

SKRIPSI

Menentukan Kualitas Kopi *Arabica* Berdasarkan Kekentalan dan Keasaman Kopi *Arabica*

Oleh

Sigit Widodo Sukarji

065117122



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2023**

SKRIPSI

Menentukan Kualitas Kopi *Arabica* Berdasarkan Kekentalan dan Keasaman Kopi *Arabica*

Diajukan sebagai salah satu syarat untuk memperoleh

Gelar Sarjana Komputer Jurusan Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Oleh

Sigit Widodo Sukarji

065117122



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2023**

HALAMAN PENGESAHAN

Judul : Menentukan Kualitas Kopi *Arabica* Berdasarkan Kekentalan dan Keasaman Kopi *Arabica*
Nama : Sigit Widodo Sukarji
NPM : 065117122

Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA – UNPAK



Teguh Puja Negara, M.Si.

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA – UNPAK



Dr. Andi Chairunnas, M.Pd., M.Kom.

Mengetahui,

Ketua Program Studi Ilmu Komputer
FMIPA – UNPAK



Arie Qur'ania, M.Kom.

Dekan
FMIPA UNPAK



Asep Denih, S.Kom., M.Sc., Ph.D.

PERNYATAAN KEASLIAN KARYA TULIS

Dengan ini saya menyatakan bahwa:

Nama : Sigit Widodo Sukarji
NPM : 065117122
Program Studi : Ilmu Komputer
Fakultas : Matematika dan Ilmu Pengetahuan Alam

Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian – bagian dimana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar – benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai peraturan yang berlaku.

Bogor, Oktober 2023

Sigit Widodo Sukarji
065117122

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Saya yang bertandatangan di bawah ini :

Nama : Sigit Widodo Sukarji
NPM : 065117122
Judul Skripsi : Menentukan Kualitas Kopi *Arabica* Berdasarkan Kekentalan dan Keasaman Kopi *Arabica*

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

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 di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, Oktober 2023

Sigit Widodo Sukarji
065117122

RIWAYAT HIDUP



Penulis bernama Sigit Widodo Sukarji yang dilahirkan di Bogor, 01 Maret 1999, anak pertama dari pasangan Bapak Widodo dan Ibu Euis Sukarsih. Penulis beragama islam dan beralamat di Kampung Cinangka Wates, RT 03 dan RW 04, Kelurahan Cinangka, Kecamatan Ciampea, Kabupaten Bogor, Jawa Barat. Penulis menyelesaikan pendidikan di Sekolah Dasar Negeri Cinangka 01 pada tahun 2011, kemudian pada tahun 2014 menyelesaikan pendidikan di MTS Tarbiyatul Falah dan Penulis adalah alumni dari SMK Pelita Ciampea lulusan angkatan tahun 2017. Pada tahun 2017 Penulis terdaftar sebagai mahasiswa di Universitas Pakuan Fakultas Matematika dan Ilmu Pengetahuan Alam Jurusan Ilmu Komputer. Dengan ketekunan dan motivasi tinggi Penulis telah berhasil menyelesaikan pengerjaan tugas akhir skripsi ini. Semoga dengan penulisan tugas akhir skripsi ini mampu memberikan kontribusi positif bagi dunia pendidikan. Akhir kata Penulis mengucapkan rasa syukur yang sebesar – besarnya atas terselesaikannya skripsi yang berjudul “Menentukan Kualitas Kopi *Arabica* Berdasarkan Kekentalan dan Keasaman Kopi *Arabica*”

RINGKASAN

Sigit Widodo Sukarji 2023. Menentukan Kualitas Kopi *Arabica* Berdasarkan Kekentalan dan Keasaman Kopi *Arabica*. Dibimbing oleh **Dr. Andi Chairunnas, M.Pd., M.Kom.** dan **Teguh Puja Negara, M.Si.**

Indonesia merupakan salah satu penghasil kopi terbesar di dunia. Pada lima tahun terakhir produksi kopi di Indonesia cenderung meningkat. Berdasarkan laporan statistik Indonesia jumlah produksi kopi mencapai 774,60 ribu ton pada tahun 2021. Meningkat sekitar 1,62% dibandingkan dengan tahun sebelumnya sebanyak 762,20 ribu ton. Salah satu jenis kopi yang dihasilkan adalah kopi dengan jenis *arabica*. Kopi *arabica* merupakan jenis kopi yang umumnya tumbuh pada ketinggian 3000-7000 kaki di atas permukaan laut. Daerah subtropis merupakan salah satu tempat yang ideal untuk kopi *arabica* karena memiliki tanah yang gembur, curah hujan yang merata, serta sinar matahari yang cukup sehingga kopi dengan jenis *arabica* dapat tumbuh dengan baik. Di pasaran sendiri kopi dengan jenis *arabica* banyak diperjual belikan dengan bebas dengan berbagai macam harga dan kualitas. Oleh karena itu, maka perlu untuk mengetahui kualitas kopi yang akan dibeli agar mendapatkan kualitas kopi yang terbaik.

Metode penelitian yang digunakan pada penelitian ini terdiri dari sepuluh tahapan yaitu: perencanaan, studi referensi, desain elektrik, pengadaan komponen, pengujian komponen, implementasi elektrik, desain *software*, implementasi *software*, uji *software*, desain mekanik, implementasi mekanik, integrasi, uji keseluruhan dan aplikasi. Pada penelitian ini menggunakan sensor pH untuk mengetahui tingkat keasaman pada sampel kopi dan menggunakan *waterflow* untuk mengetahui nilai kekentalan pada sampel kopi yang diuji. Pada penelitian ini sampel kopi yang digunakan untuk pengujian menggunakan metode manual brew menggunakan alat moka pot untuk mengekstrak sampel kopi yang digunakan. Untuk mempermudah mengetahui kualitas kopi pada sampel yang diuji pada penelitian ini menggunakan algoritma knn pada arduino. Penggunaan algoritma knn pada arduino ini mulai dari pengambilan data sensor pH dan *waterflow* pada memori arduino untuk digunakan sebagai dataset pada algoritma knn.

Hasil penelitian yang dilakukan menunjukkan kopi *arabica* yang digunakan dalam penelitian ini memiliki kualitas kopi yang baik, baik dalam keasaman maupun kekentalan. Dominan dengan status kopi dengan kualitas baik sebanyak 12 data sedangkan untuk kopi dengan kualitas kurang baik sebanyak 8. Dengan ini kopi jenis *arabica* yang digunakan pada penelitian ini mempunyai kualitas yang baik dengan rata-rata nilai pH 4.82 dan kekentalan sebesar 3.61. Pengujian untuk membandingkan nilai keasaman dan kekentalan antara kopi *arabica* dan minuman lain nya menghasilkan hasil akhir untuk kopi *arabica* dan kopi nescafe *robusta* mendapatkan hasil yang cukup mendekati dengan nilai parameter keasaman dan kekentalan. Sedangkan untuk minuman pocari sweat mendapatkan hasil yang sebaliknya.

KATA PENGANTAR

Puji dan syukur ke hadirat Allah yang senantiasa memberi nikmat dan hidayah-Nya yang begitu besar kepada penulis untuk selalu beribadah dan berdoa kepada-Nya. Serta atas rahmat-Nya penulis dapat diberikan ketenangan dan kemudahan dalam menyelesaikan laporan penelitian dengan judul **“Menentukan Kualitas Kopi Arabica Berdasarkan Kekentalan dan Keasaman Kopi Arabica”**

Atas segala bentuk upaya, bimbingan, dan bantuan dari segala pihak, penulis ingin mengucapkan banyak terima kasih sebesar besarnya kepada:

1. Dr. Andi Chairunnas, M.Pd., M.Kom. selaku Dosen Pembimbing Utama yang telah memberikan arahan, saran, dan meluangkan waktu untuk membantu pembuatan laporan ini.
2. Teguh Puja Negara, M.Si. selaku Dosen Pembimbing Pendamping yang telah memberikan saran dan waktunya.
3. Arie Qur'ania, M.Kom. selaku Ketua Program Studi Ilmu Komputer.
4. Kedua orang tua tercinta yang selalu memberikan doa, kebaikan, keikhlasan, dan dukungan baik moril dan materil selama ini.
5. Seluruh teman-teman, sahabat, dan khususnya kelas E dan F angkatan 2017, Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Pakuan yang telah memberikan semangat dalam penyusunan laporan ini.

Penulis menyadari bahwa laporan penelitian ini masih jauh dari kata sempurna, oleh karena itu segala kritik dan saran yang membangun akan penulis terima dengan senang hati. Semoga penulisan laporan ini dapat diterima dan membantu bagi penulis, rekan – rekan mahasiswa, dan seluruh pembaca pada umumnya.

Bogor, Oktober 2023

Sigit Widodo Sukarji
065117122

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	iii
PERNYATAAN KEASLIAN KARYA TULIS	iv
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	v
RIWAYAT HIDUP	vi
RINGKASAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Penelitian	2
1.3. Ruang Lingkup Penelitian.....	2
1.4. Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	3
2.1. Tinjauan Pustaka	3
2.1.1. Jenis – Jenis Kopi	3
2.1.2. Sensor PH	4
2.1.3. Sensor <i>Water Flow</i>	4
2.1.4. <i>NodeMCU ESP8266</i>	5
2.1.5. <i>Internet of Things (IOT)</i>	6
2.1.6. Android	6
2.1.7. <i>K-Nearest Neighbours (KNN)</i>	6
2.2. Penelitian Terdahulu	7
2.3. Perbandingan Penelitian.....	8
BAB III METODE PENELITIAN	9
3.1. Metode Penelitian.....	9
3.1.1. Perencanaan Penelitian	9
3.1.2. Studi Referensi.....	10
3.1.3. Desain Elektrik	10

3.1.4.	Pengadaan Komponen	10
3.1.5.	Pengujian Komponen.....	10
3.1.6.	Implementasi Elektrik.....	10
3.1.7.	Desain <i>Software</i>	10
3.1.8.	Implementasi <i>Software</i>	10
3.1.9.	Uji <i>Software</i>	10
3.1.10.	Desain Sistem Mekanik	11
3.1.11.	Impelementasi Mekanik.....	12
3.1.12.	Integrasi	12
3.1.13.	Uji Keseluruhan	12
3.1.14.	Aplikasi.....	12
3.2.	Metode <i>K-Nearest Neighbours</i> (KNN)	13
3.3.	Waktu dan Tempat Penelitian	13
3.4.	Alat dan Bahan.....	13
3.4.1.	Alat.....	13
3.4.2.	Bahan	13
BAB IV RANCANGAN DAN IMPLEMENTASI		14
4.1.	Perencanaan Penelitian.....	14
4.1.1.	Analisa Kebutuhan <i>Hardware</i>	14
4.1.2.	Analisa Kebutuhan <i>Software</i>	14
4.2.	Studi Referensi	14
4.3.	Desain Elektrik.....	14
4.4.	Pengadaan Komponen.....	15
4.5.	Pengujian Komponen	15
4.6.	Implementasi Elektrik	15
4.7.	Desain <i>Software</i>	16
4.8.	Implementasi <i>Software</i>	17
4.9.	Uji <i>Software</i>	17
4.10.	Desain Mekanik.....	17
4.11.	Implementasi Mekanik	18
4.12.	Integrasi	18
4.13.	Penerapan Metode KNN.....	18
BAB V HASIL DAN PEMBAHASAN		23
5.1.	Hasil Penelitian	23

5.2.	Pengujian Fungsional Keseluruhan Sistem (<i>Overall Sistem</i>).....	23
5.2.1.	Pengujian Struktural	23
5.2.2.	Pengujian Fungsional.....	24
5.2.3.	Pengujian Keseluruhan Sistem	25
5.2.4.	Uji Validasi	26
BAB VI	KESIMPULAN DAN SARAN	32
6.1.	Kesimpulan	32
6.2.	Saran.....	32
DAFTAR PUSTAKA	33
LAMPIRAN	36

DAFTAR GAMBAR

	Halaman
Gambar 1. Jenis – Jenis Kopi (Haidar, 2020)	3
Gambar 2. Sensor PH (Ilmu, 2019).....	4
Gambar 3. Sensor <i>Waterflow</i> (Farhan, 2022).....	5
Gambar 4. NodeMCU ESP8266 (Sindhu et al., 2021).....	5
Gambar 5. Ilustrasi IOT (Ayu, 2020).....	6
Gambar 6. Metode <i>Hardware Programming</i> (Chairunnas & Pamungka, 2019; Herman & Chairunnas, 2019; Ismangil & Ardyahadistia, 2021)	9
Gambar 7. Uji <i>Software</i>	11
Gambar 8. Diagram Alir Uji Keseluruhan	12
Gambar 9. Diagram Blok Sistem	14
Gambar 10. Diagram Blok Sistem	15
Gambar 11. Diagram Blok Sistem	16
Gambar 12. Diagram Alir Sistem.....	16
Gambar 13. Implementasi <i>Software</i>	17
Gambar 14. Desain Mekanik.....	18
Gambar 15. Model Perakitan Alat.....	18
Gambar 16. Model Perakitan Alat.....	23
Gambar 17. Proses Pengujian Fungsional: Arduino Uno (a); NodeMCU Esp8266 (b); Sensor pH (c); Sensor <i>Waterflow</i> (b)	24
Gambar 18. Pengujian Model Alat.....	25
Gambar 19. Halaman Utama Aplikasi	26
Gambar 20. Proses Pengambilan Dataset Kopi pada Algoritma KNN Arduino.....	28
Gambar 21. Ekstrasi Kopi menggunakan Mokapot	28
Gambar 22. Proses Pengujian Sampel Kopi dengan KNN	28

DAFTAR TABEL

	Halaman
Tabel 1. Perbandingan Penelitian.....	8
Tabel 2. Hasil Pengujian Komponen	15
Tabel 3. Data Latih Kopi Arabica	19
Tabel 4. Data Uji Kopi Arabica	19
Tabel 5. Hasil Perhitungan Jarak Euclidean Distance	20
Tabel 6. Urutan berdasarkan Jarak Euclidean Distance Terkecil.....	21
Tabel 7. Data Latih dengan Hasil Perhitungan Jarak Euclidean	21
Tabel 8. Hasil Voting Data Berdasarkan Nilai K.....	22
Tabel 9. Hasil Klasifikasi Data Uji dengan KNN	22
Tabel 10. Hasil Pengujian Struktural	24
Tabel 11. Hasil Pengujian Fungsional pada Komponen	25
Tabel 12. Hasil Uji Validasi Sensor PH.....	26
Tabel 13. Hasil Uji Validasi Sensor Waterflow	27
Tabel 14. Uji Validasi Kualitas Kopi	29
Tabel 15. Uji Validasi Kualitas Kopi dengan Minuman Lain	30

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. <i>Source Code</i> KNN Ardiuno Uno.....	37
Lampiran 2. <i>Source Code</i> KNN ESP	46

BAB I

PENDAHULUAN

1.1. Latar Belakang

Indonesia merupakan salah satu penghasil kopi terbesar di dunia. Pada lima tahun terakhir produksi kopi di Indonesia cenderung meningkat. Berdasarkan laporan statistik Indonesia jumlah produksi kopi mencapai 774,60 ribu ton pada tahun 2021. Meningkat sekitar 1,62% dibandingkan dengan tahun sebelumnya sebanyak 762,20 ribu ton (Mutia Annur, 2022). Salah satu jenis kopi yang dihasilkan adalah kopi dengan jenis *arabica*. Kopi *arabica* merupakan jenis kopi yang umumnya tumbuh pada ketinggian 3000-7000 kaki di atas permukaan laut. Daerah subtropis merupakan salah satu tempat yang ideal untuk kopi *arabica* karena memiliki tanah yang gembur, curah hujan yang merata, serta sinar matahari yang cukup sehingga kopi dengan jenis *arabica* dapat tumbuh dengan baik (Firdayani & Aisyah, 2020). Di pasaran sendiri kopi dengan jenis *arabica* banyak diperjual belikan dengan bebas dengan berbagai macam harga dan kualitas. Oleh karena itu, maka perlu untuk mengetahui kualitas kopi yang akan dibeli agar mendapatkan kualitas kopi yang terbaik.

