

**IMPLEMENTASI SISTEM *MONITORING* DAN PENYIRAMAN
TANAMAN OTOMATIS BERBASIS *INTERNET OF THINGS*
MENGUNAKAN *K-NEAREST NEIGHBOUR***

SKRIPSI

Oleh :
RIZQI ARI PUTRA
NIM. 16650079



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**IMPLEMENTASI SISTEM *MONITORING* DAN PENYIRAMAN
TANAMAN OTOMATIS BERBASIS *INTERNET OF THINGS*
MENGUNAKAN *K-NEAREST NEIGHBOUR***

SKRIPSI

HALAMAN PE
NGAJU
AN

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
RIZQI ARI PUTRA
NIM. 16650079**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN

**IMPLEMENTASI SISTEM *MONITORING* DAN PENYIRAMAN
TANAMAN OTOMATIS BERBASIS *INTERNET OF THINGS*
MENGUNAKAN *K-NEAREST NEIGHBOUR***

SKRIPSI

Oleh :

RIZQI ARI PUTRA
NIM. 16650079

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : 2020

Dosen Pembimbing I

Dosen Pembimbing II

Fresy Nugroho, M.T
NIP. 19710722 201101 1 001

Fajar Rohman H, M.Kom
NIP. 19890515 201801 1 001

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN
IMPLEMENTASI SISTEM *MONITORING* DAN PENYIRAMAN
TANAMAN OTOMATIS BERBASIS *INTERNET OF THINGS*
MENGGUNAKAN *K-NEAREST NEIGHBOUR*

SKRIPSI

Oleh:
Rizqi Ari Putra
NIM. 16650079

Telah Dipertahankan di Depan Penguji Skripsi dan Dinyatakan Diterima Sebagai
Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)
Tanggal 2020

Susunan Dewan Penguji

| | | Tanda Tangan |
|------------------|---|---------------------|
| 1. Penguji Utama | : <u>Hani Nurhayati, M.T</u> NIP. 19780625 200801 2 006 | () |
| 2. Ketua | : <u>Ajib Hanani, M.T</u> NIDT. 19840731 20160801 1 076 | () |
| 3. Sekretaris | : <u>Fresy Nugroho, M. T</u> NIP. 19710722 201101 1 001 | () |
| 4. Anggota | : <u>Fajar Rohman Hariri, M.Kom</u> NIP. 19890515 201801 1 001 | () |

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crys dian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Rizqi Ari Putra

NIM : 16650079

Fakultas/ Jurusan : Sains dan Teknologi/ Teknik Infomatika

Judul Skripsi : Implementasi Sistem *Monitoring* Dan Penyiraman
Tanaman Otomatis Berbasis *Internet Of Things*
Menggunakan *K-Nearest Neighbour*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 19 Juni 2020

Yang membuat pernyataan,



Rizqi Ari Putra
NIM. 116650079

HALAMAN MOTTO



HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Puji syukur kehadiran Allah, shalawat dan salam bagi Rasul-Nya

Penulis persembahkan sebuah karya ini kepada:

Kedua orang tua penulis tercinta, Bapak Ashari dan Ibu Siti May Saroh yang selalu memberikan suntikan motivasi yang tak terhingga.

Dosen pembimbing penulis Bapak Fresy Nugroho, M. T dan Bapak Fajar Rohman Hariri, M. Kom yang telah dengan sabar membimbing jalannya penelitian skripsi ini dan selalu memberikan stimulus positif untuk tetap semangat menjalani setiap tahap ujian skripsi.

Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, dan seluruh guru-guru penulis yang telah membimbing dan memberikan ilmunya yang sangat bermanfaat.

Teman-teman organisasi selama di UIN Maulana Malik Ibrahim Malang yang telah mengajarkanku berbagai hal untuk terus belajar semakin baik.

Keluarga Teknik Informatika 2016 “Andromeda” yang telah memberikan semangat dan doanya

Orang-orang yang penulis sayangi, yang tak bisa penulis sebutkan satu per satu yang selalu memberikan semangat dan motivasinya kepada penulis untuk menyelesaikan skripsi ini.

Penulis ucapkan terima kasih yang luar biasa. Semoga *ukhwah* kita tetap terjaga dan selalu mendapat *ridho* Allah SWT. *Allahumma Aamiin.*

KATA PENGANTAR

Bissmillahirrohmanirrohim

Alhamdulillah puji syukur kehadiran Allah SWT yang masih melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang menjadi salah satu syarat untuk menyelesaikan program studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Dengan segala kerendahan hati, penulis menyadari bahwa dalam menyelesaikan skripsi ini tidak lepas dari peran berbagai pihak yang telah memberi bantuan, dorongan dan semangat. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Abdul Haris, M.Ag, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Ibu Dr. Sri Harini, M. Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak M. Imamudin Lc, MA, selaku Dosen Wali yang senantiasa memberikan bimbingan dan arahan.
5. Bapak Fresy Nugroho, M. T, selaku Dosen Pembimbing I yang telah banyak memberikan arahan, waktu, dan kontribusi dalam penyelesaian skripsi ini.
6. Bapak Fajar Rohman Hariri, M. Kom, selaku Dosen Pembimbing II yang telah banyak memberikan arahan, waktu, dan kontribusi dalam penyelesaian skripsi ini.
7. Segenap sivitas akademika Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
8. Mbak Citra Fidya Atmalia, S.H selaku admin Jurusan Teknik Informatika yang selalu sabar dan tidak pernah lelah dalam membantu penulis dalam menyelesaikan permasalahan terkait administrasi jurusan.
9. Bapak, Ibu, Adik dan keluarga tercinta yang senantiasa memberikan do'a, restu serta semangat kepada penulis untuk menuntut ilmu.

10. Teman-teman Teknik Informatika angkatan 2016 “Andromeda”, yang selalu memberikan dorongan, semangat dan bantuannya dalam menyelesaikan skripsi ini.
11. Para *Virtual Liver* Nijisanji ID, khususnya Miyu Ottavia yang telah menghibur dan mengisi kekosongan waktu penulis melalui konten-konten dan *joke* recehnya.
12. Semua pihak yang turut membantu dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap skripsi ini dapat memberikan manfaat kepada para pembaca dan khususnya penulis sendiri. *Aamiin.*

Malang, 18 Juni 2020

Penulis

Rizqi Ari Putra

DAFTAR ISI

| | |
|---|-------------|
| HALAMAN PENGAJUAN | i |
| LEMBAR PERSETUJUAN | ii |
| LEMBAR PENGESAHAN | iii |
| PERNYATAAN KEASLIAN TULISAN | iv |
| HALAMAN MOTTO | v |
| HALAMAN PERSEMBAHAN | vi |
| KATA PENGANTAR | vii |
| DAFTAR ISI | ix |
| DAFTAR TABEL | xii |
| DAFTAR GAMBAR | xiii |
| ABSTRAK | xv |
| ABSTRACT | xvi |
| المخلص | xvii |
| BAB I | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Identifikasi Masalah | 4 |
| 1.3 Tujuan Penelitian..... | 4 |
| 1.4 Batasan Masalah..... | 4 |
| 1.5 Manfaat..... | 5 |
| BAB II | 6 |
| 2.1 Penelitian Terdahulu..... | 6 |
| 2.2 Dasar Teori | 8 |
| 2.2.1 <i>Strawberry</i> | 8 |
| 2.2.2 <i>Internet of Things</i> | 9 |
| 2.2.3 <i>K-Nearest Neighbor</i> | 11 |
| 2.2.4 <i>Raspberry Pi</i> | 12 |
| 2.2.5 <i>Sensor BME280</i> | 14 |
| 2.2.6 <i>OLED Display 0.96in</i> | 15 |
| 2.2.7 <i>Raspberry Camera Module 2</i> | 17 |
| 2.2.8 <i>Sensor Ultrasonic HC SR-04</i> | 18 |

| | | |
|----------------|--|-----------|
| 2.2.9 | NodeMCU ESP8266 | 19 |
| 2.2.10 | <i>Soil Moisture Sensor</i> | 20 |
| 2.2.11 | <i>LDR (Light Dependent Resistor)/ Sensor Cahaya</i> | 20 |
| 2.2.12 | <i>Analog Multiplexer</i> | 21 |
| 2.2.13 | Sensor Hujan | 22 |
| 2.2.14 | Relay | 23 |
| BAB III | | 24 |
| 3.1 | Desain Sistem | 24 |
| 3.1.1 | <i>Hardware System</i> | 25 |
| 3.1.1.1 | Raspberry Pi..... | 27 |
| 3.1.1.2 | <i>Camera Module</i> | 27 |
| 3.1.1.3 | <i>OLED Display</i> | 27 |
| 3.1.1.4 | NodeMCU..... | 27 |
| 3.1.1.5 | HC SR-04..... | 28 |
| 3.1.1.6 | <i>Multiplexer</i> | 30 |
| 3.1.1.7 | Relay | 31 |
| 3.1.2 | <i>Software System</i> | 31 |
| 3.1.2.1 | Sistem Operasi | 31 |
| 3.1.2.2 | MySQL Server..... | 32 |
| 3.1.2.3 | <i>Web Server</i> | 32 |
| 3.1.2.4 | <i>REST Server</i> | 32 |
| 3.1.3 | Pembacaan Sensor..... | 33 |
| 3.1.4 | Metode KNN..... | 33 |
| 3.1.5 | Desain Tampilan Antarmuka | 37 |
| 3.2 | Desain Jaringan | 38 |
| 3.3 | Rencana Uji Coba..... | 39 |
| BAB IV | | 42 |
| 4.1 | Hasil dan Uji Coba | 42 |
| 4.1.1 | Uji coba sensor..... | 42 |
| 4.1.2 | Pengujian Pompa Air | 45 |
| 4.1.3 | Metode <i>K-Nearest Neighbor</i> | 46 |
| 4.1.4 | Pengujian metode <i>K-Nearest Neighbor</i> | 53 |

