

RANCANG BANGUN APLIKASI PANTAU DAN KENDALI *MOBILE* MESIN PENGERING BIJI KOPI BERBASIS *INTERNET OF THINGS*



NYSSA ROMANA

H071181313



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN
ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

**RANCANG BANGUN APLIKASI PANTAU DAN KENDALI *MOBILE*
MESIN PENGERING BIJI KOPI BERBASIS *INTERNET OF THINGS***

**NYSSA ROMANA
H071181313**



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

**RANCANG BANGUN APLIKASI PANTAU DAN KENDALI *MOBILE*
MESIN PENGERING BIJI KOPI BERBASIS *INTERNET OF THINGS***

NYSSA ROMANA
H071181313

Skripsi

Sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Sistem Informasi

pada

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

SKRIPSI**RANCANG BANGUN APLIKASI PANTAU DAN KENDALI *MOBILE*
MESIN PENGERING BIJI KOPI BERBASIS *INTERNET OF THINGS*****NYSSA ROMANA****H071181313**

Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana Sistem Informasi pada 26
Agustus 2024 dan dinyatakan telah memenuhi syarat kelulusan pada



Mengesahkan:
Pembimbing tugas akhir,

Pembimbing Pendamping

Dr. Eng. Armin Lawi, S.Si., M.Eng
NIP 197204231995121001

Dr. Muhammad Hasbi, M.Sc
NIP 196307201989031003

Mengetahui:
Ketua Program Studi,

Prof. Drs. Jeffry Kusuma, Ph.D.
NIP 196411121987031002

PERNYATAAN KEASLIAN SKRIPSI DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, skripsi berjudul “Rancang Bangun Aplikasi Pantau dan Kendali Mobile Mesin Pengering Biji Kopi Berbasis Internet Of Things” adalah benar karya saya dengan arahan dari pembimbing Dr.Eng. Armin Lawi, S.Si., M.Eng. sebagai Pembimbing Utama dan Dr. Muhammad Hasbi, M.Sc sebagai Pembimbing Pendamping. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 26 Agustus 2024

Nyssa Romana
H071181313

Ucapan Terima Kasih

Puji syukur Penulis panjatkan kehadiran Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya Penulis dapat menyelesaikan skripsi ini yang berjudul “Rancang Bangun Aplikasi Pantau Dan Kendali *Mobile* Mesin Pengering Biji Kopi Berbasis *Internet Of Things*”. Saya menyadari bahwa dalam penyusunan Skripsi ini tidak luput dari bantuan dan bimbingan berbagai pihak. Oleh karena itu, pada kesempatan ini dengan segala kerendahan hati Penulis menyampaikan terimakasih yang sebesar-besarnya kepada :

1. Rektor Universitas Hasanuddin Makassar **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, dan seluruh Wakil Rektor dalam Lingkungan Universitas Hasanuddin.
2. Bapak Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam **Dr. Eng Amiruddin** dan para Wakil Dekan serta seluruh staf yang telah memberikan bantuan selama penulis mengikuti pendidikan di FMIPA Universitas Hasanuddin.
3. Bapak **Dr. Firman, S.Si., M.Si.** selaku Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. Bapak **Prof. Drs. Jeffry Kusuma, Ph.D.** sebagai Ketua Program Studi Sistem Informasi Universitas Hasanuddin.
5. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng.** selaku dosen pembimbing utama atas segala ilmu, bantuan, saran, nasehat, dan motivasi yang telah diberikan selama proses menjalani pendidikan serta kesabaran dalam membimbing penulis dalam proses penyusunan skripsi ini.
6. Bapak **Dr. Muhammad Hasbi, M.Sc** selaku Dosen Pembimbing pendamping yang senantiasa menyediakan waktu, pikiran dan saran-sarannya yang membangun dalam penulisan Skripsi ini.
7. Bapak **Dr. Agustinus Ribal, S.Si., M.Sc.** dan **Muhammad Sadno, S.Si., M.Si.** selaku penguji yang telah memberikan kritik dan saran yang sangat membangun demi penyempurnaan Skripsi ini.
8. Bapak Anthonius Gala dan Ibunda Ery S. Pagallungan selaku orang tua penulis yang senantiasa memberikan dukungan dan bantuan material serta kasih sayang dan selalu mendoakan sehingga Penulis mampu menyelesaikan skripsi ini.
9. Kakak tercinta dr.Lhorensia dan adik-adik tercinta Satrya Imanuel dan Jhosua Gala yang telah banyak memberi dukungan semangat dan doa sehingga penulis mampu menyelesaikan skripsi ini.
10. Psikiater dr. Mayamariska Sanusi, Sp.KJ yang telah banyak memberi bantuan konseling sehingga penulis mampu menyelesaikan skripsi ini.
11. Keluarga besar Sistem Informasi 2018 yang telah memberikan dukungan dan menjadi teman seperjuangan yang baik.
12. Semua Pihak yang tidak dapat saya sebutkan satu persatu.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua belia pihak yang telah membantu Penulis. Semoga Skripsi ini membawa manfaat bagi pengembangan ilmu.

Makassar , 26 Agustus 2024

Nyssa Romana

ABSTRAK

NYSSA ROMANA. **Rancang Bangun Aplikasi Pantau Dan Kendali *Mobile* Mesin Pengering Biji Kopi Berbasis *Internet Of Things*** (dibimbing oleh Dr. Eng. Armin Lawi, S.Si., M.Eng dan Dr. Muhammad Hasbi, M.Sc).

Latar belakang. Indonesia, sebagai salah satu produsen kopi terbesar di dunia, menghadapi tantangan dalam meningkatkan produktivitas kopi dibandingkan dengan negara penghasil utama lainnya. Proses pengeringan biji kopi yang sering dilakukan secara tradisional dan bergantung pada cuaca menyebabkan kualitas yang tidak konsisten. **Tujuan.** Penelitian ini bertujuan untuk merancang dan membangun aplikasi pantau dan kendali mobile berbasis Internet of Things (IoT) untuk mesin pengering biji kopi berbentuk tabung berputar (rotary machine). Mesin ini mengintegrasikan sensor DHT22 untuk pemantauan suhu dan kelembaban, dan dikendalikan oleh mikrokontroler NodeMCU ESP8266 dengan aktuator seperti motor listrik, pemanas, dan kipas. Aplikasi mobile yang dikembangkan memungkinkan pemantauan dan pengendalian proses pengeringan secara real-time melalui perangkat Android, menawarkan solusi yang efisien untuk mengatasi tantangan cuaca. **Metode.** Metode yang diunakan dalam penelitian ini adalah metode waterfall. **Hasil.** Pengujian menunjukkan bahwa sistem ini mampu menjaga suhu dalam rentang optimal dan mengurangi kadar air biji kopi secara signifikan, dengan penurunan kadar air dari 22% menjadi 12,4% dalam waktu 2 jam. **Kesimpulan.** Dengan sistem ini, proses pengeringan tidak hanya menjadi lebih konsisten tetapi juga lebih terkontrol, sehingga memberikan manfaat yang berpotensi besar untuk industri kopi.

Kata Kunci: Produktivitas Kopi, Mesin Pengering, IoT, Sensor DHT22, Aplikasi Mobile

ABSTRACT

NYSSA ROMANA. **Design and Development of a Mobile Monitoring and Control Application for a Coffee Bean Dryer Based on the Internet of Things** (Supervised by Dr. Eng. Armin Lawi, S.Si., M.Eng and Dr. Muhammad Hasbi, M.Sc).

Background. Indonesia, as one of the world's largest coffee producers, faces challenges in improving coffee productivity compared to other leading coffee-producing countries. The traditional coffee bean drying process, which often relies on weather conditions, results in inconsistent quality. **Aim.** This research aims to design and develop a mobile monitoring and control application based on the Internet of Things (IoT) for a rotary coffee bean drying machine. The machine integrates a DHT22 sensor for temperature and humidity monitoring, and is controlled by a NodeMCU ESP8266 microcontroller with actuators such as an electric motor, heater, and fan. The developed mobile application enables real-time monitoring and control of the drying process via Android devices, providing an efficient solution to weather-related challenges. **Method.** The research employed the waterfall method. **Results.** Testing showed that the system could maintain temperature within an optimal range and significantly reduce the moisture content of coffee beans, decreasing it from 22% to 12.4% within 2 hours. **Conclusion.** This system not only makes the drying process more consistent but also more controlled, offering significant potential benefits for the coffee industry.

Keywords: Coffee Productivity, Coffee Dryer, IoT, DHT22 Sensor, Mobile Application

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN PENGAJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
UCAPAN TERIMA KASIH.....	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Tinjauan Pustaka	5
1.6.1 Internet of Things	5
1.6.2 Aktuator	10
1.6.3 Perangkat Lunak.....	12
1.6.4 Integrasi IoT dalam Pertanian	14
1.6.5 Penelitian Terdahulu	18
1.6.6 Kerangka Konseptual.....	19
BAB II METODE PENELITIAN	21
2.1 Waktu dan Tempat Penelitian	21
2.2 Tahapan Penelitian	21
2.2.1 Analisis Kebutuhan	22
2.2.2 Perancangan Sistem.....	22

2.2.3 Implementasi Sistem	24
2.2.4 Pengujian	24
2.2.5 Hasil	25
2.3 Arsitektur Sistem	25
2.4 Sumber Data	27
2.5 Instrumen Penelitian.....	27
BAB III HASIL DAN PEMBAHASAN.....	28
3.1 Hasil Rancangan Prototipe.....	28
3.1.1 Rancangan Prototipe Mesin Pengering	28
3.1.2 Rancangan Prototipe Mekanik	31
3.1.3 Rancangan Prototipe Perangkat Lunak	32
3.2 Hasil Implementasi Perangkat Keras Mesin Pengering Biji Kopi Berbasis <i>Internet of Things</i>	32
3.2.1 Rangkaian NodeMCU dan DHT22.....	32
3.2.2 Rangkaian NodeMCU dan relay	33
3.2.3 Rangkaian Keseluruhan Perangkat Keras Sistem.....	34
3.3 Hasil Implementasi Perangkat Lunak Pantau dan Kendali <i>Mobile</i> Mesin Pengering Biji Kopi Berbasis <i>Internet of Things</i>	38
3.3.1 Pembangunan Perangkat Lunak Menggunakan <i>Arduino IDE</i>	38
3.3.2 Implementasi Perangkat Lunak Sistem	47
3.4 Hasil Kinerja Aplikasi Pantau dan Kendali <i>Mobile</i> Mesin Pengering Biji Kopi Berbasis <i>Internet of Things</i>	48
3.4.1 Hasil Pengujian Aplikasi Pantau dan Kendali <i>Mobile</i> Mesin Pengering Biji Kopi Berbasis <i>Internet of Things</i>	49
3.4.2 Evaluasi Aplikasi Pantau dan Kendali <i>Mobile</i> Mesin Pengering Biji Kopi Berbasis <i>Internet of Things</i>	54
3.5 Pembahasan Kinerja Aplikasi Pantau dan Kendali <i>Mobile</i> Mesin Pengering Biji Kopi Berbasis <i>Internet of Things</i>	57
BAB IV KESIMPULAN DAN SARAN	59
4.1 Kesimpulan.....	59
4.2 Saran	60
DAFTAR PUSTAKA.....	xv
LAMPIRAN	xvii

DAFTAR TABEL

Tabel 1. Spesifikasi NodeMCU ESP8266	7
Tabel 2. Spesifikasi sensor DHT22.....	8
Tabel 3. Standar Mutu Umum Biji Kopi	16
Tabel 4. Standar Mutu Khusus Kopi Robusta Pengolahan Kering	16
Tabel 5. Daftar Alat dan harga Perangkat Keras.....	35
Tabel 6. Penelitian pertama menggunakan waktu 1 jam.....	50
Tabel 7. Penelitian kedua menggunakan waktu 1 jam 30 menit.....	51
Tabel 8. Penelitian ketiga menggunakan waktu 2 jam	53

DAFTAR GAMBAR

Gambar 1. Grafik peringkat negara penghasil utama kopi berdasarkan Luas Lahan	1
Gambar 2. Grafik Konsumsi kopi Indonesia 2015-2020	2
Gambar 3. <i>Internet Of Things</i>	5
Gambar 4. Mikrokontroler NodeMCU ESP8266	7
Gambar 5. Sensor DHT22	8
Gambar 6. Relay	9
Gambar 7. LCD 16x2	9
Gambar 8. Modul I2C	10
Gambar 9. Motor listrik	10
Gambar 10. Elemen Pemanas	11
Gambar 11. Kipas	11
Gambar 12. Arduino IDE	12
Gambar 13. Firebase	13
Gambar 14. MIT App Inventor	13
Gambar 15. Proses Biji Kopi	15
Gambar 16. Lapisan Biji Kopi	15
Gambar 17. Pengukur kadar air	17
Gambar 18. Timbangan digital	17
Gambar 19. Flowchart Alur Penelitian	21
Gambar 20. Blok Diagram	22
Gambar 21. <i>Use case diagram</i>	23
Gambar 22. Deployment Diagram	24
Gambar 23. Flowchart Sistem Pantau	25
Gambar 24. Flowchart Sistem Kendali	26
Gambar 25. Rancangan prototipe mesin pengering	28
Gambar 26. Rancangan letak komponen pada mesin pengering	29
Gambar 27. Rancangan prototipe mesin pengering tampak samping dan penutup atas terbuka	30
Gambar 28. Rancangan silinder wadah kopi dalam ruang pengering	30
Gambar 29. Rancangan prototipe mekanik	31
Gambar 30. Rancangan prototipe perangkat lunak	32
Gambar 31. Rangkaian NodeMCU dan DHT22	32
Gambar 32. Rangkaian NodeMCU dan relay	33
Gambar 33. Rangkaian Keseluruhan Perangkat Keras Sistem	34
Gambar 34. Hasil rancangan Elektronik	35
Gambar 35. Ilustrasi Letak Alat Pada Mesin Pengering	36
Gambar 36. Implementasi Perangkat Keras Mesin pengering	37
Gambar 37. Tampilan Arduino IDE	38
Gambar 38. Program Pendefinisian Library yang Digunakan	38
Gambar 39. Program Inisialisasi wifi dan firebase	39
Gambar 40. Program deklarasi firebase	39

Gambar 41. Program Inisialisasi pin DHT22 dan LCD	39
Gambar 42. Program Inisialisasi pin relay	40
Gambar 43. Program Inisialisasi dan deklarasi variabel.....	40
Gambar 44. Program Inisialisasi Koneksi WiFi dan Perangkat I/O	41
Gambar 45. Program Konfigurasi Koneksi Firebase dan Sinkronisasi Waktu	41
Gambar 46. Program Fungsi Pengambilan dan Penampilan Data dari Firebase ...	42
Gambar 47. Program Fungsi update data firebase	42
Gambar 48. Program pembacaan dan menampilkan suhu dan kelembapan pada serial monitor dan LCD	43
Gambar 49. Program pengiriman data suhu dan kelembapan ke firebase	43
Gambar 50. Program Penyimpanan Data Historis ke Firebase.....	45
Gambar 51. Program fungsi update relay.....	45
Gambar 52. Program fungsi manualcontrol.....	45
Gambar 53. Program fungsi autocontrolMode.....	46
Gambar 54. Program fungsi loop.....	46
Gambar 55. Tampilan Firebase	47
Gambar 56. Tampilan Aplikasi Mobile	48
Gambar 57. Tampilan Firebase dan Aplikasi mobile penelitian pertama	49
Gambar 58. Tampilan Firebase dan Aplikasi mobile penelitian kedua	50
Gambar 59. Tampilan Firebase dan Aplikasi mobile penelitian ketiga	52
Gambar 60. Grafik suhu pada percobaan pertama	54
Gambar 61. Pembacaan massa dan kadar air pertama.....	54
Gambar 62. Grafik suhu pada percobaan kedua.....	55
Gambar 63. Pembacaan massa dan kadar air kedua	55
Gambar 64. Grafik suhu pada percobaan ketiga	56
Gambar 65. Pembacaan massa dan kadar air ketiga	56

DAFTAR LAMPIRAN

Lampiran 1 Source Code Program	xviii
Lampiran 2 Gambar Mesin Pengering Berbasis IoT	xxiv
Lampiran 3 Gambar Aplikasi Pantau dan Kendali Mobile	xxv
Lampiran 4 Blok Aplikasi Mobile	xxvi

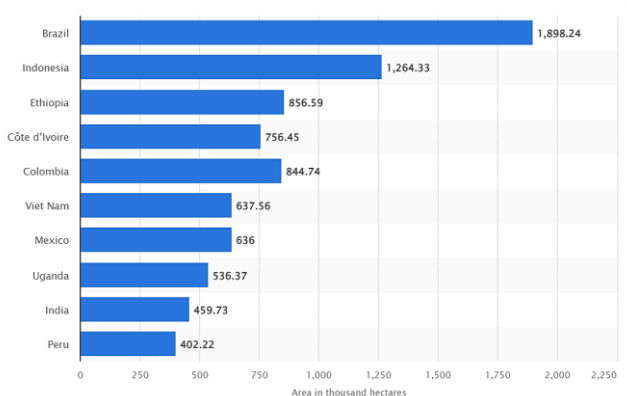
BAB I PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara agraris yang artinya sebagian besar penduduknya bekerja dalam sektor pertanian. Hal ini karena Indonesia didukung oleh kekayaan alam yang melimpah dan juga posisi Indonesia yang dinilai sangat strategis. Berdasarkan letak geografis, Indonesia terletak pada daerah tropis yang memiliki curah hujan yang tinggi. Kondisi ini membuat Indonesia memiliki lahan yang subur dan banyak jenis tumbuhan yang dapat tumbuh dengan cepat.

Ditinjau dari kontribusinya, menurut data Badan Pusat Statistik (BPS) sektor pertanian Indonesia menyumbang 13,28% dari PDB pada tahun 2021, dimana sektor pertanian menjadi penyumbang terbesar kedua setelah sektor industri pengolahan. Sekitar 3,52% kontribusi sektor pertanian tersebut berasal dari subsektor perkebunan yang salah satunya adalah kopi (Kusnandar, 2022).

Dalam perdagangan skala internasional, jika dibandingkan dengan negara pengekspor utama kopi, Indonesia memiliki keunggulan pada faktor terkait sumber daya alam. Adapun luas lahan perkebunan kopi negara-negara penghasil utama kopi di dunia tahun 2020 adalah sebagai berikut:



Gambar 1. Grafik peringkat negara penghasil utama kopi berdasarkan Luas Lahan

Sumber: (FAO) *Food and Agriculture Organization*

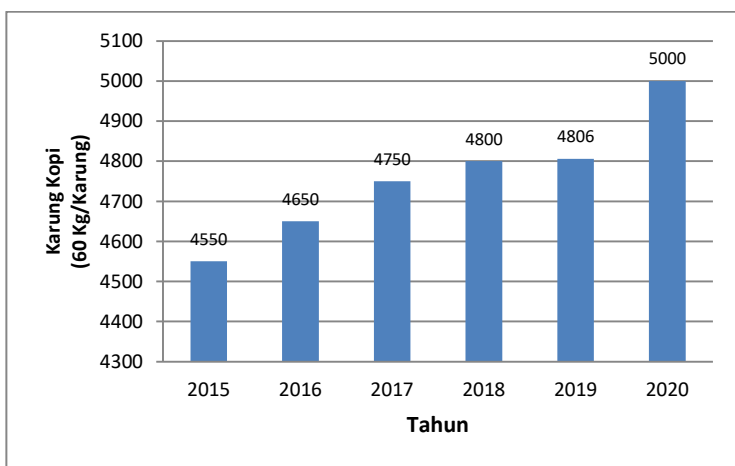
Data di atas menunjukkan bahwa Indonesia menduduki peringkat kedua setelah Brazil yang memiliki luas lahan 1.898.240 Ha. Adapun luas lahan perkebunan kopi di Indonesia pada tahun 2020 adalah 1.264.330 Ha. Selanjutnya disusul oleh Ethiopia dengan luas lahan 856.590 Ha. Berdasarkan data tersebut dapat disimpulkan bahwa Indonesia memiliki potensi dan sumber daya alam yang cukup memadai dalam perkembangan industri kopi.

