

**Rancang Bangun Sistem Klasifikasi Kelayakan Madu
Berdasarkan Kadar Gula dan Warna untuk Penderita
Diabetes Mellitus menggunakan Metode *K-Nearest
Neighbors***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Niko Aji Nugroho
NIM: 195150300111029



PROGRAM STUDI TEKNIK KOMPUTER
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2023

PENGESAHAN

RANCANG BANGUN SISTEM KLASIFIKASI KELAYAKAN MADU BERDASARKAN
KADAR GULA DAN WARNA UNTUK PENDERITA DIABETES MELLITUS
MENGUNAKAN METODE K-NEAREST NEIGHBORS

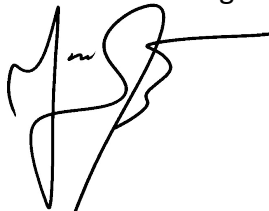
SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Niko Aji Nugroho
NIM: 195150300111029

Skripsi ini telah diuji dan dinyatakan lulus pada
20 Juli 2023
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Dahnia Syauqy, S.T., M.T., M.Sc.
NIP. 198704232020121008

Dosen Pembimbing II

Prof.Dr.Eng. Fitri Utaminingrum, S.T., M.T.
NIP. 198207102008122001

Mengetahui
Ketua Departemen Teknik Informatika

Achmad Basuki, S.T., M.MG., Ph.D.
NIP: 197411182003121002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 20 Juli 2023

Niko Aji Nugroho

NIM: 195150300111029

PRAKATA

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan berkat, rahmat, dan karunianya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Klasifikasi Kelayakan Madu Berdasarkan Kadar Gula dan Warna untuk Penderita Diabetes Mellitus menggunakan Metode K-Nearest Neighbors” dengan baik, lancar, dan tepat waktu sesuai dengan rencana penulis sebagai syarat dalam memperoleh gelar Sarjana Teknik pada program studi Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya.

Pengerjaan penelitian dan penyusunan skripsi ini tidak lepas dari dukungan, bantuan, serta doa dari beberapa pihak. Dengan ini, penulis mengucapkan rasa hormat dan banyak terima kasih kepada:

1. Orang tua penulis yaitu Bapak Joko dan Ibu Setiani, serta keluarga yang telah memberikan dukungan, memberikan semangat, serta selalu mendoakan penulis sehingga skripsi ini dapat selesai,
2. Bapak Prof. Ir. Wayan Firdaus Mahmudy, S.Si., MT., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya,
3. Bapak Achmad Basuki, S.T, M.MG., Ph.D. selaku Ketua Departemen Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya,
4. Bapak Barlian Henryranu Prasetyo, S.T., M.T., Ph.D. selaku Ketua Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya,
5. Bapak Dahnil Syauqy, S.T, M.T M.Sc. selaku dosen pembimbing pertama yang telah memberikan saran, masukan, serta semangat kepada penulis selama proses penelitian dan penyusunan skripsi,
6. Ibu Prof. Fitri Utaminigrum, ST., MT., Dr. Eng. selaku dosen pembimbing kedua yang telah memberikan saran, masukan, serta semangat kepada penulis selama proses penelitian dan penyusunan skripsi,
7. Seluruh jajaran dosen dan tenaga pendidik Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya,
8. Reza Putri Azhara selaku teman dekat dari penulis yang selalu memberikan dukungan, semangat dan bantuannya kepada penulis,
9. Leo, Denis, Abdan, Raihan, Andy, Ricko, Arga, Ghifari, Hilman, dan seluruh teman-teman sedulur Teknik Komputer yang telah membantu dan mendukung penulis untuk menyelesaikan skripsi, serta seluruh pihak yang tidak dapat dituliskan satu persatu.

Dengan segala kerendahan hati, penulis menyadari bahwa laporan ini jauh dari kesempurnaan, penulis mengharapkan apabila terdapat kritik maupun saran yang bersifat membangun dari para dosen penguji dan pembaca, Penulis berharap laporan ini dapat memberikan manfaat bagi yang membutuhkan.

Malang, 20 Juli 2023

Penulis

nikonugroho@student.ub.ac.id

ABSTRAK

Niko Aji Nugroho, Rancang Bangun Sistem Klasifikasi Kelayakan Madu Berdasarkan Kadar Gula dan Warna untuk Penderita Diabetes Mellitus menggunakan Metode K-Nearest Neighbors

Pembimbing: Dahnia Syauqy, S.T., M.T., M.Sc. dan Prof.Dr.Eng. Fitri Utaminigrum, S.T., M.T.

Madu merupakan salah satu jenis produk alam yang diproduksi oleh koloni lebah madu liar maupun lebah madu peternakan. Madu terdiri dari beberapa komponen yaitu fruktosa, glukosa, sukrosa, mineral, vitamin, dan berbagai enzim. Madu juga dianjurkan untuk dikonsumsi oleh penderita Diabetes mellitus. Penderita Diabetes mellitus yang mengkonsumsi madu tentu dapat menggantikan peranan gula sebagai asupan energi. Namun, sangat disayangkan karena ada saja beberapa oknum penjual madu yang nakal dengan mencampur madu murni dengan pemanis murah. Tentu dengan adanya fenomena tersebut, madu menjadi momok yang menakutkan bagi penderita Diabetes mellitus. Pasalnya, jika penderita Diabetes mellitus rutin mengkonsumsi madu yang tidak layak maka akan mengakibatkan naiknya kadar gula darah pada penderita. Berdasarkan permasalahan yang ada, diperlukanlah sistem yang bisa mengklasifikasikan kelayakan konsumsi madu bagi penderita Diabetes mellitus. Sistem menggunakan Arduino Mega sebagai pemroses utama, beserta komponen lain seperti sensor kapasitansi, sensor TCS3200, RTC, Modul Micro SD, LCD, dan tiga buah button. Penulis juga menggunakan salah satu algoritma machine learning yaitu K-Nearest Neighbors untuk mengklasifikasi madu. Setelah melakukan pengujian, didapatkan nilai K terbaik untuk algoritma ini yaitu K=3. Selain itu, sistem dapat bekerja dengan optimal sesuai fungsinya dan memiliki akurasi sebesar 90% untuk 10 sampel yang diujikan.

Kata kunci: madu, kapasitansi, TCS3200, Diabetes mellitus, *datalogger*, K-Nearest Neighbors

ABSTRACT

Niko Aji Nugroho, Rancang Bangun Sistem Klasifikasi Kelayakan Madu Berdasarkan Kadar Gula dan Warna untuk Penderita Diabetes Mellitus menggunakan Metode K-Nearest Neighbors

Supervisors: Dahnial Syauqy, S.T., M.T., M.Sc. and Prof.Dr.Eng. Fitri Utaminingrum, S.T., M.T.

One of the natural products made by both cultivated and wild honeybee colonies is honey. Numerous substances, including fructose, glucose, sucrose, minerals, vitamins, and different enzymes, can be found in honey. People with diabetes should also consume honey, according to recommendations. Patients with diabetes mellitus who consume honey can substitute it for sugar as a source of energy. Unfortunately, some dishonest people combine pure honey with inexpensive sweets. Patients with diabetes mellitus should be concerned about this issue because regular use of inappropriate honey can raise blood sugar levels. Consequently, a method is required to categorize. As a result, a system is required to categorize whether honey eating is appropriate for those with diabetes mellitus. The system includes an LCD, three buttons, a capacitance sensor, a TCS3200 sensor, an RTC, a Micro SD module, and an Arduino Mega as its primary processor. The K-Nearest Neighbors algorithm, one of the machine learning algorithms, is also used by the author to categorize honey. Testing has revealed that $K=3$ is the ideal value for this algorithm's constant K . Additionally, the system can operate at peak efficiency for its intended purpose and has a 90% accuracy for the 10 evaluated samples.

Keywords: honey, capacitance, TCS3200, Diabetes mellitus, datalogger, K-Nearest Neighbors

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Kepustakaan.....	6
2.1.1 Sistem Klasifikasi Kualitas Jenis-Jenis Madu Berdasarkan Warna, Kecerahan, dan PH menggunakan JST <i>Backpropagation</i>	7
2.1.2 Rancang Bangun Alat Pendeteksi Kadar Gula pada Minuman Berperisa menggunakan Sensor Kapasitif	7
2.1.3 Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree	7
2.2 Dasar Teori.....	8
2.2.1 Kadar Gula	8
2.2.2 Madu	8
2.2.3 Diabetes Mellitus	8
2.2.4 Kapasitor Keping Sejajar.....	9
2.2.5 Kapasitansi	9
2.2.6 Datalogger	9

2.2.7 K-Nearest Neighbors	9
BAB 3 METODOLOGI	12
3.1 Tipe Penelitian	12
3.2 Metode Penelitian	12
3.2.1 Studi Literatur	13
3.2.2 Rekayasa Kebutuhan	14
3.2.3 Perancangan dan Implementasi	14
3.2.4 Pengambilan dan Pengolahan Data	14
3.2.5 Pengujian Sistem	15
3.2.6 Analisis Hasil Pengujian	15
3.2.7 Kesimpulan dan Saran	15
3.3 Stakeholder Penelitian	15
3.4 Partisipan Penelitian	15
3.5 Lokasi Penelitian	16
3.6 Peralatan Pendukung Yang Digunakan	16
BAB 4 REKAYASA KEBUTUHAN	17
4.1 Kajian Masalah	17
4.2 Identifikasi Stakeholder	17
4.3 Kebutuhan Fungsional	17
4.4 Spesifikasi Sistem	18
4.5 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak	19
4.5.1 Kebutuhan Perangkat Keras	19
4.5.2 Kebutuhan Perangkat Lunak	27
BAB 5 PERANCANGAN DAN IMPLEMENTASI	30
5.1 Perancangan	30
5.1.1 Perancangan Perangkat Keras	30
5.1.2 Perancangan Perangkat Lunak	41
5.2 Implementasi Sistem	49
5.2.1 Implementasi Perangkat Keras	49
5.2.2 Implementasi Perangkat Lunak	53
BAB 6 PENGUJIAN DAN ANALISIS	68
6.1 Pengujian Sensor	68

6.1.1 Pengujian Sensor Kapasitansi.....	68
6.1.2 Pengujian Sensor TCS3200.....	69
6.2 Pengujian Nilai K	71
6.2.1 Prosedur Pengujian Nilai K.....	71
6.2.2 Hasil dan Analisis Pengujian Nilai K.....	71
6.3 Pengujian Keseluruhan Sistem	72
6.3.1 Prosedur Pengujian Keseluruhan Sistem	72
6.3.2 Hasil dan Analisis Pengujian Keseluruhan Sistem	73
BAB 7 PENUTUP	80
7.1 Kesimpulan.....	80
7.2 Saran	80
DAFTAR REFERENSI	82
LAMPIRAN A KODE PROGRAM.....	85
A.1 Kode Program Keseluruhan Sistem	85
LAMPIRAN B DOKUMENTASI	102
B.1 Dokumentasi dengan Pakar Madu	102
B.2 Dokumentasi Pengujian Sistem	103

DAFTAR TABEL

Tabel 2.1 Tinjauan Kepustakaan	6
Tabel 4.1 Spesifikasi Arduino Mega	20
Tabel 4.2 Spesifikasi Sensor Kapasitansi	21
Tabel 4.3 Spesifikasi Sensor TCS3200	22
Tabel 4.4 Spesifikasi LCD 20x4	23
Tabel 4.5 Spesifikasi I2C	24
Tabel 4.6 Spesifikasi RTC	24
Tabel 4.7 Spesifikasi Modul Micro SD	25
Tabel 4.8 Spesifikasi Micro SD Card	26
Tabel 4.9 Spesifikasi Push Button	27
Tabel 5.1 Konfigurasi Pin	31
Tabel 5.2 Konfigurasi Pin (Lanjutan)	32
Tabel 5.3 Fitur dan Kelas	44
Tabel 5.4 Implementasi Pembuatan Dataset	66
Tabel 6.1 Hasil Pengujian Sensor Kapasitansi	69
Tabel 6.2 Hasil Pengujian Sensor TCS3200	70
Tabel 6.3 Hasil Pengujian Nilai K	72
Tabel 6.4 Hasil Pengujian Keseluruhan Sistem (Fungsional)	73
Tabel 6.5 Hasil Pengujian Keseluruhan Sistem (Akurasi)	79

DAFTAR GAMBAR

Gambar 2.1 <i>K-Nearest Neighbors</i>	10
Gambar 3.1 Diagram Alir Penelitian.....	13
Gambar 4.1 Arduino Mega.....	20
Gambar 4.2 Sensor Kapasitansi.....	21
Gambar 4.3 Sensor Warna TCS3200	22
Gambar 4.4 LCD 20x4.....	23
Gambar 4.5 I2C.....	23
Gambar 4.6 RTC DS3231	24
Gambar 4.7 Modul Micro SD	25
Gambar 4.8 Micro SD Card.....	26
Gambar 4.9 Push Button	27
Gambar 4.10 Arduino IDE	28
Gambar 4.11 Jupyter Notebook.....	28
Gambar 5.1 Diagram Blok Perancangan Sistem	30
Gambar 5.2 Skematik Perancangan Sistem	31
Gambar 5.3 Perancangan <i>Cover</i> Sistem.....	33
Gambar 5.4 Perancangan Sensor Kapasitansi.....	33
Gambar 5.5 Diagram Alir Sensor Kapasitansi.....	34
Gambar 5.6 Perancangan Sensor TCS3200	35
Gambar 5.7 Diagram Alir Sensor TCS3200.....	36
Gambar 5.8 Perancangan Micro SD Card.....	37
Gambar 5.9 Diagram Alir Penyimpanan Modul Micro SD	38
Gambar 5.10 Diagram Alir Penyimpanan EEPROM	39
Gambar 5.11 Cara Kerja Protokol SPI.....	41
Gambar 5.12 Cara Kerja Protokol I2C	41
Gambar 5.13 Diagram Alir Data <i>Training</i>	42
Gambar 5.14 Diagram Alir Pembuatan Dataset.....	43
Gambar 5.15 Diagram Alir Penentuan Nilai K.....	44
Gambar 5.16 Diagram Alir Keseluruhan Sistem.....	45
Gambar 5.17 Diagram Alir Menu Klasifikasi	46

Gambar 5.18 Diagram Alir Menu History.....	47
Gambar 5.19 Diagram Alir Menu Reset Harian.....	48
Gambar 5.20 Diagram Alir Reset Total.....	49
Gambar 5.21 Sistem Tampak Depan.....	50
Gambar 5.22 Sistem Tampak Belakang	51
Gambar 5.23 Sistem Tampak Samping Kanan	51
Gambar 5.24 Sistem Tampak Atas	52
Gambar 5.25 Sistem Tampak Dalam.....	53
Gambar 6.1 Grafik Hasil Pengujian Nilai K	71
Gambar 6.2 Menu Utama	74
Gambar 6.3 Menu Klasifikasi.....	75
Gambar 6.4 Menu Klasifikasi (Lanjutan)	75
Gambar 6.5 Tampilan Menu History.....	75
Gambar 6.6 Tampilan History Klasifikasi.....	76
Gambar 6.7 Menu Reset Harian.....	76
Gambar 6.8 Menu Reset Harian (Lanjutan)	77
Gambar 6.9 Menu History Setelah Reset Harian	77
Gambar 6.10 Menu Reset Total	78
Gambar 6.11 Menu Reset Total (Lanjutan).....	78
Gambar 6.12 Menu History Setelah Reset Total.....	78

BAB 1 PENDAHULUAN

Pada bab 1 atau bab pendahuluan ialah bab pertama dalam penelitian ini yang berisikan uraian terkait dengan latar belakang penelitian, rumusan masalah yang diangkat oleh penulis, tujuan dan manfaat penelitian, serta batasan masalah yang membuat lingkup penelitian ini menjadi lebih spesifik. Pada bab ini juga terdapat sistematika penulisan mengenai isi secara general dari masing-masing bab yang termuat dalam penelitian ini.

1.1 Latar Belakang

Madu merupakan salah satu hasil alam yang diproduksi oleh koloni lebah madu liar ataupun lebah peternakan. Madu memiliki beberapa zat penyusun yaitu glukosa, fruktosa, sukrosa, vitamin, dan beberapa macam enzim. Madu juga memiliki kandungan gizi yang tinggi didalamnya sehingga mengakibatkan madu sangat digemari untuk dikonsumsi secara rutin oleh orang dewasa, anak-anak, bahkan juga bayi (Rochman, 2018).

Menurut Handayani (2021), madu juga dianjurkan untuk dikonsumsi oleh penderita Diabetes mellitus. Madu dapat dikonsumsi penderita Diabetes mellitus dimana madu akan menggantikan peranan gula sebagai asupan energi. Karena dengan mengkonsumsi madu menyebabkan resiko kenaikan kadar gula darah yang minim. Diabetes mellitus dapat diartikan sebagai sekumpulan gejala yang muncul akibat dari meningkatnya kadar gula darah secara berkelanjutan (Rachmawati, 2015). Di Indonesia, terdapat sekiranya 19,7% (Kementerian Kesehatan RI, 2019) masyarakat dengan prevalensi pra diabetes. Menurut Fitriani & Sanghati (2021), pra diabetes adalah suatu keadaan dimana kadar glukosa darah berada diatas ambang normal namun belum dapat didiagnosis sebagai Diabetes mellitus. Kasus Diabetes mellitus di Indonesia diperkirakan akan mencapai 21,3 juta orang pada tahun 2030. Hal ini karena saat ini Indonesia sudah masuk 10 besar negara dengan kasus diabetes tertinggi di dunia (Rukmana et al, 2019). Tingginya prevalensi dan angka kejadian Diabetes mellitus di Indonesia mengakibatkan perlunya diadakan pengecekan kadar glukosa darah secara rutin dan mengontrol pola makanan dengan asupan gizi yang seimbang.

Di Indonesia terdapat banyak jenis madu dengan berbagai merek, mulai dari cap lokal hingga internasional. Hal ini mengakibatkan mulai menjamurnya penjual madu di Indonesia. Namun, sangat disayangkan karena tidak semua penjual madu jujur dalam menjual produk mereka. Ada saja beberapa oknum penjual madu yang nakal dengan mencampur madu murni dengan pemanis murah. Ada pula yang dengan sengaja memberi makan lebah peternakan miliknya dengan sirup gula. Selain itu, penurunan kualitas pada madu juga bisa disebabkan oleh kontaminasi patogen saat proses produksi madu yang kurang terjaga kebersihannya. Berdasarkan berita yang beredar di masyarakat, setidaknya ada dua orang pelaku yang ditangkap polisi karena mengoplos madu di Palembang, Sumatera Selatan. Dalam proses pengoplosan madu, kadar madu murni yang digunakan hanya sekitar 5-10 persen saja dan selebihnya dicampur menggunakan bahan lain

(Tasmalinda, 2022). Tentu dengan adanya fenomena tersebut, madu menjadi momok yang menakutkan bagi penderita Diabetes mellitus. Pasalnya, jika penderita Diabetes mellitus rutin mengkonsumsi madu yang tidak layak maka akan mengakibatkan naiknya kadar gula darah pada penderita. Hal ini akan memicu berbagai jenis komplikasi penyakit pada penderita Diabetes mellitus seperti penyakit jantung, stroke, kerusakan ginjal, dan lain sebagainya (Adrian, 2020).

Saat ini, kadar gula pada suatu cairan dapat diukur melalui suatu instrumen yang disebut dengan refraktometer. Refraktometer ialah suatu instrumen yang dapat digunakan untuk mengukur kadar dari suatu zat terlarut, misalnya seperti gula. Kekurangan dari refraktometer adalah alat ini hanya dapat digunakan secara manual. Artinya harus ada pengguna yang mengoperasikannya dengan langkah-langkah tertentu. Selain itu, refraktometer juga tidak dapat melakukan pencatatan pembacaan harian atau pun total terkait jumlah madu yang dikonsumsi. Sementara bagi penderita Diabetes mellitus, adanya fitur *history* konsumsi madu harian sangatlah diperlukan. Hal ini karena, dengan mengetahui jumlah konsumsi harian madu diharapkan menjadi tolak ukur bagi penderita Diabetes mellitus dalam mempertimbangkan konsumsi madu per harinya dengan lebih bijak.

Berlandaskan adanya masalah dan kekurangan pada instrumen yang ada saat ini, penulis akan membuat sistem yang dapat mencatat dan menyimpan *history* jumlah konsumsi harian bagi penderita Diabetes mellitus. Sistem ini akan menyimpan *datalogger* pada EEPROM dan Micro SD Card, agar nantinya dapat ditampilkan dan diamati oleh penderita Diabetes mellitus. Sistem ini juga menggunakan sensor kapasitansi dan TCS3200 untuk mengukur kadar gula dan warna pada madu. Maka dengan ini, solusi yang ditawarkan oleh penulis yaitu berupa suatu Sistem Klasifikasi Kelayakan Madu Berdasarkan Kadar Gula dan Warna untuk Penderita Diabetes Mellitus menggunakan Metode K-Nearest Neighbors.

Parameter kadar gula pada madu digunakan karena penderita Diabetes mellitus tidak dianjurkan untuk mengkonsumsi makanan atau minuman dengan komposisi kadar gula yang tinggi. Parameter kadar gula untuk mengukur kelayakan madu diperoleh dari penelitian Hartini (2021) dan beberapa penelitian lain terkait pengukuran kadar gula yang menggunakan sensor kapasitansi. Pada sensor kapasitansi yang digunakan oleh Hartini (2021), didapatkan tingkat error sejumlah 2.54% dari penggunaan sensor kapasitif terhadap empat buah sampel larutan gula dengan kadar gula yang berbeda-beda. Penelitian tersebut mengungkapkan bahwasanya hasil pengukuran sensor kapasitansi yang berupa nilai kapasitansi dapat disetarakan dengan pengukuran refraktometer berupa satuan kadar gula menggunakan regresi linier.

Parameter warna pada madu digunakan karena warna termasuk ke dalam karakteristik fisik dari madu. Parameter warna untuk mengukur kelayakan madu diperoleh dari penelitian Evahelda et al (2017), dimana madu dengan warna cerah mengandung lebih banyak gula jika dibandingkan dengan madu berwarna gelap. Hal ini karena madu berwarna gelap mengandung banyak komponen fenolik yang

bersifat antioksidan. Untuk memperoleh nilai parameter warna pada madu dapat diukur menggunakan sensor warna TCS3200. Berdasarkan penelitian Alhamdani (2022), sensor TCS3200 yang digunakan dapat melakukan deteksi warna RGB dengan baik dimana saat pengujian sensor ini memiliki rata-rata error hanya sebesar 7,49%. Jika dibandingkan dengan pengambilan warna menggunakan citra digital, sensor TCS3200 akan unggul karena memiliki keluaran dataset yang berupa angka numerik sehingga mempercepat proses komputasi. Kemudian jika dibandingkan dengan sensor warna TCS230, sensor TCS3200 merupakan penyempurnaan dari sensor TCS230 dimana terdapat perbedaan dalam konsumsi arusnya (Syam & Mustika, 2018).

Pemilihan algoritma *K-Nearest Neighbors* dipilih berlandaskan dengan penelitian Tangkelayuk & Mailoa (2022) yang membandingkan tingkat akurasi antara algoritma *K-Nearest Neighbors*, *Decision Tree*, dan *Naïve Bayes*. Berdasarkan dataset kualitas air yang digunakan, ketiga algoritma tersebut yaitu *K-Nearest Neighbors*, *Decision Tree*, dan *Naïve Bayes* menghasilkan akurasi berturut-turut sebesar 86,88%, 80,84%, dan 63,60%. Sehingga menempatkan algoritma *K-Nearest Neighbors* sebagai algoritma terbaik untuk melakukan klasifikasi data. Untuk mendapatkan hasil akurasi yang maksimal pada algoritma *K-Nearest Neighbors*, diperlukan pemilihan nilai K atau jumlah tetangga terdekat secara tepat. Menurut Indrayanti et al (2017), dilakukan pengujian nilai K atau jumlah tetangga terdekat dari $K=1$ sampai $K=49$ dengan nilai ganjil saja. Hasil dari pengujian ini didapatkan nilai $K=13$ yang menjadi nilai K terbaik dengan akurasi 75,14%. Hal ini membuktikan bahwasanya pemilihan nilai K atau tetangga terdekat yang tepat dapat mempengaruhi hasil dari akurasi algoritma *K-Nearest Neighbors*.

1.2 Rumusan Masalah

Adapun rumusan masalah yang penulis angkat berlandaskan dengan latar belakang yang telah dipaparkan sebelumnya ialah sebagai berikut:

1. Bagaimana fungsionalitas sensor kapasitansi dan TCS3200, serta komponen pendukung lainnya pada sistem?
2. Bagaimana pengaruh nilai tetangga terdekat atau nilai K terhadap hasil akurasi dari algoritma *K-Nearest Neighbors* dalam proses klasifikasi kelayakan madu?
3. Bagaimana akurasi sistem secara keseluruhan dalam hal implementasi sebelum digunakan oleh penderita Diabetes mellitus?

1.3 Tujuan

Adapun beberapa tujuan yang ingin penulis capai dalam penelitian ini berlandaskan dengan rumusan masalah yang diangkat oleh penulis ialah sebagai berikut:

1. Mengetahui fungsionalitas sensor kapasitansi dan TCS3200, serta komponen pendukung lainnya pada sistem.

2. Mengetahui pengaruh nilai tetangga terdekat atau nilai K terhadap hasil akurasi dari algoritma *K-Nearest Neighbors* dalam proses klasifikasi kelayakan madu.
3. Mengetahui akurasi sistem secara keseluruhan dalam hal implementasi sebelum digunakan oleh penderita Diabetes mellitus.

1.4 Manfaat

Sistem klasifikasi ini diharapkan mampu membantu penderita Diabetes mellitus dalam membedakan mana madu yang layak dikonsumsi dan mana yang tidak. Selain itu, dengan adanya sistem ini diharapkan para penderita Diabetes mellitus dapat mempertimbangkan jumlah harian dalam mengonsumsi madu dengan lebih bijak.

1.5 Batasan Masalah

Agar lingkup penelitian ini menjadi lebih spesifik, maka batasan-batasan yang sengaja dibuat. Adapun batasan-batasan masalah tersebut adalah:

1. Implementasi sistem berupa prototipe
2. Subjek penelitian yang digunakan adalah madu
3. Madu yang akan dijadikan sampel merupakan beberapa madu dengan merek tertentu dan madu yang dijual di pinggir jalan sekitaran Kota Malang
4. Wadah untuk melakukan pengujian madu adalah wadah kue puding
5. Madu yang digunakan dalam pengujian memiliki ketinggian 1,5 cm dari dasar wadah puding

1.6 Sistematika Pembahasan

Pada sistematika pembahasan terdapat isi secara general dari masing-masing bab yang termuat dalam penelitian ini. Penelitian ini terbagi menjadi beberapa bagian bab yang telah disusun secara runtut. Sistematika penulisan skripsi dibagi menjadi beberapa bab, yaitu:

BAB I PENDAHULUAN

Pada bab 1 atau bab pendahuluan ialah bab pertama dalam penelitian ini yang berisikan uraian terkait dengan latar belakang penelitian, rumusan masalah yang diangkat oleh penulis, tujuan dan manfaat penelitian, serta batasan masalah yang membuat lingkup penelitian ini menjadi lebih spesifik. Pada bab ini juga terdapat sistematika penulisan mengenai isi secara general dari masing-masing bab yang termuat dalam penelitian ini.

BAB II LANDASAN KEPUSTAKAAN

Pada bab 2 atau bab landasan kepastakaan ialah bab kedua dalam penelitian ini yang berisikan uraian terkait dengan tinjauan kepastakaan dan dasar teori agar dapat menunjang ketersediaan informasi pada saat penyusunan penelitian. Pada

tinjauan kepustakaan terdapat beberapa literatur penelitian sebelumnya yang berpengaruh terhadap penyusunan penelitian dan akan menjadi sumber rujukan penulis. Kemudian pada dasar teori berisikan istilah-istilah yang akan digunakan oleh penulis.

