

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas tentang persiapan, detail dari pembuatan, hasil pengujian proyek dan Analisa. Analisa dilakukan untuk mengetahui kelebihan dan kekurangan dari sistem *smart* jemuran yang telah dibuat.

4.1 Proses Pembuatan Sistem

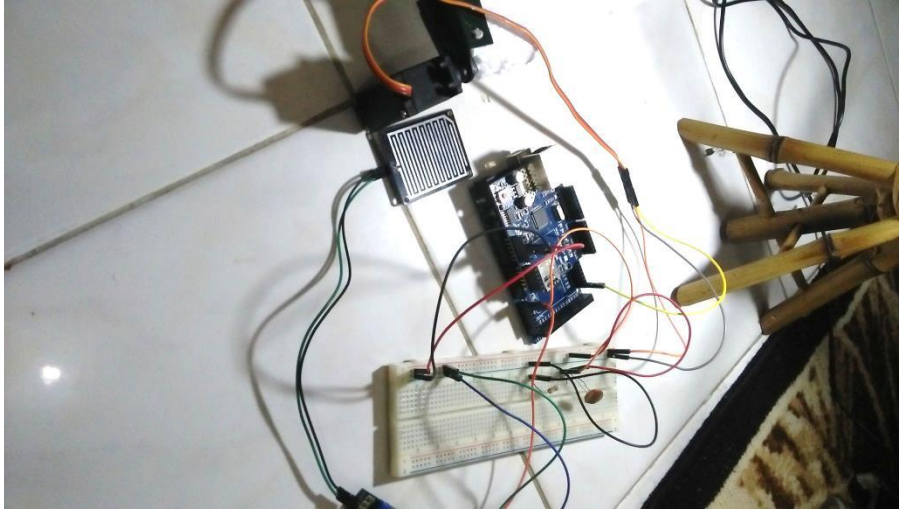
Proses pembuatan sistem terbagi menjadi empat tahapan yang perlu dilakukan, yaitu tahap persiapan, perangkaian komponen, penulisan kode program, pembuatan website pada webserver dan pengujian.

4.1.1 Persiapan

Tahap persiapan ialah tahap untuk mempersiapkan segala hal yang dibutuhkan dalam pembuatan sistem. Pertama yang harus dipersiapkan adalah komponen-komponen apa saja yang diperlukan dalam pembuatan sistem, kemudian dirangkai dan dihubungkan menjadi satu kesatuan agar membentuk sistem yang utuh. Komponen-komponen yang perlu dipersiapkan dalam pembuatan sistem *smart* jemuran adalah sebagai berikut:

- a. Mikrokontroler Arduino ATmega
- b. *Breadboard*
- c. *Jumper wire*
- d. *Arduino Ethernet Shield*
- e. Sensor LDR atau cahaya
- f. Sensor hujan atau air
- g. Servo
- h. Kabel USB

Rangkaian mikrokontroler Arduino yang sudah saling terhubung dengan *Ethernet shield*, sensor LDR, sensor air, dan servo menggunakan *jumper wire* melalui *breadboard* dapat dilihat pada Gambar 4.1. Rangkaian tersebut hanya awal untuk mendapat kode program sebelum diaplikasikan pada purwarupa nantinya.



Gambar 4.1 Rangkaian mikrokontroler



Gambar 4.2 Servo modifikasi

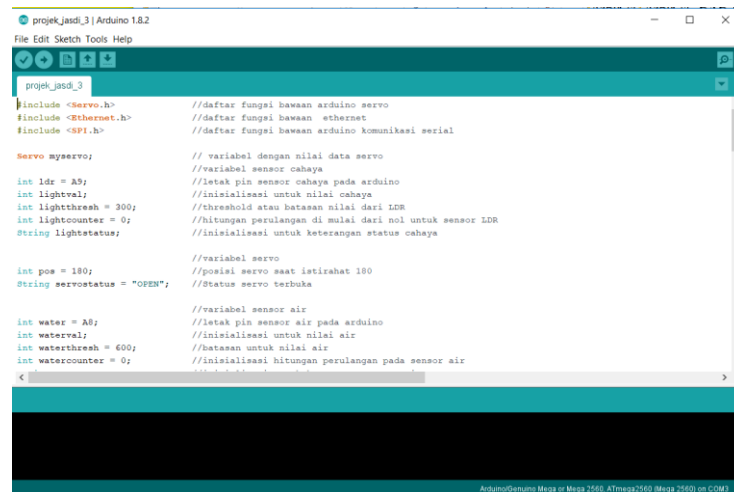
Gambar 4.2 adalah komponen servo yang telah dimodifikasi dengan sterofom sebagai atap kanopi penutup jemuran. Yang nanti akan melindungi jemuran dari hujan.

Persiapan yang kedua adalah persiapan pembuatan purwarupa yaitu rumah-rumahan. Purwarupa dibutuhkan untuk mensimulasikan sistem mikrokontroler yang digunakan. Penulis menggunakan bahan kayu triplek untuk membuat rumah-rumahan tersebut dan sterofom untuk membuat atap jemuran. Miniatur rumah dapat dilihat pada Gambar 4.3



