

**SISTEM KONTROL DAN *MONITORING AIR CONDITIONER*
BERBASIS *INTERNET OF THINGS (IOT)***

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat
kelulusan Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Lara Septiasari

NPM : 0031815

Muhammad Fahriza Firdausy

NPM : 0031820

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2021**

LEMBAR PENGESAHAN

SISTEM KONTROL DAN *MONITORING AIR CONDITIONER* BERBASIS *INTERNET OF THINGS (IOT)*

Oleh :

Lara Septiasari / 0031815

Muhammad Fahriza Firdausy / 0031820

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

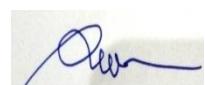
Menyetujui,

Pembimbing 1



Muhammad Iqbal Nugraha, M.Eng.

Pembimbing 2



Irwan, M.Sc., Ph.D.

Penguji 1



Surojo, M.T.

Penguji 2



Dr. Parulian Silalahi, M.Pd.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Lara Septiasari NIM : 0031815

Nama Mahasiswa 2 : Muhammad Fahriza Firdausy NIM : 0031820

Dengan judul : Sistem Kontrol dan *Monitoring Air Conditioner* Berbasis *Internet of Things* (IoT)

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 1 Agustus 2021

Nama Mahasiswa

Tanda Tangan

1. Lara Septiasari



2. Muhammad Fahriza Firdausy



ABSTRAK

Sistem monitoring suhu dan pengendali pendingin ruangan jarak jauh merupakan suatu alat yang dapat digunakan untuk memantau kondisi suhu ruangan dan mengendalikan suhu pada ruangan agar sesuai dengan yang kita inginkan. Alat ini menggunakan kontrol tipe proporsional (P) dengan sistem pemantauan memanfaatkan teknologi Internet of Things (IoT). NodeMCU digunakan sebagai perangkat pengendali berbasis mikrokontroler yang sudah difasilitasi dengan modul wifi. Untuk mengukur suhu ruangan, sensor DHT11 digunakan. Sebuah LED infrared dirangkai dan dihubungkan ke mikrokontroler sebagai pengirim data ke unit AC. Sistem ini menggunakan aplikasi android yaitu Blynk yang memiliki fitur untuk memonitoring dan mengendalikan suhu pada ruangan. Cara kerja alat ini yaitu sebagai remote yang mengendalikan Air Conditioner untuk mendapatkan suhu yang diinginkan pada ruangan dan suhu yang terbaca pada alat dapat dimonitoring pada smartphone dari jarak jauh. Pengujian respon alat terhadap nilai suhu dilakukan dengan nilai penguat Proportional yang berbeda. Nilai steady state error yang didapatkan yaitu kurang dari 1 yang menunjukkan nilai suhu pada ruangan cukup akurat dibandingkan dengan nilai suhu yang kita inginkan. Untuk nilai rise time dan settling time setiap pengujian memiliki nilai yang berbeda-beda karena nilai K_p yang berbeda yaitu 0.5, 1, dan 2. Berdasarkan hasil dari pengujian alat baik dalam menurunkan dan menaikkan suhu maka diperoleh nilai K_p = 2 yang lebih cepat mencapai rise time dan settling time. Kesimpulan akhirnya adalah penguat proposisional K_p cukup baik digunakan untuk mengendalikan pendingin ruangan.

Kata kunci – monitoring, kontrol, air conditioner, Internet of Things, otomatis

ABSTRACT

The temperature monitoring system and remote air conditioning controller is a tool that can be used to monitor the condition of the room temperature and control the temperature in the room to match what we want. This tool uses proportional type control (P) with a monitoring system utilizing Internet of Things (IoT) technology. NodeMCU is used as a microcontroller-based control device that has been facilitated with a wifi module. To measure the room temperature, the DHT11 sensor is used. An infrared LED is assembled and connected to the microcontroller as a data sender to the AC unit. This system uses an android application, namely Blynk which has features for monitoring and controlling the temperature in the room. The way this tool works is as a remote that controls the Air Conditioner to get the desired temperature in the room and the temperature that is read on the device can be monitored on a smartphone remotely. Testing the response of the tool to the temperature value is carried out with different Proportional amplifier values. The steady state error value obtained is less than 1 which indicates the temperature value in the room is quite accurate compared to the temperature value we want. For the value of rise time and settling time each test has a different value because of the different K_p values, namely 0.5, 1, and 2. time and settling time. The final conclusion is that the proportional amplifier K_p is used to control the air conditioner with a proportional.

Key words : Monitoring, Control, Air Conditioner, Internet of Things, Automatic

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Segala puji bagi Allah SWT yang telah memberikan kami kemudahan sehingga kami dapat menyelesaikan Laporan Proyek Akhir dengan judul “Sistem Kontrol dan Monitoring AC Ruangan Berbasis *Internet of Things*” ini dengan tepat waktu. Shalawat serta salam terlimpah curahkan kepada baginda tercinta Nabi Muhammad SAW yang dinanti-nantikan syafa’atnya di akhirat nanti.

Laporan ini merupakan salah satu syarat kelulusan bagi Program Studi D-III Teknik Elektronika di Politeknik Manufaktur Negeri Bangka Belitung. Kami menyadari bahwa laporan ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan di dalamnya. Untuk itu, kami mengharapkan kritik serta saran dari pembaca, agar laporan ini dapat menjadi laporan yang lebih baik lagi kedepannya.

Kami ingin mengucapkan terima kasih kepada pihak-pihak berikut yang telah membantu kami selama masa pembuatan alat dan proses penulisan laporan ini yaitu kepada:

1. Orang tua dan keluarga kami yang selalu memberikan doa dan dukungan selama mengerjakan proyek akhir dan dalam pengerajan makalah ini.
2. Bapak I Made Andik Setiawan, M.Eng, Ph.D. Selaku Direktur di Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak memberikan kemudahan bagi kami dalam menyelesaikan pendidikan.
3. Bapak Muhammad Iqbal Nugraha, M.Eng selaku Pembimbing I yang telah membimbing, mengarahkan dan memberi saran-saran dalam pengerajan proyek akhir dan dalam penyusunan laporan proyek akhir ini.
4. Bapak Irwan, M.Sc., Ph.D selaku Pembimbing II yang telah membimbing, mengarahkan dan memberi saran-saran dalam pembuatan dan penyusunan laporan proyek akhir ini.

5. Dosen dan Staf Pengajar di Politeknik Manufaktur Negeri Bangka Belitung yang telah mengajar dan membina penulis sehingga memiliki banyak pengetahuan dalam bidang elektronika.
6. Teman-teman dan pihak yang telah memberikan bantuannya.
7. Mitra kerja penulis selama mengerjakan proyek akhir ini yang selalu berjuang bersama hingga Proyek Akhir ini dapat diselesaikan dengan sebaik-baiknya.

Setelah melalui proses yang panjang dan penuh tantangan, akhirnya penulis dapat menyelesaikan pembuatan alat dan laporan proyek akhir ini yang tentunya masih banyak kekurangan dan jauh dari kesempurnaan. Walaupun demikian, penulis berharap laporan proyek akhir ini dapat bermanfaat bagi kita semua dan penulis khususnya.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Sungailiat, 6 Agustus 2021

Hormat kami,

Tim Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
<i>ABSTRAK</i>	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah	1
1.3 Tujuan Proyek Akhir	2
BAB II TEORI DASAR.....	3
2.1 Sistem Monitoring	3
2.2 Sensor Suhu DHT11	3
2.3 Internet of Things	4
2.4 Infra Merah.....	5
BAB III METODE PELAKSANAAN	8
3.1 Jenis Metode.....	8
3.2 Perancangan Rangkaian Kontrol	8
3.3 Pengujian Sistem <i>Monitoring</i>	9
3.4 Analisis Data	9
BAB IV PEMBAHASAN	10
4.1 Alat dan Bahan	10
4.1.1 Bahan Yang Digunakan	10
4.1.2 Alat Yang Digunakan	11
4.1.3 Aplikasi Yang Digunakan	11

4.2	Rancangan Alat	11
4.3	Flowchart Sistem	12
4.4	Rancangan Sensor Suhu DHT11	14
4.5	Rancangan <i>Infrared Receiver</i>	17
4.6	Rancangan Infrared Transmiter	19
4.7	Perancangan Keseluruhan Alat.....	21
4.8	Pembuatan Kode Pada Alat.....	21
4.9	Pengujian Sistem Monitoring	21
4.10	Pengujian Sistem Kontrol Alat	24
	4.10.1 Pengujian Alat Untuk Menurunkan Suhu	25
	4.10.1.1 Pengujian Alat Menurunkan suhu Dengan Nilai Setpoint 24 ..	25
	4.10.1.2 Pengujian Alat Menurunkan Suhu Dengan Nilai Setpoint 25 .	26
	4.10.1.3 Pengujian Alat Menurunkan Suhu Dengan Nilai Setpoint 26 .	27
	4.10.2 Pengujian Alat Untuk Menaikkan Suhu	29
	4.10.2.1 Pengujian Alat Menaikkan Suhu Dengan Nilai Setpoint 26....	29
	4.10.2.2 Pengujian Alat Menaikkan Suhu Dengan Nilai Setpoint 27	30
	4.10.2.3 Pengujian Alat Menaikkan Suhu Dengan Nilai Setpoint 28....	31
	BAB V	34
	PENUTUP	34
5.1	Kesimpulan.....	34
5.2	Saran	35
	DAFTAR PUSTAKA	36
	LAMPIRAN	38

DAFTAR TABEL

Table 4.1 Rangkaian Sensor Suhu DHT11 Dan Nodemcu	15
Tabel 4.2 Hasil Pengujian Sensor DHT11	16
Table 4.3 Sambungan Ir Receiver Dan Nodemcu	18
Tabel 4.4 Hasil Pengujian Alat Untuk Menurunkan Suhu.....	28
Tabel 4.5 Hasil Pengujian Alat Untuk Menaikkan Suhu	33

DAFTAR GAMBAR

Gambar 2.1 Sensor Suhu DHT11	4
Gambar 2.2 Nodemcu ESP8266.....	5
Gambar 2.3 Diagram Blok Pengendali Proporsional.....	6
Gambar 2.4 Grafik Respon Pengendali Terhadap Setpoint	6
Gambar 3.1 Blok Diagram	8
Gambar 4.1 Skema Alat	12
Gambar 4.2 Flowchart Sistem.....	13
Gambar 4.3 Rangkaian DHT11	14
Gambar 4.4 Hasil Percobaan DHT11	16
Gambar 4.5 Rangkaian Infrared Receiver.....	17
Gambar 4.6 Hasil Percobaan Infrared Receiver	18
Gambar 4.7 Gambar Rancangan IR Transmitter.....	20
Gambar 4.8 Alat Keseluruhan	21
Gambar 4.9 Menu Login Aplikasi Blynk.....	22
Gambar 4.10 Menu Pembuatan Proyek Baru	22
Gambar 4.11 Email Kode Authentic.....	23
Gambar 4.12 Tampilan Monitoring Pada Aplikasi Blynk	23
Gambar 4.13 Grafik Respon Alat Pada Setpoint 24	25
Gambar 4.14 Grafik Respon Alat Pada Setpoint 25	26
Gambar 4.15 Grafik Respon Alat Pada Setpoint 26	27
Gambar 4.16 Grafik Respon Alat Untuk Menaikkan Suhu Pada Setpoint 26	29
Gambar 4.17 Grafik Respon Alat Untuk Menaikkan Suhu Pada Setpoint 27	30
Gambar 4.18 Grafik Respon Alat Untuk Menaikkan Suhu Pada Setpoint 28	31

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup	39
Lampiran 2 Daftar Riwayat Hidup	40
Lampiran 3 Kode Program Keseluruhan.....	41
Lampiran 4 Kode Program Infrared Receiver	46

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Penggunaan pendingin ruangan atau *air conditioner* di Indonesia sangatlah banyak dikarenakan kondisi di Indonesia yang sangat panas. Hampir di setiap tempat kita akan menemukan pendingin ruangan, seperti di rumah-rumah, sekolah, perusahaan, pabrik, hotel dan banyak tempat yang lain. Salah satu contoh pemanfaatan pendingin ruangan yaitu pada ruang-ruang Anjungan Tunai Mandiri(ATM) yang setiap ruangan menggunakan pendingin ruangan. Pendingin ruangan pada setiap ruangan dibiarkan dalam kondisi selalu hidup. Disini kami mendapatkan permasalahan yaitu untuk mengendalikan pendingin ruangan dan untuk mengetahui suhu pada ruangan dalam kondisi yang ideal para pekerja harus mendatangi setiap ruangan.

Maka dari itu dibutuhkan monitoring dan juga kontrol suhu pada ruangan tersebut yang dapat dilakukan dari jarak jauh. Monitoring dan kontrol pada pendingin ruangan berguna untuk memudahkan tenaga pekerja untuk mengetahui kondisi suhu pada ruangan dan memudahkan tenaga pekerja untuk mengendalikan suhu pada suhu yang diinginkan dari jarak jauh menggunakan teknologi *Internet of Things*.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah diatas, maka dapat diketahui rumusan permasalahan yaitu sebagai berikut:

1. Bagaimana membuat alat monitoring dan kontrol Air Conditioner yang dapat dikendalikan melalui internet?
2. Bagaimana membuat alat dapat bekerja otomatis untuk mengendalikan suhu pada suatu ruangan?

1.3 Tujuan Proyek Akhir

Adapun tujuan dari proyek akhir ini yaitu sebagai berikut:

1. Membuat alat monitoring dan kontrol Air Conditioner yang dapat dikendalikan melalui internet.
2. Membuat alat dapat bekerja otomatis untuk mengendalikan suhu pada ruangan.

