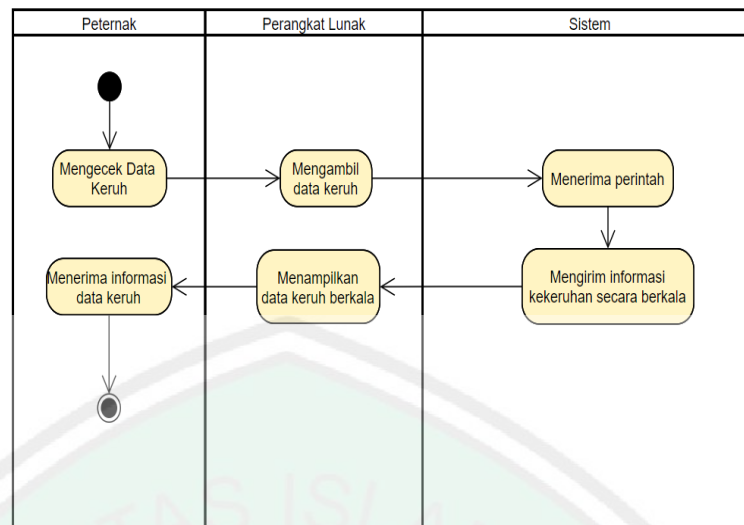
Kemudian berikut ini adalah *activity diagram* monitor level akan dipaparkan pada gambar 3.5.



Gambar 3.3 Aktivitas Diagram Monitor Level Air

Diagram aktivitas pada gambar 3.3 di atas menggambarkan alur yang digunakan sistem untuk melakukan pengecekan terhadap level air yang menggunakan perangkat lunak sebagai kontrol dari perangkat keras, dimana peternak akan mendapatkan informasi secara berkala mengenai level air tanaman melalui sensor yang terdapat pada perangkat keras yang nantinya akan dikirim ke server.

Tidak hanya melakukan pengecekan level air namun juga dilakukan pengecekan kekeruhan yang merupakan salah satu factor yang digunakan untuk melakukan pengisian air minum puyuh secara otomatis. Berikut merupakan activity diagram monitor keruh air minum burung puyuh.



Gambar 3.4 Activity Diagram Monitor Keruh Air

Diagram aktivitas di atas menggambarkan alur bagaimana sistem melakukan pengecekan kekeruhan pada air minum puyuh menggunakan perangkat lunak sebagai remot kendali untuk pengisian air, dimana peternak akan mendapatkan informasi mengenai air minum puyuh dengan keruh, normal atau jernih melalui sensor yang terdapat pada wadah air minum lalu dikirimkan kepada sistem.

2. Use Case Diagram

Use Case diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, diagram use case digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.

Dibawah ini merupakan pemaparan yang dapat dilihat pada gambar 3.5 yang akan menampilkan *Use Case Diagram* dari perangkat lunak yang akan dibangun.



Gambar 3.5 Use Case Diagram

Dari gambar 3.5 diatas mengenai *Use* menjelaskan interaksi antara aktor dengan sistem perangkat keras. Yang dimana perangkat lunak digunakan untuk Memonitoring dan mengontrol air minum puyuh, dalam penggambaran *Use Case* memiliki 4 fungsionalitas yang terdiri dari bagaimana memonitoring air minum puyuh secara *realtime* dan mengetahui data dari kondisi air secara berkala. Dimana hal ini aktornya berperan sebagai peternak yang memiliki tugas dalam memonitoring dan mengontrol air minum puyuh.

3.1.1.2 *Prototype* Perangkat Lunak

Pada tahapan pembagunan perangkat lunak akan memberikan gambaran bagaimana perangkat lunak akan dibangun, mulai dari desain tampilan antar muka dan juga fitur-fitur yang akan dimuat didalam perangkat lunak tersebut, sehingga dalam proses pembuatan peragkat lunak dapat tergambar secara jelas. Berikut merupakan gambaran dari pembagunan peragkat lunak

1. Perancangan *Prototype Splash*

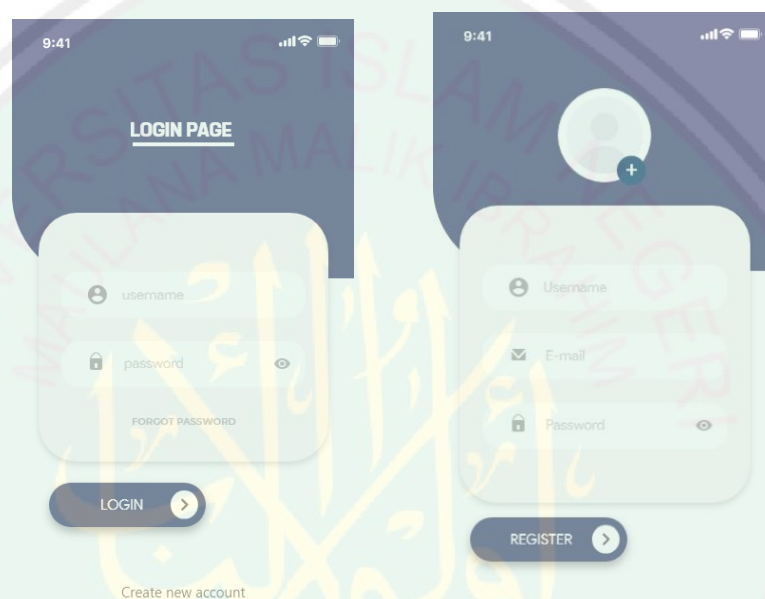
Antar muka halaman *Splash screen* adalah halaman yang pertama kali akan muncul Ketika masuk dalam aplikasi, di dalam tampilan *splash screen* terdapat nama aplikasi dan *Button start* untuk memulai masuk kedalam halaman selanjutnya. Berikut adalah tampilan *splash screen* perangkat lunak monitoring dan kontrol air minum burung puyuh :



Gambar 3.6 Tampilan Splash Screen

2. Perancangan *login* dan *register*

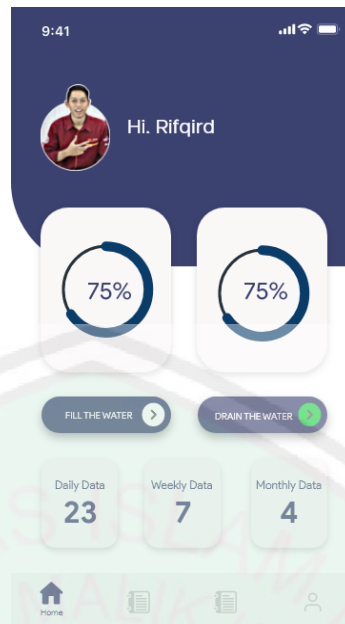
perancangan antarmuka untuk halaman *Login* dan *Register*, Pada antarmuka *Login* terdapat *form* untuk mengisi *username* dan *password*,s sebagai verifikasi akun yang sudah terdaftar dalam aplikasi. jika user belum punya akun maka harus melakukan registrasi aku didalam halaman register, dengan memasukkan *username*, *email* dan *password*. Berikut adalah tampilan *login* dan *register*



Gambar 3.7 Tampilan *Login* dan *Register*

3. Perancangan Halaman *Dashborad*

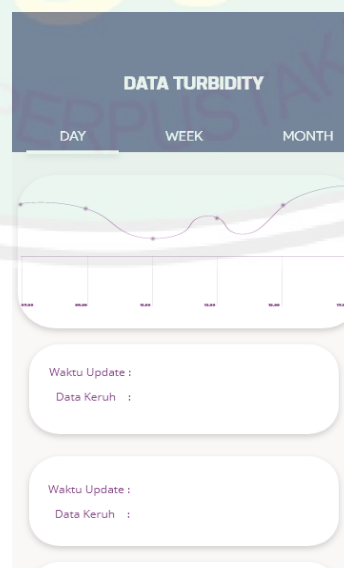
Pada halaman dashboard terdapat informasi level air dan kekeruhan air secara realtime yang dikonverikan dalam bentuk persen. Kemudian juga ada jumlah data level dan kekeruhan air yang masuk berdasarkan jam, hari dan minggu. Dan juga terdapat button mengisi dan menguras air yang dapat menyalakan pompa secara otomatis maupun manual dari perangkat lunak sesuai yang dibutuhkan oleh peternak. Berikut adalah tampilan halaman *dashboard*.



Gambar 3.8 Tampilan *Dashboard*

4. Perancangan Halaman Data Kekeruhan

Pada gambar 3.8 dibawah ini menunjukkan tampilan dari data kekeruhan yang diambil dari penyimpanan database *Firebase* secara berkala. Data yang ditampilkan berpura grafik data keruh dan waktu update data. Pada aplikasi ini tersedia tiga pilihan periode yaitu periode harian, mingguan, dan bulanan. Grafik yang digunakan dibuat dengan *library MPAndroidChart*. *MPAndroidChart* merupakan

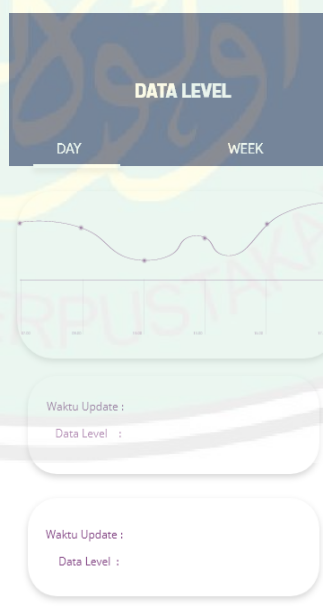


Gambar 3.8 Tampilan Halaman Data Kekeruhan

sebuah tampilan bagan pada android / *library* tampilan grafik yang sangat lengkap. MPAndroidChart mendukung grafik bentuk garis, batang, lingkaran, radar, gelembung, kandil maupun skala, seret dan animasi. Berikut adalah tampilan halaman monitoring data kekeruhan.

