SISTEM KENDALI PADA ALAT PENGERING KERUPUK BERBASIS IOT MENGUNAKAN MODUL ESP32

PROYEK AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Program Studi Teknik Elektronika Untuk Memperoleh Gelar Ahli Madya Fakultas Teknik Universitas Negeri Padang

Oleh:

RIJALUL FAHMI

NIM 2019/19066027

PROGRAM STUDI TEKNIK ELEKTRONIKA

DEPARTEMEN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2023

PERSETUJUAN PEMBIMBING PROYEK AKHIR

Judul	: Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32
Nama	: Rijalul Fahmi
NIM	: 19066027
Program Studi	: D3 Teknik Elektronika
Departemen	: Teknik Elektronika
Fakultas	: Teknik

Padang, Juni 2023

Disetujui Oleh:

Pembimbing,

Drs. Eddas, M.T

NIP. 196302091988031004

Mengetahui,

Ketua Departemen Teknik Elektronika

Fakultas Teknik Universitas Negeri Padang

Thamrin,S.Pd.,M.T.

NIP.197701012008121001

Dipindai dengan CamScanner

PENGESAHAN PROYEK AKHIR

Nama : Rijalul Fahmi

NIM : 19066027

Dinyatakan lulus setelah mempertahankan proyek akhir di depan Tim Penguji

Program Studi DIII Teknik Elektronika

Departemen Elektronika

Fakultas Teknik

Universitas Negeri Padang

dengan judul:

"Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32"

Padang, 24 Juni 2023

Tim Penguji:

	Nama	Tanda Tangan
1. Ketua	: Sartika Anori S.Pd., M.Pd.T	1.
2. Anggota	: Dr. Edidas, MT	2.
3. Anggota	: Zulwisli S.Pd., M.Eng	3.

Dipindai dengan CamScanner

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis saya, tugas akhir berupa proyek akhir dengan judul "Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32" adalah asli karya saya sendiri.
2. Karya tulis ini murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali dari pembimbing.
3. Pernyataan ini saya buat dengan sesungguhnya dan apabila terdapat penyimpangan di dalam pernyataan ini, saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma dan ketentuan hukum yang berlaku.

Padang, 24 Juni 2023

Yang Menyatakan:

Rijalul Fahmi

NIM. 2019/19066027

Dipindai dengan CamScanner

ABSTRAK

Rijalul Fahmi, 2023. "Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32".

Industri kerupuk mengalami kemajuan pesat karena dapat dilihat kerupuk sudah banyak mulai dijual di toko-toko yang besar, ini merupakan kemajuan dari industri dan hasil dari riset untuk menemukan bahan baku alternatif untuk kerupuk, tidak luput pula varian rasa dan kemasan dari kerupuk membuat kerupuk makin banyak diminati sebagai pendamping makan atau untuk cemilan. Alat pengering kerupuk dibuat dengan ESP32 sebagai proses kendali kontroler, Bot Telegram sebagai menerima pesan kondisi alat dan mengirim perintah ke ESP32, perintahnya yaitu menghidupkan pemanas dan mengatur waktu pemanas, Sensor DHT11 untuk pendeteksi suhu dan kelembapan, Limit switch sebagai pembatas dalam membuka dan membuka wadah yang dikendalikan oleh motor power window, Rain sensor sebagai pendeksi air hujan yg mengenai sensor, Motor power window sebagai pembuka dan penutup wadah, Drvier motor sebagai pengontrol motor power window, auto buck booster sebagai penurun tegangan, dan LCD sebagai tampilan dan memberikan infomasi mengenai status, Wi-fi, bot Telegram suhu, kelembapan dan timer, Power Supply sebagai pemberi daya ke beberapa komponen, Hairdryer digunakan menaikkan suhu dan mengurangi kelembapan, Lampu sebagai penerang serta juga memberikan suhu panas, fan berfungsi penstabil kelembapan ruangan.

Kata Kunci: Pengering Kerupuk, Industri, ESP32, Bot Telegram, Sensor DHT11, IDE Arduino, LCD.

KATA PENGANTAR

Segala puji dan syukur penulis sampaikan kehadirat Allat SWT, atas rahmat, ridho dan karunia-Nya sehingga penulis dapat menyelesaikan laporan yang berjudul "**Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32".** Selanjutnya shalawat beserta salam semoga disampaikan Allah SWT kepada junjungan umat, yakninya Nabi Muhammad SAW yang menjadi suri tauladan dalam setiap sikap dan Tindakan sebagai seorang muslim.

Pembuatan Proyek Akhir ini merupakan salah satu syarat dalam menyelesaikan Pendidikan Diploma Tiga (DIII) Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang. Penyelesaian Proyek Akhir ini tidak terlepas dari bantuan dan bimbingan serta dorongan dari berbagai pihak sehingga dapat diselesaikan segala hambatan dan rintangan yang dihadapi, untuk itu penulis mengucapkan terima kasih kepada:

1. Allah SWT atas nikmat yang luar biasa yang telah diberikan kepada penulis sehingga dapat menyelesaikan laporan dalam keadaan tanpa kekurangan apapun.
2. Keluarga yang selalu memberikan dukungan baik moril maupun materil.
3. Bapak Dr. Fahmi Rizal, M.Pd, M.T, selaku Dekan Fakultas Teknik Universitas Negeri Padang.
4. Bapak Thamrin S.Pd, M.T selaku ketua Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.
5. Bapak Zulwisli, S.Pd, M.Eng selaku ketua Program Studi DIII Teknik Elektronika.

6. Bapak Dr. Edidas, MT selaku pembimbing tugas akhir yang telah banyak memberi saran dan masukan dalam penyelesaian Proyek Akhir ini.
7. Ibuk Sartika Anori S.Pd., M.Pd.T selaku ketua penguji.
8. Bapak Zulwisli, S.Pd, M.Eng selaku anggota penguji.
9. Teman seperjuangan tugas akhir yang sama-sama berjuang.
10. Seluruh pihak yang telah banyak membantu dalam Menyusun laporan yang tidak bisa disebut namanya satu persatu.

Semoga segala motivasi, dorongan dan bantuan serta bimbingan yang diberikan menjadi amal jariah dan mendapat balasan yang setimpal dari Allah SWT. Penulis menyadari masih ada banyak kekurangannya, oleh karena itu, penulis sangat mengharapkan kritik dan juga saran yang sifatnya membangun untuk kesempurnaan Proyek Akhir ini. Semoga Proyek Akhir ini dapat bermanfaat bagi semua pihak dan bernilai ibadah di sisi Allah SWT.

Padang, Juni 2023

Rijalul Fahmi

DAFTAR ISI

PERSETUJUAN PEMBIMBING PROYEK AKHIR..... **Kesalahan! Bookmark tidak ditentukan.**

PENGESAHAN PROYEK AKHIR .....**Kesalahan! Bookmark tidak ditentukan.**

SURAT PERNYATAAN .....**Kesalahan! Bookmark tidak ditentukan.**

ABSTRAK v

KATA PENGANTAR..... vi

DAFTAR ISI..... viii

DAFTAR GAMBAR..... xi

BAB I PENDAHULUAN..... 1

A. Latar Belakang Masalah..... 1

B. Identifikasi Masalah..... 6

C. Batasan Masalah..... 7

D. Rumusan Masalah..... 7

E. Tujuan Proyek Akhir..... 7

F. Manfaat Proyek Akhir..... 7

BAB II LANDASAN TEORI 9

A. Internet of Things (IoT) 9

B. Modul ESP32 10

1. Spesifikasi ESP32..... 11

2. Bagian-bagian ESP32..... 11

C. IDE Arduino 12
D. Algoritma 15
1. Jenis-Jenis Algoritma 15
2. Keuntungan Memakai Algoritma 16
3. Hal yang Perlu Diperhatikan dalam Algoritma 17
4. Ciri-ciri Algoritma 19
E. Flowchart 19
F. Bahasa Pemograman C Arduino 22
1. Struktur 22
2. Elemen Bahasa C yang dibutuhkan dalam format penulisan 23
BAB III METODE PERANCANGAN DAN PEMBUATAN SOFTWARE 24
A. Diagram Blok Sistem 24
B. Prinsip Kerja Sistem 25
C. Rancangan Fisik Alat 26
D. Flowchart Sistem Alat Kerja 27
E. Pengujian Alat 28
1. Install Arduino IDE 29
2. Setting Arduino IDE 30
BAB IV PENGUJIAN ALAT 35
A. Pengujian Software 35
B. Pengujian Fungsional 43
BAB V KESIMPULAN DAN SARAN 46

A. Kesimpulan 46

B. Saran..... 46

DAFTAR PUSTAKA 48

DAFTAR GAMBAR

Gambar 1. Ilustrasi *Internet Of Thing's* 10

Gambar 2. NodeMCU ESP32 12

Gambar 3. Tampilan Awal IDE Arduino 13

Gambar 4. Tampilan Software IDE Arduino 13

Gambar 5. Simbol-Simbol Flowchart 20

Gambar 6. Flowchart Dasar 21

Gambar 7. Blok Diagram 24

Gambar 8. Tampak alat dari depan 29

Gambar 9. Tampak alat dari samping 29

Gambar 10. Flowchart Sistem Kerja Alat 25

Gambar 11. Percakapan dengan Bot Telegram 45

DAFTAR TABEL

Tabel 1. Pin Relay 43

Tabel 2. Pin Sensor Suhu 44

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Industri menjadi salah satu faktor yang sangat penting dalam siklus perekonomian dunia. Baik itu industri skala besar (Perusahaan), industri skala menengah maupun industri skala kecil (UMKM), bukti nyata pentingnya peran industri ini terhadap jalannya roda pemerintahan antara lain adalah konsistensi sumbangsihnya yang terbesar pada PDB nasional. Pada tahun 2020, kontribusi sektor industri pengolahan mencapai 17,89%. Selain itu, kinerja gemilang sektor industri tercemin pada capaian nilai ekspor dan investasi. Ada tujuh sektor industri yang sangat penting untuk saat ini, ketujuh sektor prioritas itu adalah industri makanan dan minuman, kimia, tekstil dan pakaian, otomotif, elektronik, farmasi, serta alat kesehatan. Sektor tersebut mampu memberikan lebih dari 60 persen pada PDB nasional, sehingga diharapkan target besarnya Indonesia menjadi negara 10 besar ekonomi terkuat di dunia pada tahun 2030. (kemenperin, 2021)

UU No. 5 Tahun 1984 tentang Perindustrian, industri adalah kegiatan ekonomi yang mengolah bahan mentah, bahan baku, barang setengah jadi, dan/atau barang jadi menjadi barang dengan nilai yang lebih tinggi untuk penggunaannya, termasuk kegiatan rancang bangun dan perekayasaan industri. Pembangunan industri dan pengembangan industri harus disesuaikan dengan potensi daerah dan dengan memperhatikan masalah-masalah yang ada pada daerah yang bersangkutan,

sebagai salah satu bentuk upaya untuk mensejahterakan masyarakat daerah yang bersangkutan pula.

Industri kerupuk mengalami kemajuan pesat karena dapat dilihat kerupuk sudah banyak mulai dijual di toko-toko yang besar, dan tidak lupur pula jenis kerupuk yang sekarang beredar di masyarakat sudah bervariasi, ini merupakan kemajuan dari industri dan hasil dari riset untuk menemukan bahan baku alternatif untuk kerupuk, tidak luput pula varian rasa dan kemasan dari kerupuk membuat kerupuk makin banyak diminati sebagai pendamping makan atau untuk cemilan.

