**RANCANG BANGUN AIR HUMIDIFIER BERBASIS**

**INTERNET of THINGS ( IOT)**

**LAPORAN PROJECT 3**

****

**Disusun oleh:**

**Ahmad Rouf 1122140071**

**Azrul Barja 1122140040**

**Fajar Eka Jaya Prasetya 1122140038**

**Natasya Rizki Ardilla 1122140119**

**Rifki Juliana 1122140067**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**KONSENTRASI COMPUTER NETWORKING**

**INSTITUT TEKNOLOGI DAN BISNIS**

**BINA SARANA GLOBAL**

**TANGERANG**

**2025**

# KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat, karunia, dan ridha-Nya, penulis dapat menyelesaikan laporan yang berjudul “Rancang Bangun Air Humidifier Berbasis Internet of Things (IoT)” ini dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban akademik dalam menyelesaikan proyek rancang bangun sistem berbasis IoT, khususnya dalam bidang pengendalian kelembaban udara secara otomatis dan real-time melalui platform Internet of Things. Dalam laporan ini, penulis menjelaskan secara rinci mengenai proses perancangan, implementasi, serta pengujian sistem air humidifier yang dapat dipantau dan dikontrol melalui perangkat mobile.

Penulis berharap laporan ini dapat memberikan manfaat bagi pembaca serta dapat menjadi referensi bagi pengembangan sistem serupa di masa mendatang. Penulis juga menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan guna perbaikan di masa yang akan datang. Akhir kata, semoga laporan ini dapat bermanfaat dan memberikan kontribusi positif dalam pengembangan teknologi IoT di bidang pengendalian lingkungan.

Tangerang, 16 Juni 2025

Penulis

# DAFTAR ISI

[**KATA PENGANTAR II**](#_Toc201157895)

[**DAFTAR ISI III**](#_Toc201157896)

[**DAFTAR GAMBAR V**](#_Toc201157897)

[**DAFTAR TABEL VI**](#_Toc201157898)

[BAB I PENDAHULUAN 7](#_Toc201157899)

[1.1 Latar Belakang 7](#_Toc201157900)

[1.2 Rumusan Masalah 8](#_Toc201157901)

[1.3 Batasan Masalah 8](#_Toc201157902)

[1.4 Tujuan Penulisan 9](#_Toc201157903)

[BAB II LANDASAN TEORI 10](#_Toc201157904)

[2.1 Teori Umum 10](#_Toc201157905)

[2.1.1 ESP 32 10](#_Toc201157906)

[2.1.2 DHT 22 11](#_Toc201157907)

[2.1.3 LCD 1602 I2C 12](#_Toc201157908)

[2.1.4 Mist Maker 14](#_Toc201157909)

[2.1.5 Relay SRD-05VDC-SL-C 15](#_Toc201157910)

[2.1.6 Power Supply Breadboard MB102 3.3-5V 17](#_Toc201157911)

[2.1.7 Breadboard 18](#_Toc201157912)

[2.1.8 Jumper Cable 19](#_Toc201157913)

[2.2 Teori Khusus 20](#_Toc201157914)

[2.2.1 Internet of Things ( IoT ) 20](#_Toc201157915)

[2.2.2 Blynk 22](#_Toc201157916)

[2.2.3 Rancang Bangun Berbasis IoT 23](#_Toc201157917)

[BAB III IMPLEMENTASI & PEMBAHASAN 25](#_Toc201157918)

[3.1 Wiring Diagram 25](#_Toc201157919)

[3.1.1 Komponen yang Digunakan 25](#_Toc201157920)

[3.1.2 Skema Pengkabelan 25](#_Toc201157921)

[3.1.3 Sketsa Gambar 26](#_Toc201157922)

[3.2 Source Code 27](#_Toc201157923)

[3.2.1 Platformio.ini 27](#_Toc201157924)

[3.2.2 Main.cpp 28](#_Toc201157925)

[3.3 Blynk Setup 34](#_Toc201157926)

[3.3.1 Informasi Template 34](#_Toc201157927)

[3.3.2 Datastreams 34](#_Toc201157928)

[3.3.3 Widget 34](#_Toc201157929)

[BAB IV PENUTUP 35](#_Toc201157930)

[4.1 Kesimpulan 35](#_Toc201157931)

[4.2 Saran 36](#_Toc201157932)

[**LAMPIRAN 37**](#_Toc201157933)

[**DAFTAR PUSTAKA 38**](#_Toc201157934)

# DAFTAR GAMBAR

[Gambar 2 . 1 : ESP32 C6 WROOM 1 10](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157851)

[Gambar 2 . 2 : Sensor DHT22 11](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157852)

[Gambar 2 . 3 : LCD 1602 with I2C interface 12](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157853)

[Gambar 2 . 4 : Mist Maker module 14](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157854)

[Gambar 2 . 5 : Relay module 15](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157855)

[Gambar 2 . 6 : Power Supply Module 17](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157856)

[Gambar 2 . 7 : Breadboard 18](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157857)

[Gambar 2 . 8 : Jumper Cable 19](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc201157858)

[Gambar 3 . 1 : Sketsa Gambar 26](file:///E:\AAFILE%20KULIAH\SEMESTER%206\Project%203\RANCANG%20BANGUN%20AIR%20HUMIDIFIER%20BERBASIS.docx#_Toc200958515)

# DAFTAR TABEL

[Tabel 3. 1 : Komponen 23](#_Toc200958460)

[Tabel 3. 2 : DHT22 Wiring 23](#_Toc200958461)

[Tabel 3. 3 : LCD Wiring 23](#_Toc200958462)

[Tabel 3. 4 : Relay Wiring 23](#_Toc200958463)

[Tabel 3. 5 : Mist Maker Wiring 24](#_Toc200958464)

[Tabel 3. 6 : Informasi Template 31](#_Toc200958465)

[Tabel 3. 7 : Datastreams 32](#_Toc200958466)

[Tabel 3. 8 : Widget 32](#_Toc200958467)

## 

**PENDAHULUAN**

### Latar Belakang

Perkembangan teknologi Internet of Things (IoT) telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk di bidang otomasi rumah tangga (smart home). IoT memungkinkan perangkat elektronik untuk saling terhubung, berbagi data, dan dikendalikan secara jarak jauh melalui jaringan internet. Salah satu penerapan IoT yang kini semakin relevan adalah sistem pengatur kelembaban udara atau air humidifier yang dapat dipantau dan dikendalikan secara otomatis dan real-time.

Kelembaban udara yang optimal memiliki peran penting dalam menjaga kesehatan manusia, khususnya dalam ruangan ber-AC atau di daerah dengan cuaca kering. Udara yang terlalu kering dapat menyebabkan iritasi pada saluran pernapasan, kulit kering, hingga menurunnya kualitas tidur. Oleh karena itu, dibutuhkan alat yang mampu menjaga tingkat kelembaban udara secara stabil dan efisien. Namun, penggunaan air humidifier konvensional masih memiliki keterbatasan, seperti tidak adanya sistem pemantauan otomatis, kontrol manual yang kurang fleksibel, serta tidak adanya integrasi dengan sistem digital. Dengan memanfaatkan teknologi IoT, air humidifier dapat dirancang sedemikian rupa sehingga mampu membaca tingkat kelembaban udara melalui sensor, mengaktifkan atau menonaktifkan kabut secara otomatis, serta memberikan laporan kondisi secara langsung ke perangkat pengguna melalui aplikasi atau antarmuka digital lainnya.

Melalui proyek ini, penulis merancang dan membangun air humidifier berbasis IoT yang menggunakan mikrokontroler (seperti ESP32), sensor kelembaban (DHT22), modul mist maker, dan sistem kontrol berbasis aplikasi seperti Blynk. Dengan adanya sistem ini, pengguna tidak hanya dapat memantau kondisi lingkungan secara langsung, tetapi juga mengontrol perangkat secara remote dengan mudah, efisien, dan hemat energi. Dengan latar belakang tersebut, proyek ini diharapkan dapat menjadi solusi inovatif dalam menjaga kualitas udara dalam ruangan sekaligus sebagai implementasi nyata dari penerapan teknologi Internet of Things dalam kehidupan sehari-hari.

### Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam proyek ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem air humidifier yang dapat dikendalikan secara otomatis dan manual menggunakan teknologi IoT?
2. Bagaimana sistem dapat memantau dan menampilkan data kelembaban udara secara real-time?
3. Bagaimana integrasi sensor kelembaban dan perangkat mist maker dapat dilakukan secara efisien melalui platform IoT seperti Blynk?
4. Bagaimana sistem dapat memberikan kontrol jarak jauh dan umpan balik kondisi kelembaban melalui perangkat mobile?

### Batasan Masalah

Untuk memfokuskan ruang lingkup proyek dan laporan, maka batasan-batasan masalah yang ditetapkan adalah sebagai berikut:

1. Sistem hanya memantau dan mengendalikan tingkat kelembaban udara di dalam ruangan tertutup.
2. Mikrokontroler yang digunakan adalah ESP32 sebagai unit utama pengendali dan penghubung ke jaringan internet.
3. Sensor yang digunakan untuk membaca kelembaban dan suhu udara adalah DHT22.
4. Perangkat penghasil uap adalah mist maker yang dikendalikan melalui modul relay 5V.
5. Platform aplikasi yang digunakan untuk pemantauan dan pengendalian adalah Blynk IoT.
6. Sistem hanya mendukung koneksi melalui jaringan WiFi dan belum menggunakan komunikasi berbasis seluler atau LoRa.
7. Tidak membahas aspek perawatan dan konsumsi daya jangka panjang dari perangkat humidifier.

