

**PENERAPAN METODE NAIVE BAYES DALAM SISTEM PENDETEKSI
KUALITAS TANAH PADA TANAMAN KEDELAI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Johannes Riski Sitinjak
NIM: 195150301111021



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

PENGESAHAN

PENERAPAN METODE NAIVE BAYES DALAM SISTEM PENDETEKSI KUALITAS TANAH
PADA TANAMAN KEDELAI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh:
Johannes Riski Sitinjak
NIM: 195150301111021

Skripsi ini telah diuji dan dinyatakan lulus pada
24 Juli 2023

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1



Mochammad Hannats Hanafi Ichsan, S.ST., M.T
NIP: 198812292019031010

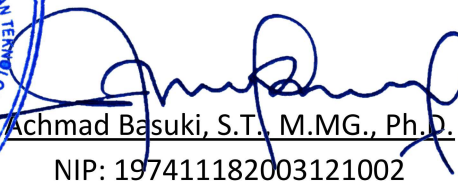
Dosen Pembimbing 2



Eko Setiawan, S.T., M.Eng., Ph.D
NIP: 198706102022031001

Mengetahui:

Ketua Departemen Teknik Informatika



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

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenarbenarnya 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 unsurunsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundangundangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 27 Juli 2023



Johannes Riski Sitinjak

NIM:195150301111021

PRAKATA

Puji syukur kepada Tuhan yang Maha Esa yang telah memberikan kelancaran dan kemudahan kepada penulis sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul Penerapan Metode Naïve Bayes dalam Sistem Pendeteksi Kualitas Tanah pada Tanaman Kedelai. Penulis sangat menyadari bahwa skripsi ini tidak dapat diselesaikan tanpa adanya bantuan dan masukan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan banyak terima kasih kepada:

1. Bapak Mochammad Hannats Hanafi Ichsan, S.ST., M.T dan Bapak Eko Setiawan, S.T., M.Eng., Ph.D selaku dosen pembimbing yang sudah banyak memberikan masukan, bimbingan dan mengarahkan penulis serta dukungan dan semangat sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Achmad Basuki, S.T., M.MG., Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya.
3. Bapak Barlian Henryranu Prasetio, S.T., M.T., PH.D. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya.
4. Bapak Aden Sitinjak dan Ibu Nurtianna Simamora selaku orangtua penulis yang tidak pernah berhenti mendoakan, memberikan semangat dan yang menjadi motivasi bagi penulis dalam menyelesaikan skripsi ini.
5. Eva Helen Sitinjak, Tien Pratiwi Sitinjak, Dick Wanda Lestari Sitinjak, Lilly Sitinjak, Sonia Togi Marito Sitinjak selaku kakak penulis dan Aber Tulus Sitinjak selaku adik penulis yang memberikan semangat dalam menyelesaikan skripsi ini.
6. Kristina Triany Mandalahi yang telah menemani dan meluangkan banyak waktu serta memberikan dorongan dan motivasi, terimakasih sudah menjadi tempat cerita dan telah menjadi bagian dari perjalanan penulis hingga menyelesaikan skripsi ini.
7. Teman-teman Filkom Keras Ito Desi Marinda Oktavia Sitinjak, Dek Paulina Siregar, Lae Arnold Siahaan dan ces Batara Nababan selaku teman yang selalu bisa menghibur dan membantu pengerjaan skripsi ini.
8. Andreas Pandapotan Sirait dan Daniel Sinaga selaku teman satu kosan yang selalu meberikan motivasi dan menjadi keluarga di Malang.

Penulis menyadari bahwa dalam penulisan ini masih terdapat banyak kekurangan maka dengan segala kerendahan hati, penulis mengharapkan kritik dan saran yang membangun agar penelitian ini dapat lebih bermanfaat dan lebih baik.

Malang, 14 Juli 2023

Penulis

johannes.7tin@gmail.com

ABSTRAK

Johannes Riski Sitinjak, Penerapan Metode Naïve Bayes dalam Sistem Pendeteksi Kualitas Tanah pada Tanaman Kedelai.

Pembimbing: Mochammad Hannats Hanafi Ichsan, S.ST., M.T dan Eko Setiawan, S.T., M.Eng., Ph.D

Kedelai berperan penting dalam meningkatkan gizi masyarakat karena kandungan proteinnya, selain baik bagi kesehatan bila dibandingkan dengan protein hewani, kedelai jauh relatif lebih murah. Namun pada tahun 2013 kedelai mengalami penurunan produksi karena adanya penurunan luas panen seluas 16,83 ribu hektar (2,97 persen) dan penurunan produktivitas sebesar 0,69 kuintal/hektar. Kualitas tanah memiliki peran yang krusial dalam kehidupan tanaman, dan tanah yang berkualitas tinggi dapat membantu meningkatkan produktivitas pertanian. Untuk mengatasi masalah yang timbul akibat penurunan luas panen dan produktivitas kedelai yang berdampak pada penurunan hasil produksi pangan kedelai, diperlukan suatu sistem yang dapat mengklasifikasikan kualitas tanah yang optimal untuk tanaman kedelai. Pada penelitian yang dilakukan Bhayangkara, A. (2020), membuat sebuah sistem pendeteksi kualitas tanah tanaman kedelai menggunakan sensor pH tanah dan Capacitive Soil Moist dengan metode klasifikasi K-Nearest Neighbor (KNN) yang mendapatkan tingkat akurasi 86,6%. Pada penelitian ini akan berfokus meningkatkan tingkat akurasi dari sistem dengan menggunakan metode klasifikasi Naive Bayes dan menggunakan sensor yang sama. Pengujian dilakukan sebanyak 30 kali yang menghasilkan tingkat akurasi klasifikasi Naive Bayes sebesar 90%. Rata-rata waktu komputasi yang dibutuhkan untuk menjalankan sistem sebesar 2,26 milisekon untuk proses training Naive Bayes dan 2,3 milisekon untuk proses klasifikasi Naive Bayes. Penggunaan SRAM sistem pada mikrokontroler Arduino Uno berada pada angka rata-rata sebesar 1226 bytes.

Kata kunci: Kedelai, Naive Bayes, Waktu Komputasi, *SRAM*.

ABSTRACT

Johannes Riski Sitinjak, Application of the Naïve Bayes Method in the Soil Quality Detection System in Soybean Plants.

Supervisors: Mochammad Hannats Hanafi Ichsan, S.ST., M.T and Eko Setiawan, S.T., M.Eng., Ph.D

Soybean plays an important role in increasing public nutrition because of its protein content, apart from being good for health when compared to animal protein, soy is relatively cheaper. However, in 2013 soybean production decreased due to a decrease in harvested area of 16.83 thousand hectares (2.97 percent) and a decrease in productivity of 0.69 quintals/hectare. Soil quality plays a crucial role in plant life, and high-quality soil can help increase agricultural productivity. To overcome the problems arising from the decrease in soybean harvested area and productivity which has an impact on decreasing soybean food production, we need a system that can classify optimal soil quality for soybean plants. In research conducted by Bhayangkara, A. (2020), created a system for detecting the quality of the soil for soybean plants using a soil pH sensor and Capacitive Soil Moisture by classification method K-Nearest Neighbor (KNN) which gets an accuracy rate of 86.6%. This research will focus on increasing the accuracy of the system using the Naive Bayes classification method and using the same sensor. The test was carried out 30 times which resulted in a Naive Bayes classification accuracy rate of 90%. The average computing time needed to run the system is 2.26 milliseconds for the process training Naive Bayes and 2.3 milliseconds for the Naive Bayes classification process. The use of system SRAM on the Arduino Uno microcontroller is at an average rate of 1226 bytes.

Keywords: Soybean, Naive Bayes, Computing Time, SRAM

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Penelitian	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka.....	6
2.2 Dasar Teori.....	7
2.2.1 Naïve Bayes	8
2.2.2 Waktu Komputasi.....	11
2.2.3 Penggunaan Memori.....	11
BAB 3 METODOLOGI	13
3.1 Tipe penelitian.....	13
3.2 Strategi dan Rancangan Penelitian.....	13
3.2.1 Metode Umum.....	13

3.2.2	Lokasi Penelitian.....	14
3.2.3	Teknik Pengumpulan Data	14
3.2.4	Teknik Analisis Data dan Pembahasan.....	15
3.2.5	Peralatan Pendukung yang Digunakan	15
BAB 4	REKAYASA KEBUTUHAN.....	16
4.1	Kajian Masalah	16
4.2	Identifikasi <i>Stakeholder</i>	17
4.3	Kebutuhan Fungsional.....	17
4.4	Spesifikasi Sistem.....	18
4.5	Analisis Kebutuhan Perangkat Keras dan perangkat Lunak	18
4.5.1	Kebutuhan Perangkat Keras.....	18
4.5.2	Kebutuhan Perangkat Lunak	22
BAB 5	PERANCANGAN DAN IMPLEMENTASI	23
5.1	Perancangan Sistem	23
5.1.1	Perancangan <i>Prototype</i> Sistem	23
5.1.2	Perancangan Perangkat Keras Sistem.....	26
5.1.3	Perancangan Perangkat Lunak Sistem	28
5.2	Implementasi Sistem	38
5.2.1	Implementasi <i>Prototype</i> Alat	38
5.2.2	Implementasi Perangkat Keras	39
5.2.3	Implementasi Perangkat Lunak.....	40
BAB 6	PENGUJIAN DAN ANALISIS	53
6.1	Pengujian Metode Naïve Bayes.....	53
6.1.1	Tujuan Pengujian.....	53
6.1.2	Prosedur Pengujian	53
6.1.3	Hasil dan Analisis.....	53
6.2	Pengujian Waktu Komputasi Metode Naïve Bayes.....	56
6.2.1	Tujuan Pengujian.....	56

6.2.2	Prosedur Pengujian	56
6.2.3	Hasil dan Analisis.....	56
6.3	Penggunaan SRAM	60
6.3.1	Tujuan Pengujian.....	60
6.3.2	Prosedur Pengujian	61
6.3.3	Hasil dan Analisis.....	61
BAB 7	PENUTUP	63
7.1	Kesimpulan	63
7.2	Saran.....	63
	DAFTAR PUSTAKA.....	64
	LAMPIRAN A - DATA SISTEM	66
	LAMPIRAN B - KODE PROGRAM UTAMA	69
	LAMPIRAN C - DOKUMENTASI	74

DAFTAR TABEL

Tabel 2. 1 Kajian Pustaka.....	6
Tabel 4. 1 Spesifikasi Laptop	18
Tabel 4. 2 Spesifikasi Arduini Uno	21
Tabel 5. 1 Keterangan Komponen.....	25
Tabel 5. 2 Koneksi Pin Sensor pH Tanah dengan Arduino Uno.....	27
Tabel 5. 3 Koneksi Pin Sensor Capacitive Soil Moist dengan Pin Arduino Uno	27
Tabel 5. 4 Koneksi Pin LCD 16x2 dengan Pin Arduino Uno	27
Tabel 5. 5 Data Latih.....	32
Tabel 5. 6 Nilai Standar Deviasi	33
Tabel 5. 7 Keterangan Implementasi Perangkat Keras	40
Tabel 5. 8 Implementasi Kode Program Utama	41
Tabel 5. 9 Implementasi Kode Program <i>Training</i> Naïve Bayes	46
Tabel 5. 10 Implementasi Kode Program Klasifikasi Naïve Bayes.....	48
Tabel 5. 11 Implementasi Program Waktu Komputasi dan Penggunaan Memori	50
Tabel 5. 12 Implementasi Kode Program Sensor pH Tanah	51
Tabel 5. 13 Implementasi Kode Program Sensor <i>Capacitive Soil Moist</i>	52
Tabel 6. 1 Pengujian Klasifikasi Metode Naïve Bayes	54
Tabel 6. 2 Pengujian Waktu Komputasi <i>Traning</i> Naïve Bayes	57
Tabel 6. 3 Pengujian Waktu Komputasi Klasifikasi Naïve Bayes	59

DAFTAR GAMBAR

Gambar 2. 1 Diagram Blok Naïve Bayes.....	9
Gambar 3. 1 Diagram Alir Metode Umum.....	14
Gambar 4. 1 Sensor pH Tanah.....	19
Gambar 4. 2 Capacitive Soil Moisture Sensor.....	20
Gambar 4. 3 Arduino Uno	20
Gambar 4. 4 Liquid Crystal Display	22
Gambar 4. 5 Arduino IDE.....	22
Gambar 5. 1 Diagram Blok Sistem.....	23
Gambar 5. 2 Desain <i>Prototype</i> Alat.....	25
Gambar 5. 3 Skematik Perangkat Keras.....	26
Gambar 5. 4 Perancangan Program Utama	29
Gambar 5. 5 Perancangan <i>Traning</i> Metode Naïve Bayes	30
Gambar 5. 6 Perancangan Kode Program Standar Deviasi.....	31
Gambar 5. 7 Perancangan Klasifikasi Metode Naïve Bayes.....	34
Gambar 5. 8 Perancangan Perhitungan Waktu Komputasi	35
Gambar 5. 9 Perancangan Perhitungan Penggunaan Memori	36
Gambar 5. 10 Perancangan Kode Program Sensor pH Tanah	37
Gambar 5. 11 Perancangan Kode Program Sensor <i>Capacitive Soil Moist</i>	38
Gambar 5. 12 Implementasi <i>Prototype</i> Alat	39
Gambar 5. 13 Implementasi Perangkat Keras	40
Gambar 6. 1 Grafik Waktu Komputasi <i>Traning</i> Naïve Bayes	58
Gambar 6. 2 Grafik Waktu Komputasi Klasifikasi Naïve Bayes	60
Gambar 6. 3 Grafik Pengujian Penggunaan SRAM.....	61

DAFTAR LAMPIRAN

LAMPIRAN A - DATA SISTEM	66
LAMPIRAN B - KODE PROGRAM UTAMA	69
LAMPIRAN C - DOKUMENTASI	74

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Tanaman Kedelai atau yang sering disebut dalam bahasa ilmiah dengan *Glycine max* adalah komoditas bahan pangan strategis ketiga setelah padi dan jagung, ini disebabkan karena hampir setiap hari kedelai dikonsumsi oleh sebagian besar masyarakat dengan rata-rata konsumsi kedelai mencapai 8,12 kg/kapita/tahun (Sudaryanto dan Swastika 2007). Karena kandungan protein kedelai, yang baik untuk kesehatan dan lebih murah daripada protein hewani, kedelai mampu membantu meningkatkan gizi masyarakat. (Damardjati et al, 2005).

Banyaknya kedelai yang diproduksi pada tahun 2013 adalah sebesar 779,99 ribu ton biji kering, yang mengalami penurunan sebesar 63,16 ribu ton (7,49 persen) dari jumlah kedelai yang diproduksi pada tahun 2012. Produksi kedelai mengalami penurunan disebabkan oleh adanya penurunan luas panen sebesar 2,97 persen (16,83 ribu hektar) dan adanya penurunan produktivitas sebesar 4,65 persen (0,69 kuintal/hektar)(Badan Pusat Statistik 2014)

Penurunan produksi akibat dari menurunnya luas lahan yang menjadi sebuah masalah penting yang perlu diperhatikan. Lahan memiliki peranan utama dalam kesuksesan pertanian, terutama dalam konteks Indonesia di mana lahan menjadi tempat tumbuh dan berkembangnya tanaman kedelai. Tanah memiliki peranan penting sebagai media tanam secara umum yang banyak digunakan dalam pertanian di Indonesia. Kualitas tanah sangat berperan penting dalam pertumbuhan tanaman, dan tanah yang berkualitas baik dapat membantu meningkatkan produksi tanaman. Selain itu, kualitas tanah yang baik juga dapat mengatasi penurunan hasil produksi yang disebabkan oleh adanya penurunan jumlah luas lahan. Pertumbuhan dan perkembangan tanaman sangat dipengaruhi oleh kualitas tanah, terutama untuk kedelai. Kedelai, jenis tanaman aerobik, sangat bergantung pada tingkat kelembapan tanah dan tingkat kandungan unsur hara. Selain itu, tanaman ini juga membutuhkan tingkat pH tanah yang cocok untuk pertumbuhannya. (Sumarno et al, 2013). Sebagai akibatnya, parameter-parameter seperti pH tanah dan kelembapan memainkan peran yang sangat penting untuk pertumbuhan kedelai.

Akibat masalah penurunan luas panen dan produktivitas kedelai yang berdampak pada produksi bahan pangan kedelai, penulis mengusulkan pengembangan sebuah sistem pendeteksi kualitas tanah. Sistem ini didasarkan pada penelitian sebelumnya yang dilakukan oleh Bhayangkara, A. (2020) dan menggunakan fitur-fitur yang sama, yaitu kelembapan dan pH tanah. Tujuan dari sistem ini adalah untuk membantu mengklasifikasikan kualitas tanah yang baik untuk pertumbuhan tanaman kedelai, sehingga dapat mendukung peningkatan hasil produksi kedelai secara efektif. Namun pada penelitian kali ini peneliti mengusulkan agar metode yang akan digunakan dari penelitian sebelumnya berbeda, yaitu dengan menggunakan metode Naïve Bayes yang dimana sebelumnya menggunakan metode K-NN (*K-Nearest Neighbors*). Metode Naive Bayes merupakan salah satu algoritma klasifikasi yang sering digunakan dalam berbagai bidang, termasuk pendeteksian kualitas tanah. Algoritma ini menggunakan probabilitas dan statistik untuk memprediksi kelas suatu objek. Penelitian ini juga didasari dari sebuah penelitian sebelumnya yang membandingkan kemampuan metode K-NN (*K-Nearest Neighbors*) dengan metode Naïve Bayes yang dimana hasil dari penelitian tersebut menunjukkan bahwa tingkat akurasi yang dihasilkan dari metode Naïve Bayes lebih tinggi dibandingkan tingkat akurasi yang dihasilkan dari metode K-NN. Tingkat akurasi yang dihasilkan dari metode Naïve Bayes yaitu sebesar 91% sedangkan K-NN menghasilkan tingkat akurasi sebesar 75% (Fayaz Itoo 2021).

Sistem yang akan dirancang memiliki tujuan agar meningkatkan tingkat akurasi dari penelitian sebelumnya oleh Bhayangkara, A. (2020), dengan tingkat akurasi yang dihasilkan sebesar 86,6%. Karena pada penelitian ini menggunakan fitur yang sama dengan penelitian terdahulu yaitu pH dan kelembapan tanah, maka pada penelitian ini tidak akan melakukan pengujian sensor yang digunakan karena pada penelitian sebelumnya sudah melakukan pengujian sensor dan mendapatkan hasil tingkat akurasi dari sensor pH sebesar 99,15% dan sensor *Capacitive Soil Moist* sebesar 97,31% (Bhayangkara, A. 2020). Selain untuk meningkatkan tingkat akurasi dari penelitian terdahulu, penelitian ini juga bertujuan untuk menguji efektivitas metode Naive Bayes dalam sistem pendeteksian kualitas tanah untuk tanaman kedelai dengan menguji waktu komputasi dan penggunaan memori sistem pada Arduino Uno. Implementasi pada sistem ini berupa nilai dari sensor pH tanah dan kelembapan tanah akan digunakan sebagai inputan ke Arduino Uno. Nantinya nilai yang didapatkan dari sensor pH dan kelembapan diklasifikasikan menggunakan metode Naïve Bayes dan nantinya di tampilkan pada LCD 16x2.

