



**RANCANG BANGUN PEMBERI PAKAN KUCING OTOMATIS BERBASIS
*INTERNET OF THINGS (IOT)***



Oleh:

Frisilia Lisika

14410200012

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

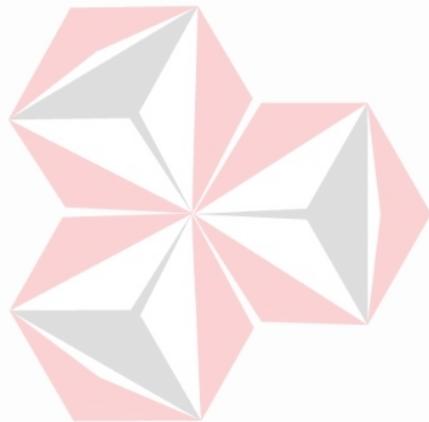
2022

**RANCANG BANGUN PEMBERI PAKAN KUCING OTOMATIS
BERBASIS *INTERNET OF THINGS (IOT)***

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik



UNIVERSITAS
Dinamika

Disusun Oleh:

Nama	: Frisilia Lisika
NIM	: 14410200012
Program Studi	: S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2022

TUGAS AKHIR
RANCANG BANGUN PEMBERI PAKAN KUCING OTOMATIS BERBASIS
INTERNET OF THINGS (IOT)

Dipersiapkan dan disusun oleh

Frisilia Lisika

NIM: 14410200012

Telah diperiksa, diuji, dan disetujui oleh Dewan Penguji

Pada: 11 Januari 2022

Susunan Dewan Penguji

Pembimbing

I. **Harianto, S.Kom., M.Eng.**

NIDN. 0722087701

II. **Ira Puspasari, S.Si., M.T.**

NIDN. 0710078601

Pembahasan

I. **Weny Indah Kusumawati, S.Kom., M.MT.**

NIDN. 0721047201


DN: cn=Harianto, c=ID, o=Universitas
dinamika, ou=Fakultas Teknologi
dan Informatika,
email=har@dinamika.ac.id
Date: 2022.01.28 13:24:31 +07'00'


Digitally signed by Ira Puspasari
DN: cn=Ira Puspasari, o=Universitas
Dinamika, ou=Prodi SI Teknik
Komputer,
email=ira@dinamika.ac.id, c=ID
Date: 2022.01.29 21:33:04 +07'00'


Weny Indah Kusumawati
cn=Weny Indah Kusumawati,
o=Teknologi dan Informatika,
Undika, ou=Teknik Komputer,
email=weny@dinamika.ac.id,
c=ID
2022.01.31 12:41:32 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar sarjana


Digitally signed by
Universitas Dinamika
Date: 2022.02.02 16:38:49
+07'00'

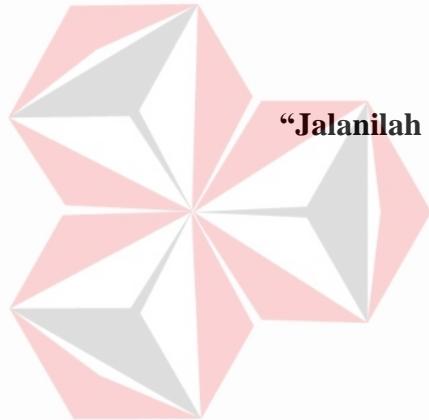
Tri Sagirani, S.Kom., M.MT.

NIDN. 0731017601

Dekan Fakultas Teknologi dan Informatika

FAKULTAS TEKNOLOGI DAN INFORMATIKA

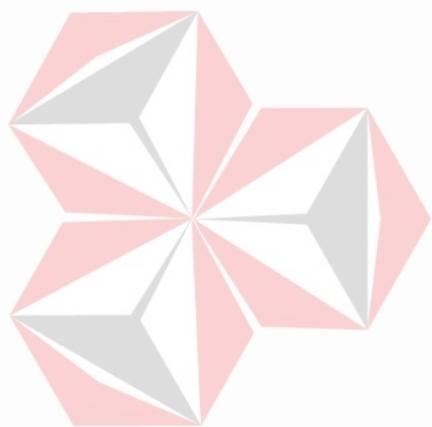
UNIVERSITAS DINAMIKA



“Jalanilah hidup dengan selalu bersyukur, karena tuhan tidak pernah tidur”

UNIVERSITAS
Dinamika

Tugas akhir ini penulis persembahkan terutama kepada kedua orangtua serta adik – adikku yang telah memberi banyak dukungan baik secara moril maupun materil, selain itu penulis juga persembahkan untuk orang – orang yang ikut berjuang bersama dalam menyelesaikan laporan ini.



UNIVERSITAS
Dinamika

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, Saya :

Nama : **Frisilia Lisika**
NIM : **14410200012**
Program Studi : **SI Teknik Komputer**
Fakultas : **Fakultas Teknologi dan Informatika**
Jenis Karya : **Laporan Tugas Akhir**
Judul Karya : **RANCANG BANGUN PEMBERI PAKAN KUCING OTOMATIS BERBASIS INTERNET OF THINGS (IOT)**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Surabaya, 11 Januari 2022



Frisilia Lisika
NIM : 14410200012

ABSTRAK

Salah satu permasalahan pemilik kucing dalam pemeliharaan kucing adalah ketika pergi meninggalkan rumah dalam beberapa hari atau saat bekerja. Alternatif yang dilakukan dalam pemeliharaan harus dititipkan ke penitipan hewan atau ditinggalkan dengan diberi porsi makanan yang banyak. Hal ini tentunya memerlukan biaya yang lebih besar dan tidak baik untuk kesehatan kucing karena pola pakan tidak terkontrol. Pada penelitian sebelumnya sudah dilakukan pembuatan alat untuk pemberi pakan otomatis, namun alat tidak dapat dikontrol dari jarak jauh dan hanya menggunakan EEPROM dalam penyimpanan datanya. Pada alat sebelumnya hasil takarannya diperoleh sebesar 82% tingkat keberhasilannya. Dari permasalahan di atas muncul ide membuat rancang bangun alat pemberi makanan kucing otomatis yang memiliki keakuratan jumlah pakan sesuai *Input* menggunakan *Loadcell* yang takaran keberhasilannnya 88.4%, juga dapat diatur penjadwalan pakan secara langsung (*offline*) melalui display LCD maupun secara jarak jauh melalui aplikasi Android (*online*). Alat pemberi makan kucing ini dapat bekerja secara otomatis dan juga dapat dikontrol melalui aplikasi yang terdapat di HP yang terhubung dengan koneksi internet. Pembuatan sistem dari alat ini menggunakan Android Studio untuk pembuatan program aplikasi dan Arduino Mega 2560 sebagai mikrokontroler yang diintegrasikan dengan ESP-01.

Kata kunci: *Firebase, Internet of Things, Kucing, Loadcell*

UNIVERSITAS
Dinamika

KATA PENGANTAR

Segala puji bagi Tuhan, karena atas berkat yang diberikan penulis berhasil menyelesaikan dan menyusun naskah Tugas Akhir yang berjudul Rancang bangun Pemberi Pakan Kucing Otomatis Berbasis *Internet of Things* (IoT). Tugas Akhir ini merupakan salah satu syarat yang harus ditempuh oleh setiap mahasiswa untuk meraih gelar kesarjanaan di Jurusan Teknik Komputer, Fakultas Teknologi dan Informatika, Universitas Dinamika.

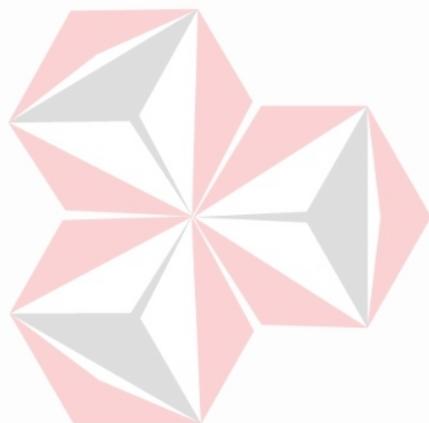
Dalam mengerjakan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Budi Jatmiko, M.Pd., selaku Rektor Universitas Dinamika.
2. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
3. Bapak Harianto, S.Kom., M.Eng., selaku dosen pembimbing pertama yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
4. Ibu Ira Puspasari, S.Si., M.T., selaku dosen pembimbing kedua yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
5. Ibu Weny Indah Kusumawati, S.Kom., M.T., selaku dosen pembahas yang telah memberikan arahan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
6. Seluruh dosen pengajar Program Studi S1 Teknik Komputer Universitas Dinamika yang telah mendidik dan memberikan motivasi kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan baik.
7. Rekan-rekan mahasiswa S1 Teknik Komputer dan semua pihak yang telah memberikan dorongan dan bantuan baik secara langsung maupun secara tidak langsung.
8. Teman-teman yang selalu memberikan dorongan baik itu dari dalam maupun dari luar kampus, terimakasih telah menemani baik dalam suka maupun duka.

Akhir kata, penulis berharap semoga segala sesuatu yang telah dihasilkan dalam pelaksanaan Tugas Akhir ini dapat bermanfaat.

Surabaya, 11 Januari 2022

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah	2
1.4 Tujuan	2
BAB II LANDASAN TEORI.....	3
2.1 Kucing.....	3
2.2 Makanan Kucing.....	4
2.3 Arduino Mega 2560	5
2.4 ESP 01.....	6
2.5 RTC DS3231.....	7
2.6 Sensor <i>Loadcell</i>	7
2.7 <i>Radio Frequency Idenfication (RFID)</i> Reader MFRC 522	8
2.8 <i>Keypad</i>	8
2.1. LCD (<i>Liquid Crystal Display</i>)	9
2.9 Motor Servo	10
2.10 Database <i>Firebase</i>	10
2.11 Android Studio.....	11

2.12	Arduino IDE	11
BAB III	METODOLOGI PENELITIAN	12
3.1	Metode Penelitian	12
3.2	Rancangan <i>Hardware</i>	13
3.3	Rancangan Program	15
3.4	Desain Alat	21
BAB IV	HASIL DAN PEMBAHASAN	22
4.1	Pengujian Pengiriman data antara ESP dengan <i>Firebase</i>	22
4.1.1	Tujuan Pengujian Pengiriman data antara ESP dengan <i>Firebase</i>	22
4.1.2	Alat Yang Digunakan Untuk Pengujian Pengiriman data antara ESP dengan <i>Firebase</i>	23
4.1.3	Prosedur Pengujian Pengiriman data antara ESP dengan <i>Firebase</i>	23
4.1.4	Hasil Pengujian.....	27
4.2	Pengujian <i>Loadcell</i> Terhadap <i>Input</i> yang Diberikan Oleh <i>User</i>	27
4.2.1	Tujuan Pengujian <i>Loadcell</i> Terhadap <i>Input</i> yang Diberikan Oleh <i>User</i>	27
4.2.2	Alat Yang Digunakan Pengujian <i>Loadcell</i> Terhadap <i>Input</i> yang Diberikan Oleh <i>User</i>	28
4.2.3	Prosedur Pengujian <i>Loadcell</i> Terhadap <i>Input</i> yang Diberikan Oleh <i>User</i>	28
4.2.4	Hasil Pengujian <i>Loadcell</i> Terhadap <i>Input</i> yang Diberikan Oleh <i>User</i>	30
4.3	Pengujian Keseluruhan	31
4.3.1	Tujuan Pengujian Keseluruhan	31
4.3.2	Alat Yang Digunakan Pengujian Keseluruhan.....	31

4.3.3 Prosedur Pengujian Keseluruhan.....	32
4.3.4 Hasil Pengujian Keseluruhan	38
BAB V PENUTUP.....	40
5.1 Kesimpulan	40
5.2 Saran	40
DAFTAR PUSTAKA	41
BIODATA PENULIS	82

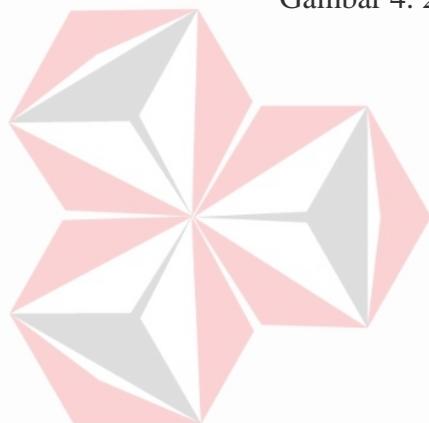


DAFTAR GAMBAR

Halaman

Gambar 2. 1 Aturan porsi kucing	4
Gambar 2. 2 Bentuk fisik Arduino Mega 2560.....	5
Gambar 2. 3 Bentuk fisik dan pin ESP8266-01	6
Gambar 2. 4 Bentuk fisik RTC DS3231	7
Gambar 2. 5 Bentuk fisik <i>Loadcell</i>	7
Gambar 2.6 Bentuk fisik modul HX 711	8
Gambar 2. 7 Bentuk fisik MFRC522 dan RFID Tag.....	8
Gambar 2.8 Bentuk fisik Keypad.....	9
Gambar 2. 9 Bentuk fisik LCD 16x2	9
Gambar 2. 10 Bentuk fisik motor servo	10
Gambar 2. 11 Struktur Json tree database Firebase	11
Gambar 2. 12 Tampilan user interface Arduino IDE.....	11
Gambar 3. 1 Blok diagram alur pengerjaan	12
Gambar 3. 2 Blok diagram sistem.....	13
Gambar 3. 3 Blok diagram dari hardware	14
Gambar 3. 4 Flowchart Arduino Mega 2560 (1)	15
Gambar 3. 5 Flowchart Arduino Mega 2560 (2)	16
Gambar 3. 6 Flowchart aplikasi Android	19
Gambar 3. 7 Struktur Json Tree database Firebase	20
Gambar 3. 8 Desain alat	21
Gambar 4. 1 Rangkaian ESP-01 ke USB-TTL	23
Gambar 4. 2 Auth key Firebase untuk ESP	24
Gambar 4. 3 Host Firebase untuk ESP	24
Gambar 4.4 Pengujian create data ke Firebase	25
Gambar 4. 5 Pengujian update data ke Firebase	26
Gambar 4. 6 Data pada Firebase yang dibaca oleh ESP01	26
Gambar 4. 7 Data pada Firebase setelah dihapus.....	27
Gambar 4. 8 Rangkaian pengujian Loadcell	28
Gambar 4. 9 Input porsi untuk pengujian Loadcell.....	29
Gambar 4. 10 Pakan keluar	29