Berdasarkan status tanaman, lahan perkebunan kopi di Indonesia terbagi atas tiga macam, yaitu sebagai berikut:

1. Tanaman Belum Menghasilkan (TBM) adalah tanaman yang sampai pada saat pengamatan belum pernah memberikan hasil, karena masih muda atau tanaman sudah cukup umur tetapi belum dapat menghasilkan karena tidak cocok dengan iklim, ketinggian tempat, kondisi tanah dan sebagainya.
2. Tanaman Menghasilkan (TM) adalah tanaman yang sebelum saat pengamatan pernah memberikan hasil dan masih akan memberikan hasil, meskipun pada saat pengamatan sedang tidak menghasilkan.
3. Tanaman Tidak Menghasilkan/Tua/Rusak (TTM) adalah tanaman yang sampai dengan saat pengamatan tidak pernah memberikan hasil atau tidak akan memberikan hasil lagi disebabkan tua, rusak atau mandul.

Menurut data BPS, luas lahan Tanaman Menghasilkan (TM) pada tahun 2020 adalah 920.492 Ha dengan total jumlah produksi 757.290 ton. Sehingga produktifitas kopi Indonesia pada tahun tersebut adalah 823 Kg/Ha (Badan Pusat Statistik, 2021). Produktivitas tersebut masih tergolong rendah dibandingkan negara-negara penghasil utama kopi. Sebagai contohnya, Vietnam dengan luas lahan 637.560 Ha mampu memproduksi kopi sebesar 1.740.000 ton. Kemampuan produksi tersebut membuat Vietnam memiliki produktivitas sekitar 2.729 Kg/Ha (ICO, 2021). Faktor seperti produktivitas ini tentunya mempengaruhi jumlah kopi yang dapat diekspor oleh suatu negara.

Sementara itu berdasarkan data ICO tahun 2015-2020 konsumsi kopi di dalam negeri juga terus meningkat setiap tahunnya. Hal ini berkaitan dengan mulai menjamurnya kedai-kedai kopi di berbagai tempat termasuk kota-kota besar dan konsumsi kopi yang mulai menjadi budaya baru dalam masyarakat dari berbagai kalangan.



Gambar 2. Grafik Konsumsi kopi Indonesia 2015-2020

Sumber: (International Coffee Organization, 2021)

Berdasarkan data tersebut, konsumsi kopi Indonesia pada periode 2015-2020 mencapai 5000 karung kopi (dengan kapasitas 60 Kg/karung). Konsumsi tersebut terus naik sejak tahun 2015 yakni 9% persen sampai pada tahun 2020. Hal ini menunjukkan bahwa Indonesia juga memiliki peluang industri kopi dalam negeri.

Kualitas dan nilai ekspor kopi yang baik ditentukan oleh beberapa hal, salah satunya adalah kadar air. Kopi dengan kandungan air tinggi akan menghasilkan kopi dengan kualitas buruk seperti menjadi rusak, busuk, berjamur dan berubah warna. Kadar air yang dianjurkan berdasarkan mutu ekspor adalah maksimal 12,5% namun pada tingkat petani dan pedagang pengumpul, kadar air umumnya di atas 16%. Bahkan di beberapa lokasi, kadar air di atas 20% (Karolina Tarigan, 2020). Hal ini tentunya juga dipengaruhi oleh cuaca pada tiap-tiap daerah penghasil kopi. Faktor cuaca sangat berpengaruh pada lamanya waktu pengeringan yang dibutuhkan oleh para petani.

Pengeringan biji kopi merupakan salah satu tahap penting dalam proses pascapanen kopi yang mempengaruhi kualitas akhir produk. Pada umumnya, metode pengeringan tradisional yang digunakan adalah pengeringan secara alami di bawah sinar matahari. Menurut (Kusmiyati et al., 2023) pada proses alami memiliki kekurangan yaitu bergantung pada musim/cuaca, jika memasuki musim hujan maka para petani membutuhkan waktu sekitar 1 minggu untuk memastikan bahwa biji kopi telah benar-benar kering. Sedangkan proses pengeringan menggunakan oven memiliki kekurangan yaitu daya listrik yang dibutuhkan sangat besar.

Dalam beberapa tahun terakhir, teknologi Internet of Things (IoT) telah banyak diimplementasikan dalam sektor pertanian untuk meningkatkan efisiensi dan kualitas produksi. IoT memungkinkan pengumpulan data secara real-time, analisis data, dan otomatisasi proses yang dapat membantu dalam pengambilan keputusan yang lebih baik. Dalam konteks pengeringan biji kopi, IoT dapat digunakan untuk memantau dan mengontrol kondisi lingkungan seperti suhu dan kelembaban secara otomatis, sehingga proses pengeringan dapat berjalan lebih efisien dan konsisten. Mesin pengering ini didesain dapat terkoneksi dengan internet. Dengan inovasi ini diharapkan mesin pengering kopi dapat menjadi alternatif kepada masyarakat khususnya petani kopi untuk melakukan proses pengeringan biji kopi tanpa adanya kendala cuaca maupun daya listrik yang tinggi. Metode ini memiliki beberapa kelemahan, antara lain ketergantungan pada kondisi cuaca, waktu pengeringan yang lama, dan risiko kontaminasi oleh debu, kotoran, atau mikroorganisme. Hal ini dapat menyebabkan kualitas biji kopi menurun, baik dari segi cita rasa maupun nilai ekonomis (Kusmiyati et al., 2023).

Berdasarkan latar belakang ini, penelitian ini bertujuan untuk **merancang dan membangun aplikasi pantau dan kendali mobile untuk mesin pengering biji kopi berbasis IoT**. Aplikasi ini akan memungkinkan monitoring dan pengendalian kondisi pengeringan secara real-time melalui perangkat mobile, sehingga diharapkan dapat meningkatkan efisiensi dan kualitas pengeringan biji kopi.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah di atas, dapat dikemukakan pertanyaan penelitian sebagai berikut:

1. Bagaimana merancang dan membangun mesin pengering biji kopi yang dapat dipantau dan dikendalikan menggunakan perangkat IoT?
2. Bagaimana merancang dan membangun aplikasi pantau dan kendali *mobile* yang datanya diperoleh dari mesin pengering biji kopi berbasis IoT?
3. Bagaimana menguji kinerja aplikasi pantau dan kendali *mobile* untuk mesin pengering kopi berbasis IoT?

1.3 Batasan Masalah

Berdasarkan uraian di atas maka batasan masalah yang akan dibahas dalam penelitian sebagai berikut:

1. Hanya melakukan pemantauan dan pengendalian pada proses pengeringan biji kopi.
2. Tidak mengamati semua faktor yang mempengaruhi pengeringan namun hanya mengamati suhu dan kelembapan ruang pada mesin pengering biji kopi.
3. Biji kopi yang dikeringkan adalah jenis kopi robusta.
4. Tidak melakukan analisa lebih lanjut terhadap database pengeringan biji kopi.
5. Aplikasi *interface* hanya dapat dijangkau oleh pengguna android.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas maka tujuan penelitian ini adalah sebagai berikut:

1. Merancang aplikasi pantau dan kendali mesin pengering biji kopi berbasis IoT.
2. Membangun aplikasi pantau dan kendali *mobile* yang datanya diperoleh dari mesin pengering biji kopi berbasis IoT.
3. Mengevaluasi kinerja aplikasi pantau dan kendali mobile untuk mesin pengering kopi berbasis IoT.

1.5 Manfaat Penelitian

Berdasarkan uraian di atas, manfaat dari penelitian ini adalah sebagai berikut:

1. Meningkatkan kualitas produk dengan menghasilkan biji kopi dengan kualitas yang baik.

Internet of Things (IoT) adalah jaringan terintegrasi dari objek fisik atau hal-hal yang disimpan dalam perangkat elektronik, perangkat lunak, sensor, dan

jaringan internet, yang memungkinkan objek tersebut untuk mengumpulkan dan bertukar data. *Internet of Things* memungkinkan suatu objek untuk digunakan dan dikendalikan melalui *remote* pada infrastruktur jaringan yang ada, sehingga menciptakan peluang untuk menggabungkan dunia fisik dan sistem berbasis komputer langsung sehingga menghasilkan peningkatan efisiensi, akurasi dan manfaat ekonomi (Widiastuti & Mirnawati, 2020).

Melalui penggunaan layanan yang sesuai penerapan IoT memberikan solusi dan fungsi yang mengembangkan wawasan serta meningkatkan potensi serta kemampuan pemantauan dan pengendalian proses ataupun aset sebuah industri (Lampropoulos dkk., 2019). Hal tersebut membuat penerapan IoT dapat menunjang peningkatan kualitas dari suatu industri.

Salah satu bentuk penerapan teknologi berbasis IoT adalah penanganan pascapanen produk pertanian. Proses pengeringan merupakan tahapan pascapanen yang memerlukan pemantauan dan pengendalian suhu dan kelembapan udara. Hal ini dikarenakan suhu dan kelembapan sangat mempengaruhi lamanya proses pengeringan. Faktor cuaca yang cenderung berubah-ubah menjadi salah satu hal yang mempengaruhinya, sehingga diharapkan teknologi IoT mampu meningkatkan kualitas hasil pertanian dan membuat proses pengeringan lebih efisien dari segi waktu dan tenaga petani.

1. Mikrokontroler

Mikrokontroler adalah sebuah komputer kecil (*special purpose computer*) di dalam satu IC yang berisi CPU, memori, *timer*, saluran komunikasi serial dan paralel, port input/output, serta ADC. Mikrokontroler dapat kita gunakan untuk berbagai aplikasi misalnya untuk pengendalian, otomasi industri, akuisisi data, telekomunikasi, dan lain-lain. Keuntungan menggunakan mikrokontroler yaitu harganya murah, dapat diprogram berulang kali, dan dapat kita program sesuai dengan keinginan (Suhaeb dkk., 2017).

Mikrokontroler yang digunakan pada penelitian ini adalah NodeMCU. NodeMCU adalah sebuah board elektronik yang berbasis chip ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet melalui *WiFi*. NodeMCU ESP8266 merupakan modul turunan pengembangan dari modul platform IoT (*Internet of Things*) keluarga ESP8266 tipe ESP-12. Secara fungsi modul ini hampir menyerupai dengan platform modul arduino, tetapi yang membedakan yaitu dikhususkan untuk konek ke internet.



Gambar 4. Mikrokontroler NodeMCU ESP8266

NodeMCU memiliki beberapa pin I/O sehingga dapat dikembangkan menjadi sebuah aplikasi monitoring maupun controlling pada proyek IoT. nodeMCU ESP8266 dapat diprogram dengan *compiler* arduino yaitu menggunakan Arduino IDE. Bentuk fisik dari nodeMCU ESP8266, terdapat port USB (*mini USB*) sehingga akan memudahkan dalam pemrogramannya. Adapun spesifikasi dari NodeMCU adalah sebagai berikut:

Tabel 1. Spesifikasi NodeMCU ESP8266

Spesifikasi	NODEMCU V1
Mikrokontroler	ESP8266
Ukuran Board	57 mmx 30 mm
Tegangan Input	3.3 ~ 5V
GPIO	13 PIN
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4 MB
Clock Speed	40/26/24 MHz
WiFi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 Ghz
USB Port	Micro USB
USB to Serial Converter	CH340G

2. Sensor

Sensor adalah perangkat yang digunakan untuk mendeteksi perubahan besaran fisik seperti tekanan, gaya, besaran listrik, cahaya, gerakan, kelembaban, suhu, kecepatan dan fenomena-fenomena lingkungan lainnya. Input yang terdeteksi tersebut akan dikonversi mejadi output yang dapat dimengerti oleh manusia baik

melalui perangkat sensor itu sendiri ataupun ditransmisikan secara elektronik melalui jaringan untuk ditampilkan atau diolah menjadi informasi yang bermanfaat bagi penggunaannya. Adapun pada penelitian ini menggunakan dua sensor yaitu DHT22.



Gambar 5. Sensor DHT22

Sensor DHT-22 atau AM2302 merupakan sensor suhu dan kelembaban. Sensor ini memiliki keluaran berupa sinyal digital dengan konversi dan perhitungan dilakukan oleh MCU 8-bit terpadu. Sensor DHT22 memiliki rentang pengukuran suhu dan kelembaban yang luas, DHT22 mampu mentransmisikan sinyal keluaran melewati kabel hingga 20 meter sehingga sesuai untuk ditempatkan di mana saja.

Tabel 2. Spesifikasi sensor DHT22

Spesifikasi	DHT22
Tegangan input	3.3-6V DC
Signal Output	Signal digital via 1 kawat
Range	Suhu: -40°C – 80°C Kelembapan 0% - 100%
Akurasi	$\pm 2^{\circ}\text{C}$ (Suhu) $\pm 5\%$ RH (Kelembapan)
Ukuran	15.1 mm x 25 mm x 7.7 mm

3. Relay

Relay merupakan perangkat elektronika yang dapat menghubungkan atau memutuskan arus listrik yang besar dengan memanfaatkan arus listrik yang kecil. Selain itu relay merupakan saklar yang bekerja dengan menggunakan prinsip elektromagnet, dimana ketika ada arus lemah yang mengalir melalui kumparan, inti besi akan menjadi magnet. Setelah menjadi magnet, inti besi tersebut akan menarik jangkar besi sehingga kontak saklar akan terhubung dan arus listrik dapat mengalir. Sementara itu pada saat arus lemah yang masuk melalui kumparan diputuskan maka saklar akan terputus.

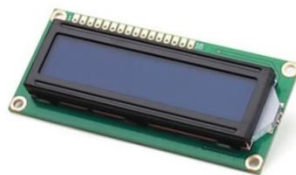


Gambar 6. Relay

Relay terdiri dari *coil* dan *contact*. *Coil* adalah gulungan kawat yang mendapat arus listrik. Sedangkan, *contact* adalah sejenis saklar yang dipengaruhi dari ada tidaknya arus listrik pada *coil*.

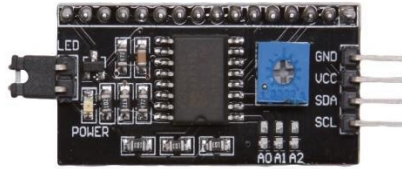
4. LCD 16x2 I2C

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. LCD (*Liquid Crystal Display*) bisa menampilkan suatu gambar/karakter dikarenakan terdapat banyak sekali titik cahaya (piksel) yang terdiri dari satu buah kristal cair sebagai titik cahaya. LCD 16x2 dapat menampilkan sebanyak 32 karakter yang terdiri dari 2 baris dan tiap baris dapat menampilkan 16 karakter.



Gambar 7. LCD 16x2

LCD 16x2 pada umumnya menggunakan 16 pin sebagai kontrolnya. Namun akan sangat boros apabila menggunakan 16 pin tersebut sehingga digunakan driver khusus sehingga LCD dapat dikontrol dengan modul I2C atau *Inter-Integrated Circuit*. Dengan modul I2C, maka LCD 16x2 hanya memerlukan dua pin untuk mengirimkan data dan dua pin untuk pemasok tegangan.



Gambar 8. Modul I2C

Dengan adanya modul I2C, LCD 16x2 hanya memerlukan empat pin yang perlu dihubungkan ke NodeMCU yaitu :

1. GND : Terhubung ke ground
2. VCC : Terhubung dengan 5V
3. SDA : Sebagai I2C data dan terhubung ke pin D2
4. SCL : Sebagai I2C data dan terhubung ke pin D1

1.6.2 Aktuator

Aktuator adalah perangkat yang mengubah sinyal listrik menjadi output fisik. Aktuator digunakan untuk menggerakkan atau mengontrol suatu sistem yang terhubung dan dikendalikan oleh program pada mikrokontroler. Adapun aktuator yang digunakan pada penelitian ini adalah sebagai berikut:

1. Motor Listrik

Motor listrik adalah alat untuk mengubah energi listrik menjadi energi mekanik. Perubahan ini dilakukan dengan mengubah tenaga listrik menjadi magnet yang disebut sebagai elektromagnet. Pada penelitian ini motor listrik digunakan untuk menggerakkan tabung/silinder pengeringan kopi. Motor listrik digunakan bersama dengan *gearbox* WPA 40 dengan rasio 1:30 sebagai pereduksi kecepatan pada motor listrik.



Gambar 9. Motor listrik

Adapun spesifikasi motor listrik yang digunakan pada penelitian ini adalah sebagai berikut:

1. Speed: 2900 RPM
2. Voltage: 220V/50Hz
3. Current: 1.5 A
4. Diameter as: 12 mm

2. Elemen Pemanas

Elemen pemanas merupakan alat yang mengubah energi listrik menjadi energi panas. Elemen pemanas memiliki berbagai jenis bentuk dan ukuran sesuai dengan aplikasinya pada industri.



Gambar 10. Elemen Pemanas

Gambar 10 menunjukkan Elemen Pemanas. Pada penelitian ini jenis elemen pemanas yang digunakan adalah *tubular heater*. *Tubular heater* adalah elemen pemanas dirancang berbentuk tabung dengan ukuran panjang dan diameter tertentu. Pada penelitian ini tubular heater berfungsi untuk memanaskan ruang pengeringan biji kopi sehingga mencapai suhu tertentu. Adapun spesifikasi dari *tubular heater* pada penelitian ini adalah sebagai berikut:

1. Tegangan: 220V
2. Daya: 300W
3. Diameter: 6,5 mm
4. Panjang: 290 mm

3. Kipas

Kipas merupakan suatu alat yang digunakan untuk menghasilkan angin guna mendinginkan udara. Pada penelitian ini kipas bertujuan untuk menurunkan suhu dalam mesin pengeringan.



Gambar 11. Kipas

penyimpanan data, pengiriman pesan dan laporan *error* atau *bug* aplikasi (Mahendra, 2019). Sehingga dengan berbagai kemampuan tersebut firebase dapat mempermudah para *developer* untuk mengembangkan suatu aplikasi yang berhubungan dengan beberapa perangkat.

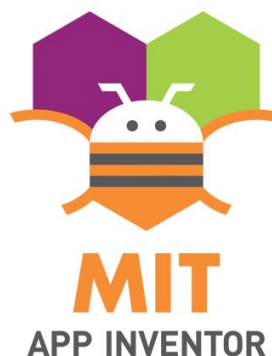


Gambar 13. Firebase

Gambar 13 menunjukkan Firebase. Pada penelitian ini fitur yang digunakan pada firebase adalah *realtime database*. Dengan menggunakan fitur tersebut data akan disimpan dan dieksekusi dalam bentuk JSON dan disinkronkan secara *realtime* ke setiap user yang terkoneksi. Data yang tersimpan pada firebase akan ditampilkan pada aplikasi android untuk memudahkan pemantauan dan pengendalian jarak jauh.

3. MIT App Inventor

MIT App Inventor merupakan platform berbasis web untuk memudahkan proses pembuatan aplikasi sederhana tanpa harus mempelajari atau menggunakan bahasa pemrograman yang terlalu banyak. MIT App Inventor awalnya dikembangkan oleh Google, dan saat ini dikelola oleh Massachusetts Institute of Technology.



Gambar 14. MIT App Inventor

Gambar 14 menunjukkan MIT App Inventor. Dengan app inventor, pengguna bisa melakukan pemrograman komputer untuk menciptakan aplikasi perangkat lunak dengan sistem operasi berbasis android. App inventor ini berbasis *visual*

block programming karena memungkinkan pengguna bisa menggunakan, melihat, menyusun dan *men-drag and drops block* yang merupakan simbol perintah dan fungsi *event handler* untuk menciptakan sebuah aplikasi yang bisa berjalan di sistem android.

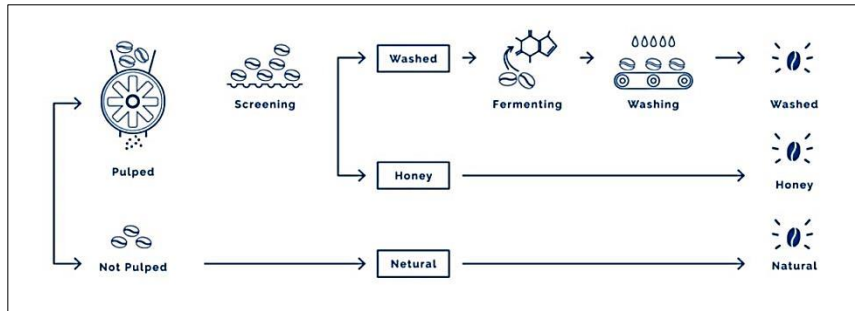
1.6.4 Integrasi IoT dalam Pertanian

Sistem pengering biji kopi berbasis Internet of Things (IoT) merupakan inovasi dalam proses pengeringan biji kopi yang memanfaatkan teknologi modern untuk meningkatkan efisiensi dan kualitas. Sistem ini mengintegrasikan mesin pengering biji kopi dengan berbagai komponen IoT seperti sensor, aktuator, dan platform komunikasi berbasis internet. Sensor suhu dan kelembapan, seperti DHT22, berperan penting dalam mengukur kondisi lingkungan di dalam ruang pengering, sementara aktuator seperti pemanas dan kipas digunakan untuk mengontrol suhu dan sirkulasi udara guna mencapai kondisi ideal pengeringan. Data yang dikumpulkan oleh sensor kemudian diproses oleh NodeMCU ESP8266, sebuah microcontroller dengan konektivitas Wi-Fi, yang mengirimkan informasi ini ke platform IoT seperti Firebase.

