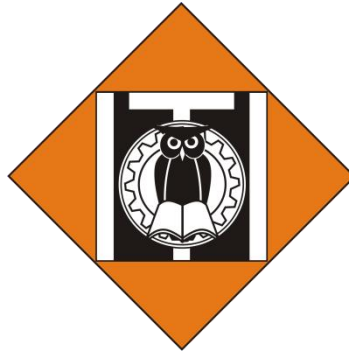


**LAPORAN AKHIR PENELITIAN  
DANA MANDIRI**



**PEMANFAATAN SISTEM IOT BERBASIS ESP32 UNTUK  
*REMOTE CONTROL* PADA INKUBATOR TEMPE**

**Ketua : Novy Hapsari, ST. MSc. (0312117803)**  
**Anggota : 1. Ir. Tita Aisyah, MT, IPM ( 0307046301)**  
**2. Saharudin, ST, M.EngSc, IPM (0310107702)**  
**Anggota Mahasiswa : 1. Melin Carmelinda (1111900033)**  
**2. M Yusrillah Surya (1111900013)**  
**3. M Haris (1111900036)**

**Dibiayai oleh:  
Dana Mandiri**

**Nomor surat tugas:  
001/ST-PLT/PRPM-PP/ITI/V/2023**

**PROGRAM STUDI TEKNIK ELEKTRO  
INSTITUT TEKNOLOGI INDONESIA**

## HALAMAN PENGESAHAN

Judul Penelitian : Pemanfaatan Sistem IoT Berbasis ESP32 Untuk *Remote Control* Pada Inkubator Tempe

Jenis Penelitian : Material Manufaktur

Bidang Fokus Penelitian : Engineering

Tujuan Sosial Ekonomi : Otomatisasi Industri Tempe

TKT (Tingkat Kesiapterapan Teknologi) : 6

Peneliti

a. Nama Lengkap : Ir. Novy Hapsari, ST. MSc.

b. NIDN : 0312117803

c. Jabatan Fungsional : Lektor

d. Program Studi : Teknik Elektro

e. Nomor HP : 0858-80808837

f. Alamat Surel (*e-mail*) : novy.des2014@gmail.com

Anggota Peneliti 1

a. Nama Lengkap : Ir. Tita Aisyah, MT. IPM.

b. NIDN : 0307046301

c. Institusi : Institut Teknologi Indonesia

Anggota Peneliti 2

a. Nama Lengkap : Ir. Saharudin, M.Eng.Sc. IPM.

b. NIDN : 0310107702

c. Institusi : Institut Teknologi Indonesia

Anggota Mahasiswa

1. a. Nama Lengkap : Melin Carmelinda

b. NIM : 1111900033

c. Jurusan : Teknik ELEktro

2. a. Nama Lengkap : M. Yusrillah Surya

b. NIM : 1111900013

c. Jurusan : Teknik ELEktro

3. a. Nama Lengkap : Mohamad Haris

b. NIM : 1111900036

c. Jurusan : Teknik ELEktro


Institusi Sumber Dana : Mandiri

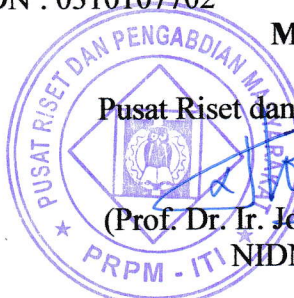
Biaya Penelitian : Rp. 12.000.000,-

Mitra Penelitian : Pabrik Tempe H. Mochtar

Tangerang Selatan, 02 Agustus 2023

**Mengetahui,**  
**Ketua Program Studi**  
(Ir. Saharudin, ST. M.Eng.Sc. IPM)  
NIDN : 0310107702

**Ketua Tim**  
  
(Ir. Novy Hapsari, ST. MSc.)  
NIDN : 0312117803

**Menyetujui,**  
**Kepala**  
Pusat Riset dan Pengabdian Masyarakat  
(Prof. Dr. Ir. Joelianingsih, M.T., IPM)  
NIDN : 031007640



# INSTITUT TEKNOLOGI INDONESIA

Jl. Raya Puspiptek, Tangerang Selatan - 15314  
(021) 7562757

[www.iti.ac.id](http://www.iti.ac.id) [institutteknologiindonesia](https://www.instagram.com/institutteknologiindonesia) [@kampusITI](https://www.facebook.com/kampusITI) Institut Teknologi Indonesia

## SURAT TUGAS

No. : 001/ST-PLT/PRPM-PP/ITI/V/2023

Pertimbangan : Bahwa dalam rangka melaksanakan kegiatan Penelitian bagi Dosen Program Studi Teknik Elektro Institut Teknologi Indonesia, perlu dikeluarkan surat tugas.

Dasar : 1. Pembebanan Tugas Dosen Program Studi Teknik Elektro;  
2. Surat Permohonan Tanggal 17 Mei 2023;  
3. Kepentingan Institut Teknologi Indonesia.

### DITUGASKAN

Kepada : Dosen Program Studi Teknik Elektro – ITI (Terlampir)

Untuk : 1. Melaksanakan kegiatan Penelitian pada Semester Genap Tahun Akademik 2022/2023;  
2. Melaporkan hasil tugas kepada Kepala PRPM - ITI;  
3. Dilaksanakan dengan penuh rasa tanggung jawab.

Tangerang Selatan, 25 Mei 2023  
Pusat Riset dan Pengabdian  
Masyarakat



Kepala,

Prof. Dr. Ir. Joelianingsih, M.T., IPM

Tembusan Yth.

1. Wakil Rektor Bid Akademik, Penelitian dan Kemahasiswaan
2. Ka. Biro SDMO
3. Ka. Prodi Teknik Elektro
4. Arsip

Lampiran Surat Tugas  
No. 001/ST-PLT/PRPM-PP/ITI/V/2023  
Tanggal 25 Mei 2023

DAFTAR PENELITIAN DOSEN PROGRAM STUDI TEKNIK ELEKTRO SEMESTER GENAP THN AKADADEMIK: 2022/2023

NO	TOPIK PENELITIAN	BIDANG	SUSUNAN TIM	SUMBER DANA	JUMLAH DANA (Rp)	KETERLIBATAN PRODI/INSTITUSI LAIN	KETERLIBATAN MAHASISWA
1	Rancang Bangun Inkubator Tempe Otomatis Skala Industri Rumah Tangga dalam Rangka Peningkatan Tingkat Kesiapteraan Teknologi (TKT)	Material Manufaktur	1.Ir.Novy Hapsari, S.T., M.Sc (ketua) 2.Ir. Tita Aisyah, M.T., IPM 3.Ir.Saharudin, S.T., M.Eng.Sc., IPM	Mandiri	12.000.000	Pabrik tempe haji Mochtar	1. M Yusrilah Surya 2. Melin Carmelinda 3. M Harris
2	Pengujian PLTB Menggunakan Turbin Angin Savonius dan Tipe H	Energi	1. Ir Edwin Kamal, ST.M.Eng.Sc.IPM (ketua) 2. Dra. Ir.Sri Yatmani, M.Si., IPM 3. Ir. Adi Setiawan, S.T., M.Eng.Sc., IPM	Mandiri	15.000.000	Tidak ada	1. Izha Mahendro 2. Bimo Saputro
3	Rancang Bangun Robot Pembaca Gerakan Tungkai Bawah	Material Manufaktur	1.Dr. Ir Tris Dewi Indraswati, S.T., M.T (ketua) 2. Ir. Ulfah Khairiyah Luthfiyani, S.T., M.Eng 3. Dra. Ir. Ratnawati, M.Si., IPM	Mandiri	13.500.000	Tidak ada	1. Uko Meyamin 2. Abdulllah Restu 3. Wahyudi
4	Effisiensi Daya Listrik Sel Surya Menggunakan Konverter DC-DC pada Beban Listrik Terhadap Perubahan Cahaya Matahari	Energi	Ir. Parlindungan P. Marpaung, MT	Mandiri	10.000.000	Tidak ada	1. Lutfiansyah Aljar 2. Fadillah
5	Rancang Bangun Sistem Monitoring Suhu Ruang Praktikum Dasar Sistem Kendali dan Kondisi Sekitar	Material Manufaktur	1. Ir.Saharudin, S.T., M.Eng.Sc., IPM Dra.Ir. Ratnawati, M.Si., IPM	Mandiri	12.500.000	Tidak ada	Joan Agrio Syahputra
6	Pemodelan Maintenance Transformator Distribusi Menggunakan Machine Learning	Energi	1. Ir.Ulfah Khairiyah Luthfiyani, S.T., M.Eng 2.Ir.Saharudin, S.T., M.Eng.Sc., IPM	Mandiri	10.000.000	Tidak ada	Ahmad Farhan

Pusat Riset dan Pengabdian Masyarakat  
Institut Teknologi Indonesia  
Kepala



Prof. Dr. Ir. Joelianningsih, M.T., IPM

## **PRAKATA**

Puji Puji Syukur kehadiran Tuhan YME yang telah melimpahkan rahmat, nikmat, dan petunjuk Nya sehingga laporan akhir penelitian periode Semester Genap 2023 dapat diselesaikan. Laporan penelitian ini dibuat sebagai hasil penelitian yang telah kami lakukan pada industry pengrajin tempe dengan judul Pemanfaatan Sistem IoT Berbasis ESP32 Untuk Remote Control pada Inkubator Tempe. Laporan penelitian ini telah kami buat sebaik mungkin, namun kami sadar bahwa masih banyak kekurangan dan perbaikan. Oleh karena itu, masukan dan saran dalam rangka perbaikan serta pengembangan atas penelitian kami sangat kami harapkan.

Diakhir kata, semoga laporan kami dapat memberikan wawasan tambahan dan sumber referensi bagi Bapak/Ibu yang membutuhkan.

Tangerang Selatan, Agustus 2023  
Ketua Peneliti,

Novy Hapsari, ST. MSc.

## ABSTRAK

Perkembangan teknologi yang sangat pesat di era globalisasi saat ini telah memberikan banyak manfaat dalam kemajuan diberbagai aspek. Di Indonesia, tempe merupakan salah satu makanan tradisional yang digemari oleh masyarakat. Oleh karena itu, industri pembuat tempe merupakan salah satu industri rumah tangga yang cukup potensial. Menurut data yang diperoleh dari Badan Standarsisasi Nasional (BSN) tahun 2012, produsen tempe sekitar 81000 yang memproduksi 2,4 juta ton tempe per tahun di Indonesia. Pengembangan dari inkubasi tempe yang sudah ada yaitu penambahan beberapa sistem monitoring otomatis dimana suhu dan kelembapan dapat di monitoring dan di akses dimana saja menggunakan aplikasi android. Penerapan *Internet of Things* pada proses fermentasi tempe dapat mempermudah dalam memonitoring, mengkontrol suhu dan kelembapan, serta mengatur waktu pada inkubator secara *realtime* dan *mobile* ketika pengrajin tidak berada disekitar inkubator dapat menggunakan aplikasi android. Tujuan dari penelitian ini adalah merancang sebuah alat yang dapat mengkontrol kondisi ruangan fermentasi tempe berdasarkan kelembapan dan suhu menggunakan sensor yang dapat memonitoring secara *realtime* dan dikontrol melalui *smartphone android*. Hasil pengujian memperlihatkan alat yang dirancang bangun bekerja dengan baik, dimana ESP32 yang digunakan sebagai mikrokontroler dapat berfungsi dan bekerja dengan baik dengan memperoleh tegangan pada kondisi *high* sebesar 2,69 volt; Sensor DHT dapat mengirimkan data berupa *temperature* dan *humidity* dengan rata-rata pembacaan sebesar 27,53 °C dengan nilai rata-rata simpangan sebesar 0,47 °C; Platform Blynk dapat terkoneksi dengan Blynk dengan terkoneksi WiFi dan dapat memonitoring dengan menampilkan pembacaan data dalam bentuk *gauge*, digit angka digital, dan grafik, serta Blynk dapat mengatur suhu batas atas dan suhu batas bawah, serta dapat mengkontrol mulainya *countdown* dengan rata-rata selisih waktu pembacaan sebesar 0.8654 sampai dengan 2 detik, yang sangat dipengaruhi oleh jaringan internet.

**Kata Kunci:** DHT22, fermentasi tempe, IoT, platform Blynk, remote

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	<b>Error! Bookmark not defined.</b>
PRAKATA.....	1
ABSTRAK.....	2
DAFTAR ISI.....	3
BAB I PENDAHULUAN.....	5
1.1 Latar Belakang.....	5
1.2 Rumusan Masalah.....	6
1.3 Batasan Masalah.....	6
1.4 Tujuan Penelitian.....	7
1.5 Metodologi Penelitian.....	7
BAB II TINJAUAN PUSTAKA.....	8
2.1 Tempe.....	8
2.2 Inkubator Tempe.....	10
2.3 <i>Remote control</i> .....	11
2.3.1 <i>Komponen Remote control</i> .....	12
2.3.2 <i>Prinsip Cara kerja remote control</i> .....	12
2.4 Komunikasi Data.....	13
2.5 ESP32.....	14
2.5 <i>Internet of Things (IoT)</i> .....	16
2.6 Perangkat Lunak Pendukung.....	18
2.6.1 <i>Arduino IDE</i> .....	18
2.7.2 <i>Blynk</i> .....	19
BAB III RANCANG BANGUN DAN INTEGRASI ALAT.....	22
3.1 Perancangan Fungsional.....	22
3.1.1 <i>Diagram Blok</i> .....	22
3.1.2 <i>Arsitektur Integrasi Sistem</i> .....	23
3.1.3 <i>Protocol Stack</i> .....	23
3.2 Perancangan Struktural.....	24
3.2.1 <i>Perancangan Elektronika</i> .....	25

3.2.2 Alur Kerja Sistem .....	25
3.3 Integrasi Bagian Hardware .....	27
3.4 Perancangan Integrasi Perangkat Lunak atau Software .....	27
3.4.1 Flowchart System .....	28
3.4.2 Hubungan Komunikasi Antara Sistem dengan Platform IoT Blynk .....	30
3.5.3 Perancangan Komunikasi antar Aplikasi (API) .....	30
3.5 Manufaktur Aplikasi Blynk .....	31
<b>BAB IV HASIL PENGUJIAN DAN ANALISA DATA .....</b>	<b>35</b>
4.1 Pengujian Mikrokontroler ESP32 .....	35
4.1.1 Tujuan Pengujian .....	35
4.1.2 Langkah pengujian .....	35
4.1.3 Hasil Pengujian dan Analisa mikrokontroler ESP32 .....	36
4.2 Pengujian Pembacaan Sensor DHT22 .....	38
4.2.1 Tujuan Pengujian .....	38
4.2.2 Langkah pengujian .....	38
4.2.3 Hasil Pengujian dan Analisa Pembacaan DHT22 .....	39
4.3 Pengujian Sistem Software Komunikasi ke Blynk .....	42
4.3.1 Tujuan Pengujian .....	42
4.3.2 Langkah Pengujian .....	42
4.3.3 Hasil Pengujian dan Analisa Koneksi Jaringan Sistem dengan Blynk .....	44
4.3.4 Hasil Pengujian dan Analisa pembacaan nilai suhu dan kelembapan pada serial monitor dan platform Blynk .....	45
4.3.5 Hasil Pengujian dan Analisa pembacaan nilai suhu pada platform Blynk dan display .....	47
4.3.6 Hasil Pengujian dan Analisa waktu respon controlling mulai hitung mundur .....	48
4.4 Pengujian Sistem Secara Keseluruhan .....	49
4.4.1 Tujuan Pengujian .....	49
4.4.2 Langkah Pengujian .....	49
4.4.3 Hasil Pengujian dan Analisa Sistem Secara Keseluruhan .....	49
<b>BAB V PENUTUP .....</b>	<b>52</b>
<b>DAFTAR PUSTAKA .....</b>	<b>53</b>



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi yang sangat pesat di era globalisasi saat ini telah memberikan banyak manfaat dalam kemajuan diberbagai aspek. Di Indonesia sendiri merupakan salah satu negara yang berkembang khususnya dalam perkembangan dan kemajuan teknologi di berbagai bidang baik industri besar, pendidikan maupun industri rumah tangga.

