

BAB 1

PENDAHULUAN

A. Latarbelakang

Petani merupakan seseorang yang bergerak dibidang pertanian utamanya dengan cara melakukan pengelolaan tanam dengan tujuan untuk menumbuhkan dan memelihara tanaman. Dalam hal bercocok tanam, media tanam merupakan hal utama yang diperlukan. Mulai dari petani pengguna media tanam tanah hingga air atau biasa disebut dengan Hidroponik. Media tanam tanah tidak mengalami banyak masalah namun berbeda dengan media tanam hidroponik yang mengalami masalah dalam pengairan. Dari kedua media tanam tersebut, media tanam hidroponik yang sering mengalami masalah untuk pengairan. Pada saat tinggi air untuk media tanam berkurang banyak dari petani yang harus mengairinya dengan system manual. Hal ini menghambat pertumbuhan tanaman. Tanaman yang di tanam pada hidroponik biasanya tanaman sayur. Jenis sayur yang banyak mengalami masalah dalam penanamannya ialah sayur Sawi.

Pada tanaman sawi kekurangan air menyebabkan tanaman menjadi tidak tumbuh subur dan tumbuh dengan lambat. Karena air berperan sebagai media utama pada penanaman hidroponik. Sehingga kadar air untuk media tanamnya juga harus tetap terjaga dan stabil sesuai kebutuhan tanaman. Jika hal ini terus dibiarkan maka petani hidroponik sayur sawi mengalami banyak kerugian dan menghabiskan banyak waktu untuk pengecekan masalah pengairan.

Untuk mengatasi masalah pengairan pada hidroponik tanaman sawi digunakan sebuah sensor kelembapan dengan cara kerja untuk mengukur suhu kelembapan tanah sehingga petani mampu memperoleh kelembapan pada media tanam sawi. Sensor kelembapan tanah akan bekerja bersamaan dengan alat siram otomatis menggunakan IOT(internet of thing). Dengan IOT(internet Of Thing) penyiraman otomatis pada tanaman hidroponik

sayur sawi akan lebih cepat terselesaikan dan sangat membantu petani hidroponik dalam masalah pertaniannya.

B. Identifikasi Masalah

Dari latar belakang yang dijelaskan di atas dapat diidentifikasi beberapa masalah :

1. Media air pada tanaman sayur sawi hidroponik kurang terkontrol.
2. Kurangnya pengetahuan petani sayur terhadap waktu penambahan air pada media tanam sayuran sawi.
3. Pengairan system manual menyebabkan sebagian dari tanaman sawi hidroponik kurang terawat.

C. Maksud dan Tujuan

Maksud dari penelitian ini adalah sebagai berikut : Membuat system penyiraman otomatis menggunakan IOT untuk penyiraman pada tanaman hidroponik sayur sawi, dimulai dari mendeteksi media tanam pada tanaman sawi menggunakan sensor suhu kelembapan tanah.

Tujuan Penelitian :

1. Menstabilkan media tanam hidroponik tanaman sawi.
2. Membantu pengairan otomatis pada tanaman hidroponik sayur sawi.
3. Mempercepat pengecekan kekurangan air pada media tanam hidroponik sayur sawi.

D. Batasan Masalah

Agar dalam pembahasannya lebih terarah dan sesuai dengan tujuan yang akan dicapai maka diperlukan batasan masalah. Adapun batasan masalah dalam pelaksanaan penelitian ini adalah :

1. Membangun prototype system IOT untuk penyiraman otomatis hidroponik tanaman sawi dengan mengatur media tanam menggunakan sensor suhu kelembapan tanah.
2. IOT untuk petani hidroponik sawi.

3. IOT untuk penyiraman hidroponik sayur sawi dengan mengatur media tanam hidroponik sayur sawi.

BAB II

KAJIAN TOERI

A. DASAR TEORI

1. Hidroponik Sawi

Hidroponik merupakan budidaya menanam dengan memanfaatkan air tanpa menggunakan tanah dengan menekankan pada pemenuhan kebutuhan nutrisi bagi tanaman. Kebutuhan air pada hidroponik lebih sedikit daripada kebutuhan air pada budidaya dengan tanah. Hidroponik menggunakan air yang lebih efisien, jadi cocok diterapkan pada daerah yang memiliki pasokan air yang terbatas. Menanam tanaman dengan sistem hidroponik merupakan suatu metoda yang ramah lingkungan. Karena dalam pembudidayaannya tidak perlu menggunakan pestisida atau bahkan herbisida yang beracun. Meskipun sistem hidroponik menggunakan air sebagai media tanamnya akan tetapi dalam prakteknya air yang diperlukan dalam bercocok tanam tidaklah sebanyak seperti budidaya dengan cara konvensional. Dan dalam perawatannya juga tidak perlu dilakukan penyiraman secara rutin. Sehingga ini menjadi faktor mengapa hasil tanamannya lebih aman dan sehat.



Sawi merupakan salah satu sayuran yang banyak digemari oleh masyarakat. Dalam prosesnya tanaman sawi hanya memerlukan perhatian yang biasa saja, namun satu hal yang perlu diperhatikan adalah disaat proses tumbuh berkembangnya tanaman sawi sistem pengairan tidak boleh tersendat. Karena jika itu terjadi maka tanaman sawi tidak akan tumbuh dengan baik dan mengalami hasil panen yang tidak berkualitas baik sesuai pasaran.

2. Firebase

Firestore Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis. Firestore Realtime Database memungkinkan Anda untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data disimpan di drive lokal. Bahkan saat offline sekalipun, peristiwa realtime terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, Realtime Database akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama klien offline, sehingga setiap perbedaan akan otomatis digabungkan. Realtime Database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau disebut juga Aturan Keamanan Firestore Realtime Database, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firestore Authentication, developer dapat menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya. Realtime Database adalah database NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait. API Realtime Database dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda untuk membangun pengalaman realtime yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut.



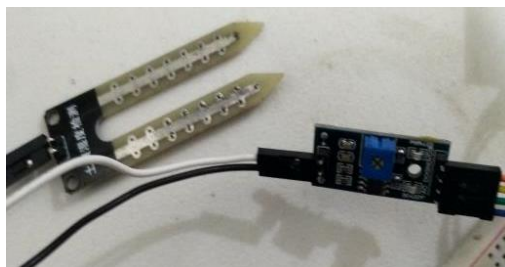
3. NodeMCU ESP8266

NodeMCU pada dasarnya adalah pengembangan dari ESP 8266 dengan firmware berbasis e-Lua. Pada NodeMcu dilengkapi dengan micro usb port yang berfungsi untuk pemrograman maupun power supply.



Selain itu juga pada NodeMCU dilengkapi dengan tombol push button yaitu tombol reset dan flash. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan package dari esp8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan c hanya berbeda syntax. Jika menggunakan bahasa Lua maka dapat menggunakan tool Lua loader maupun Lua uploader. Selain dengan bahasa Lua NodeMCU juga support dengan software Arduino IDE dengan melakukan sedikit perubahan board manager pada Arduino IDE. Sebelum digunakan Board ini harus di Flash terlebih dahulu agar support terhadap tool yang akan digunakan. Jika menggunakan Arduino IDE menggunakan firmware yang cocok yaitu firmware keluaran dari AiThinker yang support AT Command. Untuk penggunaan tool loader Firmware yang digunakan adalah firmware NodeMCU.