### Tujuan Penulisan

Adapun tujuan dari pembuatan proyek dan penyusunan laporan ini adalah:

1. Merancang dan membangun sebuah perangkat air humidifier berbasis Internet of Things (IoT) yang dapat bekerja secara otomatis dan manual.
2. Mengembangkan sistem monitoring kelembaban udara secara real-time menggunakan sensor DHT22.
3. Mengintegrasikan perangkat keras dengan aplikasi Blynk agar pengguna dapat mengontrol dan memantau sistem melalui smartphone.
4. Meningkatkan efisiensi dan kenyamanan pengguna dalam mengatur kelembaban udara di dalam ruangan.
5. Memberikan solusi praktis dalam pemanfaatan teknologi IoT untuk kebutuhan rumah tangga dan lingkungan kerja.

## 

**LANDASAN TEORI**

### Teori Umum

#### ESP 32

Gambar 2 . 1 : ESP32 C6 WROOM 1

ESP32 adalah sebuah mikrokontroler yang dikembangkan oleh Espressif Systems dan dirancang khusus untuk memenuhi kebutuhan aplikasi Internet of Things (IoT). Mikrokontroler ini merupakan penerus dari ESP8266 dan memiliki peningkatan signifikan dalam hal performa dan fitur. ESP32 dilengkapi dengan prosesor dual-core Xtensa® 32-bit LX6 yang dapat berjalan hingga 240 MHz serta mendukung konektivitas Wi-Fi 802.11 b/g/n dan Bluetooth v4.2 termasuk BLE (Bluetooth Low Energy). Dengan kombinasi fitur ini, ESP32 mampu menangani berbagai jenis aplikasi IoT seperti sistem pemantauan, perangkat rumah pintar, dan sistem otomatisasi industri.

Dari sisi hardware, ESP32 menyediakan lebih dari 30 pin GPIO yang dapat dikonfigurasi untuk berbagai fungsi seperti PWM, ADC, DAC, I2C, SPI, dan UART. Selain itu, modul ini juga dilengkapi dengan fitur tambahan seperti sensor sentuh kapasitif, hall effect sensor, serta modul RTC (Real Time Clock), yang menjadikannya sangat fleksibel untuk proyek-proyek berbasis sensor dan kendali jarak jauh. ESP32 umumnya memiliki memori internal sebesar 520 KB SRAM dan 4 MB flash (tergantung pada modul yang digunakan), serta kompatibel dengan banyak platform pemrograman seperti Arduino IDE, PlatformIO, dan MicroPython, sehingga sangat mudah diakses baik oleh pemula maupun pengembang profesional.

Salah satu keunggulan utama ESP32 adalah kemampuannya dalam pengelolaan konsumsi daya. Mikrokontroler ini memiliki beberapa mode hemat daya, seperti light sleep, deep sleep, dan hibernate, yang sangat berguna dalam aplikasi berbasis baterai yang membutuhkan waktu operasi panjang. Dengan menggunakan fitur deep sleep, ESP32 dapat beroperasi dalam konsumsi daya yang sangat rendah dan hanya aktif saat dibutuhkan, misalnya ketika menerima sinyal dari sensor atau alarm waktu. Oleh karena itu, ESP32 sangat cocok digunakan untuk perangkat seperti weather station, alat pelacak, dan sistem alarm berbasis IoT. Dukungan komunitas dan dokumentasi yang luas semakin memperkuat posisinya sebagai salah satu mikrokontroler paling populer di dunia saat ini.

#### DHT 22

Gambar 2 . 2 : Sensor DHT22

DHT22, atau dikenal juga sebagai AM2302, adalah sensor digital yang dirancang untuk mengukur suhu dan kelembaban secara akurat dan stabil. Sensor ini merupakan versi yang lebih presisi dan memiliki jangkauan yang lebih luas dibandingkan saudaranya, DHT11. DHT22 dapat mengukur suhu dalam rentang -40°C hingga +80°C dengan akurasi sekitar ±0.5°C, serta kelembaban dalam rentang 0% hingga 100% RH dengan akurasi sekitar ±2–5% RH. Sensor ini bekerja dengan protokol komunikasi digital satu kabel (single-wire), yang memungkinkan komunikasi data secara efisien antara sensor dan mikrokontroler seperti Arduino atau ESP32. Karena itu, sensor ini sangat cocok digunakan dalam berbagai aplikasi seperti stasiun cuaca, sistem pendingin otomatis, rumah pintar, dan pemantauan lingkungan.

Sensor DHT22 memiliki elemen penginderaan kelembaban kapasitif dan termistor untuk pengukuran suhu. Setiap beberapa detik, sensor membaca nilai dari elemen-elemen ini dan mengirimkan data dalam bentuk sinyal digital. Keunggulan DHT22 dibandingkan sensor lainnya terletak pada kestabilan datanya serta waktu respons yang relatif cepat, yaitu sekitar 2 detik untuk pembaruan data. Selain itu, sensor ini sudah dikalibrasi di pabrik, sehingga pengguna tidak perlu melakukan kalibrasi manual sebelum penggunaan. Pengoperasian DHT22 juga sangat hemat daya, yang menjadikannya ideal untuk aplikasi berbasis baterai atau perangkat IoT yang harus berjalan dalam jangka waktu panjang.

Dalam hal koneksi perangkat keras, DHT22 biasanya memiliki empat pin, namun pada versi breakout board umumnya hanya tiga pin yang digunakan: VCC, GND, dan Data. Tegangan operasionalnya berkisar antara 3.3V hingga 6V, sehingga dapat dengan mudah diintegrasikan dengan mikrokontroler 3.3V seperti ESP32 atau 5V seperti Arduino Uno. Data yang dikirim dari DHT22 terdiri dari 40 bit, yang mencakup 16 bit data kelembaban, 16 bit data suhu, dan 8 bit checksum untuk verifikasi keakuratan data. Karena menggunakan satu jalur komunikasi, penting untuk memberikan waktu tunda (delay) antar pembacaan untuk menghindari kesalahan data.

Secara keseluruhan, DHT22 adalah sensor yang seimbang antara biaya, akurasi, dan kemudahan integrasi. Meski tidak secepat sensor industri, DHT22 memberikan performa yang lebih dari cukup untuk kebanyakan aplikasi amatir hingga semi-profesional. Sensor ini telah menjadi standar dalam proyek-proyek pengukuran suhu dan kelembaban karena dokumentasinya yang luas dan dukungan dari komunitas maker serta pengembang. Dengan pemrograman sederhana dan ketersediaan pustaka (library) seperti DHT.h untuk Arduino atau Adafruit\_DHT untuk Python, integrasi sensor ini menjadi sangat mudah bahkan bagi pemula.

#### LCD 1602 I2C

Gambar 2 . 3 : LCD 1602 with I2C interface

LCD 1602 I2C adalah modul tampilan berbasis Liquid Crystal Display (LCD) yang dirancang untuk menampilkan karakter dalam format 2 baris × 16 kolom (total 32 karakter). Modul ini merupakan versi hemat pin dari LCD 1602 standar karena dilengkapi dengan interface I2C (Inter-Integrated Circuit), yang mengurangi jumlah pin yang diperlukan dari 16 menjadi hanya 4 (VCC, GND, SDA, dan SCL). Hal ini membuatnya sangat praktis dan efisien dalam proyek mikrokontroler seperti Arduino atau ESP32, terutama jika pengguna memiliki keterbatasan jumlah pin I/O. Modul I2C biasanya menggunakan chip ekspander PCF8574 yang berfungsi sebagai perantara antara mikrokontroler dan LCD, mentranslasi komunikasi I2C menjadi kontrol paralel untuk LCD.

Dari sisi teknis, LCD 1602 mampu menampilkan huruf, angka, dan beberapa karakter khusus ASCII, namun tidak bisa menampilkan grafis atau gambar. Modul ini menggunakan pengontrol HD44780, yang merupakan standar industri untuk LCD karakter. Konektivitas I2C memungkinkan komunikasi dua kabel melalui pin SDA (data) dan SCL (clock), serta dapat disesuaikan alamatnya melalui jumper di belakang modul, sehingga beberapa LCD I2C dapat digunakan sekaligus dalam satu sistem tanpa konflik alamat. Tegangan kerja umum dari modul ini adalah 5V, tetapi sebagian mikrokontroler seperti ESP32 yang berbasis 3.3V masih dapat menggunakannya dengan level shifter atau langsung jika modul mendukung.

Keuntungan utama menggunakan LCD 1602 I2C adalah penghematan ruang, pin, dan kabel, serta kemudahan integrasi. Banyak pustaka (library) yang tersedia untuk mendukung penggunaannya, seperti LiquidCrystal\_I2C.h di Arduino IDE, yang menyederhanakan proses pemrograman. Dengan beberapa baris kode, pengguna dapat dengan mudah menampilkan teks, mengatur kursor, dan menghapus layar. Selain itu, LCD 1602 memiliki potensiometer bawaan untuk mengatur kontras layar, dan LED backlight yang memungkinkan visibilitas di lingkungan dengan pencahayaan rendah. Oleh karena itu, modul ini sering digunakan dalam sistem monitoring suhu, tampilan status perangkat, proyek robotika, dan antarmuka pengguna dalam proyek DIY.