Kemajuan ilmu pengetahuan dan teknologi meringankan pekerjaan di dunia industri dan mendorong manusia untuk berusaha mengatasi masalah yang timbul di sekitarnya. Salah satu kemudahan yang diberikan yaitu penggunaan mikrokontroler sangat luas, tidak hanya untuk akuisi data melainkan juga untuk pengendalian di pabrik – pabrik, kebutuhan peralatan kantor, peralatan rumah tangga, automobile, dan sebagainya. Dapat dilihat dari kemajuan teknologi dari modernitas yang ditandai dengan pertumbuhan ekonomi, mobilitas sosial, hingga perluasan budaya. Menurut Philip Sporn yang merupakan insinyur di bidang kelistrikan, kemajuan teknologi dapat diamati melalui cara-cara sistematis yang berlandaskan penemuan ilmiah melalui proses eksperimen. Hasil dari teknologi tersebut tidak lain merupakan produk yang dapat digunakan secara praktis ataupun mengenai jasa tertentu. (Fadly, 2020)

Perkembangan teknologi juga dapat dirasakan dalam bidang pendidikan. Seiring perkembangan teknologi, kamu dapat memanfaatkan media internet untuk

menambah wawasan dan pengetahuan yang mungkin tidak bisa kamu temukan di buku. Selain itu, dalam hal pendaftaran sekolah yang dulunya harus datang langsung ke sekolah yang diinginkan, sekarang sudah mulai menerapkan registrasi berbasis online yang dinilai menghemat waktu dan lebih efisien. Bahkan sekarang ini sudah ada universitas yang memberikan fasilitas belajar mengajar jarak jauh. Melalui perantara internet, kamu sudah bisa terhubung dengan dosen tanpa harus bertatap muka secara langsung. (Gradianto, 2023)

Mikrokontroler yang digunakan ialah mikrokontroler ESP32 yang merupakan pengendali dari setiap komponen-komponen yang digunakan komponen elektronik yang dapat digunakan untuk memenuhi kebutuhan teknologi dibidang elektronika. ESP32 ini mikrokontroler yang dikenalkan oleh Espessif System dan merupakan penerus dari mikrokontroler ESP8266, mikrokontroler ini *open source* yang digunakan untuk kebutuhan IoT. Proyek IoT terutama difokuskan pada aplikasi Home Automation dan Smart Home tetapi juga sudah banyak pengembangan proyek IoT untuk komersial dan industri yang memiliki implementasi jauh lebih kompleks seperti Machine Learning, Artificial Intelligence, Wireless Sensor Network, dll. Pada mikrokontroler ini dilengkapi dengan sebuah chip yang dimana didalamnya sudah termasuk *processor*, memori dan juga akses ke GPIO. Hal ini menyebabkan ESP32 dapat secara langsung menggantikan seperti Arduino dan ditambah lagi dengan kemampuannya untuk mensupport koneksi wifi secara langsung (Iqbal, 2022).

Modul ESP32 merupakan platform yang sangat murah tetapi benar-benar efektif untuk digunakan berkomunikasi atau kontrol melalui internet baik digunakan secara standalone (berdiri sendiri) maupun dengan menggunakan mikrokontroler tambahan dalam hal ini Arduino IDE sebagai pengendalinya. Melakukan pemograman pada modul ESP32 ini bisa dibilang cukup mudah, yaitu dengan mengintall Arduino IDE pada laptop dan kemudian menambahkan paket board ESP32, sehingga tidak memerlukan software lain dalam memprogram modul ini. (Sujono, 2019)

Bahan utama kerupuk ini adalah singkong yang jumlahnya berlebih di Jawa khususnya pada abad ke-19. Pada masa ini singkong menjadi salah satu komoditas pangan yang paling diandalkan oleh masyarakat Jawa. “Singkong bisa direbus, digoreng atau dijadikan gablek, kemudian diolah menjadi tepung dan jadi aci. Dan salah satu produk dari singkong ya kerupuk,” paparnya. Diduga kerupuk aci baru muncul pada abad ke-19, sehingga masyarakat Indonesia saat itu bertahan hidup dengan kerupuk. Masyarakat memanfaatkan kerupuk sebagai bahan pangan pokok karena wilayah tersebut mengalami devisit pangan akibat perang dan bisa jadi tanam paksa. Tepung singkong dimanfaatkan sebagai kerupuk dan dijadikan lauk bagi rakyat biasa. Tepung singkong diolah lalu dicetak kemudian dijemur dan akhirnya digoreng. Rakyat Indonesia yang kurang berpunya hanya bisa menyantap kerupuk sebagai lauk. Sebab bahan makanan seperti daging sangat minim, dan jikalau ada di pasar harganya sangat mahal. Masyarakat hanya bisa makan dari

kerupuk dan nasi, selain itu juga olahan bahan pangan yang murah seperti singkong. (Fadly, 2020)

Mengeringkan kerupuk diperlukan kadar air tertentu dari kerupuk mentah untuk menghasilkan tekanan uap yang maksimum pada proses penggorengan sehingga gel pati kerupuk bisa mengembang. Pengeringan kerupuk dapat dilakukan dengan dua cara yaitu penjemuran dibawah sinar matahari dan dengan menggunakan mesin pengering oven. Secara umum sistem pengeringan terdiri dari ruang tempat bahan yang akan dikeringkan, alat penghembus udara kering blower, dan pemanas tambahan untuk menaikkan suhu udara pengering. Keuntungan pengeringan dengan oven adalah suhu dan waktu dapat diatur (Ir. Sutrisno Koswara, 2013)

Proses penjemuran kerupuk dengan cara konvensional lainnya adalah saatproses penjemuran harus ada yang menunggu untuk mengangkat kerupuk jika tiba-tiba terjadihujan, terkadang ada banyak aktivitas lainselain harus menunggu kerupuk. Ini tentu menambah pekerjaan dan merepotkanmanusia. Untuk itu perlu dibuat alat pengontrol pengering yang otomatis sehingga saat tidak ada sinar matahari proses pengeringan masih bisa berjalan terus menerus tanpa tergantungcuaca dan kondisi. (Elwin Mulyanah, 2015)

Berdasarkan permasalahan tersebut dapat diatasi dengan membuat alat bantu untuk mengeringkan minyak kerupuk sebelum proses penggorengan. Ini tentu melibatkan sumber daya manusia yang kreatif, yang mampu memahami keadaan menjadi gagasan yang bagus untuk menciptakan alat yang praktis dan nyaman

digunakan. Penjemur kerupuk otomatis ini akan berfungsi membuka, menutup serta memanaskan. Saat ada sinar matahari maka tempat kerupuk akan bergeser keluar dan saat tidak ada matahari atau hujan maka tempat kerupuk akan masuk ke dalam danmelakukan proses pemanasan selanjutnya. Pembuatan alat tersebut bertujuan untuk mengurangi penggunaan tempat yang digunakan untuk mengeringkan kerupuk, mempercepat proses pengeringan. Selain itu dibuatnya alat tersebut juga diharapkan mampu menambah produktivitas UMKM.

Dari latar belakang masalah diatas maka penulis mencoba merancang dan membuat program pengontrol alat yang dapat mengetahui kadar air yang terdapat pada kerupuk tersebut, yang dibuat dalam bentuk proyek akhir dengan judul " **Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32"** sedangkan bagian perangkat keras (Hardware) dibuat oleh Muhammad Saddam dengan judul ***"Perancangan Dan Pembuatan Alat Pengering Kerupuk Mengunakan Modul ESP32 Berbasis IOT"***

B. Identifikasi Masalah

Berdasarkan latar belakang masalah, maka dapat diidentifikasi masalah sebagai berikut:

1. Pengusaha kerupuk sulit mengeringkan kerupuknya.
2. Cuaca yang tidak menentu pada iklim Indonesia.
3. Pengeringan kerupuk hanya mengandalkan cuaca.
4. Penjemuran kerupuk yang memakan tempat.

C. Batasan Masalah

Agar perancangan dan pembuatan alat ini sesuai dengan konsep awal dan tidak meluas, maka diberikan batasan-batasan sebagai berikut:

1. Penelitian ini hanya terbatas pada pembuatan alat pengering kerupuk.
2. Alat pengering ini hanya terbatas pada proses pengeringan kerupuk yang sudah di buat.
3. Sistem kontrol pengeringan kadar air pada kerupuk yang dilakukan secara realtime.
4. Hasil dari pemantauan dapat dilihat pada aplikasi telegram melalui chat bot.

D. Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat rumusan masalah dari penelitian ini yaitu: “Bagaimana merancang dan membuat Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32”.

E. Tujuan Proyek Akhir

Beberapa tujuan dari pembuatan tugas akhir yaitu, Merancang dan membuat Sistem Kendali Pada Alat Pengering Kerupuk Berbasis IoT Mengunakan Modul ESP32.

F. Manfaat Proyek Akhir

Adapun manfaat dari proyek akhir ini adalah:

1. Untuk menghasilkan alat pengering yang dapat memudahkan pengusaha kerupuk pada saat proses pengeringan kerupuk jika perubahan cuaca tidak stabil.

2. Memudahkan para pengrajin kerupuk untuk mendapatkan kerupuk pada saat cuaca yang tidak mendukung
3. Membantu dalam proses mendapatkan kualitas kerupuk yang bagus.
4. Memperoleh hasil pengeringan yang lebih baik.

BAB II

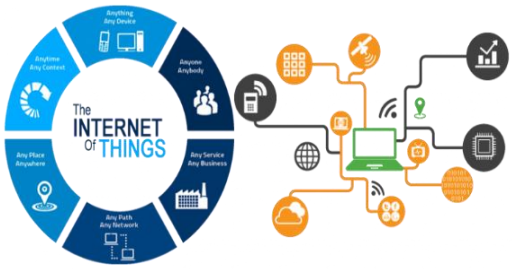
LANDASAN TEORI

A. Internet of Things (IoT)

Internet of Thing's adalah sebuah pemikiran/konsep dimana sebuah benda di pasangkan teknologi-teknologi seperti contohnya sensor dan software dengan target mengendalikan, menginformasikan, mengubungkan serta bertukar informasi atau data memalui internet melalui perangkat lain.

Simulasi dari *Smart Home* dengan Konsep *Internet of Things* (IoT)" sebuah konsep yang memungkinkan kendali sebuah device melalui internet. Adapun hasil dari penggunaan prototype *smart home* dengan konsep *Internet of Things* adalah seseorang dapat mengetahui konsep *smart home* lebih interaktif melalui alat peraga dan menjadi bahan pertimbangan yang baik bagi seorang yang mengembangkan konsep *Smart Home* dan *Internet of Things.*

Cara kerja *Internet of Thing's* adalah memanfaatkan sebuah argumentasi dari algoritma bahasa pemrograman yang telah tersusun. Setiap argumen yang terbentuk akan menghasilkan sebuah interaksi yang akan membantu perangkat keras atau mesin dalam melakukan fungsi atau kerja. Mesin tersebut tidak memerlukan bantuan dari manusia dan dapat dikendalikan secara otomatis. Faktor terpenting dari jalannya program tersebut terletak pada jaringan internet yang menjadi penghubung antar sistem dan perangkat keras. Manfaaat *Internet of Thing's* (IoT) adalah mempermudah dalam melakukan proses konektivitas perangkat atau alat.



Gambar 1. Ilustrasi *Internet Of Thing's*

B. Modul ESP32

ESP32 dibuat oleh Espressif Systems, perusahaan berbasis di Shanghai, Tiongkok. Menggunakan TSMC sebagai pemproduksi inti dengan besar 40 nm. mikrokontroller berharga rendah dan hemat energi dengan wifi dan dual-mode bluetooth terintegrasi. Generasi ESP32 menggunakan mikroprosesor Tensilica Xtensa LX6 sebagai inti. Baik dalam mode single-core maupun dual-core, juga sebagai pengembang dari SoC ESP8266 yang terkenal dengan NodeMCU.

ESP32 adalah penerus SoC ESP8266 dengan menggunakan Mikroprosesor Xtensa LX6 32-bit Tensilica dengan Wi-Fi dan *Bluetooth* yang terintegrasi. Pada mikrokontroler ini sudah tersedia modul WiFi dan *Bluetooth* dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. ESP32 memiliki fitur yang cukup lengkap karena mendukung input/output Analog dan Digital, PWM, SPI, I2C, dll. (Iqbal, 2022)

Hal yang baik tentang ESP32, seperti ESP8266 adalah komponen RF terintegrasi seperti *Power Amplifier*, *Low-Noise Receive Amplifier*, Antena Switch,

dan Filter. Hal ini membuat perancangan hardware pada ESP32 menjadi sangat mudah karena hanya memerlukan sedikit komponen eksternal.