BAB III METODOLOGI PENELITIAN

Pada bab 3 atau bab metodologi ialah bab ketiga dalam penelitian ini yang memiliki beberapa sub bab yang akan dibahas. Mula-mula bab ini akan menjelaskan terkait tipe penelitian dari penelitian penulis. Selanjutnya terdapat metode penelitian yang penulis akan gunakan yaitu disajikan dalam bentuk diagram alir penelitian. Kemudian juga membahas mengenai partisipan dalam penelitian ini. Keberadaan partisipan dalam penelitian ini sangatlah penting bagi penulis. Hal ini karena dengan adanya partisipan, penulis dapat banyak menggali informasi mengenai objek yang akan diteliti. Setelah itu, akan diberitahukan juga mengenai lokasi penelitian ini bertempat. Terakhir, penulis akan menyantumkan juga peralatan pendukung yang akan digunakan.

BAB IV REKAYASA KEBUTUHAN

Pada bab 4 atau bab rekayasa kebutuhan ialah bab keempat dalam penelitian ini yang berisikan uraian terkait dengan pengkajian masalah secara umum, mengidentifikasi stakeholder, menyusun keperluan fungsional beserta spesifikasi yang harus dimiliki sistem, juga menganalisis keperluan perangkat keras dan lunak penyusun sistem.

BAB V PERANCANGAN DAN IMPLEMENTASI

Pada bab 5 atau bab perancangan dan implementasi ialah bab kelima dalam penelitian ini yang berisikan uraian terkait dengan bagaimana perancangan perangkat keras dan lunak penyusun sistem. Setelah merancang skema, diperlukan implementasi untuk menerapkan rancangan tersebut.

BAB VI PENGUJIAN DAN ANALISIS

Dalam bab ini memiliki tujuan untuk melihat performa dan mengevaluasi kinerja baik komponen penyusun sistem ataupun sistem secara keseluruhan. Hal-hal yang dibahas akan meliputi hasil dan juga analisis hasil pengujian.

BAB VII PENUTUP

Pada bab 7 atau bab penutup ialah bab ketujuh dalam penelitian ini yang berisikan uraian terkait dengan kesimpulan yang dapat dipetik dan menjawab pertanyaan pada rumusan masalah. Kemudian juga disertai dengan adanya saran-saran untuk penelitian selanjutnya guna memaksimalkan hasil dari penelitian yang dilakukan oleh penulis.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab 2 atau bab landasan kepustakaan ialah bab kedua dalam penelitian ini yang berisikan uraian terkait dengan tinjauan kepustakaan dan dasar teori agar dapat menunjang ketersediaan informasi pada saat penyusunan penelitian. Pada tinjauan kepustakaan terdapat beberapa literatur penelitian sebelumnya yang berpengaruh terhadap penyusunan penelitian dan akan menjadi sumber rujukan penulis. Kemudian pada dasar teori berisikan istilah-istilah yang akan digunakan oleh penulis.

2.1 Tinjauan Kepustakaan

Sebelum memulai penelitian, penulis sudah melakukan literasi terhadap beberapa pustaka yang menunjang penelitian ini. Berikut adalah isi dari kajian pustaka yang berisikan penelitian-penelitian sebelumnya, disajikan melalui Tabel 2.1.

Tabel 2.1 Tinjauan Kepustakaan

No.	Nama Penulis (Tahun), Judul Penelitian	Kesamaan Penelitian	Perbedaan	
			Penelitian Sebelumnya	Penelitian Sekarang
1	Muhammad Habib Jufah Alhamdani (2022), "Sistem Klasifikasi Kualitas Jenis-Jenis Madu Berdasarkan Warna, Kecerahan, dan PH menggunakan JST <i>Backpropagation</i> "	Menggunakan madu sebagai subjek dari penelitian	Menggunakan parameter warna, kecerahan, dan pH, serta menggunakan metode JST <i>Backpropagation</i>	Menggunakan parameter kadar gula pada madu dan metode <i>K-Nearest Neighbor</i> untuk melakukan klasifikasi
2	Vina Tri Hartini (2021), "Rancang Bangun Alat Pendeteksi Kadar Gula pada Minuman Berperisa menggunakan Sensor Kapasitif"	Menggunakan sensor kapasitif untuk mengukur kadar gula	Mendeteksi kadar gula pada minuman berperisa. Klasifikasi hanya menggunakan seleksi kondisi	Mendeteksi kadar gula pada madu dan menggunakan metode <i>K-Nearest Neighbor</i> untuk melakukan klasifikasi

3	Aldi Tangkelayuk (2022), "Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree"	Menggunakan metode <i>K-Nearest Neighbors</i>	Mendeteksi kualitas air konsumsi dengan metode KNN, Naïve Bayes Dan Decision Tree	Mendeteksi kelayakan madu untuk penderita Diabetes mellitus
---	---	---	---	---

2.1.1 Sistem Klasifikasi Kualitas Jenis-Jenis Madu Berdasarkan Warna, Kecerahan, dan PH menggunakan JST *Backpropagation*

Penelitian yang dilakukan oleh Muhammad Habib Jufah Alhamdani (2022), dengan judul "Sistem Klasifikasi Kualitas Jenis-Jenis Madu Berdasarkan Warna, Kecerahan, dan PH menggunakan JST *Backpropagation*" merupakan pembuatan alat untuk membedakan kualitas jenis madu. Terdapat enam kelas madu yang akan diklasifikasikan, yaitu kelas madu hutan murni, buatan, dan campuran, serta madu klaceng murni, buatan, dan campuran. Sensor yang digunakan ada tiga jenis, yaitu sensor TCS3200 untuk mengidentifikasi warna RGB, sensor LDR untuk mengidentifikasi cahaya, dan sensor pH untuk mengidentifikasi pH madu. Dengan menggunakan algoritme JST *Backpropagation*, sistem ini mampu mencapai akurasi 94.45% dengan rata-rata waktu komputasi 0.80076 sekon.

2.1.2 Rancang Bangun Alat Pendeteksi Kadar Gula pada Minuman Berperisa menggunakan Sensor Kapasitif

Penelitian yang dilakukan oleh Vina Tri Hartini (2021) yang berjudul "Rancang Bangun Alat Pendeteksi Kadar Gula pada Minuman Berperisa menggunakan Sensor Kapasitif" merupakan penelitian yang telah membuka pikiran penulis dalam menemukan ide. Penelitian ini dilakukan menggunakan sensor kapasitif yang dibuat dengan prinsip kapasitor pelat sejajar untuk mendeteksi kadar gula yang terdapat pada minuman berperisa. Terdapat tiga opsi dalam alur seleksi kondisi kadar gula, yaitu kadar gula rendah, sedang, dan tinggi. Pada penelitian ini juga diperlukan bantuan alat berupa refraktometer, untuk menguji keseluruhan sistem yang telah dibuat. Rata-rata nilai *error* yang didapat dari membandingkan dengan hasil pengukuran refraktometer yaitu sebesar 2.54%.

2.1.3 Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree

Penelitian selanjutnya yaitu penelitian yang dilakukan oleh Aldi Tangkelayuk (2022) dengan judul "Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree". Penelitian ini dilakukan untuk membandingkan tingkat akurasi dari kualitas air konsumsi dengan menggunakan metode *K-Nearest Neighbors*, *Naïve Bayes* Dan *Decision Tree*. Berdasarkan dataset kualitas air yang digunakan, ketiga algoritma tersebut yaitu *K-Nearest Neighbors*, *Decision Tree*, dan *Naïve Bayes* menghasilkan akurasi berturut-turut sebesar 86,88%, 80,84%,

dan 63,60%. Sehingga menempatkan algoritma *K-Nearest Neighbors* sebagai algoritma terbaik untuk melakukan klasifikasi data.

2.2 Dasar Teori

Pada dasar teori akan dijabarkan terkait teori-teori yang dapat menunjang pemahaman untuk mengimplementasikan penelitian ini. Keberadaan dasar teori ini sangat diperlukan untuk mendapatkan hasil yang maksimal, sesuai dengan harapan penulis.

2.2.1 Kadar Gula

Gula ialah senyawa alami yang signifikan sebagai sumber kalori karena gula memiliki zat penyusun yang mudah untuk dicerna didalam tubuh. Jumlah gula yang terkandung didalam suatu bahan atau produk disebut dengan kadar gula. Gula juga dapat menjadi berbahaya jika dikonsumsi secara berlebihan, terlebih lagi ketika makanan atau minuman yang memiliki komposisi kadar gula yang cukup tinggi. Menurut Nareza (2020), kadar gula yang baik untuk dikonsumsi per hari oleh masyarakat adalah tidak lebih dari 30 gram gula atau setara dengan 7 sendok teh per orangnya bagi orang dewasa.

2.2.2 Madu

Madu merupakan salah satu jenis produk alam yang diproduksi oleh koloni lebah madu liar maupun lebah madu peternakan. Madu memiliki beberapa zat penyusun yaitu glukosa, fruktosa, sukrosa, vitamin, dan beberapa macam enzim. Madu juga memiliki kandungan gizi yang tinggi didalamnya sehingga mengakibatkan madu sangat digemari untuk dikonsumsi secara rutin oleh orang dewasa, anak-anak, bahkan juga bayi (Rochman, 2018). Secara umum madu dapat dikategorikan menjadi tiga jenis, yaitu madu murni, sirupan, dan kurasan.

Pada dasarnya hampir seluruh produk makanan atau minuman mengandung gula, termasuk madu. Madu yang layak dikonsumsi oleh masyarakat terutama penderita Diabetes mellitus adalah madu murni yang memiliki kadar gula rendah. Namun akibat beberapa oknum yang tidak bertanggungjawab, madu yang benar-benar murni sudah sangat sulit untuk dibedakan. Pasalnya, oknum madu ini juga kerap mencampur madu dengan beberapa bahan lain seperti pemanis murah, sirup gula, dan bahan lainnya demi meraup keuntungan semata. Alhasil madu dari oknum inilah yang dapat dikategorikan sebagai madu yang tidak layak konsumsi bagi masyarakat, terutama penderita Diabetes mellitus.

2.2.3 Diabetes Mellitus

Diabetes mellitus (DM) dapat diartikan sebagai sekumpulan gejala yang muncul akibat dari meningkatnya kadar gula darah secara berkelanjutan (Rachmawati, 2015). Menurut Fadli (2022), gejala ini terjadi sebab kadar gula darah puasa seseorang melebihi 125 mg/dL. Penderita Diabetes mellitus wajib membatasi asupan kadar gula yang masuk ke dalam tubuh. Karena jika kadar gula darah penderita meningkat drastis, maka akan memicu berbagai jenis komplikasi

penyakit pada penderita Diabetes mellitus seperti penyakit jantung, stroke, kerusakan ginjal, dan lain sebagainya (Adrian, 2020).

2.2.4 Kapasitor Keping Sejajar

Kapasitor keping sejajar atau pelat sejajar ialah suatu konsep rancangan untuk membuat sensor kapasitansi. Kapasitor keping sejajar ini tersusun atas dua buah keping atau pelat konduktor yang diletakkan dengan posisi sejajar pada jarak tertentu yang dipisahkan oleh bahan dielektrik. Bahan dielektrik adalah sebutan untuk bahan-bahan yang bersifat isolator terhadap listrik, contohnya seperti porselen, plastik, dan kaca. Kapasitor keping sejajar baru dapat bekerja ketika terhubung oleh suatu muatan listrik, seperti dari baterai. Ketika sudah saling terhubung, kapasitor keping sejajar akan mengarahkan muatan dari setiap konduktor hingga selisih potensial antara kutub positif dan negatif sama dengan selisih potensial antara kutub positif dan negatif pada baterai atau sumber daya lain yang digunakan (Hartini, 2021).

2.2.5 Kapasitansi

Kapasitansi merupakan salah satu besaran fisis yang memiliki simbol C , dan memiliki satuan Standar Internasional (SI) yaitu Farad (F). Kapasitansi adalah sekumpulan muatan listrik yang dapat disimpan pada konduktor untuk setiap potensial listrik yang sudah ditetapkan. Muatan listrik ini biasanya disimpan dalam sebuah kapasitor dua keping. Secara garis besar, nilai kapasitansi dapat dicari menggunakan persamaan 2.1.

$$C = \frac{Q}{V} \quad (2.1)$$

Berdasarkan persamaan tersebut, dapat disimpulkan bahwasanya nilai kapasitansi (C) memiliki hubungan berbanding terbalik dengan nilai tegangan (V). Ini berarti bahwa semakin tinggi nilai tegangan (V) maka semakin kecil nilai kapasitansi (C), begitu pula sebaliknya.

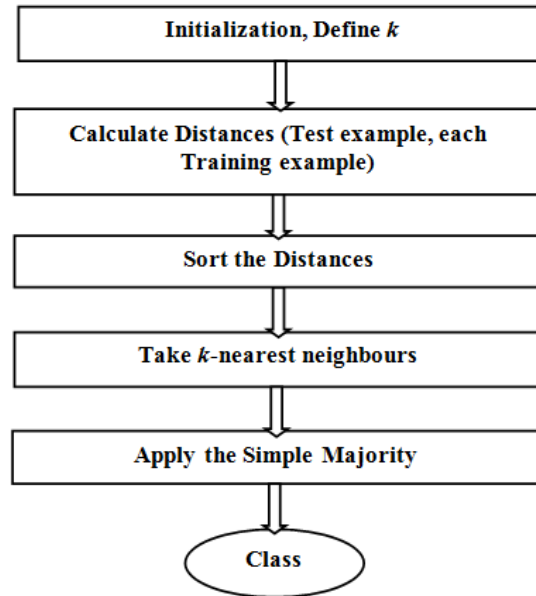
2.2.6 Datalogger

Datalogger adalah suatu fitur pada sistem yang dapat menyimpan *history* pengukuran sebelumnya. Cara kerja datalogger yaitu setiap kali user menggunakan menu klasifikasi pada sistem, maka sistem akan menyimpan keluaran dari hasil klasifikasi beserta tanggal dan waktunya. Fitur ini diberikan pada sistem agar pengguna alat ini mampu untuk melihat tanggal dan waktu terakhir pasca melakukan klasifikasi madu.

2.2.7 K-Nearest Neighbors

K-Nearest Neighbors (KNN) ialah sebuah metode yang biasa diterapkan untuk pengelompokan atau pengklasifikasian suatu objek data. Algoritma *K-Nearest Neighbors* merupakan algoritma *supervised learning* yang menggunakan mayoritas dari tetangga terdekat atau nilai K untuk mendapatkan hasil dari objek yang baru diklasifikasi. Selain itu, objek baru juga dikategorikan oleh algoritma ini menggunakan atribut dan sampel-sampel dari data latih.

Dalam menentukan nilai K pada metode ini terdapat beberapa syarat yang perlu untuk diperhatikan. Syarat-syarat tersebut meliputi nilai K yang tidak boleh melewati jumlah data *training*, nilai K harus bernilai ganjil, dan tidak disarankan menggunakan $K=1$ (Rivki, 2017). Pada algoritma ini juga terdapat empat cara yang digunakan untuk melakukan perhitungan jarak antara setiap objek yaitu *Euclidean*, *Manhattan*, *Minkowski*, dan *Hamming Distance*. Secara garis besar, alur kerja algoritma ini terlihat melalui Gambar 2.1.



Gambar 2.1 K-Nearest Neighbors

(Sumber: (Kaushal & Koundal, 2019))

Gambar 2.1 menyajikan bahwa algoritma *K-Nearest Neighbors* diawali dengan melakukan inisialisasi terlebih dahulu, disertai dengan mendefinisikan nilai K yang akan digunakan. Selanjutnya dilakukan perhitungan jarak antara titik data yang ingin diprediksi dengan setiap titik dalam dataset pelatihan. Perhitungan jarak antar titik ini dapat menggunakan *Euclidean Distance* seperti pada persamaan 2.2.

$$d(x, y)^2 = \sum_{i=1}^n |x_i - y_i|^2 \quad (2.2)$$

Selepas selesai melakukan perhitungan jarak menggunakan persamaan 2.2, seluruh jarak tersebut akan disusun sedemikian rupa dari yang terdekat jaraknya menuju ke terjauh. Kemudian berdasarkan dengan jumlah tetangga terdekat atau nilai K yang telah ditentukan diawal (semisal $K=3$), maka dari ketiga tetangga tersebut dapat mengklasifikasikan data uji sesuai dengan mayoritas kelas tetangga terdekatnya.

Selain itu, pada algoritma *K-Nearest Neighbors* juga terdapat nilai *confidence*. Nilai *confidence* dinyatakan sebagai persentase atau probabilitas bahwa klasifikasi yang diberikan adalah benar, sesuai dengan persamaan 2.3.

$$Confidence = \frac{Max\ Count}{K} \quad (2.3)$$

Persamaan untuk mencari nilai *confidence* algoritma *K-Nearest Neighbors* seperti pada persamaan 2.3 didapatkan dari kode program penyusun *library* *Arduino_KNN.h* yang digunakan oleh penulis. Nilai *confidence* pada algoritma *K-Nearest Neighbors* didapat dari hasil pembagian antara jumlah tetangga kelas mayoritas dengan jumlah total tetangga (K). Nilai *confidence* memiliki jumlah maksimum yaitu bernilai 1.

BAB 3 METODOLOGI

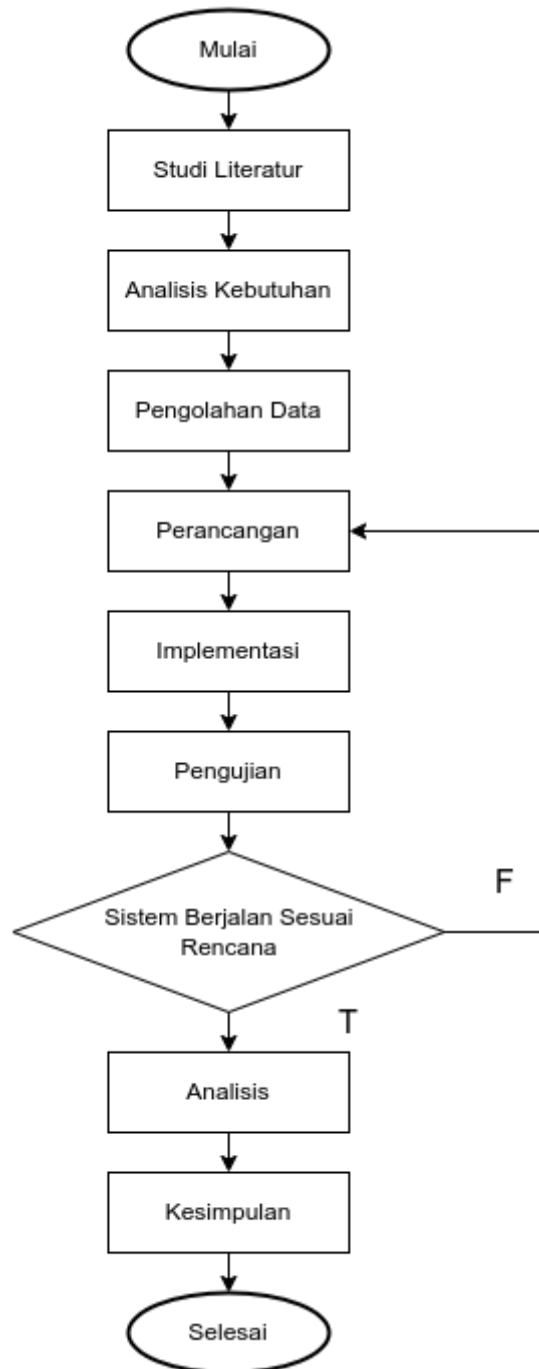
Pada bab 3 atau bab metodologi ialah bab ketiga dalam penelitian ini yang memiliki beberapa sub bab yang akan dibahas. Mula-mula bab ini akan menjelaskan terkait tipe penelitian dari penelitian penulis. Selanjutnya terdapat metode penelitian yang penulis akan gunakan yaitu disajikan dalam bentuk diagram alir penelitian. Kemudian juga membahas mengenai partisipan dalam penelitian ini. Keberadaan partisipan dalam penelitian ini sangatlah penting bagi penulis. Hal ini karena dengan adanya partisipan, penulis dapat banyak menggali informasi mengenai objek yang akan diteliti. Setelah itu, akan diberitahukan juga mengenai lokasi penelitian ini bertempat. Terakhir, penulis akan menyantumkan juga peralatan yang mendukung penelitian.

3.1 Tipe Penelitian

Tipe penelitian yang dilakukan merupakan penelitian bertipe Implementatif-Pengembangan (*Development*) dengan berfokus pada rancang bangun sistem klasifikasi kelayakan madu untuk penderita Diabetes mellitus. Implementatif yang dimaksud pada penelitian penulis adalah dengan melakukan sebuah pengujian langsung terhadap madu selaku objek penelitian. Dimana sistem dapat melakukan akuisisi data yaitu nilai kapasitansi dan warna pada madu. Sedangkan pengembangan yang dimaksudkan dalam penelitian ini adalah karena penelitian yang dilakukan oleh penulis ini juga merupakan bagian dari beberapa penelitian sebelumnya yang telah melalui tahap pengembangan.

3.2 Metode Penelitian

Metode penelitian yang penulis gunakan akan disajikan dalam bentuk diagram alir. Diagram alir ini akan menjelaskan secara spesifik terkait jalannya penelitian dari awal hingga akhir. Mula-mula penulis melakukan literasi terhadap beberapa pustaka yang dapat menunjang dan berpengaruh dengan objek penelitian. Kemudian melakukan penyusunan analisis kebutuhan untuk sistem dan pengolahan data. Selanjutnya melakukan perancangan dan implementasi perangkat lunak serta keras yang akan diperlukan. Setelah itu, tibalah pada tahap pengujian dan juga analisis penelitian yang dilakukan berdasarkan implementasi sebelumnya. Akhir dari penelitian ini dilakukan penarikan kesimpulan dan juga saran oleh penulis terhadap penelitian ini. Gambar 3.1. menyajikan metode penelitian yang digunakan oleh penulis.



Gambar 3.1 Diagram Alir Penelitian

3.2.1 Studi Literatur

Penulis mengawali tahap ini dengan melakukan literasi terhadap beberapa pustaka untuk menyusun dasar teori masing-masing kebutuhan baik fungsional dan non fungsional yang akan diaplikasikan pada sistem. Dengan melakukan studi literatur ini, penulis dapat mengimplementasikan aspek-aspek yang sudah diimplementasikan dari jurnal-jurnal yang berkaitan. Adapun beberapa aspek tersebut, yaitu meliputi jenis-jenis madu, Diabetes mellitus, pengimplementasian

metode KNN pada Arduino, penggunaan sensor kapasitansi dan sensor warna TCS3200, serta pembuatan *datalogger* untuk fitur *history* sistem.

3.2.2 Rekayasa Kebutuhan

Tahap selanjutnya yaitu dilakukan proses rekayasa kebutuhan. Proses perencanaan perangkat keras dan perangkat lunak yang nantinya akan digunakan selama proses perancangan dan implementasi sistem merupakan proses dari rekayasa kebutuhan. Adapun perangkat keras (*hardware*) yang digunakan meliputi Arduino Mega, sensor kapasitansi, sensor warna TCS3200, RTC, LCD 20x4, modul I2C, modul Micro SD, Micro SD Card. Sedangkan untuk perangkat lunak (*software*) yang digunakan yaitu Arduino IDE dan Jupyter Notebook, serta beberapa library seperti library Arduino KNN, RTC, SD, dan SPI.

3.2.3 Perancangan dan Implementasi

Kemudian tahap berikutnya adalah perancangan dan implementasi. Tahap perancangan terdiri dari perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*). Pada tahap perancangan perangkat keras dilakukan pembuatan diagram skematik terhadap seluruh komponen perangkat keras penyusun sistem, seperti Arduino Mega, LCD, RTC, modul Micro SD, dan sensor-sensor. Di tahap ini juga dilakukan perancangan *cover* pada sistem guna menunjang mobilitas sistem, serta melindungi sistem dari kerusakan seperti korsleting akibat terkena air atau zat lainnya.

Adapun tahap perancangan perangkat lunak dilakukan dengan menyusun kode program agar seluruh komponen perangkat keras yang digunakan dapat terintegrasi. Sedangkan pada tahap implementasi perangkat keras dilakukan pemasangan atau *wiring* terhadap seluruh komponen perangkat keras yang akan digunakan dalam sistem. Kemudian pada tahap implementasi perangkat lunak dilakukan dengan mengunggah kode program yang telah terintegrasi ke dalam sistem.

3.2.4 Pengambilan dan Pengolahan Data

Selanjutnya adalah tahap pengambilan data, dimana data yang diambil merupakan data primer. Data primer ini menggunakan cara pengambilan data yang dilakukan secara langsung dari beberapa sampel madu yang akan diteliti. Namun sebelum itu, beberapa sampel madu yang akan digunakan akan divalidasi terlebih dahulu oleh seorang pakar madu. Penulis membawa langsung beberapa sampel madu ke pakar madu, kemudian pakar madu tersebut akan menyicipi dan mencium aroma dari sampel madu. Selanjutnya, pakar madu akan melakukan validasi apakah madu tersebut layak atau tidak layak konsumsi.

Dataset madu yang digunakan berjumlah 30 dataset. Pada dataset madu ini terdapat empat buah fitur yaitu nilai kapasitansi, R, G, dan B. Setelah mendapatkan validasi dari pakar madu, 30 dataset madu tersebut dibagi menjadi dua buah kelas. Kelas pertama memiliki label 0 yaitu kelas "TIDAK LAYAK" dengan

jumlah 15 dataset madu. Kemudian kelas kedua yaitu kelas “LAYAK” dengan jumlah 15 dataset juga memiliki label 1.

3.2.5 Pengujian Sistem

Tahap pengujian sistem dilakukan menggunakan sepuluh sampel madu. Tahap pengujian ini berfungsi untuk mengetahui kinerja baik dari masing-masing komponen ataupun keseluruhan sistem. Sebelum melakukan pengujian, terdapat prosedur pengujian. Prosedur pengujian berfungsi agar pengujian menjadi lebih terarah dan terstruktur. Prosedur pengujian ini nantinya juga dapat bermanfaat bagi penelitian selanjutnya. Pada tahap pengujian terdapat beberapa hal yang perlu untuk diuji. Adapun beberapa pengujian yang akan diterapkan mencakup pengujian masing-masing sensor yaitu sensor kapasitansi dan TCS3200, pengujian nilai K untuk memilih nilai K atau jumlah tetangga terdekat yang terbaik, kemudian pengujian keseluruhan sistem berupa pengujian fungsional dan akurasi sistem.

3.2.6 Analisis Hasil Pengujian

Tahap analisis hasil pengujian diterapkan untuk merangkum data yang diperoleh dari hasil pengujian yang telah dikerjakan sebelumnya. Dimana penulis data menghubungkan hasil dengan suatu persamaan yang berlandaskan pada dasar teori.

3.2.7 Kesimpulan dan Saran

Pada tahapan akhir dari penelitian ini, penulis akan menarik suatu kesimpulan yang akan menjawab rumusan masalah dari penelitian ini. Kesimpulan tersebut juga didapat berdasarkan hasil dari pengujian yang telah diperoleh. Selain itu, terdapat juga beberapa saran dari penulis yang harapannya agar penelitian dengan topik yang sama dapat dikembangkan lagi di masa mendatang.

3.3 Stakeholder Penelitian

Stakeholder pada penelitian ini adalah masyarakat penderita Diabetes mellitus. Sistem ini dirancang khusus untuk penderita Diabetes mellitus karena sesuai dengan fitur-fitur yang diperlukan seperti dapat melakukan klasifikasi kelayakan madu dan juga adanya fitur history yang dapat data pengukuran madu terakhir dilakukan beserta jumlah konsumsi harian dan total madu.