Gambar 4.3 Miniatur purwarupa rumah-rumahan

Persiapan yang ketiga ialah mempersiapkan perangkat lunak. Perangkat lunak dibutuhkan untuk penulisan kode program yang akan diproses oleh mikrokontroler Arduino sesuai kondisi yang ada. Penulis menggunakan *software* IDE Arduino dan *driver* Arduino untuk memberikan perintah dan kondisi yang nantinya Arduino akan memprosesnya. Setelah melakukan instalasi IDE arduino, agar kode dapat dibaca/berjalan oleh Arduino maka perlu menentukan jenis Arduino yang digunakan pada menu *Tool – board*, di sini penulis menggunakan Arduino mega2560. Kemudian perlu menyesuaikan prosesor yang di gunakan pada IDE Arduino dan juga perlu menentukan *port* pada Arduino yang digunakan yang terletak pada menu *Tool-port*. *Port* yang digunakan untuk menghubungkan antara Arduino dengan server melalui *Ethernet shield* adalah *port* 80. Terakhir adalah menghubungkan rangkaian Arduino pada komputer dengan menggunakan kabel USB dan UTP sebagai penghubung data Arduino dengan webserver. Setelah persiapan ini IDE Arduino dapat digunakan untuk menuliskan kode program dan diunggah ke mikrokontroler Arduino untuk diproses.



```

projek_jasdi_3 | Arduino 1.8.2
File Edit Sketch Tools Help
projek_jasdi_3
#include <Servo.h> //daftar fungsi bawasan arduino servo
#include <Ethernet.h> //daftar fungsi bawasan ethernet
#include <SPI.h> //daftar fungsi bawasan arduino komunikasi serial

Servo myservo; // variabel dengan nilai data servo
//variabel sensor cahaya
int ldr = A9; //letak pin sensor cahaya pada arduino
int lightval; //inisialisasi untuk nilai cahaya
int lightthresh = 300; //threshold atau batasan nilai dari LDR
int lightcounter = 0; //hitungnng perulangan di mulai dari nol untuk sensor LDR
String lightstatus; //inisialisasi untuk keterangan status cahaya

//variabel servo
int pos = 180; //posisi servo saat istirahat 180
String servostatus = "OPEN"; //Status servo terbuka

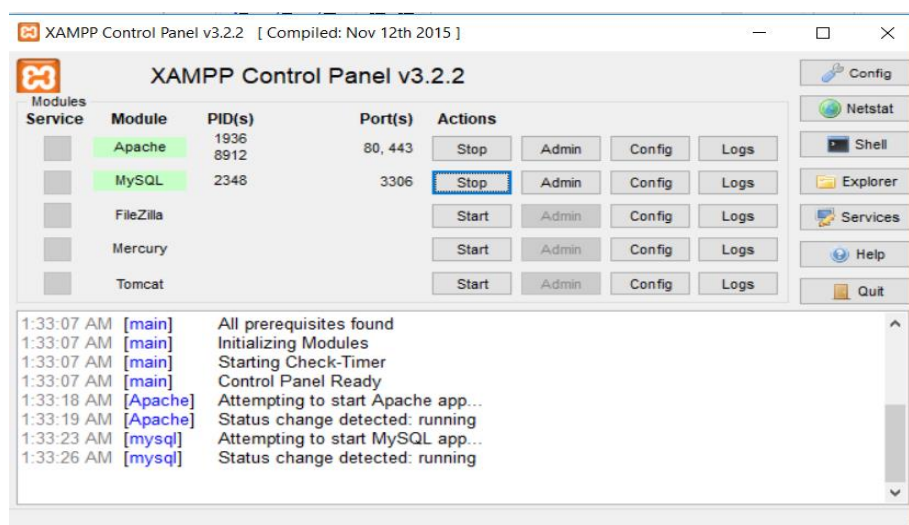
//variabel sensor air
int water = A8; //letak pin sensor air pada arduino
int waterval; //inisialisasi untuk nilai air
int watertresh = 600; //batasan untuk nilai air
int watercounter = 0; //inisialisasi hitungan perulangan pada sensor air

```

Gambar 4.4 Tampilan aplikasi Arduino IDE

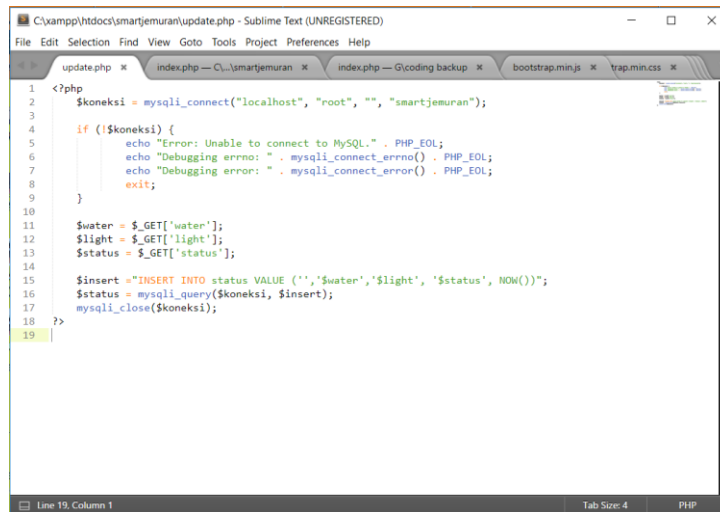
Penulisan program Arduino IDE menggunakan bahas C. Kode tersebut yang nanti akan dibaca dan diproses oleh Arduino untuk menjalankan fungsi dari komponen-komponen alat.

Persiapan yang terakhir adalah mempersiapkan *software* untuk penulisan kode website dan webserver pada konsep *smart* jemuran. Persiapan ini dibutuhkan untuk pemetukan website penampil data yang telah didapatkan dan diproses oleh mikrokontroler arduino. Penulis menggunakan sistem operasi *windows* 10 pada pembuatan sistem ini. Untuk membuat webserver penulis menggunakan aplikasi *xampp* yang membuat komputer menjadi *local server* untuk menyimpan data yang Arduino kirimkan dan sebagai jalan antara Arduino dengan komputer yang di sini sebagai lokal server. Kemudian aplikasi *Sublime text* digunakan untuk menuliskan kode pada tampilan website sistem *smart* jemuran ini.



Gambar 4.5 Antarmuka Xampp

Gambar 4.5 adalah tampilan dari aplikasi *Xampp*, pada *Xampp* mempunyai beberapa fungsi. Fungsi yang digunakan disini adalah fungsi dari *apache* untuk pembacaan kode php atau kode program website.



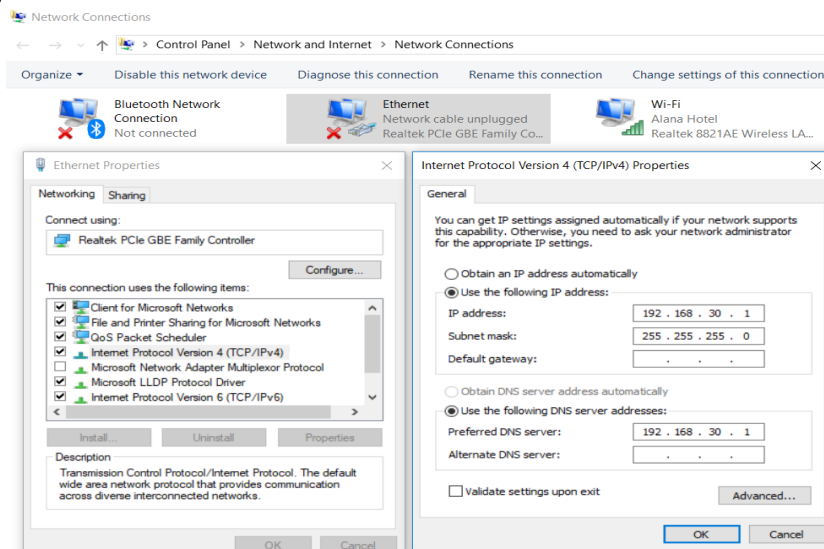
```

1 <?php
2 $koneksi = mysqli_connect("localhost", "root", "", "smartjemuran");
3
4 if (!$koneksi) {
5     echo "Error: Unable to connect to MySQL." . PHP_EOL;
6     echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;
7     echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;
8     exit;
9 }
10
11 $water = $_GET['water'];
12 $light = $_GET['light'];
13 $status = $_GET['status'];
14
15 $insert = "INSERT INTO status VALUE ('', '$water', '$light', '$status', NOW())";
16 $status = mysqli_query($koneksi, $insert);
17 mysqli_close($koneksi);
18
19 >

```

Gambar 4.6 Antarmuka Sublime Text

Setelah semua persiapan selesai, hal selanjutnya adalah memberi akses pada *Ethernet Shield* agar terhubung dengan server dengan memberikan IP pada *Ethernet* melalui kode program Arduino dan sesuaikan *gateway* dengan server yang sudah ditentukan alamat IP. Untuk memberikan IP pada *Ethernet Shield* dilakukan pada pengkodean program dan untuk setting IP pada server melalui *internet protocol properties* (TCP/ipv4), dapat dilihat pada Gambar 4.7



Gambar 4.7 Setting IP pada *local server*

Kemudian bentuk database pada server melalui *php myadmin* dengan menjalankan terlebih dahulu aplikasi *xampp* agar dapat membuat *database* untuk sever, yang nantinya database akan digunakan untuk memasukan nilai yang didapatkan oleh Arduino setelah pemproses dan nilai yang didapatkan Arduino bisa ditampilkan melalui website.