Di Indonesia, tempe merupakan salah satu makanan tradisional yang digemari oleh masyarakat. Tempe juga merupakan sumber pangan yang digunakan sebagai lauk pauk yang mempunyai kandungan gizi yang sangat baik. Tidak hanya kalangan masyarakat kelas bawah, namun masyarakat menengah keatas pun juga mengonsumsinya. Oleh karena itu, industri pembuat tempe merupakan salah satu industri rumah tangga yang banyak terdapat di indonesia yang cukup potensial. Menurut data yang diperoleh dari Badan Standarsisasi Nasional (BSN) tahun 2012, produsen tempe sekitar 81000 yang memproduksi 2,4 juta ton tempe per tahun di indonesia.

Tempe dibuat dari bahan utama biji kedelai serta bahan lainnya yang menunjang proses fermentasi oleh kapang *Rhizopus sp* (“ragi tempe”). Pada umumnya, para produsen pembuat tempe masih menggunakan cara tradisional atau manual dimana cara ini masih mengandalkan pada situasi cuaca. Dalam proses fermentasi dipengaruhi oleh suhu dan kelembapan ruangan. Suhu dan kelembapan ruangan sangat berhubungan satu sama lain, karena apabila kelembapan udara berubah, maka suhu juga akan berubah. Selama musim hujan suhu ruangan rendah dan kelembapan tinggi. Selain itu proses fermentasi secara tradisional membutuhkan waktu yang lama sekitar 24-40 jam, dengan suhu sekitar 20 °C - 25 °C . (Putra, 2022)

Pengembangan dari inkubasi tempe yang sudah ada yaitu penambahan beberapa sistem monitoring otomatis dimana suhu dan kelembapan dapat di monitoring dan di akses dimana saja menggunakan aplikasi android. Aplikasi android digunakan karena menggunakan konsep internet of things pengrajin akan mendapat kemudahan dalam pembuatan tempe yaitu hanya mamantau suhu dan kelembapan, on/off fermentasi pada inkubator melalui aplikasi yang ada pada

android walaupun orang tersebut berada di kejauhan dari tempat produksi pembuatan tempe dengan konsep *internet of things*. *Internet of things* (IoT) adalah sebuah konsep dimana objek dalam bentuk fisik dapat dihubungkan dan mentransfer data melalui jaringan internet oleh sensor. Konsep IoT menggambarkan setiap objek yang dapat terhubung ke internet dan dapat dikendalikan dari jarak jauh tanpa harus berada di tempat. Dengan dukungan IoT, pekerjaan akan menjadi lebih cepat dan mudah. (Safrianti, et al., 2022)

Dari uraian permasalahan tersebut, maka pada penelitian ini akan membuat suatu alat yang dapat memonitoring dan mengontrol suhu dan kelembapan pada proses fermentasi tempe. Alat pendeteksi suhu dan kelembapan menggunakan sensor DHT22 dan dapat di monitoring secara tidak langsung dengan menerapkan *Internet of Things* dengan ESP32 menggunakan aplikasi Blynk agar sistem dapat bekerja secara *realtime* dengan *smartphone android*. Penerapan *Internet of Things* pada proses fermentasi tempe dapat mempermudah dalam memonitoring, mengontrol suhu dan kelembapan, serta mengatur waktu pada inkubator secara *realtime* dan *mobile* ketika pengrajin tidak berada disekitar inkubator dapat menggunakan aplikasi android.

## **1.2 Rumusan Masalah**

Dari penjelasan yang telah dijelaskan dalam latar belakang, maka dapat dirumuskan masalah yang haru diselesaikan dalam penelitian ini sebagai berikut :

1. Bagaimana merancang alat yang dapat mengontrol pada kelembapan suhu untuk fermentasi tempe berbasis Android.
2. Bagaimana mengintegrasikan algoritma ke alat inkubator dengan sistem IoT.
3. Bagaimana memonitoring suhu dan kelembapan secara optimal dari jarak jauh.

## **1.3 Batasan Masalah**

Agar penelitian ini lebih spesifik dan terarah, maka pembahasan masalah dalam penelitian ini memiliki batasan masalah sebagai berikut :

1. Penelitian alat ini hanya menggunakan sistem IoT berbasis Blynk.
2. Alat ini hanya dapat mengontrol suhu dan timer pada dengan menggunakan aplikasi Blynk.
3. Menggunakan ESP32 untuk mengkoneksikan ke *internet of things*.

4. Variabel *inputan* yang digunakan yaitu suhu dan kelembapan yang berasal dari sensor DHT22.

#### **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah merancang sebuah alat yang dapat mengontrol kondisi ruangan fermentasi tempe berdasarkan kelembapan dan suhu menggunakan sensor yang dapat memonitoring secara *realtime* dan dikontrol melalui *smartphone android*.

#### **1.5 Metodologi Penelitian**

Dalam penelitian ini metode yang digunakan adalah sebagai berikut :

1. Studi Pustaka, yaitu dengan mempelajari literatur dan ide yang sudah ada dilakukan studi yang digunakan berupa buku-buku, artikel-artikel baik dari internet maupun jurnal serta data-data penelitian dan paten yang telah ada sejak awal.
2. Studi Observasi, yaitu untuk mendapatkan tolak ukur nilai suhu dan kelembapan yang digunakan dalam proses fermentasi tempe dilakukan studi pada tempat pengrajin tempe konvensional dan membandingkan alat yang sudah ada sebelumnya.
3. Perancangan dan Perakitan Sistem elektronika, yaitu Perancangan dilakukan dalam beberapa tahap. Pertama adalah perancangan diagram blok sistem ditentukan. Diantaranya perancangan sistem yang dibuat meliputi rangkaian sistem pengkabelan untuk menghubungkan antara sensor DHT22 dan komponen lainnya dengan Arduino Mega 2560 yang berkomunikasi dengan Nodemcu Esp32 untuk mengkoneksikan terhadap internet. Berikutnya adalah perancangan software berupa coding yang digunakan untuk mengontrol kinerja komponen-komponen yang digunakan secara keseluruhan.
4. Integrasi Sistem Hardware dan Software, yaitu Bagian yang sudah siap digunakan kemudian dirakit menjadi satu kesatuan sistem yang kemudian diimplementasikan.
5. Pengujian dan Evaluasi Sistem Keseluruhan, yaitu sistem yang telah dirakit selanjutnya diuji, termasuk semua bagian pemrograman yang terkait. Tujuan dari pengujian ini membuktikan bahwa alat dapat bekerja dengan baik dan sesuai dengan yang dikehendaki ketika perancangan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tempe

Tempe adalah salah satu makanan tradisional khas Indonesia. Di tanah air, tempe sudah lama dikenal selama berabad-abad silam. Makanan ini diproduksi dan dikonsumsi secara turun temurun, khususnya di daerah Jawa Tengah dan sekitarnya. Tempe merupakan makanan yang terbuat biji kedelai atau beberapa bahan lain yang diproses melalui fermentasi dari apa yang secara umum dikenal sebagai “ragi tempe”. Lewat proses fermentasi ini, biji kedelai mengalami proses penguraian menjadi senyawa sederhana sehingga mudah dicerna. (PUSIDO, 2012).

Tempe adalah makanan yang dibuat dari fermentasi terhadap biji kedelai atau beberapa bahan lain yang menggunakan beberapa jenis kapang terbuat dari fermentasi terhadap biji kedelai atau beberapa bahan lain yang menggunakan beberapa jenis kapang *rhizopus* seperti *rhizopus oligosporus*, *Rh. oryzae*, *Rh. stolonifer* (kapang roti), atau *Rh. arrhizus*. Sediaan fermentasi ini secara umum dikenal sebagai “ragi tempe”.

Indonesia merupakan negara produsen tempe terbesar di dunia dan menjadi pasar kedelai terbesar di Asia. Sebanyak 50% dari konsumsi kedelai di Indonesia dilakukan dalam nemtul tempe, 40 tahu dan 10% dalam produk lain (seperti tauco, kecap dan lain-lain). Konsumsi tempe rata-rata per orang per tahun di Indonesia diduga sekitar 6,45 kg (Alvina & Hamdani, 2019).



Gambar 2.1 Tempe

Produksi tempe di Indonesia sebagian besar masih dilakukan dengan cara tradisional. Hal ini dikarenakan para pelaku usaha tempe berasal dari kalangan Industri Rumah Tangga yang masih menggunakan peralatan dan proses produksi

yang tidak standar. Dikarenakan proses produksi masih secara tradisional maka dimasa yang akan datang akan sulit bersaing dengan industri yang lebih besar yang telah menerapkan cara berproduksi yang baik dan benar serta menerapkan *system hygiene* dan sanitasi yang benar unruk menghasilkan produk yang berkualitas.

### **Proses Pembuatan Tempe**

Proses pembuatan tempe secara tradisional dapat dilakukan dalam sepuluh tahap, sebagai berikut :

#### 1. Proses Pencucian

Proses pembuatan tempe yang pertama adalah membersihkan kedelai. Dilakukan dengan cara dicuci menggunakan air bersih hingga kedelai terlihat bersih.

#### 2. Proses perendaman

Setelah di cuci bersih, kedelai di rendam selama kurang lebih 12 jam kemudian di tiriskan sebelum masuk kedalam proses perebusan.

#### 3. Proses Perebusan

Setelah direndam selama 12 jam , kedelai direbus menggunakan air bersih. Proses perebusan ini berlangsung selama 11/2 jam berfungsi untuk melepaskan kedelai dari kulitnya. Kemudian tiriskan lalu bilas menggunakan air bersih.

#### 4. Proses peredaman setelah perebusan

Setelah matang, kedelai kemudian direndam dalam. Perendaman dilakukan antara 18 jam hingga 20 jam sampai air rendaman mengental. Kemudian kedelai di saring lalu di giling.

#### 5. Proses Penggilingan

Proses berikutnya adalah menggiling kedelai agar biji-biji kedelai yang telah matang terbelah menjadi dua, dan memisahkan dari kulit arinya. Kedelai diangkat dari wadah rendaman dan dimasukkan ke dalam alat penggilingan. Penggilingan kedelai ini menggunakan mesin penggiling, Kemudian kedelai yang sudah digiling dibilas menggunakan air bersih.

#### 6. Proses Pengukusan

Bila sudah benar – benar bersih kedelai kemudian dikukus, tujuannya untuk menghilangkan kadar air sisa bilasan usai digiling dan mengurangi aroma

tak sedap setelah direndam semalaman. Pengukusan ini selama 2 jam, kemudian di dinginkan.

#### 7. Proses Pendinginan

Setelah kedelai panas dan tidak lagi terlihat basah, kedelai kemudian didinginkan. Kedelai diangkat dari panci lalu di dinginkan selama 2 jam sebelum proses berikutnya, yaitu pencampuran ragi.

#### 8. Proses Peragian

Setelah kedelai dingin, kedelai kemudian dicampur dengan ragi tempe. Ragi tempe ditaburkan pada kedelai dingin lalu dicampur hingga merata. Untuk takaran banyak sedikitnya ragi yang dicampurkan disesuaikan dengan kondisi suhu masing – masing daerah. Di daerah panas proses fermentasi akan lebih cepat dari daerah bersuhu dingin. Karena itu pemberian ragi lebih sedikit. Sedangkan di daerah bersuhu dingin proses fermentasi lebih lambat. Jadi, untuk pemberian ragi juga lebih banyak. Pencampuran ragi tempe sangat dipengaruhi oleh cuaca.

#### 9. Pengemasan

Proses pembuatan tempe berikutnya adalah pengemasan. Pengemasan dilakukan setelah kedelai dicampur dengan ragi. Kedelai dikemas menggunakan plastik bening. Kedelai dikemas sesuai ukuran besar kecil atau panjang pendek yang diinginkan.

#### 10. Proses fermentasi

Kemudian setelah pengemasan tempe mulai di tata dengan cara di jejer dan di beri sepaasi sekitar 2 cm. Lalu diamankan selama 24 jam hingga tempe berubah warna yang di sebut proses frementasi kemudian dapat di konsumsi.

(Sunarto Putra, 2022)

## **2.2 Inkubator Tempe**

Inkubator fermentasi tempe merupakan suatu alat yang dibuat untuk mengoptimalkan proses fermentasi tempe yang dapat meminimalisir kegagalan dalam proses pembuatan tempe.



Gambar 2.2 Inkubator Tempe

Suhu dan kelembapan menjadi faktor penting dalam proses pembuatan tempe. Apabila suhu dan kelembapan dilingkungan sekitar tidak stabil, berpengaruh terhadap waktu proses fermentasi yang lama hingga menyebabkan kegagalan dalam pembuatan tempe.

Pengoptimalan yang dilakukan inkubator fermentasi tempe yaitu dengan menjaga suhu dan kelembapan agar stabil, serta sistemnya dibuat secara otomatis. Inkubator fermentasi tempe dirancang secara *software* Arduino IDE serta secara *hardware* menggunakan beberapa komponen seperti sensor DHT22, ESP32, lampu, *heater*, kipas, LCD dan komponen lainnya yang dibutuhkan.

