

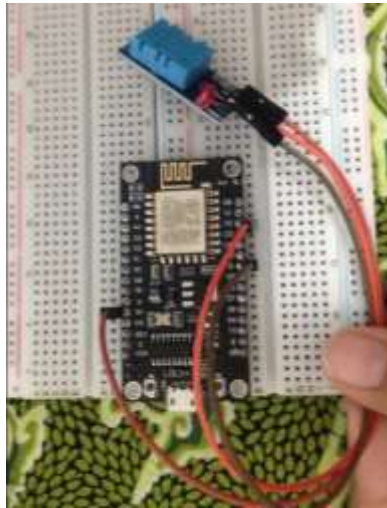
### **3.3 Realisasi**

#### **3.3.1.1 Realisasi PCB**

Realisasi PCB belum dapat direalisasikan karena tahap realisasi masih menggunakan breadboard. Tahap realisasi hanya sampai tahap pengerjaan sensor node yang terdiri dari NodeMCU dan sensor DHT11. Untuk merealisasinya maka diperlukan aplikasi Arduino.

#### **3.3.1.2 Realisasi Perakitan**

Berikut ini merupakan perakitan dari sensor node yang terdiri dari NodeMCU dan sensor DHT11.



Gambar 3.10 Realisasi Perakitan

#### **3.3.1.3 Realisasi Pengkabelan**

Realisasi Pengkabelan belum dapat di realisasikan.

### **3.3.2 Realisasi Perangkat Lunak**

#### **3.3.2.1 Realisasi Program**

Program yang dibuat pertama kali merupakan program untuk mengkonfigurasi database dimana data sensor suhu dan kelembaban akan disimpan dengan nama dbconfig.php. Sebelumnya buatlah database pada web hosting. Terdapat keterangan yang perlu diisi berupa username database, nama database,

password database, dan server database. Keterangan tersebut lalu digunakan untuk mengkonfigurasi agar dapat mengakses database yang dibuat.

```
<?php

define('DB_USER', "id9279931_kumbungjamur");    // User name
database
define('DB_PASSWORD', "kumbungjamur");          // Password
database
define('DB_DATABASE', "id9279931_kumbungjamur"); // Nama database
define('DB_SERVER', "localhost");               // Server database
(Kebanyakan 'local' host)

?>
```

Gambar 3.11 Program dbconfig.php

Kemudian selanjutnya adalah program untuk dapat mengkoneksikan ke database dengan nama db\_connect. Disini dilakukan import dari file dbconfig.php untuk mengambil kredensial database atau keterangan database yang dibutuhkan untuk mengakses database. Jika kredensial yang dimasukan benar maka pengaksesan ke database diperbolehkan. Lalu pilih database yang ingin diakses setelah itu koneksi ditutup.

```
<?php

class DB_CONNECT {

    // Constructor
    function __construct() {
        // Koneksi ke database
        $this->connect();
    }

    // Destructor
    function __destruct() {
        // Menutup koneksi ke database
        $this->close();
    }

    // Konek ke database
    function connect() {
```

```

        //mengimport dbconfig.php yang mengandung kredensial
database
        $filepath = realpath (dirname(__FILE__));

        require_once($filepath."/dbconfig.php");

        // Menghubungkan ke mysql (phpmyadmin) database
        $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or
die(mysql_error());

        // Memilih database
        $db = mysql_select_db(DB_DATABASE) or die(mysql_error())
or die(mysql_error());

        return $con;
    }

    // Menutup koneksi database
    function close() {
        mysql_close();
    }
}

?>

```

Gambar 3.12 Program db\_connect.php

Program untuk memasukan nilai suhu dan kelembaban dari sensor ke database. Metoda GET digunakan untuk mengirim data dari sensor ke database. Jika data sensor tidak ada maka JSON akan mengirim respon data hilang silahkan coba lagi. Namun jika terdapat data diterima maka data tersebut akan ditampung ke “S\_GET” sesuai dengan variable datanya. Kemudian koneksi dilakukan pada database yang dipilih. Data yang diterima dari sensor kemudian dimasukkan pada tabel “monitoring”. Kemudian dilakukan pengecekan kembali apakah memasukkan data berhasil atau tidak.

```

<?php

header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");

//Membuat array untuk respon JSON
$response = array();

