

# **Laporan Akhir Produk IoT Catfeeder**

**“OnlyFeed”**

Ditujukan untuk memenuhi tugas mata kuliah IoT



Disusun Oleh

Kelompok :

Rafi Putra Darmawan	15-2021-082
La Ode Rafif Setiawan	15-2021-117
Alfa Sendy Wiranata	15-2021-129
Anggit Nur Ridho	15-2021-134
Muhammad Ghaza A.L	15-2021-166

**PROGRAM STUDI INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL**

**2024**

# 1. Deskripsi Produk IoT

## Latar belakang

Perkembangan teknologi saat ini semakin pesat khususnya di bidang Internet of Things (IoT). IoT adalah sebuah konsep di mana teknologi seperti sensor dan perangkat lunak ditanamkan ke dalam objek untuk berkomunikasi, mengontrol, menghubungkan, dan bertukar data dengan perangkat lain sambil tetap terhubung ke Internet. Perkembangan IoT merambah ke berbagai sektor, termasuk sektor hewan peliharaan. Salah satu contohnya adalah perangkat Onlyfeed, yang dirancang untuk memberi makan kucing secara otomatis. Perangkat ini menggunakan teknologi IoT untuk memberi pemilik kucing akses jarak jauh melalui aplikasi seluler atau situs web khusus.

Perangkat Onlyfeed merupakan solusi bagi pemilik kucing yang tidak mempunyai waktu untuk terus menerus memberi makan hewan peliharaannya. Perangkat ini memungkinkan pemilik kucing dapat otomatis mengatur pemberian makan dan dapat memantau suhu tubuh kucing dan lingkungan melalui aplikasi seluler atau situs web. Perangkat Onlyfeed berpotensi meningkatkan kualitas hidup kucing dan pemiliknya. Perangkat ini memastikan kucing mendapat nutrisi yang tepat dan memungkinkan pemilik kucing tetap terhubung dengan kucingnya meski tidak di rumah.

## Tujuan pengembangan

Berdasarkan latar belakang di atas, dapat kita simpulkan bahwa tujuan pengembangan produk OnlyFeed IoT adalah:

- Mengatasi masalah pada kucing milik orang yang mempunyai sedikit waktu untuk merawatnya.
- Pemilik kucing dapat dengan mudah memberi makan kucingnya kapan saja, di mana saja.
- Meningkatkan kualitas hidup kucing dan pemiliknya.

Lebih spesifiknya, produk OnlyFeed dikatakan memiliki manfaat sebagai berikut:

- Kucing bisa mendapatkan asupan nutrisi yang tepat meski pemiliknya tidak di rumah.
- Pemilik kucing dapat tetap berhubungan dengan kucingnya meskipun mereka tidak di rumah.
- Pemilik kucing dapat memantau pola makan kucingnya dan melihat apakah kucingnya mendapatkan nutrisi yang cukup dan seimbang.

Produk OnlyFeed ditujukan sebagai solusi tepat bagi pemilik kucing yang tidak memiliki waktu untuk terus-menerus memberi makan hewan peliharaannya. Produk ini membantu meningkatkan kualitas hidup kucing dan pemiliknya.

## Manfaat dari produk

Berdasarkan latar belakang dan tujuan pengembangan yang telah dibahas di atas, maka dapat kita simpulkan bahwa keunggulan produk Onlyfeed adalah:

- Meningkatkan Kualitas Hidup Kucing  
Kucing dijamin asupan nutrisinya benar baik jumlah maupun jumlahnya. Ketika kualitas meningkat, kualitas hidup juga meningkat. Kucing memiliki energi yang cukup untuk beraktivitas dan terhindar dari berbagai masalah kesehatan akibat kekurangan gizi.
- Membantu Pemilik Kucing  
Pemilik kucing yang menggunakan Onlyfeed akan mendapat dukungan dalam merawat kucingnya. Anda tidak perlu khawatir kucing Anda akan lapar atau makan sembarangan saat Anda tidak di rumah. Selain itu, Onlyfeed juga membantu pemilik kucing memantau pola makan kucingnya dan memastikan kucingnya mendapatkan pola makan yang tepat dan seimbang.

Berikut adalah beberapa manfaat khusus produk Onlyfeed:

- Menjaga pola makan kucing Anda tetap teratur.

Onlyfeed dapat diatur untuk memberi makan kucing Anda secara otomatis pada jadwal yang telah ditentukan. Hal ini penting untuk memastikan kucing Anda menerima cukup makanan secara teratur.

- Mencegah kucing makan secara sembarangan  
Kucing yang terjamin asupan nutrisinya dengan Onlyfeed akan terhindar dari keinginan makan yang tidak disengaja. Ini membantu mencegah masalah kesehatan seperti obesitas dan kecanduan pada kucing.
- Alternatif pemilik kucing dalam mengurus kucing  
Onlyfeed adalah solusi sempurna bagi pemilik kucing yang sibuk. Alat ini dapat memberi makan kucing secara otomatis, sehingga pemilik kucing tidak perlu khawatir kucingnya kelaparan saat berada di luar.

Secara umum Onlyfeed merupakan produk inovatif yang memberikan manfaat besar bagi kucing dan pemiliknya.

## 2. Poster

**OnlyFeed**  
"ONLY-FEED"  
FEED YOUR CAT ANYWHERE & ANYTIME

**APA ITU ONLY FEED?**  
Onlyfeed adalah perangkat atau produk yang dirancang untuk memberi makan kucing secara otomatis. Dalam hubungan dengan IOT produk tersebut dapat terhubung ke koneksi internet sehingga memungkinkan pemilik kucing untuk mengaksesnya dari jarak jauh melalui aplikasi seluler atau situs web khusus.

**PASTIKAN KUCING ANDA TETAP KENYANG!!!**  
Beri makan kucing kapanpun dimanapun

**BISA PAKE APLIKASI ATAU WEBSITE!**  
Website dan Aplikasi terhubung ke alat secara langsung!

**PANTAU SUHU KUCING ANDA!!**  
Pantau suhu tubuh kucing melalui alat yang dibuat dan terkoneksi langsung secara realtime

**ANALISIS KEBUTUHAN KUCING ANDA**  
Analisis dari history yang ada!

### **3.Fitur Produk IoT**

a. Cat feeder

Fitur utama dari produk IoT ini mencakup kemampuan inovatif untuk memberikan pakan pada kucing dengan sekali sentuhan tombol yang terdapat di perangkat fisik, situs web, dan aplikasi seluler. Dengan fitur ini, pengguna dapat memberi makan kucing mereka dengan mudah, bahkan ketika mereka berada jauh dari rumah

b. Real—time data stock makanan, suhu ruangan dan suhu hewanya

Fitur ini memungkinkan pengguna untuk mengalirkan data secara real-time yang diambil dari sensor. Data yang dimasukkan akan ditampilkan secara langsung pada halaman yang tersedia dalam aplikasi mobile atau web.

c. Chart grafik berapa banyak pemberian makan dan rata-rata suhu lingkungan di 5 hari terakhir

Fitur ini melibatkan pengambilan data dari API database dan menampilkan data tersebut dalam bentuk grafik.

d. History Pemberian makan

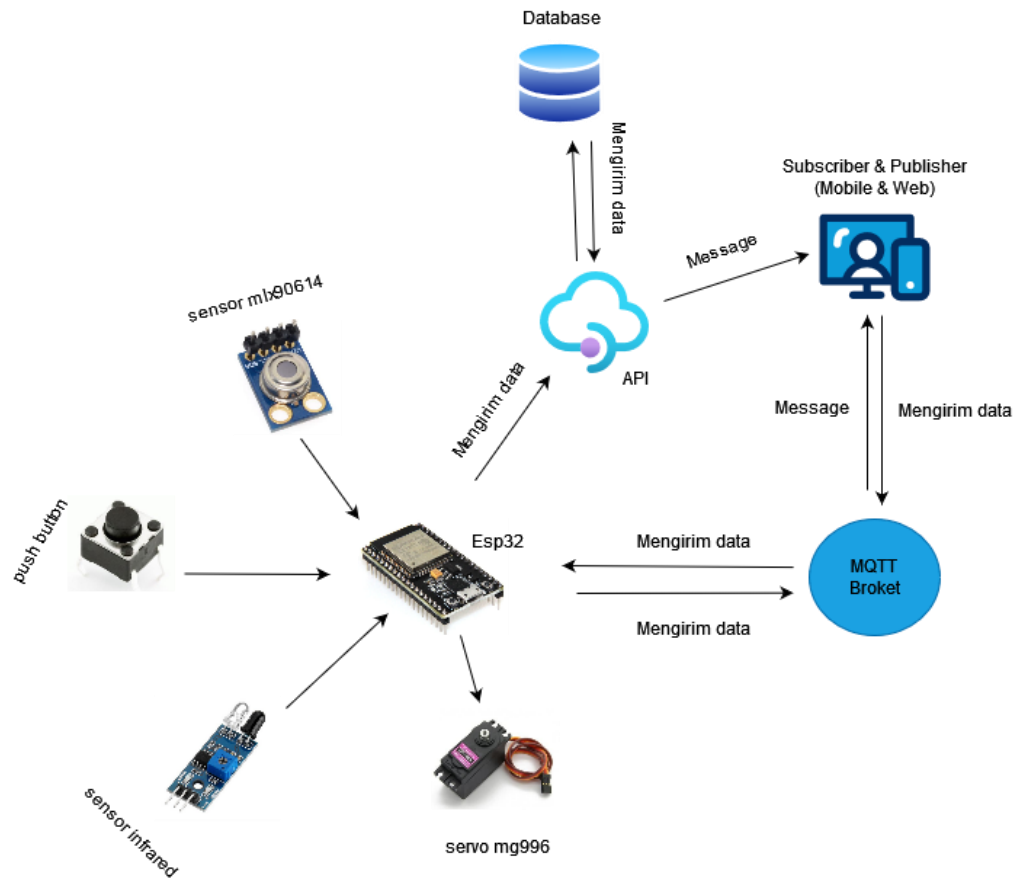
Fitur ini melibatkan pengambilan data dari API database lalu data tersebut akan di tampilkan dalam listview, dan data akan di ambil terus-menerus dari API.

