

BAB IV IMPLEMENTASI DAN PENGUJIAN

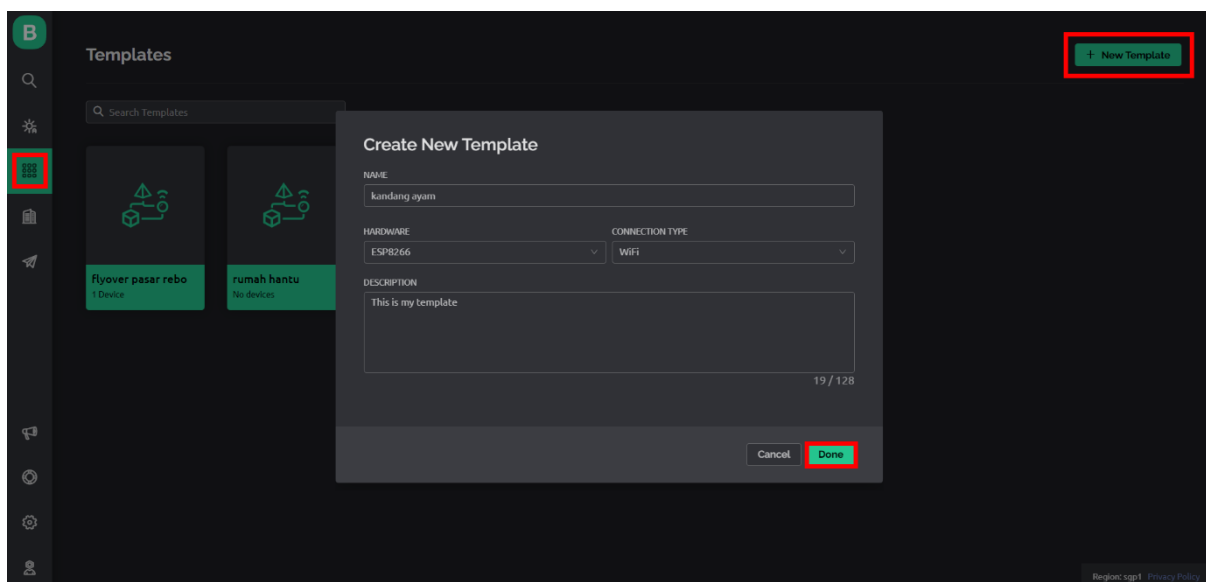
4.1 Implementasi Sistem

Implementasi sistem merupakan penerapan sistem yang didasari perancangan sistem yang sudah dibuat sebelumnya. Hasil dari implemetasi sistem dibuat dibuat dengan rancangan yang sesuai dengan kebutuhan sistem.

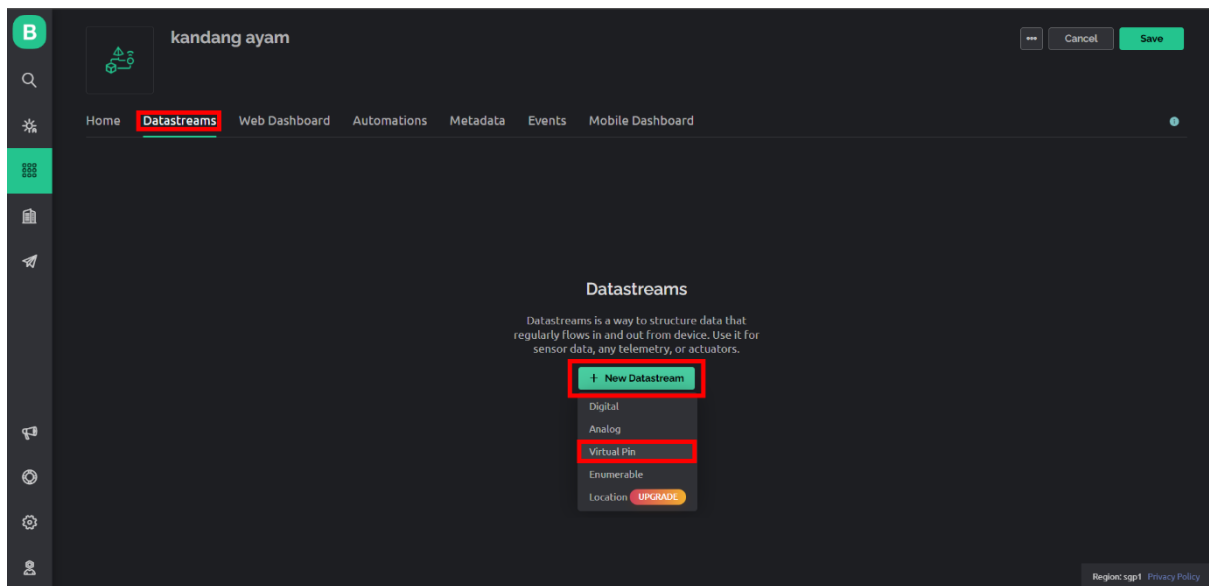
4.1.1 Konfigurasi Blynk di browser

Sebelum menggunakan platform Blynk, perlu dilakukan konfigurasi terlebih dahulu supaya data-data yang diperlukan bisa diakses pada perangkat *smartphone*.