3.4 Partisipan Penelitian

Partisipan pada penelitian ini merupakan seorang pakar madu di Peternakan Lebah, Kabupaten Lawang, Provinsi Jawa Timur. Beliau menyebutkan bahwasanya sudah berkecimpung dalam dunia peternakan lebah dari tahun 1990-an, mengikuti jejak sang ayah. Dengan kata lain beliau sudah berpengalaman selama kurang lebih 20 tahun pada bidang ini, sehingga sudah dapat dikatakan kredibel. Peran dari beliau sebagai pakar madu ini meliputi validasi dataset sampel madu dan diskusi hal-hal seputar madu yang belum diketahui oleh penulis.

3.5 Lokasi Penelitian

Penelitian yang dilakukan penulis berlokasi di Peternakan Lebah, Kabupaten Lawang, Provinsi Jawa Timur. Lokasi penelitian ini ditentukan oleh penulis karena jarak dari Kota Malang dengan Kabupaten Lawang yang masih tergolong mudah untuk dijangkau. Selain itu, penulis juga sudah mengadakan janji dengan pakar madu terlebih dahulu dan sudah berdiskusi singkat melalui *whatsapp* mengenai penelitian ini.

3.6 Peralatan Pendukung Yang Digunakan

Sistem ini tersusun dari beberapa peralatan yang mendukung dan sangat diperlukan. Peralatan pendukung yang digunakan terbagi menjadi dua tipe yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Adapun perangkat keras dan perangkat lunak yang akan digunakan meliputi:

- Perangkat Keras:
 - Arduino Mega
 - Sensor Kapasitansi
 - Sensor TCS3200
 - LCD 20x4
 - Real Time Clock (RTC)
 - Modul I2C
 - Modul Micro SD
 - SD Card
 - Tiga buah button
- Perangkat Lunak:
 - Arduino IDE
 - Jupyter Notebook
 - Pustaka Arduino_KNN.h
 - Pustaka RTCLib.h
 - Pustaka LiquidCrystal_I2C.h
 - Pustaka Wire.h
 - Pustaka SPI.h
 - Pustaka SD.h
 - Pustaka EEPROM.h

BAB 4 REKAYASA KEBUTUHAN

Pada bab 4 atau bab rekayasa kebutuhan ialah bab keempat dalam penelitian ini yang berisikan uraian terkait dengan pengkajian masalah secara umum, mengidentifikasi stakeholder, menyusun keperluan fungsional beserta spesifikasi yang harus dimiliki sistem, juga menganalisis keperluan perangkat keras dan lunak penyusun sistem.

4.1 Kajian Masalah

Sistem ini berupa sebuah alat yang dapat melakukan klasifikasi kelayakan madu berdasarkan kadar gula dan warna. Pemroses sistem ini adalah Arduino Mega yang sudah dihubungkan dengan dua buah sensor. Sensor pertama menggunakan sensor kapasitansi untuk mengukur kadar gula dari madu. Sensor kapasitansi ini terdiri dari dua buah plat konduktor yang berukuran sama, dimana nantinya kedua plat konduktor ini akan sesuai dengan prinsip kerja kapasitor keping sejajar. Sensor kedua yaitu sensor TCS3200 yang memiliki kemampuan yaitu dapat mendeteksi warna merah, hijau, dan biru (RGB). Kemudian input dari kedua buah sensor ini akan melalui suatu proses oleh Arduino Mega yang memanfaatkan algoritma *K-Nearest Neighbors* untuk dilakukan klasifikasi. Sehingga nantinya akan mengeluarkan output berupa dua buah kelas yaitu "Layak" dan "Tidak Layak". Output dari hasil klasifikasi tersebut selanjutnya akan diperlihatkan melalui LCD. Selain itu, sistem ini juga menggunakan modul RTC dan Micro SD. Modul RTC digunakan untuk mengolah dan menyimpan waktu. Sedangkan modul Micro SD digunakan untuk menyimpan keluaran sistem beserta tanggal dan waktunya. Sesuai dengan kegunaannya, pada sistem ini modul RTC dan Micro SD berfungsi sebagai komponen yang dapat menyediakan fitur history. Fitur history ini memungkinkan pengguna sistem dapat mengetahui riwayat penggunaan sistem ini sebelum mengonsumsi madu.

4.2 Identifikasi Stakeholder

Sistem dengan Metode *K-Nearest Neighbors* Berbasis Arduino Mega dirancang untuk digunakan oleh *user* dengan keperluan untuk mengetahui kelayakan pada madu. *User* yang menggunakan termasuk kedalam golongan masyarakat umum, tetapi lebih dikhususkan untuk masyarakat yang menderita penyakit Diabetes mellitus atau masyarakat pecinta madu namun tetap ingin mengonsumsi madu tanpa harus khawatir akan kadar gula berlebih.

4.3 Kebutuhan Fungsional

Pada kebutuhan fungsional sistem akan menjelaskan fungsi atau fitur yang dapat dikerjakan sistem agar sistem yang dirancang dapat bekerja sesuai dengan kemauan penulis. Adapun beberapa kebutuhan fungsional sistem yaitu:

1. Sistem dapat mengakuisisi data melalui sensor kapasitansi dan sensor TCS3200.

Kedua sensor yang dipakai dalam sistem harus dapat berfungsi dengan baik. Hal ini karena dari data yang diambil dari kedua sensor inilah yang nantinya akan menjadi dataset yang digunakan untuk melakukan *training* pada algoritma *machine learning* yang akan digunakan oleh penulis yaitu *K-Nearest Neighbors*

2. Sistem dapat melakukan klasifikasi kelayakan madu menggunakan algoritma *K-Nearest Neighbors*.

Sistem harus dapat melakukan klasifikasi kelayakan madu menggunakan algoritma *K-Nearest Neighbors* dengan benar. Ada banyak faktor yang dapat mempengaruhi hasil klasifikasi atau tingkat akurasi pada algoritma *K-Nearest Neighbors* seperti penentuan nilai *K* yang digunakan, banyak dataset yang digunakan, pengolahan dataset, dan tingkat akurasi sensor dalam melakukan pengambilan data.

3. Sistem dapat menyimpan *history* hasil klasifikasi sebelumnya.

History atau datalogger pada sistem digunakan untuk menyimpan jumlah konsumsi harian dan total madu oleh penderita Diabetes mellitus. Hal ini diperlukan karena dapat menjadi acuan bagi penderita Diabetes mellitus untuk mempertimbangkan jumlah konsumsi madu per harinya dengan lebih bijak.

4. Sistem dapat menggeser opsi menu yang tertera pada LCD menggunakan *push button* yang tersedia.

Pada sistem dilengkapi dengan tiga buah *push button* yang digunakan untuk menggeser sub menu pada LCD. *Button* hijau digunakan untuk memilih opsi ke arah atas, *button* merah digunakan untuk memilih opsi ke arah bawah, dan *button* kuning digunakan untuk memilih opsi yang diinginkan. Terdapat empat sub menu yang dapat dipilih pada layar LCD yaitu menu klasifikasi, *history*, *reset* harian, dan *reset* total.

5. Sistem dapat menampilkan hasil klasifikasi dan *history* kelayakan madu pada LCD.

Liquid Crystal Display (LCD) berfungsi sebagai layar atau tampilan utama sistem agar lebih nyaman untuk dilihat mata. Pada layar LCD, pengguna dapat melihat hasil klasifikasi dari madu yang diukur berupa kelas "LAYAK" dan "TIDAK LAYAK" bagi penderita Diabetes mellitus. Kemudian pengguna juga dapat melihat informasi pada sub menu *history* yang menampilkan tanggal dan waktu terakhir kali melakukan pengukuran, kemudian jumlah konsumsi harian dan jumlah konsumsi total madu.

4.4 Spesifikasi Sistem

Sistem tersusun atas dua buah sensor yaitu sensor kapasitansi dan sensor TCS3200. Oleh karena itu, untuk tata letaknya saat perancangan harus disesuaikan dengan sampel uji. Hal ini karena kedua sensor tersebut akan bekerja secara maksimal ketika berada dekat dengan sampel uji. Untuk tingkat kompleksitas kode

program mungkin dapat dikatakan cukup kompleks, karena terdapat empat buah menu yang mana keempat menu tersebut juga mempunyai kegunaan yang berbeda-beda. Walaupun cukup kompleks karena kode program mencapai kurang lebih 500 baris, sistem dapat tetap melakukan perhitungan dengan waktu komputasi yang tidak terlalu lama. Agar sistem dapat diimplementasikan dengan baik dan berfungsi secara optimal, spesifikasi sistem yang harus terpenuhi yaitu:

1. Sistem harus ditutupi oleh *cover* atau *casing* agar mudah ketika ingin dipindahkan ke tempat lain
2. Sistem harus memiliki komponen-komponen penunjang tujuan utama dalam pembuatan sistem, yaitu seperti sensor kapasitansi, TCS3200, RTC, Modul Micro SD, dan LCD
3. Sistem harus dapat mengetahui kelayakan konsumsi madu yang diuji. Sehingga memerlukan mikrokontroler yang menunjang untuk melakukan penyimpanan dan perhitungan data dalam jumlah besar.
4. Sistem harus dapat menyimpan data hasil pengukuran klasifikasi sebelumnya sebagai acuan untuk penderita Diabetes mellitus mempertimbangkan takaran konsumsi kadar gula per harinya.

4.5 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Kebutuhan perangkat keras dan lunak dapat disebut juga sebagai kebutuhan non-fungsional. Kebutuhan ini juga sangat diperlukan sebagai penyusun kebutuhan sistem dalam melaksanakan kinerjanya.

4.5.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras sangat perlu disesuaikan dengan masing-masing komponen atau perangkat penyusun sistem. Hal ini bertujuan untuk mencocokkan masing-masing komponen sesuai dengan spesifikasinya. Adapun kebutuhan perangkat keras yang akan digunakan oleh penulis antara lain:

4.5.1.1 Arduino Mega

Arduino Mega merupakan pemroses utama dari sistem ini. Arduino Mega, yang didasarkan pada mikrokontroler ATmega2560, memiliki lebih banyak pin I/O, memori program, dan daya komputasi daripada model Arduino lainnya. Arduino Mega banyak digunakan dalam proyek robotika, automasi, kendali industri, dan proyek skala besar yang membutuhkan lebih banyak I/O dan sumber daya pemrosesan karena fitur-fiturnya yang kuat. Arduino Mega dapat dijalankan dengan mengunggah kode program pada Arduino Mega untuk mengontrol berbagai perangkat elektronik seperti sensor, motor, lampu, dan lainnya dengan menggunakan software Arduino IDE (*Integrated Development Environment*). Gambar 4.1 menggambarkan tampilan fisik Arduino Mega, dan Tabel 4.1 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.1 Arduino Mega

(Sumber: (More, et al., 2019))

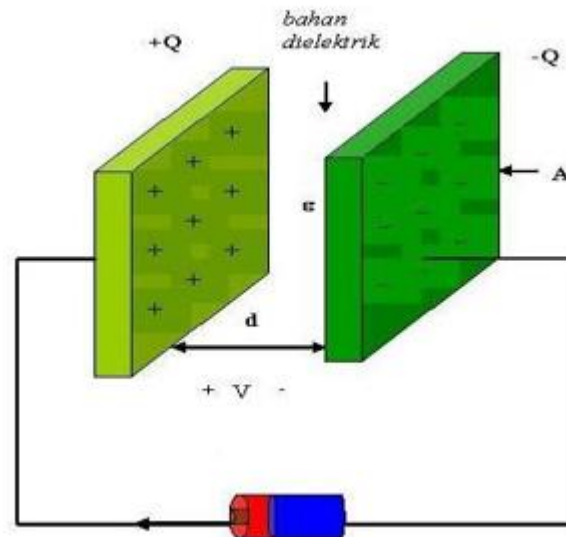
Tabel 4.1 Spesifikasi Arduino Mega

Aspek	Spesifikasi
Mikrokontroler	Atmega2560
Arus DC Per I/O Pin	20 mA
Pin I/O Digital PWM	15 buah
Pin Analog	16 buah
Pin I/O Digital	54 buah
Kecepatan Clock	16 MHz
LED_BUILTIN	13
EEPROM	4 KB
Massa	37 gram
SRAM	8 KB
Ukuran	101,52 x 53,3 mm
Flash Memory	256 KB
Tegangan Operasi	5 V
Batas Tegangan Input	6-20 V

4.5.1.2 Sensor Kapasitansi

Sensor kapasitansi adalah sensor yang dibuat menggunakan prinsip kerja dari kapasitor keping sejajar. Sensor Kapasitansi terdiri dari dua buah pelat konduktor logam sejajar yang dipisahkan oleh bahan dielektrik ditengahnya. Ketika ada perbedaan potensial tertentu antara dua buah pelat, medan elektrostatis akan didistribusikan diantara keduanya. Medan listrik akan tersebar seragam di area antara kedua buah pelat.

Sensor Kapasitansi digunakan untuk mengukur besarnya muatan listrik atau kapasitansi pada suatu objek. Kapasitansi adalah kemampuan suatu objek untuk menyimpan muatan listrik ketika diberikan tegangan. Nilai kapasitansi diukur dalam satuan farad (F), yang menunjukkan jumlah muatan yang dapat disimpan kapasitor untuk setiap volt tegangan yang diterapkan. Semakin tinggi nilai kapasitansi kapasitor, berarti semakin banyak muatan yang dapat disimpan. Gambar 4.2 menggambarkan tampilan fisik sensor kapasitansi, dan Tabel 4.2 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.2 Sensor Kapasitansi

(Sumber: (Purnamasari, 2017))

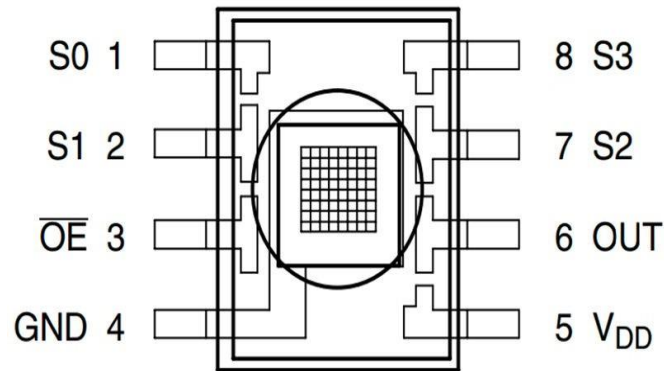
Tabel 4.2 Spesifikasi Sensor Kapasitansi

Aspek	Spesifikasi
Bahan	Dua buah plat konduktor dengan ukuran yang sama
Posisi	Sejajar
Panjang	4 cm
Lebar	4 cm
Luas Penampang	16 cm ²

4.5.1.3 Sensor TCS3200

Sensor TCS3200 adalah sebuah sensor warna yang digunakan untuk melakukan deteksi dan pengukuran intensitas frekuensi warna pada suatu objek. Sensor TCS3200 mampu mengidentifikasi warna dasar yaitu merah (R), hijau (G), dan biru (B). Selain itu, sensor ini juga dapat menghasilkan data frekuensi yang berkaitan dengan tingkat kecerahan warna tersebut.

Sensor TCS3200 terdiri dari beberapa fotodiode dan filter warna yang ditempatkan diatas sensor. Filter warna memungkinkan sensor ini untuk merespon cahaya yang masuk dalam rentang spektrum warna tertentu. Ketika cahaya masuk ke sensor, fotodiode akan mengubah intensitas cahaya menjadi sinyal listrik yang dapat diukur. Gambar 4.3 menggambarkan tampilan fisik sensor TCS3200, dan Tabel 4.3 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.3 Sensor Warna TCS3200

(Sumber: (More, et al., 2019))

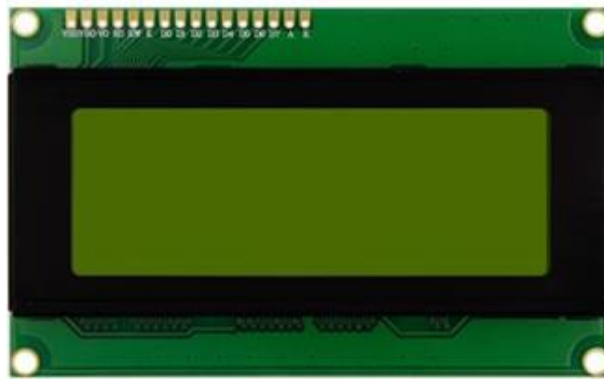
Tabel 4.3 Spesifikasi Sensor TCS3200

Aspek	Spesifikasi
Seri	TCS3200
Tegangan Kerja	2,7 – 5,5 V
Dimensi	28,4 x 28,4 mm
Jumlah Pin	8 buah

4.5.1.4 LCD 20x4

Liquid Crystal Display (LCD) 20x4 ialah jenis layar berkarakter yang memakai teknologi *liquid crystal*. LCD sering digunakan untuk menampilkan suatu teks, perhitungan, dan hasil dari suatu fungsi tertentu. LCD 20x4 memiliki 20 baris karakter yang membentang dari kiri hingga kanan, serta terdapat 4 kolom karakter yang membentang dari atas ke bawah.

Kelebihan dari layar 20x4 ini adalah kemampuan untuk menampilkan teks dengan jelas, cukup banyak karakter yang dapat diperlihatkan, dan kompatibilitas yang luas dengan berbagai platform mikrokontroler. Kemudian layar dari LCD juga memiliki pencahayaan *backlight* yang dapat diatur, sehingga memudahkan pembacaan teks dalam kondisi pencahayaan yang rendah. Tampilan layar LCD 20x4 dapat diatur menggunakan mikrokontroler dilengkapi dengan antarmuka komunikasi seperti I2C (*Inter-Integrated Circuit*) atau parallel agar mempermudah *wiring*. Gambar 4.4 menggambarkan tampilan fisik LCD 20x4, dan Tabel 4.4 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.4 LCD 20x4

(Sumber: (Stefanov, et al., (2020))

Tabel 4.4 Spesifikasi LCD 20x4

Aspek	Spesifikasi
Format Tampilan	20 karakter x 4 baris
Ukuran Karakter	2.95 x 4,75 mm
Ukuran dimensi LCD	98 x 60 x 13,6 mm
Operating Temperature	-20 sampai 70°C

4.5.1.5 I2C

Inter-Integrated Circuit (I2C) ialah suatu alat elektronika yang bertugas untuk menyederhanakan input dari LCD. Dimana yang semula LCD membutuhkan 16 input kabel, dengan menggunakan komponen I2C hal tersebut dapat disederhanakan menjadi hanya 4 buah input saja. I2C menggunakan arsitektur komunikasi *master-slave*. Artinya ada satu perangkat yang berfungsi sebagai *master* dan mengontrol komunikasi, sedangkan perangkat lain berfungsi sebagai *slave* dan menerima instruksi dari master. I2C memiliki SDA dan SCL sebagai dua jalur komunikasinya. *Serial Data Line* (SDA) bertugas mengirim dan menerima data, sedangkan *Serial Clock Line* (SCL) bertugas mengatur sinkronisasi waktu dalam komunikasi. Gambar 4.5 menggambarkan tampilan fisik I2C, dan Tabel 4.5 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.5 I2C

(Sumber: (Stefanov, et al., (2020))

Tabel 4.5 Spesifikasi I2C

Aspek	Spesifikasi
Jumlah Pin	4 buah (VCC, GND, SDA, SCL)
Device Address	0x27 atau 0x3F
Ukuran	41,5x19x15,3 mm

4.5.1.6 RTC

Real-Time Clock (RTC) bertipe DS3231 ialah alat elektronika yang bertugas untuk pengukuran waktu secara *real-time*. Ketika sistem tidak mendapatkan daya, maka komponen inilah yang nantinya akan terus menghitung waktu agar tetap berjalan. Hal tersebut terjadi karena RTC DS3231 memiliki rangkaian baterai cadangan internal yang memungkinkan modul tetap menjaga waktu yang tepat bahkan ketika daya utama terputus. Rangkaian baterai ini secara otomatis mengaktifkan RTC dan mempertahankan waktu yang akurat.

RTC DS3231 juga memiliki tingkat presisi yang sangat tinggi dalam mengukur waktu. Kesalahan presisi harian yang dihasilkan oleh DS3231 hanya dalam kisaran beberapa detik per tahunnya. Hal ini membuat RTC DS3231 dapat dikatakan sangat akurat dalam hal pengukuran waktu. Gambar 4.6 menggambarkan tampilan fisik RTC DS3231, dan Tabel 4.6 memuat spesifikasi untuk informasi lebih lanjut.



Gambar 4.6 RTC DS3231

(Sumber: (Mnati, et al., 2021))

Tabel 4.6 Spesifikasi RTC

Aspek	Spesifikasi
Seri	RTC DS3231
Tegangan Operasi	2,3 V – 5,5 V
Konsumsi Arus	500 nA

Tegangan MAX pada Pin SDA dan SCL	VCC + 0,3 V
Operating Temperature	-45°C sampai +80°C
Jumlah Pin	6 buah
Baterai	CR2032

4.5.1.7 Modul Micro SD

Pemrosesan Micro SD Card dan penyimpanan data ditangani oleh Modul Micro SD. Menggunakan sistem antarmuka *Serial Parallel Interface* (SPI), Modul Micro SD memiliki akses ke kartu micro SD untuk operasi baca dan tulis data. Oleh karena itu, Modul Micro SD baru dapat berjalan ketika telah ditambahkan akses *library* SD.h dan SPI.h pada Arduino IDE. Modul Micro SD juga memiliki konektor yang sesuai dengan kartu memori mikro SD, sehingga mempermudah pemasangan kartu memori mikro SD ke dalam modul. Gambar 4.7 menggambarkan tampilan fisik Modul Micro SD, dan Tabel 4.7 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.7 Modul Micro SD

(Sumber: (Mnati, et al., 2021))

Tabel 4.7 Spesifikasi Modul Micro SD

Aspek	Spesifikasi
<i>Card Supported</i>	Micro SD Card (up to 2 GB) dan Micro SDHC (up to 32 GB)
Tegangan Operasi	4,5 V – 5,5 V
Konsumsi Arus	0,2 mA – 200 mA
<i>Logic Level</i>	3,3 V atau 5 V
Jumlah Pin	6 buah
Antarmuka	VCC, GND, SS, MISO, MOSI, dan CLK

4.5.1.8 Micro SD Card

Micro SD Card merupakan kartu memori bersifat *non-volatile* yang dipergunakan dalam perangkat portabel. *Non-volatile* artinya dapat menyimpan data ketika komponen tidak terhubung catu daya. Kartu ini nantinya akan digunakan untuk menyimpan history hasil pengukuran klasifikasi pada sistem. Gambar 4.8 menggambarkan tampilan fisik Micro SD Card, dan Tabel 4.8 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.8 Micro SD Card

(Sumber: <https://v-gen.co.id/flashmemory/microsd-8gb-turbo/>)

Tabel 4.8 Spesifikasi Micro SD Card

Aspek	Spesifikasi
Brand	V-GeN
Speed	14,8 MB/s
Ukuran Memori	8 GB
Tipe	<i>Secure Digital Card</i>
Suhu saat beroperasi	0 – 60°C
Dimensi	32 x 24 x 2,1 mm
Keamanan	Built-in White Protect
Ketahanan	Magnetproof, Shockproof, Temperature Extremes, Waterproof, X-Rayproof

4.5.1.9 Push Button

Push button atau tombol ialah sebuah alat elektronik yang bertugas sebagai saklar sementara yang diaktifkan saat ditekan dan kembali ke posisi semula setelah dilepaskan. *Push button* terdiri dari sebuah tombol yang dapat ditekan dan mekanisme pegas di dalamnya. Ketika tombol ditekan, mekanisme pegas akan menghubungkan arus listrik. Sedangkan ketika tombol dilepas, koneksi arus listrik akan terputus kembali. Gambar 4.9 menggambarkan tampilan fisik Push Button, dan Tabel 4.9 memuat spesifikasinya untuk informasi lebih lanjut.



Gambar 4.9 Push Button

(Sumber: <https://i.ebayimg.com/images/g/PssAAOSwsZJbT6du/s-l1600.jpg>)

Tabel 4.9 Spesifikasi Push Button

Aspek	Spesifikasi
Arus/Tegangan Maksimal	3 A / 250 V
Jumlah Pin	2 pin
Diameter	12 mm

4.5.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak juga sangat diperlukan dalam pembuatan sistem. Perangkat lunak digunakan untuk mengoperasikan komponen-komponen pada sistem berupa kode program. Kebutuhan perangkat lunak yang diperlukan ialah:

4.5.2.1 Arduino IDE

Arduino Integrated Development Environment (IDE) ialah suatu platform yang bertugas mengirim perintah atau kode program untuk kemudian diproses melalui perangkat keras Arduino Mega. Arduino IDE ini tidak hanya terbatas pada board Arduino Mega saja. Melainkan juga dapat digunakan untuk berbagai jenis dari perangkat keras Arduino lainnya, seperti Arduino Nano, Arduino Uno, Arduino Mini, dan lainnya.

Arduino IDE menyediakan antarmuka yang mudah digunakan dan mudah dipahami. Platform ini berbasis C/C++ sebagai bahasa pemrogramannya, namun terdapat beberapa perubahan dan tambahan sintaks khusus untuk Arduino. Sehingga dapat digunakan untuk menulis dan mengedit kode program dalam editor teks, serta mengakses berbagai fitur dan menu untuk mengatur pengaturan proyek, mengelola pustaka, dan mengunggah kode ke papan Arduino. Pada Gambar 4.10 menyajikan tampilan dari logo Arduino IDE.



Gambar 4.10 Arduino IDE

(Sumber: https://content.arduino.cc/assets/arduino_logo_1200x630-01.png)

4.5.2.2 Jupyter Notebook

Jupyter Notebook ialah aplikasi *website* berbasis *open-source* yang dapat membuat dokumen secara interaktif yang berisi kode live, visualisasi, dan banyak teks naratif. Jupyter Notebook tersusun atas sel-sel yang dapat diisi dengan kode atau teks naratif. Setiap sel dapat dieksekusi secara independen, memungkinkan pengguna untuk menjalankan kode secara bertahap atau mengorganisir dokumen notebook dengan lebih baik. Jupyter Notebook memungkinkan pengguna untuk menjalankan kode langsung di dalam sel dengan mengklik tombol "Run". Hasil dari kode yang dieksekusi akan diperlihatkan dibawah sel, sehingga memungkinkan pengguna untuk melihat output secara interaktif. Selain itu, penulis menggunakan Jupyter Notebook karena sebagai alternatif dari penggunaan Visual Studio Code. Hal ini karena dengan Jupyter Notebook yang bersifat *open-source*, sehingga penulis tidak perlu lagi melakukan instalasi *software*. Pada Gambar 4.11 ialah tampilan logo Jupyter Notebook.



Gambar 4.11 Jupyter Notebook

(Sumber: <https://jupyter.org/assets/share.png>)

4.5.2.3 Pustaka *Arduino_KNN.h*

Arduino KNN Library digunakan agar penulis dapat melakukan klasifikasi menggunakan algoritma K-Nearest Neighbors. Dengan menggunakan library ini, penulis dapat melakukan klasifikasi algoritma K-Nearest Neighbors hanya dengan memasukkan dataset, fitur, jumlah kelas klasifikasi, dan input jumlah K atau tetangga terdekat.

4.5.2.4 Pustaka *LiquidCrystal_I2C.h*

Library I2C LCD ialah sebuah pustaka yang tidak terdapat di perangkat lunak Arduino IDE secara otomatis. Library ini berfungsi untuk mempermudah pemanggilan fungsi atau perintah yang akan diperlihatkan pada LCD. Perintah atau fungsi yang sering digunakan misalnya seperti *lcd.backlight()*, *lcd.clear()*, *lcd.setCursor()*, *lcd.init()*, dan *lcd.print()*.