1.2 Rumusan Masalah

Berdasarkan informasi sebelumnya, beberapa rumusan masalah dapat disimpulkan sebagai berikut:

1. Bagaimana tingkat akurasi metode Naïve Bayes dalam sistem pendeteksi kualitas tanah tanaman kedelai?
2. Bagaimana hasil dari pengujian waktu komputasi pada metode Naïve Bayes?
3. Bagaimana hasil dari pengujian penggunaan memori pada metode Naïve Bayes?

1.3 Tujuan

Berdasarkan rumusan masalah, tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui tingkat akurasi metode Naïve Bayes pada sistem pendeteksi kualitas tanah tanaman kedelai.
2. Mengetahui hasil dari pengujian waktu komputasi pada metode Naïve Bayes.
3. Mengetahui hasil dari pengujian penggunaan memori pada metode Naïve Bayes.

1.4 Manfaat

Penelitian ini memiliki manfaat untuk membantu masyarakat khususnya para petani agar dapat mengetahui kualitas tanah sebelum menanam tanaman kedelai dan memastikan bahwa tanah yang akan digunakan adalah kualitas yang baik, supaya dapat meningkatkan produktivitas tanaman kedelai. Sistem pendeteksi kualitas tanah ini juga dapat mempermudah pemantauan kualitas tanah, sehingga dapat membantu petani dalam memastikan bahwa tanah memiliki kualitas yang baik.

1.5 Batasan Masalah

Tujuan dari menetapkan batasan masalah pada penelitian yang akan dilakukan adalah untuk menjaga fokus dan menghindari penyimpangan dari arah yang telah ditentukan oleh peneliti. Berikut adalah rangkuman dari batasan-batasan masalah yang ada pada penelitian ini:

1. Sensor yang digunakan adalah sensor pH dan kelembaban.
2. Sistem mengeluarkan output berupa hasil kualitas tanah pada LCD.

3. Tujuan dari alat ini dibuat hanya untuk pendeteksi kualitas tanah pada tanaman kedelai.
4. Fokus penelitian ini hanya pada pengukuran tiga kualitas tanah: baik, sedang, dan buruk.

1.6 Sistematika Penelitian

Berikut ini merupakan sistematika pembahasan yang digunakan dalam penelitian ini, masing-masing bab berikut diberikan penjelasan dan gambaran umum:

- **BAB 1 PENDAHULUAN**

Bab pendahuluan dalam penelitian ini mencakup elemen-elemen penting yang menjelaskan aspek-aspek fundamental dari penelitian tersebut. Penelitian berjudul "Penerapan Metode Naïve Bayes dalam Sistem Pendeteksi Kualitas Tanah pada Tanaman Kedelai dengan Arduino Uno".

- **BAB 2 LANDASAN KEPUSTAKAAN**

Landasan kepastakaan berisikan tinjauan pustaka oleh dasar teori yang relevan dengan penelitian ini. Kajian pustaka ini mencakup referensi dari buku, jurnal, dan sumber ilmiah lainnya yang memiliki keterkaitan dengan implementasi sistem pendeteksi kualitas tanah. Seluruh referensi yang dipilih akan mendukung dan relevan dengan batasan masalah yang telah ditetapkan dalam penelitian ini.

- **BAB 3 METODOLOGI**

Bab metodologi menjelaskan apa saja yang perlu dilakukan dalam implementasi sistem pendeteksi kualitas tanah menggunakan metode naïve bayes.

- **BAB 4 ANALISIS KEBUTUHAN**

Analisis kebutuhan akan dibahas kebutuhan dari sistem yang digunakan, termasuk kebutuhan fungsional dan non-fungsional, perangkat lunak, dan perangkat keras.

- **BAB 5 PERANCANGAN DAN IMPLEMENTASI**

Perancangan dan implementasi membahas bagian perancangan sistem yang terstruktur dalam penelitian ini, mulai dari *software* hingga *hardware* yang akan diperlukan. Bab ini juga memberikan penjelasan tentang penggunaan pseudocode dan prototype dalam penelitian..

- **BAB 6 PENGUJIAN DAN ANALISIS**

Bab pengujian dan analisis membahas pengujian dan hasilnya, dan menentukan apakah sistem telah memenuhi syarat.

- **BAB 7 PENUTUP**

Bab penutup menyampaikan hasil dari pengujian yang telah dilakukan dan menyarankan bagaimana penelitian selanjutnya dapat mencapai hasil terbaik.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Bagian ini akan dibahas mengenai penelitian sebelumnya yang relevan atau memiliki informasi yang berkaitan dengan penelitian yang ingin dilakukan. Tabel 2.1 menunjukkan penelitian yang digunakan untuk membandingkan dan mendukung penelitian ini. Tujuan dari kajian pustaka ini adalah untuk mengumpulkan informasi dari penelitian sebelumnya.

Tabel 2. 1 Kajian Pustaka

No	Nama Penulis (Tahun) Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Bhayangkara, A. (2020). “Sistem Pendeteksi Kualitas Tanah Tanaman Kedelai Menggunakan Metode K-Nearest Neighbor (K-NN) dengan Arduino Nano”	Menggunakan sensor pH dan sensor kelembapan tanah dan digunakan untuk tanaman kedelai.	Menggunakan metode KNN.	Menggunakan metode Naïve Bayes.
2	Anwar, Khairul (2018). “Sistem Pendeteksi Kandungan Nutrisi dalam Tanah Berdasarkan Warna dan Kelembapan dengan Menggunakan Metode Naive Bayes”	Menggunakan metode Naïve Bayes.	Mendeteksi nutrisi tanah dan menggunakan sensor warna	Mendeteksi kualitas tanah dan menggunakan sensor pH dan kelembapan.
3	Firdaus, Jeffry Atur(2023). “Analisis	Menguji waktu komputasi dan	Menganalisis Performa	Melakukan Implementasi

	Performa Algoritma Machine Learning Pada Perangkat Embedded Atmega328P”	penggunaan memori metode Naïve Bayes.	Algoritma dan tidak ada implementasi.	Pendeteksi Kualitas tanah.
--	---	---------------------------------------	---------------------------------------	----------------------------

Tujuan dari studi literatur ini adalah untuk mendapatkan informasi untuk mendukung penelitian tentang cara mengidentifikasi kualitas tanah pada tanaman kedelai. Pada Tabel 2.1 terdapat penelitian terdahulu yang dapat membantu peneliti sehingga dapat dilakukan dengan baik. Berdasarkan Bhayangkara, A.(2020) bahwa metode KNN dapat melakukan klasifikasi kualitas tanah dengan tingkat akurasi sebesar 86,6%. Pada penelitian tersebut juga melakukan pengujian sensor pH tanah dan kelembapan yang memiliki tingkat akurasi yang sudah sangat baik sehingga sangat membantu memberikan informasi sebagai penunjang dalam penelitian ini.

Kemudian pada penelitian Anwar, Khairul (2018) melakukan penelitian yang menggunakan metode yang sama pada penelitian yang akan dilakukan yaitu metode naïve bayes. Pada penelitian yang berjudul “Sistem Pendeteksi Kandungan Nutrisi dalam Tanah Berdasarkan Warna dan Kelembapan dengan Menggunakan Metode Naive Bayes” menjelaskan bagaimana sistem dapat mendeteksi kandungan nutrisi yang ada pada tanah. Kandungan nutrisi diidentifikasi berdasarkan kelembapan dan warna sehingga dari penelitian tersebut dapat menunjang penelitian ini dengan memberikan informasi bagaimana mengimplementasikan naïve bayes dengan baik.

Kemudian pada penelitian Firdaus, Jeffry Atur (2023) dengan judul “Analisis Performa Algoritma Machine Learning Pada Perangkat Embedded Atmega328P” menjelaskan bagaimana hasil analisis metode naïve bayes dalam waktu komputasi dan penggunaan memori. Penelitian ini juga akan dilakukan analisis waktu komputasi dan penggunaan memori algoritma naïve bayes pada Arduino uno.

2.2 Dasar Teori

Bagian ini berisikan penjelasan dari teori-teori yang berhubungan dengan implementasi sistem pendeteksi kualitas tanah pada tanaman kedelai menggunakan metode Naïve Bayes menggunakan Arduino uno.

2.2.1 Naïve Bayes

Metode pengklasifikasi Naive Bayes sangat efektif dan sederhana. Metode Naive Bayes adalah metode probabilitas dan statistik yang dikembangkan oleh ilmuwan Inggris Thomas Bayes, yang bertujuan untuk memprediksi kemungkinan di masa depan berdasarkan pengalaman masa lalu. Metode ini terdiri dari dua tahap dalam proses pengklasifikasian: tahap pelatihan dan tahap klasifikasi. Hasil uji menunjukkan Metode Naive Bayes menghasilkan akurasi yang lebih baik hanya dengan memperhitungkan pemilihan fitur daripada frekuensi fitur.. (Negara et al., 2020).

Rumus Naïve Bayes ditunjukkan pada persamaan (2.1) berikut:

$$P(A|B) = P(B|A)P(A)P(B) \quad (2.1)$$

Keterangan:

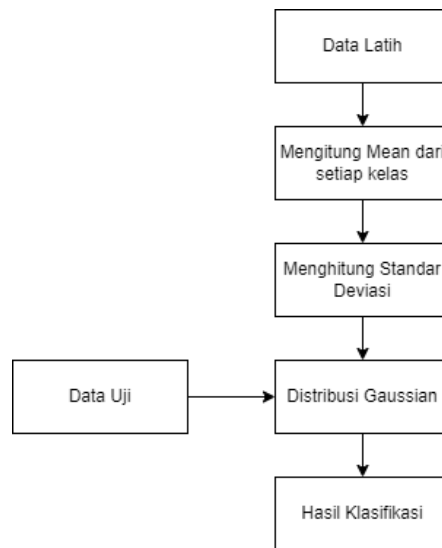
$P(A|B)$ = Probabilitas A terjadi dengan bukti bahwa B telah terjadi (probabilitas superior)

$P(B|A)$ = Probabilitas B terjadi dengan bukti bahwa A telah terjadi

$P(A)$ = Peluang A

$P(B)$ = Peluang B

Penjelasan metode naïve bayes juga akan dijelaskan dengan diagram blok pada **Gambar 2.1**.



Gambar 2. 1 Diagram Blok Naïve Bayes

Pada blok diagram metode naïve bayes diatas dapat dilihat beberap poin penting yang perlu diperhatikan yaitu data latih, mean, standar deviasi, data uji dan rumus naïve bayes. Untuk data latih didapatkan dengan pengumpulan data secara sekunder dan untuk data uji didapatkan dari pengumpulan data yang dilakukan secara premier.

2.2.1.1 Mean

Rata-rata, atau yang biasa disebut mean, dari suatu kumpulan nilai adalah bilangan yang mewakili data dengan menjumlahkan nilai-nilai tersebut dan membaginya dengan jumlah data. Rata-rata memberikan gambaran keseluruhan atau representasi dari seluruh set data tersebut. Cara untuk mendapatkan *Mean* dari sebiah kelompok data menggunakan rumus:

$$\bar{x} = \frac{x_i + x_{i+1} + x_{i+2} + \dots + x_{n-1} + x_n}{n} \quad (2.2)$$

Keterangan:

$$\bar{x} = \text{Mean}$$

$$x_i = \text{Data Ke } i$$

$$n = \text{Jumlah Data}$$

2.2.1.2 Standar Deviasi

Standar deviasi merupakan ukuran statistik yang dipakai untuk mengukur sejauh mana data tersebar atau menyebar dari rata-rata. Standar deviasi menghitung seberapa jauh setiap nilai dalam himpunan data dari rata-rata, dan memberikan gambaran tentang keragaman data. Cara untuk mendapatkan nilai dari standar deviasi dapat menggunakan dengan persamaan 2.3 dibawah:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (2.3)$$

Keterangan:

s = Standar Deviasi

\bar{x} = Mean

x_i = Data Ke - i

n = Jumlah Data

2.2.1.3 Distribusi Gaussian

Distribusi Gaussian, juga dikenal sebagai distribusi normal, adalah salah satu jenis distribusi probabilitas yang paling umum digunakan dalam statistika. Distribusi ini memiliki bentuk kurva lonceng yang simetris dan sering digunakan untuk menganalisis data kontinu di berbagai bidang ilmu. Distribusi Gaussian memiliki beberapa karakteristik utama. Pertama, distribusi ini simetris di sekitar nilai rata-rata (mean) dengan puncak kurva berada pada nilai tersebut. Ini berarti bahwa rata-rata, median, dan modus dari distribusi ini memiliki nilai yang sama. Kedua, sebaran data dalam distribusi Gaussian dikendalikan oleh standar deviasi (standard deviation). Semakin besar standar deviasi, semakin besar sebaran data di sekitar rata-rata. Cara untuk mendapatkan nilai dari distribusi gaussian dapat menggunakan dengan persamaan 2.4 dibawah:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-(x-\mu)^2}{2\sigma^2} \quad (2.4)$$

Keterangan:

g = Distribusi Gaussian.

x = Nilai input.

μ = Rata – rata (mean).

σ = Standar deviasi.

exp = Bilangan konstan 2.71828.

π = Bilangan pi.

2.2.2 Waktu Komputasi.

Waktu komputasi mengacu pada jumlah waktu yang diperlukan oleh sebuah program komputer atau algoritma untuk menyelesaikan tugas tertentu. Ini dapat mencakup waktu yang diperlukan untuk menjalankan instruksi-instruksi komputer, melakukan operasi matematika, membaca atau menulis data, atau menjalankan proses yang kompleks. Waktu komputasi sering diukur dalam satuan waktu seperti detik, milidetik, mikrodetik, atau langkah-langkah komputasi (misalnya, jumlah operasi yang dilakukan). Penting untuk memahami dan mengukur waktu komputasi karena dapat membantu dalam mengoptimalkan kinerja program, membandingkan efisiensi algoritma yang berbeda, atau memprediksi seberapa cepat suatu tugas dapat diselesaikan dalam konteks komputasi.

Ada beberapa metode untuk mengukur waktu komputasi, termasuk penggunaan fungsi waktu pada bahasa pemrograman, penggunaan instrumen profil, atau analisis teoritis. Dalam beberapa kasus, waktu komputasi juga dapat diukur menggunakan notasi O (Big O) untuk menganalisis tingkat pertumbuhan waktu komputasi algoritma saat ukuran masukan meningkat. Secara keseluruhan, waktu komputasi adalah ukuran penting dalam komputasi yang melibatkan pemahaman, pengukuran, dan optimasi waktu yang dibutuhkan oleh program atau algoritma untuk menyelesaikan tugas tertentu.

2.2.3 Penggunaan Memori.

Penggunaan memori mengacu pada jumlah ruang atau kapasitas memori yang digunakan oleh sebuah program komputer atau algoritma dalam menjalankan tugasnya. Memori komputer digunakan untuk menyimpan instruksi program, data, variabel, dan struktur data yang diperlukan dalam proses komputasi.

Penggunaan memori adalah faktor penting dalam pengembangan perangkat lunak karena memori komputer terbatas. Semakin besar penggunaan memori oleh program, semakin banyak memori yang diperlukan, dan ini dapat mempengaruhi kinerja dan efisiensi program. Penggunaan memori yang berlebihan dapat

menyebabkan program berjalan lambat atau bahkan menyebabkan kegagalan program jika memori yang tersedia tidak cukup.

BAB 3 METODOLOGI

3.1 Tipe penelitian

Tipe penelitian dalam studi ini adalah penelitian implementasi pengembangan yang lebih lanjut. Dalam penelitian sebelumnya yang dilakukan Bhayangkara, A. (2020) memiliki tingkat akurasi klasifikasi yang masih dapat ditingkatkan. Pada penelitian ini yang dimaksud implementatif pengembangan lanjut tersebut yaitu berupa sistem pendeteksi kualitas tanah pada tanaman kedelai dengan menggunakan sensor pH dan kelembaban untuk menentukan apakah tanah yang diuji masuk dalam kategori baik, sedang atau buruk. Dalam jenis penelitian ini, pengembangan lanjut mengacu pada menciptakan ide baru dalam penelitian yang sudah ada sebelumnya. Hal tersebut bertujuan untuk menghasilkan metode yang baru dan lebih baik, yang dapat digunakan sebagai pembanding saat melakukan pengujian. Dalam penelitian ini, pengembangan dilakukan dengan menggunakan metode naïve bayes untuk meningkatkan akurasi klasifikasi.

Penelitian ini mengadopsi pendekatan implementatif, yang bertujuan untuk memberikan solusi dalam bentuk implementasi produk, baik itu dalam bentuk perangkat lunak maupun perangkat keras, terhadap permasalahan yang dikaji dalam penelitian. Pada penelitian ini, peneliti melakukan serangkaian kegiatan untuk merancang sistem yang siap digunakan, yang meliputi analisis, perancangan perangkat keras, dan pengujian langsung terhadap sistem tersebut. Fokus penelitian adalah pengembangan sebuah sistem yang terdiri dari perangkat keras yang mampu mengimplementasikan pendeteksi kualitas tanah pada tanaman kedelai..

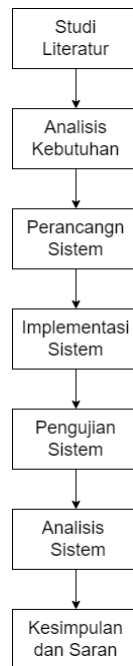
3.2 Strategi dan Rancangan Penelitian

Dalam bagian ini adalah subbagian yang membahas metode umum, subjek atau subjek penelitian, lokasi penelitian, metode pengumpulan data, metode analisis data, diskusi, dan peralatan pendukung yang digunakan dalam penelitian ini.

3.2.1 Metode Umum

Metode penelitian yang diharapkan untuk membangun sistem yang sistematis dan runtut. Hal ini dibuat untuk memastikan bahwa penelitian ini berjalan lancar dan sesuai dengan harapan. Untuk memulai, peneliti mencari literatur untuk mendapatkan referensi. Selanjutnya, peneliti mengumpulkan dan menganalisis teori

yang terkait dengan masalah penelitian saat ini. Selanjutnya, peneliti mengevaluasi *software* dan *hardware* yang dibutuhkan untuk menjalankan penelitian dan mencapai hasil yang diharapkan. Setelah itu, peneliti merancang dan menerapkan sistem untuk memenuhi rumusan masalah dan tujuan penelitian. Selanjutnya, untuk pengujian dan memeriksa kesesuaian sistem. Langkah - langkah yang dilakuakn pada penelitian ini dapat dilihat pada diagram alir Gambar 3.1.