4. Soil Moisture Sensor

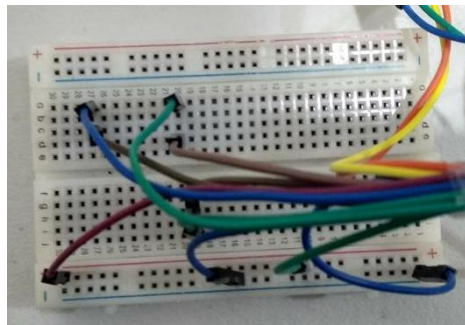


Moisture sensor adalah sensor kelembaban yang dapat mendeteksi kelembaban dalam tanah. Sensor ini sangat sederhana, tetapi ideal untuk memantau taman kota, atau tingkat air pada tanaman pekarangan anda. Sensor ini terdiri dua probe untuk melewati arus melalui tanah, kemudian membaca resistansinya untuk mendapatkan nilai tingkat kelembaban. Semakin banyak air membuat tanah lebih

mudah menghantarkan listrik (resistansi kecil), sedangkan tanah yang kering sangat sulit menghantarkan listrik (resistansi besar). Sensor ini sangat membantu Anda untuk mengingatkan tingkat kelembaban pada tanaman Anda atau memantau kelembaban tanah di kebun Anda. IO Expansion Shield adalah shield yang sempurna untuk menghubungkan Sensor dengan Arduino.

5. Project Board

BreadBoard atau disebut juga dengan project board adalah dasar



konstruksi sebuah sirkuit elektronik yang merupakan bagian prototipe dari suatu rangkaian elektronik yang belum disolder sehingga masih dapat dirubah skema atau pengantian komponen.

Jenis-jenis breadboard ditentukan berdasarkan banyak lubang yang terdapat pada papan itu, misal breadboard 400 lubang, 170 lubang, dan lain sebagainya. Hal terpenting yang harus diketahui sebelum menggunakan project board ini yaitu memahami dengan baik bagaimana jalur yang saling terhubung antara satu lubang dengan lainnya. **Project board** ini cocok digunakan untuk tahap awal develop project rangkaian elektronika. Merakit menjadi mudah karena tidak perlu melakukan penyolderan sehingga komponen komponen masih tetap bisa dipergunakan untuk project lain dikemudian hari.

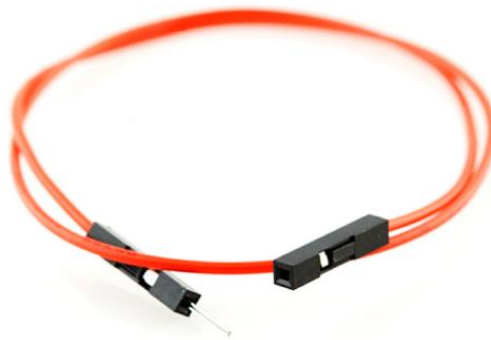
6. Kabel Jumper

Kabel jumper adalah kabel elektrik untuk menghubungkan antar komponen di breadboard tanpa memerlukan solder. Kabel jumper umumnya memiliki connector atau pin di masing-masing ujungnya. Kabel jumper dibagi menjadi 3 yaitu

- a. Male to male



b. Male to female



c. Female to female



BAB III

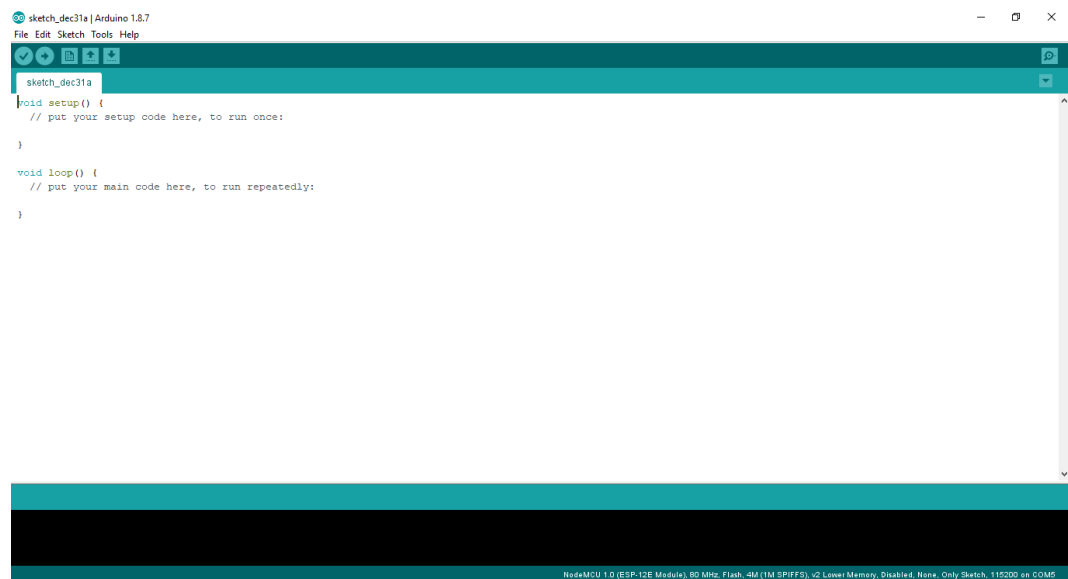
METODOLOGI

A. Penginstalan Aplikasi Arduino 1.8.7

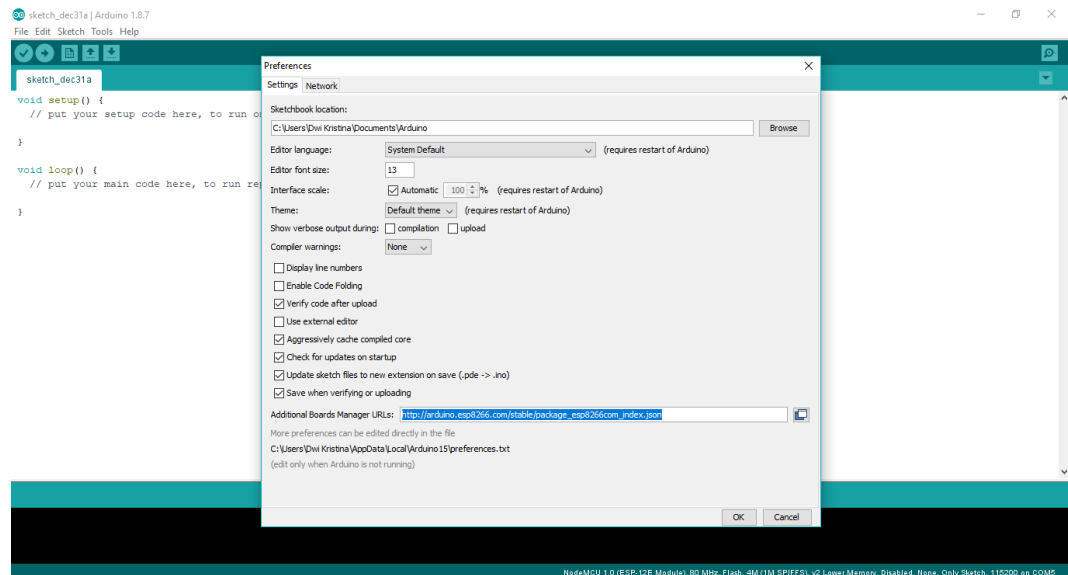
Langkah pertama download aplikasi Arduino IDE 1.8.7

Langkah kedua install Arduino IDE 1.8.7 pada laptop yang akan digunakan.