4.1.2 Perangkaian Komponen

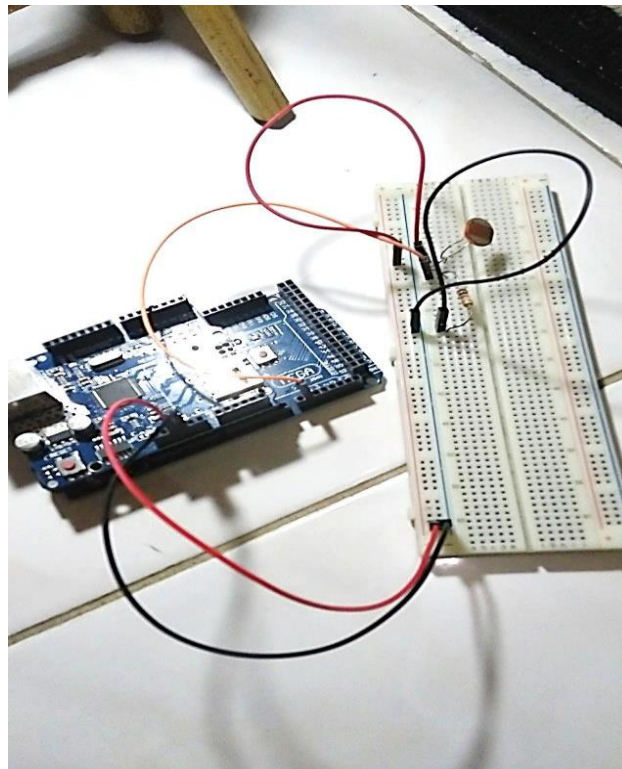
Proses perangkaian komponen sistem jemuran otomatis dimulai dengan menghubungkan segala komponen berupa sensor dan servo pada mikrokontroler, baru kemudian dipasang pada purwarupa dan disambungkan pada *local server* untuk ditampilkan pada website.

Hal pertama yang dilakukan yaitu perangkaian pada Arduino mega2560 dengan menggabungkan antara *Ethernet shield* dan Arduino menjadi satu, kemudian hubungkan Arduino dengan *breadboard*. Tujuan menggunakan *breadboard* adalah untuk mempermudah dan memberikan banyak slot pin pada arduino, cara menghubungkan Arduino dengan *breadboard* yaitu menggunakan *jumper wire* sebagai penghubung. Penulis mengelompokan warna *jumper* agar lebih mudah dalam pembagiannya, warna merah atau cerah penulis gunakan untuk *jumper* tegangan positif(+) atau 5v, warna hitam atau gelap penulis gunakan untuk *jumper ground(GND)* atau negatif(-), warna kuning atau mencolok penulis gunakan untuk pin digital atau analog.



Gambar 4.8 Arduino dengan *Ethernet Shield*

Penulis menghubungkan terlebih dahulu *jumper* merah pada pin 5v pada Arduino dengan positif(+) pada *breadboard*, gunakan *jumper* hitam untuk menghubungkan *ground*(GND) pada Arduino dengan tegangan negatif(-) pada *breadboard* dengan begitu *breadboard* akan dapat beroperasi karena sudah mendapatkan tegangan dari arduino. Kemudian menghubungkan sensor LDR pada *breadboard* dengan menggunakan *jumper*, untuk mengetahui bagian positif dan negatif dari sensor LDR, dengan cara memperhatikan panjang pendek kaki sensor dari kaki tersebut dapat diketahui bagian positif(+) dan negatif(-) dari sensor. Kaki yang panjang adalah bagian positif(+) dari sensor LDR dan yang pendek adalah bagian negatif(-), hubungkan bagian positif(+) LDR pada slot pin positif(+) *breadboard*. Pasang dahulu resistor pada *breadboard* dekat dengan *jumper* positif(+) LDR, letakkan bagian negatif(-) LDR pada kaki pertama resistor yang dipasang pada *breadboard*, kemudian hubungkan kaki kedua resistor tadi pada slot pin negatif(-) pada *breadboard*, penulis memberikan kabel *jumper* warna kuning pada bagian kaki pertama resistor pasang dan hubungkan langsung pada pin analog 9 atau A9 diarduino. Pemasangan *jumper* antara Arduino dengan *breadboard* dan sensor LDR dapat dilihat pada Gambar 4.9

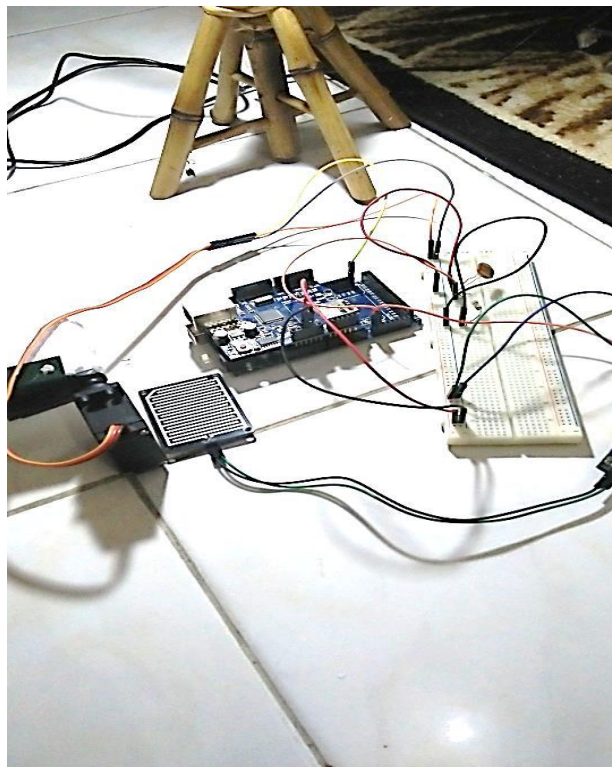


Gambar 4.9 Rangkaian komponen LDR

Langkah kedua adalah perangkaian komponen sensor air dan servo pada *breadboard*. Tidak jauh berbeda dengan sensor LDR, pada sensor air hanya memiliki 2 pin yang akan