Gambar 3. 1 Diagram Alir Metode Umum

3.2.2 Lokasi Penelitian

Pengambilan data, pengujian data, dan pengerjaan sistem dilakukan di lokasi penelitian, yaitu di kota Malang, Jawa Timur.

3.2.3 Teknik Pengumpulan Data

Alat yang dirancang pada penelitian dapat digunakan secara langsung untuk menerapkan teknik pengumpulan data ini. Teknik pengumpulan data yang digunakan oleh penulis dalam penelitian ini adalah pengumpulan data primer dan sekunder yaitu untuk data latih yang digunakan diambil dari penelitian sebelumnya yang dilakukan oleh Bhayangkara, A. (2020) sedangkan untuk data uji diambil langsung oleh peneliti dengan menggunakan sampel tanah yang memiliki nilai pH tanah dan kelembapan

yang bervariasi. Data yang telah diperoleh dari sampel tanah yang dikumpulkan berupa nilai Kelembapan tanah dan nilai pH tanah dan kemudian diklasifikasikan menggunakan Naive Bayes yang sudah deprogram pada sistem.

3.2.4 Teknik Analisis Data dan Pembahasan

Bagian ini merupakan tahap yang dilakukan untuk mengetahui waktu komputasi dan penggunaan memori yang direncanakan untuk menghasilkan output, serta tingkat akurasi dari metode pengklasifikasian yang digunakan. Teknik analisis data dan pembahasan ini akan dapat memastikan bahwa waktu komputasi dan penggunaan memori yang direncanakan untuk menghasilkan output sesuai dengan harapan dari dibuatnya rumusan masalah.

3.2.5 Peralatan Pendukung yang Digunakan

Peralatan pendukung yang dibutuhkan agar dapat memenuhi kebutuhan dari sistem yang ingin dibuat dan relasi setiap komponen supaya sistem bisa dijalankan sesuai dengan apa yang diinginkan, kemudian dapat mendukung perancangan dan pengimplementasian sistem pada penelitian ini.

3.2.5.1 Perangkat Lunak

1. Arduino IDE

3.2.5.2 Perangkat Keras

1. Arduino Uno
2. Sensor pH Tanah
3. Sensor Soil Moisture
4. Liquid Crystal Display (LCD)

BAB 4 REKAYASA KEBUTUHAN

Dalam bagian ini, diuraikan beberapa teknik pengembangan yang dibutuhkan pada penelitian ini, yaitu kebutuhan fungsional, *hardware* dan *software*, analisis masalah, spesifikasi sistem, dan pengidentifikasian pihak-pihak terkait (stakeholder).

4.1 Kajian Masalah

Berdasarkan beberapa sumber literatur yang sudah dipaparkan, karena kandungan protein kedelai, yang baik untuk kesehatan dan lebih murah daripada protein hewani, kedelai sangat penting untuk meningkatkan gizi masyarakat. Namun kedelai mengalami penurunan produksi karena adanya luas panen yang menurun sebesar 2,97% (16,83 ribu hektar) dan produktivitas yang menurun sebesar 4,65% (0,69 kuintal per hektar).

Produksi tanaman yang menurun diakibatkan oleh penyusutan luas lahan adalah isu yang penting dan memerlukan perhatian serius. Kualitas tanah memiliki peran yang krusial dalam kehidupan tanaman, dan tanah yang berkualitas tinggi dapat membantu meningkatkan produktivitas pertanian. Selain itu, kualitas tanah yang baik juga dapat mengatasi penurunan hasil panen yang terkait dengan penyusutan luas lahan. Tanah dengan kualitas yang optimal dapat berdampak signifikan pada pertumbuhan dan perkembangan tanaman, terutama dalam kasus tanaman kedelai. Kedelai adalah tanaman aerobik yang sangat sensitif terhadap kelembapan dan keseimbangan nutrisi tanah, serta membutuhkan tingkat pH tanah yang sesuai.

Untuk mengatasi masalah yang timbul akibat penurunan luas panen dan produktivitas kedelai yang berdampak pada penurunan hasil produksi pangan kedelai, diperlukan suatu sistem yang bisa mengklasifikasikan tanah optimal untuk tanaman kedelai. Sistem ini akan berperan dalam mendukung pertumbuhan kedelai dengan memberikan informasi tentang kualitas tanah yang memadai.

Pada penelitian terdahulu yang telah dilakukan oleh Bhayangkara, A. (2020), membuat sistem pendeteksi kualitas tanah tanaman kedelai. Penelitian ini menggunakan sensor pH tanah dan *Capacitive Soil Moisture* dengan metode K-NN (*K-Nearest Neighbors*). Penelitian tersebut menghasilkan tingkat akurasi klasifikasi sebesar 86,6%, sehingga pada penelitian yang akan dilakukan bertujuan untuk meningkatkan tingkat akurasi klasifikasi menggunakan metode yang berbeda yaitu metode Naive Bayes.

4.2 Identifikasi *Stakeholder*

Terdapat beberapa *stakeholder* yang terkait pada sistem yang akan dibuat, yaitu sebagai berikut:

1. Petani yang terlibat langsung dalam pertanian kedelai dan memiliki kepentingan langsung terhadap kualitas tanah yang mendukung pertumbuhan dan hasil panen kedelai.
2. Peneliti pertanian yang dapat memberikan wawasan dan pemahaman tentang aspek ilmiah dan teknis terkait peningkatan kualitas tanah serta pemilihan metode pendeteksian yang tepat.
3. Pemerintah daerah yang bertanggung jawab dalam mengatur kebijakan pertanian dan berperan dalam menyediakan dukungan, regulasi, dan sumber daya untuk meningkatkan produktivitas tanaman kedelai.

4.3 Kebutuhan Fungsional

Sebuah sistem harus memenuhi persyaratan atau persyaratan tertentu agar dapat melakukan fungsi-fungsi tertentu, yang dikenal sebagai kebutuhan fungsional. Kebutuhan fungsional mendefinisikan apa yang harus dilakukan oleh sistem, bagaimana sistem harus berperilaku, dan apa yang harus dihasilkan oleh sistem.

1. Sistem mampu mengukur kadar pH dan kelembaban dalam tanah.
Menggunakan sensor pH tanah dan *capacitive Soil Moist* untuk mengumpulkan data pH dan kelembapan tanah. Kedua sensor ini bersifat kapasitif dan memiliki kemampuan untuk mendeteksi perubahan pH dan kadar kelembapan tanah.
2. Sistem dapat menentukan kualitas tanah dengan algoritma Naïve Bayes.
Algoritma Naïve Bayes diharapkan mampu melakukan klasifikasi kualitas tanah yang telah di deteksi oleh alat. Metode klasifikasi Naïve Bayes merupakan metode *machine learning* yang dimana probabilitas dan teori Bayes digunakan untuk menghasilkan model prediksi yang dapat mengklasifikasikan data baru ke dalam kategori yang tepat.
3. Sistem mampu menampilkan hasil klasifikasi kualitas tanah yang sudah dideteksi.
Setelah mendapatkan hasil klasifikasi kualitas tanah menggunakan algoritma Naïve Bayes, sistem akan menampilkannya pada *display LCD 16x2*. Pada *display LCD 16x2* nantinya akan menunjukkan hasil klasifikasi kualitas tanah yaitu baik, sedang dan buruk.

4.4 Spesifikasi Sistem

Spesifikasi sistem merincikan persyaratan yang bukan bersifat fungsional dari sistem guna memastikan bahwa persyaratan fungsional utama terpenuhi dengan baik. Oleh karena itu, terdapat beberapa kebutuhan non-fungsional yang harus dipenuhi.

1. Sistem harus memiliki tingkat keandalan tinggi dalam mengambil sampel dan mengukur parameter kualitas tanah agar hasil yang diperoleh akurat dan konsisten.
2. Sistem harus memiliki waktu respons yang cepat dan *real-time* dalam mengukur dan menganalisis kualitas tanah untuk memastikan efisiensi dan ketepatan waktu dalam memberikan informasi kepada petani.

4.5 Analisis Kebutuhan Perangkat Keras dan perangkat Lunak

Analisis kebutuhan dilakukan agar dapat mengetahui kebutuhan perangkat lunak dan perangkat keras sistem untuk mencapai tujuan.

4.5.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras sistem mencakup komponen yang digunakan untuk mendukung kinerja sistem:

1. Laptop

Laptop yang digunakan untuk membuat sistem adalah *Lenovo IdePad Gaming 3*. Laptop ini berfungsi untuk tempat pembuatan *sourcecode* pada *software Arduino Ide*. Pada laptop juga akan ditampilkan waktu komputasi dan penggunaan memori. Berikut pada **Tabel 4.1** adalah spesifikasi laptop yang digunakan.

Tabel 4. 1 Spesifikasi Laptop

No	Spesifikasi	Keterangan
1	Merk	<i>Lenovo IdePad Gaming 3</i>
2	CPU	AMD Ryzen 5 5600H
3	<i>Memory</i>	8GB SO-DIMM DDR4 RAM
4	Penyimpanan	SSD 512GB
5	Sistem Operasi	Windows 11

2. Sensor pH Tanah

Sensor pH tanah merupakan perangkat yang digunakan untuk mengukur pH dari tanah. pH tanah sangat penting untuk menentukan kondisi tanah dan mempengaruhi pertumbuhan tanaman. Tanah yang memiliki pH yang baik memiliki kandungan unsur hara dan air yang dibutuhkan tanaman agar berkembang dan berproduksi dengan baik.



Gambar 4. 1 Sensor pH Tanah

Sumber: (Indobot Academy, 2023)

Sensor pH tanah serupa dengan sensor pH, tetapi dirancang khusus untuk pengukuran pH tanah. Sensor ini mengandung elektroda yang dapat memantau variasi potensial listrik yang terjadi pada permukaan elektroda, dan mengubah informasi tersebut menjadi nilai pH tanah. Nilai pH yang diukur oleh sensor dapat digunakan oleh petani untuk memantau kondisi tanah.

3. *Capacitive Soil Moisture Sensor*

Capacitive Soil Moisture Sensor adalah sebuah sensor kelembaban tanah yang dipakai untuk mengukur tingkat kelembaban dalam tanah. Ini sangat berguna bagi mereka yang tertarik pada bidang pertanian, hortikultur, atau pemeliharaan taman, karena membantu memantau tingkat air dalam tanah dan memastikan tanaman mendapatkan jumlah air yang tepat.



Gambar 4. 2 Capacitive Soil Moisture Sensor

Sumber: (Taryana Suryana, 2021)

Kelebihan dari sensor kelembaban tanah kapasitif adalah kemampuannya untuk mengukur kelembaban tanah secara langsung, tidak dipengaruhi oleh faktor eksternal seperti suhu atau tekanan.

4. Arduino Uno

Arduino Uno merupakan mikrokontroler berbasis sistem terbuka yang digunakan untuk mengontrol proyek elektronik dan membangun sistem Internet of Things (IoT). Mikrokontroler ini didesain untuk membuat pengembangan proyek elektronik lebih mudah dan aksesibel bagi pemula dan profesional.



Gambar 4. 3 Arduino Uno

Sumber: (Elga Aris Prastyo, 2022)

Arduino Uno menggunakan mikrokontroler ATmega328P dan memiliki port input/output yang akan digunakan untuk mengendalikan perangkat elektronik lain, seperti sensor, actuator, display, dan lain-lain. Arduino Uno juga memiliki

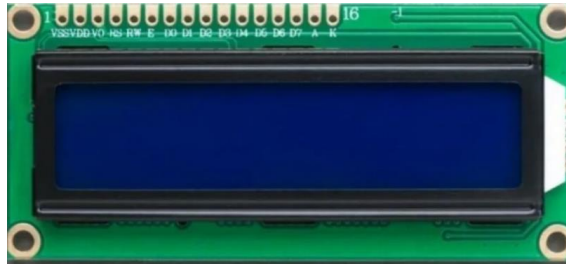
antarmuka pemrograman yang mudah digunakan, memungkinkan pengguna untuk menulis program menggunakan bahasa pemrograman C dan menguploadnya ke mikrokontroler. Berikut pada **Tabel 4.2** merupakan spesifikasi dari Arduino Uno.

Tabel 4. 2 Spesifikasi Arduini Uno

Parameter	Rating
Mikrokontroler	ATMega32P
Tegangan Operasional	5 V
Tegangan Masukan	7 – 12 Vdc
Jumlah Digital I/O	14 Pin
Jumlah analog Input	6 Pin
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

5. Liquid Crystal Display (LCD)

Liquid Crystal Display (LCD) adalah jenis display elektronik yang digunakan dalam berbagai perangkat elektronik seperti televisi, komputer, ponsel, dan lain-lain. LCD menggunakan teknologi cairan kristal untuk memproduksi gambar. LCD terdiri dari beberapa lapisan cairan kristal yang ditempatkan antara dua lapisan transparan bermuatan. Saat arus listrik dialirkan melalui cairan kristal, molekul-molekul cairan berubah posisi dan memungkinkan atau menghambat cahaya yang melewati lapisan transparan.



Gambar 4. 4 Liquid Crystal Display

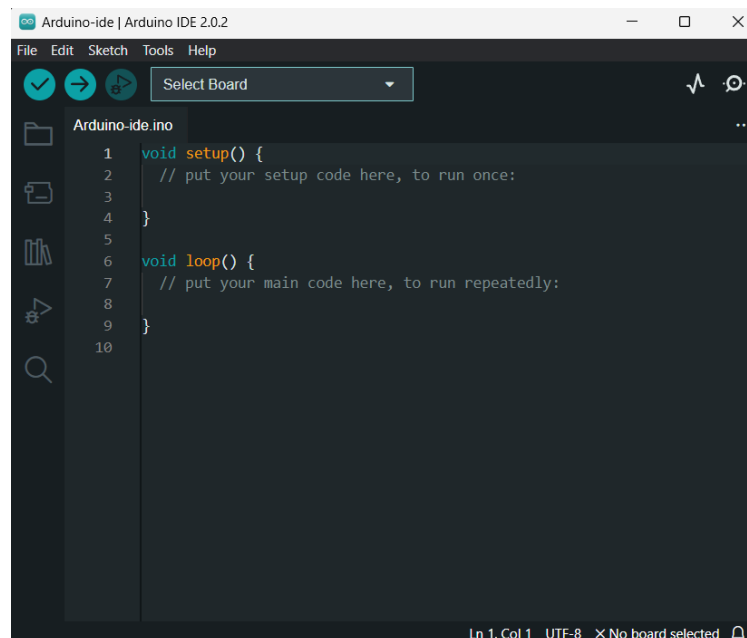
Sumber: (Indobot Academy, 2023)

4.5.2 Kebutuhan Perangkat Lunak

Bertujuan agar sistem dapat dikendalikan sehingga nantinya mampu memproses kesesuaian data dengan yang diinginkan.

1. Arduino IDE

Pada penelitian ini Arduino IDE berfungsi sebagai tempat pembuatan kode program sistem pendeteksi kualitas tanah tanaman kedelai. Kemudian kode program akan dimasukkan kedalam mikrokontroler Arduino Uno. Pada Arduino IDE juga akan menampilkan waktu komputasi dan penggunaan memori sistem pada serial monitor.

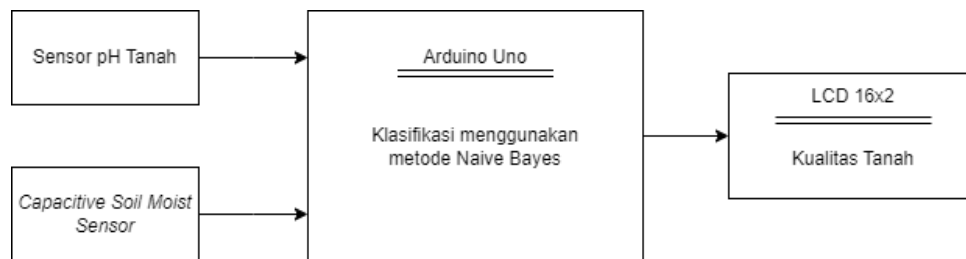


Gambar 4. 5 Arduino IDE

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

Pada bab perancangan sistem akan membahas dimana proses perancangan sistem pendeteksi kualitas tanah tanaman kedelai pada lahan tanaman dengan memanfaatkan metode klasifikasi Naïve Bayes untuk mengklasifikasikan kualitas tanah tanaman kedelai berdasarkan fitur pH tanah dan kelembapan tanah. Perancangan sistem ini digunakan untuk memastikan bahwa sistem dapat digunakan dan dapat membantu mendeteksi kualitas tanah tanaman kedelai yang dideteksi oleh sistem. Diagram blok dapat dilihat pada **Gambar 5.1**.

