

BAB IV HASIL DAN PEMBAHASAN

4.1 Temuan Penelitian

Tahap ini bertujuan untuk mengetahui tahapan penerapan alat apa saja yang digunakan pada *prototype Automatic Cat Feeding*. Tahapan ini dilakukan setelah pembahasan mengenai Analisis dan Perancangan selesai dilakukan. Kemudian Perancangan pada bab sebelumnya diterapkan ke dalam sebuah sistem sebagai berikut:

4.1.1 Lingkungan Implementasi

Pada proses implementasi ini, dibutuhkan *Software* dan *Hardware* penunjang untuk membangun dan menjalankan sebuah sistem. Berikut merupakan penjelasan dari *Software* dan *Hardware* yang menjadi penunjang pada implementasi penelitian ini:

a. Software

Software yang menjadi penunjang dalam membangun sebuah Sistem *Automatic Cat Feeding* adalah sebagai berikut:

1. Sistem Operasi Windows 10
2. Arduino IDE versi 1.8.19
3. Telegram for Android versi 8.8.6
4. Balsamiq Wireframes versi 4.1.2

b. Hardware

Hardware yang menjadi penunjang dalam membangun sebuah Sistem *Automatic Cat Feeding* adalah sebagai berikut:

- i. Laptop
 - a. Processor AMD A8
 - b. Mouse
 - c. Keyboard
 - d. RAM 8 GB
- ii. Smartphone
 - a. OPPO A53 2020
 - b. RAM 6 GB

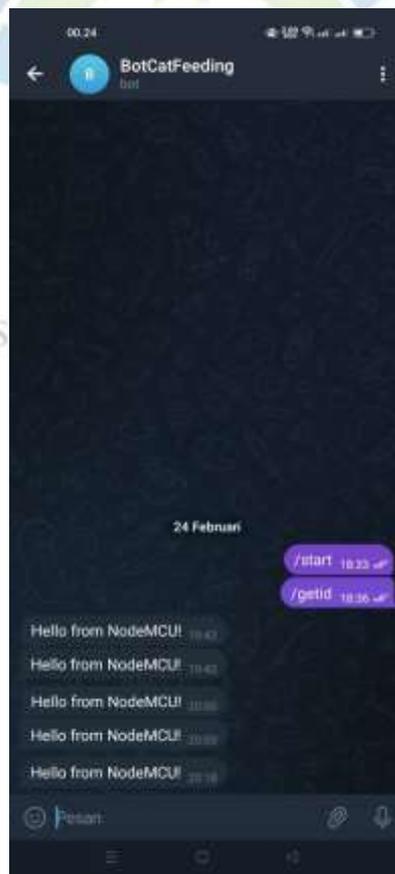
- c. Internal 64 GB
- iii. Box Toolkit Arduino
 - a. Arduino Uno
 - b. Load Cell
 - c. RTC
 - d. Servo
 - e. Modul Wifi NodeMCU ESP8266

4.1.2 Implementasi Antarmuka (*Interface*)

Proses implementasi Antarmuka ini merupakan tahap mengubah sebuah rancangan desain menjadi sebuah bentuk aplikasi. Aplikasi yang digunakan yaitu Telegram Messenger yang nantinya Telegram Messenger ini digunakan sebagai media Kontrol Sistem.

a. Implementasi Tampilan Halaman Utama

Gambar 4.1 merupakan halaman utama yang ditampilkan oleh Telegram kepada user.



Gambar 4. 1 halaman utama

Gambar di atas merupakan *Interface* dari bot Telegram untuk Halaman Utama yang dapat digunakan oleh *user*. Pada tampilan diatas *user* dapat menginputkan berat dari kucing dan sistem akan memeriksa waktu sebelum membuka servo.

b. Implementasi Pembuatan Bot Telegram

Berikut Gambar 4.2 merupakan *code* untuk membuat bot Telegram yang akan disambungkan ke sistem sebagai media Kontrol Sistem yang dapat digunakan oleh Sistem.