Firebase berfungsi sebagai basis data real-time, memungkinkan aplikasi mobile atau web untuk mengakses, menampilkan, dan menganalisis data yang relevan. Dengan sistem ini, pengguna dapat memantau kondisi pengeringan secara real-time, melakukan pengendalian jarak jauh, dan mendapatkan notifikasi serta analisis data historis. Keuntungan utama dari sistem ini meliputi peningkatan efisiensi proses pengeringan, kontrol kualitas biji kopi yang lebih baik, dan kemudahan dalam pemantauan dan pengendalian dari jarak jauh. Namun, tantangan seperti memastikan konektivitas yang stabil dan melindungi data dari akses yang tidak sah memerlukan perhatian khusus dan penerapan solusi yang tepat, seperti penggunaan jaringan Wi-Fi yang handal dan implementasi enkripsi data. Secara keseluruhan, sistem pengering biji kopi berbasis IoT menawarkan pendekatan yang modern dan efektif dalam meningkatkan proses pengeringan biji kopi, dengan dampak positif yang signifikan terhadap kualitas dan efisiensi operasional.

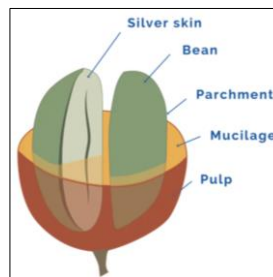
1. Metode Pengolahan

Pengolahan buah kopi pascapanen yang umum digunakan saat ini adalah *semi wash process*, *full washed process*, *honey process* dan *natural/dry process*. Berikut adalah gambaran perbedaan proses pengolahan biji kopi tiap metode pengolahan.



Gambar 15. Proses Biji Kopi

Gambar 15 menunjukkan proses pengolahan biji kopi. Proses basah (*washed*) air digunakan untuk mengupas buah ceri sampai ke bagian *mucilagenya*, sehingga hanya tersisa biji kopinya saja sebelum kopi tersebut dikeringkan. Pada metode *honey process* buah kopi dikupas dan dikeringkan dengan lapisan *mucilage* yang masih menyelimuti biji kopi tersebut. Pada saat tahap pengeringan lapisan *mucilage* tersebut membuat biji kopi menjadi semakin lengket yang mirip tekstur madu sehingga disebut *honey process*. Sedangkan, *natural process* adalah proses pengeringan kopi langsung dengan kulit paling luar di bawah sinar matahari. Berikut adalah gambaran lapisan pada biji kopi untuk mempermudah pemahaman.



Gambar 16. Lapisan Biji Kopi

Gambar 16 menunjukkan lapisan biji kopi. Adapun pada penelitian ini peneliti menggunakan biji kopi dengan pengolahan *honey process*. Kopi yang diolah dengan *honey process* cenderung memiliki cita rasa manis yang tinggi dibandingkan dengan proses pengolahan lainnya. Metode ini banyak dipakai di Amerika tengah dan disebut dengan kata *miel*, yang artinya madu sehingga saat ini dikenal dengan *honey process*. Belakangan proses ini juga semakin banyak digunakan oleh petani Indonesia. Hal ini dikarenakan kapasitas olah kecil, mudah dilakukan dan menggunakan peralatan sederhana.

Pada *honey process*, ceri kopi akan dikupas dengan mesin pulper untuk membuang kulit dan sebagian besar daging buahnya. Mesin pulper dapat diatur untuk menentukan seberapa banyak daging buah diingankan tetap melekat dengan

biji sebelum dijemur. Kemudian biji kopi dijemur di meja – meja pengering. Lamanya tahap pengeringan pada *honey process* sekitar 8-30 hari tergantung pada ketebalan lapisan lendir. Hal tersebut cukup menghabiskan waktu yang lama padahal pengeringan merupakan suatu proses penting yang terjadi dalam industri pangan.

Berdasarkan penjelasan di atas, adapun tahapan-tahapan yang ada pada *honey process* yaitu:

1. Pemetikan ceri kopi
2. Sortasi
3. Pengupasan menggunakan mesin pulper
4. Pengeringan

2. Standar Mutu Biji Kopi

Syarat mutu umum dan khusus biji kopi berdasarkan pada SNI biji kopi nomer 01-2907-2008 adalah sebagai berikut:

Tabel 3. Standar Mutu Umum Biji Kopi

No	Kriteria	Satuan	Persyaratan
1	Serangga hidup	-	Tidak ada
2	Biji berbau busuk dan berbau kapang	-	Tidak ada
3	Kadar air	% fraksi massa	Maks. 12.5%
4	Kadar kotoran	% fraksi massa	Maks. 0.5%

Sumber : (Badan Standardisasi Nasional, 2008)

Berdasarkan **Tabel 1** mutu umum biji kopi di atas maka kriteria seperti serangga hidup, biji berbau busuk dan berbau kapang diharapkan tidak ada agar biji kopi memenuhi SNI. Selain itu terdapat persyaratan untuk kadar air yaitu maksimal 12,5% dan kadar kotoran maksimal 0,5%.

Tabel 4. Standar Mutu Khusus Kopi Robusta Pengolahan Kering

Ukuran	Kriteria	Satuan	Persyaratan
Besar	Tidak lolos ayakan berdiameter 6,5 mm (Sieve No.16)	% fraksi massa	Maks lolos 5%
Sedang	Lolos ayakan diameter 6,5 mm, tidak lolos ayakan berdiameter 3,5 mm (Sieve No.9)	% fraksi massa	Maks lolos 5%

Sumber : (Badan Standardisasi Nasional, 2008)

Berdasarkan tabel di atas maka syarat mutu khusus kopi robusta pengolahan kering terbagi atas dua yaitu ukuran besar dan sedang. Adapun ukuran besar dengan kriteria yaitu tidak lolos ayakan berdiameter 6,5 mm (*Sieve No.16*) atau dengan persyaratan maksimal lolos 5% fraksi massa. Sedangkan ukuran sedang dengan kriteria yaitu lolos ayakan berdiameter 6,5 mm (*Sieve No.16*) dan tidak lolos ayakan berdiameter 3.5 mm (*Sieve No.9*) atau dengan persyaratan maksimal lolos 5% fraksi massa.

3. Metode Pengukuran Kadar Air



Gambar 17. Pengukur kadar air

Gambar 17 menunjukkan alat pengukur air. Kadar air adalah kandungan air yang terdapat dalam suatu bahan berupa biji-bijian ataupun material. Penelitian ini menggunakan Digital Grain Moisture sebagai alat pengukur kadar air biji kopi.

4. Metode Pengukuran Massa Biji Kopi



Gambar 18. Timbangan digital

Massa biji kopi adalah biji kopi yang diukur dalam satuan gram atau kilogram. Biji kopi dengan kadar air tinggi memiliki massa yang lebih besar karena air menambah massanya. Selama proses pengeringan, kadar air berkurang, menyebabkan penurunan massa bijikopi

5. Laju Pengeringan

Laju pengeringan terhadap suatu produk didapat dengan membandingkan perubahan berat produk yang terjadi dengan durasi proses pengeringan (Gultom, 2019). Laju pengeringan dapat dihitung dengan menggunakan persamaan berikut ini .

$$Q_a = \frac{W_{kb} - W_f}{N} \quad (1)$$

Dimana:

Q_a = laju pengeringan (gram/jam)

W_{kb} = massa biji kopi awal (gram)

W_f = massa biji kopi akhir (gram)

N = durasi pengeringan (jam)

1.6.5 Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan pengembang mengambil rangkuman dari beberapa penelitian terdahulu yang berkaitan, sebagai berikut:

1. Andry Petrus Launda, Dringhuzen J Mamahit, Elia Kendek Allo 2017 dengan judul Prototipe Sistem Pengering Biji Pala Berbasis Mikrokontroler Arduino Uno. Penelitian ini menggunakan sensor DHT22 untuk mengukur suhu dan kelembapan relatif. Kekurangan dari penelitian ini adalah pemantauan terbatas hanya melalui LCD dan tidak dapat dilakukan dengan jarak jauh.
2. Arshila Ariadna Dwirossi 2017, dengan judul penelitian Rancang Bangun Sistem Monitoring Kadar Air Biji Kopi Pada Mesin Pengering Biji Kopi Berbasis Penjejak Matahari Aktif Dengan Mikrokontroler Atmega16. Penelitian ini menggunakan objek penelitian yang sama yaitu mesin pengering biji kopi namun belum melakukan pemantauan dan pengendalian jarak jauh.
3. Anizar Indriani, Yovan Witanto, dan Hendra tahun 2019 dengan judul penelitian Pembuatan Alat Pengering Berputar (Rotary) Kopi Dan Lada Hitam Menggunakan Mikrokontroler Arduino Uno Desa Air Raman Kabupaten Kepahiang Propinsi Bengkulu. Penelitian ini menggunakan model mesin pengering yang sama yaitu model *rotary*.
4. Asyiva Nurbaeti, Mila Kusumawardani, dan Hendro Darmono tahun 2021 dengan judul Rancang Bangun Alat Pengering Biji Kopi Berbasis Internet Of Things. Penelitian ini menggunakan perangkat IoT yang sama namun

menggunakan model mesin pengeringan yang berbeda yaitu model box pengering.

5. Simon Santo T Gultom tahun 2019 dengan judul Rancang Bangun dan Pengujian Alat Pengering Biji Kopi Tenaga Listrik dengan Pemanfaatan Energi Surya. Penelitian ini menggunakan objek penelitian yang sama yaitu biji kopi serta menghitung kadar air berdasarkan massa awal dan akhir namun tidak melakukan pemantauan dan pengendalian berbasis *Internet of Things*.

1.6.6 Kerangka Konseptual

Kopi (*coffea sp*) adalah salah satu komoditas unggulan di Indonesia karena memiliki nilai jual yang cukup tinggi. Kopi memiliki kontribusi yang signifikan dalam perekonomian yaitu sebagai penghasil devisa, sumber pendapatan petani, penghasil bahan baku industri. membuka lapangan kerja dan pembangunan daerah.



Permasalahan umum yang ditemukan pada petani dan pedagang pengumpul biji kopi adalah kadar air dalam biji kopi yang tidak sesuai dengan SNI sehingga mempengaruhi nilai jual dan kualitas biji kopi. Hal tersebut disebabkan oleh minimnya penerapan teknologi pasca panen dan faktor cuaca yang terkadang tidak mendukung proses pengeringan.



Internet of Things merupakan sebuah teknologi yang memungkinkan adanya sebuah pengendalian, komunikasi, kerjasama dengan berbagai perangkat keras, dan pengiriman data melalui jaringan internet.



Berdasarkan hal di atas maka akan dibuatkan aplikasi pantau dan kendali *mobile* mesin pengeringan biji kopi berbasis *Internet of Things*.



Melalui aplikasi tersebut akan dilakukan pemantauan suhu, kelembapapan dan massa kopi. Adapun di dalam mesin dilengkapi dengan sensor DHT22 sebagai pendeteksi suhu, aplikasi mobile untuk input massa, adar air dan waktu pengeringan, relay untuk mengatur ON/OFF aktuator, dan mikrokontroler untuk menerima dan mengirim data.

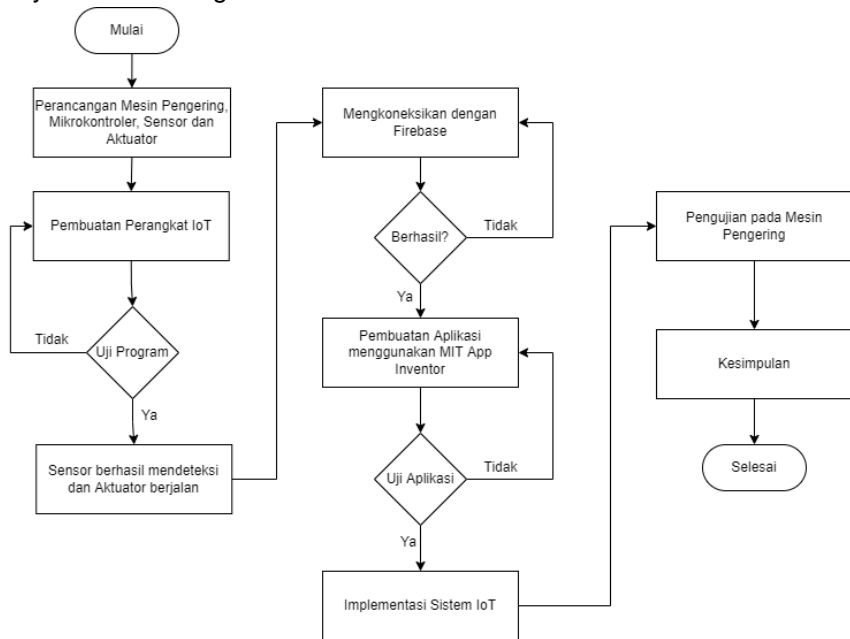
BAB II METODE PENELITIAN

2.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan mulai bulan Juni – Juli 2024. Penelitian ini dilakukan di BTN Asal Mula E2 No.10 Kelurahan Tamalanrea Indah, Kecamatan Tamalanrea, Kota Makassar.

2.2 Tahapan Penelitian

Tahapan penelitian ini menggunakan metode *waterfall*, adapun gambaran tahapan alur penelitiannya adalah sebagai berikut:



Gambar 19. Flowchart Alur Penelitian

Gambar 19 menjelaskan tentang alur penelitian. Proses dimulai dengan perancangan mesin pengering, termasuk komponen mikrokontroler, sensor, dan aktuator. Setelah perancangan, dilakukan pembuatan perangkat IoT yang bertujuan untuk menghubungkan sistem dengan jaringan internet. Setelah perangkat IoT dibuat, program diuji untuk memastikan bahwa sensor dan aktuator berfungsi dengan benar. Jika pengujian berhasil, proses dilanjutkan dengan mengoneksikan sistem ke Firebase. Setelah berhasil terhubung dengan Firebase, dibuatlah aplikasi mobile menggunakan MIT App Inventor untuk memantau dan mengendalikan mesin pengering melalui perangkat seluler. Aplikasi ini kemudian diuji untuk memastikan semua fungsinya berjalan sesuai harapan. Jika aplikasi lulus uji, sistem IoT diimplementasikan sepenuhnya. Setelah implementasi, sistem diuji

langsung pada mesin pengering untuk memastikan semua komponen bekerja sesuai spesifikasi. Akhirnya, proses ditutup dengan menyusun kesimpulan berdasarkan hasil pengujian, dan seluruh proyek dinyatakan selesai.

2.2.1 Analisis Kebutuhan

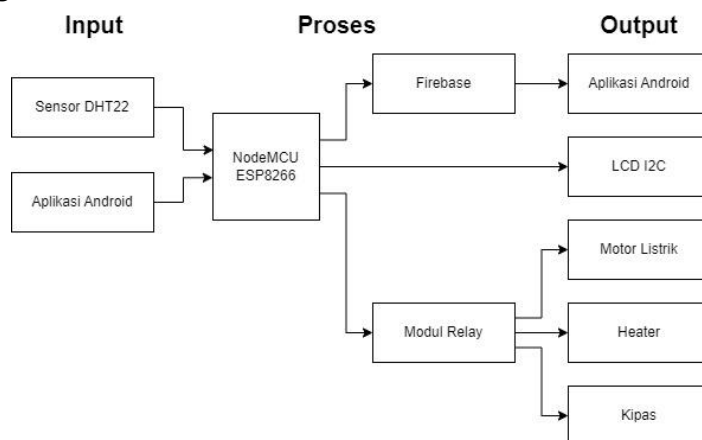
Pada tahap ini dilakukan analisis terhadap kebutuhan rancang bangun aplikasi pantau dan kendali *mobile* mesin pengering biji kopi berbasis *Internet of Things*.

Adapun hasil analisis kebutuhan untuk membangun sistem tersebut adalah sensor DHT22 untuk mendeteksi suhu dan kelembapan udara di dalam mesin pengering, aplikasi *mobile* untuk menginput massa, kadar air awal dan waktu pengeringan, motor listrik digunakan untuk menggerakkan silinder agar berputar sehingga pengeringan biji kopi merata, elemen pemanas untuk menaikkan suhu, kipas untuk menurunkan suhu, nodeMCU ESP8266 yang terhubung dengan internet untuk menerima dan mengirim data ke firebase, dan *smartphone* untuk melakukan pemantauan dan pengendalian mesin pengering.

2.2.2 Perancangan Sistem

Pada tahap ini dilakukan perancangan aplikasi pantau dan kendali mesin pengering kopi berbasis Internet of Things. Sistem dapat dioperasikan apabila sudah terhubung dengan *power supply*. Sensor yang terhubung dengan mikrokontroler kemudian otomatis mengirim data ke *firebase*. Data kemudian diteruskan ke aplikasi android beserta dengan data status aktuator (*ON/OFF*). Berdasarkan data pembacaan sensor tersebut juga mikrokontroler akan mengatur *ON/OFF* dari aktuator melalui modul relay. Dengan adanya aplikasi yang terpasang pada *smartphone* maka pengguna dapat memantau dan mengendalikan mesin dari jarak yang jauh.

1. Blok Diagram



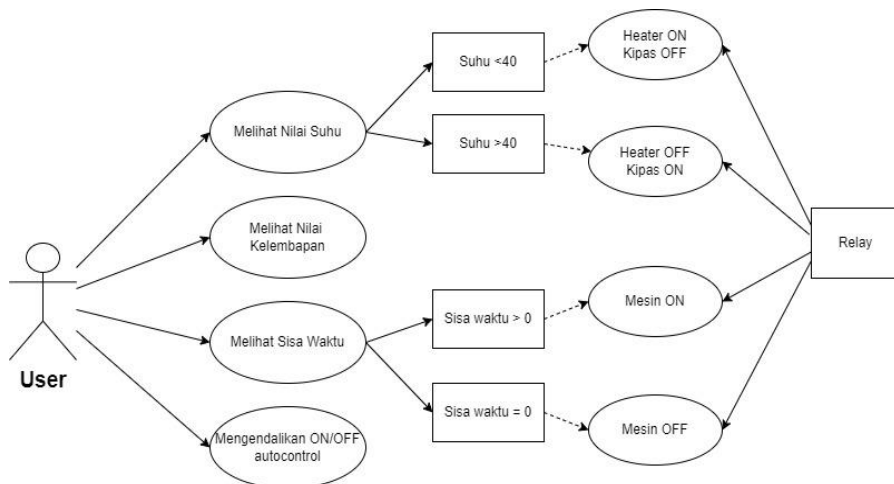
Gambar 20. Blok Diagram

Keterangan :

- Fungsi input : Sensor DHT22 mendeteksi suhu dan kelembapan udara di dalam mesin pengering dan aplikasi android menginput massa, kadar air dan waktu.
- Fungsi proses : Dilakukan penyimpanan data hasil deteksi sensor yang digunakan ke *firebase*, yang menjadi parameter sensor yakni suhu dan waktu untuk menjalankan perintah-perintah yang diberikan seperti mengaktifkan/mematikan modul *relay*.
- Fungsi output : Menampilkan hasil pembacaan dari sensor yang digunakan pada LCD dan aplikasi android yang telah dibuat.

2. Use Case Diagram

Use case diagram merupakan gambaran atau representasi dari interaksi yang terjadi antara user dan sistem. Sebuah sistem digambarkan ke dalam bentuk persegi berfungsi untuk membatasi use case dengan interaksi dari luar sistem.