e. Manual push button

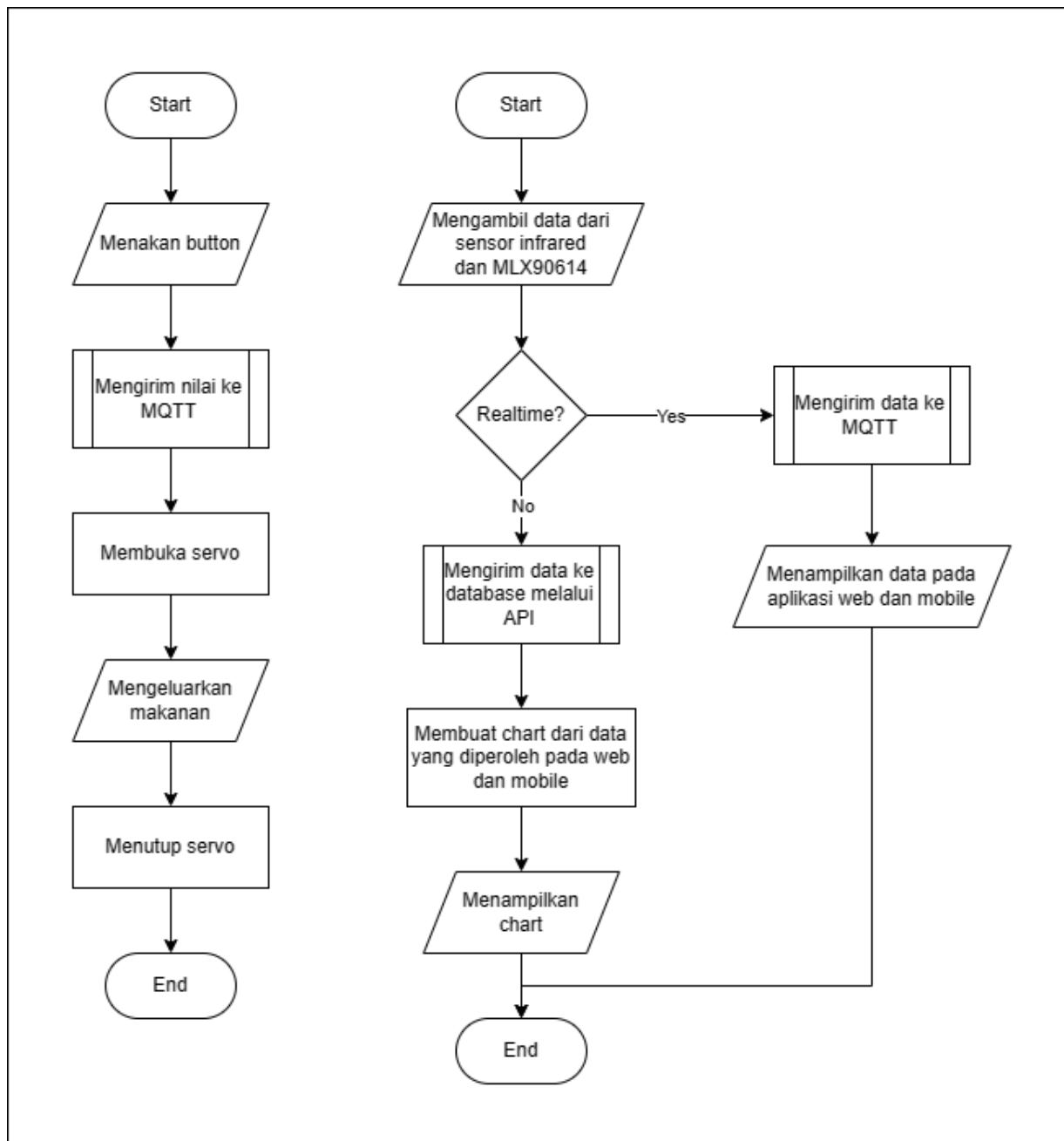
Dengan fitur ini, pengguna dapat memberikan makanan secara langsung atau secara manual melalui perangkat menggunakan push button.

#### 4. Skema Produk IoT

Adapun skema dari produk furryFeeder kelompok kami seperti berikut:



#### 5. Cara Kerja



Penjelasan:

### 1. Memberi makan kucing

- User menekan button pada alat atau pada aplikasi mobile atau web
- Servo akan terbuka selama 1 detik
- Makanan akan terbuka
- Lalu servo akan kembali menutup

### 2. Menampilkan data

- Sensor infrared dan MLX90614 akan mengirim data ke MQTT secara realtime dan ke database melalui API.
- Data dari MQTT akan ditampilkan secara realtime pada aplikasi web dan mobile.
- Data dari database akan dibuatkan chart, lalu ditampilkan pada aplikasi web dan mobile. Chart berupa rata-rata suhu pada lima hari terakhir dan jumlah pemberian makan pada lima hari terakhir.

## 6. Mekatronika

### 1. Esp32



Penjelasan: ESP32 adalah mikrokontroler yang memiliki kemampuan WiFi dan Bluetooth. Dikembangkan oleh Espressif Systems, ESP32 menawarkan kecepatan tinggi, konektivitas nirkabel, serta kemampuan pemrosesan yang baik untuk aplikasi IoT.

Spesifikasi:

- a. Prosesor: Dual-core Tensilica LX6 240 MHz
- b. WiFi: 802.11 b/g/n
- c. Bluetooth: v4.2 BR/EDR dan BLE
- d. GPIO: Bervariasi tergantung pada model

Kegunaan pada sistem: ESP32 digunakan sebagai otak utama atau mikrokontroler pusat pada sistem mekatronika ini. Mengatur dan mengkoordinasikan interaksi antara sensor dan aktuator, serta bertanggung jawab untuk pemrosesan data dan pengiriman informasi melalui koneksi WiFi atau Bluetooth.

## 2. Sensor Infrared



Penjelasan: Sensor Infrared digunakan untuk mendeteksi radiasi inframerah yang dipancarkan oleh suatu objek. Sensor ini bekerja dengan mendeteksi perubahan suhu pada objek.

Spesifikasi:

- a. Rentang Panjang Gelombang: 700 nm - 1 mm
- b. Tegangan Operasional: 2.5V - 5.5V
- c. Arus Operasional: 0.5 mA

Kegunaan pada sistem: Sensor Infrared dapat digunakan untuk mendeteksi keberadaan atau pergerakan objek dalam suatu ruangan. Pada proyek ini infrared berguna untuk menentukan apakah makanan kucing tersedia atau tidak.

## 3. Sensor MLX90614



Penjelasan: Sensor MLX90614 adalah sensor suhu berbasis inframerah yang mampu mengukur suhu suatu objek tanpa kontak fisik.

Spesifikasi:

- a. Rentang Pengukuran:  $-70^{\circ}\text{C}$  hingga  $+380^{\circ}\text{C}$
- b. Presisi:  $\pm 0.5^{\circ}\text{C}$  (dalam rentang  $0-50^{\circ}\text{C}$ )

- c. Tegangan Operasional: 3.6V - 5V

Kegunaan pada sistem: Sensor ini digunakan untuk mengukur suhu ruangan dan suhu kucing.

#### 4. Servo (Aktuator)



Penjelasan: Servo merupakan aktuator yang dapat dikendalikan secara presisi untuk menggerakkan suatu objek ke posisi tertentu.

Spesifikasi:

- Weight: 13.4g
- Dimension: 22.8×12.2×28.5mm
- Stall torque: 1.8kg/cm (4.8V); 2.2kg/cm (6.6V)
- Operating speed: 0.10sec/60degree (4.8V); 0.08sec/60degree (6.0V)
- Operating voltage: 4.8V
- Temperature range: 0°C\_ 55°C
- Dead band width: 1us
- Power Supply: Through External Adapter
- servo wire length: 25 cm

Kegunaan pada sistem: Servo digunakan untuk menggerakkan suatu mekanisme atau objek berdasarkan input dari mikrokontroler. Pada projek ini servo digunakan untuk membuka dan menutup tempat penampung makanan pada alat.

#### 5. Push Button



Penjelasan: Push Button adalah tombol sederhana yang dapat ditekan untuk memberikan input atau memicu suatu tindakan pada sistem.

Spesifikasi:

- Arus : 5mA
- Tegangan : 3.3V

Kegunaan pada sistem: Push Button dapat digunakan sebagai input pengguna untuk menggerakkan servo secara manual.

#### 7. Web

Tampilan UI :





Penjelasan : *Loading page* yang akan mendarat ke halaman utama. Untuk pengembangan lebih lanjut, bisa menambahkan sistem login di page ini.

Hello, xxx!

Today is : Tue, 09 Jan 2024



### Dashboard

Suhu Hewan	26.87 °C	Suhu Ruangan	30.28 °C
Kondisi Makanan	On Stock	Waktu Terakhir	2024-01-09 16:12:20

### Grafik Rata-rata Suhu

### Menu Control

Manual Feeding	Schedule
Data History	Profile



Penjelasan : Halaman utama, yang akan menampilkan streaming data dari mqtt(Suhu Hewan, Suhu Ruangan, Kondisi Makanan) juga mengolah dari link API(Waktu Terakhir, grafik rata-rata suhu). Juga Menampilkan "Menu Control" yang berfungsi untuk melakukan manual feeding, mengatur schedule, melihat data History, juga melakukan update profile yang akan diarahkan ke masing-masing landing page nya.

Hello, xxx!

Today is : Tue, 09 Jan 2024




### Dashboard

Suhu Hewan	26.87 °C	Suhu Ruangan	30.28 °C
Kondisi Makanan	On Stock	Waktu Terakhir	2024-01-09 16:12:20

Grafik Rata-rata Suhu

### Menu: Manual



Tap the button to feed

Penjelasan : Ketika menu manual di klik, maka akan menampilkan button "tap to feed" yang ketika gambar di klik, akan mempublish data '1' ke broker mqtt

Search EngineGoogleSocial MediaCourseTools Pro

127.0.0.1:8000 says  
Feeding success!

Home TimerAll Bookmarks

Hello, xxx!


Today is : Tue, 09 Jan 2024

### Dashboard

Suhu Hewan	26.87 °C	Suhu Ruangan	30.28 °C
Kondisi Makanan	On Stock	Waktu Terakhir	2024-01-09 16:12:20

Grafik Rata-rata Suhu

### Menu: Manual



Tap the button to feed

Penjelasan : berhasil atau tidaknya publish data akan ditampilkan di informasi jendela pop up yang muncul.

Hello, xxx!