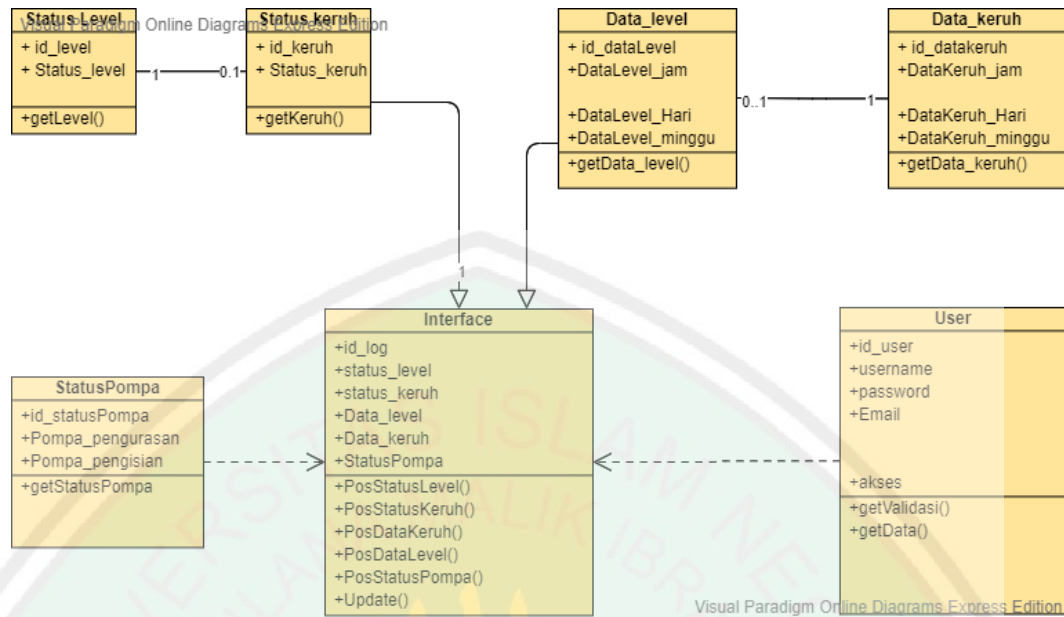
5. Perancangan Halaman Data Level

Halaman ini menampilkan grafik dan data sensor *Ultrasonic* berdasarkan perjam dan hari. Data *hourly* diperoleh dari data sensor pada hardware yang jika level air menurun dan pada tingkatan terendah dari kriteria yang diberikan maka akan dilakukan pengisian secara otomatis. Dan data *daily* diperoleh dari rata-rata data level perharinya. Halaman ini hanya menggunakan 2 kondisi pemantauan, kondisi perjam dan hari berbeda dengan kekeruhan dikarenakan tingkat perubahan data yang diperoleh level lebih signifikan dibanding dengan kekeruhan. Berikut adalah tampilan halaman monitoring data level.



Gambar 3.9 Tampilan Halaman Data Level

3.1.1.3 Perancangan data



Gambar 3.10 Struktur kelas model

Dari beberapa penggambaran dari relasi antar kelas model pada gambar 3.10 maka menentukan data yang akan di buat didalam database *Firestore* dari perangkat lunak monitoring dan kontroling air minum burung puyuh tergambar pada tabel 3.1 berikut ini:

Tabel 3.1 Perancangan data

No	Nama Kelas	Nama Child	Nama Child	Type Data
1	Control_air	Control_keruh	-	Bollean
		Control_level	-	Bollean
2	Data_realtime	Status_keruh	-	int(3)
		Status_level	-	Int(3)
3	Data_keruh	Data_jam	Id_datajamL	Int(15)
			Waktu_update	Time
			Data_sensor	Int(5)
		Data_hari	Id_dataahriL	Int(15)
			Waktu_update	int(11)
			Data_sensor	Time

		Data_minggu	Id_dataMingguL	Int(15)
			Waktu_update	int(11)
			Data_sensor	Time
4.	Data_Level	Data_jam	Id_datajamK	Int(15)
			Waktu_update	Time
			Data_sensor	Int(5)
		Data_hari	Id_dataahriK	Int(15)
			Waktu_update	int(11)
			Data_sensor	Time
		Data_minggu	Id_data-MingguK	Int(15)
			Waktu_update	int(11)
			Data_sensor	Time

3.2 Pengujian Sistem

Pengujian Sistem dilakukan untuk memeriksa keselarasan antar komponen sistem yang akan diimplementasikan. Tujuan utama dari pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. proses pengujian akan melakukan pengecekan terhadap keakuratan dari pemantauan perangkat lunak yang berbasis *android* terhadap air minum burung puyuh yang dimana menggunakan Arduino uno sebagai perangkat keras pendukung dalam pemantau tersebut. Dimana pengujian sistem ini akan menggunakan konsep *black box* yang mengacu terhadap tahapan perancangan yang telah dirancang.

Pada pengujian ini berfokus pada persyaratan fungsional dari perangkat lunak. yang bertujuan untuk mencari kesalahan terhadap fungsi-fungsi yang akan dijalankan, baik kesalahan dalam tampilan maupun sistem.

Berikut merupakan fungsi-fungsi yang diujikan pada pengujian fungsional pada perangkat lunak monitoring dan kontroling air minum burung puyuh:

Tabel 3.2 Pengujian Fungsional Perangkat Lunak

Fungsi yang diuji	Test Case
Login	Memasukan username dan password yang sudah terdaftar
Register	Input Username, password email
Update Data Sensor	Memasukan data terbaru dengan <i>refresh</i> sistem dari server mikrokontroler Arduino
Status pompa pada wadah air minum	Merubah kondisi pada menu Home
Status Kondisi air realtime dalam bentuk persen	Memasukkan data sensor ke database realtime
Update jumlah data masuk	Memasukkan data dan menghitung data masuk
Update Data level air berkala dalam dalam bentuk grafik	Memasukan data terbaru dengan <i>refresh</i> sistem dari server mikrokontroler
Update Data keruh air berkala dalam dalam bentuk grafik	Memasukan data terbaru dengan <i>refresh</i> sistem dari server mikrokontroler

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi Program

Pada bab ini peneliti melakukan analisis proses bisnis untuk menemukan alur kerja sistem, untuk mempermudah peneliti dalam menemukan jalannya kerja sistem.

Berikut merupakan tahapan yang dibutuhkan dalam proses pembangunan perangkat lunak yang telah dibagi dalam beberapa sub implementasi, yaitu sebagai berikut :

4.1.1 Kebutuhan implementasi

Tabel 4.1 Spesifikasi Software dan Hardware

KEBUTUHAN	KETERANGAN
<i>Sistem Operasi</i>	Windows 10
	<i>Android / IOS (Smartphone)</i>
<i>Processor</i>	<i>Dual Core</i> atau di atasnya
<i>RAM</i>	512 MB atau di atasnya
<i>Browser</i>	<i>Mozilla Firefox, Google Chrome, Opera</i>
<i>Software</i>	Bahasa Server side : JSON Bahasa Style Sheet : XML Bahasa Script : JAVA
<i>Data Server</i>	<i>Firebase</i>

4.1.1 Struktur Pemograman dan kode program

Pogram perangkat lunak untuk sistem kontrol dan monitoring air minum burung puyuh secara otomatis dirancang dan dibangun menggunakan bahasa pemrograman JAVA native, serta database menggunakan *Firestore*. Berikut merupakan file JAVA utama untuk menunjukkan data hasil dari pembangunan perangkat lunak pada sistem monitoring dan kontroling air minum burung puyuh yang didasarkan pada level dan kekeruhan air yang berbasis *mobile android*.

1. Login

```

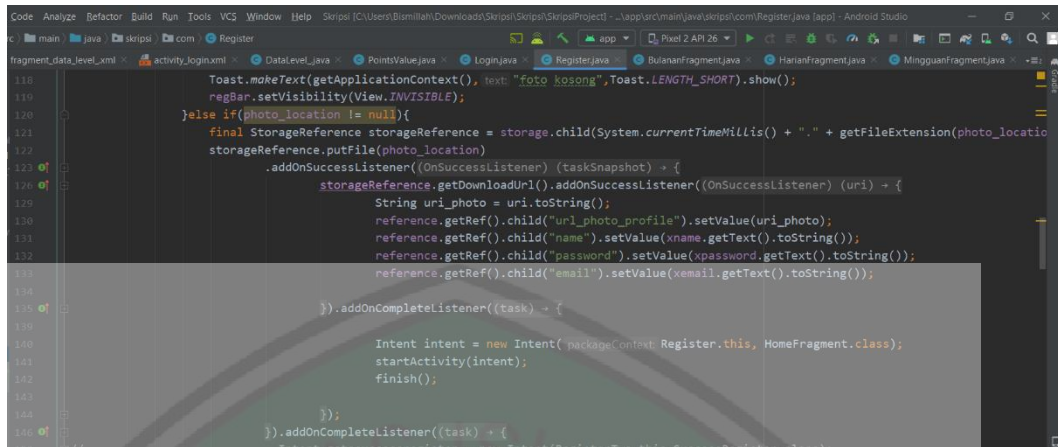
77         reference = FirebaseDatabase.getInstance().getReference().child("Users").child(username);
78         reference.addListenerForSingleValueEvent(new ValueEventListener() {
79             @Override
80             public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
81                 if (dataSnapshot.exists()) {
82                     //cek data password
83                     String datapassword = dataSnapshot.child("password").getValue().toString();
84
85                     if (password.equals(datapassword)) {
86
87                         SharedPreferences sharedPreferences = getSharedPreferences(USERNAME_KEY, MODE_PRIVATE);
88                         SharedPreferences.Editor editor = sharedPreferences.edit();
89                         editor.putString(username_key, username.getText().toString());
90                         editor.apply();
91

```

Gambar 4.1 Program Data Login

Pada gambar diatas pada baris program ke 77 ditunjukkan bahwasanya data username dan password di ambil dari *child* "Users" pada database *Firestore*. Jika data yang dimasukkan user tidak ada dalam database maka user tidak dapat melakukan login.