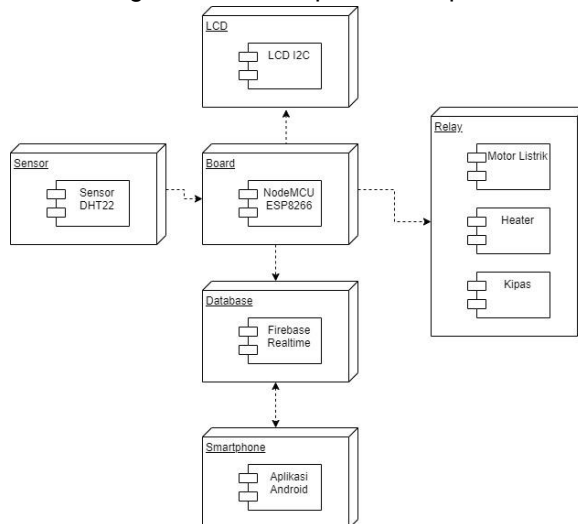


Gambar 21. Use case diagram

Pada **Gambar 21** diatas merupakan aktivitas keseluruhan yang terjadi yaitu *user* bisa memantau dan melakukan pengontrolan pada mesin pengering biji kopi. Aktivitas dalam *use case diagram* tersebut dimulai ketika *user* membuka aplikasi android yang dibuat untuk mesin pengering. Setelah itu maka akan muncul tampilan suhu, kelembapan dan sisa waktu untuk pengontrolan otomatis. User bisa menghidupkan/mematikan autocontrol yang digunakan sebagai mode manual untuk mengendalikan aktuatur.

3. Deployment Diagram

Deployment diagram dibuat untuk menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, dan keterhubungan antara komponen-komponen *hardware* tersebut.



Gambar 22. Deployment Diagram

Berdasarkan **Gambar 22**, terdapat sensor yang terhubung ke NodeMCU yang digunakan untuk mengetahui suhu dan kelembapan ruang di dalam mesin. Aplikasi android untuk menginput massa biji kopi, kadar air, dan waktu pengeringan ke firebase. Aplikasi android juga digunakan untuk memantau nilai suhu dan kelembapan ruang pengering serta mengendalikan aktuator. Program relay untuk mengaktifkan dan mematikan aktuator untuk menjaga kestabilan suhu di dalam mesin. Kemudian modul NodeMCU yang sudah terhubung dengan jaringan internet akan mengirim data ke *firebase* lalu sistem dapat dipantau pada aplikasi android yang telah dibuat.

2.2.3 Implementasi Sistem

Pada tahap ini dilakukan implementasi sesuai dengan perancangan sistem pada tahapan sebelumnya. Adapun pada tahap ini sistem akan dipasang pada mesin pengering sehingga dapat dilakukan pemantauan dan pengendalian melalui aplikasi *mobile*.

2.2.4 Pengujian

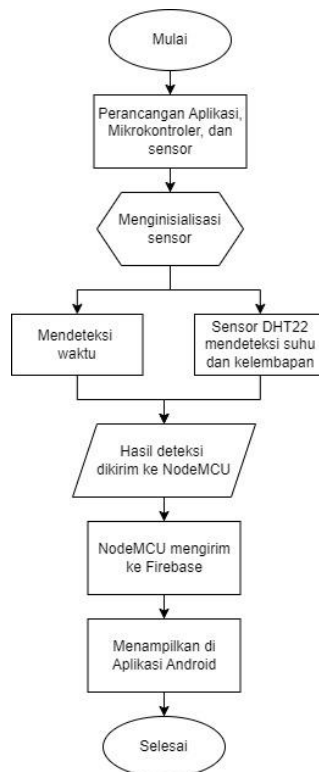
Pada tahap ini akan dilakukan pengujian kinerja pada sistem setelah tahap implementasi. Adapun beberapa pengujian yang akan dilakukan adalah sebagai berikut:

1. Menghidupkan perangkat sistem
2. Hasil pembacaan sensor pada sistem
3. Pengiriman data dari sensor ke firebase
4. Pemantauan dan pengendalian mesin pengering melalui aplikasi *mobile*

2.2.5 Hasil

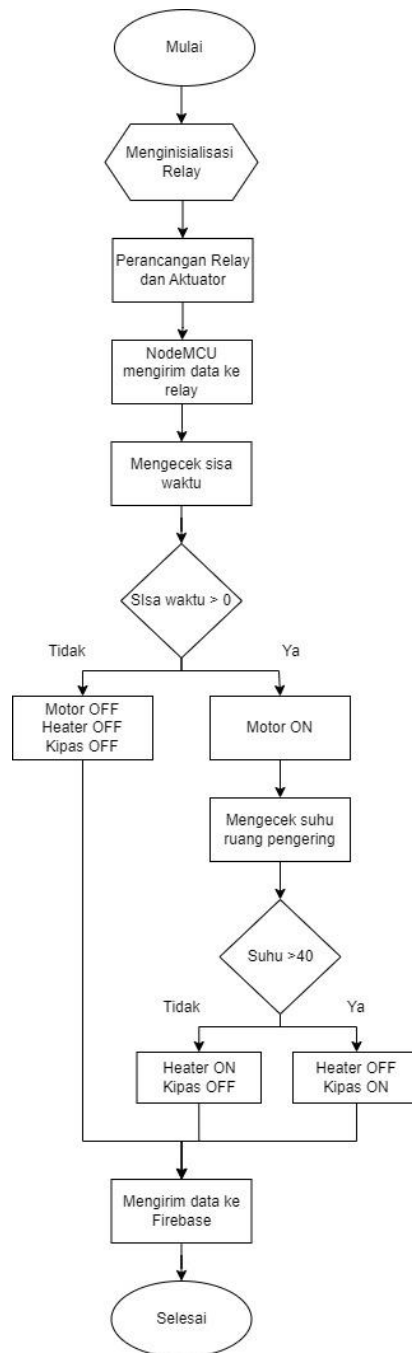
Tahap ini merupakan tahapan akhir dari keseluruhan tahap penelitian. Pada tahap ini akan diperoleh hasil akhir dari sistem yang telah dibangun sehingga penelitian ini dapat memperoleh beberapa kesimpulan.

2.3 Arsitektur Sistem



Gambar 23. Flowchart Sistem Pantau

Berdasarkan **Gambar 23.** ditunjukkan bahwa sistem pantau dimulai dari perancangan aplikasi, mikrokontroler, dan sensor. Selanjutnya dilakukan inisialisasi sensor sehingga di tahap selanjutnya sensor dapat mendeteksi nilai suhu, kelembapan dan waktu pengeringan. Hasil pembacaan sensor tersebut kemudian dikirimkan ke NodeMCU yang selanjutnya bertugas mengirimkan data tersebut ke Firebase. Data dari Firebase tersebut kemudian ditampilkan pada aplikasi android untuk memudahkan pemantauan jarak jauh.



Gambar 24. Flowchart Sistem Kendali

Berdasarkan **Gambar 24** sistem kendali dimulai dari menginisialisasi relay. Setelah tahap tersebut berhasil maka dilakukan perancangan relay dan aktuator.

Selanjutnya NodeMCU mengirim data hasil pantau kepada relay. Jika sisa waktu > 0 maka Motor akan ON. Jika suhu >40 maka heater OFF dan Kipas ON. Sebaliknya jika suhu ≤ 0 maka Heater ON dan Kipas OFF. Jika sisa waktu tidak lebih dari 0 maka semua aktuator akan OFF.

2.4 Sumber Data

Dalam penelitian ini, data yang diperoleh bersumber dari data primer yaitu hasil pembacaan sensor yang digunakan pada aplikasi pantau dan kendali *mobile* mesin pengering biji kopi berbasis *Internet of Things*.

2.5 Instrumen Penelitian

Instrumen penelitian ini meliputi kebutuhan perangkat lunak dan perangkat keras yaitu sebagai berikut:

1. Kebutuhan Perangkat Lunak:
 - Operating System Windows 10
 - Arduino IDE
 - Firebase
 - MIT App Inventor
2. Kebutuhan Perangkat Keras:
 - Mikrokontroler NodeMCU ESP8266
 - Sensor DHT22
 - Sensor Loadcell
 - Relay
 - Kabel Jumper
 - Kabel Elemen
 - Kabel Listrik
 - Power Supply
 - Adaptor
 - Heater
 - Kipas
 - Motor Listrik
 - Isolator
 - Kerangka
 - Mesin Pengering *rotary*
 - Laptop
 - Smartphone

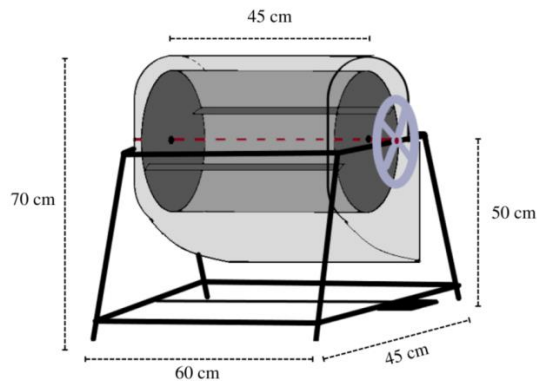
BAB III

HASIL DAN PEMBAHASAN

3.1 Hasil Rancangan Prototipe

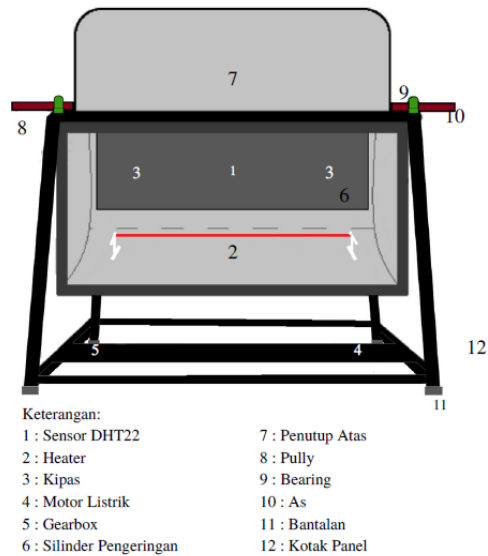
Pada penelitian ini akan dilakukan perancangan prototipe yang terbagi atas beberapa bagian yaitu sebagai berikut:

3.1.1 Rancangan Prototipe Mesin Pengering



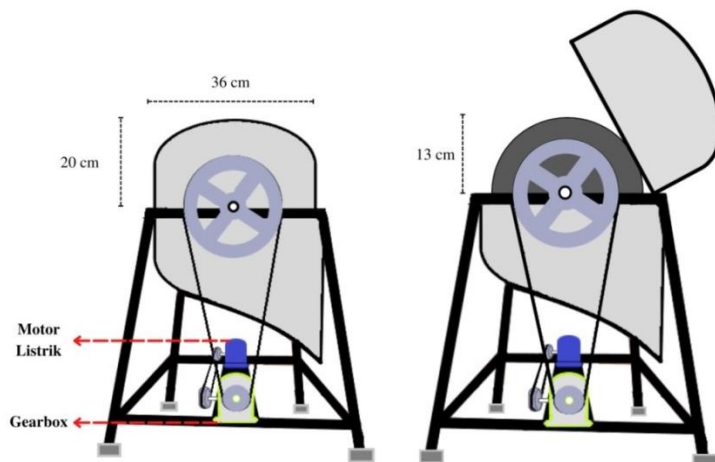
Gambar 25. Rancangan prototipe mesin pengering

Gambar 25 menunjukkan desain mesin pengering biji kopi berbentuk silinder dengan panjang rangka besi 60 cm dan lebar 45 cm, didukung oleh rangka penopang setinggi 50 cm. Panjang ruang pengering adalah 45 cm. Mesin ini dilengkapi dengan mekanisme putar yang digerakkan oleh motor listrik untuk memastikan pengeringan biji kopi yang merata.



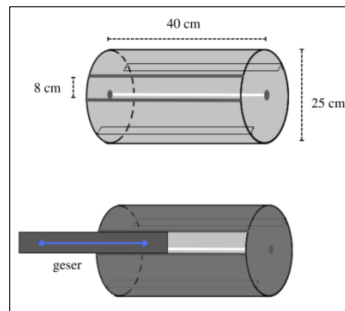
Gambar 26. Rancangan letak komponen pada mesin pengering

Gambar menunjukkan letak komponen pada mesin pengering yang dilengkapi dengan sensor DHT22 untuk mengukur suhu dan kelembaban, heater untuk menghasilkan panas, kipas untuk sirkulasi udara, motor listrik untuk menggerakkan komponen, gearbox untuk mengatur kecepatan dan torsi, serta silinder pengeringan untuk menampung biji kopi. Komponen tambahan termasuk penutup atas untuk akses, pully untuk transmisi daya, bearing untuk mengurangi gesekan, as untuk mentransmisikan putaran, dan kotak panel untuk kontrol dan pengaturan mesin secara otomatis melalui aplikasi mobile.



Gambar 27. Rancangan prototipe mesin pengering tampak samping dan penutup atas terbuka

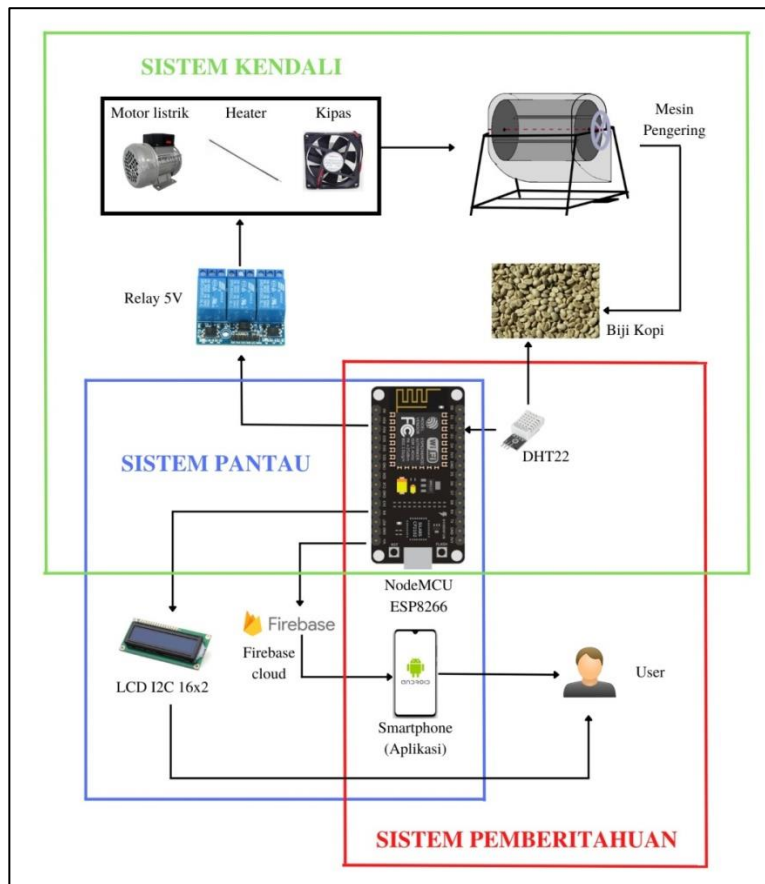
Gambar 27 menunjukkan rancangan prototipe tampak samping. Adapun tinggi ruang pengering dari rangka adalah 20 cm dan lebar 36 cm. Selanjutnya jari-jari tabung pengering adalah 13 cm. Alat ini digerakkan oleh motor listrik dan gearbox untuk memperlambat putaran. Gambar menunjukkan tampilan mesin saat penutup atasnya terbuka.



Gambar 28. Rancangan silinder wadah kopi dalam ruang pengering

Gambar 28 menunjukkan rancangan silinder wadah kopi dalam ruang pengering. Diameter silinder adalah 25 cm dengan panjang 40 cm. Penutup wadah memiliki ukuran 8 cm dan dibuka dengan cara digeser.

3.1.2 Rancangan Prototipe Mekanik



Gambar 29. Rancangan prototipe mekanik

Gambar 29 menunjukkan sistem pengering biji kopi berbasis IoT yang terbagi menjadi tiga subsistem utama: sistem kendali, sistem pantau, dan sistem pemberitahuan. Sistem kendali terdiri dari motor listrik, heater, dan kipas yang diatur oleh relay 5V untuk mengoperasikan mesin pengering. Sistem pantau menggunakan sensor DHT22 untuk mengukur suhu dan kelembaban, kemudian data dikirimkan ke NodeMCU ESP8266 yang terhubung dengan Firebase cloud dan ditampilkan pada LCD I2C 16x2. Sistem pemberitahuan memungkinkan pengguna untuk memantau dan mengendalikan mesin pengering melalui aplikasi smartphone berbasis Android, yang menerima data real-time dari NodeMCU melalui Firebase.

3.1.3 Rancangan Prototipe Perangkat Lunak

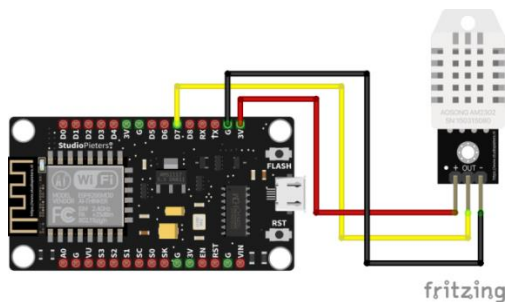


Gambar 30. Rancangan prototipe perangkat lunak

Gambar 30 menunjukkan bahwa data yang akan dipantau pada aplikasi yaitu suhu, kelembapan, sisa waktu pengeringan dan status aktuator. Selain itu aktuator juga dapat dikendalikan dari aplikasi.

3.2 Hasil Implementasi Perangkat Keras Mesin Pengering Biji Kopi Berbasis *Internet of Things*

3.2.1 Rangkaian NodeMCU dan DHT22

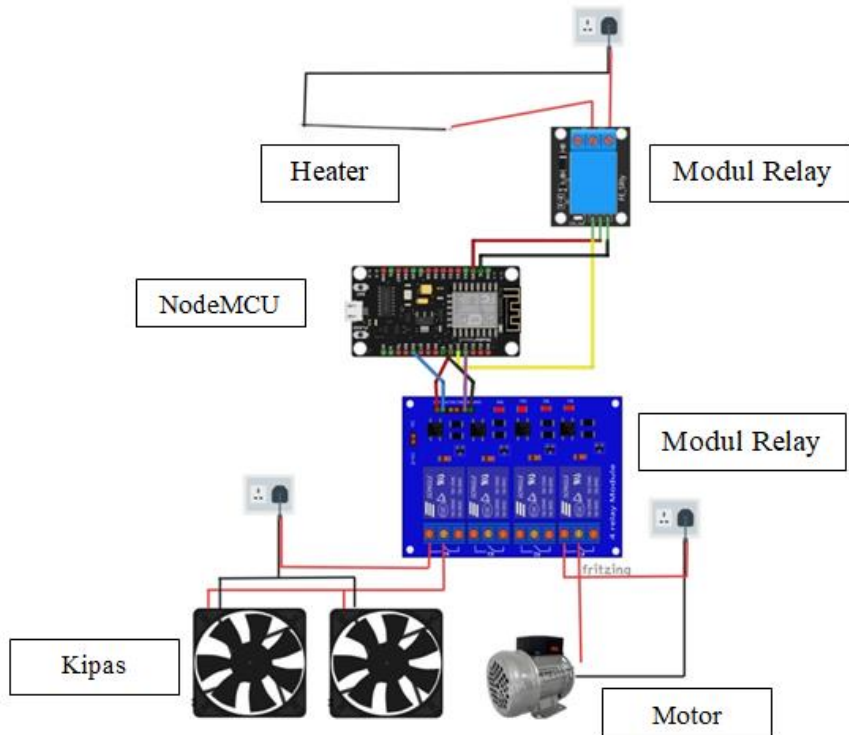


Gambar 31. Rangkaian NodeMCU dan DHT22

Gambar 31 menunjukkan rangkaian NodeMCU dan sensor DHT22. Komponen-komponen pada penerapan NodeMCU dengan sensor suhu dan kelembapan pada sistem ini yaitu satu buah NodeMCU yang terhubung dengan sebuah sensor DHT22. Sensor suhu yang terhubung dengan NodeMCU berfungsi mengukur suhu dan kelembapan pada ruang mesin pengering. Besar nilai suhu

dan kelembapan yang terbaca didapat dari sensor DHT22 yang disimpan didalam mesin pengering.