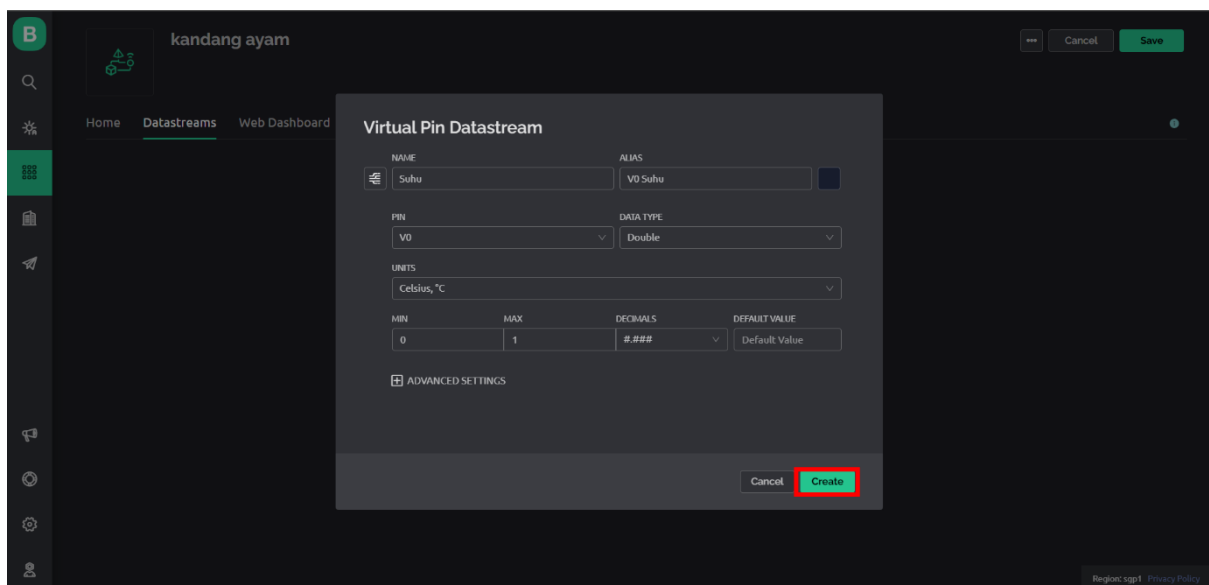
1. Mengunjungi halaman web Blynk dan melakukan klik pada menu Templates di sebelah kiri. Selanjutnya melakukan klik pada tombol + New Template kemudian mengisi seluruh kolom yang tersedia pada kotak dialog untuk membuat template baru. Ketika sudah selesai langkah terakhir yaitu melakukan klik pada tombol Done.



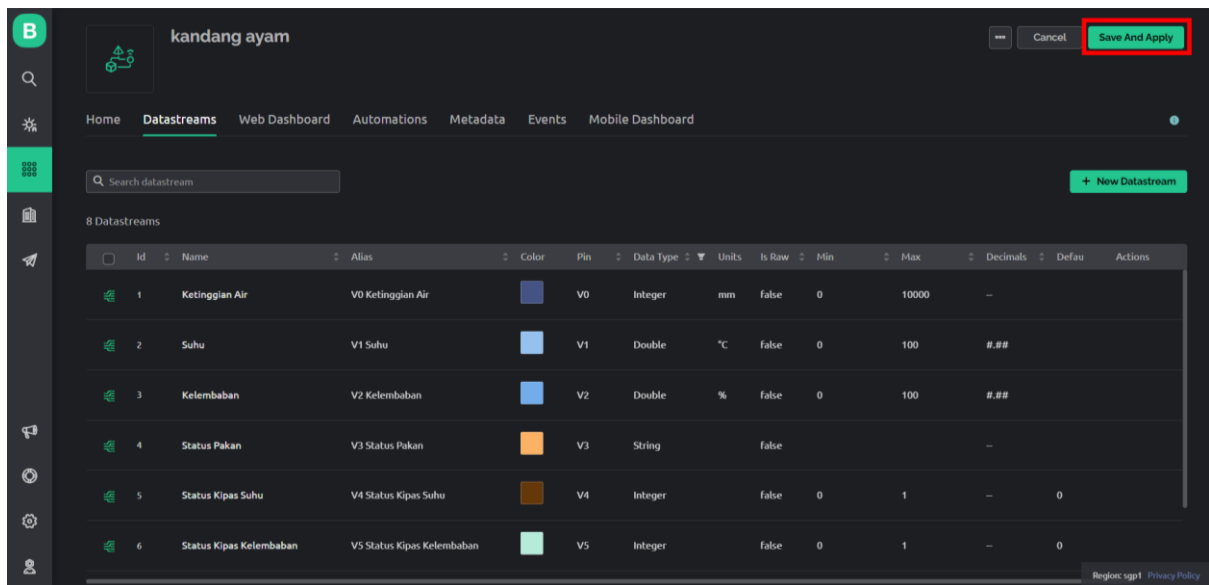
2. Setelah selesai maka akan diarahkan ke halaman berikutnya untuk langkah konfigurasi lebih lanjut. Melakukan klik pada menu Datastreams dan tombol + New Datastream dan memilih opsi Virtual Pin.



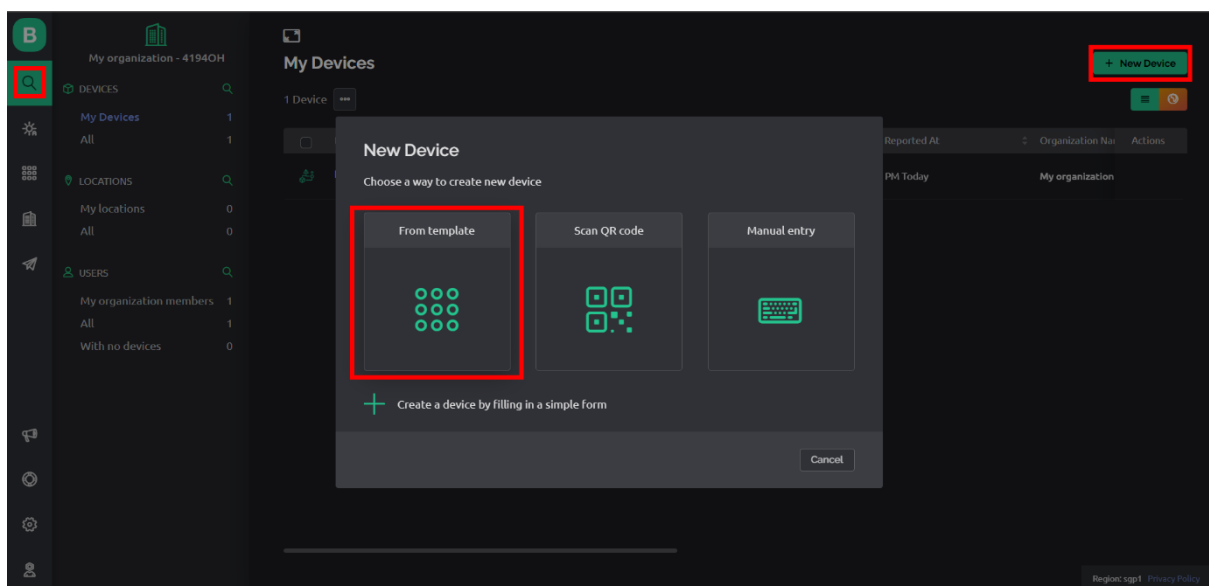
3. Selanjutnya akan muncul kotak dialog untuk membuat virtual pin. Virtual pin berperan untuk menyimpan data-data yang dikirimkan oleh mikrokontroler ke platform Blynk. Langkah selanjutnya mengisi seluruh kolom yang tersedia kemudian melakukan klik pada tombol Create apabila sudah selesai.



