

MANUAL BOOK
“SISTEM MONITORING DAN KONTROL PADA SERIKULTUR DENGAN IOT
BERBASIS TELEGRAM BOT”

Projek Akhir Praktikum Internet of Things



Disusun Oleh : Kelompok 3 / IOT C

Muhammad Abdullah

Filipus Adriel Manik

Muhammad Arif

Dani Aprilianto

2109106094

2109106096

2109106106

2109106092

Asisten :

Didi Nur Rahmad

2009106117

Alan Nuzulan

2009106032

Indro Dwi Saputro

2009106099

INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN

2024

DAFTAR ISI

DAFTAR ISI	2
A. LATAR BELAKANG SISTEM.....	3
B. FUNGSI SISTEM.....	4
C. KONSEP YANG DIGUNAKAN.....	4
D. BOARD SCHEMATIC	5
E. TAHAPAN PERANCANGAN SISTEM	5
1. Merangkai Komponen Elektronik	5
2. Persiapan Telegram BOT.....	6
3. Perancangan Program pada Arduino IDE.....	7
4. Pengujian Sistem	9

A. LATAR BELAKANG SISTEM

Pengendalian suhu dalam ruangan tertutup merupakan aspek krusial dalam berbagai aplikasi, seperti laboratorium, pusat data, gudang penyimpanan, dan lingkungan kerja yang memerlukan kondisi suhu tertentu. Suhu yang tidak terkontrol dapat menyebabkan kerusakan peralatan, degradasi bahan, dan bahkan risiko kesehatan bagi penghuni ruangan. Oleh karena itu, sistem monitoring dan kontrol suhu yang efisien sangat penting untuk memastikan stabilitas dan keamanan lingkungan tertutup.

Teknologi Internet of Things (IoT) telah berkembang pesat dan menawarkan solusi canggih untuk pemantauan dan pengendalian suhu secara real-time. IoT memungkinkan integrasi sensor yang dapat mengukur suhu secara terus-menerus dan mengirimkan data tersebut ke platform yang bisa diakses dari mana saja, memungkinkan pengawasan kondisi ruangan yang lebih akurat dan responsif terhadap perubahan.

Telegram Bot merupakan salah satu platform yang memungkinkan interaksi antara sistem dan pengguna melalui pesan instan. Dengan menggunakan Telegram Bot, sistem monitoring dan kontrol suhu dapat memberikan notifikasi otomatis kepada pengguna tentang perubahan suhu yang signifikan atau keadaan darurat. Selain itu, pengguna dapat mengirimkan perintah melalui Telegram untuk mengontrol perangkat pendingin atau pemanas dari jarak jauh.

Sistem monitoring dan kontrol suhu berbasis IoT dengan integrasi Telegram Bot menawarkan berbagai keuntungan, seperti pemantauan real-time, notifikasi instan, kontrol jarak jauh, dan efisiensi energi. Data suhu dapat dipantau secara terus-menerus dan real-time, sehingga setiap perubahan kondisi dapat segera diketahui. Pengguna mendapatkan notifikasi langsung melalui Telegram saat suhu mencapai ambang batas tertentu, memungkinkan tindakan cepat untuk mencegah masalah. Pengguna juga dapat mengontrol perangkat pendingin atau pemanas dari jarak jauh melalui perintah yang dikirimkan via Telegram, meningkatkan fleksibilitas dan responsivitas. Dengan pengaturan suhu yang lebih terkontrol, penggunaan energi untuk pendinginan atau pemanasan dapat dioptimalkan, mengurangi biaya operasional.

Implementasi sistem ini diharapkan dapat memberikan solusi yang lebih efektif dan efisien dalam pengendalian suhu ruangan tertutup. Dengan teknologi IoT dan Telegram Bot, pengawasan dan kontrol lingkungan menjadi lebih mudah dan cepat, serta mampu

meningkatkan produktivitas dan keamanan dalam berbagai aplikasi yang memerlukan kondisi suhu yang terkontrol.