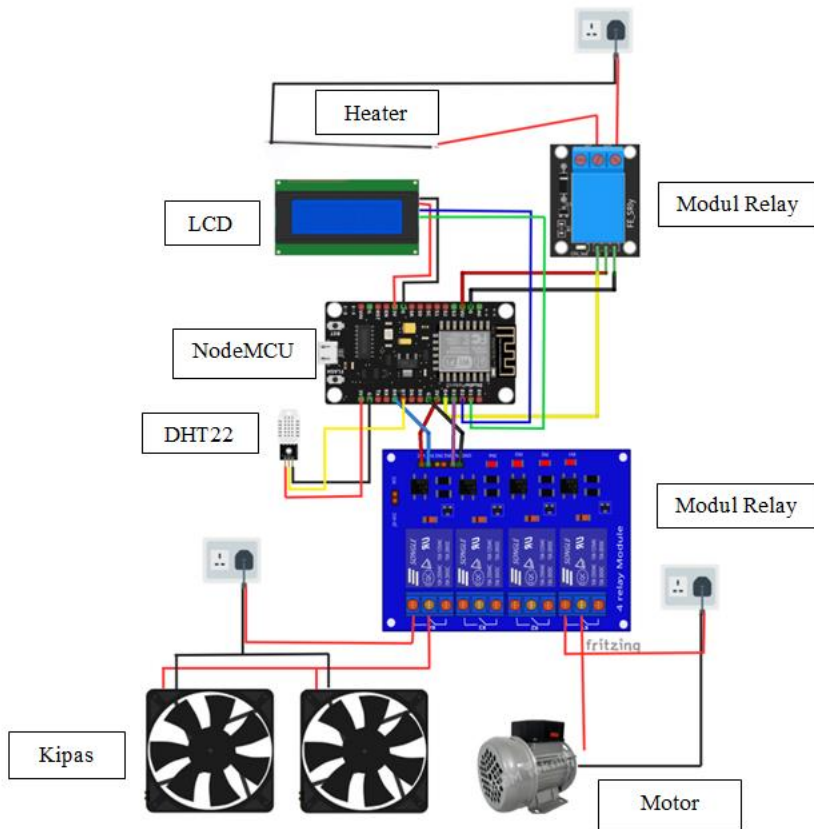
3.2.2 Rangkaian NodeMCU dan relay



Gambar 32. Rangkaian NodeMCU dan relay

Gambar 32 menunjukkan rangkaian kontrol mesin pengering biji kopi berbasis IoT. NodeMCU berfungsi sebagai pengontrol utama yang terhubung ke modul relay. Modul relay mengendalikan daya untuk heater, kipas, dan motor. Heater terhubung ke relay 1 channel, sedangkan kipas dan motor terhubung ke relay pada modul relay 4 channel. Relay ini mengaktifkan atau menonaktifkan perangkat sesuai perintah dari NodeMCU, memungkinkan kontrol otomatis melalui koneksi internet

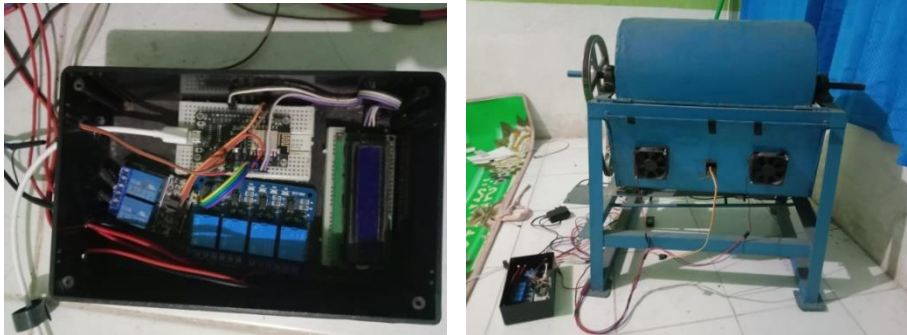
3.2.3 Rangkaian Keseluruhan Perangkat Keras Sistem



Gambar 33. Rangkaian Keseluruhan Perangkat Keras Sistem

Gambar 33 menunjukkan rangkaian kontrol mesin pengering biji kopi berbasis IoT. NodeMCU berfungsi sebagai pengontrol utama yang terhubung ke modul relay. Modul relay mengendalikan daya untuk heater, kipas, dan motor. Relay ini mengaktifkan atau menonaktifkan perangkat sesuai perintah dari NodeMCU, memungkinkan kontrol otomatis melalui koneksi internet. Selain itu, terdapat sensor DHT22 yang mengukur suhu dan kelembaban, serta LCD I2C yang menampilkan informasi real-time dari sensor. Data dari sensor DHT22 dikirim ke NodeMCU untuk memantau dan mengontrol kondisi pengeringan, sementara LCD I2C memberikan visualisasi langsung kepada pengguna.

1. Hasil Pembangunan Perangkat Keras Sistem



Gambar 34. Hasil rancangan Elektronik

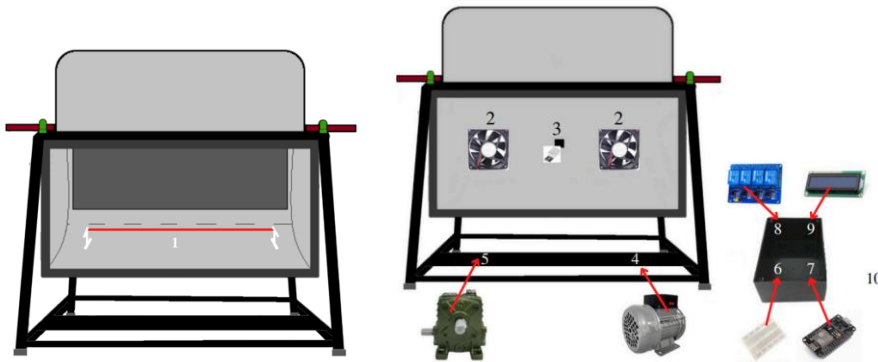
Gambar 34 menunjukkan hasil rancangan elektronik dalam kotak panel dan mesin pengering tampak luar.

Tabel 5. Daftar Alat dan harga Perangkat Keras

No	Nama alat	Jumlah	Harga
1	NodeMCU ESP8266	1	Rp. 42.000
2	DHT22	1	Rp. 32.000
3	LCD I2C	1	Rp 40.000
4	Relay 4 Channel	1	Rp 37.000
5	Relay 2 Channel	1	Rp. 15.000
6	Kipas DC	2	Rp. 27.000
7	Heater	1	Rp. 80.000
8	Motor Listrik	1	Rp. 600.000
9	Mesin Pengering	1	Rp. 800.000
10	Gearbox	1	Rp. 390.000
11	Breadboard	1	Rp. 12.000
12	Kabel Jumper	21	Rp. 10.000
13	Kabel Elemen	3	Rp. 33.000
14	Kabel listrik	2	Rp. 100.000
15	Isolator	2	Rp. 110.000
Total			Rp. 2.328.000

Berdasarkan **Gambar 34** dan **Tabel 5**, *NodeMCU ESP8266* digunakan sebagai *microcontroller*. Sensor DHT22 untuk membaca suhu dalam ruang pengeringan. *Breadboard* digunakan untuk menghubungkan alat satu sama lain. Modul relay berfungsi sebagai saklar yang mengatur ON/OFF dari alat. *Heater* digunakan sebagai pemanas dalam mesin pengering. Kipas untuk menurunkan suhu mesin pengering apabila suhunya terlalu tinggi.

2. Implementasi Perangkat Keras Sistem



Gambar 35. Ilustrasi Letak Alat Pada Mesin Pengering

Gambar 35 menunjukkan sebagai berikut:

1. *Heater*, diletakkan di bagian bawah ruang pengeringan dan berfungsi menghasilkan panas untuk menaikkan suhu dalam sistem.
2. Kipas, diletakkan di penutup depan mesin pengering dan berfungsi untuk menurunkan suhu dalam sistem.
3. Sensor DHT22, diletakkan dibagian tutup depan dan diarahkan ke dalam ruang pengeringan. Berfungsi mengukur suhu dan kelembapan udara untuk pemantauan kondisi sistem.
4. Motor Listrik, diletakkan dibagian bawah mesin. Berfungsi mengonversi energi listrik menjadi energi mekanik untuk menggerakkan mesin pengering.
5. *Gearbox*, diletakkan di bawah mesin pengering dan berfungsi mengurangi kecepatan motor.
6. *Breadboard*, berada di dalam kotak panel dan berfungsi sebagai papan untuk membuat rangkaian elektronik sementara tanpa soldering.
7. Nodemcu ESP8266, berada dalam kotak panel dan berfungsi sebagai mikrokontroler yang memungkinkan konektivitas internet untuk kontrol dan monitoring jarak jauh.
8. Modul Relay, berada di dalam kotak panel dan berfungsi mengontrol perangkat berdaya tinggi. Dalam penelitian ini yaitu motor listrik, heater, dan kipas.
9. LCD I2C berada dalam kotak panel berfungsi sebagai layar tampilan yang menggunakan protokol I2C untuk menghemat pin mikrokontroler. Dalam penelitian ini berfungsi menampilkan data seperti suhu dan kelembapan
10. Kotak Panel berfungsi sebagai tempat untuk mengatur dan melindungi komponen listrik atau elektronik sistem, menyediakan ruang yang aman untuk pemasangan, penghubungan, dan pengorganisasian komponen seperti saklar, relay, dan pengontrol, serta membantu mencegah kontak langsung dengan bagian berbahaya dan melindungi komponen dari debu, kelembapan, atau kerusakan fisik.



Tampilan Depan



Tampilan Samping



Tampilan Atas



Tampilan saat mesin terbuka

Gambar 36. Implementasi Perangkat Keras Mesin pengering

Gambar 36 merupakan implementasi mesin pengering biji kopi meliputi pengujian *hardware*. Sensor mengukur suhu dan kelembaban ruang, sensor DHT22 diletakkan mengarah ke dalam mesin guna mengukur suhu ruang mesin. Rangkaian alat kemudian dihubungkan ke sakelar menggunakan kabel data dan adaptor. *Microcontroller NodeMCU* akan menerima koneksi *WiFi*, kemudian pengujian dilakukan dengan memasukkan biji kopi ke dalam mesin pengering.

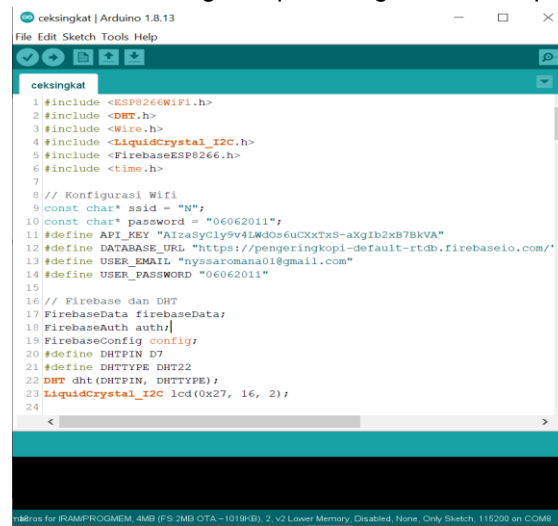
Adapun Cara Kerja dari Mesin pengering biji kopi *Internet of Things* sebagai berikut:

1. Timbang dan cek kadar air awal biji kopi yang akan dilakukan pengeringan.
2. Hubungkan mesin ke listrik dan *wifi*
3. Jika sudah terhubung, masukkan gabah yang telah ditimbang tadi untuk memulai pengeringan.
4. *Input* massa awal, kadar air awal yang didapatkan dilangkah awal pada aplikasi *mobile*, dan waktu pengeringan
5. Tunggu sampai waktu habis
6. Setelah waktu habis maka mesin akan otomatis mati.
7. Lakukan pengukuran kembali massa dan kadar air biji kopi.

3.3 Hasil Implementasi Perangkat Lunak Pantau dan Kendali *Mobile* Mesin Pengering Biji Kopi Berbasis *Internet of Things*

3.3.1 Pembangunan Perangkat Lunak Menggunakan *Arduino IDE*

Arduino IDE (Integrated Development Environment) digunakan untuk memprogram mikrokontroler NodeMCU ESP8266 agar dapat mengirim data ke perangkat lunak.



Gambar 37.Tampilan *Arduino IDE*

Berdasarkan **Gambar 37**, tampilan *Arduino IDE* dapat terintegrasi dan berfungsi secara realtime dengan *Internet of Things*, maka seluruh perangkat seperti *Board NodeMCU*, sensor DHT 22 , modul relay, dan aktuator diberikan perintah *coding* bahasa C. Melalui *Arduino IDE* atau (*Integrated Development Environment*) dilakukan pemrograman untuk melakukan fungsi-fungsi melalui sintaks pemrograman.

```

#include <ESP8266WiFi.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <FirebaseESP8266.h>
#include <time.h>

```

Gambar 38. Program Pendefinisian *Library* yang Digunakan

Kode di atas bertujuan untuk mendefinisikan *library* yang digunakan. Adapun *library* digunakan yaitu :

1. *Library ESP8266WiFi.h*, digunakan untuk menghubungkan ESP8266 ke jaringan WiFi dan melakukan komunikasi melalui internet.

2. *Library DHT.h*, digunakan untuk membaca data suhu dan kelembapan dari sensor DHT.
3. *Library Wire.h*, digunakan untuk mengelola komunikasi I2C dengan perangkat lain seperti sensor dan display.
4. *Library LiquidCrystal_I2C.h*, digunakan untuk mengontrol LCD dengan antarmuka I2C untuk menampilkan teks dan data.
5. *Library FirebaseESP8266.h*, digunakan untuk menghubungkan ESP8266 ke Firebase untuk membaca dan menulis data ke Realtime Database.
6. *Library time.h*, digunakan untuk mendapatkan waktu dari server NTP

```
const char* ssid = "N";
const char* password = "06062011";
#define API_KEY "AlzaSyC1y9v4LWdOs6uCXxTxS-aXglb2xBkVA"
#define DATABASE_URL "https://pengeringkopi-default-rtdb.firebaseio.com/"
#define USER_EMAIL "nyssaromana01@gmail.com"
#define USER_PASSWORD "06062011"
```

Gambar 39. Program Inisialisasi wifi dan firebase

*Const char** digunakan untuk membuat pointer konstan ke string, yang berarti nilai-nilai ini tidak dapat diubah selama program berjalan. *#define* digunakan untuk mendefinisikan nilai konstan yang dapat digunakan di seluruh kode dalam hal ini digunakan untuk *API_KEY*, *DATABASE_URL*, *USER_EMAIL*, dan *USER_PASSWORD*.

```
FirestoreData firestoreData;
FirestoreAuth auth;
FirestoreConfig config;
```

Gambar 40. Program deklarasi firebase

Deklarasi tersebut mendefinisikan objek yang diperlukan untuk berinteraksi dengan Firebase pada proyek ESP8266 Anda: *FirestoreData firestoreData* untuk menangani komunikasi data, *FirestoreAuth auth* untuk mengelola autentikasi pengguna, dan *FirestoreConfig config* untuk menyimpan konfigurasi koneksi ke Firebase.

```
#define DHTPIN D7
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Gambar 41. Program Inisialisasi pin DHT22 dan LCD

Konfigurasi ini menetapkan pin dan jenis sensor DHT serta menginisialisasi LCD I2C untuk tampilan.

```
const int relayPins[] = {0, 2, 15};
bool relayStates[] = {false, false, false}; // {dinamo, heater, kipas}
```

Gambar 42. Program Inisialisasi pin relay

Const int relayPins[] = {0, 2, 15}; menetapkan pin digital 0, 2, dan 15 yang akan digunakan untuk mengendalikan relay yang terhubung ke perangkat seperti dinamo, pemanas, dan kipas. *bool relayStates[] = {false, false, false};* menyimpan status awal dari masing-masing relay, di mana nilai *false* menunjukkan bahwa semua relay dalam keadaan mati.

```
float hum, temp;
int sisaMenit, sisaDetik;
char timestamp[30];
bool autoControl = true;
```

Gambar 43. Program Inisialisasi dan deklarasi variabel

Gambar 43 adalah program untuk menginisialisasi dan mendeklarasikan nilai awal dari variabel-variabel yang digunakan yaitu:

1. *float hum, temp;* digunakan untuk mendeklarasikan variabel *hum* dan *temp* untuk menyimpan nilai suhu dan kelembapan dalam tipe data float.
2. *int sisaMenit, sisaDetik;* untuk mendeklarasikan variabel *sisaMenit* dan *sisaDetik* untuk menyimpan sisa menit dan detik dalam tipe data integer.
3. *char timestamp[30];* digunakan untuk mendeklarasikan array *timestamp* dengan ukuran 30 karakter untuk menyimpan data waktu atau stempel waktu.
4. *bool autoControl = true;* digunakan untuk mendeklarasikan variabel boolean *autoControl* yang diinisialisasi dengan nilai *true*, menunjukkan bahwa kontrol otomatis diaktifkan.

```
void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");

  dht.begin();
  Wire.begin(D2, D1);
  lcd.begin();
  lcd.backlight();

  for (int i = 0; i < 3; i++) {
```

```
pinMode(relayPins[i], OUTPUT);
}
```

Gambar 44. Program Inisialisasi Koneksi WiFi dan Perangkat I/O

Gambar 44 adalah program menginisialisasi komunikasi serial dengan kecepatan 115200 bps, menghubungkan ESP8266 ke jaringan WiFi dengan nama dan kata sandi yang telah ditentukan, dan menunggu hingga koneksi berhasil. Setelah itu, program memulai sensor DHT, komunikasi I2C, dan LCD I2C dengan backlight. Program juga mengatur tiga pin relay sebagai output untuk mengendalikan perangkat yang terhubung.

```
config.api_key = API_KEY;
auth.user.email = USER_EMAIL;
auth.user.password = USER_PASSWORD;
config.database_url = DATABASE_URL;
Firebase.begin(&config, &auth);

configTime(8 * 3600, 0, "asia.pool.ntp.org", "time.nist.gov");
while (!time(nullptr)) {
    delay(1000);
    Serial.print(".");
}
Serial.println("NTP time synchronized");
}
```

Gambar 45. Program Konfigurasi Koneksi Firebase dan Sinkronisasi Waktu

Pada bagian ini, program mengonfigurasi koneksi ke Firebase dengan menetapkan kunci API, email pengguna, kata sandi, dan URL database yang diperlukan untuk autentikasi dan akses data. Kemudian, program memulai koneksi ke Firebase menggunakan konfigurasi tersebut. Selanjutnya, program mengatur waktu dengan server NTP (Network Time Protocol) untuk mendapatkan waktu yang akurat, menunggu hingga waktu berhasil disinkronkan, dan menampilkan status sinkronisasi waktu melalui komunikasi serial.

```
void fetchData() {
    if (Firebase.getString(firebaseData, "/Data/waktu"))    sisaMenit    =
firebaseData.stringData().toInt();
    if (Firebase.getString(firebaseData, "/Data/detik"))    sisaDetik    =
firebaseData.stringData().toInt();
    if (Firebase.getBool(firebaseData, "/Data/Mesin"))    relayStates[0] =
firebaseData.boolData();
    if (Firebase.getBool(firebaseData, "/Data/Heater"))    relayStates[1] =
firebaseData.boolData();
    if (Firebase.getBool(firebaseData, "/Data/Kipas"))    relayStates[2] =
```



```

firebaseData.boolData();
if      (Firebase.getBool(firebaseData,      "/Data/auto"))      autoControl      =
firebaseData.boolData();

Serial.println("Data berhasil diambil dari Firebase:");
Serial.print("Sisa Menit: "); Serial.println(sisaMenit);
Serial.print("Sisa Detik: "); Serial.println(sisaDetik);
Serial.print("Mesin (Relay 1): "); Serial.println(relayStates[0] ? "ON" : "OFF");
Serial.print("Heater (Relay 2): "); Serial.println(relayStates[1] ? "ON" : "OFF");
Serial.print("Kipas (Relay 3): "); Serial.println(relayStates[2] ? "ON" : "OFF");
Serial.print("Auto Control: "); Serial.println(autoControl ? "ON" : "OFF");
}

```

Gambar 46. Program Fungsi Pengambilan dan Penampilan Data dari Firebase

Fungsi *fetchData()* mengambil data dari Firebase dan memperbarui variabel-variabel yang relevan dengan nilai-nilai yang diterima. Ia mengambil waktu sisa dalam menit dan detik dari Firebase, serta status *boolean* untuk tiga relay (Mesin, Heater, Kipas) dan status kontrol otomatis. Setelah mendapatkan data, fungsi ini mencetak nilai-nilai yang diperoleh ke serial monitor untuk memantau status saat ini.