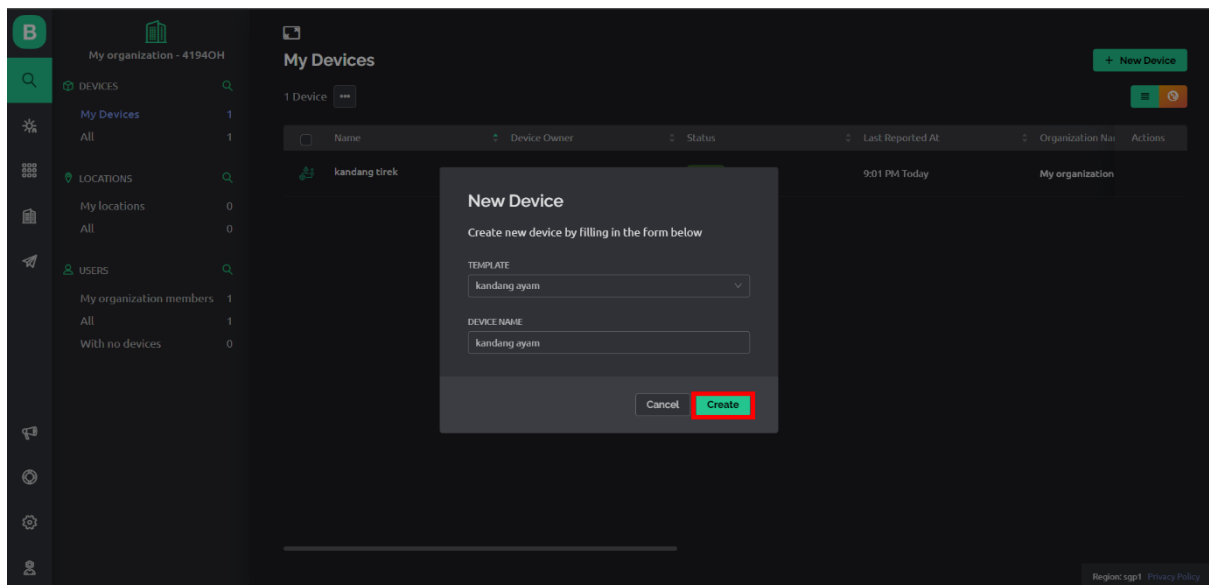
4. Setelah selesai membuat beberapa virtual pin yang dibutuhkan dapat dilanjutkan dengan melakukan klik pada tombol Save And Apply.



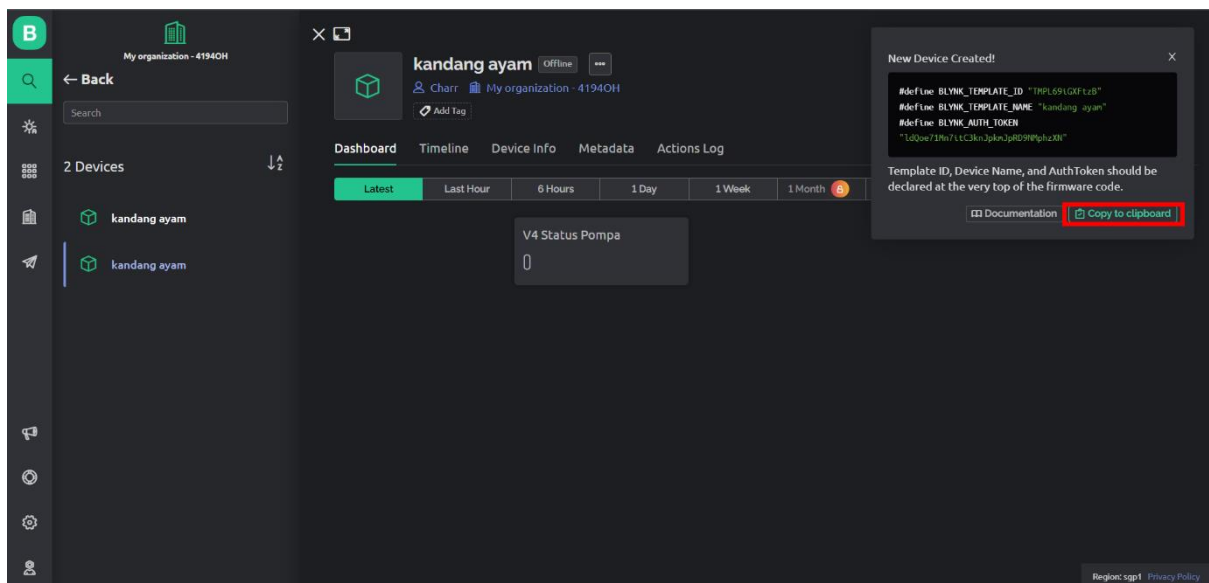
5. Kemudian melakukan klik pada menu Search serta tombol + New Device. Setelah itu memilih opsi From Template pada kotak dialog yang muncul.



6. Pada kotak dialog yang muncul dapat memilih template yang sudah dibuat sebelumnya dan melakukan klik pada tombol Create apabila sudah selesai.

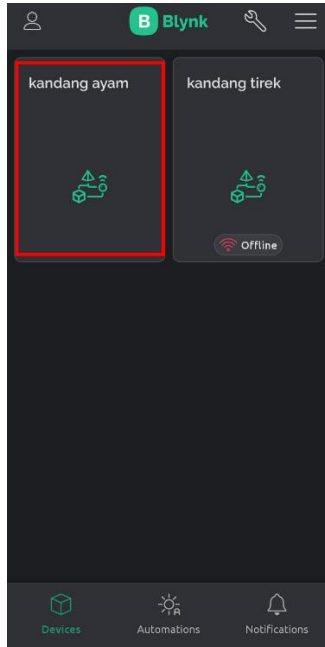


7. Pada kotak dialog yang muncul di halaman baru dapat melakukan klik pada tombol Copy to clipboard untuk menyalin ID Template, Nama Perangkat, serta Token Autentikasi yang akan digunakan pada kode yang sudah dibuat.