1. Spesifikasi ESP32

   ESP32 memiliki lebih banyak fitur daripada ESP8266. Memulai dengan ESP32 ini. Berikut ini daftar beberapa spesifikasi penting dari ESP32. Tetapi untuk spesifikasi lengkap, dapat melihat pada Datasheet.

   a. Mikroprosesor LX6 Single atau Dual-Core 32-bit dengan frekuensi clock hingga 240 MHz.
   b. 520 KB SRAM, 448 KB ROM dan 16 KB RTC SRAM.
   c. Mendukung konektivitas Wi-Fi 802.11 dengan kecepatan hingga 150 Mbps.
   d. Dukungan untuk spesifikasi Bluetooth v4.2 dan BLE klasik.
   e. 34 GPIO yang Dapat Diprogram.
   f. Hingga 18 saluran SAR ADC 12-bit dan 2 saluran Konektivitas Serial DAC 8-bit mencakup 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
   g. Ethernet MAC untuk Komunikasi LAN fisik (memerlukan PHY eksternal).
   h. 1 Host controller untuk SD/SDIO/MMC dan 1 Slave controller untuk SDIO/SPI.
   i. Motor PWM dan hingga 16-saluran LED PWM.

2. Bagian-bagian ESP32

   a. Modul ESP-WROOM-32
   b. Dua baris Pin IO (dengan 15 pin di setiap sisi)
   c. CP2012 USB – IC Jembatan UART

d. Konektor micro–USB (untuk daya dan pemrograman)

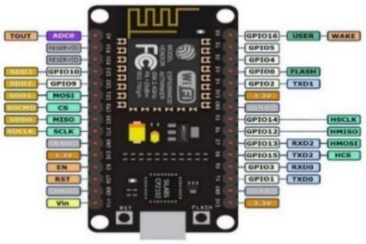
e. AMS1117 3.3V Regulator IC

f. Tombol Aktifkan (untuk Reset)

g. Tombol Boot (untuk berkedip)

h. LED Daya (Merah)

i. LED Pengguna (Biru – terhubung ke GPIO2)



Gambar 2. NodeMCU ESP32

C. IDE Arduino

IDE itu merupakan kependekan dari *Integrated Development Environment*, 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 3. Tampilan Awal IDE Arduino

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.

Gambar 4. Tampilan Software IDE Arduino

Berikut adalah perintah dan menu eksekusi serta proses yang ada pada IDE Arduino:

1. *Verify/Compile*

   Biasakan untuk memverifikasi terlebih dahulu *sketch* yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses *Verify/Compile* mengubah *sketch* ke *binary code* untuk diupload ke mikrokontroler.

2. *Upload*

   Tombol ini berfungsi untuk menguploat sketch ke board Arduino. Walaupun kita tidak mengklik tombol verify, maka sketch akan di-compile, kemudian langsung diupload ke board.

3. *New Sketch*

   Membuka *window* dan membuat *sketch* baru.

4. *Open Sketch*

   Membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi *file* .ino.

5. Save *Sketch*

   Menyimpan *sketch*, tapi tidak disertai dengan mengkompile.

6. *Serial Monitor*

   Membuka *interface* untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.

7. Keterangan Aplikasi

   Pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal "Compiling" dan "*Done Uploading*" ketika kita meng*compile* dan menguploat sketch ke *board* Arduino.

8. *Konsol*

   Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang *sketch* akan muncul pada bagian ini. Misalnya, Ketika aplikasi meng*compile* atau

ketika ada kesalahan pada *sketch* yang kita buat, maka informasi *error* dan baris akan diinformasikan di bagian ini.

D. Algoritma

Algoritma adalah sistem kerja komputer memiliki brainware, hardware, dan software. Tanpa salah satu dari ketiga sistem tersebut komputer tidak akan berguna. Software terbangun atas susunan program untuk membangun diperlukan langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan software.

Pengertian algoritma adalah susunan logis dan sistematis untuk memechakan suatu masalah atau untuk mencapai tujuan tertentu. Persamaan algoritma dan flowchart adalah merincikan suatu proses dari awal sampai akhir. Sedangkan perbedaan algoritma dan flowchart adalah kalau algoritma menjelaskan suatu proses tanpa menggunakan simbol atau disajikan dengan kode pemrograman sedangkan flowchart menjelaskan suatu proses menggunakan simbol-simbol tertentu atau disajikan dalam bentuk diagram. masing-masing diagram memiliki makna tersendiri, seperti input, proses, output, dan lain sebagainya.

1. Jenis-Jenis Algoritma

   Terdapat beberapa klasifikasi algoritma berdasarkan paradigma dan metode yang digunakan dalam perancangan algoritma tersebut. Paradigma yang digunakan untuk menyusun suatu algoritma, beberapa diantaranya sebagai berikut:

a. *Divide* and *conquer*, merupakan paradigma untuk membagi suatu permasalahan yang besar menjadi permasalahan-permasalahan yang kecil. Pembagian masalah ini dilakukan secara trus-menerus sampai ditemukan bagian masalah yang kecil dan mudah untuk dipecahkan.

b. *Dynamic Programming*, paradigma pemrograman dinamik akan sesuai jika digunakan pada suatu masalah yang mengandung sub-struktur yang optimal dan mngandung beberapa bagian permasalahan yang tumpang tindih. Paradigma ini seklias terlihat mirip dengan paradigma divide andconquer, sama-sama mencoba untuk menbagi permasalahan menjadi sub permasalahan yang lebih kecil, tapi secara intrinsic ada perbedaan dari karakter permasalahan yang dihadapi.

c. Metode serakah, merupakan paradigma yang mirip dengan pemrograman dinamik, namun jawaban dari setiap submasalah tidak perlu diketahui dari setiap tahap, dan menggunakan pilihan apa yang terbaik pada saat ini.

2. Keuntungan Memakai Algoritma

a. Pembuatan atau penulisan algoritma tidak tergantung pada bahasa pemrograman manapun, artinya penulisan algoritma independen dari bahasa pemrograman dan komputer yang melaksanakannya.

b. Notasi algoritma dapat diterjemahkan ke berbagai bahasa pemrograman.

c. Apapun bahasa pemrogramannya, output yang akan dikeluarkan sama karena algoritmanya sama.

3. Hal yang Perlu Diperhatikan dalam Algoritma

   a. Teks algoritma berisi deskripsi langkah-langkah penyelesaian masalah. Deskripsi tersebut dapat ditulis dalam notasi apapun asalkan mudah dimengerti dan dipahami.

   b. Tidak ada notasi yang baku dalam penulisan teks algoritma seperti notasi bahasa pemrograman. Notasi yang digunakan dalam menulis algoritma disebut notasi algoritmik.

   c. Setiap orang dapat membuat aturan penulisan dan notasi algoritmik sendiri. Hal ini dikarenakan teks algoritma tidak sama dengan teks program. Namun, supaya notasi algoritmik mudah ditranslasikan ke dalam notasi bahasa pemrograman tertentu, maka sebaiknya sebuah notasi algoritmik tersebut berkorespondensi dengan notasi bahasa pemrograman secara umum.

   d. Notasi algoritmik bukan notasi bahasa pemrograman, karena itu pseudocode dalam notasi algoritmik tidak dapat dijalankan oleh komputer. Agar dapat dijalankan oleh komputer, pseudocode dalam notasi algoritmik harus ditranslasikan atau diterjemahkan ke dalam notasi bahasa pemrograman yang dipilih.

   e. Algoritma sebenarnya digunakan untuk membantu kita dalam mengkonversikan suatu permasalahan ke dalam bahasa pemrograman agar mudah direalisasikan.

f. Algoritma merupakan hasil pemikiran konseptual, supaya dapat dilaksanakan oleh komputer, algoritma harus ditranslasikan ke dalam notasi bahasa pemrograman. Ada beberapa hal yang harus diperhatikan pada translasi tersebut, yaitu:

1) Pendeklarasian Variable

Untuk mengetahui dibutuhkannya pendeklarasian variabel dalam penggunaan bahasa pemrograman apabila tidak semua bahasa pemrograman membutuhkannya.

2) Pemilihan Tipe Data

Apabila bahasa pemrograman yang akan digunakan membutuhkan pendeklarasian variabel maka perlu hal ini dipertimbangkan pada saat pemilihan tipe data.

3) Pemakaian instruksi-instruksi

Beberapa instruksi mempunyai kegunaan yang sama tetapi masing-masing memiliki kelebihan dan kekurangan yang berbeda.

4) Aturan sintaksis

Pada saat menuliskan program kita terikat dengan aturan sintaksis dalam bahasa pemrograman yang akan digunakan.

5) Tampilan hasil

Pada saat membuat algoritma kita tidak memikirkan tampilan hasil yang akan disajikan. Hal-hal teknis ini diperhatikan ketika mengkonversikannya menjadi program.

4. Ciri-ciri Algoritma

Menurut Abdul Kadir Heriyanto (2005:6) ada lima ciri-ciri penting yang harus dimiliki sebuah algoritma yaitu :

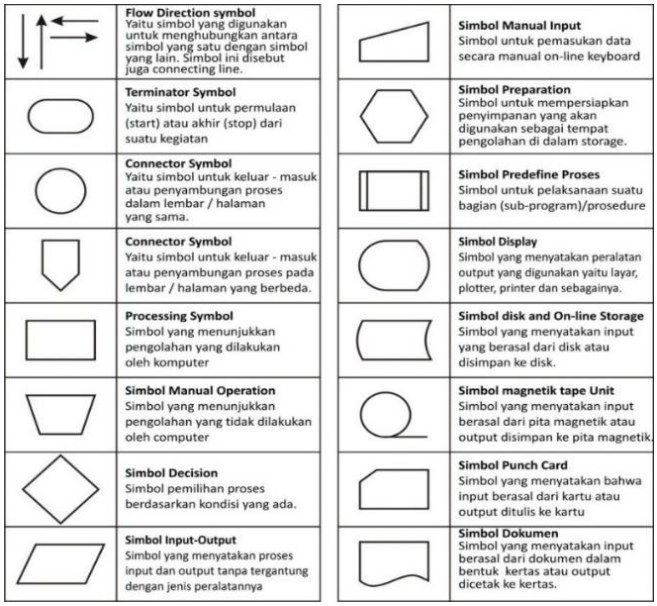
1. *Finiteness*, menyatakan bahwa suatu algoritma harus berakhir untuk semua kondisi setelah memproses sejumlah langkah.
2. *Defineteness*, menyatakan bahwa setiap langkah harus dinyatakan dengan jelas (tidak rancu).
3. Masukan, setiap algoritma dapat tidak memiliki masukan atau mempunyai satu atau beberapa masukan. Masukan merupakan suatu besaran yang diberikan di awal sebelum algoritma diproses.
4. Keluaran, setiap algoritma memiliki keluaran, baik itu satu keluaran atau banyak keluaran. Keluaran merupakan besaran yang mempunyai kaitan atau hubungan dengan masukan.
5. Efektivitas, setiap algoritma diharapkan bersifat efektif, artinya semua operasi yang dilaksanakan algoritma harus sederhana dan dapat dikerjakan dalam waktu yang terbatas. Secara prinsip, setiap intruksi dalam algoritma dapat dikerjakan oleh orang dengan hanya menggunakan kertas dan pensil.

E. Flowchart

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau pengembaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* membantu analisis

dan programmer untuk memecahkan masalah dalam segmen-segmen yang lebih kecil dan membantu dalam menganalisis alternatif lain dalam pengoperasian.

Sistem *flowchart* adalah urutan proses dalam sistem dengan menunjukkan alat media *input*, *output* serta jenis media penyimpanan dalam proses pengolahan data. Program *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (interuksi) dengan proses lainnya dalam suatu program.



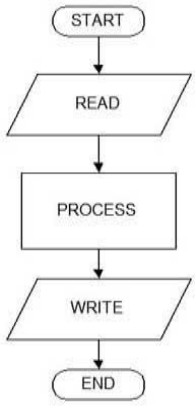
Gambar 5. Simbol-Simbol Flowchart

Dalam pembuatan *flowchart* tidak ada rumus atau patokan yang bersifat mutlak. Karena *flowchart* merupakan gambaran hasil pemikiran dalam menganalisa suatu masalah dengan komputer. Sehingga *flowchart* yang dihasilkan dapat bervariasi antara satu pemrogram dengan pemrogram lainnya. Namun secara garis besar, setiap pengolahan selalu terdiri dari tiga bagian utama, yaitu:

1. *Input* berupa bahan mentah.
2. Proses pengolahan.
3. *Output* berupa bahan jadi.

Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu:

1. *Start*: berisi instruksi untuk persiapan perlatan yang diperlukan sebelum menangani pemecahan masalah.
2. *Read*: berisi instruksi untuk membaca data dari suatu peralatan *input*.
3. *Process*: berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
4. *Write*: berisi instruksi untuk merekam hasil kegiatan ke perlatan *output*.
5. *End*: mengakhiri kegiatan pengolahan.