Secara keseluruhan, LCD 1602 I2C adalah solusi tampilan yang ekonomis, handal, dan mudah digunakan untuk berbagai aplikasi mikrokontroler. Dibandingkan dengan layar OLED atau TFT yang lebih kompleks dan mahal, LCD 1602 I2C tetap menjadi pilihan utama bagi banyak pengembang karena kestabilannya dan konsumsi daya yang rendah. Dukungan luas dari komunitas maker dan dokumentasi online juga menjadikan modul ini ramah bagi pemula. Dengan kemampuan menampilkan data secara real-time dan kemampuan antarmuka I2C yang efisien, LCD 1602 I2C adalah pilihan ideal untuk proyek-proyek sistem tertanam yang memerlukan informasi visual secara sederhana dan efektif.

#### Mist Maker



Gambar 2 . 4 : Mist Maker module

Mist maker, atau disebut juga ultrasonic humidifier/fogger, adalah sebuah alat elektronik yang digunakan untuk menghasilkan kabut air (mist) melalui proses ultrasonik. Alat ini biasanya tidak menggunakan panas, melainkan memanfaatkan frekuensi ultrasonik tinggi (sekitar 1,7 MHz) untuk menggetarkan permukaan air dengan intensitas sangat cepat sehingga membentuk partikel air mikroskopis yang tampak seperti kabut. Teknologi ini umum digunakan dalam berbagai aplikasi seperti sistem pelembap udara (humidifier), efek visual kabut pada taman atau akuarium, hingga pada sistem pertanian seperti aeroponik dan rumah kaca untuk menjaga kelembaban.

Cara kerja mist maker cukup sederhana namun efektif. Di dalamnya terdapat transduser piezoelektrik yang berfungsi sebagai komponen penggetar. Ketika diberi arus listrik berfrekuensi tinggi, transduser ini berosilasi dan menciptakan gelombang tekanan di dalam air. Ketika tekanan ini cukup tinggi, air terpecah menjadi partikel halus dan tersebar ke udara tanpa menaikkan suhu air secara signifikan. Karena menggunakan metode non-termal, mist maker sangat cocok digunakan pada lingkungan atau sistem yang sensitif terhadap perubahan suhu, seperti inkubator, sistem penetas telur, atau area tanam hidroponik.

Mist maker umumnya tidak memiliki sensor internal untuk mengukur kabut atau air, tetapi dapat dikombinasikan dengan sensor kelembaban (seperti DHT22), sensor level air, atau relay untuk mengatur operasinya secara otomatis dalam sistem IoT atau otomatisasi rumah. Dalam pengaplikasiannya, modul mist maker sering dihubungkan dengan power supply 24V DC dan dikendalikan melalui mikrokontroler seperti ESP32 menggunakan relay modul. Keamanan juga menjadi perhatian penting, karena kabut yang dihasilkan bisa menyebabkan kelembaban berlebih yang berdampak pada komponen elektronik jika tidak diatur dengan baik.

Secara keseluruhan, mist maker adalah alat yang efisien, hening, dan ramah lingkungan untuk menghasilkan kabut atau uap air tanpa pemanasan. Selain fungsinya dalam menjaga kelembaban, kabut yang dihasilkannya juga dapat menciptakan suasana visual menarik pada dekorasi ruangan atau taman. Di dunia otomasi dan IoT, mist maker sering dikombinasikan dengan sistem kendali berbasis mikrokontroler dan sensor untuk menciptakan solusi cerdas seperti humidifier otomatis, sistem irigasi kabut, atau efek visual dinamis. Dukungan dari komunitas pembuat dan tersedianya banyak proyek open-source menjadikan mist maker komponen yang sangat fleksibel untuk eksplorasi dan inovasi.

#### Relay SRD-05VDC-SL-C

Gambar 2 . 5 : Relay module

Modul relay tipe SRD-05VDC-SL-C adalah salah satu jenis relay elektromekanis yang paling umum digunakan dalam proyek-proyek elektronika dan Internet of Things (IoT). Modul ini menggunakan tegangan kendali 5V DC untuk mengaktifkan saklar internal yang dapat menghubungkan atau memutuskan arus listrik ke perangkat berdaya lebih tinggi seperti pompa, lampu, atau dalam kasus kamu, sebuah mist maker. Relay ini biasanya digunakan bersama dengan mikrokontroler seperti ESP32, Arduino, atau Raspberry Pi, karena dapat dikontrol hanya dengan sinyal digital (HIGH/LOW) dari GPIO.

Modul ini terdiri dari beberapa bagian penting, yaitu komponen relay utama, indikator LED, dan driver transistor. LED merah dan hijau pada modul berfungsi sebagai indikator status: salah satu menunjukkan bahwa daya telah masuk ke modul, dan yang lainnya menunjukkan bahwa relay sedang aktif (ON). Modul ini juga dilengkapi dengan optocoupler (pada beberapa varian) untuk melindungi mikrokontroler dari lonjakan tegangan balik (back EMF) saat saklar bekerja, walaupun tidak semua versi menyertakannya.

Di bagian output, modul ini memiliki 3 terminal koneksi utama: NO (Normally Open), NC (Normally Closed), dan COM (Common). Saat relay tidak aktif, COM terhubung ke NC. Ketika relay diaktifkan (diberi sinyal HIGH), COM akan berpindah terhubung ke NO. Dengan mekanisme ini, pengguna dapat memilih apakah ingin mengontrol perangkat agar menyala saat relay aktif (NO) atau saat relay tidak aktif (NC). Pada sistem humidifier kamu, biasanya COM dihubungkan ke VCC sumber daya, dan NO ke mist maker, sehingga mist maker hanya aktif saat relay diaktifkan oleh mikrokontroler.

Dalam konteks proyek Air Humidifier berbasis ESP32, relay ini berperan sebagai saklar utama untuk menghidupkan atau mematikan mist maker berdasarkan logika dari pembacaan sensor kelembapan DHT22. Ketika sistem dalam mode otomatis, ESP32 akan mengevaluasi nilai kelembapan dan memutuskan apakah perlu mengaktifkan relay. Dalam mode manual, status relay akan mengikuti input dari pengguna via aplikasi Blynk. Penggunaan relay ini memungkinkan ESP32 yang hanya mampu memberikan arus kecil untuk mengontrol beban yang lebih besar dengan aman dan efisien.

#### Power Supply Breadboard MB102 3.3-5V

Gambar 2 . 6 : Power Supply Module

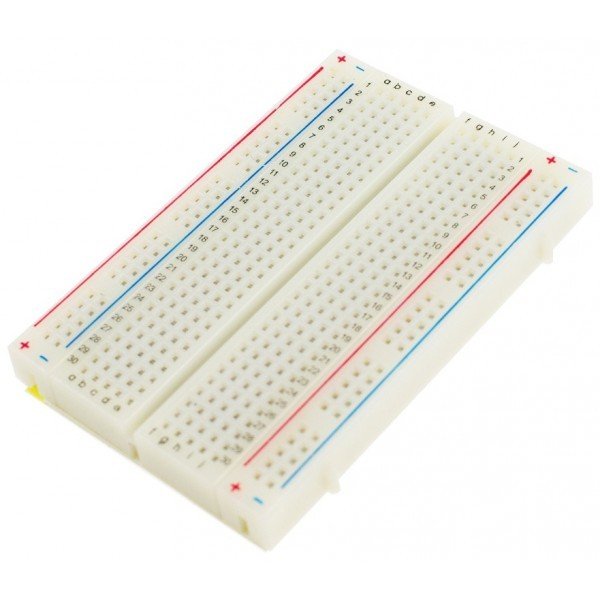
Power Supply Breadboard MB102 adalah sebuah modul catu daya mini yang dirancang khusus untuk digunakan bersama breadboard standar, terutama breadboard tipe 830 titik. Modul ini memiliki kemampuan untuk menyuplai tegangan 3.3V dan 5V secara langsung ke jalur distribusi (rail) breadboard, sehingga sangat memudahkan proses pengujian dan pengembangan rangkaian elektronik berbasis mikrokontroler seperti Arduino, ESP32, dan sensor lainnya. MB102 mendapatkan pasokan listrik melalui konektor DC barrel jack (7–12V DC) atau melalui port USB, dan kemudian menyalurkannya melalui regulator onboard seperti AMS1117 untuk menghasilkan tegangan output yang stabil.

Modul ini memiliki dua saluran keluaran utama yang dapat dikonfigurasi secara independen menggunakan jumper atau saklar geser kecil. Dengan kata lain, pengguna dapat mengatur salah satu sisi breadboard bekerja pada 5V, sementara sisi lainnya bekerja pada 3.3V, tergantung kebutuhan rangkaian. Keunggulan ini menjadikan MB102 sangat fleksibel, terutama dalam proyek yang melibatkan komponen dengan tegangan kerja berbeda, misalnya ESP32 (3.3V) dan beberapa sensor atau modul relay (5V). Modul ini juga biasanya dilengkapi dengan dua header keluaran tambahan, sehingga pengguna dapat mengakses 3.3V dan 5V secara langsung tanpa melalui breadboard.

Secara internal, MB102 menggunakan regulator linear AMS1117, yang cukup populer karena kesederhanaannya dan kestabilan outputnya. Namun, perlu diperhatikan bahwa regulator ini memiliki efisiensi rendah karena sifatnya sebagai regulator linear — artinya sebagian energi dibuang dalam bentuk panas, terutama jika inputnya jauh lebih tinggi dari output (misalnya 12V input untuk 3.3V output). Karena itu, disarankan untuk tidak menarik arus terlalu besar dari MB102 (biasanya maksimal 800 mA untuk 5V dan sekitar 300–500 mA untuk 3.3V) agar tidak terjadi overheat atau kerusakan pada modul.