### **2.3 Remote control**

*Remote control* atau yang biasa disebut pengendali jarak jauh merupakan sebuah alat elektronik yang digunakan untuk mengoperasikan sebuah mesin dari jarak jauh. Istilah *remote control* juga sering disingkat menjadi remot saja. Pada umumnya, pengendali jarak jauh digunakan untuk memberikan perintah dari kejauhan kepada televisi atau barang-barang elektronik lainnya seperti *system stereo*, mainan dan pemutar DVD. (Perdana Kristyabudi, 2016)

Remot kontrol untuk perangkat biasanya berupa benda kecil nirkabel yang digenggam dengan sederetan tombol untuk menyesuaikan berbagai *setting*. Kebanyakan remot berkomunikasi dengan perangkatnya melalui sinyal-sinyal infra merah dan melalui sinyal radio.

### 2.3.1 Komponen *Remote control*

Komponen-komponen *remote control* yang dijelaskan adalah jenis *remote control* yang sering dijumpai di peralatan-peralatan elektronika rumah, menggunakan gelombang infra merah sebagai pembawa sinyal. Sebuah sistem *remote control* terdiri dari beberapa bagian :

- a. *Transmitter* (pengirim sinyal), Alat ini berfungsi untuk mengirimkan instruksi ke peralatan elektronika. Alat ini adalah sebuah LED (*light emitting diode*) sinar infra merah yang berada di pesawat *remote control*.
- b. Panel ini berisi sejumlah tombol di pesawat *remote control*. Setiap tombol memiliki fungsi yang berbeda-beda. Bentuk panel ini tergantung dari jenis alat yang dikendalikannya.
- c. Papan rangkaian elektronik, dalam setiap *remote control* terdapat sebuah papan rangkaian elektronik, dalam bentuk sirkuit terintegrasi. Fungsi komponen ini adalah membaca tombol yang ditekan pengguna kemudian membangkitkan *transmitter* untuk mengirimkan sinyal dengan pola sesuai tombol yang ditekan
- d. *Receiver* (penerima sinyal), alat ini berada di dalam alat elektronika yang akan menerima instruksi. Untuk jenis sinar infra merah alat yang digunakan adalah foto transistor infra merah. Alat ini berperan dalam mendeteksi pola sinyal infra merah yang dikirimkan *remote control*. Gelombang infra merah adalah salah satu nama untuk lebar frekuensi pada spektrum gelombang elektromagnetik. Pada spektrum gelombang elektromagnet, panjang gelombang *infrared* lebih panjang dari cahaya tampak dan lebih pendek dari gelombang radio.

### 2.3.2 Prinsip Cara kerja *remote control*

Prinsip cara kerja *remote control* sendiri sebetulnya cukup sederhana, sinyal sinar infra merah dipancarkan dari pemancar *remote control* membentuk pola sinyal tertentu. Selanjutnya pola sinyal tersebut akan diterima oleh peralatan elektronik, lalu pola sinyal tersebut akan diterjemahkan menjadi instruksi tertentu.

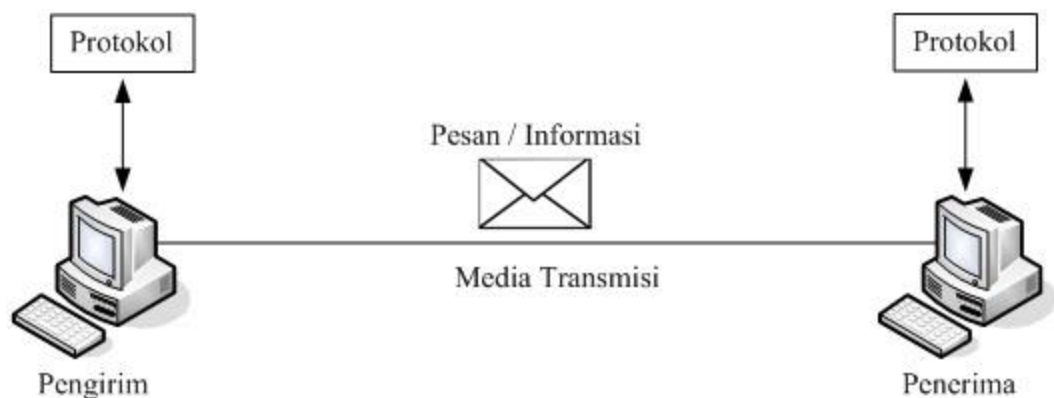


## 2.4 Komunikasi Data

Komunikasi data adalah proses pengiriman dan penerimaan data atau informasi dari dua atau lebih perangkat melalui beberapa media transmisi seperti kabel kawat, kabel optik dan gelombang radio. Perangkat berkomunikasi yang terdiri dari kombinasi perangkat keras (fisik atau peralatan perangkat keras) dan perangkat lunak (program) agar dapat terjadi komunikasi data.

Komunikasi data terdiri dari beberapa komponen seperti sumber pengirim, sistem transmisi, penerima, dan target tujuan.

1. Pesan (*Message*): Data atau informasi yang ingin dikirimkan dalam bentuk perangkat *input* dari pengirim ke penerima. Mengubah informasi audio, suara, video, atau teks menjadi unit data untuk diproses.
2. Pengirim (*Sender*): Sumber data atau informasi yang ingin dikirimkan ke penerima.
3. Penerima (*Receiver*): Tujuan akhir dari data yang ingin dikirimkan.
4. Media Transmisi (*Transmission Media*): Jalan atau media yang digunakan untuk mengirimkan data seperti kabel, gelombang radio, serat optik, dll.
5. Protokol (*Protocol*): Aturan dan prosedur yang digunakan untuk mengirim dan menerima data.



Gambar 2.3 Komunikasi Data

Dalam proses pertukaran informasi antara komputer, komunikasi komputer melibatkan pengiriman data antara dua sistem yang berbeda. Beberapa tugas khusus yang harus dipenuhi dalam proses transfer data antara dua komputer adalah:

1. Mengaktifkan jalur komunikasi data: Sistem sumber harus memastikan ada jalur komunikasi data langsung atau melalui jaringan komunikasi antara sistem sumber dan sistem tujuan.
2. Identifikasi sistem tujuan: Sistem sumber harus memberikan informasi identitas sistem tujuan kepada jaringan komunikasi agar data bisa diteruskan ke sistem yang tepat.
3. Persiapan sistem tujuan: Sistem sumber harus memastikan bahwa sistem tujuan siap menerima data dengan benar dan memiliki program manajemen file yang sesuai untuk menyimpan file dari beberapa pengguna tertentu.
4. Penerjemahan format file: Jika format file pada kedua sistem tidak kompatibel, salah satu sistem harus dapat melakukan fungsi penerjemahan format agar data bisa dipahami dan digunakan oleh sistem lain.

Komunikasi komputer melibatkan entitas dan sistem yang berbeda, termasuk program aplikasi user, program transfer file, sistem manajemen database, terminal, komputer, dan lainnya. *Protocol* adalah aturan yang memungkinkan entitas-entitas ini berkomunikasi dengan bahasa yang sama. Elemen penting dalam *protocol* termasuk :

- syntax (format data dan level sinyal),
- semantics (informasi kontrol dan pengendalian kesalahan),  
dan
- timing (urutan dan kecepatan komunikasi).

Arsitektur *protocol* adalah kerjasama tingkat tinggi di antara dua komputer yang melibatkan kesepakatan dan aturan dalam pertukaran data. Dengan menggunakan *protocol* dan arsitektur *protocol* yang tepat, komunikasi antara komputer dan jaringan komputer dapat berjalan lancar dan efisien.

## 2.5 ESP32

Mikrokontroler ESP32 dibuat oleh perusahaan bernama *Espressif Systems*. Perbedaan yang menjadi keunggulan mikrokontroler ESP32 dibanding dengan mikrokontroler yang lain, mulai dari pin out nya yang lebih banyak, pin analog lebih banyak, memori yang lebih besar, terdapat *bluetooth 4.0 low energy* serta

tersedia WiFi yang memungkinkan untuk mengaplikasikan *Internet of Things* dengan mikokontroler ESP32 (Suriana, Setiawan, & Graha, 2021)

### Spesifikasi mikrokontroler ESP32

ESP32 adalah mikrokontroler yang dikenalkan oleh *Espressif System* merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*. ESP32 sendiri tidak jauh berbeda dengan ESP8266 yang familiar di pasaran, hanya saja ESP32 lebih kompleks dibandingkan ESP8266, cocok untuk sobat dengan proyek yang besar. berikut ini merupakan spesifikasi yang dimiliki oleh mikrokontroler ESP32:

Atribut	Detail
CPU	Tensilica Xtensa LX6 32bit Dual-Core di 160/240MHz
SRAM	520 KB
FLASH	2MB (max. 64MB)
Tegangan	2.2V sampai 3.6V
Arus Kerja	Rata-rata 80mA
Dapat diprogram	Ya (C, C++, Python, Lua, dll)
Open Source	Ya
<b>Konektivitas</b>	
Wi-Fi	802.11 b/g/n
Bluetooth®	4.2BR/EDR + BLE
UART	3
<b>I/O</b>	
GPIO	32
SPI	4
I2C	2
PWM	8
ADC	18 (12-bit)
DAC	2 (8-bit)

Gambar 2.6 Tabel Spesifikasi Mikrokontroler ESP32

- *Prosesor: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz.*
- *Memori: 520 KB SRAM.*

- *Wireless connectivity: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE (shares the radio with Wi-Fi).*
- *Peripheral I/O: 12-bit SAR ADC (up to 18 channels), 2x 8-bit DACs, 10x touch sensors (capacitive sensing GPIOs), 4x SPI, 2x I2S interfaces, 2x I2C interfaces, 3x UART, SD/SDIO/CE-ATA/MMC/eMMC host controller, SDIO/SPI slave controller, Ethernet MAC interface, CAN bus 2.0, infrared remote controller (TX/RX, up to 8 channels), motor PWM, LED PWM (up to 16 channels), hall effect sensor, ultra low power analog pre-amplifier.*
- *Security : IEEE 802.11 standard security, secure boot, flash, encryption, 1024-bit, OTP (up to 768-bit for customers), cryptographic hardware acceleration (AES, SHA-2, RSA, ECC), random number generator (RNG).*

## **2.5 Internet of Things (IoT)**

*Internet of Things* atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen. (Efendi, 2018)

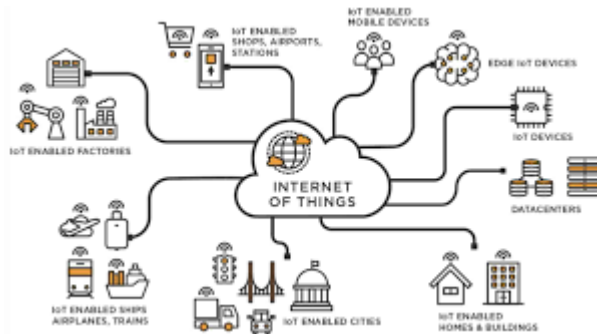
*Internet of Things* atau sering disebut IoT adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan internet sebagai penghubung. misalnya CCTV yang terpasang di sepanjang jalan dihubungkan dengan koneksi internet dan disatukan di ruang kontrol yang jaraknya mungkin puluhan kilometer. atau sebuah rumah cerdas yang dapat dimana lewat *smartphone* dengan bantuan koneksi internet. pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima sensor dan untuk analisa.

Ide awal *Internet of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai mendalami *Internet of Things* sebut saja *Intel, Microsoft, Oracle*, dan banyak lainnya.

Banyak yang memprediksi bahwa pengaruh *Internet of Things* adalah “ *the next big thing* ” di dunia teknologi informasi, hal ini karena IoT menawarkan banyak potensi yang bisa digali. Contoh sederhana manfaat dan implementasi dari *Internet of Things* misalnya adalah kulkas yang dapat memberitahukan kepada pemiliknya via SMS atau email tentang makanan dan minuman apa saja yang sudah habis dan harus di stok lagi.

### Cara Kerja Internet of Things

Konsep IoT ini sebetulnya cukup sederhana dengan cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, yakni: Barang Fisik yang dilengkapi modul IoT, Perangkat Koneksi ke Internet seperti Modem dan *Router Wireless Speedy* seperti di rumah anda, dan *Cloud Data Center* tempat untuk menyimpan aplikasi beserta *data base*.



Gambar 2.7 Konsep IoT

Dasar prinsip kerja perangkat IoT adalah benda di dunia nyata diberikan identitas unik dan dapat dikali di sistem komputer dan dapat di representasikan dalam bentuk data di sebuah sistem komputer. Pada awal-awal implementasi gagasan IoT pengenalan yang digunakan agar benda dapat diidentifikasi dan dibaca oleh komputer adalah dengan menggunakan kode batang (*Barcode*), Kode QR (*QR Code*) dan Identifikasi Frekuensi Radio (*RFID*). Dalam perkembangannya sebuah benda dapat diberi pengenalan berupa *IP address* dan menggunakan jaringan internet untuk bisa berkomunikasi dengan benda lain yang memiliki pengenalan *IP address*.

Cara Kerja *Internet of Things* yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang dimana tiap-tiap perintah argumennya itu menghasilkan sebuah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan manusia dan dalam jarak berapa pun.

Internetlah yang menjadi penghubung di antara kedua interaksi mesin tersebut, sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung.

## 2.6 Perangkat Lunak Pendukung

Dalam pembuatan sistem inkubator tempe dengan memanfaatkan sistem IoT ini tentunya harus ditunjang dengan diguankannya beberapa beberapa perangkat lunak seperti Arduino IDE, dan Blynk. Kedua perangkat lunak tersebut memiliki fungsi yang berbeda dalam membangun sistem inkubator tempe ini. Perangkat lunak Arduino IDE berfungsi untuk melakukukan pemrograman yang akan digunakan pada sistem inkubator tempe serta menghubungkan sistem dengan koneksi internet secara nirkabel. Sedangkan perangkat lunak Blynk berfungsi untuk menampilkan hasil pada sistem inkubator dalam bentuk grafik.

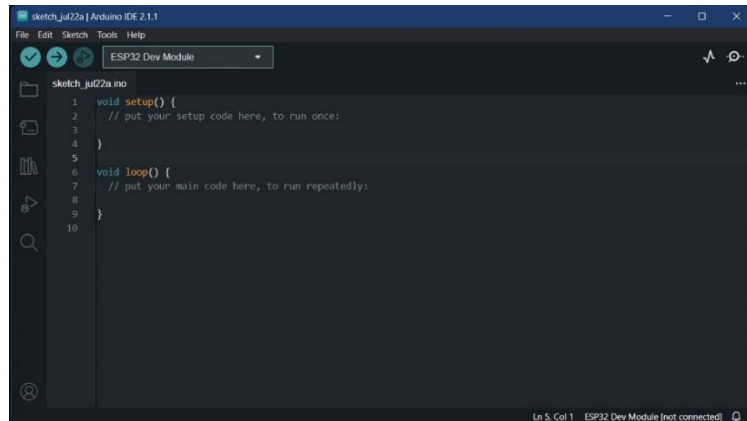
### 2.6.1 Arduino IDE



Gambar 2.8 Logo Arduino IDE

IDE itu merupakan kependekan dari *Integrated Development Enviroenment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC

mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.