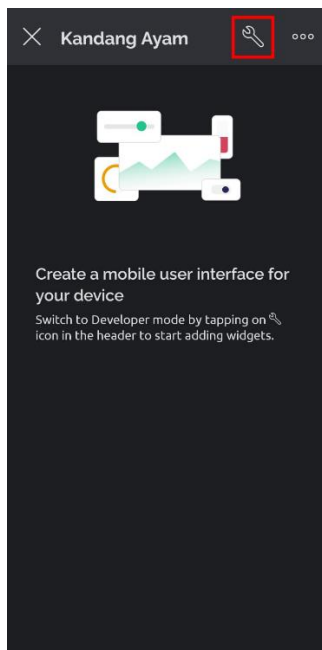


4.1.2 Konfigurasi Blynk di ponsel

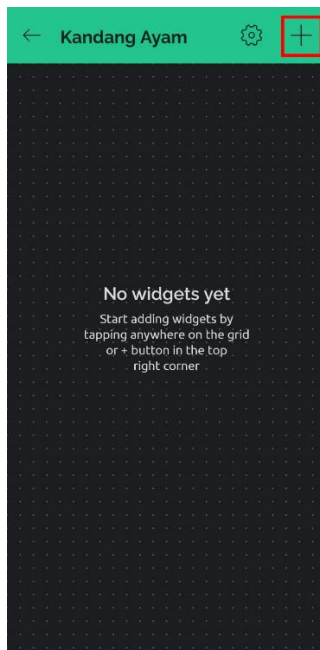
1. Konfigurasi platform Blynk dilakukan pada dua perangkat yaitu komputer dan *smartphone*. Untuk konfigurasi pada perangkat *smartphone* dapat dimulai dengan memilih device yang sudah dibuat sebelumnya ketika melakukan konfigurasi menggunakan komputer.



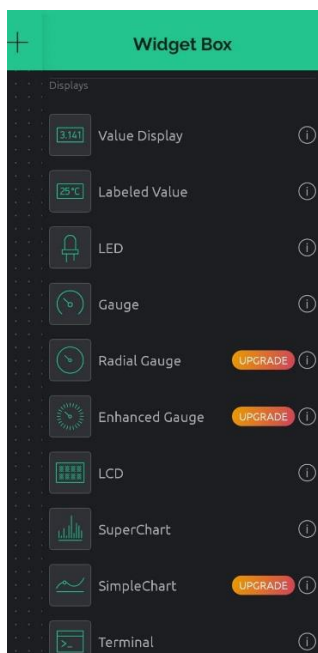
2. Pada halaman berikut masih kosong dikarenakan belum ada *widget* yang dipilih. Langkah berikutnya menambahkan *widget* dengan menekan tombol kunci pas. *Widget* ini digunakan untuk menampilkan data-data dari mikrokontroler yang sudah ditampung pada virtual pin yang sudah dibuat sebelumnya.



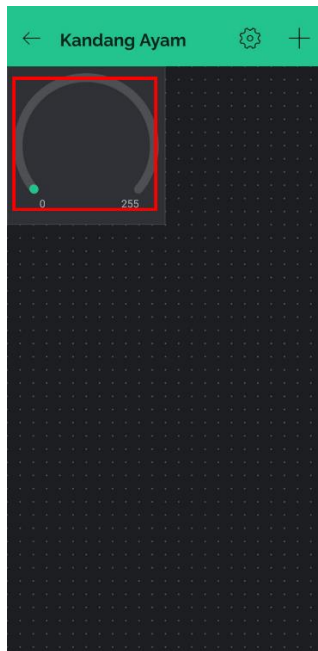
3. Dilanjutkan dengan menekan tombol + untuk menambahkan dan memilih *widget*.



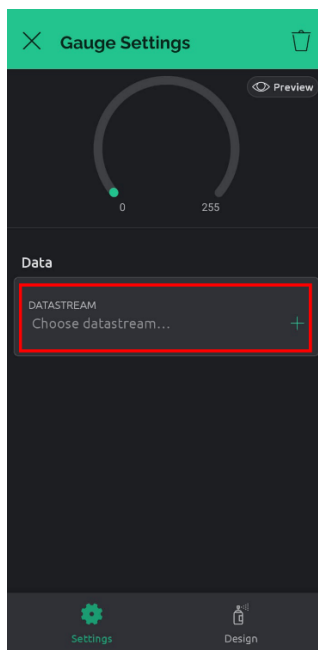
4. Pada halaman ini dapat memilih berbagai macam jenis *widget* yang sesuai dengan kebutuhan dan data-data yang digunakan.



5. *Widget* yang sudah dipilih sebelumnya akan tampil di halaman *widget*. Langkah selanjutnya melakukan konfigurasi pada *widget* tersebut dengan menekan *widget*.



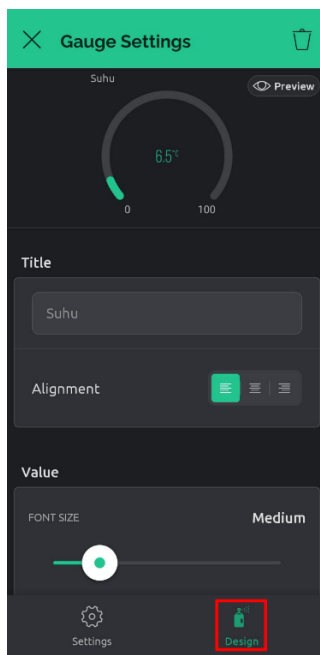
6. Konfigurasi *widget* dilakukan dengan memilih *datastream* yang sudah dibuat sebelumnya ketika melakukan konfigurasi melalui komputer.



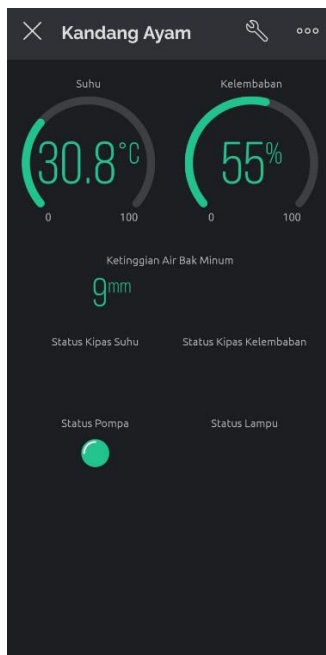
7. Memilih datastream yang sudah dibuat sebelumnya ketika melakukan konfigurasi melalui komputer.