| | | |
|-----------------------|--|-----------|
| 4.2 | Implementasi Antarmuka | 62 |
| 4.2.1 | Halaman Dashboard | 63 |
| 4.2.2 | Halaman <i>Camera</i> | 65 |
| 4.2.3 | Halaman <i>K-Nearest Neighborhod</i> | 65 |
| 4.2.4 | Menu <i>Setting</i> | 68 |
| 4.3 | Implementasi REST Service..... | 69 |
| 4.4 | Implementasi Hardware..... | 72 |
| 4.4.1 | Server (Raspberry Pi 3 B+)..... | 72 |
| 4.4.2 | Sensor dan <i>Watering Node</i> | 75 |
| 4.5 | Integrasi Islam | 77 |
| BAB V | | 79 |
| 5.1 | Kesimpulan..... | 79 |
| 5.2 | Saran..... | 80 |
| DAFTAR PUSTAKA | | 81 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Spesifikasi Raspberry Pi 3 B+ | 13 |
| Tabel 2.2 Spesifikasi Utama OLED Display 0.96in | 16 |
| Tabel 2.3 Spesifikasi Utama Camera Module V2..... | 17 |
| Tabel 3.1 Tabel kebenaran mux 8 <i>channel</i> | 31 |
| Tabel 3.2 Contoh Data Sensor | 33 |
| Tabel 3.3 Contoh <i>dataset</i> | 35 |
| Tabel 3.4 Tabel hasil perhitungan jarak..... | 36 |
| Tabel 3.5 Tabel hasil <i>sorting</i> | 37 |
| Tabel 3.6 Tetangga terdekat..... | 37 |
| Tabel 3.7 Tabel Kalibrasi Sensor | 40 |
| Tabel 3.8 Tabel <i>Confusion Matrix 2 Class</i> | 40 |
| Tabel 4.1 Tabel uji akurasi sensor <i>ultrasonic</i> SR-HC 04..... | 42 |
| Tabel 4.2 Tabel uji akurasi sensor BME-280..... | 43 |
| Tabel 4.3 Tabel uji akurasi sensor Soil Moisture..... | 44 |
| Tabel 4.4 Tabel laju pompa air 5V..... | 46 |
| Tabel 4.5 Tabel label kelas klasifikasi | 47 |
| Tabel 4.6 Tabel <i>Dataset</i> | 48 |
| Tabel 4.7 Tabel <i>Sort Euclidean Distance</i> | 51 |
| Tabel 4.8 Tabel Tetangga Terdekat | 52 |
| Tabel 4.9 Tabel Data Latih..... | 53 |
| Tabel 4.10 Tabel Data <i>Test</i> | 56 |
| Tabel 4.11 Tabel Perbandingan Hasil Prediksi dan Klasifikasi Aktual..... | 58 |
| Tabel 4.12 Tabel Confussion Matrix Data Uji coba | 60 |
| Tabel 4.13 Tabel Daftar REST <i>method</i> | 69 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Konsep <i>internet of things</i> | 11 |
| Gambar 2.2 Raspberry Pi 3B+ | 14 |
| Gambar 2.3 BME280 Sensor | 15 |
| Gambar 2.4 Perbandingan Teknologi OLED dan LCD | 16 |
| Gambar 2.5 OLED Display 0.96in..... | 17 |
| Gambar 2.6 Camera Module V2 | 18 |
| Gambar 2.7 Cara Kerja sensor ultrasonik | 18 |
| Gambar 2.8 NodeMCU V1 (kiri) dan NodeMCU V1 (kanan) | 19 |
| Gambar 2.9 NodeMCU V3 | 19 |
| Gambar 2.10 Bentuk dan <i>pin out soil moisture sensor</i> | 20 |
| Gambar 2.11 Bentuk LDR dan simbolnya | 21 |
| Gambar 2.12 Multiplexer 4 channel..... | 21 |
| Gambar 2.13 Pin out IC CD4051 | 22 |
| Gambar 2.14 Bentuk dan <i>pin out</i> sensor hujan | 22 |
| Gambar 2.15 <i>Relay module 2 channel</i> | 23 |
| Gambar 3.1 Diagram Desain Sistem..... | 24 |
| Gambar 3.2 <i>Wiring Diagram</i> Raspberry Pi sebagai <i>server</i> | 25 |
| Gambar 3.3 <i>Wiring Diagram</i> NodeMCU <i>Watering Node</i> | 26 |
| Gambar 3.4 <i>Wiring Diagram</i> NodeMCU <i>Sensor Node</i> | 26 |
| Gambar 3.5 Contoh perhitungan kapasitas air | 29 |
| Gambar 3.6 <i>Flowchart</i> proses klasifikasi dengan metode KNN..... | 34 |
| Gambar 3.7 Tampilan <i>Prototype web interface</i> | 38 |
| Gambar 3.8 Desain Jaringan | 39 |
| Gambar 4.1 <i>Scatter Plot</i> persebaran <i>dataset</i> | 49 |
| Gambar 4.2 <i>Scatter Plot</i> persebaran baru | 52 |
| Gambar 4.3 Halaman <i>Dashboard</i> tampilan pada komputer..... | 63 |
| Gambar 4.4 Halaman <i>Dashboard</i> tampilan pada tablet dan <i>mobile</i> | 64 |
| Gambar 4.5 Halaman <i>camera</i> | 65 |
| Gambar 4.6 Halaman <i>Calculate KNN</i> | 66 |
| Gambar 4.7 Halaman <i>Dataset</i> | 66 |

| | |
|---|----|
| Gambar 4. 8 Halaman Label | 67 |
| Gambar 4.9 Modal Tambah Data..... | 67 |
| Gambar 4.10 Modal Edit Data | 67 |
| Gambar 4.11 Modal <i>Camera Setting</i> | 68 |
| Gambar 4.12 Modal <i>Water Level Setting</i> | 68 |
| Gambar 4.13 Modal <i>Watering Control Setting</i> | 68 |
| Gambar 4.14 konfigurasi <i>Port Forwarding</i> pada <i>router</i> | 73 |
| Gambar 4.15 Konfigurasi <i>Hostname</i> di halaman <i>no-ip.com</i> | 73 |
| Gambar 4.16 Raspberry Pi tampak depan..... | 74 |
| Gambar 4.17 Raspberry Pi tampak dalam..... | 74 |
| Gambar 4.18 rangkaian NodeMCU tampak dalam..... | 75 |
| Gambar 4.19 rangkaian NodeMCU tampak samping | 76 |
| Gambar 4.20 WiFi Portal NodeMCU ESP8266 | 76 |

ABSTRAK

Ari Putra, Rizqi. 2020. **Implementasi Sistem Monitoring Dan Penyiraman Tanaman Otomatis Berbasis *Internet Of Things* Menggunakan *K-Nearest Neighbour***. Skripsi Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing: (I) Fresy Nugroho, M.T (II) Fajar Rohman Hariri, M. Kom

Kata Kunci: *Internet of Things*, *K-Nearest Neighbor*, Klasifikasi, *Supervised Learning*

Strawberry merupakan salah satu komoditas buah-buahan yang dibudidayakan di Indonesia. Budidaya *strawberry* tidaklah mudah, petani harus memperhatikan penyiraman secara teratur dan tidak bisa sembarangan. Hal itu karena *strawberry* tidak suka tanah yang terlalu basah atau kering. Hal tersebut sangat merepotkan petani karena petani harus selalu melihat kondisi tanah secara manual sebelum melakukan penyiraman.

Sistem *monitoring* dan penyiraman tanaman berbasis *internet of things* merupakan sistem yang dibangun untuk membantu petani dalam merawat *strawberry*. Sistem ini memantau kondisi tanah, suhu dan kelembaban secara *real-time* sehingga memudahkan bagi petani. Sistem ini juga dilengkapi penyiraman otomatis menggunakan metode *K-Nearest Neighbor*. Untuk melakukan penyiraman otomatis dibutuhkan parameter berupa suhu dan kelembaban. Kedua parameter tersebut akan diklasifikasi menggunakan metode *K-Nearest Neighbor* sehingga diketahui kondisi tanahnya. Dari kondisi tanah tersebut sistem akan menentukan intensitas air penyiraman yang diperlukan.

Sistem sudah berhasil dibangun dan dapat diakses melalui jaringan lokal maupun internet. Menggunakan antarmuka berupa web, pengguna bisa melihat data sensor secara *real-time* dan dapat juga melakukan penyiraman tanpa harus pergi ke kebun. Dalam pengujian metode *K-Nearest Neighbor* diperoleh nilai *macro precision* sebesar 94,05%, *macro recall* 63,33% dan *macro f1* 67,7%.

ABSTRACT

Ari Putra, Rizqi. 2020. **Implementation Of Internet Of Things-Based Monitoring and Automatic Plant Watering System Using K-Nearest Neighbor**. Undergraduate Thesis. Department of Informatics Engineering, Faculty of Science and Technology. State Islamic University of Maulana Malik Ibrahim Malang.