Gambar 6. Flowchart Dasar

Dari gambar 3 terlihat bahwa suatu *flowchart* harusterdapat proses persiapan dan proses akhir. Dan yang menjadi topik dalam pembahasan ini adalah tahap proses. Karena kegiatan ini banyak mengandung variasi sesuai dengan

kompleksitas masalah yang akan dipecahkan. Walaupun tidak ada kaidah-kaidah yang baku dalam penyusunan *flowchart*, namun ada beberapa anjuran yaitu:

1. Hindari pengulangan proses yang tidak perlu dan logika yangberbelit sehingga jalannya proses menjadi singkat.
2. Penggambaran *flowchart* yang simetris dengan arah yang jelas.
3. Sebuah *flowchart* diawali satu titik *START* dan diakhiri dengan *END*.

F. Bahasa Pemograman C Arduino

Luh Joni Erawati Dewi (2010), bahasa pemrograman C/C++ merupakan bahasa pemrograman tingkat tinggi.

1. Struktur

   Setiap program Arduino (biasa disebut sketch) mempunyai dua buah fungsi yang harus ada. (Abdul Kadir 2003:3)

   a. void setup () { }

      Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya. Biasanya ini digunakan pada awal program.

   b. void setup () {//

      Semua kode yang disini akan dibaca sekali oleh Arduino.

   c. loop () { }

      Fungsi ini akan dijalankan setelah setup (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

d. void setup () {//

Semua kode yang ada disini akan dibaca berulang kali oleh Arduino.

2. Elemen Bahasa C yang dibutuhkan dalam format penulisan

a. // (komentar satu baris): Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

b. /* */ (komentar banyak baris): Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

c. { } (kurung kurawal): Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir, digunakan juga pada fungsi dan pengulangan.

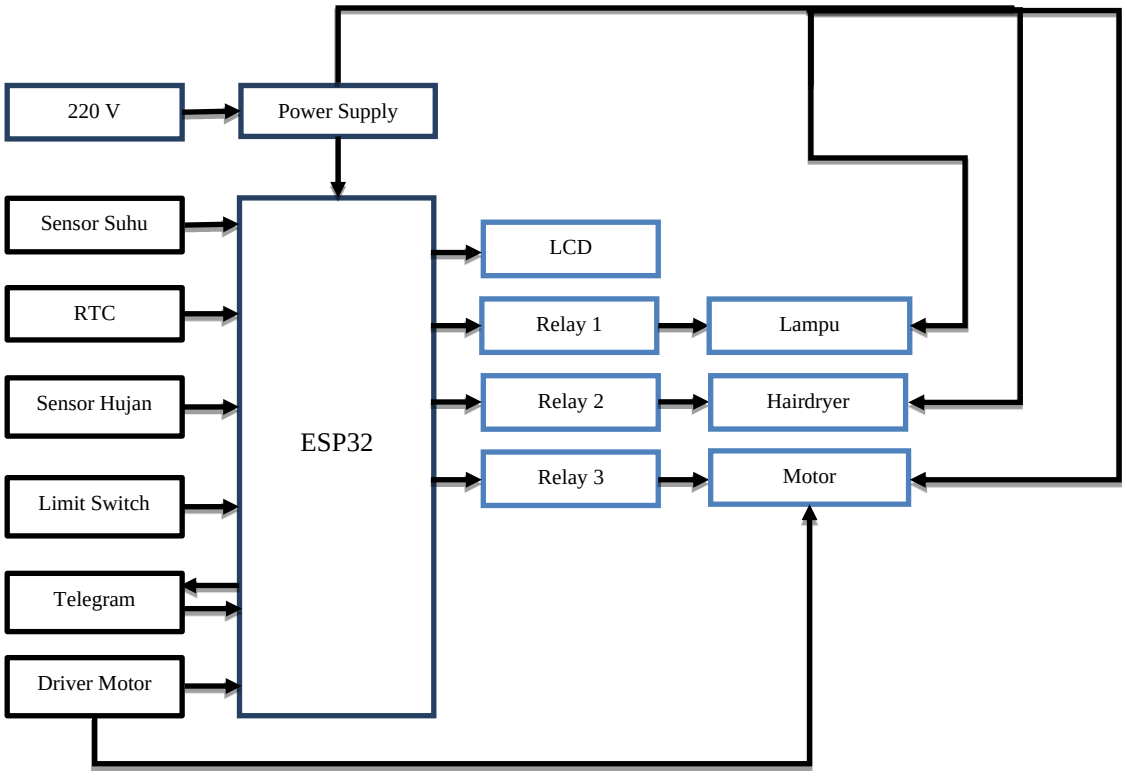
d. ; (titk koma): Setiap baris kode harus diakhiri dengan tanda titik koma, jika ada titik koma yang hilang maka program tidak akan bisa dijalankan.

BAB III

METODE PERANCANGAN DAN PEMBUATAN PROGRAM

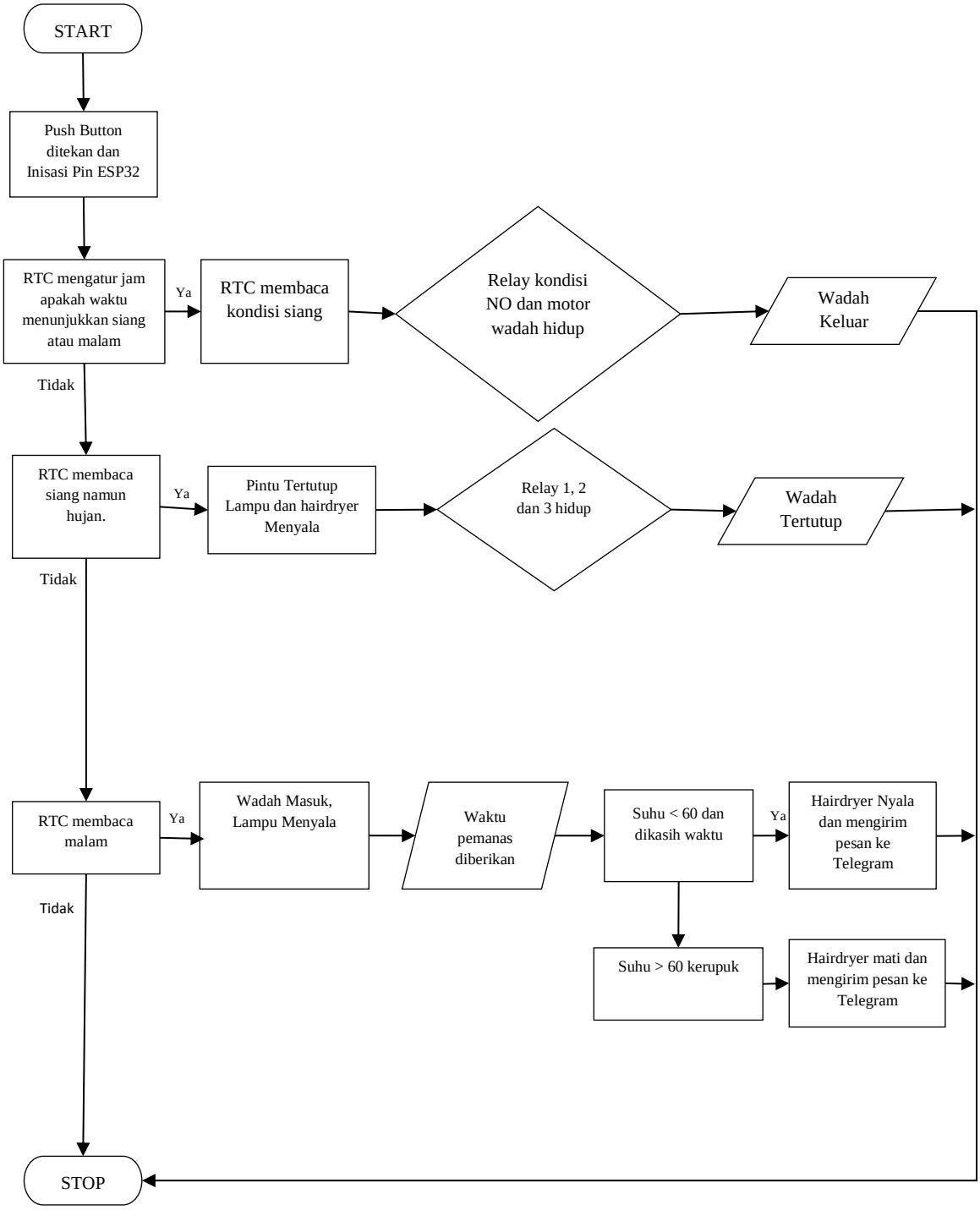
A. Diagram Blok Sistem

Perancangan dan pembuatan sistem ini dibutuhkan blok diagram untuk menjelaskan sistem secara keseluruhan. Setiap blok memiliki fungsi tertentu yang saling terkait sehingga membentuk sistem dari perangkat yang dibuat. Adapun skema blok diagram hardware sebagai berikut:



Gambar 7. Blok Diagram

B. Flowchart



Gambar 8. Flowchart Sistem Kerja Alat

Penjelasan flowchart diatas berfungsi untuk melihat kondisi yang akan digunakan alat yang akan dirancang. Berikut penjelasan dari flowchart diatas yaitu:

1. RTC melakukan set waktu pembagian siang dan malam untuk menentukan bisa melakukan pengeringan secara manual.
2. Wadah akan terbuka pada kondisi siang hari dengan limit switch sebagai pembatas, untuk pengatur motor power window oleh driver motor LM35.
3. Ketika tidak terdeksi hujan, maka proses pengeringan kerupuk dengan menggunakan sinar matahari.
4. Penentuan malam dan siang tergantung dari pengaturan RTC, karena kita mengatur siang dan malam dengan RTC.
5. Apabila suhu pada proses pengeringan kerupuk sudah melewati jam yang ditentukan, maka alat otomatis akan mengirimkan pesan ke telegram bahwa kerupuk sudah kering hairdryer akan mati, tapi ketika belum cukup proses pemanasan bakal tetap berlangsung.
6. Kondisi yang berubah akan dikirim ke user oleh ESP32 melalui bot telegram

C. Prinsip Kerja Sistem

Pada alat pengering kerupuk ini berlandasan Mikrokontroller ESP32 sebagai pengontrol utama dengan cara kerja alatnya. Langkah pertamanya adalah menekan *push button power* untuk mengaktifkan alat dengan cara memberi sumber daya secara keseluruhan blok rangkaian melalui power supply. Pada *push button* ditekan blok diagram langsung aktif keseluruhannya serta LCD display akan menampilkan kalimat “Selamat Datang” kemudian ESP32 meghubungkan ke wifi yang sudah di

diatur di program, diwaktu yang bersmaan LCD menampilkan “Menghubungkan ke Telegram”. Pada saat itu juga ESP32 juga menghubungkan ke telegram, saat sudah terhubung maka LCD menampilkan “Wi-Fi sudah terhubung” setelah itu tugas ESP32 yaitu menghubungkan dengan Telegram, hampir sama dengan Langkah sebelumnya, LCD akan menampilkan “Menghubungkan dengan bot Telegram”, ketika sudah terhubung maka LCD menampilkan “Bot sudah terhung dan otomatis bot menerima pesan. Karena alat ini tidak memakai sensor cahaya untuk mengetahui siang dan maka kita menggunakan RTC. alat ini akan diletakkan pada ruangan terbuka pada saat siang hingga malam hari, RTC akan mengatur pembukaan alat dengan melihat pengaturan jam yang sudah diatur.

Pada saat siang hari. Lampu, dan *Hairdryer* tidak akan aktif karna pada siang hari pengeringan kerupuk dilakukan dengan cara dijemur di dalam wadah alat pengering kerupuk. Namun ketika hujan saat siang hari maka wadah alat pengering kerupuk tersebut akan otomatis masuk ke dalam alat pengering kerupuk tersebut, dan pada saat sensor hujan tidak mendeteksi hujan, maka wadah alat pengering kerupuk akan otomatis keluar kembali. Pada saat waktu malam yang sudah diatur oleh RTC. wadah akan masuk otomatis ke dalam alat pengering kerupuk lalu hairdryer, dan lampu akan hidup, apabila suhu di dalam alat pengering kerupuk kurang panas untuk mengeringkan kerupuk yang terbaca oleh sensor DHT11. Maka hairdryer akan hidup, dan lampu akan hidup juga.