Gambar 4. 11 Hasil timbang pakan yang keluar	30
Gambar 4. 12 Rangkaian Schematic untuk pengujian sistem secara keseluruhan.	32
Gambar 4. 13 Halaman waiting for tapping.....	33
Gambar 4. 14 Display LCD untuk tap create Tag baru.....	33
Gambar 4. 15 RFID Tag telah terdaftar di EEPROM.....	34
Gambar 4. 16 Display LCD jika RFID Tag belum terdaftar.....	34
Gambar 4.177 Halaman Input nama dan umur kucing	34
Gambar 4. 18 Halaman Input porsi, timer, dan first feed	35
Gambar 4. 19 Halaman user action	36
Gambar 4. 20 Tampilan LCD ketika delete	36
Gambar 4. 21 Tampilan LCD ketika update	36
Gambar 4. 22 RFID Tag Ter-tapping.....	37
Gambar 4. 23 Servo terbuka dan tertutup sesuai penjadwalan	38



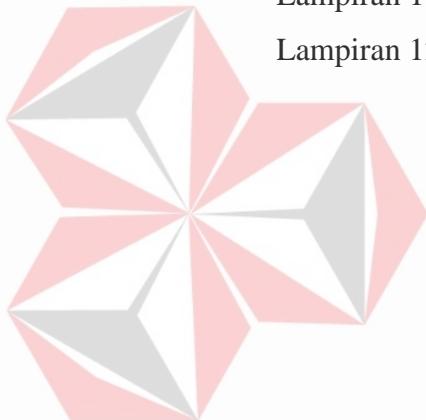
DAFTAR TABEL

	Halaman
Tabel 2. 1 Spesifikasi Arduino Mega 2560.....	6
Tabel 4.1 Tabel pengujian Loadcell.....	31
Tabel 4.2 Pengujian takaran porsi makanan	39
Tabel 4.3 Pengujian penjadwalan pemberian makanan	39



DAFTAR LAMPIRAN

	Halaman
Lampiran 1. PROGRAM ARDUINO MEGA	42
Lampiran 2. PROGRAM ESP-01	52
Lampiran 3. PROGRAM first_page	56
Lampiran 4. PROGRAM second_page.....	58
Lampiran 5. PROGRAM no_delete_confirmation	61
Lampiran 6. PROGRAM no_select_cat.....	64
Lampiran 7. PROGRAM no_user_action	66
Lampiran 8. PROGRAM yes_adjust_portion	70
Lampiran 9. PROGRAM yes_done	74
Lampiran 10. PROGRAM yes_register_name	76
Lampiran 11. PROGRAM yes_tapping	79
Lampiran 12. Hasil Turnitin.....	81



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Disaat bekerja atau pergi meninggalkan rumah timbul masalah jika memiliki hewan peliharaan. Alternatif yang dilakukan dalam pemeliharaan harus dititipkan ke penitipan hewan atau ditinggalkan dengan diberi porsi makanan yang banyak. Hal ini tentunya memerlukan biaya yang lebih besar dan tidak baik untuk kesehatan kucing karena pola pakan tidak terkontrol. Selain obesitas, pola pakan yang tidak terkontrol juga dapat mempengaruhi fisiologi pencernaan kucing, sehingga memicu terjadinya berbagai resiko penyakit pada kucing (Nusdianto, 2016).

Pada penelitian sebelumnya sudah dilakukan pembuatan alat untuk pemberi pakan otomatis, namun alat tidak dapat dikontrol dari jarak jauh dan hanya menggunakan EEPROM dalam penyimpanan datanya (Beth, Harianto, & Ira, 2019). Pada alat sebelumnya hasil takarannya diperoleh sebesar 82% tingkat keberhasilannya. Berdasarkan permasalahan yang ada, maka dilakukan penelitian tentang bagaimana cara membuat rancang bangun alat pemberi makanan kucing otomatis yang memiliki keakuratan jumlah pakan sesuai *Input* menggunakan *Loadcell*, juga dapat diatur penjadwalan pakan secara langsung (*offline*) melalui display LCD maupun secara jarak jauh melalui aplikasi Android (*online*).

Fungsi alat ini untuk memudahkan pemilik kucing yang kesulitan memberi makan kucingnya saat tidak di rumah. Alat makan kucing ini dapat bekerja secara otomatis dan juga dapat dikontrol dengan aplikasi pada *handphone* yang terkoneksi dengan internet. Pembuatan sistem dari alat ini menggunakan Android Studio untuk pembuatan program aplikasi dan Arduino Mega 2560 sebagai mikrokontroler yang diintegrasikan dengan ESP 01.

1.2 Rumusan Masalah

Dari latar belakang yang dijelasakan, dapat simpulkan permasalahan dalam penggeraan Tugas Akhir ini adalah:

1. Bagaimana cara penyempurnaan agar jumlah berat pakan yang diberikan pada kucing sesuai dengan *Input* menggunakan *Loadcell*.
2. Bagaimana cara melakukan pengaturan penjadwalan pakan dan pemberian pakan secara jarak jauh (*online*) melalui aplikasi Android untuk mengimplementasikan *Internet Of Things* (IoT) maupun secara langsung (*offline*) melalui display LCD.

1.3 Batasan Masalah

Adapun yang menjadi batasan masalah dari tugas akhir ini agar pembahasan tidak meluas adalah sebagai berikut:

1. Ukuran makanan kucing yang diberikan berukuran tidak lebih dari 5mm.
2. Makanan kucing yang diberikan adalah makanan kering (berbentuk seperti biskuit).
3. Maksimal 3 buah identitas (*Tag RFID*).

1.4 Tujuan

Berdasarkan rumusan masalah di atas, adapun tujuan dari tugas akhir atau penelitian yang ingin dicapai yakni:

1. Dapat menyempurnakan jumlah pakan yang diberikan pada kucing sesuai dengan *Input* menggunakan *Loadcell*.
2. Dapat melakukan pengaturan penjadwalan pakan dan pemberian pakan secara jarak jauh (*online*) melalui aplikasi Android untuk mengimplementasikan *Internet Of Things* (IoT) maupun secara langsung (*offline*) melalui display LCD

BAB II

LANDASAN TEORI

2.1 Kucing

Hewan yang cukup populer dipelihara, dan sangat bersahabat dengan manusia. Kucing telah berbaur dan menjadi teman manusia semenjak tahun 4000SM. Dipulau Siprus tahun 7500SM ditemukan kerangka seekor kucing (Suwed & Napitupulu, 2011). Banyak jenis ras kucing, yang tersebar diseluruh dunia. Setiap ras memiliki ciri khasnya masing-masing, namun karena seringnya terjadi perkawinan silang antar ras, banyak kucing yang dibagi menjadi dua jenis yaitu yang berbulu panjang dan yang berbulu pendek.

1. Persia

Berasal dari Iran yang dulunya dikenal dengan nama Persia. Di Indonesia kucing jenis ini paling banyak dipelihara. Memiliki ciri khas berkepala bulat, berbulu panjang, dan juga memiliki hidung pesek. Selain memiliki sifat yang kalem, kucing juga dikenal dengan sifa yang tenang. Kucing dengan jenis Persia ini, juga mudah akrab dengan manusia. Untuk menjaga dan merawat kesehatan kulit serta rambutnya, kucing Persia membutuhkan makanan dengan kandungan omega 3 (EPA dan DHA) juga omega 6.

2. Anggora

Dengan postur tubuh yang atletis serta bulu yang Panjang, kucing dengan nama lengkap Turkish Angora, sesuai namanya kucing ini berasal dari Turki. Kucing ini juga dekat dengan manusia.

3. Siam

Berasal dari negeri gajah putih Thailand, jenis ini juga mempunyai tubuh yang lebih ramping. Dengan ciri khas bulu yang lebih pendek. Selain itu sinar matanya, merupakan perpaduan antara warna almond, dengan warna biru.

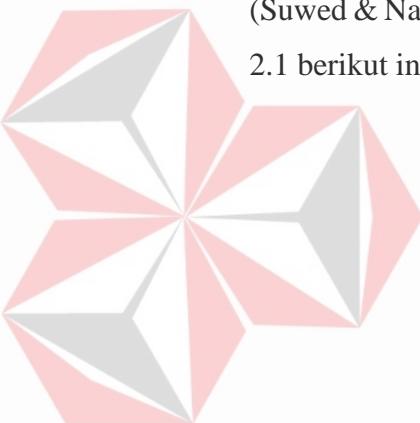
4. Maine Coon

Dengan postur badan yang mencapai 7 sampai dengan 11 kilogram untuk maine coon jantan dewasa, dan 5 sampai 7 kilogram untuk betina. Kucing dengan jenis

bulu yang cukup panjang ini berasal dari daratan Amerika Utara. Selain EPA dan DHA, kucing jenis ini membutuhkan taurine untuk menjaga kesehatan jantungnya.

Selain ditimbang, dengan membelai punggung kucing, dapat merasakan kondisi kelebihan berat badan kucing. Jika dilihat dengan teliti, batas pinggangnya terlihat jelas, jika dilihat dari atas. Tulang punggung dan rusuknya mudah teraba namun tidak terlihat, selain itu perutnya juga melorot.

Obesitas pada hewan, terutama pada kucing berdampak buruk pada kesehatannya. Banyak faktor, yang mempengaruhi obesitas pada kucing. Faktor bawaan seperti usia dan jenis kelamin, faktor genetik seperti keturunan juga ikut berpengaruh. Namun pada kenyataannya pemberian pakan berlebih dan tidak terkontrol juga mempengaruhi kucing obesitas, pemberian takaran kucing yang salah misalnya. (Suwed & Napitupulu, 2011). Acuan untuk pemberian porsi kucing sesuai pada gambar 2.1 berikut ini.



**100% Complete & Balanced for Maintenance
Complète et équilibrée à 100 % pour le
maintien de la santé**

**c = cups
g = grammes
1 cup = 88 grams**

**FEEDING GUIDELINES
GUIDE ALIMENTAIRE**

Cat Weight (lbs.) Poids du chat (kg)	Daily Feeding Guide Maintenance		Ration quotidienne Maintien	
	Weight Loss	Maintain	Perte de poids	Maintien
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. 1 Aturan porsi kucing
(Sumber: Suwed & Napitupulu, 2011)

2.2 Makanan Kucing

Kucing memiliki pola makan dan perilaku yang sangat spesifik. Kucing makan sekitar 3-4 kali sehari dengan total sekitar 30 menit sehari. Kucing hanya makan beberapa gram makanan per makanan. Rasa sangat penting dan merupakan mekanisme

dimana hewan memilih makanan untuk memenuhi kebutuhannya. Studi oleh Hewson-Hughes dkk, memberikan gambaran yang sangat jelas tentang kebutuhan nutrisi kucing, terutama protein, lemak dan karbohidrat. Seekor kucing membutuhkan sekitar 26 gram protein, 9 gram lemak, dan 8 gram karbohidrat sehari, yang setara dengan 52% protein, 36% lemak, dan 12% karbohidrat untuk kebutuhan kalori. Beberapa penelitian juga mengatakan bahwa kucing yang tinggal di rumah hanya mengkonsumsi sedikit energi. Seekor kucing dengan berat \pm 3.5 Kg hanya membutuhkan asupan normal sekitar 50 g makanan kering per hari. (Suwed & Napitupulu, 2011)

2.3 Arduino Mega 2560



Gambar 2. 2 Bentuk fisik Arduino Mega 2560
(Sumber: Andrianto, 2010)

Penggunaan yang cukup sederhana, memungkinkan Arduino mega dapat berjalan dengan power dari usb PC/laptop, serta jack adaptor dengan tegangan 7 sampai dengan 12 VDC. Selain itu Arduino mega dilengkapi dengan board yang memiliki pin I/O yang relatif banyak. Dengan digital I/O sebanyak 54 buah, dan 15 diantaranya dapat dimanfaatkan sebagai *output* PWM. Arduino mega mikrokontroler dilengkapi dengan IC ATmega 2560 sebagai IC Mikronya (Andrianto, Pemrograman Mikrokontroller AVR ATMEGA16, 2010) Untuk spesifikasi lebih jelasnya dapat di lihat pada Tabel 2.1 dibawah ini:

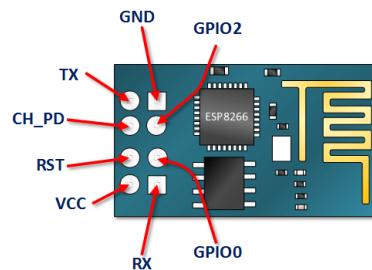
Tabel 2. 1 Spesifikasi Arduino Mega 2560

Mikrokontroler	ATmega2560
Tegangan Operasional	5V
Tegangan <i>Input</i> (rekomendasi)	7-12V
Tegangan <i>Input</i> (limit)	6-20V
PIN Digital I/O	54 (of which 15 provide PWM output)
PIN Analog <i>Input</i>	16
Arus DC per Pin I/O	20 Ma
Arus DC untuk Pin 3.3 V	50 Ma
Memori Flash	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	101.52 mm
Lebar	53.3 mm
Berat	37 g

Memiliki resolusi sebesar 10 bit, pada tiap PIN. Selain 54 pin digital, Arduino mega memiliki 16 PIN analog sebagai ADC.