4.5.2.5 Pustaka *RTClib.h*

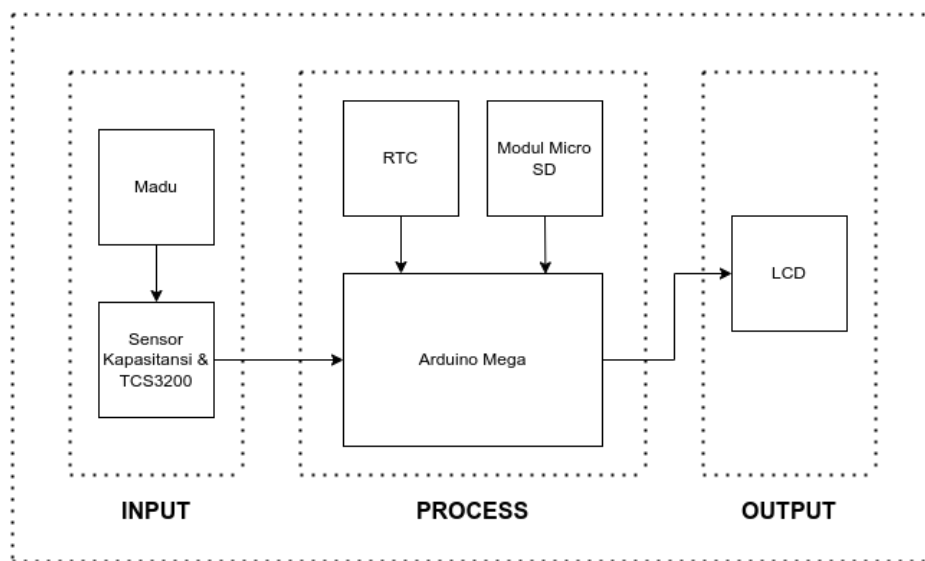
Library RTC adalah suatu library yang tidak terdapat pada perangkat lunak Arduino IDE secara otomatis. Untuk mengaksesnya dapat dilakukan dengan cara mengunduh file dari library tersebut di internet melalui web browser. RTC Library diperlukan agar penulis dapat menggunakan perintah atau fungsi untuk melakukan akses pada modul RTC DS3231. Baik perintah maupun fungsi yang sering digunakan misalnya seperti fungsi untuk menyimpan waktu pada komputer penulis ke dalam RTC.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab 5 atau bab perancangan dan implementasi ialah bab kelima dalam penelitian ini yang berisikan uraian terkait dengan bagaimana perancangan perangkat keras dan lunak penyusun sistem. Setelah merancang skema, diperlukan implementasi untuk menerapkan rancangan tersebut.

5.1 Perancangan

Dalam bentuk diagram blok yang disajikan melalui Gambar 5.1, bagian perancangan ini akan memberikan penjelasan lebih mendalam tentang prosedur perancangan sistem yang akan diimplementasikan.



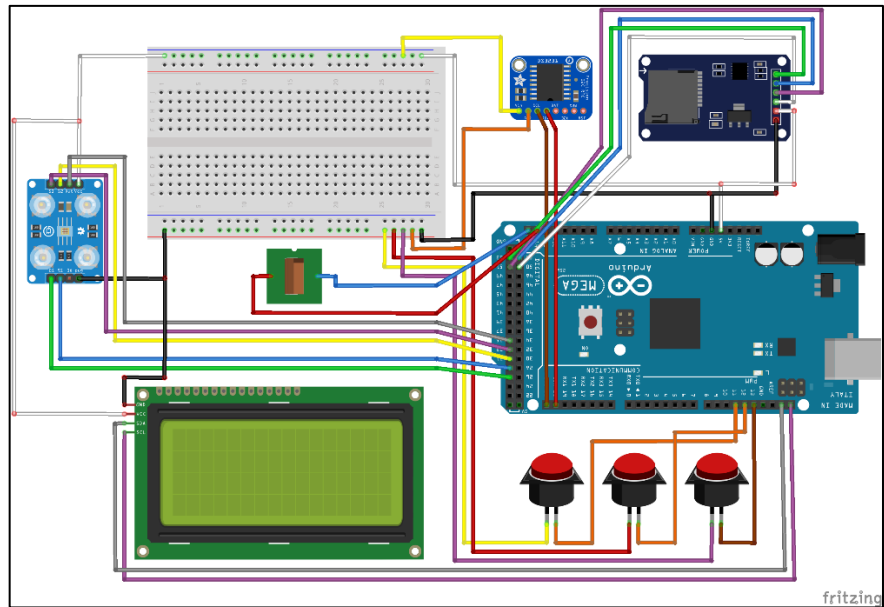
Gambar 5.1 Diagram Blok Perancangan Sistem

(Sumber: Dokumen Pribadi)

Dilihat dari Gambar 5.1, mula-mula sistem akan mendapatkan nilai input dari sensor kapasitansi dan juga sensor TCS3200. Selanjutnya, nilai input tadi akan melalui tahap proses pada Arduino Mega memanfaatkan algoritma K-Nearest Neighbors. Setelah melalui proses perhitungan dalam Arduino Mega, sistem akan menampilkan keluaran pada layar LCD berupa menu dengan beberapa opsi yaitu klasifikasi, *history*, *reset* harian, dan *reset* total.

5.1.1 Perancangan Perangkat Keras

Tahap dimana penulis membuat rancangan sistem agar saling terkoneksi antar komponen yang satu dengan lainnya. Adapun komponen-komponen yang dimaksud yaitu Arduino Mega, sensor kapasitansi, sensor TCS3200, RTC, Modul *Micro SD*, dan LCD. Untuk lebih jelasnya, rangkaian skematik antar komponen tersebut disajikan melalui Gambar 5.2.



Gambar 5.2 Skematik Perancangan Sistem

(Sumber: Dokumen Pribadi)

Gambar 5.2 disusun dengan menyesuaikan *wiring* kabel pada tabel konfigurasi pin yang disajikan melalui Tabel 5.1.

Tabel 5.1 Konfigurasi Pin

Pin Arduino	Pin Komponen	Warna	Komponen
GND	GND	Hitam	LCD I2C
VCC	VCC	Putih	
SDA	SDA	Abu-Abu	
SCL	SCL	Ungu	
GND	GND	Oranye	RTC
VCC	VCC	Kuning	
SDA	SDA	Merah	
SCL	SCL	Coklat	
GND	GND	Hitam	Micro SD
VCC	VCC	Putih	
51	MOSI	Ungu	
50	MISO	Putih Pudar	
52	SCK	Biru	
53	CS	Hijau	

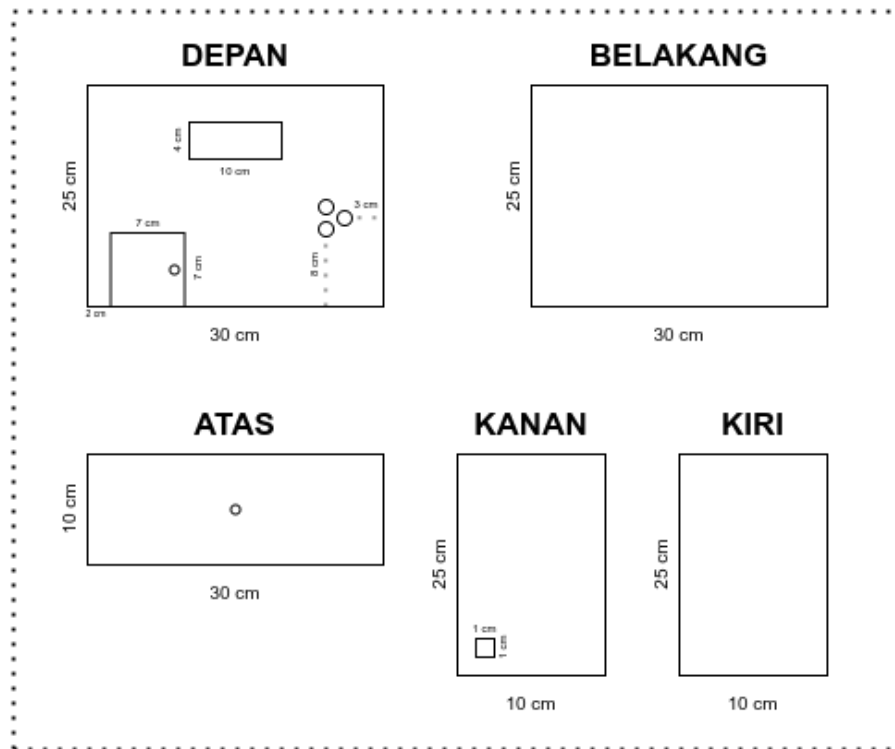
Tabel 5.2 Konfigurasi Pin (Lanjutan)

13	VCC	Coklat	Button OK
GND	GND	Ungu	
12	VCC	Oranye	Button Bawah
GND	GND	Merah	
11	VCC	Oranye	Button Atas
GND	GND	Kuning	
27	S0	Hijau	Sensor TCS3200
29	S1	Biru	
31	S2	Kuning	
33	S3	Ungu	
35	sensorOut	Abu-Abu	
VCC	VCC	Putih	
GND	GND	Hitam	
A14	OUT_PIN	Merah	
A15	IN_PIN	Biru	

Tabel 5.1 berisikan daftar seluruh *wiring* yang bertugas menjembatani setiap komponen pada sistem dengan menggunakan kabel jumper. Kabel jumper yang digunakan memiliki panjang yang bervariasi tergantung dari jarak antara pin pada papan Arduino Mega dengan pin setiap komponen.

5.1.1.1 Perancangan Cover

Penulis juga melakukan perancangan *cover* atau *casing* yang berguna untuk melindungi sistem. *Cover* atau *casing* juga berfungsi agar sistem dapat dengan mudah dibawa kemana-mana (portabel). *Cover* yang penulis gunakan sebagai wadah dari sistem ini terbuat dari bahan yang solid, yaitu akrilik. Akrilik yang digunakan memiliki ketebalan 3 mm dan berwarna hitam pekat yang tidak tembus cahaya. Hal ini dilakukan oleh penulis karena pada sistem ini menggunakan sensor warna TCS3200. Sensor ini sangat peka terhadap pencahayaan yang terang maupun redup. Maka dari itu diperlukan *cover* yang tidak tembus cahaya agar sensor ini dapat bekerja dengan maksimal dalam mengidentifikasi setiap warna pada madu. Adapun rancangan dari *cover* atau *casing* disajikan melalui Gambar 5.3.

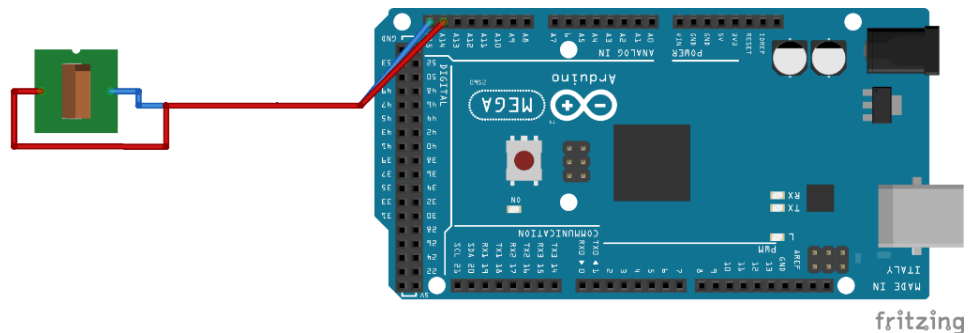


Gambar 5.3 Perancangan Cover Sistem

(Sumber: Dokumen Pribadi)

5.1.1.2 Perancangan Sensor Kapasitansi

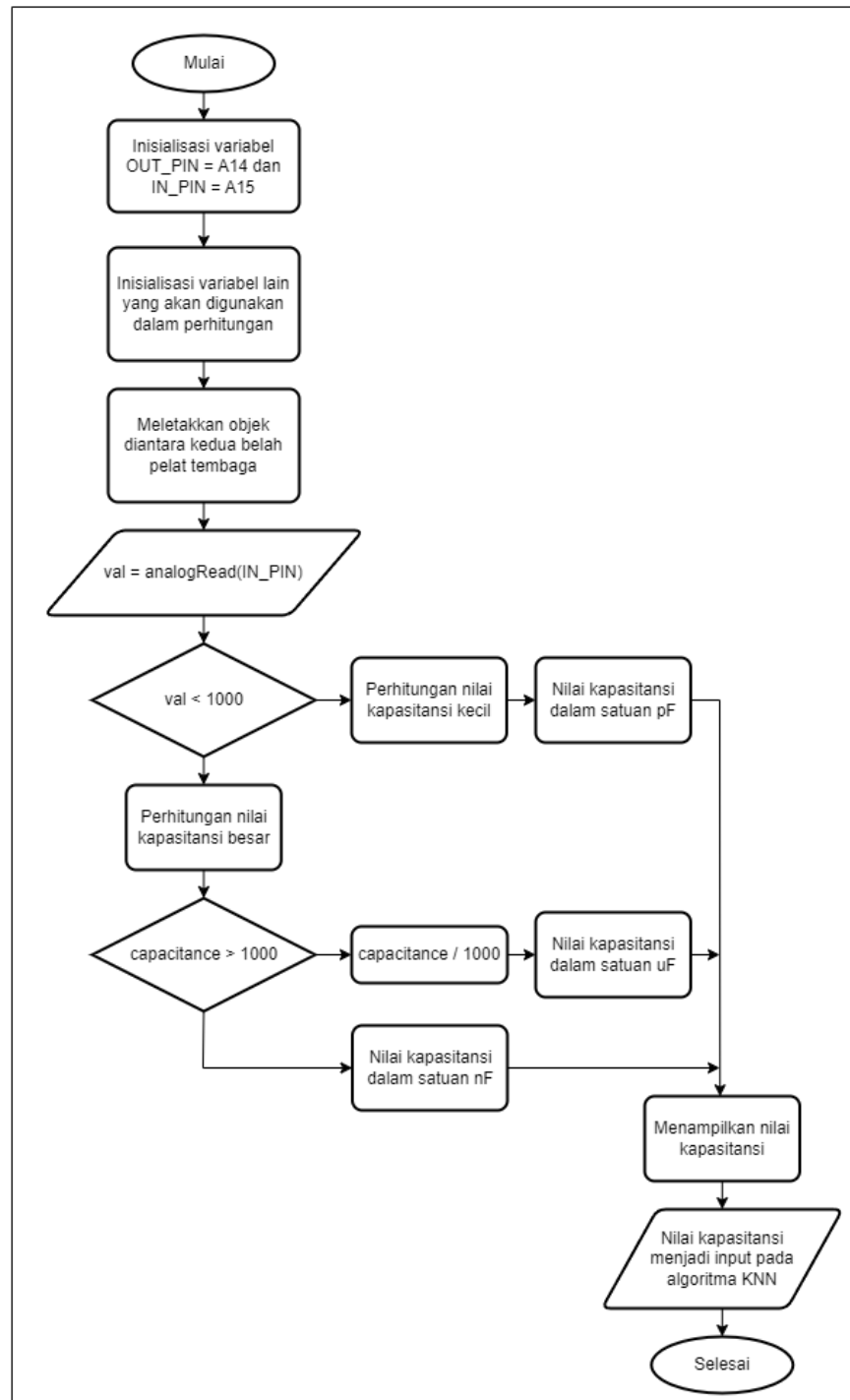
Pada tahap ini, terdapat perancangan sensor kapasitansi terhadap Arduino Mega yang disajikan melalui Gambar 5.4.



Gambar 5.4 Perancangan Sensor Kapasitansi

(Sumber: Dokumen Pribadi)

Pada Gambar 5.4, sensor kapasitansi yang terbuat menggunakan prinsip kapasitor keping sejajar terhubung pada pin analog (A14 dan A15) Arduino Mega sesuai dengan konfigurasi pin pada Tabel 5.2. Sensor kapasitansi digunakan untuk menghitung nilai kapasitansi pada suatu objek. Adapun teknis pembacaan sensor oleh sensor kapasitansi akan disajikan melalui Gambar 5.5.



Gambar 5.5 Diagram Alir Sensor Kapasitansi

(Sumber: Dokumen Pribadi)

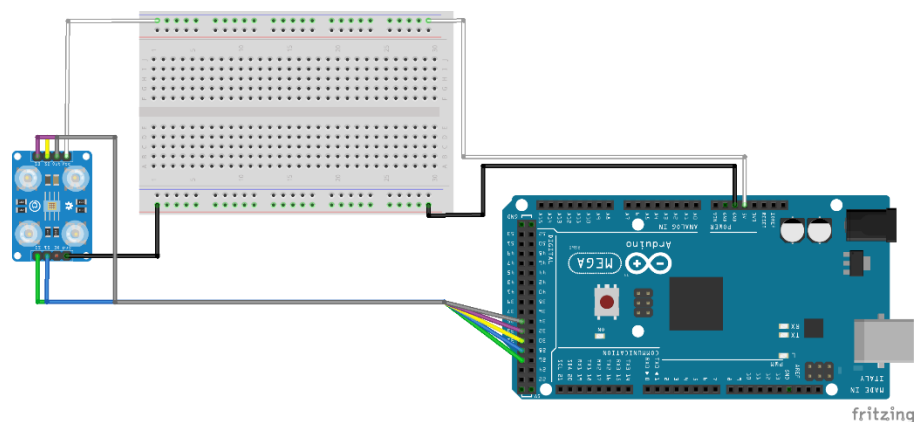
Gambar 5.5 menyajikan teknis pembacaan sensor oleh sensor kapasitansi. Perhitungan nilai kapasitansi dapat digunakan menggunakan persamaan 5.1.

$$capacitance = \frac{val * IN_CAP_TO_GND}{(MAX_ADC_VALUE - val)} \quad (5.1)$$

Nilai variabel *val* diambil dari nilai analog sensor kapasitansi. Kemudian nilai variabel *IN_CAP_TO_GND* dan *MAX_ADC_VALUE* diinisialisasikan berturut-turut 24,48 dan 1023. Terdapat tiga buah satuan yang dapat menjadi keluaran dari perhitungan sensor ini yaitu nilai kapasitansi dalam pikoFarad (pF), mikroFarad (μ F), dan nanoFarad (nF). Terakhir, keluaran dari sensor kapasitansi yang berupa nilai kapasitansi akan menjadi masukan atau *input* pada algoritma *K-Nearest Neighbors*.

5.1.1.3 Perancangan Sensor TCS3200

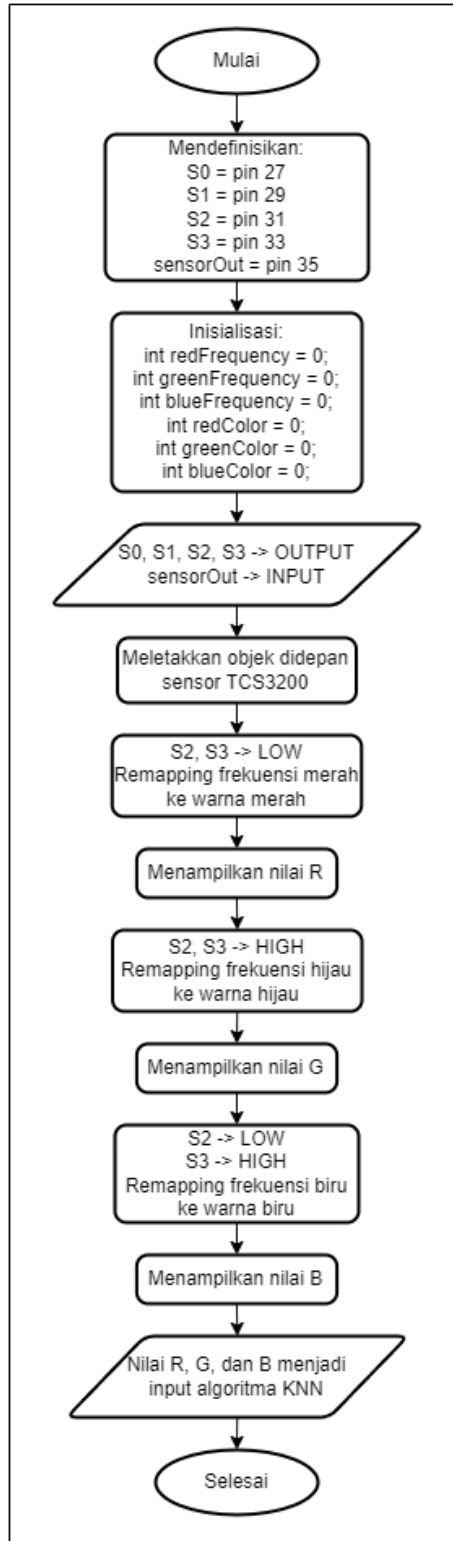
Selanjutnya, terdapat perancangan sensor warna TCS3200 ke Arduino Mega disajikan melalui Gambar 5.6.



Gambar 5.6 Perancangan Sensor TCS3200

(Sumber: Dokumen Pribadi)

Pada Gambar 5.6 sensor warna TCS3200 terhubung pada pin 27, 29, 31, 33, 35, VCC, dan Ground Arduino Mega sesuai dengan konfigurasi pin pada Tabel 5.2. Alasan mengapa pada Gambar 5.6 penulis menggunakan *breadboard* yaitu karena pada sistem keseluruhan juga menggunakannya. *Breadboard* digunakan karena ketidaktersediaan pin Ground dan VCC pada Arduino Mega guna mencukupi kebutuhan setiap komponen yang akan digunakan. Sensor TCS3200 digunakan untuk menghitung nilai warna R, G, dan B pada suatu objek. Adapun teknis pembacaan sensor oleh sensor TCS3200 disajikan melalui Gambar 5.7.



Gambar 5.7 Diagram Alir Sensor TCS3200

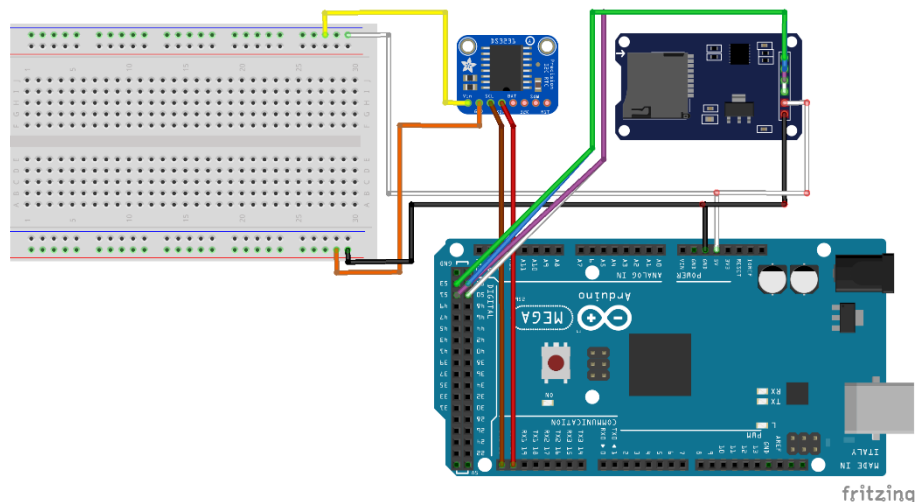
(Sumber: Dokumen Pribadi)

Gambar 5.7 menyajikan teknis pembacaan sensor oleh sensor TCS3200. Pembacaan sensor diawal dengan inisialisasi variabel yang akan digunakan.

Kemudian menentukan mode pin mana yang menjadi *input* dan *output*. Setelah itu, dapat meletakkan objek yang akan diuji coba tepat didepan sensor TCS3200. Sensor TCS3200 akan melakukan remapping frekuensi warna R, G, dan B sesuai dengan kalibrasi sensor yang telah dilakukan sebelumnya menggunakan kertas berwarna merah, hijau, dan biru. Nilai warna RGB akan diperlihatkan pada serial monitor Arduino IDE. Selanjutnya nilai R, G, dan B ini menjadi *input* untuk algoritma *K-Nearest Neighbors*.

5.1.1.4 Perancangan Penyimpanan Micro SD Card

Gambar 5.8 menggambarkan desain atau perancangan Kartu Micro SD terhadap Arduino Mega.



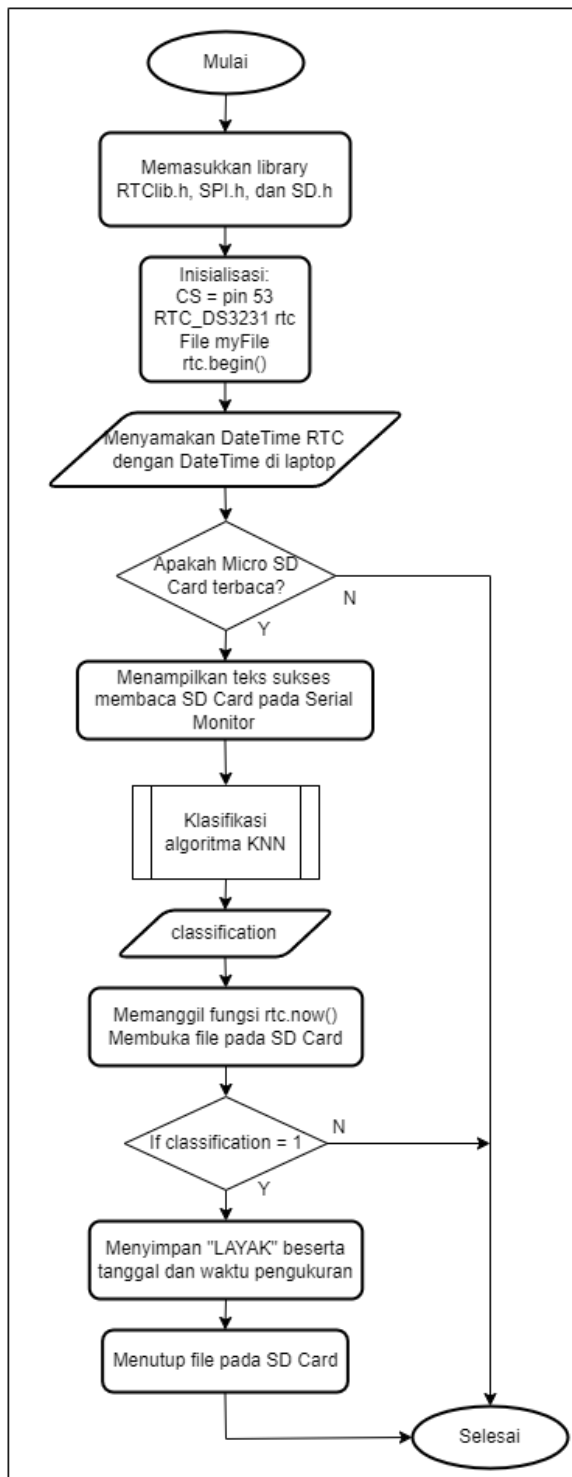
Gambar 5.8 Perancangan Micro SD Card

(Sumber: Dokumen Pribadi)

Pada Gambar 5.8 dapat terlihat perancangan Modul Micro SD beserta RTC ke Arduino Mega. RTC digunakan untuk menghitung tanggal dan waktu secara terus menerus selama ada daya dari baterai RTC. Sedangkan Modul Micro SD digunakan untuk menyimpan hasil dari klasifikasi sistem beserta tanggal dan waktunya ke dalam Micro SD Card. Hasil dari klasifikasi sistem yang disimpan hanya hasil yang bernilai 1, atau yang berkelas “LAYAK” saja. Hal ini karena madu yang direkomendasikan untuk dikonsumsi oleh penderita Diabetes mellitus adalah yang memiliki kelas “LAYAK”. Selain itu, juga sebagai acuan bagi penderita Diabetes mellitus untuk mempertimbangkan jumlah konsumsi harian madu dengan lebih bijak. Sedangkan madu berkelas “TIDAK LAYAK” tidak perlu dicatat dalam sistem karena seharusnya tidak dikonsumsi oleh penderita Diabetes mellitus.