```

void updateFirebaseData(const String &path, bool value) {
  if (Firebase.setBool(firebaseData, path, value)) {
    Serial.print("Status "); Serial.print(path); Serial.println(" berhasil dikirim ke
Firebase");
  } else {
    Serial.print("Gagal mengirim status "); Serial.print(path); Serial.println(" ke
Firebase");
    Serial.println(firebaseData.errorReason());
  }
}

```

Gambar 47. Program Fungsi update data firebase

Fungsi *updateFirebaseData()* mengirimkan nilai boolean ke Firebase pada jalur yang ditentukan oleh parameter *path*. Fungsi ini mencoba mengatur nilai *boolean* pada jalur tersebut menggunakan *Firebase.setBool()*. Jika pengiriman berhasil, fungsi mencetak pesan keberhasilan ke serial monitor. Jika gagal, fungsi mencetak pesan kesalahan dan alasan kegagalan yang diperoleh dari *firebaseData.errorReason()*.

```

hum = dht.readHumidity();
temp = dht.readTemperature();

// Print the results to the Serial Monitor

```

```

Serial.print("Humidity: ");
Serial.print(hum);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(temp);
Serial.println(" *C");

lcd.clear();
lcd.setCursor(0, 0);
lcd.printf("Temp: %.2f C", temp);
lcd.setCursor(0, 1);
lcd.printf("Humid: %.2f %%", hum);
delay(2500);

```

Gambar 48. Program pembacaan dan menampilkan suhu dan kelembapan pada serial monitor dan LCD

Fungsi ini membaca data kelembapan dan suhu dari sensor DHT, lalu menampilkan hasilnya di serial monitor dan LCD.

```

if (Firebase.setFloat(firebaseData, "/Data/suhu", temp)) {
  Serial.println("Suhu berhasil dikirim ke Firebase");
} else {
  Serial.println("Gagal mengirim suhu ke Firebase");
  Serial.println(firebaseData.errorReason());
}

if (Firebase.setFloat(firebaseData, "/Data/kelembapan", hum)) {
  Serial.println("Kelembapan berhasil dikirim ke Firebase");
} else {
  Serial.println("Gagal mengirim kelembapan ke Firebase");
  Serial.println(firebaseData.errorReason());
}

```

Gambar 49. Program pengiriman data suhu dan kelembapan ke firebase

Program ini mengirimkan data suhu dan kelembapan ke *firebase*. Jika pengiriman suhu ke *firebase* berhasil, program mencetak pesan sukses di serial monitor; jika gagal, program mencetak pesan kesalahan dan alasan kegagalannya.

```

time_t now = time(nullptr);
struct tm *p_tm = localtime(&now);
strftime(timestamp, sizeof(timestamp), "%Y-%m-%d %H:%M:%S", p_tm);
String historyPath = "/History/" + String(timestamp);

if (Firebase.setFloat(firebaseData, historyPath + "/suhu", temp)) {

```

```

    Serial.println("Suhu history berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim suhu history ke Firebase");
    Serial.println(firebaseData.errorReason());
  }

  if (Firebase.setFloat(firebaseData, historyPath + "/kelembapan", hum)) {
    Serial.println("Kelembapan history berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim kelembapan history ke Firebase");
    Serial.println(firebaseData.errorReason());
  }

  if (Firebase.setString(firebaseData, historyPath + "/Mesin", relayStates[0] ? "ON" :
"OFF")) {
    Serial.println("Status Mesin history berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim status Mesin history ke Firebase");
    Serial.println(firebaseData.errorReason());
  }

  if (Firebase.setString(firebaseData, historyPath + "/Heater", relayStates[1] ? "ON" :
"OFF")) {
    Serial.println("Status Heater history berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim status Heater history ke Firebase");
    Serial.println(firebaseData.errorReason());
  }

  if (Firebase.setString(firebaseData, historyPath + "/Kipas", relayStates[2] ? "ON" :
"OFF")) {
    Serial.println("Status Kipas history berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim status Kipas history ke Firebase");
    Serial.println(firebaseData.errorReason());
  }

  if (Firebase.setFloat(firebaseData, historyPath + "/sisaMenit", sisaMenit)) {
    Serial.println("Sisa menit history berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim sisa menit history ke Firebase");
    Serial.println(firebaseData.errorReason());
  }
}

```

```

if (Firebase.setFloat(firebaseData, historyPath + "/sisaDetik", sisaDetik)) {
  Serial.println("Sisa detik history berhasil dikirim ke Firebase");
} else {
  Serial.println("Gagal mengirim sisa detik history ke Firebase");
  Serial.println(firebaseData.errorReason());
}

```

Gambar 50. Program Penyimpanan Data Historis ke Firebase

Pembaruan data suhu, kelembapan, status relay, serta waktu sisa dalam menit dan detik dikirim ke Firebase berdasarkan tanggal dan waktu. Program memeriksa apakah setiap pengiriman data berhasil dan mencetak pesan di serial monitor untuk menunjukkan keberhasilan atau kegagalan

```

void updateRelays() {
  for (int i = 0; i < 3; i++) {
    digitalWrite(relayPins[i], relayStates[i] ? LOW : HIGH);
    updateFirebaseData("/Data/relay" + String(i + 1), relayStates[i]);
    Serial.print("Relay "); Serial.print(i + 1); Serial.print(": ");
    Serial.println(relayStates[i] ? "ON" : "OFF");
  }
}

```

Gambar 51. Program fungsi update relay

Fungsi *updateRelays()* mengatur status tiga *relay* berdasarkan nilai yang tersimpan dalam *array relayStates*. Fungsi ini menggunakan *digitalWrite()* untuk mengubah status pin relay menjadi *LOW* jika relay aktif (*true*) atau *HIGH* jika tidak aktif (*false*). Selain itu, fungsi ini memperbarui status setiap relay ke Firebase dan mencetak status *relay* saat ini ke serial monitor, menunjukkan apakah masing-masing relay dalam keadaan *ON* atau *OFF*.

```

void manualControl() {
  Serial.println("Auto-kontrol OFF");
  updateRelays();
}

```

Gambar 52. Program fungsi manualcontrol

Fungsi *manualControl()* menampilkan pesan "Auto-kontrol OFF" di serial monitor untuk menunjukkan bahwa mode kontrol otomatis dinonaktifkan. Setelah itu, fungsi memanggil *updateRelays()* untuk memperbarui status relay sesuai dengan pengaturan saat ini, yang memungkinkan kontrol manual atas *relay*.

```

void autoControlMode() {
  Serial.println("Auto-kontrol ON");

  if (sisaMenit != 0 || sisaDetik != 0) {

```

```

    relayStates[0] = true; // Mesin
    if (temp < 40) {
        relayStates[1] = true; // Heater
        relayStates[2] = false; // Kipas
    } else {
        relayStates[1] = false; // Heater
        relayStates[2] = true; // Kipas
    }
} else {
    relayStates[0] = relayStates[1] = relayStates[2] = false; // Semua OFF
}

updateRelays();
}

```

Gambar 53. Program fungsi *autocontrolMode*

Fungsi *autoControlMode()* mengaktifkan mode kontrol otomatis dan menampilkan pesan "Auto-kontrol ON" di serial monitor. Jika waktu sisa (menit atau detik) tidak sama dengan nol, fungsi ini mengatur status relay untuk menghidupkan mesin dan mengontrol heater serta kipas berdasarkan suhu. Heater dihidupkan jika suhu di bawah 40°C, dan kipas dihidupkan jika suhu di atas 40°C. Jika waktu sisa mencapai nol, semua relay dimatikan. Fungsi kemudian memanggil *updateRelays()* untuk menerapkan perubahan status relay