2.4 ESP 01

Modul ini merupakan salah satu varian dari modul ESP8266. Walaupun modul ini dapat berdiri sendiri, modul ini hanya memiliki 2 pin I/O saja. Modul ini dapat terhubung ke jaringan wifi, karena sudah terintegrasi dengan protocol TCP/IP (Eka, 2017). Untuk penggunaannya, agar lebih jelas pinnya dapat dilihat pada gambar 2.3. di bawah ini.



Gambar 2. 3 Bentuk fisik dan pin ESP8266-01
(Sumber: Eka, 2017)

2.5 RTC DS3231

Modul yang digunakan untuk waktu digital. RTC atau yang biasa dikenal dengan *real time clock*, dilengkapi dengan IC EEPROM dengan tipe AT24C32. Untuk komunikasi dengan RTC, dapat menggunakan media *interfaces* I2C (SCL dan SDA). Sebagai modul waktu, RTC memiliki presisi yang cukup akurat. RTC dapat menghitung waktu detik, menit, jam, hari, sampai dengan tahun 2100. RTC dapat bekerja dengan tegangan yang rendah, sehingga fleksibel. Berjalan ditegangan 3.3VDC sampai dengan 5.5V DC, RTC dapat menggunakan baterai maupun dihubungkan langsung ke *output* power Arduino sebagai sumber tegangannya (Hapsari, 2012). Gambar 2.4 merupakan bentuk fisik dari RTC.

2.6 Sensor Loadcell

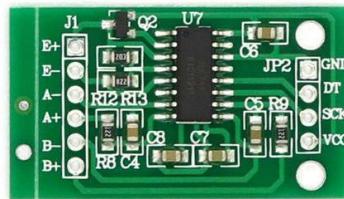


Gambar 2. 4 Bentuk fisik RTC DS3231
(Sumber: Hapsari, 2012)

Gambar 2. 5 Bentuk fisik Loadcell
(Sumber: Owen, 2004)

Loadcell seperti pada gambar 2.5 adalah sebuah alat uji elektronika yang digunakan untuk mengukur berat dari suatu benda. *Output* sinyal listrik yang dihasilkan biasanya kecil dalam satuan beberapa *millivolt* dan membutuhkan *amplifier* sebelum dapat digunakan (Owen, 2004). Disini penulis menggunakan modul HX 711 sebagai

amplifier dari masukkan *Loadcell*. HX 711 pada Gambar 2.6 adalah sebuah modul yang berfungsi sebagai penguat (*amplifier*) untuk sensor *Loadcell* dan ADC.



Gambar 2.6 Bentuk fisik modul HX 711

(Sumber: Owen, 2004)

2.7 *Radio Frequency Idenfication (RFID) Reader MFRC 522*

RFID Reader MFRC 522 pada Gambar 2.7 adalah modul yang berfungsi untuk membaca ID dari RFID, baik itu dari RFID Tag, RFID Sticker, RFID Card, dan lainnya. Bekerja dengan menggunakan *interface SPI*, catu daya 3.3 V, dan frekuensi 13.56Mhz dalam pengoperasiannya (Beth, Harianto, & Ira, 2019). Memiliki dimensi sebesar 40x50mm dengan jarak pembacaan ~60mm (Mahali, 2016).



Gambar 2. 7 Bentuk fisik MFRC522 dan RFID Tag

(Sumber: Mahali, 2016)

2.8 *Keypad*

Digunakan sebagai media *Input*, *keypad* terdiri dari saklar yang saling terhubung. Tersusun rapi membentuk baris dan kolom. Untuk memindai *keypad*, mikrokontroler dapat melakukan proses pengecekan pada port. Dengan mengirim logika *low* salah satu dari 4 bit pada kolom-kolom tersebut jika tombol ditekan, dan mikrokontroler menerima sinyal logika *high* jika tombol pada *keypad* tidak ditekan (Owen, 2004).



Gambar 2.8 Bentuk fisik *Keypad*
(Sumber: Owen, 2004)

2.1. LCD (*Liquid Crystal Display*)



Gambar 2. 9 Bentuk fisik LCD 16x2
(Sumber: Nugroho, 2008)

LCD menggunakan memori dan register dalam pengoperasiannya. Memori yang digunakan adalah DDRAM, CGRAM, dan CGROM, dan untuk *register control* yang terdapat pada LCD adalah *register perintah* dan *register data*. Memiliki 5 pin, yakni pin data, RS (*register select*), R/W (*Read/Write*), E (*Enable*), dan VLCD (Nugroho, 2008). Bentuk Fisik LCD tampak seperti pada gambar 2.9.

2.9 Motor Servo



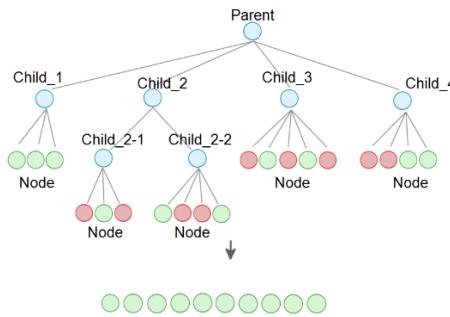
Gambar 2. 10 Bentuk fisik motor *servo*

(Sumber: Owen, 2004)

Motor Servo adalah sebuah perangkat *actuator* yang biasa digunakan untuk menggerakkan sesuatu. Terdiri dari serangkaian *gear*, motor DC, rangkaian kontrol, dan potensiometer. Motor servo bekerja dikendalikan dengan diberi sinyal PWM (*Pulse Width Modulation*) melalui kabel kontrol. Lebar pulsa yang diberikan mempengaruhi posisi sudut putaran dari poros motor servo. Ketika lebar pulsa diberikan, poros motor servo bergerak dan berputar ke posisi yang telah ditentukan (Owen, 2004). Gambar 2.10 di atas adalah contoh dari motor servo yang digunakan pada alat Tugas Akhir yang penulis kerjakan.

2.10 Database *Firebase*

Firebase merupakan *realtime* database yang berjalan pada *cloud* database. Disimpan dengan format JSON, dan juga memungkinkan komunikasi dan sinkronisasi secara *realtime*, ke seiap klien. *Firebase* dapat digunakan untuk menyimpan data audio dan visual (Hapsari, 2012). *Firebase* adalah database yang menggunakan struktur JSON *tree* yang secara garis besar contohnya dapat dilihat pada Gambar 2.11 di bawah ini.



Gambar 2. 11 Struktur *Json tree database Firebase*
(Sumber: Hapsari, 2012)

2.11 Android Studio

Android Studio adalah lingkungan pengembangan terintegrasi (IDE), untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain menjadi editor kode dan alat pengembang IntelliJ yang andal, Android Studio juga menyediakan lebih banyak fitur. (Hapsari, 2012)

2.12 Arduino IDE

Berfungsi untuk membuat dan melakukan *check* kode program yang diload. Tampilan *user interface* dapat dilihat pada gambar 2.12.



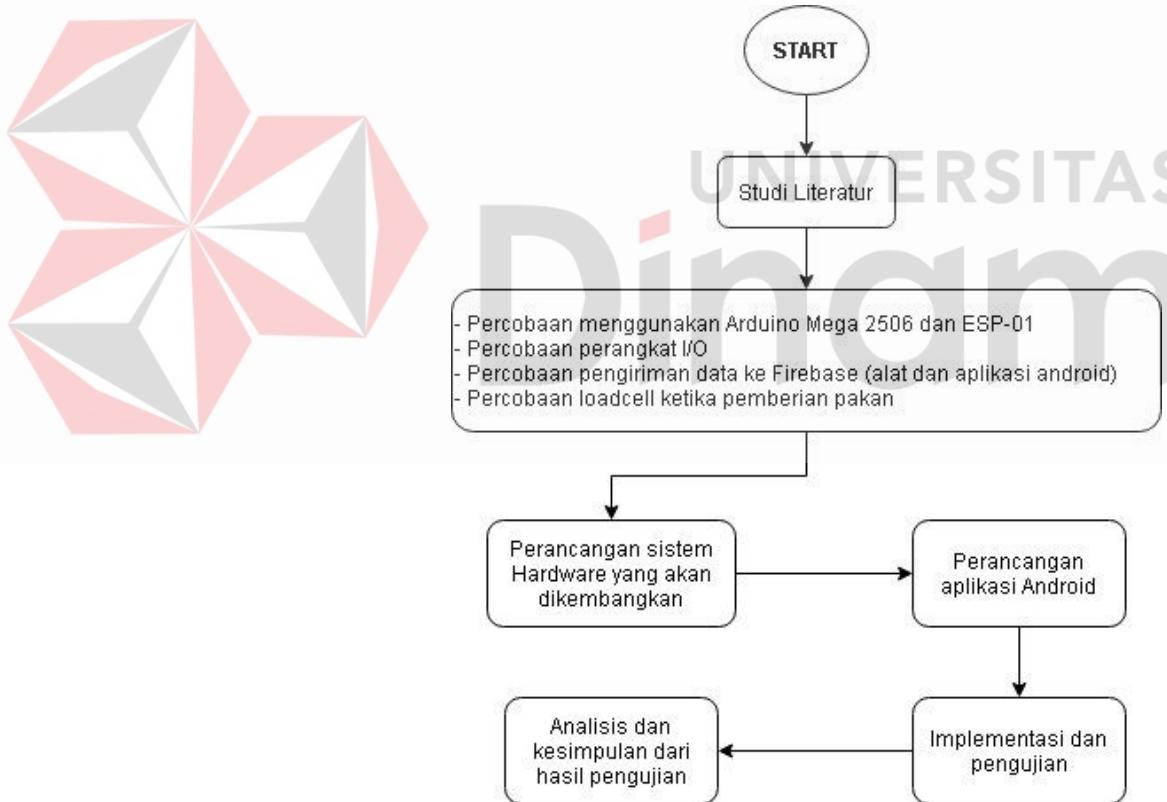
Gambar 2. 12 Tampilan *user interface* Arduino IDE
(Sumber: Sulaiman, 2017)

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

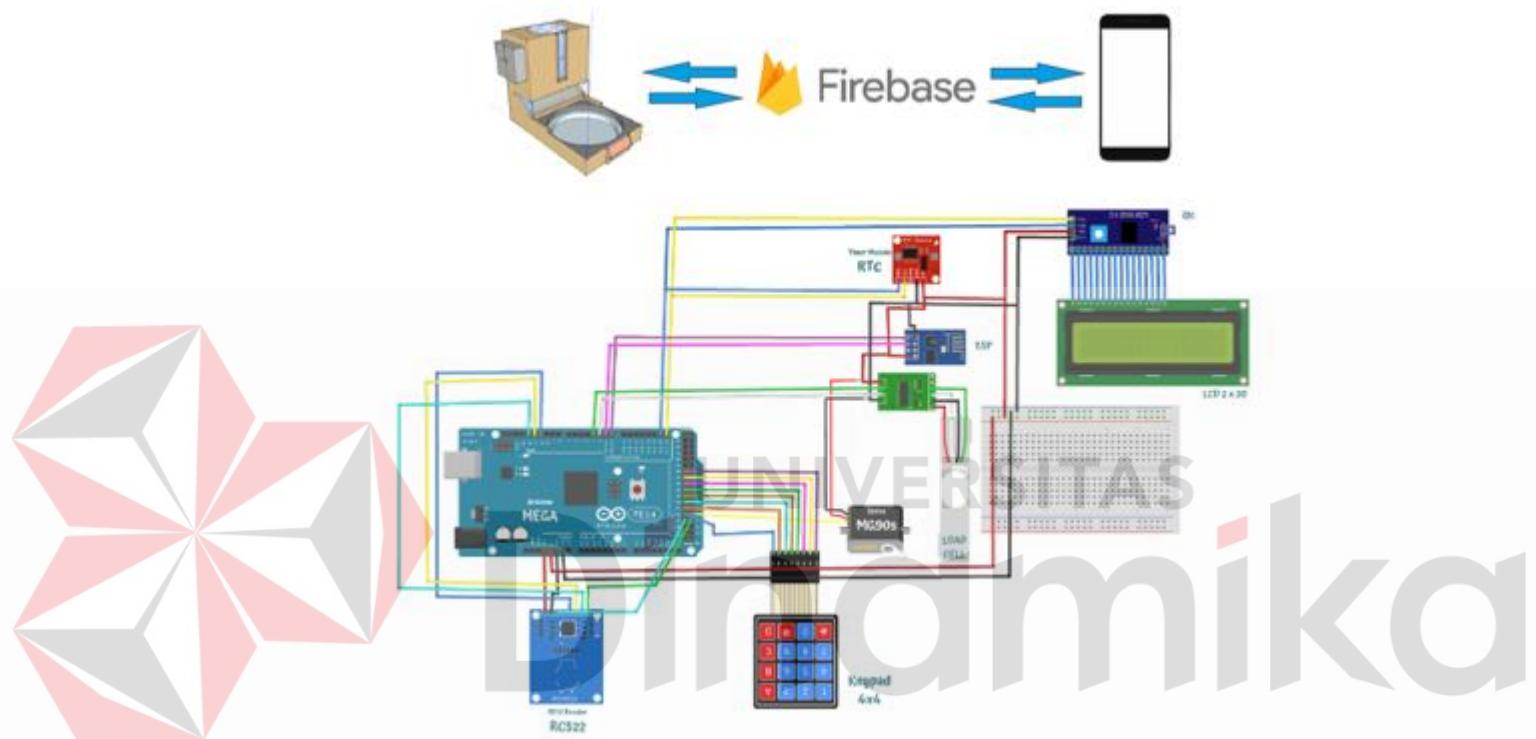
Pada penelitian ini, digunakan metode penelitian dengan studi literatur sampai dengan perancangan perangka dan implementasi serta pengujian. Sebelum mengerjakan penulis lebih dahulu mengumpulkan referensi berupa data informasi, dan relevan. Setelah studi literasi kemudian menuju tahap perancangan. Kemudian dilakukan integrasi perangkat keras dan perangkat lunak, setelah itu dilakukan pengujian dari seluruh racangan. Blok diagram alur penggerjaan dituliskan pada gambar 3.1.