8. Apabila ingin melakukan konfigurasi tampilan *widget* dapat menekan tombol Design dan melakukan konfigurasi yang diinginkan seperti judul dan ukuran *font widget*.

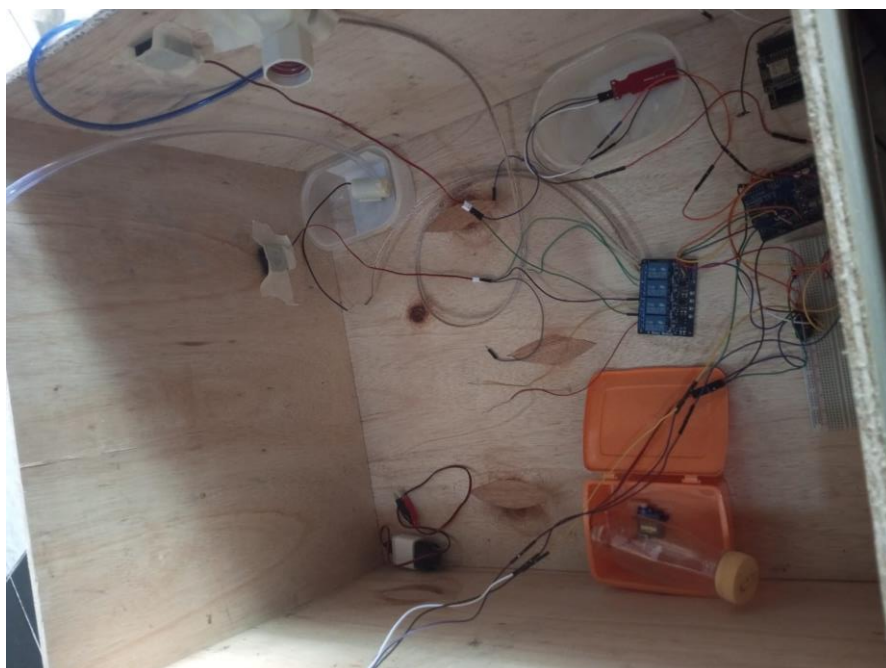


9. Berikut merupakan halaman *widget* yang sudah selesai dikonfigurasi sesuai dengan kebutuhan dan data-data yang digunakan.



4.1.3 Implementasi *Hardware*

Implementasi *hardware* dalam pengembangan "Purwarupa Kandang Ayam Pintar berbasis *Internet of Things (IoT)* menggunakan Platform Blynk" melibatkan mikrokontroler NodeMCU ESP8266 dan Arduino Uno yang dihubungkan dengan sensor ketinggian air, sensor DHT11, modul RTC DS3231, servo, relay, kipas kecil, pompa air, serta lampu bohlam. Setelah semua komponen yang diperlukan terhubung satu sama lain dan diberikan tenaga dengan cara dihubungkan ke adaptor 5 V serta dihubungkan dengan koneksi internet melalui *Wi-Fi* supaya bisa mengirim data ke platform Blynk. Gambar di bawah merupakan tampilan dari purwarupa alat yang digunakan.



4.2 Pengkodean

Dalam pengembangan "Purwarupa Kandang Ayam Pintar berbasis *Internet of Things (IoT)*" menggunakan Platform Blynk" menggunakan dua *source code* yang diupload pada 2 mikrokontroler yaitu Arduino Uno dan NodeMCU ESP8266.