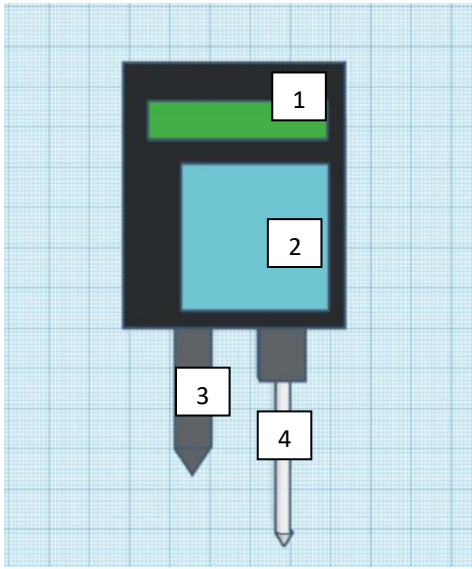


Gambar 5. 1 Diagram Blok Sistem

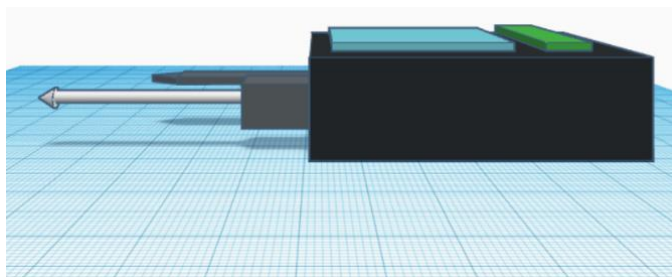
Sistem ini merupakan sebuah alat yang dapat mengambil data dari tanah. Sistem ini dibuat menggunakan dua buah sensor, yaitu sensor kelembapan tanah yang menggunakan *Capacitive Soil Moist Sensor* dan sensor pH tanah. Untuk memproses data yang didapatkan dari kedua sensor, sistem ini menggunakan Arduino Uno sebagai unit pemrosesnya. Semua komponen yang sudah ada nantinya akan dihubungkan pada Arduino Uno dan akan dirangkai sesuai dengan kebutuhan dari sistem yang diinginkan. Nilai yang dihasilkan dari pembacaan sensor *Capacitive Soil Moist Sensor* dan sensor pH tanah akan diproses menggunakan algoritma klasifikasi Naïve Bayes. Setelah mendapatkan hasil dari perhitungan algoritma Naïve Bayes, sistem akan menampilkan hasil tersebut pada *display LCD 16x2*

5.1.1 Perancangan *Prototype* Sistem

Dalam perancangan desain sistem dibuat pada platform website yaitu *tinkercad.com*. Desain dari *prototype* alat adalah sebuah kotak yang terbuat dari bahan *plastic* yang memiliki ukuran panjang 15 cm, lebar 10 cm dan tinggi 5 cm. Hasil dari desain *prototype* alat dapat dilihat pada Gambar 5.1 dan akan dijelaskan pada Tabel 5.1.



a. Desain Tampak Atas



b. Desain Tampak Samping

Gambar 5. 2 Desain *Prototype* Alat

Tabel 5. 1 Keterangan Komponen

No	Keterangan
1	LCD 16x2
2	Arduino Uno
3	Sensor <i>Capacitive soil moist</i>
4	Sensor pH Tanah

Pada Gambar 5.1 adalah hasil desain perancangan *prototype* sistem tampak atas dan tampak samping dari bagian komponen-komponen yang ada didalamnya. Pada bagian angka satu merupakan LCD 16x2 yang akan menampilkan hasil klasifikasi kualitas tanah dari perhitungan sistem pendeteksi kualitas tanah tanaman kedelai. LCD 16x2 ini diposisikan pada bagian depan alat dan menghadap keluar. Sehingga LCD 16x2 dapat terlihat dengan jelas tanpa harus membuka bagian dalam alat.

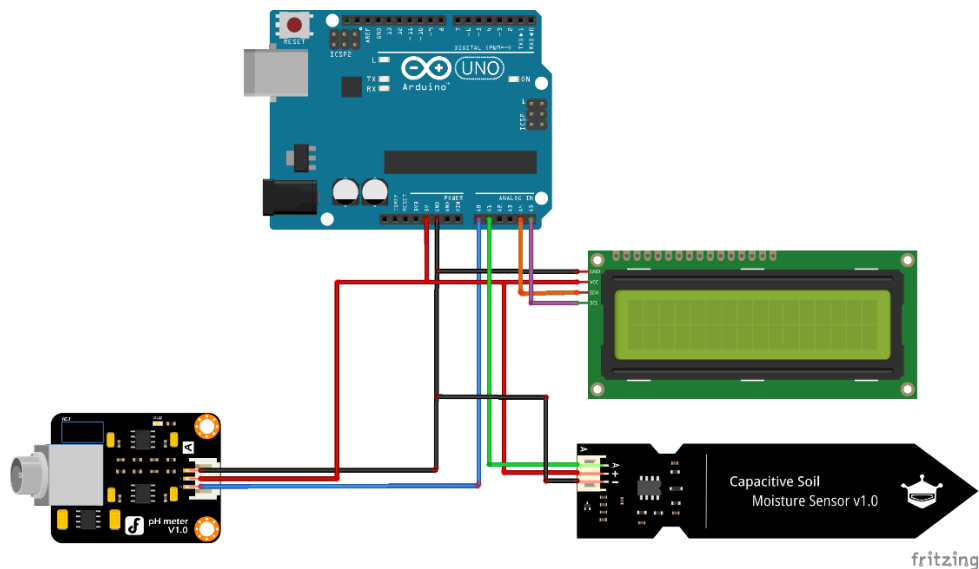
Arduino Uno pada Gambar 5.1 diidentifikasi dengan angka dua yang berada pada bagian dalam alat. Alasan Arduino Uno ditempatkan pada bagian dalam alat dikarenakan Arduino Uno nantinya hanya berfungsi sebagai otak atau yang memproses perhitungan sistem pendeteksi kualitas tanah tanaman kedelai dengan algoritma Naïve Bayes.

Pada Gambar 5.1 juga terdapat dua sensor yang ditunjukkan oleh angka tiga dan empat. Sensor pertama adalah sensor pH tanah, sedangkan sensor kedua adalah

sensor kelembapan tanah menggunakan metode *capacitive soil moist*. Penempatan kedua sensor tersebut sama dengan penempatan LCD 16x2, yaitu mereka diletakkan secara sejajar di luar alat. Mereka memiliki posisi di bagian bawah alat dan akan ditancapkan ke dalam tanah tempat tanaman kedelai tumbuh. Kedua sensor ini akan mengambil data dari tanah dan kemudian data tersebut akan diproses oleh papan pemrosesan Arduino Uno.

5.1.2 Perancangan Perangkat Keras Sistem

Pada bab ini akan dilakukan perancangan perangkat keras agar sistem dapat berjalan sesuai dengan yang diharapkan. Pada perancangan perangkat keras ini akan dijelaskan hubungan antar komponen yang digunakan dan pin mana saja yang akan dihubungkan pada setiap komponennya. Komponen-komponen yang digunakan yaitu Arduino Uno, sensor pH tanah, sensor capacitive soil moist dan LCD 16x2 seperti yang ditunjukkan pada perancangan perangkat keras pendeteksi kualitas tanah tanaman kedelai pada Gambar 5.1.



Gambar 5. 3 Skematik Perangkat Keras

Sensor pH tanah yang digunakan untuk pengambilan data pH tanah akan dihubungkan dengan arduino uno menggunakan kabel jumper. Data yang telah diambil akan diproses oleh sistem dan ditampilkan pada LCD 16x2 yang telah terpasang pada arduino uno. Keterangan pin sensor pH tanah dan arduino uno ditunjukkan pada tabel 5.1.

Tabel 5. 2 Koneksi Pin Sensor pH Tanah dengan Arduino Uno

Sensor pH Tanah	Arduino Uno
GND	GND
Data (Output)	Pin A0

Kemudian terdapat sensor capacitive soil moist yang terhubung pada pin arduino uno. Sensor ini berfungsi untuk pengambilan data kelembapan tanah pada tanah tanaman kedelai yang akan dideteksi dan ditampilkan pada LCD 16x2 yang telah terpasang pada arduino uno. Keterangan koneksi pin sensor capacitive soil moist dengan pin arduino uno ditunjukkan pada Tabel 5.2.

Tabel 5. 3 Koneksi Pin Sensor Capacitive Soil Moist dengan Pin Arduino Uno

Capacitive Soil Moist	Arduino Uno
GND	GND
VCC	VCC
Data (Output)	Pin A1

Data yang didapatkan oleh sensor ph tanah dan sensor capacitive soil moist akan ditampilkan pada sebuah LCD 16x2. LCD 16x2 yang dihubungkan dengan arduino uno akan dihubungkan pada modul I2C kemudian akan dihubungkan pada arduino uno. Pin LCD 16x2 dan arduino uno ditunjukkan pada Tabel 5.3.

Tabel 5. 4 Koneksi Pin LCD 16x2 dengan Pin Arduino Uno

LCD 16x2	Arduino Uno
GND	GND
VCC	VCC
SDA	Pin A4

SCL	Pin A5
-----	--------

5.1.3 Perancangan Perangkat Lunak Sistem

Pada bagian perancangan perangkat lunak, akan dijelaskan tentang pembuatan program yang digunakan untuk sistem pendeteksi kualitas tanah tanaman kedelai. Bagian perancangan perangkat lunak ini akan menjelaskan tentang rancangan program untuk mengambil data dari sensor, waktu komputasi, penggunaan memori dan melakukan klasifikasi kualitas tanah menggunakan metode Naïve Bayes.

5.1.3.1 Perancangan Program Utama

Pada perancangan program utama akan dibahas bagaimana keseluruhan sistem berjalan dari awal hingga akhir. Perancangan program utama sistem pada penelitian ini ditunjukan pada Gambar 5.3.

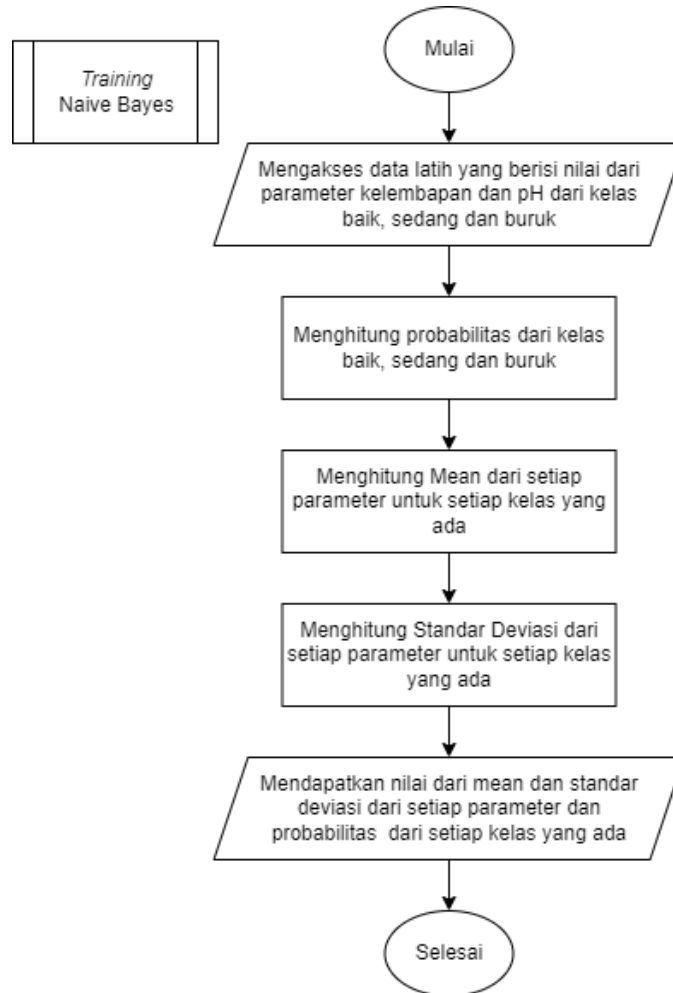


Gambar 5. 4 Perancangan Program Utama

Perancangan program utama dimulai dengan menginisialisasi mode pin, variabel dan penggunaan *library* tambahan kemudian dilanjutkan pada tahap *training* Naïve Bayes. Setelah tahap *training*, akan dilanjutkan dengan pembacaan nilai kedua sensor yang akan digunakan sebagai inputan. Setelah itu adalah klasifikasi dari algoritma Naïve Bayes dan melakukan perhitungan waktu komputasi serta penggunaan memori. Kemudian hasil klasifikasi, waktu komputasi dan penggunaan memori akan ditampilkan pada LCD 16x2.

5.1.3.2 Perancangan *Training* Metode Naïve Bayes

Perancangan *training* metode naïve bayes dapat dilihat pada **Gambar 5.4** dibawah:

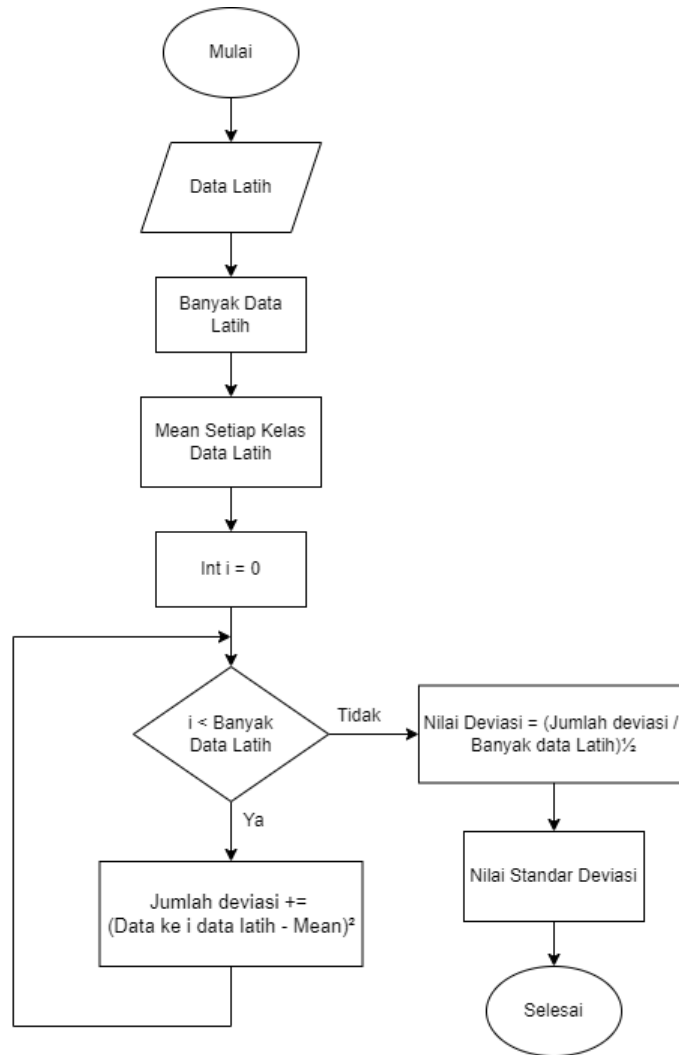


Gambar 5. 5 Perancangan *Traning* Metode Naïve Bayes

Perancangan *training* metode naïve bayes diawali dengan mencari nilai probabilitas dari setiap kelas baik, sedang dan buruk. Kemudian mencari nilai mean dan standar deviasi dari parameter kelembapan dan pH tanah untuk kelas baik, sedang dan buruk sebagai hasil training untuk digunakan pada tahap klasifikasi naïve bayes.

5.1.3.3 Perancangan Kode Program Standar Deviasi

Sebelum melakukan proses klasifikasi oleh algoritma Naïve Bayes, diperlukan nilai standar deviasi dari setiap kelas kualitas tanah. Perancangan kode program standar deviasi dapat dilihat pada **Gambar 5.5**.



Gambar 5. 6 Perancangan Kode Program Standar Deviasi

Sebelum melakukan tahap klasifikasi dengan metode Naïve Bayes, diperlukan nilai standar deviasi yang dibutuhkan nantinya dalam tahap klasifikasi. Perancangan kode program standar deviasi diawali dengan memasukkan data latih, kemudian menghitung banyaknya data latih yang ada. Setelah itu mencari nilai mean dari setiap kelas yang ada pada data latih. Setelah mendapatkan nilai mean dari setiap kelas, langkah selanjutnya adalah melakukan perulangan sebanyak jumlah data latih.

Perulang ini bertujuan untuk menjumlahkan hasil kuadrat dari nilai data latih dikurang dari rata-rata. Setelah selesai melakukan perulangan, hasil yang didapatkan sebelumnya dibagi dengan banyaknya data latih dikurang satu dan kemudian diakarkan dan hasil akhirnya adalah nilai dari standar deviasi.

Pada proses mencari nilai standar deviasi, hal yang pertama dibutuhkan adalah data latih. Data latih yang digunakan pada penelitian ini sebanyak 30 data yang dapat dilihat pada Lampiran 1 sekaligus dengan gambar tanahnya. Data latih diambil dari penelitian sebelumnya yang dilakukan oleh Bhayangkara, A. (2020) yang dimana data latih ini didapat dari sampel tanah pada Balai Penelitian Pertanian Aneka Kacang dan Umbi. Data latih dapat dilihat pada **Tabel 5.5**.

Tabel 5. 5 Data Latih

No	Kelembapan	pH	Kelas
1	80%	6	BAIK
2	61%	7	BAIK
3	65%	7	BAIK
4	74%	7	BAIK
5	70%	7	BAIK
6	67%	7	BAIK
7	77%	6	BAIK
8	73%	7	BAIK
9	69%	6	BAIK
10	69%	7	BAIK
11	65%	6	SEDANG
12	82%	7	SEDANG
13	75%	8	SEDANG
14	68%	5	SEDANG
15	78%	5	SEDANG
16	63%	6	SEDANG
17	53%	6	SEDANG
18	66%	6	SEDANG
19	65%	5	SEDANG
20	53%	5	SEDANG
21	64%	4	BURUK
22	45%	6	BURUK
23	70%	4	BURUK
24	65%	4	BURUK
25	47%	6	BURUK
26	44%	8	BURUK

27	80%	3	BURUK
28	83%	4	BURUK
29	40%	8	BURUK
30	48%	7	BURUK

Setelah mendapatkan data latih maka perhitungan untuk mencari nilai standar deviasi sudah dapat dilakukan. Setiap kelas yang ada pada data latih akan dicari nilai standar deviasinya. Salah satu contoh perhitungan manual mencari nilai standar deviasi dari kelas baik dengan parameter pH:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

$$s = \sqrt{\frac{(6 - 6,7)^2 + (7 - 6,7)^2 + (7 - 6,7)^2 + \dots + (7 - 6,7)^2 + (6 - 6,7)^2 + (7 - 6,7)^2}{10 - 1}}$$

$$s = \sqrt{\frac{2,1}{9}}$$

$$s = 0,48$$

Setelah melakukan perhitungan, nilai standar deviasi setiap kelas dan parameter dapat dilihat pada **Tabel 5.6**.