2. Register

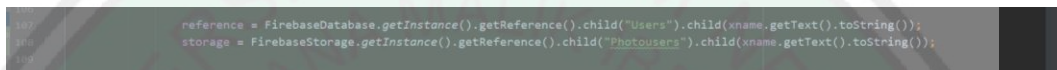


```

118 Toast.makeText(getApplicationContext(), text: "foto kosong", Toast.LENGTH_SHORT).show();
119 regBar.setVisibility(View.INVISIBLE);
120 }else if(photo_location != null){
121     final StorageReference storageReference = storage.child(System.currentTimeMillis() + "." + getFileExtension(photo_location));
122     storageReference.putFile(photo_location)
123         .addOnSuccessListener((OnSuccessListener) (taskSnapshot) -> {
124             storageReference.getDownloadUrl().addOnSuccessListener((OnSuccessListener) (uri) -> {
125                 String uri_photo = uri.toString();
126                 reference.getRef().child("url_photo_profile").setValue(uri_photo);
127                 reference.getRef().child("name").setValue(xname.getText().toString());
128                 reference.getRef().child("password").setValue(xpassword.getText().toString());
129                 reference.getRef().child("email").setValue(xemail.getText().toString());
130             });
131         }).addOnCompleteListener((task) -> {
132             Intent intent = new Intent(getApplicationContext(), Register.this, HomeFragment.class);
133             startActivity(intent);
134             finish();
135         });
136     });
137     });
138     });
139     });
140     });
141     });
142     });
143     });
144     });
145     });
146     });
147     });
148     });
149     });
150     });
151     });
152     });
153     });
154     });
155     });
156     });
157     });
158     });
159     });
160     });
161     });
162     });
163     });
164     });
165     });
166     });
167     });
168     });
169     });
170     });
171     });
172     });
173     });
174     });
175     });
176     });
177     });
178     });
179     });
180     });
181     });
182     });
183     });
184     });
185     });
186     });
187     });
188     });
189     });
190     });
191     });
192     });
193     });
194     });
195     });
196     });
197     });
198     });
199     });
200     });
201     });
202     });
203     });
204     });
205     });
206     });
207     });
208     });
209     });
210     });
211     });
212     });
213     });
214     });
215     });
216     });
217     });
218     });
219     });
220     });
221     });
222     });
223     });
224     });
225     });
226     });
227     });
228     });
229     });
230     });
231     });
232     });
233     });
234     });
235     });
236     });
237     });
238     });
239     });
240     });
241     });
242     });
243     });
244     });
245     });
246     });
247     });
248     });
249     });
250     });
251     });
252     });
253     });
254     });
255     });
256     });
257     });
258     });
259     });
260     });
261     });
262     });
263     });
264     });
265     });
266     });
267     });
268     });
269     });
270     });
271     });
272     });
273     });
274     });
275     });
276     });
277     });
278     });
279     });
280     });
281     });
282     });
283     });
284     });
285     });
286     });
287     });
288     });
289     });
290     });
291     });
292     });
293     });
294     });
295     });
296     });
297     });
298     });
299     });
300     });
301     });
302     });
303     });
304     });
305     });
306     });
307     });
308     });
309     });
310     });
311     });
312     });
313     });
314     });
315     });
316     });
317     });
318     });
319     });
320     });
321     });
322     });
323     });
324     });
325     });
326     });
327     });
328     });
329     });
330     });
331     });
332     });
333     });
334     });
335     });
336     });
337     });
338     });
339     });
340     });
341     });
342     });
343     });
344     });
345     });
346     });
347     });
348     });
349     });
350     });
351     });
352     });
353     });
354     });
355     });
356     });
357     });
358     });
359     });
360     });
361     });
362     });
363     });
364     });
365     });
366     });
367     });
368     });
369     });
370     });
371     });
372     });
373     });
374     });
375     });
376     });
377     });
378     });
379     });
380     });
381     });
382     });
383     });
384     });
385     });
386     });
387     });
388     });
389     });
390     });
391     });
392     });
393     });
394     });
395     });
396     });
397     });
398     });
399     });
400     });
401     });
402     });
403     });
404     });
405     });
406     });
407     });
408     });
409     });
410     });
411     });
412     });
413     });
414     });
415     });
416     });
417     });
418     });
419     });
420     });
421     });
422     });
423     });
424     });
425     });
426     });
427     });
428     });
429     });
430     });
431     });
432     });
433     });
434     });
435     });
436     });
437     });
438     });
439     });
440     });
441     });
442     });
443     });
444     });
445     });
446     });
447     });
448     });
449     });
450     });
451     });
452     });
453     });
454     });
455     });
456     });
457     });
458     });
459     });
460     });
461     });
462     });
463     });
464     });
465     });
466     });
467     });
468     });
469     });
470     });
471     });
472     });
473     });
474     });
475     });
476     });
477     });
478     });
479     });
480     });
481     });
482     });
483     });
484     });
485     });
486     });
487     });
488     });
489     });
490     });
491     });
492     });
493     });
494     });
495     });
496     });
497     });
498     });
499     });
500     });
501     });
502     });
503     });
504     });
505     });
506     });
507     });
508     });
509     });
510     });
511     });
512     });
513     });
514     });
515     });
516     });
517     });
518     });
519     });
520     });
521     });
522     });
523     });
524     });
525     });
526     });
527     });
528     });
529     });
530     });
531     });
532     });
533     });
534     });
535     });
536     });
537     });
538     });
539     });
540     });
541     });
542     });
543     });
544     });
545     });
546     });
547     });
548     });
549     });
550     });
551     });
552     });
553     });
554     });
555     });
556     });
557     });
558     });
559     });
560     });
561     });
562     });
563     });
564     });
565     });
566     });
567     });
568     });
569     });
570     });
571     });
572     });
573     });
574     });
575     });
576     });
577     });
578     });
579     });
580     });
581     });
582     });
583     });
584     });
585     });
586     });
587     });
588     });
589     });
590     });
591     });
592     });
593     });
594     });
595     });
596     });
597     });
598     });
599     });
600     });
601     });
602     });
603     });
604     });
605     });
606     });
607     });
608     });
609     });
610     });
611     });
612     });
613     });
614     });
615     });
616     });
617     });
618     });
619     });
620     });
621     });
622     });
623     });
624     });
625     });
626     });
627     });
628     });
629     });
630     });
631     });
632     });
633     });
634     });
635     });
636     });
637     });
638     });
639     });
640     });
641     });
642     });
643     });
644     });
645     });
646     });
647     });
648     });
649     });
650     });
651     });
652     });
653     });
654     });
655     });
656     });
657     });
658     });
659     });
660     });
661     });
662     });
663     });
664     });
665     });
666     });
667     });
668     });
669     });
670     });
671     });
672     });
673     });
674     });
675     });
676     });
677     });
678     });
679     });
680     });
681     });
682     });
683     });
684     });
685     });
686     });
687     });
688     });
689     });
690     });
691     });
692     });
693     });
694     });
695     });
696     });
697     });
698     });
699     });
700     });
701     });
702     });
703     });
704     });
705     });
706     });
707     });
708     });
709     });
710     });
711     });
712     });
713     });
714     });
715     });
716     });
717     });
718     });
719     });
720     });
721     });
722     });
723     });
724     });
725     });
726     });
727     });
728     });
729     });
730     });
731     });
732     });
733     });
734     });
735     });
736     });
737     });
738     });
739     });
740     });
741     });
742     });
743     });
744     });
745     });
746     });
747     });
748     });
749     });
750     });
751     });
752     });
753     });
754     });
755     });
756     });
757     });
758     });
759     });
760     });
761     });
762     });
763     });
764     });
765     });
766     });
767     });
768     });
769     });
770     });
771     });
772     });
773     });
774     });
775     });
776     });
777     });
778     });
779     });
780     });
781     });
782     });
783     });
784     });
785     });
786     });
787     });
788     });
789     });
790     });
791     });
792     });
793     });
794     });
795     });
796     });
797     });
798     });
799     });
800     });
801     });
802     });
803     });
804     });
805     });
806     });
807     });
808     });
809     });
810     });
811     });
812     });
813     });
814     });
815     });
816     });
817     });
818     });
819     });
820     });
821     });
822     });
823     });
824     });
825     });
826     });
827     });
828     });
829     });
830     });
831     });
832     });
833     });
834     });
835     });
836     });
837     });
838     });
839     });
840     });
841     });
842     });
843     });
844     });
845     });
846     });
847     });
848     });
849     });
850     });
851     });
852     });
853     });
854     });
855     });
856     });
857     });
858     });
859     });
860     });
861     });
862     });
863     });
864     });
865     });
866     });
867     });
868     });
869     });
870     });
871     });
872     });
873     });
874     });
875     });
876     });
877     });
878     });
879     });
880     });
881     });
882     });
883     });
884     });
885     });
886     });
887     });
888     });
889     });
890     });
891     });
892     });
893     });
894     });
895     });
896     });
897     });
898     });
899     });
900     });
901     });
902     });
903     });
904     });
905     });
906     });
907     });
908     });
909     });
910     });
911     });
912     });
913     });
914     });
915     });
916     });
917     });
918     });
919     });
920     });
921     });
922     });
923     });
924     });
925     });
926     });
927     });
928     });
929     });
930     });
931     });
932     });
933     });
934     });
935     });
936     });
937     });
938     });
939     });
940     });
941     });
942     });
943     });
944     });
945     });
946     });
947     });
948     });
949     });
950     });
951     });
952     });
953     });
954     });
955     });
956     });
957     });
958     });
959     });
960     });
961     });
962     });
963     });
964     });
965     });
966     });
967     });
968     });
969     });
970     });
971     });
972     });
973     });
974     });
975     });
976     });
977     });
978     });
979     });
980     });
981     });
982     });
983     });
984     });
985     });
986     });
987     });
988     });
989     });
990     });
991     });
992     });
993     });
994     });
995     });
996     });
997     });
998     });
999     });
1000    });