4.2.1 Pengkodean pada Arduino Uno

Berikut merupakan *source code* yang diupload pada Arduino Uno.



```
1 //dht11
2 #include <DHT.h>
3 DHT dht(7, DHT11); //Pin, Jenis DHT
4
5 //rtc
6 #include "RTClib.h"
7 RTC_DS3231 rtc;
8 char dataHari[7][12] = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"};
9 String hari;
10 int tanggal, bulan, tahun, jam, menit, detik;
11 //rtc
12
13 //servo
14 #include <Servo.h>
15 Servo servoku;
16 //servo
17
18 //sensor ketinggian air
19 int analogPin = A1;
20 int ketinggianAir;
21 int ketinggianAirMm;
22 //sensor ketinggian air
23
24 // relay
25 const int kipas1 = 10; //relay1 suhu
26 const int kipas2 = 11; //relay2 kelembaban
27 const int pompa = 12; //relay3
28 const int lampu = 13; //relay4
29
30 //on off relay
31 int relayON = LOW; //relay nyala
32 int relayOFF = HIGH; //relay mati
33
34 // status relay
35 bool statusKipasSuhu;
36 bool statusKipasKelembaban;
37 bool statusPompa;
38 bool statusLampu;
39 // relay
```

```
1 void setup() {
2   Serial.begin(9600);
3
4   // dht11
5   dht.begin();
6
7   // servo
8   servoku.attach(6);
9
10  // rtc ds3231
11  if (! rtc.begin()) {
12    Serial.println("RTC Tidak Ditemukan");
13    Serial.flush();
14    abort();
15  }
16
17  //Atur Waktu
18  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
19  // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
20  // rtc ds3231
21
22  // setup relay
23  pinMode(kipas1, OUTPUT);
24  pinMode(kipas2, OUTPUT);
25  pinMode(pompa, OUTPUT);
26  pinMode(lampu, OUTPUT);
27  digitalWrite(kipas1, relayOFF);
28  digitalWrite(kipas2, relayOFF);
29  digitalWrite(pompa, relayOFF);
30  digitalWrite(lampu, relayOFF);
31 }
```



```
1 void loop() {
2   // BACA Suhu DAN KELEMBABAN
3   float kelembaban = dht.readHumidity();
4   float suhu = dht.readTemperature();
5   // BACA Suhu DAN KELEMBABAN
6
7   // BACA KETINGGIAN AIR
8   ketinggianAir = analogRead(analogIn);
9   ketinggianAir = map(ketinggianAir, 0, 1023, 0, 255);
10
11   // KONDISI MODUL MONITORING SUSHU DAN KELEMBABAN
12   // Suhu normal
13   if (suhu > 29 && suhu <= 31) {
14     matikanModulPeningkatanSuhu();
15     matikanModulPenurunanSuhu();
16
17     // suhu di bawah standar minimum
18   } else if (suhu < 29) {
19     modulPeningkatanSuhu();
20
21     // suhu di atas standar maksimal
22   } else if (suhu > 31) {
23     modulPenurunanSuhu();
24
25     // kelembaban normal
26   } else if (kelembaban >= 50 && kelembaban <= 70) {
27     matikanModulPeningkatanKelembaban();
28     matikanModulPenurunanKelembaban();
29
30     // kelembaban di bawah standar minimum
31   } else if (kelembaban < 50) {
32     modulPeningkatanKelembaban();
33
34     // kelembaban di atas standar maksimal
35   } else if (kelembaban > 70) {
36     modulPenurunanKelembaban();
37   }
38   // KONDISI MODUL MONITORING SUSHU DAN KELEMBABAN
39
40   // KONDISI MODUL MINUM
41   // kondisi kosong
42   if (ketinggianAir < 30) {
43     modulIsiBakMinum();
44
45   } else if (ketinggianAir > 50) {
46     matikanModulIsiBakMinum();
47   }
48   // KONDISI MODUL MINUM
49
50   // BACA WAKTU
51   DateTime now = rtc.now();
52   hari = now.dayOfTheWeek();
53   tanggal = now.day(), DEC;
54   bulan = now.month(), DEC;
55   tahun = now.year(), DEC;
56   jam = now.hour(), DEC;
57   menit = now.minute(), DEC;
58   detik = now.second(), DEC;
59   // BACA WAKTU
60
61   // KONDISI MODUL PAKAN
62   // kondisi kosong
63   if (jam == 13 && menit == 49 && detik == 10) {
64     modulPakan();
65   }
66
67   if (jam == 13 && menit == 49 && detik == 30) {
68     modulPakan();
69   }
70
71   if (jam == 13 && menit == 49 && detik == 50) {
72     modulPakan();
73   }
74   // KONDISI MODUL PAKAN
75
76   // KONDISI MODUL MODE MALAM
77   // kondisi kosong
78   if (jam == 13 && menit == 0 && detik == 0) {
79     matikanModulModeMalam();
80
81   } else if (jam == 13 && menit == 2 && detik == 0) {
82     modulModeMalam();
83   }
84   // KONDISI MODUL MODE MALAM
85
86   // KIRIM DATA KE NODEMCU
87   modulKirimData();
88   // KIRIM DATA KE NODEMCU
89 }
```

```
1 // MODUL MINUM
2 void modulIsiBakMinum() {
3   // mengisi bak minum dengan menyala pompa air ketika bak minum hampir kosong
4   //relay1
5   digitalWrite(pompa, relayON);
6   statusPompa = HIGH;
7 }
8
9 void matikanModulIsiBakMinum() {
10  // mematikan pompa air ketika bak minum sudah hampir penuh
11  //relay1
12  digitalWrite(pompa, relayOFF);
13  statusPompa = LOW;
14 }
15 // MODUL MINUM
16
17 // MODUL MONITORING SUSHU DAN KELEMBABAN
18 void modulPeningkatanSuhu() {
19  // meningkatkan suhu kandang ketika suhu di bawah normal dengan menyala Lampu
20  //relay2
21  digitalWrite(lampu, relayON);
22  statusLampu = HIGH;
23 }
24
25 void matikanModulPeningkatanSuhu() {
26  // mematikan modul peningkatan suhu ketika suhu normal tercapai dengan mematikan Lampu
27  //relay2
28  digitalWrite(lampu, relayOFF);
29  statusLampu = LOW;
30 }
31
32 void modulPenurunanSuhu() {
33  // menurunkan suhu kandang ketika suhu di atas normal dengan menyala kipas1
34  //relay1
35  digitalWrite(kipas1, relayON);
36  statusKipasSuhu = HIGH;
37 }
38
39 void matikanModulPenurunanSuhu() {
40  // mematikan modul penurunan suhu ketika suhu normal tercapai dengan mematikan kipas1
41  //relay1
42  digitalWrite(kipas1, relayOFF);
43  statusKipasSuhu = LOW;
44 }
45
46 void modulPeningkatanKelembaban() {
47  // meningkatkan kelembaban kandang ketika kelembaban di bawah normal dengan menyala kipas
48  //relay2
49  digitalWrite(kipas2, relayON);
50  statusKipasKelembaban = HIGH;
51 }
52
53 void matikanModulPeningkatanKelembaban() {
54  // mematikan modul peningkatan kelembaban ketika kelembaban normal tercapai dengan mematikan kipas
55  //relay2
56  digitalWrite(kipas2, relayOFF);
57  statusKipasKelembaban = LOW;
58 }
59
60 void modulPenurunanKelembaban() {
61  // menurunkan kelembaban kandang ketika kelembaban di atas normal dengan menyala Lampu
62  //relay2
63  digitalWrite(lampu, relayON);
64  statusLampu = HIGH;
65 }
66
67 void matikanModulPenurunanKelembaban() {
68  // mematikan modul penurunan kelembaban ketika kelembaban normal tercapai dengan mematikan Lampu
69  //relay2
70  digitalWrite(lampu, relayOFF);
71  statusLampu = LOW;
72 }
73 // MODUL MONITORING SUSHU DAN KELEMBABAN
```

```
1 // MODUL PAKAN
2 void modulPakan() {
3   // kirim perintah pakan setiap waktu
4   for (int posisi = 0; posisi <= 100; posisi++) {
5     // kirim perintah ke motor driver untuk pakan
6     servo.write(posisi);
7     delay(10);
8   }
9
10  // kirim perintah pakan pada setiap 10 detik satu
11  for (int posisi = 100; posisi >= 0; posisi--) {
12    servo.write(posisi);
13    delay(10);
14  }
15 }
16 // MODUL PAKAN
17
18 // MODUL MODE MALAM
19 void modulModeMalam() {
20  // mematikan lampu kandang ketika waktu sudah memasuki malam hari
21  //relay1
22  digitalWrite(lampu, relayOFF);
23  statusLampu = LOW;
24 }
25
26 void matikanModulModeMalam() {
27  // mematikan lampu kandang ketika waktu sudah memasuki pagi hari
28  //relay1
29  digitalWrite(lampu, relayON);
30  statusLampu = HIGH;
31 }
32 // MODUL MODE MALAM
33
34 // MODUL KIRIM DATA
35 void modulKirimData() {
36  // kirim data ke NODEMCU
37  float kelembaban = dht.readHumidity();
38  float suhu = dht.readTemperature();
39  // kirim data ke NODEMCU
40
41  String dataStr1 = "suhu:" + String(suhu) + "&" + String(kelembaban) + "&" + String(tahun) + "&" + String(tanggal) + "&" + String(bulan) + "&" + String(jam) + "&" + String(minit) + "&" + String(detik) + "&" + String(statusPompa) + "&" + String(statusLampu);
42  Serial.println(dataStr1);
43  delay(1000);
44 }
45 // MODUL KIRIM DATA
```