Pembukaan wadah dan penutupan wadah dikontrol oleh limit switch sebagai sinyal input, jadi Ketika limit switch 1 kondisi ditekan maka wadah yang dikontrol

oleh *motor power window* berhenti menutup ke dalam, sedangkan jika limit switch 2 kondisi ditekan maka kondisi sebaliknya, wadah akan berhenti menutup ke luar, disini memakai sistem IoT maka pemantauan aktivitas alat dapat diketahui dari Smartphone pada Aplikasi Telegram. Jadi di Aplikasi Telegram akan dibuat bot khusus untuk memantau dan mengirimkan perintah pada alat seperti contoh malam hari, cuaca hujan dan kerupuk yang sudah kering atau matang penjemuran.

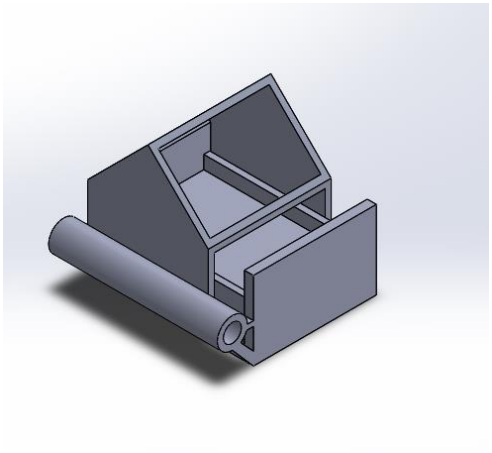
D. Perancangan Software

Dalam perancangan software, yang dahulu dilakukan adalah dengan membuat diagram blok bagaimana alat akan bekerja. Setelah diagram blok selesai kita merancang flowchart bagaimana sistematis alat bekerja dan bagaimana alur alat.

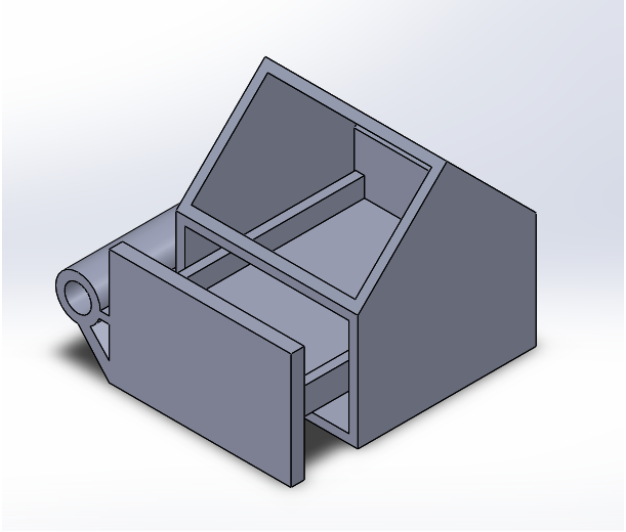
Flowchart adalah bagian terpenting dalam merancang sebuah perangkat lunak, jadi awal dari sebuah alat adalah merancang blok diagram dan flowchart. Sesudah merancang 2 hal tersebut, selanjutnya menulis codingan untuk dimasukkan ke dalam ESP32, yang diperlukan untuk mebuat barisan coding adalah Laptop, Software Arduino IDE dan beberapa file tambahan untuk komponen.

E. Pengujian Alat

NodeMCU ESP32 adalah hardware yang dikendalikan dengan software, untuk hardware sudah dijelaskan mengenai jenis-jenisnya, karakteristik dan konfigurasi pin yang tersedia, untuk software, ESP32 menggunakan software sendiri dan bahasa pemprograman sendiri yang dinamakan Arduino IDE, bahasa pemprograman Arduino yang dipakai yaitu C/C++. Kemudian kita perlu menginstall board ESP32 agar bisa membuat program ESP32.



Gambar 9. Tampak alat dari depan



Gambar 10. Tampak alat dari samping

1. Install Arduino IDE

   Langkah awal adalah dengan mendownload software Arduino IDE pada website resmi mereka pada *https://www.arduino.cc/en/software* pilih yang sesuai dengan versi komputer, kemudian jalankan file yang didownload tadi kemudian install seperti biasa. Ketika sudah selesai, maka Arduino IDE sudah berhasil terinstall pada komputer Anda.

2. Setting Arduino IDE

   a. Menambahkan Board ESP32

      Klik menu File, kemudian pilih preferences, Pada Additional Board Manager URLs tambahkan link berikut:

      https://dl.espressif.com/dl/package_esp32_index.json

      Kemudian jika sudah, klik *Tools > Board > Boards Manager*

      Pada input pencarian ketik ESP32, pilih versi terbaru yang tersedia, kemudian install.

   b. Menghubungkan Mikrokontroler ke PC

      Tentukan serial port yang dipakai, ini digunakan untuk memprogram mikrokontroler, jika belum tahu serial port berapa yang dipakai dapat dilihat di device manager > Ports, lalu cek nomer COM-nya, pada contoh tertulis COM5 artinya Port komunikasi yang digunakan adalah COM5, ini bisa berubah-ubah, maka dari itu pastikan dahulu COM yang dipakai sesuai dengan setting Arduino, setelah mengetahui COM yang digunakan, kembali ke Arduino IDE, Pilih Tools Serial Port COM5. Jika berhasil maka pada bagian editor bawah akan tampak COM yang dipakai.

   c. Upload Program ke IDE Arduino

      Setelah proses setting, selanjutnya meng-upload program yang sudah kita buat, pada contoh ini menggunakan program kosong lalu di upload ke Arduino, jika berhasil maka akan ada keterangan berhasil, namun jika error

maka di bawah editor aka tertulis error. Langkah-langkahnya seperti berikut:

a) Langkah 1

Tuliskan sebuah program pada editor yang mana program tersebut akan dieksekusi oleh Arduino. Berikut contoh beberapa contoh program:

1) Program LCD

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup() {
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("hello, world!");
}
void loop() {
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
lcd.setCursor(0, 1);
// print the number of seconds since reset:
lcd.print(millis() / 1000);
}
```

2) Program RTC

```
if (! rtc.begin()) {
Serial.println("Could not find RTC! Check circuit.");
while (1); }
rtc.adjust(DateTime(__DATE__, __TIME__));
}
```

3) Program Bot Telegram

```
void loginTelegram(){
lcd.setCursor(0,0);
lcd.print("Menghubungkan Ke");
```

```
lcd.setCursor(0,1);
lcd.print("  Telegram Bot  ");
Serial.println("login in telegram...");
while (!myBot.testConnection())  {
      myBot.setTelegramToken(token);
      Serial.print(".");
      delay(100);}
if(myBot.testConnection()){
     Serial.println("Telegram connection OK!");
     lcd.setCursor(0,0);
     lcd.print("  Telegram Bot  ");
     lcd.setCursor(0,1);
     lcd.print("   Terhubung    ");
```

Lanjutan program Bot Telegram ada pada Lampiran 1.

4) Rain Sensor

```
 if (pagiCerah==1){
      lcd.clear();
      lcd.setCursor(0,0); lcd.print("Cuaca Cerah     ");
      lcd.setCursor(0,1); lcd.print("Pukul");
      lcd.setCursor(6,1); lcd.print(jam);
      lcd.setCursor(8,1); lcd.print(":");
      lcd.setCursor(9,1); lcd.print(menit);
      lcd.setCursor(11,1); lcd.print(":");
      lcd.setCursor(12,1); lcd.print(detik);
      myBot.sendMessage(id, "Cuaca Cerah ⛅
\n\n"+String(hari)+", Pukul "+String(jam)+" :
"+String(menit)+" : "+String(detik)+" WIB.\nAtap Telah
dibuka!! 😇");
```

Lanjutan Program Rain Sensor ada pada Lampiran 1.

5) Motor Driver

```
if (jam>=7 && jam<18 && hujan==1){
   motor=1; pagiCerah++;      //membuka
 } else if ((jam<7 || jam>=18) && hujan==1){
   motor=2; malamCerah++;     //menutup
 } else if (jam>=7 && jam<18 && hujan==0){
   motor=2; pagiHujan++;      //menutup
 } else if ((jam<7 || jam>=18) && hujan==0){
   motor=2; malamHujan++;     //menutup
 } else {motor=0; }
```

6) Sensor Suhu

```
 if(suhu<=55){
   digitalWrite(blower,LOW);
   digitalWrite(fan,LOW);
  } else if (suhu>55){
   digitalWrite(blower,HIGH);
   digitalWrite(fan,HIGH);
  }
 } else{
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("Suhu : ");
  lcd.setCursor(8,0); lcd.print(String(suhu));
  lcd.setCursor(14,0); lcd.write(Simbol_derajat);
  lcd.setCursor(15,0); lcd.print("C");
  lcd.setCursor(0,1); lcd.print("Pemanas OFF     ");

  if(x>=1){future = now;}
```

7) Limit Switch

```
   if (LSwitch1==0 || LSwitch2==0){
    motor=0;
   }

   switch(motor){
    case 0 : //diam
     digitalWrite(motor1a,LOW);
     digitalWrite(motor1b,LOW);
     break;
```

8) Relay

```
 if(suhu<=55){
   digitalWrite(blower,LOW);
   digitalWrite(fan,LOW);
  } else if (suhu>55){
   digitalWrite(blower,HIGH);
   digitalWrite(fan,HIGH);
  }
```

b) Langkah 2

Lalu upload program dengan cara yaitu tekan tombol *verify* kemudian jika tidak ada *error*, tekan tombol upload lalu tunggu sampai proses selesai, Ketika sudah selesai upload program bisa di check apakah program berjalan normal pada hardware dengan cara check sinyal pada serial monitor.

BAB IV

PENGUJIAN DAN PEMBAHASAN

Suatu alat atau program dapat dikatakan bekerja dengan baik apabila sudah dilakukan pengujian sesuai dengan fungsi dari alat tersebut. Pengujian bermaksud untuk mendapatkan evaluasi terhadap program-program yang telah dikerjakan agar mendapatkan kinerja yang lebih baik dengan melakukan perbaikan terhadap program yang mengalami kekurangan saat melakukan pengujian.

Pada bab ini penulis akan menjelaskan cara pengujian pada perangkat lunak (software) seperti bagian program dan program utama. Pengujian ini dilakukan untuk mendapatkan data serta bukti hasil akhir dari kenyataan bahwa perangkat lunak yang telah dibuat bisa bekerja dengan baik dan juga dapat digabungkan dengan perangkat keras (hardware).Setelah perancangan dan pembuatan alat selesai, maka langkah selanjutnya yang dilakukan ialah melakukan pengujian alat menggunakan program keseluruhan untuk mengetahui apakah alat yang dibuat sesuai dengan yang diharapkan, kemudian dilakukan pengukuran pada parameter dengan menggunakan program yang telah dibuat sebelumnya.

A. Pengujian Software

Pengujian merupakan salat satu langkah yang penting yang harus dilakukan untuk mengetahui apakah sistem yang kita buat telah sesuai dengan yang direncanakan dalam pengujian sistem. Selain mengetahui apakah program bekerja

dengan baik sesuai dengan yang diharapkan, pengujian bertujuan untuk melihat kekurangan dari sistem yang telah dibuat. Program merupakan bagian utama dari sebuah alat yang akan digunakan untuk mengontrol mikrokontroler yang akan menggerakkan alat.

1. Deklarasi dan Konfigurasi Program

   Bagian awal dari sebuah program harus terdapat deklarasi dan konfigurasi. Pada bagian deklarasi dan konfigurasi semua unit akan diolah *compiler* dan mikrokontroler harus dinyatakan dalam bentuk Bahasa pemrograman yang sesuai dengan kaidah dan aturan pemrograman yang digunakan, karena *compiler* dan mikrokontroler hanya akan mengeksekusi program sesuai dengan deklarasi dan konfigurasi yang ditulis di awal program. Berikut deklarasi dan konfigurasi dari program dalam proyek akhir ini.

```
#include <WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#include "RTClib.h"
#include "CTBot.h"
```

   Kode program diatas merupakan *Library* yang akan digunakan untuk menjalankan beberapa sensor.

2. Pengujian LCD

   Pengujian LCD dilakukan untuk memastikan agar LCD menampilkan kalimat, kata atau angka sesuai intruksi yang diketikkan pada program.

