

**SISTEM KENDALI OTOMATISASI PEMBERIAN
MAKAN DAN MINUM PADA HEWAN PELIHARAAN
BERBASIS INTERNET OF THINGS**

**MUHAMMAD ASWIN SIGIT
H1051191051**

SKRIPSI



**PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS TANJUNGPURA PONTIANAK
2024**

**SISTEM KENDALI OTOMATISASI PEMBERIAN
MAKAN DAN MINUM PADA HEWAN PELIHARAAN
BERBASIS INTERNET OF THINGS**

**MUHAMMAD ASWIN SIGIT
H1051191051**

Skripsi

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer pada Program Studi Rekayasa Sistem Komputer



**PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS TANJUNGPURA PONTIANAK
2024**

LEMBAR PENGESAHAN

Judul Tugas Akhir : Sistem Kendali Otomatisasi Pemberian Makan Dan Minum Pada Hewan Peliharaan Berbasis *Internet Of Things*

Nama Mahasiswa : Muhammad Aswin Sigit
NIM : H1051191051
Jurusan/Program Studi : Rekayasa Sistem Komputer
Tanggal Lulus :
SK. Pembimbing : No. /Tanggal
SK. Penguji : No. /Tanggal

Dosen Pembimbing

Pembimbing I

Pembimbing II

Suhardi , S.T., M.Eng.
NIP. 198606182020121006

Tedy Rismawan, S.Kom., M.Cs
NIP. 198609222014041002

Ketua Penguji

Anggota Pengudi

Drs. Cucu Suhery, M.A.
NIP. 196108291989031002

Irma Nirmala, S.T., M.T.
NIP. 198404052019032015

Pimpinan Sidang
(merangkap anggota penguji)

Sekretaris Sidang
(merangkap anggota penguji)

Nama
NIP.

Nama
NIP

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Tanjungpura

Dr. Gusrizal, S.Si., M.Si
NIP. 19710802200031001

PERNYATAAN INTEGRITAS AKADEMIK

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Aswin Sigit
NIM : H1051191051
Program Studi/ Jurusan : Rekayasa Sistem Komputer
Fakultas : Matematika dan Ilmu Pengetahuan Alam

dengan ini menyatakan bahwa dokumen ilmiah Tugas Akhir yang disajikan ini tidak mengandung unsur pelanggaran integritas akademik sesuai Peraturan Menteri Pendidikan, Kebudayaan, Riset, Dan Teknologi Republik Indonesia Nomor 39 Tahun 2021. Apabila di kemudian hari dokumen ilmiah Tugas Akhir ini mengandung unsur pelanggaran integritas akademik sesuai ketentuan perundangan tersebut, maka saya bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Demikian pernyataan ini untuk digunakan sebagaimana mestinya.

Pontianak, 2024

Materai 10.000,-
Muhammad Aswin Sigit
H1051191051

**Sistem Kendali Otomatisasi Pemberian Makan Dan Minum Pada Hewan
Peliharaan Berbasis Internet Of Things**

Abstrak

Kucing

Kata Kunci : Kucing, Arduino, *Internet of Things*, *Mobile Application*

Internet of Things-Based Automated Control System for Feeding and Drinking Pets

Abstract

The cat is

Keywords: *Cat, Arduino, Internet of Things, Mobile Application*

PRAKATA

Sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Sistem Kendali Otomatisasi Pemberian Makan Dan Minum Pada Hewan Peliharaan Berbasis Internet Of Things**” sebagai salah satu syarat bagi penulis untuk memperoleh gelar Sarjana Komputer (S.Kom) di Jurusan Rekayasa Sistem Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tanjungpura Pontianak.

Penulis menyadari bahwa semua hasil yang diperoleh tidak lepas dari dukungan dan bantuan yang diberikan berbagai pihak seperti memberi bimbingan, saran serta data untuk menyelesaikan tugas akhir ini. Dalam penulisan tugas akhir ini penulis juga memperoleh banyak motivasi dan dorongan baik secara moril dan materil. Oleh karena itu, penulis ingin mengucapkan banyak terima kasih dengan hormat kepada:

1. Bapak Dr.Gusrizal, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tanjungpura.
2. Bapak Ikhwan Ruslianto., S.Kom., M.Cs selaku Ketua Jurusan Rekayasa Sistem Komputer Fakultas Matematika dan Ilmu Pengatahuan Alam Universitas Tanjungpura.
3. Bapak Suhardi , S.T., M.Eng. selaku dosen pembimbing pertama yang meluangkan banyak waktu dan pikiran untuk memberikan bimbingan, arahan dan saran dalam proses penyusunan skripsi ini.
4. Bapak Tedy Rismawan, S.Kom., M.Cs selaku dosen pembimbing kedua yang meluangkan banyak waktu dan pikiran untuk memberikan dukungan, arahan, motivasi, bimbingan dan saran dalam proses penyusunan skripsi ini.
5. Bapak Drs. Cucu Suhery, M.A selaku dosen penguji pertama yang meluangkan waktu dan pikiran untuk memberikan saran dan masukan dalam penyusunan skripsi ini.

6. Ibu Irma Nirmala, S.T., M.T. selaku dosen penguji kedua yang meluangkan waktu dan pikiran untuk memberikan saran dan masukan dalam penyusunan skripsi ini.
7. Bapak dan Ibu dosen serta staf administrasi Program Studi Rekayasa Sistem Komputer.
8. Ibu Widhasa Putri di Desa Arang Limbung, Kabupaten Kubu Raya, Kalimantan Barat sebagai pemilik hewan peliharann kucing yang meluangkan waktu dan lokasi sebagai lokasi pengumpulan data dan pengujian alat.
9. Kedua orangtua saya tercinta, Bapak Heri Nurhan dan Ibu Sri Ayunda yang tidak lelah memberikan semangat, doa, dukungan, dan motivasi yang sungguh luar biasa kepada penulis sejak awal perkuliahan hingga pada saat ini. Penulis menyadari bahwa penyusunan skripsi ini memiliki banyak kekurangan dan masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun akan diterima oleh penulis dengan senang hati. Semoga tugas akhir ini dapat bermanfaat bagi penulis maupun pembaca.

Pontianak, 2024

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	2
PERNYATAAN INTEGRITAS AKADEMIK.....	3
Abstrak	4
Abstract	5
PRAKATA.....	6
DAFTAR ISI.....	8
DAFTAR GAMBAR	11
DAFTAR TABEL.....	xiii
KODE PROGRAM.....	xiv
BAB 1 PENDAHULUAN	2
1.1. Latar Belakang.....	2
1.2. Perumusan Masalah.....	6
1.3. Batasan Masalah	6
1.4. Tujuan Penelitian	7
1.5. Manfaat Penelitian	7
1.6. Sistematika Penulisan	7
BAB II LANDASAN TEORI	9
2.1. Perbandingan Dengan Penelitian Terdahulu.....	9
2.2. Kucing	11
2.3. Makanan dan Minuman Kucing	15
2.4. <i>Mobile Application</i>	16
2.5. <i>Internet of Things (IoT)</i>	18
2.6. Arduino Uno R3 <i>Built-in IOT WiFi ESP8266</i>	18
2.7. Sensor <i>Load Cell</i>	20
2.8. Sensor Ultrasonic	21
2.9. Real Time Clock (RTC)	21
2.10. Liquid Crystal Display (LCD 20x4) I2C.....	22
2.11. Motor Servo.....	23
2.12. <i>Mini Pump Water</i>	24
2.13. Relay.....	24
2.14. Stepdown (Module XL4015).....	25
2.15. <i>Flutter</i>	25
2.16. <i>Firebase Real-Time Database</i>	26
2.17. Arduino IDE.....	27
2.18. Pertumbuhan Kucing	28
BAB III METODE PENELITIAN.....	29
3.1. Studi Literatur.....	30
3.2. Metode Pengumpulan Data.....	30
3.3. Analisis Kebutuhan.....	30
3.3.1 Kebutuhan Perangkat Keras.....	30

3.3.2	Kebutuhan Perangkat Lunak.....	31
3.4.	Perancangan Sistem.....	32
3.4.1	Perancangan Perangkat Keras.....	32
3.4.2	Perancangan Perangkat Lunak.....	32
3.5.	Implementasi.....	32
3.5.1	Implementasi Perangkat Keras	32
3.5.2	Implementasi Perangkat Lunak	33
3.6.	Pengujian	33
3.7.	Kesimpulan dan Saran	34
BAB IV PERANCANGAN		35
4.1.	Deskripsi Sistem	35
4.2.	Perancangan Arsitektur Sistem.....	36
4.3.	Perancangan Perangkat Keras.....	41
4.3.1	Perancangan Sistem Penjadwalan Real Time Clock (RTC)	41
4.3.2	Perancangan Sistem Pengukuran Berat Wadah Makan	42
4.3.3	Perancangan Sistem Pengukuran Tabung Air Minum	43
4.3.4	Perancangan Sistem Kendali Mini Pump Water	44
4.3.5	Perancangan Sistem Kendali Servo.....	46
4.3.7	Perancangan Sistem Pengukuran Wadah Air Minum	47
4.3.7	Perancangan Sistem Tampilan Liquid Crystal Display (LCD) 20x4-I2C Pembacaan Sensor	48
4.3.8	Perancangan Sistem Daya Pada Keseluruhan Sistem	49
4.3.9	Perancangan Keseluruhan Sistem	51
4.4.	Perancangan Perangkat Lunak.....	52
4.4.1	Perancangan perangkat Lunak pada Arduino Uno R3 Built-in IoT WiFi ESP8266.....	55
4.4.2	Perancangan Antarmuka Mobile Application	57
4.5.	Perancangan Basis Data.....	65
4.6.	Perancangan Pengujian <i>Black Box</i>	70
BAB V.....		77
5.1	Implementasi Sistem.....	77
5.2	Implementasi Perangkat Keras	77
5.2.1	Implementasi Sistem Penjadwalan <i>Real Time Clock</i>	77
5.2.2	Implementasi Sistem Pengukuran Berat Wadah Makan	78
5.2.3	Implementasi Sistem Pengukuran Volume Tabung Air Minum	79
5.2.4	Implementasi Kendali Mini Pump Water.....	79
5.2.5	Implementasi Kendali Servo	80
5.2.6	Implementasi Sistem Pengukuran Volume Wadah Air Minum	81
5.2.7	Implementasi Tampilan LCD 20X4 Pembacaan Sensor	82
5.2.8	Implementasi Keseluruhan Sistem	82
5.3	Implementasi Perangkat Lunak	84
5.3.1	Implementasi Antarmuka Halaman <i>Login</i>	84
5.3.2	Implementasi Antarmuka Halaman Lupa <i>Password</i>	85
5.3.3	Implementasi Antarmuka Halaman <i>Home</i>	86

5.3.4	Implementasi Antarmuka Form <i>Feeder</i>	87
5.3.5	Implementasi Antarmuka Halaman <i>Statistic</i>	88
5.3.6	Implementasi Antarmuka Halaman Detail Data	89
5.3.7	Implementasi Antarmuka Halaman Penjadwalan	90
5.3.8	Implementasi Antarmuka Halaman Data Penjadwalan.....	91
5.3.9	Implementasi Antarmuka Halaman Data Feeder	92
5.3.10	Implementasi Antarmuka Halaman <i>Settings</i>	93
5.4	Kode Program.....	98
5.4.1	Kode Program Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266	99
5.4.2	Kode Program Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266	120
5.5	Pengujian Sistem	131
5.5.1	Pengujian Akurasi Penjadwalan Real Time Clock.....	132
5.5.2	Pengujian Pembacaan dan Akurasi Pengukuran Berat Wadah Makan	133
5.5.3	Pengujian Pembacaan dan Akurasi Pengukuran Volume Tabung Air Minum	134
5.5.4	Pengujian Kendali Mini Pump Water	136
5.5.5	Pengujian Kendali Servo	136
5.5.6	Pengujian Pembacaan dan Akurasi Pengukuran Volume Wadah Air Minum	137
5.5.7	Pengujian Keseluruhan Sistem.....	138
5.5.8	Pengujian Black Box	140
5.6	Pembahasan	149
	BAB VI PENUTUP	150
6.1.	Kesimpulan	150
6.2.	Saran	150
	DAFTAR PUSTAKA	151
	LAMPIRAN	152

DAFTAR GAMBAR

Gambar 2.1 Kucing	12
Gambar 2.2 Aturan Porsi Makan Kucing.....	13
Gambar 2.3 Feline BCS Chart	14
Gambar 2.4 Makanan Kucing	15
Gambar 2.5 Mobile Application	17
Gambar 2.6 Internet of Things (IoT).....	18
Gambar 2.7 Arduino Uno R3 Built-in IOT WiFi ESP8266.....	19
Gambar 2.8 Load Cell.....	20
Gambar 2.9 Modul HX711	21
Gambar 2.10 Sensor Ultrasonic HC-SR04.....	21
Gambar 2.11 Real Time Clock (RTC) DS3231	22
Gambar 2.12 Liquid Crystal Display (LCD) 20x4 I2C.....	23
Gambar 2.13 Towerpro Motor Servo MG996R.....	23
Gambar 2.14 Mini Pump Water	24
Gambar 2.15 Modul Relay.....	25
Gambar 2.16 Stepdown (Modul XL4015)	25
Gambar 2.17 Logo Flutter.....	26
Gambar 2.18 Logo Firebase Real-Time Database	27
Gambar 2.19 Arduino IDE.....	28
Gambar 4.1 Rancangan Sistem Perangkat Otomatisasi Makan dan Minum Kucing	36
Gambar 4.2 Rancangan Arsitektur Sistem	37
Gambar 4.3 Design 3D Automatic Pet Feeder.....	39
Gambar 4.4 Rangkaian Sistem Makan Kucing	40
Gambar 4.5 Rangkaian Sistem Minum Kucing	40
Gambar 4.6 Rancangan Sistem Penjadwalan.....	41
Gambar 4.7 Rancangan Sistem Timbangan Tabung dan Wadah Makan.....	42
Gambar 4.8 Rangkaian Sistem Ketinggian Tabung Air.....	44
Gambar 4.9 Rancangan Sistem Kendali Mini Pump Water.....	45
Gambar 4.10 Rancangan Sistem Kendali Servo	46
Gambar 4.11 Rangkaian Sistem Ketinggian Wadah Air.....	47
Gambar 4.12 Perancangan Sistem Display Data.....	49
Gambar 4.13 Perancangan Sistem Daya Pada Keseluruhan Sistem	50
Gambar 4.14 Perancangan Keseluruhan Sistem	51
Gambar 4.15 Diagram Alir Keseluruhan	54
Gambar 4.16 Diagram Alir Arduino Uno R3 dengan built-in IoT WiFi ESP8266 Untuk Pengiriman Data Sensor	56
Gambar 4.17 Halaman Login.....	57
Gambar 4.18 Halaman Lupa Password.....	58
Gambar 4.19 Halaman Home.....	58
Gambar 4.20 a) Form Feeder pagi, b) Form Feeder sore	59
Gambar 4.21 Halaman statistic	60
Gambar 4.22 a) Chart Penjadwalan dan Diagram Monitoring, b) Chart Feeder.....	60
Gambar 4.23 a) Form Tambah Jadwal, b) Form Edit Jadwal, c) Detail Jadwal	61
Gambar 4.24 a) Halaman Data Jadwal Pagi, b) Halaman Data Jadwal Sore	62
Gambar 4.25 a) Halaman Data Feeder Pagi, b) Halaman Data Feeder Sore	62
Gambar 4.26 Halaman Setting	63
Gambar 4.27 Halaman Update Profile	63
Gambar 4.28 a) Status alat, b) Form Tambah status alat, c) Form Edit status alat	64
Gambar 4.29 Halaman Ubah Password	65

Gambar 5.1 Implementasi Sistem Penjadwalan Real Time Clock

78

Gambar 5.2 Implementasi Sistem Pengukuran Berat Wadah Makan	78
Gambar 5.3 Implementasi Sistem Pengukuran Volume Tabung Air Minum	79
Gambar 5.4 Implementasi Kendali Mini Pump Water	80
Gambar 5.5 Implementasi Kendali Servo	81
Gambar 5.6 Implementasi Sistem Pengukuran Volume Wadah Air Minum	82
Gambar 5.7 Implementasi Tampilan LCD 20X4 Pembacaan Sensor	82
Gambar 5.8 Implementasi Keseluruhan Sistem	83
Gambar 5.9 Halaman Login	85
Gambar 5.10 Halaman Lupa Password	86
Gambar 5.11 Halaman Home	87
Gambar 5.12 a) Form Feeder pagi, b) Form Feeder sore	88
Gambar 5.13 Halaman Statistic	89
Gambar 5.14 Halaman Detail Data	90
Gambar 5.15 a) Form tambah jadwal, b) Form edit jadwal, c) Detail jadwal	91
Gambar 5.16 a) Halaman Data Jadwal Pagi, b) Halaman Data Jadwal Sore	92
Gambar 5.17 a) Halaman data feeder pagi, b) Halaman data feeder sore	93
Gambar 5.18 Halaman Settings	94
Gambar 5.19 Halaman Update Profile	95
Gambar 5.20 a) Halaman tambah status alat, b) Halaman edit status alat, c) Halaman status alat	96
Gambar 5.21 Halaman Ubah Password	97
Gambar 5.22 Halaman History Feeder	98
Gambar 5.24 Pengujian Akurasi Penjadwalan Real Time Clock	132
Gambar 5.25 Pengujian Pembacaan dan Akurasi Pengukuran Berat Wadah Makan	133
Gambar 5.26 Pengujian Pembacaan dan Akurasi Pengukuran Volume Tabung Air Minum	135
Gambar 5.27 Pengujian Kendali Mini Pump Water	136
Gambar 5.28 Pengujian Kendali Motor Servo	137
Gambar 5.29 Pengujian Pembacaan dan Akurasi Pengukuran Volume Wadah Air Minum	137
Gambar 5.30 Implementasi Pengujian Keseluruhan Sistem	139

DAFTAR TABEL

Tabel 2.1 Perbandingan dengan penelitian terdahulu 9

Tabel 4.1 Koneksi Pin RTC dengan Arduino Uno R3 Built-in IoT WiFi ESP8266.....	42
Tabel 4.2 Koneksi Pin Modul HX711 dengan Arduino Uno R3 Built-in IoT WiFi ESP8266.....	43
Tabel 4.3 Pin Modul HX711 dan Load Cell	43
Tabel 4.4 Koneksi Pin Ultrasonic Tabung dengan Arduino Uno R3 Built-in IoT WiFi ESP8266	44
Tabel 4. 5 Koneksi Pin Modul Relay dengan Arduino Uno R3 Built-in IoT WiFi ESP8266	45
Tabel 4.6 Koneksi Pin Servo dengan Arduino Uno R3 Built-in IoT WiFi ESP8266	47
Tabel 4.7 Koneksi Pin Ultrasonic Wadah dengan Arduino Uno R3 Built-in IoT WiFi ESP8266	48
Tabel 4.8 Koneksi Pin LCD 20x4 I2C dengan Arduino Uno R3 Built-in IoT WiFi ESP8266	49
Tabel 4.9 Koneksi Daya Pada Keseluruhan Sistem.....	51
Tabel 4.10 Tabel <i>Users Data</i>	65
Tabel 4.11 Tabel <i>Users Profile</i>	66
Tabel 4.12 Tabel iot	67
Tabel 4.13 Tabel Penjadwalan	68
Tabel 4.14 Tabel manual.....	69
Tabel 4.15 Tabel Status Alat.....	70
Tabel 4. 16 Tabel Perancangan Pengujian Black Box	70

Tabel 5.1 Tabel Hasil Pengujian Akurasi Penjadwalan Real Time Clock..... 132

Tabel 5.2 Tabel Hasil Pengujian Pembacaan dan Akurasi Pengukuran Berat Wadah Makan 133

Tabel 5.3 Tabel Hasil Pengujian Pembacaan dan Akurasi Pengukuran Volume Tabung Air Minum	135
---	-----

Tabel 5.4 Tabel Hasil Pengujian Kendali Mini Pump Water..... 136

Tabel 5.5 Tabel Hasil Pengujian Kendali Motor Servo 137 |

Tabel 5.6 Tabel Hasil Pengujian Pembacaan dan Akurasi Pengukuran Volume Wadah Air Minum	137
--	-----

Tabel 5.7 Tabel Pengujian Keseluruhan Sistem 139 |

Tabel 5.8 Pengujian Black Box 140 |

KODE PROGRAM

Kode Program 5.1 Kode Program Transmitter pada Arduino Uno R3 Built-in IoT WiFi ESP8266	99
Kode Program 5.2 Void Setup Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266	101
Kode Program 5.3 Void Loop Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266	103
Kode Program 5.4 Inisialisasi Liquid Crystal Display	103
Kode Program 5.5 Inisialisasi Real Time Clock	105
Kode Program 5.6 Kode Program Menghitung Berat Wadah pada Makanan Kucing	106
Kode Program 5.7 Kode Program Data Sensor Ultrasonic	107
Kode Program 5.8 Kalibrasi volume air pada ultrasonic	108
Kode Program 5.9 Kode Program Pembacaan Data Sensor	109
Kode Program 5.10 Kode Program Menghidupkan Servo	109
Kode Program 5.11 Kode Program Menghidupkan Mini Pump Water.....	110
Kode Program 5.12 Kode Program Notifikasi Monitoring	111
Kode Program 5.13 Kode Program untuk Monitoring Makanan dan Minuman Kucing	113
Kode Program 5.14 Kode Program Penjadwalan Makan dan Minum Kucing.....	115
Kode Program 5.15 Kode Program Menampilkan Data pada LCD	116
Kode Program 5.16 Kode Program Mengirimkan Data pada Receiver Arduino Uno R3 Built-In IoT WiFi ESP8266	118
Kode Program 5.17 Kode Program Penerimaan Data dari Receiver pada Transmitter Arduino UNO R3 Built-In IoT WiFi ESP8266	119
Kode Program 5.18 Kode Program Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266	121
Kode Program 5.19 Void Setup Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266.	122
Kode Program 5.20 Void Loop Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266 .	123
Kode Program 5.21 Inisialisasi Jaringan WiFi	124
Kode Program 5.22 Inisialisasi Firebase Real-Time Database.....	125
Kode Program 5.23 Kode Program Penerimaan data dari Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266.....	127
Kode Program 5.24 Kode Pengiriman data pada Receiver ke Firebase Real-Time Database....	128
Kode Program 5.25 Pengiriman Data pada Receiver ke Transmitter	130

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Hewan peliharaan adalah hewan yang dipelihara sebagai teman sehari-hari oleh manusia, salah satu hewan yang sering dipelihara dan digemari oleh kebanyakan orang adalah kucing karena kucing memiliki karakteristik sifat yang manja dan memiliki bulu yang indah. Sama seperti hewan peliharaan lainnya, kucing pun membutuhkan makanan dan minuman yang harus diberikan oleh pemiliknya sesuai dengan porsi kucing tersebut. Semakin padatnya aktifitas masyarakat pada saat ini membuat masyarakat banyak mencari kemudahan dalam berbagai hal karena terbatasnya waktu yang dimiliki. Bagi orang – orang dengan aktifitas padat tentunya kegiatan memelihara hewan peliharaan seperti kucing menjadi sangat susah karena terbatasnya waktu dan padatnya aktifitas, padahal hewan peliharaan yang tidak mendapatkan makanan dan minuman yang cukup akan menjadi kurus, liar, dehidrasi dan bahkan sakit. Idealnya, jadwal pemberian makan dan minum kucing sebanyak 2 kali sehari dikategorikan sangat baik (Hartuti, 2013). Hal ini sesuai dengan pernyataan American Society Prevention of Cruelty to Animals ASPCA (2013), bahwa pemberian makan dan minum kucing 2 kali sehari berfungsi untuk mengurangi rasa lapar dan dehidrasi diantara waktu makan dan minum. Serta meminimalisasi masalah perilaku terkait kesehatan kucing. Waktu yang tepat dalam pemberian makan dan minum sebaiknya pada waktu yang tepat yaitu, pagi hari pukul 08.00 dan sore hari pukul 17.00.

Kucing seharusnya dapat mengatur asupan makanan sesuai dengan kebutuhannya, namun 25% sampai 40% dari kucing yang telah diamati oleh dokter hewan menunjukkan tanda kelebihan berat badan. Kucing yang sering beraktifitas di luar rumah membutuhkan sekitar 300 kkal/hari, sementara untuk kucing di dalam ruangan hanya membutuhkan sebanyak 200 kkal/hari atau 30% lebih sedikit dari porsi kucing yang sering beraktifitas di luar rumah (Susetyo, 2004). Jika kucing sering mengonsumsi makanan basah, kita mungkin memerhatikan bahwa kucing

tidak minum banyak air. Itu karena kucing mengonsumsi air saat makan. Makanan basah terbuat dari air hingga 80 persen. Kucing yang mengonsumsi makanan kering di sisi lain, tidak akan minum banyak air dari makanannya. Jadi, kucing perlu terhidrasi dengan minum dari piring atau mangkuk air. Pastikan kucing peliharaan kita mendapatkan cukup air sebagai bagian dari diet hariannya. Jika tidak mendapatkan cukup air, kucing mungkin akan mengalami dehidrasi. Oleh karena itu, pemelihara harus waspada dalam melakukan pemberian asupan makanan pada kucing tersebut. Ketika kucing mengalami kelebihan berat badan, maka kucing akan mengalami gangguan pada pencernaan dan saluran kemih karena beban berat tubuh yang berlebihan pada tulang belakang dan sendi, kucing pun akan mengalami dystocial atau kesulitan pada saat melakukan persalinan (pada kucing betina), heat stroke yang disebabkan karena menurunnya kemampuan untuk mengontrol suhu panas yang masuk pada tubuh, turunnya daya imun pada kucing dan akan menjadi sangat rentan terhadap penyakit ringan seperti flu.

Dehidrasi pada kucing bisa terjadi ketika hewan tersebut kehilangan cairan secara berlebihan dari tubuhnya tanpa diganti. Saat hal ini terjadi, kucing bukan hanya kehilangan air, tetapi juga beberapa mineral penting dalam tubuh seperti klorida, natrium, dan kalium. Oleh sebab itu, mendapatkan kecukupan air sangat penting untuk kesehatan kucing sekaligus mengganti cairan yang hilang melalui urin dan feses. Air juga diperlukan untuk sirkulasi, pencernaan dan pembuangan kotoran kucing. Untuk membantu kucing peliharaan kita mempertahankan hidrasi yang sehat, pastikan kucing selalu dapat mengakses air bersih di mangkuk yang bersih setiap hari. Kucing membutuhkan jumlah air harian sekitar 1 hingga 130 mililiter air per 2,2 kg berat tubuhnya. Misalnya, jika kita memiliki kucing peliharaan seberat 4,5 kg, maka kucing kita harus minum antara 200 hingga 260 mililiter air setiap hari (Suwed, 2011). Jika dehidrasi pada kucing tidak segera diatasi, maka kondisi ini dapat menyebabkan masalah kesehatan yang lebih parah. Dehidrasi pada kucing biasanya disebabkan oleh kucing tidak minum cukup air atau kehilangan air secara berlebihan.

Dari masalah tersebut dibutuhkan teknologi untuk memecahkan masalah itu. Salah satu cara untuk memecahkan permasalahan tersebut, kita dapat memanfaatkan internet sebagai media untuk membantu mengontrol dan memonitoring jadwal makan dan minum pada kucing, salah satunya pemanfaatan teknologi Internet of Things (IoT). Segala sesuatu yang terhubung ke internet termasuk dalam teknologi yang berbasis IoT. Salah satu teknologi yang bisa digunakan yaitu wireless atau tanpa kabel contohnya adalah menggunakan mikrokontroller arduino uno. Selain memanfaatkan mikrokontroller Arduino uno sebagai pengendali utama dalam sistem alat kendali, media yang bisa digunakan pada teknologi yang berbasus IoT yaitu mobile application yang mana hal tersebut bisa langsung diakses oleh pengguna melalui smartphone.

Penelitian sebelumnya dilakukan berjudul “Rancang Bangun Sistem Pengisian Pakan Dan Minum Burung Otomatis Berbasis Arduino Uno” (Tubagus Helmi Nashiruddin Sandi, 2020). Pada penelitian ini sensor yang digunakan yaitu infrared, water level dan pembuatan sistem menggunakan mikrokontroller Arduino Uno. Hasil dari perancangan alat pemberi pakan burung otomatis, sensor infrared menyala maka akan mengisi pakan berupa millet secara otomatis ke dalam wadah pakan yang mana membutuhkan waktu yang sedikit berbeda – beda yaitu diantara 19,7-22,3 detik, dan ketika water level sensor menyala maka akan mengisi air minum secara otomatis ke dalam wadah minum membutuhkan waktu yang sedikit berbeda – beda yaitu diantara 08,4-11,1 detik.

Penelitian lain yang dilakukan sebelumnya yaitu “Pengembangan Prototipe Sistem Otomasi Alat Pemberi Makan Ikan Terjadwal Pada Aquarium Berbasis Arduino Uno” (Dipo Ahmad Harel, 2018). Komponen utama yang digunakan pada penelitian ini adalah Arduino Uno sebagai mikrokontroler pembangun sistem otomasi dan komponen pendukung adalah servo yang berfungsi sebagai alat penutup dan pembuka menuangkan makanan kedalam akuarium, komponen tambahan yang menjadi keunikan penelitian ini adalah sensor kelembaban yang membaca suhu dan kelembaban pada ruang makanan dan selanjutnya mengakibatkan berkurangnya kesegaran makanan ikan tersebut, hasil suhu dan kelembaban akan ditampilkan pada layar LCD, sensor photodiode akan mendeteksi ruang tempat makanan bila masih terisi atau kosong.

Penelitian selanjutnya yang berkaitan dengan yang berkaitan dengan system kendali otomatisasi pada hewan peliharaan yaitu “Rancang Bangun Alat Pemberi Pakan Kucing Otomatis Menggunakan Mikrokontroler Nodemcu Berbasis Internet Of Things (Iot)” (Regar Devitasari, 2020). Pada penelitian menggunakan mikrokontroler NodeMCU yang terintegrasi dengan jaringan internet, dan alat otomatisasi ini dapat memberikan pakan secara real time dengan memanfaatkan komponen RTC yang dapat mengatur waktu kapan pakan akan diberikan. Pengguna juga akan mendapatkan notifikasi langsung melalui foxpath tanpa harus membuka laman web untuk mengetahui apakah pakan sudah selesai diberikan atau belum. Setelah melakukan berbagai pengujian, hasil rata-rata persentase keberhasilan keseluruhan sistem adalah 68%.

Berdasarkan permasalahan tersebut penulis merancang suatu sistem pengontrolan makan dan minum kucing secara otomatis dan terjadwal dengan menggunakan *Real Time Clock* (RTC) berfungsi untuk mengatur jam makan dan minum kucing, pemberian makan dan minum pada kucing dilakukan jam 08.00 wib dan 17.00 wib untuk pagi dan sore hari. Sensor Load Cell berfungsi untuk mengukur berat makanan pada wadah makanan dan sensor Ultrasonic berfungsi sebagai mengukur ketinggian air pada wadah minuman kucing. Perancangan alat makan dan minum otomatis pada kucing ini terjadwal agar kucing tetap bisa mendapatkan asupan makanan dan minuman meskipun pemilik kucing tersebut sedang melakukan aktifitas, penggunaan Arduino Uno ini digunakan sebagai alat pengontrol utama yang dapat diprogram untuk kerja alat sehingga motor Servo membuka wadah makanan dan mini pump dapat menyalurkan air ke wadah minuman secara otomatis, sesuai dengan waktu yang telah ditetapkan. Dengan menggunakan alat ini maka kucing akan tetap mendapatkan asupan makanan dan minuman yang cukup ketika pemelihara kucing tidak dapat menyiapkan makanan dan minuman untuk hewan peliharaannya. Setelah servo dan mini pump water memproses, kemudian data akan dikirimkan ke database dan ditampilkan di Liquid Crystal Display (LCD) dan mobile application yang dapat diakses oleh pengguna melalui smartphone.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana mengontrol ketersediaan stok pada wadah makanan dan minuman kucing yang terjadwal menggunakan mobile application yang dapat diakses oleh pengguna melalui smartphone?
2. Bagaimana pengaruh penerapan system terhadap pertumbuhan kucing?

1.3. Batasan Masalah

Agar pembahasan tetap fokus pada pokok masalah serta mempermudah pembahasan dan tujuan penelitian tercapai, maka batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut:

1. Jenis hewan peliharaan yang digunakan untuk objek penelitian ini adalah kucing dan sistem ini dirancang hanya untuk sample 1 (satu) ekor kucing. Jenis kucing untuk segala ras kucing, dengan usia 3-6 bulan dan berat badan 2 kilogram.
2. Perangkat keras yang digunakan sebagai mikrokontroller adalah Arduino Uno
3. Perangkat lunak yang digunakan sebagai kendali system yaitu Arduino IDE.
4. Pengaturan penjadwalan pemberian makan dan minum kucing menggunakan Real Time Clock (RTC).
5. Penelitian ini menggunakan 3 sensor yaitu sensor Ultrasonic sebagai sensor untuk mengukur ketinggian air pada wadah minum, sensor Load Cell untuk mengukur berat wadah makanan dengan kapasitas maksimal 1 Kg dan Real Time Clock (RTC) digunakan untuk penjadwalan waktu pemberian makan dan minum pada kucing.
6. Pengaturan pengeluaran makanan pada wadah menggunakan Motor Servo dengan ketetapan jam.
7. Pengaturan pengeluaran minuman pada wadah menggunakan Mini Pump Water dengan ketetapan jam.
8. Penelitian ini difokuskan tempat makan dan minum kucing yang berukuran 16x18cm.
9. Tabung makanan menampung sebanyak 1 kg makanan dan tabung air menampung sebanyak 1 liter air.

10. Wadah makanan hanya menampung sebanyak 120 gram makanan kering dan wadah minuman hanya dapat menampung sebanyak 300 mililiter air.
11. Sistem ini hanya untuk wilayah/perumahan yang tersedia jaringan internet.
12. Aplikasi yang digunakan untuk system kendali otomatisasi makan dan minum pada kucing yaitu mobile application yang dapat diakses melalui smartphone.
13. Ukuran makanan kucing yang digunakan adalah makanan kering dengan ukuran 1 - 2 cm.

1.4. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Melakukan proses kontrol otomatis ketersediaan stok pada wadah makanan dan minuman kucing yang terjadwal menggunakan mobile application yang dapat diakses oleh pengguna melalui smartphone.
2. Mengetahui pengaruh penerapan system terhadap pertumbuhan kucing.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Dengan adanya system kendali otomatisasi makanan dan minum pada hewan peliharaan khusus nya kucing dapat membantu pemelihara kucing yang saat ini masih terkendala waktu untuk selalu mengontrol jadwal makan dan minum kucing melalui smartphone.
2. Dengan adanya alat ini akan menampilkan data-data dari pembacaan sensor yang digunakan. Data yang ditampilkan dapat diakses menggunakan mobile application yang dapat diakses oleh pemelihara kucing melalui smartphone dan Liquid Crystal Display (LCD) pada tempat makan dan minum kucing.

1.6. Sistematika Penulisan

Sistematika penulisan dari tugas akhir ini dibagi menjadi 6 bab untuk pembahasan dan penyusunannya, sehingga permasalahan dapat dijelaskan dan diuraikan untuk menyelesaikan penelitian. Secara garis besar sistematika penulisan pada penelitian ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini menjelaskan tentang latar belakang yang menjadi alasan dalam melakukan penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab landasan teori ini berisikan teori-teori pedukung yang digunakan dalam pembuatan dan perancangan perangkat dalam penulisan tugas akhir. Dasar teori pada penelitian ini merujuk pada buku, jurnal, artikel ilmiah, wawancara, dan observasi sebagai pendukung bahasan penelitian.

BAB 3 METODE PENELITIAN

Bab ini menjelaskan metode yang digunakan untuk menyelesaikan penelitian tugas akhir ini. Metode penelitian ini menguraikan prosedur pekerjaan yang dijelaskan secara sistematis dalam diagram alir. Metode penelitian yang digunakan meliputi: studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, analisa dan kesimpulan.

BAB 4 PERANCANGAN

Bab perancangan menjelaskan tentang perancangan perangkat pada sistem otomatisasi makan dan minum pada hewan peliharaan berbasis Internet of Things. Tahapan ini meliputi perancangan perangkat keras dan perancangan perangkat lunak.

BAB 5 IMPLEMENTASI, PENGUJIAN DAN PEMBAHASAN

Bab ini menjelaskan implementasi, pengujian dan pembahasan berdasarkan perancangan yang telah dibuat sebelumnya. Implementasi yang dilakukan adalah implementasi antarmuka *mobile application* serta implementasi kode program dalam melakukan otomatisasi makan dan minum pada kucing. Setelah implementasi dilakukan selanjutnya melakukan tahapan pengujian. Pengujian yang dilakukan adalah berupa pengukuran akurasi penggunaan sensor terhadap kesehatan kucing serta hasil kondisi yang ditampilkan dalam *mobile*. Pembahasan adalah untuk membahas secara keseluruhan hasil dari pengujian sistem pemantauan kesehatan kucing.

BAB 6 PENUTUP

Bab ini berisi kesimpulan yang diambil berdasarkan hasil implementasi, pengujian dan pembahasan pada penelitian yang telah dilakukan. Pada bab ini pula berisi tentang saran yang dianggap perlu untuk perbaikan ke depannya tugas akhir ini.

BAB II

LANDASAN TEORI

2.1. Perbandingan Dengan Penelitian Terdahulu

Perbandingan dengan penelitian terdahulu ini dijadikan sebagai bahan acuan supaya tidak terjadi kesamaan dalam melakukan penelitian. Perbandingan ini akan menyajikan hasil-hasil penelitian terdahulu yang berkaitan dengan penelitian ini. Perbandingan dengan penelitian terdahulu dapat dilihat pada tabel 2.1.