BAB II

TEORI DASAR

2.1 Sistem Monitoring

Saat ini semakin banyak pengaplikasian teknologi monitoring dalam kehidupan sehari-hari. Monitoring adalah proses mendapatkan informasi suatu indikator yang berjalan secara sistematis dan terus menerus agar dapat dilakukan koreksi terhadap indikator tersebut untuk menyempurnakan program atau informasi selanjutnya [1].

Monitoring akan memberikan informasi tentang status suatu pengukuran yang berulang dari waktu ke waktu, pemantauan biasanya dilakukan untuk suatu tujuan yaitu untuk memeriksa terhadap suatu proses atau mengevaluasi kondisi untuk mendapatkan hasil yang diinginkan dari tindakan yang dilakukan [1].

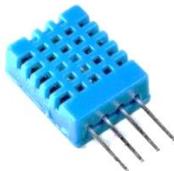
Monitoring suhu adalah salah satu teknologi yang saat ini banyak dimanfaatkan. Monitoring suhu merupakan pemantauan terhadap suhu pada suatu ruangan. Tujuannya yaitu mendapatkan informasi tentang kondisi suhu pada ruangan yang ingin dimonitoring berupa kenaikan atau penurunan suhu dalam ruangan tersebut.

2.2 Sensor Suhu DHT11

Sensor merupakan alat yang digunakan untuk mengukur nilai suatu parameter tertentu. Sensor memiliki banyak jenis tergantung apa yang ingin diukur misalkan suhu, kelembapan, cahaya, tekanan, jarak, kecepatan, suara dan lain lain. Sensor bekerja dengan mendeteksi perubahan pada sensor dengan kondisi aslinya.

Penggunaan sensor suhu sangat banyak jenisnya, semua tergantung keperluan. Pemilihan jenis sensor didasarkan kepada beberapa aspek seperti kualitas kerja sensor, kecepatan respon, akurasi pembacaan nilai, harga, jumlah dan lainnya. Sensor DHT11 merupakan satu dari banyak sensor suhu dengan

beberapa aspek yang baik, sangat mudah ditemukan, respon dan keakuratan yang baik, dan harga yang murah.



Gambar 2.1 Sensor suhu DHT11 [2]

Berikut merupakan kelebihan dari sensor DHT11 [3]:

- *Ultra low cost*
- *3 to 5V power and I/O*
- *2.5mA max current use during conversion (while requesting data)*
- *Good for 20-80% humidity readings with 5% accuracy*
- *Good for 0-50°C temperature readings ±2°C accuracy*
- *No more than 1 Hz sampling rate (once every second)*
- *Body size 15.5mm x 12mm x 5.5mm*
- *4 pins with 0.1 " spacing*

2.3 Internet of Things

Internet of things atau biasa kita ketahui dengan IoT merupakan sebuah konsep dimana sebuah alat atau objek dimungkinkan terhubung dengan alat dan objek lainnya melalui jaringan tanpa campur tangan manusia secara langsung. Pemanfaatan IoT saat ini merupakan trend yang hampir kita temui di semua alat-alat disekitar kita. Banyaknya pemanfaatan teknologi ini jelas memudahkan pekerjaan karena tanpa perlu kontak langsung dengan manusia dan karena tersambung dengan jaringan maka teknologi ini memungkinkan kita untuk mengaksesnya dari jarak jauh selama jaringan tidak terkendala.

Untuk dapat memanfaatkan teknologi ini diperlukan alat yang memungkinkan kita memproses data dan mengirimkan data tersebut ke jaringan

internet. Saat ini sudah banyak alat yang digunakan untuk mengaplikasikan IoT, salah satunya yaitu NodeMCU ESP8266. NodeMCU merupakan *platform* IoT yang bersifat *open source* buatan Espressif System. NodeMCU ESP8266 bekerja sebagai *board* Arduino yang juga terkoneksi dengan ESP8266 sehingga bekerja selayaknya mikrokontroler sekaligus dapat diakses ke jaringan. Berikut merupakan konstruksi dari NodeMCU ESP8266.

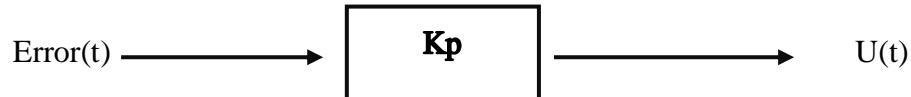
NodeMCU ESP8266



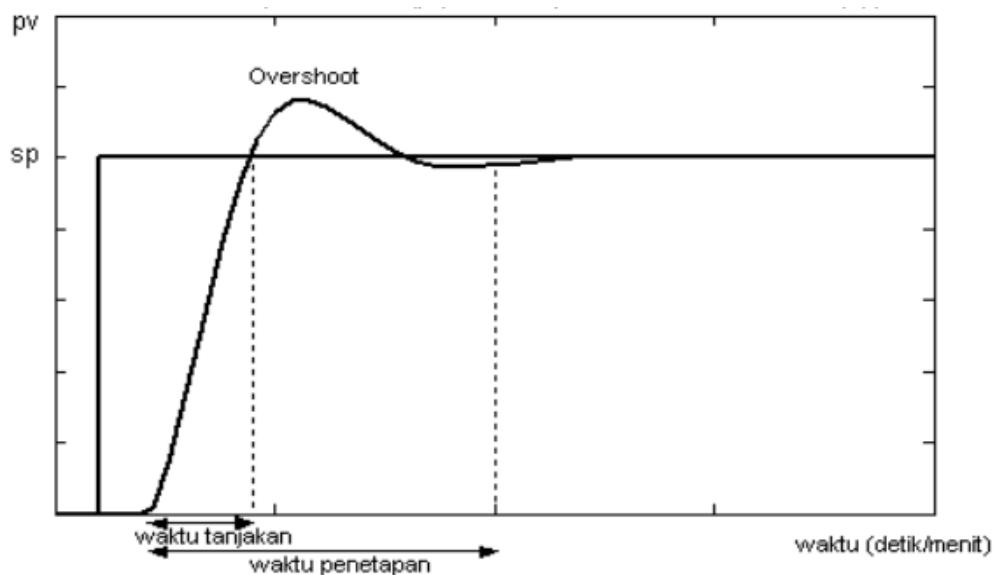
Dengan : K_p : Penguat proporsional
 e : *error*

error = suhu yang kita inginkan – suhu yang terbaca pada alat

Dan bentuk diagram blok dari pengendali Proporsional dapat dilihat pada gambar 2.3 seperti berikut.



Gambar 2.3 Diagram blok pengendali proporsional



Gambar 2.4 Grafik respon pengendali terhadap setpoint

Pada gambar 2.4 dapat dilihat bentuk grafik respon yang didapat terhadap nilai setpoint yang diinginkan. Terdapat pengaruh-pengaruh yang ditimbulkan selama pada proses pengendali seperti berikut.

Rise time : Waktu yang diperlukan respon untuk naik dari 0 sampai 90% harga akhirnya.

Overshoot : Lonjakan maksimum yang dialami oleh respon proses.

Settling time : Waktu yang diperlukan respon untuk mencapai dan menetap disekitar 95%-98% dari harga akhirnya.

Steady state error : Perbedaan nilai masukan dan keluaran

2.6 Software Blynk

Aplikasi Blynk merupakan *platform* IoT gratis untuk *mobile* yang memiliki fungsi untuk kendali *board* NodeMCU atau yang sejenisnya menggunakan internet. Aplikasi ini banyak digunakan dikarenakan gratis dan dengan mode *drag and drop* sehingga memudahkan penggunaannya dan tampilannya yang simple dan mudah untuk dipelajari dan digunakan.

BAB III

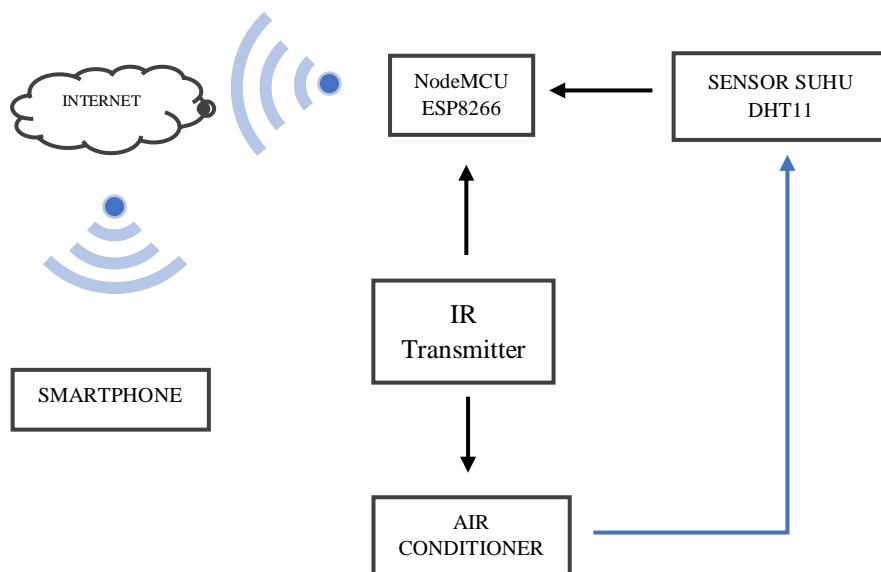
METODE PELAKSANAAN

3.1 Jenis Metode

Penulisan tugas akhir ini menggunakan metode terapan yang dimaksudkan agar alat yang dibuat memiliki bermanfaat dan berguna untuk banyak orang. Dengan judul Sistem Kontrol dan Monitoring *Air Conditioner* Berbasis *Internet of Things* ini, alat akan menontrol otomatis suhu pada ruangan tetap pada suhu yang ideal untuk ruangan, dengan menggunakan sensor suhu DHT11 sebagai pembacaan nilai suhu pada ruangan dan hasil pembacaan akan ditampilkan pada *smartphone* menggunakan aplikasi Blynk.

3.2 Perancangan Rangkaian Kontrol

Pada tahapan perancangan alat kontrol dan *monitoring Air Conditioner* yaitu dengan menentukan komponen alat dan bahan yang akan digunakan untuk pembuatan alat seperti sensor suhu DHT11, NodeMCU, Inframerah. Setelah menentukan komponen elektrik, kemudian membuat blok diagram *hardware*.



Gambar 3.1 Blok Diagram

Setelah perancangan blok diagram, selanjutnya dilakukan perakitan semua komponen elektrik sesuai dengan blok diagram pada gambar 3.1. Kemudian dilanjutkan pengujian pada alat untuk mengetahui kerja dari alat apakah berfungsi dengan baik. Pengujian dilakukan pada setiap komponen antara lain:

- Uji coba koneksi antara NodeMCU dan sensor suhu
- Uji coba koneksi antara NodeMCU dan *Infrared Receiver*
- Uji coba koneksi antara NodeMCU dan *Infrared Transmitter*

3.3 Pengujian Sistem *Monitoring*

Pengujian sistem *monitoring* dilakukan setelah proses perancangan dan perakitan pada alat selesai. Uji coba ini bertujuan untuk mengetahui apakah alat bekerja dan dapat menampilkan apa yang kita mau pada alat monitoring yaitu *smartphone*. Uji coba ini dilakukan pada aplikasi Android Blynk untuk memonitoring suhu pada ruangan.

3.4 Analisis Data

Analisis data merupakan tahap menganalisa data hasil uji coba monitoring suhu pada ruangan. Hal ini bertujuan untuk mengetahui fungsi dari alat apakah dapat mengontrol dengan baik dan mengetahui permasalahan atau kekurangan pada alat.

BAB IV

PEMBAHASAN

Pada Bab pembahasan ini menjelaskan proses pengerjaan proyek akhir sesuai metode pelaksanaan yang telah dijelaskan pada bab sebelumnya. Secara umum bab ini menguraikan tentang:

1. Alat dan bahan
2. Rancangan alat
3. Rancangan sensor suhu DHT11
4. Rancangan *infrared receiver*
5. Rancangan *infrared transmitter*
6. Perancangan keseluruhan alat
7. Pengujian sistem kontrol
8. Pengujian sistem monitoring

4.1 Alat dan Bahan

Untuk menyelesaikan proyek akhir ini diperlukan alat dan bahan untuk dapat membuat sistem tersebut. Alat dan bahan yang digunakan tergantung dengan kebutuhan. Dan dalam penyelesaian alat diperlukan aplikasi pada komputer untuk pembuatan kode program dan rangkaian skema alat.