4.2.2 Pengkodean pada NodeMCU ESP8266

Berikut *source code* yang diupload pada NodeMCU ESP8266.

```
1  //define BLYNK_TEMPLATE_ID "TMPL69iGXftzB"
2  //define BLYNK_TEMPLATE_NAME "kandang ayam"
3  //define BLYNK_AUTH_TOKEN "7RyZrouJIw0UjHCwF0YN1iWY-qcmfkdN"
4
5  #define BLYNK_AUTH_TOKEN "7RyZrouJIw0UjHCwF0YN1iWY-qcmfkdN"
6
7  #define BLYNK_PRINT Serial
8  #include <ESP8266WiFi.h>
9  #include <BlynkSimpleEsp8266.h>
10
11  char auth[] = BLYNK_AUTH_TOKEN;
12
13  char ssid[] = "RINJANI01";
14  char pass[] = "12345678";
15  //char ssid[] = "bringthaton";
16  //char pass[] = "12345666";
17  //char ssid[] = "Hotspot Area";
18  //char pass[] = "sekolahvokasimadiun";
19
20  BlynkTimer timer;
21
22  String dataIn;
23  String dt[11];
24  int i;
25  boolean parsing = false;
```

```
1  void setup(){
2    Serial.begin(9600);
3
4    dataIn = "";
5
6    // koneksi ke wifi
7    WiFi.begin(ssid, pass);
8
9    // cek koneksi wifi
10   while (WiFi.status() != WL_CONNECTED) {
11     Serial.print(".");
12     delay(1000);
13   }
14   Serial.println("Berhasil terhubung ke WiFi");
15   // koneksi ke wifi
16
17   Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
18   timer.setInterval(1000L, sendSensor);
19 }
20
21 void loop(){
22   if(Serial.available() > 0) {
23     char inChar = (char)Serial.read();
24     dataIn += inChar;
25
26     if (inChar == '\n') {
27       parsing = true;
28     }
29   }
30
31   if(parsing) {
32     parsingData();
33     parsing = false;
34     dataIn = "";
35   }
36
37   Blynk.run();
38   timer.run();
39 }
```

```
1  void parsingData() {
2    int j = 0;
3
4    // kirim data yang telah diterima sebelumnya
5    Serial.print("data masuk: ");
6    Serial.print(dataIn);
7
8    //inisialisasi variabel, (reset isi variabel)
9    dt[j] = "";
10
11    //proses parsing data
12    for(i=1; i<dataIn.length(); i++) {
13      //SEPARATOR (#)
14      if((dataIn[i] == '#')) {
15        //increment variabel j, digunakan untuk merubah index array penampung
16        j++;
17        dt[j] = ""; //inisialisasi variabel array dt[j]
18      } else {
19        //proses menampung data saat pengecekan karakter sudah selesai
20        dt[j] = dt[j] + dataIn[i];
21      }
22    }
23
24    // print data hasil parsing
25    // Serial.print("Ketinggian Air: ");
26    // Serial.println(dt[0].toInt());
27    // Serial.print("Suhu: ");
28    // Serial.println(dt[1].toFloat());
29    // Serial.print("Kelembaban: ");
30    // Serial.println(dt[2].toFloat());
31    // Serial.print("Status Pakan: ");
32    // Serial.println(dt[3]);
33    // Serial.print("StatusKipasSuhu: ");
34    // Serial.println(dt[4]);
35    // Serial.print("statusKipasKelembaban: ");
36    // Serial.println(dt[5]);
37    // Serial.print("Status Pompa: ");
38    // Serial.println(dt[6]);
39    // Serial.print("StatusLampu: ");
40    // Serial.println(dt[7]);
41    // delay(5000);
42  }
```

```
1  void sendSensor(){
2    Blynk.virtualWrite(V0, dt[0].toInt()); // ketinggian air
3    Blynk.virtualWrite(V1, dt[1].toFloat()); // suhu
4    Blynk.virtualWrite(V2, dt[2].toFloat()); // kelembaban
5    Blynk.virtualWrite(V3, dt[3]); // statusPakan
6    Blynk.virtualWrite(V4, dt[4].toInt()); // statusKipasSuhu
7    Blynk.virtualWrite(V5, dt[5].toInt()); // statusKipasKelembaban
8    Blynk.virtualWrite(V6, dt[6].toInt()); // statusPompa
9    Blynk.virtualWrite(V7, dt[7].toInt()); // statusLampu
10
11    // kondisi coba2
12    if(dt[3] == "15:35:0"){
13      Blynk.logEvent("status_pakan","Pakan sudah diberikan pada jam 7");
14    }
15    else if(dt[3] == "15:36:0"){
16      Blynk.logEvent("status_pakan","Pakan sudah diberikan pada jam 12");
17    }
18    else if(dt[3] == "15:37:0"){
19      Blynk.logEvent("status_pakan","Pakan sudah diberikan pada jam 17");
20    }
21
22    // kondisi real
23    // if(dt[3] == "7:0:0"){
24    //   Blynk.logEvent("status_pakan","Pakan sudah diberikan pada jam 7");
25    // }
26    // } else if(dt[3] == "12:0:0"){
27    //   Blynk.logEvent("status_pakan","Pakan sudah diberikan pada jam 12");
28    // }
29    // } else if(dt[3] == "17:0:0"){
30    //   Blynk.logEvent("status_pakan","Pakan sudah diberikan pada jam 17");
31    // }
32 }
```

4.3 Pengujian Hardware

Purwarupa Kandang Ayam Pintar berbasis Internet of Things akan dilakukan pengujian terhadap semua komponen dan modul yang digunakan secara satu per satu. Pengujian akan dilakukan terhadap pembacaan sensor DHT11, sensor ketinggian air, modul RTC DS3231, dan juga cara kerja servo, relay, kipas DC, lampu pijar, serta pompa air mini.

4.3.1 Pengujian Sensor DHT11

Untuk mengukur suhu dan kelembaban di kandang ayam maka dilakukan pengujian terhadap sensor DHT11.