Proses penyimpanan data pada Micro SD Card dimulai dengan memasukkan library yang akan digunakan, dan inisialisasi variabel. Selanjutnya terdapat penyamaan tanggal dan waktu RTC dengan tanggal dan waktu di jam pada laptop. Ini hanya dilakukan sekali saja, karena RTC akan menyimpan dan terus menghitung waktu tersebut. Kemudian melakukan pengecekan Micro SD Card, dan diakhiri dengan penyimpanan data hasil klasifikasi “LAYAK” beserta tanggal dan waktu

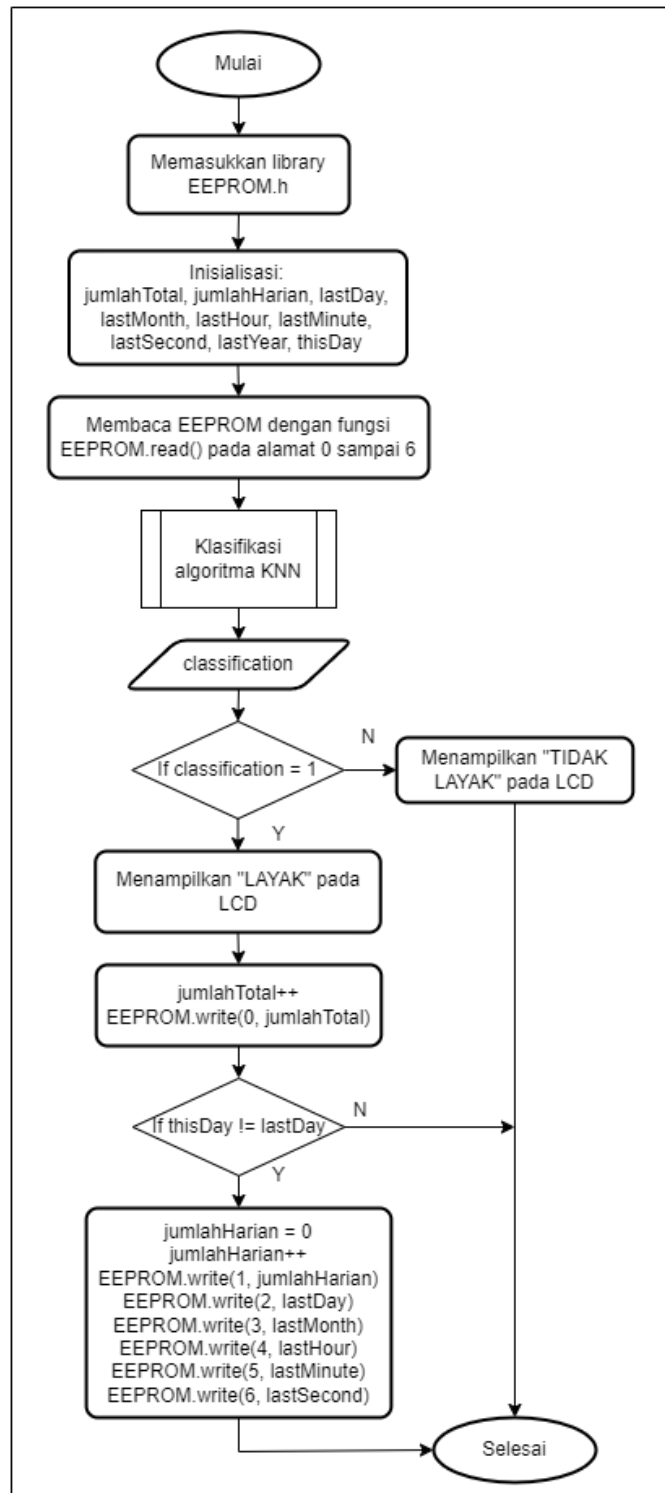
pengukuran dilakukan. Adapun proses teknis lebih jelasnya mengenai penyimpanan data pada Micro SD Card akan disajikan melalui Gambar 5.9.



Gambar 5.9 Diagram Alir Penyimpanan Modul Micro SD
(Sumber: Dokumen Pribadi)

5.1.1.5 Perancangan Penyimpanan EEPROM

Perancangan selanjutnya yaitu terdapat perancangan EEPROM pada Arduino Mega yang disajikan melalui Gambar 5.10.



Gambar 5.10 Diagram Alir Penyimpanan EEPROM

(Sumber: Dokumen Pribadi)

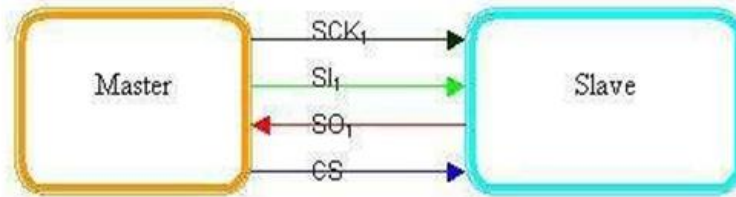
Gambar 5.10 terdapat alur penyimpanan data pada EEPROM Arduino Mega. Penulis menggunakan 7 buah alamat pada EEPROM untuk menyimpan data yaitu dari alamat 0 sampai 6. Adapun penjabaran dari penggunaan alamat EEPROM ini adalah sebagai berikut:

1. Alamat 0 pada EEPROM digunakan untuk menyimpan data jumlah konsumsi total madu.
2. Alamat 1 pada EEPROM digunakan untuk menyimpan data jumlah konsumsi harian madu.
3. Alamat 2 pada EEPROM digunakan untuk menyimpan data variabel *lastDay* atau tanggal terakhir dilakukan pengukuran.
4. Alamat 3 pada EEPROM digunakan untuk menyimpan data variabel *lastMonth* atau bulan terakhir dilakukan pengukuran.
5. Alamat 4 pada EEPROM digunakan untuk menyimpan data variabel *lastHour* atau jam terakhir dilakukan pengukuran.
6. Alamat 5 pada EEPROM digunakan untuk menyimpan data variabel *lastMinute* atau menit terakhir dilakukan pengukuran.
7. Alamat 6 pada EEPROM digunakan untuk menyimpan data variabel *lastSecond* atau detik terakhir dilakukan pengukuran.

Pada sistem juga terdapat pengecekan hari, dengan cara mencocokkan variabel *lastDay* dengan *thisDay*. Jika *thisDay* tidak sama dengan *lastDay*, maka sistem akan mereset kembali nilai jumlah konsumsi harian yang ada pada EEPROM alamat 1.

5.1.1.6 Perancangan Protokol Komunikasi

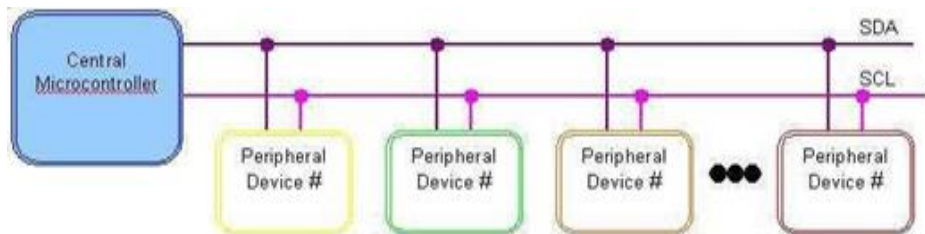
Terdapat dua buah protokol komunikasi yang digunakan oleh sistem, yaitu SPI dan I2C. Protokol *Serial Peripheral Interface* (SPI) ialah protokol sinkron yang dapat melakukan transfer data dengan kecepatan tinggi antara mikrokontroler dengan berbagai perangkat periferan seperti sensor, memori, dan tampilan. SPI menggunakan empat jalur komunikasi yaitu jalur *Data Input* (MISO), *Data Output* (MOSI), *Clock* (SCK), dan *Chip Select* (CS). Pada Modul Micro SD yang terdapat pada sistem menggunakan protokol SPI ini karena protokol SPI mendukung transfer data dengan kecepatan tinggi, yang penting dalam aplikasi penyimpanan seperti Modul Micro SD. Protokol SPI juga memungkinkan kontrol penuh terhadap operasi Modul Micro SD, termasuk transfer data, pengaturan, dan baca/tulis ke *Micro SD Card*. Adapun skema dari protokol SPI disajikan melalui Gambar 5.11.



Gambar 5.11 Cara Kerja Protokol SPI

(Sumber: (Jyothi, et al., 2012))

Sementara itu, protokol *Inter-Integrated Circuit* (I2C) ialah protokol yang menggunakan dua buah jalur komunikasi, yaitu SDA (*Serial Data*) sebagai jalur data dan SCL (*Serial Clock*) sebagai jalur sinkronisasi *clock*. Sistem menggunakan protokol I2C untuk menyederhanakan *wiring* dari layar LCD ke Arduino Mega, dari yang semula memiliki jalur 16 pin menjadi hanya 4 pin saja untuk dapat digunakan. Adapun skema dari protokol SPI disajikan melalui Gambar 5.12.



Gambar 5.12 Cara Kerja Protokol I2C

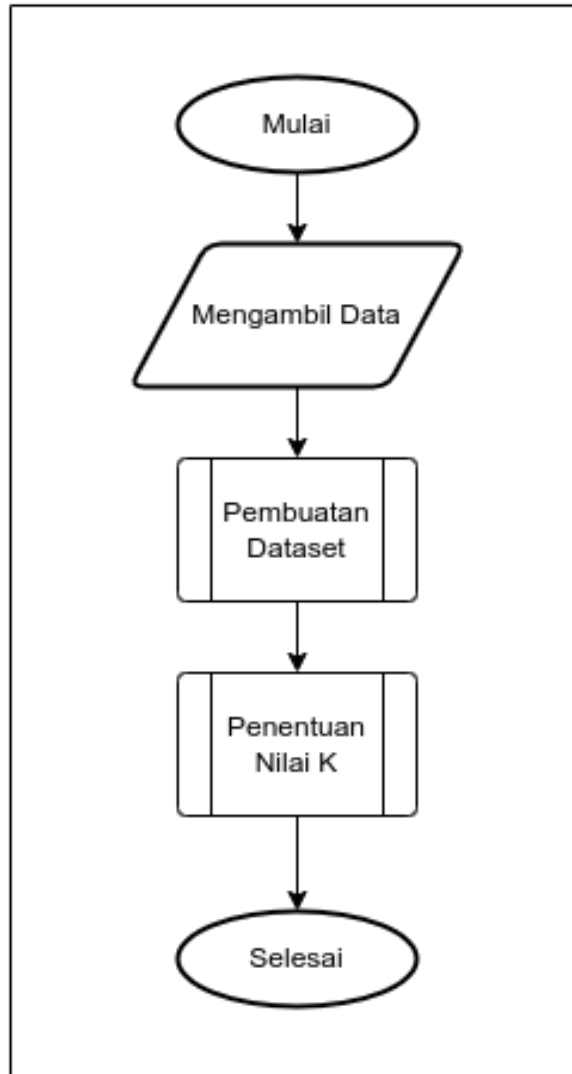
(Sumber: (Jyothi, et al., 2012))

5.1.2 Perancangan Perangkat Lunak

Tahap dimana penulis menyusun rancangan yang diperlukan dalam pembuatan sistem berbasis perangkat lunak. Adapun beberapa rancangan perangkat lunak yang akan dibuat meliputi:

5.1.2.1 Perancangan Diagram Alir Data Training

Pada tahap ini, penulis merancang diagram alir data *training* yang disajikan melalui Gambar 5.13.

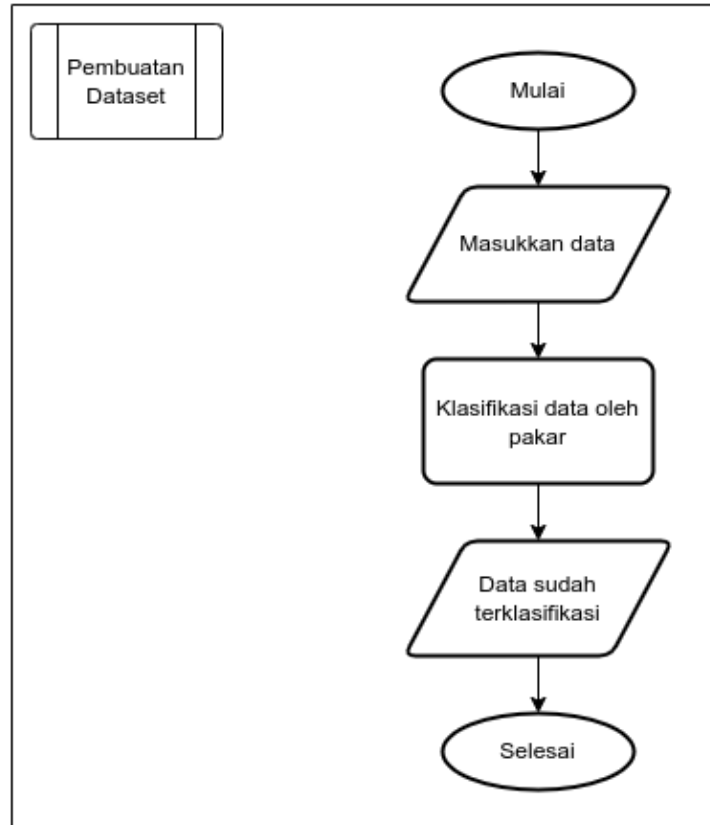


Gambar 5.13 Diagram Alir Data *Training*

(Sumber: Dokumen Pribadi)

5.1.2.2 Perancangan Diagram Alir Pembuatan Dataset

Perancangan diagram alir pembuatan dataset berguna sebab data yang penulis gunakan merupakan data primer. Perancangan diagram alir pembuatan dataset disajikan melalui Gambar 5.14.



Gambar 5.14 Diagram Alir Pembuatan Dataset

(Sumber: Dokumen Pribadi)

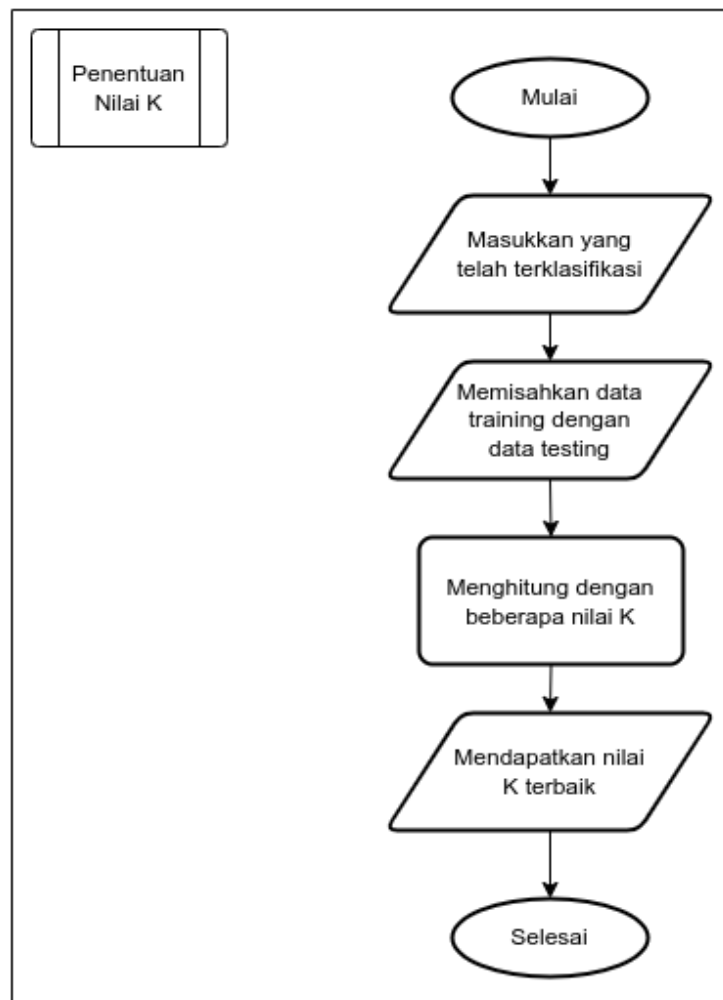
Setelah mencari sampel madu sebanyak 30 data, penulis melakukan klasifikasi sampel dengan cara mendatangi salah satu pakar madu di Peternakan Tawon Rimba Raya yang berada di Kecamatan Lawang, Kabupaten Malang, Jawa Timur. Pakar madu tersebut sudah berpengalaman selama kurang lebih 20 tahun pada bidang ini, sehingga sudah dapat dikatakan kredibel. Berdasarkan hasil diskusi penulis dengan pakar tersebut, dapat disimpulkan bahwa madu terbagi menjadi madu murni, sirupan dan kurasan. Madu murni tentu memiliki khasiat yang banyak jika dikonsumsi oleh masyarakat, sehingga dapat diklasifikasikan layak konsumsi. Sedangkan madu sirupan dan kurasan merupakan madu yang kadar kemurniannya tidak 100 persen, artinya terdapat zat lain yang sengaja dicampur ke dalam madu tersebut. Oleh karena itu, Madu sirupan dan kurasan ini diklasifikasikan sebagai madu yang tidak layak konsumsi oleh masyarakat terutama penderita Diabetes mellitus karena kadar gula yang tinggi. Tabel 5.3 menyajikan fitur dan kelas dari dataset setelah tervalidasi oleh pakar madu.

Tabel 5.3 Fitur dan Kelas

Merek Madu	Jumlah Sampel	Fitur				Kelas
		C	R	G	B	
Madu X	15					Layak
Madu Y	15					Tidak Layak

5.1.2.3 Perancangan Diagram Alir Penentuan Nilai K

Nilai K terbaik pada algoritma KNN ditentukan berdasarkan perancangan diagram alir ini. Hal ini bertujuan karena nilai K berpengaruh besar terhadap akurasi dari algoritma KNN tersebut. Perancangan diagram alir penentuan nilai K disajikan melalui Gambar 5.15.

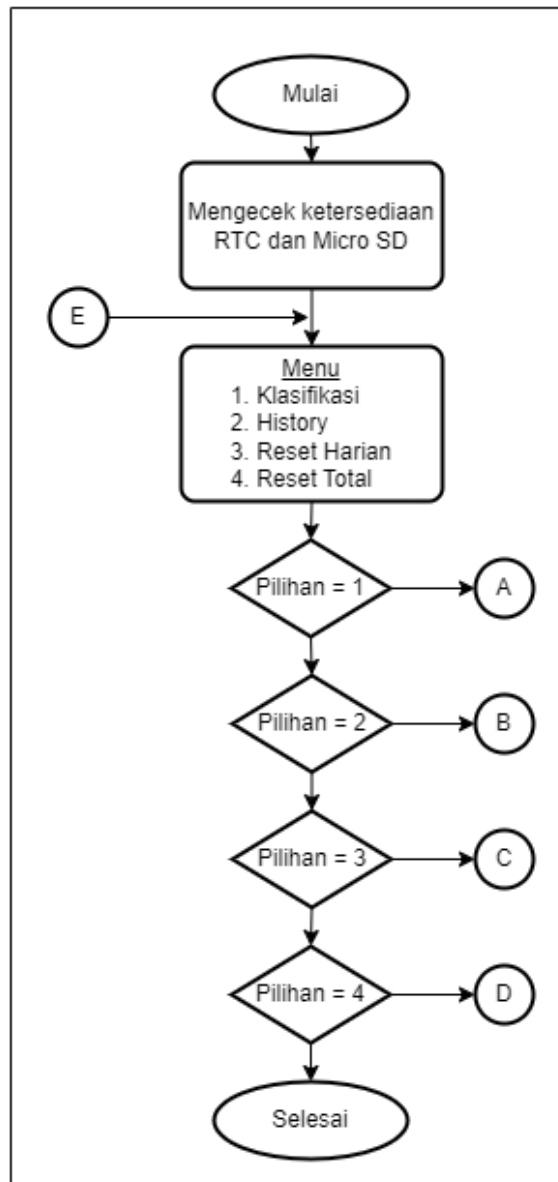


Gambar 5.15 Diagram Alir Penentuan Nilai K

(Sumber: Dokumen Pribadi)

5.1.2.4 Perancangan Diagram Alir Keseluruhan Sistem

Perancangan terakhir yaitu perancangan diagram alir keseluruhan sistem. Perancangan ini dibuat agar penulis dapat menjelaskan lebih rinci terkait logika dan fitur-fitur yang dapat dilakukan oleh sistem. Perancangan keseluruhan sistem disajikan melalui Gambar 5.16.



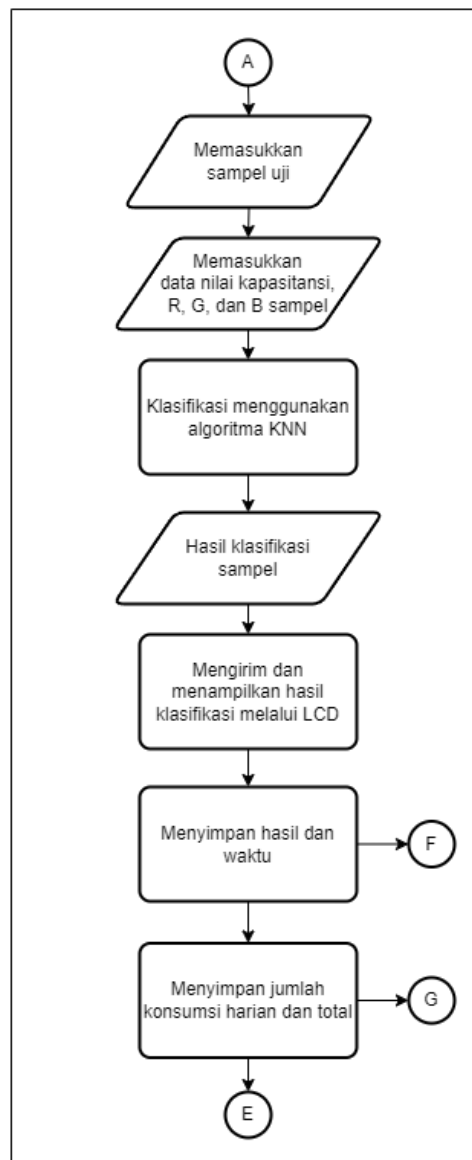
Gambar 5.16 Diagram Alir Keseluruhan Sistem

(Sumber: Dokumen Pribadi)

Pada diagram alir keseluruhan sistem ketika awal mula dijalankan akan melakukan pengecekan dari dua buah komponen yaitu RTC dan Micro SD. Pengecekan RTC pada sistem ini digunakan sebagai komponen yang dapat menghitung waktu walaupun sistem dalam keadaan mati atau tidak terhubung dengan daya. Sedangkan pengecekan Micro SD digunakan untuk memastikan apakah Micro SD pada sistem ini dapat terbaca atau tidak. Karena jika tidak

terbaca, maka sistem tidak dapat menyimpan history pengukuran yang telah dilakukan sebelumnya.

Kemudian masukan yang memiliki simbol E merupakan keluaran dari setiap menu yang akan diakses atau keluaran dari A, B, C, dan D. Ini terjadi secara berkelanjutan selama sistem menyala. Hal ini karena, setiap proses yang terjadi pada sub menu akan mengembalikan nilai pada menu utama. Pada diagram alir keseluruhan sistem juga terdapat beberapa menu yang bisa dipilih oleh pengguna, yaitu klasifikasi, history, reset harian, dan reset total. Untuk lebih jelasnya, penulis akan membedah setiap sub menu dimulai dari sub menu A Gambar 5.17.



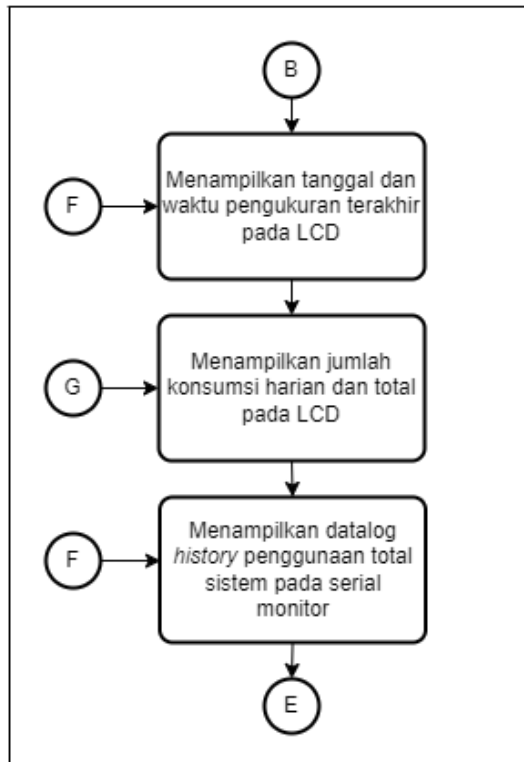
Gambar 5.17 Diagram Alir Menu Klasifikasi

(Sumber: Dokumen Pribadi)

Diagram alir menu klasifikasi yang terlihat pada Gambar 5.17, terdapat serangkaian proses yang terjadi. Ketika menu klasifikasi dipilih, maka pertama-tama sistem akan melakukan training dataset terlebih dahulu. Selanjutnya pada

LCD akan muncul teks “Silakan Masukkan Madu”, untuk itu madu sebagai sampel uji perlu dimasukkan ke dalam sistem melalui pintu kecil pada bagian depan sistem. Kemudian terjadi proses pembacaan data secara analog dari sistem menggunakan sensor kapasitansi dan sensor warna TCS3200. Data yang dibaca berupa nilai kapasitansi, R, G, dan B. Setelah data analog diambil, sistem akan melakukan klasifikasi memanfaatkan algoritma *K-Nearest Neighbors* disertai dengan diperlihatkannya hasil klasifikasi melalui LCD. Selanjutnya, sistem akan menyimpan hasil klasifikasi dan waktunya pada Micro SD Card.

Hasil klasifikasi yang disimpan hanya untuk kelas “LAYAK” saja. Hal ini karena ketika hasil klasifikasi menunjukkan kelas “TIDAK LAYAK” itu berarti madu tidak direkomendasikan untuk dikonsumsi oleh penderita Diabetes mellitus. Kemudian sistem juga akan menyimpan jumlah hasil konsumsi madu harian dan total pada EEPROM Arduino Mega. Simbol E pada Gambar 5.17 menunjukkan keadaan setelah serangkaian proses menu klasifikasi dijalankan maka akan kembali lagi ke menu utama pada Gambar 5.16. Sedangkan simbol F dan G berarti nilai tersebut akan dikirimkan ke menu history yang disajikan melalui Gambar 5.18.