4.1.1 Bahan yang digunakan

1. Sensor suhu DHT11
2. NodeMCU Esp8266
3. *Infrared transmitter*
4. *Infrared receiver*
5. Resistor
6. Transistor
7. Kabel konektor
8. *Board PCB*

4.1.2 Alat yang digunakan

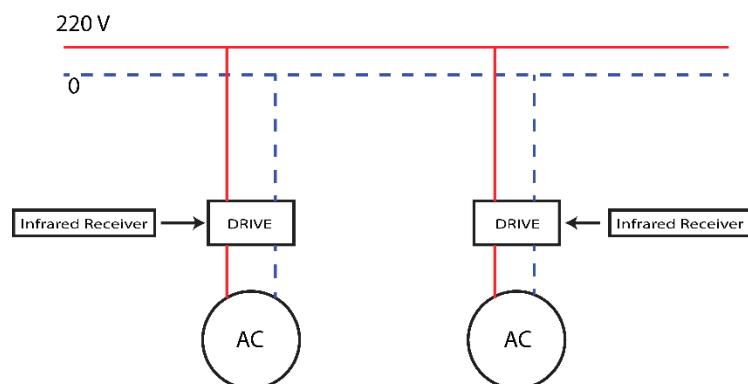
1. Solder
2. Tenol
3. Multimeter
4. Laptop

4.1.3 Aplikasi yang digunakan

1. Arduino IDE
2. Blynk
3. Fritzing

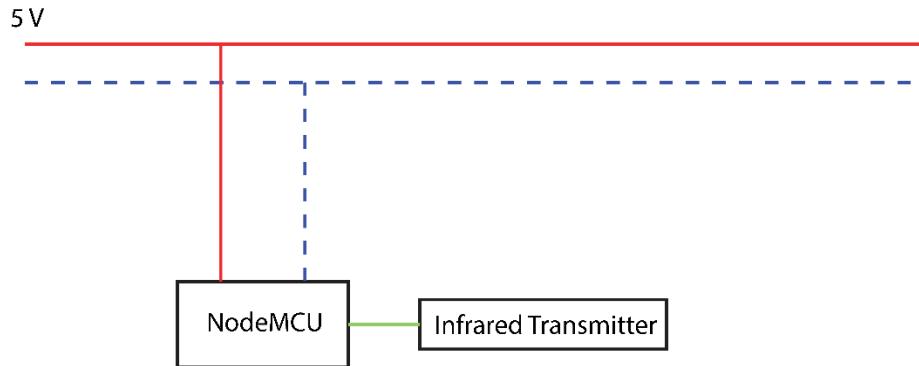
4.2 Rancangan Alat

Setelah mengetahui alat dan bahan yang akan digunakan untuk menyelesaikan alat. Berikutnya yaitu memulai rancangan awal berupa perancangan skema alat yang akan dibuat. Untuk pembuatan skema dapat menggunakan aplikasi yang mendukung untuk pembuatan skema elektronika dan disini aplikasi yang kami gunakan yaitu Fritzing pada laptop atau *computer*. Pada gambar 4.1 berikut merupakan gambar rangkaian kontrol pada *Air Conditioner*. Pada gambar terlihat pada *Air Conditioner* sudah terpasang *infrared receiver*.



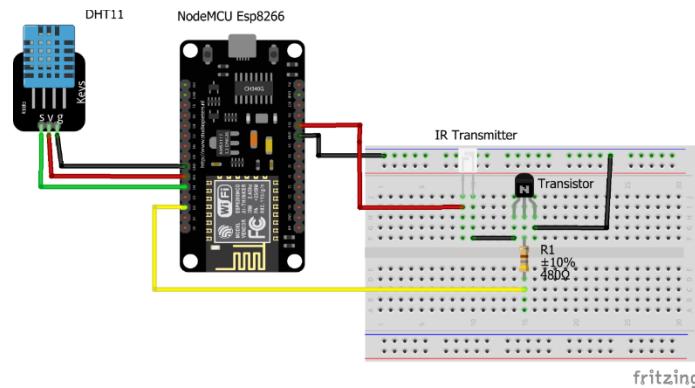
Gambar 4.1 Rangkaian kontrol pada Air Conditioner

Pada gambar 4.2 berikut merupakan gambar rangkaian kontrol pada alat yang akan dibuat. Pada alat akan menggunakan NodeMCU sebagai *board* mikrokontroler dan akan terpasang *infrared transmitter*.



Gambar 4.2 Rangkaian kontrol pada alat

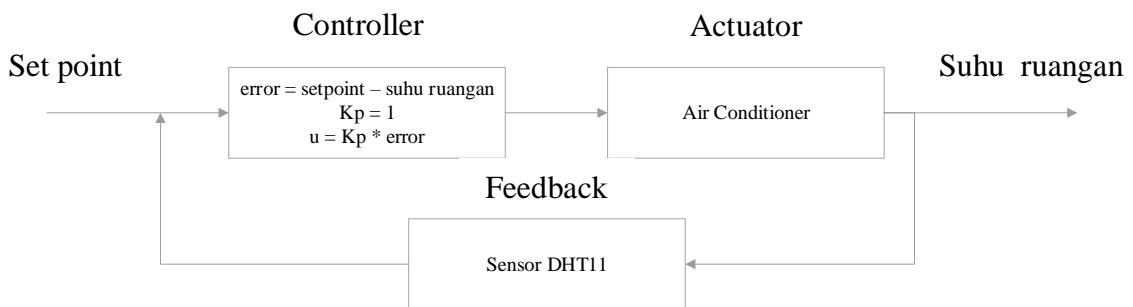
Setelah mengetahui rangkaian kontrol pada *Air conditioner* dan pada alat yang akan kita buat selanjutnya kita membuat skema alat seperti pada gambar 4.3 berikut.



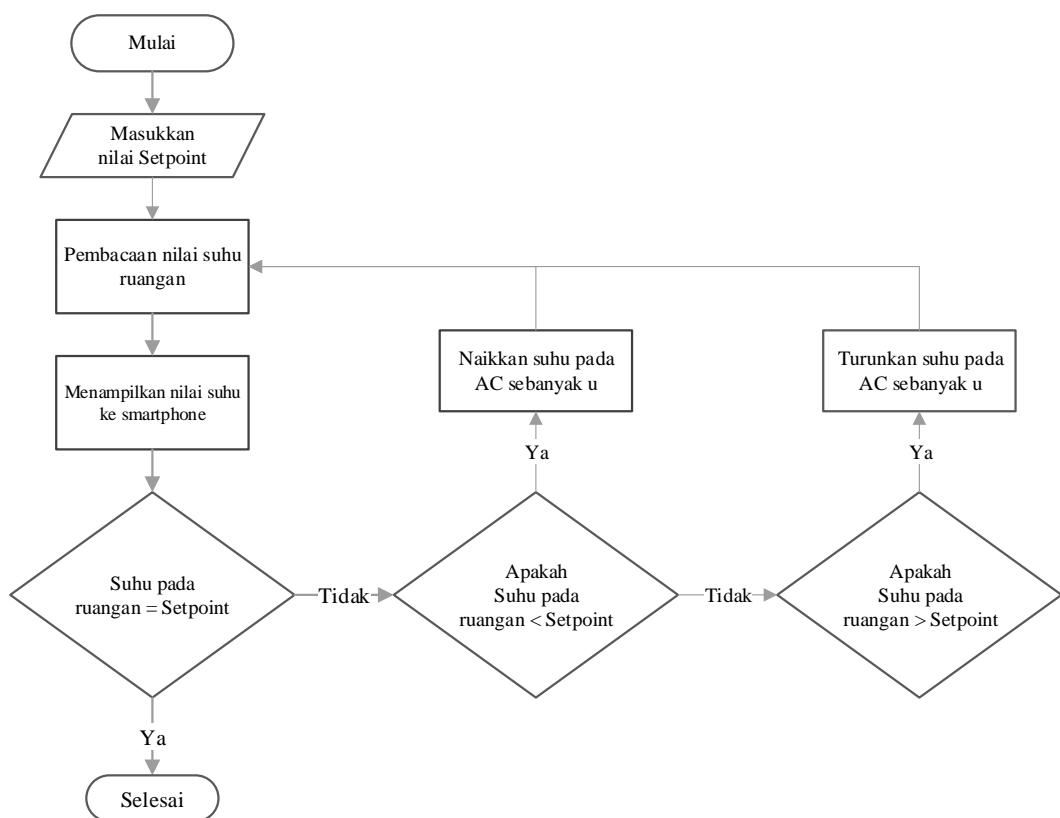
Gambar 4.3 Skema alat

4.3 Flowchart Sistem

Flowchart merupakan alur yang menggambarkan atau menjelaskan langkah-langkah penyelesaian suatu masalah. Tujuan penulisan *flowchart* agar memudahkan pembacaan urutan sistematis dalam penggerjaan alat agar dalam penggerjaannya lebih rapi dan teratur.



Gambar 4.4 blok diagram sistem kontrol



Gambar 4.5 Flowchart sistem kontrol

Penjelasan flowchart :

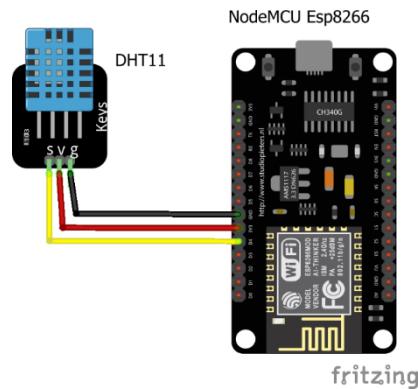
1. Mulai
2. Masukkan nilai setpoint atau suhu yang diinginkan.
3. Pembacaan nilai suhu pada ruangan oleh sensor suhu DHT11.
4. Menampilkan nilai pembacaan nilai suhu pada ruangan oleh sensor suhu DHT11 ke aplikasi Blynk di Smartphone.

5. Apakah suhu pada ruangan sama dengan suhu *setpoint* yang dimasukkan. Jika suhu sama maka sistem selesai.
6. Jika suhu pada ruangan lebih besar ($>$) dari nilai *setpoint* suhu yang dimasukkan pada *smartphone*. Jika iya maka alat akan mengirimkan data untuk menurunkan suhu pada *Air Conditioner* sebanyak u.
7. Jika suhu pada ruangan lebih kecil ($<$) dari suhu *setpoint* yang dimasukkan pada *smartphone* maka alat mengirimkan data untuk menaikkan suhu pada *Air Conditioner* sebanyak u.
8. Selesai

4.4 Rancangan Sensor Suhu DHT11

Sensor DHT11 merupakan sensor yang digunakan untuk mengukur nilai suhu pada suatu ruangan yang mana nilai *output* dari sensor berupa tegangan digital yang kemudian diolah pada *board* NodeMCU. Setelah mengetahui skema alat yang akan kita bangun selanjutnya merancang rangkaian sensor suhu DHT11 ke *board* NodeMCU untuk mengetahui apakah sensor berfungsi dengan baik.

Berdasarkan gambar 3.4. berikut merupakan rangkaian DHT11 ke *board* NodeMCU.



Gambar 4.6 Rangkaian DHT11

Pada gambar 4.2 merupakan rangkaian pengujian sensor suhu menggunakan sensor suhu DHT11. Untuk sambungan rangkaian dapat dilihat pada tabel skema rangkaian sensor suhu DHT11 berikut :

Table 4.1 Rangkaian sensor suhu DHT11 dan NodeMCU

Pin sensor suhu DHT11	Pin NodeMCU
Pin S	Pin D4
Pin V	Pin 3.3
Pin G	Pin g

Setelah perancangan sensor DHT11 ke NodeMCU selesai, selanjutnya pengujian sensor suhu. Pengujian dilakukan dengan menggunakan program yang pada NodeMCU. Penulisan kode program dilakukan pada aplikasi Arduino IDE dengan Kode program seperti berikut.

```
#include "DHT.h"
#define dht_pin D4
#define DHTTYPE DHT11

DHT dht(dht_pin, DHTTYPE);
int suhu;

void setup(){
Serial.begin(9600);
dht.begin();
}
void loop() {
temp = dht.readTemperature();
Serial.print("Suhu = ");
Serial.print(temp);
Serial.print(" C");
Serial.print("\n");
delay(2000);
}
```

Setelah rangkaian sensor suhu DHT11 dengan NodeMCU selesai dan kode program berhasil diupload pada *board* NodeMCU, selanjutnya melihat hasil pembacaan sensor suhu DHT11 pada serial monitor pada Arduino IDE. Dan hasil yang didapatkan dapat dilihat pada gambar 4.4 berikut.

The screenshot shows the Arduino Serial Monitor window titled 'COM6'. It displays a series of temperature readings from a DHT11 sensor. The data is timestamped and shows values fluctuating between 29.60°C and 29.70°C.

```

08:27:01.312 -> Suhu = 29.60 C
08:27:03.449 -> Suhu = 29.60 C
08:27:05.508 -> Suhu = 29.60 C
08:27:07.568 -> Suhu = 29.60 C
08:27:09.580 -> Suhu = 29.60 C
08:27:11.592 -> Suhu = 29.60 C
08:27:13.652 -> Suhu = 29.60 C
08:27:15.758 -> Suhu = 29.60 C
08:27:17.817 -> Suhu = 29.60 C
08:27:19.876 -> Suhu = 29.60 C
08:27:21.888 -> Suhu = 29.60 C
08:27:23.948 -> Suhu = 29.60 C
08:27:25.960 -> Suhu = 29.60 C
08:27:28.066 -> Suhu = 29.60 C
08:27:30.125 -> Suhu = 29.60 C
08:27:32.185 -> Suhu = 29.70 C
08:27:34.197 -> Suhu = 29.70 C
08:27:36.209 -> Suhu = 29.70 C
08:27:38.269 -> Suhu = 29.70 C

```

Gambar 4.7 Hasil percobaan DHT11

Jika serial monitor pada Arduino IDE berhasil menampilkan nilai dari sensor suhu DHT11 berikutnya pengujian kerja dari sensor suhu DHT11. Pengujian dilakukan untuk mengetahui keakuratan, respon dan error pembacaan sensor suhu DHT11. Pengujian dilakukan dengan membandingkan nilai suhu yang terbaca pada sensor DHT11 dengan nilai thermometer pada ruangan yang sama. Pengujian dilakukan pada ruangan dengan ukuran 4x2 meter dan pengujian dilakukan selama 30 menit pada jam 10.30 siang. Hasil dapat dilihat pada table berikut.