```

Gambar 4.2 (a) Kode Program Input Data User

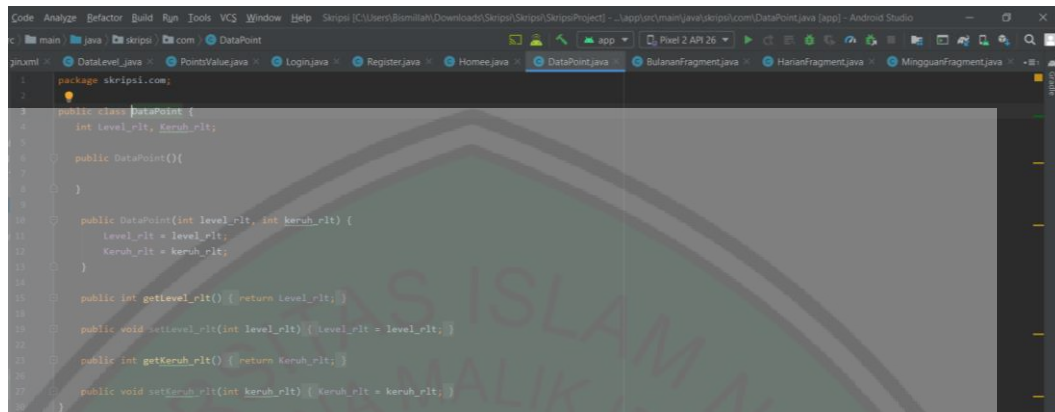


```

1007 reference = FirebaseDatabase.getInstance().getReference().child("Users").child(xname.getText().toString());
1008 storage = FirebaseStorage.getInstance().getReference().child("PhotoUsers").child(xname.getText().toString());
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306

```

Pada gambar 4.3 menjelaskan kode program untuk menampilkan data relative dari sensor dalam bentuk grafik persen. Dimana data tersebut didapatkan dari database *Child* “Data” dengan menggunakan fungsi constructor Seperti gambar dibawah ini.



```

package skripsi.com;

public class DataPoint {
    int level_rlt, keruh_rlt;

    public DataPoint(){}

    public DataPoint(int level_rlt, int keruh_rlt) {
        level_rlt = level_rlt;
        keruh_rlt = keruh_rlt;
    }

    public int getLevel_rlt() {return level_rlt;}

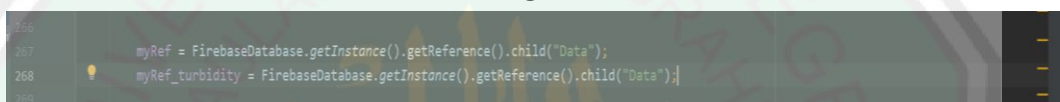
    public void setLevel_rlt(int level_rlt) {level_rlt = level_rlt;}

    public int getKeruh_rlt() {return keruh_rlt;}

    public void setKeruh_rlt(int keruh_rlt) {keruh_rlt = keruh_rlt;}
}

```

Gambar 4.4 (a) Kode Program Konstruktor *Get Data*



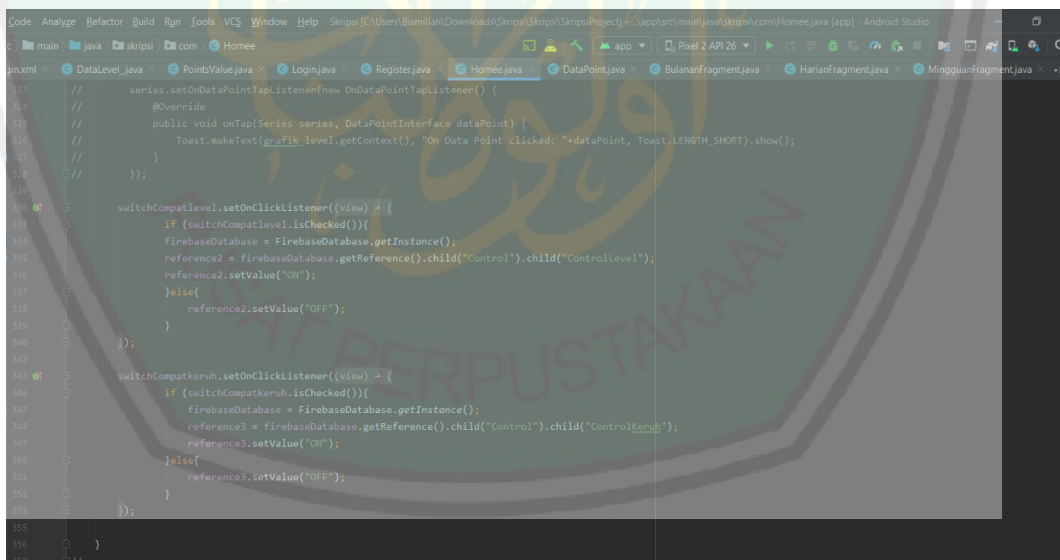
```

256
257 myRef = FirebaseDatabase.getInstance().getReference().child("Data");
258 myRef_turbidity = FirebaseDatabase.getInstance().getReference().child("Data");
259

```

Gambar 4.4 (b) Kode Program Ambil data dari *Database*

4. Control Status Pompa



```

323 // series.setOnDataPointTapListener(new OnDataPointTapListener() {
324 // @Override
325 // public void onTap(Series series, DataPointInterface dataPoint) {
326 // Toast.makeText(grafik_level.getContext(), "On Data Point clicked: " + dataPoint, Toast.LENGTH_SHORT).show();
327 // }
328 // });
329
330 switchCompatlevel.setOnCheckedChangeListener((view) -> {
331 // if (switchCompatlevel.isChecked()){
332 // firebaseDatabase = FirebaseDatabase.getInstance();
333 // reference2 = FirebaseDatabase.getInstance().getReference().child("Control").child("Controllevel");
334 // reference2.setValue("ON");
335 // }else{
336 // reference2.setValue("OFF");
337 // }
338 // });
339
340 switchCompatkeruh.setOnCheckedChangeListener((view) -> {
341 // if (switchCompatkeruh.isChecked()){
342 // firebaseDatabase = FirebaseDatabase.getInstance();
343 // reference3 = FirebaseDatabase.getInstance().getReference().child("Control").child("Controlkeruh");
344 // reference3.setValue("ON");
345 // }else{
346 // reference3.setValue("OFF");
347 // }
348 // });
349
350 }
351 //
352 //
353 //
354 //
355 //
356 //
357 //

```

Gambar 4.5 Program Kontrol Status Pompa

Kode program pada gambar 4.5 menggambarkan bagaimana mengontrol status pompa pada perangkat keras, yang mana Ketika tombol button ON dinyalakan maka akan mengirim kondisi pada database control, lalu mengirimkannya ke perangkat keras melalui Sistem IOT dengan NodeMcu. Begitupula sebaliknya.

5. Monitoring Data Keruh

Monitoring data keruh dilakukan secara berkala, berdasarkan perjam, hari dan minggu. Data perjam didapatkan dari data sensor yang dikirimkan setiap 3 jam sekali. Kemudian data perjam di hitung rata-ratanya ketika sudah mencapai 8 data masuk untuk mendapatkan data perharinya. Kemudian data perhari juga akan dihitung rata-ratanya Ketika sudah mencapai 7 data masuk, untuk memperoleh data perminggu. Data ditampilkan dalam bentuk grafik.

Berikut merupakan kode program untuk mendapataka data perjam.

```

1 package skripsi.com.Fragment_Keruh;
2
3 public class PointsValue3 {
4
5     String Data, day;
6
7     public PointsValue3(){
8
9     }
10
11    public PointsValue3(String data, String day) {
12        Data = data;
13        this.day = day;
14    }
15
16    public String getData() { return Data; }
17
18    public void setData(String data) { Data = data; }
19
20    public String getDay() { return day; }
21
22    public void setDay(String day) { this.day = day; }
23
24 }
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Gambar 4.6 Kode Program Monitoring Data keruh Perjam

Kode program diatas data keruh perhari disimpan didalam database *child* “Datakeruh”, kemudian data tersebut dipanggil menggunakan fungsi *constructor* dengan *variable* “data” untuk menampilkan data sensor dan “day” untuk menampilkan data waktu *update* pada database. Kemudian semua data jam yang masuk akan dirata-rata untuk memperoleh data harian, dengan kode program seperti dibawah ini

```

136      @Override
137      public void onClick(View view) {
138          myRef.addValueEventListener(new ValueEventListener() {
139              @Override
140              public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
141
142                  for (DataSnapshot mydata : dataSnapshot.getChildren()) {
143                      //count child
144                      Map<String, Object> map = (Map<String, Object>) mydata.getValue();
145                      Object jumlah = map.get("yValue");
146                      int Value = Integer.parseInt(String.valueOf(jumlah));
147                      sum+=Value;
148                  }
149
150                  if (dataSnapshot.exists()){
151                      count = (int) dataSnapshot.getChildrenCount();
152                      if (count >= 4){
153                          simpandata();
154                          //hapus
155                          reference2 = FirebaseDatabase.getInstance().getReference( path: "grafik").child("DataKeruh");
156                          reference2.removeValue();
157                      }
158                  }
159              }
160          }
161
162          @Override
163          public void onCancelled(@NonNull DatabaseError databaseError) {
164
165          });
166      });
167

```

Gambar 4.7 Program Menghitung Rata-Rata Data Perhari

Dari program diatas memperoleh hasil untuk data perhari, yang disimpan dalam database *Child* “DataPerhari”. Kemudian data tersebut di tampilkan dalam bentuk grafik pada halaman datakeruh perhari. Seperti gambar dibawah ini.