Beberapa penelitian terkait dengan penentuan kualitas kopi pernah dilakukan oleh (Susilawati et al., 2020). Pada penelitian ini dilakukan pembuatan *prototype* alat pendeteksi kematangan buah kopi berbasis arduino menggunakan sensor APDS Gy-9960. Pada sistem otomasi ini menggunakan arduino uno dan sensor APDS GY-9960 sebagai sensor warna untuk melihat nilai RGB pada setiap buah kopi yang diambil untuk menentukan kelompok warna buah yaitu merah, orange dan hijau. Buah dengan tiga kategori tersebut akan dikelompokkan dengan suatu sistem pemilah, dan untuk proses pemilahan tersebut menggunakan motor servo. Setelah dilakukannya proses pemilahan maka untuk informasi hasil pemilahan akan ditampilkan pada LCD. Penelitian ini menghasilkan rancangan prototipe sistem pemilahan buah kopi berdasarkan warna yang mampu mengelompokkan berdasarkan warna.

Penelitian selanjutnya dilakukan oleh (Aprianto, 2020). Pada penelitian ini dilakukan proses rancang bangun sistem *monitoring* pH dan suhu pada larutan kopi berbasis *fuzzy logic* di PT. Harum Alam Segar. Pada penelitian ini alat yang dirancang dapat menentukan kualitas larutan kopi berbasis *fuzzy logic* pada *microcontroller* Arduino uno. Teknologi ini menggabungkan dua jenis sensor yaitu termokopel tipe K dan pH meter untuk menentukan kualitas larutan kopi pada tahap akhir *monitoring quality control* (QC). Pengambilan keputusan pada alat ini menggunakan logika *fuzzy sugeno* orde nol. Selain itu alat ini juga didesain untuk dapat berkomunikasi secara serial *personal computer* (PC) dan *liquid crystal display* (LCD) untuk memonitoring kualitas larutan kopi. Dari hasil pengujian dapat disimpulkan bahwa alat yang telah dibuat dapat mendeteksi kualitas larutan kopi. Dalam pengujian digunakan larutan kopi dengan jenis berbeda. Dengan alat ini akan mempermudah QC dalam memonitoring larutan kopi dengan cepat, tepat dan mengurangi tenaga kerja.

Mengacu pada permasalahan yang sudah dibahas dan beberapa penelitian yang sudah dilakukan maka pada penelitian ini akan dilakukan pembuatan sistem berbasis *internet of things* (IOT) untuk menentukan kualitas kopi *arabica* berdasarkan

kekentalan dan keasaman kopi *arabica*. Pada penelitian ini akan menggunakan sensor ph untuk mengenali keasaman kopi *arabica* dan *waterflow* cairan untuk membaca kekentalan kopi *arabica*.

1.2. Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan menentukan kualitas kopi *arabica* berdasarkan kekentalan dan keasaman kopi *arabica* yang berguna untuk mengetahui kualitas kopi *arabica*.

1.3. Ruang Lingkup Penelitian

Ruang lingkup pada penelitian ini meliputi:

1. Kopi yang digunakan menggunakan jenis kopi *arabica* gayo
2. Sensor yang digunakan pada penelitian ini meliputi sensor PH untuk mengenali keasaman kopi *arabica* gayo dan *water flow* untuk mengenali kekentalan kopi *arabica* gayo.
3. Metode yang digunakan pada tahap ini yaitu metode hardware programming.
4. Kontrol sistem menggunakan *mikrokontroler* Arduino dan NodeMCU untuk menerima input sensor.
5. Pada penelitian ini menggunakan algoritma *k-nearest neighbors* (KNN) untuk klasifikasi data uji kopi *arabica* guna untuk menentukan kualitas kopi *arabica* berdasarkan data sampel kopi *arabica* gayo.
6. Penelitian ini juga dilengkapi oleh aplikasi android guna untuk memudahkan pengguna melihat hasil dari klasifikasi kualitas kopi *arabica*

1.4. Manfaat Penelitian

Manfaat pada penelitian ini adalah dapat mengetahui kualitas kopi *arabica*.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

2.1.1. Jenis – Jenis Kopi

Kopi yang banyak tumbuh di tanah Indonesia terdiri dari empat jenis biji, yaitu robusta, arabika, ekselsa dan liberika. Masing-masing biji kopi memiliki karakteristik rasa sendiri-sendiri. Biji kopi ini baru bisa dikonsumsi setelah melalui proses cuci, pengeringan, pemanggangan dan penggilingan (Sunarharum et al., 2019). Pada Gambar 1 menunjukkan jenis-jenis kopi.



Gambar 1. Jenis – Jenis Kopi (Haidar, 2020)

Berdasarkan Gambar 1., terdapat 4 jenis -jenis kopi sebagai berikut :

1) Kopi Arabica

Arabica memiliki rasa lebih ramah di mulut ketimbang robusta. Memiliki citarasa manis dan asam, dan memiliki lapisan rasa lebih kompleks dibanding robusta. Arabika lebih cocok ditanam di dataran tinggi dengan curah hujan sedang dan paparan sinar matahari alami yang cukup. Kopi arabika juga gampang terkena hama. Oleh sebab itu, jenis kopi ini sebaiknya ditanam secara homogen. Ketika ditanam secara heterogen dan hama merebak, hama yang ada bisa dalam sekejap merusak budidaya tanaman yang lain. Menurut (Nurhayati, 2018) kopi jenis *arabica* memiliki kekentalan sebesar 3,61. Sedangkan menurut (Kayaputri et al., 2022) kopi *arabica* mempunyai nilai pH dengan nilai 4,85 – 5,15.

2) Kopi Robusta

Nama robusta diambil dari sifat kopi ini yang lebih “*robust*” atau kokoh, bisa hidup di lingkungan lebih fleksibel dibanding arabika. Meski lebih mudah ditanam atau dibudidayakan dan lebih kuat terhadap hama, namun kepopuleran robusta ada di bawah arabika karena robusta tak memiliki variasi rasa dan aroma layaknya arabika. Kopi robusta lebih hitam, lebih banyak mengandung kafein, dan citarasanya lebih berkuatat pada *earthy flavor*, yaitu beraroma layaknya tanah sehabis hujan, alias tidak beraroma buah-buahan layaknya arabika. Robusta banyak dicari oleh pecinta kopi yang memuja

citarasa kopi yang pahit mantap. Robusta jugalah jenis biji kopi yang paling cocok diolah menjadi espresso. Kopi Robusta memiliki pH dengan nilai sekitar 5,25 – 5,40 (Kayaputri et al., 2022). Menurut (Nurhayati, 2017) kopi robusta memiliki nilai kekentalan antara 0,61 – 3,61.

3) Kopi Ekselsa

Biji kopi ekselsa sering disamakan dengan liberika. Hal ini lantaran ekselsa juga tumbuh di lingkungan yang sama dengan liberika, dan pohonnya pun juga bisa tumbuh tinggi layaknya liberika. Namun soal citarasa dan aroma, ekselsa sedikit berbeda dengan liberika. Ekselsa memiliki citarasa cenderung ke buah-buahan. Kopi Ekselsa memiliki pH dengan nilai sekitar 5,45 – 5,60 dan memiliki kekentalan dengan nilai sekitar 0,55 – 3,80 (Wanita et al., 2021).

4) Kopi Liberika

Liberika memiliki aroma dan citarasa yang unik. Gabungan antara citarasa dan aroma bunga, buah, dan sedikit asap yang beraroma kayu. Pohon kopi liberika tumbuh lebih tinggi daripada kopi arabika dan robusta. Kopi liberika lebih hebat dalam bertahan hidup dibanding arabika. Biji kopi ini bisa hidup di tanah yang kurang subur sekalipun. Kopi Liberika memiliki pH dengan nilai sekitar 6,00 – 6,40 memiliki kekentalan dengan nilai sekitar 0,54 – 3,76 (Hanifah et al., 2022).

2.1.2. Sensor PH

Sensor pH adalah sensor yang digunakan untuk mengetahui derajat keasaman. pH meter adalah alat yang digunakan untuk mengukur tingkat keasaman atau kebasaan larutan. Prinsip utama kerja pH meter adalah terletak pada sensor *probe* berupa elektroda kaca (*glass electrode*) dengan jalan mengukur jumlah ion H_3O^+ di dalam larutan. Dalam penggunaannya, sensor pH perlu dikalibrasi berkala agar keakuratannya dapat terjaga. Beberapa produsen sensor pH pada umumnya menyertakan instrumen untuk melakukan kalibrasi secara manual. Jika sensor pH dihubungkan dengan Arduino Uno, kalibrasi dapat dilakukan melalui program antarmuka kalibrasi sensor pH (pengembangan dari *library* sensor pH yang sudah tersedia) (Mufida et al., 2020). Bentuk Sensor PH ditunjukkan pada Gambar 2.



Gambar 2. Sensor PH (Ilmu, 2019)

2.1.3. Sensor Water Flow

Sensor *waterflow* merupakan sensor yang dapat membaca aliran air pada suatu tempat. Sensor ini bekerja membaca kecepatan putaran rotor yang disebabkan oleh kecepatan aliran air. Prinsip kerja sensor ini adalah mengukur aliran air dengan cara

menghitung putaran dari sebuah kincir yang terdapat di dalam alat ini. Kincir akan otomatis berputar jika ada aliran air yang melewatinya. Di dalam kincir terdapat sebuah rotor yang terdapat magnet dan ketika berputar akan menghasilkan magnet sesuai fenomena *hall effect*. Fenomena *hall effect* didasarkan pada efek medan magnetik pada partikel bermuatan yang bergerak. Semakin cepat aliran yang mengalir melalui sensor ini, maka akan semakin cepat juga putaran rotor sehingga angka yang terbaca pada sensor tersebut menjadi besar. angka tersebut merupakan sinyal *output* berupa gelombang kotak yang nantinya akan dilakukan perhitungan sehingga kita dapat mengetahui debit dan volume air yang lewat dari alat ini (Putra & Stefanus, 2019). Bentuk sensor *waterflow* ditunjukkan pada Gambar 3.



Gambar 3. Sensor *Waterflow* (Farhan, 2022)

2.1.4. NodeMCU ESP8266

NodeMCU adalah sebuah platform IoT yang bersifat *opensource*. Terdiri dari perangkat keras berupa *system on chip* ESP8266 dari ESP8266 buatan *Espressif System* yang menggunakan bahasa pemrograman *scripting* Lua. Istilah *NodeMCU* secara *default* sebenarnya mengacu pada *firmware* yang digunakan sebagai perangkat keras *development kit*. *NodeMCU* bisa dianalogikan sebagai *board arduino*-nya ESP8266. Dalam seri tutorial ESP8266 *embeddednesia* pernah membahas bagaimana memprogram ESP8266 sedikit merepotkan karena diperlukan beberapa teknik wiring serta tambahan modul USB to serial untuk mengunduh program. Namun, *NodeMCU* telah me-package ESP8266 ke dalam sebuah *board* yang kompak dengan berbagai fitur layaknya *mikrokontroler* + kapabilitas akses terhadap Wifi juga chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan sebagai kabel data dan kabel *charging smartphone* Android (Sindhu et al., 2021). Bentuk dari *NodeMCU* ESP8266 ditunjukkan pada Gambar 4.



Gambar 4. *NodeMCU* ESP8266 (Sindhu et al., 2021)

2.1.5. *Internet of Things (IOT)*

Internet of things (IOT) adalah *embedded system* yang bertujuan untuk memperluas pemanfaatan dari konektivitas internet yang tersambung secara terus-menerus. Kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata contohnya seperti bahan pangan, elektronik, peralatan yang terhubung dengan sensor dan terhubung dengan jaringan. Keterkaitan objek dengan koneksi internet sebagai dasar pengembangan semua layanan. Benda-benda fisik diintegrasikan ke dalam jaringan informasi secara berkesinambungan, dan di mana benda-benda fisik tersebut berperan secara aktif dalam proses bisnis. Tersedia layanan pintar yang saling terkoneksi, mencari dan mengubah status mereka sesuai dengan setiap informasi yang dikaitkan, disamping memperhatikan masalah privasi dan keamanan. Cara kerja dari IOT adalah dengan memanfaatkan pemrograman di setiap perintah untuk sebuah instruksi kepada mesin tanpa bantuan manusia dengan menggunakan sambungan atau koneksi internet. Seperti bagaimana mengolah data yang diperoleh dari peralatan elektronik melalui sebuah *interface* antara pengguna dan peralatan itu. Penggunaan sensor secara *real time* mengkonversikan ke dalam mesin format yang dimengerti sehingga akan mudah dipertukarkan antara berbagai bentuk format data (*Thing*) (Susanto et al., 2022). Ilustrasi IOT ditunjukkan pada Gambar 5.