Dalam praktiknya, MB102 sangat berguna dalam prototipe sirkuit elektronik, terutama dalam tahap pengujian awal. Modul ini menghemat waktu dan ruang karena tidak memerlukan rangkaian catu daya terpisah, dan pengguna cukup mencolokkan langsung ke breadboard. Namun, karena keterbatasan arus dan efisiensi regulatornya, MB102 tidak disarankan untuk sistem yang membutuhkan konsumsi daya besar secara terus-menerus. Meskipun demikian, untuk keperluan edukasi, pembelajaran, dan eksperimen mikrokontroler sederhana, MB102 tetap menjadi salah satu pilihan power supply terbaik di kalangan maker dan hobiis.

#### Breadboard



Gambar 2 . 7 : Breadboard

Breadboard adalah papan tempat merangkai komponen elektronik tanpa perlu menyolder, sehingga sangat ideal untuk keperluan prototyping atau eksperimen rangkaian elektronik. Breadboard memungkinkan pengguna menyusun dan menguji berbagai konfigurasi rangkaian secara cepat dan fleksibel. Pada dasarnya, breadboard terdiri dari lubang-lubang kecil yang tersusun dalam pola grid, di mana setiap lubang memiliki koneksi internal tertentu yang menghubungkan komponen secara elektrik. Breadboard banyak digunakan oleh pelajar, mahasiswa, hobiis, dan insinyur dalam tahap awal pengembangan proyek elektronik.

Struktur breadboard terdiri dari dua area utama: area terminal (pada bagian tengah) dan jalur daya (power rails) yang biasanya terletak di sisi atas dan bawah. Lubang-lubang pada area terminal terhubung secara vertikal dalam kolom 5 lubang, yang cocok untuk menghubungkan kaki komponen seperti resistor, IC, dan kapasitor. Sementara itu, power rail tersusun secara horizontal dan biasanya digunakan untuk menyuplai tegangan (+) dan ground (–) dari catu daya ke seluruh rangkaian. Desain ini memungkinkan pengguna menyusun rangkaian secara modular dan terorganisir.

Breadboard tersedia dalam berbagai ukuran, seperti mini, half-size, dan full-size, dengan versi full-size umumnya memiliki lebih dari 800 titik koneksi. Untuk proyek besar, beberapa breadboard dapat disatukan dengan menghubungkan sisi-sisinya. Meskipun praktis, breadboard juga memiliki keterbatasan, seperti koneksi yang tidak sekuat solder dan resistansi kontak yang bisa mempengaruhi performa pada frekuensi tinggi. Oleh karena itu, breadboard biasanya digunakan hanya pada tahap awal, sebelum rangkaian final dipindahkan ke PCB (Printed Circuit Board).

Secara keseluruhan, breadboard adalah alat yang esensial dan sangat berguna dalam dunia elektronika, khususnya dalam pembelajaran dan pembuatan prototipe. Kelebihannya yang memungkinkan perubahan rangkaian secara instan tanpa merusak komponen menjadikan breadboard sangat efisien untuk uji coba ide-ide baru. Pengguna juga dapat menggabungkan breadboard dengan modul lain seperti power supply MB102, sensor, dan mikrokontroler (misalnya Arduino atau ESP32) untuk membangun sistem elektronik yang kompleks. Kombinasi fleksibilitas, kemudahan, dan efisiensi menjadikan breadboard alat dasar yang hampir selalu hadir di meja kerja setiap pembuat rangkaian elektronik.

#### Jumper Cable

Gambar 2 . 8 : Jumper Cable

Jumper cable atau kabel jumper adalah kabel pendek yang digunakan untuk menghubungkan komponen elektronik pada breadboard, modul, atau mikrokontroler tanpa perlu menyolder. Kabel ini sangat umum digunakan dalam tahap perakitan prototipe dan eksperimen sirkuit elektronik karena praktis, dapat digunakan ulang, dan mendukung koneksi yang cepat antar komponen. Kabel jumper hadir dalam berbagai tipe konektor, seperti male-to-male, male-to-female, dan female-to-female, yang masing-masing digunakan sesuai dengan jenis pin atau soket dari perangkat yang hendak dihubungkan.

Kabel jumper biasanya terdiri dari konduktor tembaga di dalam isolasi plastik fleksibel dan memiliki ujung konektor berbentuk pin atau soket yang kompatibel dengan lubang breadboard dan header mikrokontroler. Dalam praktiknya, male-to-male adalah jenis paling umum digunakan untuk menghubungkan titik-titik pada breadboard. Sementara itu, female-to-male biasanya digunakan untuk menghubungkan breadboard ke pin mikrokontroler seperti ESP32 atau Arduino, dan female-to-female sering dipakai untuk menghubungkan dua header pin langsung, seperti modul sensor dengan papan pengendali.

Penggunaan jumper cable sangat membantu dalam membangun sistem elektronik modular. Karena tidak memerlukan proses penyolderan, pengguna dapat dengan mudah mengganti, mengatur ulang, atau memperbaiki rangkaian jika terjadi kesalahan koneksi. Ini sangat penting dalam pembelajaran dan pengembangan sistem tertanam (embedded system), seperti pada proyek mikrokontroler yang melibatkan sensor, aktuator, dan layar. Namun, karena sifat koneksinya yang tidak permanen, kabel jumper dapat mudah terlepas jika tidak terpasang dengan kuat, sehingga pada tahap akhir proyek sering diganti dengan koneksi yang lebih stabil seperti solder atau konektor permanen.

Secara keseluruhan, jumper cable adalah komponen vital dalam dunia elektronika, terutama untuk eksperimen dan prototyping. Kabel ini memungkinkan koneksi yang cepat, fleksibel, dan efisien antar berbagai perangkat elektronik. Dalam sistem berbasis breadboard dan mikrokontroler, penggunaan kabel jumper mempercepat proses pengujian dan debugging tanpa mengorbankan komponen. Dengan banyaknya varian panjang dan warna, jumper cable juga membantu dalam manajemen kabel dan pelacakan jalur koneksi, sehingga mempermudah analisis dan pemeliharaan rangkaian.

### Teori Khusus

#### Internet of Things ( IoT )

Internet of Things (IoT) adalah konsep di mana berbagai perangkat fisik dihubungkan ke internet dan dapat saling berkomunikasi, mengumpulkan, serta bertukar data secara otomatis. Perangkat-perangkat ini bisa berupa sensor, aktuator, perangkat rumah tangga, kendaraan, mesin industri, hingga sistem kesehatan. IoT memungkinkan perangkat-perangkat tersebut menjadi "pintar" karena mampu mengambil keputusan berbasis data secara real-time tanpa campur tangan manusia secara langsung. Dengan kata lain, IoT menciptakan ekosistem di mana dunia fisik dan digital saling terintegrasi.

IoT bekerja melalui tiga komponen utama: sensor atau perangkat, konektivitas, dan pemrosesan data. Sensor digunakan untuk mengumpulkan informasi dari lingkungan (seperti suhu, kelembaban, gerakan), lalu mengirimkan data ini ke server atau cloud melalui jaringan Wi-Fi, Bluetooth, LoRa, atau seluler. Setelah data sampai di cloud atau pusat pemrosesan, informasi ini dianalisis dan digunakan untuk memberikan respons, baik berupa notifikasi ke pengguna maupun perintah ke perangkat lain, seperti menghidupkan kipas saat suhu naik. Contoh sederhana dari sistem ini adalah smart home, di mana lampu bisa menyala otomatis saat penghuni memasuki ruangan.

Implementasi IoT tidak terbatas hanya pada rumah pintar. Di sektor industri, IoT menjadi fondasi dari Industri 4.0, di mana mesin-mesin produksi saling terhubung dan dapat melaporkan status operasionalnya secara real-time. Di bidang pertanian, petani menggunakan IoT untuk memonitor kelembaban tanah dan mengatur irigasi otomatis. Dalam bidang kesehatan, perangkat IoT seperti wearable device memantau detak jantung atau aktivitas pasien secara terus menerus, sehingga memudahkan diagnosis dan perawatan preventif. Potensi penerapan IoT sangat luas karena bisa disesuaikan dengan berbagai kebutuhan dan sektor.

Namun, meskipun IoT menawarkan banyak keuntungan seperti efisiensi, otomatisasi, dan pengambilan keputusan berbasis data, ia juga menimbulkan tantangan. Salah satu tantangan utama adalah keamanan data, karena banyak perangkat yang terhubung dan mengirimkan informasi sensitif melalui internet. Selain itu, interoperabilitas antar perangkat dari berbagai produsen, ketergantungan pada koneksi internet, serta biaya infrastruktur awal juga menjadi isu penting yang perlu diperhatikan dalam implementasi IoT secara luas.

Secara keseluruhan, Internet of Things merupakan salah satu inovasi teknologi paling signifikan di era digital. IoT mendorong perubahan cara manusia berinteraksi dengan teknologi, dari sekadar mengendalikan perangkat menjadi menciptakan sistem yang dapat memahami dan merespons kondisi secara otomatis. Dengan perkembangan teknologi jaringan seperti 5G, edge computing, dan artificial intelligence, masa depan IoT diprediksi akan semakin canggih dan terintegrasi. Oleh karena itu, pemahaman tentang IoT menjadi sangat penting, terutama bagi pengembang, insinyur, pelaku bisnis, dan masyarakat umum yang ingin memanfaatkan teknologi ini dalam kehidupan sehari-hari.