Tabel 2.1 Perbandingan dengan penelitian terdahulu

No.	Peneliti	Judul	Persamaan	Perbedaan
1.	Tubagus Helmi Nashiruddin Sandi, dkk (2020)	Rancang Bangun Sistem Pengisian Pakan Dan Minum Burung Otomatis Berbasis Arduino Uno	Input: <i>Arduino Uno, Servo, Relay</i> Output: Pada penelitian ini dan penelitian yang dilakukan sama-sama untuk mengetahui bagaimana pengisian pakan dan minum secara otomatis.	Pada penelitian Tubagus Helmi Nashiruddin Sandi, dkk, menghasilkan kendali otomatis dari data sensor. Pada penelitian ini menghasilkan kendali otomatis dengan memasukan nilai waktu pada sistem.

Tabel 2.1 (Lanjutan)

No.	Peneliti	Judul	Persamaan	Perbedaan
2.	Dipo Ahmad Harel, dkk (2018)	Pengembangan Prototipe Sistem Otomasi Alat Pemberi Makan Ikan Terjadwal Pada Aquarium Berbasis Arduino UNO R3	<i>Input:</i> <i>Arduino Uno, Servo dan Liquid Crystal Display (LCD)</i> <i>Output:</i> Pada penelitian ini dan penelitian yang akan dilakukan sama-sama menampilkan hasil dari data sensor pada LCD	Pada Dipo Ahmad Harel, dkk, menggunakan LCD untuk menampilkan data. Pada penelitian ini digunakan <i>mobile application</i> yang terkoneksi secara online dengan database untuk mengakses dan mengelola data.
3.	Regar Devitasari, dkk (2020)	Rancang Bangun Alat Pemberi Pakan Kucing Otomatis Menggunakan Mikrokontroler Nodemcu Berbasis Internet Of	Input: <i>Real Time Clock (RTC), Servo, Ultrasonic, Liquid Crystal Display</i> Output : Pada penelitian	Pada Regar Devitasari, dkk, menggunakan Web server digunakan sebagai sistem monitoring data pakan dan pengelolaan data pakan dan Output yang dihasilkan berupa pemberian pakan. Pada penelitian ini menggunakan Mobile Application

Tabel 2.1 (Lanjutan)

No.	Peneliti	Judul	Persamaan	Perbedaan
		Things (IoT)	ini dan penelitian yang akan dilakukan sama-sama menggunakan Real Time Clock (RTC) sebagai penjadwalan untuk pakan dan minum.	sebagai Sistem Monitoring data pakan dan data minum. Dan output yang dihasilkan yaitu pakan dan minum.

2.2. Kucing

Kucing (*Felis Silvestris Catus*) adalah salah satu hewan peliharaan terpopuler di dunia. Kucing adalah sejenis karnivora. Kata “kucing” biasanya merujuk kepada kucing yang telah jinak, tetapi bisa juga merujuk kepada “kucing besar” seperti singa, harimau, dan lain-lain. Kucing telah berbaur dengan kehidupan manusia paling tidak sejak tahun 4000 SM. Pada tahun 7500 SM ditemukan kerangka kucing di Pulau Siprus. Orang Mesir Kuno dari 4000 SM telah menggunakan kucing untuk menjauhkan tikus atau hewan pengerat lain dari lumbung yang menyimpan hasil panen (Suwed, 2011). Jumlah jenis kucing ras di seluruh dunia sangatlah banyak.

Setiap ras memiliki ciri khusus, tetapi karena sering terjadi perkawinan silang antar ras banyak kucing yang hanya dikelompokkan dalam jenis bulu panjang dan bulu pendek, ada beberapa jenis dari kucing yang sering dipelihara di Indonesia antara lain:

1. Persia, Kucing ras ini merupakan yang paling banyak dipelihara di Indonesia, berasal dari Iran (Persia). Kucing ini berbulu panjang, berkepala bulat dan berhidung pesek. Kucing berambut panjang ini memiliki sifat kalem, tenang dan mudah akrab dengan manusia. Ukuran tubuh kucing persia tergolong sedang besar. Saat mencapai usia dewasa, berat badan ideal kucing persia jantan mencapai 5- 7 kilogram, sementara persia betina hanya mencapai 3-5 kilogram. Kucing persia membutuhkan makanan yang mengandung omega 3 (EPA dan DHA) serta omega 6 untuk merawat kesehatan kulit dan rambutnya yang panjang.
2. Angora, Kucing ini bernama lengkap Turkish Angora dan berasal dari Turki. Tubuhnya atletis dan memiliki bulu yang cukup panjang.
3. Maine Coon, merupakan kucing jenis american long hair dan berasal dari Amerika Utara. Berat badan kucing ini dapat mencapai 7–11 kilogram untuk jantan dan 5–7 kilogram untuk betina. Biasanya kucing jenis ini berbulu cukup panjang.
4. Siam, Kucing ini berasal dari daerah Siam (Thailand). Kucing ini memiliki tubuh ramping, berbulu pendek dan memiliki sinar mata dengan perpaduan warna almond dan biru. Berat badan Siamese jantan dewasa adalah 4-7 kilogram dan betina dewasa adalah 3-5 kilogram. Untuk menjaga kesehatan tubuhnya yang ramping dan berotot, Siamese membutuhkan makanan dengan kandungan protein dan lemak.



Gambar 2. 1 Kucing

Tanpa menimbang kucing, indikator kondisi kelebihan berat badannya dapat dirasakan hanya dengan membelai punggung kucing dan mengawasinya dengan teliti. Rusuk dan tulang punggungnya tidak boleh terlihat tetapi mudah teraba. Ketika dilihat dari atas, batas pinggangnya harus terlihat jelas dan perutnya tidak boleh melorot.

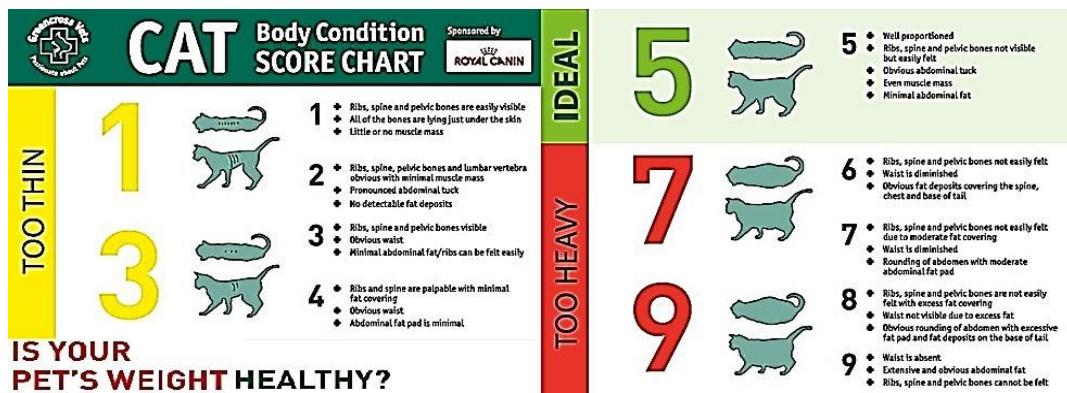
Ada banyak kejadian obesitas pada hewan terutama pada kucing yang berdampak pada kesehatannya. Obesitas adalah kelainan nutrisi yang paling umum terlihat pada kucing dan anjing. Diperkirakan lebih dari 40% kucing mengalami kelebihan berat badan. Ada beberapa faktor penyebab obesitas pada kucing, seperti keturunan, usia dan jenis kelamin. Namun, sebagian besar kasus obesitas pada kucing disebabkan karena pemberian makanan yang tidak diberi takaran per harinya, aturan pemberian porsi makanan pada kucing sesuai dengan berat badan kucing dapat dilihat pada Gambar 3 di bawah ini.

100% Complete & Balanced for Maintenance Complète et équilibrée à 100 % pour le maintien de la santé				
		FEEDING GUIDELINES GUIDE ALIMENTAIRE		
Cat Weight (lbs.) Poids du chat (kg)		Daily Feeding Guide Maintenance		Ration quotidienne Maintien
c = cups g = grammes 1 cup = 88 grams		Weight Loss	Maintain	Perte de poids
4 lbs. 1,8 kg		1/3 c	1/3 c	30 g 30 g
8 lbs. 3,6 kg		2/3 c	1/2 c	60 g 45 g
12 lbs. 5,4 kg		3/4 c	2/3 c	65 g 60 g
16 lbs. 7,3 kg		1 c	3/4 c	90 g 65 g
22 lbs. 10,0 kg		1 1/3 c	1 c	115 g 90 g

Gambar 2. 2 Aturan Porsi Makan Kucing

Resiko kesehatan pada kucing yang memiliki kelainan berat badan/obesitas dapat berupa penyakit jantung, rentan terserang infeksi, diabetes melitus, kelainan syaraf, masalah pernafasan, kanker, radang sendi, masalah kulit dan gangguan reproduksi. Tidak ada patokan berat badan ideal kucing secara khusus karena berat badan kucing tergantung pada ras dan Feline BCS (Body Condition Score), dimana Feline BCS menggunakan sistem lima poin dengan klasifikasinya yaitu Very Thin, Underweight, Ideal, Overweight, dan Obese. Setiap titik poin mewakili peningkatan dan penurunan yang tergantung pada 20% sampai dengan 30% lapisan lemak pada tubuh kucing (Teng, 2018).

Diagram BCS merupakan alat yang bermanfaat bagi pemilik hewan peliharaan untuk meningkatkan kesadaran bahwa hewan peliharaan kemungkinan memiliki masalah dengan berat badan. Diagram Feline BCS dapat dilihat pada Gambar 5 di bawah ini.



Gambar 2.3 Feline BCS Chart

Di dunia ini, terdapat 155 jenis kucing yang populer dan diakui oleh beberapa asosiasi atau organisasi kucing di dunia. Diantaranya adalah FiFe, CFA, WCF, TICA, GCCF, LOOF, ACFA, CCA / AFC, SACC, CCCA, AACE, ACF, NZCF, CatZ, WCNA, dan lain sebagainya. Sedangkan asosiasi kucing di Indonesia yaitu ICA, CFI dan CFSI yang mengacu ke asosiasi internasional tersebut, khususnya ke FiFe, CFA dan TICA.

Dehidrasi pada kucing bisa terjadi ketika hewan tersebut kehilangan cairan secara berlebihan dari tubuhnya tanpa diganti. Saat hal ini terjadi, kucing bukan hanya kehilangan air, tetapi juga beberapa mineral penting dalam tubuh seperti klorida, natrium, dan kalium. Oleh sebab itu, mendapatkan kecukupan air sangat penting untuk kesehatan kucing sekaligus mengganti cairan yang hilang melalui urin dan feses. Air juga diperlukan untuk sirkulasi, pencernaan dan pembuangan kotoran kucing. Jika dehidrasi pada kucing tidak segera diatasi, maka kondisi ini dapat menyebabkan masalah kesehatan yang lebih parah. Dehidrasi pada kucing biasanya disebabkan oleh kucing tidak minum cukup air atau kehilangan air secara berlebihan. Kucing juga dapat kehilangan kelembapan dengan mengeluarkan sedikit keringat melalui cakarnya, tetapi hal ini biasanya tidak menyebabkan kehilangan air secara drastis. Untuk membantu kucing peliharaan kita mempertahankan hidrasi yang sehat, pastikan kucing selalu dapat mengakses air bersih di mangkuk yang bersih setiap hari.

2.3. Makanan dan Minuman Kucing

Kucing mempunyai pola dan perilaku makan yang sangat spesifik. Dalam sehari seekor kucing membutuhkan sekitar 26 gram protein, 9 gram lemak dan 8 gram karbohidrat yang mana unsur tersebut setara dengan kebutuhan kalori sebanyak 52% dari protein, 36% dari lemak dan 12% dari karbohidrat. Beberapa penelitian juga menuturkan bahwa seekor kucing dengan berat badan ± 3.5 kg, hanya memerlukan asupan normal sekitar 50 gram makanan kering/harinya. Kucing tidak memiliki enzim di air liurnya untuk mencerna karbohidrat, karena itu pemberian makanan untuk kucing sangat disarankan murni daging atau mengandung daging dan tidak banyak mengandung biji-bijian karena kucing lebih membutuhkan protein hewani dan hanya membutuhkan sedikit karbohidrat. Ada macam makanan kucing, antara lain:

1. Wet Food / Canned Food, dimana makanan basah tersebut dikemas dalam kaleng dan mengandung tingkat kelembaban yang tinggi, sehingga dapat memenuhi kebutuhan akan air pada kucing namun berpotensi menimbulkan plak pada gigi.
2. Dry Food/ Kibble, merupakan makanan kering dan berbentuk seperti biscuit kecil serta mengandung tingkat kekeringan yang tinggi, sehingga kucing harus melakukan konsumsi air yang banyak untuk memenuhi asupan airnya.



Gambar 2.4 Makanan Kucing

Makanan kering mengandung 8–10% air dan biasanya diproduksi dengan cara mengekstraksi daging dengan tekanan dan suhu yang tinggi. Makanan kering dapat ditambahkan bahan lain seperti lemak makanan dengan cara disemprotkan untuk meningkatkan kandungan gizi dan rasa. Makanan kering memiliki harga yang jauh lebih murah dibandingkan dengan jenis makanan kucing lainnya. (Farndon, 2003).

Berikut ini adalah anjuran asupan nutrisi yang ada pada makanan kucing, antara lain:

1. Memiliki kandungan protein yang tinggi (minimal 25%), semakin tinggi kandungan protein akan semakin baik.
2. Memiliki kandungan lemak yang rendah.
3. Memiliki kadar magnesium yang rendah.

Setiap kucing membutuhkan jumlah air yang berbeda-beda, tergantung rutinitas yang dilakukan, berat tubuh, kesehatan, dan faktor lingkungan. Misalnya, kucing yang tinggal di dalam rumah yang sejuk mungkin tidak minum sebanyak kucing yang sangat aktif yang menghabiskan banyak waktu di luar rumah.

Kucing (usia tiga bulan) dengan berat badan 1,4 kilogram perlu meminum air sebanyak 70 mililiter dalam sehari. Kucing (usia enam bulan) dengan berat badan 2,7 kilogram perlu meminum 135 mililiter air dalam sehari. Untuk kucing dengan berat badan 4 kilogram perlu meminum 200 mililiter air dalam sehari. Kucing besar dengan berat badan 6 kilogram perlu mengonsumsi 300 mililiter air dalam sehari. Kucing yang mengonsumsi makanan kering, di sisi lain, tidak akan minum banyak air dari makanannya. Jadi, kucing perlu terhidrasi dengan minum dari piring atau mangkuk air. Pastikan kucing peliharaan kita mendapatkan cukup air sebagai bagian dari diet hariannya. Jika tidak mendapatkan cukup air, kucing mungkin akan mengalami dehidrasi.

2.4. *Mobile Application*

Aplikasi mobile dapat diartikan sebagai sebuah produk dari sistem komputasi mobile, yaitu sistem komputasi yang dapat dengan mudah dipindahkan secara fisik dan yang komputasi kemampuan dapat digunakan saat mereka sedang dipindahkan. Contohnya adalah Personal Digital Assistant (PDA), smartphone dan ponsel

(Mulyadi, 2010). Berdasarkan jenisnya aplikasi mobile dibagi menjadi beberapa kelompok yaitu:

1. Short Message Service (SMS), merupakan aplikasi mobile paling sederhana, dirancang untuk berkirim pesan dan berguna ketika terintegrasi dengan jenis aplikasi mobile lainnya.
2. Mobile Websites (Situs Web Mobile), merupakan situs web yang dirancang khusus untuk perangkat mobile. Situs web mobile sering memiliki desain yang sederhana dan biasanya bersifat memberikan informasi.
3. Mobile Web Application (Aplikasi Web Mobile), merupakan aplikasi mobile yang tidak perlu diinstal atau dikompilasi pada perangkat target. Menggunakan XHTML, CSS, dan JavaScript, aplikasi ini mampu memberikan pengguna pengalaman layaknya aplikasi native/asli.
4. Native Application (Aplikasi Asli), merupakan aplikasi mobile yang harus diinstal pada perangkat target. Aplikasi ini dapat disebut aplikasi platform, karena aplikasi ini harus dikembangkan dan disusun untuk setiap platform mobile secara khusus.



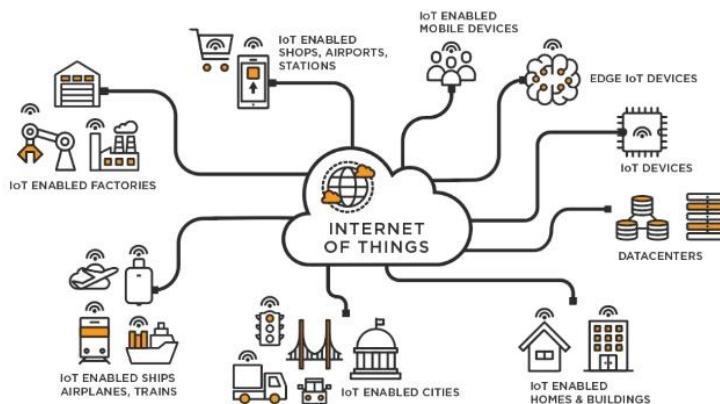
Gambar 2.5 Mobile Application

2.5. Internet of Things (IoT)

IoT didefinisikan sebagai jaringan benda cerdas yang komprehensif yang mempunyai kapasitas untuk mengatur secara otomatis, dalam berbagai informasi, data dan sumber daya, selain itu juga dapat merespon adanya perubahan pada lingkungan.

Dampak IoT pada evolusi internet menuju lingkungan cerdas generasi berikutnya yang sangat bergantung pada integrasi IoT dengan cloud computing. Saat IoT terhubung dengan cloud sejumlah data besar yang telah dikumpulkan dari banyak tempat, dapat diolah dan dianalisis untuk membuat makna informasi ke end-user (Barcelo, 2016).

Cara kerja IoT adalah setiap benda harus mempunyai sebuah IP address. Kemudian IP address pada benda-benda tersebut dikoneksikan dengan jaringan internet, selain itu pada benda tersebut dipasang sebuah sensor yang memungkinkan benda tersebut dapat mengolah informasi itu sendiri bahkan dapat berkomunikasi dengan benda-benda lain yang mempunyai IP address dan juga terkoneksi dengan jaringan intern (Shaifudin, 2016) Kemudian apabila pengolahan informasi tersebut telah selesai maka benda tersebut dapat bekerja dengan sendirinya atau dapat memerintahkan benda lain untuk ikut bekerja.



Gambar 2.6 Internet of Things (IoT)

2.6. Arduino Uno R3 Built-in IOT WiFi ESP8266

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (datasheet). Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung

mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.

ESP8266 adalah microchip Wi-Fi berbiaya rendah, dengan stack TCP / IP penuh dan kemampuan mikrokontroler, diproduksi oleh Espressif Systems. Modul kecil ini memungkinkan mikrokontroler untuk terhubung ke jaringan Wi-Fi dan membuat koneksi TCP / IP sederhana menggunakan perintah Hayes-Style. ESP8266 mampu menghosting aplikasi atau memindahkan semua fungsi jaringan WiFi dari prosesor aplikasi lain.

Modul ini memiliki kemampuan pemrosesan dan penyimpanan on-board yang cukup kuat yang memungkinkannya diintegrasikan dengan sensor dan perangkat khusus aplikasi lainnya melalui GPIO dengan pengembangan minimal di muka dan pemutaran minimal selama runtime. Tingkat integrasi on-chipnya yang tinggi memungkinkan sirkuit eksternal minimal, termasuk modul front-end, dirancang untuk menempati area PCB minimal. ESP8266 mendukung AP/SD untuk aplikasi VoIP dan antarmuka ko-eksistensi Bluetooth, dilengkapi RF yang dikalibrasi sendiri sehingga dapat bekerja dalam semua kondisi pengoperasian, dan tidak memerlukan komponen RF eksternal.

Arduino Uno R3 Built in IoT WiFi ESP8266 adalah versi khusus dari papan ARDUINO R3 klasik. Integrasi penuh mikrokontroler ATmega328 dan IC Wi-Fi ESP8266, dengan memori flash 32 Mb (megabit), dan konverter USB-TTL CH340G pada satu papan. Semua komponen dapat diatur untuk bekerja bersama atau sendiri-sendiri



Gambar 2.7 Arduino Uno R3 Built-in IOT WiFi ESP8266

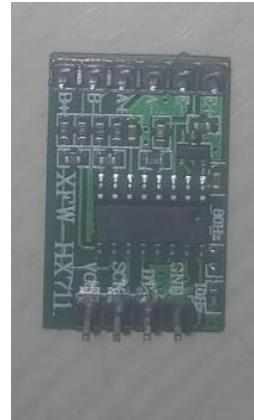
2.7. Sensor *Load Cell*

Sensor load cell merupakan sensor yang dirancang untuk mendeteksi tekanan atau berat sebuah beban, sensor load cell umumnya digunakan sebagai komponen utama pada sistem timbangan digital dan dapat diaplikasikan pada jembatan timbangan yang berfungsi untuk menimbang berat dari truk pengangkut bahan baku, pengukuran yang dilakukan oleh Load Cell menggunakan prinsip tekanan.



Gambar 2.8 Load Cell

Output Tegangan yang dihasilkan dari load cell sangat kecil, sehingga membutuhkan penguat khusus. Penguat yang dapat digunakan adalah modul HX711. HX711 adalah sebuah konverter ADC 24-bit yang dirancang untuk timbangan dan aplikasi kontrol industri. Input multiplexer akan memilih saluran input diferensial A atau B ke penguat PGA (Programmable Gain Amplifier). Saluran A dapat diprogram dengan gain penguatan 128 atau 64, yang sesuai dengan tegangan input diferensial skala penuh masing-masing yaitu kurang lebih 20 mV atau kurang lebih 40 mV. Saluran B memiliki gain penguatan tetap sebesar 32. On-chip power supply menghilangkan kebutuhan regulator eksternal untuk menyediakan daya bagi ADC dan sensor. Input Clock fleksibel, karena bisa di dapat dari sumber clock eksternal, Kristal atau dari osilator on-chip.



Gambar 2.9 Modul HX711

2.8. Sensor Ultrasonic

Sensor Ultrasonik HC SR04 adalah sensor yang berfungsi untuk merubah besaran suara menjadi besaran listrik maupun sebaliknya dengan menggunakan jarak. Prinsip kerja dari sebuah modul pada sensor ini ialah mendeteksi objek dengan cara mengirimkan gelombang ultrasonik dan kemudian menerima pantulan gelombang tersebut dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut. Modul sensor Ultrasonik ini dapat mengukur jarak antara 3 cm sampai 300 cm. (Pratama, 2014)



Gambar 2.10 Sensor Ultrasonic HC-SR04

2.9. Real Time Clock (RTC)

RTC adalah jenis perekam waktu yang bekerja berdasarkan waktu yang sebenarnya atau dengan kata lain berdasarkan waktu yang ada pada jam aktual. Agar dapat berfungsi, perekam waktu ini membutuhkan dua parameter utama yang harus ditentukan, yaitu pada saat mulai (start) dan pada saat berhenti (stop).

Biasanya RTC berbentuk chip dan memiliki fungsi sebagai penyimpanan waktu dan tanggal. Dalam proses penyimpanannya, RTC sendiri memiliki register yang dapat menyimpan data detik, menit, jam, tanggal, bulan dan tahun. RTC sendiri

memiliki 128 lokasi RAM yang terdiri dari 15byte untuk data waktu serta kontrol dan 113byte sebagai RAM umum.



Gambar 2.11 Real Time Clock (RTC) DS3231

2.10. Liquid Crystal Display (LCD 20x4) I2C

LCD merupakan salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya, tetapi memantulkan cahaya yang ada pada sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari backlight. Display LCD sebuah liquid kristal atau perangkat elektronik yang dapat digunakan untuk menampilkan angka atau teks. Ada dua jenis utama layer LCD yang dapat menampilkan numerik (digunakan dalam jam tangan, kalkulator dan lain sebagainya) dan menampilkan teks alfanumerik (digunakan pada mesin foto kopi dan telepon genggam).

Dalam menampilkan numerik tersebut, kristal dibentuk menjadi bar dan dalam menampilkan alfanumerik kristal hanya diatur ke dalam pola titik. Setiap kristal memiliki sambungan listrik individu, sehingga dapat dikontrol secara independen. Ketika kristal mati, cahaya terlihat sama dengan bahan latar belakangnya. Namun ketika sedan dalam kondisi menyala, maka cahaya akan terlihat dan membuat kristal terlihat lebih gelap dari penglihatan mata manusia, sehingga bentuk titik atau bar dapat dilihat dari perbedaan latar belakang.



Gambar 2.12 Liquid Crystal Display (LCD) 20x4 I2C

2.11. Motor Servo

Motor Servo adalah sebuah motor DC yang dilengkapi dengan rangkaian kendali dengan sistem closed feedback yang terintegrasi dalam motor tersebut. Pada motor servo, posisi putaran sumbu (axis) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalamnya. Motor servo disusun dari sebuah motor DC, gearbox, VR/ potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu. Sedangkan sudut dari sumbu tersebut akan diatur berdasarkan lebar pulsa yang ada pada pin kontrol. Motor servo, mampu bekerja dua arah (clockwise–counterclockwise) yang dimana arah dan sudut pergerakan pada rotornya dapat dikendalikan dengan memberikan variasi lebar pulsa (duty cycle) sinyal PWM pada bagian pin kontrol.



Gambar 2.13 Towerpro Motor Servo MG996R

2.12. *Mini Pump Water*

Mini water pump adalah motor pompa air celup yang berukuran kecil. Prinsip kerja pompa air adalah dengan memindahkan sejumlah volume air melalui ruang suction menuju ke ruang outlet dengan menggunakan impeller, sehingga seluruh ruang udara terisi air dan menimbulkan tekanan fluida untuk ditarik melalui tempat satu menuju ke tempat lainnya. *Mini water pump* menggunakan motor DC Brushless dan bekerja dengan tegangan DC 12V 240L/H, pompa air mini ini biasa digunakan untuk akuarium, kolam ikan, hidroponik, robotika atau proyek dalam pembuatan aplikasi yang berbasis mikrokontroller. Kelebihan *mini water pump* ini adalah tidak berisik dalam penggunaanya dan aman ketika bekerja di dalam air.



Gambar 2.14 *Mini Pump Water*

2.13. Relay

Relay merupakan komponen elektronika yang dioperasikan secara listrik dan merupakan 40 komponen electromecanical (elektromekanikal). Terdiri dari 2 bagian utama yaitu elektromagnet (coil) dan mekanikal (seperangkat kontak saklar/switch).

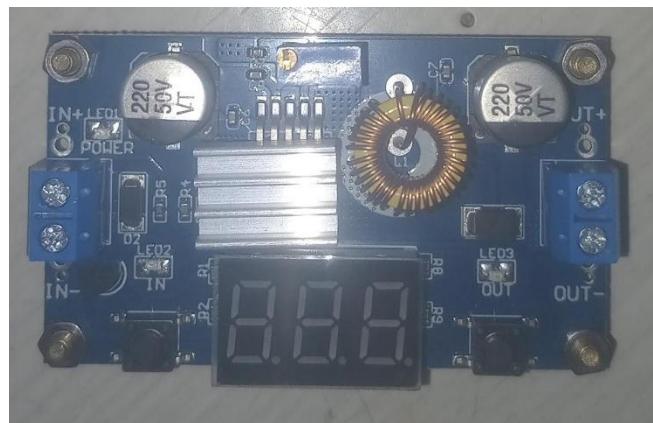
Relay didefinisikan sebagai tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid itu dialiri oleh arus listrik, maka tuas akan tertarik dikarenakan adanya gaya magnet yang terjadi pada solenoid, sehingga kontak saklar akan menutup. Dan apabila arus listrik dihentikan, gaya magnet akan hilang dan tuas akan kembali pada posisi semula, kemudian kontak saklar akan terbuka kembali. (Masinambow, 2014).



Gambar 2.15 Modul Relay

2.14. Stepdown (Module XL4015)

Modul XL4015 ialah module step down DC ke DC yang dapat melewatkkan arus mencapai 5A dengan frekuensi yang mencapai 180KHz untuk PWM. Modul ini memiliki input tegangan 4 – 38VDC dan output tegangan 1.25 – 36VDC seperti pada Gambar 2.16.



Gambar 2. 16 Stepdown (Modul XL4015)

2.15. Flutter

Framework Flutter merupakan software development kit atau bisa disingkat menjadi SDK yang dikembangkan oleh google guna dalam pengembangan aplikasi mobile. Dikarenakan hanya membutuhkan SDK android maka flutter dapat dijalankan dengan ringan di computer device, dan dapat dikembangkan dari android

studio bahkan notepad. Framework Flutter juga dilengkapi dengan fitur bot reload guna untuk melihat perubahan tanpa harus melakukan debugging ulang. Framework Flutter memiliki beberapa kelebihan diantaranya adalah banyaknya kerangka yang dapat digunakan untuk mengembangkan aplikasi lintas platform seperti native script, react native dan fuse. Namun, framework flutter tidak memiliki webview atau widget bawaan, melainkan hanya mempunyai mesin render tersendiri untuk menampilkan widgetnya. Hal tersebut dapat menguntungkan bagi pengembang tampilan UI (User Interface) yang unik juga konsisten pada semua perangkat, karena tidak tergantung pada widget bawaan OEM. Framework Flutter sudah dipaketkan dengan kode pengaksesan platform service dan API, seperti sensor, penyimpanan lokal dan lain sebagainya. Namun, untuk menghindari kendala, pengaksesan API untuk platform yang spesifik, dapat digunakan dengan plugin untuk integrasi (F. Dwi Astuti, 2020).



Gambar 2.17 Logo Flutter

2.16. *Firebase Real-Time Database*

Firebase yakni model layanan yang bekerja di belakang layar dan menghubungkan aplikasi mobile ke cloud storage. Firebase Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung.

Ketika anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis. Semua data Firebase Realtime Database disimpan sebagai objek JSON. Bisa dianggap basis data sebagai JSON tree yang di-host di awan. Tidak seperti basis data SQL, tidak ada tabel atau rekaman. Ketika ditambahkan ke JSON tree, data akan menjadi simpul dalam struktur JSON yang ada. Meskipun basis data menggunakan JSON tree, data yang

tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON yang tersedia untuk membantu anda menulis lebih banyak kode yang bisa dipertahankan. Firebase juga memiliki berbagai layanan seperti NoSQL database realtime dan penyimpanan. Prinsip Could Computing pada sistem ITL yang mana memungkinkan sistem dapat diakses dimanapun juga kapanpun (Mahali, 2019).

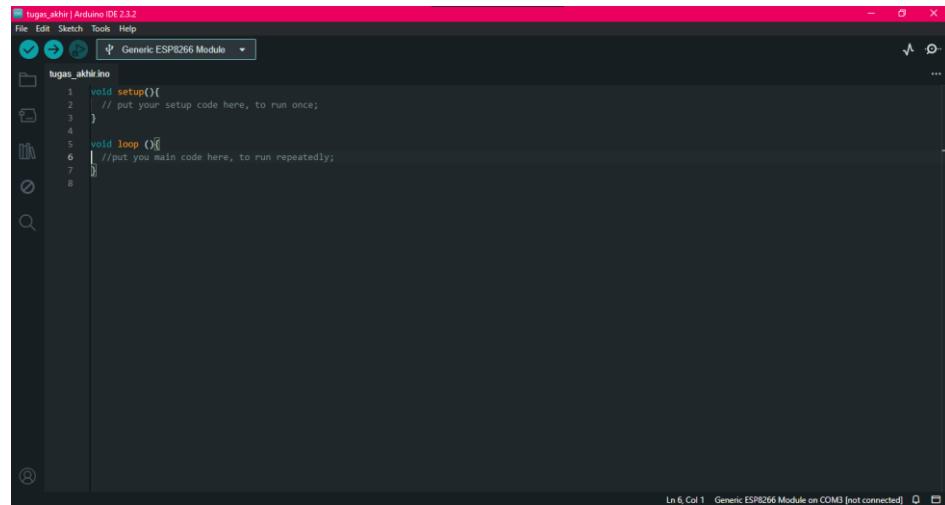


Gambar 2.18 Logo *Firebase Real-Time Database*

2.17. Arduino IDE

Software arduino yang digunakan yaitu driver dan IDE, walaupun masih ada beberapa software lain yang sangat berguna selama pengembangan arduino. (Sharmad Pasha, 2016) Arduino IDE merupakan software pengembangan terintegrasi yang hadir untuk perangkat Arduino yang digunakan untuk membantu memberikan kode mikrokontroler Arduino, selain itu digunakan untuk menghubungkan sensor dan jenis komponen lainnya, serta melakukan operasi pada domain lokal dan global dengan bantuan fungsi perpustakaan.

Integrated Development Environment (IDE) adalah suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. IDE arduino merupakan software yang sangat canggih yang ditulis dengan menggunakan java.



Gambar 2.19 Arduino IDE

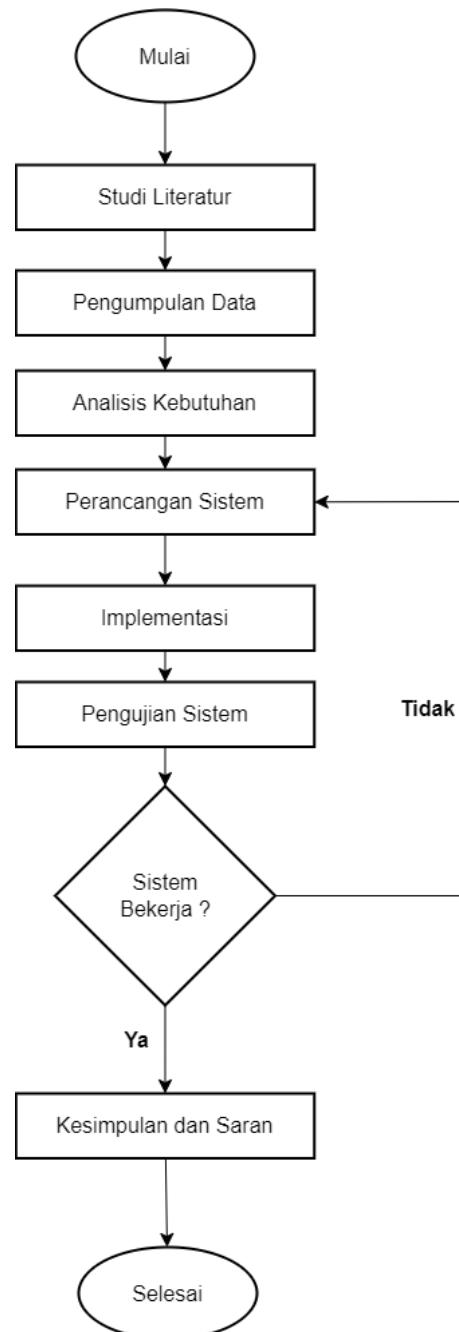
2.18. Pertumbuhan Kucing

aa

BAB III

METODE PENELITIAN

Prosedur dalam penelitian ini terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian, analisis dan kesimpulan. Adapun diagram metode penelitian dapat dilihat pada Gambar 3. 1.



Gambar 3.1 Diagram Alir Penelitian

3.1. Studi Literatur

Tahap ini merupakan studi pustaka dan dokumentasi untuk mendapatkan informasi. Literatur yang digunakan dapat berupa jurnal penelitian ilmiah sebelumnya, buku, artikel yang berkaitan dengan penelitian dan data yang dapat digunakan untuk mendukung penelitian.

Berbagai referensi yang dikumpulkan berupa jurnal terkait dengan penelitian ini yaitu pola makan dan minum kucing. Jurnal dan buku terkait pola makan dan minum kucing, pembuatan mobile application menggunakan framework flutter dipelajari dan dijadikan referensi dalam penelitian ini.

3.2. Metode Pengumpulan Data

Tahap ini berfungsi untuk mengumpulkan data yang berhubungan dengan penelitian yang akan dilakukan. Pengumpulan data pada penelitian ini dilakukan melalui wawancara terhadap pemelihara kucing yaitu Ibu Widhasa Putri di Kubu Raya. Penelitian ini juga memerlukan observasi langsung terhadap pola makan dan minum kucing yang berlokasi di Kubu Raya di lingkungan Widhasa Putry yang berlokasi di Desa Arang Limbung, Kecamatan Sungai Raya, Kabupaten Kubu Raya, Kalimantan Barat. Data yang dikumpulkan meliputi data jenis-jenis pola makan dan minum kucing dan kondisi kucing yang berhubungan dengan kesehatan kucing. Jenis makanan dan minuman pada penelitian ini adalah makanan kering dan air bersih. Pada penelitian ini terdapat dua pola dalam pemberian makan dan minum kucing antara lain pagi hari dan sore hari.

3.3. Analisis Kebutuhan

Tahap analisis kebutuhan meliputi objek yang akan diteliti dan tahap untuk melakukan analisis terhadap kebutuhan dalam membangun sistem baik perangkat keras maupun perangkat lunak.

3.3.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras meliputi komponen seperti sensor dan mikrokontroler yang digunakan untuk membangun sistem kendali otomatisasi pemberian makan dan minum pada kucing.

Adapun perangkat keras yang akan digunakan pada penelitian ini untuk membangun sistem adalah sebagai berikut:

1. Arduino Uno R3 Built-in IoT WiFi ESP8266 berfungsi sebagai Mikrokontroler.
2. Sensor Ultrasonic HC-SR04, berfungsi sebagai pengukur ketinggian air pada wadah minum.
3. Load Cell dan Modul Hx711, berfungsi sebagai pengukur berat pada wadah makanan.
4. *Real Time Clock* DS3231 (RTC), berfungsi untuk penjadwalan waktu pemberian makan dan minum pada system.
5. Motor servo, berfungsi untuk pengeluaran makanan pada wadah makanan.
6. Mini Water Pump, berfungsi untuk pengeluaran air pada wadah minuman.
7. Relay, berfungsi sebagai saklar untuk mini pump water
8. LCD 20x4, berfungsi untuk menampilkan hasil pengukuran pada perangkat.
9. Laptop Asus X441M digunakan sebagai alat penggerjaan dalam membangun sistem mobile application.