Gambar 3. 1 Blok diagram alur penggerjaan

3.2 Rancangan *Hardware*

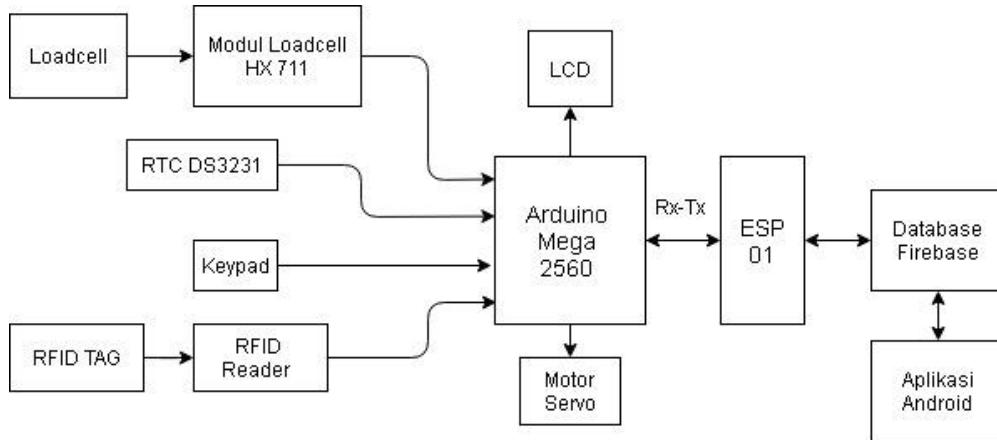
Internet of Things (IoT) adalah konsep sistem yang digunakan, untuk memberikan pakan otomatis pada kucing peliharaan di rumah. Konsep sederhana sistemnya, adalah sebagai berikut:



Gambar 3. 2 Blok diagram sistem

Dari gambar 3.2 di atas, alat dan HP android (aplikasi) terhubung dan mengakses data pada database *Firebase*. *Input* dari sistem ini ada dua, melalui perangkat langsung dan melalui apk Android. Data yang tersimpan pada *Firebase* dijadikan acuan untuk alat dalam pemberian pakan pada kucing. Untuk *flowchart* pada program lebih jelasnya dapat dilihat pada gambar 3.3 di bawah. Database *Firebase* digunakan sebagai penyimpanan data selain menggunakan EEPROM, agar alat dapat diakses dari jarak jauh baik secara *online* maupun *offline*.

Adapun blok diagram dari *Hardware* yang digunakan adalah sebagai berikut:



Gambar 3. 3 Blok diagram dari *hardware*

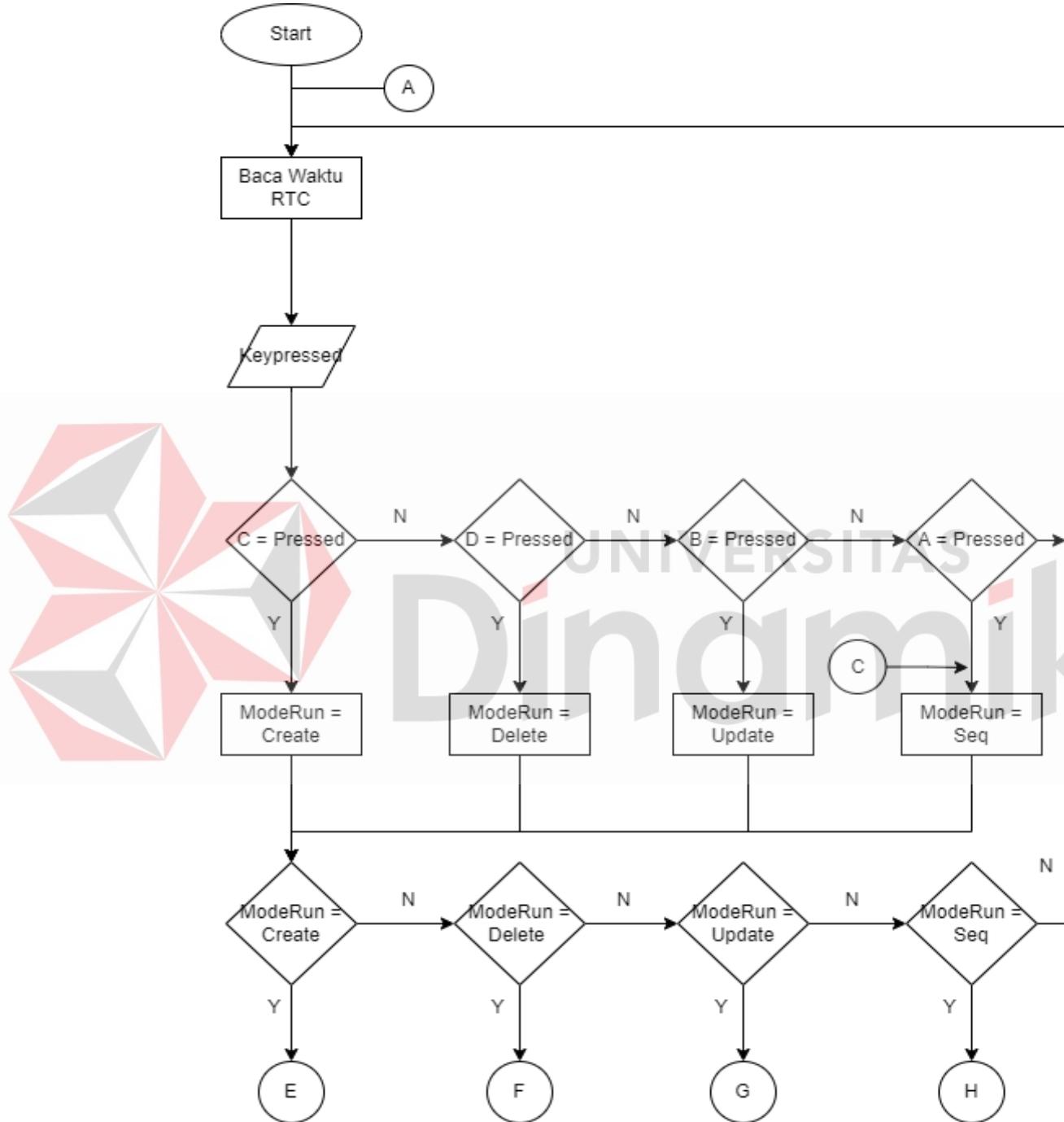
Arduino Mega 2560 dan ESP-01 berperan sebagai otak dari sistem yang dibuat. Penulis disini menggunakan Arduino Mega 2560 karena menggunakan port I/O dengan jumlah yang banyak, dan ESP-01 digunakan sebagai jembatan komunikasi antara alat dengan database *Firebase*. HX 177 berperan sebagai penguat dari hasil pembacaan sensor *Loadcell*. Sensor *Loadcell* digunakan untuk membaca berapa banyaknya pakan yang keluar supaya lebih akurat, agar tidak berlebihan dan tidak kurang.

Keypad digunakan ketika ingin melakukan pengaturan secara langsung (*offline*), *user* memberikan *Input* data seperti berapa banyak pakan yang diberikan dan berapa jeda waktu setiap kali memberi pakan. Data *Input* tadi disimpan pada EEPROM dan selanjutnya data dikirimkan juga ke *Firebase*. LCD digunakan sebagai tampilan untuk komunikasi antara alat dengan manusia. Gambar 3.3 merupakan blok diagram dari *Hardware*. Aplikasi Android digunakan ketika ingin melakukan pengaturan secara jarak jauh (*online*). Untuk detail selengkapnya dari aplikasi android dapat dilihat pada subbab 3.3.2 pada penjelasan *flowchartnya*.

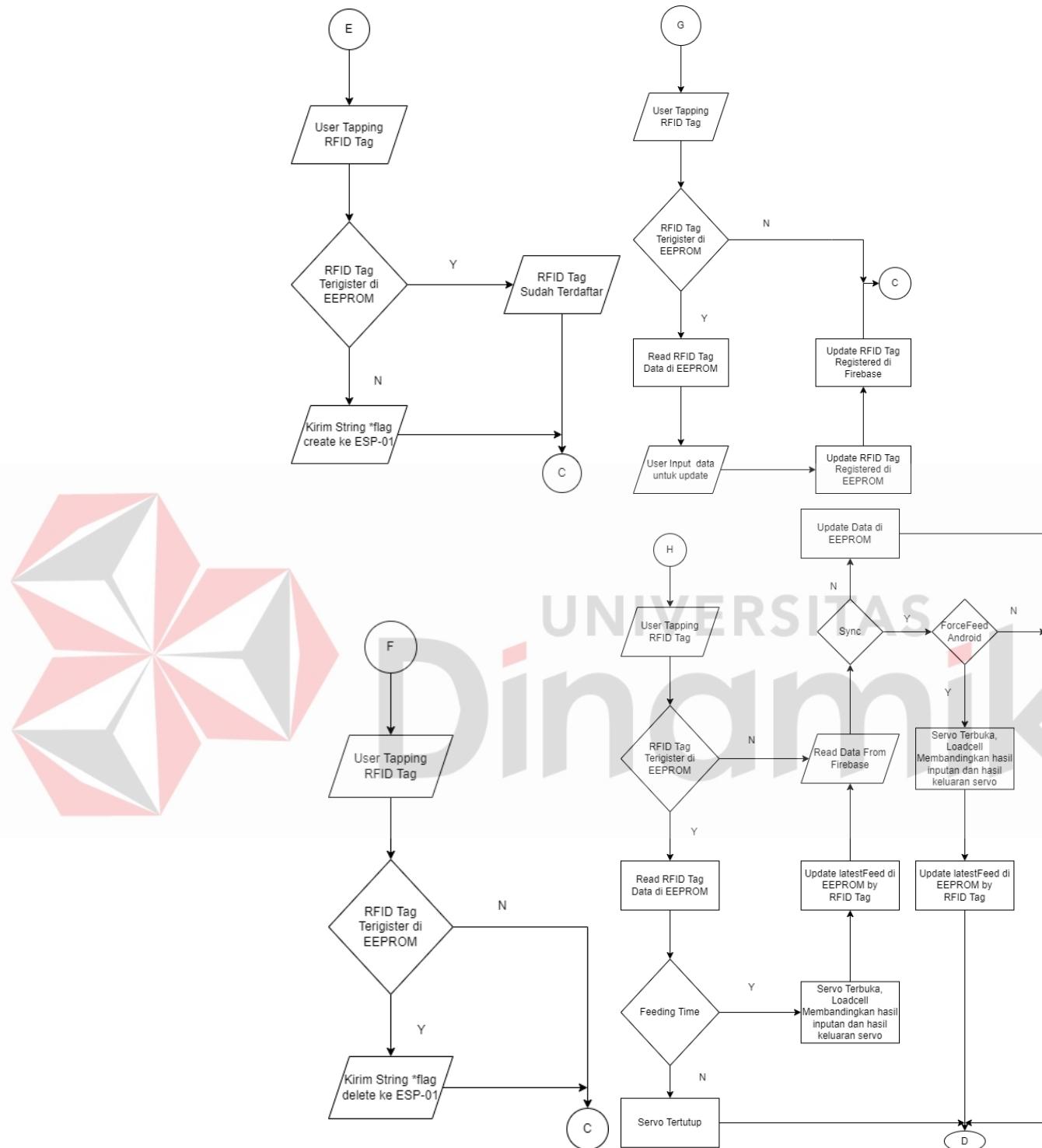
RFID Reader membaca data berupa identitas dari Tag. Pada saat identitas diketahui dan nilai dari RTC sesuai dengan waktu kasih pakan, motor servo berada pada posisi membuka memberikan porsi makanan sesuai dengan data yang diInputkan, ketika nilai *Loadcell* telah sesuai dengan *Input* yang diberikan maka servo kembali ke posisi menutup.