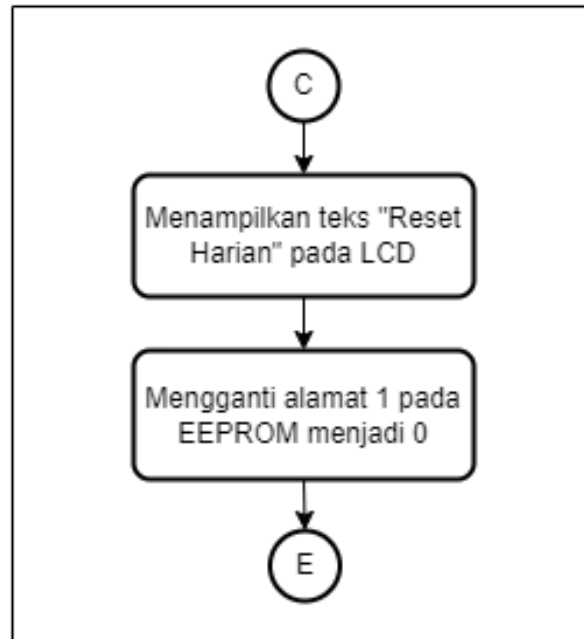


Gambar 5.18 Diagram Alir Menu History

(Sumber: Dokumen Pribadi)

Menu history yang terlihat melalui Gambar 5.18 menyajikan serangkaian proses yang terjadi. Ketika menu history dipilih, maka sistem akan menampilkan tiga buah tampilan pada LCD. Tampilan baris pertama yaitu menampilkan tanggal dan waktu pengukuran atau klasifikasi terakhir. Tampilan baris kedua menampilkan jumlah konsumsi harian madu oleh penderita Diabetes mellitus. Sedangkan pada tampilan baris ketiga akan menampilkan jumlah konsumsi total

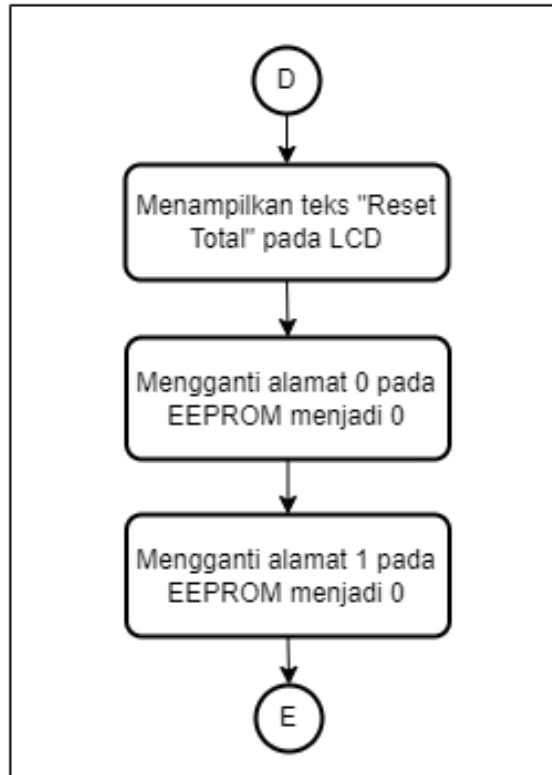
dari pertama kali sistem digunakan hingga terakhir pengukuran. Dengan catatan sistem belum mengalami reset total pada sub menu keempat. Simbol F dan G pada Gambar 5.18 merupakan nilai yang diambil dari EEPROM Arduino Mega setelah sebelumnya disimpan seperti pada Gambar 5.17. Sistem juga akan menampilkan datalog history secara detail pada serial monitor Arduino IDE. Hal ini karena informasi ini bersifat teknis, dan juga tidak cukup jika diperlihatkan semua melalui LCD. Kemudian, akhir dari diagram alir menu history akan kembali lagi menuju menu utama dengan simbol E.



Gambar 5.19 Diagram Alir Menu Reset Harian

(Sumber: Dokumen Pribadi)

Pada diagram alir menu reset harian yang terlihat pada Gambar 5.19, terdapat serangkaian proses yang terjadi. Ketika menu reset harian dipilih, maka sistem pertama-tama akan menampilkan "Reset Harian" pada LCD. Kemudian mengganti nilai yang terdapat pada alamat 1 EEPROM Arduino Mega menjadi 0. Hal ini karena alamat 1 pada EEPROM berisikan jumlah konsumsi harian yang akan bertambah seiring sistem melakukan klasifikasi pada hari tersebut. Selanjutnya, sistem akan mengembalikan tampilan kembali ke menu utama.



Gambar 5.20 Diagram Alir Reset Total

(Sumber: Dokumen Pribadi)

Pada diagram alir menu reset total yang terlihat pada Gambar 5.20, terdapat serangkaian proses yang terjadi. Ketika menu reset total dipilih, maka sistem pertama-tama akan menampilkan “Reset Total” pada LCD. Kemudian mengganti nilai yang terdapat pada alamat 0 dan 1 EEPROM Arduino Mega menjadi 0. Hal ini karena alamat 0 pada EEPROM berisikan jumlah konsumsi total yang akan bertambah seiring sistem melakukan klasifikasi. Sedangkan alamat 1 pada EEPROM berisikan jumlah konsumsi harian yang akan bertambah seiring sistem melakukan klasifikasi pada hari tersebut. Terakhir, sistem akan mengembalikan tampilan kembali ke menu utama dan begitu seterusnya selama sistem dalam keadaan menyala.

5.2 Implementasi Sistem

Bagian implementasi akan memaparkan secara mendalam mengenai proses implementasi perangkat keras dan lunak pada sistem setelah melalui tahapan perancangan sebelumnya.

5.2.1 Implementasi Perangkat Keras

Pada tahap ini penulis melakukan pemasangan seluruh komponen sesuai dengan skema yang sudah tertera sebelumnya. Setelah itu dilakukan pembuatan *cover* atau *casing* yang bertujuan untuk melindungi komponen-komponen sistem. Adapun tampilan dari *cover* atau *casing* sistem disajikan melalui Gambar 5.21 sampai Gambar 5.25.



Gambar 5.21 Sistem Tampak Depan

(Sumber: Dokumen Pribadi)

Pada Gambar 5.21, merupakan tampilan depan dari sistem. Tampilan depan sistem ini memiliki ukuran 30 cm untuk panjangnya, dan 25 cm untuk tingginya. Terdapat beberapa hal yang mendasari tampilan depan ini, yaitu pintu, *push botton*, dan LCD. Pintu kecil ini berukuran 7x7 cm yang dapat dibuka dan ditutup sesuai dengan pintu pada umumnya. Pintu ini digunakan untuk memasukkan sampel madu yang akan diuji nantinya. Kemudian dapat ditutup lagi setelah madu dimasukkan agar sistem tidak terpengaruh pencahayaan dari luar *cover*. LCD digunakan sebagai layar untuk tampilan *output* perhitungan dari sistem. Sedangkan *push button* berfungsi sebagai tombol yang dapat ditekan untuk memilih menu yang akan diperlihatkan pada LCD. *Push button* terdiri dari tiga buah dengan fungsi yang berbeda-beda. *Push button* berwarna hijau digunakan untuk menggerakkan menu ke arah atas. *Push button* berwarna merah digunakan untuk menggerakkan menu ke arah bawah. Sedangkan *push button* berwarna kuning digunakan untuk memilih menu yang pengguna inginkan atau dapat diartinya sebagai tombol *select*.



Gambar 5.22 Sistem Tampak Belakang

(Sumber: Dokumen Pribadi)

Pada Gambar 5.22, ialah tampilan belakang sistem. Tampilan belakang ini tidak memiliki suatu komponen yang menonjol, alias hanya datar saja dengan akrilik berwarna hitam kedap sebagai penyusun *cover*. Bagian atas dari tampilan belakang yang tampak pada Gambar 5.22 tersebut hanyalah pintu yang terbuka dari bagian atas sistem ini. Jika pintu bagian atas sistem dalam keadaan tertutup, maka yang terlihat pada bagian belakang hanya permukaan akrilik yang datar saja. Tampilan belakang dari sistem ini tentu memiliki ukuran yang sama dengan tampilan depannya, yaitu 30x25 cm.



Gambar 5.23 Sistem Tampak Samping Kanan

(Sumber: Dokumen Pribadi)

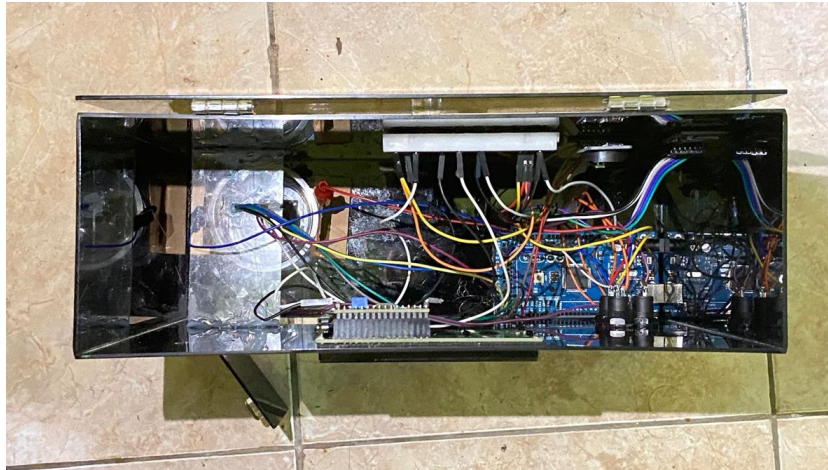
Pada Gambar 5.23 merupakan tampilan samping kanan dari sistem. Tampilan samping kanan sistem ini memiliki ukuran 10x25 cm. Pada tampilan samping kanan sistem terdapat lubang berukuran 1x1 cm yang digunakan untuk *port* USB dari Arduino Mega ke laptop sebagai catu daya. Sedangkan tampilan samping kiri dari sistem tidak ditampilkan karena sama saja dengan tampak samping kanan. Hanya saja pada tampilan samping kiri tidak terdapat lubang untuk *port* USB, alias hanya datar saja.



Gambar 5.24 Sistem Tampak Atas

(Sumber: Dokumen Pribadi)

Pada Gambar 5.24 ialah tampilan atas sistem. Tampilan atas dari sistem ini memiliki pintu berukuran 30x10 cm yang dapat dibuka dan ditutup. Ketika pintu dalam keadaan terbuka, penulis atau pengguna dapat melihat bagian dalam dari sistem secara keseluruhan. Penulis juga dapat melakukan perbaikan ketika ada suatu komponen yang bermasalah seperti kabel yang terlepas, mengganti baterai RTC yang habis, dan mengganti micro SD card ketika *error* saat pembacaan. Sedangkan ketika pintu dalam keadaan tertutup, digunakan untuk melindungi sistem dari sesuatu seperti cairan yang menetes dari atas, hewan berbuku-buku, dan banyak lagi yang bersifat merusak sistem. Selain itu, pintu yang ditutup juga akan memaksimalkan pembacaan sensor warna TCS3200 karena sangat sensitif dengan pencahayaan.



Gambar 5.25 Sistem Tampak Dalam

(Sumber: Dokumen Pribadi)

Gambar 5.25 menyajikan tampilan didalam sistem. Terlihat di bagian tampak dalam ini, penulis atau pengguna dapat berinteraksi langsung dengan seluruh komponen yang digunakan sistem. Mulai dari pemasangan awal setiap komponennya, *wiring* kabel antar komponen, dan perbaikan pasca komponen dirakit.

5.2.2 Implementasi Perangkat Lunak

Bagian ini merupakan kumpulan kode program yang terintegrasi sesuai dengan kebutuhan sistem, seperti untuk akuisisi data menggunakan sensor-sensor, menghitung waktu, dan menyimpan data.

5.2.2.1 Implementasi Program Uji Coba Sensor Kapasitansi

Program 5.1: Sensor Kapasitansi	
1	<code>#include <Wire.h></code>
2	<code>#include <LiquidCrystal_I2C.h></code>
3	
4	<code>LiquidCrystal_I2C lcd(0x3F, 20, 4);</code>
5	<code>const int OUT_PIN = A14;</code>
6	<code>const int IN_PIN = A15;</code>
7	<code>const float IN_STRAY_CAP_TO_GND = 24.48;</code>
8	<code>const float IN_CAP_TO_GND = IN_STRAY_CAP_TO_GND;</code>
9	<code>const float R_PULLUP = 34.8;</code>
10	<code>const int MAX_ADC_VALUE = 1023;</code>
11	
12	<code>void setup() {</code>
13	<code> pinMode(OUT_PIN, OUTPUT);</code>
14	<code> pinMode(IN_PIN, OUTPUT);</code>

```

15     lcd.init();
16     lcd.backlight();
17 }
18
19 void loop(){
20     pinMode(IN_PIN, INPUT);
21     digitalWrite(OUT_PIN, HIGH);
22     int val = analogRead(IN_PIN);
23     digitalWrite(OUT_PIN, LOW);
24
25     if (val < 1000){
26         pinMode(IN_PIN, OUTPUT);
27
28         float capacitance = (float)val * IN_CAP_TO_GND /
29 (float)(MAX_ADC_VALUE - val);
30
31         lcd.setCursor(0,0);
32         lcd.print(F("Capacitance = "));
33         lcd.setCursor(0,1);
34         lcd.print(capacitance, 3);
35         lcd.print(F("pF "));
36         lcd.print(val);
37         lcd.print("mS ");
38     }
39     else{
40         pinMode(IN_PIN, OUTPUT);
41         delay(1);
42         pinMode(OUT_PIN, INPUT_PULLUP);
43         unsigned long u1 = micros();
44         unsigned long t;
45         int digVal;
46
47         do{
48             digVal = digitalRead(OUT_PIN);
49             unsigned long u2 = micros();
50             t = u2 > u1 ? u2 - u1 : u1 - u2;
51         }
52

```

53	while ((digVal < 1) && (t < 400000L));
54	
55	pinMode(OUT_PIN, INPUT);
56	val = analogRead(OUT_PIN);
57	digitalWrite(IN_PIN, HIGH);
58	int dischargeTime = (int)(t / 1000L) * 5;
59	delay(dischargeTime);
60	pinMode(OUT_PIN, OUTPUT);
61	digitalWrite(OUT_PIN, LOW);
62	digitalWrite(IN_PIN, LOW);
63	
64	float capacitance = -(float)t / R_PULLUP / log(1.0 -
65	(float)val / (float)MAX_ADC_VALUE);
66	
67	lcd.setCursor(0,0);
68	lcd.print(F("Capacitance = "));
69	if (capacitance > 1000.0){
70	lcd.setCursor(0,1);
71	lcd.print(capacitance / 1000.0, 2);
72	lcd.print(F("uF "));
73	lcd.print(val);
74	lcd.print("mS");
75	}
76	
77	else{
78	lcd.setCursor(0,1);
79	lcd.print(capacitance, 2);
80	lcd.print(F("nF "));
81	lcd.print(val);
82	lcd.print("mS");
83	}
84	}
85	while (millis() % 1000 != 0);
86	}

Penjelasan Kode Program:

- Kode baris 1 memuat *library* Wire.h
- Kode baris 2 memuat *library* LiquidCrystal_I2C.h

- Kode baris 4 mendefinisikan alamat I2C pada LCD yaitu 0x3F, serta seri LCD yaitu LCD dengan jumlah karakter 20x4
- Kode baris 5 mendefinisikan variabel OUT_PIN sebagai pin A14 pada Arduino
- Kode baris 6 mendefinisikan variabel IN_PIN sebagai pin A15 pada Arduino
- Kode baris 7 mendefinisikan variabel IN_STRAY_CAP_TO_GND dengan nilai const float 24,48
- Kode baris 8 mendefinisikan variabel IN_CAP_TO_GND bernilai sama dengan variabel IN_STRAY_CAP_TO_GND
- Kode baris 9 mendefinisikan variabel R_PULLUP dengan nilai const float 34,8
- Kode baris 10 mendefinisikan variabel MAX_ADC_VALUE dengan nilai const int 1023
- Kode baris 12 mendefinisikan awal dari fungsi void setup()
- Kode baris 13 mendefinisikan variabel OUT_PIN sebagai fungsi OUTPUT pada pin arduino
- Kode baris 14 mendefinisikan variabel IN_PIN sebagai fungsi OUTPUT pada pin arduino
- Kode baris 15 melakukan inisialisasi awal LCD
- Kode baris 16 mengaktifkan *backlight* LCD
- Kode baris 17 mendefinisikan akhir dari fungsi void setup()
- Kode baris 19 mendefinisikan awal dari fungsi void loop()
- Kode baris 20 mendefinisikan variabel IN_PIN sebagai fungsi input pada pin arduino
- Kode baris 21 memberikan nilai *HIGH* pada variabel OUT_PIN
- Kode baris 22 menginisialisasi variabel val dengan nilai yang diambil dari fungsi analogRead(IN_PIN)
- Kode baris 23 memberikan nilai *LOW* pada variabel OUT_PIN
- Kode baris 25-38 mendefinisikan perhitungan nilai kapasitansi berdasarkan variabel val yang diambil dari nilai analog sensor dengan kondisi nilai val < 1000. Nilai kapasitansi yang menjadi keluaran memiliki satuan pF. Kemudian hasil perhitungan nilai kapasitansi akan ditampilkan pada LCD
- Kode baris 39-75 mendefinisikan perhitungan nilai kapasitansi berdasarkan variabel val yang diambil dari nilai analog sensor dengan kondisi nilai val > 1000. Nilai kapasitansi yang menjadi keluaran memiliki satuan uF. Kemudian hasil perhitungan nilai kapasitansi akan ditampilkan pada LCD

- Kode baris 77-82 mendefinisikan perhitungan nilai kapasitansi berdasarkan variabel val yang diambil dari nilai analog sensor dengan kondisi nilai val > 1000. Nilai kapasitansi yang menjadi keluaran memiliki satuan nF. Kemudian hasil perhitungan nilai kapasitansi akan ditampilkan pada LCD
- Kode baris 85 mendefinisikan fungsi millis()
- Kode baris 86 mendefinisikan akhir dari fungsi void loop()

5.2.2.2 Implementasi Program Uji Coba Sensor TCS3200

Program 5.2: Sensor TCS3200	
1	#define S0 27
2	#define S1 29
3	#define S2 31
4	#define S3 33
5	#define sensorOut 35
6	
7	int redFrequency = 0;
8	int greenFrequency = 0;
9	int blueFrequency = 0;
10	
11	int redColor = 0;
12	int greenColor = 0;
13	int blueColor = 0;
14	
15	void setup() {
16	pinMode(S0, OUTPUT);
17	pinMode(S1, OUTPUT);
18	pinMode(S2, OUTPUT);
19	pinMode(S3, OUTPUT);
20	pinMode(sensorOut, INPUT);
21	digitalWrite(S0,HIGH);
22	digitalWrite(S1,LOW);
23	
24	Serial.begin(9600);
25	}
26	
27	void loop() {
28	digitalWrite(S2,LOW);

29	<code>digitalWrite(S3,LOW);</code>
30	<code>redFrequency = pulseIn(sensorOut, LOW);</code>
31	<code>redColor = map(redFrequency, 24, 151, 255,0);</code>
32	<code>Serial.print("R = ");</code>
33	<code>Serial.print(redColor);</code>
34	<code>delay(100);</code>
35	
36	<code>digitalWrite(S2,HIGH);</code>
37	<code>digitalWrite(S3,HIGH);</code>
38	<code>greenFrequency = pulseIn(sensorOut, LOW);</code>
39	<code>greenColor = map(greenFrequency, 45, 129, 255, 0);</code>
40	<code>Serial.print(" G = ");</code>
41	<code>Serial.print(greenColor);</code>
42	<code>delay(100);</code>
43	
44	<code>digitalWrite(S2,LOW);</code>
45	<code>digitalWrite(S3,HIGH);</code>
46	<code>blueFrequency = pulseIn(sensorOut, LOW);</code>
47	<code>blueColor = map(blueFrequency, 19, 118, 255, 0);</code>
48	<code>Serial.print(" B = ");</code>
49	<code>Serial.print(blueColor);</code>
50	<code>delay(100);</code>
51	
52	<code>if(redColor > greenColor && redColor > blueColor){</code>
53	<code> Serial.println(" - RED detected!");</code>
54	<code>}</code>
55	<code>if(greenColor > redColor && greenColor > blueColor){</code>
56	<code> Serial.println(" - GREEN detected!");</code>
57	<code>}</code>
58	<code>if(blueColor > redColor && blueColor > greenColor){</code>
59	<code> Serial.println(" - BLUE detected!");</code>
60	<code>}</code>
61	<code>}</code>

Penjelasan Kode Program:

- Kode baris 1 mendefinisikan pin S0 sebagai pin 27 pada Arduino
- Kode baris 2 mendefinisikan pin S1 sebagai pin 29 pada Arduino
- Kode baris 3 mendefinisikan pin S2 sebagai pin 31 pada Arduino

- Kode baris 4 mendefinisikan pin S3 sebagai pin 33 pada Arduino
- Kode baris 5 mendefinisikan pin sensorOut sebagai pin 35 pada Arduino
- Kode baris 7 mendefinisikan variabel redFrequency bernilai 0
- Kode baris 8 mendefinisikan variabel greenFrequency bernilai 0
- Kode baris 9 mendefinisikan variabel blueFrequency bernilai 0
- Kode baris 11 mendefinisikan variabel redColor bernilai 0
- Kode baris 12 mendefinisikan variabel greenColor bernilai 0
- Kode baris 13 mendefinisikan variabel blueColor bernilai 0
- Kode baris 15 mendefinisikan awal dari fungsi void setup()
- Kode baris 16 mendefinisikan variabel S0 sebagai fungsi OUTPUT pada pin arduino
- Kode baris 17 mendefinisikan variabel S1 sebagai fungsi OUTPUT pada pin arduino
- Kode baris 18 mendefinisikan variabel S2 sebagai fungsi OUTPUT pada pin arduino
- Kode baris 19 mendefinisikan variabel S3 sebagai fungsi OUTPUT pada pin arduino
- Kode baris 20 mendefinisikan variabel sensorOut sebagai fungsi INPUT pada pin arduino
- Kode baris 21 memberikan nilai *HIGH* pada variabel S0
- Kode baris 22 memberikan nilai *HIGH* pada variabel S1
- Kode baris 24 memulai komunikasi pada serial monitor dengan *baudrate* 9600
- Kode baris 25 mendefinisikan akhir dari fungsi void setup()
- Kode baris 27 mendefinisikan awal dari fungsi void loop ()
- Kode baris 28 memberikan nilai *LOW* pada variabel S2
- Kode baris 29 memberikan nilai *LOW* pada variabel S3
- Kode baris 30-34 mendefinisikan *mapping* frekuensi warna merah dan disertai dengan menampilkan nilai dari variabel redColor
- Kode baris 36-42 mendefinisikan *mapping* frekuensi warna hijau dan disertai dengan menampilkan nilai dari variabel greenColor
- Kode baris 44-50 mendefinisikan *mapping* frekuensi warna biru dan disertai dengan menampilkan nilai dari variabel blueColor

- Kode baris 52-54 mendefinisikan seleksi kondisi jika variabel warna merah bernilai lebih besar dari warna hijau dan biru, kemudian ditampilkan keluaran berupa warna merah terdeteksi
- Kode baris 55-57 mendefinisikan seleksi kondisi jika variabel warna hijau bernilai lebih besar dari warna merah dan biru, kemudian ditampilkan keluaran berupa warna hijau terdeteksi
- Kode baris 58-60 mendefinisikan seleksi kondisi jika variabel warna biru bernilai lebih besar dari warna hijau dan merah, kemudian ditampilkan keluaran berupa warna biru terdeteksi
- Kode baris 61 mendefinisikan akhir dari fungsi void loop ()

5.2.2.3 Implementasi Program Uji Coba RTC

Program 5.3: RTC	
1	#include <RTClib.h>
2	#include <LiquidCrystal_I2C.h>
3	#include <Wire.h>
4	
5	RTC_DS3231 rtc;
6	LiquidCrystal_I2C lcd(0x3F, 20, 4);
7	
8	void setup() {
9	Serial.begin(9600);
10	rtc.begin();
11	rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
12	}
13	
14	void loop() {
15	DateTime waktu_sekarang = rtc.now();
16	lcd.print("Hari ini:");
17	lcd.setCursor(0, 1);
18	lcd.print(waktu_sekarang.day ()); lcd.print("-");
19	lcd.print(waktu_sekarang.month ()); lcd.print("-");
20	lcd.print(waktu_sekarang.year ()); lcd.print(" ");
21	lcd.print(waktu_sekarang.hour ()); lcd.print(":");
22	lcd.print(waktu_sekarang.minute ()); lcd.print(":");
23	lcd.print(waktu_sekarang.second ()); lcd.print("");
24	delay(1000);
25	}

Penjelasan Kode Program:

- Kode baris 1 memuat *library* RTCLib.h
- Kode baris 2 memuat *library* LiquidCrystal_I2C.h
- Kode baris 3 memuat *library* Wire.h
- Kode baris 5 mendefinisikan RTC_DS3231 sebagai rtc
- Kode baris 6 mendefinisikan alamat I2C pada LCD yaitu 0x3F, serta seri LCD yaitu LCD dengan jumlah karakter 20x4
- Kode baris 8 mendefinisikan awal dari fungsi void setup()
- Kode baris 9 memulai komunikasi pada serial monitor dengan *baudrate* 9600
- Kode baris 10 mendefinisikan untuk memulai fungsi pada library RTCLib.h
- Kode baris 11 mendefinisikan pengambilan tanggal dan waktu secara *real time*
- Kode baris 12 mendefinisikan akhir dari fungsi void setup()
- Kode baris 14 mendefinisikan awal dari fungsi void loop ()
- Kode baris 15 mendefinisikan variabel waktu_sekarang sebagai wadah untuk menampung tanggal dan waktu real time yang sudah diambil menggunakan fungsi rtc.now()
- Kode baris 16-23 mendefinisikan pengambilan tanggal dan waktu yang disimpan dalam variabel waktu_sekarang, kemudian ditampilkan melalui LCD
- Kode baris 24 mendefinisikan *delay* program selama satu detik
- Kode baris 25 mendefinisikan akhir dari fungsi void loop ()

5.2.2.4 Implementasi Program Uji Coba Micro SD Card

Program 5.4: Micro SD	
1	#include <RTCLib.h>
2	#include <Wire.h>
3	#include <SPI.h>
4	#include <SD.h>
5	#include <LiquidCrystal_I2C.h>
6	
7	RTC_DS3231 rtc;
8	
9	const int chipSelect = 53;

```

10 File myFile;
11 LiquidCrystal_I2C lcd(0x3F,20,4);
12
13 void setup(){
14     Serial.begin(9600);
15     while (!Serial) {
16         ;
17     }
18     lcd.init();
19     lcd.backlight();
20     rtc.begin();
21     rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
22
23     Serial.println(F("Read MicroSd Card"));
24     delay(1000);
25     if (!SD.begin(chipSelect))
26     {
27         Serial.println(F("Failed read microsd card on
28 module!"));
29         return;
30     }
31     else {Serial.println(F("Success read microsd card"));}
32     delay(1000);
33     Serial.print(F("Initializing SD card..."));
34 }
35
36 void loop()
37 {
38     DateTime waktu_sekarang = rtc.now();
39
40     File myFile = SD.open("datalog.txt", FILE_WRITE);
41     if (myFile) // jika file tersedia tulis data
42     {
43
44     Serial.print("Raw data: ");
45     Serial.print(waktu_sekarang.year ());   Serial.print("-");
46     Serial.print(waktu_sekarang.month ());  Serial.print("-");
47     Serial.print(waktu_sekarang.day ());     Serial.print(" ");

```

48	Serial.print(waktu_sekarang.hour ()); Serial.print(":");
49	Serial.print(waktu_sekarang.minute ()); Serial.print(":");
50	Serial.print(waktu_sekarang.second ()); Serial.println("");
51	
52	myFile.print("Raw data: ");
53	myFile.print(waktu_sekarang.year ()); myFile.print("-");
54	myFile.print(waktu_sekarang.month ()); myFile.print("-");
55	myFile.print(waktu_sekarang.day ()); myFile.print(" ");
56	myFile.print(waktu_sekarang.hour ()); myFile.print(":");
57	myFile.print(waktu_sekarang.minute ()); myFile.print(":");
58	myFile.print(waktu_sekarang.second ()); myFile.println("");
59	
60	myFile.close();
61	}
62	else
63	{
64	Serial.println("Failed open datalog.txt"); // jika gagal
65	print error
66	}
67	delay(2000);
68	}

Penjelasan Kode Program:

- Kode baris 1 memuat *library* RTCLib.h
- Kode baris 2 memuat *library* Wire.h
- Kode baris 3 memuat *library* SPI.h
- Kode baris 4 memuat *library* SD.h
- Kode baris 5 memuat *library* LiquidCrystal_I2C.h
- Kode baris 7 mendefinisikan RTC_DS3231 sebagai rtc
- Kode baris 9 mendefinisikan variabel chipSelect sebagai pin 53 pada Arduino
- Kode baris 10 mendefinisikan fungsi *File* pada *library* SD.h sebagai *myFile*
- Kode baris 11 mendefinisikan alamat I2C pada LCD yaitu 0x3F, serta seri LCD yaitu LCD dengan jumlah karakter 20x4
- Kode baris 13 mendefinisikan awal dari fungsi void setup()
- Kode baris 14 memulai komunikasi pada serial monitor dengan *baudrate* 9600

- Kode baris 15-17 mendefinisikan pengecekan pada serial monitor
- Kode baris 18 melakukan inisialisasi awal LCD
- Kode baris 19 mengaktifkan *backlight* LCD
- Kode baris 20 mendefinisikan untuk memulai fungsi pada library RTCLib.h
- Kode baris 21 mendefinisikan pengambilan tanggal dan waktu secara *real time*
- Kode baris 23 menampilkan teks pada serial monitor
- Kode baris 24 mendefinisikan *delay* program selama satu detik
- Kode baris 25-31 mendefinisikan pengecekan Micro SD dapat terbaca atau tidak
- Kode baris 32 mendefinisikan *delay* program selama satu detik
- Kode baris 33 menampilkan teks pada serial monitor
- Kode baris 34 mendefinisikan akhir dari fungsi void setup()
- Kode baris 36 mendefinisikan awal dari fungsi void loop()
- Kode baris 38 mendefinisikan variabel waktu_sekarang sebagai wadah untuk menampung tanggal dan waktu real time yang sudah diambil menggunakan fungsi rtc.now()
- Kode baris 40 membuka file pada Micro SD
- Kode baris 41 mendefinisikan seleksi kondisi jika file pada Micro SD dapat terbuka
- Kode baris 44-50 mendefinisikan pengambilan tanggal dan waktu yang disimpan dalam variabel waktu_sekarang, kemudian ditampilkan melalui serial monitor
- Kode baris 52-58 mendefinisikan pengambilan tanggal dan waktu yang disimpan dalam variabel waktu_sekarang, kemudian disimpan ke dalam Micro SD
- Kode baris 60 menjalankan fungsi untuk menutup file pada Micro SD
- Kode baris 62-66 mendefinisikan seleksi kondisi jika file pada Micro SD tidak dapat terbuka
- Kode baris 67 mendefinisikan *delay* program selama dua detik
- Kode baris 68 mendefinisikan akhir dari fungsi void loop()

5.2.2.5 Implementasi Program Penentuan Nilai K

Program 5.5: Uji Coba Penentuan Nilai K

```
In [240]: import pandas as pd
import numpy as np
dataset = pd.read_csv("datasetMadu2.csv")
dataset.head()

In [241]: X = dataset.iloc[:, [0,1,2,3]].values
print(X)
X.shape

Out[241]: (30, 4)

In [242]: y = dataset.iloc[:,4].values
print(y)
y.shape

Out[242]: (30,)

In [243]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

In [244]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
Xtrain = scaler.transform(X_train)
Xtest = scaler.transform(X_test)

In [245]: from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
for i in range (1,11):
    classifier = KNeighborsClassifier(n_neighbors=i)
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)
    print(accuracy_score(y_test, y_pred))
```

Penjelasan Kode Program:

- Kode baris 1 memuat library pandas dan numpy, selain itu juga memuat dataset dalam format csv
- Kode baris 2 mendefinisikan kolom pada dataset yang akan digunakan
- Kode baris 3 mendefinisikan baris pada dataset yang akan digunakan
- Kode baris 4 mendefinisikan pembagian antara data training dan data testing
- Kode baris 5 mendefinisikan proses preprocessing pada dataset
- Kode baris 6 mendefinisikan proses klasifikasi menggunakan algoritma K-Nearest Neighbors, serta menampilkan akurasi dengan rentangan nilai K untuk melihat nilai K terbaik

5.2.2.6 Implementasi Pembuatan Dataset

Pada implementasi pembuatan dataset, diambil data yang berasal dari 10 jenis madu. Kemudian akan dilakukan pengambilan data sebanyak 3 kali dari masing-masing jenis madu tersebut. Setelah terkumpul dengan total sebanyak 30 data,

selanjutnya data tersebut akan diberikan label LAYAK dan TIDAK LAYAK. Label ini diberikan berdasarkan dari pengamatan seorang pakar madu dari Peternakan Tawon Rimba Raya yang terletak di Kecamatan Lawang, Kabupaten Malang, Jawa Timur . Untuk lebih jelasnya mengenai implementasi pembuatan dataset disajikan melalui Tabel 5.4.

Tabel 5.4 Implementasi Pembuatan Dataset

No	Merek Madu	Fitur				Kelas
		C	R	G	B	
1	Madu Karet	3,849	57	42	110	LAYAK
2	Madu Karet	4,708	56	38	110	LAYAK
3	Madu Karet	3,564	55	46	110	LAYAK
4	Madu Alshifa	3,690	113	61	125	LAYAK
5	Madu Alshifa	3,785	113	61	117	LAYAK
6	Madu Alshifa	4,043	113	63	115	LAYAK
7	Madu Lebah Liar	3,284	88	52	103	TIDAK LAYAK
8	Madu Lebah Liar	3,346	89	44	103	TIDAK LAYAK
9	Madu Lebah Liar	3,070	88	41	103	TIDAK LAYAK
10	Madu Salsabila Kuning	3,501	147	120	137	TIDAK LAYAK
11	Madu Salsabila Kuning	3,161	144	116	134	TIDAK LAYAK
12	Madu Salsabila Kuning	3,564	148	119	134	TIDAK LAYAK
13	Madu Salsabila	3,439	84	28	97	TIDAK LAYAK
14	Madu Salsabila	3,377	79	33	88	TIDAK LAYAK
15	Madu Salsabila	3,253	82	26	87	TIDAK LAYAK
16	Madu Akasia	3,627	46	41	105	LAYAK
17	Madu Akasia	4,338	48	42	106	LAYAK
18	Madu Akasia	3,817	46	42	115	LAYAK
19	Madu Randu	3,692	64	38	108	LAYAK
20	Madu Randu	4,173	64	37	114	LAYAK

21	Madu Randu	3,946	60	34	104	LAYAK
22	Madu Hitam	3,753	41	37	110	LAYAK
23	Madu Hitam	3,658	38	33	106	LAYAK
24	Madu Hitam	4,010	38	34	116	LAYAK
25	Madu Rasa	3,284	114	63	121	TIDAK LAYAK
26	Madu Rasa	2,889	117	71	120	TIDAK LAYAK
27	Madu Rasa	3,100	113	65	115	TIDAK LAYAK
28	Madu TJ	3,315	57	37	116	TIDAK LAYAK
29	Madu TJ	3,070	59	46	109	TIDAK LAYAK
30	Madu TJ	2,800	57	38	115	TIDAK LAYAK

BAB 6 PENGUJIAN DAN ANALISIS

Dalam bab ini memiliki tujuan yaitu bisa melihat performa dan mengevaluasi kinerja, baik komponen penyusun sistem ataupun sistem secara keseluruhan. Hal-hal yang dibahas akan meliputi hasil dan juga analisis hasil pengujian.

6.1 Pengujian Sensor

Sensor-sensor yang tersusun dalam sistem diatur langkah-langkahnya dibagian ini. Pengujian ini bertujuan untuk melihat performa dan mengevaluasi kinerja masing-masing sensor.

6.1.1 Pengujian Sensor Kapasitansi

Pengujian ini diterapkan agar dapat melihat fungsionalitas beserta akurasi rata-rata dari sensor kapasitansi.

6.1.1.1 Prosedur Pengujian Sensor Kapasitansi

Adapun beberapa tahapan yang perlu dilakukan untuk melakukan pengujian sensor kapasitansi, antara lain:

1. Melakukan *wiring* kabel dari sensor kapasitansi ke Arduino
2. Menyiapkan Arduino IDE sebagai perangkat lunak pengolah kode program
3. Membuka file kode program untuk sensor kapasitansi
4. Menghidupkan Arduino Mega dengan cara menyambungkan kabel USB ke laptop atau catu daya lainnya
5. Melakukan pengunggahan kode program ke dalam Arduino Mega
6. Menyediakan kapasitor dengan kapasitas tertentu sebagai pembanding sensor kapasitansi
7. Melihat serial monitor yang terdapat di Arduino IDE dan mengamati hasil pembacaan sensor kapasitansi

6.1.1.2 Hasil dan Analisis Pengujian Sensor Kapasitansi

Adapun hasil yang didapatkan oleh pengujian sensor kapasitansi disajikan melalui Tabel 6.1.

Tabel 6.1 Hasil Pengujian Sensor Kapasitansi

Percobaan ke-	Kapasitor (Pembanding)	Sensor Kapasitansi	Persentase Akurasi (%)
1	24 pF	24,432 pF	98,20
2	24 pF	25,209 pF	94,97
3	24 pF	24,528 pF	97,80
4	330 μ F	320,85 μ F	97,27
5	330 μ F	312,02 μ F	94,55
Rata-rata Akurasi			96,56

Berdasarkan hasil pengujian yang dilakukan dengan menggunakan dua buah kapasitor dengan kapasitas 24 pF dan 330 μ F, didapatkan rata-rata akurasi sensor kapasitansi yaitu sebesar 96,56%. Oleh karena itu, dapat diasumsikan bahwa sensor kapasitansi dalam kondisi baik.

6.1.2 Pengujian Sensor TCS3200

Pengujian ini diterapkan agar dapat melihat fungsionalitas beserta akurasi rata-rata sensor TCS3200.

6.1.2.1 Prosedur Pengujian Sensor TCS3200

Adapun beberapa tahapan yang perlu dijalankan saat pengujian sensor TCS3200 ialah:

1. Melakukan *wiring* kabel dari sensor TCS3200 ke Arduino
2. Menyiapkan kertas berwarna merah, biru, dan hijau sesuai dengan Standar Warna Internasional dari RGB
3. Membuka *software* Arduino IDE
4. Menghidupkan Arduino Mega dengan cara menyambungkan kabel USB ke laptop atau catu daya lainnya
5. Masukkan kode program sensor TCS3200 untuk membaca warna
6. Memposisikan kertas berwarna merah tepat dibawah sensor dengan jarak terdekat dari sensor TCS3200
7. Amati dan catat nilai R, G, dan B yang muncul pada serial monitor
8. Ulangi langkah ke-6 sebanyak 5 kali pengujian
9. Selanjutnya ulangi langkah ke-6 sampai langkah ke-8 pada kertas hijau dan biru secara bergantian

10. Setelah data RGB didapat, lakukan perbandingan antara hasil pembacaan sensor TCS3200 dengan pembanding data Standar Warna Internasional dari nilai RGB

6.1.2.2 Hasil dan Analisis Pengujian Sensor TCS3200

Adapun hasil yang didapatkan oleh pengujian sensor TCS3200 disajikan melalui Tabel 6.2.

Tabel 6.2 Hasil Pengujian Sensor TCS3200

Warna	Hasil Nilai RGB yang diuji			Nilai Standar Warna Internasional RGB (Pembanding)			Tingkat Presisi(%)		
	R	G	B	R	G	B	R	G	B
Merah	240	205	141	255	0	0	94,12		
Merah	242	208	148	255	0	0	94,90		
Merah	240	205	143	255	0	0	94,12		
Merah	240	206	148	255	0	0	94,12		
Merah	242	210	151	255	0	0	94,90		
Hijau	161	244	164	0	255	0		95,69	
Hijau	165	246	164	0	255	0		96,47	
Hijau	172	249	166	0	255	0		97,65	
Hijau	172	248	169	0	255	0		97,25	
Hijau	167	248	169	0	255	0		97,25	
Biru	163	225	240	0	0	255			94,12
Biru	161	226	243	0	0	255			95,29
Biru	163	225	243	0	0	255			95,29
Biru	161	225	240	0	0	255			94,12
Biru	163	224	243	0	0	255			95,29
Rata-rata Presisi/Akurasi							95,37		

Berdasarkan hasil pengujian yang dilakukan dengan menggunakan Standar Warna Internasional dari RGB, didapatkan rata-rata akurasi sensor warna TCS3200 yaitu sebesar 95,37%. Oleh karena itu, dapat diasumsikan bahwa sensor warna TCS3200 ini dapat berfungsi dengan baik.

6.2 Pengujian Nilai K

Cross validatin yang dapat mencari nilai K terbaik dari dataset madu diterapkan dalam pengujian ini. Besar nilai K sangat berpengaruh terhadap akurasi dari hasil perhitungan algoritma *K-Nearest Neighbors*.

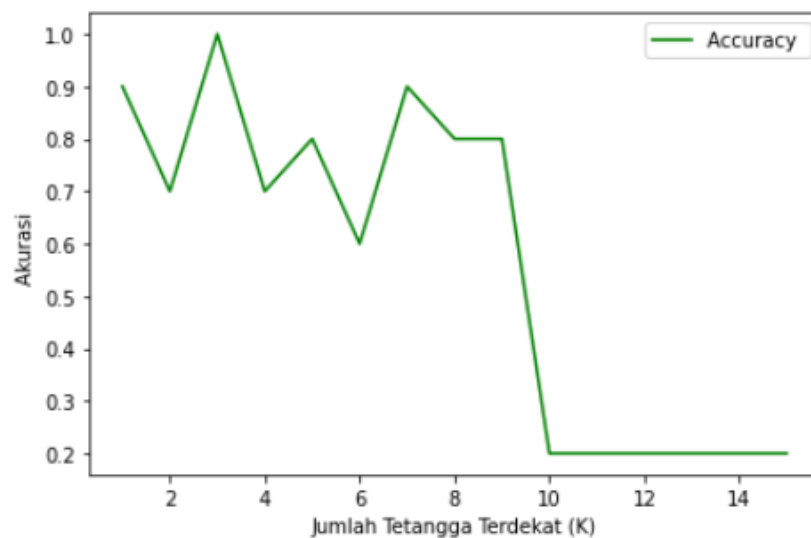
6.2.1 Prosedur Pengujian Nilai K

Nilai K atau jumlah tetangga terdekat yang terbaik dapat dicari menggunakan beberapa tahap ini:

1. Menyiapkan terminal atau Command Prompt
2. Ketikkan “Jupyter Notebook” pada terminal atau Command Prompt
3. Tunggu Jupyter Notebook akan terbuka secara otomatis pada *web browser*
4. Jika Jupyter Notebook telah tampil, masukkan kode program
5. Memasukkan data yang telah diklasifikasi dengan format file .csv
6. Menjalankan program satu per satu hingga akhir dan nantinya akan mendapatkan rentangan nilai K .

6.2.2 Hasil dan Analisis Pengujian Nilai K

Sejumlah 30 dataset madu telah diuji pada tahap ini. Dari 30 dataset madu tersebut, $1/3$ nya digunakan sebagai *data testing*. Adapun hasil dari pengujian nilai K menggunakan *cross validation* disajikan melalui Gambar 6.1.



Gambar 6.1 Grafik Hasil Pengujian Nilai K

(Sumber: Dokumen Pribadi)

Gambar 6.1 menyajikan jumlah tetangga terdekat atau nilai K yang disertakan memiliki cakupan dari 1 sampai 15. Untuk lebih jelasnya mengenai jumlah K beserta akurasi yang dihasilkan disajikan melalui Tabel 6.3.

Tabel 6.3 Hasil Pengujian Nilai K

Jumlah Tetangga Terdekat	Tingkat Presisi(%)
1	0,9
2	0,7
3	1,0
4	0,7
5	0,8
6	0,6
7	0,9
8	0,8
9	0,8
10	0,2
11	0,2
12	0,2
13	0,2
14	0,2
15	0,2

Tabel 6.3 menyajikan data yang didapat menggunakan dataset yang telah dibuat sebelumnya yaitu sebanyak 30 dataset dengan *data testing* 1/3 bagiannya. Berdasarkan hasil di Tabel 6.3, bahwa nilai K dengan akurasi terendah yaitu sebesar 20% berada pada nilai $K = 10$ sampai dengan $K = 15$. Sedangkan nilai K tertinggi dengan tingkat akurasi sebesar 100% berada pada $K = 3$. Oleh karena itu, dapat disimpulkan bahwasanya nilai $K=3$ menjadi nilai K terbaik yang tepat untuk digunakan dalam algoritma *K-Nearest Neighbors* pada sistem ini.

6.3 Pengujian Keseluruhan Sistem

Fungsionalitas dan tingkat presisi dari keseluruhan sistem dapat terlihat setelah melakukan pengujian ini. Hal tersebut meliputi pengujian terhadap setiap fungsi pada sub menu dan algoritma *K-Nearest Neighbors* dalam sistem.

6.3.1 Prosedur Pengujian Keseluruhan Sistem

Adapun tahapan yang perlu diterapkan dalam melakukan pengujian keseluruhan, antara lain:

1. Menghubungkan seluruh komponen yaitu RTC DS3231, Modul Micro SD, LCD, button, sensor kapasitansi, dan TCS3200 ke Arduino Mega
2. Menyiapkan perangkat lunak Arduino IDE
3. Membuka kode program untuk keseluruhan sistem
4. Menghidupkan Arduino Mega dengan cara menyambungkan kabel USB ke laptop atau catu daya lainnya
5. Melakukan pengunggahan kode program ke dalam Arduino
6. Jika sudah muncul tampilan menu awal pada LCD, cobalah setiap sub menu yang ada menggunakan button yang tersedia pada sistem
7. Amati dan catat hasil yang keluar dari setiap sub menu yang dipilih

6.3.2 Hasil dan Analisis Pengujian Keseluruhan Sistem

Dalam pengujian ini, penulis menerapkan dua buah pengujian antara lain pengujian fungsional dan tingkat presisi atau akurasi algoritma *K-Nearest Neighbors*.

6.3.2.1 Pengujian Fungsional

Pengujian fungsional berfungsi agar mengetahui sistem dapat berjalan sesuai yang diinginkan atau tidak. Secara umum, tujuan menggunakan pengujian fungsional adalah untuk melakukan pengecekan kesesuaian hasil yang didapatkan dengan hasil yang diinginkan. Adapun hasil pengujian fungsional dari keseluruhan sistem disajikan melalui Tabel 6.4.

Tabel 6.4 Hasil Pengujian Keseluruhan Sistem (Fungsional)

No	Skenario	Test Case	Hasil yang Diharapkan	Hasil Uji
1	Sistem ON	User menghubungkan kabel USB ke laptop sebagai catu daya sistem	Sistem dapat menyala	Berhasil
2	Memilih menu	User dapat memilih menu menggunakan button	Cursor pada menu dapat berpindah dari atas ke bawah	Berhasil
3	Menu Klasifikasi	User memilih menu klasifikasi	Sistem dapat menampilkan hasil klasifikasi	Berhasil

4	Menu History	User memilih menu history	Sistem dapat menampilkan history klasifikasi yaitu berupa jumlah konsumsi harian, total, dan waktunya.	Berhasil
5	Menu Reset Harian	User memilih menu reset harian	Sistem dapat melakukan reset pada jumlah konsumsi harian madu	Berhasil
6	Menu Reset Total	User memilih menu reset total	Sistem dapat melakukan reset pada jumlah konsumsi total madu	Berhasil
7	Sistem OFF	User melepaskan kabel USB dari laptop	Sistem dimatikan	Berhasil

Ketika sistem dinyalakan atau terhubung dengan sumber daya, sistem akan menampilkan menu utama. Pada menu utama terdapat empat buah sub menu yang dapat dipilih menggunakan button. Adapun tampilan pada menu utama sistem disajikan melalui Gambar 6.2.



Gambar 6.2 Menu Utama

(Sumber: Dokumen Pribadi)

Gambar 6.2 ialah penampakan menu utama yang terdiri dari sub menu klasifikasi, history, reset harian, dan reset total. Adapun keempat menu tersebut masing-masing akan dijelaskan secara rinci dimulai dari menu klasifikasi disajikan melalui Gambar 6.3.



Gambar 6.3 Menu Klasifikasi

(Sumber: Dokumen Pribadi)

Menu pertama adalah menu klasifikasi yang tertera pada Gambar 6.3. Pada menu ini berisikan proses klasifikasi sampel madu menggunakan algoritma *K-Nearest Neighbors*. Dapat dilihat pada Gambar 6.3, LCD menampilkan teks yang dimana saat itu juga sistem sedang melakukan training dataset.



Gambar 6.4 Menu Klasifikasi (Lanjutan)

(Sumber: Dokumen Pribadi)

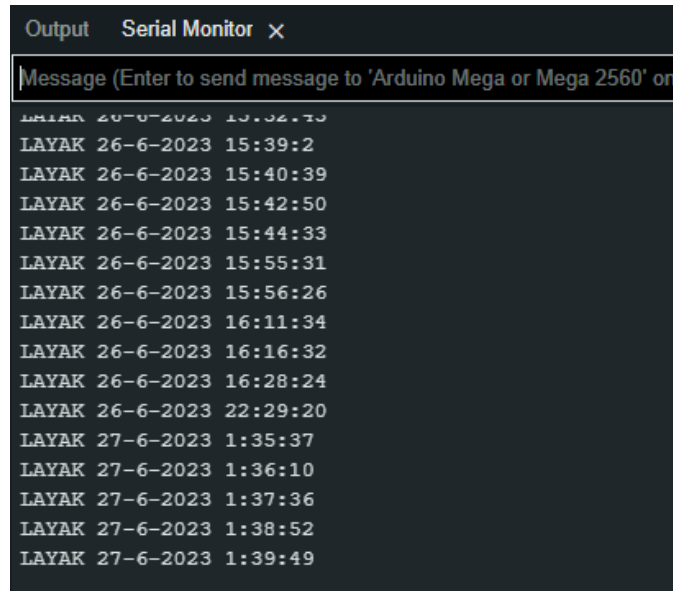
Kemudian pada Gambar 6.4, terdapat perintah untuk memasukkan madu melalui pintu kecil pada sistem. Perintah ini akan berlangsung selama 10 detik sesuai dengan kode program yang telah dibuat oleh penulis. Selanjutnya, hasil dari klasifikasi akan muncul beserta dengan nilai confidence yang dapat dilihat pada halaman lampiran pengujian.



Gambar 6.5 Tampilan Menu History

(Sumber: Dokumen Pribadi)

Menu kedua adalah menu history yang tertera pada Gambar 6.5. Pada menu ini berisikan tanggal dan waktu terakhir kali dilakukan pengukuran pada sistem, iterasi jumlah konsumsi madu harian, dan iterasi jumlah konsumsi madu total. Selain itu, pada menu ini juga terdapat datalog history pengukuran secara detail disajikan melalui Gambar 6.6.



Gambar 6.6 Tampilan History Klasifikasi

(Sumber: Dokumen Pribadi)

Gambar 6.6, terdapat datalog history pengukuran secara detail yang dapat diakses melalui serial monitor Arduino IDE. Hal ini karena datalog ini bersifat teknis, untuk keperluan penulis selaku pembuat kode program saja. Pada datalog history, hasil klasifikasi yang disimpan hanya yang memiliki kelas “LAYAK” saja. Hal tersebut berlaku juga dengan iterasi jumlah harian dan total konsumsi madu. Sedangkan ketika hasil klasifikasi madu memiliki kelas “TIDAK LAYAK”, maka tidak perlu disimpan ke dalam sistem karena sampel madu yang diujikan tersebut tidak direkomendasikan untuk dikonsumsi oleh penderita Diabetes mellitus.



Gambar 6.7 Menu Reset Harian

(Sumber: Dokumen Pribadi)

Menu ketiga adalah menu reset harian yang tertera pada Gambar 6.7. Pada menu ini berisikan berisikan fungsi untuk menghapus atau melakukan reset pada iterasi jumlah harian konsumsi madu.



Gambar 6.8 Menu Reset Harian (Lanjutan)

(Sumber: Dokumen Pribadi)

Ketika menu reset harian dipilih, maka sistem akan menampilkan teks pada Gambar 6.8. Kemudian sistem akan melakukan reset harian dengan cara mengganti nilai pada alamat 1 EEPROM Arduino Mega menjadi 0, akan disajikan melalui Gambar 6.9.



Gambar 6.9 Menu History Setelah Reset Harian

(Sumber: Dokumen Pribadi)

Pada Gambar 6.9, data harian yang semula memiliki nilai 2 sudah berubah menjadi nilai 0. Hal ini berarti fungsi untuk melakukan reset harian telah berhasil dilakukan.



Gambar 6.10 Menu Reset Total

(Sumber: Dokumen Pribadi)

Menu keempat adalah menu reset total yang tertera pada Gambar 6.10. Pada menu ini berisikan berisikan fungsi untuk menghapus atau melakukan reset pada iterasi jumlah total konsumsi madu.



Gambar 6.11 Menu Reset Total (Lanjutan)

(Sumber: Dokumen Pribadi)

Ketika menu reset total dipilih, maka sistem akan menampilkan teks pada Gambar 6.11. Kemudian sistem akan melakukan reset total dengan cara mengganti nilai pada alamat 0 dan 1 EEPROM Arduino Mega menjadi 0, akan disajikan melalui Gambar 6.12.



Gambar 6.12 Menu History Setelah Reset Total

(Sumber: Dokumen Pribadi)

Berdasarkan hasil dari Tabel 6.4 dan penjelasan melalui Gambar 6.2 sampai dengan Gambar 6.12, maka dapat disimpulkan bahwa sistem dapat berjalan

dengan baik. Hal ini dapat dibuktikan dengan seluruh bagian fungsional sistem yang dapat dijalankan. Sehingga keluaran yang didapat dari sistem memiliki kesesuaian dengan keluaran yang diinginkan penulis.

6.3.2.2 Pengujian Akurasi KNN

Pengujian akurasi algoritma *K-Nearest Neighbors* sangat diperlukan karena ini yang akan menjadi penentu tingkat kesuksesan sistem dalam menjalankan fungsi utamanya, yaitu melakukan klasifikasi pada madu. Untuk mencari rata-rata akurasi, dilakukan pengujian terhadap 10 sampel uji yang sebelumnya sudah disebutkan dalam bab sebelumnya. Dimana 10 sampel uji tersebut sudah divalidasi oleh seorang pakar madu. Adapun hasil dari pengujian akurasi algoritma KNN disajikan melalui Tabel 6.5.

Tabel 6.5 Hasil Pengujian Keseluruhan Sistem (Akurasi)

No	Merek Madu	Kelas Uji	Keluaran Sistem	Kesesuaian
1	Madu X1	LAYAK	TIDAK LAYAK	Tidak Sesuai
2	Madu X2	LAYAK	LAYAK	Sesuai
3	Madu X3	LAYAK	LAYAK	Sesuai
4	Madu X4	LAYAK	LAYAK	Sesuai
5	Madu X5	LAYAK	LAYAK	Sesuai
6	Madu Y1	TIDAK LAYAK	TIDAK LAYAK	Sesuai
7	Madu Y2	TIDAK LAYAK	TIDAK LAYAK	Sesuai
8	Madu Y3	TIDAK LAYAK	TIDAK LAYAK	Sesuai
9	Madu Y4	TIDAK LAYAK	TIDAK LAYAK	Sesuai
10	Madu Y5	TIDAK LAYAK	TIDAK LAYAK	Sesuai
Akurasi				90%

Berdasarkan hasil pengujian akurasi algoritma *K-Nearest Neighbors* pada Tabel 6.5, dapat dilihat bahwa dengan 10 kali pengujian sistem mampu mendapatkan tingkat presisi atau akurasi mencapai 90%. Oleh karena itu, dapat disimpulkan bahwa Rancang Bangun Sistem Klasifikasi Kelayakan Madu Berdasarkan Kadar Gula dan Warna untuk Penderita Diabetes Mellitus menggunakan Metode *K-Nearest Neighbors* ini telah bekerja secara optimal.