#### Blynk

Blynk adalah sebuah platform Internet of Things (IoT) yang memungkinkan pengguna untuk membuat antarmuka grafis (GUI) guna mengontrol dan memantau perangkat berbasis mikrokontroler seperti ESP32, ESP8266, Arduino, dan Raspberry Pi melalui smartphone. Aplikasi Blynk tersedia di Android dan iOS, serta didukung oleh server berbasis cloud maupun lokal. Dengan antarmuka drag-and-drop, Blynk memudahkan pengguna, bahkan yang tidak memiliki pengalaman pemrograman antarmuka, untuk membangun dashboard interaktif yang dapat mengendalikan perangkat secara real-time dari mana saja.

Blynk bekerja dengan arsitektur tiga bagian utama: aplikasi Blynk (mobile app), Blynk server, dan Blynk library (yang diinstal di mikrokontroler). Aplikasi Blynk berfungsi sebagai user interface yang digunakan pengguna untuk berinteraksi dengan perangkat IoT. Server Blynk bertugas sebagai perantara komunikasi antara aplikasi dan perangkat. Sedangkan Blynk library ditanamkan pada perangkat seperti ESP32 agar mampu menerima dan mengirimkan perintah ke server. Blynk menggunakan sistem token otentikasi yang unik untuk setiap proyek, sehingga koneksi antara aplikasi dan perangkat aman dan spesifik.

Salah satu kekuatan utama dari Blynk adalah fleksibilitasnya. Pengguna dapat dengan mudah menambahkan widget seperti tombol, slider, notifikasi, grafik data (chart), dan kontrol pin virtual ke dalam aplikasi. Hal ini memungkinkan implementasi berbagai proyek seperti sistem monitoring suhu dan kelembaban, otomatisasi rumah (smart home), pemantauan energi, hingga irigasi otomatis di bidang pertanian. Karena mendukung berbagai protokol seperti Wi-Fi, Ethernet, dan Bluetooth, Blynk sangat cocok digunakan pada berbagai perangkat keras dan kebutuhan koneksi.

Selain versi klasik (Blynk Legacy), kini Blynk telah merilis Blynk IoT (New Blynk Platform) dengan banyak peningkatan, termasuk dukungan untuk dashboard berbasis web, fitur multi-perangkat, manajemen pengguna dan organisasi, serta integrasi cloud yang lebih kuat. Versi baru ini lebih ditujukan bagi pengembang profesional dan perusahaan yang ingin membangun solusi IoT skala besar. Pengguna dapat memilih antara menggunakan server cloud Blynk atau menghosting server mereka sendiri secara lokal untuk kebutuhan data privasi dan kendali penuh atas sistem.

Secara keseluruhan, Blynk adalah solusi yang sangat ideal bagi para pengembang IoT, baik pemula maupun profesional, yang ingin membangun sistem kontrol jarak jauh dengan cepat, efisien, dan tanpa kompleksitas pengembangan front-end. Dengan dokumentasi yang lengkap, komunitas yang aktif, dan kompatibilitas luas dengan berbagai board mikrokontroler, Blynk terus menjadi salah satu platform IoT paling populer di dunia. Hal ini menjadikannya alat penting dalam pendidikan, penelitian, hingga pengembangan produk komersial berbasis IoT

#### Rancang Bangun Berbasis IoT

Rancang bangun berbasis IoT adalah proses perancangan dan pengembangan sistem yang memanfaatkan teknologi Internet of Things (IoT) untuk menghubungkan berbagai perangkat fisik agar dapat saling berkomunikasi melalui internet. Dalam konteks ini, perangkat seperti sensor, aktuator, mikrokontroler (contohnya ESP32 atau Arduino), dan perangkat lunak pengendali saling terintegrasi untuk mengumpulkan, mengirimkan, dan menganalisis data secara otomatis. Tujuan utama dari rancang bangun ini adalah menciptakan solusi yang efisien, responsif, dan terotomatisasi untuk berbagai kebutuhan di bidang rumah tangga, industri, kesehatan, pertanian, hingga transportasi.

Proses perancangan sistem IoT umumnya dimulai dari identifikasi permasalahan, lalu dilanjutkan dengan pemilihan perangkat keras (hardware) seperti sensor dan board mikrokontroler, perancangan sistem komunikasi data (Wi-Fi, LoRa, atau lainnya), serta pembangunan platform kendali seperti aplikasi Blynk, Node-RED, atau dashboard berbasis web. Selain itu, aspek pemrograman menjadi bagian penting dari rancang bangun, karena kode program bertugas mengatur cara kerja perangkat, pengambilan data, pengiriman ke cloud, hingga eksekusi aksi berdasarkan kondisi tertentu. Seluruh komponen ini harus dirancang agar bekerja secara sinkron dan stabil.

Keunggulan utama dari sistem rancang bangun berbasis IoT terletak pada kemampuan monitoring dan kontrol jarak jauh, serta fleksibilitasnya dalam menangani data real-time. Misalnya, dalam sistem pemantauan suhu dan kelembaban ruangan, data dari sensor DHT22 dikirim ke server melalui ESP32, lalu divisualisasikan melalui aplikasi Blynk. Sistem ini bisa dikembangkan lebih lanjut dengan menambahkan kontrol otomatis seperti menyalakan kipas saat suhu tinggi. Rancang bangun seperti ini tidak hanya meningkatkan efisiensi operasional, tapi juga meminimalisir kebutuhan intervensi manusia secara manual.

Namun, dalam implementasinya, rancang bangun IoT juga menghadapi beberapa tantangan, seperti keamanan data, kestabilan koneksi internet, interoperabilitas antar perangkat, serta manajemen konsumsi daya. Oleh karena itu, proses perancangan harus dilakukan dengan memperhatikan aspek keamanan, skalabilitas, dan keandalan sistem.

## 

**IMPLEMENTASI DAN PEMBAHASAN**

### Wiring Diagram

#### Komponen yang Digunakan

Tabel 3. 1 : Komponen

|  |  |
| --- | --- |
| **Komponen** | **Keterangan** |
| ESP32-C6-WROOM-1-N8 | Mikrokontroler utama |
| DHT 22 | Sensor suhu dan kelembaban |
| LCD 16x2 I2C | Menampilkan suhu, kelembaban, status |
| Relay 5V (SRD-05VDC-SL-C) | Mengontrol Mist Maker |
| Mist Maker (PCB Fogger) | Penghasil kabut |
| Breadboard + Kabel jumper | Untuk koneksi prototipe |
| Power Supply Breadboard MB102 3.3-5V | Supply daya listrik |

#### Skema Pengkabelan

##### ESP32 ke Sensor DHT22

Tabel 3. 2 : DHT22 Wiring

|  |  |
| --- | --- |
| **DHT22** | **ESP32** |
| VCC | 5V |
| GND | GND |
| Data | GPIO 4 |

##### ESP32 ke LCD I2C 16x2

Tabel 3. 3 : LCD Wiring

|  |  |
| --- | --- |
| **LCD I2C** | **ESP32** |
| VCC | 5V |
| GND | GND |
| SDA | GPIO 21 |
| SCL | GPIO 22 |

##### ESP32 ke Relay

Tabel 3. 4 : Relay Wiring

|  |  |
| --- | --- |
| **Relay** | **ESP32** |
| VCC | 5V |
| GND | GND |
| IN | GPIO 5 |
| NO | GND mist maker |
| COM | GND |
| NC | - |