3.3.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak meliputi aplikasi yang dibutuhkan untuk membangun sistem. Adapun perangkat lunak yang digunakan pada penelitian ini antara lain:

1. Balsamiq Wireframes digunakan untuk membuat perancangan antarmuka application
2. Visual Studio Code sebagai text editor.
3. *Diagrams.net*, aplikasi website yang digunakan untuk membuat *Flowchart*.
4. Bahasa pemrograman menggunakan Flutter.
5. Fritzing, sebagai media untuk membuat rancangan perangkat keras.
6. Arduino IDE sebagai perangkat pemrograman pada komponen perangkat keras.
7. Android studio sebagai *emulator* android.
8. *Google Firebase real-time* database sebagai database.

3.4. Perancangan Sistem

Tahap perancangan sistem ini dilakukan untuk merancang sistem kendali otomatisasi pemberian makan dan minum pada kucing berdasarkan data yang diperoleh dari proses pengumpulan data. Pada tahap ini dibagi menjadi 2 perancangan antara lain:

3.4.1 Perancangan Perangkat Keras

Adapun perancangan perangkat keras terdiri dari:

1. Perancangan sistem penjadwalan RTC DS3231
2. Perancangan sistem untuk mengukur berat wadah makan kucing menggunakan sensor Berat loadcell 5 kg dan Modul HX711.
3. Perancangan sistem untuk mengukur volume air pada wadah dan tabung wadah dan tabung minum kucing menggunakan sensor Jarak Ultrasonic HC-SR04.
4. Perancangan sistem pengendalian mini pump water.
5. Perancangan sistem pengendalian servo.
6. Perancangan tampilan LCD 20x4 I2C pembacaan sensor.
7. Perancangan keselurhan sistem.

3.4.2 Perancangan Perangkat Lunak

Adapun perancangan perangkat lunak pada sistem pemantauan kesehatan kambing bedasarkan suhu tubuh dan detak jantung terdiri dari:

1. Perancangan antarmuka berbasis mobile application.
2. Perancangan database
3. Perancangan kode program pada setiap sensor.

3.5. Implementasi

Tahap implementasi pada penelitian ini adalah tahap yang dilakukan setelah perancangan perangkat sudah selesai dilakukan baik perangkat keras maupun perangkat lunak. Implementasi dimulai dengan perakitan perangkat keras sesuai dengan perancangan yang sudah dilakukan. Selanjutnya melakukan pemrograman perangkat lunak hingga tampilan sesuai dengan yang diinginkan.

3.5.1 Implementasi Perangkat Keras

Implementasi perangkat keras yang dilakukan dalam penelitian ini yaitu :

1. Implementasi sistem penjadwalan RTC DS3231.

2. Implementasi sistem untuk mengukur berat wadah makan kucing menggunakan sensor Berat loadcell 5 kg dan Modul HX711.
3. Implementasi sistem untuk mengukur volume air pada wadah dan tabung wadah dan tabung minum kucing menggunakan sensor Jarak Ultrasonic HC-SR04.
4. Implementasi sistem kendali mini pump water.
5. Implementasi sistem kendali servo
6. Implementasi sistem LCD 20x4 I2C pembacaan sensor.
7. Implementasi keselurhan sistem.

3.5.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak yang dilakukan dalam penelitian ini yaitu :

1. Implementasi database
2. Implementasi antarmuka berbasis mobile application.
3. Implementasi kode program setiap sensor.

3.6. Pengujian

Tahap pengujian dilakukan untuk menguji sistem yang suda dibangun. Pengujian ini bertujuan untuk mengetahui akurasi dan fungsi-fungsi dari sistem yang sudah dibangun sesuai atau masih terdapat kesalahan. Tahap ini meliputi pengujian seluruh komponen dari sistem baik perangkat lunak maupun perangkat keras.

Apabila sistem belum berjalan sesuai dengan perancangan yang dibangun sebelumnya, maka pengujian belum berhasil. Apabila sistem belum berhasil, maka akan melakukan analisis terhadap perancangan sebelumnya untuk menemukan kekurangan dari sistem. Apabila sistem dapat berjalan dengan baik, maka pengujian berhasil. Adapun tahapan pengujian sistem adalah sebagai berikut:

1. Pengujian pembacaan dan akurasi sensor Loadcell 5kg dan Modul HX711.
2. Pengujian pembacaan dan akurasi sensor jarak Ultrasonik.
3. Pengujian akurasi penjadwalan pada sensor RTC DS3231.
4. Pengujian kendali servo MG996R pada pengisian wadah makan kucing.
5. Pengujian kendali Mini Pump Water pada pengisian wadah minum kucing.

6. Pengujian blackbox, dilakukan untuk mengetahui mobile application dapat menghasilkan hasil yang sesuai dengan fungsi yang telah ditetapkan oleh program.
7. Pengujian keselurhan sistem.

3.7. Kesimpulan dan Saran

Tahap akhir dari penelitian ini adalah menarik kesimpulan yang akan menjawab pertanyaan yang ada pada perumusan masalah berdasarkan hasil pengujian sistem yang telah dilakukan pada penelitian ini. Saran yang baik dapat mengoptimalkan kinerja pada penelitian sebelumnya.

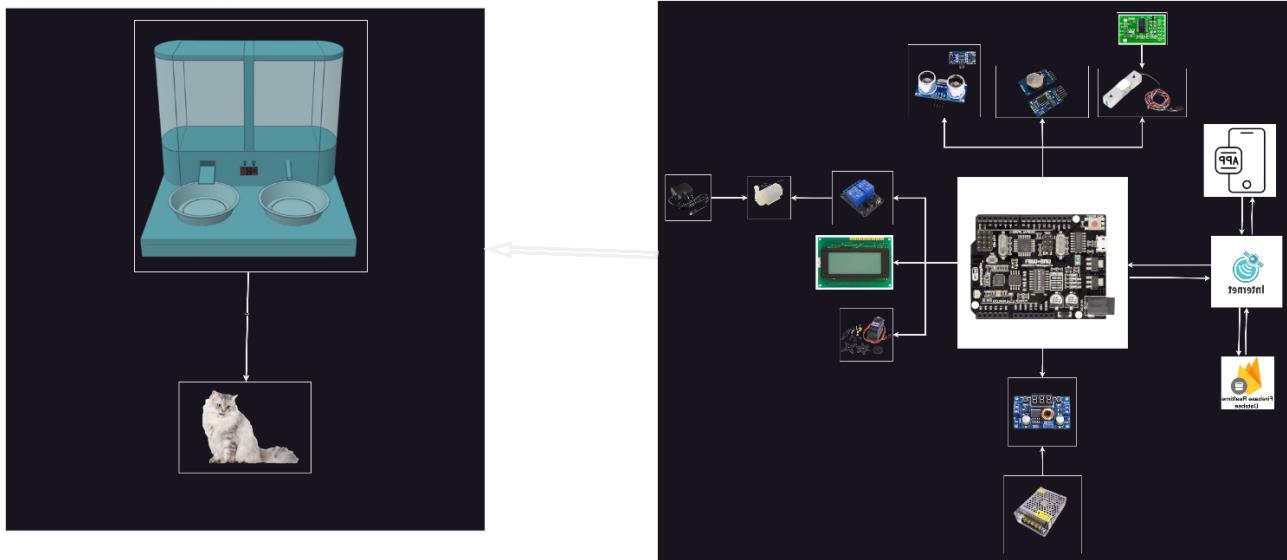
BAB IV

PERANCANGAN

4.1. Deskripsi Sistem

Penelitian yang dilakukan adalah membangun sebuah sistem kendali otomatisasi makan dan minum kucing berbasis mobile application secara online. Sistem dapat melakukukan pemberian makan dan minum pada wadah secara otomatis saat kondisi wadah makan kurang dari 120 gram dan wadah minum kurang dari 300 mililiter. Sistem juga dapat menyalakan Mini Pump Water dan Motor Servo sesuai dengan kondisi penjadwalan makan dan minum dengan kondisi jika Real Time Clock membaca jadwal pada jam 07.00 pagi dan 17.00 sore maka relay akan menyala untuk menghidupkan pump dan servo akan menyala, kemudian jika jadwal tidak sesuai dengan waktu tersebut maka pump dan servo akan mati.

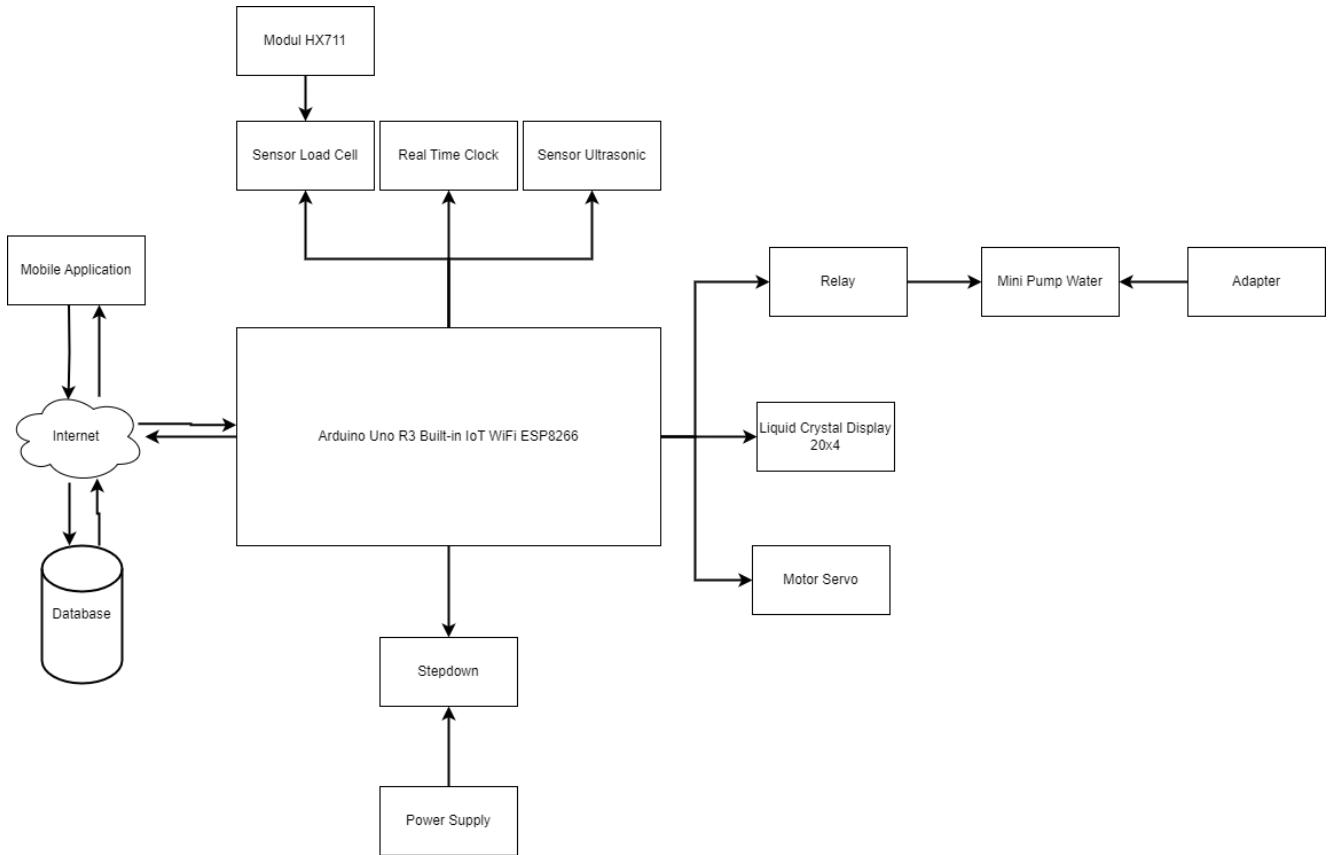
Sistem menggunakan Arduino Uno R3 Built-in IoT WiFi ESP8266 yang digunakan untuk melakukan pembacaan data pakan dari sensor Loadcell 5 Kilogram dan HX711, pengukuran data jarak dari wadah minum menggunakan Ultrasonic HC-SR04 dan pembacaan waktu penjadwalan menggunakan Real Time Clock DS3231. Pembacaan nilai dari sensor akan di tampilkan pada LCD dan data akan masuk ke firebase database lalu ditampilkan pada mobile application. Sistem juga dapat mengirimkan notifikasi jika wadah makan dan minum kosong melalui tampilan LCD dab Mobile Application. Rancangan sistem kendali otomatisasi pemberian makan dan minum pada kucing dapat dilihat pada Gambar 4.1.



Gambar 4.1 Rancangan Sistem Perangkat Otomatisasi Makan dan Minum Kucing

4.2. Perancangan Arsitektur Sistem

Perancangan sistem dalam penelitian ini terbagi menjadi dua yaitu perancangan perangkat keras dan perangkat lunak. Pada perancangan perangkat keras terdiri dari beberapa komponen perangkat keras yaitu Arduino Uno R3 Built-in IoT WiFi ESP8266, Load Cell 5 Kilogram dan Modul HX711, Real Time Clock DS3231, Sensor Ultrasonik, Liquid Crystal Display (LCD) 20X4 I2C, Mini Pump Water, Tower Servo MG996R, Jack DC, Adapter 5 Volt 1 Ampere, modul relay 1 channel, Stepdown XL4015 dan Power Supply 5 volt 5 Ampere. Sedangkan pada perancangan perangkat lunak terdiri dari perancangan logika dan pengkondisian pada Arduino Uno R3 Built-in IoT WiFi ESP8266 perancangan antar muka pada mobile application. Adapun perancangan arsitektur sistem secara umum dapat dilihat pada Gambar 4. 2.



Gambar 4.2 Rancangan Arsitektur Sistem

Untuk mempermudah pemahaman fungsi dari setiap blok, berikut penjelasan masing-masing blok tersebut:

1. Arduino Uno R3 Built-in WiFi ESP8266 sebagai mikrokontroller yang berfungsi untuk melakukan pengolahan data dari sensor dan pengendali untuk pemberian makan dan minum kucing.
2. Database berfungsi sebagai pengumpulan dan penyimpanan data.
3. Internet berfungsi sebagai akses informasi dan akses pengiriman data pada sistem.
4. Mobile Application berfungsi sebagai antarmuka pengguna untuk melihat informasi pembacaan dari setiap sensor, notifikasi penjadwalan makan dan minum, serta dapat mengakses mode manual dan mode otomatis.
5. Real Time Clock DS-3231 berfungsi sebagai penjadwalan untuk mengatur waktu pola makan dan minum kucing
6. Sensor Load Cell berfungsi sebagai timbangan digital untuk mengukur berat makanan yang ada pada wadah makan kucing.

7. Sensor Ultrasonic HC-SR04 berfungsi untuk mengukur ketinggian air pada tabung dan wadah minum kucing.
8. Modul *Relay* berfungsi sebagai *switch on/off* untuk mengontrol pompa.
9. Adapter berfungsi sebagai sumber arus listrik yang diperlukan pada modul relay.
10. Pompa untuk memompa air dari wadah, dan mengalirkan ke tempat lain dengan bantuan selang. Pompa terkoneksi dengan adaptor sebagai catu daya dan mengonversi tegangan AC menjadi DC.
11. LCD 20x4 I2C berfungsi untuk menampilkan output dari sensor yang digunakan.
12. Servo berfungsi sebagai alat penggerak yang digunakan untuk membuka katup makan pada tabung makan kucing.
13. Stepdown berfungsi sebagai mengubah tegangan tinggi dengan arus rendah menjadi tegangan rendah dengan arus tinggi.
14. Power Supply berfungsi sebagai sumber arus listrik yang diperlukan pada keseluruhan system.

Sistem otomatis untuk pemberian makan dan minum kucing yang dirancang dalam penelitian ini terdiri dari berbagai komponen utama yang bekerja secara terintegrasi untuk memastikan ketersediaan makanan dan air yang konsisten bagi kucing peliharaan. Pada sistem makan, tabung makanan dengan kapasitas 1 kilogram berfungsi sebagai penampung utama untuk makanan kering, yang kemudian disalurkan melalui pipa jalur makanan menuju wadah makan kucing. Pengaturan aliran makanan dilakukan oleh katup tabung yang dikendalikan oleh motor servo, yang membuka dan menutup katup sesuai perintah dari mikrokontroler.

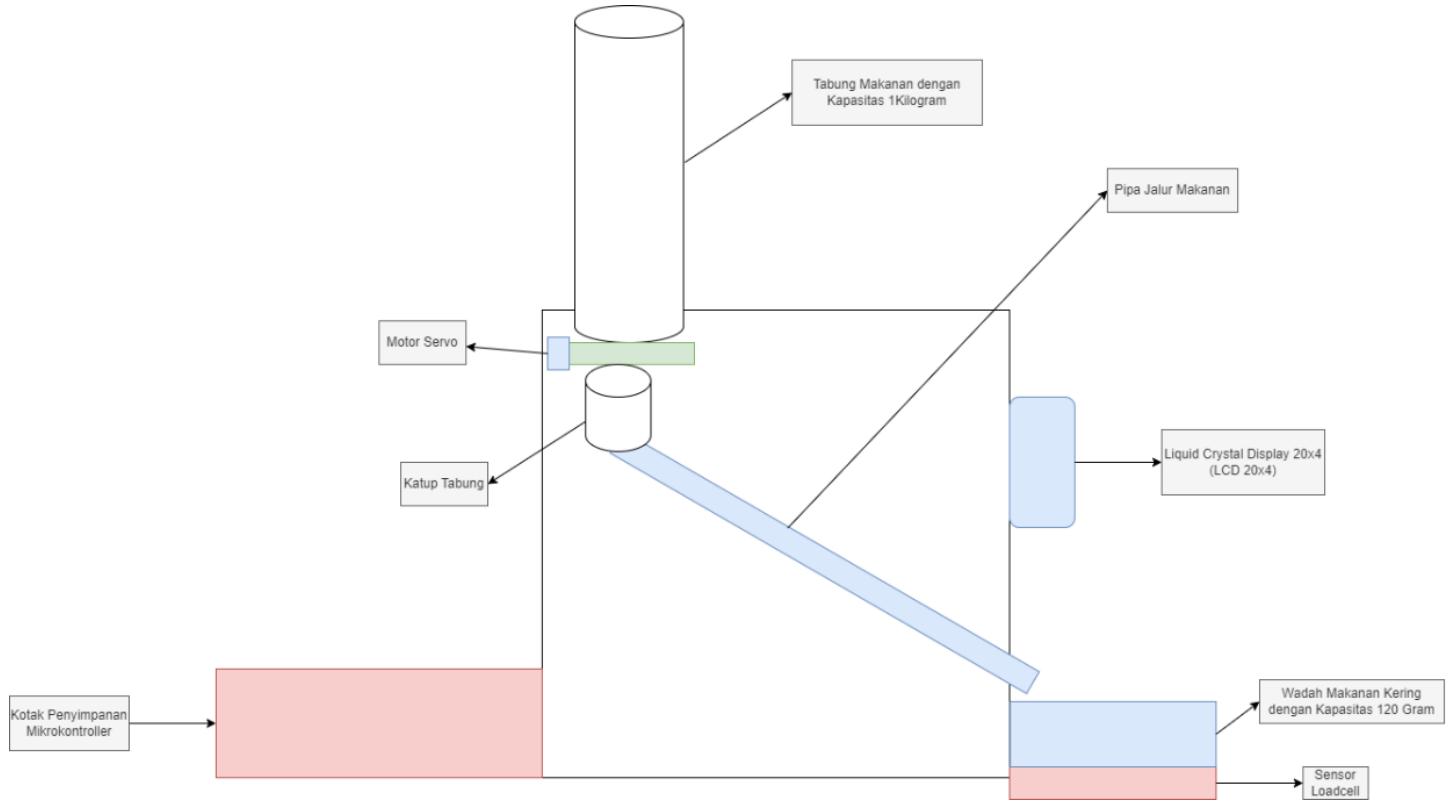
Wadah makanan kering dengan kapasitas 120 gram digunakan untuk menyajikan makanan, dan sensor Loadcell memonitor jumlah makanan yang ada di dalam wadah, dengan data yang diteruskan ke mikrokontroler untuk menentukan kebutuhan penambahan makanan. Informasi terkait status sistem makan, seperti jumlah makanan yang tersisa dan status katup, ditampilkan secara real-time pada Liquid Crystal Display (LCD) 20x4. Pada sistem minum, tabung minuman dengan kapasitas 1 liter bertindak sebagai penampung utama air minum, yang diawasi oleh

sensor ultrasonik untuk memantau level air. Sensor ini mengukur waktu tempuh gelombang suara yang dipancarkan hingga kembali setelah dipantulkan oleh permukaan air, dan data tersebut dikirim ke mikrokontroler untuk diproses. Selang air menghubungkan tabung dengan wadah minum kucing, di mana air dipindahkan oleh pompa air mini yang diaktifkan oleh mikrokontroler ketika level air dalam wadah minum terdeteksi rendah. Wadah minuman dengan kapasitas 300 mililiter, yang dipantau oleh sensor ultrasonik, berfungsi sebagai tempat minum kucing, dengan sistem pengisian ulang air yang dilakukan secara berkala. Informasi terkait status sistem minum, seperti level air dan status pompa, juga ditampilkan pada LCD 20x4.

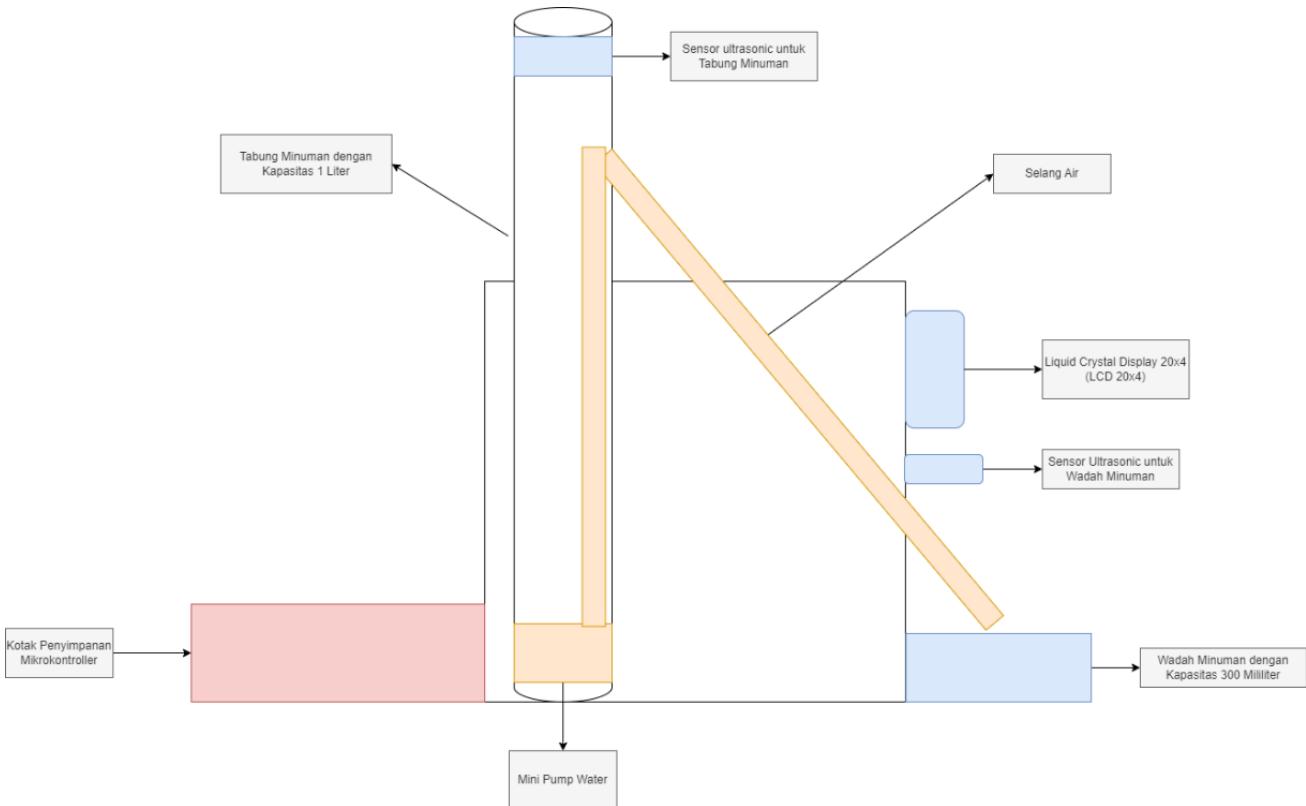
Seluruh sistem, baik untuk makan maupun minum, dikendalikan oleh mikrokontroler yang ditempatkan dalam kotak penyimpanan khusus untuk melindunginya dari gangguan eksternal, memastikan fungsi sistem yang optimal. Adapun perancangan dan design 3D rangkaian sistem makan dan minum kucing dapat dilihat pada Gambar 4.3, Gambar 4.4. dan Gambar 4.5.



Gambar 4.3 Design 3D Automatic Pet Feeder



Gambar 4.4 Rangkaian Sistem Makan Kucing



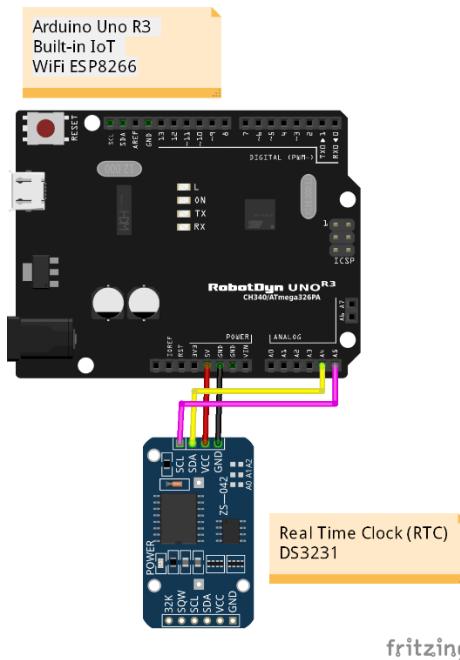
Gambar 4.5 Rangkaian Sistem Minum Kucing

4.3. Perancangan Perangkat Keras

Perancangan perangkat keras merupakan tahapan dalam membangun sebuah sistem smarterrarium. Perancangan perangkat keras ini memiliki langkah yang mengacu pada rancangan arsitektur sistem. Perancangan yang dilakukan yaitu perangkaian alat dengan beberapa komponen yang dibutuhkan sehingga menjadi sebuah sistem yang dimana setiap rancangan tersebut dapat saling terhubung dan bekerja dengan baik.

4.3.1 Perancangan Sistem Penjadwalan Real Time Clock (RTC)

Pada sistem penjadwalan, RTC digunakan untuk melakukan penjadwalan waktu untuk makan dan minum kucing. Kemudian pada saat RTC membaca waktu pukul 07.00 dan 17.00, maka mini pump water akan mengisi air pada wadah makan dan servo akan membuka katup makan untuk mengisi wadah makan kucing. Tampilan dari penjadwalan waktu dapat dilihat pada Gambar 4.6.



Gambar 4.6 Rancangan Sistem Penjadwalan

Pada koneksi RTC dengan Arduino Uno R3 Built-in IoT WiFi ESP8266 digunakan empat pin pada RTC. 5.0V sebagai pin catu daya, GND sebagai pin negatif, SDA pada pin A4 dan SCL pada pin A5. Koneksi dapat dilihat pada Tabel 4.1.

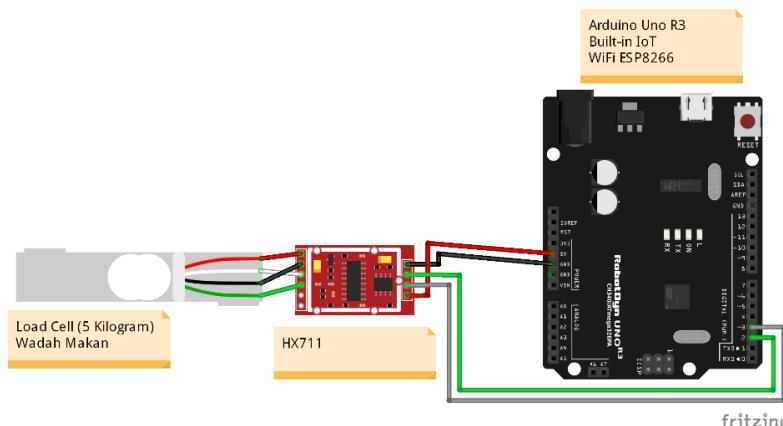
Tabel 4.1 Koneksi Pin RTC dengan Arduino Uno R3 Built-in IoT WiFi ESP8266

Arduino Uno R3 Built-in IoT WiFi ESP8266	Real Time Clock (RTC) DS3231
GND	GND
5.0 V	VCC
A4	SDA
A5	SCL

4.3.2 Perancangan Sistem Pengukuran Berat Wadah Makan

Perangkat keras yang dirancang adalah sensor load cell digunakan untuk mengukur berat badan, adapun komponen yang digunakan untuk membangun sistem adalah Arduino Uno R3 Built-in IoT WiFi ESP8266, load cell 5 kilogram(kg) dan modul HX711.

Untuk menghubungkan load cell 5 kg ke Arduino Uno R3 Built-in IoT WiFi ESP8266 diperlukan modul tambahan berupa modul HX711 sebagai penerima input dari output sensor load cell berupa sinyal analog kemudian output HX711 akan masuk ke Arduino Uno R3 Built-in IoT WiFi ESP8266 berupa data digital. Tampilan dari perancangan load cell, tabel koneksi pin Modul HX711 dan tabel koneksi pin load cell dapat dilihat pada Gambar 4.7, Tabel 4.2 dan Tabel 4.3.



Gambar 4.7 Rancangan Sistem Timbangan Tabung dan Wadah Makan

Tabel 4.2 Koneksi Pin Modul HX711 dengan Arduino Uno R3
Built-in IoT WiFi ESP8266

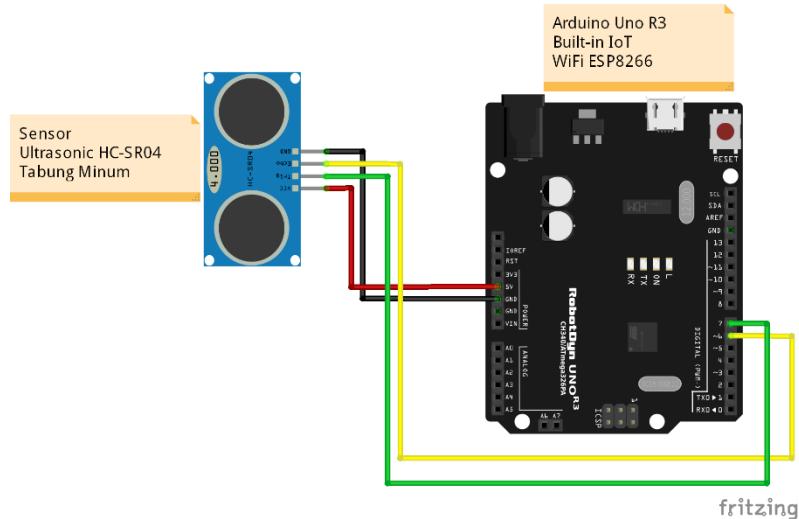
Arduino Uno R3 Built-in IoT WiFi ESP8266	Module HX711
GND	GND
5.0V	VCC
2	DT
3	SCK

Tabel 4.3 Pin Modul HX711 dan Load Cell

HX711	Load Cell
E+	Red
E-	Black
A-	White
A+	Green

4.3.3 Perancangan Sistem Pengukuran Tabung Air Minum

Sensor Jarak Ultrasonik digunakan untuk mendeteksi tinggi air pada tabung minum kucing. Dalam penelitian ini tabung minum merupakan wadah untuk air yang akan disalurkan terus menerus menggunakan mini pump water ke wadah minum kucing. Sistem deteksi jarak air digunakan untuk memberikan notifikasi ke LCD 20x4 jika air pada tabung minum mulai habis yaitu pada jarak kurang dari 12 cm. Tampilan dari perancangan deteksi jarak Ultasonik dapat dilihat Gambar 4.8.



Gambar 4.8 Rangkaian Sistem Ketinggian Tabung Air

Pada koneksi sensor Ultrasonik dengan Arduino Uno R3 Built-in IoT WiFi ESP8266 digunakan empat pin pada sensor Ultrasonik. VCC sebagai pin catu daya, Trig pada pin 6, Echo pada pin 7 dan GND sebagai pin negatif. Koneksi dapat dilihat pada Tabel 4.4.

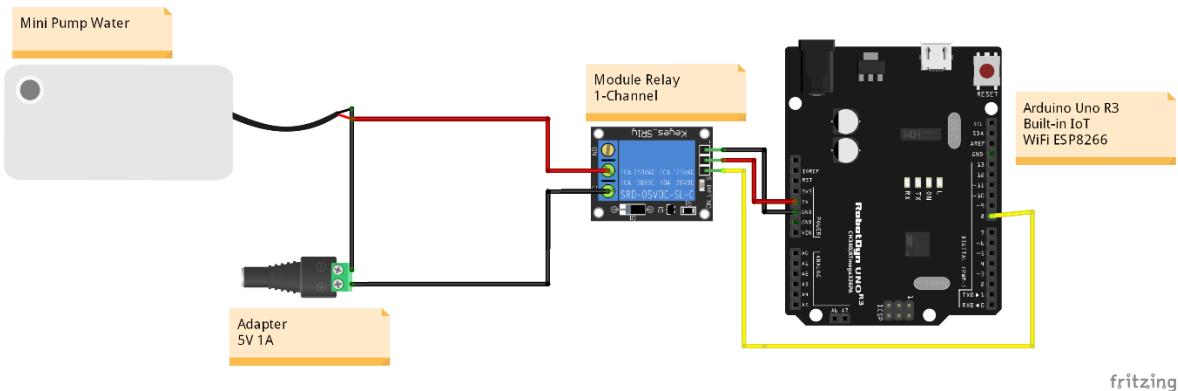
Tabel 4.4 Koneksi Pin Ultrasonic Tabung dengan Arduino Uno R3 Built-in IoT WiFi ESP8266

Arduino Uno R3 Built-in IoT WiFi ESP8266	Ultrasonic Tabung
GND	GND
5.0V	VCC
6	Trig
7	Echo

4.3.4 Perancangan Sistem Kendali Mini Pump Water

Perancangan sistem yang dibangun menggunakan modul relay dengan 1 channel sebagai saklar switch on/off untuk mengontrol mini pump water dengan pengendalian melalui Arduino Uno R3 Built-in IoT WiFi ESP8266 sebagai mikrokontroler dari sebuah sistem yang dibangun. Arduino Uno R3 Built-in IoT

WiFi ESP8266 berfungsi sebagai perangkat yang memberikan sinyal berupa hidup atau mati kepada pin yang terhubung dengan relay. Ketika sinyal berkondisi hidup, maka Arduino Uno R3 Built-in IoT WiFi ESP8266 akan menjaga relay dalam keadaan terhubung. Namun, jika sinyal berubah menjadi mati, maka Arduino Uno R3 Built-in IoT WiFi ESP8266 akan memutuskan kembali koneksi pada relay. Adapun gambar perancangan sistem pengendalian relay dapat dilihat pada Gambar 4.9.



Gambar 4.9 Rancangan Sistem Kendali Mini Pump Water

Pada koneksi relay dengan Arduino Uno R3 Built-in IoT WiFi ESP8266 digunakan 3 pin pada relay. GND sebagai pin negatif atau Ground, dan VCC sebagai pin catu daya sebesar 5V sekaligus sebagai pin masukan, IN sebagai pin masukkan mini pump water. Koneksi dapat dilihat pada Tabel 4. 5.

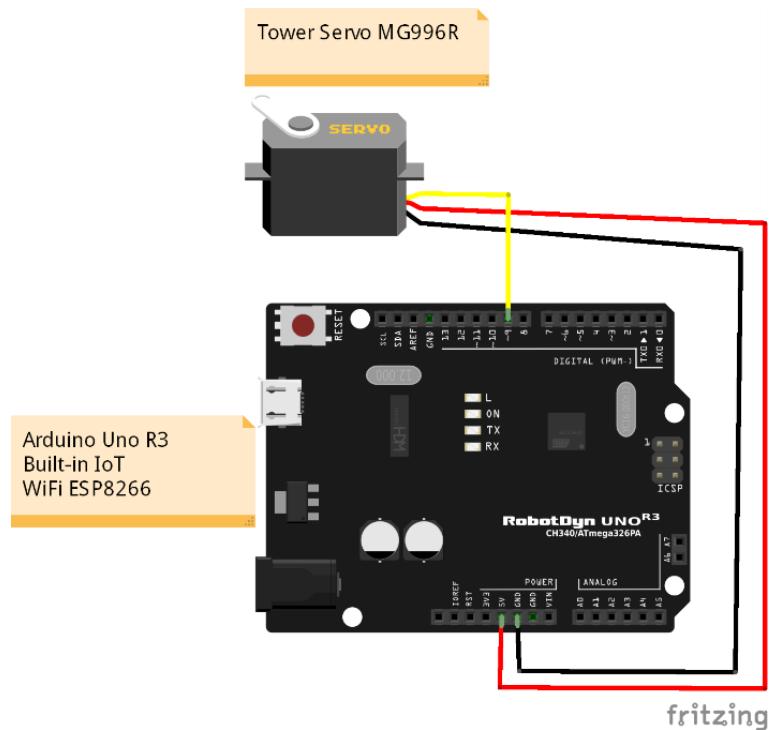
Tabel 4. 5 Koneksi Pin Modul Relay dengan Arduino Uno R3 Built-in IoT WiFi ESP8266

Arduino Uno R3 Built-in IoT WiFi ESP8266	Modul Relay
GND	GND
5.0V	VCC
8	IN

4.3.5 Perancangan Sistem Kendali Servo

Perancangan sistem kendali servo dibangun untuk mengontrol katup pada tabung pakan dengan pengendalian melalui Arduino Uno R3 Built-in IoT WiFi ESP8266 dan kemudian menyalurkan makanan ke wadah makan kucing secara otomatis berdasarkan waktu yang telah ditentukan.