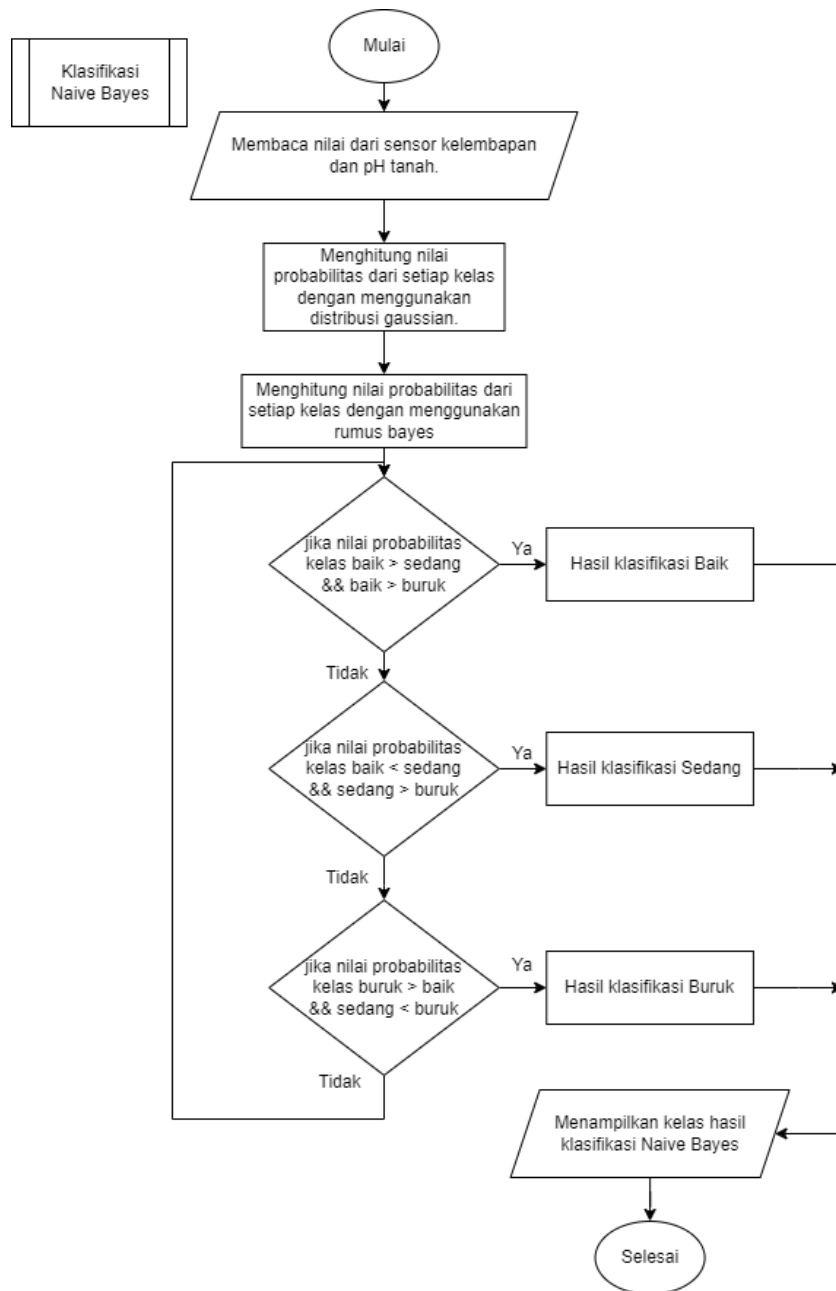
Tabel 5. 6 Nilai Standar Deviasi

Kelas Kualitas Tanah	Nilai Standar Deviasi pH	Nilai Standar Deviasi Kelembapan
Baik	0.48	0.48
Sedang	0.99	0.99
Buruk	1.84	1.84

Hasil dari perhitungan nilai standar deviasi akan digunakan pada tahap selanjutnya yaitu tahap klasifikasi dengan algoritma Naïve Bayes.

5.1.3.4 Perancangan Klasifikasi Metode Naïve Bayes

Perancangan klasifikasi metode naïve bayes dapat dilihat pada **Gambar 5.6** dibawah ini:



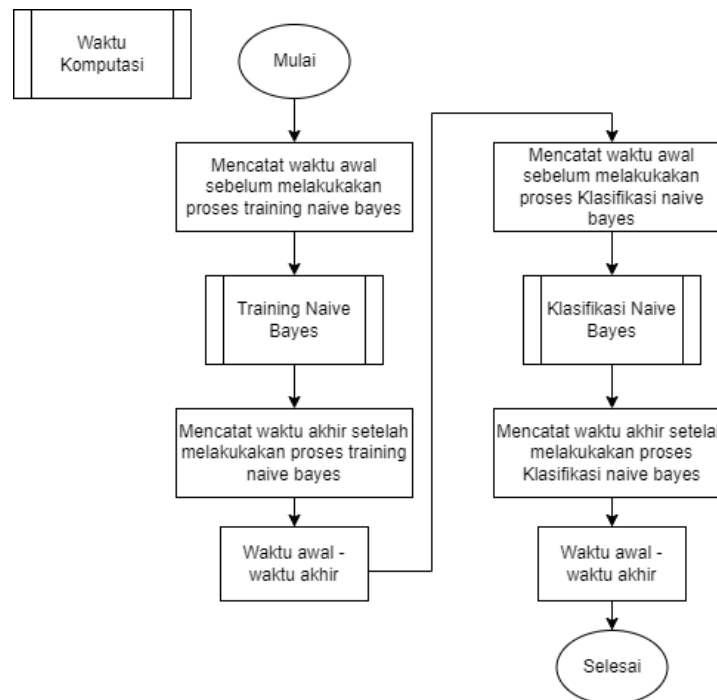
Gambar 5. 7 Perancangan Klasifikasi Metode Naïve Bayes

Perancangan klasifikasi Naïve Bayes diawali dengan membaca nilai dari kedua sensor, yaitu sensor kelembapan dan pH tanah yang nanti akan digunakan sebagai data uji. Selanjutnya dari data uji yang sudah didapatkan, akan dihitung nilai dari probabilitas dari setiap kelas dengan menggunakan rumus distribusi gaussian dan dilanjutkan perhitungan dengan menggunakan rumus bayes. Setelah nilai probabilitas dari setiap kelas yang ada sudah didapatkan, dilanjut dengan membandingkan ketiga

nilai tersebut, yang dimana nilai probabilitas yang paling besar akan menjadi hasil akhir dari klasifikasi.

5.1.3.5 Perancangan Perhitungan Waktu Komputasi dan Penggunaan Memori

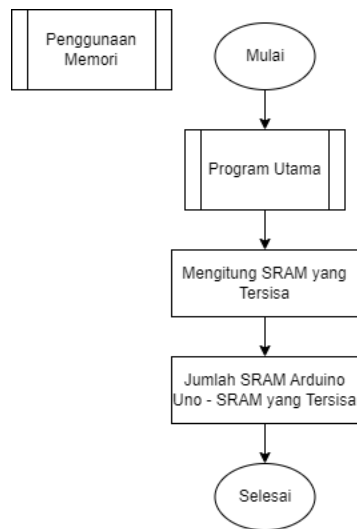
Perancangan perhitungan waktu komputasi dan penggunaan memori dapat dilihat pada **Gambar 5.7**.



Gambar 5. 8 Perancangan Perhitungan Waktu Komputasi

Perancangan perhitungan waktu komputasi dibagi menjadi dua, yaitu yang pertama waktu komputasi training naïve bayes dan waktu komputasi klasifikasi naïve bayes. Perhitungan waktu komputasi training naïve bayes adalah perhitungan banyaknya waktu yang digunakan dalam proses training naïve bayes dengan cara mencatat waktu awal sebelum melakukan proses training kemudian diakhir proses training waktunya juga dicatat kemudian dihitung selisihnya sehingga didapat jumlah waktu komputasi training dengan satuan *millisecond* (ms). Begitu juga hal nya dengan perhitungan waktu komputasi klasifikasi naïve bayes, waktu yang digunakan selama proses klasifikasi naïve bayes.

Selanjutnya adalah perancangan perhitungan penggunaan memori yang dapat dilihat pada **Gambar 5.8**.

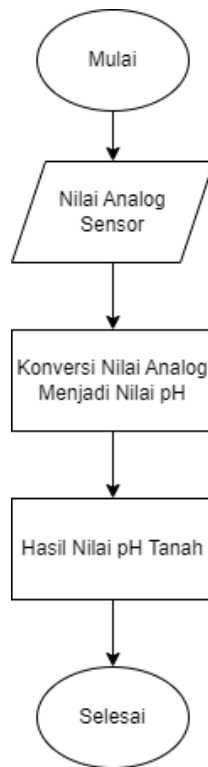


Gambar 5. 9 Perancangan Perhitungan Penggunaan Memori

Perancangan perhitungan jumlah memori yang digunakan metode naïve bayes dalam mengklasifikasikan kualitas tanah diawali dengan menjalankan program utama, kemudian menghitung SRAM yang tersisa pada Arduino Uno dan dilanjutkan dengan jumlah SRAM yang ada pada Arduino Uno dikurang dengan sisa SRAM yang sudah didapatkan tadi yang nantinya akan didapat jumlah memori yang digunakan.

5.1.3.6 Perancangan Kode Program Sensor pH Tanah

Perancangan kode program sensor pH tanah merupakan salah satu bagian penting yang dibutuhkan sistem. Kode program sensor pH tanah harus diperhatikan agar sensor pH tanah dapat melakukan pengambilan data pH pada tanah yang dideteksi. Perancangan kode program sensor pH tanah dapat dilihat pada **Gambar 5.9**.

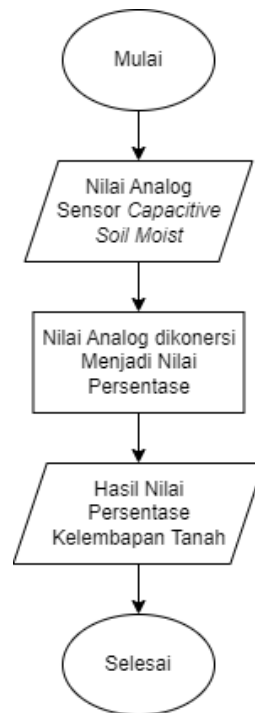


Gambar 5. 10 Perancangan Kode Program Sensor pH Tanah

Perancangan kode program sensor pH tanah diawali dengan pengambilan data oleh sensor berupa data analog. Kemudian data analog yang sudah didapatkan sensor pH tanah dikonversi menjadi nilai pH tanah. Nilai dari sensor pH tanah yang sudah dikonversi nantinya akan digunakan sebagai nilai input pada sistem yang digunakan.

5.1.3.7 Perancangan Kode Program Sensor *Capacitive Soil Moist*

Dalam tahap berikutnya, akan dijelaskan bagaimana langkah-langkah dalam merancang kode program untuk sensor *Capacitive Soil Moist*, yang bertujuan untuk mendapatkan informasi mengenai kelembapan tanah. Tujuan dari proses perancangan ini adalah agar sensor dapat berfungsi sesuai dengan kebutuhan yang diinginkan. Sensor *Capacitive Soil Moist* yang digunakan menghasilkan nilai dalam bentuk analog. Oleh karena itu, diperlukan sebuah kode program yang dapat mengkonversi nilai analog dari sensor menjadi sebuah persentase yang mewakili tingkat kelembapan tanah. Perancangan kode program sensor *Capacitive Soil Moist* dapat dilihat pada **Gambar 5.10**.



Gambar 5. 11 Perancangan Kode Program Sensor *Capacitive Soil Moist*

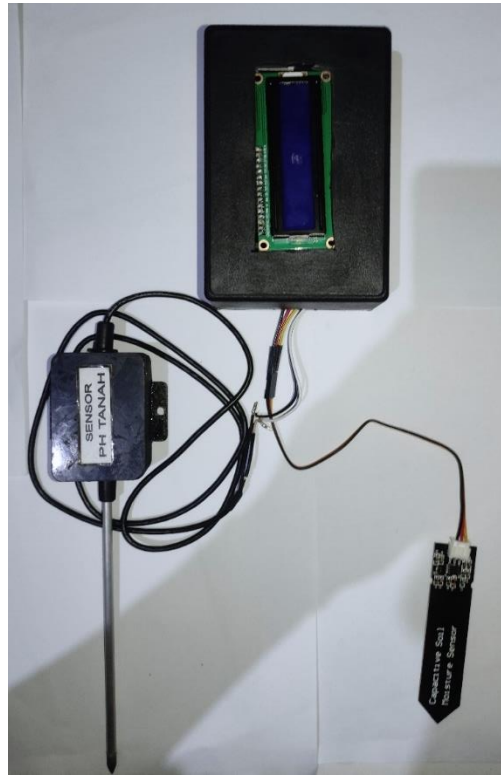
Hasil nilai konversi nilai analog menjadi nilai persentase akan digunakan sebagai nilai inputan pada sistem pendeteksi kualitas tanah tanaman kedelai.

5.2 Implementasi Sistem

Pada implementasi sistem akan menjelaskan bagaimana hasil realisasi dari perancangan yang sudah dilakukan sebelumnya. Beberapa diantaranya yaitu implementasi *prototype* alat, implementasi perangkat keras dan implementasi perangkat lunak.

5.2.1 Implementasi *Prototype* Alat

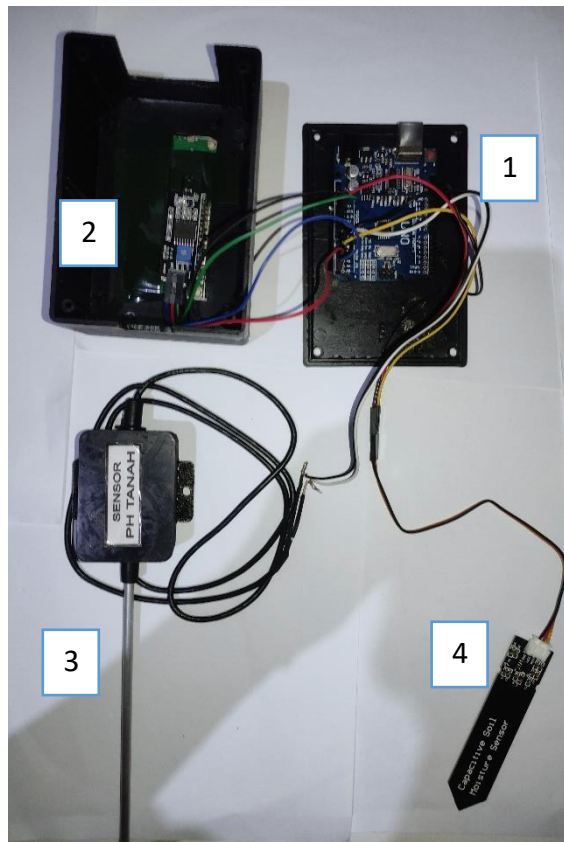
Perancangan *prototype* alat pendeteksi kualitas tanah tanaman kedelai akan diimplementasikan seperti pada **Gambar 5.11**. *Prototype* terdiri dari 4 komponen utama yaitu mikrokontroler Arduino Uno, sensor pH tanah, sensor *Capacitive Soil Moist* dan LCD 16x2. Sumber daya yang digunakan oleh sensor *Capacitive Soil Moist* berasal dari mikrokontroler sedangkan sumber daya mikrokontroler berasal dari kabel USB yang terhubung dengan laptop.



Gambar 5. 12 Implementasi *Prototype* Alat

5.2.2 Implementasi Perangkat Keras

Implementasi perangkat keras alat pendeteksi kualitas tanah tanaman kedelai akan menjelaskan apa saja komponen yang digunakan dan bagaimana cara realisasi agar sistem dapat berjalan dengan baik. Semua komponen yang sudah disiapkan seperti Arduino Uno, sensor pH tanah, sensor *Capacitive Soil Moist* dan LCD 16x2 akan dirangkai sesuai dengan kebutuhan sistem. Pada Arduino Uno akan akan dihubungkan semua komponen seperti sensor pH tanah, sensor *Capacitive Soil Moist* dan LCD 16x2 agar berjalan sesuai dengan yang sudah diprogramkan. Komponen-komponen yang sudah terpasang akan dimasukkan kedalam box elektronik agar terlihat rapi dari luar. LCD 16x2 berada pada bagian luar agar dengan mudah untuk melihat hasil klasifikasi dari sistem. Implementasi perangkat keras dari perancangan yang sudah dilakukan sebelumnya dapat dilihat pada **Gambar 5.12**.



Gambar 5. 13 Implementasi Perangkat Keras

Tabel 5. 7 Keterangan Implementasi Perangkat Keras

No	Keterangan
1	Arduino Uno
2	LCD 16x2
3	Sensor ph Tanah
4	Sensor <i>Capacitive Soil Moist</i>

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak akan menjelaskan proses pembuatan kode program dari sistem pendeteksi kualitas tanah tanaman kedelai, diantaranya yaitu

implementasi kode program algoritma naïve bayes, waktu komputasi, penggunaan memori, sensor pH tanah dan *Capacitive Soil Moist*.

5.2.3.1 Implementasi Kode Program Utama

Implementasi kode program utama akan menjelaskan bagaimana alur dari implementasi kode program utama dari sistem pendeteksi kualitas tanah tanaman kedelai. Kode program utama dapat dilihat pada **Tabel 5.8**.

Tabel 5. 8 Implementasi Kode Program Utama

Kode	Program Utama
1	#include <math.h>
2	#include <LiquidCrystal_I2C.h>
3	LiquidCrystal_I2C lcd (0x3F, 16, 2);
4	#define analogInPin A0 //sambungkan kabel hitam (output) ke
5	pin A0
6	#define sensor A1
7	#define wet 210
8	#define dry 510
9	
10	void display_freeram() {
11	Serial.print(F("- SRAM Digunakan: "));
12	Serial.println(2048 - freeRam());
13	}
14	
15	int freeRam() {
16	extern int __heap_start,*__brkval;
17	int v;
18	return (int)&v - (__brkval == 0
19	? (int)&__heap_start : (int) __brkval);
20	}
21	
22	float standarDeviasi(float latih[], float mean){
23	float dev = 0;
24	int panjangArray = 10;
25	for(int i = 0; i< panjangArray; i++){
26	dev += pow(latih[i]-mean,2);
27	}
28	dev /= panjangArray -1;
29	return sqrt(dev);
30	}
31	float klasifikasiGaussian(float inputan, float mean, float
32	deviasi) {
33	float hasil = 1 / (deviasi * sqrt(2 * PI));
34	float hasil1 = sq(inputan - mean);
35	hasil1 = (hasil1 / (2 * sq(deviasi))) * -1;
36	hasil1 = pow(exp(1), hasil1);
37	hasil = hasil * hasil1;
38	return hasil*100;
39	}
40	