##### Power Supply 5V ke Mist Maker

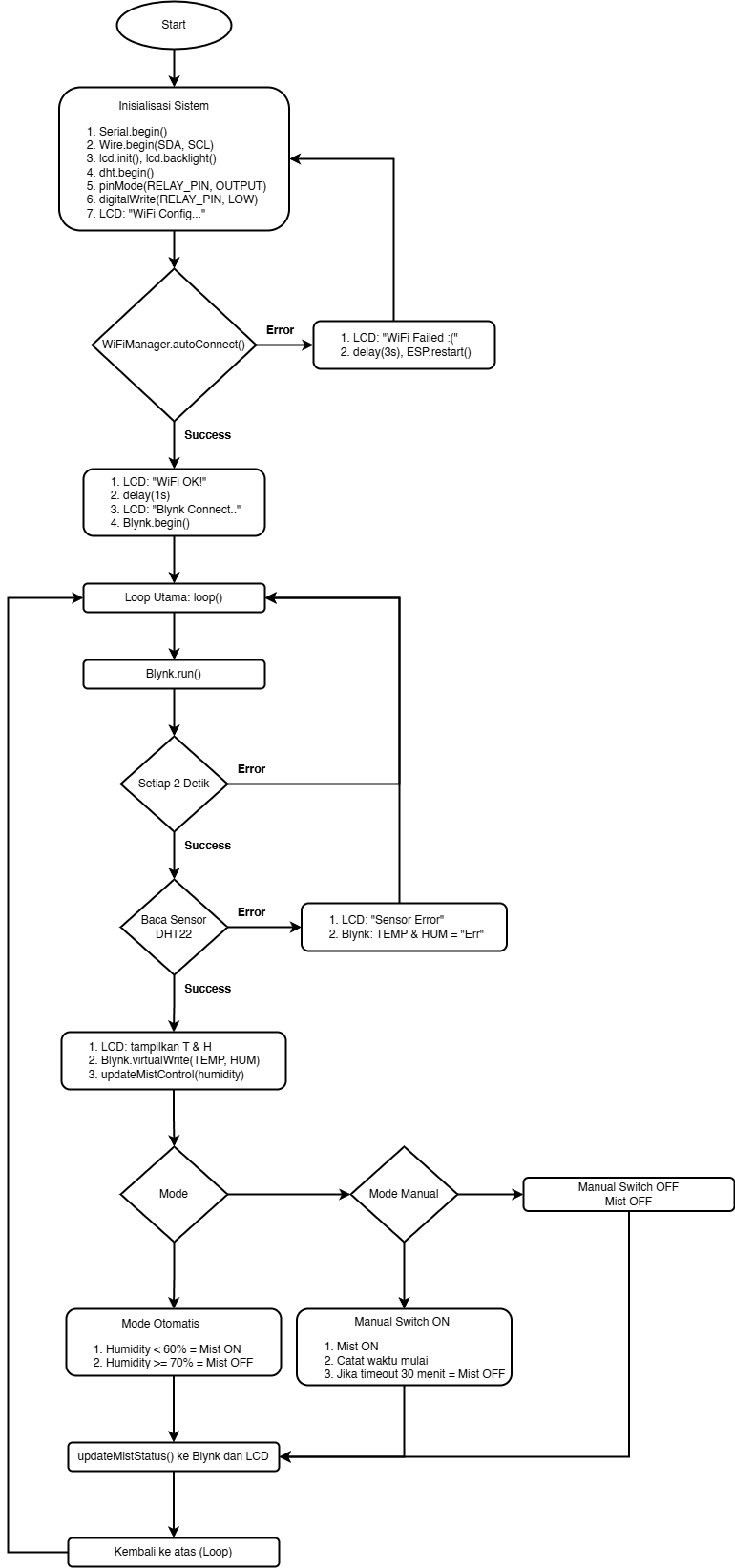
Tabel 3. 5 : Mist Maker Wiring

|  |  |
| --- | --- |
| **Mist Maker (PCB fogger)** | **Power Supply** |
| VCC | 5V power supply |
| GND | NO Relay |

#### Sketsa Gambar

Gambar 3 . 1 : Sketsa Gambar

#### Flowchart



### Source Code

#### Platformio.ini

|  |
| --- |
| [env:esp32-c6-devkitc-1]  platform = espressif32  board = esp32-c6-devkitc-1  framework = arduino  monitor\_speed = 115200  lib\_deps =  knolleary/PubSubClient@^2.8  adafruit/DHT sensor library@^1.4.6  blynkkk/Blynk@^1.3.2  marcoschwartz/LiquidCrystal\_I2C@^1.1.4  iakop/LiquidCrystal\_I2C\_ESP32@^1.1.6  tzapu/WiFiManager@^2.0.17 |

Penjelasan :

1. [env:esp32-c6-devkitc-1]

Bagian ini mendefinisikan lingkungan build (build environment) untuk board ESP32-C6 DevKitC-1. Semua parameter di bawahnya akan digunakan untuk proses kompilasi, upload, dan monitoring serial.

1. platform = espressif32

Baris ini menunjukkan bahwa proyek menggunakan platform pengembangan Espressif32, yang mendukung berbagai varian board berbasis ESP32. PlatformIO akan mengunduh toolchain dan SDK yang sesuai untuk ESP32.

1. board = esp32-c6-devkitc-1

Menentukan jenis board yang digunakan, yaitu ESP32-C6 DevKitC-1. Informasi ini penting karena akan menentukan konfigurasi pin, flash size, serta parameter lainnya yang spesifik untuk board tersebut.

1. framework = Arduino

Menginstruksikan PlatformIO untuk menggunakan Arduino framework, yang artinya Anda menulis program menggunakan sintaks dan pustaka standar Arduino. Ini adalah pilihan umum karena banyak pustaka dan contoh kode tersedia untuk Arduino.

1. monitor\_speed = 115200

Menentukan kecepatan serial monitor untuk debugging via USB. Nilai 115200 adalah standar umum dan harus sesuai dengan Serial.begin() di dalam kode sumber Anda agar data terlihat dengan benar.

1. lib\_deps =

Bagian ini mendeklarasikan semua dependensi library eksternal yang diperlukan oleh proyek. PlatformIO akan otomatis mengunduh dan menginstalnya saat Anda build pertama kali. Berikut penjelasan setiap pustaka:

1. [knolleary/PubSubClient@^2.8](mailto:knolleary/PubSubClient@%5e2.8)

Library MQTT yang umum digunakan untuk komunikasi IoT berbasis publish-subscribe. Cocok untuk koneksi ke broker seperti Mosquitto, HiveMQ, atau Cloud MQTT.

1. adafruit/DHT sensor [library@^1.4.6](mailto:library@%5e1.4.6)

Pustaka resmi dari Adafruit untuk membaca data dari sensor DHT11, DHT22, dan sejenisnya (untuk suhu dan kelembaban).

1. [blynkkk/Blynk@^1.3.2](mailto:blynkkk/Blynk@%5e1.3.2)

Library untuk berkomunikasi dengan Blynk platform, baik versi klasik maupun IoT. Digunakan untuk membuat dashboard kontrol berbasis mobile.

1. [marcoschwartz/LiquidCrystal\_I2C@^1.1.4](mailto:marcoschwartz/LiquidCrystal_I2C@%5e1.1.4)

Library untuk mengendalikan LCD 1602/2004 via antarmuka I2C. Digunakan jika Anda memakai display LCD dengan backpack I2C.

1. [iakop/LiquidCrystal\_I2C\_ESP32@^1.1.6](mailto:iakop/LiquidCrystal_I2C_ESP32@%5e1.1.6)

Ini adalah versi modifikasi dari LiquidCrystal\_I2C khusus untuk ESP32 agar lebih kompatibel dan stabil. Anda tampaknya menggabungkan dua versi pustaka ini—pastikan hanya satu yang aktif agar tidak konflik.

1. [tzapu/WiFiManager@^2.0.17](mailto:tzapu/WiFiManager@%5e2.0.17)

Digunakan untuk pengaturan koneksi Wi-Fi otomatis, sehingga pengguna tidak perlu hard-code SSID/password Wi-Fi. Saat tidak ada koneksi, WiFiManager akan membuka portal konfigurasi berbasis web.