B. FUNGSI SISTEM

1. Monitoring Suhu perangkat / ruangan tertutup dengan menggunakan DHT 11
2. Menampilkan kondisi suhu secara berkala di platform IOT
3. Monitoring suhu dari platform IOT ke dalam telegram bot
4. Menyalakan Kipas (servo) jika suhu mencapai kategori panas
5. Membunyikan buzzer jika suhu mencapai kategori panas
6. Menyalakan LED jika suhu mencapai kategori panas

C. KONSEP YANG DIGUNAKAN

1. Telegram Bot

Telegram Bot digunakan untuk membuat *user interface* pada sistem, dimana dalam telegram bot nanti user dapat memonitoring sistem serikultur dengan menggunakan bot telegram yang berbasis dengan chat, dimana menggunakan telegram akan menggunakan platform *bot father* untuk membuat 1 bot telegram untuk mengirimkan pesan perintah kepada setiap user didalam grup telegram, sehingga setiap user dalam grup telegram bisa memonitoring setiap perangkat IOT yang digunakan, dan juga bot akan mengirimkan pesan yang berisi perintah untuk memonitoring alat IOT yang digunakan. Dan nantinya diperlukan untuk API *bot father* supaya API nya bisa di baca oleh broker.

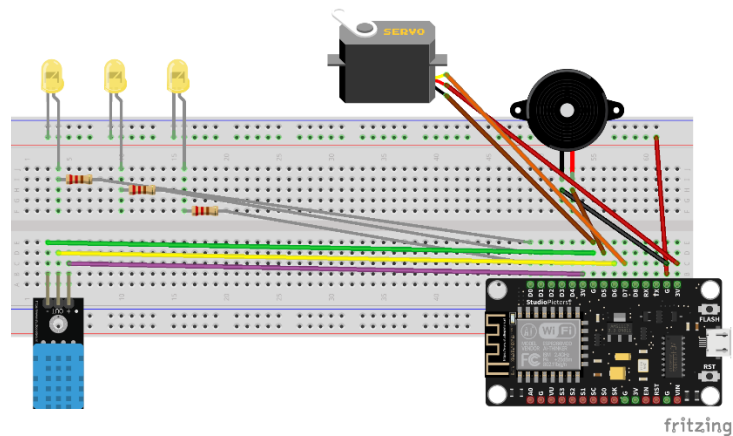
2. PythonFlask

Python Flask Berfungsi sebagai Broker atau penghubung antara Arduino dengan *user interface* yang dipakai. Dimana di dalam flask akan menggunakan method Get untuk mengambil data dari sensor yang dipakai dan menggunakan method Post untuk menampilkan data dari sensor yang tersimpan di dalam broker yaitu flask sendiri. Flask juga di jadikan sebagai server local untuk sistem serikultur, dimana semua proses pengambilan data dan menampilkan data akan digunakan dalam flask.

3. Arduino IDE

Penggunaan Arduino IDE sebagai platform untuk codingan dalam monitoring alat IOT yang digunakan, fungsinya juga juga untuk monitoring ESP32 dan juga monitoring suhu dari serial monitor Arduino sebelum dikirim ke dalam Flask, dan Arduino juga untuk monitoring API dari Flask dan Bot Telegram.

D. BOARD SCHEMATIC



Gambar 1 Board Schematic

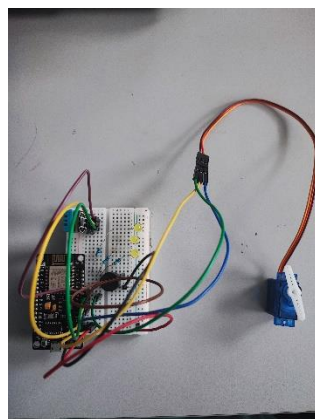
Komponen yang digunakan antara lain:

1. NodeMCU x 1
2. Kabel Jumper Male - Male x 10
3. LED x 3
4. Resistor x 3
5. Sensor suhu DHT11 x 1
6. Buzzer x1
7. Servo x1

E. TAHAPAN PERANCANGAN SISTEM

Berikut adalah cara merancang sistem monitoring serikultur berbasis IoT. Perancangan sistem terdiri dari tahap merangkai komponen elektronik, persiapan Telegram Bot, perancangan program Arduino, dan pengujian sistem.