Advisors: (I) Fresy Nugroho, M.T (II) Fajar Rohman Hariri, M. Kom

Keywords: Classification, Internet of Things, K-Nearest Neighbor, Supervised Learning

Strawberry is one of the fruit commodities cultivated in Indonesia. Strawberry cultivation is not easy, farmers must pay attention to regular watering and cannot be careless. That's because strawberries don't like too wet or dry soil. This is very troublesome for farmers because they have to look at the condition of the soil manually every time before watering.

An internet of things-based *monitoring* and plants watering system is a system that was built to assist farmers in caring for strawberries. This system monitors soil conditions, temperature and humidity in real-time making it easy for farmers. This system is also equipped with automatic watering using the K-Nearest Neighbor method. To do automatic watering, it needs parameters such as temperature and humidity. Both parameters will be classified using the K-Nearest Neighbor method so that the soil conditions are known. From the soil conditions, the system will determine the intensity of the required watering.

The system has been successfully built and can be accessed via the local network or the internet. Using a web interface, users can see sensor data in real-time and can also do watering without having to go to the garden. In testing the K-Nearest Neighbor method, the value of macro precision is 94.05%, macro recall 63.33% and macro f1 67.7%.

الملخص

أري بوترا، رزقي. ٢٠٢٠. تنفيذ شبكة الإنترنت للرصد القائم على الأشياء ونظام ري النبات التلقائي باستخدام K-Nearest Neighbor. قسم هندسة المعلوماتية لكلية العلوم والتكنولوجيا في جامعة مولانا مالك إبراهيم الإسلامية الحكومية بمالانق. المشرف : (١) فريسي نوغراها، الماجستير. (٢) فجر رحمن حريري ، ماجستير علوم الحاسوب.

الكلمات الرئيسية : تصنيف ، إنترنت الأشياء ، تعليم تحت الإشراف ، K-Nearest Neighbor.

والفراولة هي إحدى السلع الأساسية للفواكه التي تزرع في إندونيسيا. فزراعة الفراولة ليست سهلة ، وينبغي للمزارعين أن ينتبهوا إلى الري المنتظم ولا يمكن أن تكون تعسفية. لأن الفراولة لا تحب التربة الرطبة أو الجافة وهو أمر مزعج جدا للمزارعين لأن المزارعين يجب أن ينظروا دائما إلى حالة التربة يدويا قبل أن يقوموا بالسقي.

نظام رصد وسقي النباتات على أساس الإنترنت للأشياء هو نظام تم بناؤه لمساعدة المزارعين في رعاية الفراولة. وترصد هذه النظم حالة التربة ، ودرجة الحرارة والرطوبة في الوقت الحقيقي مما يسهل على المزارعين القيام بذلك. كما أن النظام مجهز بالسقي الآلي باستخدام أسلوب K-Nearest Neighbor. لأداء سقي تلقائي البارامترات المطلوبة مثل درجة الحرارة والرطوبة. من هذه البارامترات سيتم تصنيفها باستخدام K-Nearest Neighbor حتى أن الحالة المعروفة من التربة. من الظروف الأرضية النظام يحدد كثافة الماء المطلوب.

وقد بني هذا النظام بنجاح ويمكن الوصول إليه عن طريق الشبكة المحلية أو الإنترنت. باستخدام واجهة على شكل شبكة الإنترنت, يمكن للمستخدمين عرض بيانات الاستشعار في الوقت الحقيقي ويمكن أيضا القيام الري دون الحاجة إلى الذهاب إلى الحديقة. في اختبار K-Nearest Neighbor حصلت على قيمة macro precision ٩٤.٠٥% ، macro recall هو ٦٣.٣% و macro f1 ٦٧.٧%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi sudah banyak mengalami perkembangan yang pesat baik dari teknologi informasi, komputer maupun elektronika digital. Hal tersebut terbukti dengan maraknya inovasi baru yang mempengaruhi gaya hidup dan pandangan masyarakat. Teknologi juga memiliki peranan yang tak terpisahkan dalam membangun dan memudahkan masyarakat. Contohnya adalah perkembangan internet sebagai teknologi komunikasi dan informasi dapat mempermudah masyarakat dalam berkomunikasi dan memperoleh pengetahuan atau informasi tanpa terbatas pada jarak, waktu maupun tempat. Selain itu dengan perkembangan teknologi yang ada mampu meningkatkan kuantitas dan kualitas di berbagai sektor, misal pendidikan, kesehatan, industri, pertanian dan lain-lain.

Sebagai negara beriklim tropis dengan segala kekayaan alamnya, Indonesia memiliki potensi pada sektor pertanian. Kekayaan sumber daya alam dan keragaman plasma nutfah (*bio diversity*) di Indonesia merupakan terbesar kedua setelah Brasil. Dengan kekayaan jenis tanaman tersebut mampu memberikan komoditas atau produk pertanian yang unggul. Produk pertanian segar seperti sayur dan buah merupakan salah satu produk pertanian yang sangat berpotensi. Selain produk buah dan sayur potensi utama lainnya adalah tanaman rempah-rempah dan nabati. Produk pertanian pokok pangan juga banyak menghasilkan produk unggulan seperti padi, jagung, kedelai, umbi-umbian, kedelai dan segala varietas lainnya.

Sektor pertanian menjadi salah satu tulang punggung ekonomi di Indonesia dan memiliki peranan signifikan. Hal tersebut dibuktikan dengan adanya data dari Badan Pusat Statistik 2012 yang menyebutkan bahwa setidaknya ada 35.9% angkatan kerja Indonesia yang masuk pada sektor pertanian dan menyumbang 14,7% bagi GNP Indonesia. Sehingga menguatkan bahwa sektor pertanian memiliki peranan vital bagi perekonomian di Indonesia.

Seiring dengan perkembangan ilmu pengetahuan, *strawberry* mendapat perhatian pengembangannya di daerah beriklim tropis seperti Indonesia. Hal tersebut dikarenakan *strawberry* memiliki nilai ekonomi dan banyak manfaat. Adapun kandungan dari *strawberry* adalah kalori, protein, vitamin A dan C dan karbohidrat. *Strawberry* sendiri dapat dikonsumsi secara segar maupun sebagai produk olah, misal selai dan ekstrak buah.

Kota Batu merupakan salah satu kota di Indonesia yang melakukan budidaya *strawberry*. Di kota Batu sendiri tidak sedikit para petani membudidayakan *strawberry* baik di kebun maupun di perkerangan rumah. Perawatan *strawberry* cukup rumit karena memerlukan udara dan suhu yang dingin dan sejuk. Dalam hal penyiraman *strawberry* tidak bisa sembarangan karena *strawberry* tidak suka tanah yang becek dan kering. Sehingga hal tersebut harus diperhatikan oleh petani. Pada umumnya petani harus melihat tanah secara langsung dengan tangan untuk memastikan apakah tanahnya sudah sesuai apa belum sebelum menyirami. Tentunya hal tersebut menjadi masalah yang sangat merepotkan petani karena harus melakukan pemantauan kondisi tanah dan melakukan penyiraman untuk mengontrol keadaan tanah. Hal tersebut menunjukkan bahwa petani belum mampu memanfaatkan sumber daya dan teknologi secara maksimal dalam proses bertani.

Teknologi otomasi sudah sangat berkembang pesat. teknologi otomasi memiliki pengaruh yang sangat baik, terutama pada sektor pertanian di Indonesia. Dengan adanya teknologi otomasi ini akan semakin mempermudah proses bertani. Teknologi otomasi mampu mengurangi tenaga *labor* manusia mulai dari proses menanam, merawat hingga menjadi produk siap pakai. Untuk membangun sistem otomasi yang cerdas membutuhkan *machine learning* di dalamnya. *Machine learning* sendiri merupakan bagian dari kecerdasan buatan. *Machine learning* memiliki fokus pada pengembangan sistem yang mampu belajar sendiri tanpa harus di program oleh manusia. Dalam perancangan *machine learning* diperlukan data *training* (pembelajaran) sebelum mengeluarkan *output*.

K-Nearest Neighbor (KNN) merupakan salah satu metode *machine learning* bersifat *supervised learning* yang digunakan untuk menentukan

klasifikasi. Metode ini bekerja dengan mengklasifikasi kriteria baru sesuai dengan mayoritas kedekatan jarak dari kriteria target. Kelebihan dari metode ini adalah bersifat non paramatik (Putri, Suparti, & Rahmawati, 2014). Non parametrik merupakan model matematika yang tidak mengasumsikan apa-apa mengenai distribusi *instance* di dalam dataset. Salah satu kelebihan dari model non parametrik yaitu menghasilkan garis keputusan kelas yang sangat fleksibel dan tidak linear, namun model ini biasanya lebih sulit untuk diinterpretasikan. Kelebihan lainnya dari metode ini adalah sifatnya yang sederhana, di mana untuk melakukan klasifikasi berdasarkan jarak terpendek dari *testing sample* ke *training sample*. Metode ini juga memiliki kelebihan yang tangguh terhadap *training data* yang *noise* dan efektif apabila data latihnya besar (Eliyen, Tolle, & Muslim, 2017). Dalam penelitian yang dilakukan oleh Abraham Sudharson Ponraj pada tahun 2019, metode KNN memiliki tingkat akurasi hingga 94% pada manajemen tanah yang digunakan untuk melakukan prediksi kekeringan tanah.