41	float klasifikasi(float nilai1, float nilai2, float
42	peluang){
43	float hasil = nilai1*nilai2*peluang;
44	return hasil;
45	}
46	
47	int sensorValue = 0; //ADC value from sensor
48	float inputPh = 0.0; //pH value after conversion
49	const size_t jmlh = 30;
50	int jmlhBaikPh = 0;
51	int jmlhSedangPh = 0;
52	int jmlhBurukPh = 0;
53	int jmlhBaikKelembapan = 0;
54	int jmlhSedangKelembapan = 0;
55	int jmlhBurukKelembapan = 0;
56	float banyakBaik= 0;
57	float banyakSedang= 0;
58	float banyakBuruk= 0;
59	float latihBaikPh[10];
60	float latihSedangPh[10];
61	float latihBurukPh[10];
62	float latihBaikKelembapan[10];
63	float latihSedangKelembapan[10];
64	float latihBurukKelembapan[10];
65	float latih[jmlh][3] = {
66	{80, 6, 0},
67	{61, 7, 0},
68	{65, 7, 0},
69	{74, 7, 0},
70	{70, 7, 0},
71	{67, 7, 0},
72	{77, 6, 0},
73	{73, 7, 0},
74	{69, 6, 0},
75	{69, 7, 0},
76	{65, 6, 1},
77	{82, 7, 1},
78	{75, 8, 1},
79	{68, 5, 1},
80	{78, 5, 1},
81	{63, 6, 1},
82	{53, 6, 1},
83	{66, 6, 1},
84	{65, 5, 1},
85	{53, 5, 1},
86	{64, 4, 2},
87	{45, 6, 2},
88	{70, 4, 2},
89	{65, 4, 2},
90	{47, 6, 2},
91	{44, 8, 2},
92	{80, 3, 2},
93	{83, 4, 2},
94	{40, 8, 2},

```

95     {48, 7, 2}
96 };
97
98
99 void setup() {
100     lcd.init();
101     lcd.backlight();
102     Serial.begin(9600);
103     int indexArrayBaik = 0;
104     int indexArraySedang = 0;
105     int indexArrayBuruk = 0;
106     for(int i=0; i<jmlh; i++){
107         if(latih[i][2] == 0){
108             banyakBaik += 1;
109             jmlhBaikPh = jmlhBaikPh + latih[i][1];
110             jmlhBaikKelembapan = jmlhBaikKelembapan +
111 latih[i][0];
112             latihBaikPh[indexArrayBaik] = latih[i][1];
113             latihBaikKelembapan[indexArrayBaik] = latih[i][0];
114             indexArrayBaik++;
115         }else if(latih[i][2] == 1){
116             banyakSedang += 1;
117             jmlhSedangPh = jmlhSedangPh + latih[i][1];
118             jmlhSedangKelembapan = jmlhSedangKelembapan +
119 latih[i][0];
120             latihSedangPh[indexArraySedang] = latih[i][1];
121             latihSedangKelembapan[indexArraySedang] =
122 latih[i][0];
123             indexArraySedang++;
124         }else if(latih[i][2] == 2){
125             banyakBuruk += 1;
126             jmlhBurukPh = jmlhBurukPh + latih[i][1];
127             jmlhBurukKelembapan = jmlhBurukKelembapan +
128 latih[i][0];
129             latihBurukPh[indexArrayBuruk] = latih[i][1];
130             latihBurukKelembapan[indexArrayBuruk] = latih[i][0];
131             indexArrayBuruk++;
132         }
133     }
134 }
135
136 void loop(){
137     lcd.clear();
138     //read the analog in value:
139     unsigned long startTimeKlasifikasi = millis();
140     sensorValue = analogRead(analogInPin);
141     inputPh = (-0.0693*sensorValue)+7.3855;
142
143     //sensor kelembapan
144     int value = analogRead(sensor);
145     int inputKelembapan = map(value, wet, dry, 100, 0);
146
147     //mencari peluang dari setiap kelas
148     unsigned long startTimeTraining = millis();

```

```

149 float peluangBaik = banyakBaik/jmlh;
150 float peluangSedang = banyakSedang/jmlh;
151 float peluangBuruk = banyakBuruk/jmlh;
152
153 //mencari Mean
154 float meanBaikPh = jmlhBaikPh/banyakBaik;
155 float meanBaikKelembapan = jmlhBaikKelembapan/banyakBaik;
156 float meanSedangPh = jmlhSedangPh/banyakSedang;
157 float meanSedangKelembapan =
158 jmlhSedangKelembapan/banyakSedang;
159 float meanBurukPh = jmlhBurukPh/banyakBuruk;
160 float meanBurukKelembapan =
161 jmlhBurukKelembapan/banyakBuruk;
162
163 //mencari standar deviasi
164 float stdDevBaikPh = standarDeviasi(latihBaikPh,
165 meanBaikPh);
166 float stdDevSedangPh = standarDeviasi(latihSedangPh,
167 meanSedangPh);
168 float stdDevBurukPh = standarDeviasi(latihBurukPh,
169 meanBurukPh);
170 float stdDevBaikKelembapan =
171 standarDeviasi(latihBaikKelembapan, meanBaikKelembapan);
172 float stdDevSedangKelembapan =
173 standarDeviasi(latihSedangKelembapan, meanSedangKelembapan);
174 float stdDevBurukKelembapan =
175 standarDeviasi(latihBurukKelembapan, meanBurukKelembapan);
176 unsigned long endTimeTraining = millis();
177
178
179
180 //memanggil rumus gaussian
181 double kelasPhBaik = klasifikasiGaussian(inputPh,
182 meanBaikPh, stdDevBaikPh);
183 double kelasPhSedang = klasifikasiGaussian(inputPh,
184 meanSedangPh, stdDevSedangPh);
185 double kelasPhBuruk = klasifikasiGaussian(inputPh,
186 meanBurukPh, stdDevBurukPh);
187 double kelasKelembapanBaik =
188 klasifikasiGaussian(inputKelembapan, meanBaikKelembapan,
189 stdDevBaikKelembapan);
190 double kelasKelembapanSedang =
191 klasifikasiGaussian(inputKelembapan, meanSedangKelembapan,
192 stdDevSedangKelembapan);
193 double kelasKelembapanBuruk =
194 klasifikasiGaussian(inputKelembapan, meanBurukKelembapan,
195 stdDevBurukKelembapan);
196
197 //klasifikasi
198 double kelasBaik = klasifikasi(kelasPhBaik,
199 kelasKelembapanBaik, peluangBaik);
200 double kelasSedang = klasifikasi(kelasPhSedang,
201 kelasKelembapanSedang, peluangSedang);
202

```



```

203     double kelasBuruk = klasifikasi(kelasPhBuruk,
204     kelasKelembapanBuruk, peluangBuruk);
205     unsigned long endTimeKlasifikasi = millis();
206
207     lcd.setCursor(0, 0);
208     lcd.print("Kelembapan: ");
209     lcd.setCursor(13, 0);
210     lcd.print(inputKelembapan);
211     lcd.print("%");
212     lcd.print(" ");
213     lcd.setCursor(0, 1);
214     lcd.print("pH Tanah: ");
215     lcd.setCursor(11, 1);
216     lcd.print(inputPh);
217
218     delay(2000);
219     unsigned long waktuKomputasiKlasifikasi =
220     endTimeKlasifikasi - startTimeKlasifikasi;
221
222     if(kelasBaik > kelasSedang && kelasBaik > kelasBuruk){
223         lcd.clear();
224         lcd.setCursor(0, 0);
225         lcd.print("Kualitas: ");
226         lcd.setCursor(10, 0);
227         display_freeram();
228         lcd.print("Baik");
229         lcd.setCursor(0, 1);
230         lcd.print("Waktu : ");
231         lcd.setCursor(9, 1);
232         lcd.print(waktuKomputasiKlasifikasi);
233         Serial.println(waktuKomputasiKlasifikasi);
234         lcd.setCursor(11, 1);
235         lcd.print("ms");
236
237     }else if(kelasBaik < kelasSedang && kelasSedang >
238     kelasBuruk){
239         lcd.clear();
240         lcd.setCursor(0, 0);
241         lcd.print("Kualitas: ");
242         lcd.setCursor(10, 0);
243         display_freeram();
244         lcd.print("Sedang");
245         lcd.setCursor(0, 1);
246         lcd.print("Waktu : ");
247         lcd.setCursor(9, 1);
248         lcd.print(waktuKomputasiKlasifikasi);
249         Serial.println(waktuKomputasiKlasifikasi);
250         lcd.setCursor(11, 1);
251         lcd.print("ms");
252     }else if(kelasBuruk > kelasBaik && kelasSedang <
253     kelasBuruk){
254         lcd.clear();
255         lcd.setCursor(0, 0);
256         lcd.print("Kualitas: ");

```

257	lcd.setCursor(10, 0);
258	lcd.print("Buruk");
259	display_freeram();
260	lcd.setCursor(0, 1);
261	lcd.print("Waktu : ");
262	lcd.setCursor(9, 1);
263	lcd.print(waktuKomputasiKlasifikasi);
264	Serial.println(waktuKomputasiKlasifikasi);
265	lcd.setCursor(11, 1);
266	lcd.print("ms");
267	};
268	delay(5000);
	}

Kode program utama dibagi sistem terbagi menjadi beberapa bagian, mulai dari inisialisasi mode pin, variabel dan *library* pendukung, *training* Naïve Bayes, klasifikasi Naïve Bayes, waktu komputasi dan penggunaan memori. Pada **Tabel 5.8** pertama akan memanggil *library* yang digunakan seperti baris 1-3, kemudian dilanjutkan dengan inisialisasi mode pin seperti pada baris 4-8, kemudian inisialisasi variabel seperti pada baris 47-65. Selanjutnya adalah tahap *training* Naïve Bayes yang dapat dilihat pada baris 148-176, kemudian dilanjutkan dengan tahap klasifikasi dengan algoritma Naïve Bayes yang dilihat pada baris 183-205, selanjutnya menghitung waktu komputasi yang dapat dilihat pada baris 220-221, dan untuk proses perhitungan penggunaan memori dapat dilihat pada baris 244.

5.2.3.2 Implementasi Kode Program *Training* Naïve Bayes

Implementasi kode program *training* Naïve Bayes akan menjelaskan bagaimana kode program *training* Naïve Bayes diimplementasikan pada Arduino IDE. Implementasinya dapat dilihat pada **Tabel 5.9**.

Tabel 5. 9 Implementasi Kode Program *Training* Naïve Bayes

Training Metode Naïve Bayes	
1	float standarDeviasi(float latih[], float mean){
2	float dev = 0;
3	int panjangArray = 10;
4	for(int i = 0; i< panjangArray; i++){
5	dev += pow(latih[i]-mean,2);
6	}
7	dev /= panjangArray -1;
8	return sqrt(dev);
9	}
10	float latih[jmlh][3] ={
11	{80, 6, 0},
12	{61, 7, 0},
13	{65, 7, 0},
14	{74, 7, 0},
15	{70, 7, 0},

```

16 {67, 7, 0},
17 {77, 6, 0},
18 {73, 7, 0},
19 {69, 6, 0},
20 {69, 7, 0},
21 {65, 6, 1},
22 {82, 7, 1},
23 {75, 8, 1},
24 {68, 5, 1},
25 {78, 5, 1},
26 {63, 6, 1},
27 {53, 6, 1},
28 {66, 6, 1},
29 {65, 5, 1},
30 {53, 5, 1},
31 {64, 4, 2},
32 {45, 6, 2},
33 {70, 4, 2},
34 {65, 4, 2},
35 {47, 6, 2},
36 {44, 8, 2},
37 {80, 3, 2},
38 {83, 4, 2},
39 {40, 8, 2},
40 {48, 7, 2}
41 };
42 void loop(){
43     //mencari peluang dari setiap kelas
44     float peluangBaik = banyakBaik/jmlh;
46     float peluangSedang = banyakSedang/jmlh;
47     float peluangBuruk = banyakBuruk/jmlh;
48
49     //mencari Mean
50     float meanBaikPh = jmlhBaikPh/banyakBaik;
51     float meanBaikKelembapan = jmlhBaikKelembapan/banyakBaik;
52     float meanSedangPh = jmlhSedangPh/banyakSedang;
53     float meanSedangKelembapan =
54     jmlhSedangKelembapan/banyakSedang;
55     float meanBurukPh = jmlhBurukPh/banyakBuruk;
56     float meanBurukKelembapan = jmlhBurukKelembapan/banyakBuruk;
57
58     //mencari standar deviasi
59     float stdDevBaikPh = standarDeviiasi(latihBaikPh, meanBaikPh);
60     float stdDevSedangPh = standarDeviiasi(latihSedangPh,
61     meanSedangPh);
62     float stdDevBurukPh = standarDeviiasi(latihBurukPh,
63     meanBurukPh);
64     float stdDevBaikKelembapan =
65     standarDeviiasi(latihBaikKelembapan, meanBaikKelembapan);
66     float stdDevSedangKelembapan =
67     standarDeviiasi(latihSedangKelembapan, meanSedangKelembapan);
68     float stdDevBurukKelembapan =
69     standarDeviiasi(latihBurukKelembapan, meanBurukKelembapan);
70 }

```

Pada **Tabel 5.9**, baris 1-9 merupakan sebuah fungsi untuk mencari nilai standar deviasi dari setiap parameter dan kelasnya. Selanjutnya pada baris 10-41 merupakan data latih yang digunakan pada sistem pendeteksi kualitas tanah tanaman kedelai. Sebelum memanggil fungsi untuk mencari standar deviasi, hal yang perlu dilakukan adalah mencari peluang dari ketiga kelas kualitas tanah seperti pada baris 44-47. Selanjutnya mencari nilai mean dari setiap parameter dan kelasnya seperti pada baris 50-56, kemudian pada baris 59-70 fungsi untuk mencari nilai standar deviasi akan dipanggil dan akan mendapatkan nilai standar deviasi dari setiap parameter dan ketiga kelas kualitas tanah. Nilai standar deviasi yang sudah didapat nantinya akan digunakan pada tahap klasifikasi Naïve Bayes.

5.2.3.3 Implementasi Kode Program Klasifikasi Naïve Bayes

Implementasi kode program klasifikasi Naïve Bayes akan dijelaskan pada **Tabel 5.10**.

Tabel 5. 10 Implementasi Kode Program Klasifikasi Naïve Bayes

Klasifikasi Metode Naive Bayes	
1	float klasifikasiGaussian(float inputan, float mean, float
2	deviasi) {
3	float hasil = 1 / (deviasi * sqrt(2 * PI));
4	float hasil1 = sq(inputan - mean);
5	hasil1 = (hasil1 / (2 * sq(deviasi))) * -1;
6	hasil1 = pow(exp(1), hasil1);
7	hasil = hasil * hasil1;
8	return hasil*100;
9	}
10	
11	float klasifikasi(float nilai1, float nilai2, float peluang){
12	float hasil = nilai1*nilai2*peluang;
13	return hasil;
14	}
15	void loop(){
16	//memanggil rumus gaussian
17	double kelasPhBaik = klasifikasiGaussian(inputPh, meanBaikPh,
18	stdDevBaikPh);
19	double kelasPhSedang = klasifikasiGaussian(inputPh,
20	meanSedangPh, stdDevSedangPh);
21	double kelasPhBuruk = klasifikasiGaussian(inputPh, meanBurukPh,
22	stdDevBurukPh);
23	double kelasKelembapanBaik =
24	klasifikasiGaussian(inputKelembapan, meanBaikKelembapan,
25	stdDevBaikKelembapan);
26	double kelasKelembapanSedang =
27	klasifikasiGaussian(inputKelembapan, meanSedangKelembapan,
28	stdDevSedangKelembapan);
29	
30	

```

31 double kelasKelembapanBuruk =
32 klasifikasiGaussian(inputKelembapan, meanBurukKelembapan,
33 stdDevBurukKelembapan);
34
35 //klasifikasi
36 double kelasBaik = klasifikasi(kelasPhBaik,
37 kelasKelembapanBaik, peluangBaik);
38 double kelasSedang = klasifikasi(kelasPhSedang,
39 kelasKelembapanSedang, peluangSedang);
40 double kelasBuruk = klasifikasi(kelasPhBuruk,
41 kelasKelembapanBuruk, peluangBuruk);
42 if(kelasBaik > kelasSedang && kelasBaik > kelasBuruk){
43     lcd.clear();
44     lcd.setCursor(0, 0);
46     lcd.print("Kualitas: ");
47     lcd.setCursor(10, 0);
48     display_freeram();
49     lcd.print("Baik");
50     lcd.setCursor(0, 1);
51     lcd.print("Waktu : ");
52     lcd.setCursor(9, 1);
53     lcd.print(waktuKomputasiKlasifikasi);
54     Serial.println(waktuKomputasiKlasifikasi);
55     lcd.setCursor(11, 1);
56     lcd.print("ms");
57
58 }else if(kelasBaik < kelasSedang && kelasSedang > kelasBuruk){
59     lcd.clear();
60     lcd.setCursor(0, 0);
61     lcd.print("Kualitas: ");
62     lcd.setCursor(10, 0);
63     display_freeram();
64     lcd.print("Sedang");
65     lcd.setCursor(0, 1);
66     lcd.print("Waktu : ");
67     lcd.setCursor(9, 1);
68     lcd.print(waktuKomputasiKlasifikasi);
69     Serial.println(waktuKomputasiKlasifikasi);
70     lcd.setCursor(11, 1);
71     lcd.print("ms");
72 }else if(kelasBuruk > kelasBaik && kelasSedang < kelasBuruk){
73     lcd.clear();
74     lcd.setCursor(0, 0);
75     lcd.print("Kualitas: ");
76     lcd.setCursor(10, 0);
77     lcd.print("Buruk");
78     display_freeram();
79     lcd.setCursor(0, 1);
80     lcd.print("Waktu : ");
81     lcd.setCursor(9, 1);
82     lcd.print(waktuKomputasiKlasifikasi);
83     Serial.println(waktuKomputasiKlasifikasi);
84     lcd.setCursor(11, 1);
85     lcd.print("ms");

```

86	};
87	}

Pada **Tabel 5.10**, baris 1-9 merupakan fungsi untuk mencari nilai dari distribusi gaussian dari setiap parameter dan kelasnya. Fungsi ini memiliki tiga parameter yaitu, yang pertama adalah inputan yang dimana parameter ini adalah nilai inputan dari sensor yang digunakan, parameter kedua adalah mean yang dimana sudah didapatkan sebelumnya pada tahap *training* Naïve Bayes. Parameter yang ketiga adalah deviasi yang merupakan nilai standar deviasi yang sudah didapatkan juga pada tahap *training* Naïve Bayes. Pada baris 11-14 merupakan sebuah fungsi untuk mendapatkan nilai akhir probabilitas data uji.

Tahap selanjutnya setelah pembuatan fungsi yang dibutuhkan untuk klasifikasi adalah tahap perulangan yang akan memanggil fungsi-fungsi yang sudah dibuat sebelumnya. Pada baris 17-31 fungsi untuk mencari nilai distribusi gaussian akan dipanggil dan akan mendapatkan nilai distribusi gaussian pada setiap parameter dan kelas kualitas tanah yang ada. Kemudian fungsi untuk mendapatkan nilai akhir probabilitas akan dipanggil seperti pada baris 34-40. Setelah mendapatkan nilai probabilitas data uji pada ketiga kelas kualitas tanah, nilainya akan diidentifikasi untuk menentukan nilai terbesar yang akan menjadi penentu hasil klasifikasi kualitas tanah seperti pada baris 41-85.

5.2.3.4 Implementasi Program Waktu Komputasi dan Penggunaan Memori

Implementasi program waktu komputasi dan penggunaan memori akan dijelaskan pada **Tabel 5.11**.