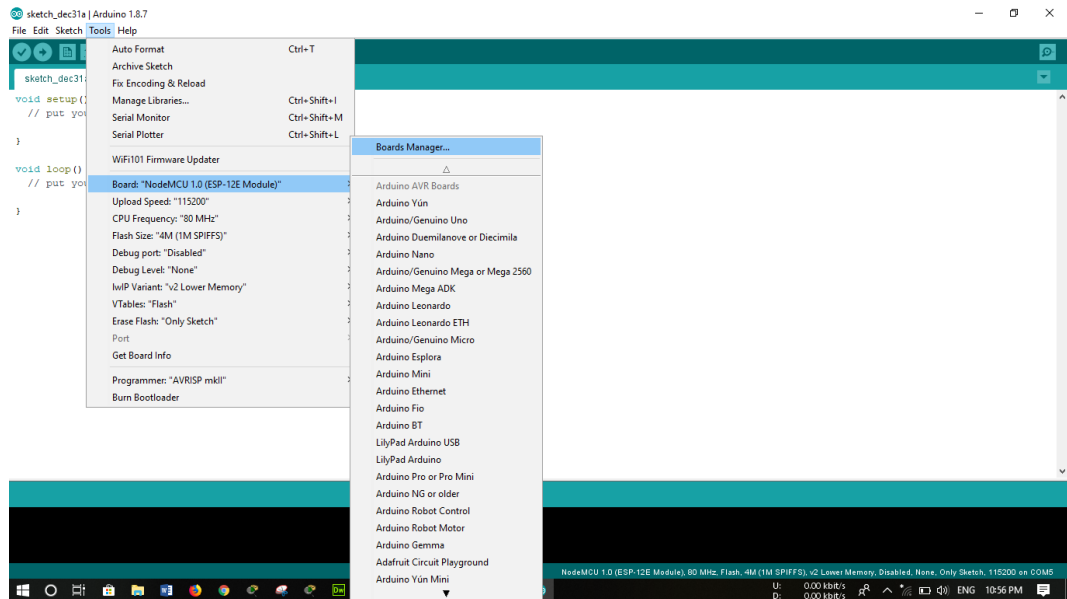
Langkah ketiga buka Arduino



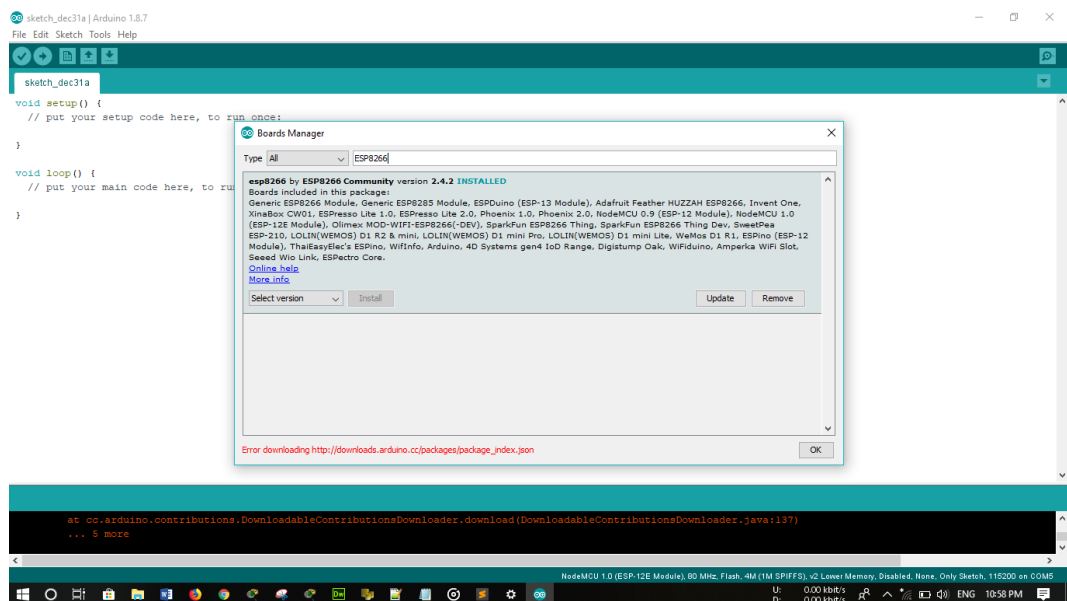
Kemudian pilih menu file>preferences>isikan pada url



Kemudian buka menu tools>board manage>



pada bagian search di bodr manage tuliskan ESP8266>Klik Install maka akan terinstall seperti gambar dibawah ini.



tunggu hingga proses selesai.

B. Firebase

1. Langkah pertama Login ke web <http://firebase.google.com> kemudian masuk ke console
2. Membuat project baru dengan nama hidroponik01 (Bisa menggunakan nama sendiri)