Untuk membantu melakukan pemantauan dan pengambilan data jarak jauh dapat dilakukan dengan teknologi *Internet of Things*. Konsep dasar teknologi *internet of things* adalah menghubungkan objek/ benda ke dalam sebuah jaringan internet baik publik maupun lokal. Objek-objek tersebut memiliki kemampuan untuk merasakan, berkomunikasi dan berbagi informasi di dalamnya. Selanjutnya objek-objek tersebut akan mengumpulkan data secara teratur, menganalisisnya dan menggunakannya sebagai *trigger* suatu *action*. Ada tiga kategori utama *internet of things*, yaitu *people to people*, *people to machine* dan *machine to machine*.

Dari paparan dan masalah yang sudah disebutkan, peneliti memiliki ide untuk mengatasi permasalahan petani tentang pemantauan tanah dan penyiraman tanaman secara otomatis dengan membangun penyiram otomatis berbasis *internet of things* dengan mengimplementasikan metode *K-Nearest Neighborhood*. Untuk melakukan penyiraman secara otomatis ada dua parameter yang dibutuhkan, yaitu kelembaban tanah dan suhu. Dari dua parameter tersebut akan diklasifikasikan dengan metode *K-Nearest Neighborhood* untuk mengidentifikasi apakah tanah dalam keadaan kering

maupun basah. Dari data klasifikasi sistem akan menentukan seberapa banyak volume air yang diperlukan untuk melakukan penyiraman. Data kelembapan dan suhu juga akan disimpan di *server* sebagai data *training* dan akan ditampilkan sehingga dapat dilakukan *monitoring* oleh petani.

1.2 Identifikasi Masalah

Berdasarkan paparan dari latar belakang masalah, dapat diidentifikasi permasalahan yaitu bagaimana membangun sistem penyiraman otomatis berbasis *Internet of Things* menggunakan parameter suhu dan kelembapan dengan metode *K-Nearest Neighbor* ?

1.3 Tujuan Penelitian

Merujuk pada identifikasi masalah, maka penelitian diharapkan dapat mencapai tujuan yaitu untuk membangun sistem yang dapat melakukan *monitoring* dan penyiraman otomatis berbasis *Internet of Things* menggunakan parameter suhu dan kelembapan dengan metode *K-Nearest Neighbor*.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah:

1. Tanaman yang diamati dijadikan penelitian adalah tanaman *strawberry* dengan penanaman dalam pot.
2. Digunakan *Raspberry Pi 3 B+* sebagai *web server*, *database* dan pengolah data.
3. Data suhu dan kelembapan udara diambil menggunakan sensor *BME28*, sedangkan untuk mengambil kelembapan tanah digunakan sensor *soil moisture*.
4. Mikrokontroler *NodeMCU ESP8266* digunakan untuk mengambil data dari sensor dan mengirimkannya ke server.
5. Jaringan *WiFi* digunakan untuk berkomunikasi antara *NodeMCU* dan *Raspberry PI*.

6. Digunakan *Pi Camera module* untuk melakukan *streaming* pemantauan pada tanaman.
7. Digunakan antarmuka berupa web untuk melakukan pemantauan pada tumbuhan secara *real-time*.

1.5 Manfaat

Secara teoritis manfaat dari penelitian ini adalah sebagai sarana penulis untuk mengaplikasikan teori dan praktikum tentang kecerdasan buatan dengan mengimplementasikan terhadap perangkat berbasis *Internet of Thing*. Di samping itu dengan adanya penelitian ini diharapkan mampu menjadi pendorong berkembangnya teknologi pertanian cerdas. Secara praktis manfaat dari penelitian ini adalah untuk memudahkan masyarakat, khususnya pertanian *strawberry* untuk melakukan pemantauan dan penyiraman otomatis pada tanaman mereka. Sehingga pada akhirnya bias didapatkan hasil panen yang berkualitas secara kuantitas dan kualitas.

BAB II KAJIAN PUSTAKA

2.1 Penelitian Terdahulu

Abraham Sudharson Ponraj pada tahun 2019 melakukan penelitian mengenai pendekatan *machine learning* untuk *internet of things* pertanian. Dalam penelitiannya dilakukan perbandingan beberapa *metode machine learning*. Kemudian dari analisa tersebut dilakukan kategorisasi. Kategori ini menjelaskan tentang kemampuan metode *machine learning* dalam menjalankan fungsinya, setidaknya ada lima kategori. Adapun kategori-kategori yang dimaksud adalah:

1. Manajemen lahan dan hasil
2. Manajemen Tanah
3. Manajemen Penyakit
4. Deteksi Gulma
5. Manajemen Air

Pada manajemen tanah ditemukan bahwa *K-Nearest Neighbor* memiliki akurasi sebesar 94% dalam melakukan prediksi terhadap penyerapan dan penguapan air.

A. Neelima dkk pada tahun 2018 melakukan penelitian mengenai sistem irigasi *internet of Things* berbasis sensor otomatis. Dalam perancangannya digunakan sensor kelembaban tanah dan arduino. Sedangkan untuk terkoneksi ke internet dibutuhkan *module* wifi. Data dari pembacaan sensor tersebut kemudian dikirim ke platform *internet of things* bernama thingspeak. Otomasi penyiraman diolah di Arduino kemudian status penyiraman dikirimkan ke thingspeak. Kelemahan dari sistem ini adalah otomasi masih bersifat statis dan belum melibatkan proses kecerdasan di dalamnya.

Pavankumar Naik dkk pada tahun 2018 melakukan penelitian mengenai penyiraman otomatis berbasis internet of things menggunakan Arduino Yun sebagai otaknya. Pada penelitian tersebut digunakan sensor kelembaban udara, sensor suhu dan sensor kelembaban tanah. Data sensor tersebut dikirim dan disimpan di Thingspeak Cloud Server. Data yang sudah tersimpan tadi

disajikan dalam bentuk aplikasi Android. Pada aplikasi tersebut menampilkan status pompa air, suhu, kelembaban udara dan tanah. Pada penelitian tersebut penyiraman bersifat statis dimana akan dilakukan penyiraman jika kelembaban di bawah 30%.

Srishi Rawal pada tahun 2017 membangun sebuah penyiraman otomatis berbasis *internet of things*. Dalam penelitiannya tersebut digunakan Arduino UNO sebagai basisnya. Arduino tersebut dikombinasikan dengan modul GSM/GPRS SIM900A untuk terhubung dengan internet. Pada penelitian tersebut untuk melakukan penyiraman otomatis digunakan sensor kelembaban tanah. Dari pembacaan sensor tersebut akan dikirimkan ke Thingspeak Cloud Server untuk disimpan dan ditampilkan dalam bentuk graf. Pada penelitian tersebut juga disediakan antarmuka berupa web untuk melihat status penyiraman dan sensor kelembaban tanah. Namun, pada sistem yang dibangun penyiraman tanah bersifat statis. Penyiraman akan dilakukan jika kelembaban tanah berada pada level 79% dan berhenti melakukan penyiraman pada 100%.

Pada tahun 2017 Yuthika Shekhar dkk telah melakukan penelitian dalam makalahnya yang berjudul *Intelligent IoT Based Automated Irrigation System*. Dalam penelitiannya dikembangkan sistem penyiraman otomatis berbasis iot menggunakan metode *K Nearest Neighborhood*. Metode tersebut digunakan untuk mengklasifikasikan kondisi tanah menjadi empat kategori, yaitu basah, sedikit basah, kering, sedikit kering. Parameter yang digunakan dalam klasifikasinya adalah kelembaban tanah dan suhu tanah. Penelitian ini menerapkan *internet of thing* berbasis *machine to machine*. Sensor suhu dan kelembaban dihubungkan dengan papan pengembangan arduino dengan menghubungkan secara serial menggunakan kabel usb ke raspberry pi. Data sensor tadi kemudian akan di olah dengan *K Nearest Neighborhood* di dalam Raspberry PI. Data yang telah diambil dan klasifikasi kemudian disimpan ke dalam bentuk CSV. Kemudian data hasil klasifikasi ditampilkan dalam bentuk *scartered* plot melalui tampilan web. Namun dalam sistemnya belum dapat melakukan pemantauan kondisi tanah, karena data yang ditampilkan

berupa hasil klasifikasi. Selain itu penelitian ini belum mampu menentukan intensitas penyiraman.

Pada penelitian ini peneliti akan mengimplementasikan metode *K-Nearest Neighbor* untuk *monitoring* dan penyiraman tumbuhan otomatis berbasis *internet of things*. Dengan menerapkan metode tersebut diharapkan dapat melakukan penyiraman tanpa khawatir tumbuhan kelebihan atau kekurangan air. Selain itu sistem yang dibangun juga dilengkapi mode manual untuk mengatasi jika terdapat masalah tidak terduga. Sistem yang dibangun juga dilengkapi dengan *interface* berbasis web yang menampilkan data sensor secara *real-time*. Selain itu pada web *interface* juga disediakan kontrol penyiraman dan juga kamera untuk melakukan *streaming* maupun pengambilan gambar. Untuk akses web *interface* bisa diakses menggunakan internet dengan membuka alamat IP publik sistem di browser baik melalui perangkat komputer maupun *mobile*.