Gambar 5. Ilustrasi IOT (Ayu, 2020)

2.1.6. **Android**

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau *smartphone*. Android merupakan generasi baru *platform mobile* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan dibawah GNU, *General Public Licensi* versi 2 (GPLv2), yang sering dikenal dengan istilah “*copyleft*” lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah terms. Android didistribusikan di bawah lisensi *Apache Software* (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya (Nurhidayati & Nur, 2021)

2.1.7. *K-Nearest Neighbours (KNN)*

KNN merupakan salah satu algoritma *machine learning* yang bekerja dengan mencari sejumlah k pola terdekat dengan pola masukan kemudian menentukan kelas berdasarkan pola terbanyak di antara k pola tersebut. Proses pelatihan KNN menghasilkan nilai K yang memberi akurasi tertinggi dalam menggeneralisasi data

data yang akan datang. Proses pelatihan nilai k ini berguna untuk melihat nilai k beserta hasilnya sampai dihasilkan hasil nilai k yang paling optimum. KNN banyak digunakan untuk aplikasi *data mining*, *statistical pattern recognition*, *image processing* dan banyak hal lain. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru berdasarkan atribut dan training sampel. KNN biasa digunakan untuk prediksi, prediksi digunakan untuk nilai dari hasil yang akan datang. Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Algoritma KNN memiliki beberapa kelebihan yaitu tidak terlalu terpengaruh terhadap data *training* yang memiliki banyak *noise* dan efektif apabila *training* data-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter K (jumlah dari tetangga terdekat)(Admojo & Ahsanawati, 2020; Cahyanti et al., 2020; Hidayati et al., 2020; Puspita & Widodo, 2021; Yulianto & Darwis, 2021). Adapun urutan algoritma KNN adalah sebagai berikut:

- 1) Menentukan parameter K (jumlah tetangga paling dekat)
- 2) Menghitung kuadrat jarak *euclidian* (*euclidean distance*) antara data baru terhadap data latih (*training data*). Adapun persamaan yang digunakan untuk menghitung jarak *euclidean distance* menggunakan persamaan (1) (Wang et al., 2020; Zhao et al., 2021).

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \dots\dots\dots(1)$$

Dimana d merupakan jarak data; pi adalah sampel data; qi adalah data uji atau data latih; i adalah variabel data; dan n adalah jumlah data.

- 3) Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak euclidean terkecil
- 4) Mengumpulkan kategori Y (klasifikasi nearest neighbors)
- 5) Keluarkan hasil / voting berdasarkan nilai K yang telah ditentukan

Keluaran dari algoritma KNN akan berupa kelas dari data yang dicari atau data sampel yang dicari dan akan dihitung berdasarkan kemunculan data terbanyak menurut nilai K yang telah diatur sebelumnya (Stanley et al., 2020).

2.2. Penelitian Terdahulu

Penelitian terdahulu terkait dengan penelitian yang akan dilakukan sebagai berikut:

- Nama : Helfy Susilawati, Akhmad Fauzi Ikhsan, Faizal Salman (2020)
- Judul : *Prototyping* Alat Pendeteksi Kematangan Buah Kopi Berbasis Arduino Menggunakan Sensor Apds Gy-9960
- Isi : Pada sistem otomasi ini menggunakan Arduino Uno dan sensor APDS GY-9960 sebagai sensor warna untuk melihat nilai RGB pada setiap buah kopi yang diambil untuk menentukan kelompok warna buah yaitu merah, orange dan hijau. Buah dengan 3 kategori tersebut akan dikelompokkan dengan suatu sistem pemilah, dan untuk proses pemilahan tersebut menggunakan motor servo. Setelah dilakukannya proses pemilahan maka untuk informasi hasil pemilahan akan ditampilkan pada LCD. Penelitian ini menghasilkan rancangan prototipe sistem pemilahan buah kopi berdasarkan warna yang mampu mengelompokkan berdasarkan warna

Nama : Zarif Aprianto (2020)
 Judul : Rancang bangun sistem *monitoring* pH dan suhu pada larutan kopi berbasis *fuzzy logic* di PT. Harum Alam Segar
 Isi : Pada penelitian ini alat dapat menentukan kualitas larutan kopi berbasis *fuzzy logic* pada *microcontroller* Arduino uno. Teknologi ini menggabungkan dua jenis sensor yaitu termokopel tipe K dan pH meter untuk menentukan kualitas larutan kopi pada tahap akhir monitoring QC. Pengambilan keputusan pada alat ini menggunakan logika *fuzzy* sugeno orde nol. Selain itu alat ini juga didesain untuk dapat berkomunikasi secara serial PC dan LCD untuk memonitoring kualitas larutan kopi. Dari hasil pengujian dapat disimpulkan bahwa alat yang telah dibuat dapat mendeteksi kualitas larutan kopi. Dalam pengujian digunakan larutan kopi dengan jenis berbeda. Dengan alat ini akan mempermudah QC dalam memonitoring larutan kopi dengan cepat, tepat dan mengurangi tenaga kerja

2.3. Perbandingan Penelitian

Perbandingan penelitian ditampilkan pada Tabel 1.

Tabel 1. Perbandingan Penelitian

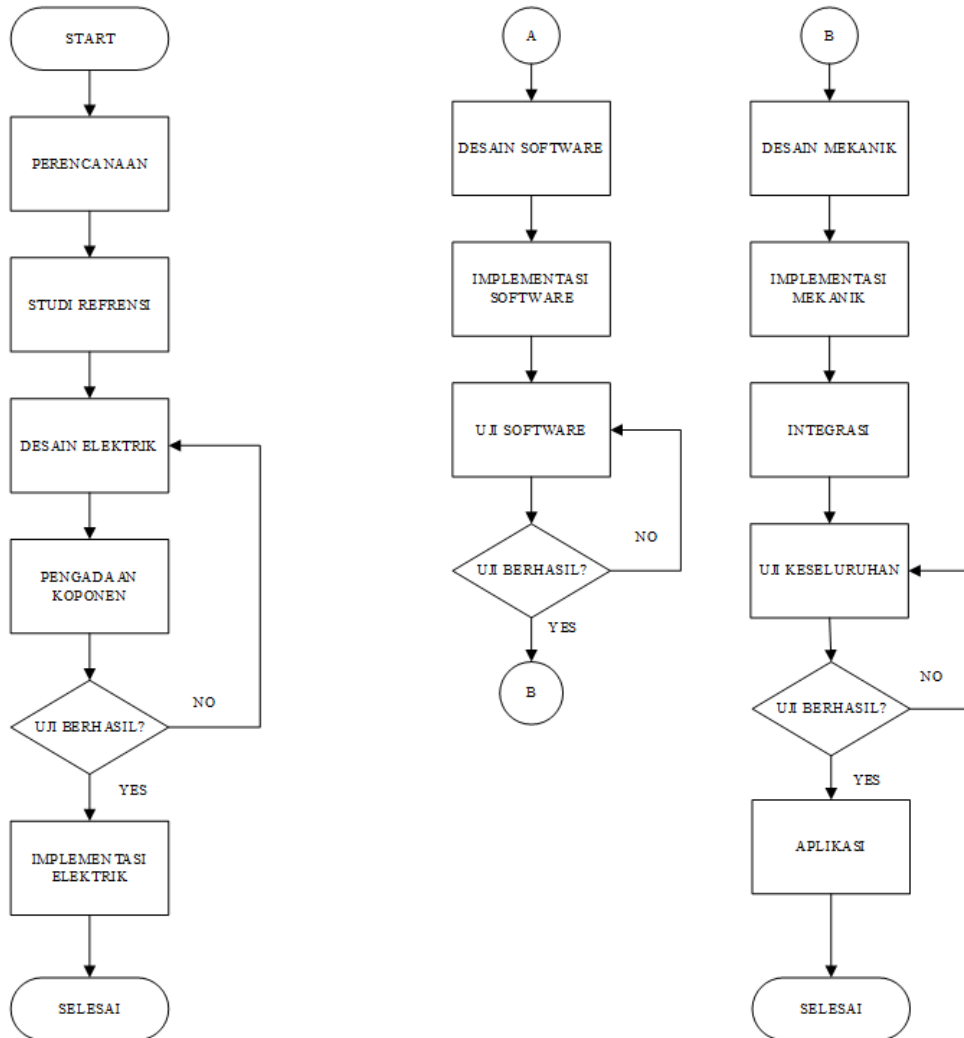
No	Nama & Tahun	Input				Proses		Output		
		sensor APDS GY-9960	termokopel tipe K	PH	Waterflow	Arduino	NodeMCU	Serial PC	Aplikasi	LCD
1	Helpy Susilawati, Akhmad Fauzi Ikhsan, Faizal Salman (2020)	√	-	-	-	√	-	-	-	√
2	Zarif Aprianto (2020)	-	√	√	-	√	-	√	-	√
3	Sigit Widodo Sukarji (2022)	-	-	√	√	√	√	-	√	√

BAB III

METODE PENELITIAN

3.1. Metode Penelitian

Pada penelitian ini dilakukan menggunakan metode penelitian bidang *hardware programming* (Chairunnas & Pamungka, 2019; Herman & Chairunnas, 2019; Ismangil & Ardyahadistia, 2021) yang dapat dilihat pada Gambar 6.



Gambar 6. Metode *Hardware Programming* (Chairunnas & Pamungka, 2019; Herman & Chairunnas, 2019; Ismangil & Ardyahadistia, 2021)

Berdasarkan Gambar 6, maka tahapan penelitian dapat diuraikan sebagai berikut:

3.1.1. Perencanaan Penelitian

Dalam perencanaan penelitian ini, terdapat beberapa hal penting yang harus diperhatikan dan dipertimbangkan, antara lain :

1. Keterangan awal penelitian

2. Estimasi kebutuhan alat dan bahan
3. Estimasi anggaran
4. Kemungkinan penerapan dari sistem yang dirancang

3.1.2. Studi Referensi

Setelah perencanaan telah matang dilanjutkan dengan penelitian awal dari aplikasi yang akan dibuat mulai dari pemilihan dan pengetesan komponen (alat dan bahan) yang akan digunakan, kemungkinan rancangan awal dan akhir dalam merancang alat untuk menentukan kualitas kopi arabica berdasarkan kekentalan dan keasaman kopi arabica.

3.1.3. Desain Elektrik

Dalam merancang desain elektrik terdapat beberapa hal yang harus diperhatikan, antara lain :

1. Sumber catu daya (seperti baterai atau *rectifier*) dan pembagian daya untuk masing-masing komponen.
2. Kebutuhan tegangan dan arus untuk mikrokontroler dan sensor.
3. Desain skema rangkaian

3.1.4. Pengadaan Komponen

Pengadaan komponen adalah tahap persiapan pengumpulan komponen-komponen yang akan di pakai nantinya agar pada saat proses perakitan tidak terhenti karena kekurangan komponen. Setelah pengadaan komponen selesai lalu dilanjut ke proses pengujian komponen.

3.1.5. Pengujian Komponen

Dalam tahapan pengujian komponen dilakukan pengujian komponen terhadap fungsi kerja komponen berdasarkan kebutuhan sistem yang akan dibuat. Pengujian komponen alat sangat penting dilakukan hal ini dilakukan agar pada tahapan selanjutnya komponen - komponen yang nantinya akan dirangkai dan dihubungkan satu persatu tidak ada masalah pada masing – masing komponennya.

3.1.6. Implementasi Elektrik

Implementasi elektrik adalah pengimplementasian dari gambaran rangkaian desain listrik yang telah dibuat sebelumnya.

3.1.7. Desain Software

Software yang pada umumnya dibutuhkan perancangan perangkat keras antara lain, software untuk sistem kontrol alat (aplikasi) dan *software interface* pada komputer PC. Dalam desain *software* yang digunakan dalam penelitian ini menggunakan perangkat lunak Ms. Office, Google Chrome, Visio 2019, Fritzing, Arduino IDE 1.8.10, Visual Studio Code dan Sketch Up untuk Bahasa pemrograman yang digunakan yaitu pemrograman Bahasa C.

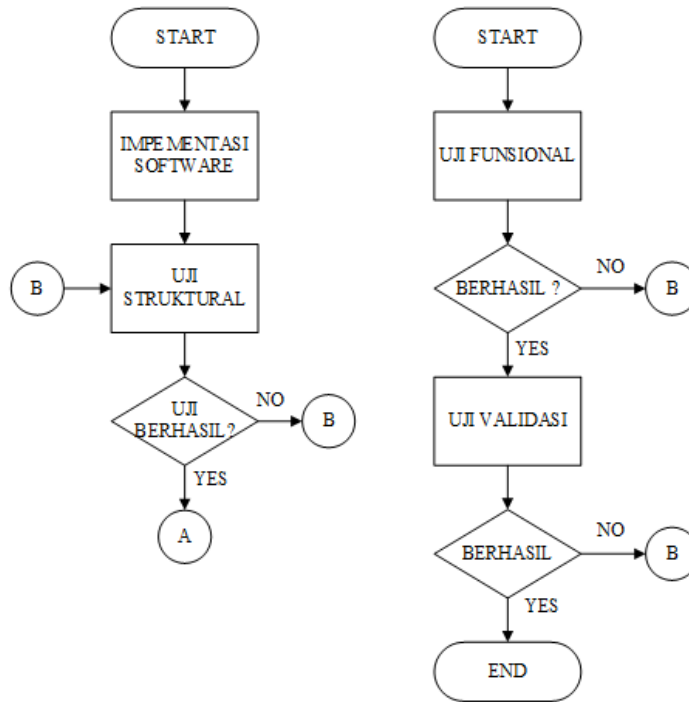
3.1.8. Implementasi Software

Implementasi *software* adalah pengimplementasian dari gambaran desain *software* yang telah di buat sebelumnya.

3.1.9. Uji Software

Pengujian *software* dilakukan agar desain yang telah dibuat sebelumnya sesuai dengan yang di inginkan sehingga pada saat penelitain bisa berfungsi dengan baik. Uji

software meliputi uji struktural, uji fungsional dan uji validasi yang dapat dilihat pada Gambar 7.



Gambar 7. Uji Software

Berdasarkan Gambar 7, maka pengujian *software* dapat diuraikan sebagai berikut:

1. Uji Struktural
Uji struktural pada *software* dilakukan dengan tujuan untuk mengetahui apakah *software* yang telah di buat dapat berfungsi dengan benar atau tidak.
2. Uji Fungsional
Uji fungsional pada *software* dilakukan dengan tujuan untuk mengintegrasikan sistem *software* yang telah di desain sebelumnya.
3. Uji Validasi
Uji Validasi pada *software* dilakukan dengan tujuan untuk menguji kinerja dari *software* yang telah dibuat apakah *software* tersebut dapat berjalan dengan baik atau tidak.

3.1.10. Desain Sistem Mekanik

Dalam perancangan perangkat keras, desain mekanik merupakan hal penting. Tahap desain sistem mekanis merupakan tahap dilakukannya pertimbangan meliputi kebutuhan sistem yang akan dibuat terhadap desain mekanik, diantaranya:

1. Dimensi dan massa keseluruhan sistem
2. Penempatan modul-modul elektronik
3. Ketahanan dan fleksibilitas terhadap lingkungan
4. Pengujian sistem mekanik yang telah dirancang

3.1.11. Implementasi Mekanik

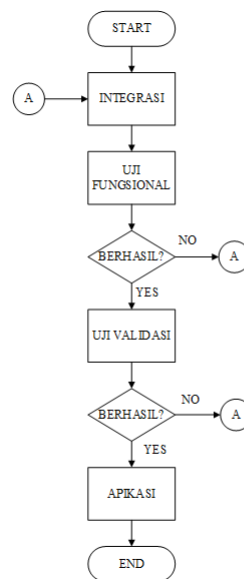
Implementasi mekanik adalah tahap pengimplementasian dari desain mekanik sebelumnya.