Tabel 4.2 Hasil pengujian sensor DHT11

Jam	Suhu thermometer(°C)	Sensor suhu DHT11(°C)	Selisih suhu (°C)	Presentase <i>error (%)</i>
10.30	26.3	26.5	0.2	0.7
10.35	26.2	26.3	0.1	0.3
10.40	26.2	26.3	0.1	0.3
10.45	26.3	26.4	0.1	0.3
10.50	26.3	26.3	0	0
10.55	26.3	26.3	0	0
11.00	26.3	26.4	0.1	0.3
Rata-rata				0.27

Pada tabel 4.2 merupakan tabel pengujian sensor DHT11 pada suatu ruangan. Pengujian dilakukan pada pukul 10.30 sampai pukul 11.00. dan didapatkan nilai suhu pada *thermometer* dan juga nilai suhu pada sensor suhu DHT11. Untuk mencari selisih suhu:

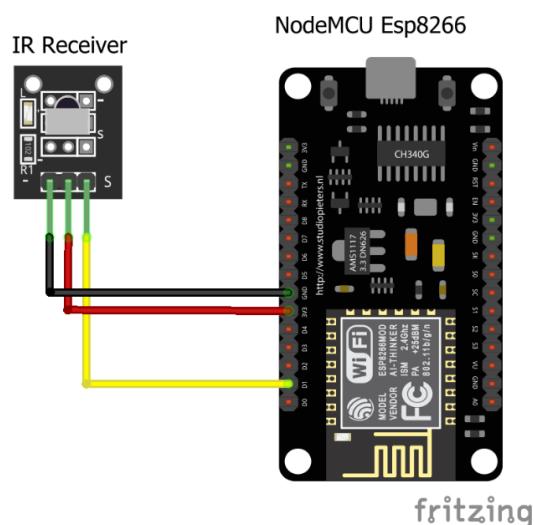
Selisih suhu = nilai sensor – nilai alat ukur(4.1)

Dan perhitungan persentase *error* saat pengujian sensor DHT11 didapatkan dengan rumus berikut

Berdasarkan nilai persentase *error* yang didapatkan dapat diketahui bahwa nilai *error* sensor DHT11 adalah 1.18 persen. Dan waktu respon dari sensor DHT11 selama pengujian memerlukan waktu 5-10 detik.

4.5 Rancangan *Infrared Receiver*

Infrared Receiver digunakan untuk mendapatkan kode pada *remote AC* yang akan kita kontrol dan monitoring. Dikarenakan setiap *remote* memiliki kode yang berbeda tergantung jenis dari AC tersebut, oleh karena itu satu *remote* tidak bisa digunakan untuk AC yang berbeda jenis dan kode tersebut yang membedakan AC jenis satu dengan jenis yang lainnya. Berikut ini adalah rangkaian *infrared Receiver* yang kita gunakan untuk mendapatkan kode tersebut.



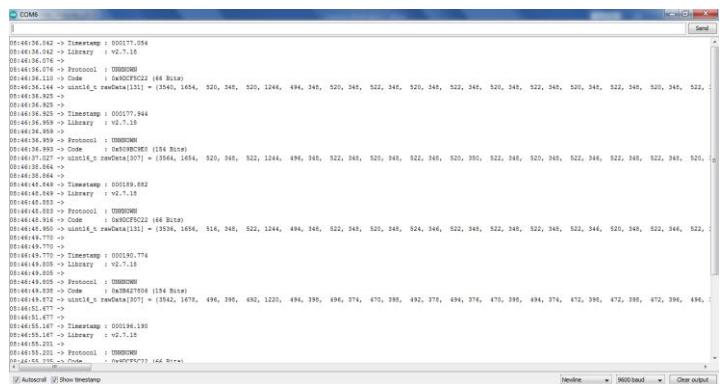
Gambar 4.8 Rangkaian Infrared Receiver

Dan untuk sambungan rangkaianya seperti pada table 4.2 berikut.

Table 4.3 Sambungan Ir Receiver dan NodeMCU

Pin IR Receiver	Pin NodeMCU
Pin V	Pin 3.3
Pin G	Pin G
Pin S	Pin D1

Setelah perangkaian alat yang akan dibuat selesai dirangkai sesuai dengan gambar 4.5 diatas selanjutnya penulisan kode pemrograman pada *board* NodeMCU pada Arduino IDE. Kode untuk *infrared receiver* dapat dilihat pada lampiran 4. Setelah rangkaian selesai dirangkai dan kode program berhasil diupload pada *board* NodeMCU selanjutnya adalah membuka serial monitor pada Arduino IDE, lalu arahkan *remote AC* yang ingin kita ambil kodennya. Lalu tekan tombol pada *remote*. Setiap tombol pada *remote AC* memiliki kode yang berbeda, digunakan untuk membedakan setiap tombol. Hasil dapat dilihat pada gambar 4.6 berikut.



Gambar 4.9 Hasil percobaan Infrared Receiver

Setelah mendapatkan kode dari tombol yang akan digunakan untuk kontrol otomatis alat, selanjutnya kode tadi di *copy* dan *paste* ke kode utama alat. Kode yang kita gunakan yaitu kode *rawData* yang bisa didapatkan saat merekam kode remote. Berikut ini merupakan hasil pengambilan kode pada *remote AC* merk Daikin. Data yang diambil merupakan data untuk tombol suhu 30 dan 29 yang mana setiap suhu memiliki data yang berbeda.

```

uint16_t dataSuhu30[137] = {9876, 9732, 9876, 9732, 4696, 2406,
466, 260, 468, 874, 414, 904, 412, 296, 440, 900, 414, 292,
466, 266, 440, 302, 442, 286, 464, 876, 412, 294, 466, 264,
444, 898, 412, 294, 442, 290, 442, 302, 466, 258, 444, 292,
444, 294, 442, 288, 444, 292, 466, 878, 438, 270, 466, 276,
468, 866, 440, 874, 438, 268, 468, 264, 468, 268, 466, 878,
438, 268, 468, 274, 468, 258, 470, 266, 444, 292, 468, 870,
440, 266, 446, 288, 468, 264, 468, 274, 446, 890, 440, 264,
468, 878, 440, 266, 468, 268, 468, 266, 446, 288, 468, 276,
470, 256, 470, 266, 468, 268, 444, 288, 468, 872, 440, 876,
438, 268, 446, 300, 468, 256, 470, 266, 470, 876, 416, 292,
468, 872, 438, 266, 468, 846, 462, 274, 442, 20278, 4694};

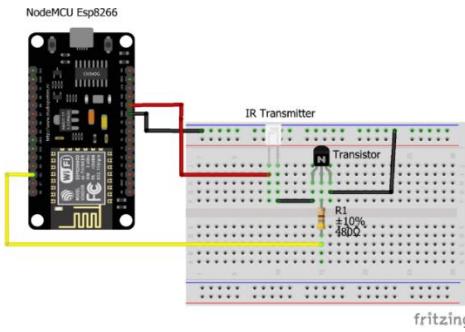
uint16_t dataSuhu29[137] = {9900, 9706, 9902, 9704, 4698, 2406,
466, 260, 468, 872, 440, 876, 440, 268, 466, 872, 440, 294,
440, 262, 470, 276, 468, 258, 468, 872, 440, 268, 468, 262,
470, 870, 442, 264, 468, 264, 468, 276, 466, 258, 470, 268,
468, 264, 472, 260, 472, 294, 442, 874, 440, 266, 470, 274,
468, 864, 440, 870, 440, 268, 468, 264, 468, 268, 466, 876,
440, 266, 470, 274, 470, 256, 472, 264, 470, 266, 466, 870,
442, 264, 472, 264, 470, 262, 470, 274, 468, 864, 442, 266,
470, 874, 442, 266, 468, 264, 470, 266, 468, 262, 470, 274,
462, 870, 440, 266, 470, 266, 470, 868, 442, 264, 470, 874,
418, 290, 446, 298, 468, 256, 446, 290, 468, 876, 440, 266,
468, 848, 464, 266, 468, 844, 466, 854, 464, 20252, 4670};

```

Dapat kita lihat kode diatas merupakan contoh kode hasil merekam dari *remote Air Conditioner DAIKIN*. Kode yang diambil yaitu berupa kode biner data suhu 30 dan data suhu 29. Kode diatas digunakan untuk mengendalikan *Air Conditioner* berada pada nilai 30 atau nilai lainnya sesuai dengan yang kita inginkan dengan mengirim kode melewati *infrared transmitter*.

4.6 Rancangan Infrared Transmiter

Infrared transmitter digunakan untuk mengirimkan data kode yang kita dapatkan dari *infrared receiver* dan telah diolah pada *board NodeMCU*, lalu kode dikirimkan ke AC untuk kita kontrol dan monitoring. Kerja dari *transmitter* yaitu dengan mengirimkan data melalui gelombang elektromagnetik yang dikeluarkan *infrared*. Untuk percobaan *infrared Transmitter* dapat dilakukan dengan rancangan seperti gambar 4.7 berikut.



Gambar 4.10 Gambar rancangan IR transmitter

Spesifikasi alat

1. NodeMCU Esp8266
2. IR Transmitter
3. Transistor NPN 2n2222
4. Resistor 330 ohm

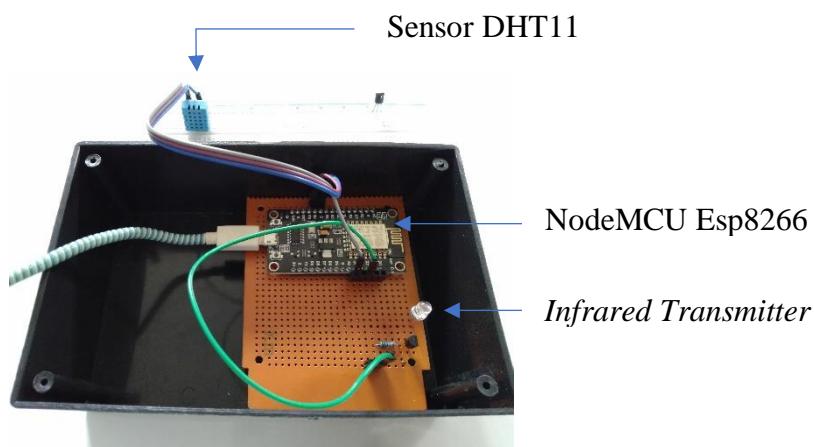
Pada spesifikasi diatas dapat dilihat bahwa pada rangkaian menggunakan transistor NPN dengan tipe 2n2222. Kegunaan transistor untuk penguatan sinyal dari *board* NodeMCU sehingga keluaran dari IR *transmitter* memiliki jangkauan *transmitter* lebih jauh. Untuk rangkaianya dapat dilihat pada penjelasan berikut.

Rangkaian:

1. Pin 3.3 *board* NodeMCU dihubungkan ke kutub (+) LED IR. Pin 3.3 berarti tegangan yang dihasilkan yaitu 3.3 volt. Tegangan tersebut cukup untuk menghidupkan LED *Infrared*.
2. Kutub (-) LED IR dihubungkan ke kaki *collector* (c) pada transistor. Arus pada kutub –, akan mengalir ke pin *collector* (c) pada transistor dan pada kaki *base* pada transistor disambungkan pada pin D2 NodeMCU sehingga tidak ada arus keluaran pada kaki *emitor* pada transistor jika tidak ada pemicu pada kaki *base*. Jika ada nilai pada pin D2 pada NodeMCU maka kaki *base* transisitor akan mengaliri arus menuju *ground* dan membuat IR *Transmitter* aktif.

4.7 Perancangan Keseluruhan Alat

Setelah selesai melalukan percobaan perancangan sensor suhu DHT11 dan IR *transmitter* selanjutnya menyatukan keseluruhan alat menjadi satu seperti skema gambar 4.1 sebelumnya. Selanjutnya alat disusun pada box berukuran 14.5 cm x 9.5 cm x 5 cm, pada alat juga ditambahkan adaptor dengan nilai keluaran 5 Volt. Sehingga perancangan keseluruhan alat dapat dilihat seperti pada gambar 4.8 berikut.



Gambar 4.11 Alat kontrol dan monitoring Air Conditioner berbasis IoT

4.8 Pembuatan Kode Pada Alat

Setelah selesai merancang alat hingga selesai berikutnya yaitu melakukan pembuatan kode pada *board* NodeMCU untuk membuat sistem kontrol dan monitoring pada alat. Kode dibuat pada aplikasi Arduino IDE seperti pada lampiran 3.

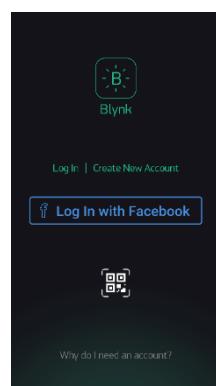
Setelah kode selesai dibuat selanjutnya kode di *compile* untuk mengetahui apakah masih terdapat *error* atau permasalahan pada kode pemrograman. Jika *compile* berhasil dan tidak terdapat permasalahan pada kode pemrograman selanjutnya mengupload kode ke *board* NodeMCU.

4.9 Pengujian Sistem Monitoring

Pengujian sistem monitoring dilakukan untuk mengetahui apakah alat dapat mengirimkan data ke *smartphone* dan pada *smartphone* data dapat ditampilkan.

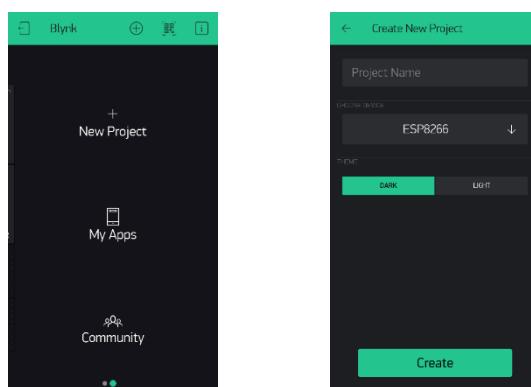
Aplikasi yang digunakan yaitu Blynk. Cara pengujian monitoring pada aplikasi Blynk seperti berikut ini.

1. Unduh aplikasi Blynk di *playstore* dengan klik Blynk di pencarian atau langsung ke website blynk.io. Setelah aplikasi terpasang pada smartphone buka aplikasi Blynk.
2. Lalu *login* Blynk dengan membuat akun baru atau menggunakan akun facebook anda.