RTC DS3231 berfungsi sebagai pengatur waktu yang akurat, di mana Arduino Uno R3 Built-in IoT WiFi ESP8266 akan membaca waktu yang disediakan oleh RTC DS3231 dan memberikan sinyal kepada servo untuk membuka atau menutup katup pada tabung makan. Ketika waktu yang ditentukan tercapai, Arduino Uno R3 Built-in IoT WiFi ESP8266 akan mengaktifkan servo untuk membuka katup tabung makan, dan setelah periode waktu tertentu, servo akan kembali menutup katup tabung makan secara otomatis. Gambar perancangan sistem pengendalian servo dapat dilihat pada Gambar 4.10.



Gambar 4.10 Rancangan Sistem Kendali Servo

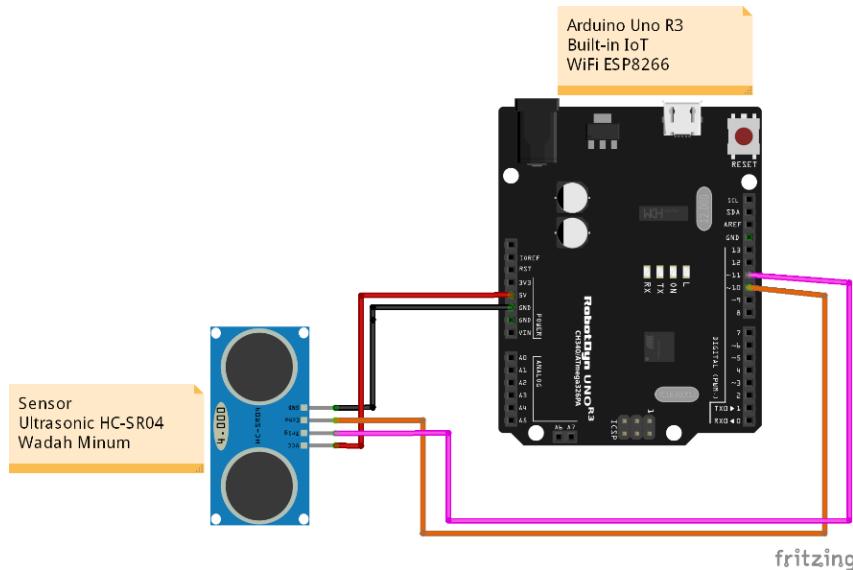
Pada koneksi servo dengan Arduino Uno R3 Built-in IoT WiFi ESP8266 digunakan 3 pin pada servo. VCC sebagai pin catu daya, GND sebagai pin negatif dan PWM sebagai pin masukkan pulse. Koneksi dapat dilihat pada Tabel 4.6.

Tabel 4.6 Koneksi Pin Servo dengan Arduino Uno R3 Built-in IoT WiFi ESP8266

Arduino Uno R3 Built-in IoT WiFi ESP8266	Real Time Clock (RTC) DS3231
GND	GND
5.0V	VCC
9	PWM

4.3.7 Perancangan Sistem Pengukuran Wadah Air Minum

Pada sistem pengukuran wadah air minum, sensor jarak Ultrasonik digunakan untuk mendekripsi tinggi air pada wadah minum kucing. Dalam penelitian ini wadah minum merupakan wadah untuk air yang akan diminum oleh kucing dan dialirkan dari tabung air minum menggunakan mini pump water ke wadah minum kucing. Sistem deteksi jarak air digunakan untuk memberikan notifikasi ke LCD 20x4 jika air pada wadah minum mulai habis yaitu pada jarak kurang dari 12cm. Tampilan dari perancangan pengukuran wadah air minum dapat dilihat Gambar 4.11.



Gambar 4.11 Rangkaian Sistem Ketinggian Wadah Air

Pada koneksi sensor Ultrasonik dengan Arduino Uno R3 Built-in IoT WiFi ESP8266 digunakan empat pin pada sensor Ultrasonik. VCC sebagai pin catu daya, Echo pada pin 11, Trig pada pin 10 dan GND sebagai pin negatif. Koneksi dapat dilihat pada Tabel 4.7.

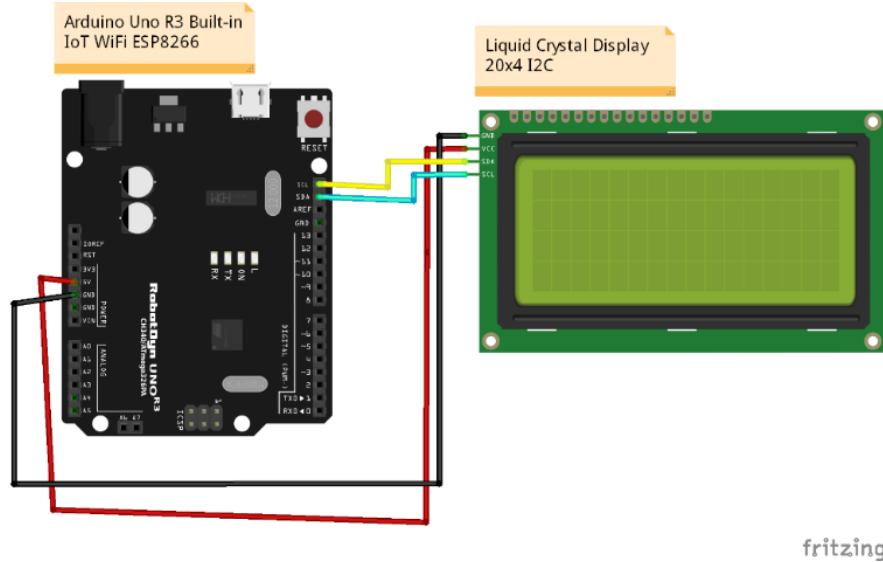
Tabel 4.7 Koneksi Pin Ultrasonic Wadah dengan Arduino Uno R3 Built-in IoT WiFi ESP8266

Arduino Uno R3 Built-in IoT WiFi ESP8266	Ultrasonic Wadah
GND	GND
5.0V	VCC
10	Trig
11	Echo

4.3.7 Perancangan Sistem Tampilan Liquid Crystal Display (LCD) 20x4-I2C

Pembacaan Sensor

Liquid Crystal Display (LCD) 20x4-I2C digunakan untuk menampilkan hasil pembacaan penjadwalan waktu makan dan minum kucing dan pada Real Time Clock berupa waktu pada pukul 07.00 dan 17.00. Pada penelitian ini penggunaan LCD juga digunakan untuk menampilkan notifikasi dari Sensor jarak Ultrasonik dan load cell yang akan memberikan notifikasi jika berat wadah makan dan wadah air sudah mulai habis. Tampilan dari perancangan pembacaan sensor dapat dilihat Gambar 4.12.



Gambar 4.12 Perancangan Sistem Display Data

Pada koneksi LCD dengan Arduino Uno R3 Built-in IoT WiFi ESP8266 digunakan empat pin pada LCD. VCC sebagai pin catu daya 5 Volt, SDA pada pin SDA, SCL pada pin SCL dan GND sebagai pin negatif. Koneksi dapat dilihat pada Tabel 4.8.

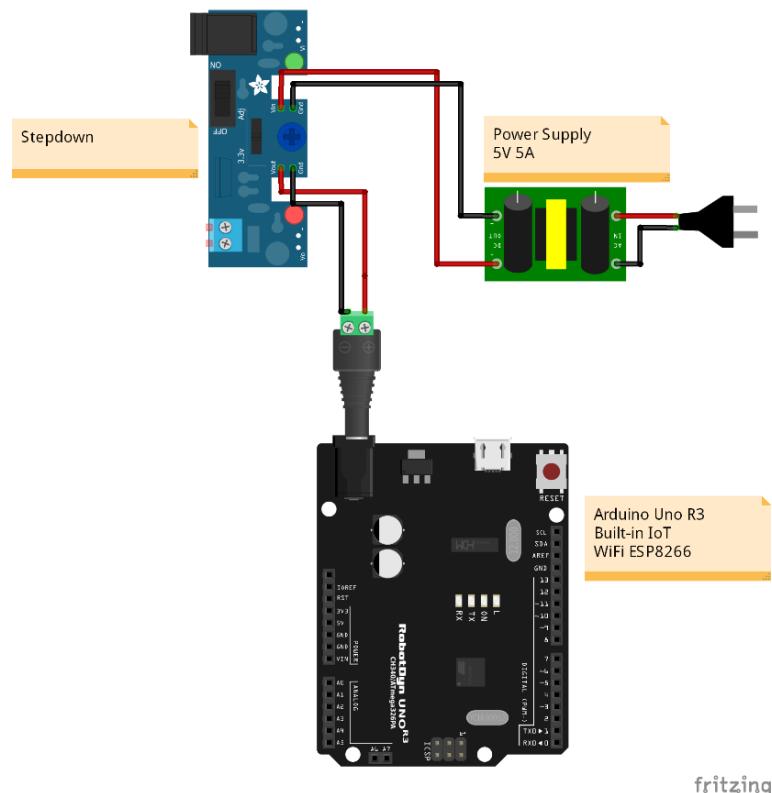
Tabel 4.8 Koneksi Pin LCD 20x4 I2C dengan Arduino Uno R3 Built-in IoT WiFi ESP8266

Arduino Uno R3 Built-in IoT WiFi ESP8266	LCD 20x4 I2C
GND	GND
5.0V	VCC
SDA	SDA
SCL	SCL

4.3.8 Perancangan Sistem Daya Pada Keseluruhan Sistem

Perancangan sistem daya pada gambar ini melibatkan beberapa komponen utama untuk menyuplai daya ke Arduino Uno R3 Built-in IoT WiFi ESP8266. Sistem ini menggunakan sumber daya utama berupa catu daya 5V dengan arus 5A,

yang kemudian diturunkan dan distabilkan oleh modul stepdown. Modul stepdown ini menyesuaikan tegangan output agar sesuai dengan kebutuhan daya Arduino Uno. Tegangan yang sudah disesuaikan tersebut kemudian disalurkan ke Arduino Uno melalui konektor DC, memastikan bahwa perangkat mendapatkan pasokan daya yang stabil dan aman untuk operasi. Tampilan dari perancangan daya pada keseluruhan sistem dapat dilihat Gambar 4.13.



Gambar 4.13 Perancangan Sistem Daya Pada Keseluruhan Sistem

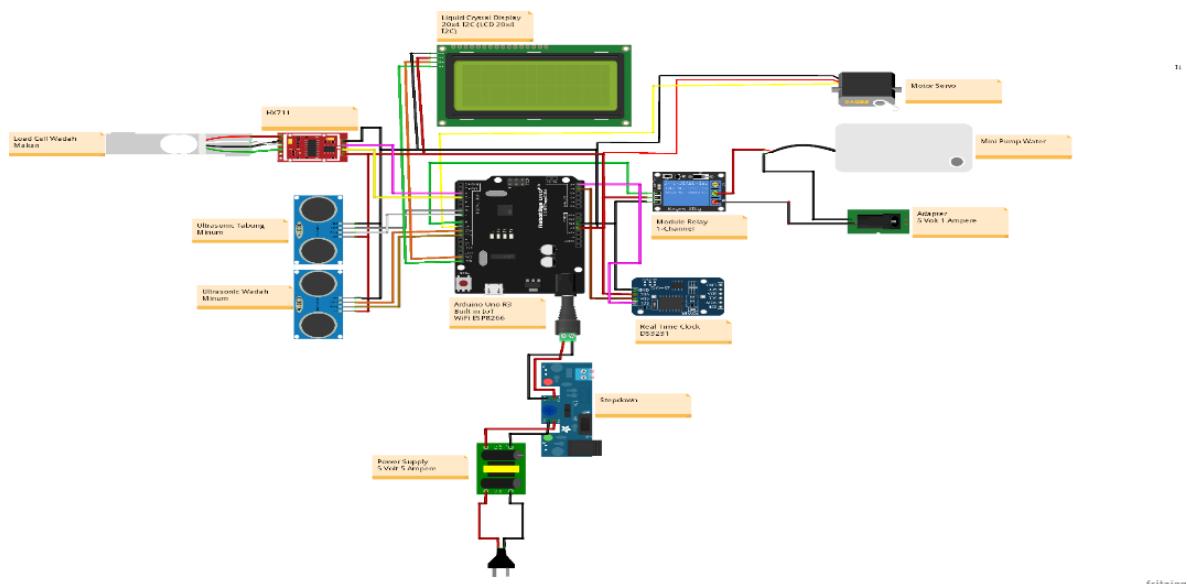
Pada koneksi catu daya dengan Arduino Uno R3 Built-in IoT WiFi ESP8266 dihubungkan melalui stepdown menggunakan DC Jack, Vout sebagai daya output ke arduino dan Vin sebagai daya input dari catu daya ke Arduino Uno R3 Built-in IoT WiFi ESP8266. Koneksi dapat dilihat pada tabel 4.9.

Tabel 4.9 Koneksi Daya Pada Keseluruhan Sistem

Arduino Uno R3 Built-in IoT WiFi ESP8266	Step Down		Power Supply	Sumber Listrik
Power Jack	Vout	Vin		
VCC (Red)	+	+	DC Out (+)	AC IN
GND (Black)	-	-	DC In (-)	

4.3.9 Perancangan Keseluruhan Sistem

Perancangan keseluruhan sistem adalah gabungan perangkat dan koneksi keseluruhan dari perancangan sistem penjadwalan Real Time Clock, perancangan berat wadah makan, perancangan tabung air minum, perancangan pengendalian relay, perancangan kendali servo, perancangan wadah air minum dan perancangan Tampilan Liquid Crystal Display (LCD) 20x4-I2C menjadi suatu kesatuan sistem. Tampilan dari perancangan keseluruhan sistem dapat dilihat pada Gambar 4.14.

**Gambar 4.14** Perancangan Keseluruhan Sistem

4.4. Perancangan Perangkat Lunak

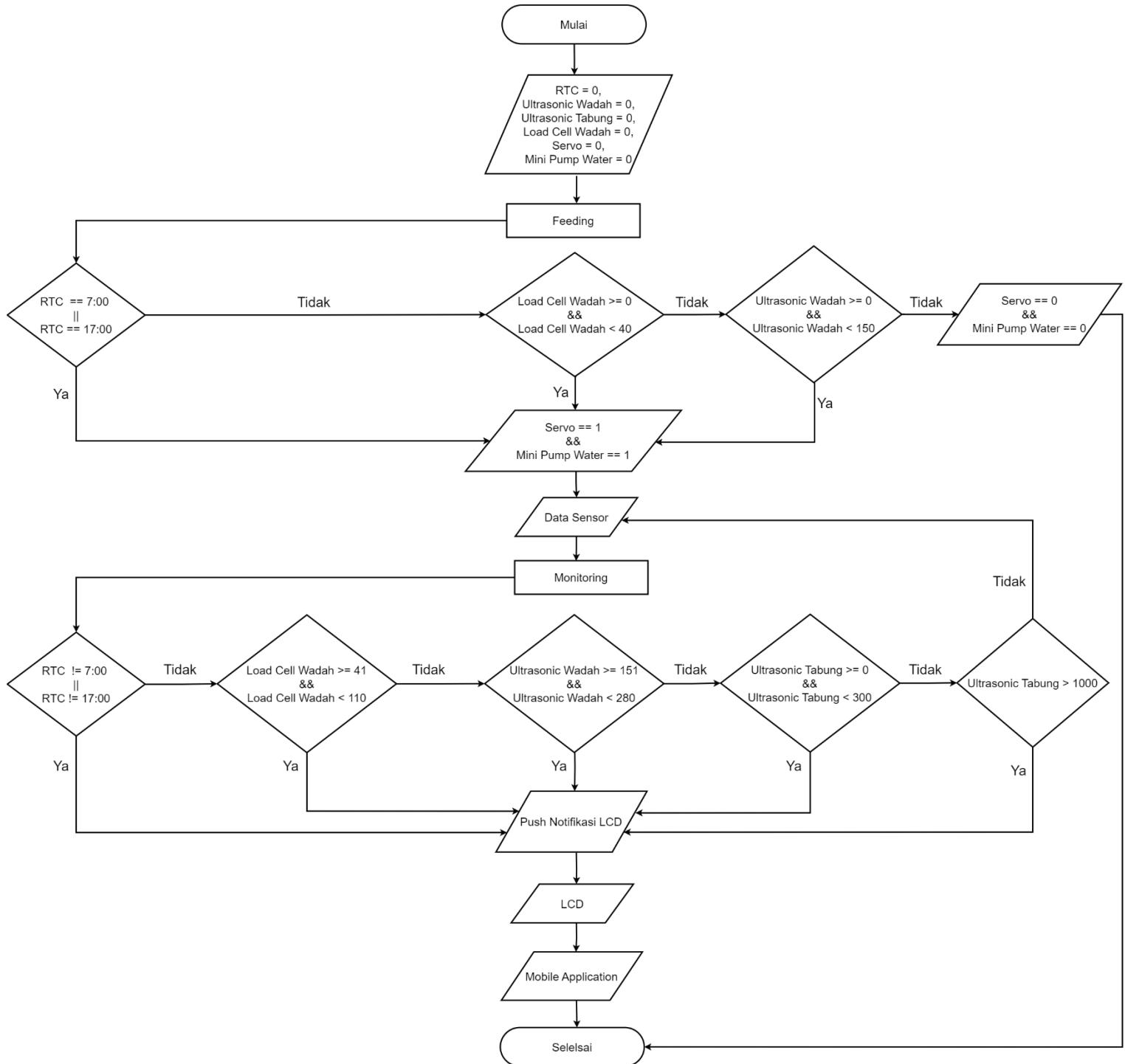
Perancangan perangkat lunak ini merupakan langkah krusial dalam memastikan seluruh komponen perangkat keras dapat beroperasi secara sinkron sesuai alur kerja yang telah ditentukan. Pada penelitian ini, perangkat lunak yang dikembangkan memulai prosesnya dengan menginisialisasi berbagai komponen yang terhubung dengan Arduino Uno R3 *Built-in IoT WiFi ESP8266*. Komponen tersebut meliputi modul *Real Time Clock* (RTC) untuk penjadwalan waktu, sensor *ultrasonic* yang digunakan untuk mengukur ketinggian air pada wadah dan tabung penampungan, sensor *load cell* untuk mengukur berat makanan pada wadah, serta servo yang berfungsi mengontrol pembukaan katup makanan. Selain itu, modul relay digunakan sebagai saklar untuk menghidupkan dan mematikan *mini pump water* yang bertugas mengalirkan air dari tabung ke wadah minum, dan *Liquid Crystal Display* (LCD) digunakan untuk menampilkan data hasil pembacaan sensor. Semua data sensor juga disimpan ke dalam database guna keperluan monitoring dan kendali sistem.

Proses dimulai dengan pembacaan waktu dari modul RTC untuk menentukan waktu pemberian makan dan minum (*feeding*). Jika waktu menunjukkan pukul 07:00 pagi atau 17:00 sore, sistem akan memulai proses pengaktifan servo untuk membuka katup makanan dan mengaktifkan relay untuk menyalaikan *mini pump water* guna mengisi wadah minum. Dalam proses ini, sensor load cell akan mengukur berat makanan di wadah makan, sementara sensor Ultrasonic mengukur ketinggian air di wadah minum. Apabila berat makanan berkisar antara 0 hingga 40 gram, serta ketinggian air di wadah minum berada antara 0 hingga 150 mililiter, servo dan *mini pump water* akan terus aktif hingga kondisi ini berubah. Jika berat makanan melebihi 40 gram atau ketinggian air di wadah mencapai lebih dari 150 mililiter, sistem akan mematikan servo dan *mini pump water* untuk menghentikan pengisian.

Setelah pemberian makan dan minum, sistem akan memasuki mode monitoring dengan terus membaca data dari sensor-sensor yang terhubung. Dalam mode ini, apabila waktu pada RTC tidak menunjukkan pukul 07:00 pagi atau 17:00 sore, sistem akan menampilkan push notifikasi pada LCD yang menginformasikan status pengisian wadah makanan dan minum. Monitoring juga mencakup

pengukuran ulang berat makanan menggunakan *load cell*, serta ketinggian air pada wadah minum dan tabung penampungan air dengan sensor *ultrasonic*. Jika berat makanan berada antara 41 hingga 110 gram dan volume air di wadah minum berada antara 151 hingga 280 mililiter, sistem akan berada dalam mode pasif tanpa melakukan pengisian ulang. Selain itu, sistem juga memantau ketinggian air pada tabung penampungan, jika volume air di tabung berada dalam kisaran 300 hingga 1000 mililiter, kondisi dianggap aman dan tidak diperlukan pengisian ulang.

Setelah seluruh proses pembacaan data sensor selesai, sistem akan menampilkan data hasil pengukuran pada LCD serta mengirimkannya ke *mobile application* untuk memantau status makanan dan minuman kucing secara *real-time*. Diagram alir lengkap dari sistem ini dapat dilihat pada Gambar 4.15.



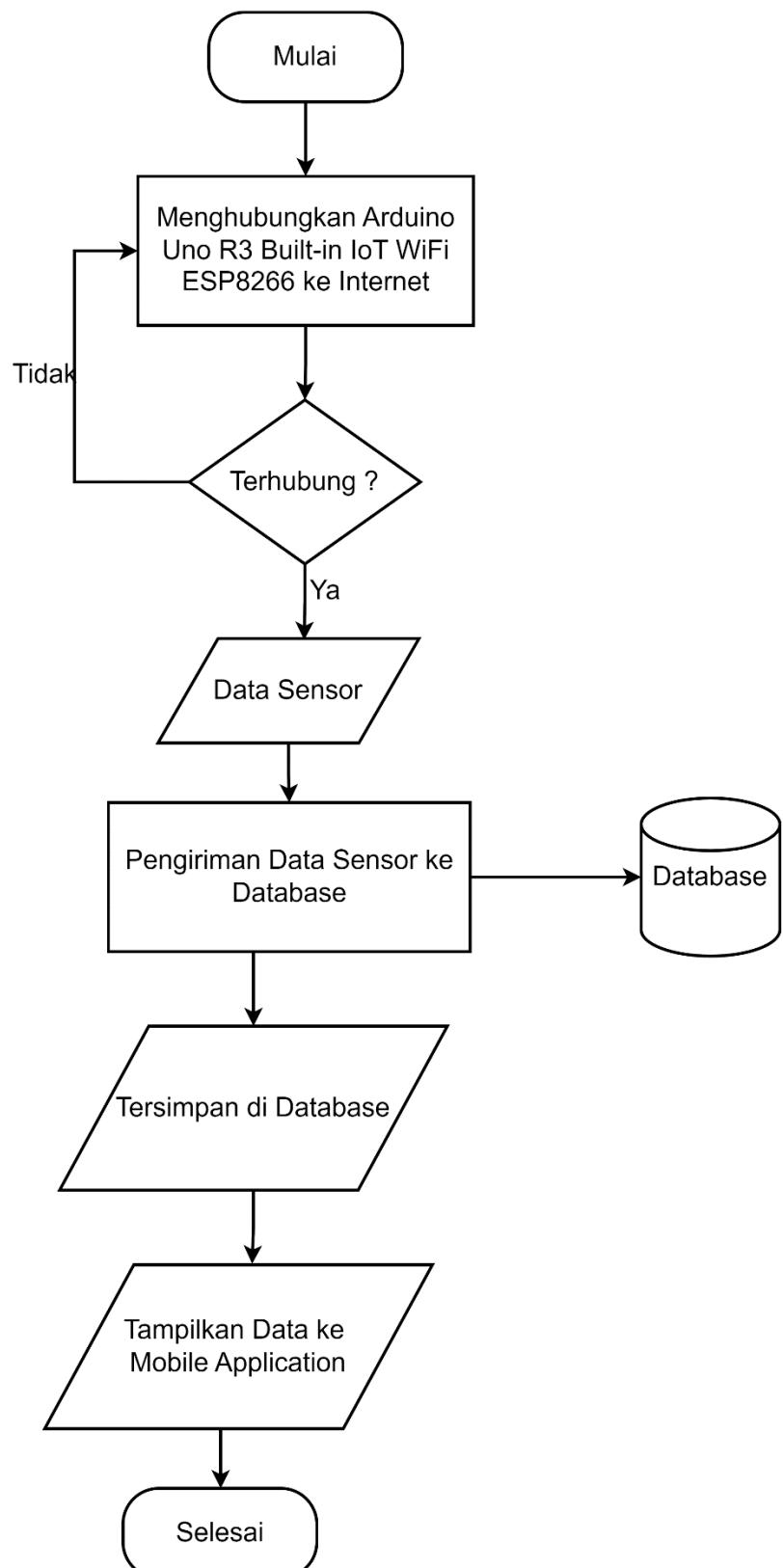
Gambar 4.15 Diagram Alir Keseluruhan

4.4.1 Perancangan perangkat Lunak pada Arduino Uno R3 Built-in IoT WiFi ESP8266

Pada sistem ini, Arduino Uno R3 dengan built-in IoT WiFi ESP8266 digunakan sebagai penghubung utama antara perangkat dan internet untuk memungkinkan pengiriman data sensor ke Firebase Realtime Database. Proses diawali dengan inisialisasi perangkat Arduino, di mana modul WiFi ESP8266 mencoba melakukan koneksi ke jaringan WiFi yang telah dikonfigurasi sebelumnya.

Setelah koneksi internet berhasil, Arduino akan melanjutkan untuk melakukan pembacaan data dari sensor-sensor yang terhubung, seperti sensor ultrasonik, load cell, dan modul RTC. Data sensor yang terkumpul kemudian diproses dan dikirimkan ke Firebase Realtime Database melalui protokol internet. Firebase menyimpan data tersebut secara real-time, yang kemudian dapat diakses oleh pengguna untuk kendali dan otomatisasi pemberian makan dan minum kucing pada *mobile application*.

Alur kerja ini memastikan pengumpulan dan pengiriman data sensor secara efisien, serta memungkinkan sistem untuk terus memantau status koneksi internet dan kondisi sensor. Jika terjadi gangguan pada koneksi, sistem akan berupaya melakukan koneksi ulang secara otomatis. Adapun diagram alir Arduino Uno R3 dengan built-in IoT WiFi ESP8266 untuk pembacaan sensor dapat dilihat pada Gambar 4.16.



Gambar 4.16 Diagram Alir Arduino Uno R3 dengan built-in IoT WiFi ESP8266 Untuk Pengiriman Data Sensor

4.4.2 Perancangan Antarmuka Mobile Application

Tahapan ini merupakan perancangan antar muka pada *mobile application*.

1. Antarmuka Halaman *Login*

Halaman *login* digunakan untuk masuk pada aplikasi. Pada halaman ini terdapat dua *form* isian untuk memasukkan *email* dan *password* yang harus diisi sesuai dengan data akun yang ada di *database firebase*. Ketika data *email* dan *password* yang dimasukkan benar, maka akan dapat masuk ke dalam menu utama aplikasi, sebaliknya jika *email* dan *password* yang dimasukkan salah atau tidak benar, maka tidak akan bisa masuk ke aplikasi. Pada bagian bawah terdapat tautan *Lupa Password* yang memudahkan pengguna untuk memulihkan akses jika pengguna lupa kata sandi. Rancangan halaman *login* dapat dilihat pada Gambar 4.17.



Gambar 4.17 Halaman *Login*

2. Antarmuka Halaman Lupa *Password*

Halaman Lupa *Password* dirancang untuk memudahkan pengguna dalam memulihkan akses ke akun pengguna. Pada halaman ini terdapat *form* untuk memasukkan alamat email yang sudah terdaftar pada *firebase database*, di mana nantinya tautan untuk mereset kata sandi akan dikirimkan. Tombol "Reset Password" untuk memastikan kemudahan navigasi bagi pengguna yang ingin memulihkan kata sandi. Rancangan halaman login dapat dilihat pada Gambar 4.18.



Gambar 4.18 Halaman Lupa Password

3. Antarmuka Halaman *Home*

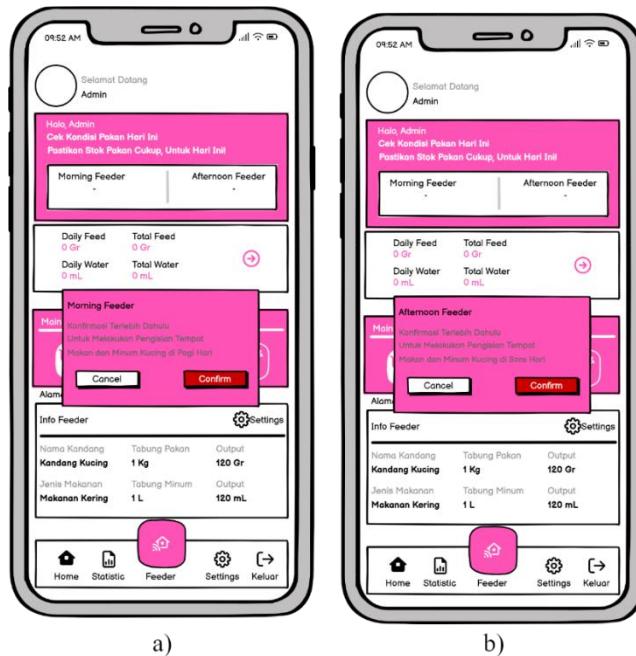
Halaman *Home* berfungsi sebagai pusat navigasi utama yang menyediakan akses cepat dan mudah ke berbagai fitur inti aplikasi. Pada halaman ini berisikan informasi penjadwalan *real time clock* DS3231 terkini terkait status pemberian makan dan minum kucing. Selain itu, terdapat juga pembacaan nilai sensor *load cell* untuk mengukur berat pada wadah makan dan ultrasonik untuk mengukur ketinggian air pada tabung dan wadah minum. Pada bagian bawah terdapat tombol navigasi untuk memudahkan pengguna untuk akses ke halaman lain, seperti *statistic*, *feeder*, *settings* dan keluar. Rancangan halaman *Home* dapat dilihat pada Gambar 4.19.



Gambar 4.19 Halaman *Home*

4. Antarmuka *Form Feeder*

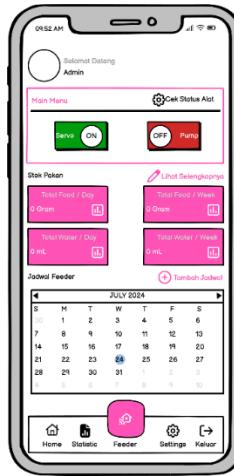
Halaman *feeder* dirancang untuk memberikan kendali pengguna untuk melakukan penjadwalan pemberian makan dan minum kucing secara otomatis. Pada halaman ini, pengguna dapat mengatur *feeder* yaitu pagi hari dan sore hari. Rancangan halaman *feeder* dapat dilihat pada Gambar 4.20.



Gambar 4.20 a) *Form Feeder* pagi, b) *Form Feeder* sore

5. Antarmuka Halaman *Statistic*

Halaman *statistic* berfungsi untuk memberikan informasi terperinci tentang pola konsumsi makanan dan air kucing secara periodik. Pada halaman ini, pengguna dapat melihat total jumlah makanan yang telah diberikan per hari dan per minggu, serta total volume air yang telah dikonsumsi dalam periode yang sama. Tampilan ini memudahkan pengguna untuk memantau kesehatan dan kebiasaan makan kucing mereka serta kontrol manual maupun otomatis untuk pemberian makan dan minum kucing. Selain itu, halaman ini dilengkapi dengan tampilan kalender yang memudahkan pengguna untuk melihat dan menyesuaikan jadwal pemberian makan dan minum kucing. Fitur Tambah Jadwal memungkinkan pengguna untuk menambahkan atau mengubah jadwal makan dan minum kucing. Rancangan halaman *statistic* dapat dilihat pada Gambar 4.21.



Gambar 4.21 Halaman *statistic*

6. Antarmuka Halaman Detail Data

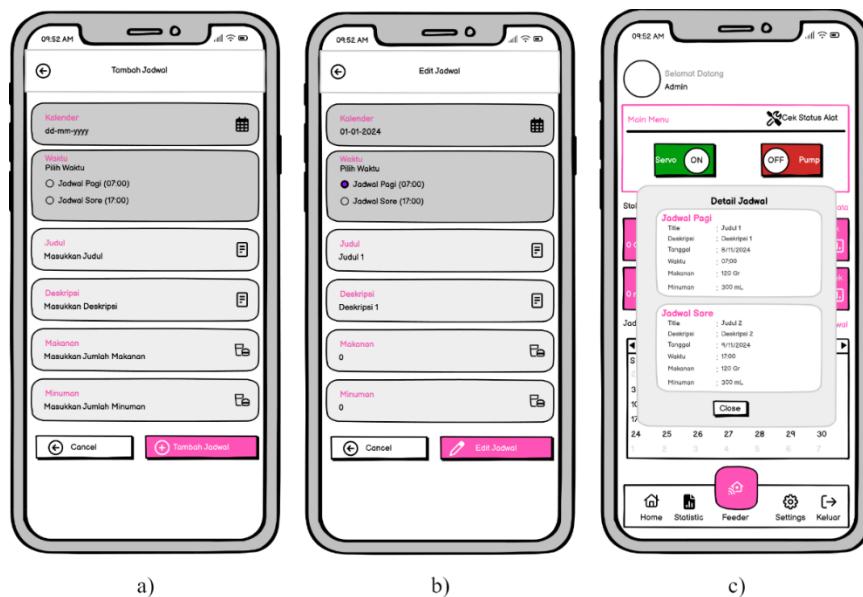
Halaman Detail Data berfungsi untuk menampilkan analisis mendalam mengenai konsumsi makanan dan air kucing secara visual. Pada bagian atas halaman ini, terdapat grafik batang yang menunjukkan data penjadwalan harian yang di input oleh *user* secara *manual* pada *form* tambah jadwal. Selain itu, halaman ini juga menampilkan diagram lingkaran yang menggambarkan persentase pada data sensor *ultrasonic* dan *load cell* untuk mengetahui pengukuran ketinggian air dan berat pakan secara real-time. Pada halaman detail data juga terdapat grafik batang untuk menampilkan data feeder harian yang dilakukan oleh sistem yaitu pagi dan sore hari. Grafik ini membantu pengguna memahami pola konsumsi kucing secara lebih mendetail. Rancangan halaman Detail Data dapat dilihat pada Gambar 4.22.



Gambar 4.22 a) Chart Penjadwalan dan Diagram Monitoring, b) Chart Feeder

7. Antarmuka Halaman Penjadwalan

Halaman "Penjadwalan" memungkinkan pengguna untuk menambahkan, memperbarui dan melihat detail jadwal pemberian makan dan minum kucing. Pada halaman ini, pengguna dapat menentukan tanggal dan waktu pemberian pakan, jumlah pakan yang akan diberikan. Desain antarmuka yang sederhana dan user-friendly memudahkan pengguna dalam mengatur jadwal yang sesuai dengan kebutuhan kucing mereka. Setelah semua informasi diisi, pengguna dapat menyimpan jadwal dengan menekan tombol "Simpan," yang kemudian akan mengaktifkan pengaturan otomatis sesuai dengan jadwal yang telah ditentukan. Rancangan halaman tambah jadwal dapat dilihat pada Gambar 4.23.



Gambar 4.23 a) Form Tambah Jadwal, b) Form Edit Jadwal, c) Detail Jadwal

8. Antarmuka Halaman Data Penjadwalan

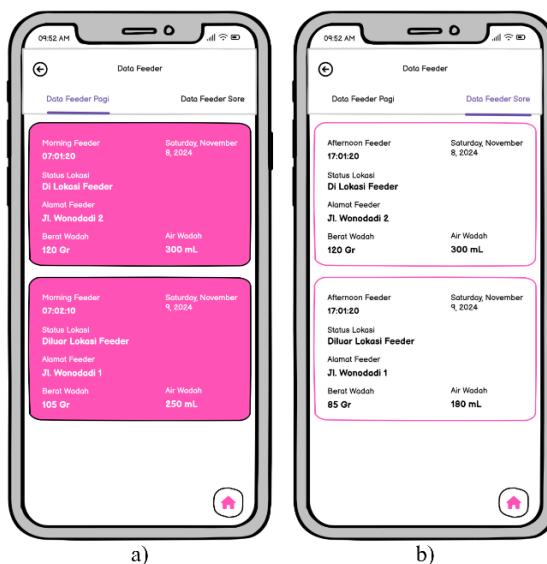
Halaman "Data Penjadwalan" menampilkan jadwal pemberian makan dan minum kucing yang telah diatur untuk pagi dan sore hari. Pada halaman ini, pengguna dapat melihat informasi tanggal, waktu, jumlah makanan, dan jumlah minuman yang telah dijadwalkan. Terdapat pula opsi untuk mengedit atau menghapus jadwal yang sudah ada, serta tombol untuk menambahkan jadwal baru. Antarmuka ini dirancang untuk memberikan kemudahan dalam mengelola jadwal pemberian makan dan minum secara efisien dan tepat waktu. Rancangan halaman data penjadwalan dapat dilihat pada Gambar 4.24.



Gambar 4.24 a) Halaman Data Jadwal Pagi, b) Halaman Data Jadwal Sore

9. Antarmuka Halaman Data Feeder

Halaman “Data Feeder” menampilkan data pemberian makan dan minum kucing pada *form feeder* untuk pagi dan sore hari. Pengguna dapat melihat informasi terkait tanggal, waktu, lokasi, status lokasi, jumlah makanan, dan jumlah minuman. Desain antarmuka ini dibuat untuk memudahkan pengguna dalam mengelola pemberian makan dan minum kucing di pagi dan sore hari dengan lebih teratur dan efisien. Rancangan halaman data *feeder* dapat dilihat pada Gambar 4.25.