dihubungkan dengan modul 2-4 dari sensor air tersebut. Modul tersebut merubah sensor air yang tadinya hanya 2 pin tanpat input data menjadi 4 pin dengan input data, gunakan 3 pin pada modul yaitu pin VCC, GND, dan AO. Penulis menghubungkan bagian VCC yang berarti 5v dengan *jumper* merah pada slot pin positif(+) *breadboard*, bagian GND yang berarti *ground* dengan *jumper* warna hitam pada slot pin negatif(-) pada *breadboard*, dan bagian AO yang berarti input data dengan *jumper* kuning pada slot pin 8 atau A8 pada arduino. Kemudian untuk menghubungkan servo, penulis membedakan kabel *jumper* pada servo, normalnya servo memiliki kabel 3 *jumper female* yang dipasangkan dengan kabel *jumper male-male*. Pada servo bagian *jumper* yang berwarna terang biasanya berwarna merah adalah 5v atau positif(+), bagian kabel *jumper* yang berwarna gelap berarti GND atau negatif(-) pada servo, dan kabel *jumper* yang berwarna mencolok itu biasanya adalah kabel input.

Penulis memasang kabel *jumper* positif(+) servo pada pin slot positif(+) di *breadboard*, dan menghubungkan *jumper* negatif(-) servo pada pin slot negatif(-) *breadboard*, lalu menghubungkan *jumper* input pada pin slot Arduino 21. Perangkat sensor air dan servo dapat dilihat pada Gambar 4.9



Gambar 4.10 Rangkaian komponen lengkap

Setelah semua langkah sudah selesai, penulis memasang seluruh rangkaian komponen pada purwarupa rumah-rumahan

4.1.3 Kode Program

Penulisan Kode program Arduino ditulis dengan menggunakan *software* IDE Arduino guna memberikan program perintah untuk pada arduino, dan untuk penulisan kode website menggunakan *Sublime text* guna memberikan *interface* pada data yang Arduino dapatkan.

Penulisan kode program Arduino sebagai berikut:

```
//variabel ethernet shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
//setting alamat mac ethernet shield
byte ip[] = { 192, 168, 30, 2 };
//setting alamat ip
byte gateway[] = { 192, 168, 30, 1 };
//setting alamat gateway
byte subnet[] = { 255, 255, 255, 0 };
//setting subnet network

//Ethernet CLient library
EthernetClient client; //ethernet sebagai client
unsigned long lastConnectionTime = 0; // koneski terakhir
ethernet pada server. dalam milisecond
const unsigned long delayInterval = 10L * 1000L; // delay
update pada ethernet shield dalam milisecond
char server[] = "192.168.30.1"; //alamat
ip server

void setup() //program utama
{
myservo.attach(21); // peletakan pin servo pada
arduino
Serial.begin(9600); //komunikasi serial
pinMode(ldr, INPUT); //mengakses data sensor ldr
pinMode(water, INPUT); //mengakses data sensor air
myservo.write(pos); //servo membaca posisi awal
Ethernet.begin(mac, ip, gateway, subnet); //memulai
komunikasi ethernet
delay(1000); // jeda waktu

}
```

Gambar 4.11 Contoh kode program pemberian alamat IP dan input servo.

Kode program pada Gambar 4.11 menjelaskan pembuatan alamat IP pada Ethernet Shield, IP yang digunakan adalah 192.168.30.1. Pada kode program koneksi antara arduino dengan lokal host diberikan jeda waktu sekitar 10 detik. Void setup() digunakan untuk menginisialisasi variabel-variabel yang akan digunakan, dan hanya dijalankan satu kali saat

Arduino mulai menyala. Fungsi `pinMode(pin, SET)` digunakan untuk menginisialisasi sebuah pin, dan menentukan pin tersebut akan digunakan sebagai input ataupun output.

```

void loop() //program perulangan
{
  if (millis() - lastConnectionTime > delayInterval) {

    lightval = analogRead(ldr);
    waterval = analogRead(water

//kondisi untuk penentuan perhitungan dan nilai data yang di
dapat
    if(waterval < waterthresh){
//kondisi, jika nilai air lebih kecil dari batas
      if(watercounter < 0){
//kondisi perulangan, jika perhitungan masukan data kurang
dari 0 atau -
        watercounter = 0; //
perulangan hitungan air reset 0/dimuali dari 0 dahulu
      }
      watercounter++; //perhitungan di tambah 1 atau +1
      waterstatus = "hujan"; //status [ada serial monitor
hujan
    }else{ //jika bukan
      if(watercounter > 0){ //jika perhitungan lebih besar
dari 0
        watercounter = 0; //perhitungan di reset jadi 0
      }
      watercounter--; //perhitungan di kurangi 1 atau -1
      waterstatus = "kering";
    }
    if(lightval < lightthresh){ //jika
nilai cahaya lebih kecil dari batas
      if(lightcounter < 0){ //jika
perhitungan cahaya mulai lebih kecil dari 0
        lightcounter = 0; //hitungan di reset 0
      }
      lightcounter++;
//hitungan di tambah +1
      lightstatus = "gelap";
//status pada serial monitor gelap
    }else{ //jika
bukan
      if(lightcounter > 0){ //jika
nilai hitungan cahaya lebih besar dari 0
        lightcounter = 0; //
hitungan di reset 0
      }
      lightcounter--; //hitungan di kurangi 1 atau -1
      lightstatus = "terang";
    }
  }
}

```

Gambar 4.12 Contoh kode program perulangan hitungan dan kondisi.

```

    }
        lightcounter--; //hitungan di kurangi 1 atau -1
        lightstatus = "terang"; //status terang
    }
    // penulisan pada serial monitor, nilai perhitungan air
    dan cahaya, status dari cahaya dan air
    Serial.println("Wcount
: "+(String)watercounter+"("+waterstatus+") Lcount
: "+(String)lightcounter+"("+lightstatus+")");

//kondisi untuk menggerakan servo
    if(watercounter >= 5||lightcounter>=5){
//kondisi jika hitungan air sudah menyampai lebih dari sama
dengan 5 atau hitungan cahaya lebih dari sama dengan 5
        watercounter = 0; //perhitungan air dari 0 lagi
        lightcounter = 0; //perhitungan cahaya dari 0 lagi
        myservo.write(15); //servo akan bergerak searah
jarum jam
        update(waterval,lightval,"CLOSE");
//update data nilai air dan cahaya .status tutup
        Serial.println("close");
//serial monitor memunculkan status atap close
        else if (watercounter <= -5 && lightcounter <= -5){
//kondisi jika bukan, hitungan air lebih kecil sama dengan -5
dan nilai hitungan cahay lebih kecil sama dengan -5
        watercounter = 0; //nilai di hitungan air di reset 0
        lightcounter = 0; //nilai hitungan cahaya di reset 0
        myservo.write(175); //servo bergerak berlawanan arah
jarum jam
        update(waterval, lightval,"OPEN"); //update data nilai
air dan cahaya status buka
        Serial.println("open");//serial monitor memunculkan
status open
        }
        Serial.println("Water : "+(String) waterval+" Light : "+
(String) lightval); //serial monitor memunculkan nilai
dari cahaya dan air

    }

    delay(1L * 1000L);
//jeda waktu 1detik
}

```

Gambar 4.13 Contoh kode program perhitungan nilai 5 dan pengaturan servo.