Tambahkan project

Nama proyek

hidroponik01

+ iOS + </>

Tips: Project tersedia untuk aplikasi lintas platform

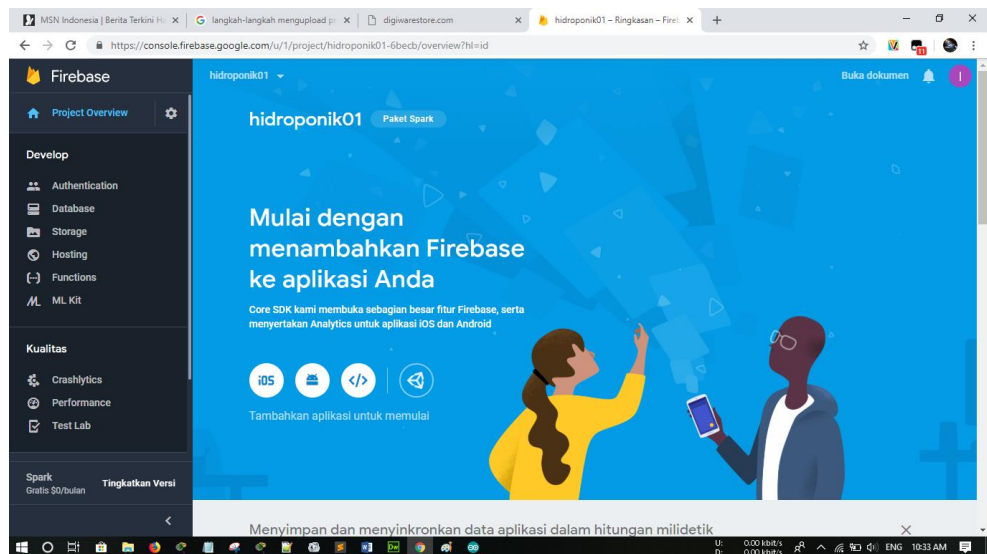
ID project

hidroponik01-e182c

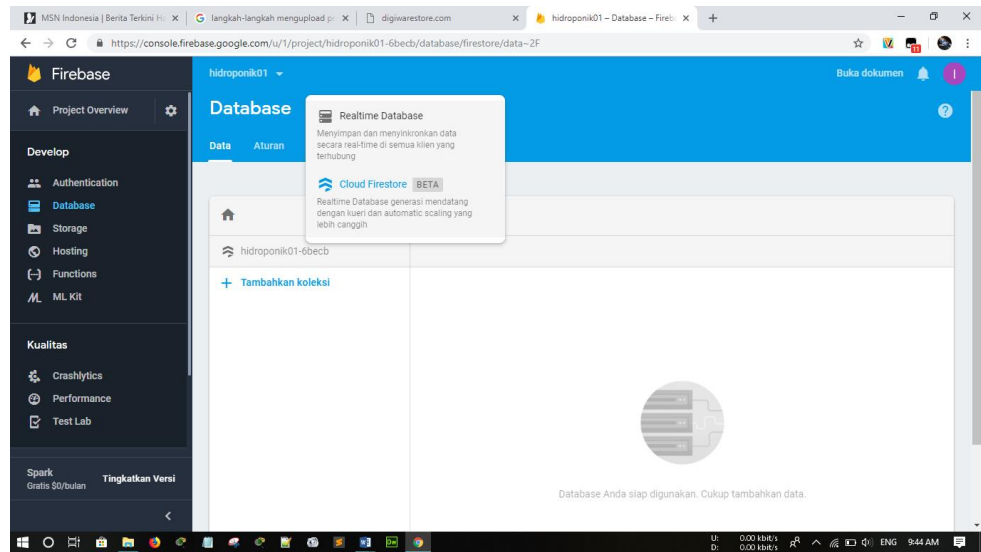
Lokasi Analytics

Indonesia

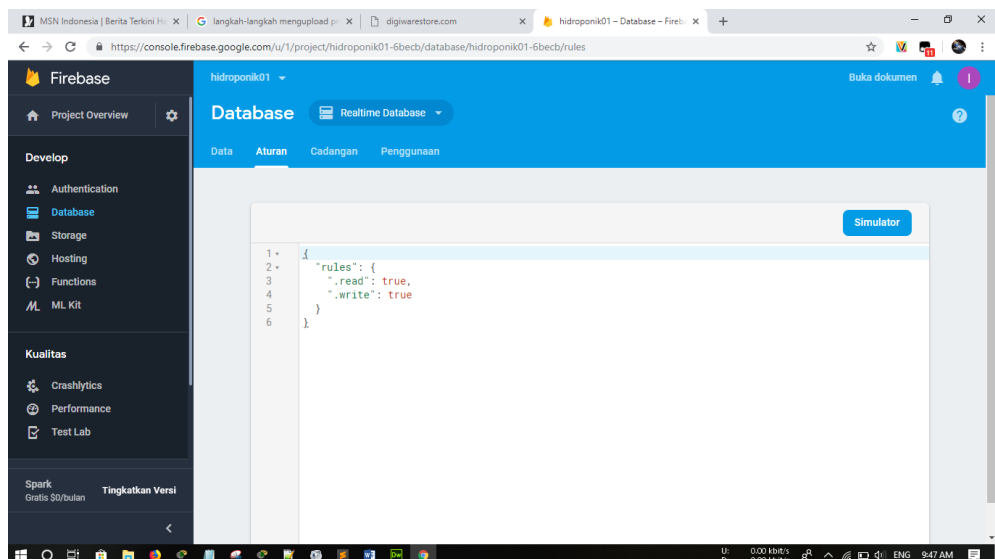
- Langkah berikutnya klik create Project, kemudian ditampilkan halaman console untuk pengaturan firebase



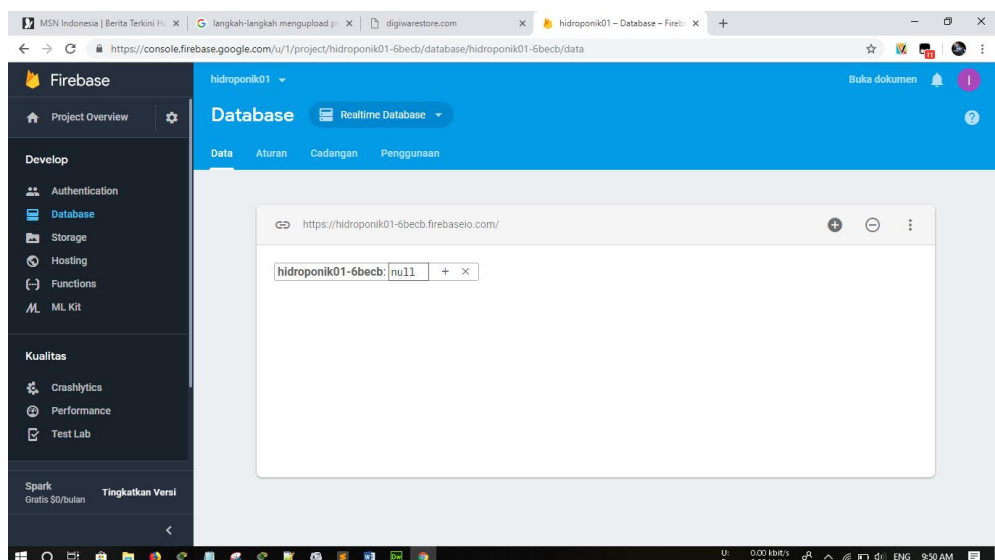
4. Kemudian klik database, lalu masuk ke menu Realtime Database.