Resistansi	Keterangan
Suhu > 29°C - 31°C	Panas
Suhu = 29°C - 31°C	Normal
Suhu < 29°C - 31°C	Dingin

Resistansi	Keterangan
Kelembaban < 50% - 70%	Kering
Kelembaban = 50% - 70%	Normal
Kelembaban > 50% - 70%	Lembab

4.3.2 Pengujian Sensor Ketinggian Air

Tujuan dilakukannya pengujian terhadap sensor ketinggian air untuk mengukur nilai resistansi ketinggian air.

Resistansi	Keterangan
30 - 0	Kosong
31 - 40	Sedang
41 - 60	Penuh

4.3.3 Pengujian Modul RTC DS3231

Modul RTC DS3231 dilakukan pengujian dengan cara dibandingkan dengan jam tangan guna mengetahui ketepatan waktu yang ditunjukkan oleh modul RTC DS3231.

Waktu RTC	Waktu Digital	Keterangan
07.00	07.00	Sesuai
12.00	12.00	Sesuai
17.00	17.00	Sesuai

4.3.4 Pengujian Servo

Pada servo dilakukan pengujian untuk mengetahui pergerakan servo saat waktu makan ayam sudah tiba. Hal ini dikarenakan tempat pakan ayam menempel pada servo sehingga perlu dilakukan pengujian untuk mengetahui apakah servo bergerak atau tidak.

Kondisi	Sudut dalam derajat	Keterangan
Servo ON	180°	Servo terbuka
Servo OFF	0°	Servo tertutup

4.3.5 Pengujian Relay

Pengujian relay dilakukan untuk mengetahui apakah relay menyalakan dan mematikan kipas DC, lampu pijar, serta pompa air mini sesuai dengan kondisi yang sudah ditentukan atau tidak.

Kondisi	Keterangan
Relay Channel 1 ON	Kipas menyala
Relay Channel 1 OFF	Kipas mati
Relay Channel 2 ON	Kipas menyala
Relay Channel 2 OFF	Kipas mati
Relay Channel 3 ON	Pompa menyala
Relay Channel 3 OFF	Pompa mati
Relay Channel 4 ON	Lampu menyala
Relay Channel 4 OFF	Lampu mati

4.3.6 Pengujian Kipas DC

Kipas DC digunakan untuk menurunkan suhu dan meningkatkan kelembaban pada kandang ayam. Pengujian yang dilakukan terhadap kipas DC bertujuan untuk mengetahui apakah kipas bisa menyala dan mati pada kondisi yang sudah ditentukan.

Kondisi	Keterangan
Suhu > 29°C - 31°C	Kipas 1 menyala
Suhu ≤ 29°C - 31°C	Kipas 1 mati
Kelembaban < 50% - 70%	Kipas 2 menyala
Kelembaban > 50% - 70%	Kipas 2 mati

4.3.7 Pengujian Lampu Pijar

Lampu pijar digunakan untuk meningkatkan suhu dan menurunkan kelembaban pada kandang ayam. Lampu pijar dilakukan pengujian untuk mengetahui apakah lampu pijar bisa menyala dan mati sesuai dengan kondisi yang sudah ditentukan.

Kondisi	Keterangan
Suhu > 29°C - 31°C	Lampu mati
Suhu ≤ 29°C - 31°C	Lampu menyala
Kelembaban < 50% - 70%	Lampu mati
Kelembaban > 50% - 70%	Lampu menyala

4.3.8 Pengujian Pompa Air Mini

Pompa air mini digunakan untuk mengalirkan air ke tempat minum pada kandang ayam berdasarkan nilai resistansi dari sensor ketinggian air. Pada pompa air mini dilakukan pengujian untuk mengetahui apakah pompa air bisa menyala dan mati sesuai dengan kondisi yang sudah ditentukan.

Resistansi	Keterangan
Ketinggian air < 30	Pompa menyala
Ketinggian air > 60	Pompa mati

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Apabila modul RTC DS3231 menunjukkan waktu 7.00, 12.00, dan 17.00 maka servo akan berputar untuk menuangkan pakan ke wadah yang sudah tersedia. Sedangkan ketika modul RTC DS3231 menunjukkan waktu 17.00 maka relay akan memicu lampu pijar untuk menyala. Ketika sensor DHT11 menunjukkan nilai suhu lebih dari 29°C - 31°C maka relay akan memicu kipas suhu supaya menyala untuk menurunkan suhu kandang. Saat sensor DHT11 menunjukkan nilai suhu kurang dari 29°C - 31°C maka relay akan memicu lampu pijar supaya menyala untuk menaikkan suhu kandang. Saat sensor DHT11 menunjukkan nilai kelembaban di bawah 50% - 70% maka relay akan memicu kipas kelembaban supaya menyala untuk menaikkan persentase kelembaban. Jika sensor DHT11 menunjukkan nilai kelembaban di atas 50% - 70% maka relay akan memicu lampu pijar supaya menyala untuk menurunkan persentase kelembaban. Ketika nilai sensor ketinggian air di bawah 30 maka relay akan memicu pompa air supaya menyala untuk mengisi bak minum.

Dari penelitian yang sudah dilakukan dengan judul "Purwarupa Kandang Ayam Pintar berbasis *Internet of Things (IoT)* menggunakan Platform Blynk" menghasilkan alat yang dapat berfungsi dengan baik sesuai dengan rancangan di tahap awal. Keseluruhan modul dan komponen yang digunakan mulai dari modul RTC DS3231, sensor ketinggian air, dan sensor DHT11 yang bisa memberikan data yang cukup akurat. Kemudian Arduino Uno dengan NodeMCU ESP8266 yang bisa melakukan komunikasi serial dengan lancar. Selain itu, NodeMCU ESP8266 juga dapat dengan mudah mengirimkan data kepada platform Blynk.

5.2 Saran

Dari penelitian yang sudah dilakukan oleh penulis, saran yang dapat diberikan untuk pengembangan penelitian ini ke depannya adalah sebagai berikut:

1. Purwarupa yang dibuat masih menggunakan listrik sebagai sumber daya utama. Hal ini akan sangat ~~merugikan~~ ketika tidak ada sumber listrik. Oleh karena itu diperlukan sumber daya utama selain listrik, salah satunya baterai supaya bisa mengisi kekurangan tersebut.
2. Hasil dari penelitian yang dilakukan masih dalam bentuk prototipe dan belum diterapkan ke kandang ayam yang sebenarnya. Di masa yang akan datang diharapkan hasil dari penelitian ini sudah bisa diterapkan dan digunakan ke kandang ayam yang sebenarnya.