Gambar 4.25 a) Halaman Data Feeder Pagi, b) Halaman Data Feeder Sore

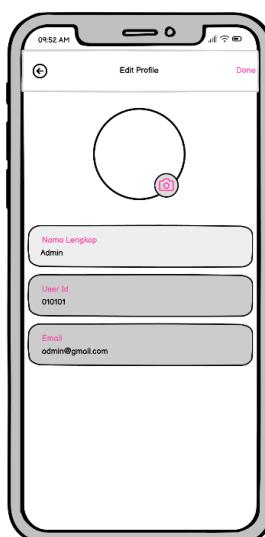
10. Antarmuka Halaman Setting

Halaman "Setting" dirancang untuk memberikan akses kepada pengguna dalam mengelola berbagai pengaturan terkait akun dan perangkat pada aplikasi. Pada halaman ini, pengguna dapat memperbarui profil, memeriksa status alat, mengubah kata sandi, melihat riwayat pemberian makan dan minum kucing, serta melakukan logout dari aplikasi. Rancangan halaman setting dapat dilihat pada Gambar 4.26.



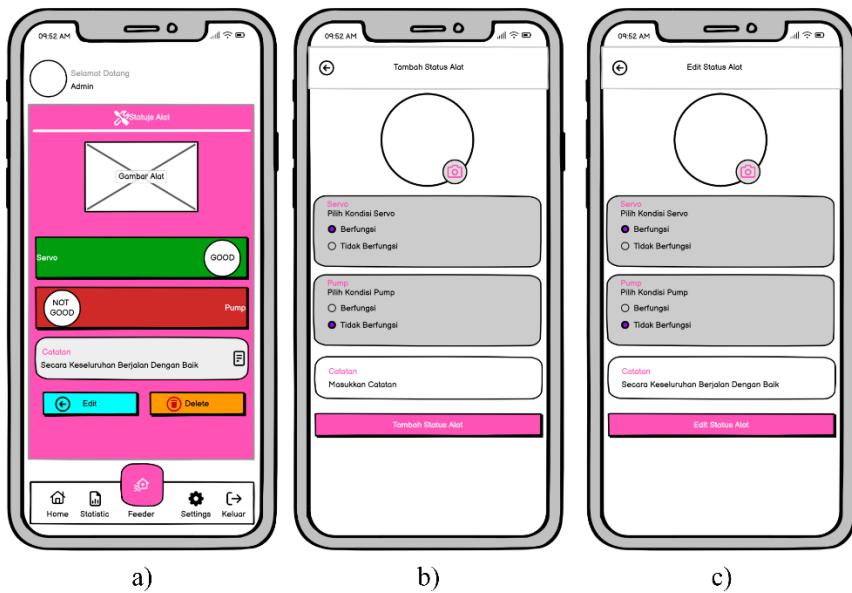
Gambar 4.26 Halaman Setting

Halaman "*Update Profile*" dirancang untuk memungkinkan *user* untuk memperbarui informasi pribadi yang sudah terdaftar pada database. Pada halaman ini, pengguna dapat mengubah nama lengkap dan foto profile. Rancangan halaman *Update Profile* dapat dilihat pada Gambar 4.27.



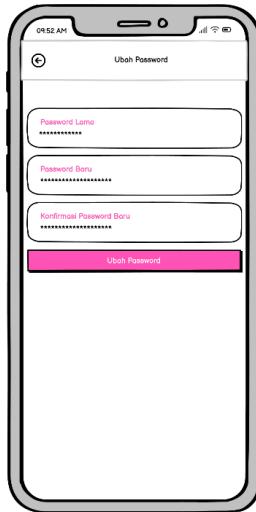
Gambar 4.27 Halaman *Update Profile*

Halaman "Status Alat" dirancang untuk membantu user mengelola perangkat masih berfungsi atau tidak. Pada halaman ini, pengguna dapat memantau status perangkat pemberi makan dan minum kucing, seperti kondisi dan status operasional mini pump water dan servo yang telah diinput oleh user pada form tambah status alat. Untuk memperbarui kondisi dan status operasional perangkat user dapat menggunakan form edit status alat. Informasi ini membantu pengguna memastikan bahwa alat berfungsi dengan baik dan dapat mengambil tindakan yang diperlukan jika terjadi masalah. Rancangan halaman status alat dapat dilihat pada Gambar 4.28.



Gambar 4.28 a) Status alat, b) *Form Tambah status alat*, c) *Form Edit status alat*

Halaman "Ubah Password" dirancang untuk memungkinkan pengguna mengganti kata sandi dengan mudah. Halaman ini terdiri dari tiga kolom input yang masing-masing dilabeli dengan "Password Lama", "Password Baru", dan "Konfirmasi Password Baru". Pengguna harus memasukkan kata sandi lama mereka untuk verifikasi, kemudian memasukkan kata sandi baru yang diinginkan dan mengonfirmasinya dengan mengisi kolom "Konfirmasi Password Baru". Setelah semua kolom terisi dengan benar, pengguna dapat menekan tombol "Ubah Password" yang berwarna merah muda untuk menyimpan perubahan. Desain halaman ini dibuat sederhana dan intuitif untuk memastikan kemudahan penggunaan serta keamanan bagi pengguna. Rancangan halaman ubah password dapat dilihat pada Gambar 4.29.



Gambar 4.29 Halaman Ubah Password

4.5. Perancangan Basis Data

Dalam penelitian ini, basis data yang digunakan adalah *Firebase* untuk menyimpan data secara *real-time*. Adapun data yang akan di simpan pada *database* yaitu, informasi tentang penjadwalan pemberian makan dan minum kucing, data pengguna aplikasi, data sensor, dan kendali servo dan mini pump water. Struktur basis data ini mencakup tabel-tabel seperti tabel *Users Data*, tabel *Users Profile*, tabel *iot*, dan tabel manual.

Tabel *User Data* bertugas sebagai tabel utama yang menyimpan data pengguna. Setiap pengguna memiliki UID (*Unique Identifier*) yang berfungsi sebagai kunci utama. Di dalam tabel ini, terdapat sub-tabel yang menyimpan detail profil pengguna (*Users Profile*), serta pengaturan terkait otomatisasi *Internet of Things* (*iot*), pengaturan penjadwalan pemberian makan dan minum kucing secara otomatis dan manual serta status alat. Tabel *Users Profile* terdiri dari UID, address, avatar, created_at, email, password, name, position, role. Tabel *Data UsersProfile* dapat dilihat pada tabel 4.10.

Tabel 4.10 Tabel *Users Data*

No	Nama	Type	Keterangan
1.	UID	<i>String</i>	<i>Primary Key</i>
2.	<i>UsersProfile</i>	<i>String</i>	Berisi informasi pengguna
3.	<i>Iot</i>	<i>String</i>	Berisi informasi data otomatisasi <i>internet of things</i>
4.	Penjadwalan	<i>String</i>	Berisi informasi data penjadwalan

Tabel 4.10 (Lanjutan)

5	manual	<i>String</i>	Berisi informasi data pemberian makan dan minum kucing secara manual
6.	<i>statusAlat</i>	<i>String</i>	Berisi informasi data status alat

Tabel *Users Profile* digunakan untuk menyimpan data profil pengguna dalam aplikasi yang digunakan untuk mengidentifikasi setiap pengguna secara individual dalam sistem dan digunakan sebagai kredensial login ke dalam aplikasi. Tabel *Users Profile* terdiri dari *UID*, *address*, *avatar*, *created_at*, *email*, *password*, *name*, *position*, *role*. Tabel Data *UsersProfile* dapat dilihat pada tabel 4.11.

Tabel 4.11 Tabel *Users Profile*

No	Nama	Type	Keterangan
1.	<i>UID</i>	<i>String</i>	<i>Primary Key</i>
2.	<i>address</i>	<i>String</i>	Berisi informasi alamat lengkap pengguna rincian lokasi pengguna seperti jalan, kecamatan, dan daerah.
3.	<i>Avatar</i>	<i>String</i>	Berisi informasi penyimpanan gambar <i>profile</i> pengguna.
4.	<i>Email</i>	<i>String</i>	Berisi informasi menyimpan alamat email pengguna yang digunakan sebagai kredensial utama untuk masuk ke dalam sistem.
5.	<i>password</i>	<i>String</i>	Berisi informasi menyimpan kata sandi pengguna yang digunakan otentikasi saat login.
6.	<i>Name</i>	<i>String</i>	Berisi informasi menyimpan nama pengguna yang digunakan untuk menampilkan identitas pengguna dalam antarmuka aplikasi.
7.	<i>position</i>	<i>String</i>	Berisi informasi menyimpan informasi mengenai detail koordinat lokasi pengguna berupa <i>latitude</i> dan <i>longitude</i> .
8.	<i>created_at</i>	<i>String</i>	Berisi informasi tanggal dan waktu pembuatan akun pengguna.
9.	<i>Role</i>	<i>String</i>	Berisi informasi menyimpan peran atau tingkat akses yang dimiliki pengguna dalam sistem.

Tabel iot digunakan untuk menyimpan riwayat data pengukuran dari sensor Ultrasonic HC-SR04, *Load Cell* dan *HX711*, *Real Time Clock DS3231* dan untuk mengendalikan aktuator, yaitu Mini Pump Water dan Servo. Pengendalian aktuator ini dikirimkan ke Arduino Uno R3 dengan Built-in IoT WiFi ESP8266, yang bertugas untuk mengatur mekanisme pemberian makan dan minum pada sistem otomatisasi untuk pemberian makan dan minum kucing. Jika Arduino menerima sinyal untuk mengaktifkan aktuator, sistem akan memicu aksi yang sesuai, seperti mengalirkan air atau menggerakkan servo untuk membuka wadah makanan dan minuman. Sebaliknya, jika tidak ada sinyal pengendalian, sistem tetap dalam keadaan hidup. Tabel iot terdiri dari UID, control, pumpControl, servoControl, monitoring, beratWadah, ketHari, ketWaktu, pumpStatus, servoStatus, volumeMLTabung, volumeMLWadah. Tabel Data iot dapat dilihat pada tabel 4.12.

Tabel 4.12 Tabel iot

No	Nama	Type	Keterangan
1.	UID	<i>String</i>	<i>Primary Key</i>
2.	<i>control</i>	<i>String</i>	Berisi informasi kendali pump dan servo.
3.	<i>pumpControl</i>	<i>Bool</i>	Berisi informasi kendali pump.
4.	<i>servoControl</i>	<i>Bool</i>	Berisi informasi kendali servo.
5.	<i>monitoring</i>	<i>String</i>	Berisi informasi monitoring makan dan minum kucing
6.	beratWadah	<i>Int</i>	Berisi informasi pembacaan sensor <i>Load Cell</i> dan <i>HX711</i> pada wadah makan kucing.
7.	ketHari	<i>String</i>	Berisi informasi pembacaan tanggal pada sensor <i>Real Time Clock DS3231</i> .
8.	ketWaktu	<i>String</i>	Berisi informasi pembacaan waktu pada sensor <i>Real Time Clock DS3231</i> .
9.	<i>pumpStatus</i>	<i>Bool</i>	Berisi informasi status pump.
10.	<i>servoStatus</i>	<i>Bool</i>	Berisi informasi status servo.
11.	<i>volumeMLTabung</i>	<i>Int</i>	Berisi informasi pembacaan sensor Ultrasonic HC-SR04 pada tabung minum kucing.
12.	<i>volumeMLWadah</i>	<i>int</i>	Berisi informasi pembacaan sensor Ultrasonic HC-SR04 pada wadah minum kucing.

Tabel penjadwalan digunakan untuk menyimpan informasi lengkap mengenai jadwal pemberian makan dan minum kucing, termasuk jadwal pagi dan sore, tanggal, waktu, dan jumlah makanan serta minuman yang diberikan. Setiap jadwal diidentifikasi dengan UID sebagai kunci primer, dengan tambahan judul dan deskripsi untuk memudahkan identifikasi. Tabel ini juga mencatat waktu dan tanggal pembuatan jadwal untuk keperluan pelacakan. Tabel penjadwalan terdiri dari UID, jadwalPagi, jadwalSore, created_at, date, deskripsi, makanan, minuman, tanggal, title, waktu. Tabel Data manual dapat dilihat pada tabel 4.13.

Tabel 4.13 Tabel Penjadwalan

No	Nama	Type	Keterangan
1.	UID	<i>String</i>	<i>Primary Key</i>
2.	jadwalPagi	<i>String</i>	Berisi informasi pemberian makan dan minum kucing pada pagi hari.
3.	jadwalSore	<i>String</i>	Berisi informasi pemberian makan dan minum kucing pada sore hari
4.	tanggal	<i>String</i>	Berisi informasi menyimpan tanggal pemberian makan dan minum kucing. Ini penting untuk penjadwalan otomatis berdasarkan tanggal yang ditetapkan oleh pengguna.
5.	Waktu	<i>String</i>	Berisi informasi menyimpan waktu pemberian makan dan minum kucing. Ini penting untuk penjadwalan otomatis berdasarkan waktu yang ditetapkan oleh pengguna.
6.	Title	<i>String</i>	Berisi informasi menyimpan judul tentang pemberian makan dan minum kucing. judul atau nama dari pengaturan manual. Judul ini memudahkan identifikasi database penjadwalan makan dan minum kucing.
7.	deskripsi	<i>String</i>	Berisi informasi menyimpan deskripsi singkat atau catatan tentang pemberian makan dan minum kucing.
8.	makanan	<i>String</i>	Berisi informasi jumlah makanan yang diatur untuk diberikan pada waktu yang ditentukan.
9.	minuman	<i>String</i>	Berisi informasi jumlah makanan yang diatur untuk diberikan pada waktu yang ditentukan.
10.	<i>created_at</i>	<i>String</i>	Berisi informasi waktu pembuatan database penjadwalan makan dan minum kucing.

11.	<i>Date</i>	<i>String</i>	Berisi informasi tanggal pembuatan database penjadwalan makan dan minum kucing.
-----	-------------	---------------	---

Tabel Manual digunakan untuk menyimpan data terkait pemberian makan dan minum kucing secara manual oleh pengguna. Tabel manual terdiri dari UID, morningFeeder, afternoonFeeder, alamat, date, *distance*, *in_area*, *latitude*, *longitude*. Tabel Data Feeder dapat dilihat pada tabel 4.12.

Tabel 4.14 Tabel manual

No	Nama	Type	Keterangan
1.	UID	<i>String</i>	<i>Primary Key</i>
2.	<i>morningFeeder</i>	<i>String</i>	Berisi informasi pemberian makan dan minum kucing di pagi hari.
3.	<i>afternoonFeeder</i>	<i>String</i>	Berisi informasi pemberian makan dan minum kucing di sore hari.
4.	Alamat	<i>String</i>	Berisi informasi alamat pengguna melakukan pemberian makan dan minum kucing.
5.	<i>Date</i>	<i>String</i>	Berisi informasi tanggal pengguna melakukan pemberian makan dan minum kucing.
6.	<i>distance</i>	<i>String</i>	Berisi informasi jarak pengguna melakukan pemberian makan dan minum kucing.
8.	<i>in_area</i>	<i>String</i>	Berisi informasi area pemberian makan dan minum kucing.
9.	<i>latitude</i>	<i>String</i>	Berisi informasi koordinat <i>latitude</i> alamat detail pemberian makan dan minum kucing.
10.	<i>longitude</i>	<i>String</i>	Berisi informasi koordinat <i>longitude</i> alamat detail pemberian makan dan minum kucing.

Tabel Status Alat digunakan untuk menyimpan informasi terkait status operasional alat makan dan minum kucing. Tabel status alat terdiri dari UID, avatar, Catatan, pump_status, servo_status. Tabel Data status alat dapat dilihat pada tabel 4.15.

Tabel 4.15 Tabel Status Alat

No	Nama	Type	Keterangan
1.	UID	<i>String</i>	<i>Primary Key</i>
2.	<i>Avatar</i>	<i>String</i>	Berisi informasi penyimpanan gambar perangkat pemberian makan dan minum kucing.
3.	Catatan	<i>String</i>	Berisi informasi catatan tentang alat makan dan minum kucing.
4.	<i>pump_status</i>	<i>Bool</i>	Berisi informasi kondisi tentang status operasional pump
5.	<i>servo_status</i>	<i>Bool</i>	Berisi informasi kondisi tentang status operasional servo.

4.6. Perancangan Pengujian *Black Box*

Pengujian menggunakan metode black box bertujuan untuk mengevaluasi apakah elemen-elemen dalam antarmuka sistem berfungsi sesuai dengan yang diharapkan. Fokus utama pengujian ini adalah pada interaksi pengguna dengan antarmuka, memastikan bahwa setiap fitur, seperti form input, tombol, dan navigasi, beroperasi dengan benar. Selain itu, pengujian ini juga bertujuan untuk mengidentifikasi kesalahan atau kegagalan yang mungkin terjadi selama penggunaan. Rancangan pengujian black box dapat dilihat pada Tabel 4.16, yang mencakup berbagai halaman seperti halaman login, home, feeder, dan lainnya, dengan skenario pengujian spesifik dan hasil yang diharapkan untuk setiap interaksi pengguna. Berikut rancangan pengujian black box yang terdapat pada Tabel 4.16.

Tabel 4. 16 Tabel Perancangan Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan
1.	<i>Login</i>	Menu <i>Login</i>	Menampilkan halaman <i>login</i>
		<i>Form Email</i>	Pengguna bisa memasukkan alamat <i>email</i> pada <i>form email</i>
		<i>Form Password</i>	Pengguna bisa memasukkan <i>password</i> pada <i>form password</i>

Tabel 4. 16 Tabel Perancangan Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan
		Memasukkan <i>email</i> dan <i>password</i> yang sudah terdaftar, lalu menekan tombol login	Sistem menampilkan halaman home dan notifikasi login sukses.
		Memasukkan <i>email</i> dan <i>password</i> yang belum terdaftar, lalu menekan tombol masuk dan memunculkan modal untuk status <i>login</i>	Sistem memeriksa email dan password, jika belum terdaftar maka muncul modal untuk memasukkan <i>email</i> dan <i>password</i> tidak terdaftar.
		Tombol Login	Sistem melakukan verifikasi <i>email</i> dan <i>password</i>
		Tombol Lupa Password	Sistem menampilkan halaman lupa <i>password</i>
2.	Lupa Password	Menu Lupa Password	Menampilkan halaman lupa <i>password</i>
		Form Email	Pengguna bisa memasukkan alamat <i>email</i> yang sudah terdaftar pada form <i>email</i>
		Memasukkan email yang sudah terdaftar, lalu menekan tombol <i>reset password</i> .	Sistem mengirimkan form ke alamat email pengguna yang sudah terdaftar untuk mengubah <i>password</i> yang lama menjadi <i>password</i> yang baru.

Tabel 4. 16 Tabel Perancangan Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	
	Home	Memasukkan email yang belum terdaftar, lalu menekan tombol <i>reset password</i>	Sistem memeriksa alamat <i>email</i> , jika belum terdaftar maka muncul modal untuk memasukkan alamat <i>email</i> yang sudah terdaftar	
		Tombol <i>Reset Password</i>	Sistem melakukan <i>verifikasi email</i> dan <i>password</i> .	
3.		Menu Home	Menampilkan halaman home	
		Menampilkan status <i>morning feeder</i> dan <i>afternoon feeder</i>	Menampilkan status pemberian makan dan minum kucing pada penjadwalan pagi dan sore hari.	
		Nilai kondisi penjadwalan pemberian jumlah makanan dan minuman secara harian.	Menampilkan jumlah pemberian makanan dan minuman harian secara real-time.	
		Menampilkan alamat feeder	Menampilkan alamat pemberian makan dan minum kucing.	
4.	Feeder	Menu feeder	Menampilkan halaman feeder	
		Menampilkan kendali <i>morning feeder</i>	Menampilkan halaman otomatisasi pemberian makan dan minum kucing pada pagi hari.	
		Menampilkan kendali <i>afternoon feeder</i>	Menampilkan halaman otomatisasi pemberian makan dan minum kucing pada sore hari.	
5.	Main	Menu Main	Menampilkan halaman main	

Tabel 4. 16 Tabel Perancangan Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan
		Tombol Automatic	Menampilkan pilihan mode untuk mengaktifkan pemberian makan dan minum kucing secara otomatis
		Tombol on/off servo	Mengendalikan on/off servo untuk melakukan pemberian makan pada wadah makanan kucing.
		Tombol on/off pump	Mengendalikan on/off servo untuk melakukan pemberian minum pada wadah minuman kucing.
		Tombol Cek Status Alat	Menampilkan halaman status alat.
		Tombol Log Data	Menampilkan halaman log data.
		Menampilkan jumlah data sensor	Menampilkan nilai data sensor untuk monitoring pemberian makan dan minum kucing secara harian dan mingguan.
		Tombol Tambah Jadwal	Menampilkan halaman untuk formulir tambah penjadwalan.
		Menampilkan data penjadwalan pada tabel kalender.	Menampilkan data hasil dari penjadwalan pemberian makan dan minum kucing.
6.	Status Alat	Menu Status Alat	Menampilkan halaman status alat.
		Menampilkan form operasional perangkat pemberian makan dan minum kucing.	Menampilkan informasi status operasional perangkat pemberian makan dan minum kucing
7.	Statistik Data	Menu Log Data	Menampilkan halaman log data
		Menampilkan Data Statistik Penjadwalan	Menampilkan informasi chart data penjadwalan pemberian makan dan minum.

Tabel 4. 16 Tabel Perancangan Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan
8.	List Data	Menampilkan Data IoT	Menampilkan informasi status sensor pada wadah makan dan minum kucing berupa <i>radial bar chart</i> secara <i>realtime</i> .
		Menampilkan Statistik Data Feeder	Menampilkan informasi data feeder dalam bentuk <i>chart</i> secara <i>realtime</i>
		Menu Data Latih	Menampilkan Halaman Data Latih
		Menampilkan Data Jadwal Pagi	Menampilkan Data Penjadwalan Pagi Hari dalam bentuk <i>list</i> data
		Menampilkan Data Jadwal Sore	Menampilkan Data Penjadwalan Sore Hari dalam bentuk <i>list</i> data
		Tombol Edit	Menampilkan form untuk mengedit data jadwal
		Tombol Delete	Menampilkan form untuk menghapus data jadwal
		Tombol Tambah	Menampilkan form untuk menambah data jadwal
		Menampilkan Data Feeder Pagi	Menampilkan Data Feeder Pagi dalam bentuk <i>list</i> data
9.	Tambah Jadwal	Menampilkan Data Feeder Sore	Menampilkan Data Feeder Sore dalam bentuk <i>list</i> data
		Menu Tambah Jadwal	Menampilkan form untuk tambah jadwal
		Form Kalendar	Pengguna bisa memilih tanggal untuk penjadwalan

Tabel 4. 16 Tabel Perancangan Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan
		Form Waktu	Pengguna bisa memilih waktu untuk penjadwalan
		Form Judul	Pengguna bisa memasukkan judul untuk penjadwalan
		Form Deskripsi	Pengguna bisa memasukkan deskripsi untuk penjadwalan
		Form Makanan	Pengguna bisa memasukkan jumlah makanan untuk penjadwalan.
		Form Minuman	Pengguna bisa memasukkan jumlah minuman untuk penjadwalan
		Tombol Tambah Jadwal	Sistem menyimpan data penjadwalan
		Tombol Cancel	Sistem membatalkan penambahan jadwal
10.	Edit Jadwal	Menu Edit Jadwal	Menampilkan form untuk mengedit jadwal
		Form Kalendar	Pengguna bisa memilih tanggal untuk mengedit penjadwalan
		Form Waktu	Pengguna bisa memilih waktu untuk mengedit penjadwalan
		Form Judul	Pengguna bisa mengedit judul untuk penjadwalan
		Form Deskripsi	Pengguna bisa mengedit deskripsi untuk penjadwalan

Tabel 4. 16 Tabel Perancangan Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan
11.	<i>Settings</i>	Form Makanan	Pengguna bisa mengedit jumlah makanan untuk penjadwalan
		Form Minuman	Pengguna bisa mengedit jumlah minuman untuk penjadwalan
		Tombol Edit Jadwal	Sistem menyimpan perubahan data penjadwalan
		Tombol Cancel	Sistem membatalkan pengeditan jadwal
		Menu <i>settings</i>	Menampilkan halaman <i>settings</i>
		Tombol <i>update profile</i>	Menampilkan <i>form</i> untuk <i>update profile</i>
		Tombol status alat	Menampilkan <i>form</i> untuk tambah status alat
		Tombol ubah <i>password</i>	Menampilkan <i>form</i> untuk mengubah <i>password</i>
		Tombol history <i>feeder</i>	Menampilkan riwayat aktivitas <i>feeder</i>
		Tombol keluar	Sistem akan mengeluarkan pengguna dari tampilan <i>mobile application</i> dan kembali ke halaman <i>login</i> .

BAB V

IMPLEMENTASI, PENGUJIAN DAN PEMBAHASAN

Pada bab implementasi, pengujian, dan pembahasan menjelaskan tahapan penerapan sistem yang telah dibangun berdasarkan pembahasan yang telah dijelaskan pada bab sebelumnya. Tahapan tersebut meliputi implementasi sistem, kode program, pengujian dan pembahasan.

5.1 Implementasi Sistem

Implementasi sistem adalah tahap di mana perancangan sistem direalisasikan untuk siap dioperasikan. Pada tahap ini, komponen perangkat keras dan perangkat lunak diintegrasikan untuk membentuk sistem yang siap dioperasikan sesuai dengan perancangan. Implementasi perangkat keras mencakup pengaturan Arduino Uno R3 Built-in IoT WiFi ESP8266 untuk mengendalikan Mini Pump Water dan Motor Servo berdasarkan data dari sensor *Load Cell* 5 Kilogram dan HX711 untuk mengukur berat makanan, serta sensor Ultrasonic HC-SR04 untuk mengukur volume air. Selain itu, *Real Time Clock* DS3231 digunakan untuk membaca waktu penjadwalan yang mengaktifkan pump dan servo pada waktu 07.00 pagi dan 17.00 sore. Hasil pembacaan sensor ditampilkan pada LCD 20x4 I2C, dan data dikirimkan ke Firebase real-time database. Sedangkan implementasi perangkat lunak meliputi implementasi antarmuka *mobile application* dan implementasi kode program di perangkat keras.

5.2 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan langkah-langkah penyusunan komponen perangkat keras yang telah dirancang pada bab perancangan. Berikut Implementasi perangkat keras meliputi:

5.2.1 Implementasi Sistem Penjadwalan *Real Time Clock*

Implementasi penjadwalan menggunakan *Real Time Clock* (RTC) pada sistem ini bertujuan untuk mengatur waktu pemberian makan dan minum kucing secara otomatis. Komponen yang digunakan yaitu RTC DS3231 dihubungkan pin A4 dan A5 pada Arduino Uno R3 Built-in IoT WiFi ESP8266 untuk melakukan pembacaan waktu secara *real-time*. Pada saat RTC mencapai waktu yang telah ditentukan, yaitu pukul 07.00 pagi dan 17.00 sore, mini pump water akan aktif

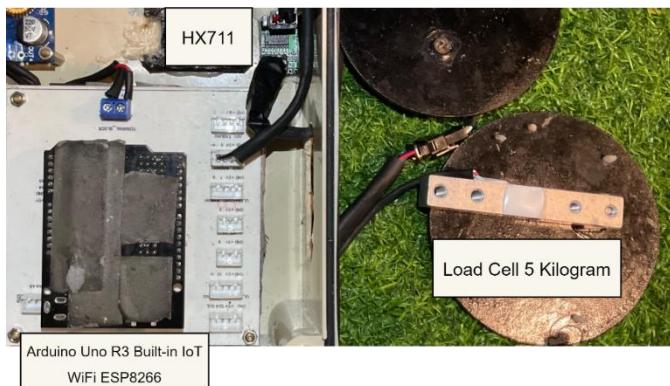
untuk mengisi air pada wadah minum, dan servo akan membuka katup makan untuk mengisi wadah makan kucing. Hasil dari implementasi penjadwalan RTC ini dapat dilihat pada Gambar 5.1.



Gambar 5.1 Implementasi Sistem Penjadwalan *Real Time Clock*

5.2.2 Implementasi Sistem Pengukuran Berat Wadah Makan

Implementasi sistem pengukuran berat wadah makan menggunakan sensor load cell 5 kilogram (kg) dan modul HX711 yang dihubungkan ke pin digital 4 dan 5 pada Arduino Uno R3 Built-in IoT WiFi ESP8266. Pada modul HX711, pin DT terhubung ke digital pin 4, sedangkan pin SCK terhubung ke digital pin 5 pada Arduino. Sensor load cell berfungsi untuk mengukur berat makanan di dalam wadah, sementara modul HX711 bertindak sebagai penguat sinyal dan konverter dari sinyal analog yang dihasilkan oleh load cell menjadi data digital yang dapat dibaca oleh Arduino. Hasil dari implementasi sistem pengukuran berat wadah makan ini dapat dilihat pada Gambar 5.2.

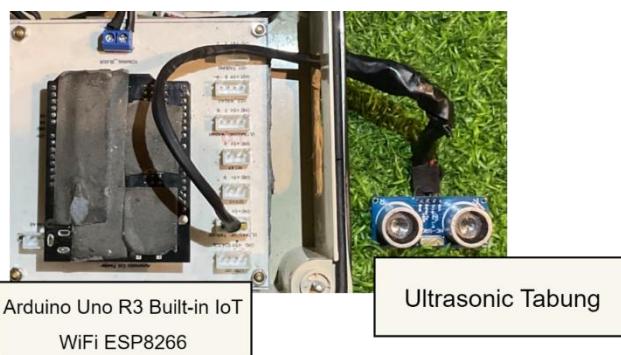


Gambar 5.2 Implementasi Sistem Pengukuran Berat Wadah Makan

5.2.3 Implementasi Sistem Pengukuran Volume Tabung Air Minum

Implementasi sistem pengukuran volume tabung air minum menggunakan sensor Ultrasonic HC-SR04 yang dihubungkan ke pin digital 6 dan 7 pada Arduino Uno R3 Built-in IoT WiFi ESP8266. Pada sensor Ultrasonic HC-SR04, pin TRIG terhubung ke pin digital 6, sedangkan pin ECHO terhubung ke pin digital 7 pada Arduino. Sensor ini berfungsi untuk mengukur tinggi air dalam tabung minum dengan cara mengirimkan gelombang ultrasonik dari pin TRIG, yang kemudian dipantulkan kembali oleh permukaan air dan diterima oleh pin ECHO. Jarak yang diukur oleh sensor adalah selisih antara tinggi tabung dan jarak air dari sensor.

Untuk mengkonversi hasil pengukuran dari satuan cm ke mililiter, digunakan rumus volume tabung, yaitu dengan menghitung volume berdasarkan tinggi air yang terukur. Mengingat radius tabung adalah 4,25 cm dan tinggi tabung adalah 34,5 cm, volume air dalam tabung dihitung menggunakan rumus $V = \pi \times r^2 \times h$, di mana π (pi) adalah konstanta yang bernilai sekitar 3,14159, r adalah radius tabung (4,25 cm), dan h adalah tinggi air yang terukur dari dasar tabung. Hasil perhitungan volume (V) akan memberikan volume dalam satuan cm^3 , yang setara dengan mililiter (ml). Data hasil pengukuran ini digunakan untuk menentukan volume air dalam tabung dan memberikan notifikasi pada LCD 20x4 jika tinggi air kurang dari 500 mililiter dan lebih dari 1 liter. Hasil dari implementasi sistem pengukuran volume tabung air minum ini dapat dilihat pada Gambar 5.3



Gambar 5.3 Implementasi Sistem Pengukuran Volume Tabung Air Minum

5.2.4 Implementasi Kendali Mini Pump Water

Implementasi kendali Mini Pump Water pada sistem ini menggunakan modul relay 1 channel yang berfungsi sebagai saklar untuk mengontrol nyala atau mati Mini Pump Water. Modul relay ini dihubungkan ke pin digital 8 pada Arduino Uno

R3 Built-in IoT WiFi ESP8266. Arduino berfungsi sebagai pengendali utama yang mengirimkan sinyal ke relay. Ketika Arduino mengirimkan sinyal 'HIGH' ke pin relay, relay akan aktif, sehingga Mini Pump Water menyala dan mulai mengalirkan air ke wadah minum. Sebaliknya, jika sinyal yang dikirimkan adalah 'LOW', relay akan mati dan Mini Pump Water akan berhenti bekerja.

Kendali Mini Pump Water ini juga berfungsi otomatis berdasarkan dua kondisi, yaitu jadwal yang diatur melalui RTC DS3231 pada pukul 07.00 pagi dan 17.00 sore, serta kondisi monitoring volume air di tabung minum. Jika sensor ultrasonik mendeteksi volume air lebih dari 0 dan kurang dari 40 mililiter, maka relay akan otomatis menyala untuk mengaktifkan Mini Pump Water, sehingga air dapat ditambahkan ke wadah minum hingga volume mencapai batas aman. Hasil dari implementasi kendali Mini Pump Water ini dapat dilihat pada Gambar 5.4.



Gambar 5.4 Implementasi Kendali Mini Pump Water

5.2.5 Implementasi Kendali Servo

Implementasi servo otomatis pada sistem ini bertujuan untuk mengontrol katup pada tabung pakan kucing secara otomatis. Kendali servo dilakukan melalui Arduino Uno R3 Built-in IoT WiFi ESP8266 yang terhubung ke pin digital 9 untuk mengatur pergerakan servo. Kendali servo otomatis pada sistem ini berfungsi untuk mengontrol katup pada tabung pakan kucing secara otomatis berdasarkan dua kondisi. Pertama, jadwal yang diatur melalui RTC DS3231 pada pukul 07.00 pagi dan 17.00 sore, di mana Arduino Uno R3 Built-in IoT WiFi ESP8266 akan mengirimkan sinyal ke servo untuk membuka katup tabung pakan dan menyalurkan makanan ke wadah makan kucing. Kedua, jika sensor load cell mendeteksi berat pakan dalam wadah lebih dari 0 gram dan kurang dari 40 gram, maka servo akan aktif untuk mengisi ulang pakan secara otomatis. Setelah jumlah pakan yang cukup disalurkan, servo akan kembali ke posisi semula untuk menutup katup. Hasil dari implementasi kendali servo otomatis ini dapat dilihat pada Gambar 5.5.



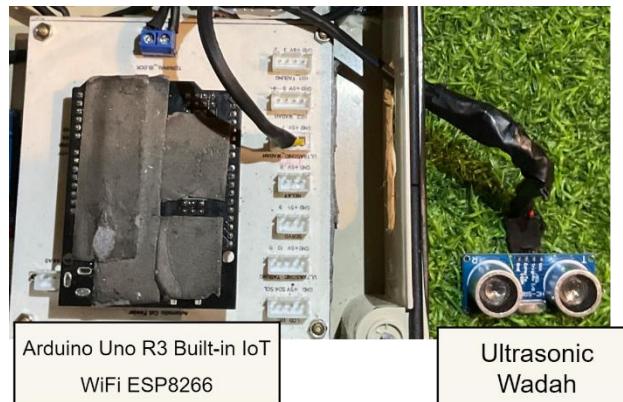
Gambar 5.5 Implementasi Kendali Servo

5.2.6 Implementasi Sistem Pengukuran Volume Wadah Air Minum

Implementasi sistem pengukuran volume wadah air minum dilakukan menggunakan sensor Ultrasonic HC-SR04 yang terhubung ke pin digital 10 dan 11 pada Arduino Uno R3 Built-in IoT WiFi ESP8266. Pada sensor Ultrasonic HC-SR04, pin TRIG terhubung ke pin digital 11, sedangkan pin ECHO terhubung ke pin digital 10. Sensor ini berfungsi untuk mengukur tinggi air dalam wadah minum dengan cara mengirimkan gelombang ultrasonik dari pin TRIG, yang kemudian dipantulkan kembali oleh permukaan air dan diterima oleh pin ECHO. Jarak yang diukur oleh sensor adalah selisih antara tinggi wadah dan jarak air dari sensor.

Untuk mengkonversi hasil pengukuran dari satuan cm ke mililiter, digunakan rumus volume wadah, yaitu dengan menghitung volume berdasarkan tinggi air yang terukur. Mengingat radius wadah adalah 4,5 cm dan tinggi wadah adalah 15,0 cm, volume air dalam wadah dihitung menggunakan rumus $V = \pi \times r^2 \times h$, di mana π (pi) adalah konstanta yang bernilai sekitar 3,14159, r adalah radius tabung (4,5 cm), dan h adalah tinggi air yang terukur dari dasar tabung. Hasil perhitungan volume (V) akan memberikan volume dalam satuan cm^3 , yang setara dengan mililiter (ml).

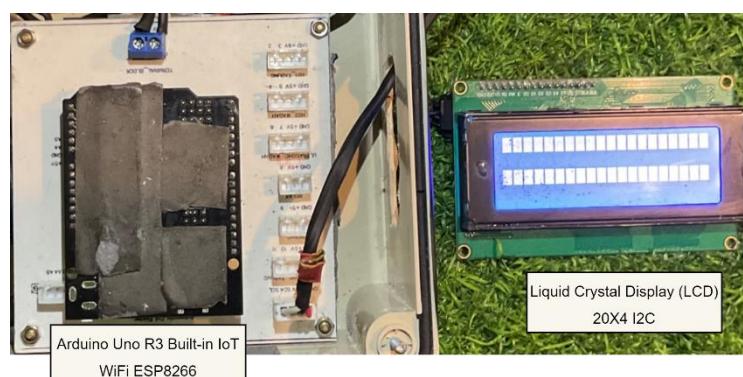
Kondisi pada sistem monitoring ditetapkan yaitu jika sensor ultrasonik mendekripsi volume air kurang dari atau sama dengan 40 mililiter, maka pompa akan menyala untuk mengalirkan air dari tabung minum ke wadah minum. Sementara itu, jika volume air berada di antara 40 mililiter dan 300 mililiter, notifikasi akan muncul pada LCD 20x4 I2C untuk memberikan informasi kepada pengguna. Hasil dari implementasi sistem pengukuran volume wadah air minum ini dapat dilihat pada Gambar 5.6.