Pada Gambar 4.12 dan Gambar 4.13 menjelaskan fungsi loop() berguna untuk melaksanakan / mengeksekusi perintah program yang telah dibuat. Fungsi ini akan secara aktif mengontrol board Arduino baik membaca input atau merubah output. Fungsi loop()

dijalankan terus menerus selama board arduino hidup. Fungsi ini adalah fungsi perulangan untuk pada setiap kondisi intensitas yang diterima oleh sensor dan akan diproses oleh Arduino. Kode program pada intinya dibuat jika intensitas tersebut kurang atau melebihi batas *threshold* Arduino akan memproses *output* pada servo sesuai program yang telah dibuat.

```

    }
    delay(1L * 1000L);
//jeda waktu 1detik
}
void update(int waterval, int lightval, String status){
//program update pada server, variabel nilai cahaya, air dan
status

    client.stop();
//client arduino berhenti

    // jika koneksi berhasil
    if (client.connect(server, 80)) {
//jika client arduino terhubung dengan server pada port 80
        Serial.println("connecting...");
//serial monitor memunculkan status terkoneksi
        client.println("GET
/smartjemuran/update.php?water="+ (String)
waterval+"&light="+ (String) lightval+"&status="+status+"
HTTP/1.1"); // client arduino akses pada server nilai
air dan cahaya pada URL melalui method get
        client.println("Host: 192.168.30.1");
//pada alamat ip
        client.println("User-Agent: arduino-ethernet");
//keterangan akses atas nama arduino ethernet
        client.println("Connection: close");
//koneksi di tutup agar mengganggu koneksi baru
        client.println();
        lastConnectionTime = millis();
//waktu koneksi terakhir sama dengan waktu koneksi saat ini
dalam milisecond
    } else {
        // jika koneksi gagal
        Serial.println("connection failed");
// serial monitor memunculkan koneksi gagal
    }
}

```

Gambar 4.14 Contoh kode program *update* pada lokal server.

Gambar 4.14 menjelaskan data yang diterima arduino akan diupdate ke website untuk informasi yang diterima oleh arduino menggunakan fungsi void update(). Fungsi void update adalah untuk meneruskan data yang diterima oleh Arduino kepada jaringan yang tersedia melalui perantara *Ethernet Shield* atau perangkat jaringan lain pada Arduino.

```

    if (!$koneksi) {
        echo "Error: Unable to connect to MySQL." .
PHP_EOL;
        echo "Debugging errno: " .
mysqli_connect_errno() . PHP_EOL;
        echo "Debugging error: " .
mysqli_connect_error() . PHP_EOL;
        exit;
    }

    $water = $_GET['water'];
    $light = $_GET['light'];
    $status = $_GET['status'];

    $insert = "INSERT INTO status VALUE
('','$water','$light', '$status', NOW())";
    $status = mysqli_query($koneksi, $insert);
    mysqli_close($koneksi);
?>

```

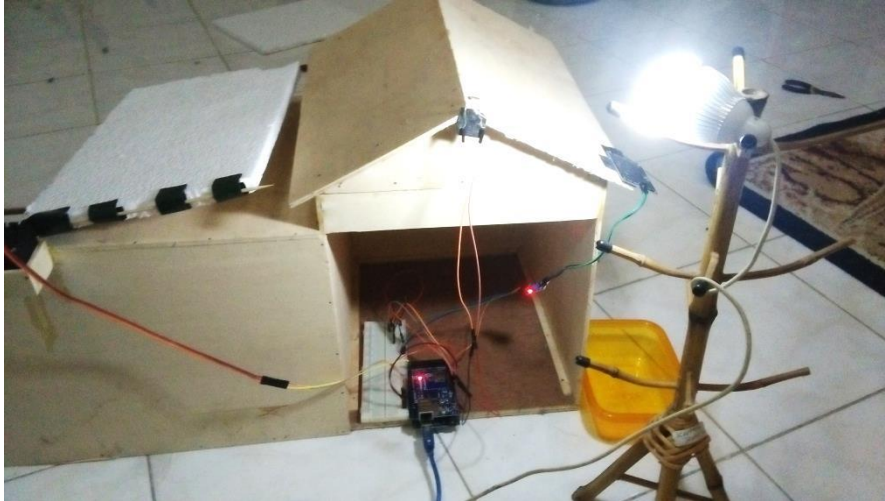
Gambar 4.15 Contoh kode program *update* pada *database*.

Gambar 4.15 menjelaskan kode program pada website. Fungsi tersebut untuk mendapatkan data dari Arduino dan akan ditampilkan pada website. Fungsi GET untuk mengambil data, pada kode program tersebut data yang diambil adalah nilai dari air, cahaya, dan status yang didapat sesuai hasil dari kode program pada Arduino. Fungsi INSERT untuk memasukan data kedalam database yang sudah dibuat.

4.2 Pengujian

Tahap pengujian ialah tahap dimana semua komponen telah dirangkai, kode program sudah siap untuk dioperasikan, dan website pada *local server* sudah dapat menerima data dari arduino. Pengujian ini dilakukan dengan tujuan agar mengetahui sistem sudah beroperasi sesuai dengan perintah atau tidak. Dalam tahap pengujian akan dibuat beberapa skenario layaknya hujan dan beberapa kondisi.

Sebelum melakukan pengujian perlu adanya pengecekan pada setiap komponen dan kabel *jumper* pada mikrokontroler arduino. Pengujian dilakuan dengan sensor yang sudah terpasang pada miniatur rumah dengan pengoperasian oleh mikrokontroler arduino. Sensor akan dipasangkan pada atap miniatur rumah dan servo sebagai poros atap kanopi untuk jemuran.



Gambar 4.16 Rangkaian sudah diposisikan pada miniatur rumah

Gambar 4.16 adalah gambar rangkaian dari mikrokontroler yang semua sensor sudah diletakan pada posisinya, dikarenakan kabel jumper tidak menjangkau *breadboard*, penulis melakukan *extension* pada kabel jumper. Pada Gambar 4.16 dapat dilihat ada sebuah lampu, lampu tersebut diskenariokan sebagai matahari. Lampu tersebut bukan merupakan satu rangkaian dari mikrokontroler, lampu tersebut hanya untuk membantu membuat sensor mendapatkan nilai cahaya berdasarkan kondisi terang.