2.2 Dasar Teori

2.2.1 *Strawberry*

Strawberry pertama kali ditemukan di Chili, Amerika. *Strawberry* sendiri merupakan tanaman buah berupa herba. Ada dua jenis spesies *strawberry*, yaitu *Fragaria chiloensis L.* dan *Fragaria vesca L.* Jenis *strawberry* *Fragaria chiloensis L.* tersebar luas ke berbagai negara Amerika, Eropa, dan Asia. Sedangkan spesies selanjutnya *Fragaria vesca L.* lebih menyebar luas dibanding spesies sebelumnya. Jenis *strawberry* inilah yang pertama kali masuk ke Indonesia. (Kementerian Riset dan Teknologi / Badan Riset dan Inovasi Nasional, 2000)

Strawberry yang umum di pasar swalayan merupakan *strawberry* hibrida dari persilangan *Fragaria virginiana L. var Duchesne* asal Amerika Utara dengan *Fragaria chiloensis L. var Duchesne* asal Chili. Dari persilangan tersebut dihasilkan *strawberry* modern *Fragaria x annanassa var Duchesne*. Adapun varietas yang dapat ditanam di Indonesia adalah *Pajero*, *Ostara*, *Selva*, *Tenira*, *Osogrande*, *Bogota*, *Robunda*, *Grella*, *Red Gantlet* dan

Elvira. (Kementerian Riset dan Teknologi / Badan Riset dan Inovasi Nasional, 2000)

Strawberry dapat tumbuh optimal di daerah dengan curah hujan 600-700 mm/ tahun. *Strawberry* juga memerlukan setidaknya 8-10 jam penyinaran cahaya matahari setiap harinya. Kelembaban udara yang baik untuk pertumbuhan *strawberry* antara 80-90%. Sedangkan untuk suhu optimal yang diperlukan adalah 17-20 derajat celcius. (Kementerian Riset dan Teknologi / Badan Riset dan Inovasi Nasional, 2000)

2.2.2 *Internet of Things*

Internet of things atau biasa disingkat IoT merupakan sebuah konsep teknologi yang akhir-akhir ini ramai diperbincangkan. Secara umum *internet of things* dapat diartikan sebagai segala benda yang dapat terhubung dan saling berkomunikasi melalui jaringan internet. Konsep utama dari *internet of things* adalah dengan memperluas manfaat dari konektivitas internet yang terhubung atau tersambung secara terus menerus. (Panduardi & Haq, 2016)

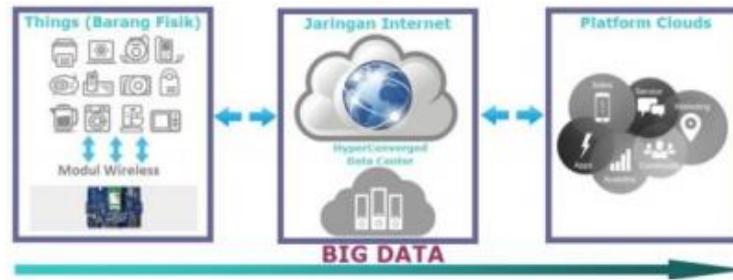
Awal mula perkembangan *internet of things* sendiri dimulai tahun 1989. Pada tahun tersebut internet mulai dikenal dan menjadi awal dari kegiatan daring. Selanjutnya pada tahun 1990, John Romkey melakukan penelitian mengenai perangkat yang dapat dikendalikan melalui internet. Pada penelitian tersebut diciptakan pemanggang roti yang dapat dimatikan dan dihidupkan melalui daring. Istilah *internet of things* sendiri pertama kali disebutkan oleh Kevin Ashton pada tahun 1997. Beliau yang saat itu menjabat sebagai Direktur Eksekutif Auto-IDE juga menciptakan teknologi *RFID* (*Radio Frequency Identification*) pada tahun yang sama. Penemuan tersebut merupakan lompatan dalam teknologi *internet of things*. Kemudian pada tahun 2003 teknologi RFID mulai digunakan oleh militer Amerika Serikat secara besar-besaran. (Junaidi, 2015)

Selanjutnya kelompok perusahaan meluncurkan IPSO Alience pada tahun 2008 untuk mengenalkan dan mempromosikan penggunaan *Internet Protocol (IP)* dalam jaringan dari “*Smart Object*”. Hal tersebut juga digunakan untuk mengaktifkan *internet of things*. Setelah diluncurkannya

IPv6 pada tahun 2011 memicu pertumbuhan besar *internet of things*. Banyak perusahaan besar yang mendukung perkembangan *internet of things*, seperti Cisco, IBM dan Ericson. (Junaidi, 2015)

Dengan adanya perkembangan *internet of things* semua peralatan bisa dikendalikan dan dipantau. Sebagian besar proses dalam *internet of things* dilakukan menggunakan bantuan dari sensor. Sensor yang dipasang tersebut mengubah *raw physical data* menjadi *digital signal* dan mengirimkannya ke pusat kendali. Dengan demikian bisa dilakukan pemantauan perubahan lingkungan secara jarak jauh dari berbagai penjuru melalui internet. Arsitektur sistem *internet of things* sendiri didasarkan pada operasi dan proses yang *real-time*. Pada sistem *home automation* setiap stop kontak akan terhubung dengan *smartphone* (terkadang remote kontrol) sehingga dapat dioperasikan secara jarak jauh. Namun, skenario tersebut tidak membutuhkan perangkat penyimpanan maupun *processor* di setiap saklar stop kontak. Sistem seperti itu hanya membutuhkan sensor untuk menerima sinyal dan kebanyakan hanya untuk melakukan *toggle ON/ OFF*. Jadi sistem pada *internet of things* sangat bervariasi dan tergantung pada kebutuhan implementasinya. (Junaidi, 2015)

Secara sederhana konsep *internet of things* mengacu pada perangkat yang dilengkapi modul *internet of things*, koneksi ke jaringan/ internet dan *cloud data center* sebagai tempat menyimpan aplikasi beserta *database*. *Internet of things* memanfaatkan sebuah argumen pemrograman yang menghasilkan interaksi antara sesama mesin yang terhubung secara otomatis tanpa adanya campur tangan dari pengguna dari mana saja tanpa terikat jarak. Interaksi antara mesin-mesin tersebut dihubungkan oleh internet. Pengguna hanya mengatur dan melakukan pengawasan dari alat apakah bekerja atau tidak. (Efendi, 2018)



Gambar 2.1 Konsep *internet of things*

2.2.3 *K-Nearest Neighbor*

Algoritma *K-Nearest Neighbor* (K-NN) merupakan metode klasifikasi yang bersifat *supervised learning*. Metode ini melakukan klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. Hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam K-NN. Perbedaan antara *unsupervised learning* dengan *supervised learning* adalah pada *unsupervised learning* bertujuan untuk menemukan pola dalam sebuah data dan sebelumnya belum memiliki pola apa pun. Sedangkan pada *supervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. (Krisandi & Helmi, 2013)

Algoritma K-NN memiliki tujuan untuk mengklasifikasi objek baru berdasarkan atribut dan sampel uji. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN. Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apa pun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma K-NN menggunakan klasifikasi ke-tetangga-an/ *neighborhood* sebagai nilai prediksi dari sampel uji yang baru. Jarak yang digunakan adalah jarak *Euclidean Distance*. Jarak Euclidean adalah jarak yang paling umum digunakan pada data numerik. (Krisandi & Helmi, 2013)

Euclidean distance merupakan metode perhitungan jarak dari dua buah titik dalam *Euclidean space*. *Euclidean space* sendiri bisa meliputi bidang euclidean dua dimensi, euclidean tiga dimensi atau lebih. (Nishom, 2019)

Adapun rumus dasar untuk menghitung *euclidean distance* adalah sebagai berikut:

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

dimana:

d = jarak antara x dan y

x = data pusat kluster

y = data pada atribut

i = setiap data

n = jumlah data

x_i = data pada pusat kluster ke i

y_i = data pada setiap data ke i

2.2.4 Raspberry Pi

Raspberry Pi adalah perangkat yang luar biasa. Raspberri merupakan sebuah komputer berukuran kecil dan berfungsi layaknya komputer pada umumnya. Raspberry Pi merupakan papan sirkuit tunggal/ *System on Chip (SoC)* yang di dalamnya terdiri dari CPU, RAM dan I/O. Meskipun berukuran kecil, raspberry dapat melakukan segala hal yang bisa dilakukan komputer yang lebih besar dan haus daya lainnya. Raspberry Pi dapat digunakan untuk bermain *game*, mengerjakan tugas kantor maupun untuk membuat dan menjalankan program. (Raspberrypi.org, 2019)

Raspberry Pi 3 Model B + merupakan jajaran dari variasi Raspberry Pi 3. Model terbaru ini menawarkan prosesor *quad core* 64-bit dan berjalan pada 1.4GHz. Untuk konektivitas model ini memiliki *dual-band* 2.4GHz dan 5GHz LAN nirkabel, Bluetooth 4.2 / BLE, Ethernet lebih cepat, dan PoE kapabilitas melalui HAT PoE terpisah. (Raspberrypi.org, 2019)