5. Ganti coding pada aturan yang awalnya false pindah ke true.

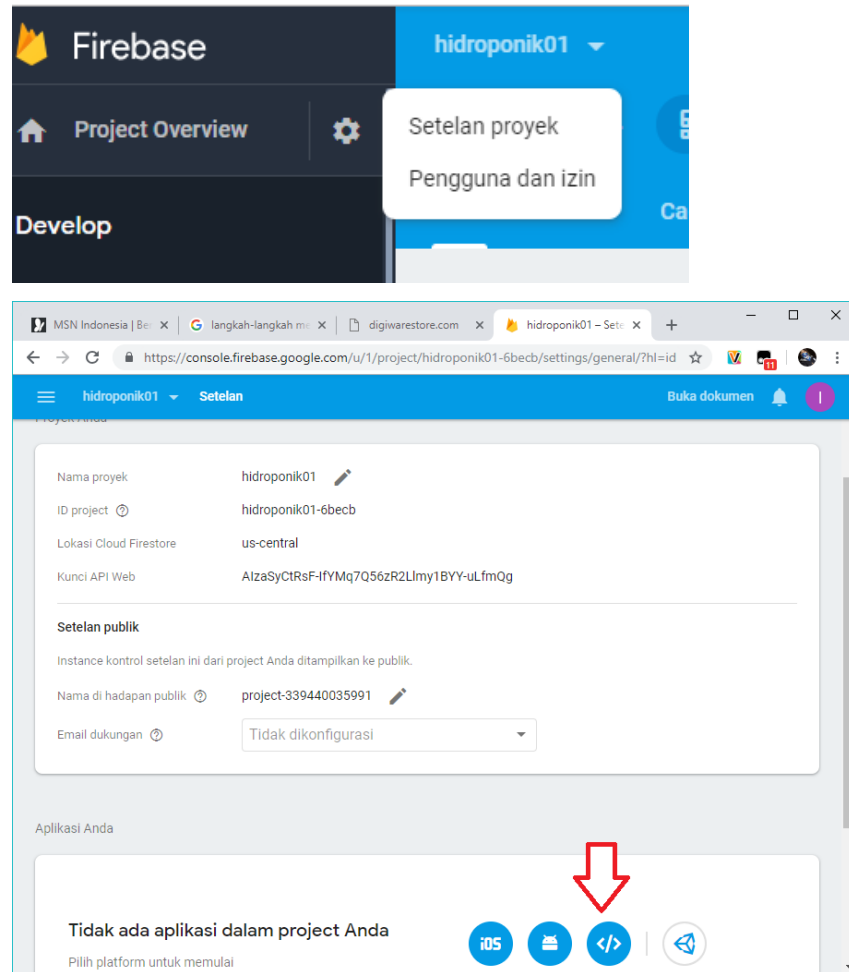


6. Kemudian pindah ke menu data lalu copy URL dan salinkan ke script Arduino.

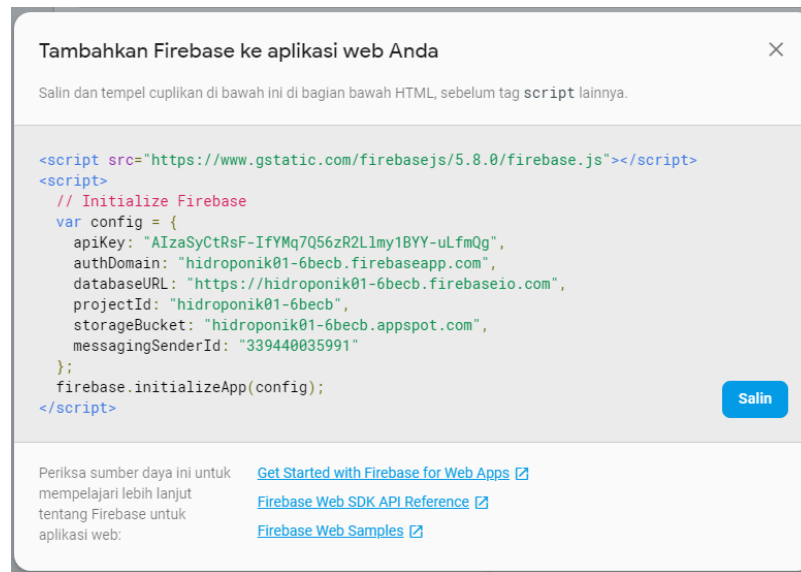


7. Dalam mengkoneksikan firebase dan UI digunakan sebagai pengendali, perlu menggunakan API firebase yang berformat javascript yang berguna agar UI dan firebase dapat terkoneksi dan bertukar data, langkahnya sebagai berikut.

Project Overview > Setelan Proyek



Lalu klik pada symbol tersebut maka akan muncul seperti ini



kode diatas disalin dan digunakan didalam script web

8. Untuk mengkoneksikan firebase dan NodeMCU agar dapat saling bertukar data maka diperlukan konstanta FIREBASE_HOST dan FIREBASE_AUTH yang berfungsi sebagai autentikasi NodeMCU terhadap Firebase.

```
// Set these to run example.
#define FIREBASE_HOST "wortel01.firebaseio.com"
#define FIREBASE_AUTH "o2tOOFpbcvEcXaoFUsZyWus2tijWBWsesv8MzoX8"
#define WIFI_SSID "area501"
#define WIFI_PASSWORD "gazplml23"
```

C. Cara Kerja

- Soil Moisture Sensor membaca objek dengan kadar kelembaban objek sesuai yang telah ditentukan.
- Data yang diperoleh NodeMCU ESP8266 dari Soil Moisture Sensor diolah.
- Jika Soil Moisture Sensor membaca sensor maka dengan otomatis Led akan menyala sebagai indikator dari tanda alat penyiraman bekerja.

D. Prototype Sistem

1. Masukkan script berikut ini ke Arduino

```
#include <ESP8266WiFi.h>
```

```

#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST "wortel01.firebaseio.com"
#define FIREBASE_AUTH
"o2tOOFpbcvEcXaoFUsZyWus2tijWBWsesv8MzoX8"
#define WIFI_SSID "area501"
#define WIFI_PASSWORD "qazplm123"

int soilMoisturePin1 = 2; // VCC ke Pin Digital D4
    int soilMoisturePin2 = 14; // GNG bukan ke GND - Pin output D5
    int soilMoisturePin3 = A0; // A0 ke Pin A0

// 1023 = 100% RAW-Wert is between 0 wet and 1023 dry on 5 Volt
// 901 = 100% RAW-Wert is between 0 wet and 903 dry on VIN Volt
// 672 = 100% RAW-Wert is between 0 wet and 672 dry on 3,3 Volt
// The sensor returned 18% with RAW 188 when fully sunk into the
water and light wave movements.

const int dry = 1023;

const int relayPin = 12; // SIGNAL from Relay
int pump_threshold = 50; // it soil DRY >=50% then activate a pump
int siram = 0;
void setup() {
    Serial.begin(9600);

    while(!Serial); //Waiting for Serial connection
    pinMode(soilMoisturePin1, OUTPUT);
    pinMode(soilMoisturePin2, OUTPUT);
    pinMode(relayPin, OUTPUT); //Set pin 6 as an OUTPUT
    digitalWrite(relayPin, LOW); // Relay off - no pump at start

```

```

// connect to wifi.
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("connecting");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
Serial.println();
Serial.print("connected: ");
Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
//Firebase.setInt("led1", 0);
}

int readSoilMoisture()
{
  int value;
  digitalWrite(soilMoisturePin1, HIGH);
  digitalWrite(soilMoisturePin2, LOW);
  delay(1000);
  value = analogRead(soilMoisturePin3);
  digitalWrite(soilMoisturePin1, LOW);
  digitalWrite(soilMoisturePin2, HIGH);
  delay(1000);
  digitalWrite(soilMoisturePin2, LOW);
  return value;
}

void loop() {
  // Option "easy": int value_water = analogRead(soilMoisturePin3);
  int value_water = readSoilMoisture();

```



```

String water = String((int)((((double)value_water/dry)*100.0)));
Serial.print("Tingkat Kekeringan Tanah : ");
Serial.print(water);
Serial.print("% Nilai : ");
Serial.println(value_water);

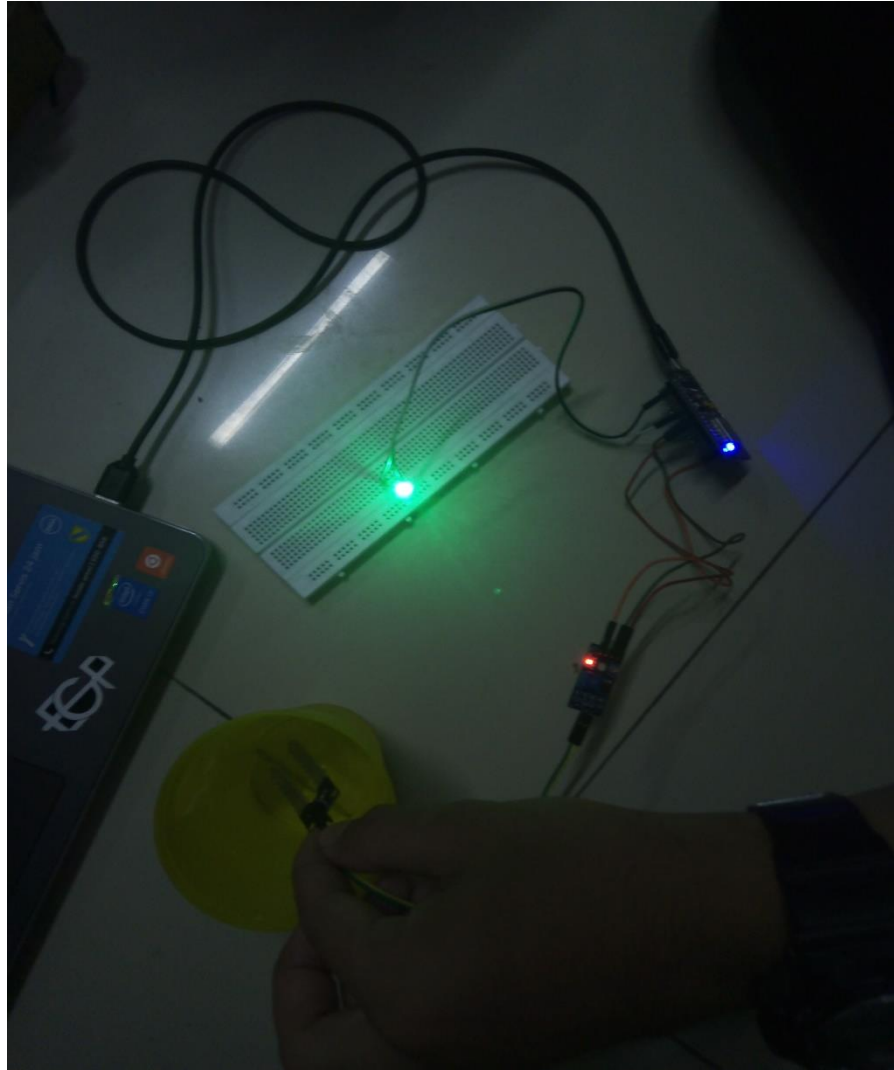
//Change String2Integer and check if threshold is reached
if (water.toInt() >= pump_threshold) //
{
Serial.println("Tanah Terlalu Kering, Dibutuhkan Penyiraman.");
digitalWrite(relayPin, LOW);
siram = 0;
}
else
{
Serial.println("Terlalu Basah, Tidak Butuh Disiram.");
digitalWrite(relayPin, HIGH);
siram = 1;
}
Firebase.setString("kekeringan_tanah", water);
//   Firebase.setInt("water", value_water);
Firebase.setInt("Siram", siram);
}