Gambar 5.6 Implementasi Sistem Pengukuran Volume Wadah Air Minum

5.2.7 Implementasi Tampilan LCD 20X4 Pembacaan Sensor

Implementasi Liquid Crystal Display (LCD) 20x4-I2C pada sistem ini bertujuan untuk menampilkan hasil pembacaan dari sensor-sensor yang terpasang. LCD terhubung ke Arduino Uno R3 Built-in IoT WiFi ESP8266 melalui pin SDA dan SCL. Tampilan pada LCD 20x4 menampilkan informasi terkait penjadwalan waktu makan dan minum kucing, yaitu pada pukul 07.00 dan 17.00, serta notifikasi dari sensor jarak ultrasonik dan load cell. Notifikasi tersebut akan muncul jika berat wadah makan dan wadah air sudah mulai habis, sehingga pengguna dapat melakukan pengisian ulang. Hasil dari implementasi tampilan sensor pada LCD 20x4 dapat dilihat pada Gambar 5.7.



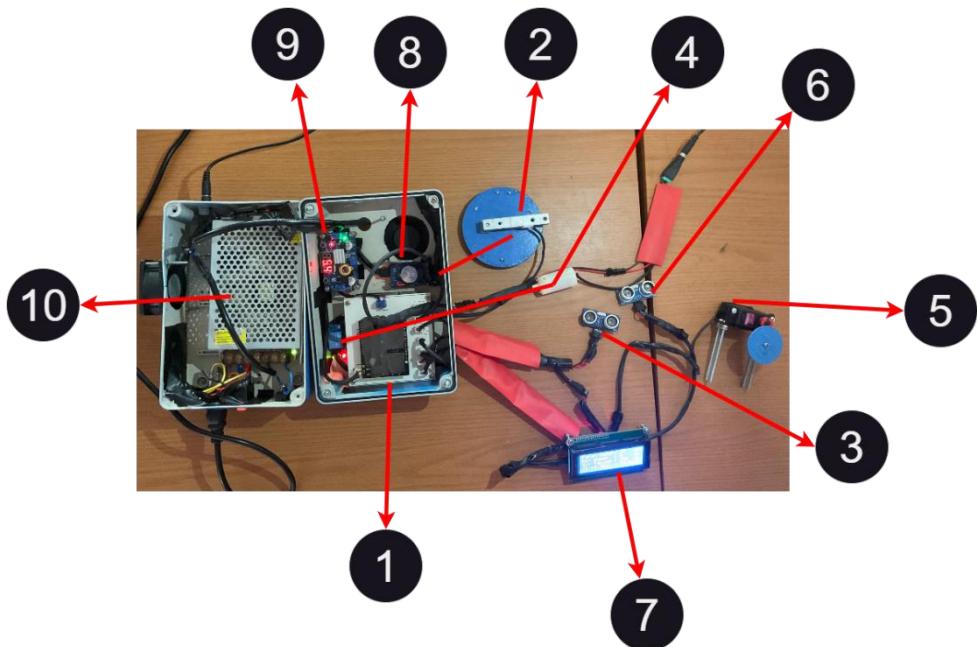
Gambar 5.7 Implementasi Tampilan LCD 20X4 Pembacaan Sensor

5.2.8 Implementasi Keseluruhan Sistem

Implementasi keseluruhan sistem ini menggabungkan berbagai komponen perangkat keras dan perangkat lunak yang saling terintegrasi untuk membentuk sistem otomatisasi pemberian makan dan minum kucing. Komponen utama yang digunakan meliputi Arduino Uno R3 Built-in IoT WiFi ESP8266 sebagai pusat

kendali, sensor Load Cell dan HX711 untuk mengukur berat makanan pada wadah, sensor Ultrasonik HC-SR04 untuk memantau level air dalam wadah minum, serta Real Time Clock DS3231 untuk penjadwalan waktu makan dan minum. Sistem ini juga dilengkapi dengan Mini Pump Water, servo untuk mengendalikan katup makanan, dan Liquid Crystal Display (LCD) 20x4 I2C yang menampilkan informasi terkait status sensor dan kondisi sistem secara real-time.

Sistem akan secara otomatis menghidupkan Mini Pump Water dan motor servo pada waktu yang telah dijadwalkan, yaitu pukul 07.00 pagi dan 17.00 sore, untuk mengisi wadah makanan dan minuman. Selain itu, sistem dapat menampilkan notifikasi pada LCD dan mengirimkan pemberitahuan ke aplikasi mobile jika kondisi makanan atau air dalam wadah mulai habis. Semua komponen ini disusun secara terintegrasi, di mana data sensor dikirim ke Firebase Realtime Database dan dapat diakses melalui aplikasi mobile. Tampilan implementasi keseluruhan sistem dapat dilihat pada Gambar 5.8.



Gambar 5.8 Implementasi Keseluruhan Sistem

Keterangan gambar:

1. Arduino Uno R3 Built-in IoT WiFi ESP8266.
2. Loadcell and Modul HX711.
3. Ultrasonic HC-SR04 (Wadah Minum).
4. Modul Relay 1 Channel and Mini Pump Water.

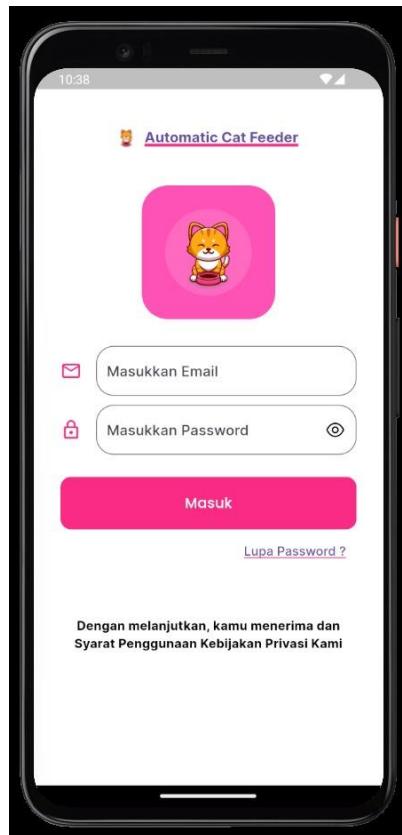
5. Servo MG996R.
6. Ultrasonic HC-SR04 (Tabung Minum).
7. Liquid Crystal Display 20X4 I2C.
8. Real Time Clock DS3231
9. Stepdown XL4015.
10. Power Supply.

5.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak yang dibangun meliputi implementasi antarmuka website yang telah dirancang pada bab perancangan. Tampilan antarmuka website bertujuan untuk memberikan informasi kepada pengguna dengan antarmuka yang interaktif.

5.3.1 Implementasi Antarmuka Halaman *Login*

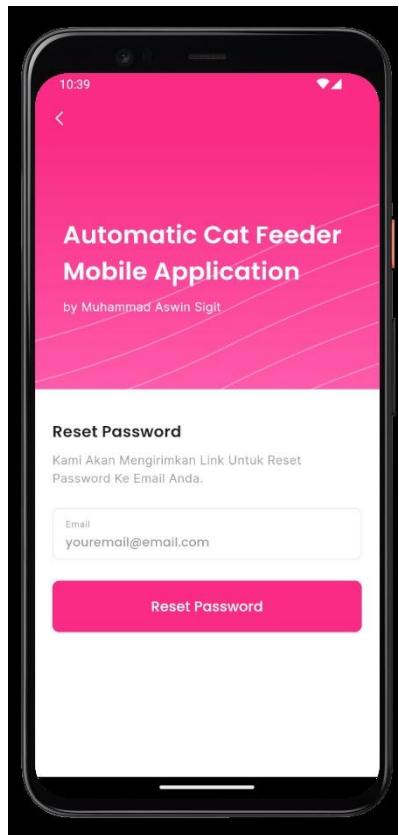
Implementasi halaman login terintegrasi dengan database Firebase untuk autentikasi pengguna. Pada halaman ini, terdapat dua input field yang berfungsi untuk memasukkan email dan password. Pengguna diharuskan mengisi kedua form tersebut dengan data yang terdaftar di database. Setelah pengguna mengklik tombol "Masuk", sistem akan melakukan validasi terhadap input yang diberikan. Jika email dan password yang dimasukkan cocok dengan data di Firebase, pengguna akan diarahkan ke menu utama aplikasi. Namun, jika terdapat kesalahan dalam input, sistem akan menampilkan pesan kesalahan yang sesuai dan mencegah akses ke aplikasi. Selain itu, tautan "Lupa Password" juga disediakan untuk membantu pengguna yang mengalami kesulitan dalam mengingat kata sandi mereka, dengan mengarahkan pengguna ke proses pemulihan kata sandi. Implementasi teknis halaman login ini dirancang agar user-friendly dan aman, serta dapat dilihat pada Gambar 5.9.



Gambar 5.9 Halaman *Login*

5.3.2 Implementasi Antarmuka Halaman Lupa Password

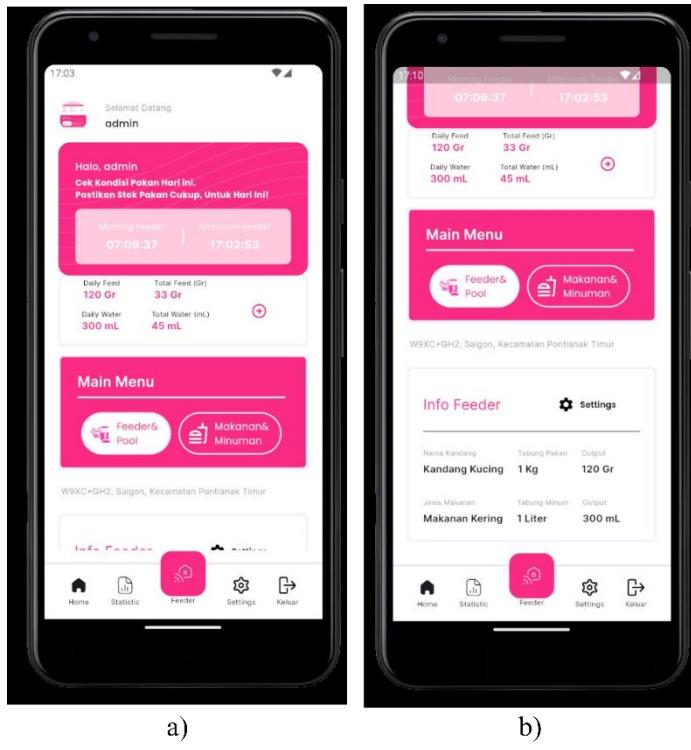
Implementasi halaman Lupa Password dilakukan dengan tujuan untuk memberikan kemudahan bagi pengguna dalam memulihkan akses ke akun mereka. Pada halaman ini, pengguna akan diminta untuk memasukkan alamat email yang telah terdaftar dalam database Firebase. Setelah pengguna mengisi form tersebut dan mengklik tombol "Reset Password", sistem akan mengirimkan tautan reset kata sandi ke alamat email yang dimasukkan. Proses ini diintegrasikan dengan layanan email Firebase, memastikan bahwa pengguna menerima email secara cepat dan aman. Desain halaman ini juga memperhatikan aspek usability agar pengguna dapat dengan mudah menavigasi proses pemulihan. Implementasi teknis halaman Lupa Password dapat dilihat pada Gambar 5.10.



Gambar 5.10 Halaman Lupa Password

5.3.3 Implementasi Antarmuka Halaman *Home*

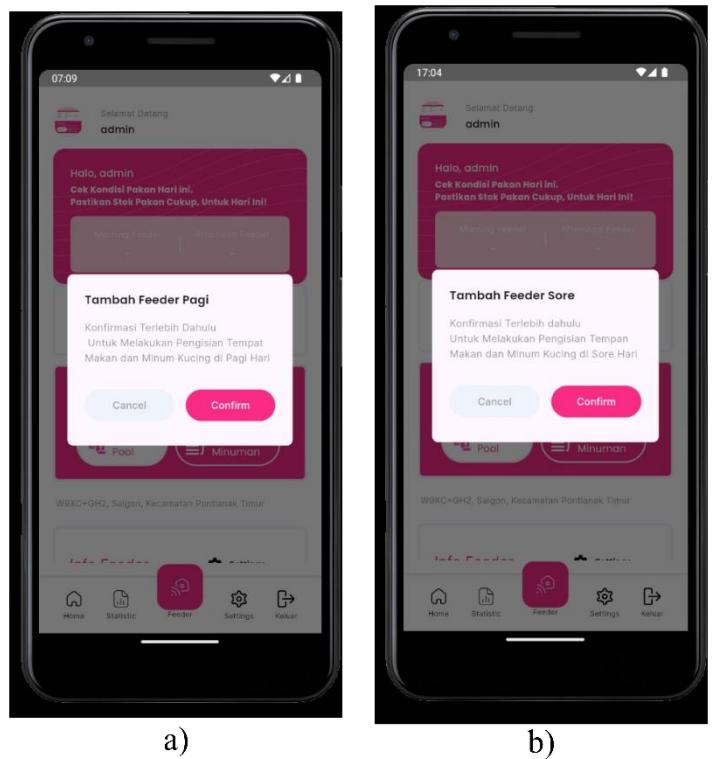
Implementasi halaman Home dirancang sebagai pusat navigasi utama yang memungkinkan pengguna untuk mengakses berbagai fitur inti aplikasi dengan cepat dan mudah. Halaman ini menampilkan informasi terkini dari penjadwalan real time clock DS3231 yang berkaitan dengan status pemberian makan dan minum kucing. Selain itu, sistem juga mengintegrasikan pembacaan nilai sensor load cell untuk mengukur berat pada wadah makan, serta sensor ultrasonik untuk mengukur ketinggian air di tabung dan wadah minum. Di bagian bawah halaman, terdapat tombol navigasi yang memudahkan pengguna untuk berpindah ke halaman lain, seperti statistik, feeder, pengaturan, dan keluar. Implementasi teknis halaman Home ini memastikan antarmuka yang intuitif dan informatif, serta dapat dilihat pada Gambar 5.11.



Gambar 5.11 Halaman Home

5.3.4 Implementasi Antarmuka Form Feeder

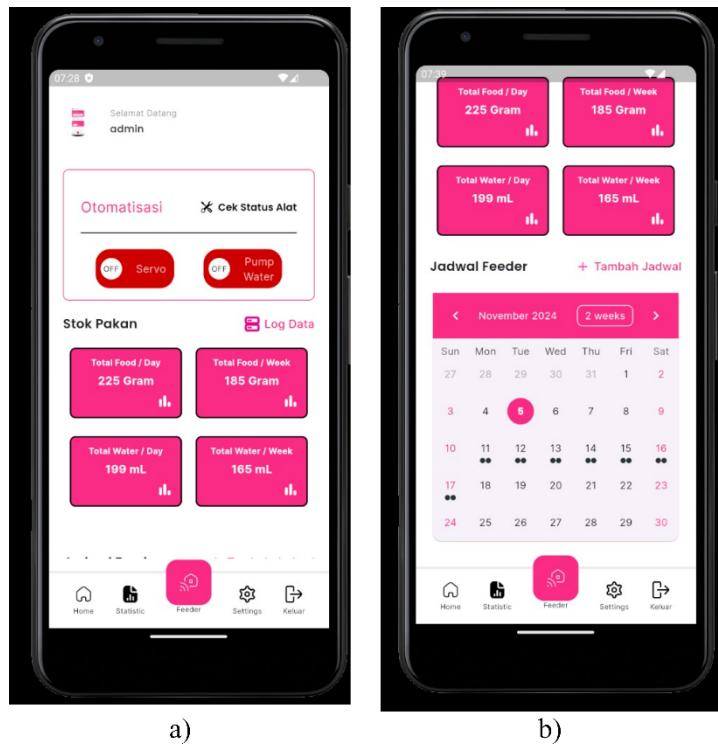
Implementasi halaman Feeder bertujuan untuk memberikan kontrol penuh kepada pengguna dalam penjadwalan pemberian makan dan minum kucing secara otomatis. Halaman ini memungkinkan pengguna untuk mengatur waktu penjadwalan, yang terdiri dari sesi pagi dan sore hari, dengan antarmuka yang sederhana dan mudah dipahami. Pengguna dapat memilih waktu spesifik untuk setiap sesi, dan sistem akan secara otomatis menjalankan proses pemberian makan dan minum sesuai dengan jadwal yang ditentukan. Implementasi ini memastikan bahwa pengguna dapat mengelola kebutuhan nutrisi kucing mereka dengan efisien. Desain teknis halaman Feeder dapat dilihat pada Gambar 5.12.



Gambar 5.12 a) Form *Feeder* pagi, b) Form *Feeder* sore

5.3.5 Implementasi Antarmuka Halaman *Statistic*

Implementasi halaman Statistik dirancang untuk menyediakan informasi terperinci mengenai pola konsumsi makanan dan air kucing secara periodik. Pada halaman ini, pengguna dapat mengakses data total jumlah makanan yang telah diberikan per hari dan per minggu, serta total volume air yang telah dikonsumsi dalam periode yang sama. Fitur ini memungkinkan pengguna untuk memantau kesehatan dan kebiasaan makan kucing mereka secara lebih efektif, serta memberikan kontrol manual maupun otomatis terhadap pemberian makan dan minum. Selain itu, halaman ini dilengkapi dengan tampilan kalender yang memudahkan pengguna dalam melihat dan menyesuaikan jadwal pemberian makan dan minum. Fitur "Tambah Jadwal" juga disediakan untuk memungkinkan pengguna menambahkan atau mengubah jadwal sesuai kebutuhan. Implementasi teknis halaman Statistik dapat dilihat pada Gambar 5.13.



Gambar 5.13 Halaman Statistic

5.3.6 Implementasi Antarmuka Halaman Detail Data

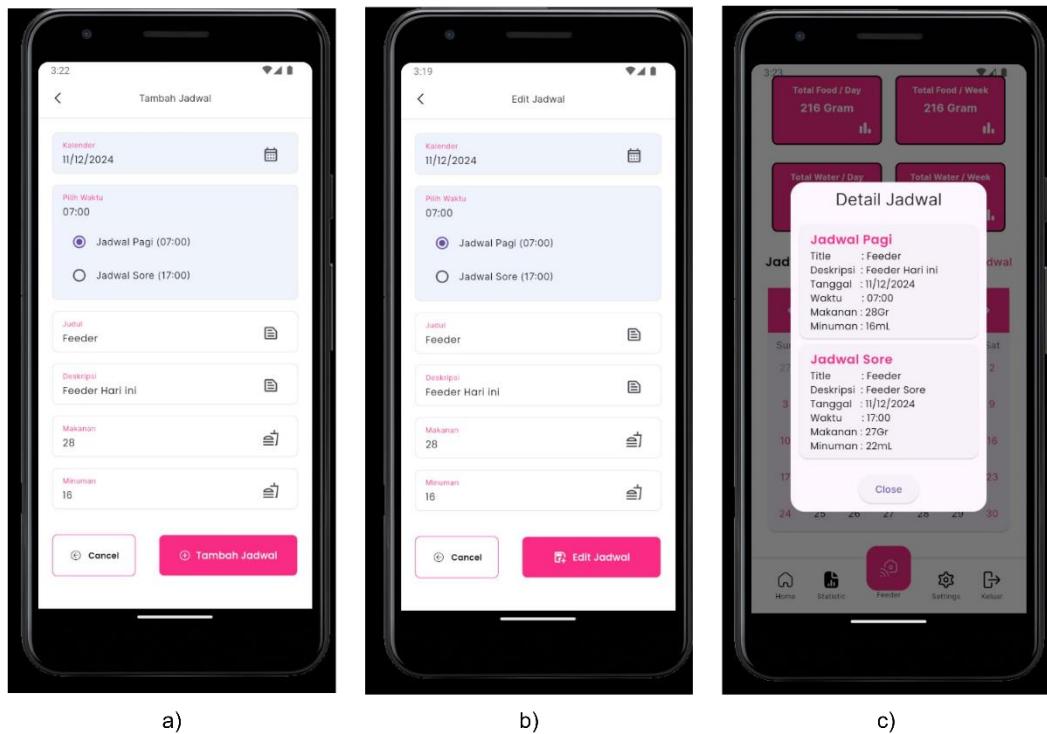
Implementasi halaman Detail Data pada aplikasi mobile berfungsi untuk menyajikan analisis mendalam terkait konsumsi makanan dan air kucing secara visual. Pada bagian atas halaman, ditampilkan grafik batang yang merepresentasikan data harian yang dimasukkan oleh pengguna melalui form tambah jadwal. Selain itu, halaman ini juga menyertakan diagram lingkaran yang menggambarkan persentase hasil pembacaan sensor ultrasonic dan load cell, yang digunakan untuk mengukur ketinggian air serta berat pakan secara real-time. Visualisasi ini memudahkan pengguna dalam memahami pola konsumsi kucing secara lebih terperinci. Implementasi dari desain halaman detail data dapat dilihat pada Gambar 5.14.



Gambar 5.14 Halaman Detail Data

5.3.7 Implementasi Antarmuka Halaman Penjadwalan

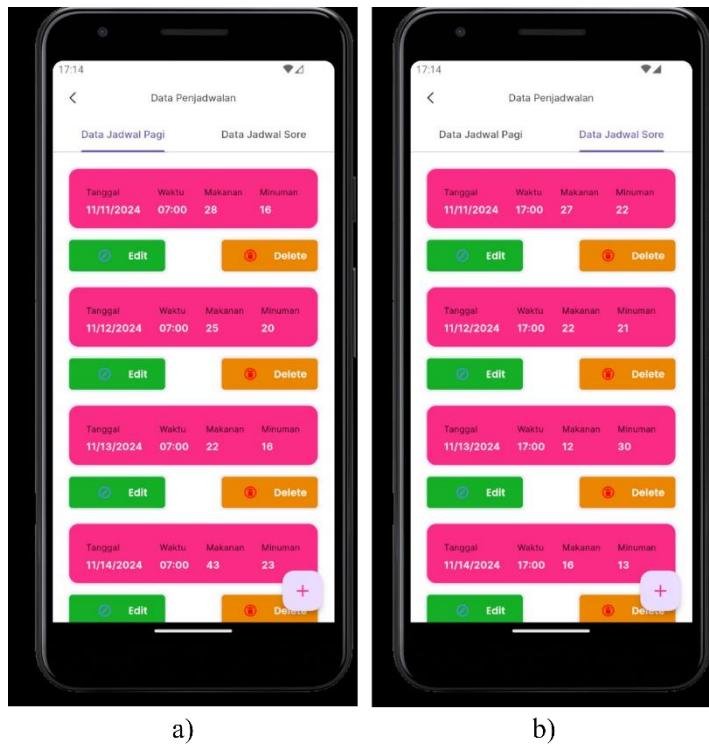
Implementasi halaman "Penjadwalan" pada aplikasi mobile memungkinkan pengguna untuk menambahkan jadwal pemberian makan dan minum kucing secara lebih terstruktur. Pada halaman ini, pengguna dapat menentukan parameter seperti waktu pemberian pakan, jumlah pakan yang akan diberikan, serta frekuensi pemberian, baik harian maupun mingguan. Antarmuka yang dirancang dengan sederhana dan mudah digunakan memastikan kemudahan pengguna dalam mengatur jadwal sesuai dengan kebutuhan spesifik kucing. Setelah seluruh informasi diinput, pengguna dapat menyimpan jadwal dengan menekan tombol "Simpan," yang secara otomatis akan mengaktifkan pengaturan sesuai jadwal yang telah ditentukan. Implementasi dari desain halaman tambah jadwal dapat dilihat pada Gambar 5.15.



Gambar 5.15 a) Form tambah jadwal, b) Form edit jadwal, c) Detail jadwal

5.3.8 Implementasi Antarmuka Halaman Data Penjadwalan

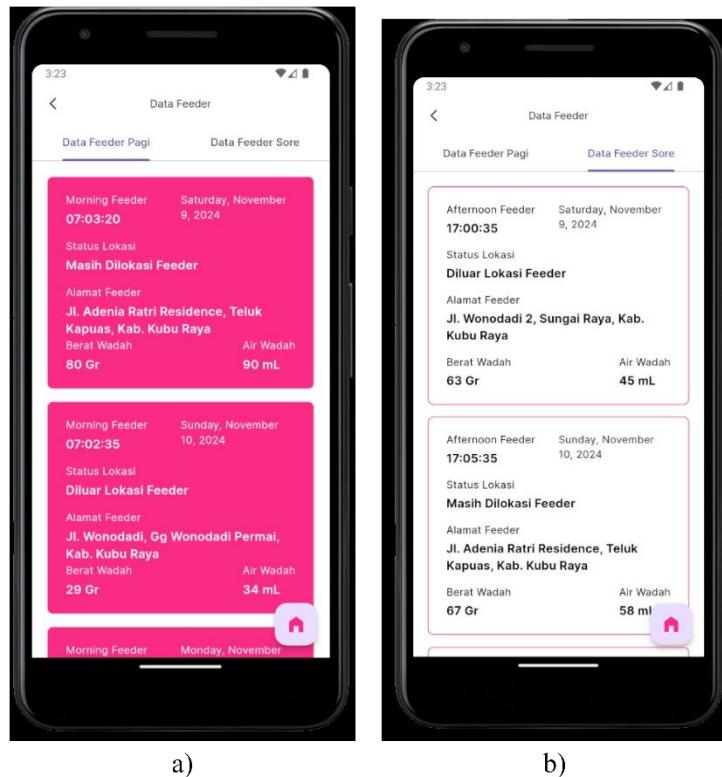
Implementasi halaman "Data Jadwal Penjadwalan" pada aplikasi mobile menyediakan tampilan jadwal pemberian makan dan minum kucing yang telah diatur untuk pagi hari. Pada halaman ini, pengguna dapat melihat secara rinci informasi terkait tanggal, waktu, jumlah makanan, dan jumlah minuman yang telah dijadwalkan. Selain itu, tersedia fitur untuk mengedit atau menghapus jadwal yang sudah ada, serta tombol untuk menambahkan jadwal baru sesuai kebutuhan. Antarmuka ini dirancang untuk mempermudah pengguna dalam mengelola jadwal pemberian makan dan minum dengan efisien dan tepat waktu. Implementasi dari desain halaman data jadwal pagi dapat dilihat pada Gambar 5.16.



Gambar 5.16 a) Halaman Data Jadwal Pagi, b) Halaman Data Jadwal Sore

5.3.9 Implementasi Antarmuka Halaman Data Feeder

Implementasi halaman "Data Feeder" pada aplikasi mobile menampilkan informasi terkait jadwal pemberian makan dan minum kucing yang telah diatur untuk sore hari. Pada halaman ini, pengguna dapat melihat detail seperti tanggal, waktu, jumlah makanan, dan jumlah minuman yang sudah dijadwalkan. Tersedia pula fitur untuk mengedit atau menghapus jadwal yang ada, serta tombol untuk menambahkan jadwal baru sesuai kebutuhan. Antarmuka ini dirancang untuk memudahkan pengguna dalam mengelola jadwal pemberian makan dan minum kucing pada sore hari secara lebih teratur dan efisien. Implementasi dari desain halaman data jadwal sore dapat dilihat pada Gambar 5.17.



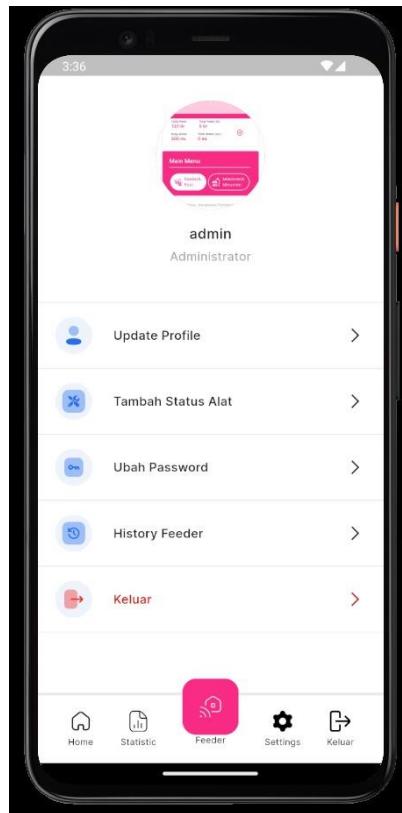
a)

b)

Gambar 5.17 a) Halaman data feeder pagi, b) Halaman data feeder sore

5.3.10 Implementasi Antarmuka Halaman *Settings*

Implementasi halaman "Setting" pada aplikasi mobile memberikan akses kepada pengguna untuk mengelola berbagai pengaturan terkait akun dan perangkat yang terhubung. Pada halaman ini, pengguna dapat memperbarui profil, memeriksa status alat yang digunakan, mengubah kata sandi, melihat riwayat pemberian makan dan minum kucing, serta melakukan logout dari aplikasi. Halaman ini dirancang untuk mempermudah pengguna dalam melakukan penyesuaian pengaturan sesuai kebutuhan mereka. Implementasi dari desain halaman settings dapat dilihat pada Gambar 5.18.

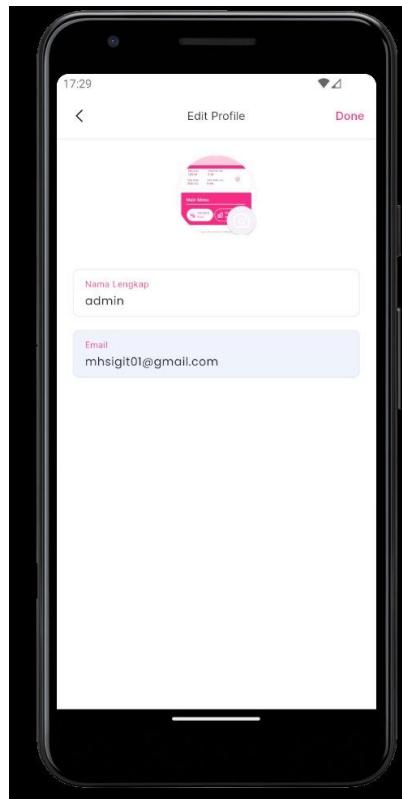


Gambar 5.18 Halaman *Settings*

Halaman "Setting" pada aplikasi mobile berfungsi sebagai pusat pengaturan yang memungkinkan pengguna untuk mengelola akun dan perangkat yang terhubung dengan sistem otomatisasi pemberian makan dan minum kucing. Pada halaman ini, terdapat beberapa fitur utama, antara lain:

1. Halaman Update Profile

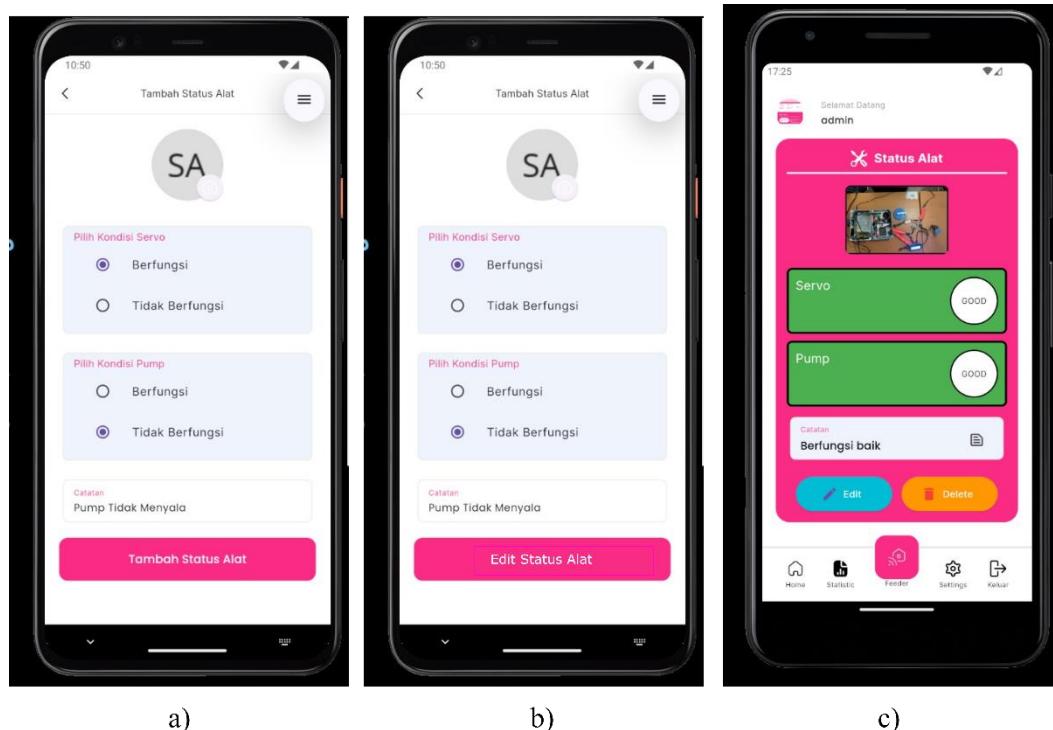
Fitur ini memungkinkan pengguna untuk memperbarui informasi profil mereka, seperti nama pengguna, foto profil, dan detail lainnya yang relevan dengan akun. Dengan memperbarui profil, pengguna dapat memastikan data yang terhubung dengan sistem selalu terbaru. Implementasi dari desain halaman update profile dapat dilihat pada Gambar 5.19.



Gambar 5.19 Halaman *Update Profile*

2. Halaman Status Alat

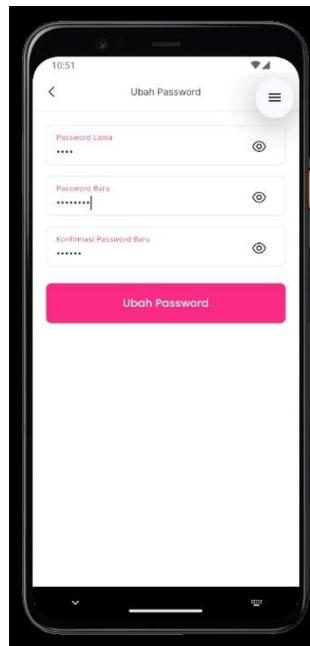
Fitur ini berfungsi untuk mengelola status perangkat yang terhubung dengan sistem otomatisasi. Jika pengguna belum memasukkan data status alat, maka aplikasi akan langsung mengarahkan pengguna ke halaman Tambah Status Alat. Pada halaman ini, pengguna dapat menambahkan perangkat baru yang akan dikenali dan dipantau oleh sistem. Namun, jika data status alat sudah tersedia, pengguna akan langsung diarahkan ke halaman "Status Alat", di mana mereka dapat melihat status perangkat yang sudah terdaftar. Selain itu, pengguna juga memiliki opsi untuk mengedit atau menghapus data perangkat yang sudah ada, sehingga memungkinkan penyesuaian atau pembaruan informasi sesuai kebutuhan. Fitur-fitur ini dirancang untuk memastikan pengguna dapat dengan mudah mengelola perangkat yang terhubung, sehingga koneksi dan integrasi perangkat dengan sistem tetap berjalan optimal. Implementasi dari desain halaman status alat dapat dilihat pada Gambar 5.20.



Gambar 5.20 a) Halaman tambah status alat, b) Halaman edit status alat,
c) Halaman status alat

3. Halaman Ubah Password

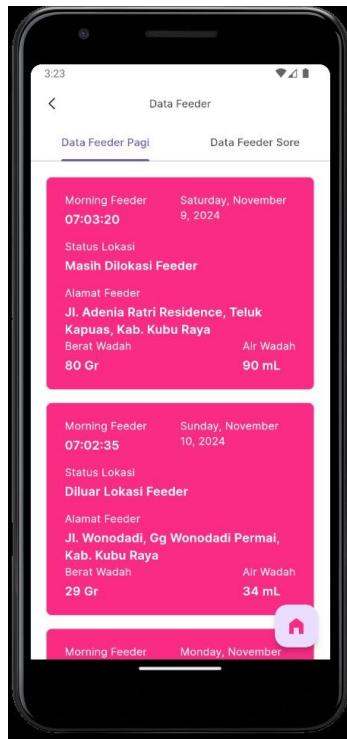
Fitur ini memberikan kemudahan bagi pengguna untuk mengganti kata sandi akun mereka guna meningkatkan keamanan akun. Dengan fitur ini, pengguna dapat melakukan penggantian kata sandi secara berkala untuk menjaga privasi dan keamanan data. Implementasi dari desain halaman ubah password dapat dilihat pada Gambar 5.21.



Gambar 5.21 Halaman Ubah *Password*

4. Halaman History Feeder

Fitur ini menyediakan riwayat pemberian makan dan minum kucing, termasuk detail seperti waktu dan jumlah pemberian. Dengan adanya riwayat ini, pengguna dapat memantau aktivitas sistem secara akurat dan memastikan kucing mendapatkan asupan yang tepat sesuai jadwal. Implementasi dari desain halaman history feeder dapat dilihat pada Gambar 5.22



Gambar 5.22 Halaman History Feeder

5. Logout

Fitur ini memungkinkan pengguna untuk keluar dari aplikasi secara aman setelah selesai menggunakan layanan. Dengan fitur logout, keamanan akun pengguna dapat terjaga, terutama jika aplikasi diakses melalui perangkat bersama.

5.4 Kode Program

Kode program merupakan langkah untuk mengkoneksikan antara perangkat keras dengan sistem yang akan digunakan. Kode program ini akan menghubungkan perangkat keras dengan *website* sehingga dapat menampilkan hasil pengukuran data sensor dan mengirim data ke dalam database untuk disimpan sesuai dengan kebutuhan *user*. Kode program ini dibuat dengan menggunakan Arduino IDE dengan menggunakan bahasa program C++. Kode program ini mendefinisikan *library*, *function*, *class* serta variabel sesuai dengan kebutuhan komponen atau sensor yang digunakan. Pada sistem ini dibagi menjadi beberapa bagian antara lain kode program pengukuran suhu, kode program pengukuran detak jantung, kode program pengiriman hasil pengukuran ke *database*, serta kode program keseluruhan.