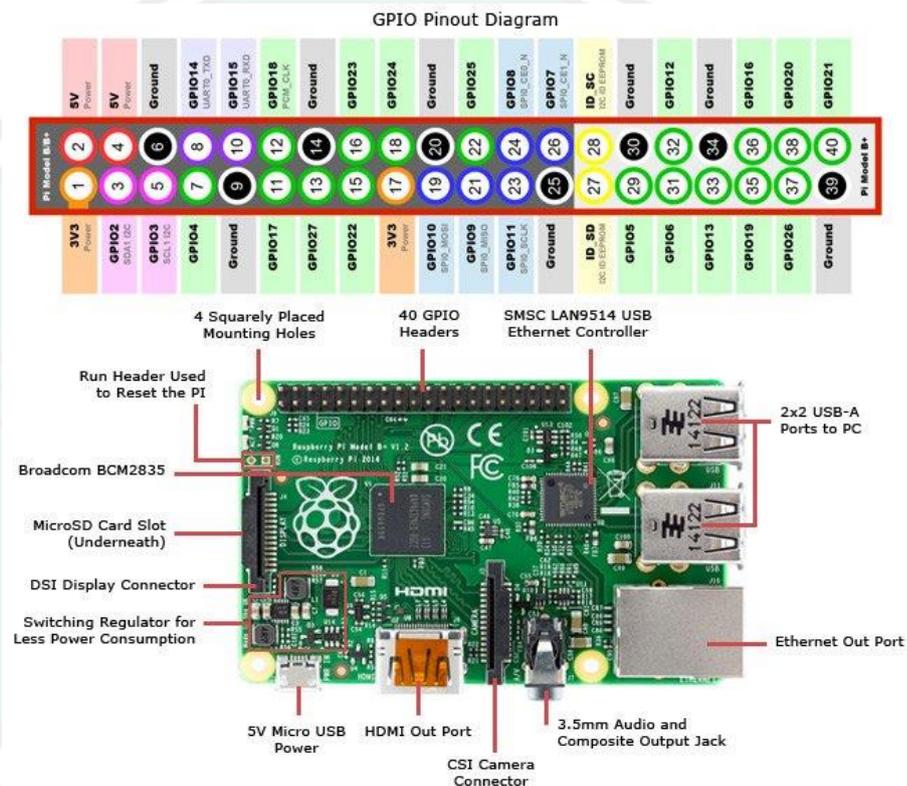
LAN nirkabel *dual-band* hadir dengan sertifikasi kepatuhan modular. Hal tersebut memungkinkan untuk merancang *board* ini menjadi produk jadi dengan mengurangi pengujian pada LAN nirkabel secara signifikan. Hal tersebut mampu meningkatkan biaya dan waktu pemasaran. Raspberry Pi 3

Model B + mempertahankan *footprint* mekanis yang sama seperti Raspberry Pi 2 Model B dan Raspberry Pi 3 Model B. Secara lengkap spesifikasi utama Raspberry Pi 3 Model B + ditunjukkan oleh **Tabel 2.1**. (Raspberrypi.org, 2019)

Tabel 2.1 Spesifikasi Raspberry Pi 3 B+

| Spesifikasi | Keterangan |
|------------------|--|
| Prosesor | Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz |
| Memori | 1GB LPDDR2 SDRAM |
| Konektivitas | 2.4GHz dan 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) 4 × USB 2.0 ports |
| Akses | Extended 40-pin GPIO header |
| Video dan Audio | 1 × full size HDMI MIPI DSI display port MIPI CSI camera port 4 pole stereo output and composite video port |
| Multimedia | H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics |
| Dukungan SD Card | Format micro SD untuk memuat sistem operasi dan penyimpanan data |
| Daya masukan | 5 V/2.5A DC via micro USB connector 5V DC via GPIO header Power over Ethernet (PoE)-enabled (requires separate PoE HAT) |
| Operasional | Operating temperature, 0–50°C |

Raspberry Pi 3 Model B + dilengkapi dengan beberapa pin yang biasa disebut *General Purpose Input Output* atau disingkat GPIO. GPIO dapat diprogram untuk membaca *input* maupun untuk menghasilkan/ mengatur *output*. Pin GPIO pada Raspberry Pi 3 Model B+ dapat digunakan sebagai *input* maupun *output* digital. Selain itu sebagian pin GPIO yang tersedia juga bisa digunakan sebagai antarmuka protokol komunikasi. (Bachrudin, Widodo, & Adi, 2017)



Gambar 2.2 Raspberry Pi 3B+

2.2.5 Sensor BME280

BME280 merupakan sensor yang terintegrasi dengan sensor kelembaban, tekanan udara dan suhu digital berdasarkan prinsip-prinsip penginderaan yang telah terbukti. Modul sensor ini dikemas dalam bentuk paket *Land Grid Array* (LGA) dengan tutup logam yang sangat kompak. Modul sensor ini memiliki *footprint* hanya $2,5 \times 2,5 \text{ mm}^2$ dan ketinggian 0,93 mm. Modul sensor ini dapat diterapkan dalam berbagai perangkat yang ditenagai oleh baterai seperti halnya jam tangan, *GPS module* maupun

handset karena dimensi yang kecil dan konsumsi daya yang rendah. Modul sensor ini memiliki kinerja tinggi di berbagai implementasi yang membutuhkan pengukuran tekanan dan kelembaban. (Bosch Sensortec, 2018)

Modul sensor ini memberikan waktu respons yang sangat cepat untuk aplikasi dengan kesadaran konteks cepat dan akurasi keseluruhan tinggi pada rentang suhu yang luas. Sensor tekanan yang terdapat pada modul ini merupakan sensor tekanan barometrik absolut. Dibandingkan dengan Bosch Sensortec BMP180, sensor tekanan yang terintegrasi ini memiliki akurasi dan resolusi sangat tinggi dan *noise* yang jauh lebih rendah. Begitu halnya sensor suhu yang terintegrasi juga sudah dioptimalkan sehingga menghasilkan *noise* serendah-rendahnya dan resolusi yang tinggi. Sensor ini menyediakan antarmuka berupa *Serial Peripheral Interface* (SPI) dan *Inter-Integrated Circuit* (I2C). Untuk dapat bekerja modul ini memerlukan daya sebesar 1,71V - 3,6V untuk VDD dan 1,2V - 3,6V untuk pasokan *interface* VDDIO. Pengukuran dapat dipicu atau dilakukan secara berkala. Ketika sensor dalam keadaan nonaktifkan, konsumsi akan turun menjadi 0,1 μ A. (Bachrudin, Widodo, & Adi, 2017)

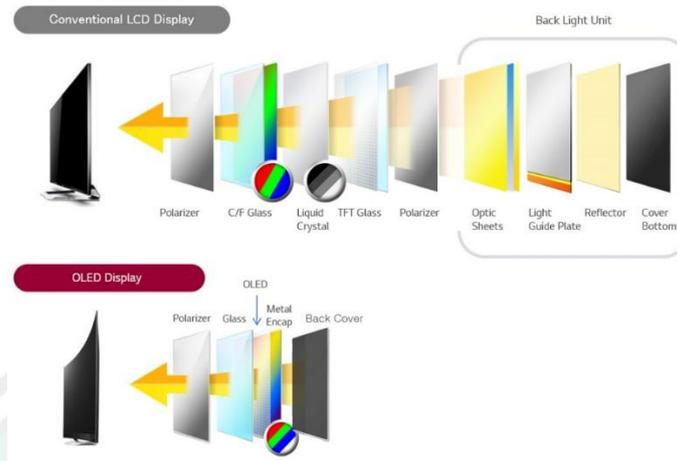


Gambar 2.3 BME280 Sensor

2.2.6 OLED Display 0.96in

OLED (*Organic Light Emitting Diodes*) merupakan salah satu teknologi pemancar cahaya datar. Teknologi ini dibuat dengan menempatkan serangkaian film tipis organik antara dua konduktor. Ketika arus listrik diterapkan, cahaya terang dipancarkan. Berbeda seperti LCD yang membutuhkan lampu latar putih, teknologi OLED memancarkan cahaya

sehingga tidak memerlukan cahaya latar. Hal tersebut menjadikan OLED lebih tipis dan efisien dibanding dengan layar LCD.

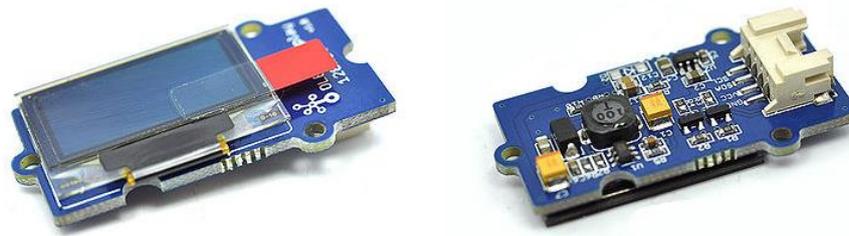


Gambar 2.4 Perbandingan Teknologi OLED dan LCD

OLED Display 0.96in adalah modul *display* matriks monokrom 128×64 dot. Modul ini memiliki 4pin I2C *Interface* untuk dihubungkan ke mikrokontroller. Dibandingkan dengan LCD, layar OLED lebih kompetitif, yang memiliki sejumlah keunggulan seperti kecerahan tinggi, emisi mandiri, kontras tinggi rasio, garis tipis, sudut pandang lebar, rentang suhu lebar, dan konsumsi daya rendah. Ia memiliki layar lebih besar sehingga dapat menampilkan lebih banyak konten daripada OLED 96×96 . (SEEED Studio, 2020) Adapun spesifikasi utamanya ditunjukkan pada Tabel 2.2.