Gambar 4.12 menu login aplikasi Blynk

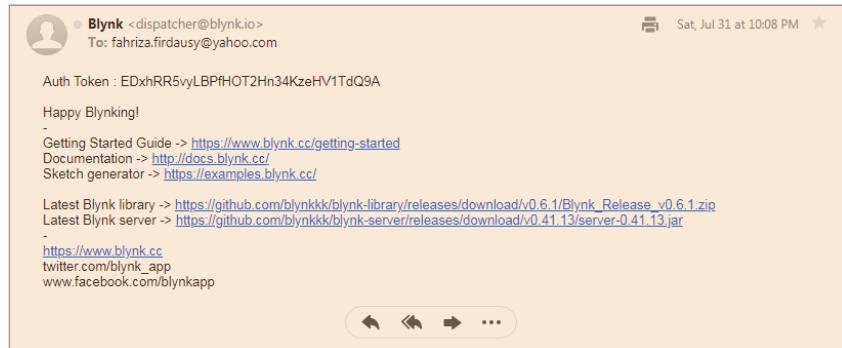
3. Setelah masuk ke aplikasi selanjutnya klik *new project*. Lalu masukkan nama proyek yang akan dibuat, pilih perangkat yang akan digunakan dan tema tampilan gelap atau terang. Jika telah selesai klik *Create*.



Gambar 4.13 Menu pembuatan proyek baru

4. Setelah anda mengklik *create* maka otomatis kode *authentic* dari Blynk akan terkirim ke email akun kita. Untuk mengetahui kode *authentic* bisa dilihat

dengan *login* ke email anda dan dapat dilihat bahwa ada email yang dikirimkan dari pihak Blynk.



Gambar 4.14 Email kode authentic

5. Selanjutnya pada aplikasi Blynk klik tambah untuk menambahkan *widget* atau alat yang akan kita gunakan. Penggunaan alat pada Blynk dibatasi untuk pengguna baru mendapatkan 2000 *battery* yang setiap alat memiliki nilai *battery* yang berbeda-beda. Apabila penggunaan *battery* telah habis maka kita perlu menambah dengan membayar minimal Rp 41000. *Widget* yang akan digunakan untuk sistem monitoring seperti pada gambar 4.12 berikut.



Gambar 4.15 Tampilan monitoring pada aplikasi Blynk

Setelah membuat tampilan monitoring di aplikasi Blynk pada *smartphone*, selanjutnya pengujian sistem monitoring alat. Selanjutnya pembuatan kode agar alat dapat melakukan monitoring. Kode seperti berikut ini.

```

#include <BlynkSimpleEsp8266.h>
#include <ESP8266WiFi.h>
#define BLYNK_PRINT Serial

BlynkTimer timer;
int setPoint
float temp;

char auth[] = "Authentic code";
char ssid[] = "USERNAME";
char pass[] = "PASSWORD"; } Kode authentic untuk menghubungkan board NodeMCU dengan Aplikasi

void temperature() {
    Blynk.virtualWrite(V0, temp); } Untuk menampilkan nilai pembacaan suhu DHT11 ke Blynk

BLYNK_WRITE(V2)
{
    setPoint = param.asInt(); } Untuk menerima nilai setpoint dari Blynk

void setup(){
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(400L, temperature);
}

void loop() {
    Blynk.run();
    timer.run();
}

```

4.10 Pengujian Sistem Kontrol Alat

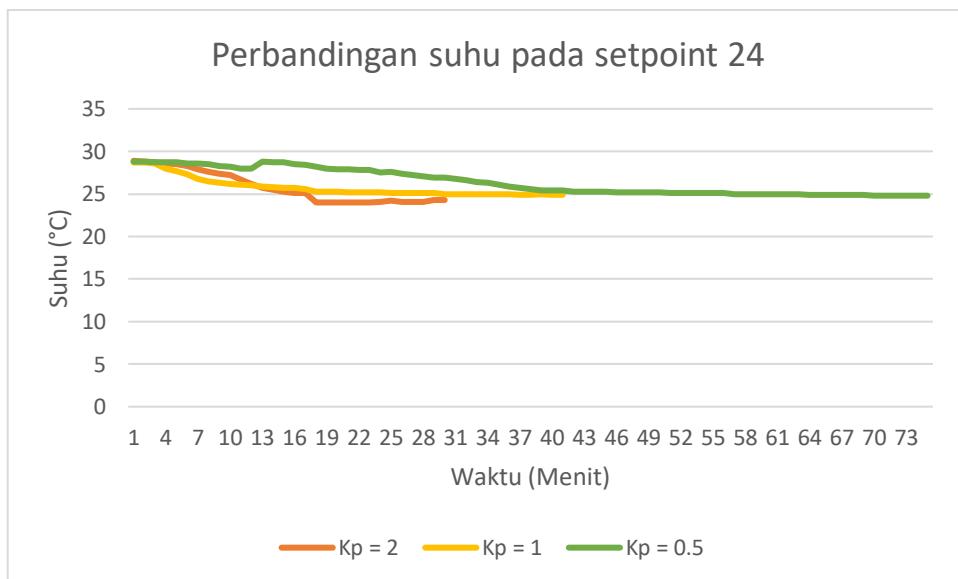
Pengujian sistem kontrol pada alat dilakukan untuk mengetahui kontrol pada alat berfungsi dengan baik. Untuk mengetahui kerja kontrol dapat dilihat pada grafik monitoring pada aplikasi Blynk. Pengendali yang digunakan yaitu Proporsional. Berikut ini pengujian alat dengan nilai *setpoint* yang berbeda. Nilai *setpoint* dapat diatur melalui smartphone dengan nilai yang diinginkan dan alat otomatis mengendalikan AC pada ruangan sehingga membuat suhu pada ruangan sama dengan nilai yang diinginkan. Pengujian dilakukan untuk mengetahui respon alat terhadap nilai *setpoint* dan berapa lama waktu yang diperlukan alat untuk mendapatkan suhu ruangan yang diinginkan.

4.10.1 Pengujian Alat Untuk Menurunkan Suhu

Pengujian alat untuk menurunkan suhu dilakukan untuk mengetahui apakah dapat mengendalikan pendingin ruangan untuk mendapatkan suhu yang kita inginkan pada ruangan.

4.10.1.1 Pengujian Alat menurunkan suhu Dengan Nilai Setpoint 24

Pengujian alat untuk menurunkan suhu dengan nilai suhu yang diinginkan pada ruangan/setpoint yaitu 24 dan nilai penguat (K_p) variari 0.5, 1 dan 2. Hasil percobaan dapat dilihat pada gambar 4.15 berikut.



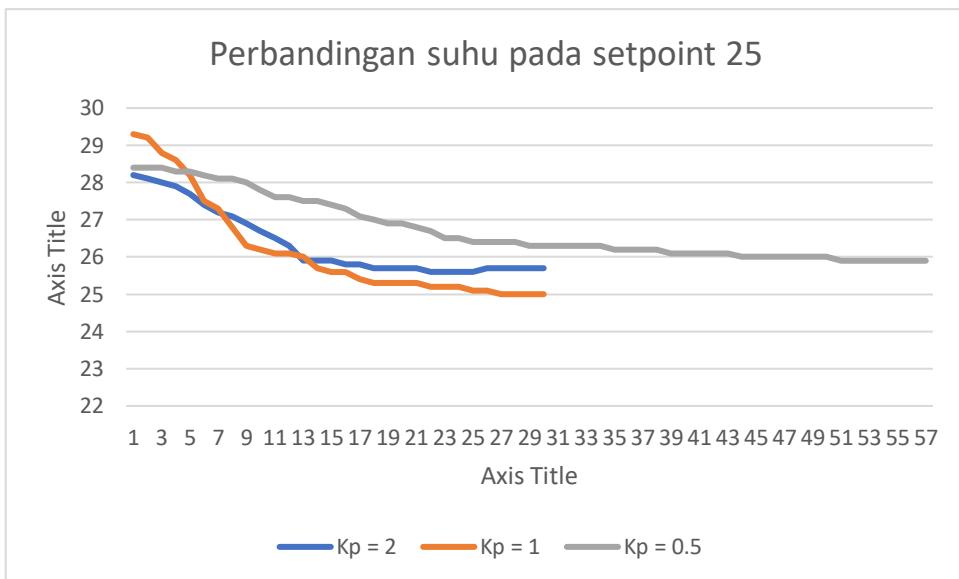
Gambar 4.16 grafik respon alat pada setpoint 24

Analisa : pengujian alat dilakukan pada setpoint 24 dan nilai penguat (K_p) = 0.5. pada gambar 4.15 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.7°C . Pada grafik dapat dilihat diperlukan 40 menit untuk mencapai rise time dan diperlukan 67 menit untuk mencapai settling time nilai setpoint 24 dan perbedaan masukan dan keluaran pada grafik yaitu $24 - 24.8 = -0.8$. Selanjutnya pengujian alat untuk nilai $K_p = 1$. Pada gambar 4.15 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.7°C . Pada grafik dapat dilihat diperlukan 18-20 menit untuk mencapai rise time dan diperlukan 25 menit untuk mencapai settling time dan perbedaan nilai masukan dan nilai keluaran yaitu $24 - 24.9 = -0.9$.

Terakhir pengujian alat dengan nilai penguat (K_p) = 2. pada gambar 4.15 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.9°C . Pada grafik dapat dilihat diperlukan 16 menit untuk mencapai rise time dan diperlukan 17 menit untuk mencapai settling time nilai setpoint 24 dan perbedaan nilai masukan dan nilai keluaran yaitu $24 - 24.3 = -0.3$.

4.10.1.2 Pengujian Alat Menurunkan Suhu Dengan Nilai Setpoint 25

Pengujian alat untuk menurunkan suhu dengan nilai suhu yang diinginkan pada ruangan/setpoint adalah 25 dan nilai penguat (K_p) variasi yaitu 0.5, 1 dan 2. Hasil percobaan dapat dilihat pada gambar 4.16 berikut.



Gambar 4.17 grafik respon alat pada setpoint 25

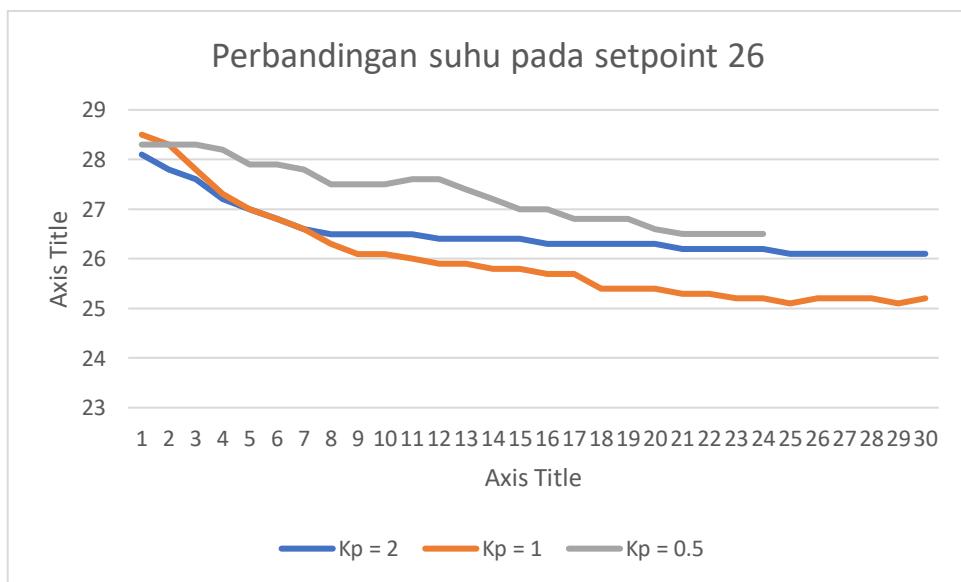
Analisa : pengujian alat dilakukan pada setpoint 25 dan nilai penguat (K_p) = 0.5. pada gambar 4.16 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.4°C . Pada grafik dapat dilihat diperlukan 23-25 menit untuk mencapai rise time dan diperlukan 40 menit untuk mencapai settling time nilai setpoint 25 dan perbedaan masukan dan keluaran yaitu $25 - 25.9 = -0.9$. Selanjutnya pengujian alat dilakukan dengan nilai penguat (K_p) = 1. pada gambar 4.16 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 29.3°C . Pada grafik dapat dilihat diperlukan 16-19 menit untuk mencapai rise time dan diperlukan 25

menit untuk mencapai settling time nilai setpoint 25 dan perbedaan nilai masukan dan nilai keluaran yaitu $25 - 25 = 0$.

Terakhir pengujian alat dilakukan dengan nilai penguat (K_p) = 2. pada gambar 4.16 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.1°C . Pada grafik dapat dilihat diperlukan 13 menit untuk mencapai rise time dan diperlukan 21 menit untuk mencapai settling time nilai setpoint 25 dan perbedaan nilai masukan dan nilai keluaran yaitu $25 - 25.7 = -0.7$.

4.10.1.3 Pengujian Alat Menurunkan Suhu Dengan Nilai Setpoint 26

Pengujian alat untuk menurunkan suhu dengan nilai suhu yang diinginkan pada ruangan/setpoint adalah 26 dan nilai penguat (K_p) variasi yaitu 0.5, 1 dan 2. Hasil percobaan dapat dilihat pada gambar 4.17 berikut.



Gambar 4.18 grafik respon alat pada setpoint 26

Analisa : pengujian alat dilakukan pada setpoint 26 dan nilai penguat (K_p) = 0.5. pada gambar 4.17 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.3°C . Pada grafik dapat dilihat diperlukan 13-15 menit untuk mencapai rise time dan diperlukan 21 menit untuk mencapai settling time nilai setpoint 26 dan perbedaan nilai masukan dan keluaran yaitu $26 - 26.5 = -0.5$. Selanjutnya pengujian alat dilakukan dengan nilai penguat (K_p) = 1. Pada gambar 4.17 dapat

dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.5°C . Pada grafik dapat dilihat diperlukan 5-7 menit untuk mencapai rise time dan diperlukan 9 menit untuk mencapai settling time nilai setpoint 26 dan perbedaan nilai masukan dan nilai keluaran yaitu $26 - 25.1 = 0.9$.