Gambar 4.17 Skenario kondisi terang

Gambar 4.17 adalah skenario pada saat sedang menjemur selimut hotel pada terik matahari, terdapat banyak cahaya. Skenario tersebut adalah kondisi pada saat terang dan tidak

ada hujan, servo terletak pada posisi yang diinginkan dengan membuka atap kanopi jemuran. Karena cahaya yang didapatkan lebih besar dari batas servo akan beroperasi membuka kanopi, setiap kondisi yang terjadi akan dilakukan pengecekan ulang sebanyak 5 hitungan. Jika kondisi tetap sama hingga hitungan ke 5 servo akan beroperasi bergerak sesuai kondisi yang didapat saat terang, gelap, hujan, kering. Jika kondisi tidak terpenuhi, mikrokontroler akan melakukan pengecekan ulang dari hitungan 0 hingga mendapatkan nilai kondisi yang sama hingga 5 kali.

Skenario tersebut adalah skenario untuk melakukan pengecekan pada nilai yang didapatkan. Tujuan skenario tersebut dimaksudkan untuk mengatasi skenario jika datang bayangan awan yang menutupi cahaya atau sinar cahaya yang datang lalu pergi dan skenario ada tetesan air yang terkena pada sensor air dan juga untuk membedakan hujan yang hanya sementara atau hujan yang datang terus menerus. Dengan begitu mikrokontroler akan lebih akurat dalam pengoperasian pada servo.



Gambar 4.18 Skenario pada kondisi gelap

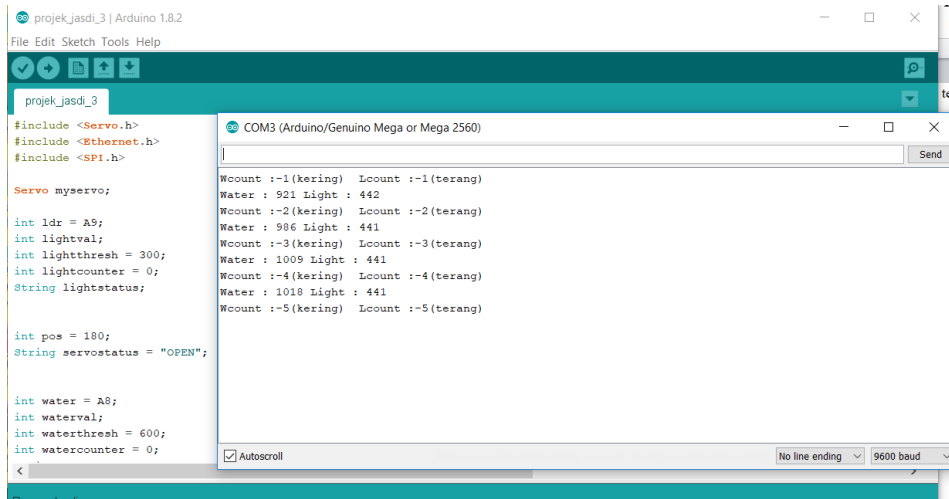
Gambar 4.18 adalah gambar skenario dimana kondisi tidak ada matahari. Kondisi pada saat tidak ada cahaya dan tidak ada hujan yang dapat diartikan sebagai mendung, untuk mengantisipasi hujan datang mikrokontroler Arduino memproses servo agar mengerjakan atap kanopi untuk menutupi jemuran. Proses dilakukan setelah pengecekan ulang sebanyak 5 kali, jika nilai cahaya lebih kecil dari batas. Baru setelah itu jika kondisi tetap sama servo akan bergerak menutup jemuran



Gambar 4.19 Skenario hujan dengan ada cahaya

Gambar 4.19 menggambarkan skenario pada saat hujan turun tetapi ada sinar matahari. Kondisi tersebut diartikan pada saat sensor menerima masukan nilai air lebih kecil dari batas nilai tetapi nilai cahaya yang didapatkan lebih besar dari pada batas, kondisi tersebut mutlak membuat Arduino memproses servo agar bergerak menutup jemuran supaya jemuran tidak basah terkena air. Tetap pada kondisi tersebut mikrokontroler akan melakukan pengecekan sebanyak 5 kali untuk mengantisipasi hujan sementara atau hujan yang berkelanjutan dan juga untuk mengantisipasi jika hujan reda tetapi masih ada tetesan air yang tertinggal pada sensor air. Karena ada jeda waktu saat menunggu kering sama halnya dengan Arduino memastikan apakah masih terjadi hujan atau sudah berhenti, maka sensor air diatur kemiringannya agar cepat mengering. Karena diatur kemiringannya sensor akan lebih pasti memastikan hujan karena tidak berhenti meneteskan air dengan intensitas yang banyak.

Pada sensor air nilai intensitas pada saat sensor dalam keadaan kering, sensor mendapatkan banyak tegangan jadi jika tegangan yang didapatkan sensor lebih kecil dari batas berarti sensor terkena basah air. Pada sensor LDR jika nilai cahaya yang didapat lebih besar dari pada batas, itu berarti sensor mendapatkan banyak cahaya yang membuat tegangan yang dihasilkan tinggi atau kondisi terang.



The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the following code:

```
#include <Servo.h>
#include <Ethernet.h>
#include <SPI.h>

Servo myservo;

int ldr = A9;
int lightval;
int lightthresh = 300;
int lightcounter = 0;
String lightstatus;

int pos = 180;
String servostatus = "OPEN";

int water = A8;
int waterval;
int waterthresh = 600;
int watercounter = 0;
```

The serial monitor on the right displays the following output:

```
Wcount :-1(kering) Lcount :-1(terang)
Water : 921 Light : 442
Wcount :-2(kering) Lcount :-2(terang)
Water : 986 Light : 441
Wcount :-3(kering) Lcount :-3(terang)
Water : 1009 Light : 441
Wcount :-4(kering) Lcount :-4(terang)
Water : 1018 Light : 441
Wcount :-5(kering) Lcount :-5(terang)
```

Gambar 4.20 Nilai dari kedua sensor dan batasan nilai


Gambar 4.21 adalah skenario gambaran dari datangnya hujan dan mendung, yang berarti sensor cahaya tidak mendapatkan nilai cahaya di atas batasan nilai cahaya dan sensor air menerima masukan nilai air yang di bawah batas nilai air. Maka Arduino akan langsung melakukan pengecekan ulang sebanyak 5 kali, jika nilai pengecekan yang didapat tetap sama sesuai batasan hingga hitungan ke 5 Arduino akan memproses servo agar menutup atap kanopi jemuran.



Gambar 4.21 Skenario pada saat mendung dan hujan

Setiap nilai yang didapat pada Arduino dengan kondisi yang ada, dapat dilihat atau ditampilkan oleh Arduino di *serial monitor* Arduino IDE. Dengan maksud agar penulis dapat melakukan pengecekan pada kode program yang berjalan. Seperti pada Gambar 4.22 yang menunjukkan *status* dan nilai yang didapatkan pada masing-masing sensor. Pada sensor air nilai tegangan di atas angka 1000 dengan nilai batas 600 yang berarti statusnya kering dan

pada sensor cahaya nilai tegangan di bawah angka batas yaitu 300 yang berarti nilai kecil, *status* yang didapat gelap. Kondisi tersebut dapat diartikan sebagai skenario pada saat tidak ada hujan dan tidak ada cahaya yang artinya adalah mendung. Pada *serial monitor* juga dapat dilihat perulangan yang dilakukan setiap 5 kali hitungan, jika hitungan sudah mendapatkan nilai 5, Arduino akan memberikan status servo terbuka atau tertutup.