Tabel 2.2 Spesifikasi Utama OLED Display 0.96in

| Item | Min | Normal | Max | Unit |
|----------------|-----|------------------------------|-----|------|
| Tegangan (VCC) | 3.3 | 5.0 | 5.5 | V |
| Driver IC | - | SSD1308Z | - | - |
| Warna tampilan | - | White | - | - |
| Dot Matrix | - | 128×64 | - | - |
| Ukuran Panel | - | $26.7(W) \times 19.26(H)$ | - | mm |
| Area aktif | - | $21.74(W) \times 11.175 (H)$ | - | mm |
| Dot Pitch | - | $0.17(W) \times 0.175 (H)$ | - | mm |
| Dot Size | - | $0.15(W) \times 0.15 (H)$ | - | mm |



Gambar 2.5 OLED Display 0.96in

2.2.7 Raspberry Camera Module 2

Modul Kamera Raspberry Pi v2 merupakan modul kamera resmi keluaran dari Raspberry Pi Foundation. Modul kamera ini dikeluarkan tahun 2016 dan merupakan pengganti dari seri sebelumnya yang dikeluarkan pada tahun 2013. Modul Kamera v2 memiliki sensor Sony IMX219 8-megapiksel. Adapun spesifikasi utama yang diambil dari situs resmi (Raspberrypi.org, 2020) adalah sebagaimana Tabel 2.3.

Tabel 2.3 Spesifikasi Utama Camera Module V2

| Spesifikasi | Keterangan |
|--------------------------|-------------------------------------|
| Berat | 3g |
| Resolusi | 8 Megapixels |
| Mode video | 1080p30, 720p60 dan 640 × 480p60/90 |
| Integrasi Linux | V4L2 driver |
| C programming API | OpenMAX IL dan lainnya tersedia |
| Sensor | Sony IMX219 |
| Resolusi sensor | 3280 × 2464 pixels |
| Sensor area gambar | 3.68 x 2.76 mm (4.6 mm diagonal) |
| Ukuran pixel | 1.12 μm x 1.12 μm |
| Ukuran lensa | 1/4" |
| Panjang fokal | 3.04 mm |
| Horizontal field of view | 62.2 degrees |
| Vertical field of view | 48.8 degrees |
| Focal ratio (F-Stop) | 2.0 |



Gambar 2.6 Camera Module V2

2.2.8 Sensor Ultrasonic HC SR-04

Sensor *Ultrasonic* HC-SR04 merupakan modul dengan 4 pin. Dimana masing-masing pin-nya adalah *Vcc*, *Trigger*, *Echo* dan *Ground*. Sensor ini merupakan sensor yang sangat populer digunakan dalam banyak aplikasi yang membutuhkan pengukuran jarak atau deteksi objek. Modul ini berbentuk seperti memiliki dua mata dengan yang menjorok ke depan, masing-masing berfungsi sebagai pemancar dan penerima gelombang ultrasonik. Penghitungan jarak dihitung menggunakan rumus sederhana:

$$\text{Jarak} = \text{Kecepatan} \times \text{Waktu}$$

Pemancar ultrasonik mentransmisikan gelombang ultrasonik, gelombang ini bergerak di udara. Ketika gelombang tersebut menabrak sebuah objek atau benda, kemudian gelombang tersebut akan memantul kembali ke penerima gelombang pada modul. Adapun ilustrasinya sebagaimana gambar berikut. (components101.com, 2020)

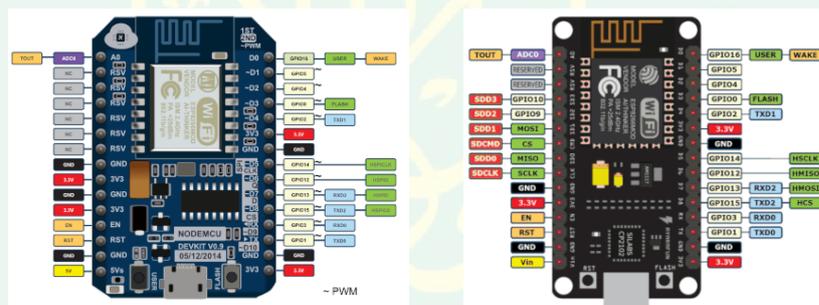


Gambar 2.7 Cara Kerja sensor ultrasonik

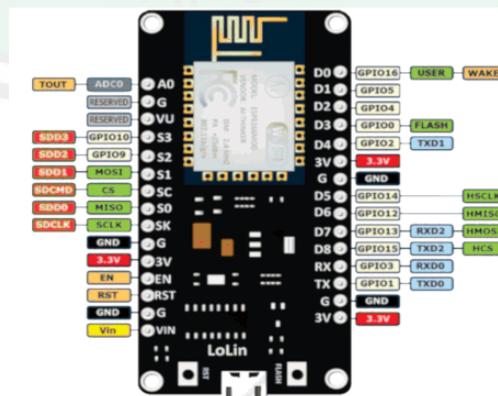
2.2.9 NodeMCU ESP8266

NodeMCU merupakan salah satu *development board* yang mampu terkoneksi dengan internet. *Board* ini dibangun berdasarkan modul wifi ESP8266. NodeMCU berjalan pada tegangan 3.3v dan menawarkan banyak serta mendukung protokol seperti SPI atau I2C. Dibandingkan dengan *board* lain dari Arduino atau Particle, NoceMCU hanya mendukung sebuah *analog input*. (Pulver, 2019)

Dalam perkembangannya NodeMCU memiliki 3 versi. V1 adalah versi keluaran pertama kali dan sudah tidak dipasarkan. V2 (Amica) merupakan pengembangan dari V1 dan memiliki *form factor* yang lebih baik. Sedang V3 (LoLin) tidak memiliki peningkatan yang besar, namun ukuran *board*-nya lebih besar dibandingkan dengan V2. Meskipun awalnya NodeMCU didesain untuk dapat diprogram dengan bahasa Lua, namun juga dapat diprogram menggunakan Arduino IDE. Dengan memasang *board* terkait pada Arduino *Board Manager* pengguna dapat memprogram layaknya seperti Arduino menggunakan bahasa C++. (Pulver, 2019)



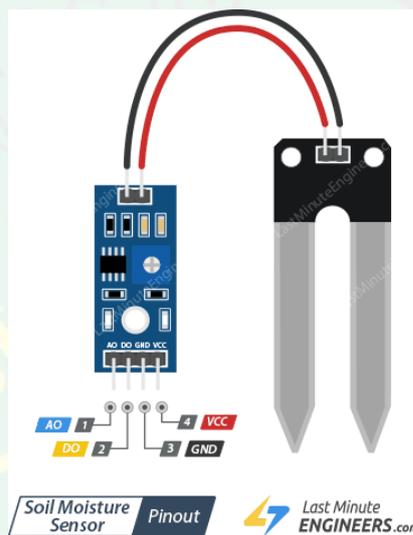
Gambar 2.8 NodeMCU V1 (kiri) dan NodeMCU V1 (kanan)



Gambar 2.9 NodeMCU V3

2.2.10 Soil Moisture Sensor

Soil moisture sensor atau sensor kelembaban tanah merupakan sensor kelembaban yang dapat digunakan untuk mendeteksi kelembaban dalam tanah. Sensor ini terdiri dari dua buah *probe* untuk melewatkan arus melalui tanah yang kemudian membaca resistensinya untuk mendapatkan tingkat nilai kelembaban. Semakin kering tanah maka akan didapatkan nilai resistansi yang besar karena sulit menghantarkan arus. Sedangkan dalam keadaan tanah yang basah atau banyak air maka akan semakin mudah dalam menghantarkan arus sehingga nilai resistansinya rendah. (Yahwe, Isnawaty, & Aksara, 2016)

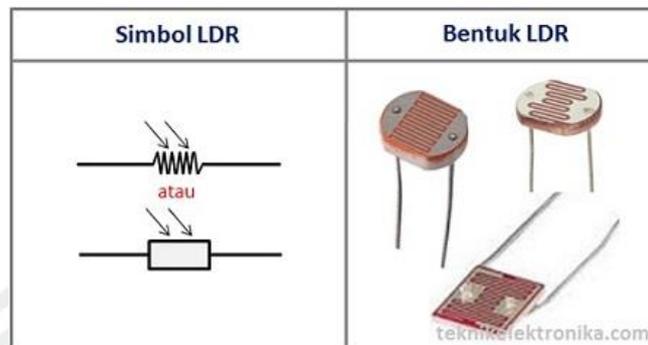


Gambar 2.10 Bentuk dan pin out soil moisture sensor

2.2.11 LDR (Light Dependent Resistor)/ Sensor Cahaya

Light dependent resistor atau biasa disingkat LDR merupakan suatu jenis resistor yang nilai resistensi atau hambatannya dipengaruhi oleh cahaya yang diterima. Besar nilai hambatan pada sebuah LDR dipengaruhi oleh besar kecilnya cahaya yang diterima. Nilai hambatan sebuah LDR akan semakin menurun jika ada penambahan intensitas cahaya yang mengenainya. Pada umumnya LDR memiliki resistensi sekitar 10M dalam keadaan gelap dan kurang lebih 150 dalam keadaan terang. Umumnya LDR terbuat dari bahan semikonduktor *cadmium sulfida*. Dengan menggunakan bahan semikonduktor ini energi cahaya yang jatuh menyebabkan lebih banyak