Today is : Tue, 09 Jan 2024



## Dashboard

Suhu Hewan26.87 °C

Suhu Ruangan30.28 °C

Kondisi MakananOn Stock

Waktu Terakhir2024-01-09 16:12:20

Grafik Rata-rata Suhu

## Menu: Data History

Data Pemberian Makan(API)
2024-01-09 16:12:20
2024-01-09 16:11:50
2024-01-09 16:09:32
2024-01-09 16:08:49
2024-01-09 16:08:46
2024-01-09 16:08:44
2024-01-09 16:08:41
2024-01-09 16:08:36
2024-01-09 16:08:22
2024-01-09 16:07:49

Penjelasan : ketika menu data history di klik, akan menampilkan beberapa data API yang pastinya sudah dikonfigurasi dengan alat.

Source code :

### Dashboard.blade.php

```
resources > views > layouts > dashboard.blade.php > html > head
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7
8     <title>{{ $title }} - OnlyFeed</title>
9     <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>
10    <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js" type="text/javascript"></script>
11    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/4.4.1/chart.min.js" integrity="sha512-L0Sh17nXXzI1BSUUPpxrokqq40jqgZFQczTY1" type="text/javascript"></script>
12    <script src="mqttws31.js" type="text/javascript"></script>
13    <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
14    <!-- Style -->
15    <link rel="stylesheet" href="{{ asset('css/style.css') }}">
16
17
18    <script type="text/javascript">
19        var defaultServer = "broker.hivemq.com"; // Ganti dengan alamat broker default Anda
20        var defaultPort = "8080"; // Ganti dengan port default Anda
21        var topikDataSuhuHewan = "catfeeder-iot-klmpk2-datasuhuhewan"; // Nilai default untuk Subscribe Topic
22        var topikDataSuhuLingkungan = "catfeeder-iot-klmpk2-datasuhulingkungan";
23        var topikDataInfrared = "catfeeder-iot-klmpk2-datainfrared";
24        var connected_flag = 0;
25        var mqtt;
26        var reconnectTimeout = 2000;
27
28        document.addEventListener("DOMContentLoaded", function() {
29            // Panggil MQTTconnect() secara otomatis
30            MQTTconnect();
31        });
32    </script>
```

```

32
33     function onConnectionLost() {
34         console.log("connection lost");
35         connected_flag = 0;
36     }
37
38     function onFailure(message) {
39         console.log("Failed");
40         setTimeout(MQTTconnect, reconnectTimeout);
41     }
42
43     function onMessageArrived(r_message) {
44         out_msg = r_message.payloadString + " &nbsp;&deg;C";
45         // Check the topic to determine which element to update
46         if (r_message.destinationName === topikDataSuhuHewan) {
47             // Update element for suhu hewan
48             document.getElementById("pesanSuhuHewan").innerHTML = out_msg;
49         } else if (r_message.destinationName === topikDataSuhuLingkungan) {
50             // Update element for suhu lingkungan
51             document.getElementById("pesanSuhuLingkungan").innerHTML = out_msg;
52         } else if (r_message.destinationName === topikDataInfrared) {
53             if (r_message.payloadString == "1") {
54                 out_msg = "On Stock";
55                 document.getElementById("pesanInfrared").innerHTML = out_msg;
56             } else if (r_message.payloadString == "0") {
57                 out_msg = "Off Stock";
58                 document.getElementById("pesanInfrared").innerHTML = out_msg;
59             }
60         }
61     }

```

```

64         console.log(" in onConnected " + reconn);
65     }
66
67     function onConnect() {
68         connected_flag = 1;
69         console.log("on Connect " + connected_flag);
70
71         // Gunakan nilai default untuk Subscribe Topic
72         var stopic = topikDataSuhuHewan;
73         console.log("Subscribing to topic =" + stopic);
74         mqtt.subscribe(stopic);
75
76         var stopiklingkungan = topikDataSuhuLingkungan;
77         console.log("Subscribing to topic lingkungan =" + stopiklingkungan);
78         mqtt.subscribe(stopiklingkungan);
79
80         var stopikInfrared = topikDataInfrared;
81         console.log("Subscribing to topik Infrared =" + stopikInfrared);
82         mqtt.subscribe(stopikInfrared);
83     }
84
85     function MQTTconnect() {
86         document.getElementById("pesanSuhuHewan").innerHTML = "";
87         var s = defaultServer;
88         var p = defaultPort;
89
90         if (p != "") {
91             console.log("ports");
92             port = parseInt(p);
93             console.log("port" + port);
94         }

```

```

93         console.log("port" + port);
94     }
95     if (s != "") {
96         host = s;
97         console.log("host");
98     }
99
100     console.log("connecting to " + host + " " + port);
101     var x = Math.floor(Math.random() * 10000);
102     var cname = "orderform-" + x;
103     mqtt = new Paho.MQTT.Client(host, port, cname);
104
105     var options = {
106         timeout: 3,
107         onSuccess: onConnect,
108         onFailure: onFailure,
109     };
110
111     mqtt.onConnectionLost = onConnectionLost;
112     mqtt.onMessageArrived = onMessageArrived;
113
114     mqtt.connect(options);
115     return false;
116 }
117 </script>
118
119 </head>

```

```

121 <body>
122     <!-- tinggal include layouts . navbarnya -->
123     <div class="dashboard">
124         <div class="container pb-10">
125             <div class="flex justify-between">
126                 <div>
127                     <h2>Hello, xxx!</h2>
128                     <h3>Today is : {{today()->format('D, d M Y')}}</h3>
129                 </div>
130                 <div>
131                     <a href="{{route('dashboard')}}">
132                         
133                     </a>
134                 </div>
135             </div>
136
137             <hr style="margin-bottom: 0">
138
139             <div class="flex flex-row">
140                 <div class="w-1/2 flex-col id="dashboard-col">
141                     <h1 class="text-center font-semibold">Dashboard</h1>
142
143                     <div class="grid grid-cols-2 gap-4">
144                         <div class="box-primary flex">
145                             <div class="flex align-center">
146                                 <h4 class="font-bold box-text">Suhu Hewan</h4>
147                             </div>
148                             <div class="box-line"></div>
149                             <div class="flex align-center justify-center">
150                                 <h3 id="suhu-hewan" class="font-semibold box-text">
151                                     <p id="pesanSuhuHewan"></p>

```

```

154                                     </p>
155                                 </h3>
156                             </div>
157                         <div class="box-primary flex">
158                             <div class="flex align-center">
159                                 <h4 class="font-semibold box-text">Suhu Ruangan</h4>
160                             </div>
161                             <div class="box-line"></div>
162                             <div class="flex align-center justify-center">
163                                 <h3 class="font-semibold box-text">
164                                     <p id="pesanSuhuLingkungan"></p>
165                                 </h3>
166                             </div>
167                         </div>
168                     </div>
169                     <!-- ===== -->
170                     <div class="grid grid-cols-2 gap-4 mt-5">
171                         <div class="box-primary flex">
172                             <div class="flex align-center">
173                                 <h4 class="font-semibold box-text">Kondisi Makanan</h4>
174                             </div>
175                             <div class="box-line"></div>
176                             <div class="flex align-center justify-center">
177                                 <h3 id="suhu-hewan" class="font-semibold box-text">
178                                     <p id="pesanInfrared"></p>
179                                 </h3>
180                             </div>
181                         </div>
182                         <div class="box-primary flex">
183                             <div class="flex align-center">
184                                 <h4 class="font-semibold box-text">Waktu Terakhir</h4>
185                             </div>
186                             <div class="box-line"></div>

```

```

resources > views > layouts > dashboard.blade.php > html > head > script > onMessageArrived
183     </div>
184     <div class="box-line"></div>
185     <div class="flex align-center justify-center">
186         <h3 class="font-semibold box-text">
187             @forelse($lastDay as $apiData)
188                 <h4 class="font-semibold box-text">{{ $apiData['ts'] }}
189             </h3>
190             @empty
191                 <h3 class="font-semibold box-text">Tidak ada Data Terkini</h3>
192             @endforelse
193         </h3>
194     </div>
195 </div>
196 </div>
197
198 <div class="box-graph mt-5">
199     <h2 class="font-semibold text-center">Grafik Rata-rata Suhu</h2>
200     <div id="canvasContainer">
201         <canvas id="chartRataSuhu"></canvas>
202     </div>
203 </div>
204
205 </div>
206
207 <div class="line mx-6"></div>
208
209 <div class="w-1/2 flex-col id="menu-col">
210     @yield('content')
211 </div>
212
213 </div>

```

Penjelasan : Ini merupakan file main dari website ini, yield content akan berubah tergantung button menu mana yang akan di klik.. pada halaman ini juga akan menampilkan data suhu hewan, suhu lingkungan, dan sensor infrared.

Index.blade.php :

```
1 @extends('layouts.dashboard', ['title' => 'Dashboard'])
2
3 @section('content')
4 <h1 class="text-center font-semibold">Menu Control</h1>
5
6 <div class="grid grid-cols-2 gap-4">
7     <div class="box-primary flex align-center justify-center">
8         <a href="{{route('dashboard.manual-feeding')}}" class="decoration-none text-dark font-semibold text-xl">Manual
9         Feeding</a>
10    </div>
11    <div class="box-primary flex align-center justify-center">
12        <a href="" class="decoration-none text-dark font-semibold text-xl">Schedule
13        </a>
14    </div>
15 </div>
16 <div class="grid grid-cols-2 gap-4 mt-5">
17     <div class="box-primary flex align-center justify-center">
18         <a href="{{route('dashboard.data-history')}}" class="decoration-none text-dark font-semibold text-xl">Data
19         History</a>
20    </div>
21    <div class="box-primary flex align-center justify-center">
22        <a href="" class="decoration-none text-dark font-semibold text-xl">Profile</a>
23    </div>
24 </div>
25 @endsection
```

Penjelasan : mengisikan yield content default, yang jikalau salah satu menu diatas di klik, akan menggantikan settingan defaultnya

Data-history.blade.php :

```
1 @extends('layouts.dashboard', ['title' => 'Data History'])
2
3 @section('content')
4 <h1 class="text-center font-semibold">Menu: Data History</h1>
5
6 <div class="menu-data-history-box flex flex-col align-center">
7     <table class="table mt-10">
8         <thead>
9             <tr>
10                 <th>Data Pemberian Makan(API)</th>
11                 <!-- <th>Data Pemberian Makan(Database)</th> -->
12             </tr>
13         </thead>
14         <tbody>
15             @forelse($apiTimeStamp as $apiData)
16                 <tr class="text-center">
17                     <td>{{ $apiData['ts'] }}</td>
18                 </tr>
19                 @empty
20                     <tr>
21                         <td colspan="2">
22                             <h4 class="font-semibold text-center">Tidak ada data</h4>
23                         </td>
24                     </tr>
25                 @endforelse
26             </tbody>
27         </table>
28     </div>
29 @endsection
```