3.1.12. Integrasi

Modul listrik yang diintegrasikan dengan *software* di dalam kontrollernya, kemudian diintegrasikan dalam struktur mekanik yang telah dirancang.

3.1.13. Uji Keseluruhan

Pada tahapan ini dilakukan pengujian fungsi dari keseluruhan sistem. Pengetesan ini bertujuan untuk mengetahui apakah sistem yang dibuat sesuai dengan rancangannya atau tidak. Bila ada sistem yang tidak dapat bekerja dengan baik maka harus dilakukan proses perakitan ulang pada setiap desain sistemnya. Diagram alir uji keseluruhan disajikan pada Gambar 8.



Gambar 8. Diagram Alir Uji Keseluruhan

Berdasarkan Gambar 8, maka uji keseluruhan dapat diuraikan sebagai berikut:

1. Uji Fungsional
Uji fungsional dilakukan untuk mengintegrasikan sistem listrik dan *software* yang telah di desain. Pengujian ini dilakukan untuk meningkatkan performa dari perangkat lunak untuk pengontrolan desain listrik dan mengeliminasi *error (bug)* dari *software* yang telah dibuat.
2. Uji Validasi
Tahap ini bertujuan untuk menguji kinerja dari alat yang telah dibuat apakah alat tersebut dapat berjalan dengan baik atau tidak.

3.1.14. Aplikasi

Pengoptimalan dilakukan untuk meningkatkan performa dari aplikasi yang telah dirancang. Lalu optimasi ditekankan pada desain mekanik dan perangkat lunak agar penggunaan lebih maksimal dan tidak terjadi *error*.

3.2. Metode *K-Nearest Neighbours* (KNN)

Pada penelitian dilakukan proses klasifikasi terhadap kualitas kopi arabica. Penentuan klasifikasi kopi arabica dilakukan berdasarkan kekentalan dan keasamaan pada kopi arabica menggunakan metode KNN. Adapun Urutan Algoritma KNN sebagai berikut:

- 1) Menentukan parameter K (jumlah tetangga paling dekat)
- 2) Menghitung kuadrat jarak *euclidean* (*euclidean distance*) antara data baru terhadap data latih (*training data*). Adapun persamaan yang digunakan untuk menghitung jarak *euclidean distance* menggunakan persamaan (1) (Wang et al., 2020; Zhao et al., 2021).

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \dots\dots\dots(1)$$

Dimana d merupakan jarak data; pi adalah sampel data; qi adalah data uji atau data latih; i adalah variabel data; dan n adalah jumlah data.

- 3) Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *euclidean* terkecil
- 4) Mengumpulkan kategori Y (klasifikasi *nearest neighbors*)
- 5) Keluarkan hasil / voting berdasarkan nilai K yang telah ditentukan

3.3. Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan mulai Bulan Agustus 2022 sampai Oktober 2022. Waktu pelaksanaan dilaksanakan di Kampung Sindangbarang 03/02 Desa Pasireurih Kecamatan Tamansari Kabutapen Bogor.

3.4. Alat dan Bahan

3.4.1. Alat

Adapun alat yang akan digunakan adalah sebagai berikut:

- 1) Hardware
 - Laptop atau *Personal Computer* (PC): merk Laptop Asus A455L dengan spesifikasi intel core i5, RAM 4 Gb, VGA 2 Gb 2. MPU-6050
 - NodeMCU
 - Sensor PH
 - *Waterflow*
- 2) Software meliputi: *Visual Studio Code*, *Ms. Office*, *Visio*, *Sketchup*, *Arduino IDE*, dan *Chrome Browser*,

3.4.2. Bahan

Bahan-bahan yang akan digunakan pada penelitian didapatkan melalui studi pustaka yang berkaitan dengan perancangan dan implementasi sistem yang akan dibangun. Sumber informasi tersebut berupa buku, media cetak, media internet, serta Buku Panduan Skripsi dan Tugas Akhir.

BAB IV

RANCANGAN DAN IMPLEMENTASI

4.1. Perencanaan Penelitian

Dalam perencanaan penelitian ini merupakan tahapan perencanaan dalam pemilihan kebutuhan alat dan bahan yang akan digunakan dalam penelitian ini. Sensor yang digunakan adalah sensor pH dan sensor *waterflow*. Selain itu digunakan juga arduino dan *nodemcu*.

4.1.1. Analisa Kebutuhan *Hardware*

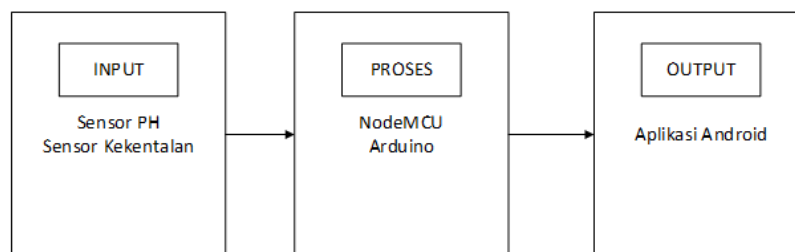
Pada tahapan ini dilakukan proses analisa kebutuhan perangkat keras (*hardware*) yang akan digunakan. *Hardware* pada penelitian ini menggunakan mikrokontroller arduino yang digunakan untuk menerima *input* sensor dan nodemcu yang digunakan untuk mengirim data sensor ke *database*. Selain itu, pada penelitian ini menggunakan sensor PH untuk mengetahui tingkat keasaman kopi yang diuji dan *waterflow* yang digunakan untuk mengetahui kekentalan dari kopi yang diuji.

4.1.2. Analisa Kebutuhan *Software*

Pada tahapan ini dilakukan proses analisa kebutuhan perangkat lunak (*software*) yang akan digunakan. *Software* pada penelitian ini menggunakan Arduino IDE yang digunakan untuk mempermudah pembuatan dan pengembangan sistem yang akan dibangun mulai dari menuliskan *source* program sampai upload hasil kompilasi, dan uji coba secara terminal serial. Disamping itu, pada penelitian ini juga menggunakan android studio yang digunakan untuk membuat aplikasi android.

4.2. Studi Referensi

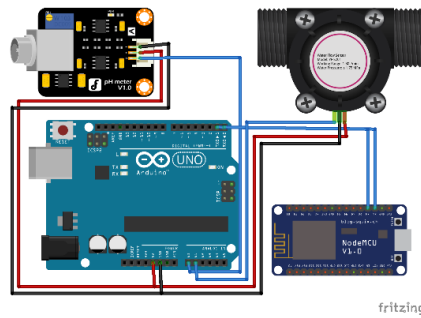
Setelah perencanaan telah matang dilanjutkan dengan penelitian awal dari aplikasi yang akan dibuat mulai dari pemilihan dan pengujian komponen (alat dan bahan) yang akan digunakan, kemungkinan rancangan awal dan akhir dalam menentukan kualitas kopi arabica berdasarkan kekentalan dan keasaman kopi arabica. Pada Gambar 9. ditampilkan diagram blok untuk sistem yang akan berjalan.



Gambar 9. Diagram Blok Sistem

4.3. Desain Elektrik

Dalam desain sistem listrik dan mekanis terdapat beberapa hal yang harus diperhatikan, antara lain: sumber catu daya dan pembagian daya untuk masing-masing komponen; kebutuhan tegangan dan arus untuk mikrokontroler, sensor dan aktuator; dan desain skema rangkaian. Adapun desain elektrik ditunjukkan pada Gambar 10.



Gambar 10. Diagram Blok Sistem

Berdasarkan Gambar 10., dapat diuraikan penjelasan sebagai berikut:

- 1) Sumber listrik (seperti baterai atau *power supply*).
Baterai atau *power supply* yang akan digunakan pada rangkaian ini sebesar 12V untuk sensor PH untuk mengetahui keasaman kopi dan *waterflow* untuk mengetahui kekentalan kopi, *NodeMCU* ESP8266 dan arduino uno.
- 2) Mikrokontroler
Mikrokontroler yang akan digunakan dalam perancangan desain listrik ini menggunakan *NodeMCU* esp8266 sebagai pusat pemrosesan input sinyal elektronik menjadi output sinyal elektronik yang dibutuhkan, serta dapat menghubungkan ke dalam jaringan wifi yang tersedia.
- 3) Desain sistem kontrol yang diterapkan dalam penelitian ini menggunakan sistem kontrol pemrograman bahasa pemrograman C untuk mengakusisi data serial dari rangkaian mikrokontroler.

4.4. Pengadaan Komponen

Pengadaan komponen adalah tahap persiapan pengumpulan komponen-komponen yang akan di pakai nantinya agar pada saat proses perakitan tidak terhenti karena kekurangan komponen. Komponen yang dipakai pada penelitian meliputi Arduino, *NodeMCU*, sensor PH dan *waterflow*. Setelah pengadaan komponen selesai lalu dilanjut ke proses pengujian komponen.

4.5. Pengujian Komponen

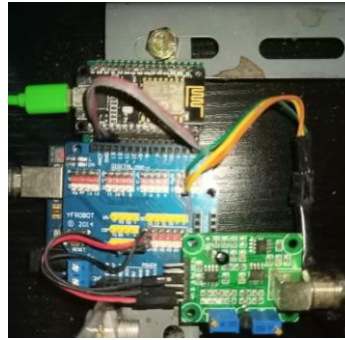
Dalam tahap pengujian komponen dilakukan pengujian alat terhadap fungsi kerja komponen berdasarkan kebutuhan sistem yang akan dibuat. Agar alat dapat berjalan sebagaimana semestinya. Pengujian komponen ini meliputi komponen yang dipakai pada pengujian ini. Pada Tabel 2. ditampilkan hasil pengujian komponen.

Tabel 2. Hasil Pengujian Komponen

Komponen	Tegangan <i>Input</i>	Tegangan <i>Output</i>
<i>NodeMCU</i> ESP8266	5 V	3,28 V
Arduino Uno	5 V	5,12 V
Sensor PH	5 V	4.89
<i>Waterflow</i>	5 V	4.73

4.6. Implementasi Elektrik

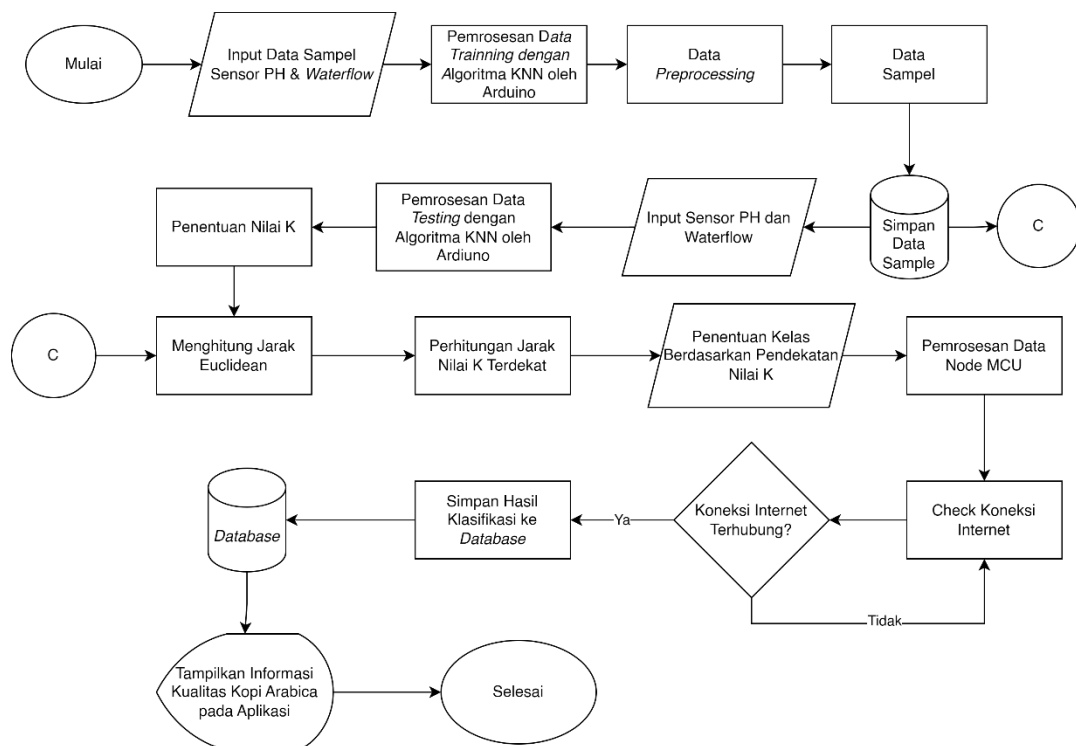
Implementasi elektrik adalah pengimplementasian dari gambaran rangkaian desain listrik yang telah dibuat sebelumnya. Implementasi elektrik dapat dilihat pada Gambar 11.



Gambar 11. Diagram Blok Sistem

4.7. Desain Software

Dalam desain perangkat lunak yang digunakan dalam penelitian ini menggunakan perangkat lunak *Ms. Office, Google Chrome, Visio 2019, Fritzing, Arduino IDE 1.8.10, Visual Studio Code* dan *Sketch Up* untuk bahasa pemrograman yang digunakan yaitu pemrograman bahasa C. Diagram alir sistem ditunjukkan pada Gambar 12.



Gambar 12. Diagram Alir Sistem

Berdasarkan Gambar 12. dapat diuraikan penjelasan sebagai berikut:

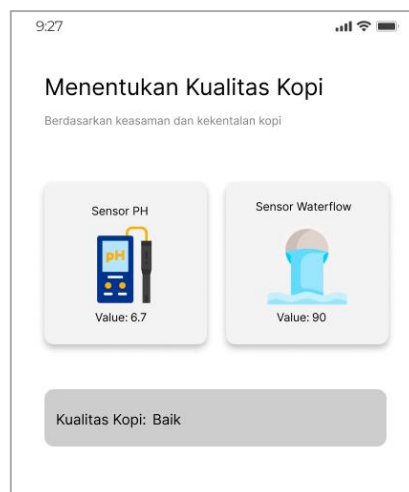
- 1) Sensor pH dan sensor *waterflow* membaca nilai dari objek yang di teliti
- 2) Nilai dari kedua sensor tersebut akan di olah sebagai data latih (*data training*) untuk algoritma KNN
- 3) Pemrosesan data dilakukan oleh Arduino menggunakan *library* KNN dan hasil dari data latih akan di tersimpan pada memori Arduino
- 4) *Input* dari sensor Ph dan sensor kekentalan selanjutnya adalah sebagai data uji

(*data testing*)

- 5) Sebelum melakukan klasifikasi, perlu dilakukan penetapan nilai K untuk algoritma KNN
- 6) Menghitung jarak Euclidean dengan menggunakan data latih yang tersimpan pada memori Arduino.
- 7) Penentuan jarak nilai K terdekat
- 8) Penentuan kelas berdasarkan jarak yang telah dilakukan pada proses mencari jarak terdekat dengan nilai K
- 9) Pemrosesan pengiriman data ke nodemcu
- 10) Pengecekan koneksi internet, jika ada maka data akan di simpan pada *database*
- 11) Hasil klasifikasi yang telah dilakukan oleh Arduino untuk menentukan kualitas kopi dapat dilihat melalui aplikasi.