```
#include <LiquidCrystal_I2C.h>
```

Kode program diatas digunakan untuk menyertakan *Library* LCD pada program Arduino, 12C merupakan tipe IC yang diintegrasi ke LCD agar penggunaan pin Arduino tidak banyak.

```
#include <Wire.h>
```

Kode program diatas digunakan untuk pemanggilan *Library* komunikasi serial I2C.

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

Kode program di atas mendefinisikan *Library* LCD sebanyak 27 kotak dengan ukuran 16 cm x 2 cm.

```
if (pagiCerah==1){
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("Cuaca Cerah     ");
    lcd.setCursor(0,1); lcd.print("Pukul");
    lcd.setCursor(6,1); lcd.print(jam);
    lcd.setCursor(8,1); lcd.print(":");
    lcd.setCursor(9,1); lcd.print(menit);
    lcd.setCursor(11,1); lcd.print(":");
    lcd.setCursor(12,1); lcd.print(detik);
    myBot.sendMessage(id, "Cuaca Cerah ⛅ \n\n"+String(hari)+", Pukul "+String(jam)+" : "+String(menit)+" : "+String(detik
    malamCerah=0;pagiHujan=0; malamHujan=0;
```

Kode yang menampilkan kondisi cerah, pada LCD akan menampilkan cuaca, jam, menit dan detik. setCursor berfungsi sebagai lokasi dimana teks yang ditulis ke LCD akan ditampilkan.

3. Pengujian Sensor Suhu (DHT 11)

```
#include "DHT.h"
```

Kode diatas diatas digunakan untuk menyertakan *Library* sensor suhu (DHT11) pada program Arduino. Sensor suhu disini berfungsi sebagai sensor untuk mengatur agar suhu ruangan tetap di 60 dan mengatur kelembapan suhu pada ruang pengeringan.

```
if(suhu<=55 && counter1==0 && ((future - now).hours()>0 || (future - now).minutes()>0 || (future - now).seconds()>0 )){
  digitalWrite(blower,LOW);
  digitalWrite(fan,LOW);
  lcd.setCursor(0,0); lcd.print("  Suhu  Rendah  ");
  lcd.setCursor(0,1); lcd.print("   Pemanas ON   ");
  myBot.sendMessage(id, "Suhu Rendah! ❄\nMenyalakan Pemanas!");
  statusBlow = 1; counter1=1; counter2=0;
} else if (suhu>55 && counter2==0){
  digitalWrite(blower,HIGH);
  digitalWrite(fan,HIGH);
  lcd.setCursor(0,0); lcd.print("  Suhu  Tinggi  ");
  lcd.setCursor(0,1); lcd.print("  Pemanas OFF   ");
  myBot.sendMessage(id, "Suhu Terlalu Tinggi!🔥 \nMematikan Pemanas! ");
  statusBlow = 0; counter1=0; counter2=1;
```

Kode diatas berfungsi mendeteksi suhu dan langsung menampilkan nilai sensor ke LCD, seperti pada perintah diatas, jika suhu dibawah 55 ℃, maka hairdyer tidak hidup, dan apabila waktu hitung mundur ada, maka hairdryer akan hidup juga, setCursor berfungsi menampilkan tulisan ke LCD.

4. Pengujian *Real Time Clock* (RTC)

Pengujian modul ini dilakukan untuk mengetahui kinerja dari modul RTC tersebut.

```
#include "RTClib.h"
```

Kode diatas berfungsi sebagai library dari *Real Time Clock* (RTC).

```
if (! rtc.begin()) {
  Serial.println("Could not find RTC! Check circuit.");
  while (1); }

rtc.adjust(DateTime(__DATE__, __TIME__));
```

Kode diatas berfungsi untuk memulai RTC, *rtc.adjust* untuk mengatur waktu sesuai dengan waktu yang sekarang (realtime). Pada RTC kita bisa mengatur waktu dari hairdyer dan jangka waktu berapa hairdyer hidup.

Pada Program diatas, RTC bisa menyesuaikan jam dari internet karena menuliskan perintah rtc.adjust.

```
kbd.addButton("15 Menit", "m15", CTBotKeyboardButtonQuery);
kbd.addButton("30 Menit", "m30", CTBotKeyboardButtonQuery);
kbd.addButton("45 Menit", "m45", CTBotKeyboardButtonQuery);
kbd.addRow();
kbd.addButton("1 Jam", "j1", CTBotKeyboardButtonQuery);
kbd.addButton("2 Jam", "j2", CTBotKeyboardButtonQuery);
kbd.addButton("3 Jam", "j3", CTBotKeyboardButtonQuery);
kbd.addButton("5 Jam", "j5", CTBotKeyboardButtonQuery);
ya.addButton("YAA!!", "yes", CTBotKeyboardButtonQuery);
ya.addButton("TIDAK", "no", CTBotKeyboardButtonQuery);
```

Pada program diatas, kita mengatur waktu hidup hairdyer selama 15 menit, 30 menit, 45 menit, 1 jam, 2 jam, 3 jam, dan 5 jam, bisa ditambahkan sesuai dengan kebutuhan.

5. Pengujian Connect Wifi

```
#include <WiFi.h>
```

Kode diatas sebagai *Library* untuk konektivitas mikrokontroler ke jaringan wifi. Wifi disini berfungsi untuk menghubungkan ESP32 dengan internet, agar bisa berkomunikasi antara ESP32 dengan user melalui Telegram.

```
#define ssid      "OPPO A17"
#define password  ""
```

Kode diatas #define sebagai distribusi untuk variable tanpa memakan memori. Fungsi dari koding diatas menyambungkan ke jaringan SSID wifi dan password dari wifi tersebut.

```
void connectWifi(){
    Serial.println("Connecting To Wifi");
    lcd.setCursor(0,0);
    lcd.print("   Connecting   ");
    lcd.setCursor(0,1);
    lcd.print("    To  Wifi    ");
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
```

Kode diatas menampilkan status konektivitas jaringan ke LCD. Ketika wifi baru menyambungkan, maka LCD akan menampilkan “Connecting To Wifi”, “Connecting” berarti sedang menyambungkan ke wifi.

```
Serial.println("Wifi Connected");
Serial.println(WiFi.SSID());
Serial.println(WiFi.RSSI());
Serial.println(WiFi.macAddress());
Serial.println(WiFi.localIP());
Serial.println(WiFi.gatewayIP());
Serial.println(WiFi.dnsIP());
lcd.setCursor(0,0);
lcd.print("     WiFi       ");
lcd.setCursor(0,1);
lcd.print("   Connected    ");
delay(1500);
```

Kode diatas menampilkan konektivitas wifi. Yaitu Ketika wifi terhubung, maka akan akan tampil pada layar LCD “Wifi Connected” ESP32 otomatis mendapatkan macAddres. LocalIP, GatewayIP, dnsIP.

Pada line program diatas dapat dilihat bahwa program membaca SSID, RSSID, MacAdrdress, Local IP, gateway IP untuk menyambungkan dengan perangkat seperti hotspot, Acces Point ataupun perangkat lain yang memancarkan sinyal Wi-Fi.

Pada line program diatas, untuk semuanya seperti SSD, RSSI dan seterusnya sudah terisi otomatis ketika ESP32 tersambung ke WiFi yang sudah disetting. Warna LED ketika tersambung WiFi adalah biru. Ketika sudah tersambung ke jaringan WiFi, LED akan menampilkan jaringan sudah tersambung.

6. Pengujian Bot Telegram

```
void loginTelegram(){
    lcd.setCursor(0,0);
    lcd.print("Menghubungkan Ke");
    lcd.setCursor(0,1);
    lcd.print("  Telegram Bot  ");
    Serial.println("login in telegram...");
    while (!myBot.testConnection())    {
        myBot.setTelegramToken(token);
        Serial.print(".");
        delay(100);}

  if(myBot.testConnection()){
        Serial.println("Telegram connection OK!");
        lcd.setCursor(0,0);
        lcd.print("  Telegram Bot  ");
        lcd.setCursor(0,1);
        lcd.print("   Terhubung    ");
  } else {
        Serial.println("Connection NOK");
        lcd.setCursor(0,0);
        lcd.print(" Telegram  Bot  ");
        lcd.setCursor(0,1);
        lcd.print("Gagal  Terhubung");
  }delay(1500);
}
```

Kode diatas sebagai penghubung ke bot telegram, dengan memasukkan token Bot Telegram dan ID Telegram, seperti yang tertera:

```
#define token     "6234639140:AAEAVHJCiSCAnyEM_COs1N-rk_1cgEMOnLA"
#define id        647924843
```

7. Pengujian Sensor Hujan (*Rain Sensor)*

```
  if(suhu<=temp){
    digitalWrite(blower,LOW);
    digitalWrite(fan,LOW);
  } else if (suhu>temp){
    digitalWrite(blower,HIGH);
    digitalWrite(fan,HIGH);
  }
```

Kode diatas berfungsi untuk membuat sebuah kondisi ketika sensor hujan mendeteksi air, maka wadah otomatis akan masuk ke dalam dan hairdryer akan hidup, dan ketika sensor hujan tidak mendeteksi adanya hujan maka wadah tetap terbuka.

8. Pengujian Relay

Pengujian relay berfungsi untuk mengontrol ON/OFF komponen output.

a. Pengujian relay pada hairdryer

```
if(suhu<=temp){
  digitalWrite(blower,LOW);
  digitalWrite(fan,LOW);
} else if (suhu>temp){
  digitalWrite(blower,HIGH);
  digitalWrite(fan,HIGH);
}
```

Pada program diatas, digunakan untuk ON/OFF hairdryer pada beberapa kondisi.

b. Pengujian relay pada motor wadah pengering dan lampu