```

168
169 private void Tampil() {
170     myRef = FirebaseDatabase.getReference( path: "DataPerhari");
171     myRef.addValueEventListener(new ValueEventListener() {
172
173         @Override
174         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
175             ArrayList<BarEntry> dataValue= new ArrayList<>();
176             ArrayList<BarEntry> lData = new ArrayList<>();
177             ArrayList<String> lLabels = new ArrayList<>();
178
179             if (dataSnapshot.hasChildren()){
180                 for (DataSnapshot mydata : dataSnapshot.getChildren()){
181                     Model_minggu model_minggu = mydata.getValue(Model_minggu.class);
182                     lData.add(new BarEntry(model_minggu.getDay(),model_minggu.getData()));
183                 }
184
185                 BarDataSet dataSet = new BarDataSet(lData, label: "Day");
186                 BarData data = new BarData(dataSet);
187
188                 data.setBarWidth(0.9f);
189                 data.setValueTextSize(10f);
190                 data.setValueTextColor(Color.BLACK);
191                 dataSet.setColors(ColorTemplate.COLORFUL_COLORS);
192                 dataSet.setValueTextColor(Color.BLACK);
193                 dataSet.setValueTextSize(12);
194                 String[] hari=new String[]{"Day"};
195                 XAxis xAxis =bargraph.getXAxis();
196                 xAxis.setValueFormatter(new MyXAxis(hari));
197                 xAxis.setPosition(XAxis.XAxisPosition.BOTH_SIDED);
198                 bargraph.setDrawBarShadow(false);
199                 bargraph.setDrawGridBackground(false);
200                 bargraph.setData(data);
201                 bargraph.animate();
202                 bargraph.animateY( durationMillis: 1500);
203
204             }
205         }
206     });
207 }

```

Gambar 4.8 Kode Program Tampil Dataperhari

Seperti dataperjam di atas data perhari juga akan dihitung rata-ratanya untuk memperoleh data perminggu. Proses untuk memperoleh data perminggu sama dengan dataperhari, yaitu dengan menghitung jumlah rata-rata dari data perhari. Kemudian ditampilkan dalam bentuk grafik pada halaman “datakeruhperminggu”.

6. Monitoring Data Level

Monitoring data level dilakukan secara berkala seperti data keruh, hanya saja monitoring data level mengklasifikasi berdasarkan 2 katagori perjam dan hari. Data perjam didapatkan dari data sensor yang dikirimkan setiap 3 jam sekali. Kemudian data perjam di hitung rata-ratanya ketika sudah mencapai 8 data masuk untuk mendapatkan data perharinya.

Metode pengolahan data monitoring keruh dan monitoring level sama. Pada tabel 4.2 dibawah ini menjelaskan *file sourcecode* yang digunakan untuk pembangunan monitoring data level.

Tabel 4.2 *File Sourcecode* Pembangunan Halaman Monitoring Level Air

No	Pages	File PHP	Metode	Data
1	DataLevelPerJam	Data-poinlevel.java	GET	Child “datalevel”, “datalevelperjam” : data, waktu
		DataPerjamL.java	POST	Sensor level : Data_level Waktu Update: waktu
2	DataLevelPerHari	Data-poinlevel.java	GET	Child “datalevel”, “datalevelperhari” : data, waktu
		DataPerhariL.java	POST	Sensor level : Data_level Waktu Update: waktu

4.1.3 Implementasi Jadwal Monitoring

Dalam penerapan penjadwalan, pengambilan data sensor level dan kekeruhan sangat berpengaruh. Sehingga dilakukanlah pengambilan kondisi dengan langkah – langkah berikut ini :

Langkah pertama merupakan pengklasifikasian level dan kekeruhan sensor berdasarkan level air serta status dari kekeruhan pada wadah air minum puyuh. Dalam proses pengklasifikasiannya terdapat beberapa aturan yang telah dibuat, aturan untuk level air yaitu jika jumlah lebih dari 70% maka tanah tersebut berstatus “Tinggi”, kemudian jika jumlah level antara 50% sampai 70% maka air tersebut berstatus “Sedang”, kemudian jika jumlah level antara kurang dari 50% sampai dengan 10% maka air tersebut berstatus “rendah”. Untuk kekeruhan yaitu jika jumlah nilai kekeruhan air (Ntu) lebih dari 70% maka status air tersebut adalah “Keruh”, kemudian jika jumlah kekeruhan antara 50% sampai 70% maka air

tersebut bersetatus “Normal”, dan jika jumlah kekeruhan antara 10% sampai 50% maka air tersebut bersetatus “Jernih”.

Data yang kami dapatkan dilakukan dengan monitoring menggunakan perangkat *Android* selama 7 hari, dengan pengiriman data setiap 3 jam sekali. Uji coba dilakukan selama 2 kali, uji coba pertama dilakukan selama 2 hari dengan menggunakan 2 ekor burung puyuh berumur 15 bulan. Uji coba kedua dilakukan dengan 5 hari dengan menggunakan 4 ekor burung puyuh berumur 15 bulan. wadah air minum yang digunakan berukuran 2 liter. Spesifikasi selengkapnya seperti gambar dibawah ini.



Gambar 4.9 Prototype Alat Uji Coba

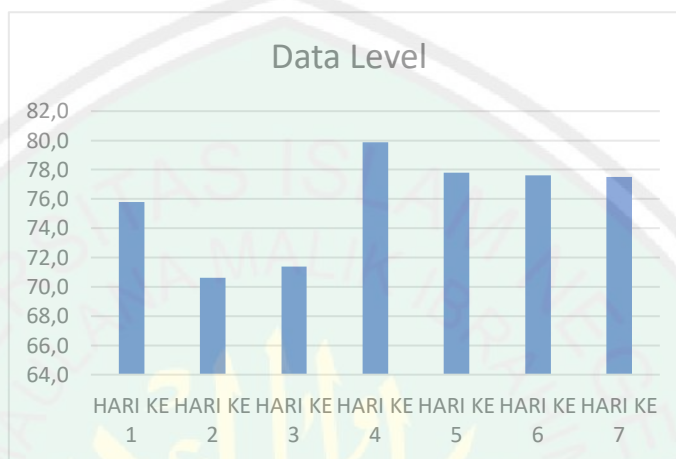
Waktu pengujian dilakukan pada pukul 06:57:27 tanggal 5 Oktober 2020, dengan kondisi air ketinggian 5.88 cm dan kekeruhan 7.5 Ntu. Berikut merupakan tabel data sensor level dan kekeruhan yang diambil serta waktu pengambilan data dan terdapat kondisi dari wadah air minum burung puyuh. Adapun untuk data dari sensor level dan kekeruhan dengan menjadikan nilai level kebentuk persentase agar dapat dilakukan pengkualifikasian “Status Level”

Tabel 4.3 Data Kondisi Wadah Air Minum Puyuh dengan “Status Level”

No,	Waktu	Nilai Level Air (%)	Turbidity Sensor(Ntu)	Status
Uji Coba Hari Pertama dengan 2 ekor puyuh				
0	07:57:27/5:10:2020	39,2	7,5	Rendah
1	11:01:24/5:10:2020	82,7	7,38	Tinggi
2	14:01:33/5:10:2020	82,3	7,34	Tinggi
3	17:05:24/5:10:2020	81,2	7,26	Tinggi
4	20:05:33/5:10:2020	80,6	7,23	Tinggi
5	23:05:42/5:10:2020	80,1	7,16	Tinggi
6	02:05:51/6:10:2020	80,1	7,16	Tinggi
7	05:05:59/6:10:2020	80,0	7,17	Tinggi
8	08:06:09/6:10:2020	76,1	7,23	Tinggi
Uji Coba Hari Kedua dengan 4 ekor puyuh				
1	11:06:18/6:10:2020	76,1	7,23	Tinggi
2	14:06:27/6:10:2020	72,3	7,4	Tinggi
3	17:06:36/6:10:2020	72,2	7,41	Tinggi
4	20:10:11/6:10:2020	68,8	7,43	Sedang
5	23:12:06/6:10:2020	71,7	7,38	Tinggi
6	02:12:16/7:10:2020	68,9	7,44	Sedang
7	05:17:40/7:10:2020	66,1	6,82	Sedang
8	08:23:22/7:10:2020	68,8	7,43	Sedang
Uji Coba Hari Ketiga dengan 4 ekor puyuh				
1	11:29:03/7:10:2020	66,7	7,46	Sedang
2	14:34:53/7:10:2020	66,5	7,46	Sedang
3	17:41:32/7:10:2020	58,7	8,62	Sedang
4	20:48:52/7:10:2020	56,7	8,55	Sedang
5	23:56:07/8:10:2020	81,9	8,03	Tinggi
6	02:56:11/8:10:2020	81,3	7,98	Tinggi
7	05:56:19/8:10:2020	79,6	7,15	Tinggi
8	08:56:29/8:10:2020	79,6	7,15	Tinggi
Uji Coba Hari Keempat dengan 4 ekor puyuh				

1	11:56:37/8:10:2020	81,9	7,06	Tinggi
2	15:56:50/8:10:2020	81,3	6,98	Tinggi
3	18:56:59/8:10:2020	72,9	7,49	Tinggi
4	21:57:02/8:10:2020	68,7	7,42	Sedang
5	23:57:07/8:10:2020	86,4	7,71	Tinggi
6	02:57:23/9:10:2020	83,1	7,43	Tinggi
7	05:59:08/9:10:2020	81,7	7,32	Tinggi
8	08:59:45/9:10:2020	80	7,28	Tinggi
Uji Coba Hari Kelima dengan 4 ekor puyuh				
1	11:59:37-9:10:2020	81,3	6,99	Tinggi
2	15:59:50/9:10:2020	81,3	7	Tinggi
3	18:59:55/9:10:2020	79,5	7,27	Tinggi
4	21:59:58/9:10:2020	73,9	9,63	Tinggi
5	23:59:07/9:10:2020	70,9	9,42	Tinggi
6	02:59:23/10:10:2020	69,1	9,17	Sedang
7	06:01:08/10:10:2020	83,7	6,82	Tinggi
8	09:01:15/10:10:2020	82,0	6,93	Tinggi
Uji Coba Hari Keenam dengan 4 ekor puyuh				
1	12:01:37/10:10:2020	81,9	7,94	Tinggi
2	15:56:50/10:10:2020	80,7	8,12	Tinggi
3	18:56:59/10:10:2020	74,9	8,54	Tinggi
4	21:57:02/10:10:2020	73,0	8,62	Tinggi