4.8. Implementasi *Software*

Implementasi *software* adalah pengimplementasian dari gambaran desain *software* yang telah di buat sebelumnya. Pada tahapan implementasi *software* ini merupakan gambaran dari desain sistem yang sudah dibuat sebelumnya. Implementasi *software* ini dapat dilihat pada Gambar 13.



Gambar 13. Implementasi *Software*

4.9. Uji *Software*

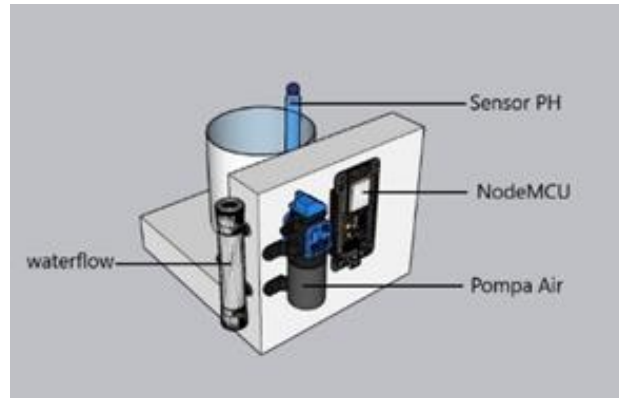
Pengujian *software* dilakukan agar desain yang telah dibuat sebelumnya sesuai dengan yang di inginkan sehingga pada saat penelitain bisa berfungsi dengan baik. Pengujian ini dilakukan dengan menjalankan fungsi yang ada pada aplikasi yang berjalan apakah sudah sesuai dengan fungsi yang sudah dirancang sebelumnya.

4.10. Desain Mekanik

Dalam perancangan perangkat keras, desain mekanik merupakan hal penting yang harus dipertimbangkan. Pada umumnya kebutuhan aplikasi terhadap desain mekanik antara lain :

- 1) Bentuk dan ukuran *Printed Circuit Board (PCB)*.
- 2) Ketahanan dan fleksibilitas terhadap lingkungan.
- 3) Penempatan modul-modul elektronik.
- 4) Pengujian sistem mekanik yang telah di rancang.

Desain mekanik pada penelitian ini ditunjukkan pada Gambar 14.



Gambar 14. Desain Mekanik

4.11. Implementasi Mekanik

Pada tahap ini dilakukan proses implementasi dari desain mekanik yang sudah dibuat sebelumnya. Setelah pengimplementasian mekanik selesai dilanjutkan ke tahap integrasi.

4.12. Integrasi

Modul listrik yang diintegrasikan dengan *software* di dalam kontrollernya, kemudian diintegrasikan dalam struktur mekanik yang telah dirancang menjadi satu keseluruhan dari alat yang dibangun. Pada perakitan model yang sudah dibuat memiliki panjang 25 cm, lebar 20 cm dan tinggi 15 cm. Pada Gambar 15. ditampilkan model perakitan alat.



Gambar 15. Model Perakitan Alat

4.13. Penerapan Metode KNN

Pada penelitian ini dilakukan proses klasifikasi kualitas dari kopi *arabica* berdasarkan kekentalan dan keasamaan kopi *arabica* menggunakan metode KNN. Adapun proses yang dilakukan untuk menentukan klasifikasi kualitas dari kopi arabica berdasarkan kekentalan dan keasamaan kopi *arabica* menggunakan metode KNN diuraikan sebagai berikut:

- 1) Menentukan parameter K (jumlah tetangga paling dekat)
 Pada penelitian dilakukan proses klasifikasi kualitas kopi *arabica* berdasarkan kekentalan dan keasaman kopi *arabica* menggunakan metode KNN. Pada tahapan ini terdapat data latih sebanyak sepuluh data dengan dua variabel yaitu keasaman (X1) dan kekentalan (X2) dengan dua kelas yaitu kurang baik dan baik yang ditunjukkan pada Tabel 3 serta satu data uji yang ditunjukkan pada Tabel 4.

Tabel 3. Data Latih Kopi *Arabica*

Data Latih ke	Variabel		Kelas (Y)
	Keasaman (X1)	Kekentalan (X2)	
-			
1	4.39	2.34	Kualitas kopi kurang baik
2	4.56	2.21	Kualitas kopi kurang baik
3	4.28	3.72	Kualitas kopi kurang baik
4	4.56	3.6	Kualitas kopi kurang baik
5	4.19	3.21	Kualitas kopi kurang baik
6	4.87	3.36	Kualitas kopi baik
7	4.88	3.48	Kualitas kopi baik
8	4.91	3.61	Kualitas kopi baik
9	4.43	2.97	Kualitas kopi baik
10	4.93	3.01	Kualitas kopi baik

Tabel 4. Data Uji Kopi *Arabica*

Data Uji	Variabel	
	Keasaman (X1)	Kekentalan (X2)
1	4.33	2.16

Tahap pertama yang dilakukan dalam proses klasifikasi kualitas kopi *arabica* adalah dengan menentukan parameter atau nilai K terlebih dahulu. Penentuan nilai K ini tidak menggunakan rumus yang pasti. Namun, ada beberapa hal yang perlu dipertimbangkan adalah jika kelas nya berjumlah genap maka sebaiknya nilai K-nya ganjil. Begitupun sebaliknya, jika kelasnya berjumlah ganjil maka nilai K-nya genap (Putri, 2021). Pada penelitian ini terdapat dua kelas yang merupakan bilangan genap sehingga nilai K yang ditentukannya adalah adalah bilangan ganjil. Nilai K yang ditentukan pada penelitian ini adalah 3.

- 2) Menghitung kuadrat jarak *euclidian (euclidean distance)* masing-masing obyek terhadap data sampel yang diberikan.
 Setelah ditentukan nilai K nya. Tahap selanjutnya dilakukan proses perhitungan jarak antara data baru terhadap semua data latih menggunakan persamaan (1). Adapun proses perhitungan jarak *euclidean distance* dilakukan sebagai berikut:

$$d(\text{data latih ke } - 1, \text{datauji}) = \sqrt{(4.39 - 4.33)^2 + (2.34 - 2.16)^2} \\ = 0.1897$$

$$d(\text{data latih ke } - 2, \text{datauji}) = \sqrt{(4.56 - 4.33)^2 + (2.21 - 2.16)^2} \\ = 0.2354$$

$$d(\text{data latih ke } - 3, \text{datauji}) = \sqrt{(4.28 - 4.33)^2 + (3.27 - 2.16)^2} \\ = 1.5608$$

$$d(\text{data latih ke } - 4, \text{datauji}) = \sqrt{(4.56 - 4.33)^2 + (3.6 - 2.16)^2} \\ = 1.4583$$

$$d(\text{data latih ke } - 5, \text{datauji}) = \sqrt{(4.19 - 4.33)^2 + (3.21 - 2.16)^2} \\ = 1.0593$$

$$d(\text{data latih ke } - 6, \text{datauji}) = \sqrt{(4.87 - 4.33)^2 + (3.36 - 2.16)^2} \\ = 1.3159$$

$$d(\text{data latih ke } - 7, \text{datauji}) = \sqrt{(4.88 - 4.33)^2 + (3.48 - 2.16)^2} \\ = 1.4300$$

$$d(\text{data latih ke } - 8, \text{datauji}) = \sqrt{(4.91 - 4.33)^2 + (3.61 - 2.16)^2} \\ = 1.5617$$

$$d(\text{data latih ke } - 9, \text{datauji}) = \sqrt{(4.43 - 4.33)^2 + (2.97 - 2.16)^2} \\ = 0.8161$$

$$d(\text{data latih ke } - 10, \text{datauji}) = \sqrt{(4.93 - 4.33)^2 + (3.01 - 2.16)^2} \\ = 1.0404$$

Berdasarkan proses perhitungan jarak *euclidean distance* maka didapatkan hasil perhitungan yang ditampilkan pada Tabel 5.

Tabel 5. Hasil Perhitungan Jarak *Euclidean Distance*

Data Ke -	Variabel		Hasil Perhitungan
	(X1)	(X2)	
1	4.39	2.34	0.1897
2	4.56	2.21	0.2354
3	4.28	3.72	1.5608
4	4.56	3.6	1.4583
5	4.19	3.21	1.0593
6	4.87	3.36	1.3159
7	4.88	3.48	1.4300
8	4.91	3.61	1.5617
9	4.43	2.97	0.8161
10	4.93	3.01	1.0404

- 3) Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *euclidean* terkecil.

Tahapan selanjutnya dilakukan proses pengurutan data kedalam kelompok berdasarkan jarak *euclidean* terkecil. Hasil pengurutan data ditunjukkan pada

Tabel 6.

Tabel 6. Urutan berdasarkan Jarak *Euclidean Distance* Terkecil

Data Ke -	Variabel		Hasil Perhitungan	Urutan
	(X1)	(X2)		
1	4.39	2.34	0.1897	1
2	4.56	2.21	0.2354	2
3	4.28	3.72	1.5608	9
4	4.56	3.6	1.4583	8
5	4.19	3.21	1.0593	5
6	4.87	3.36	1.3159	6
7	4.88	3.48	1.4300	7
8	4.91	3.61	1.5617	10
9	4.43	2.97	0.8161	3
10	4.93	3.01	1.0404	4

Berdasarkan Tabel 6. Dapat diketahui bahwa data ke -1 memiliki jarak *Euclidean* terdekat atau terkecil dengan data uji.

- 4) Mengumpulkan kategori Y (klasifikasi *nearest neighbors*).

Pada Tabel 7. ditunjukkan data latih beserta hasil perhitungan jarak *Euclidean* serta urutannya berdasarkan jarak *euclidean* terdekat atau terkecil dengan data uji.

Tabel 7. Data Latih dengan Hasil Perhitungan Jarak *Euclidean*

Data Ke -	Variabel		Hasil Perhitungan	Urutan	Kelas (Y)
	(X1)	(X2)			
1	4.39	2.34	0.1897	1	Kualitas kopi kurang baik
2	4.56	2.21	0.2354	2	Kualitas kopi kurang baik
3	4.28	3.72	1.5608	9	Kualitas kopi kurang baik
4	4.56	3.6	1.4583	8	Kualitas kopi kurang baik
5	4.19	3.21	1.0593	5	Kualitas kopi kurang baik
6	4.87	3.36	1.3159	6	Kualitas kopi baik
7	4.88	3.48	1.4300	7	Kualitas kopi baik
8	4.91	3.61	1.5617	10	Kualitas kopi baik
9	4.43	2.97	0.8161	3	Kualitas kopi baik
10	4.93	3.01	1.0404	4	Kualitas kopi baik

- 5) Keluarkan hasil atau voting berdasarkan nilai K yang telah ditentukan.

Pada tahap terakhir yaitu melakukan voting berdasarkan nilai K. Pada penelitian ini nilai K = 3. Sehingga untuk melakukan voting hanya mempertimbangan data latih dari urutan 1 sampai urutan 3 saja. Kemudian lakukan voting nilai kelas (Y) berdasarkan mayoritas kelas pada ketiga data tersebut. Pada Tabel 8. ditunjukkan 3 data hasil voting berdasarkan nilai K

yang ditentukan. Dari Tabel 8. Dapat disimpulkan bahwa Adapun data hasil voting nilai kelas (Y) ditunjukkan pada Tabel 8. mayoritas kelas pada ketika data tersebut adalah kualitas kopi kurang baik. Maka dapat diketahui bahwa data uji termasuk kedalam kelas dengan kualitas kopi kurang baik. Adapun hasil proses klasifikasi KNN terhadap data uji ditunjukkan pada Tabel 9.

Tabel 8. Hasil Voting Data Berdasarkan Nilai K

Data Ke -	Variabel		Hasil Perhitungan	Urutan	Kelas (Y)
	(X1)	(X2)			
1	4.39	2.34	0.1897	1	Kualitas kopi kurang baik
2	4.56	2.21	0.2354	2	Kualitas kopi kurang baik
9	4.43	2.97	0.8161	3	Kualitas kopi baik

Tabel 9. Hasil Klasifikasi Data Uji dengan KNN

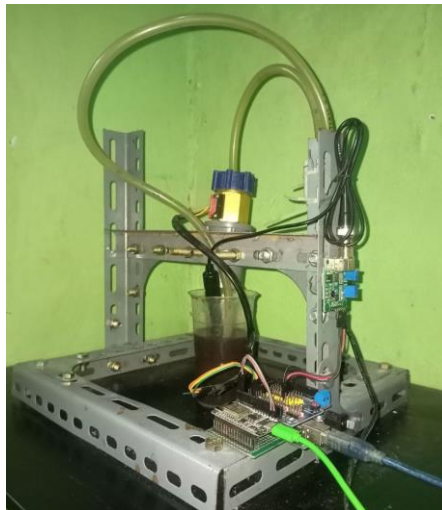
Data Uji	Variabel		Kelas (Y)
	Keasaman (X1)	Kekentalan (X2)	
1	4.33	2.16	Kualitas kopi kurang baik

BAB V

HASIL DAN PEMBAHASAN

5.1. Hasil Penelitian

Hasil penelitian ini merupakan hasil dari tahapan sebelumnya telah dijelaskan dimulai dari proses perencanaan, perancangan, hingga implementasi keseluruhan dari menentukan kualitas kopi *arabica* berdasarkan kekentalan dan keasaman kopi *arabica*. Hasil penelitian ini menyelesaikan beberapa hal yang menjadi acuan dan referensi agar mendapatkan hasil yang maksimal. Pada penelitian ini menggunakan sensor ph untuk mengetahui tingkat keasaman pada kopi yang diuji serta menggunakan *waterflow* untuk mengetahui kekentalan pada kopi yang diuji. Pada penelitian ini kopi yang digunakan dibuat menggunakan teknik manual brew dengan menggunakan alat moka pot dengan memanfaatkan tekanan air untuk mengekstrak kopi yang akan digunakan dalam pengujian ini. Penelitian ini juga menggunakan mikrokontroler Arduino untuk menerima input sensor dan NodeMCU untuk mengirim data sensor ke *database*. Selain itu, untuk menentukan kualitas pada sampel kopi yang diuji pada penelitian ini menggunakan algoritma KNN yang diterapkan pada arduino uno. Alat untuk menentukan kualitas kopi ditunjukkan pada Gambar 16.



Gambar 16. Model Perakitan Alat

5.2. Pengujian Fungsional Keseluruhan Sistem (*Overall Sistem*)

Tahapan ini dilakukan pengujian seluruh fungsi sistem, mulai dari pengujian *hardware*, program, dan *user interface* aplikasi yang dibuat. Jika, terdapat sistem yang tidak bisa berfungsi sebagaimana mestinya maka akan dilakukan proses implementasi mekanik pada sistem.