5.4.1 Kode Program Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266

Pada sistem kendali otomatisasi pemberian makan dan minum untuk hewan peliharaan berbasis Internet of Things, perangkat transmitter bertugas untuk mengumpulkan data dari berbagai sensor dan mengirimkannya ke sistem receiver. Transmitter ini menggunakan mikrokontroler Arduino Uno R3 yang dilengkapi dengan modul WiFi ESP8266. Modul ini memungkinkan pengiriman data secara real-time ke server Firebase untuk memonitor dan mengendalikan sistem dari jarak jauh melalui aplikasi mobile.

1. Library, Define, dan Deklarasi variabel yang Digunakan

Library yang digunakan dalam program ini meliputi Wire.h untuk komunikasi I2C, LiquidCrystal_I2C.h untuk antarmuka dengan LCD, RTCLib.h untuk komunikasi dengan Real Time Clock (RTC) DS3231, Servo.h untuk pengendalian servo, dan HX711.h untuk membaca data dari sensor berat. Variabel-variabel yang dideklarasikan mencakup parameter sensor, status aktuator (seperti servo dan relay), dan data waktu. Kode ini mengatur pin input dan output untuk sensor dan aktuator serta mengatur konstanta kalibrasi untuk load cell.

Kode Program 5.1 Kode Program Transmitter pada Arduino Uno R3 Built-in IoT WiFi ESP8266

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <RTCLib.h>
#include <Servo.h>
#include <HX711.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);
RTC_DS3231 rtc;

char dataHari[7][12] = {"Minggu", "Senin", "Selasa", "Rabu",
"Kamis", "Jum'at", "Sabtu"};
int tanggal, bulan, tahun, jam, menit, detik;
String hari, ketHari, ketWaktu;
#define LOADCELL_WADAH_DOUT_PIN 4
#define LOADCELL_WADAH_SCK_PIN 5
HX711 lcWadah;
```

```
float calibration_factor_wadah = -388.10;

#define trigPinTabung 6
#define echoPinTabung 7
long durasiTabung;
float maxTinggiTabung = 34.5;
float radiusTabung = 4.25;
#define relayPin 8
Servo myServo;
#define servoPin 9
#define echoPinWadah 10
#define trigPinWadah 11
long durasiWadah;
float maxTinggiWadah = 15.0;
float radiusWadah = 4.5;
String sendDataToEsp;
String request = "";
bool pumpStatus = false;
bool servoStatus = false;
unsigned long previousLCDMillis = 0;
const long lcdInterval = 1000;
unsigned long previousNotificationMonitoringMillis = 0;
const long notificationMonitoringInterval = 60000;
```

2. Void Setup

Bagian void setup pada kode program transmitter berfungsi untuk melakukan inisialisasi awal dari seluruh perangkat keras yang terhubung pada Arduino Uno R3 Built-in IoT WiFi ESP8266. Pada fungsi ini, dilakukan pengaturan komunikasi serial dengan Serial.begin(9600) untuk memantau proses melalui monitor serial. Library I2C diinisialisasi dengan Wire.begin() untuk mendukung komunikasi antara Arduino dan komponen lain, seperti LCD dan RTC DS3231. Inisialisasi RTC dilakukan untuk memastikan perangkat dapat membaca waktu secara akurat. Jika RTC tidak terdeteksi atau kehilangan daya, sistem akan menampilkan pesan "RTC Tidak Ditemukan" dan menyesuaikan waktu sesuai dengan waktu kompilasi program. Selanjutnya, sensor load cell diatur dengan lcWadah.begin() untuk

membaca berat pada wadah makanan, melakukan kalibrasi menggunakan set_scale() dan mengatur nilai awal dengan tare().

Pada fungsi ini juga ditentukan mode kerja dari berbagai pin, seperti sensor ultrasonik pada tabung dan wadah, servo, dan relay. Misalnya, pin untuk sensor ultrasonik diatur sebagai input dan output untuk membaca tinggi air pada tabung dan wadah. Servo dihubungkan dan diatur ke posisi awal dengan myServo.attach() dan myServo.write(0). Sementara itu, relay diatur sebagai output dengan kondisi awal mati menggunakan digitalWrite(relayPin, LOW), untuk memastikan pompa tidak aktif ketika sistem pertama kali dinyalakan.

Selain itu, LCD diinisialisasi melalui fungsi initLCD() untuk menampilkan data monitoring pada layar. Inisialisasi ini memastikan bahwa perangkat siap untuk mengumpulkan data dan menjalankan fungsionalitas monitoring serta kontrol sesuai dengan kebutuhan otomatisasi pemberian makan dan minum.

Kode Program 5.2 Void Setup Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266

```
void setup() {  
    Serial.begin(9600);  
    Wire.begin();  
    if (!rtc.begin()) {  
        Serial.println("RTC Tidak Ditemukan");  
        Serial.flush();  
        abort();  
    }  
    if (rtc.lostPower()) {  
        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));  
    }  
    lcWadah.begin(LOADCELL_WADAH_DOUT_PIN,  
    LOADCELL_WADAH_SCK_PIN);  
    lcWadah.set_scale(calibration_factor_wadah);  
    lcWadah.tare();  
    pinMode(echoPinTabung, INPUT);  
    pinMode(trigPinTabung, OUTPUT);  
    pinMode(echoPinWadah, INPUT);
```

```
pinMode(trigPinWadah, OUTPUT);
myServo.attach(servoPin);
myServo.write(0);
pinMode(relayPin, OUTPUT);
digitalWrite(relayPin, LOW);
initLCD();
}
```

3. Void Loop

Bagian void loop pada kode program transmitter berfungsi untuk menjalankan proses utama secara berulang dalam sistem monitoring dan kontrol pada Arduino Uno R3 Built-in IoT WiFi ESP8266. Pada fungsi ini, dilakukan pembacaan data dari sensor secara terus-menerus untuk mengumpulkan informasi mengenai berat wadah, tinggi air pada wadah, dan tinggi air pada tabung. Data ini kemudian diolah untuk menentukan tindakan yang akan diambil oleh sistem.

Proses dimulai dengan pemanggilan fungsi `readSensor()`, yang bertugas membaca data dari sensor-sensor yang terhubung, seperti sensor load cell untuk berat wadah dan sensor ultrasonik untuk tinggi air pada tabung dan wadah. Data yang diperoleh dari sensor tersebut disimpan dalam variabel `beratWadah`, `tinggiAirWadah`, dan `tinggiAirTabung` untuk digunakan dalam langkah berikutnya.

Setelah data sensor dibaca, fungsi `monitoring()` dijalankan untuk memantau kondisi terkini dari berat dan tinggi air. Fungsi ini bertujuan untuk memastikan bahwa kondisi makanan dan air sesuai dengan batas yang telah ditentukan. Jika terdeteksi kekurangan, sistem akan mengambil tindakan sesuai dengan logika yang telah diatur dalam program.

Selanjutnya, fungsi `feeder()` dipanggil untuk mengatur pergerakan servo yang berfungsi membuka tutup wadah makanan sesuai dengan kebutuhan, seperti saat waktu pemberian makan telah tiba atau kondisi wadah memerlukan pengisian ulang. Fungsi `displayLCD()` digunakan untuk menampilkan data monitoring pada layar LCD 20x4, sehingga informasi mengenai berat dan tinggi air dapat dilihat secara langsung oleh pengguna.

Terakhir, fungsi sendData() dan reqData() digunakan untuk mengirimkan data monitoring ke sistem penerima melalui koneksi WiFi dan menerima perintah kontrol dari pengguna melalui aplikasi. Hal ini memungkinkan integrasi yang lancar antara perangkat keras dan sistem cloud dalam otomatisasi pemberian makan dan minum pada kucing.

Kode Program 5.3 Void Loop Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266

```
void loop() {  
    int beratWadah, volumeMLAirWadah, volumeMLAirTabung;  
    readSensor(beratWadah, volumeMLAirWadah, volumeMLAirTabung);  
    monitoring(beratWadah, volumeMLAirWadah, volumeMLAirTabung);  
    feeder();  
    displayLCD(beratWadah, volumeMLAirWadah, volumeMLAirTabung);  
    sendDataToReceiver(beratWadah, volumeMLAirWadah  
    , volumeMLAirTabug);  
    reqDataFromReceiver();  
}
```

4. Fungsi Inisialisasi Liquid Crystal Display

Fungsi initLCD() digunakan untuk menginisialisasi layar Liquid Crystal Display (LCD). Pada awalnya, LCD menampilkan informasi mengenai tugas akhir dan identitas pembuat sistem, serta status awal dari program. Fungsi ini memastikan bahwa LCD siap digunakan untuk menampilkan data yang diperlukan selama proses monitoring.

Kode Program 5.4 Inisialisasi Liquid Crystal Display

```
void initLCD() {  
    lcd.init();  
    lcd.backlight();  
    lcd.setCursor(5, 1);  
    lcd.print("TUGAS AKHIR");  
    lcd.setCursor(1, 2);  
    lcd.print("INTERNET OF THINGS");  
    delay(3000);  
    lcd.clear();
```

```
lcd.setCursor(8, 1);
lcd.print("Oleh : ");
lcd.setCursor(0, 2);
lcd.print("Muhammad Aswin Sigit");
delay(3000);
lcd.clear();
lcd.setCursor(3, 1);
lcd.print("MULAI PROGRAM");
lcd.setCursor(0, 2);
lcd.print("AUTOMATIC CAT FEEDER");
delay(3000);
}
```

5. Fungsi Inisialisasi Real Time Clock

Fungsi initRTC() berfungsi untuk menginisialisasi dan mengambil data waktu serta tanggal terkini dari modul Real Time Clock (RTC) DS3231 yang terhubung pada sistem transmitter Arduino Uno R3 Built-in IoT WiFi ESP8266. Dengan menggunakan RTC, sistem dapat mengetahui waktu secara akurat untuk menjalankan proses otomatisasi yang bergantung pada jadwal.

Pada fungsi ini, objek DateTime now digunakan untuk menyimpan waktu saat ini yang diperoleh dari fungsi rtc.now(). Data yang diambil meliputi tanggal, bulan, tahun, jam, menit, dan detik, yang kemudian disimpan dalam variabel masing-masing seperti tanggal, bulan, tahun, jam, menit, dan detik. Selain itu, variabel hari juga diisi dengan nama hari sesuai dengan indeks hari yang diberikan oleh RTC, menggunakan array dataHari yang telah didefinisikan sebelumnya.

Kemudian, informasi tanggal dan waktu tersebut dikonversi menjadi format string yang lebih mudah dibaca, dan disimpan dalam variabel ketHari untuk tanggal dalam format "dd/MM/yyyy" dan ketWaktu untuk waktu dalam format "HH:mm". Informasi ini nantinya akan digunakan oleh sistem untuk menampilkan data waktu pada layar LCD dan mengatur jadwal pemberian makan serta minum secara otomatis.

Kode Program 5.5 Inisialisasi Real Time Clock

```
void initRTC() {  
    DateTime now = rtc.now();  
    tanggal = now.day();  
    bulan = now.month();  
    tahun = now.year();  
    detik = now.second();  
    jam = now.hour();  
    menit = now.minute();  
    hari = dataHari[now.dayOfTheWeek()];  
    ketHari = String(tanggal) + "/" + String(bulan) + "/" +  
    String(tahun);  
    ketWaktu = String(jam) + ":" + String(menit) + ":" +  
    String(detik);  
}
```

6. Fungsi Wadah Pakan

Fungsi wadahPakan() digunakan untuk membaca dan menghitung berat pakan yang terdapat di dalam wadah makanan kucing menggunakan sensor load cell yang terhubung melalui modul HX711. Fungsi ini berperan penting dalam mengetahui jumlah pakan yang tersedia di wadah agar sistem dapat memantau dan mengontrol pemberian pakan secara otomatis sesuai kebutuhan.

Pada fungsi ini, nilai berat pakan diperoleh dengan menggunakan metode get_units() dari objek lcWadah yang dikalibrasi sebelumnya. Hasil pembacaan berat pakan kemudian disimpan dalam variabel beratWadah dengan satuan gram. Untuk memastikan nilai yang diperoleh akurat dan tidak negatif, dilakukan pengecekan dengan kondisi if (beratWadah < 0), di mana jika berat yang terdeteksi kurang dari 0, maka variabel beratWadah diatur menjadi 0. Hal ini mencegah terjadinya kesalahan pembacaan yang dapat mengganggu proses monitoring.

Kode Program 5.6 Kode Program Menghitung Berat Wadah pada Makanan Kucing

```
void wadahPakan(int &beratWadah) {
    beratWadah = lcWadah.get_units(10);
    if (beratWadah < 0) {
        beratWadah = 0;
    }
}
```

7. Fungsi Ultrasonic

Fungsi `readUltrasonic()` digunakan untuk mengukur jarak menggunakan sensor ultrasonik, yang berperan dalam menghitung volume air pada wadah dan tabung minum kucing. Fungsi ini memanfaatkan dua pin pada sensor ultrasonik, yaitu pin trig dan echo, untuk mengirim dan menerima sinyal ultrasonik.

Proses pengukuran dimulai dengan mengirimkan sinyal melalui pin trig dengan mengatur pin tersebut menjadi HIGH selama 10 mikrodetik. Setelah itu, sinyal dikembalikan ke LOW dan sensor akan mendeteksi waktu yang dibutuhkan oleh sinyal untuk memantul kembali ke pin echo. Fungsi `pulseIn()` digunakan untuk mengukur durasi waktu pantulan sinyal ultrasonik tersebut dalam mikrodetik.

Nilai durasi yang diperoleh kemudian dikonversi menjadi jarak dalam satuan sentimeter dengan menggunakan rumus $\text{durasi} * 0.034 / 2$. Faktor 0.034 adalah kecepatan suara dalam $\text{cm}/\mu\text{s}$, dan hasilnya dibagi dua karena sinyal harus menempuh jarak dua kali (pergi dan kembali). Fungsi ini mengembalikan nilai `distance`, yang menunjukkan jarak antara sensor dan permukaan air.

Pengukuran jarak ini digunakan untuk menghitung ketinggian air di dalam wadah dan tabung minum. Informasi ini penting untuk memastikan ketersediaan air yang cukup bagi kucing dan untuk mengontrol pengisian ulang secara otomatis melalui sistem kendali.

Kode Program 5.7 Kode Program Data Sensor Ultrasonic

```
long readUltrasonic(int trigPin, int echoPin) {  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
    long durasi = pulseIn(echoPin, HIGH);  
    long distance = durasi * 0.034 / 2;  
    return distance;  
}
```

8. Fungsi Kalibrasi volume air pada ultrasonic

Fungsi sensorUltrasonic() bertugas untuk melakukan pengukuran dan kalibrasi volume air pada wadah dan tabung minum kucing menggunakan sensor ultrasonik. Fungsi ini mengambil data jarak air dari sensor ultrasonic yang dipasang pada wadah dan tabung, kemudian menghitung ketinggian air yang tersisa berdasarkan jarak yang terbaca.

Proses dimulai dengan pemanggilan fungsi readUltrasonic() untuk mendapatkan jarak antara permukaan air dan sensor di dalam wadah dan tabung. Data jarak ini kemudian digunakan untuk menghitung tinggi air dengan mengurangkan nilai maksimum ketinggian air yang dapat ditampung oleh wadah atau tabung (variabel maxTinggiWadah dan maxTinggiTabung) dengan nilai jarak yang terbaca.

Setelah diperoleh tinggi air, volume air dihitung dengan mengalikan tinggi air dengan luas penampang wadah atau tabung (luasWadah dan luasTabung). Hasil perhitungan ini kemudian dikonversi ke dalam satuan mililiter dan disimpan dalam variabel volumeMLAirWadah dan volumeMLAirTabung.

Fungsi ini memastikan pengukuran volume air dalam wadah dan tabung selalu akurat, sehingga sistem dapat mengontrol pengisian ulang air secara otomatis sesuai dengan kebutuhan kucing. Selain itu, kalibrasi ini juga membantu dalam memantau kondisi air dan menjaga ketersediaannya agar tetap optimal.

Kode Program 5.8 Kalibrasi volume air pada ultrasonic

```
void      sensorUltrasonic(int      &volumeMLAirWadah,      int
&volumeMLAirTabung) {
    float      jarakAirWadah      =      readUltrasonic(trigPinWadah,
echoPinWadah);
    if (tinggiAirWadah <= 1) {
        tinggiAirWadah = 0;
    }
    tinggiAirWadah = abs(maxTinggiWadah - jarakAirWadah);
    float volumeWadah = luasWadah * tinggiAirWadah;
    volumeMLAirWadah = abs(static_cast<int>(volumeWadah));
    float      jarakAirTabung      =      readUltrasonic(trigPinTabung,
echoPinTabung);
    if (maxTinggiTabung <= 1) {
        maxTinggiTabung = 0;
    }
    tinggiAirTabung = abs(maxTinggiTabung - jarakAirTabung);
    float volumeTabung = luasTabung * tinggiAirTabung;
    volumeMLAirTabung = abs(static_cast<int>(volumeTabung));
}
```

9. Fungsi Pembacaan Data Sensor

Fungsi readSensor() memastikan bahwa semua data yang dibutuhkan untuk monitoring kondisi makanan dan minuman terekam dengan baik dan dapat digunakan untuk pengambilan keputusan oleh sistem kontrol otomatis dalam memberikan pakan dan minum untuk kucing.

Pada fungsi ini, pertama-tama dilakukan inisialisasi waktu melalui pemanggilan fungsi initRTC(), sehingga waktu pembacaan data dapat dicatat dengan akurat. Setelah itu, fungsi wadahPakan() dipanggil untuk mendapatkan nilai berat makanan dalam wadah menggunakan sensor load cell. Data berat ini kemudian disimpan dalam variabel beratWadah. Selanjutnya, fungsi sensorUltrasonic() digunakan untuk mengukur ketinggian air dalam wadah dan tabung menggunakan sensor ultrasonic. Hasil pengukuran ini kemudian dikonversi menjadi nilai volume air dan disimpan dalam variabel tinggiAirWadah dan

tinggiAirTabung.

Kode Program 5.9 Kode Program Pembacaan Data Sensor

```
void readSensor(int &beratWadah, long &tinggiAirWadah, long  
&tinggiAirTabung) {  
    initRTC();  
    wadahPakan(beratWadah);  
    sensorUltrasonic(tinggiAirWadah, tinggiAirTabung);  
}
```

10. Fungsi kendali servo

Fungsi bukaServo() dan activateServo() pada kode transmitter bertanggung jawab untuk mengendalikan pergerakan servo dalam sistem otomatisasi pemberian makan hewan. Fungsi bukaServo(int jumlah) memungkinkan pergerakan servo melalui proses membuka dan menutup secara berulang sebanyak jumlah yang ditentukan. Pada fungsi ini, servo bergerak dari posisi 0 derajat ke 90 derajat dan kembali ke posisi 0 derajat, dengan setiap perubahan posisi diberi jeda 10 milidetik untuk memastikan pergerakan yang halus. Setelah perulangan selesai, variabel servoStatus diatur menjadi true, menandakan bahwa servo telah aktif.

Sedangkan, fungsi activateServo(bool status) digunakan untuk memberikan informasi tentang status aktif atau tidaknya servo melalui output serial. Jika status bernilai true, maka akan dicetak pesan "Servo dihidupkan", dan jika status bernilai false, pesan yang dicetak adalah "Servo dimatikan". Fungsi ini memudahkan proses monitoring dan debugging selama operasi sistem.

Kode Program 5.10 Kode Program Menghidupkan Servo

```
void bukaServo(int jumlah) {  
    for (int i = 0; i < jumlah; i++) {  
        for (int posisi = 0; posisi <= 90; posisi++) {  
            myServo.write(posisi);  
            delay(10);  
        }  
        for (int posisi = 90; posisi >= 0; posisi--) {  
            myServo.write(posisi);  
            delay(10);  
        }  
    }  
}
```

```

    }
}

servoStatus = true;
}

void activateServo(bool status) {
    if (status) {
        Serial.println("Servo dihidupkan");
    } else {
        Serial.println("Servo dimatikan");
    }
}

```

11. Fungsi kendali Mini Pump Water

Fungsi onPump() dan activatePump() bertujuan untuk mengontrol pengoperasian pompa air dalam sistem. Fungsi onPump() mengaktifkan pompa dengan menyalakan relay yang terhubung ke pin kendali selama 5 detik menggunakan fungsi millis() untuk mengukur durasi waktu. Selama periode ini, relay dinyalakan untuk memungkinkan aliran air, dan setelah durasi habis, relay dimatikan kembali untuk menghentikan pompa. Setelah operasi ini, status pumpStatus diatur menjadi true, menunjukkan bahwa pompa telah diaktifkan.

Fungsi activatePump(bool status) memberikan keluaran status pompa melalui serial. Jika status bernilai true, maka pesan "Pompa dihidupkan" akan dicetak, dan jika status bernilai false, maka pesan "Pompa dimatikan" akan dicetak. Fungsi ini membantu dalam proses pemantauan kondisi pompa secara real-time melalui output serial.

Kode Program 5.11 Kode Program Menghidupkan Mini Pump Water

```

void onPump() {
    unsigned long startMillis = millis();
    while (millis() - startMillis <= 5000) {
        digitalWrite(relayPin, HIGH);
    }
    digitalWrite(relayPin, LOW);
    pumpStatus = true;
}

```

```

}

void activatePump(bool status) {
    if (status) {
        Serial.println("Pompa dihidupkan");
    } else {
        Serial.println("Pompa dimatikan");
    }
}

```

12. Fungsi Notifikasi Monitoring

Fungsi monitoringNotification() dan showNotification() bertujuan untuk menampilkan pesan notifikasi pada layar LCD sebagai bagian dari sistem monitoring kondisi makanan dan minuman pada kucing. Fungsi monitoringNotification(String title, String message, int delayTime) digunakan untuk menampilkan judul dan pesan notifikasi yang spesifik terkait status makanan dan minuman di wadah. Fungsi ini membersihkan tampilan LCD, kemudian menampilkan pesan "NOTIFIKASI !!!" di baris pertama, judul notifikasi di baris kedua, dan pesan di baris ketiga, serta menahan tampilan tersebut selama delayTime milidetik sebelum berpindah ke proses berikutnya.

Fungsi showNotification(String title, String message, int delayTime) memiliki mekanisme yang serupa dengan monitoringNotification(), namun dengan format penempatan pesan yang sedikit berbeda pada layar LCD. Fungsi ini digunakan untuk menampilkan informasi umum terkait status proses, seperti status "PROSES PENGISIAN..." atau "PROCESS SUCCESSFULL!". Kedua fungsi ini memastikan bahwa setiap perubahan kondisi terdeteksi oleh sistem dapat diinformasikan secara visual melalui LCD, sehingga pengguna dapat memantau secara langsung.

Kode Program 5.12 Kode Program Notifikasi Monitoring

```

void monitoringNotification(String tittle, String message, int
delayTime) {
    lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("NOTIFIKASI !!!");
}

```

```

lcd.setCursor(0, 1);
lcd.print(tittle);
lcd.setCursor(2, 2);
lcd.print(message);
delay(delayTime);
}

void showNotification(String tittle, String message, int delayTime) {
    lcd.clear();
    lcd.setCursor(3, 1);
    lcd.print(tittle);
    lcd.setCursor(0, 2);
    lcd.print(message);
    delay(delayTime);
}

```

13. Fungsi Monitoring

Fungsi monitoring() digunakan untuk memantau secara otomatis kondisi makanan dan air pada wadah serta tabung dalam sistem. Fungsi ini memanfaatkan variabel beratWadah, tinggiAirWadah, dan tinggiAirTabung untuk mengukur tingkat makanan dan air yang tersedia. Fungsi ini dijalankan secara berkala berdasarkan interval waktu tertentu yang diukur menggunakan fungsi millis().

Dalam prosesnya, fungsi ini menampilkan pesan "MONITORING CHECKING!" pada layar LCD menggunakan showNotification(), kemudian memeriksa kondisi dari beratWadah, tinggiAirWadah, dan tinggiAirTabung. Jika berat makanan dalam wadah kurang dari 40 gram, fungsi akan memicu notifikasi "PAKAN WADAH < 40 GR" dan kemudian mengaktifkan fungsi bukaServo() untuk mengisi ulang wadah makanan. Demikian pula, jika tinggi air dalam wadah kurang dari 150 ml, sistem akan menampilkan pesan "AIR WADAH < 150 ML" dan mengaktifkan onPump() untuk mengisi air secara otomatis.

Fungsi monitoring() juga menangani situasi di mana jumlah makanan atau air melebihi kapasitas yang diharapkan, misalnya menampilkan pesan "PAKAN WADAH > 120 GR" atau "AIR WADAH > 300 ML" ketika deteksi menunjukkan kondisi penuh. Selain itu, fungsi ini mengawasi tinggi air di dalam tabung,

memberikan peringatan jika volume air kurang dari 300 ml atau melebihi 1 liter, serta menampilkan pesan ketika waktu pengecekan berada di luar jadwal yang ditentukan (07:00 dan 17:00).

Dengan menggunakan fungsi ini, sistem dapat memberikan peringatan visual terkait kondisi makanan dan minuman secara real-time, serta melakukan tindakan otomatis untuk menjaga ketersediaan makanan dan air yang optimal bagi kucing.

Kode Program 5.13 Kode Program untuk Monitoring Makanan dan Minuman Kucing

```
void monitoring(int beratWadah, long tinggiAirWadah, long tinggiAirTabung) {
    unsigned long currentMillis = millis();
    if (currentMillis - previousNotificationMonitoringMillis
        >= notificationMonitoringInterval) {
        previousNotificationMonitoringMillis = currentMillis;
        showNotification("NOTIFIKASI !!!", "MONITORING
CHECKING!", 3000);
        if (beratWadah >= 0 && beratWadah < 40) {
            monitoringNotification("PAKAN WADAH < 40 GR", "WADAH
PERLU DIISI!", 2000);
            showNotification("NOTIFIKASI !!!", "PROSES PENGISIAN
...", 1000);
            bukaServo(4);
            showNotification("NOTIFIKASI !!!", "PROCESS
SUCCESSFULL!", 2000);
        } else if (beratWadah >= 41 && beratWadah < 100) {
            monitoringNotification("PAKAN WADAH < 100 GR", "SEGERA
ISI WADAH", 2000);
        } else if (beratWadah > 120) {
            monitoringNotification("PAKAN WADAH > 120 GR", "WADAH
PAKAN PENUH", 2000);
        }
        if (tinggiAirWadah >= 0 && tinggiAirWadah < 150) {
            monitoringNotification("AIR WADAH < 150 ML", "WADAH
PERLU DIISI", 2000);
        }
    }
}
```

```

        showNotification("NOTIFIKASI !!!", "PROSES PENGISIAN
        ...", 1000);
    onPump();
    showNotification("NOTIFIKASI!!!", "PROCESS
    SUCCESSFULL!", 2000);
}
else if (tinggiAirWadah >= 151 && tinggiAirWadah < 280){
    monitoringNotification("AIR WADAH < 300 ML", "SEGERA
    ISI WADAH", 2000);
} else if (tinggiAirWadah > 300) {
    monitoringNotification("AIR WADAH > 300 ML", "WADAH
    MINUM PENUH", 2000);
}
if (tinggiAirTabung >= 0 && tinggiAirTabung < 300) {
    monitoringNotification("AIR TABUNG < 300 ML", "SEGERA
    ISI TABUNG", 2000);
} else if (tinggiAirTabung > 1000) {
    monitoringNotification("AIR TABUNG > 1 L", "TABUNG
    MINUM PENUH", 2000);
}
if (jam != 7 && jam != 17) {
    monitoringNotification("DILUAR WAKTU FEEDING", "CEK
    KET WAKTU", 2000);
}
}
}

```

14. Fungsi Feeder

Fungsi feeder() bertanggung jawab untuk mengatur penjadwalan otomatis pemberian makan dan minum pada kucing. Fungsi ini memastikan bahwa proses pengisian makanan dan minuman dilakukan pada waktu yang telah ditentukan, yaitu pukul 07:00 untuk pagi hari dan 17:00 untuk sore hari. Pada waktu yang ditetapkan, fungsi ini akan memicu serangkaian notifikasi menggunakan showNotification() yang ditampilkan pada layar LCD, memberikan informasi

kepada pengguna mengenai status pengecekan dan pengisian pakan dan air.

Setelah menampilkan notifikasi terkait proses pengecekan dan waktu makan (pagi atau sore), fungsi feeder() akan menjalankan pengisian air dengan memanggil fungsi onPump() untuk mengaktifkan pompa air, diikuti oleh bukaServo(4) untuk menggerakkan servo yang membuka wadah makanan. Setelah proses pengisian selesai, notifikasi "PROCESS SUCCESSFULL!" akan ditampilkan untuk memberi tahu bahwa pengisian pakan dan minum telah berhasil dilakukan.

Dengan adanya fungsi ini, sistem dapat melakukan pengisian pakan dan minuman secara otomatis dan terjadwal, memastikan ketersediaan makanan dan minuman yang cukup bagi kucing tanpa perlu intervensi manual. Hal ini membantu dalam menjaga konsistensi dan keteraturan dalam pemberian makan dan minum sesuai dengan waktu yang telah diatur.

Kode Program 5.14 Kode Program Penjadwalan Makan dan Minum Kucing

```
void feeder() {
    if (jam == 7 && menit == 0 && detik == 0) {
        showNotification("NOTIFIKASI!!", "FEEDING CHECKING!!..",
            3000);
        showNotification("NOTIFIKASI!!", "MORNING FEEDING!!!!",
            2000);
        showNotification("NOTIFIKASI!!", "PROSES PENGISIAN ...",
            1000);
        onPump();
        bukaServo(4);
        showNotification("NOTIFIKASI!!!", "PROCESS
        SUCCESSFULL!", 2000);
    } else if (jam == 17 && menit == 0 && detik == 0) {
        showNotification("NOTIFIKASI!!!", "FEEDING
        CHECKING!!..", 3000);
        showNotification("NOTIFIKASI!!", "AFTERNOON FEEDING!!!",
            2000);
        showNotification("NOTIFIKASI!!!", "PROSES PENGISIAN
        ...", 1000);
    }
}
```

```

    onPump();
    bukaServo(4);
    showNotification("NOTIFIKASI!!", "PROCESS SUCCESSFULL!",
    2000);
}
}

```

15. Fungsi Menampilkan Data

Fungsi displayData() berfungsi untuk menampilkan informasi terkait kondisi pakan dan air pada layar LCD secara berkala. Data yang ditampilkan meliputi berat pakan dalam wadah, tinggi air dalam wadah, dan tinggi air dalam tabung. Fungsi ini menggunakan teknik pengaturan waktu menggunakan millis() untuk memastikan bahwa tampilan pada layar LCD diperbarui pada interval yang telah ditentukan oleh variabel lcdInterval.

Setiap kali interval waktu terpenuhi, fungsi ini akan membersihkan tampilan LCD dengan lcd.clear() dan menampilkan data baru pada koordinat yang sesuai. Berat pakan ditampilkan dalam satuan gram (Gr), sedangkan tinggi air dalam wadah dan tabung ditampilkan dalam satuan mililiter (mL). Selain itu, fungsi ini juga menampilkan informasi tambahan seperti keterangan hari (ketHari) dan waktu (ketWaktu), sehingga pengguna dapat memantau kondisi secara real-time.

Fungsi ini berperan penting dalam memberikan informasi yang akurat kepada pengguna mengenai kondisi pakan dan air, sehingga pengguna dapat mengetahui kapan pengisian ulang diperlukan. Dengan menampilkan data secara langsung pada layar LCD, sistem ini memudahkan pengguna dalam melakukan monitoring tanpa harus terhubung ke perangkat lain.

Kode Program 5.15 Kode Program Menampilkan Data pada LCD

```

void displayData(int beratWadah, long tinggiAirWadah, long
tinggiAirTabung) {
    unsigned long currentMillis = millis();
    if (currentMillis - previousLCDMillis >= lcdInterval) {
        previousLCDMillis = currentMillis;
        lcd.clear();
        lcd.setCursor(0, 0);
    }
}

```

```
lcd.print("Pakan Wadah");
lcd.setCursor(11, 0);
lcd.print(": ");
lcd.print(beratWadah);
lcd.setCursor(17, 0);
lcd.print(" Gr");
lcd.setCursor(0, 1);
lcd.print("Air Wadah");
lcd.setCursor(11, 1);
lcd.print(": ");
lcd.print(tinggiAirWadah);
lcd.setCursor(18, 1);
lcd.print("mL");
lcd.setCursor(0, 2);
lcd.print("Air Tabung");
lcd.setCursor(11, 2);
lcd.print(": ");
lcd.print(tinggiAirTabung);
lcd.setCursor(18, 2);
lcd.print("mL");
lcd.setCursor(0, 3);
lcd.print(ketHari + " - " + ketWaktu);
}
}
```

16. Fungsi Pengiriman Data

Fungsi sendDataToReceiver() digunakan untuk mengirimkan data dari sistem monitoring pada Transmitter ke Receiver yang terhubung melalui komunikasi serial. Data yang dikirimkan meliputi berat pakan dalam wadah, volume air dalam wadah dalam mililiter (mL), volume air dalam tabung dalam mililiter (mL), status pompa (pumpStatus), status servo (servoStatus), serta informasi waktu berupa keterangan hari (ketHari) dan waktu (ketWaktu).

Data yang akan dikirimkan dikombinasikan menjadi satu string dengan

pemisah karakter # untuk memudahkan proses parsing data pada sisi Receiver. Fungsi Serial.println() digunakan untuk mengirimkan data tersebut melalui jalur serial ke Receiver yang menggunakan Arduino Uno R3 Built-In IoT WiFi ESP8266.

Dengan adanya fungsi ini, sistem dapat mengirimkan data secara real-time ke perangkat penerima, memungkinkan pengguna untuk memantau kondisi pakan dan air dari jarak jauh melalui platform IoT. Hal ini juga mendukung proses pengendalian yang lebih efektif, karena data monitoring yang akurat dapat langsung diterima dan diolah oleh Receiver untuk tindakan lebih lanjut.

Kode Program 5.16 Kode Program Mengirimkan Data pada Receiver
Arduino Uno R3 Built-In IoT WiFi ESP8266

```
void sendDataToReceiver(int beratWadah, int volumeMLAirWadah, int
volumeMLAirTabung) {
    sendDataToEsp=String(beratWadah)+"#" + String(volumeMLAirWadah)
    +"#" +String(volumeMLAirTabung) + " #" + String(pumpStatus) + "#"
    +String(servоСtatus)+"#" +String(ketHari)+"#" +String(ketWaktu);
    Serial.println(sendDataToEsp);
}
```

17. Fungsi Penerimaan Data dari Receiver

Fungsi reqDataFromReceiver() digunakan untuk menerima perintah dari perangkat receiver melalui komunikasi serial, yang kemudian digunakan untuk mengontrol servo dan pompa berdasarkan perintah yang diterima. Fungsi ini memeriksa apakah terdapat data yang tersedia di serial input menggunakan Serial.available(), dan jika ada, data akan dibaca sebagai string dengan Serial.readStringUntil('\n') dan diproses lebih lanjut.

Jika perintah yang diterima adalah "Pump_ON", maka fungsi onPump() akan dipanggil untuk mengaktifkan pompa, dan activatePump(true) akan menampilkan pesan status pompa yang menyala. Sebaliknya, jika perintah adalah "Pump_OFF", relay akan dimatikan dan activatePump(false) akan menampilkan pesan bahwa pompa telah dimatikan.

Demikian pula, jika perintah yang diterima adalah "Servo_ON", maka bukaServo(4) akan dipanggil untuk menggerakkan servo, dan activateServo(true)

akan menginformasikan bahwa servo dalam keadaan aktif. Perintah "Servo_OFF" akan menghentikan pergerakan servo dengan mengatur posisinya kembali ke nol dan mengubah status servoStatus menjadi false melalui activateServo(false).

Fungsi reqDataFromReceiver() memungkinkan sinkronisasi antara transmitter dan receiver dalam sistem IoT ini, memastikan bahwa setiap instruksi dari receiver dapat diterjemahkan dengan baik ke dalam tindakan fisik yang dioperasikan oleh servo dan pompa.

Kode Program 5.17 Kode Program Penerimaan Data dari Receiver pada Transmitter Arduino UNO R3 Built-In IoT WiFi ESP8266

```
void reqDataFromReceiver() {  
    if (Serial.available()) {  
        String receivedCommand = Serial.readStringUntil('\n');  
        receivedCommand.trim();  
        if (receivedCommand == "Pump_ON") {  
            onPump();  
            activatePump(true);  
        } else if (receivedCommand == "Pump_OFF") {  
            digitalWrite(relayPin, LOW);  
            pumpStatus = false;  
            activatePump(false);  
        }  
        if (receivedCommand == "Servo_ON") {  
            bukaServo(4);  
            servoStatus = true;  
            activateServo(true);  
        } else if (receivedCommand == "Servo_OFF") {  
            myServo.write(0);  
            servoStatus = false;  
            activateServo(false);  
        }  
    }  
}
```

5.4.2 Kode Program Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266

Kodingan receiver pada Arduino Uno R3 Built-in IoT WiFi ESP8266 berfungsi sebagai penerima data yang dikirimkan dari perangkat transmitter. Perangkat receiver ini menerima data yang berisi informasi mengenai berat pakan dalam wadah, volume air dalam wadah, volume air dalam tabung, serta status dari aktuator seperti pompa dan servo. Data ini dikirimkan melalui komunikasi serial dari transmitter dan diterima oleh receiver untuk kemudian diolah lebih lanjut.

Receiver menggunakan kemampuan konektivitas WiFi dari modul ESP8266 yang tertanam pada Arduino Uno R3 untuk mengintegrasikan data tersebut dengan sistem Internet of Things (IoT). Data yang diterima dapat diteruskan ke basis data cloud seperti Firebase untuk keperluan monitoring dan kontrol jarak jauh. Dengan demikian, pengguna dapat memantau kondisi pakan dan air, serta mengendalikan perangkat melalui aplikasi mobile yang terhubung dengan sistem IoT.

Kode ini memungkinkan sistem untuk berfungsi secara otomatis dan terhubung ke internet, yang meningkatkan fleksibilitas dan kemudahan pengguna dalam memonitor dan mengontrol sistem pemberian makan dan minum secara real-time

1. Library, Define, dan Deklarasi variabel yang Digunakan

Pada bagian ini, program menggunakan beberapa library dan mendeklarasikan variabel-variabel yang dibutuhkan untuk menjalankan sistem receiver pada Arduino Uno R3 Built-in IoT WiFi ESP8266. Library seperti ESP8266WiFi.h digunakan untuk mengelola koneksi WiFi pada perangkat ESP8266, sementara FirebaseESP8266.h digunakan untuk berkomunikasi dengan Firebase Realtime Database. Library tambahan seperti TokenHelper.h dan RTDBHelper.h membantu dalam pengelolaan autentikasi dan pengiriman data ke Firebase.

Bagian #define digunakan untuk mendefinisikan kredensial seperti SSID dan password WiFi yang digunakan untuk menghubungkan perangkat ke jaringan, serta API key, URL database, dan akun pengguna untuk akses ke Firebase. Deklarasi variabel seperti FirebaseDatabase, FirebaseAuth, DatabaseReference, dan FirebaseFirestore diperlukan untuk mengelola koneksi dan interaksi dengan Firebase, termasuk

autentikasi dan pengiriman data monitoring.

Variabel waktu sendToFirebasePrevMillis dan receiveFromFirebasePrevMillis digunakan untuk mengatur interval pengiriman dan penerimaan data antara perangkat dan Firebase, dengan tujuan menjaga sinkronisasi data secara berkala. Dengan menggunakan pengaturan ini, perangkat dapat mengirim data hasil monitoring serta menerima data kontrol dari cloud pada interval waktu tertentu.

Kode Program 5.18 Kode Program Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266