```

void loop() {
    fetchData();
    sendDataToFirebase();
    if (autoControl) autoControlMode();
    else manualControl();
    delay(2000);
}

```

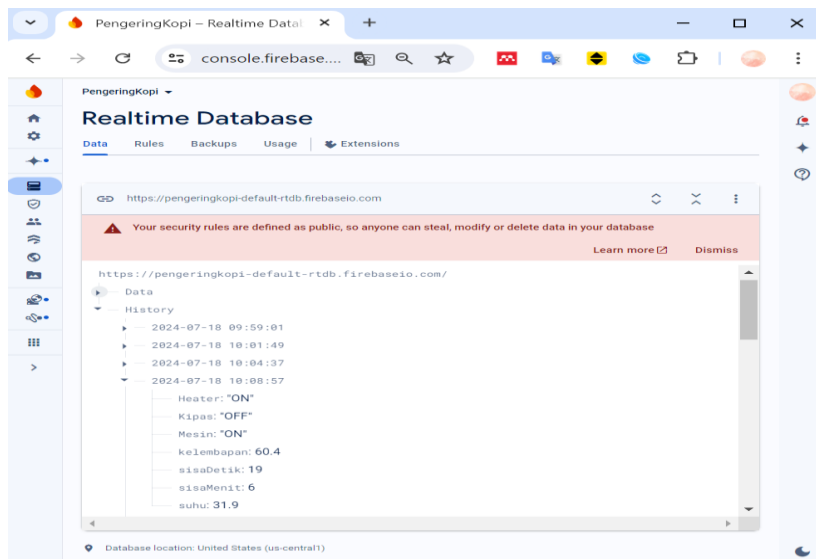
Gambar 54. Program fungsi *loop*

Fungsi *loop()* menjalankan serangkaian tugas secara berulang. Pertama, ia memanggil *fetchData()* untuk mengambil data terbaru dari Firebase. Selanjutnya, fungsi *sendDataToFirebase()* mengirimkan data sensor dan status ke Firebase. Fungsi ini kemudian memeriksa apakah mode kontrol otomatis (*autoControl*) aktif; jika ya, ia menjalankan *autoControlMode()* untuk mengatur relay berdasarkan kondisi saat ini, atau sebaliknya, menjalankan *manualControl()* untuk pengaturan manual. Setelah itu, fungsi *loop()* menunggu selama 2000 milidetik sebelum memulai siklus berikutnya.

3.3.2 Implementasi Perangkat Lunak Sistem

1. Implementasi *Firebase* pada sistem mesin pengering

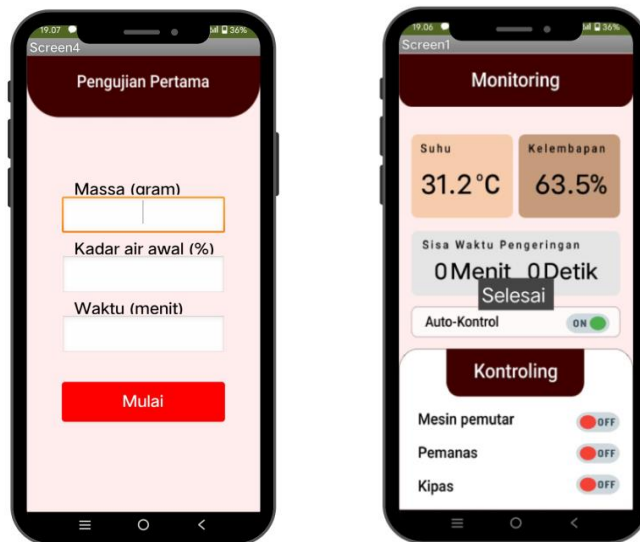
Firebase berfungsi untuk menyimpan data historis suhu, kelembapan, dan status aktuator secara real-time dari sistem pengering kopi. Setiap kali data suhu dan kelembapan baru diperoleh dari sensor, serta status aktuator (mesin, pemanas, dan kipas) diperbarui, nilai-nilai tersebut dikirim ke Firebase untuk disimpan. Selain menyimpan data terbaru, Firebase juga mencatat data historis berdasarkan waktu, memungkinkan pengguna untuk memantau perubahan kondisi lingkungan dan kinerja sistem secara detail.



Gambar 55. Tampilan Firebase

Data historis ini disimpan di *"/History/"* sehingga pengguna dapat melihat kapan dan bagaimana kondisi lingkungan serta status aktuator berubah. Dengan menyimpan informasi ini, pengguna dapat melacak kinerja sistem pengering kopi dari waktu ke waktu, membantu mereka memahami pola dan membuat keputusan yang lebih baik untuk pengelolaan sistem.

2. Implementasi Aplikasi



Gambar 56. Tampilan Aplikasi Mobile

Aplikasi ini digunakan untuk memonitor dan mengontrol proses pengeringan kopi. Pada layar monitoring, aplikasi menampilkan suhu dan kelembapan ruang pengeringan, serta sisa waktu pengeringan dalam format menit dan detik. Pengguna juga dapat mengaktifkan atau menonaktifkan mode kontrol otomatis. Pada layar kontroling, aplikasi menyediakan tombol untuk menghidupkan atau mematikan tiga aktuator utama: mesin pemutar, pemanas, dan kipas, sehingga memungkinkan kontrol manual setiap aktuator sesuai kebutuhan.

Layar pengujian pertama memungkinkan pengguna memasukkan data awal pengujian seperti massa biji kopi, kadar air awal, dan waktu pengeringan yang diinginkan, serta memulai proses pengeringan dengan tombol "Mulai". Aplikasi ini terintegrasi dengan Firebase untuk menampilkan informasi secara real-time dan menyimpan data historis suhu, kelembapan, serta status aktuator untuk analisis lebih lanjut.

3.4 Hasil Kinerja Aplikasi Pantau dan Kendali Mobile Mesin Pengering Biji Kopi Berbasis Internet of Things

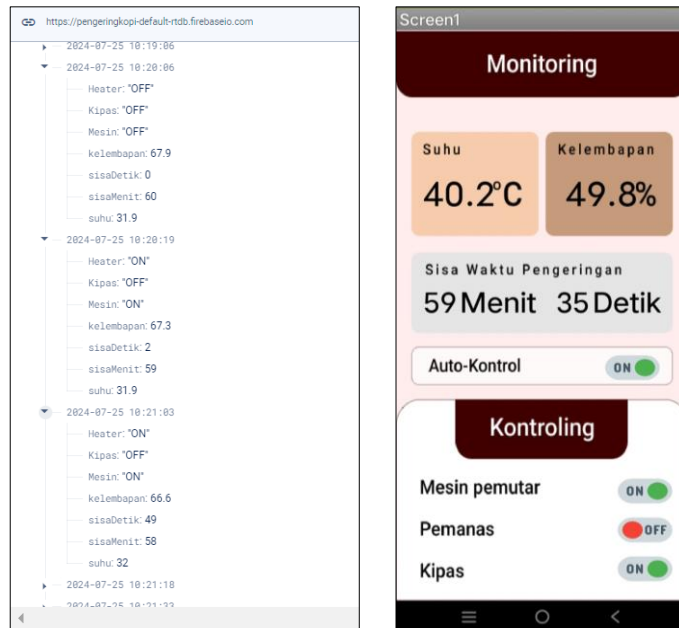
Pengujian sistem ini merupakan proses pengeksekusian sistem perangkat keras dan lunak untuk menentukan apakah sistem tersebut sesuai dengan yang diinginkan peneliti. Pada pengujian sistem terlihat data hasil pengukuran suhu, kelembapan, dan waktu pengeringan. Menampilkan data sensor secara *realtime* selama masih terhubung dengan listrik dan terkoneksi jaringan *WiFi*. Dari hasil pembacaan tersebut kemudian sistem juga menampilkan data hasil pengukuran di aplikasi berbasis android yang diakses melalui *smartphone* Android, terlihat data hasil pengukuran yang ditampilkan adalah suhu, kelembapan, dan sisa waktu

pengeringan. Sebelum memulai pengeringan aplikasi mobile digunakan untuk menginput massa awal, kadar air dan waktu pengeringan.

3.4.1 Hasil Pengujian Aplikasi Pantau dan Kendali Mobile Mesin Pengering Biji Kopi Berbasis *Internet of Things*

Pada Penelitian ini dilakukan tiga kali percobaan pengeringan biji kopi dengan massa 1 kg dan kadar air 22% dengan waktu yang berbeda-beda.

1. Penelitian pertama menggunakan waktu 1 jam



Gambar 57. Tampilan Firebase dan Aplikasi mobile penelitian pertama

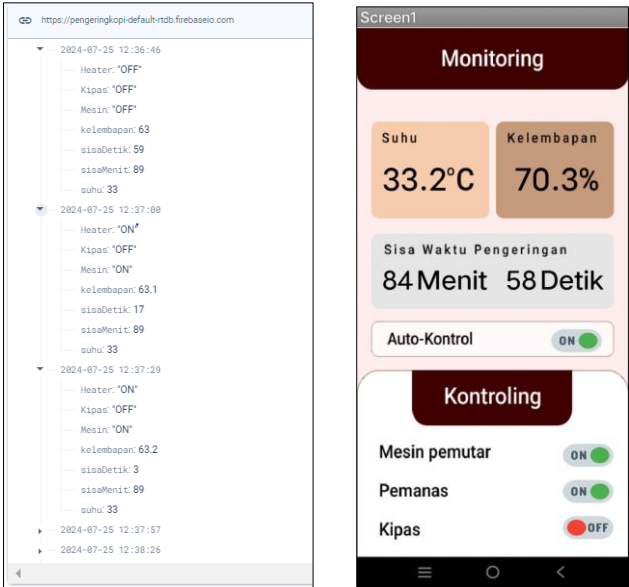
Menurut **Gambar 57** firebase menunjukkan data yang disimpan mencakup status beberapa aktuator yaitu Heater, Kipas, dan Mesin. Firebase juga menyimpan data kelembapan, suhu, serta sisa waktu pengeringan dalam detik dan menit. Aplikasi mobile menampilkan suhu sebesar 40.2°C dan kelembapan 49.8%, serta sisa waktu pengeringan selama 59 menit dan 35 detik. Fitur Auto-Kontrol diaktifkan, memungkinkan sistem untuk beroperasi otomatis. Selain itu, terdapat kontrol manual untuk mesin pemutar, pemanas, dan kipas, dengan status masing-masing: mesin pemutar aktif (ON), pemanas tidak aktif (OFF), dan kipas aktif (ON)

Tabel 6. Penelitian pertama menggunakan waktu 1 jam

timestamp	suhu	kelembapan	Mesin	Heater	Kipas	sisaMenit	sisaDetik
2024-07-25 10:20:06	31.9	67.9	OFF	OFF	OFF	60	0
2024-07-25 10:23:31	37.9	52.5	ON	ON	OFF	55	52
2024-07-25 10:28:28	36.2	55	ON	ON	OFF	50	57
2024-07-25 10:33:52	41	43	ON	OFF	ON	45	35
2024-07-25 10:39:01	42.1	41.8	ON	OFF	ON	40	43
2024-07-25 10:44:58	39	46.7	ON	OFF	ON	34	48
2024-07-25 10:48:40	41.8	42.1	ON	OFF	ON	30	52
2024-07-25 10:53:50	41.2	44.1	ON	OFF	ON	25	45
2024-07-25 10:59:46	42.4	41.1	ON	OFF	ON	20	21
2024-07-25 11:03:58	42.9	41	ON	OFF	ON	15	40
2024-07-25 11:09:23	40.4	45.8	ON	ON	OFF	10	17
2024-07-25 11:15:03	40.3	46.8	ON	ON	OFF	5	25
2024-07-25 11:18:48	38.1	50	ON	ON	OFF	1	11
2024-07-25 11:19:17	37.9	50.6	ON	ON	OFF	0	41
2024-07-25 11:21:29	40.5	46.2	OFF	OFF	OFF	0	0

Tabel 6 menunjukkan data dari sistem pengering biji kopi, dengan proses dimulai pada 2024-07-25 pukul 10:20:06 WITA dan berakhir pada 11:21:29 WITA. Ketika suhu mencapai 40°C, Heater otomatis dimatikan (OFF) dan Kipas diaktifkan (ON) untuk mencegah suhu berlebihan. Sebaliknya jika suhu kurang dari 40°C maka Heater diaktifkan (ON) dan kipas dimatikan (OFF). Data menunjukkan rentang suhu berkisar antara 31.9°C hingga 42.9°C. Suhu terendah tercatat pada 10:20:06 WITA yaitu 31.9°C, dan tertinggi pada 11:03:58 WITA yaitu 42.9°C. Dengan demikian, sistem ini memastikan bahwa suhu dalam mesin pengering dikontrol secara optimal untuk menghasilkan pengeringan biji kopi yang efisien.

2. Penelitian kedua menggunakan waktu 1 jam 30 menit



Gambar 58. Tampilan Firebase dan Aplikasi mobile penelitian kedua

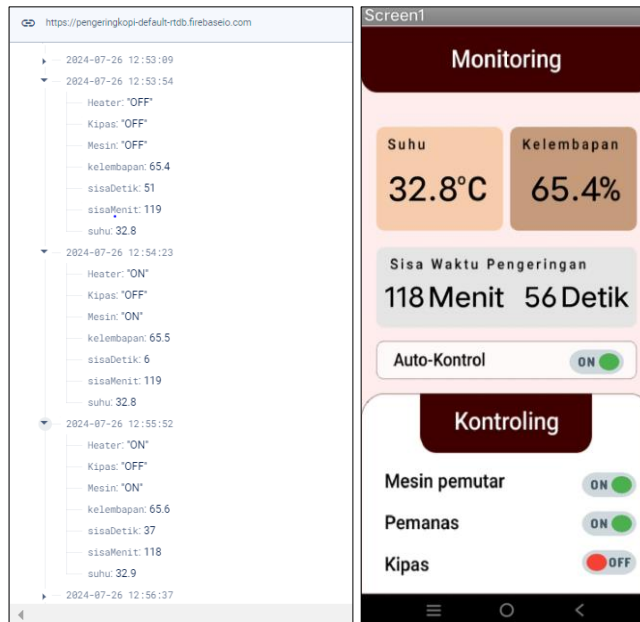
Menurut **Gambar 58** firebase menunjukkan data yang disimpan mencakup status beberapa aktuator yaitu Heater, Kipas, dan Mesin. Firebase juga menyimpan data kelembapan, suhu, serta sisa waktu pengeringan dalam detik dan menit. Aplikasi mobile menampilkan suhu sebesar 33.2°C dan kelembapan 70.3%, serta sisa waktu pengeringan selama 84 menit dan 58 detik. Fitur Auto-Kontrol diaktifkan, memungkinkan sistem untuk beroperasi otomatis. Selain itu, terdapat kontrol manual untuk mesin pemutar, pemanas, dan kipas, dengan status masing-masing: mesin pemutar aktif (ON), pemanas aktif (ON), dan kipas tidak aktif (OFF)

Tabel 7. Penelitian kedua menggunakan waktu 1 jam 30 menit

timestamp	suhu	kelembapan	Mesin	Heater	Kipas	sisaMenit	sisaDetik
2024-07-25 12:36:02	33	63.3	OFF	OFF	OFF	90	0
2024-07-25 12:42:06	43.8	41.3	ON	ON	OFF	84	27
2024-07-25 12:46:00	38.7	49.2	ON	ON	OFF	80	35
2024-07-25 12:50:51	41.5	43.2	ON	OFF	ON	75	45
2024-07-25 12:55:41	40.3	47.6	ON	OFF	ON	70	42
2024-07-25 13:00:53	38.7	48.9	ON	ON	OFF	65	32
2024-07-25 13:06:26	39.3	48.7	ON	ON	OFF	60	31
2024-07-25 13:11:04	39.2	47	ON	ON	OFF	55	25
2024-07-25 13:16:42	38.6	49.8	ON	OFF	ON	50	19
2024-07-25 13:20:53	41.2	42.8	ON	OFF	ON	45	40
2024-07-25 13:26:16	43.7	41.9	ON	OFF	ON	40	34
2024-07-25 13:31:22	40.3	48.1	ON	ON	OFF	35	31
2024-07-25 13:35:53	38.4	51.9	ON	ON	OFF	30	46
2024-07-25 13:41:13	39	51.2	ON	ON	OFF	25	28
2024-07-25 13:47:05	42.7	45.1	ON	ON	OFF	20	7
2024-07-25 13:51:03	38.7	51.5	ON	OFF	ON	15	41
2024-07-25 13:56:49	39.4	50.8	ON	ON	OFF	10	13
2024-07-25 14:01:17	38.9	52.1	ON	ON	OFF	5	46
2024-07-25 14:05:39	41.3	47.6	ON	OFF	ON	1	11
2024-07-25 14:07:53	40.8	48.5	ON	OFF	ON	0	56
2024-07-25 14:08:22	38.2	54.9	ON	OFF	ON	0	0
2024-07-25 14:09:07	37.8	55.8	OFF	OFF	OFF	0	0

Tabel 7 menunjukkan data dari sistem pengering biji kopi, dengan proses dimulai pada 2024-07-25 pukul 12:36:02 WITA dan berakhir pada 14:09:07 WITA. Ketika suhu mencapai 40°C, Heater otomatis dimatikan(OFF) dan Kipas diaktifkan (ON) untuk mencegah suhu berlebihan. Sebaliknya jika suhu kurang dari 40°C maka Heater diaktifkan (ON) dan kipas dimatikan (OFF). Data menunjukkan rentang suhu berkisar antara 33°C hingga 43.8°C. Suhu terendah tercatat pada 12:36:02 WITA yaitu 33°C, dan tertinggi pada 12:42:06 WITA yaitu 43.8°C. Dengan demikian, sistem ini memastikan bahwa suhu dalam mesin pengering dikontrol secara optimal untuk menghasilkan pengeringan biji kopi yang efisien.

3. Penelitian ketiga menggunakan waktu 2 jam



Gambar 59. Tampilan Firebase dan Aplikasi mobile penelitian ketiga

Menurut **Gambar 59** firebase menunjukkan data yang disimpan mencakup status beberapa aktuator yaitu Heater, Kipas, dan Mesin. Firebase juga menyimpan data kelembapan, suhu, serta sisa waktu pengeringan dalam detik dan menit. Aplikasi mobile menampilkan suhu sebesar 32.8°C dan kelembapan 65.4%, serta sisa waktu pengeringan selama 118 menit dan 56 detik. Fitur Auto-Kontrol diaktifkan, memungkinkan sistem untuk beroperasi otomatis. Selain itu, terdapat kontrol manual untuk mesin pemutar, pemanas, dan kipas, dengan status masing-masing: mesin pemutar aktif (ON), pemanas aktif (ON), dan kipas tidak aktif (OFF).

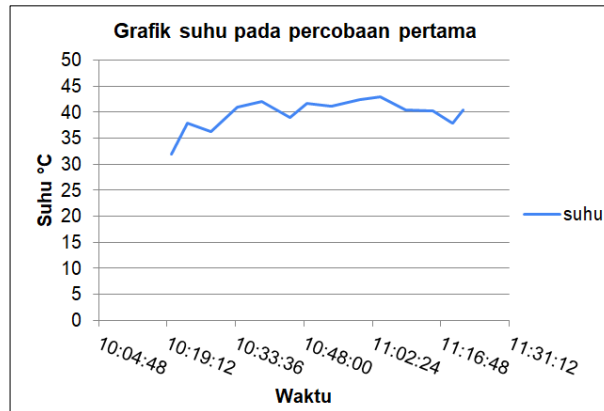
Tabel 8. Penelitian ketiga menggunakan waktu 2 jam

timestamp	suhu	kelembapan	Mesin	Heater	Kipas	sisaMenit	sisaDetik
2024-07-26 12:53:09	32.8	65	OFF	OFF	OFF	120	0
2024-07-26 12:58:25	40.8	48.9	ON	ON	OFF	115	51
2024-07-26 13:03:39	38.4	51.8	ON	ON	OFF	110	24
2024-07-26 13:09:35	38.1	52.3	ON	ON	OFF	105	29
2024-07-26 13:14:19	41	46.9	ON	OFF	ON	100	48
2024-07-26 13:17:47	43.5	44	ON	ON	OFF	95	51
2024-07-26 13:22:46	41.9	46.1	ON	OFF	ON	90	39
2024-07-26 13:29:15	42.1	46.8	ON	ON	OFF	84	43
2024-07-26 13:33:16	40	49.2	ON	OFF	ON	80	58
2024-07-26 13:38:41	44.9	41	ON	OFF	ON	75	36
2024-07-26 13:43:37	40.4	48.6	ON	OFF	ON	69	57
2024-07-26 13:48:43	41.6	46.8	ON	OFF	ON	65	37
2024-07-26 13:54:53	38.5	53	ON	ON	OFF	59	59
2024-07-26 13:57:46	41.1	47.6	ON	OFF	ON	55	53
2024-07-26 14:02:26	41.8	46.4	ON	OFF	ON	51	15
2024-07-26 14:07:45	41.4	48.4	ON	ON	OFF	45	57
2024-07-26 14:12:59	42.1	47.1	ON	OFF	ON	40	47
2024-07-26 14:18:14	42.8	46.4	ON	OFF	ON	35	48
2024-07-26 14:24:23	40.7	50.6	ON	ON	OFF	29	56
2024-07-26 14:28:49	38.9	53.6	ON	ON	OFF	25	31
2024-07-26 14:33:58	41.8	47.8	ON	OFF	ON	20	9
2024-07-26 14:38:39	39.3	52.3	ON	ON	OFF	15	15
2024-07-26 14:43:33	39.1	53.4	ON	OFF	ON	10	23
2024-07-26 14:48:56	43.1	45.6	ON	OFF	ON	5	2
2024-07-26 14:52:54	40.3	50.6	ON	OFF	ON	1	8
2024-07-26 14:53:52	39.8	51.4	ON	OFF	ON	0	23
2024-07-26 14:54:19	39.5	52	OFF	OFF	OFF	0	0

Tabel 8 menunjukkan data dari sistem pengering biji kopi, dengan proses dimulai pada 2024-07-26 pukul 12:53:09 WITA dan berakhir pada 14:54:19 WITA. Ketika suhu mencapai 40°C, Heater otomatis dimatikan(OFF) dan Kipas diaktifkan (ON) untuk mencegah suhu berlebihan. Sebaliknya jika suhu kurang dari 40°C maka Heater diaktifkan (ON) dan kipas dimatikan (OFF). Data menunjukkan rentang suhu berkisar antara 32.8°C hingga 44.9°C. Suhu terendah tercatat pada 12:53:09 WITA yaitu 32.8°C, dan tertinggi pada 13:38:41 WITA yaitu 44.9°C. Dengan demikian, sistem ini memastikan bahwa suhu dalam mesin pengering dikontrol secara optimal untuk menghasilkan pengeringan biji kopi yang efisien.

3.4.2 Evaluasi Aplikasi Pantau dan Kendali Mobile Mesin Pengering Biji Kopi Berbasis *Internet of Things*

1. Penelitian pertama menggunakan waktu 1 jam



Gambar 60. Grafik suhu pada percobaan pertama

Gambar 60 menunjukkan grafik perubahan suhu selama percobaan pertama. Pada awalnya, suhu meningkat dari sekitar 30°C ke 40°C, lalu berfluktuasi sedikit antara 35°C dan mendekati 45°C. Suhu cenderung stabil setelah kenaikan awal, dengan puncak mendekati 45°C selanjutnya menurun sedikit dan stabil di sekitar 40°C. Perubahan ini terjadi selama periode waktu sekitar 1 jam.



Gambar 61. Pembacaan massa dan kadar air pertama

Gambar 61 menunjukkan pada percobaan pertama terjadi pengurangan massa dari 1000 gram menjadi 973 gram dan kadar air dari 22% menjadi 19.2%

2. Penelitian kedua menggunakan waktu 1 jam 30 menit



Gambar 62. Grafik suhu pada percobaan kedua

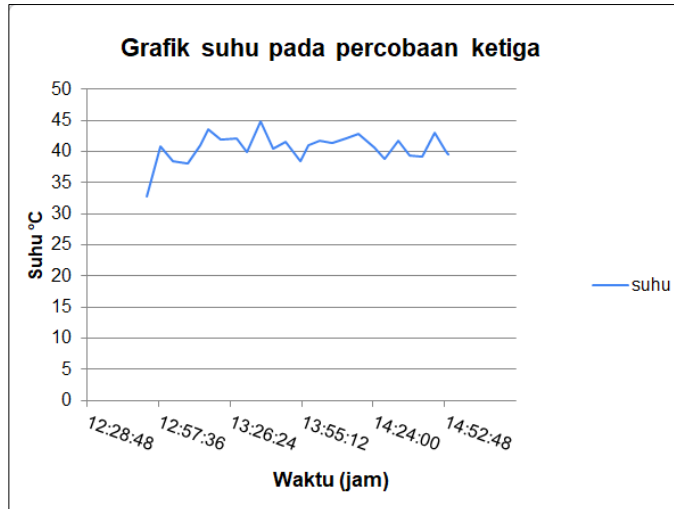
Gambar 62 menunjukkan Grafik perubahan suhu selama percobaan kedua. Pada awalnya, suhu meningkat dari sekitar 30°C ke 40°C, kemudian berfluktuasi sedikit antara 35°C dan mendekati 45°C. Suhu cenderung stabil setelah kenaikan awal, dengan puncak mendekati 45°C dan fluktuasi kecil yang tetap berada dalam rentang tersebut. Perubahan ini terjadi selama periode waktu sekitar 1,5 jam.



Gambar 63. Pembacaan massa dan kadar air kedua

Gambar 63 menunjukkan pada percobaan kedua terjadi pengurangan massa dari 1000 gram menjadi 946 gram dan kadar air dari 22% menjadi 15.4%

3. Penelitian ketiga menggunakan waktu 2 jam



Gambar 64. Grafik suhu pada percobaan ketiga

Gambar 64 menunjukkan grafik perubahan suhu selama percobaan ketiga. Pada awalnya, suhu meningkat dari sekitar 30°C ke 40°C, kemudian berfluktuasi antara 35°C dan mendekati 45°C. Suhu cenderung stabil setelah kenaikan awal, dengan beberapa puncak mendekati 45°C dan sedikit penurunan di akhir percobaan. Perubahan ini terjadi selama periode waktu sekitar 2 jam.



Gambar 65. Pembacaan massa dan kadar air ketiga

Gambar 65 menunjukkan pada percobaan ketiga terjadi pengurangan massa dari 1000 gram menjadi 885 gram dan kadar air dari 22% menjadi 12.4%

3.5 Pembahasan Kinerja Aplikasi Pantau dan Kendali *Mobile* Mesin Pengereng Biji Kopi Berbasis *Internet of Things*

Kinerja aplikasi mobile dalam memantau dan mengendalikan mesin pengereng biji kopi berbasis IoT dinilai baik dalam beberapa aspek penting. Aplikasi ini memberikan kemudahan dalam memantau parameter kritis seperti suhu, kelembapan, dan status aktuator secara real-time. Selain itu, fitur Auto-Kontrol yang diimplementasikan dalam aplikasi memungkinkan pengaturan otomatis perangkat keras seperti Heater, Kipas, dan Mesin Pemutar berdasarkan data suhu yang diperoleh.

Dari segi antarmuka pengguna (UI), aplikasi ini dirancang dengan fokus pada kemudahan penggunaan. Tampilan yang sederhana memudahkan pengguna untuk mengakses informasi yang diperlukan serta melakukan pengaturan sesuai kebutuhan. Kontrol manual juga tersedia, memberikan fleksibilitas tambahan untuk mengatur operasi mesin pengereng secara manual jika diperlukan. Pengguna dapat dengan mudah melihat status mesin pemutar, pemanas, dan kipas melalui aplikasi, serta mengaktifkan atau menonaktifkan perangkat tersebut dengan cepat.

Kinerja jaringan antara aplikasi dan server Firebase juga menunjukkan stabilitas yang baik. Data dapat diakses secara real-time tanpa adanya keterlambatan signifikan sehingga memastikan bahwa kondisi pengeringan dapat dipantau dan dikendalikan dengan akurat. Selain itu, data historis yang disimpan dalam Firebase memberikan pengguna akses untuk menganalisis hasil pengeringan, termasuk perubahan massa dan kadar air biji kopi.

Sistem pengendalian yang diterapkan dalam aplikasi, baik secara otomatis maupun manual, menunjukkan kemampuan untuk mempertahankan suhu yang diperlukan dalam rentang optimal. Sebagai contoh, sistem secara otomatis mematikan pemanas saat suhu mencapai 40°C dan mengaktifkan kipas untuk mencegah suhu berlebihan. Sebaliknya, saat suhu turun di bawah 40°C, pemanas diaktifkan kembali. Fungsi ini memastikan bahwa proses pengeringan biji kopi berlangsung dengan efisien tanpa risiko *over-heat*.

Dari hasil ketiga percobaan pengeringan biji kopi menggunakan sistem berbasis Internet of Things (IoT), terlihat adanya penurunan massa dan kadar air secara bertahap sesuai dengan durasi pengeringan. Pada percobaan pertama yang berlangsung selama 1 jam, massa biji kopi berkurang dari 1000 gram menjadi 973 gram, dengan kadar air turun dari 22% menjadi 19.2%. Pada percobaan kedua dengan durasi 1 jam 30 menit, massa biji kopi berkurang lebih lanjut menjadi 946 gram, dengan kadar air menurun lebih signifikan menjadi 15.4%. Percobaan ketiga, yang berlangsung selama 2 jam, menunjukkan pengurangan massa hingga 899 gram, dan kadar air biji kopi mencapai titik terendah pada 12.4%.

Penurunan massa dan kadar air yang semakin signifikan pada percobaan kedua dan ketiga menunjukkan bahwa sistem pengeringan bekerja dengan lebih efektif seiring dengan bertambahnya durasi pengeringan. Laju pengeringan kadar air meningkat dari 2.8% per jam pada percobaan pertama, menjadi 4.4% per jam pada percobaan kedua, dan mencapai 4.8% per jam pada percobaan ketiga.

BAB IV KESIMPULAN DAN SARAN

4.1 Kesimpulan

Dari ketiga percobaan pengeringan biji kopi dengan massa awal 1 kg dan kadar air awal 22%, terlihat adanya pengurangan massa yang signifikan seiring dengan bertambahnya durasi pengeringan. Laju pengeringan dihitung menggunakan rumus:

$$Q_a = \frac{W_{kb} - W_f}{N}$$

Dimana:

Q_a = laju pengeringan (gram/jam)

W_{kb} = massa biji kopi awal (gram)

W_f = massa biji kopi akhir (gram)

N = durasi pengeringan (jam)

1. Pada percobaan pertama, yang berlangsung selama 1 jam, massa biji kopi berkurang dari 1000 gram menjadi 973 gram, dengan penurunan kadar air dari 22% menjadi 19.2%. Ini menghasilkan laju pengeringan sebesar 27 gram/jam.
2. Percobaan kedua, yang berlangsung selama 1 jam 30 menit, menunjukkan penurunan massa dari 1000 gram menjadi 946 gram, dengan kadar air turun menjadi 15.4%, menghasilkan laju pengeringan 36 gram/jam.
3. Pada percobaan ketiga, dengan durasi pengeringan 2 jam, massa biji kopi berkurang menjadi 885 gram, dengan kadar air mencapai 12.4%, memberikan laju pengeringan sebesar 57.5 gram/jam.

Laju pengeringan yang meningkat pada percobaan kedua dan ketiga menunjukkan bahwa proses pengeringan semakin efisien dengan bertambahnya waktu pengeringan. Grafik suhu yang menunjukkan stabilisasi sekitar 40°C mendukung pengeringan yang konsisten, di mana suhu cenderung stabil setelah peningkatan awal. Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Mesin pengering biji kopi akan dirancang dengan teknologi IoT yang menggunakan sensor dan mikrocontroller untuk mengumpulkan data suhu dan kelembaban selama proses pengeringan. Data ini akan dikirimkan ke Firebase, sebuah platform database cloud, sehingga bisa diakses oleh aplikasi mobile yang dibangun dengan MIT App Inventor.

2. Aplikasi mobile akan dibangun menggunakan MIT App Inventor, sebuah platform visual yang mudah digunakan untuk membuat aplikasi Android. Aplikasi ini akan mengambil data dari Firebase, menampilkannya kepada pengguna, dan memungkinkan pengguna untuk mengontrol mesin pengering biji kopi secara real-time. Firebase akan berfungsi sebagai jembatan antara mesin dan aplikasi, menyimpan data sensor dan mengirim perintah dari aplikasi ke mesin.
3. Pengujian kinerja akan mencakup memastikan bahwa aplikasi yang dibuat dengan MIT App Inventor dapat secara stabil terhubung ke Firebase untuk menerima dan mengirim data. Aplikasi akan diuji untuk melihat apakah data sensor ditampilkan dengan benar, apakah perintah dari aplikasi dijalankan dengan cepat, dan apakah aplikasi tetap responsif dan mudah digunakan di berbagai kondisi operasional.

4.2 Saran

Rancang bangun aplikasi pantau dan kendali *mobile* mesin pengering biji kopi berbasis *internet of things* masih jauh dari kesempurnaan. Untuk menciptakan sebuah sistem yang baik tentu perlu dilakukan pengembangan, baik dari sisi manfaat maupun dari sisi kerja sistem. Berikut saran untuk pengembangan yang mungkin dapat menambah nilai dari aplikasi itu sendiri :

1. Penelitian ini menunjukkan bahwa laju pengeringan meningkat dengan waktu, namun optimisasi lebih lanjut diperlukan untuk menentukan durasi dan suhu yang ideal guna mencapai kualitas biji kopi yang optimal. Pengujian tambahan dengan variasi suhu dan durasi yang lebih spesifik bisa memberikan wawasan lebih baik mengenai titik optimal pengeringan.
2. Disarankan untuk menggunakan sensor suhu dan kelembapan dengan presisi dan akurasi yang lebih tinggi untuk memastikan pengukuran yang lebih tepat. Sensor SHT31 atau AM2302 bisa menjadi alternatif yang lebih baik untuk mendapatkan data yang lebih akurat.
3. Aplikasi mobile yang digunakan untuk memantau dan mengendalikan mesin pengering sudah berjalan dengan baik. Namun, untuk pengembangan lebih lanjut, seperti notifikasi otomatis ketika kondisi tertentu tercapai (misalnya suhu terlalu tinggi atau kadar air mencapai tingkat tertentu), dapat meningkatkan respons pengguna dan mencegah potensi kerusakan pada biji kopi.
4. Implementasi algoritma cerdas seperti *machine learning* untuk mengoptimalkan pengaturan suhu dan waktu pengeringan berdasarkan data historis bisa menjadi langkah selanjutnya. Algoritma ini bisa mempelajari pola-pola dari pengeringan sebelumnya dan secara otomatis menyesuaikan pengaturan untuk hasil yang lebih konsisten dan efisien.

DAFTAR PUSTAKA

- Adani, F., & Salsabil, S. (2019). Internet of Things: Sejarah Teknologi Dan Penerapannya. *Isu Teknologi Stt Mandala*, 14(2), 92–99.
- Badan Pusat Statistik. (2021). *Statistik Kopi Indonesia 2020* (Direktorat Statistik Tanaman Pangan Hortikultura dan Perkebunan (ed.)). <https://www.bps.go.id/>
- Badan Standardisasi Nasional. (2008). SNI 01-2907-2008: Biji Kopi. *Badan Standardisasi Nasional*, 1–16.
- Dwirossi, A. A. (2017). *Rancang Bangun Sistem Monitoring Kadar Air Biji Kopi Pada Mesin Pengering Biji Kopi Berbasis Penjejak Matahari Aktif Dengan Mikrokontroler Atmega16*. Institut Teknologi Sepuluh November.
- Food and Agriculture Organization (FAO). 2024. Grafik peringkat negara penghasil utama kopi berdasarkan luas lahan. <https://www.fao.org/faostat/en/>. Diakses pada 7 Maret 2022.
- Gultom, S. S. T. (2019). *Rancang bangun dan pengujian alat pengering biji kopi tenaga listrik dengan pemanfaatan energi surya*. Universitas Sumatera Utara.
- ICO. (2021). World Coffee Production 2017-2020. *International Coffee Organization*, 1–3. <https://www.ico.org/prices/po-production.pdf>
- Indriani, A., Witanto, Y., & Hendra, H. (2019). Pembuatan Alat Pengering Berputar (Rotary) Kopi Dan Lada Hitam Menggunakan Mikrokontroler Arduino Uno Desa Air Raman Kabupaten Kepahiang Propinsi Bengkulu. *Dharma Rafflesia : Jurnal Ilmiah Pengembangan dan Penerapan IPTEKS*, 17(1), 64–76. <https://doi.org/10.33369/dr.v17i1.6197>
- International Coffee Organization. (2021). *World Coffee Consumption*. <https://www.ico.org>
- Karolina Tarigan, E. (2020). *Alat Pengering Kopi Dengan Monitor Pengendalian Jarak Jauh Menggunakan Iot (Internet Of Things) Berbasis ATMEGA328*. Universitas Sumatera Utara.
- Kusmiyati, K., Pambudi, A. D., Arifin, Z., Wulandari, S. A., Purnomo, M. A., Setiadi, K. A., & Listianingrum, N. Y. (2023). Monitoring Sistem Kontrol Mesin Drying Kopi Secara Real Time Berbasis IoT. *Elektrika*, 15(2), 90. <https://doi.org/10.26623/elektrika.v15i2.7857>
- Kusnandar, V. B. (2022). Ini Kontribusi Sektor Pertanian terhadap Ekonomi RI Tahun 2021. *Databoks.Katadata.Co.Id*, November, 2021. <https://databoks.katadata.co.id/datapublish/2022/02/15/ini-kontribusi-sektor-pertanian-terhadap-ekonomi-ri-tahun-2021#:~:text=Berdasarkan data Badan Pusat Statistik,%2C28%25 terhadap PDB nasional.>
- Lampropoulos, G., Siakas, K., & Anastasiadis, T. (2019). Internet of Things in the Context of Industry 4.0: An Overview. *International Journal of Entrepreneurial*

Knowledge, 7(1), 4–19. <https://doi.org/10.2478/ijek-2019-0001>

- Launda, A. P., Mamahit, D. J., & Allo, E. K. (2017). Prototipe sistem pengering biji pala berbasis mikrokontroler Arduino Uno. *Jurnal Teknik Elektro dan Komputer*, 6(3), 141–147.
- Mahendra, R. A. (2019). *Sistem Pengumuman Berbasis Aplikasi Android Dengan Menggunakan Firebase (Studi Kasus: Fakultas Teknik - Universitas Muhammadiyah Magelang)*. Universitas Muhammadiyah Magelang.
- Nurbaeti, A., Kusumawardani, M., & Darmono, H. (2021). Rancang Bangun Alat Pengering Biji Kopi Berbasis Internet Of Things. *Jurnal Jartel: Jurnal Jaringan Telekomunikasi*, 11(2), 74–80. <https://doi.org/10.33795/jartel.v11i2.60>
- Suhaeb, S., Abd Djawad, Y., Jaya, H., Ridwansyah, Sabran, & Risal, A. (2017). Mikrokontroler dan Interface. *Buku Ajar Jurusan Pendidikan Teknik Elektronika UNM*, 2–3. https://scholar.google.co.id/scholar?hl=id&as_sdt=0%2C5&q=jurnal+artikel+ilmiah&btnG=
- Widiastuti, I., & Mirnawati, M. (2020). Development Of Science and Technology to Realize The Industrial Revolution 4.0 and The Internet Of Things (IoT) in Indonesia. *International Journal Of Innovations in Engineering Research and Technology*, 7(6), 49–60.

LAMPIRAN

Lampiran 1 Source Code Program

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <FirebaseESP8266.h>
#include <time.h>

// Konfigurasi Wifi
const char* ssid = "E5576_EAA5";
const char* password = "TqETY27ieLH";
#define API_KEY "AlzaSyC1y9v4LWdOs6uCXxTxS-aXglb2xB7BkVA"
#define DATABASE_URL "https://pengeringkopi-default-
rtdb.firebaseio.com/"
#define USER_EMAIL "nyssaromana01@gmail.com"
#define USER_PASSWORD "06062011"

// Firebase dan DHT
FirebaseData firebaseData;
FirebaseAuth auth;
FirebaseConfig config;
#define DHTPIN D7
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Konfigurasi Relay
const int relayPins[] = {0, 2, 15};
bool relayStates[] = {false, false, false}; // {dinamo, heater, kipas}
float hum, temp;
int sisaMenit, sisaDetik;
char timestamp[30];
bool autoControl = true;

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
```



```

dht.begin();
Wire.begin(D2, D1);
lcd.begin();
lcd.backlight();

for (int i = 0; i < 3; i++) {
    pinMode(relayPins[i], OUTPUT);
}

config.api_key = API_KEY;
auth.user.email = USER_EMAIL;
auth.user.password = USER_PASSWORD;
config.database_url = DATABASE_URL;
Firebase.begin(&config, &auth);

configTime(8 * 3600, 0, "asia.pool.ntp.org", "time.nist.gov");
while (!time(nullptr)) {
    delay(1000);
    Serial.print(".");
}
Serial.println("NTP time synchronized");
}

void fetchData() {
    if (Firebase.getString(firebaseData, "/Data/waktu")) sisaMenit =
firebaseData.stringData().toInt();
    if (Firebase.getString(firebaseData, "/Data/detik")) sisaDetik =
firebaseData.stringData().toInt();
    if (Firebase.getBool(firebaseData, "/Data/Mesin")) relayStates[0] =
firebaseData.boolData();
    if (Firebase.getBool(firebaseData, "/Data/Heater")) relayStates[1] =
firebaseData.boolData();
    if (Firebase.getBool(firebaseData, "/Data/Kipas")) relayStates[2] =
firebaseData.boolData();
    if (Firebase.getBool(firebaseData, "/Data/auto")) autoControl =
firebaseData.boolData();

    Serial.println("Data berhasil diambil dari Firebase:");
    Serial.print("Sisa Menit: "); Serial.println(sisaMenit);
    Serial.print("Sisa Detik: "); Serial.println(sisaDetik);
    Serial.print("Mesin (Relay 1): "); Serial.println(relayStates[0] ? "ON" :
"OFF");
    Serial.print("Heater (Relay 2): "); Serial.println(relayStates[1] ? "ON" :
"OFF");
    Serial.print("Kipas (Relay 3): "); Serial.println(relayStates[2] ? "ON" :