Penjelasan : menampilkan data history, yang memanggil variabel fungsi di controller

## Manual-feeding.blade.php :

```
resources > views > menu > manual-feeding.blade.php > script > ready() callback
1  @extends('layouts.dashboard', ['title' => 'Manual Feeding'])
2
3  @section('content')
4  <h1 class="text-center font-semibold">Menu: Manual</h1>
5
6  <div class="menu-manual-box flex flex-col justify-center align-center">
7      <a href="#" id="btnBukaServo" class="decoration-none text-dark font-semibold text-xl">
8          
9      </a>
10
11     <h2 class="font-semibold mt-10">Tap the button to feed</h2>
12 </div>
13
14 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
15 <script>
16     $(document).ready(function() {
17         $('#btnBukaServo').on('click', function(event) {
18             event.preventDefault(); // Mencegah navigasi default saat tombol gambar diklik
19
20             // Memanggil rute baru untuk bukaServo dari controller
21             $.ajax({
22                 type: 'GET',
23                 url: '{{ route("dashboard.manual-feeding.buka-servo") }}',
24                 success: function(response) {
25                     // Handle respons atau tampilkan pesan sukses jika diperlukan
26                     console.log(response);
27
28                     // Tampilkan pesan di halaman atau lakukan aksi lainnya
29                     alert('Feeding success!');
30                 },
31                 error: function(error) {
32                     // Handle kesalahan atau tampilkan pesan kesalahan jika diperlukan
33                 }
34             });
35         });
36     });
37 </script>
```

Penjelasan : Ketika button image di klik, maka akan memanggil route fungsi buka servo

## DashboardController.php :

```
15 class DashboardController extends Controller
16 {
17     public $mqttData;
18
19     public function __invoke()
20     {
21         $this->subscribeMQTT();
22         // $chart = $this->setDashboardGraph();
23         // $suhu = $this->setDashboardData();
24         $lastDay = $this->getLastDay();
25
26         return view('dashboard.index', compact('lastDay'));
27     }
28
29     public function subscribeMQTT()
30     {
31         $host = 'broker.hivemq.com';
32         $port = 1883;
33         $clientId = 'cat_feeder';
34         $topic = 'iot-kelompok2-suhuHewan';
35
36         $mqtt = new MqttClient($host, $port, $clientId);
37
38         try {
39             $mqtt->connect();
40
41             if ($mqtt->isConnected()) {
42                 $mqtt->subscribe($topic, function ($topic, $message) {
43                     $this->mqttData = $message; // Simpan data ke variabel mqttData
44                 });
45             }
46         } catch (\Exception $e) {
47             // Handle error
48         }
49     }
50 }
```

```

42         $mqtt->subscribe($topic, function ($topic, $message) {
43             $this->mqttData = $message; // Simpan data ke variabel mqttData
44             echo ("Received message on topic '$topic': $message");
45         });
46
47         // Tahan koneksi sejenak untuk mendengarkan pesan
48         sleep(2);
49
50         $mqtt->disconnect();
51         // dd($this->mqttData);
52     } else {
53         // Log pesan kesalahan koneksi MQTT
54         echo ('Failed to connect to MQTT broker.');
```

Penjelasan : digunakan untuk setting get API juga di dashboard

## dataHistoryController.php

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Data;
6  use LaravelDaily\LaravelCharts\Classes\LaravelChart;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Http;
9
10 class DataHistoryController extends Controller
11 {
12     public function __invoke()
13     {
14         $apiTimeStamp = $this->getTSAPI();
15         $apiManyInput = $this->getJumlahInputAPI();
16         $lastDay = $this->getLastDay();
17         // $data = $this->getDataHistory();
18         $dataFromDatabase = $this->getDataHistory();
19         // $chart = $this->setDashboardGraph();
20         $suhu = $this->setDashboardData();
21
22         // return view('menu.data-history', compact('dataFromApi', 'data', 'chart', 'suhu'));
23         return view('menu.data-history', compact('apiManyInput', 'lastDay', 'apiTimeStamp', 'dataFromDatabase', 'chart', 'suhu'));
24     }
25
26     protected function getDataHistory()
27     {
28         return Data::query()
29             ->select('suhu_data', 'ultra_data', 'created_at')
30             ->orderBy('created_at', 'desc')
31             ->take(25)
32             ->get();
33     }
34 }
```



```

35
36
37     protected function getTSAPI()
38     {
39         $response = Http::get('https://kelompokd2iot.000webhostapp.com/petfeeder/api/getlast10days.php');
40
41         if ($response->successful()) {
42             return $response->json();
43         }
44
45         return [];
46     }
47
48     protected function getJumlahInputAPI()
49     {
50         $response = Http::get('https://kelompokd2iot.000webhostapp.com/petfeeder/api/getlast5days.php');
51
52         if ($response->successful()) {
53             return $response->json();
54         }
55
56         return [];
57     }
58
59     protected function setDashboardData()
60     {
61         return Data::query()
62             ->select('suhu_data', 'ultra_data', 'created_at')
63             ->orderBy('created_at', 'desc')
64             ->first();
65     }
66
67     protected function getLastDay()
68     {
69         $response = Http::get('https://kelompokd2iot.000webhostapp.com/petfeeder/api/getlastday.php');
70
71         if ($response->successful()) {
72             return $response->json();
73         }
74
75         return [];
76     }
77 }
78

```

Penjelasan : menerima json dari API, lalu di inisialisasi di fungsi invoke

ManualFeedingController.php

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Data;
6  use LaravelDaily\LaravelCharts\Classes\LaravelChart;
7  use PhpMqtt\Client\MqttClient;
8  use PhpMqtt\Client\ConnectionSettings;
9  use Illuminate\Support\Facades\Http;
10
11 class ManualFeedingController extends Controller
12 {
13     public function index()
14     {
15         $chart = $this->setDashboardGraph();
16         $suhu = $this->setDashboardData();
17         $lastDay = $this->getLastDay();
18
19         // $this->bukaServo();
20         // ternyata doubling nya dari sini.. kan index itu otomatis ya saat load, menu.manual-feeding
21         // nah itu doubling saat refreshnya..
22         // ini saya tahunya saat di menu default, ketika saya mau pindah ke Menu-Manual itu dia malah send data
23         // ternyata dari sini
24
25         return view('menu.manual-feeding', compact('chart', 'suhu', 'lastDay'));
26     }
27
28     protected function bukaServo()
29     {
30         $host = 'broker.hivemq.com';
31         $port = 1883;
32         $clientId = 'cat-feeder';
33         $message = '1';
34
35         $mqtt = new MqttClient($host, $port, $clientId);
36
37         try {
38             $mqtt->connect();
39
40             if ($mqtt->isConnected()) {
41                 $mqtt->publish($topic, $message, 0);
42                 $mqtt->disconnect();
43             } else {
44                 // Log pesan kesalahan koneksi MQTT
45                 echo('Failed to connect to MQTT broker.');

```

Penjelasan : mengkonfigurasi mqtt di controller, mulai dari nama client, broker, port, topik, dan pesan. Lalu mengkoneksikannya dan publish data ke broker

Web.php

```

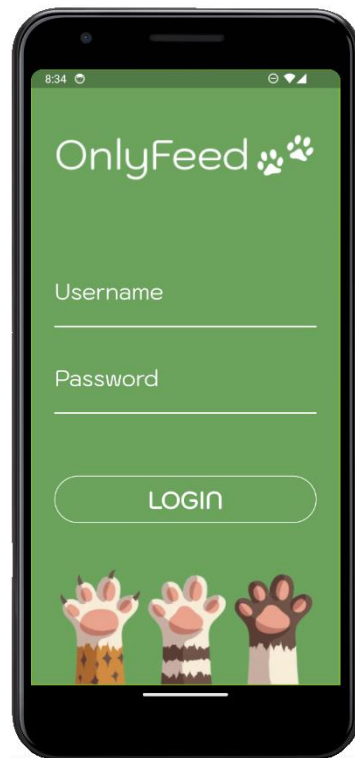
23 //Route di bawah akan mereturn view welcome, dan route di bawah memiliki name home
24 Route::get('/', function () {
25     return view('loading');
26 });
27
28
29 //Route di bawah akan meng-grouping route-route dengan middleware auth dan prefix 'dashboard' pada URL
30 Route::group(['prefix' => 'dashboard'], function () {
31     // Route di bawah akan menampilkan halaman awal dashboard
32     Route::get('/halaman', DashboardController::class->name('dashboard'));
33
34     // Route di bawah akan meng-group route-route yang menggunakan controller ManualFeedingController
35     Route::controller(ManualFeedingController::class)->group(function () {
36         // Route di bawah akan memanggil function index pada controller
37         Route::get('manual-feeding', 'index')->name('dashboard.manual-feeding');
38
39         Route::get('/dashboard/manual-feeding/buka-servo', [ManualFeedingController::class, 'bukaServo'])
40             ->name('dashboard.manual-feeding.buka-servo');
41     });
42
43     // Route di bawah akan memanggil function __invoke pada DataHistoryController
44     Route::get('data-history', DataHistoryController::class->name('dashboard.data-history'));
45 });

```

Penjelasan : digunakan untuk alamat dan pemanggilan controller sesuai dengan kebutuhan.

## 8. Mobile

- Halaman Awal



Ini merupakan tampilan awal dari aplikasi, dimana terdapat dua textfield yaitu username dan password. Selain itu terdapat button yang ketika ditekan akan mengarahkan ke halaman utama.