```
switch(motor){
 case 0 : //diam
   digitalWrite(motor1a,LOW);
   digitalWrite(motor1b,LOW);
   break;
 case 1 :
   digitalWrite(motor1a,LOW);
   digitalWrite(motor1b,HIGH);
   digitalWrite(lampu,HIGH);
   break;
 case 2 :
   digitalWrite(motor1a,HIGH);
   digitalWrite(motor1b,LOW);
   digitalWrite(lampu,LOW);
   break;
 default :
   digitalWrite(motor1a,LOW);
   digitalWrite(motor1b,LOW);
   break;
}
```

Pada program diatas, digunakan untuk membuka atau menutup wadah pada beberapa kondisi, dan ON/OFF lampu pada waktu tertentu. Yaitu pada siang tapi cerah.

B. Pengujian Fungsional

Pengujian fungsional dari flowchart dan sistem dari keseluruhan sistem yang terdiri dari pengujian driver motor, sensor hujan, *Real Time Clock* (RTC), relay dan sensor suhu DHT11.

1. Pengujian Relay

   Relay berfungsi mengendalikan hidup dan mati dari motor wadah, hairdryer dan lampu. Pada saat mikrokontroler memberkan nilai HIGH (1), maka relay otomatis akan berada pada kondisi ON. Sebaliknya apabila mikrokontroler memberikan nilai LOW(0), maka kondisi relay otomatis akan berada pada posisi OFF.

No	Port Relay	Pole	
		NC	NO
1.	Motor Wadah	Tidak Terhubung	Terhubung
2.	HairDryer	Tidak Terhubung	Terhubung
3.	Lampu	Tidak Terhubung	Terhubung
4.	Tidak dipakai	Tidak Terhubung	Tidak Terhubung

Tabel 1. Pin Relay

2. Pengujian Sensor Suhu (DHT11)

   Pengujian pada sensor suhu DHT 11 bertujuan untuk mengukur nilai tegangan, suhu serta nilai kelembapan yang ada di dalam ruangan pengering kerupuk, Pengukuran tegangan sendiri bertujuan untuk mendapatkan tegangan

perubahan yang terjadi ketika suhu dari kondisi panas. Untuk suhu maksimal berada pada 60^0C

No	Titik Pengukuran	Tegangan Terukur (VOUT)	Suhu (^{0}C)
1	TP1	5 VDC	35-60^0C

Tabel 2. Pin Sensor Suhu

3. Pnegujian Rain Sensor

Pengujian pada Rain sensor ini memastikan sensitivitas pada rain sensor dan membaca adanya air yang mengenai sensor.

No	Kondisi	Kode yang dikirimkan
1.	Siang	1
2.	Malam	0

4. Pengujian Pada Telegram

Pada pengujian ini melihat apakah ESP32 mengirim pesan pada pada telegram atau tidak. Ketika kondisi berubah seperti terdeksinya perubahan cuaca, maka ESP32 akan mengirim pesan ke Telgram melalui bot.

Untuk memulai bot bisa mengetikkan perintah “/start” maka alat pengering akan diaktifkan, alat otomatis akan cek keadaan apakah siang atau malam melalui RTC, dan melihat apakah cuaca cerah atau hujan dengan menggunakan rain sensor, untuk menghidupkan pengering harus memberikan timer dahulu, ada beberapa opsi untuk memberikan timer.

Gambar 11. Percakapan dengan Bot Telegram

Bisa dilihat pada gambar diatas penggunaan Bot Telegram sebagai perintah untuk menjalankan alat, yaitu memberikan perintah untuk mengatur waktu, serta notifikasi. Untuk gambar alat ada pada Lampiran 2.

BAB V

PENUTUP

A. Kesimpulan

Setelah melakukan tahap perencanaan, pembuatan dan proses pengujian serta analisa terhadap "Perancangan Program Alat Pengering Kerupuk Ubi Mengunakan Modul ESP32 Berbasis IoT", maka dapat ditarik kesimpulan dan saran sebagai berikut:

1. Alat pengering kerupuk sudah dapat mengeringkan kerupuk dengan baik dan mengunakan sensor suhu, kemudian ditampilkan langsung pada LCD 16x2 dan dapat menyimpan data pada database.
2. Alat ini bisa dikatakan alternatif ketika pengrajin kerupuk tidak bisa mengeringkan kerupuk karena cuaca yang kurang bersahabat atau produksi yang banyak sehingga tidak ada tempat untuk menjemur kerupuk.

B. Saran

Dari hasil proyek akhir ini masih terdapat beberapa kekurangan dan dimungkinkan untuk dikembangkan lebih lanjut, maka untuk kedepannya penulis memberikan beberapa saran, yaitu:

1. Untuk kedepannya alat dapat dikembangkan lebih baik lagi dan memberikan inovasi baru seiring berkembangnya teknologi.
2. Untuk pengukuran kekeringan kerupuk, disarankan untuk memakai tingkat kelembapan dari ruangan tersebut, agar kadar air dalam udara dapat diketahui

3. Sebaiknya lebih teliti dalam pemograman sehingga dalam uji program tidak banyak terdapat kesalahan dan menjalankan program dengan output yang tepat. Jika terlalu sering melakukan kesalahan maka pada saat melalukan compile ke arduino akan sering mengalami error.
4. Untuk kedepannya program bisa menutupi kurangnya dalam mekanik alat agar bisa meningkatkan nilai keefektifan dari fungsinya.

DAFTAR PUSTAKA

Handoko, P. (2019). MODUL PRAKTIKUM DASAR-DASAR PEMROGRAMAN. Tangerang Selatan.

Ridlo, I. A. (2017). PANDUAN PEMBUATAN FLOWCHART. 7-11.

Setiawan, R. (2021, Agustus 4). Flowchart adalah: Fungsi, Jenis, Simbol, dan Contohnya. *dicoding*.

Bilah, A. (2019, Maret 4). Mengapa Harus Belajar Bahasa Pemrograman C?

Sekreningsih Nita, A. F. (2022, Februari 2). Implementasi E-Learning Berbasis Multiplatform pada Dunia Pendidikan Sebagai Solusi dari Program MBKM. *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS TAHUN 2022*, 1, 90

Ariyanti, S. (2018). SISTEM BUKA TUTUP PINTU OTOMATIS BERBASIS SUARA. ELINVO (*Electronics, Informatics, and Vocational Education*), May 2018,3(1), 83-91, DOI:10.21831/elinvo.v3i1.19076, 84-91.

Soeroso, H. (2017). Penggunaan Bot Telegram Sebagai Announcement System pada Intansi Pendidikan. *2*(1), 45-48.

Hakim, A. J. (2015). Prototype Smart Home Dengan Konsep Internet of Things (IoT) Menggunakan Arduino Berbasis Web. 5-26.

Artiyasa, M. (2020, September). APLIKASI SMART HOMENODE MCU IOT UNTUK BLYNK. *Jurnal Rekayasa Teknologi Nusa Putra*, Vol. 7, 3-7.

Djuandi, F. (2011). *PENGENALAN ARDUINO.* 2011.

Endra, R. Y. (2019, Oktober). IMPLEMENTASI SISTEM KONTROL BERBASIS WEB PADA SMART. (Prof. Mustofa Usman, Ed.) *Jurnal Sistem Informasi dan Telematika*, 98-106.

Elwin Mulyanah, C. M. (2015). Perancangan Dan Pembuatan Alat Pengering Kerupuk Otomatis Menggunakan Mikrokontroler atmega16. *ejournal.bsi*, 2.

Fadly. (2020, Agustus 09). *Kompas*. Diambil kembali dari kompas.com: https://www.kompas.com/food/read/2020/08/09/190700775/sejarah-kerupuk-di-indonesia-makanan-pokok-pada-masa-penjajahan

Gradianto, R. A. (2023, Maret 1). *Bola*. Diambil kembali dari Bola: https://www.bola.com/ragam/read/5220487/pengertian-perkembangan-teknologi-dan-berbagai-contohnya

Iqbal, M. (2022, Mei Selasa). *Mikrokrontroler ESP32*. Diambil kembali dari Telkom University: https://miqbal.staff.telkomuniversity.ac.id/mikrokontroler-esp32/

Ir. Sutrisno Koswara, M. (2013). *Pengolahan aneka kerupuk.* Bandung: tekpan unimus.

kemenperin. (2021, Maret 7). *kemenperin.go.id*. Diambil kembali dari kementrian perindustrian: https://www.kemenperin.go.id/artikel/22346/Pemerintah-Optimalkan-Peran-Industri-Pulihkan-Ekonomi-Nasional

Soeroso, H. (2017). Penggunaan Bot Telegram Sebagai Announcement System pada Intansi Pendidikan. *2*(1), 45-48.

Sujono, W. A. (2019). RANCANG BANGUN PENDETEKSI PENGAMAN PINTU . *Exact Papers in Compilation*, 307 - 314.

Lampiran 1. Listing Program Mesin Pengering Kerupuk Otomatis

```
#include <WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#include "RTClib.h"
#include "CTBot.h"

#define ssid     "OPPO A17"
#define password  ""
#define token    "6234639140:AAEAVHJCiSCAnyEM_COs1N-rk_1cgEMOnLA"
#define id       647924843

#define sensorHujan   2
#define limitSwitch1  4
#define limitSwitch2 17
#define sensorSuhu   27
#define motor1a      12
#define motor1b      14
#define blower       25
#define lampu        32
#define fan          33

CTBot myBot;
CTBotInlineKeyboard kbd, ya;
DHT dht(sensorSuhu, DHT11);
RTC_DS3231 rtc;
LiquidCrystal_I2C lcd(0x27,16,2);
byte Simbol_derajat = B11011111;
char days[7][12] = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jum'at",
"Sabtu"};
static int motor=0, statusBlow=0;
static int x=0, counter1=0,counter2=0, counter3=0;
unsigned short int pagiCerah=0,malamCerah=0,pagiHujan=0,malamHujan=0;

void connectWifi(){
  Serial.println("Connecting To Wifi");
  lcd.setCursor(0,0);
  lcd.print("   Connecting   ");
  lcd.setCursor(0,1);
```

```
  lcd.print("    To  Wifi    ");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("Wifi Connected");
  Serial.println(WiFi.SSID());
  Serial.println(WiFi.RSSI());
  Serial.println(WiFi.macAddress());
  Serial.println(WiFi.localIP());
  Serial.println(WiFi.gatewayIP());
  Serial.println(WiFi.dnsIP());
  lcd.setCursor(0,0);
  lcd.print("     WiFi       ");
  lcd.setCursor(0,1);
  lcd.print("   Connected    ");
  delay(1500);
}

void loginTelegram(){
  lcd.setCursor(0,0);
  lcd.print("Menghubungkan Ke");
  lcd.setCursor(0,1);
  lcd.print("  Telegram Bot  ");
  Serial.println("login in telegram...");
  while (!myBot.testConnection())    {
    myBot.setTelegramToken(token);
    Serial.print(".");
    delay(100);}

 if(myBot.testConnection()){
    Serial.println("Telegram connection OK!");
    lcd.setCursor(0,0);
    lcd.print("  Telegram Bot  ");
    lcd.setCursor(0,1);
    lcd.print("   Terhubung    ");
 } else {
    Serial.println("Connection NOK");
    lcd.setCursor(0,0);
    lcd.print(" Telegram  Bot  ");
```

```
    lcd.setCursor(0,1);
    lcd.print("Gagal  Terhubung");
  }delay(1500);
}

void setup(){
  pinMode(sensorHujan, INPUT);
  pinMode(limitSwitch1, INPUT_PULLUP);
  pinMode(limitSwitch2, INPUT_PULLUP);
  pinMode(sensorSuhu, INPUT);
  pinMode(motor1a, OUTPUT);
  pinMode(motor1b, OUTPUT);
  pinMode(blower, OUTPUT);
  pinMode(lampu, OUTPUT);
  pinMode(fan, OUTPUT);

  digitalWrite(blower,HIGH);
  digitalWrite(lampu,HIGH);
  digitalWrite(fan,HIGH);

  lcd.init(); lcd.backlight();
  dht.begin(); lcd.clear();
  Serial.begin(9600);
  connectWifi();
  loginTelegram();

  kbd.addButton("15 Menit", "m15", CTBotKeyboardButtonQuery);
  kbd.addButton("30 Menit", "m30", CTBotKeyboardButtonQuery);
  kbd.addButton("45 Menit", "m45", CTBotKeyboardButtonQuery);
  kbd.addRow();
  kbd.addButton("1 Jam", "j1", CTBotKeyboardButtonQuery);
  kbd.addButton("2 Jam", "j2", CTBotKeyboardButtonQuery);
  kbd.addButton("3 Jam", "j3", CTBotKeyboardButtonQuery);
  kbd.addButton("5 Jam", "j5", CTBotKeyboardButtonQuery);
  ya.addButton("YAA!!", "yes", CTBotKeyboardButtonQuery);
  ya.addButton("TIDAK", "no", CTBotKeyboardButtonQuery);

  if (! rtc.begin()) {
    Serial.println("Could not find RTC! Check circuit.");
    while (1); }
```

```
  rtc.adjust(DateTime(__DATE__, __TIME__));
}

void loop(){
  TBMessage msg;
  DateTime now = rtc.now();
  static DateTime future;
  int detik = now.second();
  int menit = now.minute();
  int jam = now.hour();
  int tanggal = now.day();
  int bulan = now.month();
  int tahun = now.year();
  String hari = days[now.dayOfTheWeek()];

  float kelembapan = dht.readHumidity();
  float suhu = dht.readTemperature();
  int hujan = digitalRead(sensorHujan);
  int LSwitch1 = digitalRead(limitSwitch1);
  int LSwitch2 = digitalRead(limitSwitch2);

  if (myBot.getNewMessage(msg)){
    if (msg.text.equalsIgnoreCase("/start")) {
      if (x==0){
        future = now;
        myBot.sendMessage(id, "Pengering kerupuk otomatis Diaktifkan!!
☺\n\nSilahkan atur pewaktu Pemanas !! 👇", kbd);
      } else if (x>=1 && ((future - now).hours()>0 || (future - now).minutes()>0 ||
(future - now).seconds()>0)){
        myBot.sendMessage(id, "Mesin Telah diaktifkan! \nTunggu
Selama\n"+String((future - now).hours())+" Jam : "+String((future -
now).minutes())+" Menit : "+String((future - now).seconds())+ " Detik\nHingga
Kerupuk Kering 😄");
      } else if (x>=1 && (future - now).hours()<=0 && (future - now).minutes()<=0
&& (future - now).seconds()<=0){
        myBot.sendMessage(id, "Silahkan atur pewaktu dibawah Terlebih dahulu
👇" ,kbd);
      } x++;
    } else if(msg.text.equalsIgnoreCase("/status")) {
      if (statusBlow==1){
```