3.3 Rancangan Program

3.3.1 *Flowchart* Arduino Mega 2560



Gambar 3. 4 *Flowchart* Arduino Mega 2560 (1)



Gambar 3. 5 Flowchart Arduino Mega 2560 (2)

Dari gambar 3.4 dan 3.5 di atas, mula-mula Arduino Mega 2560 mengambil data waktu pada RTC, kemudian dilakukan pembacaan pada *keypad*. Jika ditekan C, maka masuk ke mode *create*, jika ditekan D, maka masuk ke mode *delete*, jika ditekan B, maka masuk ke mode *update*, dan jika ditekan A, maka masuk ke mode seq untuk sinkronisasi data antara *Firebase* dan EEPROM.

a. Mode *run create*

Mode ini digunakan ketika *user* ingin mendaftarkan / membuat data untuk RFID Tag baru. *User tapping* RFID Tag terlebih dahulu, kemudian dilakukan *scanning* pada EEPROM apakah data RFID Tag sudah ada. Jika belum ada maka “ID” RFID Tag baru tadi disimpan ke *Firebase*, lalu *user* memberikan *Inputan* melalui aplikasi android yang nantinya data *Inputan* dibaca ketika masuk ke mode seq. Jika RFID Tag sudah terdaftar, maka LCD menampilkan bahwa “RFID Tag sudah terdaftar” dan langsung masuk ke mode Seq.

b. Mode *run update*

Mode ini digunakan ketika *user* ingin meng-*update* data yang tersimpan pada EEPROM dan *Firebase*. *User tapping* RFID Tag terlebih dahulu, kemudian dilakukan *scanning* pada EEPROM apakah data RFID Tag sudah ada. Jika sudah ada, maka nilai-nilai RFID Tag pada EEPROM yang tersimpan dibaca semua. Kemudian LCD menampilkan nilai yang tersimpan pada EEPROM, kemudian *user* memasukkan *Input* data yang ingin di *update*. Setelah selesai memberikan *Inputan*, data disimpan ke EEPROM dan dikirimkan *update* data baru ke *Firebase*. Jika RFID Tag belum terdaftar di EEPROM, maka lcd menampilkan bahwa “RFID Tag belum terdaftar” dan langsung masuk ke mode Seq.

c. Mode *run delete*

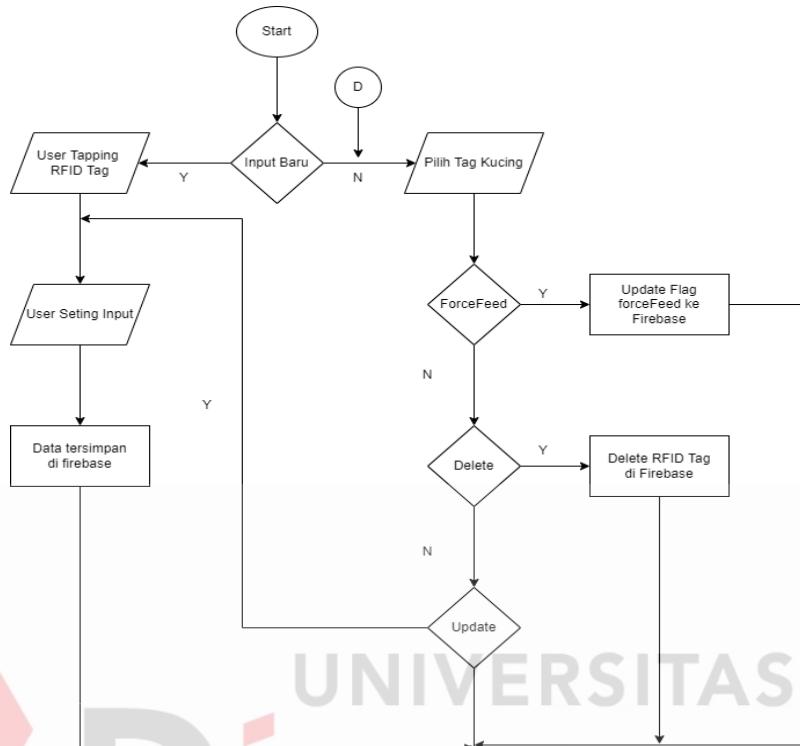
Mode ini digunakan ketika *user* ingin menghapus data RFID Tag yang tersimpan pada EEPROM dan *Firebase*. *User tapping* RFID Tag terlebih dahulu, kemudian dilakukan *scanning* pada EEPROM apakah data RFID Tag sudah ada. Jika RFID Tag sudah tersimpan di EEPROM, maka hapus data *Input* dari *user* yang tersimpan di EEPROM, kemudian mengirim flag ke ESP-01 untuk menghapus data RFID Tag yang tersimpan di *Firebase*. Jika RFID Tag belum terdaftar di EEPROM,

maka LCD menampilkan bahwa “RFID Tag belum terdaftar” dan langsung masuk ke mode Seq.

d. Mode *run seq*

Mode ini adalah mode yang berjalan terus menerus ketika alat menyala. Pada mode ini dilakukan pembacaan terus menerus apakah ada *update* data terbaru dari *Firebase* untuk menyamakan data dengan EEPROM. Mula-mula RFID Tag ter tapping oleh kucing yang mendekat ke alat ketika lapar. Lalu dilakukan pembacaan apakah RFID Tag terdaftar, jika ya maka dicek apakah sudah waktunya makan. Jika ya maka servo terbuka, dan ketika *Loadcell* sudah menyentuh nilai yang telah ditentukan servo tertutup. Kemudian meng-*update value latestFeed* pada EEPROM untuk *flag* kapan terakhir kali kucing diberi pakan. Selama tidak ada tapping dari RFID Tag, alat melakukan pembacaan pada *Firebase* melalui ESP-01. Jika paket dan *flag* yang diterima adalah paket data baru, maka dilakukan *update* data pada EEPROM. Jika *flag* yang diterima adalah *flag* untuk memberikan makan langsung, maka data jumlah pakan yang diberikan diterima, servo terbuka dan menutup ketika jumlah sudah terpenuhi yang dibaca melalui *Loadcell* lalu dilakukan *update flag latestFeed* pada EEPROM.

3.3.2 Aplikasi Android

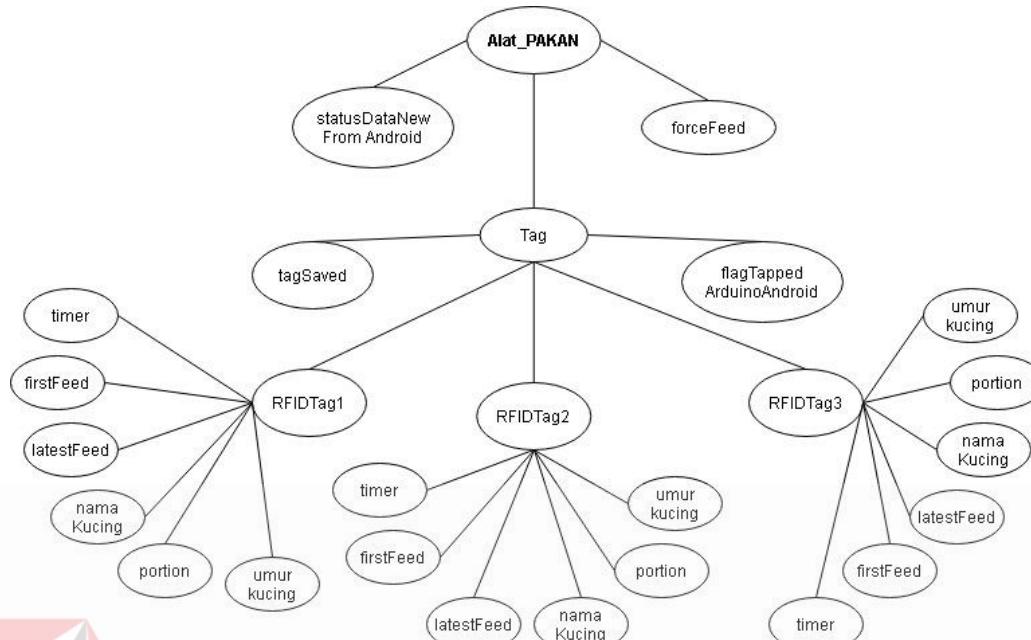


Gambar 3. 6 Flowchart aplikasi Android

Dari gambar 3.6 di atas, awalnya *user* diberi pilihan apakah mau memberikan *setting* baru atau melihat nilai variabel terkini pada alat terlebih dahulu. Jika tidak ingin *setting* baru, maka *user* memilih id *RFID Tag* yang sudah tersimpan. Lalu dilakukan aksi apakah *user* ingin memberikan pakan secara langsung (kirim *flag* ke alat melalui *Firebase*), jika ingin menghapus *RFID Tag* yang sudah tersimpan (hapus *RFID Tag* di *Firebase*), atau ingin mengupdate data yang sudah tersimpan.

Jika *user* ingin memberikan *Input setting* baru, aplikasi pindah ke halaman *register*. *User tapping* ke alat terlebih dahulu untuk didapatkan “id” dari *RFID Tag* nya, kemudian setelah di tapping halaman berpindah ke halaman *Input setting*, *user* memberikan *Input* pada halaman ini. Data yang dimasukkan kemudian disimpan ke *Firebase*.

3.3.3 Rancangan Struktur Database

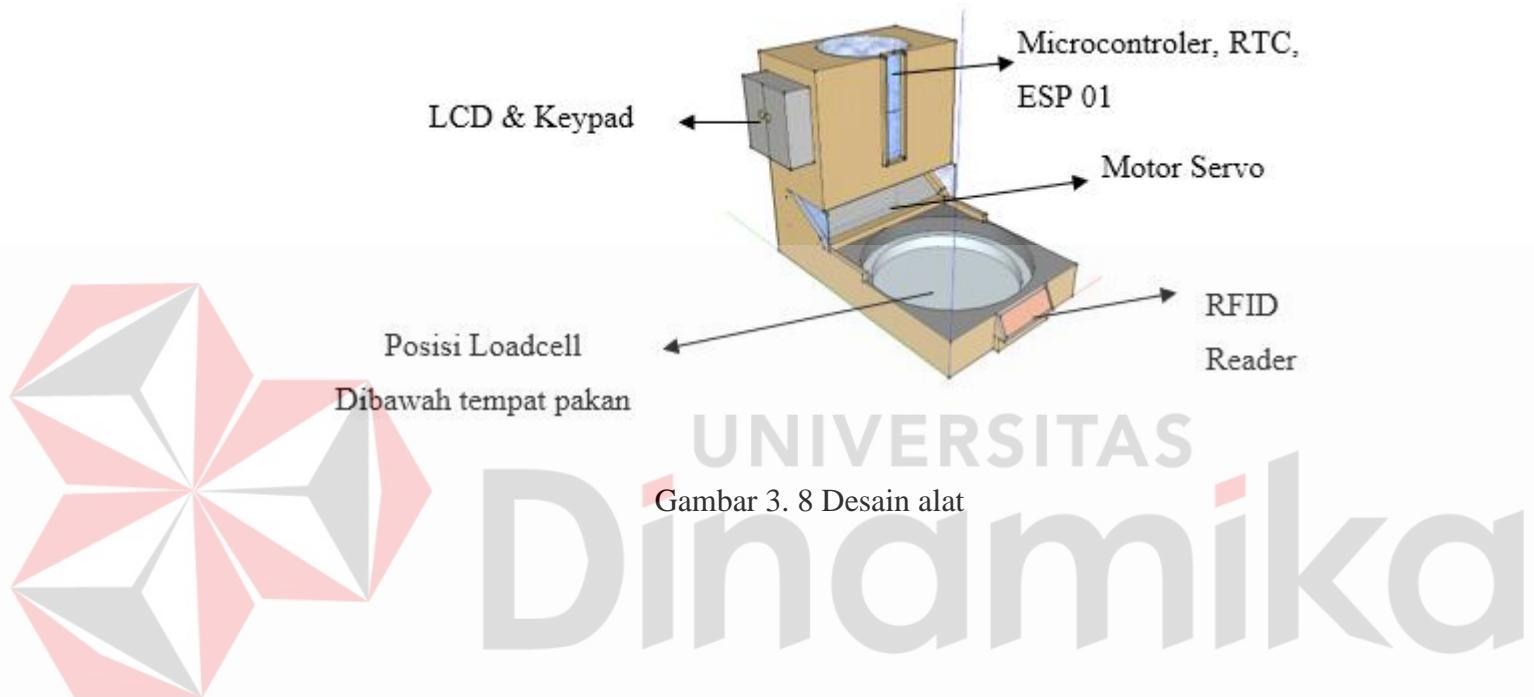


Gambar 3. 7 Struktur Json Tree database Firebase

Dari gambar 3.7 di atas, Alat_PAKAN parent dari data alat, memiliki *child* *statusDataNewFromAndroid* untuk *flag* apakah ada data baru dari *Android*, *child* *forceFeed* untuk pemberikan pakan secara langsung dari *user*, *child* *Tag* yang berisi *child* untuk pengaturan *RFID Tag*. *Child* *Tag* memiliki *child* *flagTapped* *Arduino* *Android* yang digunakan untuk *flag* ketika *user* *create new* data, *child* *TagSaved* untuk *flag* *buffer* “*id*” *RFID Tag* apa sajakah yang tersimpan di *Firebase*, dan *child* *RFIDTag* *n* yang berisi nilai-nilai *Input* dari *user* untuk tiap *RFID Tag* yang mana didalamnya terdapat nilai untuk porsi, *firstFeed*, *timer*, *latestFeed*, *namaKucing*, dan *umurKucing*.