1. Merangkai Komponen Elektronik

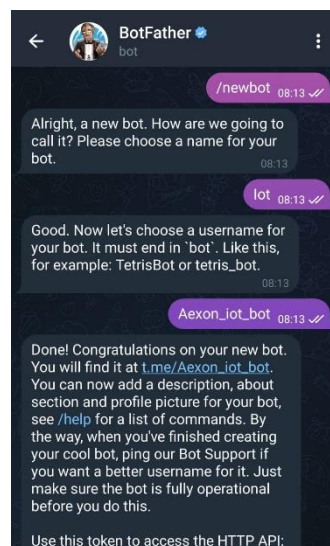


Gambar 2 Rangkaian Akhir

Rangkaian komponen elektronik seperti pada *board schematic* sebelumnya. Dimana rangkaian yang dipakai memakai komponen yang sudah di rangkai dalam *schematic* sebelumnya, 1 Buzzer digunakan untuk notifikasi suara, 3 LED untuk notifikasi cahaya kalau misalnya suhunya sudah sesuai dengan kategori DHT 11 untuk monitoring suhu dan juga 1 servo untuk menggerakkan kipas jika suhu dalam kondisi panas.

Perangkaian Komponen dimulai dengan menyambungkan 3 LED dengan ESP32 untuk menguji lampu dan juga resistor layak digunakan atau tidak, kemudian dilanjutkan dengan pemasangan buzzer dan juga Servo serta DHT 11 yang diuji pemakaian nya dengan menggunakan Arduino IDE.

2. Persiapan Telegram BOT



Gambar 3 Pembuatan BOT Telegram

Masuk ke telegram, kemudian cari channel Bot *Bot Father*. Kemudian, buatlah Bot anda dengan menggunakan menggunakan perintah `"/newbot"`, kemudian pilihlah nama untuk Bot anda, bebas namanya mau seperti apa, kemudian setelah itu akan diarahkan untuk membuat username dari Bot yang akan kita pakai, kemudian nanti akan di brekan token HTTP untuk mengakses BOT nya menggunakan platform IOT yang dapat diakses melalui arduino IDE.

Undang BOT kedalam Grup yang akan dipakai BOT nya, nanti ada keterangan perintah BOT untuk memulai BOT dengan memulai perintah `"/start"` setelah itu akan diakses dan BOT dapat digunakan di dalam Grup. Untuk pesan yang berisi list perintah akan dibuat dan di desain di alam platform Arduino IDE.

3. Perancangan Program pada Arduino IDE

Source code dapat diakses pada link dibawah.

BOT tele (py) = [pa-praktikum-iot-unmul-c3/bot_tele.py at main · Kuuhaku456/pa-praktikum-iot-unmul-c3 \(github.com\)](https://github.com/Kuuhaku456/pa-praktikum-iot-unmul-c3/blob/main/bot_tele.py)

bot di Arduino IDE = [pa-praktikum-iot-unmul-c3/mainku.ino at main · Kuuhaku456/pa-praktikum-iot-unmul-c3 \(github.com\)](https://github.com/Kuuhaku456/pa-praktikum-iot-unmul-c3/blob/main/mainku.ino)

Server = [pa-praktikum-iot-unmul-c3/server_arduino.py at main · Kuuhaku456/pa-praktikum-iot-unmul-c3 \(github.com\)](https://github.com/Kuuhaku456/pa-praktikum-iot-unmul-c3/blob/main/server_arduino.py)

Agar dapat menggunakan server flask maka di perlukan library flask dalam penggunaannya, untukm menginstal flask silahkan pergi ke command untuk menginstal flask, dengan menggunakan command

```
python -m pip install flask
```

```
PS C:\Users\Acer> python -m pip install flask
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=3.0.0 (from flask)
  Downloading werkzeug-3.0.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
  Requirement already satisfied: colorama in c:\users\acer\appdata\roaming\python\python310\site-packages (from click>=8.1.3->flask) (0.4.5)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-2.1.5-cp310-cp310-win_amd64.whl.metadata (3.1 kB)
Downloading flask-3.0.3-py3-none-any.whl (101 kB)
101.7/101.7 kB 216.8 kB/s eta 0:00:00
Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Downloading click-8.1.7-py3-none-any.whl (97 kB)
97.9/97.9 kB 562.2 kB/s eta 0:00:00
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading Jinja2-3.1.4-py3-none-any.whl (133 kB)
133.3/133.3 kB 84.6 kB/s eta 0:00:00
Downloading werkzeug-3.0.3-py3-none-any.whl (227 kB)
227.3/227.3 kB 97.1 kB/s eta 0:00:00
Downloading MarkupSafe-2.1.5-cp310-cp310-win_amd64.whl (17 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
```