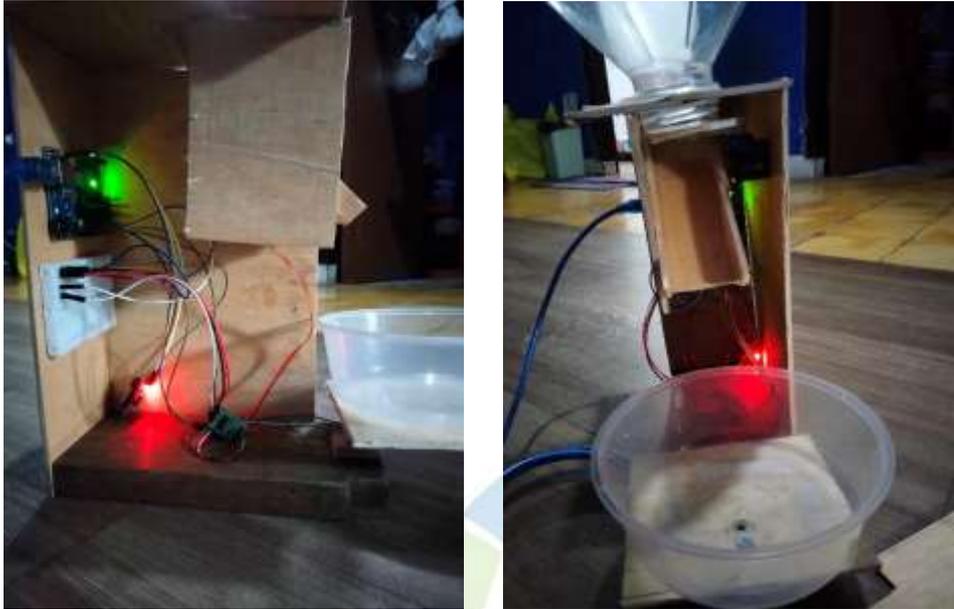


Gambar 4. 2 Konfigurasi Bot Telegram

4.1.3 Implentasi Prototype

Proses implementasi prototype ini merupakan tahap mengubah rancangan desain dari sebuah sistem menjadi bentuk rangkaian sistem secara nyata. Rangkaian sistem ini kemudian akan digunakan oleh user sebagai sistem nyata dari sistem yang dibangun pada penelitian ini.

Berikut merupakan rangkaian sistem atau prototype dari sistem yang dibangun pada gambar 4.3:



Gambar 4. 3 Tampilan Prototype Sistem

4.1.4 Implementasi Proses

Pada tahap implementasi proses ini, disesuaikan kepada aturan-aturan yang dilakukan pada proses rancangan sistem. Implementasi proses ini untuk menyesuaikan antara data yang dimasukan/dideteksi (*input*) dan hasil data yang diproses dan dikeluarkan (*output*) yang telah diproses oleh sistem.

Proses yang dilakukan pada penelitian Rancang Bangun Sistem *Automatic Cat Feeding* ini dilakukan dengan urutan proses dimulai dari pembacaan berapa berat kucing yang diinputkan kemudian data yang dideteksi dikirimkan kepada mikrokontroler untuk diproses berapa banyak pakan yang dibutuhkan sesuai dengan berat kucing yang diinputkan. Lalu sistem akan membaca waktu jika sudah sesuai maka servo akan membuka. Selanjutnya sensor berat akan membaca berapa berat pakan yang dikeluarkan.

Berikut merupakan *source code* proses yang dilakukan oleh sistem pada penelitian *Automatic Cat Feeding*:

- a. Source Code Servo dan RTC

```
#include "RTClib.h"
#include <Servo.h>

#define pinServo A0
```

```

// #define servoTutup 180//derajat
Servo myservo;
RTC_DS3231 rtc;
char dataHari[7][12] = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis",
"Jumat", "Sabtu"};
String hari;
int tanggal, bulan, tahun, jam, menit, detik;

void setup () {
  Serial.begin(9600);
  myservo.attach(pinServo);
  myservo.write(0);
  delay(500);

  if (! rtc.begin()) {
    Serial.println("RTC Tidak Ditemukan");
    Serial.flush();
    abort();
  }

  //Atur Waktu
  rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));
  //rtc.adjust(DateTime(2023,2, 16, 23, 33, 0));
}

void loop () {
  DateTime now = rtc.now();
  hari = dataHari[now.dayOfTheWeek()];
  tanggal = now.day(), DEC;
  bulan = now.month(), DEC;
  tahun = now.year(), DEC;
  jam = now.hour(), DEC;
  menit = now.minute(), DEC;
  detik = now.second(), DEC;
}

```



UNIVERSITAS ISLAM NEGERI
SUNAN GUNUNG DJATI

```

if(jam == 19 && menit == 55 && detik == 00){
  myservo.write(180);
  delay(4000);
  myservo.write(-180);
  // servoMakanIkan.write(servoTutup);
}

Serial.println(String() + hari + ", " + tanggal + "-" + bulan + "-" + tahun);
Serial.println(String() + jam + ":" + menit + ":" + detik);
Serial.println();

delay(1000);
}