Terakhir pengujian alat dilakukan dengan nilai penguat (K_p) = 2. pada gambar 4.17 dapat dilihat bahwa pada awal pengujian suhu pada ruangan yaitu 28.3°C . Pada grafik dapat dilihat diperlukan 7-8 menit untuk mencapai rise time dan diperlukan 24 menit untuk mencapai settling time nilai setpoint 26 dan nilai steady-state error yaitu 0.1

Tabel 4.4 Hasil pengujian alat untuk menurunkan suhu

Nilai Setpoint	Suhu Awal ($^{\circ}\text{C}$)	K_p	Steady State Error	Rise Time (Menit)	Settling Time (Menit)
24	28.7	0.5	0.8	40	67
	28.7	1	0.9	18	25
	28.9	2	0.3	16	17
25	28.4	0.5	0.9	23	40
	29.3	1	0	16	25
	28.1	2	0.7	13	21
26	28.3	0.5	0.5	13	26
	28.5	1	0.9	7	24
	28.3	2	0.1	5	9

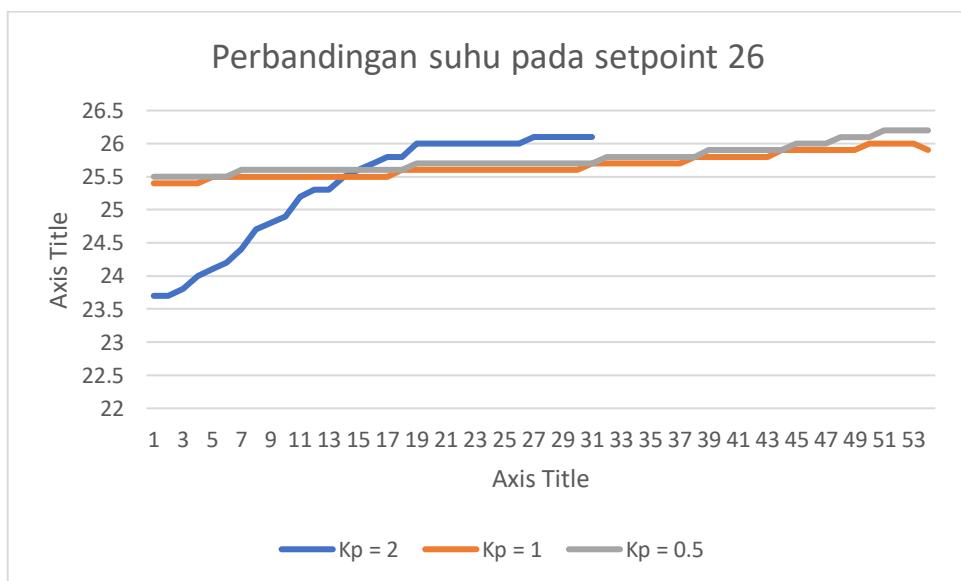
Pada tabel 4.4 diatas merupakan hasil pengujian alat untuk menurunkan suhu. . Berdasarkan hasil pengujian tabel diatas nilai $K_p = 2$ Lebih cepat mencapai rise time dan settling dibandingkan dengan nilai $K_p = 1$ atau $K_p = (0.5)$.

4.10.2 Pengujian Alat Untuk Menaikkan Suhu

Pengujian alat untuk menaikkan suhu dilakukan untuk mengetahui apakah alat dapat mengendalikan pendingin ruangan untuk menaikkan suhu pada ruangan. Pendingin ruangan biasa digunakan hanya untuk menurunkan suhu pada ruangan dan untuk menaikkan suhu ruangan dapat dilakukan dengan cara menaikkan nilai suhu pada pendingin ruangan atau dengan mematikan pendingin ruangan tersebut.

4.10.2.1 Pengujian Alat Menaikkan Suhu Dengan Nilai Setpoint 26

Pengujian alat untuk menaikkan suhu dengan nilai suhu yang diinginkan pada ruangan/setpoint yaitu 26 dan nilai penguat (K_p) variasi yaitu 0.5, 1 dan 2. Hasil pengujian dapat dilihat pada gambar 4.18 berikut.



Gambar 4.19 grafik respon alat untuk menaikkan suhu pada setpoint 26

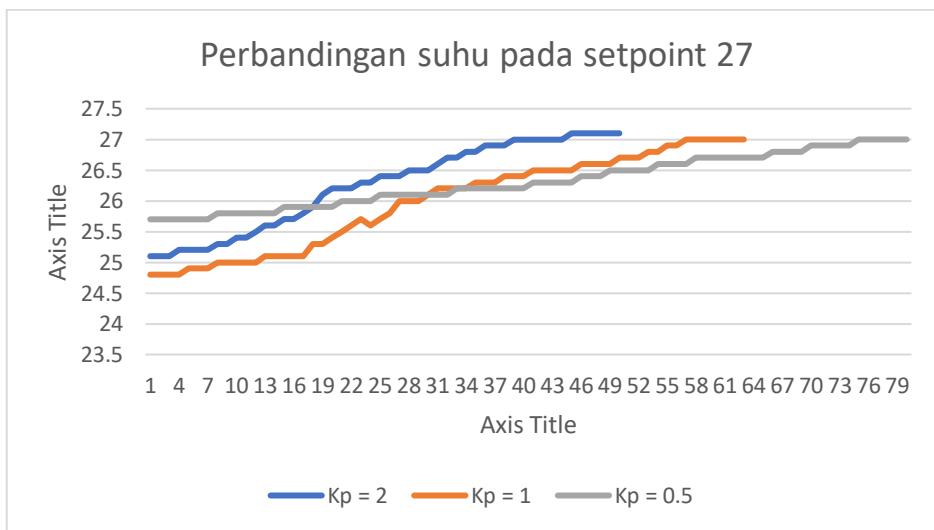
Analisa : pengujian alat dilakukan pada setpoint 26 dan nilai penguat (K_p) = 0.5. pada gambar 4.18 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 25.5°C . Pada grafik dapat dilihat untuk mencapai rise time diperlukan 44 menit. Untuk mencapai settling time diperlukan waktu 50 menit dan perbedaan nilai masukan dan nilai keluaran yaitu $26 - 26.2 = -0.2$. Selanjutnya pengujian alat dilakukan dengan nilai penguat (K_p) = 1. pada gambar 4.18 dapat dilihat

merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 25.4°C . Pada grafik dapat dilihat untuk mencapai rise time diperlukan 43-45 menit. Untuk mencapai nilai settling time diperlukan waktu 39 menit dan perbedaan nilai masukan dan nilai keluaran yaitu $26 - 25.9 = 0.1$.

Terakhir pengujian alat dilakukan dengan nilai penguat (K_p) = 2. pada gambar 4.18 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 23.7°C . Pada grafik dapat dilihat untuk mencapai rise time kira-kira diperlukan 18 menit. Untuk mencapai nilai settling time diperlukan waktu 20 menit dan perbedaan nilai masukan dan nilai keluaran yaitu $26 - 26.1 = 0.1$.

4.10.2.2 Pengujian Alat Menaikkan Suhu Dengan Nilai Setpoint 27

Pengujian alat untuk menaikkan suhu dengan nilai suhu yang diinginkan pada ruangan/setpoint yaitu 27 dan nilai penguat (K_p) variasi yaitu 0.5, 1 dan 2. Hasil pengujian dapat dilihat pada gambar 4.19 berikut.



Gambar 4.20 grafik respon alat untuk menaikkan suhu pada setpoint 27

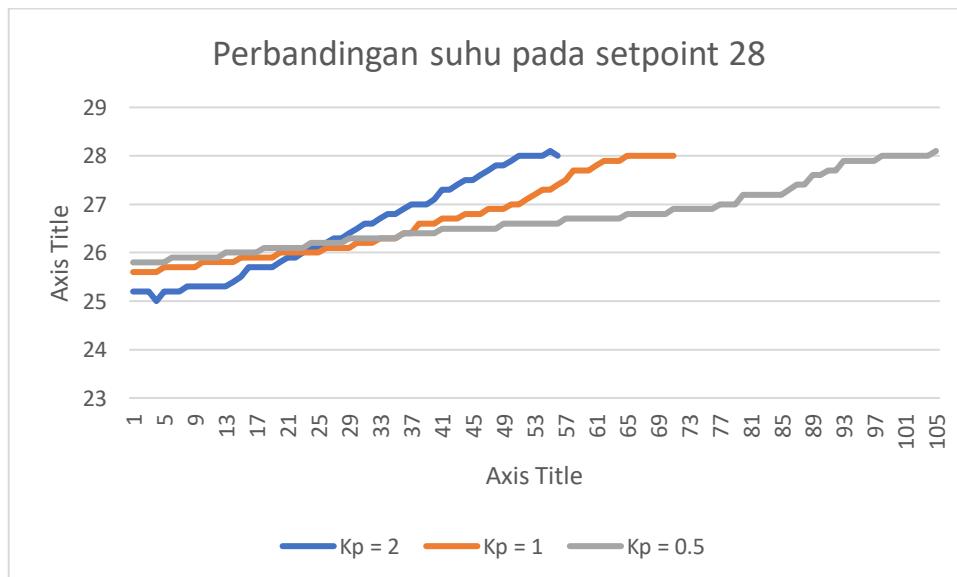
Analisa : pengujian alat dilakukan pada setpoint 27 dan nilai penguat (K_p) = 0.5. pada gambar 4.19 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu

25.7 °C. Pada grafik dapat dilihat untuk mencapai rise time diperlukan 65-70 menit. Untuk mencapai settling time diperlukan waktu 69 menit dan perbedaan nilai masukan dan nilai keluaran yaitu $27 - 27 = 0$. Selanjutnya pengujian dengan nilai penguat (K_p) = 1. pada gambar 4.19 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 24.8°C. Pada grafik dapat dilihat untuk mencapai rise time diperlukan 55-60 menit. Untuk mencapai nilai settling time diperlukan waktu 50 menit dan perbedaan nilai masukan dan nilai keluaran yaitu $27 - 27 = 0$.

Terakhir pengujian alat dilakukan dengan nilai penguat (K_p) = 2. pada gambar 4.19 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 25.1 °C. Pada grafik dapat dilihat untuk mencapai rise time kira-kira diperlukan 35 menit. Untuk mencapai nilai settling time diperlukan waktu 33 menit dan perbedaan nilai masukan dan keluaran yaitu $27 - 27.1 = 0.1$.

4.10.2.3 Pengujian Alat Menaikkan Suhu Dengan Nilai Setpoint 28

Pengujian alat untuk menaikkan suhu dengan nilai suhu yang diinginkan pada ruangan/setpoint yaitu 28 dan nilai penguat (K_p) variasi yaitu 0.5, 1 dan 2. Hasil pengujian dapat dilihat pada gambar 4.20 berikut.



Gambar 4.21 grafik respon alat untuk menaikkan suhu pada setpoint 28

Analisa : pengujian alat dilakukan pada setpoint 28 dan nilai penguat (K_p) = 0.5. pada gambar 4.20 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 25.8°C . Pada grafik dapat dilihat untuk mencapai rise time diperlukan 92 menit. Untuk mencapai nilai settling time diperlukan waktu 92 menit dan perbedaan nilai masukan dan nilai keluaran yaitu $28 - 27.9 = 0.1$. Selanjutnya pengujian alat dilakukan dengan nilai penguat (K_p) = 1. pada gambar 4.20 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 25.6°C . Pada grafik dapat dilihat untuk mencapai rise time kira-kira diperlukan 60 menit. Untuk mencapai nilai settling time diperlukan waktu 60 menit dan perbedaan nilai masukan dan keluaran yaitu $28 - 28 = 0$.

Terakhir pengujian alat dilakukan dengan nilai penguat (K_p) = 2. pada gambar 4.20 dapat dilihat merupakan hasil pengujian untuk menaikkan suhu pada ruangan. Pada awal pengujian suhu pada ruangan diturunkan pada suhu yaitu 25.2°C . Pada grafik dapat dilihat untuk mencapai rise time kira-kira diperlukan 46 menit. Untuk mencapai nilai settling time diperlukan waktu 47 menit dan perbedaan nilai masukan dan nilai keluaran yaitu $28 - 28 = 0$.

Setelah melakukan pengujian untuk menaikkan suhu dengan menggunakan penguat proporsional (K_p) dengan besar 0.5, 1 dan 2 didapatkan bahwa pada nilai $K_p = 0.5$ untuk mencapai nilai setpoint 26 diperlukan waktu 44 menit, pada $K_p = 1$ untuk mencapai nilai setpoint 26 diperlukan waktu 43-45 menit dan pada $K_p = 2$ untuk mencapai nilai setpoint 26 diperlukan waktu 18 menit. Untuk mencapai nilai setpoint 27 pada $K_p = 0.5$ diperlukan waktu 65-70 menit, pada $K_p = 1$ diperlukan waktu 55-60 menit dan pada $K_p = 2$ diperlukan waktu 39 menit. Untuk mencapai nilai setpoint 28 pada $K_p = 0.5$ diperlukan waktu 92 menit, pada $K_p = 1$ diperlukan waktu 60 menit dan pada $K_p = 2$ diperlukan waktu 46 menit.

Tabel 4.5 hasil pengujian alat untuk menaikkan suhu

Nilai Setpoint	Suhu Awal (°C)	Kp	Steady State Error	Rise Time (Menit)	Settling Time (Menit)
26	25.5	0.5	0.2	44	50
	25.4	1	0.1	43	49
	23.7	2	0.1	18	20
27	25.7	0.5	0	65	69
	24.8	1	0	55	60
	25.1	2	0.1	35	45
28	25.8	0.5	0.1	92	92
	25.6	1	0	60	60
	25.2	2	0	46	47

Pada tabel 4.4 merupakan hasil pengujian alat untuk menaikkan suhu pada suatu ruangan. Berdasarkan hasil pengujian tabel diatas nilai Kp = 2 Lebih cepat mencapai rise time dan settling dibandingkan dengan nilai Kp = 1 atau Kp = 0.5.