Tabel 5. 11 Implementasi Program Waktu Komputasi dan Penggunaan Memori

Waktu Komputasi Dan Penggunaan Memori	
1	void display_freeram() {
2	Serial.print(F("- SRAM Digunakan: "));
3	Serial.println(2048 - freeRam());
4	}
5	int freeRam() {
6	extern int __heap_start,*__brkval;
7	int v;
8	return (int)&v - (__brkval == 0
9	? (int)&__heap_start : (int) __brkval);
10	}
11	void setup() {
12	Serial.begin(9600);
13	}
14	void loop(){
15	unsigned long startTimeTraining = millis();
16	
17	//kode Program Training Naïve Bayes

18	
19	unsigned long endTimeTraining = millis();
20	unsigned long waktuKomputasiTraining = endTimeTraining -
21	startTimeTraining;
22	
23	unsigned long startTimeKlasifikasi = millis();
24	
25	//Kode Program Klasifikasi Naïve Bayes
26	
27	unsigned long endTimeKlasifikasi = millis();
28	unsigned long waktuKomputasiKlasifikasi = endTimeKlasifikasi -
29	startTimeKlasifikasi;
30	
31	Serial.println(waktuKomputasiTraining);
32	Serial.println(waktuKomputasiKlasifikasi);
33	display_freeram();
34	}

Pada **Tabel 5.11**, baris 1-10 merupakan sebuah fungsi untuk menghitung jumlah memori yang digunakan saat menjalankan sistem pendeteksi kualitas tanah tanaman kedelai. Kemudian pada baris 15-29 merupakan kode program untuk menghitung waktu komputasi *training* dan klasifikasi metode Naïve Bayes. Setelah melakukan perhitungan waktu komputasi dan penggunaan memori, nilainya akan ditampilkan pada *serial monitor* seperti pada baris 31-33.

5.2.3.5 Implementasi Kode Program Sensor pH Tanah

Implementasi kode program sensor pH tanah akan dijelaskan pada **Tabel 5.12**.

Tabel 5. 12 Implementasi Kode Program Sensor pH Tanah

Sensor pH Tanah	
1	#define analogInPin A0
2	int sensorValue = 0;
3	float inputPh = 0.0;
4	
5	void setup() {
6	Serial.begin(9600);
7	}
8	
9	void loop(){
10	sensorValue = analogRead(analogInPin);
11	inputPh = (-0.0693*sensorValue)+7.3855;
12	}

Pada **Tabel 5.12**, merupakan kode program yang digunakan untuk mendapatkan nilai pH tanah. Jalannya pembacaan data dari data berupa nilai analog menjadi data nilai pH tanah yang dapat dilihat pada baris 10-11.

5.2.3.6 Implementasi Kode Program Sensor *Capacitive Soil Moist*

Sensor *Capacitive Soil Moist* adalah perangkat yang digunakan untuk mengestimasi kelembapan tanah dengan mengukur tingkat kelembapan. Nilai yang diberikan oleh sensor ini berupa nilai analog, yang kemudian dikonversi menjadi persentase kandungan kelembapan tanah. Implementasi kode program Sensor *Capacitive Soil Moist* akan dijelaskan pada **Tabel 5.13**.

Tabel 5. 13 Implementasi Kode Program Sensor *Capacitive Soil Moist*

Sensor Kelembapan Tanah	
1	#define sensor A1
2	#define wet 210
3	#define dry 510
4	
5	void setup() {
6	Serial.begin(9600);
7	}
8	
9	void loop(){
10	//sensor kelembapan
11	int value = analogRead(sensor);
12	int inputKelembapan = map(value, wet, dry, 100, 0);
13	}

Pada **Tabel 5.13** merupakan kode program yang digunakan untuk mendapatkan nilai persentase dari kelembapan tanah yang dideteksi. Sensor akan menghasilkan nilai analog kemudian dikonversi menjadi nilai persentase yang akan digunakan sebagai inputan pada sistem pendeteksi kualitas tanah tanaman kedelai. Konversi dari nilai analog menjadi nilai persentase dapat dilihat pada baris 11-12.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab pengujian dan Analisis ini, digunakan 30 data uji untuk mengamati kinerja dari sistem pendeteksi kualitas tanah tanaman kedelai. Data uji diambil dari beberapa sampel tanah yang berbeda dan menambahkan larutan asam cuka untuk memanipulasi kadar pH dalam tanah.

6.1 Pengujian Metode Naïve Bayes

Metode Naïve Bayes digunakan dalam sistem pendeteksi kualitas tanah tanaman kedelai. Metode ini memiliki peran krusial dalam sistem yang dibuat karena berfungsi sebagai penentu kualitas tanah yang sedang dideteksi. Pengujian metode Naïve Bayes dilakukan dengan membandingkan hasil keluaran sistem dengan tanah yang sebenarnya.

6.1.1 Tujuan Pengujian

Pengujian metode Naïve Bayes dilakukan dengan tujuan untuk mengevaluasi akurasi klasifikasi Naïve Bayes yang digunakan dalam sistem. Pengujian ini bertujuan untuk memastikan apakah hasil dari sistem sesuai dengan tanah yang sebenarnya.

6.1.2 Prosedur Pengujian

Pada bagian ini akan dijelaskan prosedur melakukan pengujian dari akurasi sistem yang telah dibuat yaitu sebagai berikut:

1. Menyiapkan tanah untuk digunakan sebagai data uji.
2. Laptop dihubungkan dengan mikrokontroler Arduino uno yang telah disambungkan dengan sensor pH tanah dan sensor *Capacitive Soil Moist*.
3. Sensor pH tanah dan sensor *Capacitive Soil Moist* ditancapkan kedalam tanah yang akan di uji.
4. Melakukan proses deteksi kualitas tanah menggunakan sistem yang telah dibuat.
5. Melakukan komparasi hasil kelas klasifikasi Naïve Bayes dengan kelas sebenarnya dibandingkan untuk mengetahui tingkat akurasi dari sistem.

6.1.3 Hasil dan Analisis

Pada sub bab ini akan menampilkan hasil pengujian yang telah dilakukan yang dapat dilihat pada **Tabel 6.1**.

Tabel 6. 1 Pengujian Klasifikasi Metode Naïve Bayes

No	Sensor pH Tanah	Sensor <i>Capacitive Soil Moist</i> (%)	Kelas Aktual	Hasil Klasifikasi
1	6.35	74.5	Baik	Baik
2	7.28	69.1	Baik	Baik
3	7.44	75.7	Baik	Baik
4	7.01	75.3	Baik	Baik
5	6.58	74.4	Baik	Baik
6	7.18	7.18	Baik	Baik
7	6.24	79.6	Baik	Baik
8	6.30	70.3	Baik	Baik
9	7.30	67.2	Baik	Baik
10	6.99	62.4	Baik	Baik
11	7.01	65.3	Baik	Baik
12	6.27	60.8	Baik	Baik
13	6.78	73.3	Baik	Baik
14	6.12	64.6	Sedang	Baik
15	5.91	65.8	Sedang	Baik
16	6.30	57.2	Sedang	Sedang
17	5.23	75.9	Sedang	Sedang
18	5.90	71.5	Sedang	Sedang
19	5.42	59.2	Sedang	Sedang
20	5.87	55.3	Sedang	Sedang
21	5.40	66.2	Sedang	Sedang
22	5.03	71.9	Sedang	Sedang
23	5.09	70.7	Sedang	Sedang

24	7.21	83.2	Sedang	Sedang
25	5.13	55.9	Buruk	Sedang
26	4.90	46.5	Buruk	Buruk
27	4.07	44.9	Buruk	Buruk
28	4.11	72.2	Buruk	Buruk
29	7.12	42.2	Buruk	Buruk
30	8.27	42.6	Buruk	Buruk

Pada **Tabel 6.1** merupakan hasil percobaan yang dilakukan untuk mengetahui tingkat akurasi dari metode klasifikasi Naïve Bayes. Percobaan dilakukan sebanyak 30 kali dengan nilai ph tanah dan kelembapan yang berbeda beda. Untuk mengetahui tingkat akurasi dari metode klasifikasi Naïve Bayes, melakukan perhitungan dengan persamaan **6.1**.

(6.1)

$$Akurasi = \frac{Jumlah\ data\ uji - Jumlah\ data\ yang\ tidak\ sesuai}{Jumlah\ data\ uji} \times 100\%$$

Setelah melakukan analisis terhadap percobaan yang telah dilakukan sebanyak 30 kali, terdapat 3 data yang tidak sesuai dengan kelas aktual. Sehingga dapat dihitung tingkat akurasi dari metode klasifikasi Naïve Bayes dengan perhitungan berikut:

$$Akurasi = \frac{30 - 3}{30} \times 100$$

$$Akurasi = 90\%$$

Dari perhitungan yang telah dilakukan maka didapatkan tingkat akurasi dari metode klasifikasi Naïve Bayes yaitu sebesar 90%. Akurasi dari metode Naïve Bayes tersebut dapat ditingkatkan dengan menggunakan data latih yang lebih banyak lagi dan lebih beragam.

6.2 Pengujian Waktu Komputasi Metode Naïve Bayes

Pengujian waktu komputasi merupakan pengujian yang dilakukan untuk mengetahui waktu komputasi yang dibutuhkan metode Naïve Bayes untuk melakukan klasifikasi kualitas tanah tanaman kedelai.

6.2.1 Tujuan Pengujian

Pengujian waktu komputasi bertujuan untuk mengetahui seberapa efektif metode Naïve Bayes dalam melakukan klasifikasi pada sistem. Semakin kecil waktu komputasinya, semakin baik sistem yang telah dibuat.

6.2.2 Prosedur Pengujian

Prosedur pengujian akan menjelaskan langkah-langkah yang harus dilakukan dalam melakukan pengujian waktu komputasi Naïve Bayes. Prosedur pengujian akan dijelaskan sebagai berikut:

1. Menyiapkan tanah untuk digunakan sebagai data uji.
2. Laptop dihubungkan dengan mikrokontroler Arduino uno yang telah disambungkan dengan sensor pH tanah dan sensor *Capacitive Soil Moist*.
3. Sensor pH tanah dan sensor *Capacitive Soil Moist* ditancapkan kedalam tanah yang akan di uji.
4. Melakukan proses deteksi kualitas tanah menggunakan sistem yang telah dibuat.
5. Membuka serial monitor pada Arduino Ide.
6. Mencatat nilai waktu komputasi dan menghitung waktu rata-rata yang dibutuhkan.

6.2.3 Hasil dan Analisis

Pada sub bab ini akan menjelaskan hasil dan analisis waktu komputasi yang dibutuhkan sistem untuk melakukan proses klasifikasi kualitas tanah tanaman kedelai. Pengujian waktu komputasi dilakukan sebanyak 30 kali sesuai dengan data uji dengan waktu komputasi dibagi menjadi dua bagian yaitu waktu komputasi *training* Naïve Bayes dan waktu komputasi klasifikasi Naïve Bayes.

6.2.3.1 Waktu Komputasi *Training* Naïve Bayes

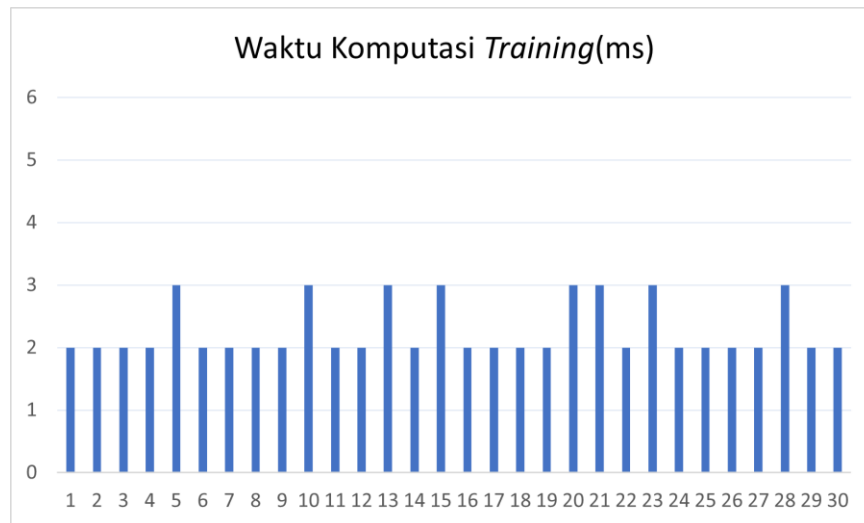
Sebelum melakukan klasifikasi perlu dilakukan proses *training* untuk membentuk model matematika data *training*. Berikut pada **Tabel 6.2** akan disajikan hasil pengujian waktu komputasi *training* Naïve Bayes:

Tabel 6. 2 Pengujian Waktu Komputasi *Traning* Naïve Bayes

No	Waktu Komputasi(ms)
1	2
2	2
3	2
4	2
5	3
6	2
7	2
8	2
9	2
10	3
11	2
12	2
13	3
14	2
15	3
16	2
17	2
18	2
19	2
20	3
21	3
22	2
23	3
24	2
25	2
26	2
27	2
28	3

29	2
30	2

Dari hasil pengujian yang telah dilakukan sebanyak 30 kali, waktu komputasi berada pada rentan waktu 2-3 ms. Berikut pada **Gambar 6.1** adalah grafik dari waktu komputasi *training* Naïve Bayes.



Gambar 6. 1 Grafik Waktu Komputasi *Traning* Naïve Bayes

Setelah melakukan pengujian sebanyak 30 kali, maka akan dihitung rata-rata waktu komputasi *training* Naïve Bayes dengan persamaan **6.2**.

$$Rata - rata = \frac{\sum Waktu Komputasi training}{Jumlah data training} \quad (6.2)$$

$$Rata - rata = \frac{68}{30} ms$$

$$Rata - rata = 2,26 ms$$

Dari perhitungan mencari rata-rata waktu komputasi dengan menggunakan persamaan **6.2**, didapatkan hasil bahwa waktu komputasi *training* Naïve Bayes membutuhkan waktu rata-rata 2,26 ms.

6.2.3.2 Waktu Komputasi Klasifikasi Naïve Bayes

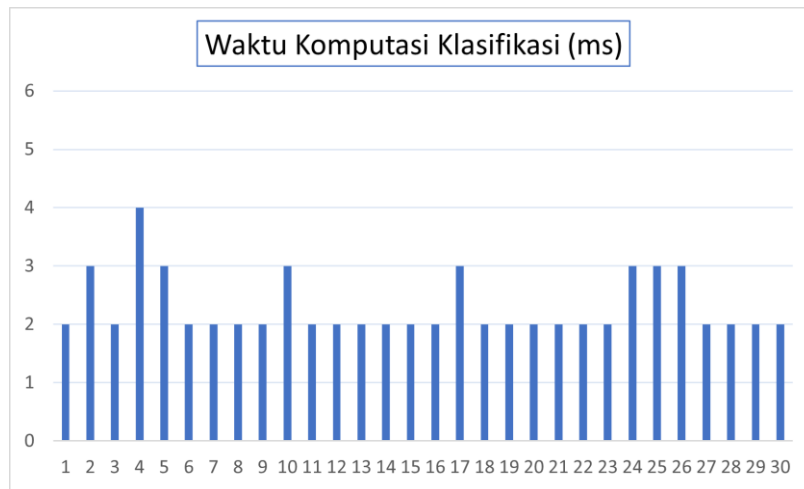
Setelah melakukan *training* metode Naïve Bayes, selanjutnya adalah proses klasifikasi Naïve Bayes. Berikut disajikan pengujian waktu komputasi klasifikasi Naïve Bayes pada **Tabel 6.3**.

Tabel 6. 3 Pengujian Waktu Komputasi Klasifikasi Naïve Bayes

No	Waktu Komputasi(ms)
1	2
2	3
3	2
4	4
5	3
6	2
7	2
8	2
9	2
10	3
11	2
12	2
13	2
14	2
15	2
16	2
17	3
18	2
19	2
20	2
21	2
22	2
23	2
24	3
25	3
26	3
27	2
28	2

29	2
30	2

Pengujian waktu komputasi klasifikasi Naïve Bayes dilakukan sebanyak 30 kali dapat dilihat pada **Gambar 6.2**.



Gambar 6. 2 Grafik Waktu Komputasi Klasifikasi Naïve Bayes

Berbeda dengan hasil percobaan waktu komputasi *training*, waktu komputasi klasifikasi Naïve Bayes berada pada rentan waktu 2-4 ms dengan rata-rata waktu yang dibutuhkan adalah 2,3 ms.

6.3 Penggunaan SRAM

SRAM merupakan singkatan dari *Static random-access memory*. SRAM digunakan mikrokontroler untuk menyimpan variabel-variabel yang akan dimanipulasi nilainya. Arduino Uno memiliki SRAM sebesar 2048 *bytes*.

6.3.1 Tujuan Pengujian

Pengujian penggunaan SRAM dilakukan dengan tujuan untuk mengetahui seberapa efisien algoritma Naïve Bayes dalam menggunakan SRAM pada mikrokontroler Arduino Uno. Pada pengujian ini akan menghitung jumlah SRAM yang digunakan metode Naïve Bayes dalam menjalankan sistem pendeteksi kualitas tanah tanaman kedelai.

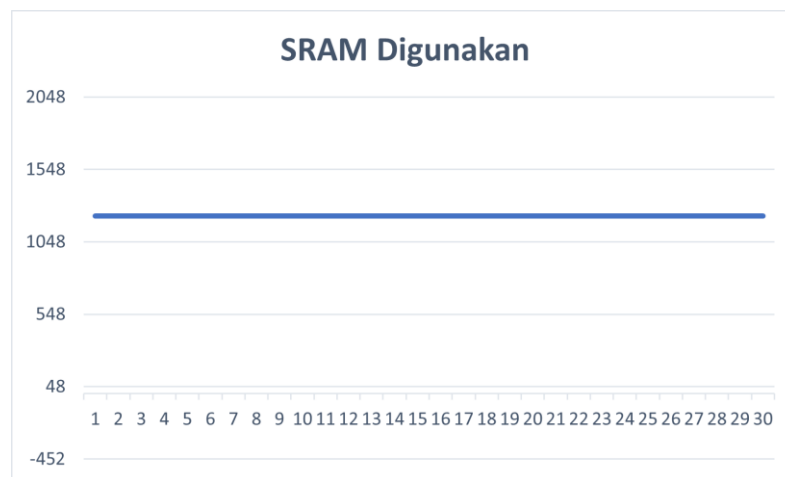
6.3.2 Prosedur Pengujian

Pada bagian ini akan menjelaskan prosedur dalam melakukan pengujian penggunaan SRAM yaitu sebagai berikut:

1. Menyiapkan tanah untuk digunakan sebagai data uji.
2. Laptop dihubungkan dengan mikrokontroler Arduino uno yang telah disambungkan dengan sensor pH tanah dan sensor *Capacitive Soil Moist*.
3. Sensor pH tanah dan sensor *Capacitive Soil Moist* ditancapkan kedalam tanah yang akan di uji.
4. Melakukan proses deteksi kualitas tanah menggunakan sistem yang telah dibuat.
5. Membuka serial monitor pada Arduino Ide.
6. Mencatat jumlah memori yang digunakan algoritma Naïve Bayes pada Arduino Uno.