```

b. Source Code Pembacaan Sensor berat (Load Cell)

```

#include "HX711.h"
#define DOUT A0
#define CLK A1
HX711 scale(DOUT, CLK);
float calibration_factor = 427.20;
int GRAM;

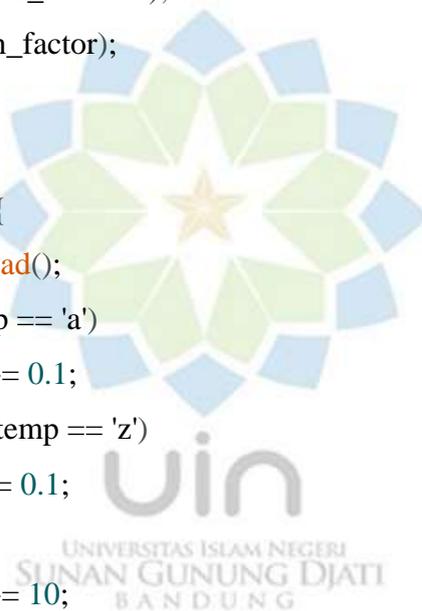
void setup() {
  Serial.begin(9600);
  Serial.println("tekan a,s,d,f untuk menaikkan calibration_factor ke
10,100,1000,10000");
  Serial.println("tekan z,x,c,v untuk menurunkan calibration_factor ke
10,100,1000,10000");
  Serial.println("Tekan T untuk Tare");
  scale.set_scale();
  scale.tare();
  long zero_factor = scale.read_average();
  Serial.print("Zero factor: ");

```

```
Serial.println(zero_factor);
delay(1000);
}

void loop() {
  scale.set_scale(calibration_factor);
  GRAM = scale.get_units(), 4;
  Serial.print("Reading: ");
  Serial.print(GRAM);
  Serial.print(" Gram");
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();

  if (Serial.available()) {
    char temp = Serial.read();
    if (temp == '+' || temp == 'a')
      calibration_factor += 0.1;
    else if (temp == '-' || temp == 'z')
      calibration_factor -= 0.1;
    else if (temp == 's')
      calibration_factor += 10;
    else if (temp == 'x')
      calibration_factor -= 10;
    else if (temp == 'd')
      calibration_factor += 100;
    else if (temp == 'c')
      calibration_factor -= 100;
    else if (temp == 'f')
      calibration_factor += 1000;
    else if (temp == 'v')
      calibration_factor -= 1000;
    else if (temp == 't')
```



```
scale.tare();  
}  
}
```

c. Source Code Pengiriman Data dari Modul Wifi NodeMCU ESP8266 ke Telegram

```
#include <ESP8266WiFi.h>  
#include <WiFiClientSecure.h>  
  
// Replace with your WiFi credentials  
const char* ssid = "JWR_Menyabungkan...";  
const char* password = "05November2021";  
  
// Replace with your Telegram bot token and chat ID  
const char* botToken = "6149539428:AAGNgLwTyL7B7BTu9-  
bky8G5VjBn9wnikVw";  
const int chatId = 5360140761;  
  
// Create a WiFiClientSecure object to connect to Telegram API  
WiFiClientSecure client;  
  
void setup() {  
  Serial.begin(9600);  
  
  // Connect to WiFi  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(1000);  
    Serial.println("Connecting to WiFi...");  
  }  
  Serial.println("WiFi connected");  
  
  // Set the client to use the Telegram API domain  
  client.setInsecure();
```

```

// Wait for the serial port to be ready
while (!Serial) {}

// Send a message to the bot
sendMessage("Hello from NodeMCU!");
}

void loop() {
  // Do nothing
}

// Function to send a message to the Telegram bot
void sendMessage(String message) {
  // Construct the API request
  String url = "https://api.telegram.org/bot" + String(botToken) +
"/sendMessage?chat_id=" + String(chatId) + "&text=" + message;

  // Send the request to the Telegram API
  if (client.connect("api.telegram.org", 443)) {
    Serial.println("Connected to Telegram API");

    // Send the HTTPS request
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
      "Host: api.telegram.org\r\n" +
      "Connection: close\r\n\r\n");
    delay(500);

    // Read the response from the Telegram API
    while (client.available()) {
      String line = client.readStringUntil('\r');
      Serial.println(line);
    }
    client.stop();
  }
}

```

```

} else {
    Serial.println("Failed to connect to Telegram API");
}
}

```

4.2 Pembahasan Hasil Penelitian

Pembahasan hasil penelitian merupakan pembahasan dari sebuah pengujian. Pengujian tersebut merupakan proses untuk menemukan kesalahan (error) atau kerusakan pada aplikasi yang telah dibangun. Proses pengujian juga dilakukan untuk mengukur kelayakan sistem dan kesesuaian sistem dengan rancangan yang telah dibuat. Proses pengujian sistem juga merupakan proses terakhir dalam metodologi pengembangan yang diterapkan pada penelitian ini. Adapun proses pengujian yang dilakukan merupakan pengujian *black box testing*.