```

```

// Mengecek jika terdapat data yang diambil
if (isset($_GET['temp']) && isset($_GET['hum'])) {

    $temp = $_GET['temp'];
    $hum = $_GET['hum'];

    // Memasukan kelas connect database
    $filepath = realpath (dirname(__FILE__));
    require_once($filepath."/db_connect.php");

    // Menghubungkan ke database
    $db = new DB_CONNECT();

    // SQL query memasukan data ke tabel monitoring
    $result = mysql_query("INSERT INTO monitoring(temp,hum)
VALUES('$temp','$hum')");

    // Mengecek keberhasilan eksekusi
    if ($result) {
        // Data berhasil dimasukan
        $response["success"] = 1;
        $response["message"] = "Monitoring successfully created.";

        // Menampilkan respon JSON
        echo json_encode($response);
    } else {
        // Gagal memasukan ke database
        $response["success"] = 0;
        $response["message"] = "Something has been wrong";

        // Menampilkan respon JSON
        echo json_encode($response);
    }
} else {
    // Jika parameter yang diinginkan hilang
    $response["success"] = 0;
    $response["message"] = "Parameter(s) are missing. Please check
the request";

    // Menampilkan respon JSON
    echo json_encode($response);
}
?>

```

Gambar 3.13 Program insert.php

Selanjutnya adalah program untuk menampilkan semua data yang diterima dari sensor bernama read\_all.php. Program ini mengambil keseluruhan data database menggunakan perintah select\*from(tabel). Program ini berfungsi untuk memberikan 500 nilai terakhir dari pembacaan sensor. Nilai yang paling terakhir akan ditampilkan pada website.

```
<?php

header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");

//Membuat array untuk respon JSON
$response = array();

// Memanggil kelas connect database
$filepath = realpath (dirname(__FILE__));
require_once($filepath."/db_connect.php");

// Menghubungkan ke database
$db = new DB_CONNECT();

// SQL query untuk mendapatkan data dari monitoring
$result = mysql_query("SELECT *FROM monitoring") or
die(mysql_error());

// Mengecek ketersediaan data
if (mysql_num_rows($result) > 0) {

    // Storing the returned array in response
    $response["monitoring"] = array();

    // While loop to store all the returned response in variable
    while ($row = mysql_fetch_array($result)) {
        // temporary user array
        $monitoring = array();
        $monitoring["id"] = $row["id"];
        $monitoring["temp"] = $row["temp"];
        $monitoring["hum"] = $row["hum"];

        // Push all the items
        array_push($response["monitoring"], $monitoring);
    }

    // On success
    $response["success"] = 1;
}
```

```

        // Show JSON response
        echo json_encode($response);
    }
    else
    {
        // Jika tidak ada data ditemukan
        $response["success"] = 0;
        $response["message"] = "No data on monitoring found";

        // Show JSON response
        echo json_encode($response);
    }
    ?>

```

Gambar 3.14 Program read\_all.php

Selanjutnya adalah pembuatan file html untuk ditampilkan pada website. Website monitoring ini terdiri dari 1 halaman saja. Halaman ini mengandung teks dan gambar. Pada html file ini akan menampilkan data yang diterima dari database. Dari data tersebut diambil data yang paling akhir dari ["https://kumbungjamur.000webhostapp.com/api/monitoring/read\\_all.php"](https://kumbungjamur.000webhostapp.com/api/monitoring/read_all.php).

```

<!DOCTYPE html>

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0"/>
        <meta http-equiv="Access-Control-Allow-Origin" content="*">

        <script type="text/javascript">
            document.write([
                "<script src='",
                ("https:" == document.location.protocol) ? "https://"
: "http://",
                "ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min
.js' type='text/javascript'>\</script>"
            ].join(''));
        </script>
        -->
        <title>Jamur Tiram Monitoring</title>
        <style>

        .footer{