6.3.3 Hasil dan Analisis

Pada sub bab ini akan menjelaskan hasil dan analisis dari penggunaan SRAM yang dibutuhkan sistem untuk menjalankan algoritma Naïve Bayes pada Arduino Uno. Pengujian dilakukan sebanyak 30 kali sesuai dengan data uji. Hasil pengujian dapat dilihat pada **Gambar 6.3** berikut:



Gambar 6. 3 Grafik Pengujian Penggunaan SRAM

Pada pengujian sebanyak 30 kali, jumlah SRAM yang digunakan algoritma Naïve Bayes adalah konstan dengan nilai sebesar 1226 *bytes*. Angka tersebut termasuk kategori efisien dalam menjalankan sistem yang telah dibuat. Angka

tersebut juga menjelaskan bahwa mikrokontroler Arduino Uno masih dapat menampung beberapa fitur dan kelas atau jumlah data latih jika dibutuhkan.

BAB 7 PENUTUP

Pada bab ini akan membahas kesimpulan yang ditarik dan saran yang diberikan berdasarkan pengujian dan analisis yang telah dilakukan.

7.1 Kesimpulan

Kesimpulan yang ditarik untuk menjawab rumusan masalah setelah melakukan pengujian dan analisis adalah sebagai berikut:

1. Metode klasifikasi Naïve Bayes yang telah dilakukan pengujian sebanyak 30 kali mendapatkan tingkat akurasi sebesar 90% yang dihitung berdasarkan kelas tanah baik, sedang dan buruk. Angka tersebut dinilai sudah cukup baik dibandingkan metode klasifikasi *K-Nearest Neighbor(K-NN)* yang memiliki tingkat akurasi pada angka 86,6% yang telah dilakukan pada penelitian terdahulu oleh Bhayangkara, A. (2020)
2. Waktu komputasi rata-rata sistem menggunakan metode klasifikasi Naïve Bayes adalah 2,26 ms untuk proses *training* Naïve Bayes dan 2,3 ms untuk proses klasifikasi Naïve Bayes. Waktu komputasi tersebut dinilai sudah cukup efektif.
3. Penggunaan SRAM pada sistem pendeteksi kualitas tanah tanaman kedelai menggunakan metode Naïve Bayes pada Arduino Uno rata-rata adalah 1226 *bytes*. Penggunaan SRAM tersebut dinilai sudah cukup baik dikarenakan masih terdapat ruang yang cukup pada Arduino Uno jika ingin menambah data latih atau fitur yang dibutuhkan.

7.2 Saran

Pada bab ini akan jelaskan mengenai saran untuk membantu penelitian kedepannya untuk mengembangkan sistem yang telah dibangun agar menjadi lebih baik lagi.

1. Menambah data latih lebih banyak lagi supaya ketika sistem melakukan klasifikasi menggunakan metode Naïve Bayes didapatkan hasil yang lebih akurat dan lebih baik.
2. Menambahkan fitur lebih banyak lagi seperti: kandungan nutrisi dalam tanah, kandungan organik dalam tanah.
3. Meningkatkan akurasi sensor yang digunakan agar data yang didapat lebih akurat sehingga hasil klasifikasi dari sistem akan lebih akurat juga.

DAFTAR PUSTAKA

- Bhayangkara, A., Setiawan, E., & Firtiyah, H. (2020). Sistem Pendeteksi Kualitas Tanah Tanaman Kedelai Menggunakan Metode K-Nearest Neighbor (K-NN) dengan Arduino Nano. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(8), 2513-2519. Diambil dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/7691>.
- Anwar, K., Syauqy, D., Fitriyah, H. (2018). Sistem Pendeteksi Kandungan Nutrisi dalam Tanah Berdasarkan Warna dan Kelembapan dengan Menggunakan Metode Naive Bayes. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(9), 2491-2498. Diambil dari <https://garuda.kemdikbud.go.id/documents/detail/638996>.
- Firdaus, J., Budi, A., & Setiawan, E. (2023). Analisis Performa Algoritma Machine Learning Pada Perangkat Embedded Atmega328P. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(2), 245-254. Diambil dari <https://jtiik.ub.ac.id/index.php/jtiik/article/view/6196>
- Sudaryono. (2007). Perbaikan teknologi kedelai mendukung PTT kedelai pada lahan kering masam. *Laporan Teknis Tahun 2007. Balai Penelitian Tanaman Kacang-kacangan dan Umbi-umbian*, Malang. 25 hlm.
- Damardjati, D.S., Marwoto, D.K.S. Swastika, D.M. Arsyad, & Y. Hilman. (2005). Prospek dan arah pengembangan agribisnis kedelai. Badan *Litbang Pertanian. Departemen Pertanian, Jakarta*.
- BPS. (2014). Angka Tetap Tahun 2013 dan Angka Ramalan I tahun 2014 produksi tanaman pangan. *BPS, Jakarta*.
- Sumarno, Ahmad Gozi, M. (2013). Persyaratan Tumbuh dan Wilayah Produksi Kedelai di Indonesia. *Balai Penelitian Tanaman Aneka Kacang dan Umbi. Malang*.
- Itoo, F., Meenakshi, & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *Int. j. inf. tecnol.* 13(4), 1503–1511.

Indobot. (2023). Monitoring pH dan Kelembaban Tanah Menggunakan Platform Thingspeak. [Online] Available at : <https://indobot.co.id/blog/iot-pertanian-monitoring-ph-dan-kelembaban-tanah-menggunakan-platform-thingspeak/> [Diakses 23 Maret 2023].

Suryana, T.(2021). Capacitive Soil Moisture Sensor Untuk Mengukur Kelembaban Tanah. *Jurnal Komputa Unikom* 2021. Diambil dari: <https://repository.unikom.ac.id/68742/> [Diakses 3 April 2023]

LAMPIRAN A - DATA SISTEM

A. Data Latih Klasifikasi Naïve Bayes

No	Kelembapan	pH	Kelas
1	80%	6	BAIK
2	61%	7	BAIK
3	65%	7	BAIK
4	74%	7	BAIK
5	70%	7	BAIK
6	67%	7	BAIK
7	77%	6	BAIK
8	73%	7	BAIK
9	69%	6	BAIK
10	69%	7	BAIK
11	65%	6	SEDANG
12	82%	7	SEDANG
13	75%	8	SEDANG
14	68%	5	SEDANG
15	78%	5	SEDANG
16	63%	6	SEDANG
17	53%	6	SEDANG
18	66%	6	SEDANG
19	65%	5	SEDANG
20	53%	5	SEDANG
21	64%	4	BURUK
22	45%	6	BURUK
23	70%	4	BURUK
24	65%	4	BURUK
25	47%	6	BURUK
26	44%	8	BURUK
27	80%	3	BURUK
28	83%	4	BURUK
29	40%	8	BURUK
30	48%	7	BURUK

B. Data Uji Klasifikasi Naïve Bayes

No	Sensor pH Tanah	Sensor <i>Capacitive Soil Moist</i> (%)	Kelas Aktual	Hasil Klasifikasi
1	6.35	74.5	Baik	Baik
2	7.28	69.1	Baik	Baik
3	7.44	75.7	Baik	Baik
4	7.01	75.3	Baik	Baik
5	6.58	74.4	Baik	Baik
6	7.18	7.18	Baik	Baik
7	6.24	79.6	Baik	Baik
8	6.30	70.3	Baik	Baik
9	7.30	67.2	Baik	Baik
10	6.99	62.4	Baik	Baik
11	7.01	65.3	Baik	Baik
12	6.27	60.8	Baik	Baik
13	6.78	73.3	Baik	Baik
14	6.12	64.6	Sedang	Baik
15	5.91	65.8	Sedang	Baik
16	6.30	57.2	Sedang	Sedang
17	5.23	75.9	Sedang	Sedang
18	5.90	71.5	Sedang	Sedang
19	5.42	59.2	Sedang	Sedang
20	5.87	55.3	Sedang	Sedang
21	5.40	66.2	Sedang	Sedang
22	5.03	71.9	Sedang	Sedang
23	5.09	70.7	Sedang	Sedang

24	7.21	83.2	Sedang	Sedang
25	5.13	55.9	Buruk	Sedang
26	4.90	46.5	Buruk	Buruk
27	4.07	44.9	Buruk	Buruk
28	4.11	72.2	Buruk	Buruk
29	7.12	42.2	Buruk	Buruk
30	8.27	42.6	Buruk	Buruk

LAMPIRAN B - KODE PROGRAM UTAMA

Kode Program Utama	
1	#include <math.h>
2	#include <LiquidCrystal_I2C.h>
3	LiquidCrystal_I2C lcd (0x3F, 16, 2);
4	#define analogInPin A0 //sambungkan kabel hitam (output) ke
5	pin A0
6	#define sensor A1
7	#define wet 210
8	#define dry 510
9	
10	void display_freeram() {
11	Serial.print(F("- SRAM Digunakan: "));
12	Serial.println(2048 - freeRam());
13	}
14	
15	int freeRam() {
16	extern int __heap_start,*__brkval;
17	int v;
18	return (int)&v - (__brkval == 0
19	? (int)&__heap_start : (int) __brkval);
20	}
21	
22	float standarDeviasi(float latih[], float mean){
23	float dev = 0;
24	int panjangArray = 10;
25	for(int i = 0; i< panjangArray; i++){
26	dev += pow(latih[i]-mean,2);
27	}
28	dev /= panjangArray -1;
29	return sqrt(dev);
30	}
31	float klasifikasiGaussian(float inputan, float mean, float
32	deviasi) {
33	float hasil = 1 / (deviasi * sqrt(2 * PI));
34	float hasil1 = sq(inputan - mean);
35	hasil1 = (hasil1 / (2 * sq(deviasi))) * -1;
36	hasil1 = pow(exp(1), hasil1);
37	hasil = hasil * hasil1;
38	return hasil*100;
39	}
40	
41	float klasifikasi(float nilai1, float nilai2, float
42	peluang){
43	float hasil = nilai1*nilai2*peluang;
44	return hasil;
45	}
46	
47	int sensorValue = 0; //ADC value from sensor
48	float inputPh = 0.0; //pH value after conversion
49	const size_t jmlh = 30;
50	int jmlhBaikPh = 0;
51	int jmlhSedangPh = 0;

```

52 int jmlhBurukPh = 0;
53 int jmlhBaikKelembapan = 0;
54 int jmlhSedangKelembapan = 0;
55 int jmlhBurukKelembapan = 0;
56 float banyakBaik= 0;
57 float banyakSedang= 0;
58 float banyakBuruk= 0;
59 float latihBaikPh[10];
60 float latihSedangPh[10];
61 float latihBurukPh[10];
62 float latihBaikKelembapan[10];
63 float latihSedangKelembapan[10];
64 float latihBurukKelembapan[10];
65 float latih[jmlh][3] ={
66     {80, 6, 0},
67     {61, 7, 0},
68     {65, 7, 0},
69     {74, 7, 0},
70     {70, 7, 0},
71     {67, 7, 0},
72     {77, 6, 0},
73     {73, 7, 0},
74     {69, 6, 0},
75     {69, 7, 0},
76     {65, 6, 1},
77     {82, 7, 1},
78     {75, 8, 1},
79     {68, 5, 1},
80     {78, 5, 1},
81     {63, 6, 1},
82     {53, 6, 1},
83     {66, 6, 1},
84     {65, 5, 1},
85     {53, 5, 1},
86     {64, 4, 2},
87     {45, 6, 2},
88     {70, 4, 2},
89     {65, 4, 2},
90     {47, 6, 2},
91     {44, 8, 2},
92     {80, 3, 2},
93     {83, 4, 2},
94     {40, 8, 2},
95     {48, 7, 2}
96 };
97
98
99 void setup() {
100     lcd.init();
101     lcd.backlight();
102     Serial.begin(9600);
103     int indexArrayBaik = 0;
104     int indexArraySedang = 0;
105     int indexArrayBuruk = 0;

```

```

106     for(int i=0; i<jmlh; i++){
107         if(latih[i][2] == 0){
108             banyakBaik += 1;
109             jmlhBaikPh = jmlhBaikPh + latih[i][1];
110             jmlhBaikKelembapan = jmlhBaikKelembapan +
111 latih[i][0];
112             latihBaikPh[indexArrayBaik] = latih[i][1];
113             latihBaikKelembapan[indexArrayBaik] = latih[i][0];
114             indexArrayBaik++;
115         }else if(latih[i][2] == 1){
116             banyakSedang += 1;
117             jmlhSedangPh = jmlhSedangPh + latih[i][1];
118             jmlhSedangKelembapan = jmlhSedangKelembapan +
119 latih[i][0];
120             latihSedangPh[indexArraySedang] = latih[i][1];
121             latihSedangKelembapan[indexArraySedang] =
122 latih[i][0];
123             indexArraySedang++;
124         }else if(latih[i][2] == 2){
125             banyakBuruk += 1;
126             jmlhBurukPh = jmlhBurukPh + latih[i][1];
127             jmlhBurukKelembapan = jmlhBurukKelembapan +
128 latih[i][0];
129             latihBurukPh[indexArrayBuruk] = latih[i][1];
130             latihBurukKelembapan[indexArrayBuruk] = latih[i][0];
131             indexArrayBuruk++;
132         }
133     }
134 }
135
136 void loop(){
137     lcd.clear();
138     //read the analog in value:
139     unsigned long startTimeKlasifikasi = millis();
140     sensorValue = analogRead(analogInPin);
141     inputPh = (-0.0693*sensorValue)+7.3855;
142
143     //sensor kelembapan
144     int value = analogRead(sensor);
145     int inputKelembapan = map(value, wet, dry, 100, 0);
146
147     //mencari peluang dari setiap kelas
148     unsigned long startTimeTraining = millis();
149     float peluangBaik = banyakBaik/jmlh;
150     float peluangSedang = banyakSedang/jmlh;
151     float peluangBuruk = banyakBuruk/jmlh;
152
153     //mencari Mean
154     float meanBaikPh = jmlhBaikPh/banyakBaik;
155     float meanBaikKelembapan = jmlhBaikKelembapan/banyakBaik;
156     float meanSedangPh = jmlhSedangPh/banyakSedang;
157     float meanSedangKelembapan =
158 jmlhSedangKelembapan/banyakSedang;
159     float meanBurukPh = jmlhBurukPh/banyakBuruk;

```

```

160     float meanBurukKelembapan =
161     jmlhBurukKelembapan/banyakBuruk;
162
163     //mencari standar deviasi
164     float stdDevBaikPh = standarDeviasi(latihBaikPh,
165     meanBaikPh);
166     float stdDevSedangPh = standarDeviasi(latihSedangPh,
167     meanSedangPh);
168     float stdDevBurukPh = standarDeviasi(latihBurukPh,
169     meanBurukPh);
170     float stdDevBaikKelembapan =
171     standarDeviasi(latihBaikKelembapan, meanBaikKelembapan);
172     float stdDevSedangKelembapan =
173     standarDeviasi(latihSedangKelembapan, meanSedangKelembapan);
174     float stdDevBurukKelembapan =
175     standarDeviasi(latihBurukKelembapan, meanBurukKelembapan);
176     unsigned long endTimeTraining = millis();
177
178
179
180     //memanggil rumus gaussian
181     double kelasPhBaik = klasifikasiGaussian(inputPh,
182     meanBaikPh, stdDevBaikPh);
183     double kelasPhSedang = klasifikasiGaussian(inputPh,
184     meanSedangPh, stdDevSedangPh);
185     double kelasPhBuruk = klasifikasiGaussian(inputPh,
186     meanBurukPh, stdDevBurukPh);
187     double kelasKelembapanBaik =
188     klasifikasiGaussian(inputKelembapan, meanBaikKelembapan,
189     stdDevBaikKelembapan);
190     double kelasKelembapanSedang =
191     klasifikasiGaussian(inputKelembapan, meanSedangKelembapan,
192     stdDevSedangKelembapan);
193     double kelasKelembapanBuruk =
194     klasifikasiGaussian(inputKelembapan, meanBurukKelembapan,
195     stdDevBurukKelembapan);
196
197     //klasifikasi
198     double kelasBaik = klasifikasi(kelasPhBaik,
199     kelasKelembapanBaik, peluangBaik);
200     double kelasSedang = klasifikasi(kelasPhSedang,
201     kelasKelembapanSedang, peluangSedang);
202     double kelasBuruk = klasifikasi(kelasPhBuruk,
203     kelasKelembapanBuruk, peluangBuruk);
204     unsigned long endTimeKlasifikasi = millis();
205
206     lcd.setCursor(0, 0);
207     lcd.print("Kelembapan: ");
208     lcd.setCursor(13, 0);
209     lcd.print(inputKelembapan);
210     lcd.print("%");
211     lcd.print(" ");
212     lcd.setCursor(0, 1);
213     lcd.print("pH Tanah: ");

```

```

214     lcd.setCursor(11, 1);
215     lcd.print(inputPh);
216
217     delay(2000);
218     unsigned long waktuKomputasiKlasifikasi =
219     endTimeKlasifikasi - startTimeKlasifikasi;
220
221     if(kelasBaik > kelasSedang && kelasBaik > kelasBuruk){
222         lcd.clear();
223         lcd.setCursor(0, 0);
224         lcd.print("Kualitas: ");
225         lcd.setCursor(10, 0);
226         display_freeram();
227         lcd.print("Baik");
228         lcd.setCursor(0, 1);
229         lcd.print("Waktu : ");
230         lcd.setCursor(9, 1);
231         lcd.print(waktuKomputasiKlasifikasi);
232         Serial.println(waktuKomputasiKlasifikasi);
233         lcd.setCursor(11, 1);
234         lcd.print("ms");
235     }else if(kelasBaik < kelasSedang && kelasSedang >
236     kelasBuruk){
237         lcd.clear();
238         lcd.setCursor(0, 0);
239         lcd.print("Kualitas: ");
240         lcd.setCursor(10, 0);
241         display_freeram();
242         lcd.print("Sedang");
243         lcd.setCursor(0, 1);
244         lcd.print("Waktu : ");
245         lcd.setCursor(9, 1);
246         lcd.print(waktuKomputasiKlasifikasi);
247         Serial.println(waktuKomputasiKlasifikasi);
248         lcd.setCursor(11, 1);
249         lcd.print("ms");
250     }else if(kelasBuruk > kelasBaik && kelasSedang <
251     kelasBuruk){
252         lcd.clear();
253         lcd.setCursor(0, 0);
254         lcd.print("Kualitas: ");
255         lcd.setCursor(10, 0);
256         lcd.print("Buruk");
257         display_freeram();
258         lcd.setCursor(0, 1);
259         lcd.print("Waktu : ");
260         lcd.setCursor(9, 1);
261         lcd.print(waktuKomputasiKlasifikasi);
262         Serial.println(waktuKomputasiKlasifikasi);
263         lcd.setCursor(11, 1);
264         lcd.print("ms");
265     };
266     delay(5000);
267 }

```

LAMPIRAN C - DOKUMENTASI

C.1 Dokumentasi