```
      myBot.sendMessage(id, "Pemanas Diaktifkan!\n\nWaktu
Tersisa  \n"+String((future - now).hours())+" Jam : "+String((future -
now).minutes())+" Menit : "+String((future - now).seconds())+" Detik\nSuhu :
"+String(suhu)+"°C\nKelembapan : "+String(kelembapan)+" %");
    } else if (statusBlow==0){
      myBot.sendMessage(id, "Pemanas Tidak Aktif! \n\nSuhu :
"+String(suhu)+" °C\nKelembapan : "+String(kelembapan)+" %");
    }
  } else if(msg.text.equalsIgnoreCase("/time")) {
    myBot.sendMessage(id, "Silahkan Pilih Tambahan Waktu!",kbd);
  } else if(msg.text.equalsIgnoreCase("/off")) {
    digitalWrite(blower,HIGH);
    digitalWrite(fan,HIGH);
    statusBlow = 0; x=0;
    future = now + TimeSpan(0, 0, 0, 0);
    myBot.sendMessage(id, "Pemanas Telah dimatikan 🙃");
  } else if (msg.messageType == CTBotMessageQuery ) {
     if (msg.callbackQueryData.equals("m15") && x>0) {
     myBot.endQuery(msg.callbackQueryID, "Waktu ditambahkan 15 Menit
🕒");
     future = future + TimeSpan(0, 0, 1, 0);
    } else if (msg.callbackQueryData.equals("m30") && x>0) {
     myBot.endQuery(msg.callbackQueryID, "Waktu ditambahkan 30 Menit
🕒");
     future = future + TimeSpan(0, 0, 30, 0);
    } else if (msg.callbackQueryData.equals("m45") && x>0) {
     myBot.endQuery(msg.callbackQueryID, "Waktu ditambahkan 45 Menit
🕒");
     future = future + TimeSpan(0, 0, 45, 0);
    } else if (msg.callbackQueryData.equals("j1") && x>0) {
     myBot.endQuery(msg.callbackQueryID, "Waktu ditambahkan 1 Jam 🕒");
     future = future + TimeSpan(0, 1, 0, 0);
    } else if (msg.callbackQueryData.equals("j2") && x>0) {
     myBot.endQuery(msg.callbackQueryID, "Waktu ditambahkan 2 Jam 🕒");
     future = future + TimeSpan(0, 2, 0, 0);
    } else if (msg.callbackQueryData.equals("j3")&& x>0) {
     myBot.endQuery(msg.callbackQueryID, "Waktu ditambahkan 3 Jam 🕒");
     future = future + TimeSpan(0, 3, 0, 0);
    } else if (msg.callbackQueryData.equals("j5")&& x>0) {
     myBot.endQuery(msg.callbackQueryID, "Waktu ditambahkan 5 Jam 🕒");
```

```
      future = future + TimeSpan(0, 5, 0, 0);
    } else if (msg.callbackQueryData.equals("yes")) { x=1;
      myBot.sendMessage(id, "Silahkan Pilih Tambahan Waktu!",kbd);
    } else if (msg.callbackQueryData.equals("no")) {
      myBot.endQuery(msg.callbackQueryID, "OKE BOSQUEE! 😉😉");
    }
  } else if (msg.text.equalsIgnoreCase("/menu")){
    myBot.sendMessage(id, "Silahkan pilih menu dibawah 👇\n/start --> 
Menyalakan Pemanas \n/status --> Melihat Status pemanas\n/time --> 
Menambah timer Pemanas\n/off --> Mematikan Pemanas\n/menu --> Pilihan 
Menu");
  } else if(msg.text.equalsIgnoreCase("/jam")) {
    myBot.sendMessage(id, "Jam Sekarang\n"+String(now.hour())+" : 
"+String(now.minute())+" : "+String(now.second())+" WIB\n"+String(hari)+", 
"+String(tanggal)+"-"+String(bulan)+"-"+String(tahun));
  } else {
    myBot.sendMessage(id, "Perintah tidak dikenali! ☠️☠️\nSilahkan pilih 
menu dibawah 👇\n/start --> Menyalakan Pemanas \n/status --> Melihat 
Status pemanas\n/time --> Menambah timer Pemanas\n/off --> Mematikan 
Pemanas\n/menu --> Pilihan Menu");
  }
 }

 if(suhu<=55 && counter1==0 && ((future - now).hours()>0 || (future - 
now).minutes()>0 || (future - now).seconds()>0 )){
  digitalWrite(blower,LOW);
  digitalWrite(fan,LOW);
  lcd.setCursor(0,0); lcd.print("  Suhu  Rendah  ");
  lcd.setCursor(0,1); lcd.print("   Pemanas ON   ");
  myBot.sendMessage(id, "Suhu Rendah! ❄️\nMenyalakan Pemanas!");
  statusBlow = 1; counter1=1; counter2=0;
 } else if (suhu>55 && counter2==0){
  digitalWrite(blower,HIGH);
  digitalWrite(fan,HIGH);
  lcd.setCursor(0,0); lcd.print("  Suhu  Tinggi  ");
  lcd.setCursor(0,1); lcd.print("  Pemanas OFF   ");
  myBot.sendMessage(id, "Suhu Terlalu Tinggi! 🔥 \nMematikan Pemanas! ");
  statusBlow = 0; counter1=0; counter2=1;
 }
```

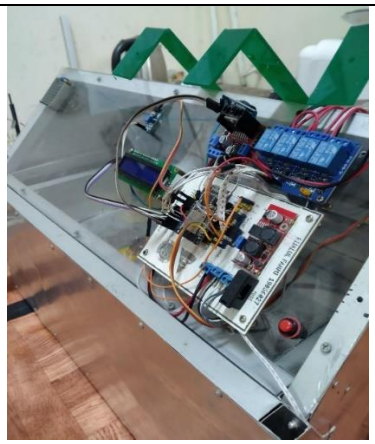
```
if (jam>=7 && jam<18 && hujan==1){
  motor=1; pagiCerah++;      //membuka
} else if ((jam<7 || jam>=18) && hujan==1){
  motor=2; malamCerah++;     //menutup
} else if (jam>=7 && jam<18 && hujan==0){
  motor=2; pagiHujan++;      //menutup
} else if ((jam<7 || jam>=18) && hujan==0){
  motor=2; malamHujan++;     //menutup
} else {motor=0; }

if (LSwitch1==0 || LSwitch2==0){
  motor=0;
}

 switch(motor){
  case 0 : //diam
    digitalWrite(motor1a,LOW);
    digitalWrite(motor1b,LOW);
    break;
  case 1 :
    digitalWrite(motor1a,LOW);
    digitalWrite(motor1b,HIGH);
    digitalWrite(lampu,HIGH);
    break;
  case 2 :
    digitalWrite(motor1a,HIGH);
    digitalWrite(motor1b,LOW);
    digitalWrite(lampu,LOW);
    break;
  default :
    digitalWrite(motor1a,LOW);
    digitalWrite(motor1b,LOW);
    break;
 }

 if (pagiCerah==1){
     lcd.clear();
     lcd.setCursor(0,0); lcd.print("Cuaca Cerah     ");
     lcd.setCursor(0,1); lcd.print("Pukul");
     lcd.setCursor(6,1); lcd.print(jam);
     lcd.setCursor(8,1); lcd.print(":");
```

```
      lcd.setCursor(9,1); lcd.print(menit);
      lcd.setCursor(11,1); lcd.print(":");
      lcd.setCursor(12,1); lcd.print(detik);
      myBot.sendMessage(id, "Cuaca Cerah ⛅ \n\n"+String(hari)+", Pukul
"+String(jam)+" : "+String(menit)+" : "+String(detik)+" WIB.\nAtap Telah
dibuka!! 😇");
      malamCerah=0;pagiHujan=0; malamHujan=0;
  } else if (malamCerah==1){
      lcd.clear();
      lcd.setCursor(0,0); lcd.print("Cuaca  Cerah    ");
      lcd.setCursor(0,1); lcd.print("Pukul ");
      lcd.setCursor(6,1); lcd.print(jam);
      lcd.setCursor(8,1); lcd.print(":");
      lcd.setCursor(9,1); lcd.print(menit);
      lcd.setCursor(11,1); lcd.print(":");
      lcd.setCursor(12,1); lcd.print(detik);
      myBot.sendMessage(id, "Cuaca Cerah ⛅ \nNamun Malam Telah tiba! 🎶
\n\n"+String(hari)+", Pukul "+String(jam)+" : "+String(menit)+" :
"+String(detik)+" WIB.\nAtap Telah Ditutup!! 😇");
      pagiCerah=0; pagiHujan=0; malamHujan=0;
  } else if (pagiHujan==1){
      lcd.clear();
      lcd.setCursor(0,0); lcd.print("Cuaca  Hujan    ");
      lcd.setCursor(0,1); lcd.print("Pukul");
      lcd.setCursor(6,1); lcd.print(jam);
      lcd.setCursor(8,1); lcd.print(":");
      lcd.setCursor(9,1); lcd.print(menit);
      lcd.setCursor(11,1); lcd.print(":");
      lcd.setCursor(12,1); lcd.print(detik);
      myBot.sendMessage(id, "Terjadi Hujan!!! 🌧\n\n"+String(hari)+", Pukul
"+String(jam)+" : "+String(menit)+" : "+String(detik)+" WIB.\nAtap Telah
Ditutup!! 😇");
      pagiCerah=0; malamCerah=0; malamHujan=0;
  } else if (malamHujan==1){
      lcd.clear();
      lcd.setCursor(0,0); lcd.print("Cuaca  Hujan    ");
      lcd.setCursor(0,1); lcd.print("Pukul");
      lcd.setCursor(6,1); lcd.print(jam);
      lcd.setCursor(8,1); lcd.print(":");
      lcd.setCursor(9,1); lcd.print(menit);
```

```
      lcd.setCursor(11,1); lcd.print(":");
      lcd.setCursor(12,1); lcd.print(detik);
      myBot.sendMessage(id, "Terjadi Hujan pada malam hari
🌧\n\n"+String(hari)+", Pukul "+String(jam)+" : "+String(menit)+" :
"+String(detik)+" WIB.\nAtap Telah ditutup!! 😇");
      pagiCerah=0; malamCerah=0; pagiHujan=0;
  } else {
   if ((future - now).hours()<=0 && (future - now).minutes()<=0 && (future -
now).seconds()<=0 && counter3==0 && statusBlow==1){
    lcd.clear();
    lcd.setCursor(0,0); lcd.print(" Waktu Pemanas  ");
    lcd.setCursor(0,1); lcd.print("  Telah Habis   ");
    digitalWrite(blower,HIGH);
    digitalWrite(fan,HIGH);
    myBot.sendMessage(id, "Waktu Pemanasan Telah Habis! \nApakah Anda
ingin Menambah Waktu?",ya);
    counter3=1; statusBlow=0; x=0;
   } else if ((future - now).hours()>0 || (future - now).minutes()>0 || (future -
now).seconds()>0 ){
    counter3=0; statusBlow=1; lcd.clear();
      lcd.setCursor(0,0); lcd.print("Suhu : ");
      lcd.setCursor(8,0); lcd.print(suhu);
      lcd.setCursor(14,0); lcd.write(Simbol_derajat);
      lcd.setCursor(15,0); lcd.print("C");
      lcd.setCursor(0,1); lcd.print("Waktu: ");
      lcd.setCursor(7,1); lcd.print((future - now).hours());
      lcd.setCursor(9,1); lcd.print(":");
      lcd.setCursor(10,1); lcd.print((future - now).minutes());
      lcd.setCursor(12,1); lcd.print(":");
      lcd.setCursor(13,1); lcd.print((future - now).seconds());
      delay(800);
      if(suhu<=55){
        digitalWrite(blower,LOW);
        digitalWrite(fan,LOW);
      } else if (suhu>55){
        digitalWrite(blower,HIGH);
        digitalWrite(fan,HIGH);
      }
   } else{
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("Suhu : ");
```

```
    lcd.setCursor(8,0); lcd.print(String(suhu));
    lcd.setCursor(14,0); lcd.write(Simbol_derajat);
    lcd.setCursor(15,0); lcd.print("C");
    lcd.setCursor(0,1); lcd.print("Pemanas OFF     ");
    if(x>=1){future = now;}
   }
  }
}
```

Lampiran 2. Gambar Mesin Pengering Kerupuk Ubi Menggunakan ESP32 berbasis IoT

	Tampak dari Depan
	**Tampak dari atas**
	Tampak dari belakang