Gambar 4 Instalasi Flask

```
Click here to ask Blackbox to help you code faster
1 from flask import Flask, request, jsonify
2 import json
3 app = Flask(__name__)
4
5
6
```

Import "flask" could not be resolved

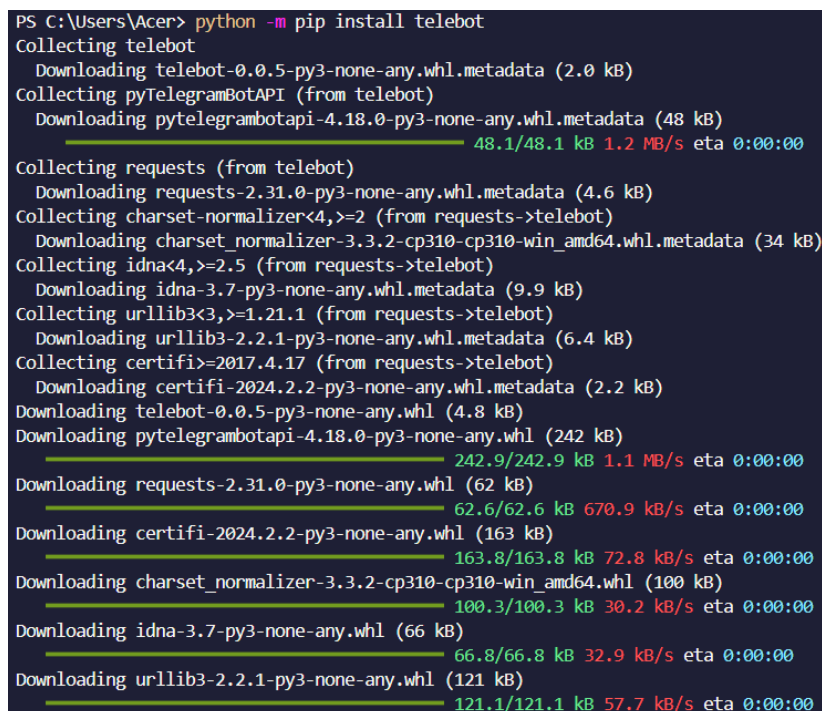
Gambar 5 Setup Flask

Setelah diinstal, diatas merupakan setup flask yang akan digunakan dalam rangkaian kali ini. Dimana sebelumnya akan mengimport modul Flask beserta komponen request dan jsonify. Modul Flask digunakan untuk membuat aplikasi web, request digunakan untuk

mengakses data dari permintaan HTTP, dan jsonify digunakan untuk mengonversi data Python menjadi format JSON. Kemudian import JSON untuk implementasi model JSON dan terakhir membuat sebuah instance dari kelas Flask dan menyimpannya dalam variabel app. Parameter __name__ digunakan untuk menentukan nama modul aplikasi yang dijalankan. Ini membantu Flask untuk mengetahui di mana letak aplikasi dan berbagai file sumber daya.