```

```

"OFF");
  Serial.print("Auto Control: "); Serial.println(autoControl ? "ON" : "OFF");
}

void updateFirebaseData(const String &path, bool value) {
  if (Firebase.setBool(firebaseData, path, value)) {
    Serial.print("Status "); Serial.print(path); Serial.println(" berhasil dikirim ke
    Firebase");
  } else {
    Serial.print("Gagal mengirim status "); Serial.print(path); Serial.println(" ke
    Firebase");
    Serial.println(firebaseData.errorReason());
  }
}

void sendDataToFirebase() {
  hum = dht.readHumidity();
  temp = dht.readTemperature();

// Print hasil ke serial monitor
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" *C");

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.printf("Temp: %.2f C", temp);
  lcd.setCursor(0, 1);
  lcd.printf("Humid: %.2f %%", hum);
  delay(2500);

  if (Firebase.setFloat(firebaseData, "/Data/suhu", temp)) {
    Serial.println("Suhu berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim suhu ke Firebase");
    Serial.println(firebaseData.errorReason());
  }

  if (Firebase.setFloat(firebaseData, "/Data/kelembapan", hum)) {
    Serial.println("Kelembapan berhasil dikirim ke Firebase");
  } else {
    Serial.println("Gagal mengirim kelembapan ke Firebase");
  }
}

```

```

    Serial.println(firebaseData.errorReason());
}

time_t now = time(nullptr);
struct tm *p_tm = localtime(&now);
strftime(timestamp, sizeof(timestamp), "%Y-%m-%d %H:%M:%S", p_tm);
String historyPath = "/History/" + String(timestamp);

if (Firebase.setFloat(firebaseData, historyPath + "/suhu", temp)) {
    Serial.println("Suhu history berhasil dikirim ke Firebase");
} else {
    Serial.println("Gagal mengirim suhu history ke Firebase");
    Serial.println(firebaseData.errorReason());
}

if (Firebase.setFloat(firebaseData, historyPath + "/kelembapan", hum)) {
    Serial.println("Kelembapan history berhasil dikirim ke Firebase");
} else {
    Serial.println("Gagal mengirim kelembapan history ke Firebase");
    Serial.println(firebaseData.errorReason());
}

if (Firebase.setString(firebaseData, historyPath + "/Mesin", relayStates[0] ?
"ON" : "OFF")) {
    Serial.println("Status Mesin history berhasil dikirim ke Firebase");
} else {
    Serial.println("Gagal mengirim status Mesin history ke Firebase");
    Serial.println(firebaseData.errorReason());
}

if (Firebase.setString(firebaseData, historyPath + "/Heater", relayStates[1]
? "ON" : "OFF")) {
    Serial.println("Status Heater history berhasil dikirim ke Firebase");
} else {
    Serial.println("Gagal mengirim status Heater history ke Firebase");
    Serial.println(firebaseData.errorReason());
}

if (Firebase.setString(firebaseData, historyPath + "/Kipas", relayStates[2] ?
"ON" : "OFF")) {
    Serial.println("Status Kipas history berhasil dikirim ke Firebase");
} else {
    Serial.println("Gagal mengirim status Kipas history ke Firebase");
    Serial.println(firebaseData.errorReason());
}
}

```

```

if (Firebase.setFloat(firebaseData, historyPath + "/sisamenit", sisaMenit)) {
    Serial.println("Sisa menit history berhasil dikirim ke Firebase");
} else {
    Serial.println("Gagal mengirim sisa menit history ke Firebase");
    Serial.println(firebaseData.errorReason());
}

if (Firebase.setFloat(firebaseData, historyPath + "/sisaDetik", sisaDetik)) {
    Serial.println("Sisa detik history berhasil dikirim ke Firebase");
} else {
    Serial.println("Gagal mengirim sisa detik history ke Firebase");
    Serial.println(firebaseData.errorReason());
}
}

void updateRelays() {
    for (int i = 0; i < 3; i++) {
        digitalWrite(relayPins[i], relayStates[i] ? LOW : HIGH);
        updateFirebaseData("/Data/relay" + String(i + 1), relayStates[i]);
        Serial.print("Relay "); Serial.print(i + 1); Serial.print(": ");
        Serial.println(relayStates[i] ? "ON" : "OFF");
    }
}

void manualControl() {
    Serial.println("Auto-kontrol OFF");
    updateRelays();
}

void autoControlMode() {
    Serial.println("Auto-kontrol ON");

    if (sisaMenit != 0 || sisaDetik != 0) {
        relayStates[0] = true; // Mesin
        if (temp < 40) {
            relayStates[1] = true; // Heater
            relayStates[2] = false; // Kipas
        } else {
            relayStates[1] = false; // Heater
            relayStates[2] = true; // Kipas
        }
    } else {
        relayStates[0] = relayStates[1] = relayStates[2] = false; // Semua OFF
    }
}

```

```
    updateRelays();  
}  
  
void loop() {  
    fetchData();  
    sendDataToFirebase();  
    if (autoControl) autoControlMode();  
    else manualControl();  
    delay(2000);  
}
```

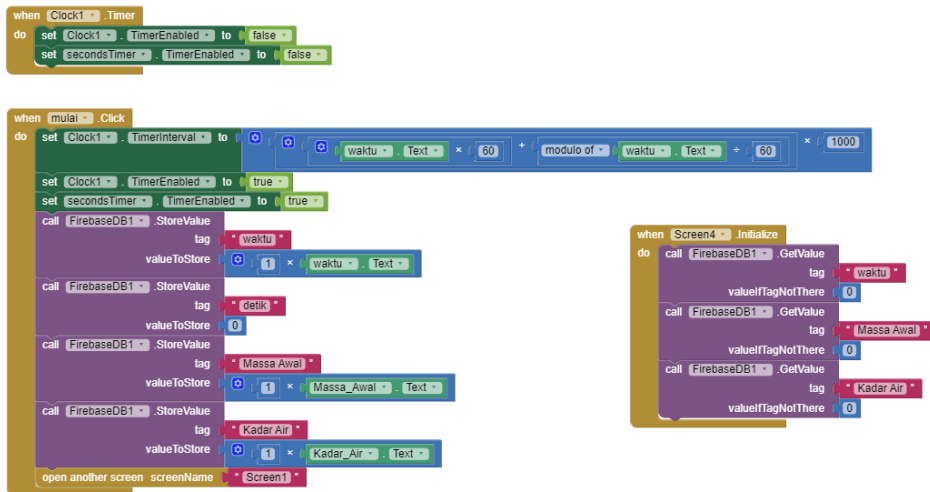
Lampiran 2 Gambar Mesin Pengering Berbasis IoT



Lampiran 3 Gambar Aplikasi Pantau dan Kendali Mobile



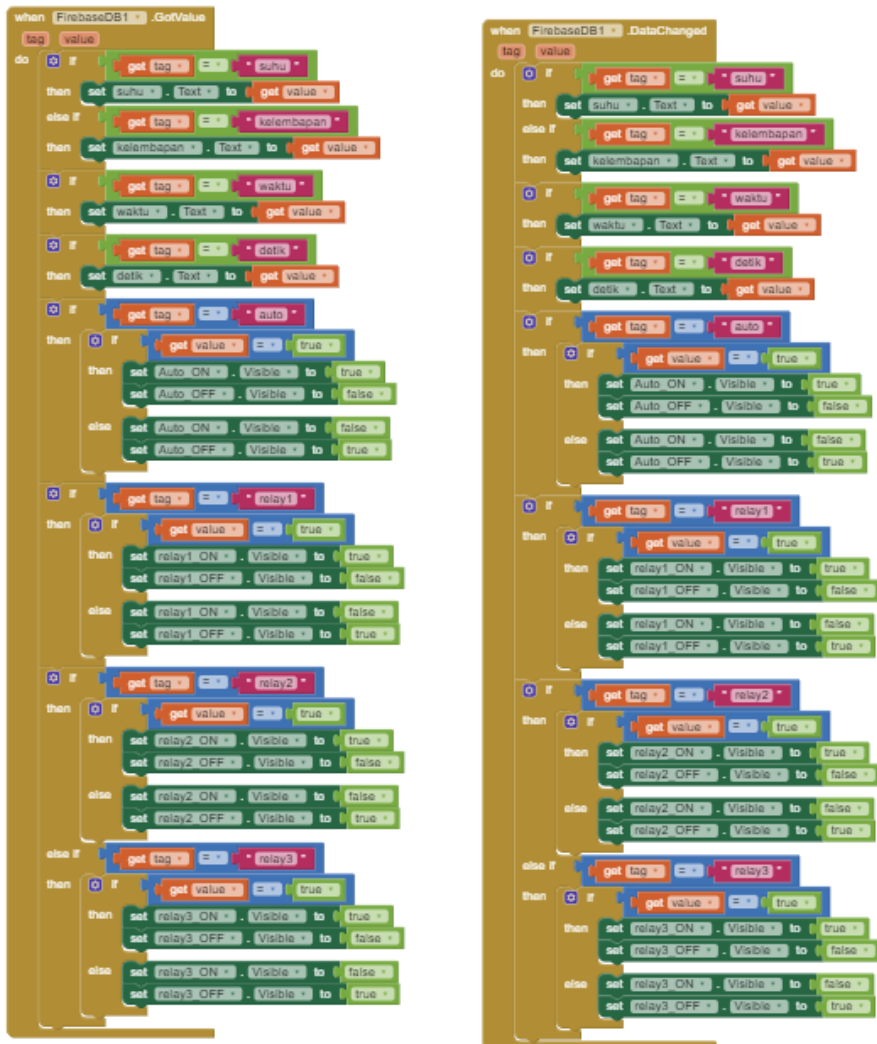
Lampiran 4 Blok Aplikasi Mobile



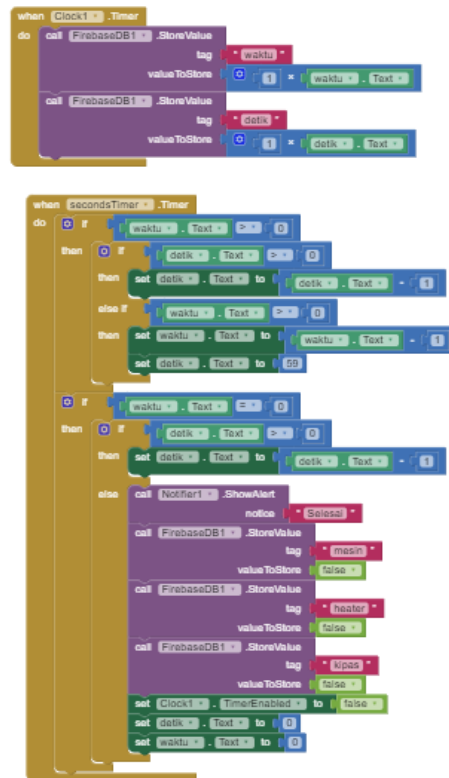
Blok tampilan awal untuk menginput Massa Awal, Kadar air, dan waktu pengeringan.



Inisialisasi variable yang digunakan



Blok Untuk Menampilkan hasil pembacaan sensor-sensor dan aktuator yang diambil dari firebase



Blok untuk mengatur timer



Blok untuk mengatur ON/OFF aktuator