```

```

        background:#64B5F6;
        width:100%;
        height:100px;
        position:absolute;
        bottom:0;
        left:0;
    }
    .banner{
        height: 0px;
        background-size: cover;
        background-position: 0px 0px;
        border-top: 10px solid black;
        border-bottom: 10px solid #9b2412;
    }
    .center {
        height: 400px;
        width: 480px;
        background: #702d1a;
        position: fixed;
        box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0
        rgba(0, 0, 0, 0.19);
        top: 50%;
        left: 50%;
        margin-top: -150px;
        margin-left: -240px;
    }

    .form{
        padding-top: 10px;
        padding-right: 10px;
        padding-bottom: 30px;
        padding-left: 10px;
    }
    .ip{
        background-color: #ffffff; /* Green */
        border: none;
        color: black;
        padding: 16px 32px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
        margin: 4px 2px;
        -webkit-transition-duration: 0.4s; /* Safari */
    }

</style>
</head>

```

```

<body bgcolor="#ffffff">
  <center>
    <img src = 'jamurtiram.png' height="80px"
width="250px" style='vertical-align: middle' />
  </center>
  <center>
    <h1 style="font-family: Helvetica;color:
black;">Pembacaan Suhu dan Kelembaban</h1>
  </center>
  <div class="banner"></div>
  <div class="center">
    <div align="center" class="form">
      <br><br>
      <p style = 'line-height: 60px;font-family:
Helvetica;color: #fff;font-size: 50px;' id="temperature">
        <img src = 'temperature.png' height="60px"
width="60px" style='vertical-align: middle' /> 00.00
      </p>
      <p style = 'line-height: 60px;font-family:
Helvetica;color: #fff;font-size: 50px;' id="humidity">
        <img src = 'humidity.png' height="60px"
width="60px" style='vertical-align: middle' /> 00.00
      </p>
    </div>
  </div>

</body>

<script>
  window.onload = function() {
    loaddata();
  };
  function loaddata(){
    var url =
"https://kumbungjamur.000webhostapp.com/api/monitoring/read_all.ph
p";

    $.getJSON(url, function(data) {
      var val= data;
      var
humid=(data['monitoring'][(Object.keys(data['monitoring']).length)
-1]['hum']);

      var
temper=(data['monitoring'][(Object.keys(data['monitoring']).length)
-1]['temp']);

    document.getElementById("temperature").innerHTML = "<img src =

```



```

'temperature.png' height=\"60px\" width=\"60px\" style='vertical-align: middle' /> " +temper;

document.getElementById("humidity").innerHTML = "<img src =
'humidity.png' height=\"60px\" width=\"60px\" style='vertical-align: middle' /> "+humid;

console.log(data['monitoring'][(Object.keys(data['monitoring']).length)-1]['temp']);
    });

    }
    window.setInterval(function(){
        loaddata();
    }, 5000);

</script>
</html>

```

Gambar 3.15 Program index.html

Berikut ini program pembacaan data dari sensor suhu dan kelembaban DHT11. Sebelum membuat program, terlebih dahulu menginstall board yang digunakan yaitu nodeMCU ESP8266 E12. Diperlukan juga untuk mendownload library untuk pembacaan sensor DHT11. Program ini juga digunakan untuk mengupload data pada database di webhosting. Pertama-tama perlu dideklarasikan ssid, password, serta host yang akan digunakan. Setelah terhubung pada wifi, kemudian dilakukan pembacaan pada sensor DHT11. Setelah data sensor didapatkan lalu data tersebut dikirim ke database yang telah dibuat sesuai dengan variabelnya. Pembacaan sensor diberi delay 3000ms yang berarti pembacaan sensor setiap 3 detik.

```

#include <ESP8266WiFi.h>
#include "DHT.h"

#define DHTPIN D1
#define DHTTYPE DHT11

const char* ssid = "N22 POJOK";
const char* password = "ayuangga";
const char* host = "kumbungjamur.000webhostapp.com";
DHT dht(DHTPIN, DHTTYPE);

```

```

void setup() {
  Serial.begin(115200);
  delay(100);
  dht.begin();
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.print("Netmask: ");
  Serial.println(WiFi.subnetMask());
  Serial.print("Gateway: ");
  Serial.println(WiFi.gatewayIP());
}

void loop() {
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("connecting to ");
  Serial.println(host);

  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
  }
}