Code:

```

1. import 'package:flutter/material.dart';
2. import 'package:Onlyfeed/navigation.dart';
3.
4. class HomePage extends StatelessWidget {
5.     HomePage({super.key});
6.
7.     final TextEditingController username = TextEditingController();
8.     final TextEditingController pass = TextEditingController();
9.
10.    @override

```

```

11. Widget build(BuildContext context) {
12.   return Scaffold(
13.     resizeToAvoidBottomInset: false,
14.     backgroundColor: Color(0xFF6BA35D),
15.     body: SafeArea(
16.       child: Padding(
17.         padding: EdgeInsets.symmetric(horizontal: 30, vertical: 0),
18.         child: Column(
19.           crossAxisAlignment: CrossAxisAlignment.center,
20.           children: <Widget>[
21.             SizedBox(height: 50),
22.             Container(
23.               child: Image.asset('assets/images/title.png'),
24.             ),
25.             SizedBox(height: 100),
26.             TextFormField(
27.               controller: username,
28.               decoration: InputDecoration(
29.                 labelText: "Username",
30.                 labelStyle: TextStyle(
31.                   color: Colors.white,
32.                 ),
33.               ),
34.               style: TextStyle(
35.                 color: Colors.white,
36.                 fontSize: 28,
37.               ),
38.             ),
39.             SizedBox(height: 25),
40.             TextFormField(
41.               controller: pass,
42.               decoration: InputDecoration(
43.                 labelText: "Password",
44.               ),
45.               style: TextStyle(
46.                 color: Colors.white,
47.                 fontSize: 28,
48.               ),
49.               obscureText: true,
50.             ),
51.             SizedBox(height: 80),
52.             Column(
53.               children: <Widget>[
54.                 MaterialButton(
55.                   minWidth: double.infinity,
56.                   height: 60,
57.                   onPressed: () {
58.                     Navigator.pushReplacement(
59.                       context,
60.                       MaterialPageRoute(
61.                         builder: (context) => NavigationPage(),
62.                       ),
63.                     );
64.                   },
65.                   shape: RoundedRectangleBorder(
66.                     side: BorderSide(color: Colors.white),
67.                     borderRadius: BorderRadius.circular(50),
68.                   ),
69.                   child: Text(
70.                     "LOGIN",
71.                     style: TextStyle(
72.                       fontSize: 30,
73.                       color: Colors.white,
74.                       fontWeight: FontWeight.bold,
75.                     ),
76.                   ),
77.                 ],
78.             ),
79.             Spacer(),
80.             Container(
81.               width: MediaQuery.of(context).size.width,
82.               child: Image.asset('assets/images/claw3.png'),
83.             ),
84.           ],
85.         ),
86.       ),

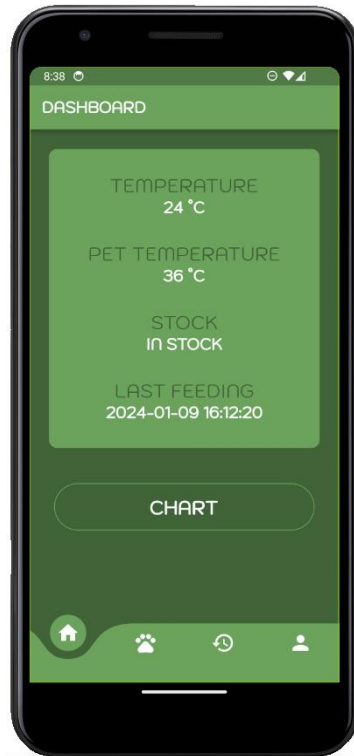
```

```

87.         ),
88.     ),
89. );
90. }
91. }
92.

```

- Halaman Utama



Pada halaman utama menampilkan data dari sensor. Pertama ada data suhu lingkungan, suhu hewan, dan data stok yang didapat dari MQTT. Selain itu terdapat informasi mengenai kapan pemberian makan terakhir yang diperoleh dari API. Berikutnya terdapat button "CHART" yang bila diklik akan mengarahkan ke halaman grafik.

Code:

```

1. import 'dart:convert';
2. import 'package:flutter/material.dart';
3. import 'package:http/http.dart' as http;
4. import 'package:mqtt_client/mqtt_client.dart';
5. import 'package:mqtt_client/mqtt_server_client.dart';
6. import 'package:Onlyfeed/chart.dart';
7.
8. class MqttManager {
9.   MqttServerClient? _client;
10.
11.   MqttManager() {
12.     _client = MqttServerClient('broker.hivemq.com', 'flutter_subscriber');
13.     _client?.port = 1883;
14.     _client?.logging(on: false);
15.     _client?.keepAlivePeriod = 60;
16.     _client?.onDisconnected = _onDisconnected;
17.     _client?.onSubscribed = _onSubscribed;
18.   }
19.
20.   Future<void> connect() async {
21.     try {
22.       await _client?.connect();
23.       print('Connected to MQTT');
24.       _subscribeToTopic('catfeeder-iot-klmpk2-datasuhilingkungan');
25.       _subscribeToTopic('catfeeder-iot-klmpk2-datasuhewan');
26.       _subscribeToTopic('catfeeder-iot-klmpk2-datainfrared');
27.     } catch (e) {
28.       print('Error connecting to MQTT: $e');

```

```

29.     }
30. }
31.
32. void _onDisconnected() {
33.     print('Disconnected from MQTT');
34. }
35.
36. void _onSubscribed(String topic) {
37.     print('Subscribed to topic: $topic');
38. }
39.
40. void _subscribeToTopic(String topic) {
41.     _client?.subscribe(topic, MqttQos.atMostOnce);
42. }
43.
44. void disconnect() {
45.     _client?.disconnect();
46. }
47. }
48.
49. class MainPageDB extends StatefulWidget {
50.     const MainPageDB({Key? key}) : super(key: key);
51.
52.     @override
53.     _MainPageDBState createState() => _MainPageDBState();
54. }
55.
56. class _MainPageDBState extends State<MainPageDB> {
57.     List<Map<String, dynamic>> temperatureData = [];
58.     String lastFeedingTimestamp = '';
59.     String currentTemperature = '';
60.     String currentPetTemperature = '';
61.     String currentStockStatus = '';
62.
63.     MqttManager mqttManager = MqttManager();
64.
65.     @override
66.     void initState() {
67.         super.initState();
68.         fetchData();
69.         mqttManager.connect();
70.         _subscribeToMqttMessages();
71.     }
72.
73.     void _subscribeToMqttMessages() {
74.         mqttManager._client?.updates
75.             ?.listen((List<MqttReceivedMessage<MqttMessage>> c) {
76.                 final MqttPublishMessage message = c[0].payload as MqttPublishMessage;
77.                 final String payload =
78.                     MqttPublishPayload.bytesToStringAsString(message.payload.message);
79.                 if (c[0].topic == 'catfeeder-iot-klmpk2-datasuhulingkungan') {
80.                     setState(() {
81.                         currentTemperature = '$payload °C';
82.                     });
83.                 } else if (c[0].topic == 'catfeeder-iot-klmpk2-datasuhuhewan') {
84.                     setState(() {
85.                         currentPetTemperature = '$payload °C';
86.                     });
87.                 } else if (c[0].topic == 'catfeeder-iot-klmpk2-datainfrared') {
88.                     setState(() {
89.                         currentStockStatus = (payload == '1') ? 'IN STOCK' : 'OUT OF STOCK';
90.                     });
91.                 }
92.             });
93.     }
94.
95.     @override
96.     void dispose() {
97.         mqttManager.disconnect();
98.         super.dispose();
99.     }
100.
101.     Future<void> fetchData() async {
102.         final response = await http.get(Uri.parse(
103.             'https://kelompokd2iot.000webhostapp.com/petfeeder/api/getrataratasuhu.php'));
104.

```



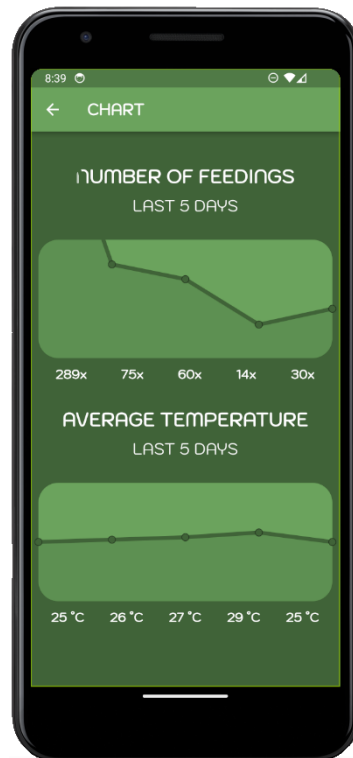
```

181.         side: BorderSide(color: Color(0xFF6BA35D)),
182.         borderRadius: BorderRadius.circular(50),
183.     ),
184.     child: Text(
185.         "CHART",
186.         style: TextStyle(
187.             fontSize: 24,
188.             color: Colors.white,
189.             fontWeight: FontWeight.bold,
190.         ),
191.     ),
192. ),
193. ],
194. ),
195. ],
196. ),
197. );
198. }
199.
200. Widget data() => Column(
201.     children: [
202.         SizedBox(height: 30),
203.         Text(
204.             'TEMPERATURE',
205.             style: TextStyle(
206.                 fontSize: 24,
207.                 color: Color(0xFF406338),
208.             ),
209.         ),
210.         Text(
211.             currentTemperature,
212.             style: TextStyle(
213.                 fontSize: 20, fontWeight: FontWeight.bold, color: Colors.white),
214.         ),
215.         SizedBox(height: 30),
216.         Text(
217.             'PET TEMPERATURE',
218.             style: TextStyle(
219.                 fontSize: 24,
220.                 color: Color(0xFF406338),
221.             ),
222.         ),
223.         Text(
224.             currentPetTemperature,
225.             style: TextStyle(
226.                 fontSize: 20, fontWeight: FontWeight.bold, color: Colors.white),
227.         ),
228.         SizedBox(height: 30),
229.         Text(
230.             'STOCK',
231.             style: TextStyle(
232.                 fontSize: 24,
233.                 color: Color(0xFF406338),
234.             ),
235.         ),
236.         Text(
237.             currentStockStatus,
238.             style: TextStyle(
239.                 fontSize: 20, fontWeight: FontWeight.bold, color: Colors.white),
240.         ),
241.         SizedBox(height: 30),
242.         Text(
243.             'LAST FEEDING',
244.             style: TextStyle(
245.                 fontSize: 24,
246.                 color: Color(0xFF406338),
247.             ),
248.         ),
249.         Text(
250.             lastFeedingTimestamp,
251.             style: TextStyle(
252.                 fontSize: 20, fontWeight: FontWeight.bold, color: Colors.white),
253.         ),
254.     ],
255. );
256. }

```



- Halaman Grafik



Pada halaman grafik menampilkan informasi mengenai jumlah pemberian makan dilima hari terakhir dan rata-rata suhu lingkungan pada lima hari terakhir. kedua informasi tadi didapatkan dari API.