```

Script yang diUnggah keArduino

2. Langkah selanjutnya Upload program pada Arduino ,tunggu proses selesai.
3. Pasang rangkaian

Berikut bentuk rangkaian yang sudah dipasang :



E. Tampilan UI(User Interface)

Untuk tampilan login system

```
<!DOCTYPE html>
<html>
<head>
  <title>iSawi</title>
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="#006faa">
```

```

        <?php
        if(isset($_GET["login_error"])){
            echo "<h2 style='color:red';>Username atau password
            salah</h2>";
        }
    ?>

```

```

<h2>Login</h2>
<form method="post" action="cek_login.php">
    <div class="imgcontainer">
        
    </div>

    <div class="container">
        <label><b>Username</b></label>
        <input type="text" placeholder="Username atau Email"
        name="username" required>
        <label><b>Password</b></label>
        <input type="password" placeholder="Password"
        name="password" required>

        <button type="submit" name="commit">Masuk</button>
        <input type="checkbox" checked="checked"><span>Ingat
        Saya</span>
    </div>

    <div class="container" style="background-color:#f1f1f1">
        <button type="button" class="cancelbtn">Cancel</button>
        <span class="psw">Lupa <a href="#">password?</a></span>
    </div>
</form>

```

```
</body>
</html>
```

Lalu script ini

```
<?php

ob_start();
session_start();
ob_end_clean();

$username=$_POST["username"];
$password=$_POST["password"];

if($username=="admin" AND $password=="admin")
{
    $_SESSION["username"]=$username;
    header("location:admin.php?p=Homi");
}else{
    header("location:index.php?login_error");
}
?>
```

Kemudian masukkan

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<style>
* {
    padding:0;
    margin:0;
}
```

```
body {  
    font-family: Verdana, Geneva, sans-serif;  
    font-size: 18px;  
    background-color: #FFF  
}
```

```
header {  
    width: 100%;  
    background-color: #006faa ;  
    z-index: 1000;  
}
```

```
.menu-bar {  
    color: #FFF;  
    font-size: 25px;  
    cursor: pointer;  
    padding: 10px 12px;  
    margin-left: 10px;  
    margin-top: 5px;  
    margin-bottom: 5px;  
}
```

```
.menu-bar:hover {  
    background-color: rgba(0, 0, 0, 0.1);  
    border-radius: 50px;  
}
```

```
#tag-menu {  
    display: none;  
}
```

```
#tag-menu:checked ~ div.jw-drawer {  
  animation: slide-in 0.5s ease;  
  animation-fill-mode: forwards;  
}
```

```
.jw-drawer {  
  position:fixed;  
  left:-280px;  
  background-color:#006faa;  
  height:100%;  
  z-index:100;  
  width:230px;  
  animation: slide-out 0.5s ease;  
  animation-fill-mode: forwards;  
}
```

```
.jw-drawer ul li {  
  list-style:none;  
}
```

```
.jw-drawer ul li a {  
  padding:10px 20px;  
  text-decoration:none;  
  display:block;  
  color:#FFF;  
  border-top:1px solid #059;  
}
```

```
.jw-drawer ul li a:hover{  
  background-color:rgba(0, 0, 0, 0.1);  
}
```

```
.jw-drawer ul li a i {
    width:50px;
    height:35px;
    text-align:center;
    padding-top:15px;
}
```

```
@keyframes slide-in {
    from {left: -280px;}
    to {left: 0;}
}
```

```
@keyframes slide-out {
    from {left: 0;}
    to {left: -280px;}
}
```

```
.content{
    padding: 100px 0 0 250px;
}
</style>
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.5.0/css/font-awesome.min.css">
```

```
<head>
```

```
<title>iSawi</title>
```

```
</head>
```

```
<header>
```

```
<tr>
```

```
<th>
```

```

<input type="checkbox" id="tag-menu"/>
<label class="fa fa-bars menu-bar" for="tag-menu"></label>
</th>
<th>
    <label style="color:#FFF">iSawi</label>
</th>
</tr>
<div class="jw-drawer">
    <nav>
        <ul>
            <li><a href="admin.php?p=Controller"><i
class=""></i>Controller</a></li>
            <li><a href="admin.php?p=Monitoring"><i
class=""></i>Monitoring</a></li>
            <li><a href="admin.php?p=logout"><i
class=""></i>Logout</a></li>
        </ul>
    </nav>
</div>
</header>
<body>

```

```

<?php
    $pages_dir = 'halaman';
    if(!empty($_GET['p'])){
        $pages = scandir($pages_dir, 0);
        unset($pages[0], $pages[1]);

        $p = $_GET['p'];
        if(in_array($p.'.php', $pages)){
            include($pages_dir.'/'.$p.'.php');
        } else {

```



```

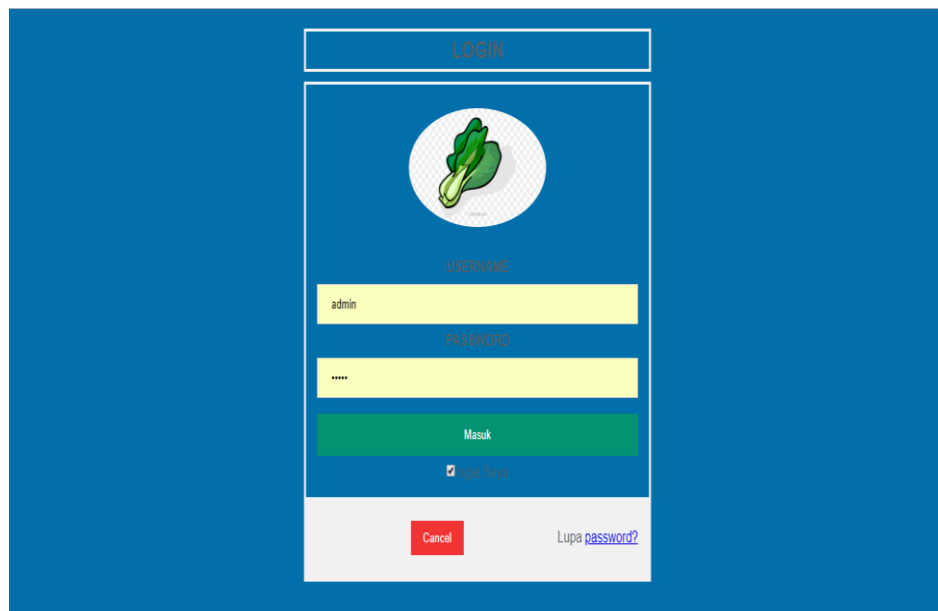
        echo 'Halaman tidak ditemukan! :(';
    }
} else {
    include($pages_dir.'/homi.php');
}