5.2.1. Pengujian Struktural

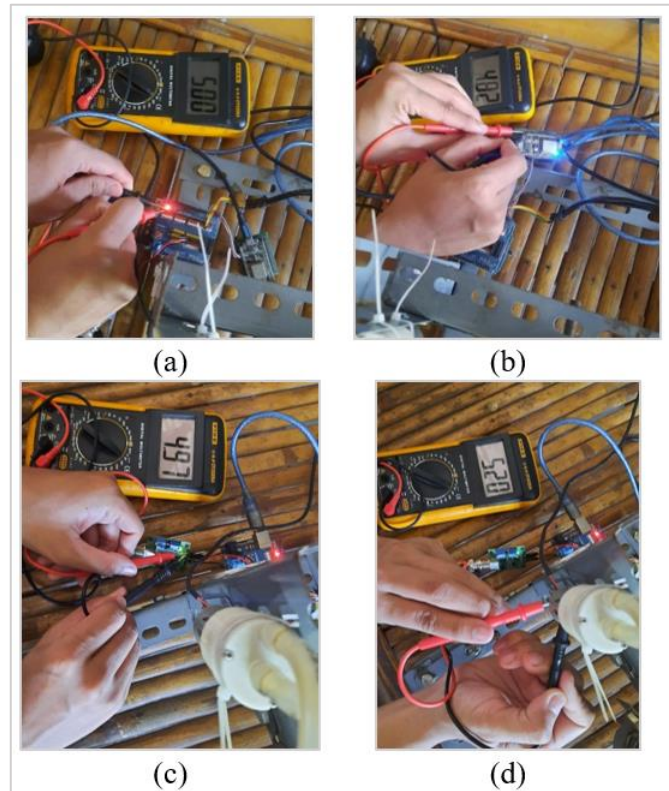
Pada tahapan ini dilakukan pengujian yang bertujuan untuk mengetahui apakah jalur sirkuit sudah terhubung dengan benar sehingga sistem dapat berfungsi dengan baik. Pengujian ini dilakukan dengan mencoba semua jalur-jalur sirkuit dengan menggunakan multimeter. Hasil pengujian struktural ditunjukkan pada Tabel 10.

Tabel 10. Hasil Pengujian Struktural

No	Komponen Sistem	Terhubung Pada Pin Arduino	Keterangan
1	Sensor Ph	Pin A0	Terhubung
2	Sensor <i>Waterflow</i>	Pin 2	Terhubung
3	Nodemcu Esp8266	Pin Rx, pin Tx	Terhubung

5.2.2. Pengujian Fungsional

Pada penelitian ini dilakukan proses pengujian fungsional dengan tujuan untuk mengetahui apakah setiap komponen dapat bekerja dengan baik. Beberapa komponen yang di uji yaitu tegangan Arduino uno, Nodemcu esp8266, sensor pH dan sensor *waterflow*. Pada proses pengujian ini menggunakan multimeter untuk mengetahui berapa tegangan yang diterima oleh komponen. Pada pengujian Arduino Uno dilakukan dengan cara mengukur tegangan pada pin Vcc dan Gnd. Pada Nodemcu esp8266 dilakukan pengujian untuk mengetahui apakah mikrokontroller dapat menerima tegangan sesuai dengan *supply* yang diberikan atau tidak. Tegangan yang diberikan sebesar 5V dengan cara menghubungkannya ke pin Vin dan Gnd pada *board*. Pengujian juga dilakukan terhadap sensor pH dengan memberikan tegangan 5V dengan cara menghubungkan kutub positif (+) dan kutub negatif (-) pada multimeter dengan pin Vcc dan pin Gnd. Selain itu, pengujian juga dilakukan terhadap sensor *waterflow* dengan memberikan tegangan 12V dengan cara menghubungkan kutub positif (+) dan kutub negatif (-) pada multimeter dengan pin Vcc dan pin Gnd. Proses pengujian fungsional pada komponen ditunjukkan pada Gambar 17. Sedangkan, untuk hasil pengujian fungsional pada komponen ditunjukkan pada Tabel 11.



Gambar 17. Proses Pengujian Fungsional: Arduino Uno (a); NodeMCU Esp8266 (b); Sensor pH (c); Sensor *Waterflow* (d)

Tabel 11. Hasil Pengujian Fungsional pada Komponen

No	Komponen	Tegangan <i>Input</i> (volt)	Tegangan <i>Output</i> (volt)	Kondisi
1	Arduino Uno	5	5	Baik
		5	4.98	Baik
		5	4.96	Baik
2	NodeMCU Esp8266	5	4.82	Baik
		5	4.76	Baik
		5	4.81	Baik
3	Sensor pH	5	4.97	Baik
		5	5	Baik
		5	4.95	Baik
4	Sensor <i>Waterflow</i>	12	5.20	Baik
		12	8.92	Baik
		12	9.20	Baik

5.2.3. Pengujian Keseluruhan Sistem

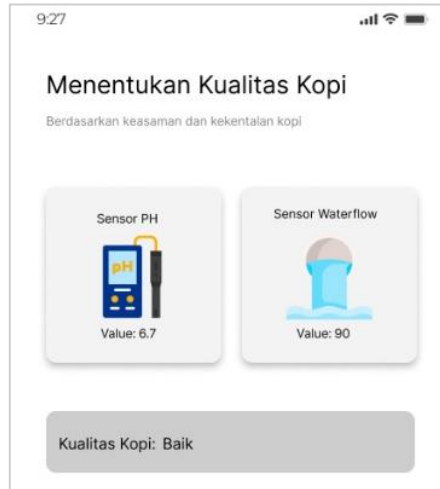
Setelah beberapa rangkaian pengujian yang telah dilakukan pada setiap komponen yang ada, maka tahap selanjutnya yang dilakukan pengujian keseluruhan pada sistem yang dibuat. Tahap pertama yang dilakukan merangkai semua komponen, selanjutnya meng-upload program ke dalam *chip* Arduino Uno. Adapun beberapa pengujian yang dilakukan pada sistem keseluruhan antara lain:

1. Pengujian pengecekan dari alat pada serial monitor pada Arduino IDE, pengecekan terhadap konektivitas dari ESP8266, apakah terkoneksi dengan jaringan sekitar sehingga mendapatkan IP dinamis yang kemudian dapat dipanggil untuk melakukan koneksi dengan *interface monitoring*.
2. Pengujian model alat untuk menentukan kualitas kopi *arabica* berdasarkan kekentalan dan keasaman kopi *arabica* dilakukan pada saat alat pertama kali diaktifkan dan semua komponen yang digunakan telah berfungsi sesuai yang dibutuhkan. Pada pengujian ini dilakukan pengujian yang meliputi menguji fungsi kerja sensor ph untuk mendapatkan data keasaman dan *waterflow* untuk mendapatkan data kekentalan pada sampel kopi yang diuji. Pengujian ini dapat dilihat pada Gambar 18.



Gambar 18. Pengujian Model Alat

3. Pada penelitian ini juga dibuatkan aplikasi untuk memudahkan user dalam menentukan kualitas kopi. Pada halaman utama aplikasi akan menampilkan nilai sensor pH dan nilai *waterflow* serta menampilkan hasil klasifikasi kualitas kopi dengan algoritma knn pada arduino. Halaman utama aplikasi dapat dilihat pada Gambar 19.



Gambar 19. Halaman Utama Aplikasi

5.2.4. Uji Validasi

Pada tahapan ini dilakukan dengan menguji alat secara keseluruhan apakah sudah sesuai dengan rancangan yang telah dibuat sebelumnya. Pengujian mengenai rancang bangun alat untuk menentukan kualitas kopi *arabica* berdasarkan kekentalan dan keasaman kopi *arabica* dengan melakukan uji coba pada sampel kopi yang digunakan.

5.2.4.1. Uji Validasi Sensor PH

Uji validasi sensor ph dilakukan untuk mengukur tingkat keasaman pada kopi arabica. Pengujian dilakukan dengan cara memasukan sensor Ph ke dalam cangkir kopi, sensor akan mendeteksi tingkat keasaman dari kopi dan hasil nilai akan di proses oleh Arduino untuk di jadikan variabel algoritma KNN. Kopi Arabika memiliki ph dengan nilai sekitar 4,85 – 5,15 (Kayaputri et al., 2022).

Tabel 12. Hasil Uji Validasi Sensor PH

No	Sensor Ph	Keterangan
1	4.82	Tidak Baik
2	4.98	Baik
3	4.44	Tidak Baik
4	4.89	Baik
5	5.05	Baik
6	4.90	Baik
7	4.93	Baik
8	5.00	Baik
9	5.30	Tidak Baik
10	5.24	Tidak Baik

Berdasarkan hasil pengujian pada Tabel 12. Dapat diketahui bahwa kopi *arabica* yang digunakan pada penelitian ini memiliki tingkat keasaman atau pH yang baik. Dimana hasil sensor pH dengan indikator **Baik** memiliki jumlah yang lebih banyak di bandingkan variabel **Tidak Baik**.

5.2.4.2. Uji Validasi Sensor *Waterflow*

Uji validasi sensor *waterflow* dilakukan untuk mengukur tingkat kekentalan pada kopi arabica. Pengujian dilakukan dengan cara mengalirkan kopi melewati sensor *waterflow* untuk diketahui kekentalan dari kopi yang di ujikan. Berdasarkan referensi, kekentalan kopi *arabica* yang sesuai adalah 3.61 (Nurhayati, 2018).

Tabel 13. Hasil Uji Validasi Sensor *Waterflow*

No	Sensor <i>Waterflow</i>	Keterangan
1	4.28	Tidak Sesuai
2	4.31	Tidak Sesuai
3	3.61	Sesuai
4	3.61	Sesuai
5	3.61	Sesuai
6	3.61	Sesuai
7	3.61	Sesuai
8	3.61	Sesuai
9	3.61	Sesuai
10	3.61	Sesuai

Berdasarkan hasil pengujian pada Tabel 13. dapat diketahui bahwa kopi *arabica* yang digunakan pada penelitian ini memiliki kekentalan yang sesuai dengan nilai yang diujikan.

5.2.4.3. Uji Validasi Kualitas Kopi

Pada tahapan ini dilakukan dengan menguji alat secara keseluruhan apakah sudah sesuai dengan rancangan yang telah dibuat sebelumnya. Pengujian mengenai rancang bangun alat untuk menentukan kualitas kopi arabica berdasarkan kekentalan dan keasaman kopi arabica dengan melakukan uji coba pada sampel kopi yang digunakan. Pada penelitian ini menggunakan sensor pH untuk mengetahui tingkat keasaman pada kopi dan menggunakan *waterflow* untuk mengetahui nilai kekentalan pada kopi yang diuji.

Mikrokontroler yang digunakan pada penelitian ini menggunakan arduino uno untuk pengolahan data kopi yang diuji dan nodemcu untuk mengirim data ke *database*. Untuk memudahkan dalam menentukan kualitas kopi pada penelitian ini menggunakan bantuan algoritma KNN pada arduino. Penerapan algoritma KNN ini dimulai pada saat alat dinyalakan untuk mendapatkan dataset yang tersimpan pada memori arduino. Dataset yang tersimpan pada memori arduino berasal dari sampel kopi yang baik dan sampel kopi yang kurang baik. Proses pengambilan dataset pada KNN arduino untuk dataset sampel kopi dapat dilihat pada Gambar 20.

```

// Ask user for the name of each object
for (int currentClass = 0; currentClass < CLASSES; currentClass++) {
  // Ask user to show examples of each object

  if (currentClass == 1) {
    for (int currentExample = 0; currentExample < EXAMPLES_PER_CLASS; currentExample++) {
      Serial.print("Show me an example ");
      Serial.println(label[currentClass]);

      sensor3(kopi, currentExample);
      // Add example color to the k-MN model
      myKNN.addExample(kopi, currentClass);

      // if (currentExample >= 29) {
      Serial.println("FINISH");
      delay (200);
      // }
    }
  }
}

```

Gambar 20. Proses Pengambilan Dataset Kopi pada Algoritma KNN Arduino

Setelah dataset yang didapatkan dari memori arduino maka selanjutnya user dapat menguji sampel kopi yang ingin diuji apakah sudah sesuai dengan kelas yang tersimpan pada knn arduino. Pada pengujian ini kelas KNN dibagi menjadi dua yaitu kelas untuk kelas kopi kualitas baik dan kopi kualitas kurang baik sesuai dengan dataset yang tersimpan pada memori arduino. Pada penelitian ini sampel kopi yang digunakan dalam pengujian ini diekstrak menggunakan metode manual brew menggunakan alat moka pot. Proses ekstraksi kopi menggunakan moka pot dapat dilihat pada Gambar 21.



Gambar 21. Ekstraksi Kopi menggunakan Moka pot

Setelah proses ekstraksi sampel kopi menggunakan moka pot sudah selesai maka sampel kopi dapat digunakan untuk pengujian. Proses pengujian sampel kopi menggunakan knn arduino dapat dilihat pada Gambar 22.

```

sensor(kopi);
// Wait for the object to move away again
Serial.println("Let me guess your object");

// Classify the object
classification = myKNN.classify(kopi, K);

// Print the classification
// Serial.print("You showed me ");
// Serial.println(label[classification]);

if (label[classification] == "Kopi lulus uji") {
  Serial.print("You showed me ");
  Serial.println(label[classification]);
}
else {
  Serial.println("Kopi dibawah standar");
}
}

```

Gambar 22. Proses Pengujian Sampel Kopi dengan KNN

Berdasarkan Gambar 22. arduino akan mengambil data sensor pH untuk tingkat keasaman dan *waterflow* untuk nilai kekentalan yang akan diklasifikasikan berdasarkan kelas pada algoritma KNN pada arduino. Proses klasifikasi kualitas kopi ini menggunakan KNN arduino akan menampilkan label kelas berdasarkan pada kelas KNN yang ada apakah kopi kualitas baik atau kopi kualitas kurang baik. Hasil pengujian sampel kopi ini akan disimpan oleh nodemcu ke *database* dan ditampilkan pada halaman utama aplikasi untuk melihat hasil kualitas kopi yang di uji. Sehingga user dapat mengetahui kopi yang di uji memiliki kualitas yang baik atau kurang baik untuk mendapatkan kualitas kopi yang terbaik di pasaran. Adapun hasil klasifikasi menggunakan KNN pada Arduino ditunjukkan pada Tabel 14.