```
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"
#define WIFI_SSID "Tugasakhir"
#define WIFI_PASSWORD "wifisigit"
#define API_KEY "AIzaSyD9cMliTs9G41vgRLcjS2VacvtMWWR1doQ"
#define DATABASE_URL "tugas-akhir-3c0d9-default-rtdb.firebaseio.com"
#define USER_EMAIL "mhsigit01@gmail.com"
#define USER_PASSWORD "muhammadsigit292001"
FirebaseData firebaseData;
FirebaseAuth auth;
FirebaseConfig config;
FirebaseJson monitoringJson;
FirebaseJsonData jsonData;
String uid, databasePath, monitoringNode, controlNode;
unsigned long sendToFirebasePrevMillis = 0;
unsigned long sendToFirebaseDelay = 3000;
unsigned long receiveFromFirebasePrevMillis = 0;
unsigned long receiveFromFirebaseDelay = 2700;
```

2. Void Setup

Bagian void setup pada kode program receiver bertanggung jawab untuk menginisialisasi pengaturan awal yang diperlukan agar sistem dapat berfungsi

dengan baik. Pada tahap ini, beberapa proses dijalankan untuk mempersiapkan perangkat Arduino Uno R3 dengan modul IoT WiFi ESP8266 agar siap berkomunikasi dengan jaringan WiFi dan Firebase.

Instruksi Serial.begin(9600); digunakan untuk memulai komunikasi serial dengan baud rate 9600, yang memungkinkan perangkat untuk mengirim dan menerima data melalui monitor serial untuk keperluan debugging dan pemantauan proses.

Selanjutnya, fungsi initWiFi(); dipanggil untuk menginisialisasi koneksi WiFi, memastikan perangkat dapat terhubung dengan jaringan yang telah ditentukan. Fungsi initFirebase(); digunakan untuk mengatur koneksi dengan Firebase Realtime Database, termasuk autentikasi pengguna dan konfigurasi dasar yang diperlukan untuk berinteraksi dengan database.

Akhirnya, delay(500); memberikan jeda singkat untuk memastikan semua inisialisasi berjalan lancar sebelum perangkat melanjutkan ke proses lainnya. Bagian ini penting untuk memastikan bahwa perangkat siap untuk menerima dan mengirim data sesuai dengan kebutuhan sistem.

Kode Program 5.19 Void Setup Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266

```
void setup() {  
    Serial.begin(9600);  
    initWiFi();  
    initFirebase();  
    delay(500);  
}
```

3. Void Loop

Bagian void loop pada kode program receiver bertanggung jawab untuk menjalankan proses utama secara berulang, sehingga sistem dapat terus memantau dan memperbarui data. Pada tahap ini, Arduino Uno R3 dengan modul IoT WiFi ESP8266 menjalankan serangkaian instruksi untuk memastikan perangkat tetap terhubung dengan Firebase Realtime Database dan berkomunikasi dengan baik dengan sistem pengirim data.

Fungsi Firebase.isTokenExpired() digunakan untuk memeriksa apakah token

autentikasi Firebase sudah kadaluarsa. Jika token kadaluarsa, maka `Firebase.refreshToken(&config);` akan memperbarui token tersebut untuk memastikan koneksi tetap aman dan stabil. Informasi terkait proses ini ditampilkan melalui `Serial.println("Memperbarui Token");` agar pengguna dapat memantau aktivitas ini melalui monitor serial.

Jika koneksi dengan Firebase dalam kondisi siap (`Firebase.ready()`), dua fungsi utama dipanggil: `receiveDataFromTransmitter()` untuk menerima data dari sistem pengirim (transmitter) dan `receiveDataFromDatabase()` untuk mengambil data dari Firebase Realtime Database. Proses ini memastikan bahwa perangkat dapat menerima data terbaru dari sensor dan memperbarui data kontrol sesuai kebutuhan.

Jika kondisi di atas tidak terpenuhi, maka `Serial.println("Kodingan Loop berjalan");` digunakan untuk menampilkan status bahwa loop masih berjalan, yang berguna untuk memantau jalannya program saat koneksi sedang dalam proses atau tidak tersedia. Bagian void loop ini penting untuk menjaga komunikasi yang berkesinambungan antara perangkat dengan jaringan dan database, serta memastikan semua data yang diperlukan selalu diperbarui.

Kode Program 5.20 Void Loop Receiver pada Arduino Uno R3 Built-In IoT WiFi ESP8266

```
void loop() {
    if (Firebase.isTokenExpired()) {
        Firebase.refreshToken(&config);
        Serial.println("Memperbarui Token");
    } else if (Firebase.ready()) {
        receiveDataFromTransmitter();
        receiveDataFromDatabase();
    } else {
        Serial.println("Kodingan Loop berjalan");
    }
}
```

4. Fungsi Initialisasi Jaringan WiFi

Fungsi `initWiFi()` digunakan untuk menginisialisasi koneksi jaringan WiFi pada perangkat Arduino Uno R3 dengan modul IoT WiFi ESP8266. Fungsi ini

bertujuan untuk memastikan perangkat dapat terhubung ke jaringan internet dengan menggunakan SSID dan password yang telah didefinisikan sebelumnya.

Pada bagian awal, WiFi.begin(WIFI_SSID, WIFI_PASSWORD); dipanggil untuk memulai koneksi WiFi. Pesan "Mencoba Menghubungkan ke WiFi" akan ditampilkan pada monitor serial untuk memberikan informasi kepada pengguna mengenai status koneksi.

Kemudian, dengan menggunakan perulangan while (WiFi.status() != WL_CONNECTED), program akan terus mencoba untuk terhubung ke jaringan WiFi sampai perangkat berhasil tersambung. Setiap kali upaya koneksi dilakukan, karakter titik (".") akan ditampilkan di monitor serial setiap detik (delay(1000);), yang menandakan proses koneksi sedang berlangsung.

Setelah perangkat berhasil terhubung ke jaringan WiFi, fungsi ini akan menampilkan pesan "Terhubung dengan IP ADDRESS :" diikuti dengan alamat IP lokal yang diperoleh perangkat melalui WiFi.localIP(). Informasi alamat IP ini penting untuk memastikan bahwa perangkat telah terhubung dengan benar ke jaringan, sehingga dapat berkomunikasi dengan Firebase dan menerima data dari sistem transmitter.

Kode Program 5.21 Inisialisasi Jaringan WiFi

```
void initWiFi() {
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Mencoba Menghubungkan ke WiFi");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(1000);
    }
    Serial.println();
    Serial.println("Terhubung dengan IP ADDRESS : ");
    Serial.println(WiFi.localIP());
}
```

5. Fungsi Inisialisasi Firebase

Fungsi initFirebase() digunakan untuk menginisialisasi koneksi antara perangkat Arduino Uno R3 yang dilengkapi dengan modul IoT WiFi ESP8266 dan

Firebase Realtime Database. Tujuan dari fungsi ini adalah untuk memastikan perangkat dapat mengakses dan berkomunikasi dengan Firebase untuk keperluan penyimpanan data monitoring dan pengambilan data kontrol.

Pada fungsi ini, beberapa parameter penting dikonfigurasi, seperti config.api_key, auth.user.email, auth.user.password, dan config.database_url yang digunakan untuk otentikasi dan pengaturan akses ke basis data di Firebase. Fungsi Firebase.reconnectWiFi(true); memungkinkan perangkat untuk secara otomatis terhubung kembali ke jaringan WiFi jika koneksi terputus.

Fungsi ini juga mengatur ukuran respons Firebase dengan firebaseData.setResponseSize(4096); untuk memastikan bahwa data yang diterima atau dikirim dapat ditangani dengan benar. Selain itu, config.token_status_callback digunakan untuk memantau status token autentikasi, dan config.max_token_generation_retry diatur untuk mencoba hingga 5 kali jika terjadi kegagalan dalam pembuatan token.

Setelah parameter konfigurasi selesai, fungsi Firebase.begin(&config, &auth); memulai koneksi ke Firebase dengan menggunakan konfigurasi dan informasi autentikasi yang telah ditetapkan. Untuk memastikan bahwa autentikasi berhasil, perangkat akan menunggu hingga User UID berhasil diperoleh melalui auth.token.uid. UID ini digunakan sebagai identitas pengguna untuk mengakses jalur database yang sesuai, yaitu "/UsersData/" + uid. Dengan adanya UID, perangkat dapat menyimpan dan mengelola data dengan lebih terstruktur dan aman dalam Firebase Realtime Database.

Kode Program 5.22 Inisialisasi Firebase Real-Time Database

```
void initFirebase() {  
    config.api_key = API_KEY;  
    auth.user.email = USER_EMAIL;  
    auth.user.password = USER_PASSWORD;  
    config.database_url = DATABASE_URL;  
    Firebase.reconnectWiFi(true);  
    firebaseData.setResponseSize(4096);  
    config.token_status_callback = tokenStatusCallback;  
    config.max_token_generation_retry = 5;
```

```
    Firebase.begin(&config, &auth);
    Serial.println("Mendapatkan User UID");
    while ((auth.token.uid) == "") {
        Serial.print(".");
        delay(1000);
    }
    uid = auth.token.uid.c_str();
    Serial.print("User UID : ");
    Serial.print(uid);
    Serial.println();
    databasePath = "/UsersData/" + uid;
}
```

6. Fungsi Penerimaan Data dari Transmitter

Fungsi receiveDataFromTransmitter() bertanggung jawab untuk menerima data dari transmitter yang terhubung melalui koneksi serial. Proses penerimaan data dimulai dengan memeriksa apakah ada data yang tersedia untuk dibaca pada serial. Data yang diterima disimpan dalam variabel receivedData hingga mencapai akhir baris yang ditandai dengan karakter newline ('\n').

Setelah data diterima, fungsi akan melakukan pemrosesan untuk mengekstrak informasi yang dipisahkan oleh tanda '#' dalam string. Tanda ini digunakan sebagai pemisah untuk mengidentifikasi bagian-bagian dari data yang diterima, seperti berat wadah, volume air di tabung, volume air di wadah, status pompa, status servo, dan informasi hari serta waktu.

Fungsi ini menggunakan metode indexOf() untuk menemukan posisi dari setiap tanda pemisah. Data yang diperoleh kemudian diambil menggunakan substring() dan dikonversi ke tipe yang sesuai, seperti int untuk berat dan volume, dan bool untuk status pompa dan servo. Setelah semua data berhasil diekstrak, fungsi sendDataToFirebase() dipanggil untuk mengirimkan data yang diterima ke Firebase Realtime Database untuk penyimpanan dan pengelolaan lebih lanjut.

Dengan demikian, fungsi ini memungkinkan sistem untuk beroperasi secara real-time, menerima data monitoring dari transmitter, dan menyimpannya dalam basis data untuk analisis lebih lanjut.

Kode Program 5.23 Kode Program Penerimaan data dari Transmitter pada Arduino Uno R3 Built-In IoT WiFi ESP8266

```
void receiveDataFromTransmitter() {
    String receivedData = "";
    while (Serial.available()) {
        receivedData = Serial.readStringUntil('\n');
    }
    if (receivedData.length() > 0) {
        int separator1 = receivedData.indexOf('#');
        int separator2 = receivedData.indexOf('#', separator1 + 1);
        int separator3 = receivedData.indexOf('#', separator2 + 1);
        int separator4 = receivedData.indexOf('#', separator3 + 1);
        int separator5 = receivedData.indexOf('#', separator4 + 1);
        int separator6 = receivedData.indexOf('#', separator5 + 1);
        int beratWadah = receivedData.substring(0, separator1).toInt();
        int volumeMLTabung = receivedData.substring(separator1 + 1, separator2).toInt();
        int volumeMLWadah = receivedData.substring(separator2 + 1, separator3).toInt();
        bool pumpStatus = receivedData.substring(separator3 + 1, separator4) == "1";
        bool servoStatus = receivedData.substring(separator4 + 1, separator5) == "1";
        String ketHari = receivedData.substring(separator5 + 1, separator6);
        String ketWaktu = receivedData.substring(separator6+1);
        sendDataToFirebase(beratWadah, volumeMLTabung, volumeMLWadah, pumpStatus, servoStatus, ketHari, ketWaktu);
    }
}
```

```
    }  
}
```

7. Fungsi Pengiriman Data ke Firebase

Fungsi sendDataToFirebase() bertugas untuk mengirimkan data monitoring dari receiver ke Firebase Realtime Database. Proses pengiriman dimulai dengan mencatat waktu saat fungsi dipanggil menggunakan fungsi millis(), yang memberikan waktu dalam milidetik sejak program dimulai. Fungsi ini memastikan bahwa data hanya dikirim pada interval tertentu dengan memeriksa selisih waktu antara pemanggilan terakhir dan waktu saat ini. Jika selisihnya melebihi batas yang ditentukan oleh sendToFirebaseDelay, atau jika fungsi ini dipanggil untuk pertama kalinya, maka data akan dipersiapkan untuk dikirim.

Data yang akan dikirim mencakup berat wadah, volume air di tabung, volume air di wadah, status pompa, status servo, serta informasi hari dan waktu. Data ini diatur dalam objek JSON menggunakan metode set() pada monitoringJson, yang merupakan objek dari kelas FirebaseJson. Setelah semua data disiapkan, fungsi ini menggunakan Firebase.setJSON() untuk mengirimkan data JSON ke node yang telah ditentukan di Firebase.

Jika pengiriman berhasil, fungsi ini akan mencetak informasi mengenai data yang terkirim, termasuk path dan tipe data, serta nilai yang dikirim ke konsol serial untuk debugging. Namun, jika terjadi kegagalan dalam pengiriman data, fungsi akan menampilkan pesan kesalahan beserta alasan kegagalan, sehingga memudahkan proses pemecahan masalah.

Dengan demikian, fungsi ini memainkan peran penting dalam menjaga integrasi antara sistem monitoring dan penyimpanan data berbasis cloud, memungkinkan data untuk diperoleh dan dianalisis secara real-time.

Kode Program 5.24 Kode Pengiriman data pada Receiver ke Firebase Real-Time Database

```
void sendDataToFirebase(int beratWadah, int volumeMLTabung, int  
volumeMLWadah, bool pumpStatus, bool servoStatus, String ketHari,  
String ketWaktu) {  
    unsigned long currentMillis = millis();  
    if(currentMillis-sendToFirebasePrevMillis>sendToFirebaseDelay
```

```

    || sendToFirebasePrevMillis== 0) {
        sendToFirebasePrevMillis = currentMillis;
        monitoringNode = databasePath + "/iot/monitoring";
        monitoringJson.set("beratWadah", beratWadah);
        monitoringJson.set("volumeMLTabung", volumeMLTabung);
        monitoringJson.set("volumeMLWadah", volumeMLWadah);
        monitoringJson.set("pumpStatus", pumpStatus);
        monitoringJson.set("servoStatus", servoStatus);
        monitoringJson.set("ketHari", ketHari);
        monitoringJson.set("ketWaktu", ketWaktu);
        if(Firebase.setJSON(firebaseData, monitoringNode.c_str(),
            monitoringJson)) {
            Serial.println("DATA TERKIRIM");
            Serial.println("PATH : " + firebaseData.dataPath());
            Serial.println("TYPE: " + firebaseData.dataType());
            Serial.print("VALUE: ");
            printResult(firebaseData);
            Serial.println("-----");
        } else {
            Serial.println("GAGAL MENGIRIM DATA");
            Serial.println("Error:" + firebaseData.errorReason());
            Serial.println("-----");
        }
    }
}

```

8. Fungsi Pengiriman Data kendali Ke Arduino

Fungsi sendDataToTransmitter() digunakan untuk mengirimkan perintah kendali dari receiver ke transmitter. Dalam fungsi ini, perintah yang diberikan dalam bentuk string dicetak ke konsol serial menggunakan Serial.println(), yang memungkinkan pemantauan dan verifikasi perintah yang dikirim selama pengujian dan debugging.

Fungsi receiveDataFromDatabase() berperan penting dalam mengambil data kontrol dari Firebase Realtime Database. Fungsi ini menggunakan pengukuran

waktu untuk memastikan bahwa pengambilan data hanya dilakukan pada interval tertentu, yang ditentukan oleh receiveFromFirebaseDelay. Setelah memeriksa apakah cukup waktu telah berlalu sejak pengambilan terakhir, fungsi ini membangun node kontrol dengan menggabungkan databasePath dan path yang relevan.

Ketika data berhasil diambil, fungsi ini mengekstrak nilai untuk kontrol pompa dan servo dari objek JSON yang diterima. Nilai-nilai ini digunakan untuk menentukan apakah pompa dan servo harus dinyalakan atau dimatikan. Jika pumpControl bernilai true, maka fungsi akan mengirim perintah "Pump_ON" ke transmitter; sebaliknya, jika bernilai false, perintah "Pump_OFF" akan dikirim. Demikian pula, kontrol untuk servo ditangani dengan cara yang sama, mengirimkan perintah "Servo_ON" atau "Servo_OFF" berdasarkan nilai yang diterima.

Apabila pengambilan data dari Firebase gagal, fungsi ini mencetak pesan kesalahan yang mencakup alasan kegagalan, sehingga memudahkan identifikasi masalah dalam komunikasi antara receiver dan Firebase.

Dengan mekanisme ini, fungsi sendDataToTransmitter() dan receiveDataFromDatabase() bekerja secara sinergis untuk menjaga kendali yang efektif atas komponen sistem, menjamin responsivitas dan fungsi sistem otomatis yang diinginkan.

Kode Program 5.25 Pengiriman Data pada Receiver ke Transmitter

```
void sendDataToTransmitter(String command) {
    Serial.println(command);
}

void receiveDataFromDatabase() {
    unsigned long currentMillis = millis();
    if(currentMillis-receiveFromFirebasePrevMillis
        >receiveFromFirebaseDelay || receiveFromFirebasePrevMillis
        == 0) {
        receiveFromFirebasePrevMillis = currentMillis;
        controlNode = databasePath + "/iot/control";
        if (Firebase.getJSON(firebaseData, controlNode.c_str())) {
            FirebaseJson controlJson = firebaseData.jsonObject();
            bool pumpControl, servoControl;
```

```

controlJson.get(jsonData, "pumpControl");
pumpControl = jsonData.boolValue;
controlJson.get(jsonData, "servoControl");
servoControl = jsonData.boolValue;
if (pumpControl == true) {
    sendDataToTransmitter("Pump_ON");
} else {
    sendDataToTransmitter("Pump_OFF");
}
if (servoControl == true) {
    sendDataToTransmitter("Servo_ON");
} else {
    sendDataToTransmitter("Servo_OFF");
}
} else {
    Serial.println("Error mendapatkan data kontrol: " +
firebaseData.errorReason());
}
}
}
}

```

5.5 Pengujian Sistem

Pengujian sistem terdiri dari pengujian perangkat keras dan pengujian perangkat lunak. Tujuan dari pengujian ini adalah untuk mengetahui bahwa semua fungsi dalam sistem berjalan sebagaimana yang diharapkan. Pengujian perangkat keras melibatkan evaluasi kontrol otomatis, sementara pengujian pengujian blackbox.

Tahap pengujian sistem dilakukan setelah tahap implementasi selesai dan tahap ini tahap penentuan untuk memastikan sebuah sistem berfungsi dengan baik atau tidak. Tahapan dari pengujian perangkat keras dan pengujian perangkat lunak. Proses pengujian ini bertujuan untuk memastikan semua fungsi sistem dapat berjalan sesuai dengan hasil yang diharapkan. Proses pengujian meliputi pengujian pembacaan dan akurasi sensor DHT22, pengujian pembacaan sensor cahaya LDR, pengujian pembacaan dan akurasi sensor jarak Ultrasonik, pengujian keseluruhan

sistem dan pengujian pemantauan website, pengujian black box.

5.5.1 Pengujian Akurasi Penjadwalan Real Time Clock

Aa



Gambar 5.23 Pengujian Akurasi Penjadwalan Real Time Clock

a

Tabel 5.1 Tabel Hasil Pengujian Akurasi Penjadwalan Real Time Clock

No	Hari	Jadwal	Servo	Mini Pump Water
1.	Hari ke-1	Jadwal Pagi	Hidup	Hidup
		Jadwal Sore	Hidup	Hidup
2.	Hari ke-2	Jadwal Pagi	Hidup	Hidup
		Jadwal Sore	Hidup	Hidup
3.	Hari ke-3	Jadwal Pagi	Hidup	Hidup
		Jadwal Sore	Hidup	Hidup

4.	Hari ke-4	Jadwal Pagi	Hidup	Hidup
		Jadwal Sore	Hidup	Hidup
5.	Hari ke-5	Jadwal Pagi	Hidup	Hidup
		Jadwal Sore	Hidup	Hidup
6.	Hari ke-6	Jadwal Pagi	Hidup	Hidup
		Jadwal Sore	Hidup	Hidup
7.	Hari ke-7	Jadwal Pagi	Hidup	Hidup
		Jadwal Sore	Hidup	Hidup

5.5.2 Pengujian Pembacaan dan Akurasi Pengukuran Berat Wadah Makan

A



Gambar 5.24 Pengujian Pembacaan dan Akurasi Pengukuran Berat Wadah Makan

A

Tabel 5.2 Tabel Hasil Pengujian Pembacaan dan Akurasi Pengukuran Berat Wadah Makan

No	Hari	Jadwal	Berat Wadah	Berat Badan Kucing

			(gram)	
1.	Hari ke-1	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
2.	Hari ke-2	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
3.	Hari ke-3	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
4.	Hari ke-4	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
5.	Hari ke-5	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
6.	Hari ke-6	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
7.	Hari ke-7	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup

5.5.3 Pengujian Pembacaan dan Akurasi Pengukuran Volume Tabung Air Minum

A



Gambar 5.25 Pengujian Pembacaan dan Akurasi Pengukuran Volume Tabung Air Minum

A

Tabel 5.3 Tabel Hasil Pengujian Pembacaan dan Akurasi Pengukuran Volume Tabung Air Minum

No	Hari	Jadwal	Berat Wadah (gram)	Berat Badan Kucing
1.	Hari ke-1	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
2.	Hari ke-2	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
3.	Hari ke-3	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
4.	Hari ke-4	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup

5.	Hari ke-5	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
6.	Hari ke-6	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
7.	Hari ke-7	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup

5.5.4 Pengujian Kendali Mini Pump Water

A



Gambar 5.26 Pengujian Kendali Mini Pump Water

A

Tabel 5.4 Tabel Hasil Pengujian Kendali Mini Pump Water

No	Gelas Ukur (mL)	Ultrasonic (mL)	Selisih (mL)	Error (%)	Akurasi (%)
1.					
2.					
3.					

5.5.5 Pengujian Kendali Servo

A



Gambar 5.27 Pengujian Kendali Motor Servo

A

Tabel 5.5 Tabel Hasil Pengujian Kendali Motor Servo

No	Gelas Ukur (mL)	Ultrasonic (mL)	Selisih (mL)	Error (%)	Akurasi (%)
1.					
2.					
3.					

5.5.6 Pengujian Pembacaan dan Akurasi Pengukuran Volume Wadah Air Minum

A



Gambar 5.28 Pengujian Pembacaan dan Akurasi Pengukuran Volume Wadah Air Minum

A

Tabel 5.6 Tabel Hasil Pengujian Pembacaan dan Akurasi Pengukuran Volume Wadah Air Minum

No	Hari	Jadwal	Berat Wadah (gram)	Berat Badan Kucing

	Hari ke-1	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
2.	Hari ke-2	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
3.	Hari ke-3	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
4.	Hari ke-4	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
5.	Hari ke-5	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
6.	Hari ke-6	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup
7.	Hari ke-7	Jadwal Pagi	120	Hidup
		Jadwal Sore	120	Hidup

5.5.7 Pengujian Keseluruhan Sistem

A



Gambar 5.29 Implementasi Pengujian Keseluruhan Sistem

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Tabel 5.7 Tabel Pengujian Keseluruhan Sistem

No	Hari	Penjadwalan	Wadah Minum (mL)	Tabung Minum (mL)	Wadah Makan (gram)	Servo	Mini Pump Water
1.	Hari ke-1	Jadwal Pagi	120	120	120	Hidup	Hidup
		Jadwal Sore	120	120	120	Hidup	Hidup
2.	Hari ke-2	Jadwal Pagi	120	120	120	Hidup	Hidup
		Jadwal Sore	120	120	120	Hidup	Hidup
3.	Hari ke-3	Jadwal Pagi	120	120	120	Hidup	Hidup
		Jadwal Sore	120	120	120	Hidup	Hidup
4.	Hari ke-4	Jadwal Pagi	120	120	120	Hidup	Hidup
		Jadwal Sore	120	120	120	Hidup	Hidup
5.	Hari ke-5	Jadwal Pagi	120	120	120	Hidup	Hidup
		Jadwal Sore	120	120	120	Hidup	Hidup
6.	Hari ke-6	Jadwal Pagi	120	120	120	Hidup	Hidup

		Jadwal Sore	120	120	120	Hidup	Hidup
7.	Hari ke-7	Jadwal Pagi	120	120	120	Hidup	Hidup
		Jadwal Sore	120	120	120	Hidup	Hidup

5.5.8 Pengujian Black Box

A

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Login	Menu <i>Login</i>	Menampilkan halaman <i>login</i>	Berhasil
		<i>Form Email</i>	Pengguna bisa memasukkan alamat <i>email</i> pada <i>form email</i>	Berhasil
		<i>Form Password</i>	Pengguna bisa memasukkan <i>password</i> pada <i>form password</i>	Berhasil
		Memasukkan <i>email</i> dan <i>password</i> yang sudah terdaftar, lalu menekan tombol login	Sistem menampilkan halaman home dan notifikasi login sukses.	Berhasil
		Memasukkan <i>email</i> dan <i>password</i> yang belum terdaftar, lalu menekan	Sistem memeriksa <i>email</i> dan <i>password</i> , jika belum terdaftar maka muncul modal untuk memasukkan	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
2.	Lupa Password	tombol masuk dan memunculkan modal untuk status <i>login</i>	<i>email</i> dan <i>password</i> tidak terdaftar.	
		Tombol Login	Sistem melakukan verifikasi <i>email</i> dan <i>password</i>	Berhasil
		Tombol Lupa <i>Password</i>	Sistem menampilkan halaman lupa <i>password</i>	Berhasil
2.	Lupa Password	Menu Lupa <i>Password</i>	Menampilkan halaman lupa <i>password</i>	Berhasil
		<i>Form Email</i>	Pengguna bisa memasukkan alamat <i>email</i> yang sudah terdaftar pada <i>form email</i>	Berhasil
		Memasukkan <i>email</i> yang sudah terdaftar, lalu menekan tombol <i>reset password</i> .	Sistem mengirimkan form ke alamat <i>email</i> pengguna yang sudah terdaftar untuk mengubah <i>password</i> yang lama menjadi <i>password</i> yang baru.	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
3.	Home	Memasukkan email yang belum terdaftar, lalu menekan tombol <i>reset password</i>	Sistem memeriksa alamat <i>email</i> , jika belum terdaftar maka muncul modal untuk memasukkan alamat <i>email</i> yang sudah terdaftar	Berhasil
		Tombol <i>Reset Password</i>	Sistem melakukan <i>verifikasi email</i> dan <i>password</i> .	Berhasil
		Menu Home	Menampilkan halaman home	Berhasil
3.	Home	Menampilkan status <i>morning feeder</i> dan <i>afternoon feeder</i>	Menampilkan status pemberian makan dan minum kucing pada penjadwalan pagi dan sore hari.	Berhasil
		Nilai kondisi penjadwalan pemberian jumlah makanan dan minuman secara harian.	Menampilkan jumlah pemberian makanan dan minuman harian secara real-time.	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
	4. Feeder	Menampilkan alamat feeder	Menampilkan alamat pemberian makan dan minum kucing.	Berhasil
		Menu feeder	Menampilkan halaman feeder	Berhasil
		Menampilkan kendali morning feeder	Menampilkan halaman otomatisasi pemberian makan dan minum kucing pada pagi hari.	Berhasil
	5. Main	Menampilkan kendali afternoon feeder	Menampilkan halaman otomatisasi pemberian makan dan minum kucing pada sore hari.	Berhasil
		Menu Main	Menampilkan halaman main	Berhasil
		Tombol Automatic	Menampilkan pilihan mode untuk mengaktifkan pemberian makan dan minum kucing secara otomatis	Berhasil
		Tombol on/off servo	Mengendalikan on/off servo untuk melakukan pemberian makan pada wadah makanan kucing	Berhasil
		Tombol on/off pump	Mengendalikan on/off servo untuk melakukan pemberian minum pada wadah minuman kucing.	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
5.	Status Alat	Tombol Cek Status Alat	Menampilkan halaman status alat.	Berhasil
		Tombol Log Data	Menampilkan halaman log data.	Berhasil
		Menampilkan jumlah data sensor	Menampilkan nilai data sensor untuk monitoring pemberian makan dan minum kucing secara harian dan mingguan.	Berhasil
		Tombol Tambah Jadwal	Menampilkan halaman untuk formulir tambah penjadwalan.	Berhasil
		Menampilkan data penjadwalan pada tabel kalender.	Menampilkan data hasil dari penjadwalan pemberian makan dan minum kucing.	Berhasil
6.	Statistik Data	Menu Status Alat	Menampilkan halaman status alat.	Berhasil
		Menampilkan form operasional perangkat pemberian makan dan minum kucing.	Menampilkan informasi status operasional perangkat pemberian makan dan minum kucing	Berhasil
7.	Statistik Data	Menu Log Data	Menampilkan halaman log data	Berhasil
		Menampilkan Data Statistik Penjadwalan	Menampilkan informasi chart data penjadwalan pemberian makan dan minum.	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
8.	List Data	Menampilkan Data IoT	Menampilkan informasi status sensor pada wadah makan dan minum kucing berupa <i>radial bar chart</i> secara <i>realtime</i> .	Berhasil
		Menampilkan Statistik Data Feeder	Menampilkan informasi data feeder dalam bentuk <i>chart</i> secara <i>realtime</i>	Berhasil
		Menu Data Latih	Menampilkan Halaman Data Latih	Berhasil
		Menampilkan Data Jadwal Pagi	Menampilkan Data Penjadwalan Pagi Hari dalam bentuk <i>list data</i>	Berhasil
		Menampilkan Data Jadwal Sore	Menampilkan Data Penjadwalan Sore Hari dalam bentuk <i>list data</i>	Berhasil
		Tombol Edit	Menampilkan form untuk mengedit data jadwal	Berhasil
		Tombol Delete	Menampilkan form untuk menghapus data jadwal	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
9.	Tambah Jadwal	Tombol Tambah	Menampilkan form untuk menambah data jadwal	Berhasil
		Menampilkan Data Feeder Pagi	Menampilkan Data Feeder Pagi dalam bentuk <i>list data</i>	Berhasil
		Menampilkan Data Feeder Sore	Menampilkan Data Feeder Sore dalam bentuk <i>list data</i>	Berhasil
		Menu Tambah Jadwal	Menampilkan form untuk tambah jadwal	Berhasil
		Form Kalendar	Pengguna bisa memilih tanggal untuk penjadwalan	Berhasil
		Form Waktu	Pengguna bisa memilih waktu untuk penjadwalan	Berhasil
		Form Judul	Pengguna bisa memasukkan judul untuk penjadwalan	Berhasil
		Form Deskripsi	Pengguna bisa memasukkan deskripsi untuk penjadwalan	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
10.	Edit Jadwal	Form Makanan	Pengguna bisa memasukkan jumlah makanan untuk penjadwalan	Berhasil
		Form Minuman	Pengguna bisa memasukkan jumlah minuman untuk penjadwalan	Berhasil
		Tombol Tambah Jadwal	Sistem menyimpan data penjadwalan	Berhasil
		Tombol Cancel	Sistem membatalkan penambahan jadwal	Berhasil
		Menu Edit Jadwal	Menampilkan form untuk mengedit jadwal	Berhasil
		Form Kalendar	Pengguna bisa memilih tanggal untuk mengedit penjadwalan	Berhasil
		Form Waktu	Pengguna bisa memilih waktu untuk mengedit penjadwalan	Berhasil
		Form Judul	Pengguna bisa mengedit judul untuk penjadwalan	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
11.		Form Deskripsi	Pengguna bisa mengedit deskripsi untuk penjadwalan	Berhasil
		Form Makanan	Pengguna bisa mengedit jumlah makanan untuk penjadwalan	Berhasil
		Form Minuman	Pengguna bisa mengedit jumlah minuman untuk penjadwalan	Berhasil
		Tombol Edit Jadwal	Sistem menyimpan perubahan data penjadwalan	Berhasil
		Tombol Cancel	Sistem membatalkan pengeditan jadwal	Berhasil
11.	Settings	Menu <i>settings</i>	Menampilkan halaman <i>settings</i>	Berhasil
		Tombol <i>update profile</i>	Menampilkan form untuk <i>update profile</i>	Berhasil
		Tombol status alat	Menampilkan form untuk tambah status alat	Berhasil
		Tombol ubah <i>password</i>	Menampilkan form untuk mengubah <i>password</i>	Berhasil

Tabel 5.8 Pengujian Black Box

No	Halaman yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
		Tombol history <i>feeder</i>	Menampilkan riwayat aktivitas <i>feeder</i>	Berhasil
		Tombol keluar	Sistem akan mengeluarkan pengguna dari tampilan <i>mobile application</i> dan kembali ke halaman <i>login</i> .	Berhasil

5.6 Pembahasan

A

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan adapun kesimpulan yang dapat diambil adalah:

1. A
2. A.

6.2. Saran

Berdasarkan hasil yang diperoleh dalam penelitian ini, adapun beberapa saran dari penulis yang dapat dijadikan sebagai bahan penelitian selanjutnya adalah :

1. A
2. A

DAFTAR PUSTAKA

- Barcelo, M. C. (2016). IoT-Cloud Service Optimization in Next Generation Smart Environments. *IEEE Journal on Selected Areas in Communications*, 12, 34.
- Dipo Ahmad Harel, H. P. (2018). Pengembangan Prototipe Sistem Otomasi Alat Pemberi Makan Ikan Terjadwal Pada Aquarium Berbasis Arduino UNO R3. *Widyakala*, 05, 104-110.
- Farndon, J. (2003). *Ensiklopedia Mini Hewan*. Jakarta: Erlangga.
- Hartuti, R. S. (2013). Kajian Kesejahteraan Kucing yang Dipelihara Pada Beberapa Pet Shop Di Wilayah Bekasi, Jawa Barat. *Medika Veterinaria*, 8(1), 37-42.
- Masinambow, N. &. (2014). Pengendali Saklar Listrik Melalui Ponsel Pintar Android. *Jurnal Teknik Elektro dan Komputer*, 03.
- Mulyadi, A. (2010). *Membangun Aplikasi Android*. Yogyakarta: Multimedia Center Publishing.
- Pratama, E. (2014). *Sistem Informasi dan Implementasinya (Teori & Konsep Sistem informasi Disertai berbagai Contoh Praktisnya Menggunakan Perangkat lunak Open Source)*. Bandung: Informatika.
- Regar Devitasari, K. P. (2020). RANCANG BANGUN ALAT PEMBERI PAKAN KUCING OTOMATIS MENGGUNAKAN MIKROKONTROLER NODEMCU BERBASIS INTERNET OF THINGS (IOT). *Jurnal Ilmiah Teknik Informatika*, 14, 152-164.
- Shaifudin, S. d. (2016). Pemantauan Ruang Inkubator Penetas Telur Ayam Dengan Berbasis Telemetri Menggunakan Arduino Uno R3. *Jurnal Teknik Elektro*, 05.
- Susetyo, B. R. (2004). *Panduan Memelihara Kucing Persia*. Jakarta Selatan: Agromedia Pustaka.
- Suwed, M. A. (2011). *Panduan lengkap Kucing*. Jakarta: Penebar Swadaya.
- Teng, K. T.-A. (2018). Strong associations of nine-point body condition scoring with survival and lifespan in cats. *Journal of Feline Medicine and Surgery*, 20 (12), 1110-1118.
- Tubagus Helmi Nashiruddin Sandi, I. A. (2020). RANCANG BANGUN SISTEM PENGISIAN PAKAN DAN MINUM BURUNG OTOMATIS BERBASIS ARDUINO UNO. *Jurnal Teknik Elektro*, 09, 755-805.

AA