Gambar 2.9 Tampilan Software Arduino IDE

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut Wiring yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Program yang ditulis dengan menggunakan Arduino *Software* (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi.

Penggunaan Perangkat lunak Arduino IDE pada sistem inkubator tempe digunakan untuk melakukan pemrograman yang mendukung terhadap pembacaan sensor DHT22 dengan mikrokontroler ESP32, dimana pembacaan sensor DHT dilakukan secara nirkabel dengan bantuan antena WiFi ESP32. Dengan menambahkan library WIFI dan WIFIClient dan DHT yang digunakan pada program untuk sistem pada Arduino IDE, pembacaan suhu secara nirkabel dapat dilakukan.

### 2.6.2 Blynk

Blynk adalah sebuah layanan *server* yang digunakan untuk mendukung *project Internet of Things*. Layanan *server* ini memiliki lingkungan *mobile user* baik Android maupun iOS. Blynk Aplikasi sebagai pendukung IoT dapat diunduh melalui *Google play* untuk pengguna *Android* dan melalui *App Store* bagi pengguna iOS. Blynk mendukung

berbagai macam *hardware* yang dapat digunakan untuk *project Internet of Things*. Blynk adalah *dashborad* digital dengan fasilitas antarmuka grafis dalam pembuatan *project*-nya. (Hariri, Novianta, & Kristiyana, 2019)



Gambar 2.10 Logo Blynk

1. Blynk *Apps*

Blynk *Apps* memungkinkan untuk membuat *project interface* dengan berbagai macam komponen *input output* yang mendukung untuk pengiriman maupun penerimaan data serta merepresentasikan data sesuai dengan komponen yang dipilih. Representasi data dapat berbentuk visual angka maupun grafik.

2. Blynk *Server*

Blynk *server* merupakan fasilitas *Back end Service* berbasis *Cloud* yang bertanggung jawab untuk mengatur komunikasi antara aplikasi *smartphone* dengan lingkungan *hardware*. Kemampuan untuk menangani puluhan *hardware* pada saat yang bersamaan semakin memudahkan bagi para pengembang sistem IoT.

3. Blynk *Library*

Blynk *Library* dapat digunakan untuk membantu pengembangan *code*. Blynk *library* tersedia pada banyak *platform* perangkat keras sehingga semakin memudahkan para pengembang IoT dengan fleksibilitas *hardware* yang didukung oleh lingkungan Blynk.

Dalam Blynk, terdapat protokol komunikasi yang digunakan untuk mengirim dan menerima data antara perangkat keras dan aplikasi Blynk. Berikut adalah beberapa protokol yang penting dalam Blynk:

1. Protokol Komunikasi:

Blynk *Protocol*: Blynk menggunakan protokol khusus yang dibangun di atas TCP/IP untuk menghubungkan perangkat keras dengan *server*



Blynk. Protokol ini memungkinkan pertukaran data antara perangkat dan aplikasi Blynk.

2. Protokol Autentikasi:

*Token Auth*: Saat membuat proyek di aplikasi Blynk, Anda akan diberikan token autentikasi. Token ini digunakan oleh perangkat keras untuk melakukan autentikasi dengan *server* Blynk, sehingga dapat terhubung dengan proyek yang telah Anda buat.

3. Protokol Pengiriman Data:

*Virtual pins*: Blynk menggunakan konsep *Virtual pins* untuk mengirim dan menerima data antara perangkat keras dan aplikasi. *Virtual pins* adalah saluran virtual yang dapat digunakan untuk menghubungkan *widget* di aplikasi Blynk dengan pin GPIO atau variabel di perangkat keras.

*Read/Write Operations*: Dalam protokol Blynk, Anda dapat melakukan operasi baca (*read*) dan tulis (*write*) ke *Virtual pins* untuk membaca data dari perangkat atau mengirim data ke perangkat.

4. Protokol *Widget*:

*Widget Event*: Saat *widget* di aplikasi Blynk diaktifkan atau digunakan, protokol Blynk akan mengirimkan peristiwa (*event*) ke perangkat keras. Misalnya, saat tombol ditekan, protokol akan mengirimkan peristiwa bahwa tombol telah ditekan, dan perangkat dapat merespons sesuai.

*Widget Property*: Beberapa *widget* di aplikasi Blynk memiliki properti khusus yang dapat dikonfigurasi, seperti rentang nilai pada *slider* atau tipe sensor pada *widget* sensor. Properti ini dapat dikirimkan dari aplikasi ke perangkat melalui protokol Blynk.

5. Protokol *Server-Push*:

*Server-Push*: *Server* Blynk dapat mengirimkan pesan atau instruksi ke perangkat keras melalui protokol *Server-Push*. Ini memungkinkan *server* Blynk untuk mengirim perintah atau pemberitahuan ke perangkat, seperti memperbarui nilai *widget* atau mengirimkan pesan.

## BAB III

### RANCANG BANGUN DAN INTEGRASI ALAT

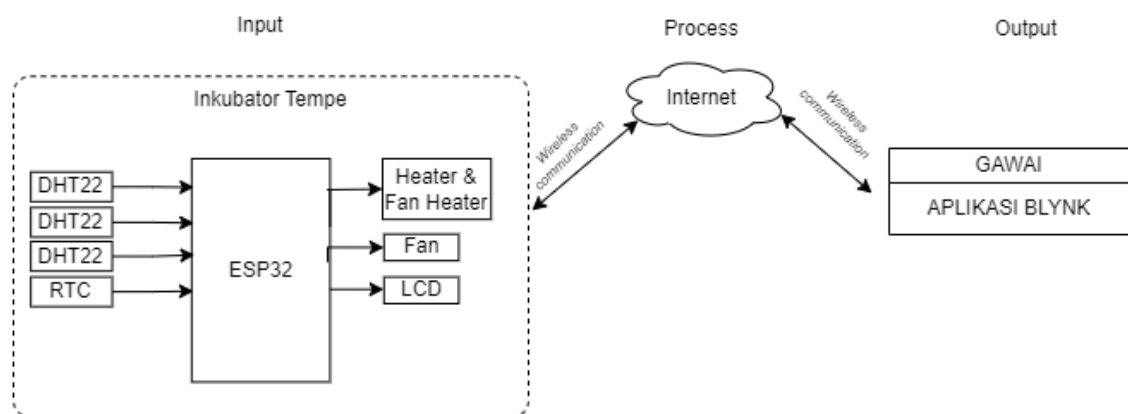
Pada bab ini akan dibahas mengenai tentang metodologi perancangan alat yang berisi pembahasan fungsi dari komponen yang digunakan pada inkubator fermentasi tempe konvensional. Pendekatan dari perancangan alat yang digunakan adalah secara fungsional dan struktural. Penjelasan tersebut ditujukan sebagai sarana untuk menjelaskan alat inkubator tempe yang akan dibuat.

#### 3.1 Perancangan Fungsional

Pada perancangan fungsional akan dibahas mengenai tentang penjelasan komponen sistem dari alat inkubator tempe. Penjelasan komponen sistem yang dirancang ditujukan sebagai sarana untuk menjelaskan beberapa komponen dan sistem yang digunakan pada rangkaian dari inkubator tempe.

##### 3.1.1 Diagram Blok

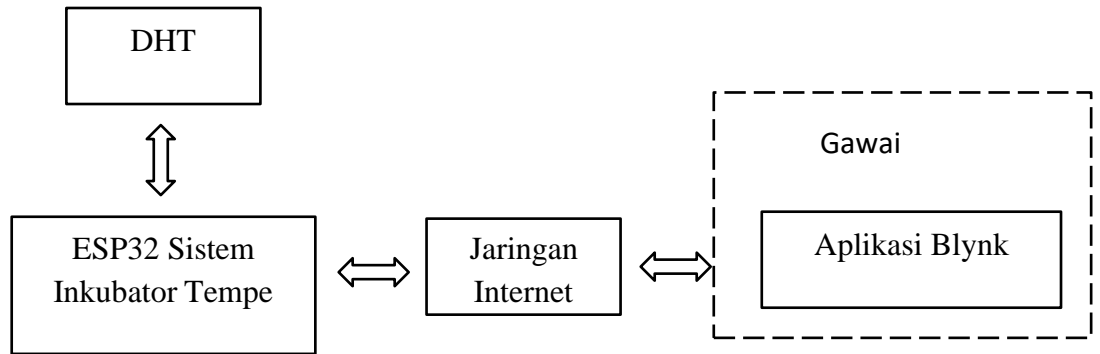
Diagram blok merupakan sebuah diagram berbentuk kotak (blok) yang digunakan sebagai gambaran mengenai sistem yang dirancang. Diagram blok sistem pada inkubasi ini ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Blok Sistem pada inkubator Tempe

Diagram blok sistem ini terdiri dari mikrokontroler ESP32 terdapat program yang akan memproses dan memberikan perintah kerja pada komponen yang ada pada *input* dan *output*. ESP32 digunakan untuk menghubungkan platform Blynk sebagai database, pengontrolan dan monitoring pada android .

### 3.1.2 Arsitektur Integrasi Sistem



Gambar 3.3 Blok Diagram Arsitektur Komunikasi Data

Arsitektur Komunikasi Data ditunjukkan pada gambar 3.3 Arsitektur ini terdiri dari Gawai yang terhubung dengan aplikasi Blynk memuat sistem pengontrol dan database inkubator tempe, jaringan internet untuk menghubungkan ESP32 dan gawai, DHT22 untuk pembacaan suhu dan kelembapan dan mikrokontroler ESP32 sebagai pengendali sistem inkubator tempe.

### 3.1.3 Protocol Stack

Tumpukan protokol atau tumpukan jaringan merupakan implementasi dari paket protokol jaringan komputer atau keluarga protokol. Dengan adanya *protocol stack* inilah semua aplikasi dan layanan di dalam jaringan komputer dapat berjalan dengan baik dan setiap protokol akan menjalankan tugasnya masing-masing. Protokol dalam rancangan integrasi pada penelitian ini ditunjukkan tabel 3.1.

<i>Application</i>	Bahasa Pemrograman C Aplikasi Blynk
<i>Transport</i>	Protokol komunikasi Serial seperti UART (Universal Asynchronous Receiver/Transmitter).
<i>Network</i>	TCP/IP (Transmission Control <i>Protocol/Internet Protocol</i> )
<i>Data Link</i>	<i>Template ID Blynk,</i> <i>Template Name Blynk</i>
<i>Physical</i>	USB Type A to Micro USB

Software

Hard

Gambar 3.4 Protocol Stack Rancangan Integrasi

Tabel 3.1 Protokol Stack

Lapisan	Deskripsi
Lapisan fisik ( <i>Physical</i> )	Menangani perangkat keras dan sinyal listrik yang digunakan untuk komunikasi. Menghubungkan papan ESP32 ke komputer menggunakan kabel USB.
Lapisan Transport ( <i>Transport</i> )	Mengatur transfer data antara papan ESP32 dan komputer menggunakan protokol komunikasi serial, seperti UART ( <i>Universal Asynchronous Receiver/Transmitter</i> ).
Lapisan jaringan ( <i>Network</i> )	Menangani komunikasi IP ( <i>Internet Protocol</i> ) antara ESP32 dan <i>server</i> Blynk. ESP32 harus terhubung ke jaringan WiFi dan menggunakan tumpukan TCP/IP ( <i>Transmission Control Protocol/Internet Protocol</i> ) untuk berkomunikasi melalui jaringan.
Lapisan Aplikasi ( <i>Application</i> )	Menggunakan protokol Blynk untuk berinteraksi dengan <i>server cloud</i> Blynk. Memungkinkan pengguna untuk membuat antarmuka pengguna di aplikasi Blynk, mengontrol proyek IoT, dan mengirim/menerima data melalui <i>server</i> Blynk.

### 3.2 Perancangan Struktural

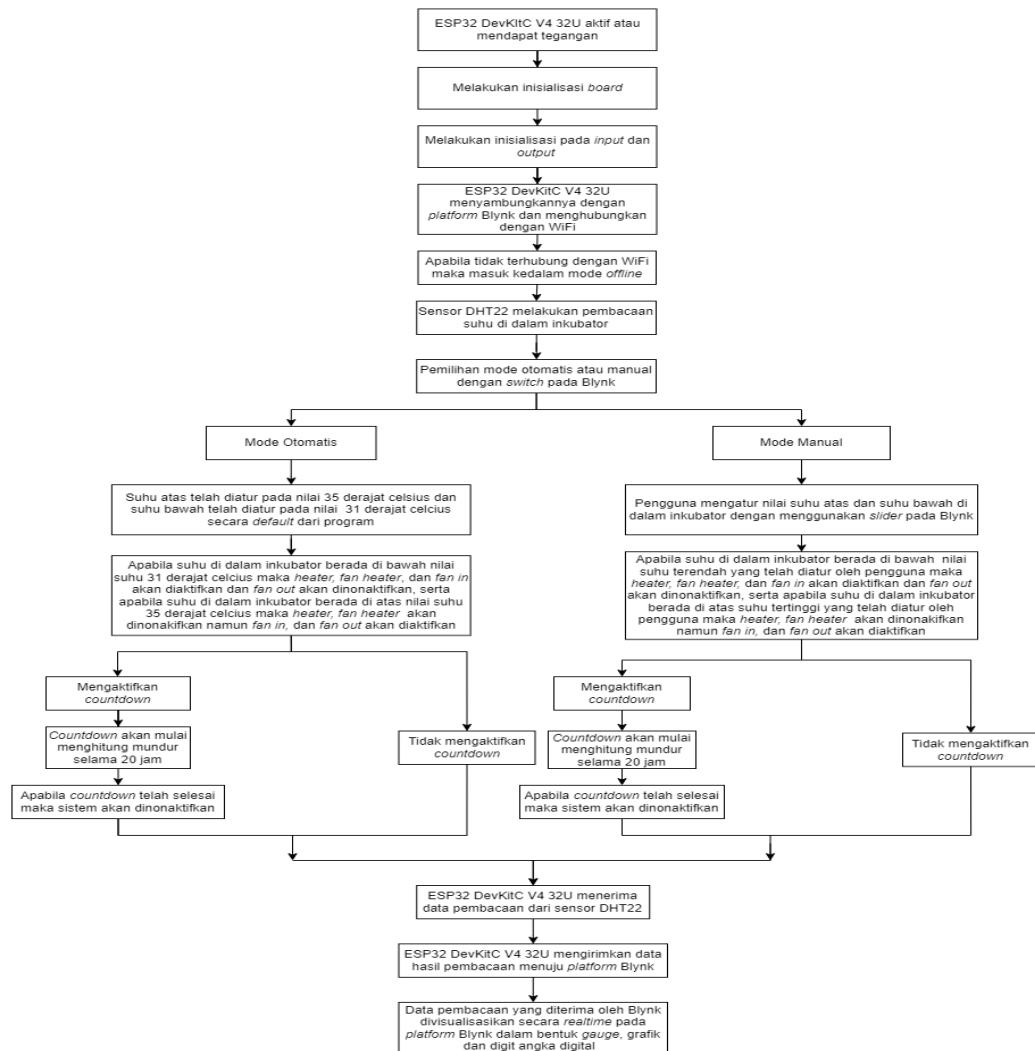
Perancangan integrasi struktural merupakan sebuah langkah awal terpenting dalam pembuatan alat. Perancangan awal dimulai dengan menyiapkan komponen dan bahan yang akan digunakan sesuai dengan bahan yang akan diperlukan agar karakteristik dari alat yang akan dibuat, pembuatan desain mekanik serta pemrograman. Pada dasarnya pemanfaatan sistem IoT pada inkubator tempe berbasis ESP32 adalah sistem pengontrolan dan monitoring suhu serta kelembapan didalam inkubator yang dapat diakses melalui *android smartphone* menggunakan aplikasi Blynk. Perancangan struktural bertujuan untuk mempermudah dalam proses perakitan dan perbaikan alat jika terjadinya kerusakan atau *error* pada proses pembuatan alat. Dalam proses perancangan dan pembuatan alat. Dalam proses pengerjaan perancangan ini terdapat beberapa tahap yaitu:

### 3.2.1 Perancangan Elektronika

Pada penelitian ini sebuah pengembangan dari suatu alat inkubator yang sebelumnya sudah dibuat yaitu dengan memanfaatkan sistem IoT yang tujuannya dapat mengontrol suhu dan memonitoring suhu dan kelembapan secara jarak jauh menggunakan aplikasi Blynk. hasil pembacaan sensor DHT suhu dan kelembapan yang telah didapat oleh sistem akan dikirimkan dengan jaringan nirkabel menuju perangkat keras android dan komputer untuk divisualisasikan dalam bentuk *gauge*, grafik dan digit angka digital pada *platform* Blynk.

### 3.2.2 Alur Kerja Sistem

Pada bagian ini akan dijelaskan tentang alur kerja sistem dalam yang digunakan untuk rancang bangun sistem kontrol suhu pada inkubator tempe.



Gambar 3.5 Flowchart Alur Kerja Sistem

Pada Gambar 3.5 diatas dapat dilihat alur kerja sistem yang dibentuk untuk sistem inkubator tempe, langkah awal yang dilakukan yaitu memastikan ESP32 aktif atau mendapatkan tegangan, lalu melakukan inisialisasi board, melakukan inisialisasi *input* dan *output* sebelum menjalankan sistem, setelah itu ESP32 yang merupakan mikrokontroler akan menyambungkan pada *platform* Blynk dan menghubungkan dengan jaringan WiFi, setelah ESP32 terhubung dengan WiFi maka sensor DHT akan membaca suhu didalam inkubator, dan apabila ESP32 tidak terhubung dengan WiFi maka masuk kedalam mode offline, lalu dalam aplikasi Blynk terdapat dua mode yaitu otomatis dan mode manual, pada mode otomatis suhu atas dan bawah sudah diatur didalam program yaitu dengan nilai 35°C untuk suhu atas di dalam inkubator dan nilai 31°C untuk suhu bawah di dalam inkubator, apabila suhu di dalam inkubator berada dibawah 31 °C maka *heater* dan *fan heater*, *fan in* akan diaktifkan, sedangkan *fan out* akan dinonaktifkan, sebaliknya apabila suhu di dalam inkubator berada diatas 35 °C maka *heater* dan kipas *heater*, *fan out* akan dinonaktifkan sedangkan *fan in* akan diaktifkan. Lalu mengaktifkan *Countdown* melalui *button* yang ada pada aplikasi Blynk, *countdown* akan hitung mundur selama 20 jam, apabila *countdown* telah selesai maka sistem akan dinonaktifkan, sedangkan apabila *countdown* tidak diaktifkan maka sistem akan terus berjalan tanpa berhenti. pada mode manual suhu batas atas dan suhu batas bawah di dalam inkubator dapat diatur oleh pengguna dengan menggunakan *slider* pada aplikasi Blynk, apabila suhu di dalam inkubator di bawah suhu yang telah diatur oleh pengguna maka *heater* dan *fan heater*, *fan in* akan diaktifkan, sedangkan *fan out* akan dinonaktifkan, sebaliknya apabila suhu di dalam inkubator berada di atas suhu yang telah diatur oleh pengguna maka *heater* dan kipas *heater*, *fan out* akan dinonaktifkan sedangkan *fan in* akan diaktifkan. Lalu mengaktifkan *Countdown* melalui *button* yang ada pada aplikasi Blynk, *countdown* akan hitung mundur selama 20 jam, apabila *countdown* telah selesai maka sistem akan dinonaktifkan, sedangkan apabila *countdown* tidak diaktifkan maka sistem akan terus berjalan tanpa berhenti. Selanjutnya ESP32 akan menerima data pembacaan dari sensor suhu kemudian hasil pembacaan data tersebut akan ditransmisikan oleh ESP32 menuju *platform* Blynk yang mana setelahnya data tersebut akan divisualisasikan dalam bentuk *gauge*, digit angka digital, dan grafik secara *realtime*.

### 3.3 Integrasi Bagian Hardware

Integrasi *hardware* pada pemanfaatan sistem Iot berbasis ESP32 untuk *remote control* inkubator tempe dengan mengintegrasikan tiga perangkat keras yang memiliki fungsi berbeda, dimana integrasi ini berhubungan dengan komunikasi data pada lapisan TCP/IP (*Transmission Control Protocol/Internet Protocol*) model, yaitu lapisan fisik. Integrasi hardware yang akan di bangun ialah sebagai berikut:

- PC/Laptop – ESP32, pada PC/Laptop telah tersedia port USB, sedangkan pada ESP32 hanya memiliki port micro USB maka dari itu memerlukan *cabel* USB *type A to micro USB* agar ESP32 dapat terkoneksi dengan PC/laptop melalui port USB.
- ESP32 – DHT22, pada sensor DHT22 telah tersedia port GPIO sehingga memerlukan kabel *jumper* agar dapat berkomunikasi dengan ESP32.

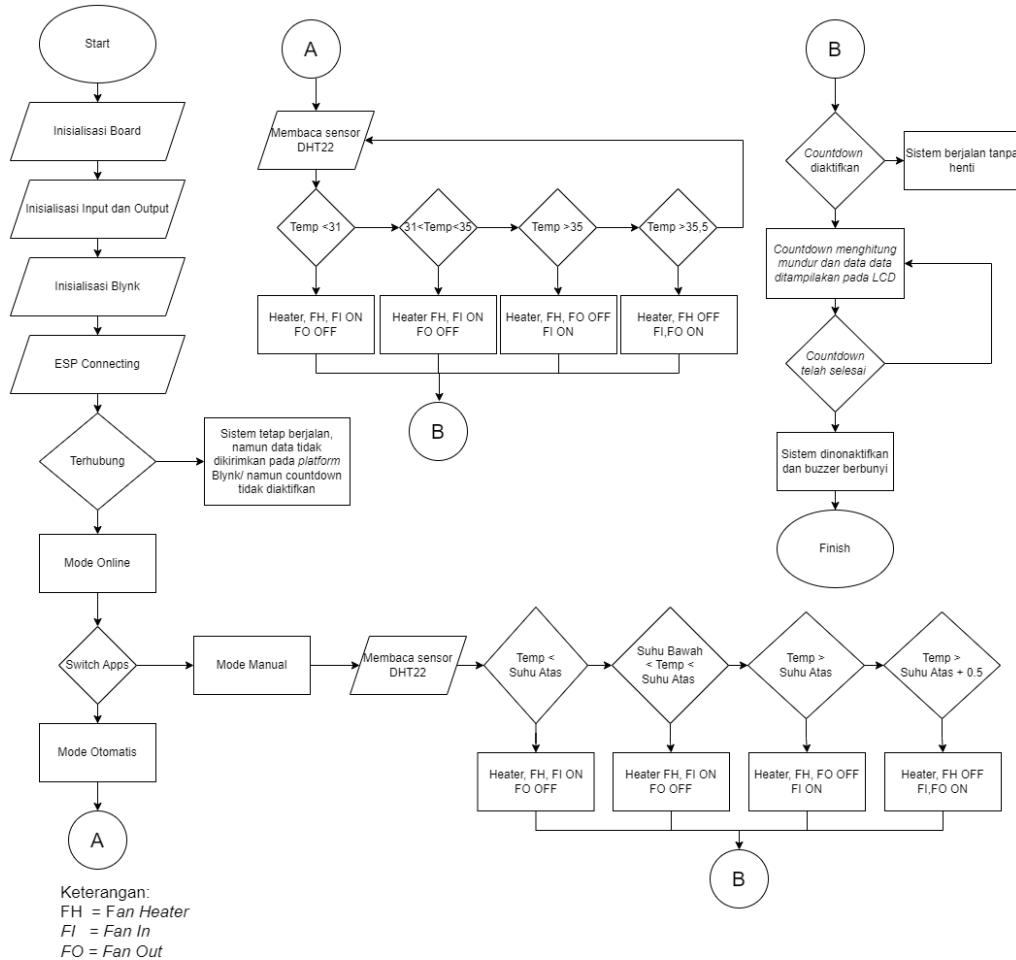
Dengan menggunakan kabel USB tipe A to micro USB dan kabel jumper, maka dapat dengan mudah menghubungkan PC/Laptop dengan ESP32 dan sensor DHT22 sehingga dapat melakukan pemrograman dan berkomunikasi dengan sensor DHT22 melalui ESP32.

### 3.4 Perancangan Integrasi Perangkat lunak atau Software

Perancangan integrasi perangkat lunak atau *software* dilakukan dengan menghubungkan bagian dari protokol stack, dan perancangan perangkat lunak dibagi dalam 2 bagian antara lain perancangan pemrogram, integrasi hubungan komunikasi antara sistem dengan *platform* Blynk dan integrasi perangkat lunak API (*Application Programming Interface*). Perancangan pemrograman menggunakan *software* Arduino IDE menggunakan PC yang akan digunakan agar ESP3 dapat mengelola setiap data *input* dan *output*. Sebagai gambaran umum dari integrasi yang dibutuhkan, maka dibuat *flowchart* sistem yang akan menjadi acuan dari program Arduino IDE dan aplikasi Blynk, agar alat dapat berfungsi sesuai prinsip kerja yang direncanakan dan dapat mengoneksikan ke sistem *internet of things* yang berfungsi untuk mengatur suhu dan memonitoring suhu dan kelembapan pada aplikasi Blynk.

### 3.4.1 Flowchart System

Flowchart atau diagram alir perancangan integrasi bagian perangkat lunak keseluruhan yaitu diagram alir sistem inkubator tempe.



Gambar 3.5 Flowchart System Inkubator Tempe

Pada gambar 3.5 terdapat *flowchart* sistem yang digunakan dalam *remote control* inkubator tempe. Gambar diagram tersebut menjelaskan bagaimana sistem bekerja. Langkah awal yang dilakukan yaitu inisialisasi mikrokontroler ESP32 sebelum menjalankan sistem, lalu inisialisasi *input* dan *output*, setelah itu ESP32 merupakan mikrokontroler yang akan menyambungkan sistem dengan Blynk dan menghubungkan jaringan WiFi, dengan memasukkan Blynk templete ID, dan Blynk *templete* nama ke dalam pemrograman, lalu apabila ESP32 telah terhubung dengan Blynk dan WiFi maka terhubung langsung dengan mode online sedangkan apabila ESP tidak terhubung pada jaringan internet maka sistem tetap berjalan namun data tidak



dapat dikirimkan pada *platform* Blynk dan *countdown* tidak dapat diaktifkan, dalam mode online terdapat *swicth* yang dapat mengubah mode otomatis dan mode manual, lalu pada mode otomatis pengaturan suhu telah ditentukan dengan suhu batas atas yaitu 35°C dan batas bawah yaitu 31°C, apabila sensor membaca suhu di dalam inkubator berada di bawah 31°C maka *heater* dan *fan heater*, *fan in* diaktifkan (*ON*) , *fan out* dinonaktifkan (*OFF*), apabila sensor membaca suhu di dalam inkubator di atas 35°C maka *heater* dan *fan heater*, *fan out* dinonaktifkan (*OFF*), *fan in* diaktifkan (*ON*), dan apabila sensor membaca suhu di dalam inkubator berada diantara 31°C sampai dengan 35 °C maka *heater* dan *fan heater*, *fan out* dinonaktifkan (*ON*), *fan in* akan diaktifkan (*ON*), namun apabila sensor membaca suhu di dalam inkubator membaca 35,5 °C atau lebih maka *heater* dan *fan heater* dinonaktifkan (*OFF*), *fan in* dan *fan out* akan diaktifkan (*ON*).

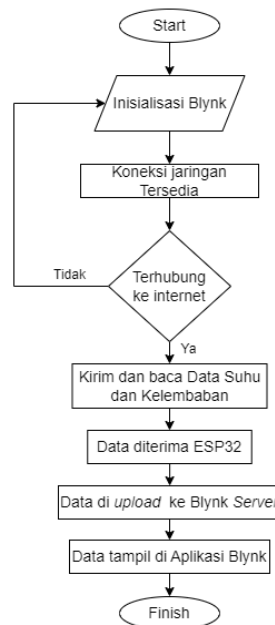
Pada mode manual pengaturan suhu dapat ditentukan atau diatur oleh pengguna dengan cara menggeser *slider* batas suhu atas dan batas suhu bawah pada aplikasi Blynk, apabila sensor membaca suhu di dalam inkubator berada di bawah batas suhu bawah yang telah diatur oleh pengguna maka *heater* dan *fan heater*, *fan in* diaktifkan (*ON*) , *fan out* dinonaktifkan (*OFF*), apabila sensor membaca suhu di dalam inkubator di atas batas suhu atas yang telah diatur oleh pengguna maka *heater* dan *fan heater*, *fan out* dinonaktifkan (*OFF*), *fan in* diaktifkan (*ON*), dan apabila sensor membaca suhu di dalam inkubator berada diantara batas suhu bawah sampai dengan batas suhu atas yang telah diatur oleh pengguna, maka *heater* dan *fan heater*, *fan out* dinonaktifkan (*ON*), *fan in* akan diaktifkan (*ON*), namun apabila sensor membaca suhu di dalam inkubator membaca berada lebih tinggi 0,5°C dari batas suhu atas yang telah diatur oleh pengguna atau lebih maka *heater* dan *fan heater* dinonaktifkan (*OFF*), *fan in* dan *fan out* akan diaktifkan (*ON*).