5	23:57:07/10:10:2020	70,4	7,8	Tinggi
6	02:57:23/11:10:2020	68,5	7,23	Sedang
7	05:59:08/11:10:2020	86,7	6,82	Tinggi
8	08:03:45/11:10:2020	85,0	6,93	Tinggi
Uji Coba Hari Ketujuh dengan 4 ekor puyuh				
1	11:56:37/11:10:2020	82,3	7,92	Tinggi
2	15:56:50/11:10:2020	79,2	6,98	Tinggi
3	18:56:59/11:10:2020	76,9	7,19	Tinggi
4	21:57:02/11:10:2020	72,0	7,27	Tinggi
5	23:57:07/11:10:2020	70,8	7,81	Tinggi
6	02:57:23/12:10:2020	66,7	7,25	Sedang
7	05:59:08/12:10:2020	86,7	6,88	Tinggi
8	08:59:45/12:10:2020	12,3	6,94	Tinggi

Dari tabel 4.2 diatas merupakan tabel data dengan pengklasifikasian dengan status kondisi Level air . Dimana data tersebut diambil selama 7 hari tanpa henti dengan pengiriman data setiap 3 jam sekali. Hari pertama menggunakan 2 ekor puyuh dan hari kedua sampai hari ketujuh menggunakan 4 ekor puyuh. Maka bisa didapatkan rata-rata konsumsi air minum burung puyuh, seperti grafik dibawah ini :



Gambar 4.10 Grafik Level Air pada Wadah Air Minum Burung Puyuh

Pada gambar diatas menggambarkan rata-rata konsumsi air minum burung puyuh dengan wadah air minum berukuran 2 liter. Pada hari pertama dengan 2 ekor puyuh didapatkan nilai rata-rata 76% yang artinya masih tergolong katagori “tinggi”. Kemudian pada hari kedua dan ketiga, ada peningkatan konsumsi air minum dari burung puyuh yaitu 70% tergolong katagori normal. Hari keempat sampai ke tujuh ada penurunan konsumsi air minum, dengan rata-rata 78%, tergolong katagori tinggi.

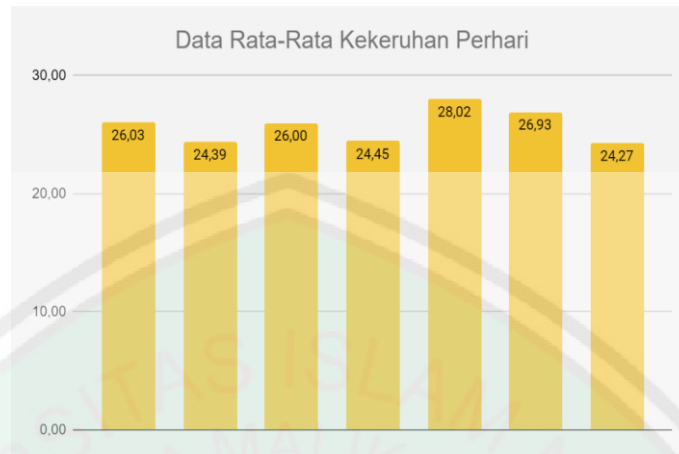
Adapun pengkulifikasian data kekeruhan adalah sebagai berikut, dengan status keruh :

Tabel 4.4 Data Kondisi Wadah Air Minum Puyuh dengan “Status Keruh”

No,	Waktu	Nilai Level Air (cm)	Turbidity Sensor(%)	Status
Uji Coba Hari Pertama dengan 2 ekor puyuh				
0	07:57:27/5:10:2020	5,88	25,00	Jernih
1	11:01:24/5:10:2020	12,41	24,60	Jernih
2	14:01:33/5:10:2020	12,34	24,47	Jernih
3	17:05:24/5:10:2020	12,18	24,20	Jernih
4	20:05:33/5:10:2020	12,09	40,30	Jernih
5	23:05:42/5:10:2020	12,01	23,87	Jernih
6	02:05:51/6:10:2020	12,01	23,87	Jernih
7	05:05:59/6:10:2020	12	23,90	Jernih
8	08:06:09/6:10:2020	11,41	24,10	Jernih
Uji Coba Hari Kedua dengan 4 ekor puyuh				
1	11:06:18/6:10:2020	11,41	24,10	Jernih
2	14:06:27/6:10:2020	10,85	24,67	Jernih
3	17:06:36/6:10:2020	10,83	24,70	Jernih
4	20:10:11/6:10:2020	10,32	24,77	Jernih
5	23:12:06/6:10:2020	10,75	24,60	Jernih
6	02:12:16/7:10:2020	10,34	24,80	Jernih
7	05:17:40/7:10:2020	9,91	22,73	Jernih
8	08:23:22/7:10:2020	10,32	24,77	Jernih
Uji Coba Hari Ketiga dengan 4 ekor puyuh				
1	11:29:03/7:10:2020		24,87	Jernih
2	14:34:53/7:10:2020	10	24,87	Jernih
3	17:41:32/7:10:2020	9,98	28,73	Jernih
4	20:48:52/7:10:2020	8,81	28,50	Jernih
5	23:56:07/8:10:2020	8,5	26,77	Jernih
6	02:56:11/8:10:2020	12,28	26,60	Jernih
7	05:56:19/8:10:2020	12,19	23,83	Jernih
8	08:56:29/8:10:2020	11,94	23,83	Jernih
Uji Coba Hari Keempat dengan 4 ekor puyuh				
1	11:56:37/8:10:2020	12,28	23,53	Jernih
2	15:56:50/8:10:2020	12,19	23,27	Jernih
3	18:56:59/8:10:2020	10,93	24,97	Jernih
4	21:57:02/8:10:2020	10,3	24,73	Jernih

5	23:57:07/8:10:2020	12,96	25,70	Jernih
6	02:57:23/9:10:2020	12,36	24,77	Jernih
7	05:59:08/9:10:2020	12,55	24,40	Jernih
8	08:59:45/9:10:2020	12,3	24,27	Jernih
Uji Coba Hari Kelima dengan 4 ekor puyuh				
1	11:59:37-9:10:2020	12,2	23,30	Jernih
2	15:59:50/9:10:2020	12,19	23,33	Jernih
3	18:59:55/9:10:2020	11,93	24,23	Jernih
4	21:59:58/9:10:2020	10,78	32,10	Jernih
5	23:59:07/9:10:2020	10,58	31,40	Jernih
6	02:59:23/10:10:2020	10,36	30,57	Jernih
7	06:01:08/10:10:2020	13	29,40	Jernih
8	09:01:15/10:10:2020	12,3	29,83	Jernih
Uji Coba Hari Keenam dengan 4 ekor puyuh				
1	12:01:37/10:10:2020	12,28	31,00	Jernih
2	15:56:50/10:10:2020	12,1	31,27	Jernih
3	18:56:59/10:10:2020	11,23	28,47	Jernih
4	21:57:02/10:10:2020	10,95	28,73	Jernih
5	23:57:07/10:10:2020	10,56	26,00	Jernih
6	02:57:23/11:10:2020	10,28	24,10	Jernih
7	05:59:08/11:10:2020	13	22,73	Jernih
8	08:03:45/11:10:2020	12,75	23,10	Jernih
Uji Coba Hari Ketujuh dengan 4 ekor puyuh				
1	11:56:37/11:10:2020	12,35	26,40	Jernih
2	15:56:50/11:10:2020	11,88	23,27	Jernih
3	18:56:59/11:10:2020	11,53	23,97	Jernih
4	21:57:02/11:10:2020	10,8	24,23	Jernih
5	23:57:07/11:10:2020	10,62	26,03	Jernih
6	02:57:23/12:10:2020	10	24,17	Jernih
7	05:59:08/12:10:2020	13	22,93	Jernih
8	08:59:45/12:10:2020	12,83	23,13	Jernih

Dari tabel 4.3 diatas, data keruh didapatkan rata-rata tingkat kekeruhan perhari dapat dilihat pada gambar grafik berikut :



Gambar 4.11 Grafik Keruh Air pada Wadah Air Minum Burung Puyuh

Pada gambar diatas menggambarkan rata-rata tingkat kekeruhan air minum burung puyuh dengan wadah air minum berukuran 2 liter, yang dilakukan uji coba selama 7 hari, dengan tempat uji coba secara *indoor* suhu kurang lebih 30°C mendapatkan rata-rata perhari adalah 30% yang dapat dikatakan sebagai kategori “jernih”.

4.1.4 Implementasi Kontroling Air Minum

Implementasi kontroling air minum burung puyuh ini bertujuan agar dapat menjaga kebersihan air minum burung puyuh. Dalam pengujian ini dilakukan observasi ke peternak senior yang berada di daerah Gondanglegi Kabupaten Malang. Degan data kualitas air minum yang diperoleh adalah sebagai berikut :

1. Kontrol Kekeruhan

Tabel 4.5 Aturan Kualifikasi Kekeruhan Air Minum

No	Kondisi Air	Jumlah Ntu (Nephelometric Turbidity Units)	Status Pompa Menguras
1	Jernih	0-10 Ntu	OFF
2	Sedang	10-20 Ntu	OFF
3	Keruh	20-30 Ntu	ON

2. Kontrol Pengisian

Tabel 4.6 Aturan Kualifikasi Level Air Minum

No	Kondisi Air	Jumlah Ntu (Nephelometric Turbidity Units)	Status Pompa Menngisi
1	Tinggi	10-15 cm	OFF
2	Sedang	5-10 cm	OFF
3	Rendah	0-5 cm	ON

4.2 Implementasi *Interface*

Dari implementasi antar muka (interface) , peneliti mengklasifikasikan menjadi 5 bagian utama yaitu *Splash Screen*, *Login*, *Register*, *Dashboard / Home Page* (halaman depan), Data level air dan Data kekeruhan air dan ada fitur tambahan digunakan untuk implementasi metode yang digunakan setiap peneliti. Berikut merupakan *interface sistem* penyiraman otomatis :

4.2.1 Implementasi Tampilan *Splash Screen*

Berikut adalah tampilan *splash screen* yang mana tampilan ini adalah tampilan yang akan pertama kali muncul didalam aplikasi, yang mempunyai fungsi sebagai pengenalan aplikasi monitoring air minum pada wadah burung puyuh



Gambar 4.12 Tampilan Antar Muka *Login and Register*

4.2.2 Implementasi Antarmuka *Login dan Register*

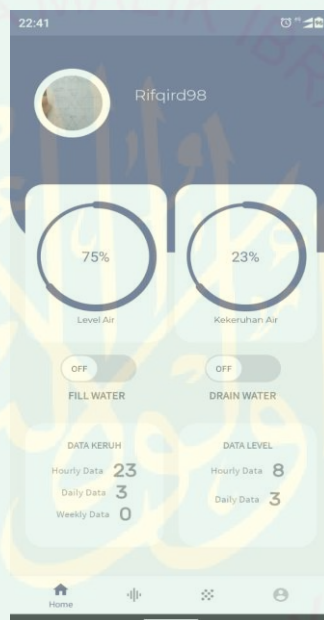
Berikut merupakan halaman *Login dan Register*, halaman ini bertujuan untuk mengetahui identitas pengguna dari aplikasi ini. Di dalam halaman ini terdapat beberapa form yang perlu di isi oleh pengguna antara lain username, password, email dan foto *profile*. Gambaran antarmuka *login dan register* dipaparkan pada gambar dibawah ini:



Gambar 4.12 Tampilan Antar Muka *Login and Register*

4.2.3 Implementasi Antar muka *Dashboard*

Berikut merupakan halaman utama yaitu dashboard . Pada halaman dashboard terdapat informasi level air dan kekeruhan air secara realtime kemudian juga ada jumlah data level dan kekeruhan air yang masuk berdasarkan jam, hari dan minggu. Dan juga terdapat *button* mengisi dan menguras air yang dapat menyalakan pompa secara otomatis maupun manual dari perangkat lunak sesuai yang dibutuhkan oleh peternak. Gambaran antarmuka dashboard akan dipaparkan pada gambar dibawah ini:



Gambar 4.13 Tampilan Antarmuka *dashboard*

4.2.4 Implementasi Antar muka Monitoring Kekeruhan

Berikut ini adalah halaman monitoring kekeruhan air, pembuatan halaman ini bertujuan agar peternak dapat memonitor kekeruhan air minum puyuh dan menjaganya agar tetap bersih dan sehat untuk diminum burung puyuh. halaman ini menampilkan grafik dan data sensor *Turbidity* berdasarkan perjam, hari, dan minggu. Data *hourly* diperoleh dari data sensor pada hardware yang jika kekeruhan air meningkat pada tingkatan tertinggi dari kriteria yang diberikan maka akan dilakukan pengurasan secara otomatis. Dan data *daily* diperoleh dari rata-rata data kekeruhan perharinya begitu pula *weekly* diperoleh dari rata-rata data kekeruhan perminggunya. Berikut adalah gambaran antarmuka halaman kekeruhan



Gambar 4.14 Tampilan Antarmuka Kekeruhan Air

4.2.5 Implementasi Antar muka Monitoring Level Air

Berikut ini adalah halaman monitoring level air, pembuatan halaman ini bertujuan agar peternak dapat memonitor level air minum puyuh dan mengetahui seberapa banyak konsumsi minum burung puyuh dan menjaganya agar tidak sampai kehabisan. halaman ini menampilkan grafik dan data sensor *Ultrasonic* berdasarkan perjam dan hari. Data *hourly* diperoleh dari data sensor pada hardware yang jika level air menurun dan pada tingkatan terendah dari kriteria yang diberikan maka akan dilakukan pengisian secara otomatis. Dan data *daily* diperoleh dari rata-rata data level perharinya. Halaman ini hanya menggunakan 2 kondisi pemantauan, kondisi perjam dan hari berbeda dengan kekeruhan dikarenakan tingkat perubahan data yang diperoleh level lebih signifikan dibanding dengan kekeruhan. Berikut adalah gambaran antarmuka halaman monitoring level :



Gambar 4.15 Tampilan Antarmuka Level Air

4.3 Pengujian Sistem

Recana pengujian dilakukan guna menguraikan kegiatan yang akan dilakuakn sebagai bentuk pengujian suatu perangkat lunak, seperti menentukan asumsi dan pengujian yang digunakan. Perangkat lunak berbasis *android* berjalan seperti pada diagram dibawah ini:

4.3.1 Metode Pengujian Sistem

Melakukan pengujian perangkat lunak penyiraman tanaman otomatis ini menggunakan pengujian *black box testing*. Dimana *black box testing* adalah pengujian yang dilakukan kepada perangkat lunak yang terfokuskan dalam fungsionalitas dari perangkat lunak dan bagaimana hasil yang diberikan dari perangkat lunak tersebut.

Adapun data pengujian yang digunakan untuk menguji fungsionalitas dari perangkat lunak monitoring dan kontroling air minum burung puyuh dapat dilihat pada tabel berikut:

Tabel 4.7 Data Pengujian

Kelas Uji	Butir Uji	Jenis Pengujian
<i>Splash Screen</i>	Tampilan <i>Splash</i> menunjukkan nama dari aplikasi.	<i>Black Box</i>
<i>Login & Register</i>	<i>Register</i> akun	<i>Black Box</i>
	<i>Login</i> aplikasi dengan akun yang sudah terdaftar	<i>Black Box</i>
<i>Dashboard</i>	<i>Update</i> data sensor realtime	<i>Black Box</i>
	Status pompa pengisian dan pengurusan	<i>Black Box</i>
	<i>Update</i> data database level dan kekeruhan, perjam, hari dan minggu	<i>Black Box</i>
Monitor Level	<i>Update</i> data Level setiap 3 jam sekali, dan dikomulasikan hasil rata-ratnya perhari.	<i>Black Box</i>
Monior Keruh	<i>Update</i> data keruh setiap 3 jam sekali, dan dikomulasikan hasil rata-ratnya perhari dan minggu.	<i>Black Box</i>

Dalam tabel 4.3 menerangkan bahwa laman-laman yang akan diuji fungsionalitasnya dan melalui *metode Black Box Testing*.

4.3.2 Kasus dan Hasil Pengujian Fungsional

Langkah ini bertujuan untuk menganalisis sistem yang sedang dibangun agar dapat meminimalkan kesalahan saat implementasinya sehingga perancangan yang dibangun akan digunakan tanpa kesalahan dalam sistem tersebut. Sehingga dalam pengujian ini menggunakan *blackbox* metode yang dimana testing berupa alur yang telah melalui perancangan sebelumnya. Dibawah ini merupakan serangkaian langkah-langkah pengujian yang dilakukan dapat dilihat hasilnya berikut ini :

1. Pengujian *Splash Screen*

Tabel 4.8 Hasil Pengujian Fungsionalitas *Splash Screen*

Aksi Aktor	Reaksi Sistem	Kesimpulan
1. Memulai Aplikasi	2. Menampilkan kelas <i>splash.java</i> untuk menampilkan nama dari aplikasi serta button untuk memulai masuk halaman utama aplikasi. Jika aplikasi belum ada akun yang terdaftar maka akan di bawah ke halaman login dan register. Jika sudah ada akun yang pernah masuk, maka langsung menuju halaman <i>dashboard</i> .	Sesuai

Hasil dari pengujian fungsionalitas laman *Login* dan *Register* sebagai laman pertama membuka aplikasi pada tabel 4.4 menunjukkan bahwa ketika user mengakses Aplikasi, reaksi dari sistem yaitu menampilkan nama dari aplikasi serta button untuk memulai masuk halaman utama aplikasi. Jika aplikasi belum ada akun yang terdaftar maka akan di bawah ke halaman login dan register. Jika sudah ada akun yang pernah masuk, maka langsung menuju halaman *dashboard*. Berdasarkan hal tersebut dapat diperoleh hasil pengujian fungsionalitas sesuai.

2. Pengujian *Login & Register*

Tabel 4.9 Hasil Pengujian Fungsionalitas *Login* dan *Register*

Aksi Aktor	Reaksi Sistem	Kesimpulan
3. Memulai halaman <i>login</i>	4. Memanggil kelas <i>login.java</i> , untuk masuk ke dalam aplikasi menggunakan akun yang sudah terdaftar.	Sesuai
5. Menekan <i>Button Login</i>	6. Mengecek data <i>Users</i> ke database. jika data user ada didalam database, maka lanjut ke halaman <i>dashboard</i>	Sesuai
7. Menekan <i>Create new akun</i>	8. Perpindah ke halaman <i>register</i> .	Sesuai
9. Memulai halaman <i>register</i>	10. Memanggil file <i>register.java</i> untuk Menampilkan form <i>username</i> , <i>password</i> , <i>email</i>	Sesuai
11. Menekan <i>button register</i>	12. <i>Save data user</i> di database <i>Firebase</i>	Sesuai

Hasil dari pengujian fungsionalitas laman *Login* dan *Register* pada tabel 4.4 menunjukkan bahwa ketika user mengakses halaman login , reaksi dari sistem yaitu menampilkan nama dari aplikasi serta button untuk memulai masuk halaman utama aplikasi. Jika aplikasi belum ada akun yang terdaftar maka akan di bawah ke halaman login dan register. Jika sudah ada akun yang pernah masuk, maka langsung menuju halaman *dashboard*. Berdasarkan hal tersebut dapat diperoleh hasil pengujian fungsionalitas sesuai.

3. Pengujian *Dashboard*

Tabel 4.10 Hasil Pengujian Fungsionalitas *Dashboard*

Aksi Aktor	Reaksi Sistem	Kesimpulan
13. Memulai halaman <i>Dashboard</i>	14. Memanggil file <i>home.java</i> untuk menampilkan menu utama aplikasi yang menampilkan status informasi sensor terbaru. 15. Melakukan inisiasi hubungan ke <i>database</i>	Sesuai
16. Menekan <i>Button fill Water</i>	17. Pompa mengisi dinyalakan	Sesuai
18. Menekan <i>Button Drain Water</i>	19. Pompa menguras dinyalakan	Sesuai

Hasil dari pengujian fungsionalitas laman *Dashboard* pada tabel 4.5 menunjukkan bahwa ketika user mengakses halaman *Dashboard* reaksi dari sistem yaitu menampilkan data sensor terbaru dengan melakukan inisiasi data dalam *database*. Dan Ketika *Button fill water* dihidupkan maka pompa mengisi menyala.

Kemudian jika *button drain water* dinyalakan maka pompa menguras menyala. Berdasarkan hal tersebut dapat diperoleh hasil pengujian fungsionalitas sesuai.

4. Pengujian Monitor Level

Tabel 4.11 Hasil Pengujian Fungsionalitas Monitor Level

Aksi Aktor	Reaksi Sistem	Kesimpulan