4.2.1. Pengujian Antarmuka Sistem

Pengujian sistem ini dilakukan dengan berfokus kepada Pengujian Antarmuka dari sistem. Pengujian sistem ini dilandasi dari Halaman Utama yang dilakukan oleh user yang diberikan oleh sistem kepada *user* melalui Telegram. Berikut adalah hasil dari pengujian Antarmuka berdasarkan pada sistem yang dibangun pada penelitian ini.

a. Pengujian Halaman Utama pada Sistem

Tabel 4.1 di bawah ini merupakan tabel dari Halaman Utama pada Telegram oleh user

Tabel 4. 1 Kontrol Sistem pada Telegram

| No | Skenario | Hasil | | Keterangan |
|----|-----------------------------|--------|-------|--|
| | | Sukses | Gagal | |
| 1. | Menginputkan berat kucing | ✓ | | Sistem berhasil menerima data inputan dari user yang data tersebut akan di terima oleh microcontroller . |
| 2. | Notifikasi dispenser kosong | ✓ | | Sistem berhasil memberikan notifikasi pada user bahwa |

| | | | | |
|--|--|--|--|--------------------------------|
| | | | | dispenser pakan kucing kosong. |
|--|--|--|--|--------------------------------|

4.2.2. Pengujian Prototype

Fungsi pada pengujian prototype ini untuk memastikan bahwa Rancang Bangun Sistem *Aotomatic Cat Feeding* ini telah sesuai dengan yang dirancang dan seluruh fungsi yang ada didalam sistem dapat berjalan secara efektif. Selain itu pada Pengujian *Prototype* ini akan diketahui kelebihan serta kekurangan dari sistem yang dibangun sehingga nantinya dapat menjadi bahan acuan untuk penelitian selanjutnya.

a. Pengujian Servo dan RTC

Dalam penelitian ini servo digunakan untuk membuka dan menutup dispenser yang berisi pakan kucing. Servo akan memutar menutup jika waktu tidak sesuai dengan jadwal yang telah ditentukan, dan servo akan memutar membuka jika waktu sudah menunjukkan jadwal pemberian pakan. RTC atau sensor waktu digunakan untuk set jadwal pemberian pakan kucing. Jika waktu sudah menunjukkan pukul 09.00 dan 16.00 servo akan memutar membuka dan otomatis pakan langsung keluar.

Tabel 4.2 di bawah ini merupakan Hasil Pengujian Servo dan RTC atau sensor waktu.

Tabel 4. 2 pengujian servo dan rtc

| No | Waktu Alat | Servo | Buka | Tutup |
|----|------------|----------|------|-------|
| 1 | 09.00 | Bergerak | ✓ | - |
| 2 | 10.00 | Diam | - | ✓ |
| 3 | 11.00 | Diam | - | ✓ |
| 4 | 12.00 | Diam | - | ✓ |
| 5 | 13.00 | Diam | - | ✓ |
| 6 | 14.00 | Diam | - | ✓ |
| 7 | 15.00 | Diam | - | ✓ |
| 8 | 16.00 | Bergerak | ✓ | |

b. Pengujian Servo dan Load cell

Load cell digunakan sebagai penimbang berat pakan ang dikeluarkan, ketika pakan dibawah timbangansudah mencapai berat 60gr atau sesuai yang ditentukan maka akan memberi perintah ke Arduino untuk menutup pintu wadah.

Tabel 4.3 di bawah ini merupakan hasil yang pengujian yang dilakukan oleh penulis dalam pengujian Sensor Berat atau *Load cell* berdasarkan berat yang sesuai data.