3.4 Desain Alat

Desain alat sama seperti dengan hasil yang telah dibuat sebelumnya seperti pada gambar 3.8 di bawah, namun lebih disempurnakan lagi. Di bawah wadah makanan kucing ditambahkan sensor *Loadcell* agar jumlah makanan yang diberikan sesuai dengan keinginan *user* dan ditambahkan sistem kontrol jarak jauh (*Internet of Things*).



BAB IV

HASIL DAN PEMBAHASAN

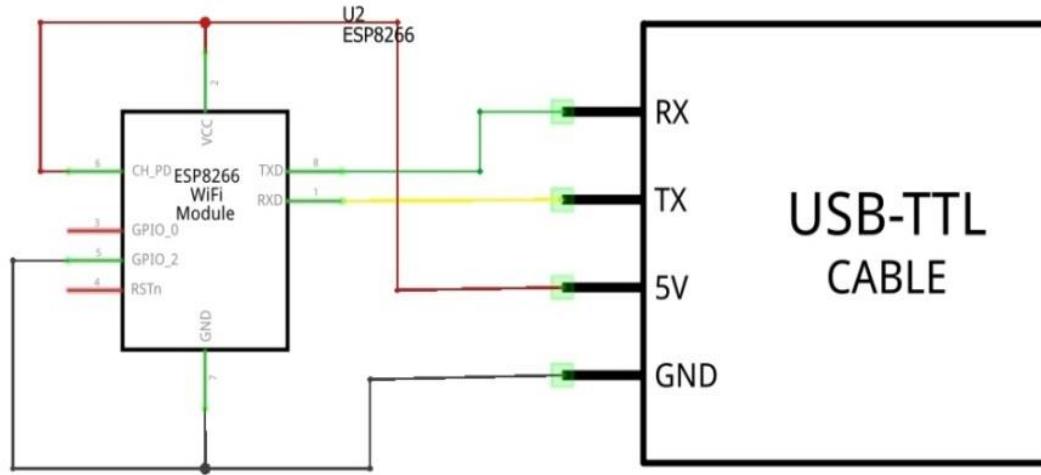
Dalam tahapan pengujian ini terbagi menjadi beberapa bagian percobaan yang telah penulis lakukan, diantaranya adalah pengujian pengiriman data antara ESP01 ke *Firebase*; *Loadcell* terhadap jumlah pakan yang diberikan melalui *Input*, serta pengujian implementasi keseluruhan sistem. Pengujian yang dilakukan meliputi pengujian terhadap sistem IoT yang dibuat mulai dari Android – Server – dan Alat maupun *Loadcell* yang dijelaskan dibawah ini:

4.1 Pengujian Pengiriman data antara ESP dengan *Firebase*

Pengujian dilakukan dengan cara pengiriman data dari ESP01 ke *Firebase*. Nantinya string yang dikeluarkan oleh ESP01 adalah paket data yang digunakan untuk Arduino mega, dan data yang terdapat pada *Firebase* yang digunakan untuk aplikasi Android.

4.1.1 Tujuan Pengujian Pengiriman data antara ESP dengan *Firebase*

Secara umum prinsip dari pemrosesan data yang berhubungan dengan *server* pasti selalu ada *create*, *read*, *update*, dan *delete* (CRUD). Pada pengujian ini dilakukan CRUD pada ESP01 yang nantinya digunakan oleh keseluruhan sistem untuk mengirim data dari Arduino ke *Firebase* dan *Firebase* ke Arduino.



Gambar 4. 1 Rangkaian ESP-01 ke USB-TTL

4.1.2 Alat Yang Digunakan Untuk Pengujian Pengiriman data antara ESP dengan *Firebase*

Agar proses pengujian pelatihan ini dapat berjalan sesuai dengan penelitian ini maka diperlukan beberapa alat sebagai berikut:

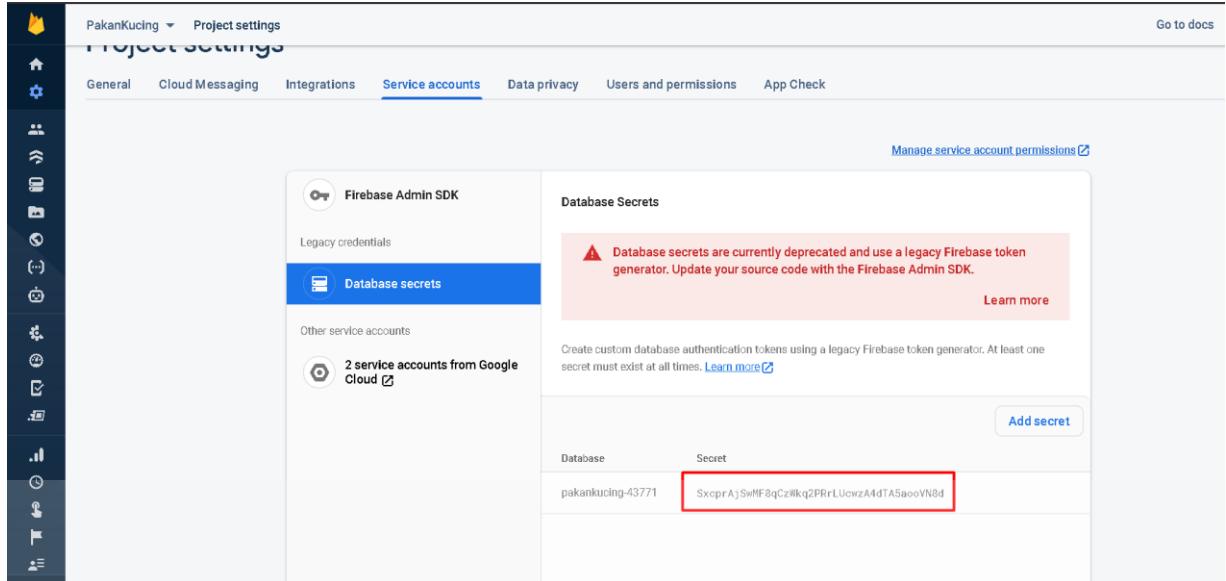
- ESP8266
- ARDUINO IDE
- Kabel Downloader
- Access Point

4.1.3 Prosedur Pengujian Pengiriman data antara ESP dengan *Firebase*

Berikut prosedur pengujian untuk kirim data dari ESP ke *Firebase*:

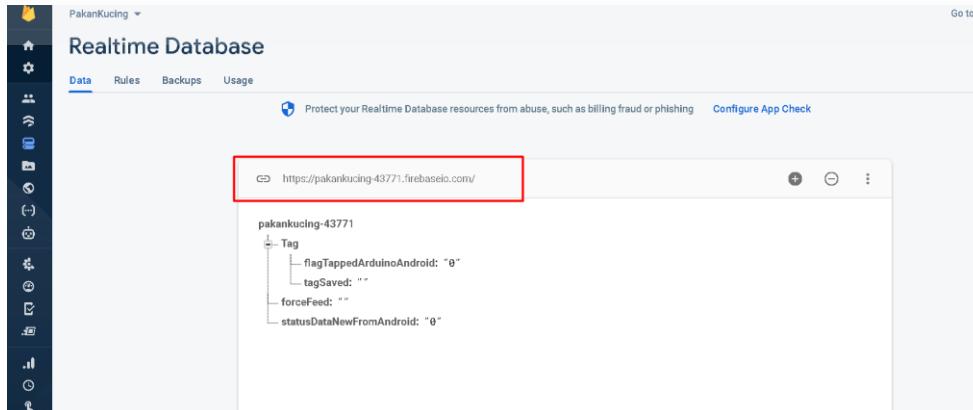
- Memasang rangkaian seperti yang ditunjukkan pada gambar 4.1 diatas.
- Menyetting database *Firebase* pada web www.firebaseio.google.com.
- Login ke *Firebase*, klik “Get Started”.
- Create database Firebase* baru:
New project -> masukkan nama *project* -> *Create project*.
- Dapatkan *Firebase host* dan *auth key*. Gambar 4.2. *Auth key Firebase* untuk ESP.

6. Membuka menu settings -> *Service accounts* -> Database secrets -> Copy secret untuk *auth key*.



Gambar 4. 2 *Auth key* *Firebase* untuk ESP

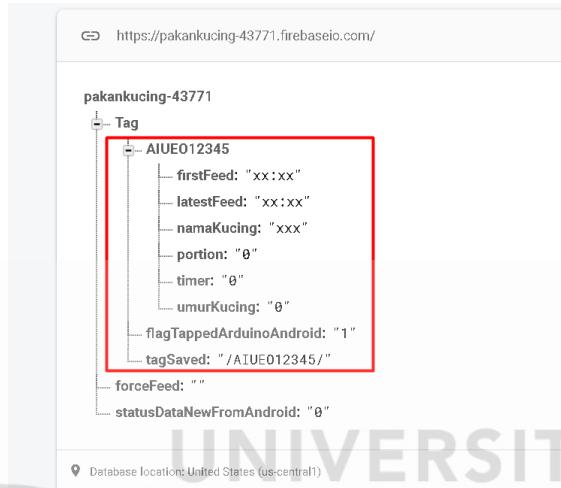
Membuka menu database -> Data -> *copy* nama *Firebase* host untuk mendapatkan nama *host*.



Gambar 4. 3 *Host* *Firebase* untuk ESP

Pada menu database -> Rules -> ubah *write* dan *read* menjadi 'false'.

7. Menyetting *Firebase* pada web sudah, kemudian menambahkan *library Firebase* pada Arduino IDE, bisa di *download* di google.
8. Mengetik *listing* program yang di *upload* ke ESP01 seperti pada contoh *script* yang terdapat pada lampiran 2.
9. Mengirim data untuk meng-*create*, kemudian melihat data yang di*create* ke *Firebase*.



Gambar 4.4 Pengujian *create* data ke *Firebase*

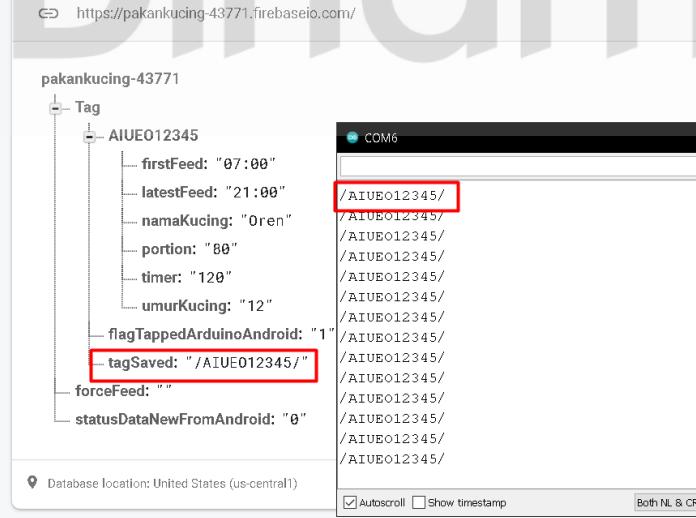
10. Mengirim data untuk *update*, kemudian melihat data yang di*update* ke *Firebase*.
11. Mengirim *string* “@AIUE012345|07:00|Oren|80|120|12|21:00” melalui serial monitor.



Gambar 4. 5 Pengujian *update* data ke *Firebase*

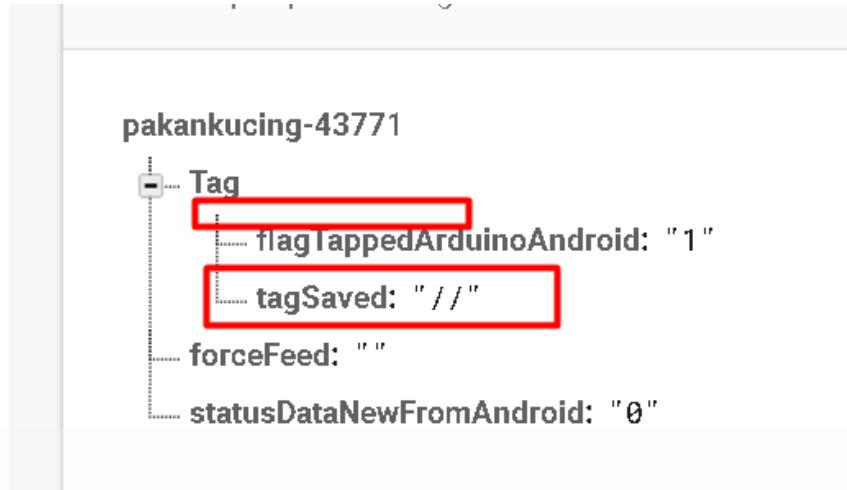
12. Mengirim data untuk *read*, melihat hasil pada serial monitor. Data yang dilihat pada serial monitor adalah data yang diterima oleh Arduino Mega.

 - Mengirimkan *string* “~” melalui serial monitor.
 - Mengirim data *Firebase* yang dibaca oleh ESP01 pada serial monitor.