Code:

```

1. import 'package:flutter/material.dart';
2. import 'package:fl_chart/fl_chart.dart';
3. import 'package:http/http.dart' as http;
4. import 'dart:convert';
5.
6. class ChartPage extends StatefulWidget {
7.   @override
8.   _ChartPageState createState() => _ChartPageState();
9. }
10.
11. class _ChartPageState extends State<ChartPage> {
12.   List<Map<String, dynamic>> temperatureData = [];
13.   List<Map<String, dynamic>> feedingData = [];
14.
15.   @override
16.   void initState() {
17.     super.initState();
18.     fetchDataTemp();
19.     fetchDataFeed();
20.   }
21.
22.   Future<void> fetchDataTemp() async {
23.     try {
24.       final response = await http.get(Uri.parse(
25.         'https://kelompokd2iot.000webhostapp.com/petfeeder/api/getrataratasuhu.php'));
26.
27.       if (response.statusCode == 200) {
28.         final List<Map<String, dynamic>> fetchedData =
29.           List<Map<String, dynamic>>.from(json.decode(response.body));
30.
31.         setState(() {
32.           temperatureData = fetchedData;
33.         });
34.       } else {
35.         print('Failed to load data. Status code: ${response.statusCode}');
36.       }

```

```

37.     } catch (e) {
38.         print('Error fetching data: $e');
39.     }
40. }
41.
42. Future<void> fetchDataFeed() async {
43.   try {
44.     final response = await http.get(Uri.parse(
45.       'https://kelompok2iot.000webhostapp.com/petfeeder/api/getlast5days.php'));
46.
47.     if (response.statusCode == 200) {
48.       final List<Map<String, dynamic>> fetchedData =
49.         List<Map<String, dynamic>>.from(json.decode(response.body));
50.
51.       setState(() {
52.         feedingData = fetchedData;
53.       });
54.     } else {
55.       print('Failed to load data. Status code: ${response.statusCode}');
56.     }
57.   } catch (e) {
58.     print('Error fetching data: $e');
59.   }
60. }
61.
62. @override
63. Widget build(BuildContext context) {
64.   return Scaffold(
65.     appBar: AppBar(
66.       title: Text(
67.         "CHART",
68.         style: TextStyle(
69.           fontWeight: FontWeight.bold,
70.         ),
71.       ),
72.       backgroundColor: Color(0xFF6BA35D),
73.     ),
74.     body: Padding(
75.       padding: EdgeInsets.all(10),
76.       child: Column(
77.         children: [
78.           SizedBox(height: 30),
79.           Text(
80.             'NUMBER OF FEEDINGS',
81.             style: TextStyle(
82.               fontSize: 24,
83.               fontWeight: FontWeight.bold,
84.               color: Colors.white),
85.           ),
86.           SizedBox(height: 10),
87.           Text(
88.             'LAST 5 DAYS',
89.             style: TextStyle(fontSize: 20, color: Colors.white),
90.           ),
91.           SizedBox(height: 30),
92.           Container(
93.             height: 150,
94.             decoration: BoxDecoration(
95.               borderRadius: BorderRadius.circular(25),
96.               color: Color(0xFF6BA35D),
97.             ),
98.             child: LineChart(
99.               LineChartData(
100.                 gridData: FlGridData(show: false),
101.                 titlesData: FlTitlesData(show: false),
102.                 borderData: FlBorderData(
103.                   show: false,
104.                 ),
105.                 minX: 0,
106.                 maxX: feedingData.length - 1.0,
107.                 minY: -20,
108.                 maxY: 100,
109.                 lineBarsData: [
110.                   LineChartBarData(
111.                     isCurved: false,
112.                     colors: [Color(0xFF406338)],

```

```

113.         dotData: FLDotData(show: true),
114.         barWidth: 5,
115.         belowBarData: BarAreaData(
116.             show: true,
117.             colors: [Color(0xFF406338)]
118.                 .map((color) => color.withOpacity(0.3))
119.                 .toList(),
120.         ),
121.         spots: List.generate(
122.             feedingData.length,
123.             (index) => FLSpot(
124.                 index.toDouble(),
125.                 double.parse(feedingData[index]["jumlah data"]),
126.             ),
127.         ),
128.     ),
129. ],
130. ),
131. ),
132. ),
133. SizedBox(height: 10),
134. Row(
135.     mainAxisAlignment: MainAxisAlignment.spaceAround,
136.     children: [
137.         for (var data in feedingData)
138.         Text(
139.             '${data["jumlah data"]}x',
140.             style: TextStyle(
141.                 color: Colors.white,
142.                 fontSize: 16,
143.                 fontWeight: FontWeight.bold),
144.         ),
145.     ],
146. ),
147. SizedBox(height: 30),
148. Text(
149.     'AVERAGE TEMPERATURE',
150.     style: TextStyle(
151.         fontSize: 24,
152.         fontWeight: FontWeight.bold,
153.         color: Colors.white),
154. ),
155. SizedBox(height: 10),
156. Text(
157.     'LAST 5 DAYS',
158.     style: TextStyle(fontSize: 20, color: Colors.white),
159. ),
160. SizedBox(height: 30),
161. Container(
162.     height: 150,
163.     decoration: BoxDecoration(
164.         borderRadius: BorderRadius.circular(25),
165.         color: Color(0xFF6BA35D),
166.     ),
167.     child: LineChart(
168.         LineChartData(
169.             gridData: FLGridData(show: false),
170.             titlesData: FLTitlesData(show: false),
171.             borderData: FLBorderData(
172.                 show: false,
173.             ),
174.             minX: 0,
175.             maxX: temperatureData.length - 1.0,
176.             minY: 0,
177.             maxY: 50,
178.             lineBarsData: [
179.                 LineChartBarData(
180.                     isCurved: false,
181.                     colors: [Color(0xFF406338)],
182.                     dotData: FLDotData(show: true),
183.                     barWidth: 5,
184.                     belowBarData: BarAreaData(
185.                         show: true,
186.                         colors: [Color(0xFF406338)]
187.                             .map((color) => color.withOpacity(0.3))
188.                             .toList(),

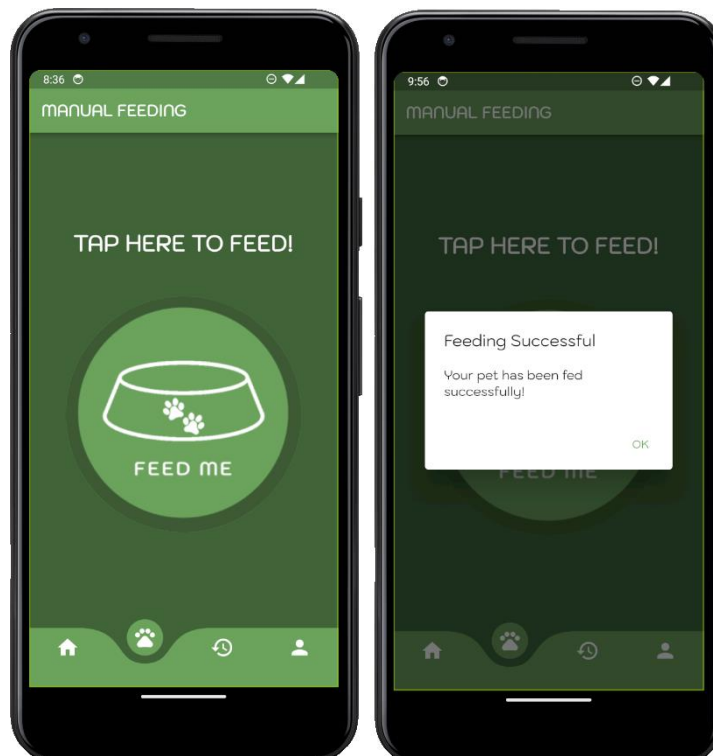
```

```

189.         ),
190.         spots: List.generate(
191.             temperatureData.length,
192.             (index) => F1Spot(
193.                 index.toDouble(),
194.                 double.parse(
195.                     temperatureData[index]["rata_rata_suhu"]),
196.             ),
197.         ),
198.     ],
199. ),
200. ),
201. ),
202. ),
203. SizedBox(height: 10),
204. Row(
205.     mainAxisAlignment: MainAxisAlignment.spaceAround,
206.     children: [
207.         for (var data in temperatureData)
208.         Text(
209.             '${data["rata_rata_suhu"]} °C',
210.             style: TextStyle(
211.                 color: Colors.white,
212.                 fontSize: 16,
213.                 fontWeight: FontWeight.bold),
214.         ),
215.     ],
216. ),
217. ),
218. ),
219. ),
220. );
221. }
222. }
223.

```

- Halaman Manual Feeding



Pada halaman ini terdapat button yang bila ditekan akan memunculkan pop-up pemberitahuan bahwa pemberian makan telah berhasil dan akan mengirim nilai ke MQTT untuk menggerakkan servo.

Code:

```

1. import 'package:flutter/material.dart';
2. import 'package:mqtt_client/mqtt_client.dart';