?>

</body>
</html>

```

Script halaman login



Tampilan Halaman Utama

```

<html>
  <head>
    <title>
      Home
    </title>
  </head>

```

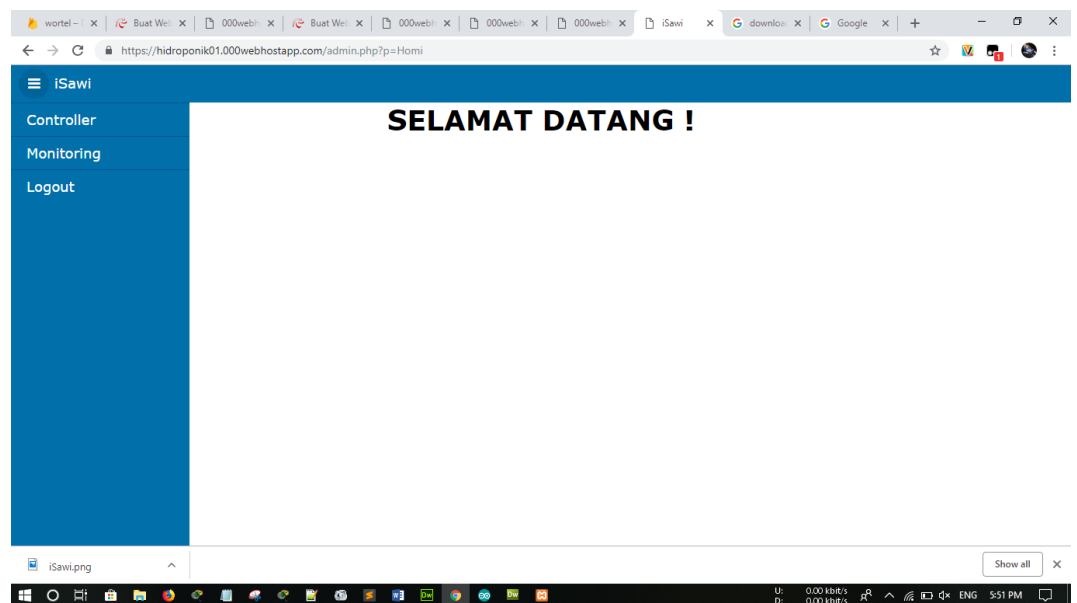
```

<body>
    <H1 align="center">SELAMAT DATANG !</H1>
</body>
</html>

```

Script Halaman utama

UI halaman utama



Tampilan Controller

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
    <td><h2>STATUS</h2></td>

    <tr>

```

```

        <td><button name="pateni" id="pateni"
onclick="pateni_()">Matikan</button></td>
    </tr>
</script>
<script src="https://www.gstatic.com/firebasejs/5.7.3/firebase.js"></script>
<script>
    // Initialize Firebase
    var config = {
        apiKey: "AIzaSyBUqzIBIEittavMEpih0aXWb_snEqSL7tY",
        authDomain: "wortel01.firebaseio.com",
        databaseURL: "https://wortel01.firebaseio.com",
        projectId: "wortel01",
        storageBucket: "wortel01.appspot.com",
        messagingSenderId: "1059964507280"
    };
    firebase.initializeApp(config);
    var Siram = 0;
    var getSiram = firebase.database().ref('/Siram');
    getSiram.on('value', function(snapshot) {
        Siram = snapshot.val();
    });

    function siram_() {
        if (Siram==1) {
            document.getElementById("pateni").innerHTML = "Matikan";
        } else if (Siram==0) {
            document.getElementById("pateni").innerHTML = "Hidupkan";
        }
    }

    function pateni_() {
        if (Siram==1) {
            firebase.database().ref('/Siram').set(0);

```

```
        } else if (Siram==0){
            firebase.database().ref('/Siram').set(1);
        }
    }

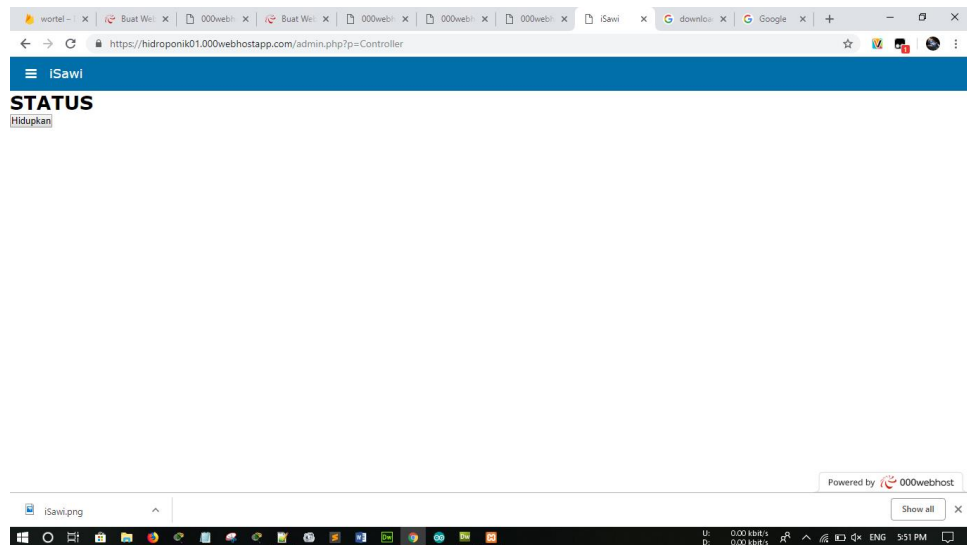
    function upd() {
        getSiram.on('value', function(snapshot) {
            Siram = snapshot.val();
        });
        siram_();
    }

    setInterval(upd, 500);

</script>
</body>
</html>
```

Script Controller

UI Halaman Controller



Tampilan Monitoring

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Monitoring</title>
</head>

<body>
<table align="center">
<tr>
<td>
<label id="kering">Tingkat Kekeringan
tanah : </label>
</td>
</tr>
<tr>
<td>
```

```

<label id="status">Status : </label>
</td>
</tr>
</table>

```

```

<script src="https://www.gstatic.com/firebasejs/5.7.3/firebase.js"></script>

```