```

```

    return;
}

String url = "%api/monitoring/insert.php?temp=" + String(t) + "&hum="+
String(h);
Serial.print("Requesting URL: ");
Serial.println(url);

client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
delay(500);

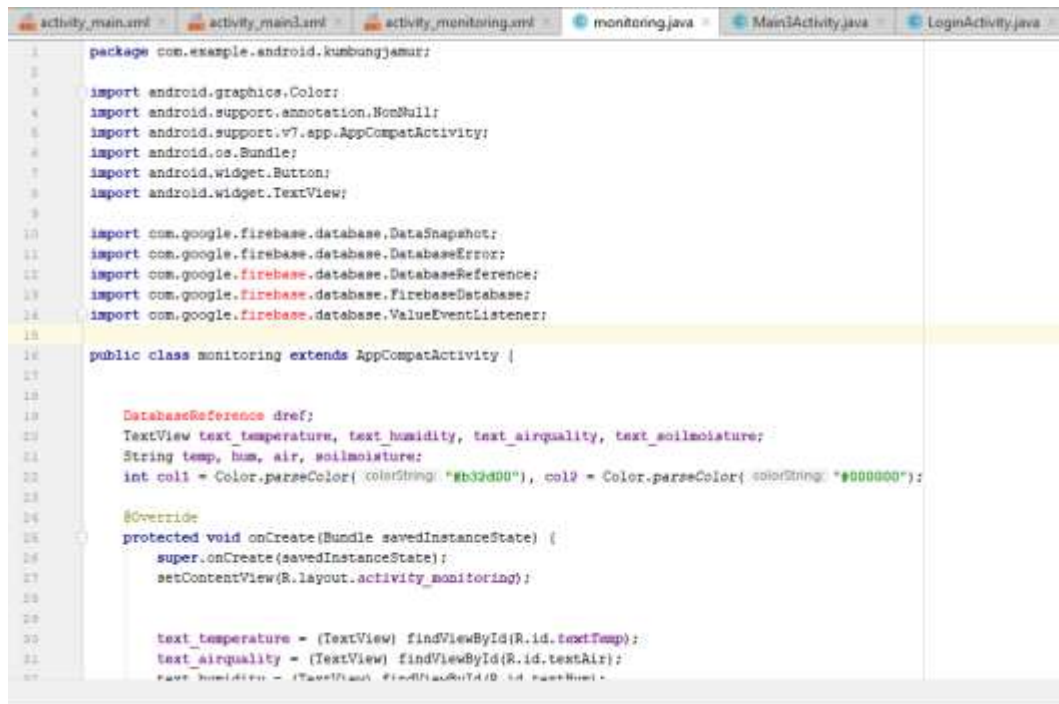
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
}

Serial.println();
Serial.println("closing connection");
delay(3000);
}

```

Gambar 3.16 Program Post\_dht11\_data

Program yang terakhir adalah pembuatan aplikasi android. Pertama- tama yang harus dilakukan adalah pembuatan layout. Aplikasi android terdiri dari 2 halaman. Halaman pertama terdiri dari logo aplikasi serta tombol untuk ke halaman selanjutnya. Di halaman kedua lalu merupakan halaman utama dimana kita dapat memonitoring kumbung jamur melalui nilai suhu dan temperature di area kumbung. Aplikasi android ini akan mengambil data dari database sama halnya dengan website. Nilai data sensor yang diambil adalah data yang paling terakhir terbaca oleh sensor.



Gambar 3.17 Program Android Studio

### 3.3.2.2 Realisasi Database

Database dibuat sesuai dengan kebutuhan dari sistem yang dibuat. Database yang dibuat bernama id9279931\_kumbungjamur dan terdiri dari 3 kolom yaitu id, nilai suhu, dan nilai temperatur. Id diberi nilai maksimal yaitu 255.

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id	int(255)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	temp	varchar(10)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 3	hum	varchar(10)	utf8_unicode_ci		No	None		

☐ Check all
 With selected:
 

Browse
 Change
 Drop
 Primary
 Unique

Gambar 3.18 Database

### 3.3.3 Realisasi Mekanik

Realisasi mekanik belum dapat direalisasikan.

#### **3.3.3.1 Realisasi Mekanik Pendukung**

Realisasi mekanik pendukung belum dapat direalisasikan.

#### **3.3.3.2 Realisasi Kemasan Alat**

Realisasi kemasan alat belum dapat direalisasikan.