```

```

3. import 'package:mqtt_client/mqtt_server_client.dart';
4.
5. class MqttManager {
6.   MqttServerClient? _client;
7.
8.   MqttManager() {
9.     _client = MqttServerClient('broker.hivemq.com', 'flutter_publisher');
10.    _client?.port = 1883;
11.    _client?.logging(on: false);
12.    _client?.keepAlivePeriod = 60;
13.    _client?.onDisconnected = _onDisconnected;
14.  }
15.
16.   Future<void> connect() async {
17.     try {
18.       await _client?.connect();
19.       print('Connected to MQTT');
20.     } catch (e) {
21.       print('Error connecting to MQTT: $e');
22.     }
23.   }
24.
25.   void _onDisconnected() {
26.     print('Disconnected from MQTT');
27.   }
28.
29.   void publish(String topic, String message) {
30.     final MqttClientPayloadBuilder builder = MqttClientPayloadBuilder();
31.     builder.addString(message);
32.
33.     _client?.publishMessage(topic, MqttQos.exactlyOnce, builder.payload!);
34.   }
35.
36.   void disconnect() {
37.     _client?.disconnect();
38.   }
39. }
40.
41. class ManualPage extends StatelessWidget {
42.   const ManualPage({super.key});
43.
44.   @override
45.   Widget build(BuildContext context) {
46.     return Scaffold(
47.       appBar: AppBar(
48.         title: Text(
49.           "MANUAL FEEDING",
50.           style: TextStyle(
51.             fontWeight: FontWeight.bold,
52.           ),
53.         ),
54.         backgroundColor: Color(0xFF6BA35D),
55.       ),
56.       body: Center(
57.         child: Column(
58.           mainAxisAlignment: MainAxisAlignment.center,
59.           children: [
60.             Text(
61.               'TAP HERE TO FEED!',
62.               style: TextStyle(
63.                 fontSize: 28,
64.                 fontWeight: FontWeight.bold,
65.                 color: Colors.white),
66.             ),
67.             button(context), // Pass context to the button function
68.           ],
69.         ),
70.       ),
71.     );
72.   }
73.
74.   Widget button(BuildContext context) => Padding(
75.     padding: EdgeInsets.only(top: 50),
76.     child: InkWell(
77.       onTap: () async {
78.         MqttManager mqttManager = MqttManager();

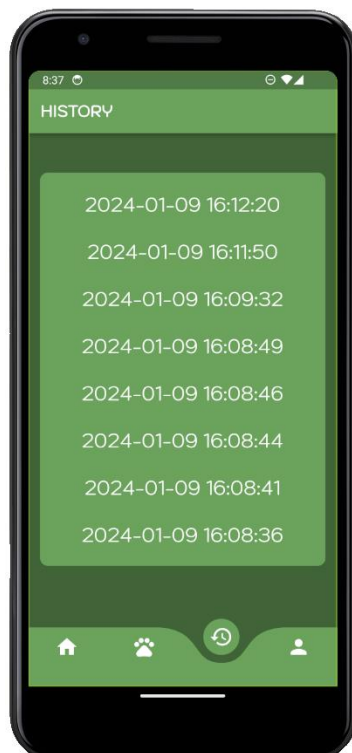
```

```

79.         await mqttManager.connect();
80.         mqttManager.publish('control_servo_iot_d2', '1');
81.         popup(context);
82.     },
83.     child: CircleAvatar(
84.         radius: 150,
85.         backgroundColor: Color(0xFF6BA35D),
86.         child: Image.asset(
87.             'assets/images/buttonfeed.png',
88.             width: 300,
89.             height: 300,
90.         ),
91.     ),
92. ),
93. );
94.
95. void popup(BuildContext context) {
96.     showDialog(
97.         context: context,
98.         builder: (BuildContext context) {
99.             return AlertDialog(
100.                title: Text("Feeding Successful"),
101.                content: Text("Your pet has been fed successfully!"),
102.                actions: [
103.                    TextButton(
104.                        onPressed: () {
105.                            Navigator.of(context).pop();
106.                        },
107.                        child: Text(
108.                            "OK",
109.                            style: TextStyle(
110.                                color: Color(0xFF6BA35D),
111.                            ),
112.                        ),
113.                    ),
114.                ],
115.            );
116.        },
117.    );
118. }
119. }
120.

```

- Halaman History



Pada halaman ini terdapat informasi mengenai history pemberian makan yang didapat dari API. Data yang ditampilkan adalah sepuluh data terakhir yang ada pada database.

Code:

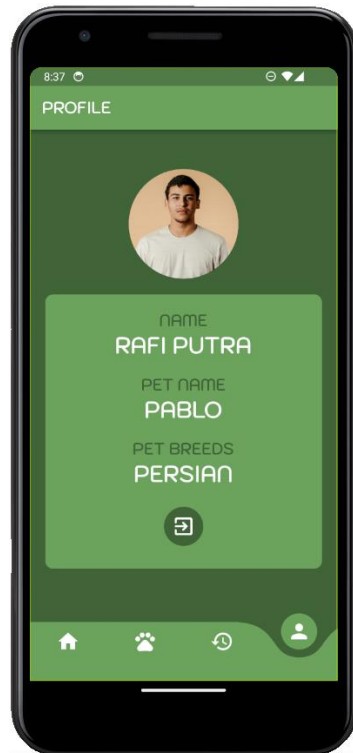
```
1. import 'package:flutter/material.dart';
2. import 'package:http/http.dart' as http;
3. import 'dart:convert';
4.
5. class HistoryPageDB extends StatefulWidget {
6.   const HistoryPageDB({super.key});
7.
8.   @override
9.   _HistoryPageDBState createState() => _HistoryPageDBState();
10. }
11.
12. class _HistoryPageDBState extends State<HistoryPageDB> {
13.   List<String> dateList = [];
14.
15.   @override
16.   void initState() {
17.     super.initState();
18.     fetchData();
19.   }
20.
21.   Future<void> fetchData() async {
22.     final response = await http.get(Uri.parse(
23.       'https://kelompok2iot.000webhostapp.com/petfeeder/api/getlast10days.php'));
24.
25.     if (response.statusCode == 200) {
26.       final List<dynamic> data = json.decode(response.body);
27.       setState(() {
28.         dateList = data.map((item) => item['ts'].toString()).toList();
29.       });
30.     } else {
31.       throw Exception('Failed to load data');
32.     }
33.   }
34.
35.   @override
36.   Widget build(BuildContext context) {
37.     return Scaffold(
38.       appBar: AppBar(
39.         title: Text(
40.           "HISTORY",
41.           style: TextStyle(
42.             fontWeight: FontWeight.bold,
43.           ),
44.         ),
45.         backgroundColor: Color(0xFF6BA35D),
46.       ),
47.       body: Padding(
48.         padding: EdgeInsets.only(left: 15, right: 15, top: 50),
49.         child: Container(
50.           height: 500,
51.           decoration: BoxDecoration(
52.             borderRadius: BorderRadius.circular(10),
53.             color: Color(0xFF6BA35D),
54.           ),
55.           child: ListView.builder(
56.             padding: EdgeInsets.only(top: 10),
57.             itemCount: dateList.length,
58.             itemExtent: 60,
59.             itemBuilder: (BuildContext context, int index) {
60.               return Center(
61.                 child: Text(
62.                   dateList[index],
63.                   style: TextStyle(fontSize: 25, color: Colors.white),
64.                 ),
65.               );
66.             },
67.           ),
68.         ),
69.       ),
    );
```

```

70.     );
71.   }
72. }
73.

```

- Halaman Profile



Pada halaman ini akan menampilkan informasi berupa nama pengguna, nama hewan, dan ras hewan. Selain itu terdapat button untuk kembali ke halaman awal.

Code:

```

1. import 'package:flutter/material.dart';
2. import 'package:Onlyfeed/homepage.dart';
3.
4. class ProfilePage extends StatelessWidget {
5.   const ProfilePage({super.key});
6.
7.   @override
8.   Widget build(BuildContext context) {
9.     return Scaffold(
10.      appBar: AppBar(
11.        title: Text(
12.          "PROFILE",
13.          style: TextStyle(
14.            fontWeight: FontWeight.bold,
15.          ),
16.        ),
17.        backgroundColor: Color(0xFF6BA35D),
18.      ),
19.      body: Profile(),
20.    );
21.  }
22. }
23.
24. class Profile extends StatefulWidget {
25.   const Profile({super.key});
26.
27.   @override
28.   State<Profile> createState() => _ProfileState();
29. }
30.
31. class _ProfileState extends State<Profile> {
32.   @override
33.   Widget build(BuildContext context) {
34.     return Scaffold(

```



```

35.         body: ListView(padding: EdgeInsets.zero, children: <Widget>[
36.             SizedBox(height: 50),
37.             profileImage(70,
38.                 'https://plus.unsplash.com/premium_photo-1689568126014-
06fea9d5d341?auto=format&fit=crop&q=60&w=500&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWZyY2h8ODF8fHB5b2ZpbGU1MjBwaWN0dXJ1fGVufDB8fDB8fHww'),
39.             SizedBox(height: 0),
40.             Container(
41.                 child: Column(
42.                     crossAxisAlignment: CrossAxisAlignment.start,
43.                     children: [
44.                         Padding(
45.                             padding: const EdgeInsets.all(20),
46.                             child: Container(
47.                                 height: 350,
48.                                 width: MediaQuery.of(context).size.width,
49.                                 decoration: BoxDecoration(
50.                                     color: Color(0xFF6BA35D),
51.                                     borderRadius: BorderRadius.circular(10),
52.                                 ),
53.                                 child: Column(
54.                                     crossAxisAlignment: CrossAxisAlignment.center,
55.                                     children: [
56.                                         description(),
57.                                     ],
58.                                 ),
59.                             )],
60.                         ),
61.                     ),
62.                 ]));
63.     });
64. }
65.
66. Widget profileImage(double sizeRadius, String linkImage) => CircleAvatar(
67.     radius: sizeRadius,
68.     backgroundColor: Colors.black,
69.     child: ClipOval(
70.         child: Image.network(
71.             linkImage,
72.             fit: BoxFit.cover,
73.             width: sizeRadius * 2,
74.             height: sizeRadius * 2,
75.         ),
76.     ),
77. );
78.
79. Widget description() => Column(
80.     children: [
81.         SizedBox(height: 20),
82.         Text(
83.             'NAME',
84.             style: TextStyle(
85.                 fontSize: 20,
86.                 color: Color(0xFF406338),
87.                 fontWeight: FontWeight.bold,
88.             ),
89.         ),
90.         Text(
91.             'RAFI PUTRA',
92.             style: TextStyle(
93.                 fontSize: 28, fontWeight: FontWeight.bold, color: Colors.white),
94.         ),
95.         SizedBox(height: 20),
96.         Text(
97.             'PET NAME',
98.             style: TextStyle(
99.                 fontSize: 20,
100.                 color: Color(0xFF406338),
101.                 fontWeight: FontWeight.bold,
102.             ),
103.         ),
104.         Text(
105.             'PABLO',
106.             style: TextStyle(
107.                 fontSize: 28, fontWeight: FontWeight.bold, color: Colors.white),
108.         ),

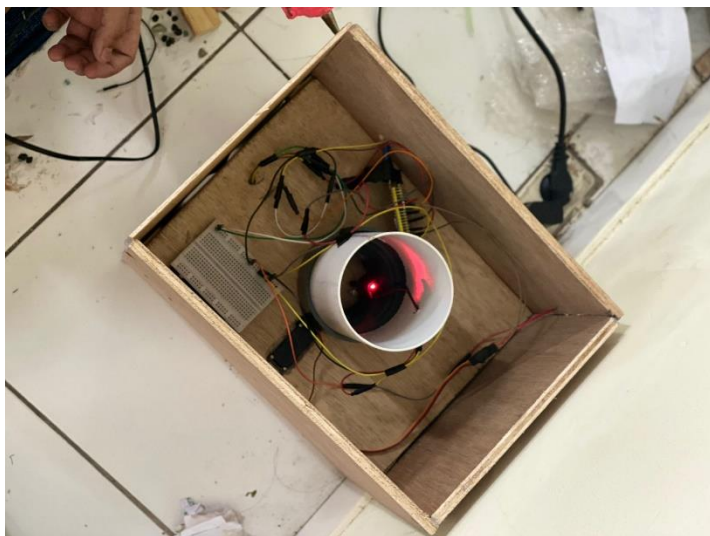
```