Gambar 4. 6 Data pada *Firebase* yang dibaca oleh ESP01

13. Mengirim data untuk *delete*, kemudian melihat hasilnya ke *Firebase*.
 - a. Mengirimkan *string* “#AIUEO12345” melalui serial monitor.
 - b. Data RFID *Tag* pada *Firebase* terhapus.



Gambar 4. 7 Data pada *Firebase* setelah dihapus

4.1.4 Hasil Pengujian

Mengacu pada subbab prosedur 4.1.3 diatas dengan hasil yang diajukan oleh beberapa gambar dapat diketahui bahwa proses kirim data dari ESP ke *Firebase* berhasil. Proses berhasil dengan ditandai dengan terkirimnya data *flag* dari ESP ke *Firebase*.

4.2 Pengujian *Loadcell* Terhadap *Input* yang Diberikan Oleh *User*

Pengujian dilakukan untuk melihat hasil keakuratan sensor *Loadcell*.

4.2.1 Tujuan Pengujian *Loadcell* Terhadap *Input* yang Diberikan Oleh *User*

Secara umum prinsip dari *Loadcell* bekerja berdasarkan regangan dan tekanan dari *strain gauge*. Ketika beban diterapkan, badan *Loadcell* mengalami deformasi elastis yang menyebabkan terjadinya kompresi (*compression*) dan tekanan (*tension*) pada strain gauge yang terpasang yang kemudian menjadi perubahan tegangan listrik. Pada pengujian ini dilakukan pemberian *Input* oleh *user*. Untuk mengetahui apakah

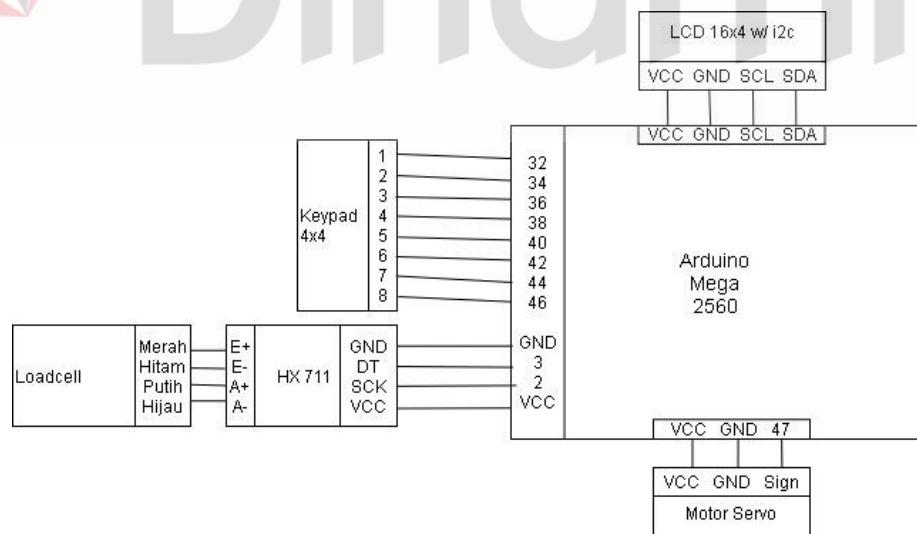
Loadcell dan motor servo sudah bekerja sesuai dengan pengaturan jumlah pakan yang diberikan.

4.2.2 Alat Yang Digunakan Pengujian *Loadcell* Terhadap *Input* yang Diberikan Oleh *User*

Agar proses pengujian pelatihan ini dapat berjalan sesuai dengan penelitian ini, maka diperlukan beberapa alat sebagai berikut:

1. *Loadcell* 1kg
2. HX711
3. Servo SG90
4. Wadah pakan
5. Timbangan dapur (gram)
6. Arduino Mega
7. Keypad

4.2.3 Prosedur Pengujian *Loadcell* Terhadap *Input* yang Diberikan Oleh *User*



Gambar 4. 8 Rangkaian pengujian *Loadcell*

Berikut prosedur pengujian untuk *Loadcell*:

1. Memasang rangkaian seperti pada gambar 4.8.
2. Membuka Arduino IDE dan mengetik *listing* program yang diupload.
3. *Upload* program ke Arduino Mega.
4. Memasukkan *Input* jumlah pakan pada *keypad*, jika sudah tekan ‘C’.



Gambar 4. 9 *Input* porsi untuk pengujian *Loadcell*

5. Servo terbuka, dan tertutup ketika jumlah pakan keluar pada gambar 4.10, yang terbaca oleh *Loadcell* telah menyentuh angka yang diberikan pada *Input*.



Gambar 4. 10 Pakan keluar

6. Menimbang jumlah pakan yang keluar menggunakan timbangan gram dapur yang digunakan untuk membuat kue pada umumnya, berapakah nilai yang didapat. Gambar 4.11 adalah hasil timbangan pakan yang keluar.



Gambar 4. 11 Hasil timbang pakan yang keluar

4.2.4 Hasil Pengujian Loadcell Terhadap Input yang Diberikan Oleh User

Dari pengujian yang telah dilakukan, didapat hasil yang menggunakan rumus menggunakan perumusan perhitungan sebagai berikut:

$$Selisih_n = output - input \quad (1)$$

$$Error = \frac{(output - input)}{input} \times 100\% \quad (2)$$

$$Rata - rata \ Error = \frac{\sum Error}{\sum n} \quad (3)$$

Tabel 4.1 Tabel pengujian *Loadcell*

No.	Input (keypad)	Output (timbangan)	Error (Input – output)	Error %
1	25	28	3	10.71
2	40	44	4	2.77
3	50	55	5	9.09
4	100	105	5	4.76
5	140	144	4	2.77
Rata–Rata Error			4.2	6.02

Dari hasil uji coba yang telah dilakukan pada tabel di atas, disimpulkan bahwa perbedaan nilai antara *Loadcell* dan timbangan gram dapur lumayan sedikit berbeda, dimana untuk hasilnya didapatkan selisih *error* sebesar 6.02%.

4.3 Pengujian Keseluruhan

4.3.1 Tujuan Pengujian Keseluruhan

Tujuan dari pengujian ini adalah untuk mengetahui keberhasilan dari keseluruhan program yang telah dibuat apakah dapat bekerja dengan baik dan berjalan lancar dengan menggunakan *Internet of Things* (IoT).

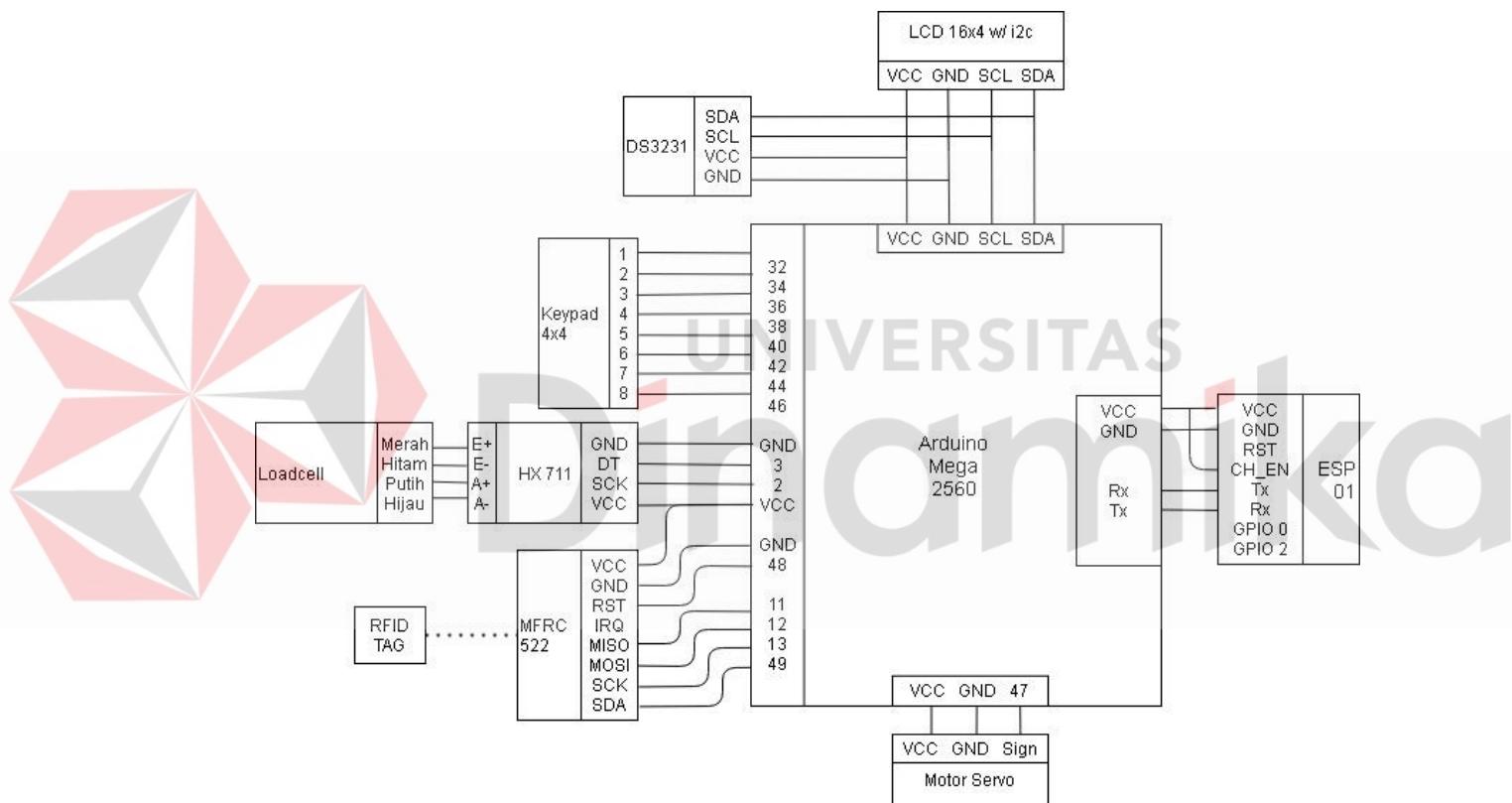
4.3.2 Alat Yang Digunakan Pengujian Keseluruhan

Agar proses pengujian pelatihan ini dapat berjalan sesuai dengan penelitian ini maka diperlukan beberapa alat sebagai berikut:

1. RTC DS3231
2. LCD
3. *Keypad*
4. *Loadcell* 1kg
5. HX711
6. Servo SG90
7. RFID Tag
8. RFID Reader MFRC 522
9. Wadah pakan

10. Arduino Mega
11. ESP01
12. Arduino IDE
13. Android Studio
14. Access poin

4.3.3 Prosedur Pengujian Keseluruhan

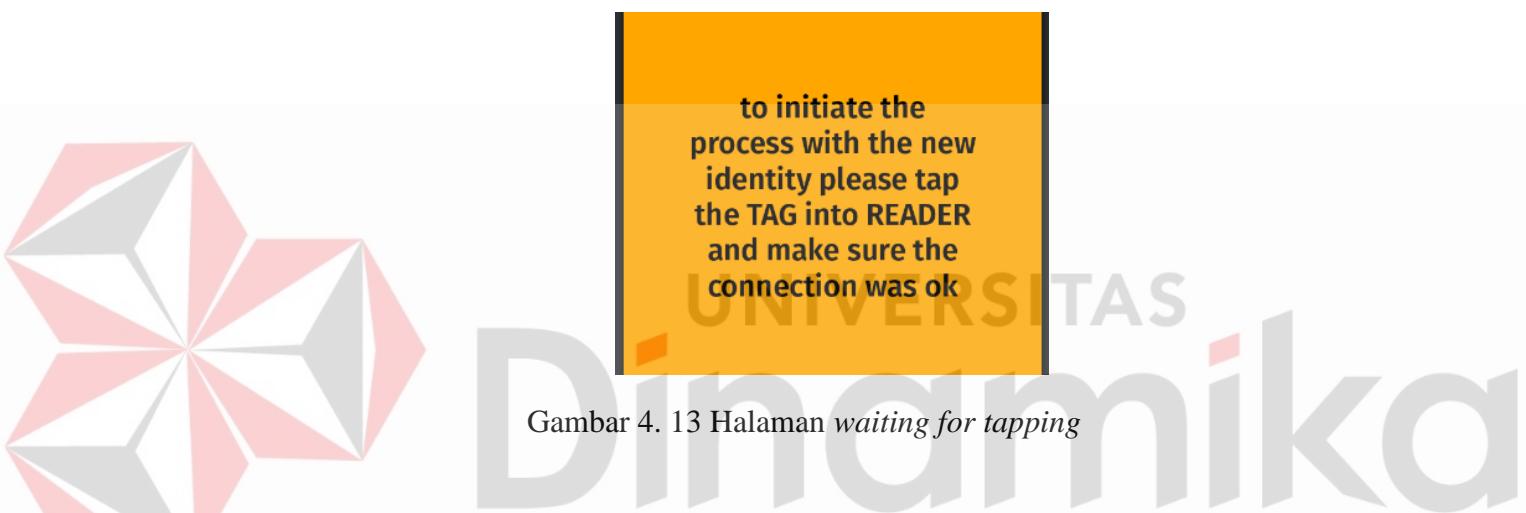


Gambar 4. 12 Rangkaian *Schematic* untuk pengujian sistem secara keseluruhan

Berikut adalah prosedur pengujian alat secara keseluruhan:

1. Membuka Arduino IDE dan mengetik *listing* program untuk ESP01 dan Arduino Mega yang *diupload*. Untuk isi programnya dapat dilihat pada lampiran 1.