BAB 7 PENUTUP

Pada bab 7 atau bab penutup ialah bab ketujuh dalam penelitian ini yang berisikan uraian terkait dengan kesimpulan yang dapat dipetik dan menjawab pertanyaan pada rumusan masalah. Kemudian juga disertai dengan adanya saran-saran untuk penelitian selanjutnya guna memaksimalkan hasil dari penelitian yang dilakukan oleh penulis.

7.1 Kesimpulan

Pada akhir dari penelitian ini, penulis akan menarik beberapa kesimpulan berlandaskan hasil tahap pengujian serta analisis yang sudah dikerjakan. Hal ini bertujuan untuk menjawab pertanyaan-pertanyaan yang tercantum pada rumusan masalah yang sebelumnya juga sudah ditentukan pada bab pertama. Adapun beberapa kesimpulan tersebut antara lain:

1. Sensor kapasitansi memiliki rata-rata akurasi mencapai 96,56%. Sedangkan sensor warna TCS3200 memiliki rata-rata akurasi mencapai 95,37%. Kemudian untuk komponen lainnya seperti RTC, Modul Micro SD, dan button juga dapat berjalan dengan baik pada saat pengujian sistem..
2. Nilai K terbaik dari algoritma *K-Nearest Neighbors* pada sistem dapat dicari menggunakan *cross validation* pada dataset madu yang sudah dilabeli kelasnya oleh seorang pakar madu. Setelah melalui pengujian tersebut, didapatkan jumlah tetangga atau nilai K terbaik berada di $K=3$ dengan akurasi sebesar 100%.
3. Pada pengujian fungsional, seluruh skenario yang diberikan telah berhasil dijalankan dengan tepat sehingga keluaran yang didapat pada sistem sudah sesuai dengan keluaran yang diinginkan oleh penulis. Hal ini terbukti dari seluruh sub menu telah memiliki fungsionalitas yang baik ketika dipilih. Kemudian untuk pengujian akurasi keseluruhan sistem, algoritma *K-Nearest Neighbors* juga memberikan performa yang baik. Itu terbukti karena setelah melakukan pengujian akurasi, sistem ini mendapatkan akurasi sebesar 90% untuk 10 sampel uji.

7.2 Saran

Adapun saran-saran oleh penulis yang berguna bagi penelitian mendatang, demi memaksimalkan hasil dari sistem ini adalah sebagai berikut:

1. Untuk memaksimalkan sistem ini, kedepannya bisa menggunakan dataset madu yang lebih banyak lagi. Serta menambahkan sensor lain agar sistem menjadi lebih akurat lagi.
2. Untuk *cover* mungkin bisa dibuat seminimalis mungkin. Hal ini berguna agar sistem bersifat lebih portabel dan memudahkan pengguna.

3. Untuk perkembangan teknologi kedepannya, mungkin bisa menggunakan algoritma *machine learning* lain yang lebih kompleks atau mungkin *deep learning*.

DAFTAR REFERENSI

- Adrian, K., 2020. Komplikasi Diabetes Melitus Bisa Menyerang Mata Hingga Ujung Kaki. Diambil dari <https://www.alodokter.com/komplikasi-diabetes-melitus-bisa-menyerang-mata-hingga-ujung-kaki>
- Alhamdani, M., Syauqy, D., & Prasetyo, B., 2022. Sistem Klasifikasi Kualitas Jenis-Jenis Madu berdasarkan Warna, Kecerahan, dan pH menggunakan Metode JST Backpropagation. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(6), 2584-2590. Diambil dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11119>
- Evahelda, E., Pratama, F., Santoso, B., 2017. Sifat Fisik dan Kimia Madu dari Nektar Pohon Karet di Kabupaten Bangka Tengah, Indonesia. *Jurnal AGRITECH*, 37(4), 363-368. Diambil dari <https://doi.org/10.22146/agritech.16424>
- Fadli, R., 2022. Kenali Perbedaan Prediabetes dan Diabetes. Diambil dari <https://www.halodoc.com/artikel/kenali-perbedaan-prediabetes-dan-diabetes>
- Fitriani, F. & Sanghati, S., 2021. Intervensi Gaya Hidup Terhadap Pencegahan Diabetes Melitus Tipe 2 Pada Pasien Pra Diabetes. *Jurnal Ilmiah Kesehatan Sandi Husada*, 10(2). Diambil dari <https://doi.org/10.35816/jiskh.v10i2.682>
- Handayani, V., 2021. Benarkah Madu Aman untuk Dikonsumsi Pengidap Diabetes. Diambil dari <https://www.halodoc.com/artikel/benarkah-madu-aman-untuk-dikonsumsi-pengidap-diabetes>
- Hartini, V., 2021. Rancang Bangun Alat Pendeteksi Kadar Gula pada Minuman Berperisa menggunakan Sensor Kapasitif. *Jurnal Fokus Elektroda Jurnal Fokus Elektroda : Energi Listrik, Telekomunikasi, Komputer, Elektronika dan Kendali*, 6(3), 149-152. Diambil dari <http://ojs.uho.ac.id/index.php/JFE/article/view/19899>
- Indrayanti, I., Sugianti, D., Karomi, A., A., 2017. Optimasi Parameter K pada Algoritma K-nearest Neighbour untuk Klasifikasi Penyakit Diabetes Mellitus. *PROSIDING SNATIF*. Diambil dari <https://jurnal.umk.ac.id/index.php/SNA/article/view/1456/993>
- Jyothi, M., et al., 2012. Implementation of SPI Communication Protocol for Multipurpose Applications with I2C Power and Area Reduction. / *International Journal of Engineering Research and Applications (IJERA)*, 2(2), 875-883. Diambil dari https://www.academia.edu/29405623/IJETR032452_pdf
- Kaushal, C., Koundal, D., 2019. Recent Trends In Big Data Using Hadoop. *International Journal of Informatics and Communication Technology (IJ-ICT)*, 8(1), 39-49. Diambil dari https://www.researchgate.net/publication/332541933_Recent_trends_in_big_data_using_hadoop

- Mnati, M., J., et al., 2021. An Open-Source Non-Contact Thermometer Using Low-Cost Electronic Components. *HardwareX*, 9, 2468-0672. Diambil dari <https://doi.org/10.1016/j.ohx.2021.e00183>
- More, A., et al., 2019. Soil Analysis Using IoT. *2nd International Conference on Advances in Science & Technology (ICAST)*. Diambil dari <http://dx.doi.org/10.2139/ssrn.3366756>
- Nareza, M., 2020. Ini Anjuran Konsumsi Gula, Garam, dan Lemak per Hari. Diambil dari <https://www.alodokter.com/ini-anjuran-konsumsi-gula-garam-dan-lemak-per-hari>
- Olaoye, J. O., Sunmonu, S. E., 2011. Development of Management Tool And Computerization of Mechanized Aquacultural Farm Operations. *Department of Agricultural and Biosystems Engineering, University of Ilorin, Ilorin, Kwara State, Nigeria*, 5(10), 3322-3332. Diambil dari https://www.researchgate.net/publication/269113570_Development_of_Management_Tool_and_Computerization_of_Mechanized_Aquacultural_Farm_Operations
- Pratama, C., L., et al., 2023. Sistem Deteksi Kelayakan Konsumsi pada Air Isi Ulang dengan Metode K-Nearest Neighbors berbasis Arduino. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(3), 1273-1280. Diambil dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12433>
- Purnamasari, P., 2017. PEMBUATAN ALAT UKUR KADAR GULA BERBASIS KAPASITANSI DENGAN MENGGUNAKAN ARDUINO UNO. *UT-Faculty of Mathematics and Natural Sciences*. Diambil dari <http://repository.unej.ac.id/handle/123456789/83167>
- Rachmawati, N., 2015. Gambaran Kontrol dan Kadar Gula Darah pada Pasien Diabetes Melitus di Poliklinik Penyakit Dalam RSJ Prof. Dr. Soerojo Magelang. Diambil dari <http://eprints.undip.ac.id/51927/>
- Rukmana, R., M., Sulistyawati, D., Herawati, R., 2019. PENYULUHAN PENGATURAN KONSUMSI MAKANAN SEHAT DAN PEMERIKSAAN GLUKOSA DARAH DI KELOMPOK POSYANDU LANSIA RW 18 PERUMNAS MOJOSONGO, SURAKARTA, JAWA TENGAH. *JURNAL CEMERLANG : Pengabdian pada Masyarakat*, 2(1), 1-12. Diambil dari <https://doi.org/10.31540/jpm.v2i1.194>
- Rivki, M., Bachtiar, A., 2017. IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR DALAM PENGKLASIFIKASIAN FOLLOWER TWITTER YANG MENGGUNAKAN BAHASA INDONESIA. *Jurnal Sistem Informasi*, 1(13), 31-37. Diambil dari <https://jsi.cs.ui.ac.id/index.php/jsi/article/view/500/342>
- Rochman, S., Mukhtar, M., 2018. Klasifikasi Kualitas Madu Lebah menggunakan Sistem Spektrofotometer dan Machine Learning Berbasis Single Board Computer. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)*. <https://karyailmiah.unipasby.ac.id/2019/01/18/klasifikasi-kualitas-madu->

lebih-menggunakan-sistem-spektrofotometer-dan-machine-learning-berbasis-single-board-computer/

- Stefanov, G., et al., 2020. Measurement and Visualization on Analog Signals with Microcomputers Connection. *Balkan Journal of Applied Mathematics and Informatics*, 3 (1). pp. 85-94. ISSN 2545-4803. Diambil dari <https://eprints.ugd.edu.mk/id/eprint/24148>
- Syam, S., & Mustika, N., 2018. PROTOTYPE ALAT BANTU DETEKSI POLA WARNA UNTUK PENDERITA BUTA WARNA. *Journal of Technology Research in Information System and Engineering*, 5(2), 1-7. Diambil dari <https://garuda.kemdikbud.go.id/documents/detail/3001423>
- Tangkelayuk, A., & Mailoa, E., 2022. Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree. *Jurnal Teknik Informatika dan Sistem Informasi*, 9(2), 1109-1119. Diambil dari <https://doi.org/10.35957/jatisi.v9i2.2048>
- Tasmalinda, 2022. Praktik Industri Rumahan Madu Oplosan Digerebek, Madu Asli Hanya 5 Persen Selebihnya Dioplos. Diambil dari <https://sumsel.suara.com/read/2022/05/22/092721/praktik-industri-rumahan-madu-oplosan-digerebek-madu-asli-hanya-5-persen-selebihnya-dioplos?page=1>

LAMPIRAN A KODE PROGRAM

A.1 Kode Program Keseluruhan Sistem

Program Keseluruhan Sistem	
<pre>//Library----- ----- #include <RTClib.h> //library RTC untuk setting waktu #include <Wire.h> //Library wire #include <LiquidCrystal_I2C.h> //Library LCD dengan modul I2C #include <SPI.h> //Library untuk komunikasi serial, miso mosi, sck #include <SD.h> //Library untuk melakukan penyimpanan data ke micro sd #include <Arduino_KNN.h> //library knn #include <EEPROM.h> //lib eeprom KNNClassifier kelayakanMadu(2); //kelayakan madu 2 kelas (layak/tidak layak) int classification; int jumlahTotal; //EE 0 int jumlahHarian; //EE 1 int lastDay; //EE 2 int lastMonth; //EE 3 int lastHour; //EE 4 int lastMinute; //EE 5 int lastSecond; //EE 6 int lastYear; int thisDay; //PushButton----- ----- int upButton = 11; //button hijau int downButton = 12; //button merah int selectButton = 13; //button kuning int menu = 1; //loop menu</pre>	

```

//RTC-----
-----
RTC_DS3231 rtc;


//Micro SD-----
-----

const int chipSelect = 53;           //inisialisasi pin CS sd
card = pin digital 53 arduino mega
File myFile;                         //inisialisasi sebagai
val. myFile


//Kapasitansi-----
-----

const int OUT_PIN = A14;
const int IN_PIN = A15;
const float IN_STRAY_CAP_TO_GND = 24.48;
const float IN_CAP_TO_GND = IN_STRAY_CAP_TO_GND;
const float R_PULLUP = 34.8;
const int MAX_ADC_VALUE = 1023;
float capacitance;


//TCS3200-----
-----

#define S0 27
#define S1 29
#define S2 31
#define S3 33
#define sensorOut 35
// Stores frequency read by the photodiodes
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
// Stores the red. green and blue colors
int redColor = 0;
int greenColor = 0;
int blueColor = 0;


//KNN-----
-----

const int fi = 30; // Banyak Data

```

```

const int co = 4; // Banyak Fitur
float dataset[fi][co] = {
    {3.690, 113, 61, 125}, //1
    {3.785, 113, 61, 117}, //2
    {4.043, 113, 63, 115}, //3

    {3.849, 57, 42, 110}, //4
    {4.708, 56, 38, 110}, //5
    {3.564, 55, 46, 110}, //6

    {3.627, 46, 41, 105}, //7
    {4.338, 48, 42, 106}, //8
    {3.817, 46, 42, 115}, //9

    {3.692, 64, 38, 108}, //10
    {4.173, 64, 37, 114}, //11
    {3.946, 60, 34, 104}, //12

    {3.753, 41, 37, 110}, //13
    {3.658, 38, 33, 106}, //14
    {4.010, 38, 34, 116}, //15
    //=====

    {3.284, 88, 52, 103}, //16
    {3.346, 89, 44, 103}, //17
    {3.070, 88, 41, 103}, //18

    {3.501, 147, 120, 137}, //19
    {3.161, 144, 116, 134}, //20
    {3.564, 148, 119, 134}, //21

    {3.439, 84, 28, 97}, //22
    {3.377, 79, 33, 88}, //23
    {3.253, 82, 26, 87}, //24

    {3.284, 114, 63, 121}, //25
    {2.889, 117, 71, 120}, //26
    {3.100, 113, 65, 115}, //27

```

```
    {3.315, 57, 37, 116}, //28  
    {3.070, 59, 46, 109}, //29  
    {2.800, 57, 38, 115} //30  
};
```

```
// Data Class
```

```
int classes[fi] = {
```

```
    1, //1
```

```
    1, //2
```

```
    1, //3
```

```
    1, //4
```

```
    1, //5
```

```
    1, //6
```

```
    1, //7
```

```
    1, //8
```

```
    1, //9
```

```
    1, //10
```

```
    1, //11
```

```
    1, //12
```

```
    1, //13
```

```
    1, //14
```

```
    1, //15
```

```
    0, //16
```

```
    0, //17
```

```
    0, //18
```

```
    0, //19
```

```
    0, //20
```

```
    0, //21
```

```
    0, //22
```

```
    0, //23
```

```
    0, //24
```

```
    0, //25
```

```
    0, //26
```

```
    0, //27
```

```
    0, //28
```

```

    0, //29
    0 //30
};

//LCD-----
-----
LiquidCrystal_I2C lcd(0x3F, 20, 4);    //address LCD 20x4
//=====
=====

//void                                         setup
starts=====
void setup() {
    Serial.begin(9600);
    while (!Serial) {
        ;
    }
}
//Setting Time RTC-----
-----
    rtc.begin();
    // rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

//Cek Micro SD-----
-----

    Serial.println(F("Read MicroSd Card"));
    delay(1000);
    if (!SD.begin(chipSelect))
    {
        Serial.println(F("Failed read microsd card on module!"));
        return;
    }
    else {Serial.println(F("Success read microsd card"));}
    delay(1000);

//Kapasitansi-----
-----

    pinMode(OUT_PIN, OUTPUT);
    pinMode(IN_PIN, INPUT);

```

```

//TCS3200-----
-----

pinMode(S0, OUTPUT);           // Setting the outputs
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(sensorOut, INPUT);     // Setting the sensorOut as an
input
digitalWrite(S0,HIGH);         // Setting frequency scaling to
20%
digitalWrite(S1,LOW);

//LCD-----
-----

lcd.init();
lcd.backlight();

//PushButton-----
-----

pinMode(upButton, INPUT_PULLUP);
pinMode(downButton, INPUT_PULLUP);
pinMode(selectButton, INPUT_PULLUP);
lcd.setCursor(2, 1);
lcd.print(F("SELAMAT DATANG.."));
delay(5000);
Serial.print(F("Initializing SD card..."));

//EEPROM-----
-----

jumlahTotal = EEPROM.read(0);
jumlahHarian = EEPROM.read(1);
lastDay = EEPROM.read(2);
lastMonth = EEPROM.read(3);
lastHour = EEPROM.read(4);
lastMinute = EEPROM.read(5);
lastSecond = EEPROM.read(6);

//Masuk Menu-----
-----

lcd.clear();

```

```

    updateMenu();
}

//void                                                                    setup
ends=====

//void                                                                    loop
starts=====
void loop() {

//PushButton-----
-----

    if (!digitalRead(downButton)){
        menu++;
        updateMenu();
        delay(100);
        while (!digitalRead(downButton));
    }
    if (!digitalRead(upButton)){
        menu--;
        updateMenu();
        delay(100);
        while (!digitalRead(upButton));
    }
    if (!digitalRead(selectButton)){
        executeAction();
        updateMenu();
        delay(100);
        while (!digitalRead(selectButton));
    }

}

//void                                                                    loop
ends=====

//funksi                                                                    update                                                                    menu
starts=====
void updateMenu() {

```



```

switch (menu) {
    case 0:
        menu = 1;
        break;
    case 1:
        lcd.clear();
        lcd.print(F("=> Klasifikasi"));
        lcd.setCursor(0, 1);
        lcd.print(F("    History"));
        lcd.setCursor(0, 2);
        lcd.print(F("    Reset Harian"));
        lcd.setCursor(0, 3);
        lcd.print(F("    Reset Total"));
        break;
    case 2:
        lcd.clear();
        lcd.print(F("    Klasifikasi"));
        lcd.setCursor(0, 1);
        lcd.print(F("=> History"));
        lcd.setCursor(0, 2);
        lcd.print(F("    Reset Harian"));
        lcd.setCursor(0, 3);
        lcd.print(F("    Reset Total"));
        break;
    case 3:
        lcd.clear();
        lcd.print(F("    Klasifikasi"));
        lcd.setCursor(0, 1);
        lcd.print(F("    History"));
        lcd.setCursor(0, 2);
        lcd.print(F("=> Reset Harian"));
        lcd.setCursor(0, 3);
        lcd.print(F("    Reset Total"));
        break;
    case 4:
        lcd.clear();
        lcd.print(F("    Klasifikasi"));
        lcd.setCursor(0, 1);

```

```

        lcd.print(F("    History"));
        lcd.setCursor(0, 2);
        lcd.print(F("    Reset Harian"));
        lcd.setCursor(0, 3);
        lcd.print(F("=> Reset Total"));
        break;
    case 5:
        menu = 4;
        break;
    }
}

//fungsi                                update                                menu
ends=====

//fungsi                                sub                                menu
starts=====

void executeAction() {
    switch (menu) {
        case 1:
            action1();
            break;
        case 2:
            action2();
            break;
        case 3:
            action3();
            break;
        case 4:
            action4();
            break;
    }
}

//fungsi                                sub                                menu
ends=====

//sub
menu=====
=====

```

```

//sub menu 1-----
-----
void action1() {
    lcd.clear();
    Serial.println("Starting KNN");
    for(int i=0; i<fi; i++){ //i untuk mengurutkan dataset
        delay(200);
        Serial.print("add data ");
        int a = classes[i];
        Serial.println(i);
        lcd.setCursor(4, 1);
        lcd.print("Processing...");
        kelayakanMadu.addExample(dataset[i], a);
    }
    Serial.println();
    Serial.print("\tBanyak Data Latih = ");
    Serial.print(kelayakanMadu.getCount());
    Serial.println();
    lcd.clear();
    lcd.setCursor(6, 0);
    lcd.print("SILAKAN");
    lcd.setCursor(3, 1);
    lcd.print("MASUKKAN MADU");
    delay(10000);
    lcd.clear();

    int x = 0;
    do{
//Kapasitansi-----
-----
        pinMode(IN_PIN, INPUT);
        digitalWrite(OUT_PIN, HIGH);
        int val = analogRead(IN_PIN);
        digitalWrite(OUT_PIN, LOW);

        if (val < 1000){
            pinMode(IN_PIN, OUTPUT);
            float capacitance = (float)val * IN_CAP_TO_GND /
(float)(MAX_ADC_VALUE - val);

```

```

        Serial.print(F("Capacitance = "));
        Serial.print(capacitance, 3);
        Serial.print(F("pF "));
        Serial.print(F(" || "));
    }else{
        pinMode(IN_PIN, OUTPUT);
        delay(1);
        pinMode(OUT_PIN, INPUT_PULLUP);
        unsigned long u1 = micros();
        unsigned long t;
        int digVal;
        do{
            digVal = digitalRead(OUT_PIN);
            unsigned long u2 = micros();
            t = u2 > u1 ? u2 - u1 : u1 - u2;
        }
        while ((digVal < 1) && (t < 400000L));

        pinMode(OUT_PIN, INPUT);
        val = analogRead(OUT_PIN);
        digitalWrite(IN_PIN, HIGH);
        int dischargeTime = (int)(t / 1000L) * 5;
        delay(dischargeTime);
        pinMode(OUT_PIN, OUTPUT);
        digitalWrite(OUT_PIN, LOW);
        digitalWrite(IN_PIN, LOW);
        float capacitance = -(float)t / R_PULLUP / log(1.0 -
(float)val / (float)MAX_ADC_VALUE);
        Serial.print(F("Capacitance = "));
        if (capacitance > 1000.0){
            Serial.print(capacitance / 1000.0, 2);
            Serial.print(F("uF "));
            Serial.print(F(" || "));
        }else{
            Serial.print(capacitance, 2);
            Serial.print(F("nF "));
            Serial.print(F(" || "));
        }
    }
}

```

```

    while (millis() % 1000 != 0);
//Kapasitansi ends-----
-----

//TCS3200-----
-----

    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);
    redFrequency = pulseIn(sensorOut, LOW);
    redColor = map(redFrequency, 255, 70, 0, 255);
    Serial.print("R = ");
    Serial.print(redColor);
    Serial.print(F(" || "));
    delay(100);

    digitalWrite(S2,HIGH);
    digitalWrite(S3,HIGH);
    greenFrequency = pulseIn(sensorOut, LOW);
    greenColor = map(greenFrequency, 255, 65, 0, 255);
    Serial.print("G = ");
    Serial.print(greenColor);
    Serial.print(F(" || "));
    delay(100);

    digitalWrite(S2,LOW);
    digitalWrite(S3,HIGH);
    blueFrequency = pulseIn(sensorOut, LOW);
    blueColor = map(blueFrequency, 255, 45, 0, 255);
    Serial.print("B = ");
    Serial.println(blueColor);
    delay(100);
//TCS3200 ends-----
-----

//KNN Klasifikasi-----
-----

    delay(100);
    float input[] = { capacitance, redColor, greenColor, blueColor
};
    classification = kelayakanMadu.classify(input, 3);

```

```

float confidence = kelayakanMadu.confidence();
Serial.print("\tclassification = ");
Serial.println(classification);
Serial.print("\tconfidence      = ");
Serial.println(confidence);
lcd.clear();
if (classification == 1){
    lcd.setCursor(7, 0);
    lcd.print("LAYAK");
    jumlahTotal ++;
    EEPROM.write(0, jumlahTotal); //saving to EEPROM

    DateTime waktu_sekarang = rtc.now();
    thisDay = waktu_sekarang.day();
    if (thisDay != lastDay){
        jumlahHarian = 0;
        EEPROM.write(1, jumlahHarian); //saving to EEPROM
    }
    jumlahHarian ++;
    EEPROM.write(1, jumlahHarian); //saving to EEPROM
    lastDay = waktu_sekarang.day();          EEPROM.write(2,
lastDay);
    lastMonth = waktu_sekarang.month();      EEPROM.write(3,
lastMonth);
    lastHour = waktu_sekarang.hour();        EEPROM.write(4,
lastHour);
    lastMinute = waktu_sekarang.minute();    EEPROM.write(5,
lastMinute);
    lastSecond = waktu_sekarang.second();    EEPROM.write(6,
lastSecond);
    lastYear = waktu_sekarang.year();
}else{
    lcd.setCursor(5, 0);
    lcd.print("TIDAK LAYAK");
}
lcd.setCursor(2, 1);
lcd.print("Confidence: ");
lcd.print(confidence);
x++;
delay(5000);

```

```

    } while (x <= 0);
//KNN Klasifikasi ends-----
-----

//Saving data-----
-----

    DateTime waktu_sekarang = rtc.now();
    File myFile = SD.open("datalog.txt", FILE_WRITE); //Membuka
    File datalogger.txt
    if (myFile){
        if (classification == 1){
            myFile.print("LAYAK");          myFile.print(" ");
            myFile.print(waktu_sekarang.day());          myFile.print("-");
        };
            myFile.print(waktu_sekarang.month());
        myFile.print("-");
            myFile.print(waktu_sekarang.year());          myFile.print("
");
            myFile.print(waktu_sekarang.hour());
        myFile.print(":");
            myFile.print(waktu_sekarang.minute());
        myFile.print(":");
            myFile.print(waktu_sekarang.second());
        myFile.println("");
    }
    myFile.close();
}
    delay(1000);
}

//sub menu 1 ends-----
-----

//sub menu 2-----
-----

void action2() {
    lcd.clear();
    DateTime waktu_sekarang = rtc.now();
    File myFile = SD.open("datalog.txt", FILE_WRITE);
    if (myFile){
        lcd.print("Data Terakhir:");
        lcd.setCursor(3, 1);
        lcd.print(lastDay);    lcd.print("-");
    }
}

```

```

        lcd.print(lastMonth);  lcd.print("-");
        lcd.print(waktu_sekarang.year());  lcd.print(" ");
        lcd.print(lastHour);  lcd.print(":");
        lcd.print(lastMinute); lcd.print(":");
        lcd.print(lastSecond); lcd.print("");

        jumlahTotal = EEPROM.read(0);
        jumlahHarian = EEPROM.read(1);
        lcd.setCursor(0, 2);
        lcd.print("Data Harian: ");
        lcd.print(jumlahHarian);
        lcd.setCursor(0, 3);
        lcd.print("Data Total: ");
        lcd.print(jumlahTotal);
        delay(5000);
        readSD();
        myFile.close();
    }else{
        Serial.println(F("Failed open datalogger.txt")); // jika
        gagal print error
    }
    delay(5000);
}

//sub menu 2 ends-----
-----

//sub menu 3-----
-----

void action3() {
    lcd.clear();
    lcd.setCursor(4, 1);
    lcd.print("RESET HARIAN");
    Serial.println("Removing the data...");
    delay(3000);
    lcd.clear();
    lcd.setCursor(5, 1);
    lcd.print("Menghapus");
    lcd.setCursor(4, 2);
    lcd.print("Data Harian");

```



```

EEPROM.write(1, 0);
delay(3000);
}
//sub menu 3 ends-----
-----

//sub menu 4-----
-----

void action4() {
    lcd.clear();
    lcd.setCursor(4, 1);
    lcd.print("RESET TOTAL");
    Serial.println("Removing the data...");
    delay(3000);
    lcd.clear();
    lcd.setCursor(5, 1);
    lcd.print("Menghapus");
    lcd.setCursor(5, 2);
    lcd.print("Data Total");
    EEPROM.write(0, 0);
    EEPROM.write(1, 0);
    delay(3000);
}
//sub menu 4 ends-----
-----

//Fungsi Read SD-----
-----

void readSD(){
    myFile = SD.open("datalog.txt");
    if (myFile){
        Serial.println("Isi File datalog.txt:");
        while (myFile.available()) {
            Serial.write(myFile.read());
        }
        myFile.close();
    }
    else{
        Serial.println("Buka file datalog.txt Error!");
    }
}

```

```
}  
//Fungsi Read SD ends-----  
-----
```

LAMPIRAN B DOKUMENTASI

B.1 Dokumentasi dengan Pakar Madu





B.2 Dokumentasi Pengujian Sistem