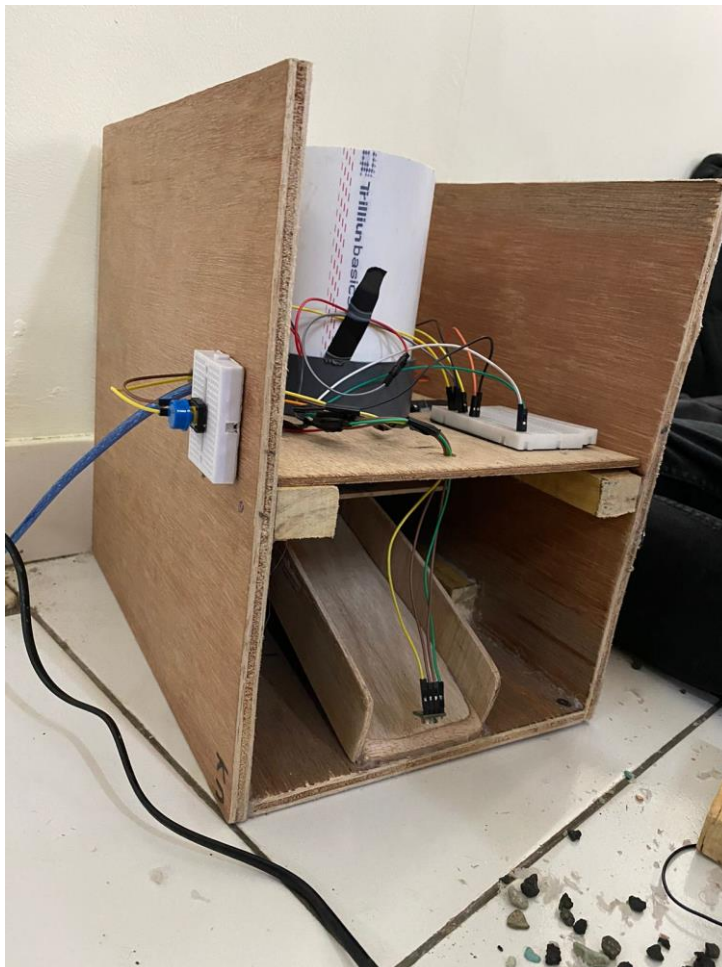
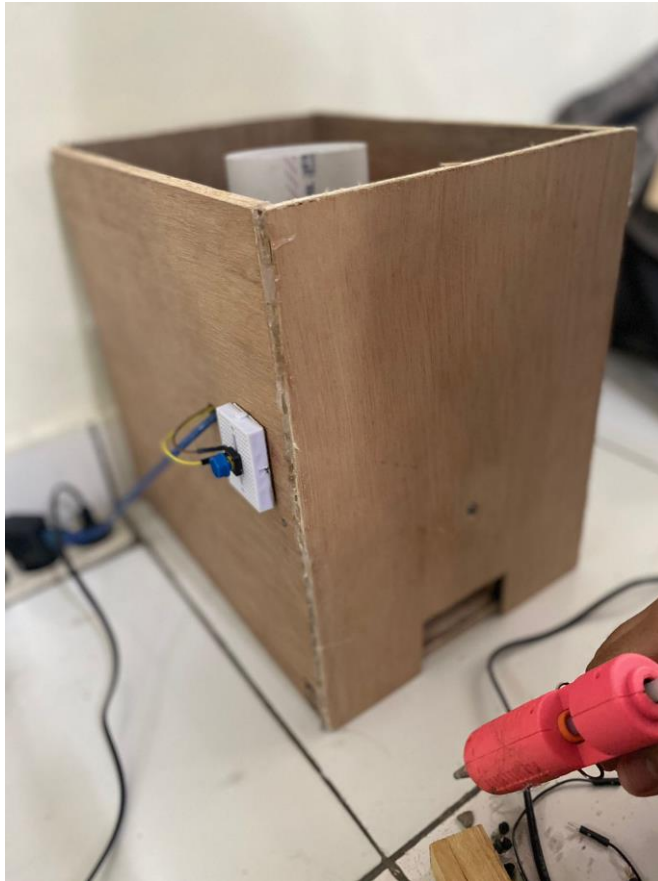
```

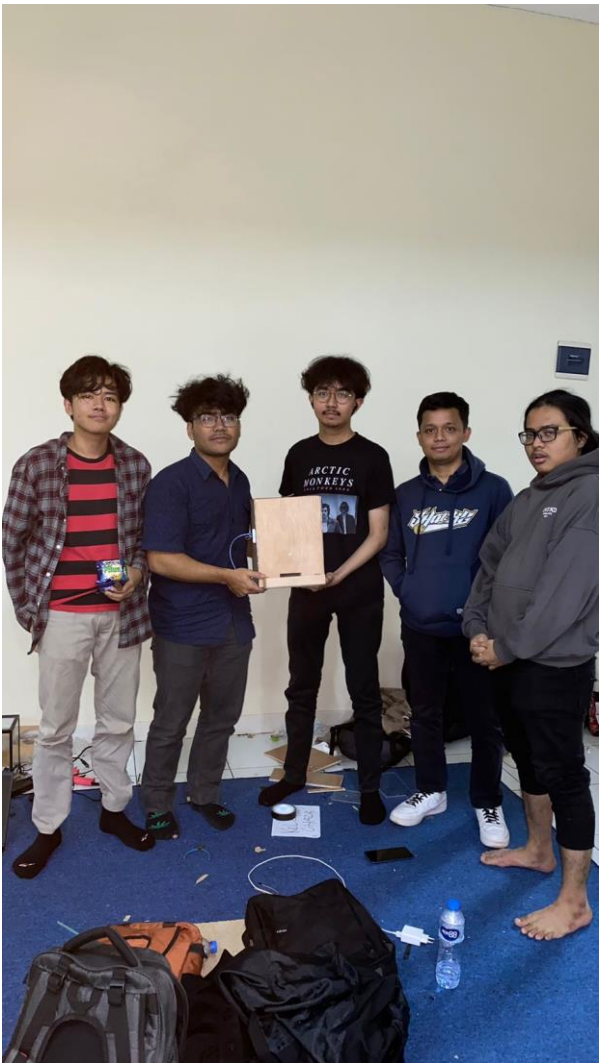
109.         SizedBox(height: 20),
110.         Text(
111.           'PET BREEDS',
112.           style: TextStyle(
113.             fontSize: 20,
114.             color: Color(0xFF406338),
115.             fontWeight: FontWeight.bold,
116.           ),
117.         ),
118.         Text(
119.           'PERSIAN',
120.           style: TextStyle(
121.             fontSize: 28, fontWeight: FontWeight.bold, color: Colors.white),
122.         ),
123.         SizedBox(height: 25),
124.         Center(
125.           child: InkWell(
126.             onTap: () {
127.               Navigator.pushReplacement(
128.                 context,
129.                 MaterialPageRoute(
130.                   builder: (context) => HomePage(),
131.                 ),
132.             );
133.           ),
134.           child: Container(
135.             width: 50,
136.             height: 50,
137.             decoration: BoxDecoration(
138.               color: Color(0xFF406338),
139.               shape: BoxShape.circle,
140.             ),
141.             child: Icon(
142.               Icons.exit_to_app_rounded,
143.               color: Colors.white,
144.               size: 30,
145.             ),
146.           ),
147.         ),
148.       ],
149.     );
150.   );
151. }
152.

```

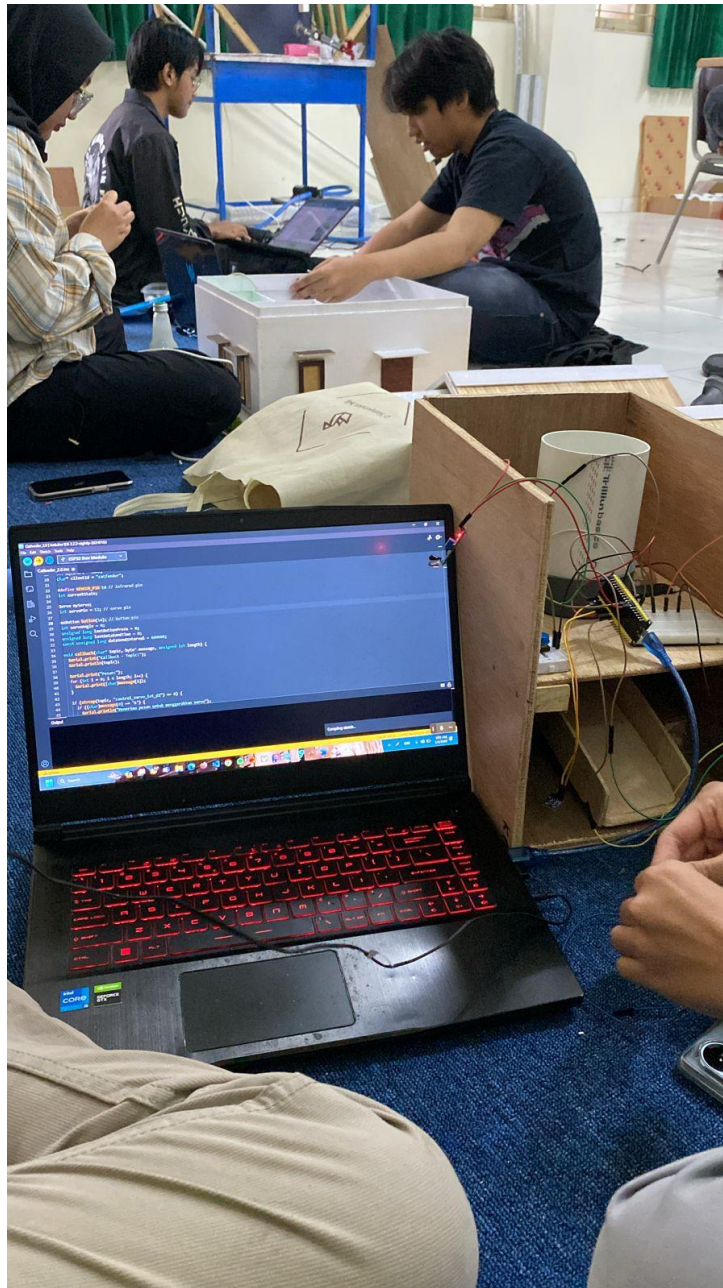
## 9. Dokumentasi Foto











## 10. Kendala yang dihadapi

Beberapa kendala yang kami hadapi selama mengerjakan alat dan implementasi di sisi user :

1. Kabel jumper yang mudah terlepas
2. Alat yang selalu tidak konsisten

## 11. Link Youtube

<https://youtu.be/7coNr8MJ5a0>