Tabel 14. Uji Validasi Kualitas Kopi

Pengujian Ke -	Kopi (gram)	Air (ml)	Keasaman	Kekentalan	Status
1	1	100	4.39	2.34	Kualitas kopi kurang baik
2	1	135	4.56	2.21	Kualitas kopi kurang baik
3	1	175	4.28	3.72	Kualitas kopi kurang baik
4	1	200	4.56	3.6	Kualitas kopi kurang baik
5	1	235	4.19	3.61	Kualitas kopi kurang baik
6	1	275	4.87	3.61	Kualitas kopi baik
7	1	300	4.88	3.61	Kualitas kopi baik
8	1	335	4.9	3.61	Kualitas kopi baik
9	1	375	4.9	3.61	Kualitas kopi baik
10	1	400	4.9	3.61	Kualitas kopi baik
11	2	100	5.12	3.61	Kualitas kopi baik
12	2	135	4.67	3.42	Kualitas kopi kurang baik
13	2	175	4.22	3.56	Kualitas kopi kurang baik
14	2	200	4.1	3.81	Kualitas kopi kurang baik
15	2	235	5.52	3.61	Kualitas kopi baik
16	2	275	4.94	3.61	Kualitas kopi baik
17	2	300	4.84	3.61	Kualitas kopi baik
18	2	335	4.86	3.61	Kualitas kopi baik
19	2	375	4.93	3.61	Kualitas kopi baik
20	2	400	4.88	3.61	Kualitas kopi baik
Rata-Rata			4.82	3.61	

Catatan : 1 sendok makan (sdm) kopi = 15 gram

Berdasarkan hasil pengujian yang telah dilakukan, pada Tabel 14. menunjukan dominan kopi dengan kualitas baik sebanyak 12 data sedangkan untuk kopi dengan kualitas kurang baik sebanyak 8. Dengan ini kopi jenis arabica yang digunakan pada penelitian ini mempunyai kualitas yang baik dengan rata-rata nilai pH atau keasaman 4.82 dan kekentalan sebesar 3.61.

5.2.4.4. Uji Validasi Kualitas Kopi dengan Minuman Lain

Pada pengujian validasi ini dilakukan beberapa pengujian untuk mengetahui kadar keasaman dan kekentalan pada kopi arabica dan minuman lain. Proses penelitian ini dimulai dari mendeteksi kadar keasaman dan kekentalan pada kopi dan minuman lain dengan menggunakan sensor pH dan sensor waterflow kemudian nilai sensor akan diolah menggunakan mikrokontroler Arduino dengan algoritma KNN. *Output* dari hasil pengolahan tersebut akan di kirimkan ke Nodemcu esp8266 dan ditampilkan dalam aplikasi. Tujuan dari pengujian ini adalah untuk mengetahui perbandingan antara nilai keasaman dan kekentalan. Adapun hasil pengujian ditunjukkan pada Tabel 15.

Tabel 15. Uji Validasi Kualitas Kopi dengan Minuman Lain

Kopi Arabica			
Pengujian Ke -	Keasaman	Kekentalan	Status
1	4.39	2.34	Kualitas kopi kurang baik
2	4.56	2.21	Kualitas kopi kurang baik
3	4.28	3.72	Kualitas kopi kurang baik
4	4.56	3.6	Kualitas kopi kurang baik
5	4.19	3.61	Kualitas kopi kurang baik
6	4.87	3.61	Kualitas kopi baik
7	4.88	3.61	Kualitas kopi baik
8	4.9	3.61	Kualitas kopi baik
9	4.9	3.61	Kualitas kopi baik
10	4.9	3.61	Kualitas kopi baik
Kopi Nescafe Robusta			
Pengujian Ke -	Keasaman	Kekentalan	Status
1	3.65	3.6	Kualitas kopi baik
2	3.21	3.6	Kualitas kopi baik
3	3.68	3.62	Kualitas kopi baik
4	4.36	3.64	Kualitas kopi baik
5	4.66	3.55	Kualitas kopi kurang baik
6	4.78	3.74	Kualitas kopi kurang baik
7	4.51	4.2	Kualitas kopi kurang baik
8	4.23	4.21	Kualitas kopi kurang baik
9	4.1	4.11	Kualitas kopi kurang baik
10	4	3.51	Kualitas kopi kurang baik
Pocari Sweat			

Pengujian Ke -	Keasaman	Kekentalan	Status
1	4.65	2.1	Kualitas kopi kurang baik
2	4.88	2.3	Kualitas kopi kurang baik
3	4.36	2.36	Kualitas kopi kurang baik
4	3.89	2.41	Kualitas kopi kurang baik
5	4.21	2.33	Kualitas kopi kurang baik
6	4.11	2.27	Kualitas kopi kurang baik
7	4.56	2.2	Kualitas kopi kurang baik
8	4.66	2.19	Kualitas kopi kurang baik
9	4.6	2.23	Kualitas kopi kurang baik
10	4.6	2.18	Kualitas kopi kurang baik

Berdasarkan Tabel 15. Dapat disimpulkan bahwa, hasil dari pengujian yang telah dilakukan menunjukkan bahwa kopi arabica dengan kopi nescafe robusta memiliki hasil yang cukup mendekati dengan nilai parameter dari masing-masing sensor. Sebaliknya, pada minuman pocari sweat di dapatkan hasil dengan kualitas yang kurang baik, hal ini dikarenakan kekentalan minuman ini berada jauh dari parameter untuk kekentalan, yaitu 3.61.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan mengenai rancang bangun alat untuk menentukan kualitas kopi arabica berdasarkan kekentalan dan keasaman kopi arabica sudah sesuai dengan perancangan alat yang dibuat sebelumnya. Pada penelitian ini menggunakan sensor pH untuk mengetahui tingkat keasaman pada sampel kopi dan menggunakan *waterflow* untuk mengetahui nilai kekentalan pada sampel kopi yang diuji. Untuk mikrokontroler menggunakan arduino untuk mengolah input sensor dan nodemcu untuk menyimpan data sensor ke *database*. Pada penelitian ini sampel kopi yang digunakan untuk pengujian menggunakan metode manual brew menggunakan alat mokapot untuk mengekstrak sampel kopi yang digunakan. Untuk mempermudah mengetahui kualitas kopi pada sampel yang diuji pada penelitian ini menggunakan algoritma knn pada arduino. Penggunaan algoritma KNN pada arduino ini mulai dari pengambilan data sensor ph dan *waterflow* pada memori arduino untuk digunakan sebagai dataset pada algoritma knn.

Setelah data sensor yang tersimpan pada memori arduino selesai digunakan untuk dataset pada algoritma knn maka tahapan selanjutnya adapalah pengujian sampel kopi yang akan digunakan pada penelitian ini. Pengujian ini menggunakan sampel kopi yang sudah dibuat menggunakan mokapot. Pada saat pengujian sampel kopi dilakukan arduino akan mengambil nilai sensor ph untuk menentukan tingkat keasaman dan *waterflow* untuk nilai kekentalan. Dari data sensor yang diambil maka nilai sensor tersebut akan diklasifikasikan oleh arduino menggunakan algoritma knn berdasarkan kelas pada algoritma knn. Hasil pengujian sampel kopi ini akan disimpan oleh nodemcu ke *database* dan ditampilkan pada halaman utama aplikasi untuk melihat hasil kualitas kopi yang diuji. Sehingga user dapat mengetahui kopi yang diuji memiliki kualitas yang baik atau kurang baik untuk mendapatkan kualitas kopi yang terbaik di pasaran.

Hasil pengujian menunjukkan kopi *arabica* yang digunakan dalam penelitian ini memiliki kualitas kopi yang baik, baik dalam keasaman maupun kekentalan. Dominan dengan status kopi dengan kualitas baik sebanyak 12 data sedangkan untuk kopi dengan kualitas kurang baik sebanyak 8. Dengan ini kopi jenis *arabica* yang digunakan pada penelitian ini mempunyai kualitas yang baik dengan rata-rata nilai pH 4.82 dan kekentalan sebesar 3.61. Pengujian untuk membandingkan nilai keasaman dan kekentalan antara kopi *arabica* dan minuman lain nya menghasilkan hasil akhir untuk kopi *arabica* dan kopi nescafe *robusta* mendapatkan hasil yang cukup mendekati dengan nilai parameter keasaman dan kekentalan. Sedangkan untuk minuman pocari sweat mendapatkan hasil yang sebaliknya.

6.2. Saran

Berdasarkan penelitian yang sudah dilakukan mengenai rancang bangun alat untuk menentukan kualitas kopi *arabica* berdasarkan kekentalan dan keasaman kopi *arabica* terdapat beberapa saran untuk pengembangan penelitian berikutnya seperti penambahan parameter untuk pengecekan kualitas kopi berdasarkan bubuk kopi untuk mempermudah pengecekan kualitas kopi yang banyak dijual dipasaran.

DAFTAR PUSTAKA

- Admojo, F. T., & Ahsanawati. (2020). Klasifikasi Aroma Alkohol Menggunakan Metode KNN. *Indonesian Journal of Data and Science*, 1(2), 34–38. <https://doi.org/10.33096/ijodas.v1i2.12>
- Aprianto, Z. (2020). *Rancang bangun sistem monitoring pH dan suhu pada larutan kopi berbasis fuzzy logic di PT HARUM ALAM SEGAR* [Universitas Muhammadiyah Gresik]. <http://eprints.umg.ac.id/id/eprint/4471>
- Ayu, M. G. (2020). *Perkembangan dan Penggunaan IoT di Indonesia Tahun 2021 Diprediksi Meningkat.* cloudcomputing. <https://www.cloudcomputing.id/berita/perkembangan-dan-penggunaan-iot-di-indonesia>
- Cahyanti, D., Rahmayani, A., & Husniar, S. A. (2020). Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara. *Indonesian Journal of Data and Science*, 1(2), 39–43. <https://doi.org/10.33096/ijodas.v1i2.13>
- Chairunnas, A., & Pamungka, T. G. (2019). Sistem Kontrol Robot Penyeimbang Berbasis Arduino Menggunakan Metode Pid Dengan Komunikasi Bluetooth Hc-05. *Komputasi: Jurnal Ilmiah Ilmu Komputer dan Matematika*, 15(2), 140–151. <https://doi.org/10.33751/komputasi.v15i2.1380>
- Farhan. (2022). *Tutorial Akses Data Sensor Water Flow Menggunakan Arduino Uno.* Indomaker. <http://indomaker.com/product/blog/tutorial-akses-data-sensor-water-flow-menggunakan-arduino-uno/>
- Firdayani, N., & Aisyah, Y. (2020). 4 Beda Kopi Robusta dengan Arabika, Daerah Tumbuh sampai Pengolahan. *kompas*. <https://www.kompas.com/food/read/2020/09/19/143029375/4-beda-kopi-robusta-dengan-arabika-daerah-tumbuh-sampai-pengolahan>
- Haidar, K. (2020). *Jenis-jenis kopi.* wixsite. <https://acupofcoffeeby.wixsite.com/posangkopi/post/jenis-jenis-kopi>
- Hanifah, D., Andarwulan, N., & Herawati, D. (2022). Karakteristik Fisikokimia dan Kapasitas Antioksidan Kopi Liberika dari Kabupaten Tanjung Jabung Barat, Jambi. *Jurnal Teknologi dan Industri Pangan*, 33(1), 39–51. <https://doi.org/10.6066/jtip.2022.33.1.39>
- Herman, H. A., & Chairunnas, A. (2019). Model Robot Troli Object Follower Menggunakan Pixy Cmcum5 Berbasis Arduino Uno 328p. *Komputasi: Jurnal Ilmiah Ilmu Komputer dan Matematika*, 16(2), 263–270. <https://doi.org/10.33751/komputasi.v16i2.1620>
- Hidayati, D. C., Al Faraby, S., & Adiwijaya, A. (2020). Klasifikasi Topik Multi Label pada Hadis Shahih Bukhari Menggunakan K-Nearest Neighbor dan Latent Semantic Analysis. *JURIKOM (Jurnal Riset Komputer)*, 7(1), 140. <https://doi.org/10.30865/jurikom.v7i1.2013>
- Ilmu, N. (2019). *Tutorial Mengakses Module ph meter sensor menggunakan Arduino.*

- Nyebar Ilmu. <https://www.nyebarilmu.com/tutorial-mengakses-module-ph-meter-sensor-menggunakan-arduino/>
- Ismangil, A., & Ardyahadistia, F. (2021). Model Pembangkit Listrik Elemen Ganda Dengan Panel Surya Dan Turbin Berbasis Internet of Things. *Komputasi: Jurnal Ilmiah Ilmu Komputer dan Matematika*, 18(2), 86–96. <https://doi.org/http://dx.doi.org/10.33751/komputasi.v18i2.3442>
- Kayaputri, I. L., Amalia, R. I., & Khairunnisa, F. (2022). Pemanfaatan Kopi Arabika (*Coffea arabica*) dalam Pembuatan Minuman Yoghurt sebagai Pangan Fungsional. *Jurnal Teknologi Hasil Peternakan*, 3(2), 49. <https://doi.org/10.24198/jthp.v3i2.40545>
- Mufida, E., Anwar, R. S., Khodir, R. A., & Rosmawati, I. P. (2020). Perancangan Alat Pengontrol pH Air Untuk Tanaman Hidroponik Berbasis Arduino Uno. *INSANtek – Jurnal Inovasi dan Sains Teknik Elektro*, 1(1), 13–10.
- Mutia Annur, C. (2022). Produksi Kopi Indonesia Naik Jadi 774,60 Ribu Ton pada 2021. *databoks.katadata.co.id*. <https://databoks.katadata.co.id/datapublish/2022/03/09/produksi-kopi-indonesia-naik-jadi-77460-ribu-ton-pada-2021>
- Nopriawan, R. (2018). *Prototype Alat Pengendali Dan Monitoring Tanaman Sebagai Pengembangan Smart Farming Berbasis Internet Of Things (IoT)*. Universitas Teknologi Yogyakarta.
- Nurhayati, N. (2018). Karakteristik Sensori Kopi Celup Dan Kopi Instan Varietas Robusta Dan Arabika. *Jurnal Ilmiah Inovasi*, 17(2). <https://doi.org/10.25047/jii.v17i2.547>
- Nurhidayati, N., & Nur, A. M. (2021). Pemanfaatan Aplikasi Android Dalam Rancang Bangun Sistem Informasi Persebaran Indeks di Wilayah Pancor Kabupaten Lombok Timur. *Infotek : Jurnal Informatika dan Teknologi*, 4(1), 51–62. <https://doi.org/10.29408/jit.v4i1.2989>
- Puspita, R., & Widodo, A. (2021). Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS. *Jurnal Informatika Universitas Pamulang*, 5(4), 646. <https://doi.org/10.32493/informatika.v5i4.7622>
- Putra, D. F. A., & Stefanus, S. (2019). Kajian Literatur –Penggunaan Sensor Waterflow pada Proses Pencampuran Cairan Dalam Industri. *Ultima Computing : Jurnal Sistem Komputer*, 11(1), 20–23. <https://doi.org/10.31937/sk.v11i1.1098>
- Putri, I. P. (2021). Analisis Performa Metode K- Nearest Neighbor (KNN) dan Crossvalidation pada Data Penyakit Cardiovascular. *Indonesian Journal of Data and Science*, 2(1), 21–28. <https://doi.org/10.33096/ijodas.v2i1.25>
- Sindhu, R. D., Sari, I., & Lestari, D. P. (2021). PEMBUATAN PROTOTYPE SMART HOME MENGGUNAKAN NODEMCU ESP8266 V3 DAN CHAT BOT PADA SMARTPHONE ANDROID. *Jurnal Ilmiah Informatika Komputer*, 26(2), 123–135. <https://doi.org/10.35760/ik.2021.v26i2.4157>