```

<script>

```

```

// Initialize Firebase

```

```

var config = {

```

```

  apiKey: "AIzaSyBUqzIBIEittavMEpih0aXWb_snEqSL7tY",

```

```

  authDomain: "wortel01.firebaseio.com",

```

```

  databaseURL: "https://wortel01.firebaseio.com",

```

```

  projectId: "wortel01",

```

```

  storageBucket: "wortel01.appspot.com",

```

```

  messagingSenderId: "1059964507280"

```

```

};

```

```

firebase.initializeApp(config);

```

```

var Kering = "50";

```

```

var Siram = 0;

```

```

var getKering = firebase.database().ref('/kekeringan_tanah');

```

```

var getSiram = firebase.database().ref('/Siram');

```

```

getKering.on('value', function(snapshot) {

```

```

  Kering = snapshot.val();

```

```

});

```

```

getSiram.on('value', function(snapshot) {

```

```

  Siram = snapshot.val();

```

```

});

```

```

function kering_() {

```

```

  document.getElementById("kering").innerHTML = "Tingkat

```

```

  Kekeringan Tanah : "+Kering+"%;

```

```

    }
    function siram_() {
        if (Siram==1) {
            document.getElementById("status").innerHTML = "Status :
Disiram";
        } else if (Siram==0) {
            document.getElementById("status").innerHTML = "Status :
Tidak Disiram";
        }
    }
}

function upd() {
    getKering.on('value', function(snapshot) {
        Kering = snapshot.val();
    });
    getSiram.on('value', function(snapshot) {
        Siram = snapshot.val();
    });

    kering_();
    siram_();
}

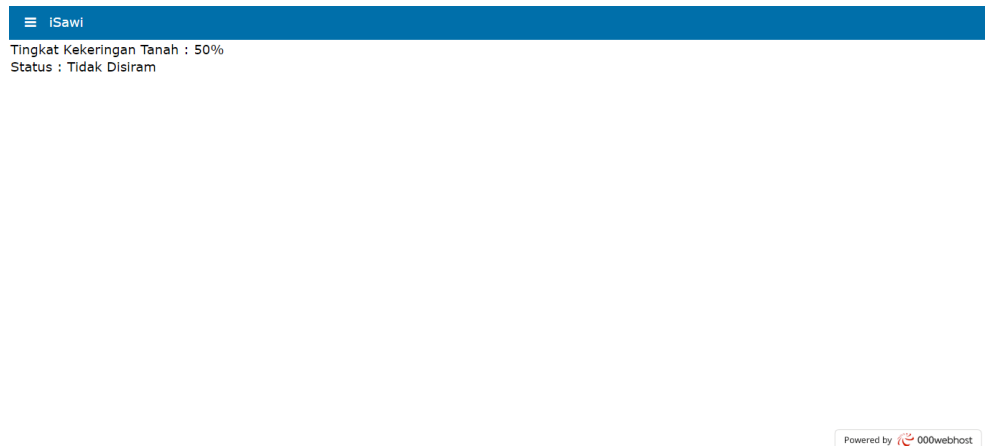
setInterval(upd, 500);

</script>
</body>
</html>

```

Script Halaman Monitoring

UI halaman Monitoring



Tampilan Halaman Logout

```
<?php
    ob_start();
    session_start();
    ob_end_clean();
    session_destroy();
    header("location: ../index.php");
?>
```

Script Halaman Logout

BAB IV

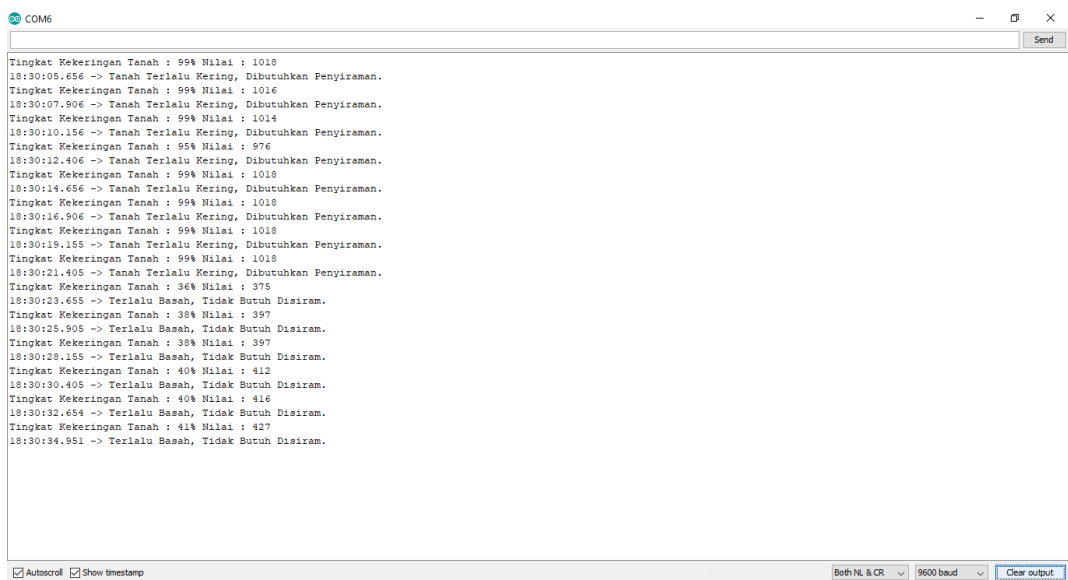
PEMBAHASAN

A. Percobaan Percobaan Menggunakan Arduino

Langkah pertama upload script sampai dengan perintah Done Compiling

Kemudian pasang sensor sesuai rangkaian.

buka serial monitoring pada Arduino



The screenshot shows a Serial Monitor window titled 'COM6' with a 'Send' button in the top right corner. The output text is as follows:

```
Tingkat Kekeringan Tanah : 99% Nilai : 1018
18:30:05.656 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 99% Nilai : 1016
18:30:07.906 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 99% Nilai : 1014
18:30:10.156 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 99% Nilai : 976
18:30:12.406 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 99% Nilai : 1018
18:30:14.656 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 99% Nilai : 1018
18:30:16.906 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 99% Nilai : 1018
18:30:19.155 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 99% Nilai : 1018
18:30:21.405 -> Tanah Terlalu Kering, Dibutuhkan Penyiraman.
Tingkat Kekeringan Tanah : 36% Nilai : 375
18:30:23.655 -> Terlalu Basah, Tidak Butuh Disiram.
Tingkat Kekeringan Tanah : 38% Nilai : 397
18:30:25.905 -> Terlalu Basah, Tidak Butuh Disiram.
Tingkat Kekeringan Tanah : 38% Nilai : 397
18:30:28.155 -> Terlalu Basah, Tidak Butuh Disiram.
Tingkat Kekeringan Tanah : 40% Nilai : 412
18:30:30.405 -> Terlalu Basah, Tidak Butuh Disiram.
Tingkat Kekeringan Tanah : 40% Nilai : 416
18:30:32.654 -> Terlalu Basah, Tidak Butuh Disiram.
Tingkat Kekeringan Tanah : 41% Nilai : 427
18:30:34.951 -> Terlalu Basah, Tidak Butuh Disiram.
```

At the bottom of the window, there are checkboxes for 'Autoscroll' and 'Show timestamp', both of which are checked. On the right side, there are dropdown menus for 'Both NL & CR' and '9600 baud', and a 'Clear output' button.

pada percobaan yang dilakukan ketika sensor membaca kondisi air > 1023 maka akan muncul pemberitahuan suhu kelembaban air dan tanah terlalu kering, sehingga lampu led akan menyala dan mengirimkan pesan bahwa lahan pertanian butuh dilakukan penyiraman otomatis. Namun, sebaliknya jika kondisi air < 901 maka akan muncul pemberitahuan terlalu kering sehingga dibutuhkan penyiraman. Dari deteksi sensor tersebut diperoleh perintah yang dikirimkan pada maka lampu Led tidak akan menyala.

BAB V

PENUTUP

Kesimpulan

Dari hasil percobaan yang dilakukan pada hidroponik tanaman Sawi yang mengalami masalah pengairan pada media tanam. Masalah pada media tanam menyebabkan sawi gagal panen dan mengalami banyak kerugian pada petani. Dengan menggunakan IOT pada penyiramannya yang dibantu oleh sensor kelembaban yang bekerja untuk mengecek kadar kelembaban pada tanah.