BAB V

PENUTUP

5.1 Kesimpulan

Setelah menyelesaikan tahap perancangan alat, pembuatan sistem *monitoring* dan kemudian diakhiri dengan tahap pengujian, maka berdasarkan hasil data yang diperoleh melalui pengujian dapat disimpulkan sebagai berikut :

1. Alat dapat memonitoring suhu pada ruangan dengan menggunakan aplikasi Blynk dan alat dapat mengendalikan pendingin ruangan dengan menaikkan atau menurunkan nilai suhu pada *air conditioner* untuk mendapatkan suhu yang diinginkan pada ruangan. Suhu pada ruangan dapat dimonitoring dan pendingin ruangan dapat dikendalikan dari jarak jauh karena tersambung dengan internet menggunakan komponen yaitu NodeMCU ESP8266.
2. Alat akan terus bekerja otomatis untuk mendapatkan suhu ruangan sesuai dengan yang diinginkan. Alat akan bekerja otomatis dengan membandingkan nilai suhu yang dimasukkan pada *smartphone* dengan nilai suhu yang terbaca oleh sensor DHT11 pada alat.
3. Pengujian respon alat terhadap nilai suhu yang kita inginkan pada ruangan menunjukkan bahwa semakin rendah nilai penguat kontrol proporsional yang kita gunakan maka waktu rise time akan semakin lambat dan jika nilai penguat proporsional yang digunakan lebih besar maka waktu rise time akan semakin cepat.

5.2 Saran

Setelah menyelesaikan penggerjaan proyek akhir ini masih terdapat beberapa kekurangan dan dimungkinkan untuk melakukan pengembangan lebih lanjut. Beberapa saran yang perlu disampaikan adalah sebagai berikut :

1. Alat hanya dapat memonitoring dan mengendalikan pendingin ruangan jika tersambung dengan jaringan internet jadi untuk pengembangan dan perbaikan kedepannya tempat alat harus memiliki akses internet yang bagus untuk membuat alat bekerja dengan maksimal.
2. Pastikan bahwa alat monitoring dan pengendalian pendingin ruangan berbasis internet tersambung secara terus menerus dengan sumber listrik.

DAFTAR PUSTAKA

- [1] Wikipedia, "Pemantauan," Wikimedia, 26 Juli 2020. [Online]. Available: <Https://Id.Wikipedia.Org/Wiki/Pemantauan>. [Accessed 7 Agustus 2021].
- [2] M. H. A. Khairi, "Cara Mengukur Suhu Dan Kelembaban Dengan Dht11 Dan Arduino," Mahir Elektro, 17 April 2021. [Online]. Available: <Https://Www.Mahirelektr.Com/2020/02/Tutorial-Menggunakan-Sensor-Dht11-Pada-Arduino.Html>. [Accessed 31 Agustus 2021].
- [3] Adafruit, "Dht11 Vs Dht22," Adafruit, 6 Agustus 2021. [Online]. Available: <Https://Learn.Adafruit.Com/Dht>. [Accessed 7 Agustus 2021].
- [4] Ajie, "Memulai Menggunakan Nodemcu Esp8266 Pada Arduino Ide," Indomaker, 22 Desember 2018. [Online]. Available: <Http://Indomaker.Com/Index.Php/2018/12/22/Memulai-Menggunakan-Nodemcu-Esp8266-Pada-Arduino-Ide/>. [Accessed 31 Agustus 2021].
- [5] Wikipedia, "Inframerah," Wikimedia, 10 Juni 2021. [Online]. Available: <Https://Id.Wikipedia.Org/Wiki/Inframerah>. [Accessed 7 Agustus 2021].
- [6] N. I. Widiastuti, "Kajian Sistem Monitoring Dokumen Akreditasi Teknik Informatika Unikom," *Majalah Ilmiah Unikom*, Vol. 12 No.2, Pp. 195-202.
- [7] S. H. W. Sasono, "Orbith," *Rancang Bangun Monitoring Dan Kontrol Suhu Pada Penyimpanan Biji Kedelai Berbasis Iot*, Vol. 16, Pp. 66-71, 2020.
- [8] E. P. Putra, "Pid (Proportional-Integral-Derivative) Controller," 21 November 2013. [Online]. Available: <Https://Putraekapermana.Wordpress.Com/2013/11/21/Pid/>. [Accessed 7 Agustus 2021].
- [9] A. R. Fenny Vinola, "Sistem Monitoring Dan Controlling Suhu Ruangan Berbasis Internet Of Things," *Teknik Elektro Dan Komputer*, Vol. 9, Pp. 117-

126, 2020.

- [10] A. Agarwal, "Automatic Arduino Based Ir Remote Control Temperature Driven," Arduino, 28 Mei 2020. [Online]. Available: <Https://Create.Arduino.Cc/Projecthub/Akarsh98/Automatic-Arduino-Based-Ir-Remote-Control-Temperature-Driven-99d1d7>. [Accessed 11 Juli 2021].

LAMPIRAN

LAMPIRAN 1 DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama : Lara Septiasari
Tempat, Tanggal lahir : Tempilang, 19 September 1999
Alamat Rumah : Jl. Tanjung Gadung Kecamatan Tempilang
Telp : -
Hp : 0831 7603 5815
Email : septiasarilara@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



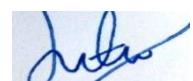
2. Riwayat Pendidikan

SD Negeri 11 Tempilang
MTS Nurul Huda Tempilang
SMA Negeri 1 Tempilang

3. Pendidikan Non Formal

-

Sungailiat, 05 Agustus 2021



Lara Septiasasari

LAMPIRAN 2 DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama : Muhammad Fahriza Firdausy
Tempat, Tanggal lahir: Sungailiat, 15 Oktober 2000
Alamat Rumah : Jl. Duyung Raya No.679 Desa Karya
Makmur Kec. Pemali
Telp : -
Hp : 0813 7374 5625
Email : firdausyfahriza@gmail.com
Jenis Kelamin : Laki-Laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 25 Sungailiat
SMP Negeri 1 Sungailiat
SMK Negeri 1 Sungailiat

3. Pendidikan Non Formal

Sungailiat, 05 Agustus 2021

Muhammad Fahriza Firdausy

LAMPIRAN 3 KODE PROGRAM KESELURUHAN

```
#include <Arduino.h>
#include <BlynkSimpleEsp8266.h>
#include <IRremoteESP8266.h>
#include <ESP8266WiFi.h>
#include "DHT.h"
#include <IRsend.h>
#include <ir_Sharp.h>
#define BLYNK_PRINT Serial
#define dht_pin D4
#define DHTTYPE DHT11
#define BLYNK_PRINT Serial

DHT dht(dht_pin, DHTTYPE);
BlynkTimer timer;
int setPoint, error, Kp, u;
float temp;

const uint16_t kIrLed = 4; // ESP8266 GPIO pin to use. Recommended:
                           // GPIO4 (D2)
IRsend irsend(kIrLed);

// ====== CONNECT TO WIFI ======
char auth[] = "IhlkdvhNqUYOSGv5rO4ggsq4yn0OKhjy";
char ssid[] = "ASUS_X00TD";
char pass[] = "mff151000";

void temperature() {

    Blynk.virtualWrite(V0, temp);
}

BLYNK_WRITE(V2)
{
    setPoint = param.asInt();
}

//RAW CODE
uint16_t dataSuhu30[137] = {9876, 9732, 9876, 9732, 4696, 2406, 466,
260, 468, 874, 414, 904, 412, 296, 440, 900, 414, 292, 466, 266,
440, 302, 442, 286, 464, 876, 412, 294, 466, 264, 444, 898, 412,
294, 442, 290, 442, 302, 466, 258, 444, 292, 444, 294, 442, 288,
444, 292, 466, 878, 438, 270, 466, 276, 468, 866, 440, 874, 438,
268, 468, 264, 468, 268, 466, 878, 438, 268, 468, 274, 468, 258,
470, 266, 444, 292, 468, 870, 440, 266, 446, 288, 468, 264, 468,
274, 446, 890, 440, 264, 468, 878, 440, 266, 468, 268, 468, 266,
446, 288, 468, 276, 470, 256, 470, 266, 468, 268, 444, 288, 468,
872, 440, 876, 438, 268, 446, 300, 468, 256, 470, 266, 470, 876,
416, 292, 468, 872, 438, 266, 468, 846, 462, 274, 442, 20278,
4694};

uint16_t dataSuhu29[137] = {9900, 9706, 9902, 9704, 4698, 2406, 466,
260, 468, 872, 440, 876, 440, 268, 466, 872, 440, 294, 440, 262,
470, 276, 468, 258, 468, 872, 440, 268, 468, 262, 470, 870, 442,
264, 468, 264, 468, 276, 466, 258, 470, 268, 468, 264, 472, 260,
472, 294, 442, 874, 440, 266, 470, 274, 468, 864, 440, 870, 440,
```

```
268, 468, 264, 468, 268, 466, 876, 440, 266, 470, 274, 470, 256,
472, 264, 470, 266, 466, 870, 442, 264, 472, 264, 470, 262, 470,
274, 468, 864, 442, 266, 470, 874, 442, 266, 468, 264, 470, 266,
468, 262, 470, 274, 462, 870, 440, 266, 470, 266, 470, 868, 442,
264, 470, 874, 418, 290, 446, 298, 468, 256, 446, 290, 468, 876,
440, 266, 468, 848, 464, 266, 468, 844, 466, 854, 464, 20252,
4670};
```

```
uint16_t dataSuhu28[137] = {9872, 9734, 9874, 9734, 4670, 2432, 442,
286, 440, 900, 412, 904, 412, 296, 438, 900, 416, 292, 440, 290,
444, 300, 442, 286, 442, 898, 414, 294, 440, 292, 442, 898, 414,
292, 440, 290, 442, 302, 442, 286, 464, 272, 442, 292, 444, 288,
444, 292, 442, 902, 414, 292, 442, 300, 444, 890, 414, 896, 414,
292, 468, 264, 444, 290, 446, 900, 414, 292, 444, 300, 444, 282,
468, 266, 468, 268, 468, 870, 438, 268, 446, 290, 444, 288, 468,
274, 444, 890, 438, 268, 444, 900, 438, 268, 444, 292, 468, 266,
468, 264, 468, 276, 468, 260, 468, 268, 442, 292, 466, 846, 464,
266, 468, 852, 462, 270, 442, 302, 464, 262, 464, 270, 440, 878,
462, 272, 442, 294, 464, 270, 464, 846, 462, 858, 460, 20254,
4694};
```

```
uint16_t dataSuhu27[137] = {9902, 9706, 9900, 9706, 4700, 2402, 470,
256, 470, 870, 442, 874, 442, 294, 440, 870, 440, 268, 466, 266,
468, 274, 468, 258, 472, 870, 442, 266, 470, 290, 440, 872, 442,
268, 468, 264, 468, 274, 470, 254, 472, 292, 442, 268, 468, 262,
468, 268, 466, 874, 442, 266, 472, 274, 468, 862, 442, 870, 442,
294, 440, 262, 470, 264, 470, 874, 442, 266, 470, 272, 470, 256,
472, 268, 468, 294, 442, 866, 442, 266, 470, 264, 470, 262, 470,
274, 468, 862, 444, 264, 470, 874, 444, 266, 468, 266, 472, 264,
470, 262, 470, 272, 470, 862, 442, 870, 440, 874, 442, 268, 472,
260, 468, 876, 442, 264, 470, 274, 470, 256, 472, 264, 468, 850,
442, 292, 442, 870, 442, 874, 464, 272, 466, 856, 464, 20252,
4694};
```

```
uint16_t dataSuhu26[137] = {9902, 9706, 9900, 9706, 4700, 2400, 464,
260, 468, 874, 414, 902, 438, 268, 468, 874, 438, 268, 444, 288,
444, 300, 466, 260, 468, 872, 438, 268, 468, 262, 468, 874, 438,
266, 470, 262, 468, 276, 470, 862, 442, 266, 468, 268, 460, 878,
440, 872, 440, 268, 468, 260, 470, 276, 468, 864, 440, 872, 440,
268, 468, 262, 470, 266, 468, 876, 418, 290, 468, 276, 466, 258,
448, 288, 468, 268, 470, 868, 416, 290, 468, 266, 468, 262, 470,
274, 470, 864, 440, 266, 468, 876, 442, 264, 446, 290, 466, 268,
466, 264, 470, 274, 444, 284, 442, 872, 464, 852, 440, 294, 464,
270, 442, 876, 462, 272, 466, 278, 464, 262, 466, 270, 442, 876,
464, 272, 442, 294, 466, 852, 462, 272, 466, 274, 464, 20256,
4694};
```

```
uint16_t dataSuhu25[137] = {9876, 9730, 9900, 9708, 4698, 2404, 468,
256, 470, 874, 422, 894, 438, 272, 464, 874, 440, 268, 466, 264,
468, 278, 466, 260, 470, 870, 424, 284, 468, 264, 468, 874, 440,
294, 438, 264, 470, 274, 468, 864, 440, 266, 470, 268, 466, 870,
440, 872, 438, 268, 466, 266, 468, 276, 466, 866, 440, 872, 440,
268, 468, 264, 468, 266, 468, 876, 440, 270, 466, 278, 464, 256,
472, 266, 466, 298, 440, 870, 440, 270, 468, 264, 468, 262, 472,
272, 468, 864, 440, 268, 466, 876, 440, 268, 468, 266, 470, 266,
468, 266, 466, 278, 464, 866, 440, 266, 470, 876, 440, 266, 468,
268, 466, 876, 440, 296, 440, 278, 468, 256, 472, 264, 470, 874,
442, 266, 468, 872, 440, 266, 468, 292, 442, 270, 470, 20250,
4700};
```

```
uint16_t dataSuhu24[137] = {9876, 9732, 9900, 9706, 4698, 2400, 468,
258, 470, 872, 440, 876, 440, 296, 440, 872, 440, 266, 470, 290,
440, 278, 466, 256, 468, 874, 440, 294, 440, 262, 470, 870, 442,
```

```

266, 470, 262, 468, 276, 470, 862, 440, 268, 468, 268, 468, 868,
440, 870, 440, 296, 440, 262, 470, 272, 470, 864, 440, 870, 442,
294, 440, 262, 472, 292, 440, 874, 442, 268, 468, 278, 464, 260,
468, 266, 468, 268, 468, 868, 440, 268, 468, 268, 468, 262, 470,
274, 470, 862, 440, 266, 470, 874, 442, 268, 468, 264, 470, 266,
470, 262, 470, 274, 470, 256, 472, 266, 468, 874, 442, 266, 470,
264, 444, 898, 444, 264, 470, 274, 468, 256, 472, 264, 472, 874,
442, 268, 468, 264, 470, 266, 446, 286, 470, 270, 468, 20252,
4694};

uint16_t dataSuhu23[137] = {9900, 9702, 9904, 9704, 4702, 2398, 470,
288, 442, 868, 440, 874, 442, 294, 440, 872, 442, 294, 442, 290,
442, 276, 466, 256, 472, 870, 442, 268, 468, 262, 470, 868, 442,
294, 442, 262, 470, 302, 440, 864, 442, 266, 470, 292, 442, 866,
442, 870, 442, 266, 470, 260, 470, 302, 442, 862, 442, 868, 442,
268, 466, 290, 442, 266, 468, 874, 442, 270, 468, 272, 470, 254,
474, 292, 442, 294, 442, 866, 442, 266, 468, 294, 442, 260, 470,
274, 470, 862, 442, 268, 466, 874, 444, 268, 468, 264, 470, 294,
440, 262, 470, 274, 468, 862, 444, 868, 442, 268, 468, 262, 470,
266, 468, 874, 442, 268, 468, 272, 470, 256, 472, 262, 446, 900,
440, 266, 470, 848, 438, 876, 464, 846, 466, 854, 440, 20276,
4694};

uint16_t dataSuhu22[137] = {9874, 9730, 9900, 9708, 4698, 2404, 470,
258, 468, 872, 440, 876, 440, 268, 466, 872, 440, 268, 468, 262,
468, 304, 440, 288, 440, 870, 442, 294, 440, 264, 470, 870, 440,
266, 468, 292, 440, 274, 470, 864, 440, 268, 468, 296, 440, 868,
442, 870, 442, 268, 468, 264, 470, 274, 468, 864, 442, 870, 442,
266, 470, 262, 470, 264, 472, 874, 442, 268, 468, 274, 468, 256,
468, 268, 468, 268, 468, 866, 442, 266, 468, 266, 468, 262, 472,
274, 468, 862, 442, 264, 470, 874, 442, 266, 468, 268, 468, 266,
468, 264, 470, 272, 446, 282, 468, 870, 442, 268, 466, 262, 446,
292, 468, 874, 442, 266, 470, 274, 470, 256, 462, 274, 470, 874,
440, 266, 470, 264, 470, 876, 442, 846, 466, 854, 464, 20250,
4670};

uint16_t dataSuhu21[137] = {9900, 9704, 9900, 9706, 4700, 2404, 468,
256, 472, 870, 438, 878, 440, 296, 440, 870, 440, 268, 468, 264,
468, 276, 468, 256, 470, 872, 440, 294, 440, 262, 468, 872, 442,
294, 440, 262, 472, 274, 466, 864, 442, 264, 468, 296, 442, 866,
440, 872, 442, 264, 468, 264, 468, 276, 466, 864, 442, 870, 442,
264, 472, 262, 470, 264, 472, 874, 442, 266, 468, 274, 468, 256,
470, 268, 468, 294, 442, 868, 440, 266, 466, 266, 468, 262, 470,
276, 466, 864, 442, 266, 468, 876, 442, 294, 442, 294, 442, 294,
442, 260, 470, 274, 468, 862, 442, 266, 470, 264, 470, 260, 472,
264, 470, 874, 442, 266, 470, 274, 470, 256, 472, 264, 470, 874,
442, 266, 446, 894, 442, 266, 444, 892, 442, 878, 442, 20246,
4682};

uint16_t dataSuhu20[137] = {9872, 9734, 9874, 9732, 4670, 2428, 440,
288, 440, 902, 412, 904, 410, 298, 438, 900, 412, 294, 440, 290,
442, 304, 440, 288, 442, 900, 410, 296, 440, 292, 440, 900, 412,
294, 440, 292, 442, 300, 440, 892, 412, 294, 440, 296, 438, 898,
412, 900, 412, 294, 442, 290, 442, 300, 444, 892, 412, 900, 412,
294, 466, 266, 442, 296, 440, 904, 412, 296, 440, 302, 442, 286,
442, 292, 442, 294, 440, 896, 414, 292, 442, 294, 442, 290, 442,
302, 442, 892, 438, 268, 466, 878, 414, 296, 462, 270, 442, 294,
440, 290, 468, 276, 442, 286, 442, 292, 442, 292, 442, 290, 442,
292, 468, 878, 438, 268, 444, 300, 442, 284, 466, 270, 442, 902,
416, 294, 466, 268, 444, 294, 466, 870, 438, 882, 438, 20250,
4682};

uint16_t dataSuhu19[137] = {9872, 9732, 9874, 9732, 4672, 2434, 440,
288, 438, 902, 410, 906, 412, 294, 442, 900, 412, 294, 442, 290,
4682};