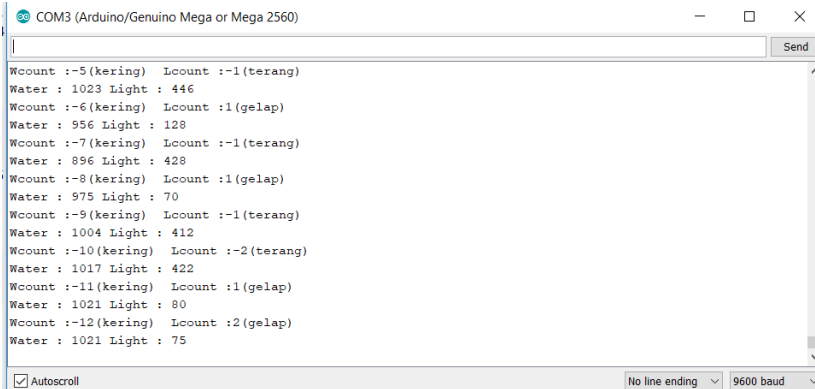


```

COM3 (Arduino/Genuino Mega or Mega 2560)
close
Water : 1019 Light : 251
Wcount :-1(kering) Lcount :1(gelap)
Water : 1004 Light : 210
Wcount :-2(kering) Lcount :2(gelap)
Water : 1015 Light : 211
Wcount :-3(kering) Lcount :3(gelap)
Water : 1019 Light : 211
Wcount :-4(kering) Lcount :4(gelap)
Water : 1020 Light : 211
Wcount :-5(kering) Lcount :5(gelap)
connecting...
close
Water : 1021 Light : 210
Wcount :-1(kering) Lcount :1(gelap)
Water : 1020 Light : 207
Autoscroll No line ending 9600 baud

```

Gambar 4.22 *Serial monitor* menampilkan status dan nilai yang di dapat



```

COM3 (Arduino/Genuino Mega or Mega 2560)
Wcount :-5(kering) Lcount :-1(terang)
Water : 1023 Light : 446
Wcount :-6(kering) Lcount :1(gelap)
Water : 956 Light : 128
Wcount :-7(kering) Lcount :-1(terang)
Water : 896 Light : 428
Wcount :-8(kering) Lcount :1(gelap)
Water : 975 Light : 70
Wcount :-9(kering) Lcount :-1(terang)
Water : 1004 Light : 412
Wcount :-10(kering) Lcount :-2(terang)
Water : 1017 Light : 422
Wcount :-11(kering) Lcount :1(gelap)
Water : 1021 Light : 80
Wcount :-12(kering) Lcount :2(gelap)
Water : 1021 Light : 75
Autoscroll No line ending 9600 baud

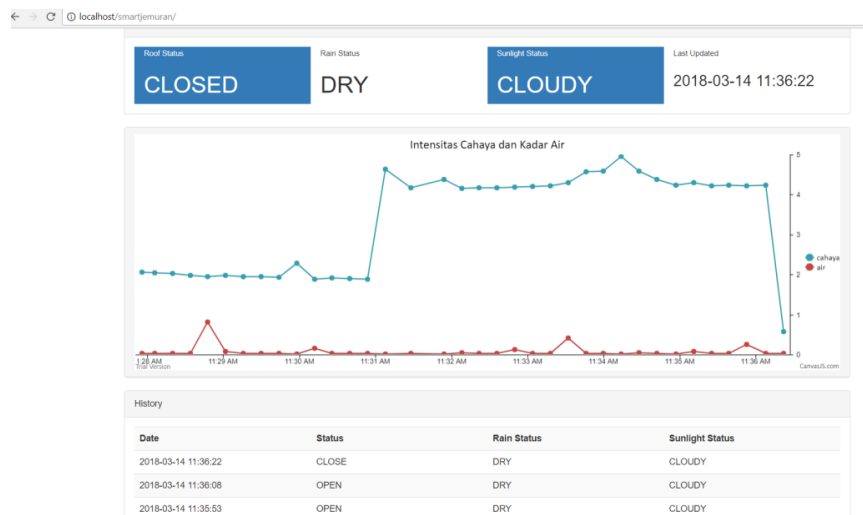
```

Gambar 4.23 *Serial monitor* perulang hitungan

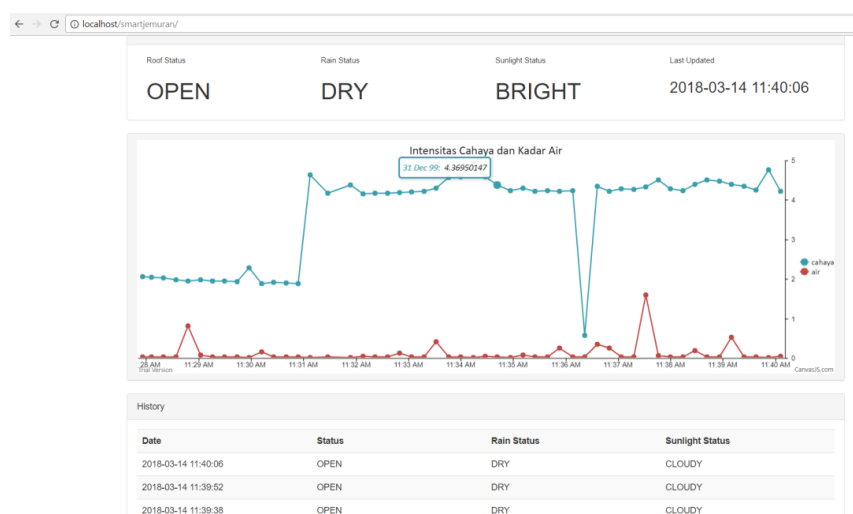
Gambar 4.23 menunjukkan skenario perulangan hitungan pengecekan tidak menyampai nilai 5, di dalam serial monitor hitungan dikembalikan terlebih dahulu menjadi nol agar hitungan tidak terlalu menumpuk. Jika hitungan sudah di *reset* menjadi 0, hitungan selanjutnya akan dibaca sesuai kondisi akan +1 atau -1. Pada Gambar 4.18 hitungan yang didapat pada kondisi gelap hitungannya akan mengarah hingga 5 dan pada kondisi kering nilainya akan mengarah pada -5. Jika sebelum hitungan ke 5 ada nilai kondisi terang, hitungan akan menjadi -1 hingga -5 dan pada sensor air jika mendapatkan nilai kondisi basah tiba-tiba, hitungan akan ditambah mulai dari +1 hingga 5. Selama kondisi pengecekan belum

terpenuhi servo tidak akan bergerak, servo akan terus diposisi terakhir. Karena hal itu pengecekan dilakukan 5 kali setiap 1 detik, menurut penulis itu cukup untuk mengatasi jika ada cahaya atau air yang datang lalu pergi sebab dapat menyebabkan servo bergerak terus menerus sesuai kondisi.