#### Main.cpp

|  |
| --- |
| #define BLYNK\_TEMPLATE\_ID "TMPL6PJEDLLXx"  #define BLYNK\_TEMPLATE\_NAME "Humidifier"  #define BLYNK\_AUTH\_TOKEN "WizPCfkBq07EVvxT-\_NP3X12bZr0dTNC"  #include <Arduino.h>  #include <WiFi.h>  #include <BlynkSimpleEsp32.h>  #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  #include <DHT.h>  #include <WiFiManager.h>  // ====== Pin Configuration ======  #define DHTPIN 4 // Pin untuk sensor DHT22  #define DHTTYPE DHT22 // Tipe sensor  #define RELAY\_PIN 5 // Pin relay mist maker  #define SDA\_PIN 21 // SDA untuk I2C  #define SCL\_PIN 22 // SCL untuk I2C  // ====== Virtual Pin Configuration untuk Blynk ======  #define BLYNK\_MODE V0  #define BLYNK\_MANUAL\_SWITCH V1  #define BLYNK\_TEMP V2  #define BLYNK\_HUM V3  #define BLYNK\_STATUS V4  #define BLYNK\_RESET\_WIFI V5  // ====== Global Variables ======  DHT dht(DHTPIN, DHTTYPE); // Inisialisasi sensor DHT  LiquidCrystal\_I2C lcd(0x27, 16, 2); // Inisialisasi LCD 16x2 dengan alamat 0x27  bool isManualMode = false; // Status mode manual  bool manualControl = false; // Status kontrol manual  bool mistOn = false; // Status mist maker  unsigned long lastUpdate = 0; // Waktu update terakhir  unsigned long manualStartTime = 0; // Waktu mulai mode manual  const unsigned long MANUAL\_TIMEOUT = 30 \* 60 \* 1000; // Timeout 30 menit  // ====== Fungsi untuk update status mist ke Blynk ======  void updateMistStatus() {  String status = mistOn ? "ON" : "OFF";  Blynk.virtualWrite(BLYNK\_STATUS, status);  }  // ====== Fungsi update status mode ke Blynk ======  void updateBlynkModeStatus() {  Blynk.virtualWrite(BLYNK\_MODE, isManualMode ? 1 : 0);  }  // ====== Fungsi update tampilan LCD ======  void updateLCD(float t, float h) {  lcd.clear();  lcd.setCursor(0, 0);  lcd.print("T:" + String(t, 1) + "C");  lcd.setCursor(9, 0);  lcd.print("H:" + String(h, 0) + "%");  lcd.setCursor(0, 1);  lcd.print("M:");  lcd.print(isManualMode ? "Manual" : "Auto ");  lcd.setCursor(9, 1);  lcd.print("S:");  lcd.print(mistOn ? "ON " : "OFF");  updateMistStatus();  }  // ====== Fungsi kontrol mist maker ======  void updateMistControl(float h) {  if (isManualMode) {  if (manualControl) {  if (!mistOn) {  digitalWrite(RELAY\_PIN, HIGH);  mistOn = true;  manualStartTime = millis();  } else if (millis() - manualStartTime > MANUAL\_TIMEOUT) {  digitalWrite(RELAY\_PIN, LOW);  mistOn = false;  manualControl = false;  Blynk.virtualWrite(BLYNK\_MANUAL\_SWITCH, 0);  }  } else if (mistOn) {  digitalWrite(RELAY\_PIN, LOW);  mistOn = false;  }  } else {  if (h < 60.0 && !mistOn) {  digitalWrite(RELAY\_PIN, HIGH);  mistOn = true;  Serial.println("Mist ON (Auto - Hum < 60%)");  } else if (h >= 70.0 && mistOn) {  digitalWrite(RELAY\_PIN, LOW);  mistOn = false;  Serial.println("Mist OFF (Auto - Hum >= 70%)");  }  }  updateMistStatus();  }  // ====== Fungsi callback ketika Blynk terhubung ======  BLYNK\_CONNECTED() {  Blynk.syncVirtual(BLYNK\_MODE);  Blynk.syncVirtual(BLYNK\_MANUAL\_SWITCH);  Blynk.syncVirtual(BLYNK\_RESET\_WIFI);  updateBlynkModeStatus();  }  // ====== Fungsi untuk menangani perubahan mode manual/auto ======  BLYNK\_WRITE(BLYNK\_MODE) {  isManualMode = param.asInt();  Serial.println(isManualMode ? "Manual Mode" : "Auto Mode");  if (!isManualMode) {  manualControl = false;  Blynk.virtualWrite(BLYNK\_MANUAL\_SWITCH, 0);  }  updateBlynkModeStatus();  }  // ====== Fungsi untuk menangani kontrol manual dari aplikasi ======  BLYNK\_WRITE(BLYNK\_MANUAL\_SWITCH) {  manualControl = param.asInt();  Serial.print("Manual switch changed to: ");  Serial.println(manualControl);  manualStartTime = millis();  }  // ====== Fungsi untuk reset pengaturan WiFi ======  BLYNK\_WRITE(BLYNK\_RESET\_WIFI) {  if (param.asInt() == 1) {  Serial.println("Resetting WiFi settings...");  lcd.clear();  lcd.setCursor(0, 0);  lcd.print("Reset WiFi...");  delay(1000);  WiFiManager wm;  wm.resetSettings();  ESP.restart();  }  }  // ====== Setup awal perangkat ======  void setup() {  Serial.begin(115200);  Wire.begin(SDA\_PIN, SCL\_PIN);  lcd.init();  lcd.backlight();  dht.begin();  pinMode(RELAY\_PIN, OUTPUT);  digitalWrite(RELAY\_PIN, LOW);  lcd.setCursor(0, 0);  lcd.print("WiFi Config...");  WiFiManager wm;  bool res = wm.autoConnect("Humidifier-Setup", "12345678");  if (!res) {  Serial.println("Failed to connect & no config saved");  lcd.clear();  lcd.setCursor(0, 0);  lcd.print("WiFi Failed :(");  delay(3000);  ESP.restart();  }  Serial.println("WiFi Connected!");  lcd.clear();  lcd.setCursor(0, 0);  lcd.print("WiFi OK!");  delay(1000);  Blynk.begin(BLYNK\_AUTH\_TOKEN, WiFi.SSID().c\_str(), WiFi.psk().c\_str());  lcd.setCursor(0, 1);  lcd.print("Blynk Connect..");  }  // ====== Loop utama ======  void loop() {  Blynk.run();  unsigned long currentMillis = millis();  if (currentMillis - lastUpdate >= 2000) {  lastUpdate = currentMillis;  float t = dht.readTemperature();  float h = dht.readHumidity();  if (isnan(t) || isnan(h)) {  Serial.println("Sensor Error");  lcd.setCursor(0, 0);  lcd.print("Sensor Error ");  lcd.setCursor(0, 1);  lcd.print("Check DHT22 ");  Blynk.virtualWrite(BLYNK\_TEMP, "Err");  Blynk.virtualWrite(BLYNK\_HUM, "Err");  return;  }  Serial.printf("Temp: %.2f C | Hum: %.2f %%\n", t, h);  updateLCD(t, h);  Blynk.virtualWrite(BLYNK\_TEMP, t);  Blynk.virtualWrite(BLYNK\_HUM, h);  updateMistControl(h);  }  } |

Penjelasan :

1. Inisialisasi dan Setup Hardware
2. Program dimulai dari fungsi setup().
3. Serial monitor dibuka dengan Serial.begin(115200) untuk debugging.
4. I2C LCD diinisialisasi pada pin SDA=21, SCL=22 (Wire.begin(SDA\_PIN, SCL\_PIN)).
5. LCD dinyalakan (lcd.init() dan lcd.backlight()).
6. Sensor DHT22 diaktifkan (dht.begin()).
7. Pin relay diset sebagai output dan dimatikan dahulu (digitalWrite(RELAY\_PIN, LOW)).
8. Konfigurasi dan Koneksi WiFi Otomatis
9. Program menampilkan pesan "WiFi Config..." di LCD.
10. Menggunakan WiFiManager untuk mencoba menyambung ke WiFi terakhir. Jika belum ada:
11. ESP32 membuat hotspot bernama "Humidifier-Setup", password: 12345678.
12. Pengguna dapat menyambungkan HP lalu memasukkan SSID dan password WiFi.
13. Jika gagal menyambung, ESP restart ulang (ESP.restart()).
14. Jika berhasil, tampilkan "WiFi OK!" dan lanjut ke koneksi Blynk.
15. Koneksi ke Server Blynk
16. Gunakan token BLYNK\_AUTH\_TOKEN dan WiFi SSID/Password untuk koneksi ke Blynk Cloud (Blynk.begin()).
17. Tampilkan "Blynk Connect.." di LCD.
18. Loop Utama

Program memasuki loop() yang terus berjalan, setiap 2 detik:

1. Baca suhu dan kelembapan dari sensor DHT22:

float t = dht.readTemperature();

float h = dht.readHumidity();

1. Cek apakah pembacaan valid (tidak NaN).
2. Tampilkan data suhu dan kelembapan ke LCD menggunakan updateLCD(t, h).
3. Kirim nilai suhu dan kelembapan ke aplikasi Blynk (Blynk.virtualWrite()).
4. Jalankan logika mist maker (otomatis/manual) dengan memanggil updateMistControl(t) (atau h jika ingin berdasarkan kelembapan).
5. Kontrol Otomatis/Mist Maker

Fungsi updateMistControl() menentukan apakah relay dinyalakan atau dimatikan:

1. Mode Manual:

Jika mode manual aktif (isManualMode == true), kontrol mist maker ditentukan oleh tombol manual (manualControl).

1. Mode Otomatis (berdasarkan suhu):

Jika suhu > 27°C dan mist belum aktif → nyalakan mist maker.

Jika suhu ≤ 25°C dan mist aktif → matikan mist maker.

1. Fungsi updateMistStatus() digunakan untuk memperbarui status mist ke Blynk.
2. Fungsi Sinkronisasi Saat Terkoneksi Blynk
3. Saat koneksi ke Blynk berhasil (BLYNK\_CONNECTED()), sinkronkan nilai tombol/toggle mode dari aplikasi:

Blynk.syncVirtual(BLYNK\_MODE);

Blynk.syncVirtual(BLYNK\_MANUAL\_SWITCH);

Blynk.syncVirtual(BLYNK\_RESET\_WIFI);

1. Fungsi Virtual Pin Blynk
2. BLYNK\_WRITE(BLYNK\_MODE): Menerima perubahan mode dari aplikasi (manual/otomatis).
3. BLYNK\_WRITE(BLYNK\_MANUAL\_SWITCH): Menerima perintah manual ON/OFF dari pengguna.
4. BLYNK\_WRITE(BLYNK\_RESET\_WIFI): Jika ditekan, sistem akan:

Menampilkan "Reset WiFi..." di LCD.

Menghapus konfigurasi WiFi.

Restart ESP32 agar masuk mode konfigurasi ulang.

### Blynk Setup

#### Informasi Template

Tabel 3. 6 : Informasi Template

|  |  |
| --- | --- |
| **Field** | **Values** |
| Template ID | TMPL6PJEDLLXx |
| Template Name | Humidifier |
| Auth Token | WizPCfkBq07EVvxT-\_NP3X12bZr0dTNC |

#### Datastreams

Tabel 3. 7 : Datastreams

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Virtual Pin** | **Min** | **Max** | **Unit** |
| Mode | Integer | V0 | 0 | 1 |  |
| Manual Switch | Integer | V1 | 0 | 1 |  |
| Temperature | Double | V2 | 0 | 100 | C |
| Humidity | Double | V3 | 0 | 100 | % |
| Mist Status | String | V4 |  |  |  |
| Reset Wifi | Integer | V5 | 0 | 1 |  |

#### Widget

Tabel 3. 8 : Widget

|  |  |  |  |
| --- | --- | --- | --- |
| **Widget** | **Datastreams** | **Type** | **Notes** |
| Mode Switch | V0 | Switch | Otomatis (0) / Manual (1) |
| Manual Switch | V1 | Switch | Kontrol ON/OFF saat manual |
| Temperature | V2 | Gauge / Display | Menampilkan suhu |
| Humidity | V3 | Gauge / Display | Menampilkan kelembapan |
| Mist Status | V4 | Label | Menampilkan ON / OFF |
| Reset Wifi | V5 | Button | Value: 1, Mode: Push |

## 

**PENUTUP**

### Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada proyek IoT Air Humidifier berbasis ESP32-C6, dapat diambil beberapa kesimpulan sebagai berikut :

1. Perancangan dan pembangunan sistem humidifier berbasis IoT berhasil dilakukan dengan menggunakan mikrokontroler ESP32-C6, yang mampu mengendalikan mist maker secara otomatis berdasarkan kelembapan dan secara manual melalui aplikasi Blynk. Sistem mendukung dual-mode operation yang memungkinkan fleksibilitas dalam pengendalian perangkat.
2. Sistem mampu memantau dan menampilkan data suhu dan kelembaban secara real-time, menggunakan sensor DHT22. Data tersebut ditampilkan melalui layar LCD I2C 1602 dan secara sinkron dikirim ke aplikasi Blynk, dengan pembaruan setiap 2 detik. Hal ini memungkinkan pengguna untuk selalu mendapatkan informasi terbaru mengenai kondisi udara di lingkungan sekitar.
3. Integrasi antara sensor kelembaban (DHT22) dan mist maker telah berhasil diimplementasikan secara efisien. Logika kontrol disusun berdasarkan ambang suhu tertentu, serta diintegrasikan ke dalam platform IoT (Blynk) menggunakan koneksi WiFi. Proses ini memungkinkan sistem berjalan otomatis tanpa intervensi manual, tetapi tetap menyediakan opsi override.
4. Kontrol jarak jauh dan umpan balik kondisi kelembaban berhasil diimplementasikan melalui perangkat mobile menggunakan aplikasi Blynk. Pengguna dapat mengakses dashboard untuk melihat suhu, kelembaban, status mist maker, serta mengganti mode kontrol dan mereset pengaturan WiFi secara langsung dari smartphone.