```

```

440, 304, 440, 286, 442, 898, 414, 292, 442, 290, 442, 898, 412,
296, 442, 290, 440, 302, 442, 892, 412, 294, 442, 292, 444, 896,
412, 898, 414, 294, 442, 290, 442, 300, 466, 868, 412, 900, 436,
268, 444, 288, 444, 292, 440, 902, 414, 292, 464, 278, 442, 284,
444, 292, 442, 292, 444, 894, 416, 292, 466, 268, 468, 264, 458,
286, 466, 866, 414, 292, 466, 878, 438, 270, 442, 292, 442, 292,
442, 290, 466, 274, 446, 890, 438, 266, 468, 268, 458, 880, 438,
874, 440, 268, 466, 264, 468, 276, 444, 284, 468, 268, 466, 880,
438, 268, 468, 268, 442, 294, 442, 870, 462, 274, 442, 20280,
4670};
```

```

uint16_t dataSuhu18[137] = {9902, 9704, 9902, 9704, 4698, 2404, 470,
258, 470, 870, 442, 874, 440, 268, 468, 870, 440, 268, 468, 292,
440, 274, 470, 256, 472, 868, 442, 268, 466, 292, 440, 872, 442,
268, 468, 262, 468, 274, 470, 258, 468, 266, 470, 294, 440, 868,
442, 870, 440, 294, 442, 262, 470, 274, 468, 862, 442, 870, 442,
266, 470, 262, 470, 266, 470, 874, 442, 294, 440, 274, 470, 256,
472, 266, 468, 266, 470, 868, 442, 264, 470, 266, 470, 260, 472,
274, 468, 862, 440, 266, 470, 874, 442, 266, 470, 266, 468, 268,
468, 262, 468, 274, 468, 258, 472, 266, 470, 264, 470, 868, 444,
868, 442, 266, 470, 262, 468, 274, 470, 258, 470, 264, 470, 874,
442, 266, 470, 264, 468, 852, 464, 268, 468, 272, 468, 20254,
4670};
```

```

uint16_t dataSuhu17[137] = {9898, 9708, 9902, 9704, 4702, 2400, 470,
256, 472, 870, 442, 874, 442, 294, 440, 870, 444, 292, 442, 290,
442, 274, 468, 254, 472, 870, 442, 292, 442, 290, 442, 870, 442,
266, 470, 262, 470, 276, 468, 256, 472, 294, 440, 294, 442, 866,
444, 868, 444, 294, 440, 264, 468, 274, 470, 860, 442, 870, 442,
294, 442, 290, 442, 268, 468, 872, 444, 294, 440, 276, 470, 254,
474, 266, 468, 266, 470, 866, 442, 266, 468, 268, 468, 290, 442,
274, 468, 860, 442, 266, 470, 874, 442, 268, 470, 262, 468, 268,
472, 260, 470, 274, 470, 862, 442, 870, 440, 876, 442, 266, 470,
870, 440, 266, 470, 262, 470, 274, 468, 256, 448, 292, 442, 874,
464, 270, 442, 870, 440, 294, 440, 292, 462, 274, 446, 20276,
4694};
```

```

uint16_t dataSuhu16[137] = {9872, 9732, 9874, 9734, 4670, 2434, 438,
286, 440, 902, 410, 906, 410, 298, 438, 900, 412, 294, 440, 290,
442, 302, 440, 286, 442, 900, 412, 294, 440, 292, 440, 900, 412,
294, 440, 290, 442, 300, 440, 288, 442, 294, 440, 296, 440, 898,
412, 900, 412, 296, 440, 292, 440, 302, 442, 892, 412, 900, 414,
294, 442, 290, 442, 292, 442, 904, 414, 292, 442, 300, 442, 286,
446, 290, 442, 292, 466, 872, 414, 292, 442, 292, 444, 288, 468,
276, 466, 868, 436, 268, 466, 880, 438, 268, 466, 268, 466, 268,
466, 264, 444, 300, 442, 284, 442, 898, 436, 880, 436, 270, 466,
850, 460, 270, 442, 290, 464, 278, 442, 286, 440, 294, 440, 878,
462, 272, 464, 270, 464, 272, 446, 286, 462, 276, 464, 20258,
4694};
```

```

void setup() {
    Serial.begin(9600);
    dht.begin();
    irsend.begin();
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(400L, temperature);
    temp = dht.readTemperature();
}

void loop() {
    Blynk.run();
```

```

timer.run();
temp = dht.readTemperature();
Serial.print("Suhu = ");
Serial.print(temp);
Serial.print(" C");
Serial.print("\n");
delay(2000);

error = setPoint - temp;
Kp = 1;
u = Kp * error;

if(u >= -6){
irsend.sendRaw(dataSuhu18, 137, 36);
delay(10000);

}
if(u == -5){
irsend.sendRaw(dataSuhu20, 137, 36);
delay(10000);
}
if(u == -4){
irsend.sendRaw(dataSuhu21, 137, 36);
delay(10000);
}
if(u == -3){
irsend.sendRaw(dataSuhu22, 137, 36);
delay(10000);
}
if(u == -2){
irsend.sendRaw(dataSuhu23, 137, 36);
delay(10000);
}
if(u == -1){
irsend.sendRaw(dataSuhu24, 137, 36);
delay(10000);
}
if(u == 1){
irsend.sendRaw(dataSuhu25, 137, 36);
delay(10000);
}
if(u == 2){
irsend.sendRaw(dataSuhu26, 137, 36);
delay(10000);
}
if(u == 3){
irsend.sendRaw(dataSuhu27, 137, 36);
delay(10000);
}
if(u == 4){
irsend.sendRaw(dataSuhu28, 137, 36);
delay(10000);
}
if(u == 5){
irsend.sendRaw(dataSuhu29, 137, 36);
delay(10000);
}
if(u == 6){
irsend.sendRaw(dataSuhu30, 137, 36);
delay(10000);
}

}

```

LAMPIRAN 4 KODE PROGRAM INFRARED RECEIVER

```
#include <Arduino.h>
#include <IRrecv.h>
#include <IRremoteESP8266.h>
#include <IRac.h>
#include <IRtext.h>
#include <IRutils.h>

const uint16_t kRecvPin = 5;
const uint32_t kBaudRate = 9600;
const uint16_t kCaptureBufferSize = 1024;

#if DECODE_AC
const uint8_t kTimeout = 50;
#else
const uint8_t kTimeout = 15;
#endif
const uint16_t kMinUnknownSize = 12;
#define LEGACY_TIMING_INFO false
IRrecv irrecv(kRecvPin, kCaptureBufferSize, kTimeout, true);
decode_results results;

void setup() {
#if defined(ESP8266)
    Serial.begin(kBaudRate, SERIAL_8N1, SERIAL_TX_ONLY);
#else
    Serial.begin(kBaudRate, SERIAL_8N1);
#endif
    while (!Serial)  establised.
    delay(50);
    Serial.printf("\n" D_STR_IRRECVDUMP_STARTUP "\n", kRecvPin);
#if DECODE_HASH
    irrecv.setUnknownThreshold(kMinUnknownSize);
#endif
    irrecv.enableIRIn();
}

void loop() {
```

```
if (irrecv.decode(&results)) {  
    uint32_t now = millis();  
    Serial.printf(D_STR_TIMESTAMP " : %06u.%03u\n", now / 1000, now % 1000);  
    if (results.overflow)  
        Serial.printf(D_WARN_BUFFERFULL "\n", kCaptureBufferSize);  
    Serial.println(D_STR_LIBRARY": v"_IRREMOTEESP8266_VERSION_ "\n");  
    Serial.print(resultToHumanReadableBasic(&results));  
    String description = IRAcUtils::resultActToString(&results);  
  
    if (description.length())    Serial.println(D_STR_MSGDESC ": " +  
        description);  
    #if LEGACY_TIMING_INFO  
    Serial.println(resultToTimingInfo(&results));  
    yield();  
    #endif  
    Serial.println(resultToSourceCode(&results));  
    Serial.println();  
    yield();  
}  
}
```