Setelah semua proses telah dilakukan, semua nilai yang didapatkan Arduino dapat dilihat pada website, penyampaian data melalui *Ethernet Shield* menuju *local server* untuk ditampilkan pada website. Pada Gambar 4.24 dan Gambar 4.25 adalah gambar yang menunjukkan nilai data pada kondisi yang didapat oleh Arduino setelah hitungan perulangan ke 5, di *update* ke website untuk ditampilkan. Melalui website dapat dilihat rekam waktu, *status* dan nilai yang didapat oleh arduino.



Gambar 4.24 Tampilan *interface* saat kondisi mendung



Gambar 4.25 Antarmuka website saat kondisi terang

Setelah semua tahapan pengujian telah dilakukan dapat ditarik kesimpulan bahwa keseluruhan sistem berjalan dengan baik sesuai dengan rancangan dan konsep awal yang telah dibuat. Hasil pengujian dapat dilihat pada Tabel 4.1

Tabel 4.1 Tabel hasil pengujian

No	Sensor Air	Sensor LDR	Servo	Hasil
1	Tidak ada air	Tidak ada cahaya	Menutup	Bekerja
2	Ada air	Tidak ada cahaya	Menutup	Bekerja
3	Tidak ada air	Ada cahaya	Membuka	Bekerja
4	Ada air	Ada cahaya	Menutup	Bekerja

Sebelum sistem berjalan dengan baik, penulis melakukan beberapa pengujian pada berbagai komponen dan nilai. Berikut uraian tahapan pengujian yang dilakukan penulis.

Penulis menggunakan sensor LDR yang menggunakan *module*, pada sensor LDR yang menggunakan *module* nilai yang didapat hanya ada 2 kemungkinan 0 atau 1, sama dengan hanya mendapatkan nilai cahaya 1023 atau dibawah 100. Karena hal itu penulis menggunakan sensor LDR tanpa *module* yang ditambahkan dengan resistor, dari situ penulis mendapatkan nilai yang beragam. Setelah nilai didapatkan penulis mencoba mengkonversi satuan pada cahaya dari tegangan menjadi lux, dengan mencari nilai-nilai cahaya pada luxmeter dan multimeter. dikarenakan tidak ada ilmu pasti tentang merubah nilai tegangan menjadi lux maka penulis hanya menggunakan nilai tegangan sebagai acuan nilai yang didapat.

Penulis juga mencoba melakukan percobaan pada servo. Pada servo yang digunakan penulis memiliki batasan putaran yaitu hanya pada 0 derajat – 360 derajat, servo penulis tidak bias melakukan putaran lebih dari 360 derajat jika ingin membuat atap bergeser diperlukan putaran lebih dari 360 derajat.

4.3 Analisa Kelebihan dan Kekurangan Sistem

Dari tahapan pengujian sebelumnya yang telah dilakukan beberapa tahapan bagaimana sistem beroperasi. Dengan mengetahui bagaimana sistem berjalan maka dapat dianalisa kelebihan dan kekurangan dari sistem ini.

4.3.1 Kelebihan Sistem

Setiap sistem tentunya memiliki kelebihannya masing-masing, setelah melakukan beberapa tahapan pengujian pada sistem *smart* jemuran, maka didapatkan beberapa kelebihan dari sistem yang sudah dibangun, yaitu:

- a. Keseluruhan sistem dapat dikendalikan secara otomatis tanpa membutuhkan tenaga dari manusia untuk mengoperasikannya.
- b. Alat yang digunakan adalah mikrokontroler Arduino yang relatif mudah untuk dikembangkan.
- c. Sistem memiliki rekap waktu alat bekerja yang bias menjadi acuan cuaca yang didapat.
- d. Website memberikan informasi sesuai kondisi yang ada.

4.3.2 Kekurangan Sistem

Setelah dilakukan beberapa tahapan pengujian sistem smart jemuran ini, maka didapatkan beberapa kekurangan dari sistem yang sudah dibangun, yaitu:

- a. Sensor air sangat susah mengering jika ditetaskan air pada jumlah sedikit dan nilai tegangan yang didapatkan tidak akurat, dengan jumlah tetesan yang tetap dan masih melekat pada sensor nilainya berubah-ubah secara drastis. Oleh karena itu sebetulnya butuh bantuan untuk mengeringkan secara cepat jika tidak ingin terjadi kesalahan kondisi. Solusi sejauh ini hanya bisa didapatkan melalui posisi kemiringan agar sisa air mengalir.
- b. Penetesan air hanya menggunakan pipet jadi banyak jeda nilai yang tak beraturan karena posisi sensor miring. Yang terkadang membuat kondisi yang didapat salah.
- c. Karena keterbatasan servo berputar, servo hanya bisa berputar 360 derajat.
- d. Nilai yang didapat bukan berdasarkan nilai acuan khusus seperti lux atau intensitas air yang ada, tetapi nilai tegangan yang didapatkan sensor sudah cukup menjadi acuan. Sehingga membuat grafik pada website tidak berdasarkan nilai yang sebenarnya yaitu nilai lux dan nilai air.
- e. Grafik pada website tidak menunjukkan waktu yang tepat, dikarenakan grafik menggunakan *plug-in* yang tersedia di mana *legend* pada waktu tidak dapat dirubah. Sehingga waktu yang ditunjukkan hanya didapatkan setiap selang waktu beberapa menit. Tujuan grafik tersebut sebetulnya hanya untuk mempermudah pengguna untuk melihat data yang banyak.

Tabel 4.2 Tabel hasil kekurangan

Kendala	Penyebab	Usulan Solusi
Tegangan pada sensor air sering berubah-ubah secara drastis	Karena masih terdapat tetesan air pada sensor dan juga sensor susah mengering sendirinya	Mengatasi dengan tingkat kemiringan sensor dan batasan nilai air pada program
Grafik pada website tidak sesuai dengan nilai asli dari intensitas cahaya dan air	Karena tidak ada ilmu pasti untuk merubah nilai tegangan menjadi lux untuk intensitas cahaya, dan pada air belum ada satuan yang pasti pada air	Nilai masih berupa tegangan yang didapat oleh sensor
Fleksibilitas atap masih kurang atau pengaplikasian pergerakan atap kurang memperhitungkan beban nantinya	Karena servo yang digunakan hanya bias berputar 180-360 derajat	Pengaplikasian atap terpaksa harus mengangkat atap untuk membuka atau menutup
Waktu pada grafik di website kurang tepat	Karena <i>legend</i> waktu pada <i>plug-in</i> tidak dapat dirubah	Data yang diterima, didapatkan selang beberapa menit agar data yang didapat tidak terlalu banyak