20. Memulai halaman <i>Monitor Level</i>	21. Memanggil file <i>datalevel.java</i> untuk menampilkan menu utama aplikasi yang menampilkan data dan grafik perkembangan level air setiap 3 jam. 22. Melakukan inisiasi hubungan ke <i>database</i>	Sesuai
23. Menekan <i>Button Count</i>	24. Menghitung nilai rata-rata data pada halaman <i>Horly</i> dan dikirim ke data <i>Daily</i> .	Sesuai
25. Menekan menu <i>Daily</i>	26. menampilkan data dan grafik perkembangan level air setiap hari. 27. Melakukan inisiasi hubungan ke <i>database</i>	Sesuai

Hasil dari pengujian fungsionalitas laman monitor level pada tabel 4.6 menunjukkan bahwa ketika user mengakses halaman monitor level reaksi dari sistem yaitu menampilkan data dan grafik sensor level secara lengkap perjam dan perhari dengan melakukan inisiasi data dalam *database*. Dan Ketika *Button Count* ditekan

maka rekasi sistem yaitu menghitung nilai rata-rata data masuk pada menu *horly* dan mengirimnya ke data *Daily*.

5. Pengujian Monitor Keruh

Tabel 4.12 Hasil Pengujian Fungsionalitas Monitor Level

Aksi Aktor	Reaksi Sistem	Kesimpulan
28. Memulai halaman monitor	29. Memanggil file <i>datakeruh.java</i> untuk menampilkan menu utama aplikasi yang menampilkan data dan grafik perkembangan keruh air setiap 3 jam. 30. Melakukan inisiasi hubungan ke <i>database</i>	Sesuai
31. Menekan <i>Button Count</i> pada menu <i>horly</i>	32. Menghitung nilai rata-rata data pada halaman <i>Horly</i> dan dikirim ke data <i>Daily</i> .	Sesuai
33. Menekan menu <i>Daily</i>	34. menampilkan data dan grafik perkembangan keruh air setiap hari. 35. Melakukan inisiasi hubungan ke <i>database</i>	Sesuai
36. Menekan <i>Button Count</i> pada menu <i>Daily</i>	37. Menghitung nilai rata-rata data pada halaman <i>Daily</i> dan dikirim ke data <i>Weekly</i> .	Sesuai
38. Menekan menu <i>weekly</i>	39. menampilkan data dan grafik perkembangan keruh air setiap 1 minggu sekali. 40. Melakukan inisiasi hubungan ke <i>database</i>	Sesuai

Hasil dari pengujian fungsionalitas laman monitor keruh pada tabel 4.7 menunjukkan bahwa ketika user mengakses halaman monitor keruh reaksi dari sistem yaitu menampilkan data dan grafik sensor kekeruhan secara lengkap perjam, perhari dan perminggu dengan melakukan inisiasi data dalam *database*. Dan Ketika *Button Count* ditekan maka reaksi sistem yaitu menghitung nilai rata-rata data masuk pada menu *horly* dan mengirimnya ke data *daily*. Kemudian didalam menu *daily* ada *button count* juga untuk menghitung nilai rata-rata data masuk pada menu *daily* dan mengirimnya ke data *weekly*, untuk menampilkan data perminggu



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan tentang rancang bangun perangkat lunak sistem monitoring dan kontrol air minum pada kandang burung Puyuh dengan menggunakan *Internet Of Things* berbasis android, dapat disimpulkan bahwa sistem telah berhasil dibangun. Perancangan perangkat lunak dilakukan sedemikian rupa untuk membuat sistem ini. *Arduino UNO* digunakan sebagai mikrokontroler yang menerima input data sensor, yang kemudian data tersebut diolah dan dikirim menggunakan *NodeMcu* ke database *Firebase*, yang kemudian data tersebut ditampilkan ke dalam perangkat lunak berupa informasi kondisi wadah air minum secara *realtime* dan juga berkala. Jadi peternak dapat memonitoring air minum burung puyuh, dan mengetahui seberapa banyak konsumsi air minum burung puyuh dan kualitas air minum burung puyuh.

Dari hasil uji coba monitoring selama 1 minggu yang dilakukan dengan menggunakan 4 ekor burung puyuh sebagai bahan uji coba dan wadah air berukuran 2 liter didapatkan hasil untuk konsumsi air minum burung puyuh setiap harinya yaitu 0.6 liter air atau 600 mililiter. Untuk kekeruhan air minum burung puyuh dalam 1 minggu didapatkan hasil rata-rata 10 Ntu yang artinya masih tergolong jernih.

Uji coba perangkat lunak yang telah dilakukan dapat disimpulkan, bahwa rekasi sistem kontrol dan monitoring secara *realtime* dan berkala berjalan sesuai dengan sistem yang dibuat.

5.2 Saran

Dalam penelitian ini masih banyak hal-hal yang kurang dan harus diperbaiki. Sehingga penulis memiliki saran untuk penelitian selanjutnya sebagai berikut.:

- a. Penambahkan fitur pengaturan durasi pengisian dan pengurasan pada perangkat *android*.
- b. Penambahan fitur mode otomatis manual pada kontrol pompa air minum burung puyuh.
- c. Penambahan fitur notifikasi ketika keadaan air habis atau keruh.
- d. Pada aplikasi ini monitoring air minum burung puyuh ini masih berupa data-data teks yang ditampilkan, sehingga peternak masih belum mengetahui keadaan air minum yang sebenarnya secara visual. Mungkin ke depannya dapat diberikan fitur kamera yang dapat mengetahui secara langsung keadaan air minum burung puyuh.

Hal tersebut diharapkan dapat menjadi perhatian bagi pengembang berikut, dan dapat dikembangkan lagi agar lebih sempurna dari ini.

DAFTAR PUSTAKA

- Fauzi. (2017). Sistem Kontrol Suhu Ruangan Pada Inkubator Anak Ayam Menggunakan Ep Wemos Di Berbasis IoT. *Universita Noratama Surabaya*.
- Setiyani, L. (2018). *Rekayasa Perangkat Lunak*. Karawang: Jatayu Catra Internusa.
- Nazzrudin.(2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Hafsa, T., Ayesha, K., & Junaid, M. (2016). *Internet of Things (IoT): An Overview of Applications and Security Issues Regarding*. *INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING*, VOL. 7, NO. 1.
- Khanifa, F. (2016). *Pembangunan Aplikasi Mobile Android Untuk Memberikan Rute Rekomendasi Angkutan Umum Kota Bandung*. Bandung: Skripsi.Elib.Unikom.Fakultas Teknik Dan Ilmu Komputer.
- Lee, W.-M. (2012). *Beginning Android 4 Application Development*. indianapolis: Jhon Wily & Sons, Inc.
- Mahali, M. I. (2016). Smart Door Locks Based on *Internet of Things* Concept With *Mobile Backend As A Service*. *Jurnal Electronics, Informatics Vocational Education*, vol 1 no 3.
- Masriwilaga, A. A., Jabar, T. A., Subagja, A., & Septiana, S. (2019). Monitoring *Sistem for Broiler Chicken Farms Based on Internet of Things* . *TELEKON-TRAN*, VOL. 7, NO. 1.

Mindriawan, Z., Arimbawa, I. W., & Wijaya, I. G. (2018). Implementasi *Internet of Things* Pada Sistem Monitoring Suhu dan Kontrol Air Pada Kandang Burung Puyuh Petelur dengan Menggunakan Protokol MQTT (. *Dept Informatics Engineering, Mataram University* .

Prayudita, R. (2018, July 15). *Metode Waterfall*. Retrieved from materikuliahif-unpas: <http://materikuliahif-unpas.blogspot.com/2018/07/metode-waterfall.html>

Sudrajat, & Yuniawan, A. (2018). Faktor-faktor yang Berpengaruh Terhadap Pendapatan Usaha Ternak Ayam Sentul Di Kabupaten Ciamis. *Jurnal Pemikiran Masyarakat Ilmiah Berwawasan Agribisnis*, vol 4 : 70-83.

Technology, S. S. (2019, September 18). *Mengenal Apa itu Internet of Things (IoT) : Defenisi, Manfaat, Tujuan dan Cara Kerja*. Retrieved from idcloudhost: <https://idcloudhost.com/mengenal-apa-itu-internet-of-things-iot-defenisi-manfaat-tujuan-dan-cara-kerja/>.

Wangefela, L., Santa, N. M., Salendu, A. H., & Kalangi, L. S. (2018). Analisis Margin Of Safety Usaha Peternakan Puyuh "MERRYLAND" Di Desa Kali Kecamatan Pineleg Kabupaten Minahasa. *Jurnal Zootek*, Vol. 38 No. 1 : 183-19.

Yahya. (2018, Agustus 3). *Internet of Things Panduan Lengkap*. Retrieved from dewaweb: <https://www.dewaweb.com/blog/internet-of-things/>.