Tabel 4. 3 pengujian servo dan load cell

| No | Timbangan | Servo (Buka) | Servo (Tutup) |
|-----|-----------|--------------|---------------|
| 1. | 60gr | - | ✓ |
| 2. | 70gr | - | ✓ |
| 3. | 80gr | - | ✓ |
| 4. | 90gr | ✓ | - |
| 5. | 100gr | - | ✓ |
| 6. | 110gr | - | ✓ |
| 7. | 120gr | - | ✓ |
| 8. | 130gr | - | ✓ |
| 9. | 140gr | - | ✓ |
| 10. | 150gr | ✓ | - |
| 11 | 160gr | ✓ | - |

c. Pengujian Modul Wifi ESP8266

Modul ESP8266 yang digunakan pada pengujian ini merupakan modul wifi yang sudah tertanam pada Wemos D1 R2 sebagai media peringatan untuk user berupa Peringatan pesan yang dikirimkan melalui *Telegram Messenger*. Pada proses pengujian ini menghitung keberhasilan Modul Wifi dalam mengirimkan data yang diperoleh dan *delay* yang terjadi saat pengiriman data ke telegram. Koneksi internet sangat berpengaruh terhadap kecepatan pengiriman data yang dilakukan oleh Modul Wifi ESP8266.

Berikut Tabel 4.4 Pengujian Koneksi Modul Wifi ESP8266 terhadap pengiriman data ke user melalui *Telegram Messenger*.

Tabel 4. 4 pengujian koneksi nodeMCU

| No | Skenario | Status | Delay (s) |
|----|------------------------------|----------|-----------|
| 1 | Pengiriman Data Sensor Waktu | Berhasil | 1 Detik |
| 2 | Pengiriman Data Sensor Waktu | Berhasil | 2 Detik |
| 3 | Pengiriman Data Sensor Waktu | Berhasil | 1 Detik |
| 4 | Pengiriman Data Sensor Waktu | Berhasil | 1 Detik |

| | | | |
|----|------------------------------|----------|---------|
| 5 | Pengiriman Data Sensor Waktu | Berhasil | 2 Detik |
| 6 | Pengiriman Data Sensor Berat | Berhasil | 3 Detik |
| 7 | Pengiriman Data Sensor Berat | Berhasil | 2 Detik |
| 8 | Pengiriman Data Sensor Berat | Berhasil | 1 Detik |
| 9 | Pengiriman Data Sensor Berat | Berhasil | 1 Detik |
| 10 | Pengiriman Data Sensor Berat | Berhasil | 3 Detik |

4.2.3. Pembahasan Hasil Pengujian

Menurut Ravianto efektivitas adalah seberapa baik pekerjaan yang dilakukan, sejauh mana orang menghasilkan keluaran sesuai yang diharapkan. Efektivitas harus dinilai atas dasar tujuan yang bisa dilaksanakan bukan atas dasar konsep tujuan yang maksimum. Efektivitas dapat diukur dengan menggunakan standar sesuai dengan acuan Litbang Depdagri (1991) seperti pada tabel di bawah:

Tabel 4. 5 tabel data acuan litbang degdagri

| Rasio Efektivitas | Tingkat Capaian |
|-------------------|----------------------|
| Dibawah 40 | Sangat tidak Efektif |
| 40 – 59,99 | Tidak Efektif |
| 60 – 79,99 | Cukup Efektif |
| Diatas 80 | Sangat Efektif |

Setelah melakukan pengujian yang telah dilakukan yaitu pengujian sensor berat atau *Load cell* dan sensor waktu atau RTC, persentase dari seluruh pengujian dapat dilihat pada tabel dibawah ini.

Tabel 4.8 dibawah ini merupakan tabel data yang Pengujian seluruh alat beserta dengan perhitungan.

Tabel 4. 6 Tabel data pengujian seluruh alat

| Jumlah Data Uji | Total Berhasil | Total Tidak Berhasil |
|-----------------|----------------|----------------------|
| 31 | 28 | 3 |

Nilai Akurasi dari Hasil Pengujian adalah:

| |
|--|
| $\frac{\text{Total Pengujian Berhasil}}{\text{Total Pengujian}} \times 100\% = \text{Nilai Akurasi}$ |
|--|

$$\text{Hasil nilai akurasi sistem Automatic Cat Feeding} = \frac{28}{31} \times 100\% = 90,3\%$$

Nilai yang didapatkan dari total 31 pengujian yaitu pengujian sensor berat sebanyak 11 kali, sensor waktu atau RTC sebanyak 8 kali, nodeMCU sebanyak 10 kali dan mengalami kegagalan sebanyak 3 kali pengujian. Kegagalan tersebut disebabkan karena kurang akuratnya sensor berat membaca datanya. Maka hasilnya yaitu mendapatkan persentase nilai akurasi sebesar 85,25%. Maka jika mengacu pada tabel efektivitas Litbang Depdagri, nilai persentase pengujian pada sistem ini berkategori Sangat Efektif.