Dengan demikian, seluruh rumusan masalah yang telah ditetapkan pada awal proyek ini telah terjawab melalui implementasi sistem yang telah dibangun dan diuji secara menyeluruh.

### Saran

Berdasarkan hasil implementasi dan pembahasan yang telah dilakukan dalam proyek ini, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan lebih lanjut di masa mendatang :

1. Peningkatan Akurasi dan Sensor Tambahan

Disarankan untuk menggunakan sensor suhu dan kelembaban dengan akurasi lebih tinggi seperti SHT31 atau BME280 agar sistem dapat bekerja lebih presisi, terutama jika digunakan untuk kebutuhan medis atau lingkungan sensitif.

1. Penambahan Fitur Monitoring Berbasis Cloud

Sistem dapat dikembangkan agar terhubung dengan layanan cloud seperti Firebase, ThingsBoard, atau MQTT Broker publik untuk menyimpan histori data dan melakukan analisis tren suhu dan kelembaban dari waktu ke waktu.

1. Integrasi Notifikasi Real-Time

Penambahan notifikasi (seperti melalui Telegram, email, atau push notification) akan memberikan informasi langsung kepada pengguna saat kondisi tertentu terjadi, seperti kelembaban rendah atau kesalahan perangkat.

1. Optimasi Konsumsi Daya

Untuk penggunaan jangka panjang terutama jika menggunakan sumber daya baterai, sistem sebaiknya dikembangkan dengan mode hemat daya (deep sleep) dan efisiensi pembacaan sensor yang lebih baik.

1. Tampilan Antarmuka Pengguna Lebih Interaktif

Meskipun aplikasi Blynk sudah cukup fungsional, sistem dapat dikembangkan lebih lanjut dengan membuat dashboard web berbasis Node-RED atau ReactJS, sehingga dapat diakses dari perangkat manapun tanpa bergantung pada satu aplikasi saja.

1. Penerapan Case atau Enclosure Khusus

Untuk keamanan dan estetika, disarankan menempatkan semua komponen ke dalam casing atau enclosure yang tahan terhadap debu dan kelembaban agar lebih aman dan profesional.

# LAMPIRAN



Repository dapat diakses pada : https://github.com/fajarejaya/project-3

# DAFTAR PUSTAKA

Adafruit Learning System. (n.d.). Breadboarding basics. Dikutip dari https://learn.adafruit.com/breadboards-for-beginners

Adafruit Learning System. (n.d.). Character LCDs. Dikutip dari https://learn.adafruit.com/character-lcds

Adafruit Learning System. (n.d.). DHT sensor tutorial. Dikutip dari https://learn.adafruit.com/dht

Adafruit Learning System. (n.d.). Humidifiers explained. Dikutip dari https://learn.adafruit.com/usb-powered-humidifier

Adafruit. (n.d.). DHT sensor library documentation. Dikutip dari https://github.com/adafruit/DHT-sensor-library

Advanced Monolithic Systems. (n.d.). Datasheet AMS1117 Voltage Regulator. Dikutip dari https://www.advanced-monolithic.com/pdf/ds1117.pdf

All About Circuits. (n.d.). Introduction to breadboards. Dikutip dari https://www.allaboutcircuits.com/tools/breadboard-view/

Aosong Electronics. (2020). DHT22 (AM2302) Sensor Datasheet. Dikutip dari https://www.aosong.com/en/products/details/22.html

Arduino Reference. (n.d.). DHT sensor library. Dikutip dari https://github.com/adafruit/DHT-sensor-library

Arduino Reference. (n.d.). LiquidCrystal\_I2C library. Dikutip dari https://github.com/johnrickman/LiquidCrystal\_I2C

Cisco. (n.d.). IoT design considerations and best practices. Dikutip dari https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html

Cisco. (n.d.). IoT explained. Dikutip dari https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html

ElectroPeak. (n.d.). Power supply module for MB102 breadboard. Dikutip dari https://electropeak.com/learn/power-supply-module-for-mb102-breadboard/

Espressif Systems. (n.d.). ESP32 Datasheet. Dikutip dari https://www.espressif.com/sites/default/files/documentation/esp32\_datasheet\_en.pdf

Espressif Systems. (n.d.). ESP-IDF Documentation. Dikutip dari https://docs.espressif.com/projects/esp-idf/en/latest/esp32/index.html

Espressif Systems. (n.d.). ESP32 DevKitC Documentation. Dikutip dari https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32c6/esp32-c6-devkitc-1/index.html

GitHub. (n.d.). Arduino core for the ESP32. Dikutip dari https://github.com/espressif/arduino-esp32

GitHub. (n.d.). Blynk Library (Classic and IoT). Dikutip dari https://github.com/blynkkk/blynk-library

GitHub. (n.d.). LiquidCrystal\_I2C\_ESP32 – iakop. Dikutip dari https://github.com/iakop/LiquidCrystal\_I2C\_ESP32

GitHub. (n.d.). LiquidCrystal\_I2C – Marcoschwartz. Dikutip dari https://github.com/marcoschwartz/LiquidCrystal\_I2C

GitHub. (n.d.). PubSubClient – Nick O’Leary. Dikutip dari https://github.com/knolleary/pubsubclient

GitHub. (n.d.). WiFiManager – by tzapu. Dikutip dari https://github.com/tzapu/WiFiManager

IBM. (n.d.). What is the Internet of Things (IoT)? Dikutip dari https://www.ibm.com/topics/internet-of-things

Instructables. (n.d.). DIY Ultrasonic Mist Maker Projects. Dikutip dari https://www.instructables.com/id/Ultrasonic-Mist-Maker-Projects

Instructables. (n.d.). How to use MB102 power supply module. Dikutip dari https://www.instructables.com/How-to-Use-MB102-Breadboard-Power-Supply-Module/

Make: Magazine. (n.d.). Breadboarding tips and tricks. Dikutip dari https://makezine.com/projects/skill-builder-breadboarding/

Microsoft Azure. (n.d.). Introduction to the Internet of Things (IoT). Dikutip dari https://learn.microsoft.com/en-us/azure/iot-fundamentals/iot-introduction

Microsoft Azure. (n.d.). IoT architecture and components. Dikutip dari https://learn.microsoft.com/en-us/azure/iot-fundamentals/iot-introduction

Oracle. (n.d.). What is IoT? Dikutip dari https://www.oracle.com/internet-of-things/what-is-iot/

PlatformIO. (n.d.). platformio.ini configuration file. Dikutip dari https://docs.platformio.org/en/latest/projectconf/index.html

Random Nerd Tutorials. (n.d.). Controlling mist maker with relay and ESP32. Dikutip dari https://randomnerdtutorials.com

Random Nerd Tutorials. (n.d.). DHT22 with Arduino and ESP32. Dikutip dari https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor

Random Nerd Tutorials. (n.d.). ESP32 tutorials and projects. Dikutip dari https://randomnerdtutorials.com

Random Nerd Tutorials. (n.d.). Guide for I2C LCD with Arduino and ESP32. Dikutip dari https://randomnerdtutorials.com/esp32-i2c-lcd-1602

ResearchGate. (2020). Design and implementation of IoT-based smart systems. Dikutip dari https://www.researchgate.net/publication/343964357\_IoT\_Smart\_Systems

SparkFun. (n.d.). LCD 16x2 character display hookup guide. Dikutip dari https://learn.sparkfun.com/tutorials/using-an-lcd

SparkFun. (n.d.). What is a breadboard? Dikutip dari https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard

SparkFun Forums. (n.d.). MB102 breadboard power supply overview. Dikutip dari https://forum.sparkfun.com/viewtopic.php?t=42630

Taidacent Electronics. (n.d.). Ultrasonic fogger working principle. Dikutip dari <https://www.taidacent.com/products/ultrasonic-mist-maker>

ResearchGate. (2019). Smart Home Automation using IoT and ESP32. Dikutip dari https://www.researchgate.net/publication/335702021\_Smart\_Home\_Automation\_using\_IoT\_and\_ESP32

Datasheet4U. (n.d.). SRD-05VDC-SL-C Datasheet. Dikutip dari https://www.datasheet4u.com/datasheet-pdf/Songle/SRD-05VDC-SL-C/pdf.php?id=1111842

Electronics Hub. (2021). How to use a Relay with Arduino. Dikutip dari https://www.electronicshub.org/interfacing-relay-with-arduino/