Lalu pada *platform* Blynk telah dibuatkan *toggle* atau tombol *countdown* bila di tekan akan mengaktifkan sistem perhitungan mundur selama 20 jam, setelah perhitungan mundur telah selesai maka sistem akan dinonaktifkan dan buzzer akan berbunyi, apabila *countdown* tidak diaktifkan maka sistem akan terus berrjalan tanpa henti, namun selama sistem

masih berjalan secara keseluruhan maka seluruh data akan ditampilkan pada LCD (keterangan, suhu, kelembapan, jam, tanggal, dan *countdown*).

### 3.4.2 Hubungan Komunikasi Antara Sistem dengan Platform IoT Blynk

Apabila data yang diperoleh dari hasil pembacaan beberapa sensor sudah didapat, maka proses selanjutnya adalah menampilkan data di platform IoT Blynk. Adapun sensor-sensor yang datanya akan ditampilkan di platform IoT diantaranya sensor suhu. Supaya bisa menampilkan data yang diperoleh di platform IoT, maka mikrokontroler ESP32 harus terkoneksi dengan jaringan internet. Apabila mikrokontroler ESP32 tidak terkoneksi dengan jaringan internet, data yang diperoleh dari pembacaan sensor tidak bisa ditampilkan di platform IoT *controlling* pada Blynk. Untuk lebih jelas hubungan antara sistem dengan platform IoT Blynk bisa dilihat pada Gambar 3.7.



Gambar 3.7 Flowchart Sistem dengan Platform IoT Blynk

### 3.4.3 Perancangan Komunikasi antar Aplikasi (API)

Berdasarkan data yang telah diperoleh bahwa untuk memungkinkan komunikasi data antar peralatan harus memperhatikan *interface* pemrograman aplikasinya (API), maka Perancangan API pada tugas ini diperlukan agar integrasi data antar perangkat dapat berjalan

dengan baik. Dengan merujuk pada diagram alir pada bab 3 sebelumnya API yang akan di bangun pada penelitian ini ialah:

➤ API untuk Komunikasi antara ESP32-DHT22

Basis program sensor DHT22 menggunakan bahasa C, dengan demikian sensor DHT22 dapat dikomunikasikan dengan program pada mikrokontroler ESP32 dengan menggunakan Library DHT22 yang sudah disediakan oleh software Arduino IDE sebagai software pemrograman ESP32.

➤ API untuk Komunikasi antara ESP32-Blynk

Untuk Mengkomunikasikan ESP32 dengan Blynk melalui koneksi nirkabel (WiFi) dan menggunakan API Key Blynk untuk berkomunikasi dengan perangkat gawai secara nirkabel dengan menambahkan library Blynk pada program Arduino IDE.

### 3.5 Manufaktur Aplikasi Blynk

Tahapan dari rancangan aplikasi sistem pengendali suhu dan kelembapan pada inkubator tempe berbasis *android* menggunakan aplikasi Blynk, pada aplikasi Blynk terdapat beberapa *future* yang dapat digunakan sesuai dengan fungsi dan program yang akan dibuat.



Gambar 3.8 Tampilan Aplikasi Blynk

Pada Gambar 3.8 merupakan tampilan aplikasi Blynk untuk inkubator tempe, untuk proses *control* dan monitoring pada inkubator tempe sehingga dapat divisualisasi

dalam bentuk *gauge*, digit digital, dan grafik serta pengaturan suhu pada *platform* Blynk dibutuhkan beberapa langkah yaitu:

a. *Template ID* dan *Auth token*

1. Masuklah ke Blynk.Console.
2. Pilih menu "*Template*" kemudian pilih "+*New Template*"
3. Tuliskan Nama dan jenis *Device* yang akan gunakan untuk *project* kali ini. Misalnya *My Project*. Kemudian Klik DONE.
4. Setelah sudah, maka akan dimunculkan *template ID* dan nama *Device* yang kita akan gunakan. *Copy* dan anda sudah dapat menambahkannya ke program di Arduino IDE.
5. Selanjutnya untuk mendapatkan *Auth Token*, kita dapat klik menu *Search > Devices > My Devices*. Kemudian pilih *New Device*.
6. Pilih *From Template* untuk membuka *Device* dari dari *template*.
7. Kemudian pilih *template* yang sudah dibuat sebelumnya dan pada *Device Name*, tuliskan nama untuk *device* yang akan digunakan nantinya. Pilih *Create*.
8. *Auth Token* sudah dimunculkan untuk *project* kali ini.
9. Masukkan semua informasi ke dalam program Arduino IDE. Kemudian *upload* program untuk mengkoneksikan antara ESP32 dengan Blynk *Cloud*.

b. Setting Datastream

1. Kembali ke *Template*. Kemudian klik *Template* tadi dan pilih Datastreams.
2. Klik Edit pada pojok kanan atas. Kemudian pilih *New DataStream > Virtual pin*.
3. Pilih Datastream dan pilih *Virtual pin* di drop menu.
4. Setting Datastream seperti berikut ini.
  - a. Untuk *temperature* dengan menggunakan *virtual pin* V0, data *type double*, unitsnya celcius, °C, dan atur nilai MIN/MAX 0-100 sesuai dengan range suhu.
  - b. Untuk *humidity* yaitu dengan menggunakan *virtual pin* V1, data *type integer*, unitsnya persentase, %, dan atur nilai MIN/MAX 0-100 sesuai dengan range kelembapan.
  - c. Untuk suhu batas atas dengan menggunakan *virtual pin* V2, data *type double*, units Celcius, °C, dan atur nilai MIN/MAX 0-100

- d. Untuk suhu batas atas dengan menggunakan *virtual pin* V3, data *type double*, units Celcius, °C, dan atur nilai MIN/MAX 0-100
- e. Untuk tombol mode menggunakan *virtual pin* V4, dengan data *type integer*, unitsnya none, nilai MIN/MAX 0/1 karena unuk *high* atau *low* saja.
- f. Untuk tombol *start* menggunakan *virtual pin* V6, dengan data *type integer*, unitsnya none, nilai MIN/MAX 0/1 karena unuk *high* atau *low* saja.
- g. Untuk tombol *se* menggunakan *virtual pin* V7, dengan data *type integer*, unitsnya none, nilai MIN/MAX 0/1 karena unuk *high* atau *low* saja.

c. Setting *Dashboard Web*

*Dashboard Web* digunakan untuk membuat *widget* pada tampilan website, *Widget* biasanya berisi terkait controlling dan monitoring. Perhatikan caranya sebagai berikut.

1. Masuk ke *Web Dashboard*.
2. *Drag n Drop Gauge, Slider, Superchart* (grafik) dan *Button*.
3. Pilih Setting pada *Gauge* dan setting untuk tampilan monitoring *temperature* dan *humidity* menggunakan data stream V0 dan V1, lalu setting nama design dan warnanya, *temperature* untuk V0 dengan warna merah dan *humidity* untuk V1 dengan warna hijau.
4. Pilih Setting pada *Slider* dan setting untuk controlling *temperature* batas atas dan batas bawah menggunakan datastream V2 dan V3, lalu setting nama design Temp Up (suhu batas atas) untuk V2 dan Temp Down(suhu batas bawah) untuk V3
5. Pilih Setting pada *Button* dan setting untuk controlling mode menu otomatis dan manual menggunakan datastream V4, lalu setting dengan mode switch nama design mode menu dan labels manual dan otomatis.
6. Pilih Setting pada *Button* dan setting untuk controlling *Countdown Start* dan *Stop* menggunakan datastream V6, lalu setting dengan mode switch nama design *Countdown* dan labels *Start* dan *Stop*.

7. Pilih Setting pada *Button* dan setting untuk controlling *Countdown* Set dan Reset menggunakan datastream V7, lalu setting dengan mode switch nama design *Countdown* dan labels Set dan Reset.
  8. Pilih Setting pada *Button* dan Setting untuk monitoring dalam bentuk grafik dengan menggunakan datastream V0 dan V1.
- d. Setting *Dashboard* Aplikasi Blynk di Android
1. Buka aplikasi Play Store di android, lalu cari Blynk apps pada kolom pencaharian. kemudian unduh dan instal Blynk apps tersebut, dan tunggu sampai selesai.
  2. Kemudian buka aplikasinya, lalu masukan email dan password yang sudah didaftarkan.
  3. Lalu buka *Switch to Develover Mode* (icon kunci inggris) dipojok kanan bawah, lalu klik garis tiga dipojok kanan atas. Pilih '+Add Template', tuliskan nama *Template* misalkan "My Project" lalu tekan *continue*.
  4. Kembali ke *Template*. Kemudian klik *Template* yang telah dibuat sebelumnya dan pilih *Datastreams*.
  5. Untuk memasang komponen atau *widget* yang diperlukan klik tombol '+' untuk menambahkan *widget*.
  6. Pilih *widget gauge, slider, Superchart* (grafik) dan *Button*.
  7. Pilih Setting pada *gauge, slider, superchart* (grafik) dan *button* sama seperti pada *dashboard web*.
  8. Setelah *widget* terpasang sesuai dengan datastream yang sudah dibuat.
  9. Lalu koneksikan Blynk dengan WiFi dengan mengklik garis tiga dipojok kanan atas lalu klik '+ Add New Device' -> find devices nearby.
  10. Nyalakan WiFi dan Bluetooth lalu klik *start*, ikuti langkah selanjutnya, *choose your device*, tunggu sampai *connect*, lalu konfigurasi ssid dan *password* WiFi yang ingin digunakan.
  11. Lalu tunggu configuring *Device connected* lalu klik *finish*
  12. Setelah selesai *device* dapat digunakan.

## **BAB IV**

### **HASIL PENGUJIAN DAN ANALISA DATA**

Pada bab ini menjelaskan tentang serangkaian hasil pengujian yang dilakukan dan analisis data hasil pengujian. Pengujian bertujuan untuk mengetahui fungsi dan kinerja dari masing masing komponen dan rangkaian serta untuk mengetahui apakah *system* dapat bekerja sesuai dengan apa yang direncanakan.

Pengujian yang dilakukan antara lain:

1. Pengujian Mikrokontroler ESP32
2. Pengujian pembacaan sensor DHT22
3. Pengujian konektivitas sistem dengan *platform* Blynk
4. Pengujian sistem secara keseluruhan

#### **4.1 Pengujian Mikrokontroler ESP32**

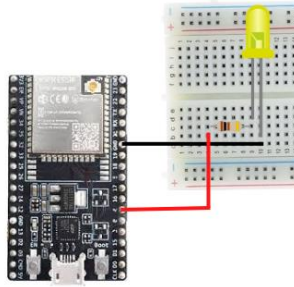
##### **4.1.1 Tujuan Pengujian**

Pengujian ini bertujuan untuk memastikan bahwa mikrokontroler ESP32 yang digunakan dapat bekerja dengan baik.

##### **4.1.2 Langkah pengujian**

Pengujian ini dapat dilakukan dengan cara menghubungkan LED dengan mikrokontroler ESP32 menggunakan komunikasi GPIO (*General Purpose Input/Output*), kemudian mikrokontroler ESP32 dihubungkan dengan komputer menggunakan kabel mikro USB untuk melakukan pemrograman. Seperti pada gambar 4.1. Alat dan bahan yang dibutuhkan pada pengujian ini yaitu:

1. Mikrokontroler ESP32
2. LED
3. Perangkat Lunak Arduino IDE
4. *Project Board*
5. Kabel *Jumper*
6. Kabel mikro USB
7. Voltmeter
8. Resistor 100 Ohm



Gambar 4.1 Rangkaian pengujian ESP32

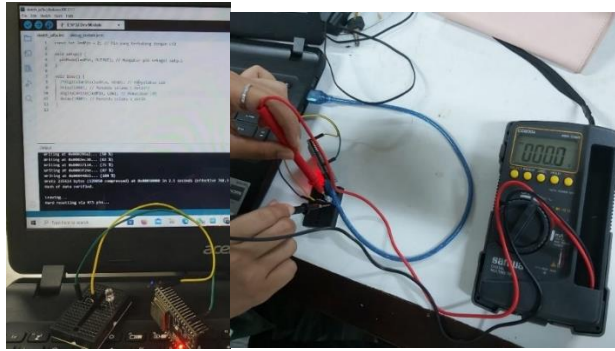
Langkah- langkah pengujian konektivitas ESP32 yang dilakukan sebagai berikut:

1. Alat dan bahan dalam pengujian ini dipersiapkan
2. Komponen disusun seperti pada gambar 4.1
3. Perangkat lunak Arduino IDE diprogram untuk memberikan perintah *HIGH* pada LED.
4. Menghubungkan ESP32 pada komputer dengan menggunakan kabel mikro USB
5. Program di *upload* dari komputer menuju mikrokontroler ESP32
6. Perhatikan reaksi dari LED serta tegangan yang terukur oleh voltmeter, lalu hasilnya dicatat.
7. Ulangi percobaan dengan mengganti perintah pada LED menjadi *LOW*.
8. Perhatikan kembali reaksi dari LED serta tegangan yang diukur oleh voltmeter, lalu catat kembali hasil yang didapatkan

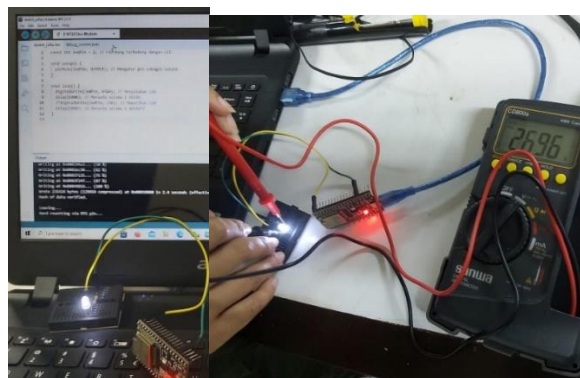
#### 4.1.3 Hasil Pengujian dan Analisa mikrokontroler ESP32

Pada pengujian ini digunakan lampu LED sebagai indikator berfungsinya mikrokontroler ESP32 yang digunakan. Ketika diberikan program perintah *HIGH* (mengaktifkan *output* pada pin yang terhubung ke LED) maka tegangan yang terukur pada voltmeter senilai 2,69 volt lampu LED menyala (LED *ON*), sedangkan ketika diberikan program perintah *LOW* (mematikan *output* pada pin yang terhubung ke LED) maka tegangan yang terukur pada voltmeter senilai 0 volt Lampu LED mati (LED *OFF*). Untuk hasil pengujian ESP32 dapat dilihat pada Gambar 4.1 dan Gambar 4.2.