- Stanley, Felix, & Lisangan, E. A. (2020). Simulasi Deteksi Banjir Untuk Peringatan Dini Diolah Memakai Metode KNN Berbasis Arduino. *TEMATIKA: Jurnal Penelitian Teknik Informatika dan Sistem Informasi*, 8(1), 9–22.
- Sunarharum, W. B., Fibrianto, K., Yuwono, S. S., & Nur, M. (2019). *Sains Kopi Indonesia*. UB Press.
- Susanto, F., Prasiani, N. K., & Darmawan, P. (2022). IMPLEMENTASI INTERNET OF THINGS DALAM KEHIDUPAN SEHARI-HARI. *Jurnal Imagine*, 2(1), 35–40. <https://doi.org/10.35886/imagine.v2i1.329>
- Susilawati, H., Ikhsan, A. F., & Salman, F. (2020). Prototyping Alat Pendeteksi Kematangan Buah Kopi Berbasis Arduino Menggunakan Sensor Apds Gy-9960. *Jurnal Penelitian dan Pengembangan Teknik Elektro Telekomunikasi Indonesia*, 11(1), 8–15.
- Wang, B., Gan, X., Liu, X., Yu, B., Jia, R., Huang, L., & Jia, H. (2020). A Novel Weighted KNN Algorithm Based on RSS Similarity and Position Distance for Wi-Fi Fingerprint Positioning. *IEEE Access*, 8, 30591–30602. <https://doi.org/10.1109/ACCESS.2020.2973212>
- Wanita, F., Mashud, Angriawan, R., & Elma Pratiwi, C. (2021). RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN PERSEDIAAN (CONTROL BUFFER STOCK) UNTUK EFISIENSI KEWIRAUSAHAAN PENJUALAN KOPI PADA SOFT COFFEE. *Jurnal Teknologi Informasi Universitas Lambung Mangkurat (JTIULM)*, 6(1), 9–18. <https://doi.org/10.20527/jtiulm.v6i1.70>
- Yulianto, A. P., & Darwis, S. (2021). Penerapan Metode K-Nearest Neighbors (kNN) pada Bearing. *Jurnal Riset Statistika*, 1(1), 10–18. <https://doi.org/10.29313/jrs.v1i1.16>
- Zhao, D., Hu, X., Xiong, S., Tian, J., Xiang, J., Zhou, J., & Li, H. (2021). k-means clustering and kNN classification based on negative databases. *Applied Soft Computing*, 110, 107732. <https://doi.org/10.1016/j.asoc.2021.107732>

LAMPIRAN

Lampiran 1. Source Code KNN Ardiuno Uno

```
#include <EEPROM.h>
#include <Arduino_KNN.h>
#include <ArduinoJson.h>
#include <Arduino.h>
#include <SoftwareSerial.h>

const int INPUTS = 2;          // Klasifikasi dari input sensor ph dan waterflow
const int CLASSES = 2;        // Jumlah dari objek klasifikasi (baik dan kurang)
const int EXAMPLES_PER_CLASS = 30; // Jumlah data yang dilatih
const int K = 3;

// esp serial communication
SoftwareSerial espSerial(5, 6); //RX TX

// Create a new KNNClassifier
KNNClassifier myKNN(INPUTS);

// Names for each class (object type)
String label[CLASSES] = { "baik", "kurang" };

// Array to store data to pass to the KNN library
float kopi[INPUTS];

int TURBINE;
int HSensor = 2;
float ambilflow;
//int arrflow [10];
double nilaiflow;
int ok;
float Calc;
int periksa;

int pHSense = A0;
int samples = 10;
float adc_resolution = 1024.0;
float voltage;

int updateDataWaterflow;
int updateDataWaterflow2;
int updateDataPH;
int updateDataPH2;

int showDataSetPH;
int showDataSetPH2;

int data;
```

```

float dataSensor = 0.0;

// ph
float ph(float voltage) {
  // kalkulasi ph
  return 7 + ((2.5 - voltage) / 0.18);
}

// waterflow
void speed_rpm() {
  TURBINE++;
}

float read_flow_sensor() {
  // sensor waterflow
  TURBINE = 00;
  sei();
  delay(1000);
  cli();
  Calc = (TURBINE * 60 / 7.5);
  return Calc;
}

// read sensor
void read_sensor(float kopi[]) {
  /*
   * fungsi untuk membaca semua sensor
   */

  // read ph sensor
  int measurings = 0;
  for (int i = 0; i < samples; i++) {
    measurings += analogRead(pHSense);
    delay(10);
  }

  voltage = 5 / adc_resolution * measurings / samples;

  // read waterflow sensor
  ambilflow = read_flow_sensor();
  ambilflow = ambilflow * 4 + 40;

  kopi[0] = ph(voltage);
  kopi[1] = ambilflow;
  delay(3000);
}

```

```

// read training data from eeprom
void train_data_baik(float kopi[], int i) {
  // read nilai sensor sample dari eeprom
  float wf = EEPROM.read(i);
  float ph = EEPROM.read(i + 60);

  kopi[0] = wf * 4;
  kopi[1] = (ph / 100) * 4, 2;

  // Serial.print("Nilai Array KOPI: ");
  // Serial.print(kopi[0]);
  // Serial.print(",");
  // Serial.print(kopi[1]);
  // Serial.println(",");
  delay(500);
}

void train_data_kurang(float kopi[], int i) {
  // read nilai sensor sample dari eeprom
  float wf = EEPROM.read(i + 30);
  float ph = EEPROM.read(i + 90);

  kopi[0] = wf * 4;
  kopi[1] = (ph / 100) * 4, 2;
  // kopi[0] = EEPROM.read(i + 30);
  // kopi[1] = EEPROM.read(i + 90);

  // Serial.print("Nilai Array KOPI: ");
  // Serial.print(kopi[0]);
  // Serial.print(",");
  // Serial.print(kopi[1]);
  // Serial.println(",");
  delay(500);
}

// store data waterflow baik
void store_waterflow_baik(int i) {
  /*
   fungsi untuk menyimpan nilai waterflow ke eeprom
  */

  float rand = random(360, 365);
  ambilflow = rand / 4;

  Serial.print("\n Waterflow = ");
  Serial.println(ambilflow);
  Serial.println(i);
}

```

```

EEPROM.write(i, ambilflow);
return ambilflow;

// ambilflow = read_flow_sensor();
// if (ambilflow > 50) {
//   Serial.println("kalibrasi waterflow");
//   for (int i = 0; i < 20; i++) {
//     Serial.print(":*");
//     delay(1000);
//   }
//   // masukin nilai
//   ambilflow = read_flow_sensor();
//   EEPROM.write(i, ambilflow);
//   Serial.print("\n Waterflow = ");
//   Serial.println(ambilflow);
//   Serial.println(i);
//   EEPROM.write(i, ambilflow);
//   return ambilflow;
// }

// store data waterflow kurang
void store_waterflow_kurang(int i) {

if (i < 50) {
  float rand = random(200, 350);
  ambilflow = rand / 4;

  Serial.print("\n Waterflow = ");
  Serial.println(ambilflow);
  Serial.println(i);
  EEPROM.write(i, ambilflow);
} else {
  float rand = random(400, 500);
  ambilflow = rand / 4;

  Serial.print("\n Waterflow = ");
  Serial.println(ambilflow);
  Serial.println(i);
  EEPROM.write(i, ambilflow);
}
return ambilflow;

// ambilflow = read_flow_sensor();
// if (ambilflow > 50) {
//   Serial.println("kalibrasi waterflow");
//   for (int i = 0; i < 20; i++) {
//     Serial.print(":*");

```

```

// delay(1000);
// }
// // masukin nilai
// ambilflow = read_flow_sensor();
// EEPROM.write(i, ambilflow);
// Serial.print("\n Waterflow = ");
// Serial.println(ambilflow);
// Serial.println(i);
// return ambilflow;
//}
}

// update waterflow baik
void update_waterflow_baik() {
  int i;
  i = -1;
  while (true) {
    i++;
    store_waterflow_baik(i);
    if (i == 30) {
      Serial.println("FINISH");
      break;
    }
  }
}

// update waterflow kurang
void update_waterflow_kurang() {
  int i;
  i = 29;
  while (true) {
    i++;

    store_waterflow_kurang(i);
    if (i == 60) {
      Serial.println("FINISH");
      break;
    }
  }
}

// store data ph baik
void store_ph_baik() {
  /*
  fungsi untuk menyimpan nilai ph ke eeprom
  */
  for (int i = 60; i < 90; i++) {
    int measurings = 0;

```

```

for (int i = 0; i < samples; i++) {
  measurements += analogRead(pHSense);
  delay(10);
}

// voltage = 5 / adc_resolution * measurements / samples;

// int hasil = ph(voltage) * 100 / 4;

// Serial.print("pH= ");
// Serial.println(ph(voltage));
// EEPROM.write(i, hasil);
// Serial.println(hasil);

int rand = random(481, 515);
float hasil = rand / 4;
Serial.print("pH= ");
Serial.print(hasil);
Serial.println(i);
EEPROM.write(i, hasil);
}
}

// store data ph kurang
void store_ph_kurang() {
  for (int i = 90; i < 120; i++) {
    int measurements = 0;

    for (int i = 0; i < samples; i++) {
      measurements += analogRead(pHSense);
      delay(10);
    }

    // voltage = 5 / adc_resolution * measurements / samples;

    // int hasil = ph(voltage) * 100 / 4;

    // Serial.print("pH= ");
    // Serial.println(ph(voltage));
    // EEPROM.write(i, hasil);
    // Serial.println(hasil);

int rand = random(200, 400);
float hasil = rand / 4;
Serial.print("pH= ");

```



```

    Serial.print(hasil);
    Serial.println(i);
    EEPROM.write(i, hasil);
  }
}

// kNN
void knn_process() {
  // json
  char json_string[256];

  StaticJsonDocument<200> doc;
  int classification;

  Serial.println("k-NN Algorithm Processing....");

  // get sensor data
  read_sensor(kopi);
  delay(1000);

  // Classify the object
  classification = myKNN.classify(kopi, K);
  delay(1000);

  if (label[classification] == "baik") {
    // to json
    doc["ph"] = kopi[0];
    doc["waterflow"] = kopi[1] / 100, 2;
    doc["status"] = label[classification];
    delay(4000);
    serializeJson(doc, Serial);
    Serial.println();
    delay(4000);

    // send to esp
    serializeJson(doc, json_string);
    delay(5000);
    espSerial.println(json_string);
  } else {
    // to json
    doc["ph"] = kopi[0];
    doc["waterflow"] = kopi[1] / 100, 2;
    doc["status"] = label[classification];
    delay(4000);
    serializeJson(doc, Serial);
    Serial.println();
    delay(4000);
  }
}

```

```

// send to esp
serializeJson(doc, json_string);
delay(5000);
espSerial.println(json_string);
}
delay(5000);
}

void menu() {
  while (Serial.available() == 0) {
  }

  int menuChoice = Serial.parseInt();

  switch (menuChoice) {
    case 1:
      Serial.println("KNN");
      knn_process();
      break;
  }
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  espSerial.begin(9600);

  pinMode(HSensor, INPUT);
  attachInterrupt(00, speed_rpm, RISING);

  Serial.println("Arduino k-NN coffee classifier");
  delay(100);

  // randomSeed(analogRead(0));

  // Ask user for the name of each object
  for (int currentClass = 0; currentClass < CLASSES; currentClass++) {
    // Ask user to show examples of each object
    if (currentClass == 1) {
      Serial.print("Show me an example ");
      Serial.println(label[currentClass]);
      Serial.println("Read Data Train From Eeprom");

      for (int currentExample = 0; currentExample < EXAMPLES_PER_CLASS;
currentExample++) {
        // baca data eeprom
        train_data_kurang(kopi, currentExample);

```

```

    // tambahkan data eeprom ke dalam k-NN model
    // myKNN.addExample(kopi, currentClass);
    delay(200);
}
} else {
    Serial.print("Show me an example ");
    Serial.println(label[currentClass]);
    Serial.println("Read Data Train From Eeprom");

    for (int currentExample = 0; currentExample < EXAMPLES_PER_CLASS;
currentExample++) {
        // baca data eeprom
        train_data_baik(kopi, currentExample);

        // tambahkan data eeprom ke dalam k-NN model
        // myKNN.addExample(kopi, currentClass);
        delay(200);
    }
}
Serial.println("Selesai Menambahkan k-NN Model");
}

// Serial.println("1. Knn");
// Serial.println("2. update data waterflow baik");
// Serial.println("3. update data waterflow kurang");
// Serial.println("4. store data ph baik");
// Serial.println("5. store data ph kuran");

// Serial.println("Which sensor would you like to read? ");
}

void loop() {
    // put your main code here, to run repeatedly:
    // menu();

    knn_process();

    // // update data waterflow baik
    // update_waterflow_baik();

    // // update data waterflow kurang
    // update_waterflow_kurang();

    // // store data ph baik
    // store_ph_baik();

    // // store data ph kurang
    // store_ph_kurang();
}

```

```
}
```

Lampiran 2. Source Code KNN ESP

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

const char* ssid = "pypy"; //ganti nama wif
const char* password = "12345678"; //ganti password wifi

//Your Domain name with URL path or IP address with path
const char* serverName = "https://7hqgsn45-3000.asse.devtunnels.ms/temporary";

// the following variables are unsigned longs because the time, measured in
// milliseconds, will quickly become a bigger number than can be stored in an int.
unsigned long lastTime = 0;
// Set timer to 5 seconds (5000)
unsigned long timerDelay = 5000;

// communication serial
#define RXp2 16
#define TXp2 17

void get_data() {
  StaticJsonDocument<200> doc;

  while (Serial2.available() > 0) {
    char json_string[256];
    // datas = Serial2.readString();
    String datas = Serial2.readString();
    Serial.println(datas);
    deserializeJson(doc, datas);
    // serializejson to string
    serializeJson(doc, json_string);

    // send data
    //Check WiFi connection status
    if (WiFi.status() == WL_CONNECTED) {
      HTTPClient http;

      // Your Domain name with URL path or IP address with path
      http.begin(serverName);

      // If you need an HTTP request with a content type: application/json, use the
      following:
```

```

http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(json_string);

Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
// Serial2.println(httpResponseCode);
// Free resources
http.end();
} else {
  Serial.println("WiFi Disconnected");
}
}
}

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  Serial2.begin(9600, SERIAL_8N1, RXp2, TXp2);

  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
}

void loop() { // run over and over
  get_data();
}

```