2. Ketika program pada Arduino Mega telah *diupload*, maka LCD menampilkan tulisan awal seperti pada gambar 4.13 di bawah ini, yang berarti alat sedang jalan dalam mode “*run*”.
3. Mendaftarkan RFID Tag yang ingin disimpan. Berikut ini adalah cara mendaftarkan RFID Tag:
 - a. Membuka aplikasi, kemudian pada halaman kedua ketika ada pesan “*are you new*” pilih “*yes*”. Kemudian aplikasi berpindah ke halaman “*waiting for tapping*”.



Gambar 4. 13 Halaman *waiting for tapping*

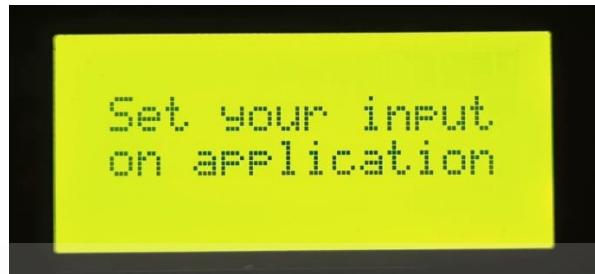
- b. Kemudian berpindah pada alat, tekan ‘C’, lalu tapping RFID Tag. Jika RFID Tag telah terdaftar pada EEPROM, maka menampilkan LCD pada gambar 4.15, dan aplikasi diback ke halaman pertama. Jika rfid Tag belum terdaftar, maka menampilkan LCD sesuai dengan gambar 4.14, 4.15, dan 4.16.



Gambar 4. 14 *Display LCD untuk tap create Tag baru*

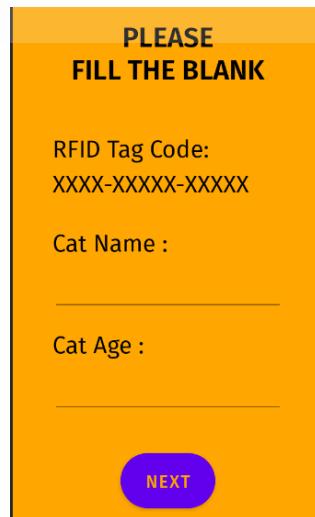


Gambar 4. 15 RFID Tag telah terdaftar di EEPROM

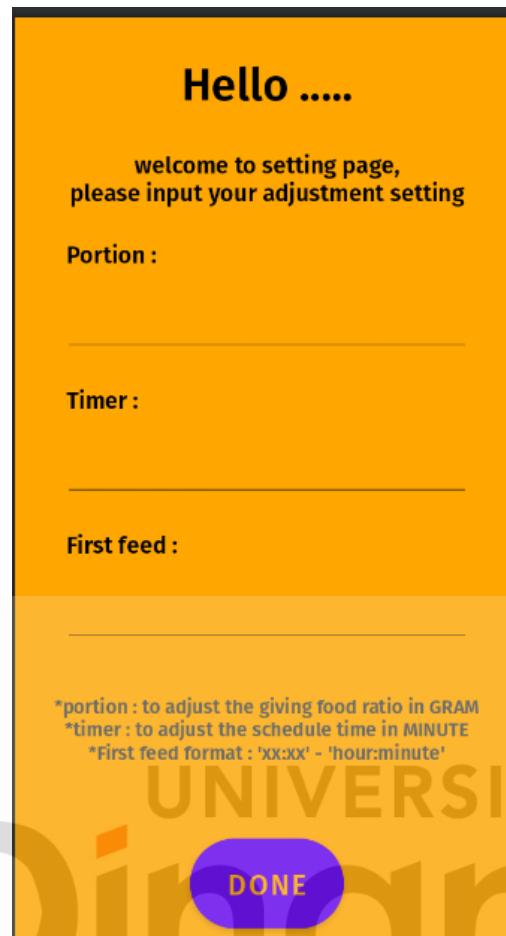


Gambar 4. 16 Display LCD jika RFID Tag belum terdaftar

- c. Berpindah ke aplikasi android. Setelah RFID Tag ditapping, halaman secara otomatis berpindah ke halaman selanjutnya untuk *Input setting*. Memasukkan *Input* settingan yang diinginkan sesuai dengan parameter-parameter nya seperti pada gambar 4.17. Gambar 4.18 merupakan halaman *Input* porsi, *timer*, dan *first feed*.

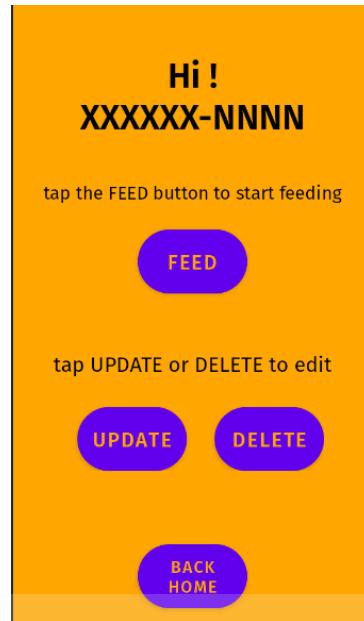


Gambar 4.177 Halaman *Input* nama dan umur kucing



Gambar 4. 18 Halaman *Input* porsi, *timer*, dan *first feed*

- d. Data RFID Tag yang baru telah tersimpan pada *Firebase*, selanjutnya Arduino Mega melakukan sinkronisasi data pada EEPROM agar sama dengan data pada *Firebase*.
- 4. Untuk melakukan *update*, *delete* data, atau memberikan pakan secara langsung (*force feed*) dapat dilakukan pada aplikasi atau alat secara langsung, berbeda dengan proses ketika *create*. Di aplikasi, pada halaman kedua ketika ada pesan “are you new” pilih “no” kemudian pilih RFID Tag pada list. Kemudian menjalankan aplikasi sesuai dengan *button-button* yang tersedia dan intruksi yang ditampilkan seperti pada gambar 4.19.



Gambar 4. 19 Halaman *user action*

Sedangkan pada alat, tekan ‘D’ untuk *delete*, tekan ‘B’ untuk *update*, dan tekan ‘A’ untuk kembali ke mode sequensial. Kemudian ikuti intruksi yang tampil pada LCD pada gambar 4.20, dan gambar 4.21.



Gambar 4. 20 Tampilan LCD ketika *delete*



Gambar 4. 21 Tampilan LCD ketika *update*

5. Ketika mode sequensial alat terus membaca apakah ada data terbaru pada *Firebase* untuk dilakukan *update data* pada EEPROM, dan dilakukan pembacaan waktu. Jika waktu pakan sesuai dengan penjadwalan dan RFID Tag menempel pada MFRC522, servo terbuka kemudian menutup ketika *Loadcell* telah membaca *Input* nilai sesuai dengan data yang tersimpan pada EEPROM. Jika ingin memberikan makan secara langsung, *user* dapat menggunakan *force feed* yang terdapat pada aplikasi. Gambar 4.22 menunjukkan RFID Tag saat tapping. Gambar 4.23 Servo terbuka dan tertutup sesuai penjadwalan.



Gambar 4. 22 RFID Tag Ter-tapping



Gambar 4. 23 *Servo* terbuka dan tertutup sesuai penjadwalan

4.3.4 Hasil Pengujian Keseluruhan

Berdasarkan dari pengujian keseluruhan diatas, berikut adalah hasil dari pengujian terhadap pemberian porsi pakan yang ditunjukkan pada tabel 4.2 di bawah ini dan penjadwalan menggunakan *Internet of Things* (IoT) yang tersinkronisasi dengan EEPROM yang ditunjukkan pada tabel 4.3 di bawah ini. Tabel 4.2 menggunakan perumusan perhitungan *error* sebagai berikut:

$$Selisih_n = output - input \quad (1)$$

$$Keberhasilan = \frac{(output - input)}{input} \times 100\% \quad (2)$$

$$Error = 100 - keberhasilan \quad (3)$$

$$Total Keberhasilan = \frac{\sum Keberhasilan}{\sum n} \quad (4)$$

Tabel 4.2 Pengujian takaran porsi makanan

Percobaan Ke (n)	Input	Output	Selisih	Keberhasilan
1	25	19	6	76%
2	25	18	7	72%
3	25	21	4	84%
4	25	26	1	96%
5	25	28	3	88%
6	50	42	8	84%
7	50	45	5	90%
8	50	44	6	88%
9	50	48	2	96%
10	50	50	0	100%
11	100	107	7	93%
12	100	99	1	99%
13	100	86	14	86%
14	100	113	13	87%
15	100	87	13	87%
Rata-Rata Keberhasilan				88,4%
Rata – Rata Error				11,6%

Berdasarkan tabel 4.2 di atas pada pengujian takaran porsi makanan, dapat disimpulkan bahwa tingkat keberhasilan yang didapat setelah dilakukan uji coba sebanyak 15 kali adalah rata-rata 88,4% dan rata rata *Error* sebesar 11,6%.

Tabel 4.3 Pengujian penjadwalan pemberian makanan

Percobaan ke-	Input Waktu	Ouput waktu	Timer (Jeda pakan/menit)	Selisih	Hasil
1	19:42	19:42	1	0	Akurat
2	19:44	19:44	2	0	Akurat
3	19:46	19:046	3	0	Akurat

Pada tabel 4.3 di atas, dari sampel hasil uji coba terhadap RTC yang digunakan untuk penjadwalan pakan, dapat disimpulkan bahwa sistem *timer* dapat digunakan dan berfungsi dengan baik tanpa ada kesalahan.

BAB V

PENUTUP

5.1 Kesimpulan

Dalam tahap pengujian terbagi menjadi beberapa bagian percobaan, diantaranya pengujian pembacaan *Loadcell* pada gram untuk banyaknya jumlah pakan; pengujian pengiriman data ke *Firebase* menggunakan Arduino Mega dan ESP-01; serta implementasi keseluruhan sistem ketika alat dijalankan secara remote jarak jauh (IoT) yang kemudian penulis dapat simpulkan adalah sebagai berikut:

1. Pada penelitian ini, *Loadcell* 1 kg yang diintegrasikan dengan Arduino Mega2560 sebagai alat kendali. Dalam pengujian, *Loadcell* dapat membaca dengan baik dan didapatkan selisih *error* sebesar 11.6%.
2. Pengiriman data ke *Firebase* dari Arduino Mega melalui ESP-01 dapat dilakukan dengan baik. Ketika sinyal dan koneksi jelek, data tidak hilang melainkan di *buffer* (masuk antrian) terlebih dahulu. Sistem *buffer* ini sudah terdapat pada *Firebase*.
3. Pada pengujian terhadap pemberian pakan jarak jauh (IoT), data-data yang dikirimkan dan tersimpan telah berjalan sesuai dengan harapan. Namun jumlah pakan yang diberikan terdapat selisih 6-13 gram.

5.2 Saran

Ada beberapa hal yang perlu dilakukan, agar hasil penelitian ini lebih baik lagi. Berikut ini merupakan saran bagi yang ingin melanjutkan penelitian ini:

1. *User Interface* pada aplikasi Android perlu diperbaiki lagi untuk ditambahkan menu karena disini penulis hanya menekankan pada fungsi dan penggunaan teknikal.
2. Posisi wadah dan *Loadcell* lebih baik jika tidak berdekatan dengan obyek lain, untuk menghindari *error* pembacaan.

DAFTAR PUSTAKA

- Andrianto. (2010). Pemrograman Mikrokontroller AVR ATMEGA16. Bandung: Informatika.
- Beth, M., Harianto, & Ira. (2019). Rancang Bangun Pemberi Pakan otomatis Pada Kucing Menggunakan Mikrokontroller. *Jcones*.
- Eka, F. (2017). *Kontrol Dan Monitoring Smarthome Dengan Modul ESP8266 Serta Server Thingspeak*. Balik Papan: Politeknik Negeri Balikpapan.
- Hapsari, N. T. (2012). *Perancangan Sistem Kendali Jarak Jauh Peralatan Listrik Rumah Tangga Dengan Kontrol Wireless Pada Orange House Menggunakan Mobile Application Berbasis Android*. Surabaya: Universitas Telkom.
- Mahali, M. I. (2016). *Smart Door Locks Based On Internet Of Things Concept With Mobile Backend as a Service*. Jakarta: Universitas Indonesia.
- Nugroho, A. (2008). *Rancang Bangun Dan Pemrograman Sistem Transmisi Data GPS Menggunakan Teknologi CSD Sebagai Aplikasi Sistem Penjejakan Posisi Berbasis Mikrokontroller AVR-ATMEGA8535*. Jakarta: Universitas Indonesia.
- Nusdianto. (2016). Pakan Dan Kucing. In *Kesehatan Dan Risiko Penyakit Akibat Pakan Pada Kucing*. Faculty of Veterinary Medicine Universitas Airlangga.
- Owen. (2004). *Dasar-dasar Elektronika*. Jakarta: Erlangga.
- Sulaiman, O. K. (2017). *Sistem Internet Of Thiings (IOT) Berbasis Cloud Computing Dalam Campus Area Network*. Jakarta: Universitas Indonesia.
- Suwed, M. A., & Napitupulu, R. M. (2011). *Panduan Lengkap Kucing*. Jakarta: Penebar Swadaya.