Gambar 4.2 Kondisi pin *low*



Gambar 4.3 Kondisi pin *high*

Tabel 4.1 Pengujian ESP32

Kondisi Pin	Tegangan	Indikator LED
<i>HIGH</i>	2.69 Volt	Nyala
<i>LOW</i>	0 Volt	Mati

Berdasarkan hasil pengujian yang telah dilakukan, dengan menunjukkan kondisi *HIGH* dengan tegangan yang diukur sebesar 2.69 volt sehingga membuat lampu LED menyala dan kondisi *LOW* dengan tegangan terukur sebesar 0 volt sehingga lampu LED mati, maka dari hasil pengujian tersebut dapat dinyatakan bahwa mikrokontroler ESP32 yang digunakan dapat berfungsi dengan baik, sehingga seluruh pin yang tersedia dapat digunakan.

## 4.2 Pengujian Pembacaan Sensor DHT22

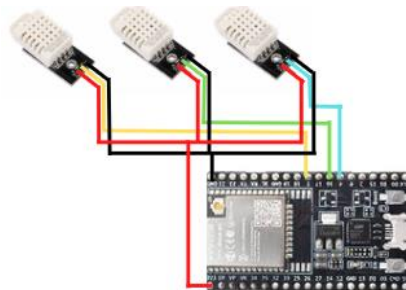
### 4.2.1 Tujuan Pengujian

Pengujian Pembacaan Sensor DHT22 bertujuan untuk memastikan bahwa sensor DHT22 dapat mengukur suhu dan kelembapan dengan akurasi yang memadai. Pengujian ini bertujuan untuk memastikan bahwa sensor berfungsi dengan baik dan memberikan data yang konsisten dan dapat diandalkan.

### 4.2.2 Langkah pengujian

Pengujian ini dapat dilakukan dengan menghubungkan 3 buah pin GPIO (*General Purpose Input/Output*) untuk pemrosesan data, yang tersedia pada mikrokontroler ESP32, lalu ESP32 dihubungkan dengan komputer menggunakan kabel mikro USB seperti pada gambar 4.4 untuk dilakukan pemrograman. Alat yang dibutuhkan pada pengujian ini yaitu:

1. Mikrokontroler ESP32
2. 3 buah sensor DHT22
3. Perangkat keras komputer
4. Perangkat lunak Arduino IDE dengan program yang telah dibuat
5. Kabel penghubung (kabel *jumper*)
6. Kabel mikro USB
7. *Project board*
8. Tang Ampere yang dapat mengukur suhu



Gambar 4. 4 Rangkaian Pengujian DHT22

Langkah- langkah pengujian pembacaan sensor DHT22 yang dilakukan sebagai berikut:

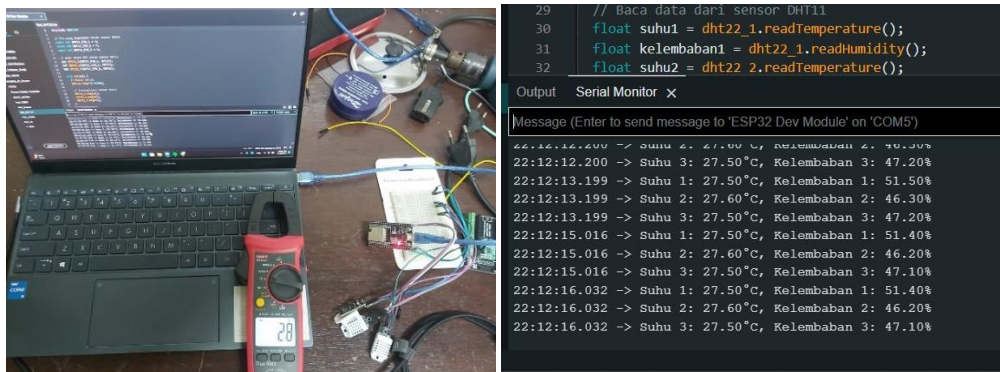
1. Persiapkan alat serta semua komponen yang dibutuhkan.

2. Rangkaian yang disusun seperti pada gambar 4.4
3. Perangkat lunak Arduino IDE diprogram dengan memasukkan *library* khusus yang diperlukan untuk mendukung sensor DHT22.
4. Menghubungkan perangkat lunak ESP32 pada komputer dengan kabel mikro USB
5. Program diupload dari komputer menuju mikrokontroler ESP32.
6. Perhatikan serial monitor pada perangkat komputer. Catat hasil pembacaan suhu dan kelembapan yang ditampilkan pada serial monitor.
7. Ulangi percobaan dengan pemrograman yang berbeda yaitu membaca suhu rata-rata yang diambil oleh 3 buah sensor.
8. Perhatikan kembali hasil pembacaan yang ditampilkan pada serial monitor, lalu catat kembali hasil pembacaan sensor.

#### **4.2.3 Hasil Pengujian dan Analisa Pembacaan DHT22**

Pada pengujian ini dilakukan dua buah percobaan. Pertama, dilakukan pembacaan masing-masing sensor DHT22 yang diletakkan pada suhu ruangan untuk mengukur temperatur dan kelembapan yang dibaca oleh masing-masing sensor. Sedangkan pada percobaan kedua, dilakukan pembacaan rata-rata temperatur dan kelembapan yang telah diprogram untuk melihat suhu rata-rata yang dihasilkan dari 3 buah sensor DHT22 yang diletakkan pada suhu ruangan. Pembacaan dilakukan sebanyak sebanyak 10 data yang muncul pada serial monitor dan memiliki nilai simpangan untuk masing-masing hasil pembacaan yang dilakukan, dimana nilai simpangan tersebut merupakan selisih hasil pembacaan dengan nilai sebenarnya, sehingga dapat mengetahui keakuratan

hasil pembacaan dengan nilai sebenarnya, sehingga dapat mengetahui kakuratan dari hasil pembacaan yang dilakukan oleh sensor DHT22



Gambar 4.5 Pengujian masing-masing sensor DHT22

Setelah melakukan pengujian pembacaan sensor DHT22, pada Gambar 4.5 menunjukkan bahwa tang ampere menunjukan bahwa suhu ruangan saat ini adalah 28°C, sedangkan pembacaan *temperature* dan kelembapan pada DHT 1 membaca suhu sebesar 27.50°C, kelembapan sebesar 51.40% , DHT 2 membaca suhu sebesar 27.60°C, kelembapan sebesar 46.20%, dan DHT 3 membaca suhu sebesar 27.50°C, kelembapan sebesar 47.10%. Rata-rata pembacaan yang dilakukan oleh DHT 1, 2, dan 3 sebesar 27,53°C, sehingga terjadi simpangan hasil pembacaan oleh DHT22 terhadap tang ampere dengan nilai rata-rata simpangan sebesar 0,47°C dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil data pengujian suhu DHT22 pada kondisi suhu ruangan

Hasil Pembacaan Suhu oleh 3 buah DHT22 pada kondisi suhu ruangan									
Pembacaan	suhu terbaca (°C)			Rata total suhu	Suhu terukur (°C)	Simpangan			rata simpangan suhu
	DHT 1	DHT 2	DHT 3			DHT 1	DHT 2	DHT 3	
Ke - 1	27,50	27,60	27,50	27,53	28	0,50	0,40	0,50	0,47
ke - 2	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 3	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 4	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 5	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 6	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 7	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 8	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 9	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47
Ke - 10	27,50	27,60	27,50	27,53		0,50	0,40	0,50	0,47



Gambar 4. 6 pembacaan suhu rata-rata

Pada pengujian yang kedua yaitu dengan melakukan pembacaan suhu rata-rata dari 3 buah DHT22 pada kondisi suhu ruangan, pada gambar 4.6 terdapat pembacaan suhu dari alat ukur tang ampere yaitu sebesar 27°C. Pada serial monitor terdapat pembacaan suhu rata-rata sebesar 27.41 °C sehingga terjadi simpangan hasil pembacaan DHT22 dengan nilai simpangan rata-rata simpangan sebesar 0.41°C, dapat dilihat pada tabel 4.3 dibawah ini.

Tabel 4.3 Pembacaan Suhu Rata-rata

Hasil Pembacaan Suhu Rata-rata 3 buah DHT22 pada kondisi Suhu Ruangan			
Pembacaan	Suhu Terbaca (°C)	Suhu terukur (°C)	Simpangan
ke - 1	27,40	27	0,40
ke - 2	27,40		0,40
ke - 3	27,43		0,43
ke - 4	27,40		0,40
ke - 5	27,40		0,40
ke - 6	27,43		0,43
ke - 7	27,40		0,40
ke - 8	27,43		0,43
ke - 9	27,40		0,40
ke - 10	27,40		0,40
<b>Rata-rata</b>	<b>27,41</b>		<b>0,41</b>

### **4.3 Pengujian Sistem Software Komunikasi ke Blynk**

Pada pengujian ini dilakukan dengan cara mengkoneksikan sistem yang telah dibuat dengan *platform* Blynk, dan pengujian ini terdapat beberapa percobaan, percobaan tersebut meliputi pengujian koneksi jaringan sistem dengan Blynk, waktu respon pengiriman data pembacaan suhu dan kelembapan pada sensor DHT22 yang ditampilkan pada serial monitor atau *display* LCD dan pada aplikasi Blynk, pembacaan suhu yang dikirimkan dari aplikasi Blynk ke *display* LCD dan waktu respon percobaan uji coba mulai *countdown* .

#### **4.3.1 Tujuan Pengujian**

Pengujian *software* dilakukan agar pemrogram bisa mengetahui kalau suatu sistem terbebas dari kesalahan. Tujuan dilakukannya pengujian *software* untuk memeriksa apakah seluruh komponen sudah berjalan sesuai dengan diagram alir yang ada pada gambar 3.5, dan dapat dikomunikasikan dari program komponen dengan proses pemrograman IoT, menggunakan Blynk sehingga data yang diperoleh dapat divisualisasikan dalam bentuk *gauge*, grafik, dan digit digital pada *platform* Blynk, serta dapat mengontrol dari aplikasi Blynk

#### **4.3.2 Langkah Pengujian**

Pengujian ini dilakukan dengan cara menyambungkan konektivitas antara *platform* Blynk dengan jaringan WiFi yang telah diatur pada aplikasi Blynk, serta memperhatikan visualisasi dalam bentuk *gauge*, grafik, dan angka digital dari data hasil pembacaan yang dilakukan oleh sistem inkubator tempe dan mengetahui sistem *controlling* berjalan atau tidak. Alat yang digunakan pada pengujian ini yaitu:

1. Smartphone
2. Perangkat keras komputer
3. Perangkat lunak Blynk dan aplikasi Blynk
4. Perangkat lunak Arduino IDE
5. Mikrokontroler ESP32
6. DHT32
7. Kabel *Jumper*

Langkah-langkah pengujian koneksi sistem Blynk dilakukan sebagai berikut:

1. Siapkan semua alat dan bahan yang akan digunakan
2. Rangkaian disusun seperti pada gambar
3. Siapkan Blynk *template ID* dan Blynk *template* nama dari *project room* Blynk yang digunakan.
4. Siapkan WiFi serta kata sandi yang akan digunakan
5. Siapkan *platform* Blynk pada Android dan atur *virtual pin* yang akan digunakan untuk melihat bentuk *gauge*, *digit digital*, dan grafik.
6. Blynk *template ID*, Blynk *template name* dimasukkan kedalam program pada perangkat lunak Arduino IDE.
7. Program di *upload* menuju mikrokontroler ESP32 menggunakan kabel mikro USB.
8. Buka *platform* Blynk dan perhatikan tampilan pada aplikasi Blynk di Android. Lalu *setting* WiFi dan kata sandi yang akan digunakan untuk menghubungkan *platform* Blynk dan ESP32, apabila ESP32 telah terhubung dengan WiFi maka akan terlihat perangkat yang terhubung dan *platform* Blynk akan terhubung dengan sistem.
9. Setelah itu uji pembacaan suhu dan kelembapan pada serial monitor ke Blynk lalu catat waktunya, suhu dan kelembapannya.
10. Setelah itu uji pembacaan pengaturan suhu dari aplikasi Blynk ke serial monitor lalu catat suhu dan kelembapannya.
11. Kemudian uji respon waktu terhadap perintah *start* pada *countdown* waktu.

### 4.3.3 Hasil Pengujian dan Analisa Koneksi Jaringan Sistem dengan Blynk

Pada percobaan pertama yaitu pengujian koneksi jaringan sistem dengan Blynk, pada Gambar 4.7.



Gambar 4. 7 Pengujian koneksi sistem dengan Blynk



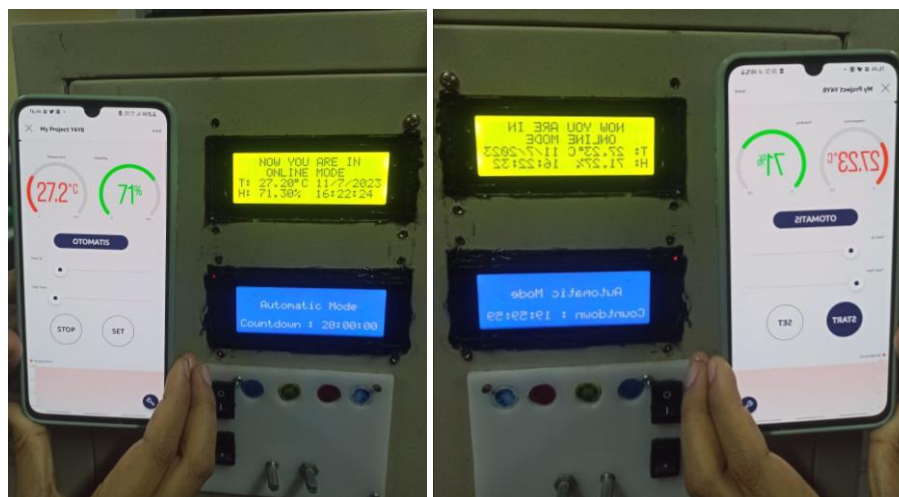


Gambar 4. 8 Hasil pengujian koneksi sistem dengan Blynk

Berdasarkan hasil pengujian, sistem berhasil terhubung dengan platform Blynk menggunakan WiFi sehingga hasil pembacaan data yang dilakukan oleh sensor DHT22 dapat divisualisasikan dalam bentuk *gauge*, digit digital, dan grafik seperti pada gambar 4.8.