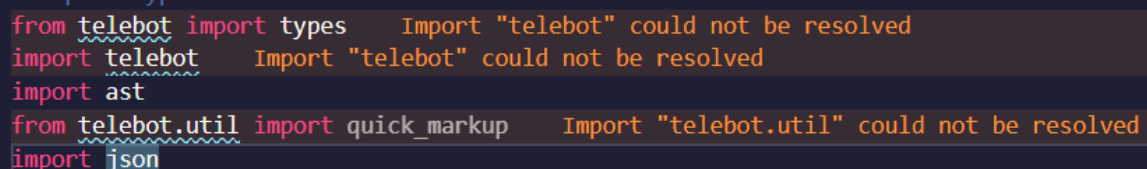
Kemudian ada penginstalan library telebot untuk mengimplementasikan Bot telegram dalam python. Hal ini dilakukan untuk membuat bot telegram dapat terkoneksi dengan broker flask dalam format python. Untuk menginstalnya cukup ke terminal dan memasukkan perintah dibawah ini

```
python -m pip install telebot
```



```
PS C:\Users\Acer> python -m pip install telebot
Collecting telebot
  Downloading telebot-0.0.5-py3-none-any.whl.metadata (2.0 kB)
Collecting pyTelegramBotAPI (from telebot)
  Downloading pytelegrambotapi-4.18.0-py3-none-any.whl.metadata (48 kB)
    48.1/48.1 kB 1.2 MB/s eta 0:00:00
Collecting requests (from telebot)
  Downloading requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)
Collecting charset-normalizer<4,>=2 (from requests->telebot)
  Downloading charset_normalizer-3.3.2-cp310-cp310-win_amd64.whl.metadata (34 kB)
Collecting idna<4,>=2.5 (from requests->telebot)
  Downloading idna-3.7-py3-none-any.whl.metadata (9.9 kB)
Collecting urllib3<3,>=1.21.1 (from requests->telebot)
  Downloading urllib3-2.2.1-py3-none-any.whl.metadata (6.4 kB)
Collecting certifi>=2017.4.17 (from requests->telebot)
  Downloading certifi-2024.2.2-py3-none-any.whl.metadata (2.2 kB)
Downloading telebot-0.0.5-py3-none-any.whl (4.8 kB)
Downloading pytelegrambotapi-4.18.0-py3-none-any.whl (242 kB)
    242.9/242.9 kB 1.1 MB/s eta 0:00:00
Downloading requests-2.31.0-py3-none-any.whl (62 kB)
    62.6/62.6 kB 670.9 kB/s eta 0:00:00
Downloading certifi-2024.2.2-py3-none-any.whl (163 kB)
    163.8/163.8 kB 72.8 kB/s eta 0:00:00
Downloading charset_normalizer-3.3.2-cp310-cp310-win_amd64.whl (100 kB)
    100.3/100.3 kB 30.2 kB/s eta 0:00:00
Downloading idna-3.7-py3-none-any.whl (66 kB)
    66.8/66.8 kB 32.9 kB/s eta 0:00:00
Downloading urllib3-2.2.1-py3-none-any.whl (121 kB)
    121.1/121.1 kB 57.7 kB/s eta 0:00:00
```

Gambar 6 instalasi telebot



```
from telebot import types      Import "telebot" could not be resolved
import telebot                 Import "telebot" could not be resolved
import ast
from telebot.util import quick_markup  Import "telebot.util" could not be resolved
import json
```

Gambar 7 setup telebot di python

Setelah diinstal, diatas adalah setup dari telebot yang akan digunakan dalam penyambungan telegram bot dengan broker flask yang digunakan nanti. Dimana di awali dengan mengimpor modul types dari pustaka telebot. Modul types berisi berbagai kelas yang digunakan untuk membuat dan mengelola elemen-elemen dalam bot Telegram, seperti tombol,

keyboard, dan pesan. Kemudian yang dilanjut dengan mengimpor modul ast (Abstract Syntax Trees). Modul ini digunakan untuk memproses dan mengelola struktur sintaksis abstrak dari kode Python. Dalam konteks bot, ini bisa digunakan untuk mengonversi string menjadi objek Python, yang mungkin diperlukan untuk memproses data yang diterima. Yang setelahnya, mengimpor fungsi `quick_markup` dari submodul `telebot.util`. Fungsi ini memudahkan pembuatan markup cepat untuk pesan bot, seperti inline keyboard, dengan format yang sederhana. Dan terakhir, mengimpor fungsi `quick_markup` dari submodul `telebot.util`. Fungsi ini memudahkan pembuatan markup cepat untuk pesan bot, seperti inline keyboard, dengan format yang sederhana.

4. Pengujian Sistem



Gambar 8 Monitoring sistem menggunakan platform telegram Bot

Setelah program di upload, pastikan hasil monitoring dapat dilihat pada grup yang sudah dimuat oleh telegram Bot dimana disini semua yang ada di grup dapat memonitoring suhu dan juga LED serta servo yang ada di dalam telegram bot dari pesan perintah yang disediakan oleh BOT.