#### 4.3.4 Hasil Pengujian dan Analisa pembacaan nilai suhu dan kelembapan pada serial monitor dan platform Blynk

Pada percobaan berikutnya yaitu pembacaan nilai suhu dan kelembapan pada serial monitor atau *display* LCD, yang ditampilkan pada platform Blynk. pada Gambar 4.9 pembacaan suhu dan kelembapan bahwa tampilan yang berada pada *display* LCD dan platform Blynk sama-sama menunjukkan angka nilai suhu sebesar 27.20°C dan nilai kelembapan sebesar 71.30% tetapi dalam platform Blynk menggunakan nilai integer sehingga mengalami pembulatan menjadi 71%.



Gambar 4. 9 Hasil pengujian pembacaan suhu dan kelembapan pada serial *display* dan Blynk

Percobaan dilakukan sebanyak 10 kali percobaan dengan waktu setiap detik, hal ini ditujukan untuk mengetahui perbandingan atau selisih waktu pengiriman data dari alat monitoring ke aplikasi Blynk secara *realtime*. Data hasil pengujian seperti pada tabel 4.4.

Tabel 4.4 Hasil perbandingan waktu yang terbaca di *display* dan aplikasi Blynk

hasil perbandingan waktu yang terbaca di <i>display</i> dan aplikasi Blynk					
percobaan ke	waktu yang terbaca pada <i>display</i> (WIB)	waktu yang terbaca pada Blynk (WIB)	Suhu dan kelembapan di <i>display</i>	Suhu dan kelembapan di Blynk	Durasi waktu
1	23:53:00.026	23:53:00.81	30,63 C 44,27%	30,63 C 44 %	0,784
2	23:53:01.167	23:53:02.07	30,63 C 44,27%	30,63 C 44 %	0,903
3	23:53:02.345	23:53:03.12	30,67 C 44,27%	30,67 C 44 %	0,775
4	23:53:03.572	23:53:04.32	30,67 C 44,27%	30,67 C 44 %	0,748
5	23:53:04.781	23:53:05.65	30,70 C 44,23%	30,70 C 44 %	0,869
6	23:53:05.974	23:53:06.84	30,70 C 44,23%	30,70 C 44 %	0,865
7	23:53:07.162	23:53:08.12	30,67 C 44,20%	30,67 C 44 %	0,985
8	23:53:08.377	23:53:09.37	30,67 C 44,20%	30,67 C 44 %	0,993
9	23:53:09.580	23:53:10.50	30,70 C 44,20%	30,70 C 44 %	0,920
10	23:53:10.768	23:53:11.58	30,70 C 44,20%	30,70 C 44 %	0,812
<b>Rata-Rata</b>					<b>0,8654</b>

Pada pengujian didapatkan rata-rata selisih waktu pembacaan data dari *display* ke aplikasi Blynk sebesar 0.8654 s. Hal ini menunjukkan bahwa

antara waktu yang terbaca pada alat dengan waktu yang dikirim pada aplikasi Blynk tergolong cepat. Tetapi cepat lambatnya data yang terkirim ke aplikasi Blynk sangat dipengaruhi oleh jaringan internet, apabila jaringan internet stabil maka data yang akan di terima oleh Blynk sama dengan waktu yang terbawa pada *display*, tetapi apabila jaringan internet kurang stabil maka data akan sedikit mengalami keterlambatan untuk mengirim data ke aplikasi Blynk.

#### 4.3.5 Hasil Pengujian Analisa pembacaan nilai suhu pada *platform* Blynk dan *display*

Selanjutnya percobaan uji coba pada proses mode manual yaitu dengan mengklik atau menekan tombol *switch* otomatis yang ada pada antarmuka aplikasi Blynk sehingga dapat memerintahkan menjadi mode manual lalu pengguna dapat mengatur *slider* suhu batas bawah dan *slider* suhu batas atas yang diinginkan.



Gambar 4. 10 Hasil Pengujian pembacaan nilai suhu pada *platform* Blynk dan *display*

Berdasarkan Gambar 4.10 merupakan hasil pengujian, pembacaan suhu yang diatur dengan menggunakan *slider* pada antarmuka aplikasi Blynk dan *display* pada LCD sama-sama, menunjukkan pada suhu pada *Temp Up* yang diatur oleh pengguna senilai 48°C dan *Temp Down* yang diatur oleh pengguna senilai 30 °C, maka pengujian ini berhasil mengirimkan data dari Blynk ke *display* LCD/ Serial Monitor.

#### 4.3.6 Hasil Pengujian dan Analisa waktu respon controlling mulai hitung mundur

Dan yang terakhir percobaan uji coba mulai hitung mundur dengan cara menekan atau mengklik sebuah *toggle* atau tombol *start* yang ada pada antarmuka pengguna Blynk. percobaan ini dilakukan sebanyak 10 kali yaitu dengan cara menggunakan kamera handphone yang ada vidio times hal ini ditujukan untuk mengetahui waktu respon atau selisih waktu perintah data dari aplikasi Blynk ke alat control. Data hasil pengujian seperti pada tabel 4.5.

Tabel 4.5 Hasil Pengujian waktu respon controlling mulai hitung mundur

No	Waktu Kirim	Waktu Respon Countdown	Durasi(s)
1	16:56:27.21	16:56:28.18	0,97
2	16:56:51.06	16:56:52.42	1,36
3	16:57:03.20	16:57:04.31	1,11
4	16:57:15.00	16:57:16.44	1,44
5	16:57:28.41	16:57:30.85	2,44
6	16:57:38.90	16:57:40.86	1,96
7	16:57:49.28	16:57:50.85	1,57
8	16:58:12.60	16:58:14.97	2,37
9	16:58:23.77	16:58:25.12	1,35
10	16:59:01.77	16:59:03.71	1,94
Rata-rata			2

Pada pengujian didapatkan rata-rata selisih waktu perintah data dari aplikasi Blynk ke alat kontrol sebesar 2 s. Hal ini menunjukkan bahwa antara waktu perintah pada aplikasi Blynk dengan waktu yang dikirim ke alat kontrol tergolong cepat Tetapi cepat lambatnya data yang terkirim ke aplikasi Blynk sangat dipengaruhi oleh jaringan internet, apabila jaringan internet stabil maka data yang akan di terima oleh Blynk sama dengan waktu yang terbawa pada *display*, tetapi apabila jaringan internet kurang stabil maka data akan sedikit mengalami keterlambatan untuk mengirim data ke aplikasi Blynk.

## 4.4 Pengujian Sistem Secara Keseluruhan

### 4.4.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui apakah sistem keseluruhan dapat membaca parameter suhu dan kelembapan pada inkubator tempe dan divisualisaikan pada *platform* Blynk dan mengontrol pada inkubator tempe dari *platform* Blynk.

### 4.4.2 Langkah Pengujian

Pengujian ini dilakukan dengan memasang komponen yang telah disusun menjadi sebuah pemanfaatan sistem IoT berbasis ESP32 untuk *remote control* pada inkubator tempe. Alat dan bahan yang akan dibutuhkan pada pengujian ini yaitu:

1. Inkubator Tempe
2. Android atau perangkat keras komputer yang di lengkapi perangkat lunak Blynk.
3. Jaringan Internet

Langkah-langkah pengujian sistem secara keseluruhan adalah sebagai berikut:

1. Siapkan alat dan bahan yang akan digunakan untuk pengujian ini
2. Hubungkan inkubator tempe pada sumber tegangan.
3. Masukkan tempe yang sudah siap difermentasi kedalam inkubator tempe.
4. Hubungkan inkubator tempe dan gawai pada jaringan internet.
5. Buka *platform* Blynk untuk melihat *gauge*, digit angka digital, dan grafik dari pembacaan sistem.
6. Lakukan pengujian selama 20 jam

### 4.4.3 Hasil Pengujian dan Analisa Sistem Secara Keseluruhan

Data hasil pengujian sistem secara keseluruhan pada alat inkubator tempe menggunakan sebanyak 10 buah dan dimonitoring secara berkala setiap satu jam berdasarkan suhu dan kelembapannya, maka dapat dilihat hasil secara keseluruhan pada tabel 4.6.

Tabel 4.6 Hasil pengujian inkubator tempe selama 20 jam

No	Waktu (WIB)	Suhu dan Kelembapan (Blynk)	
		Suhu (°C)	Kelembapan (%)
1	20.41	30,30	50
2	21.41	32,50	44
3	22.41	32,80	42
4	23.41	32,23	42
5	00.41	32,46	39
6	01.41	31,94	39
7	02.41	32,47	37
8	03.41	32,50	36
9	04.41	32,57	36
10	05.41	32,38	35
11	06.41	32,18	35
12	07.41	31,90	35
13	08.41	30,71	36
14	09.41	30,57	37
15	10.41	31,77	35
16	11.41	32,10	35
17	12.41	32,12	35
18	13.41	31,57	36
19	14.41	31,41	36
20	15.41	31,12	37
Rata-rata		31,88	37,85

Pada pengujian didapatkan hasil rata-rata suhu sebesar 31,88 °C dan kelembapan 37,85 %, suhu yang terjaga selama proses fermentasi tempe yaitu 31 °C-33 °C dengan diatur *countdown* selama 20 jam. Pengujian dilakukan pada jam 20.41 WIB dan selesai pada jam 15.41 WIB.

Tabel 4.7 Pengujian sistem inkubator tempe selama 20 jam

Pengujian	Kondisi			
	Koneksi	<i>Countdown</i>	Buzzer	Sistem
07/15/23 8:41:48 PM	Online	On	Off	On
07/16/23 4:32:06 AM	Offline	On	Off	On
07/16/23 4:32:06 AM	Online	On	Off	On
07/16/23 3:42:43 PM	Offline	Off	On	Off

Data hasil pengujian yang ditunjukkan pada Tabel 4.7 Pengujian sistem inkubator tempe selama 20 jam terkoneksi atau terhubung dengan Blynk dan WiFi,

dan *Countdown* berjalan sesuai dengan waktu yang telah di setting, setelah *countdown* selesai kondisi buzzer menyala dan sistem mati.



Gambar 4.11 Hasil fermentasi tempe pada inkubator tempe

Gambar 4.11 Dari pengujian keseluruhan menggunakan tempe yang dilakukan, hasil tempe yang difermentasi dengan menggunakan inkubator dengan inkubator tempe dibandingkan dengan tempe konvensional. Dengan menjaga kestabilan suhu selama proses fermentasi tempe, inkubator dapat menghemat 4 jam lebih cepat dibandingkan dengan proses pembuatan tempe secara konvensional. Inkubator tempe dengan memanfaatkan sistem IoT dapat memudahkan para pengrajin tempe untuk memonitoring dan mengontrol dari jarak jauh. Dan alat inkubator ini juga dapat menjaga nilai higienis.

## **BAB V**

### **PENUTUP**

berdasarkan hasil pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa:

1. ESP32 yang digunakan sebagai mikrokontroler pada pemanfaatan sistem IoT berbasis ESP32 untuk *remote control* pada inkubator tempe ini dapat berfungsi dan bekerja dengan baik dengan memperoleh tegangan pada kondisi *high* sebesar 2,69 volt dan tegangan pada kondisi *low* sebesar 0 volt.
2. Sensor DHT dapat mengirimkan data berupa *temperature* dan *humidity*. Rata-rata pembacaan yang dilakukan oleh DHT 1, 2, dan 3 sebesar 27,53 °C, sehingga terjadi simpangan hasil pembacaan oleh DHT22 terhadap tang ampere dengan nilai rata-rata simpangan sebesar 0,47 °C, maka menunjukkan bahwa DHT22 dapat berfungsi dengan baik, untuk inkubator tempe.
3. *Platform* Blynk dapat terkoneksi dengan Blynk dengan terkoneksi WiFi dan dapat memonitoring dengan menampilkan pembacaan data dalam bentuk *gauge*, digit angka digital, dan grafik, serta Blynk dapat mengatur suhu batas atas dan suhu batas bawah, serta dapat mengontrol mulainya *countdown*.
4. Pada pengiriman data rata-rata selisih waktu pembacaan data dari *display* ke aplikasi Blynk sebesar 0.8654 s. rata-rata selisih waktu perintah data dari aplikasi Blynk ke alat kontrol sebesar 2 s. Hal ini menunjukkan bahwa waktu pengiriman dan waktu perintah dapat dikatakan cepat. Tetapi cepat lambatnya data yang terkirim ke aplikasi Blynk sangat dipengaruhi oleh jaringan internet.
5. Pengujian keseluruhan berhasil dilakukan dengan hasil Pengujian sistem inkubator tempe selama 20 jam terkoneksi atau terhubung dengan Blynk dan WiFi, dan *Countdown* berjalan sesuai dengan waktu yang telah di setting, setelah *countdown* selesai kondisi buzzer menyala dan sistem mati.



## DAFTAR PUSTAKA

- Alvina, A., & Hamdani, D. (2019). PROSES PEMBUATAN TEMPE TRADISIONAL. *Jurnal Pangan Halal*, 9-12.
- Efendi, Y. (2018). *INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU MENGGUNAKAN RASPBERRY PI BERBASIS MOBILE*. *Jurnal Ilmiah Ilmu Komputer*, 19-26.
- Hariri, R., Novianta, M., & Kristiyana, S. (2019). PERANCANGAN APLIKASI BLYNKUNTUK MONITORINGDAN KENDALI PENYIRAMAAN TANAMAN. *Jurnal Elektrikal*, 1-10.
- Intern, D. (2020, September 16). *Apa itu Database? Contoh Produk dan Fungsinya*. Retrieved from Dicoding: <https://www.dicoding.com/blog/apa-itu-database/>
- Perdana Kristyabudi, H. N. (2016). *SISTEM KENDALI REMOTE CONTROL MINI-BLIMP MENGGUNAKAN ANDROID SMARTPHONE DENGAN KOMUNIKASI BLUETOOTH BERBASIS MIKROKONTROLER. SKRIPSI*.
- PUSIDO. (2012). *Tempe: Persembahan Indonesia untuk Dunia. Badan Standardisasi Nasional (p. ii+17)*. Jakarta: BSN.
- Suknia, S. L., & Ducha Rahmani, T. P. (2020). PROSES PEMBUATAN TEMPE HOME INDUSTRY BERBAHAN DASAR KEDELAI (*Glycine max (L.) Merr*) DAN KACANG MERAH (*Phaseolus vulgaris L.*) DI CANDIWESI, SALATIGA . *Southeast Asian Journal of Islamic Education*, Volume 03, No. 01.
- Sunarto Putra, G. F. (2022). *IMPLEMENTASI IOT(INTERNET OF THINGS) PADA PROTOTYPE INKUBATOR FERMENTASI TEMPE*. Tangerang Selatan: Skripsi.
- Suriana, I., Setiawan, I. A., & Graha, I. S. (2021). Rancang Bangun Sistem Pengaman Kotak Dana Punia berbasis Mikrokontroler NodeMCU ESP32 dan Aplikasi Telegram. *Jurnal Ilmiah TELSINAS*, Volume 4, No. 2 11-20.