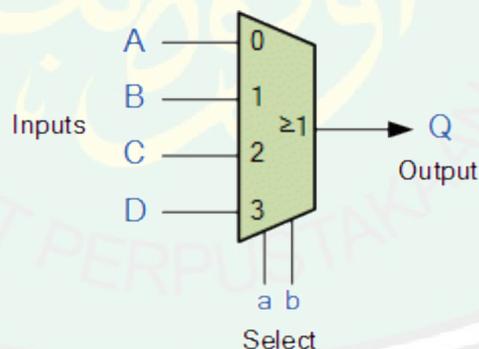
muatan yang dilepas atau arus listrik meningkat atau dengan kata lain terjadi penurunan resistensi bahan. (Sujarwata, 2012)



Gambar 2.11 Bentuk LDR dan simbolnya

2.2.12 Analog Multiplexer

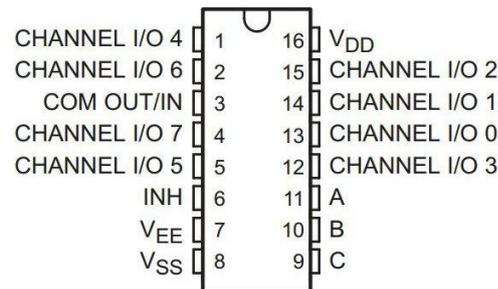
Multiplexer atau yang biasa disingkat MUX atau MPX merupakan sebuah rangkaian yang memiliki banyak *input* tetapi hanya memiliki satu *output*. Pada rangkaian *multiplexer* sinyal input bisa disalurkan menuju jalur *output* melalui sinyal kendali. Sinyal kendali tersebut akan memilih bit *address* mana yang akan diaktifkan atau dipilih. Sinyal kendali sering disebut sebagai *input select*. Penyaluran *input* menuju *output* pada *multiplexer* hanya dapat dilakukan satu kali dalam suatu waktu. (Widjanarka, 2006)



Gambar 2.12 Multiplexer 4 channel

Salah satu *Integrated Circuit (IC)* mux/ demux adalah CD4051. IC ini merupakan IC *mux single 8-channel* dan memiliki 3 *input switch* atau *binary control* dan *inhibit input*. Dengan demikian 3 *input switch* tersebut bisa memilih 1 dari 8 *channel* untuk diaktifkan dan menghubungkan 1 dari 8 *input*

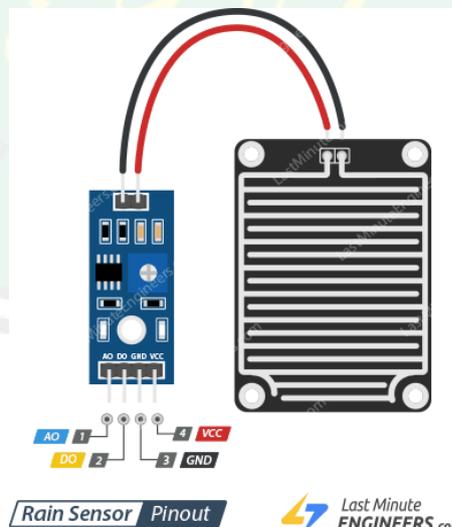
menuju *output*. Untuk beroperasi IC ini memerlukan data 5V. (Texas Instruments Incorporated, 2003)



Gambar 2.13 Pin out IC CD4051

2.2.13 Sensor Hujan

Sensor hujan atau juga disebut *raindrop* sensor merupakan sebuah sensor yang dapat digunakan untuk mendeteksi hujan. Sensor ini dapat digunakan sebagai *switch* atau saklar jika ada tetesan air yang mengenainya. Sensor ini juga bisa digunakan untuk mengukur intensitas curah hujan. Sensor ini bekerja dengan cara mendeteksi adanya air yang mengenai garis-garis sirkuit pada *sensor board*. *Sensor board* bekerja layaknya sebuah variabel resistor dan memiliki nilai resistensi 100K dalam keadaan basah dan 2M ketika dalam keadaan kering. Dengan kata lain semakin banyak terkena air atau basah maka semakin banyak arus yang dialirkan. (Raju, 2017)



Gambar 2.14 Bentuk dan *pin out* sensor hujan

2.2.14 Relay

Relay adalah *switch* atau saklar yang bisa dikendalikan menggunakan arus. *Relay* terdiri dari sebuah *low current coil* yang dililitkan pada sebuah inti. Di dalamnya terdapat sebuah armatur besi yang akan tertarik menuju inti apabila arus mengalir melewati kumparan. Armatur ini terpasang pada sebuah tuas berpegas. Ketika armatur tertarik menuju ini, kontak jalur bersama akan berubah posisinya dari kontak *normally-closed* (NC) ke kontak *normally-open* (NO). (Turang, 2015)

Relay dapat digunakan untuk mengontrol motor AC atau komponen AC lainnya dengan rangkaian kontrol DC atau beban lain dengan sumber tegangan yang berbeda. Contoh pengaplikasian *relay* biasanya adalah sebagai kontrol ON/OFF beban dengan sumber tegang berbeda. *Relay* sebagai selektor atau pemilih hubungan. *Relay* sebagai eksekutor rangkaian delay (tunda). *Relay* sebagai pelindung atau pemutus arus pada kondisi tertentu. (Turang, 2015)

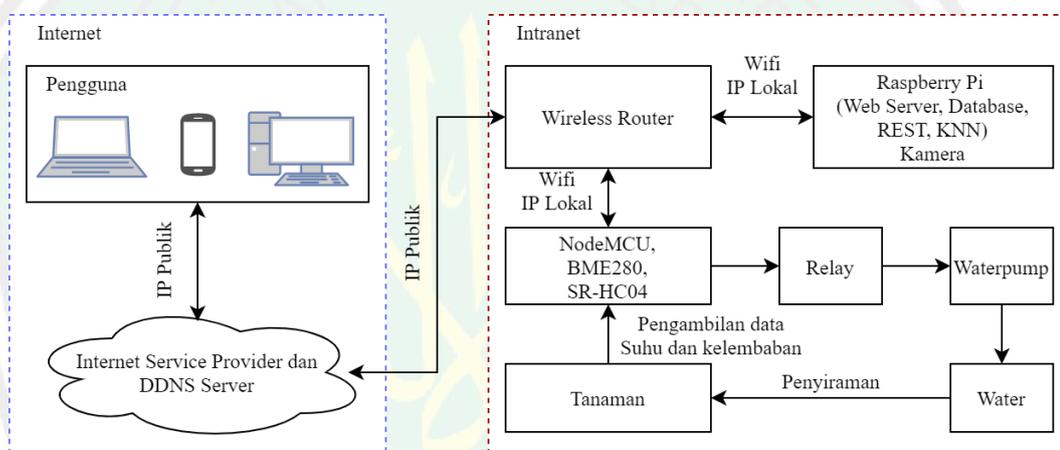


Gambar 2.15 *Relay module 2 channel*

BAB III METODOLOGI PENELITIAN

3.1 Desain Sistem

Sistem ini dibuat dengan tujuan agar dapat melakukan proses klasifikasi status kondisi tanah menggunakan metode *K-Nearest Neighbor* berdasarkan dari *input* sensor berupa sensor kelembaban tanah dan suhu. *Output* dari sistem ini merupakan hasil klasifikasi yang kemudian sistem akan melakukan penyiraman berdasarkan hasil klasifikasi tersebut. Selain itu data *monitoring* dan klasifikasi akan ditampilkan pada sebuah halaman web. Gambaran umum sistem yang akan dibangun dalam penelitian ini adalah sebagai Gambar 3.1.

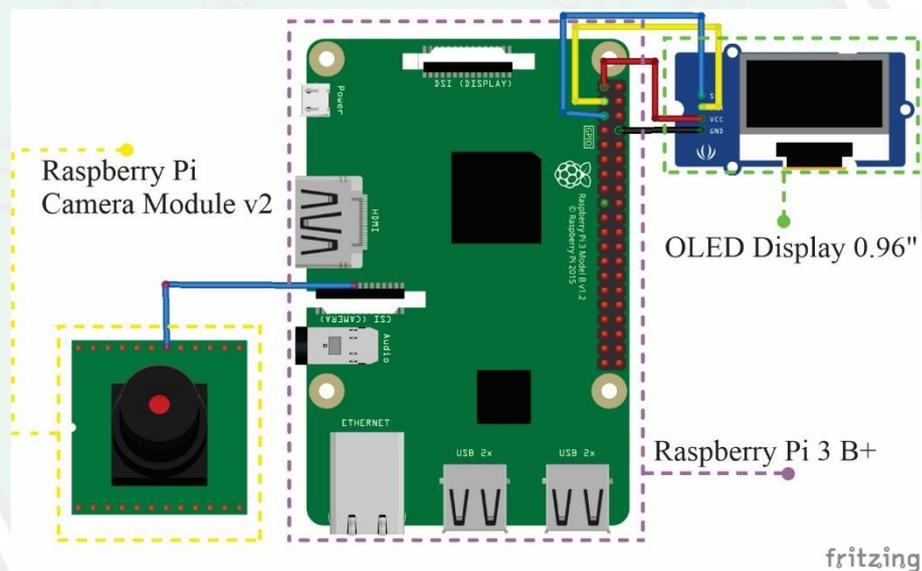


Gambar 3.1 Diagram Desain Sistem

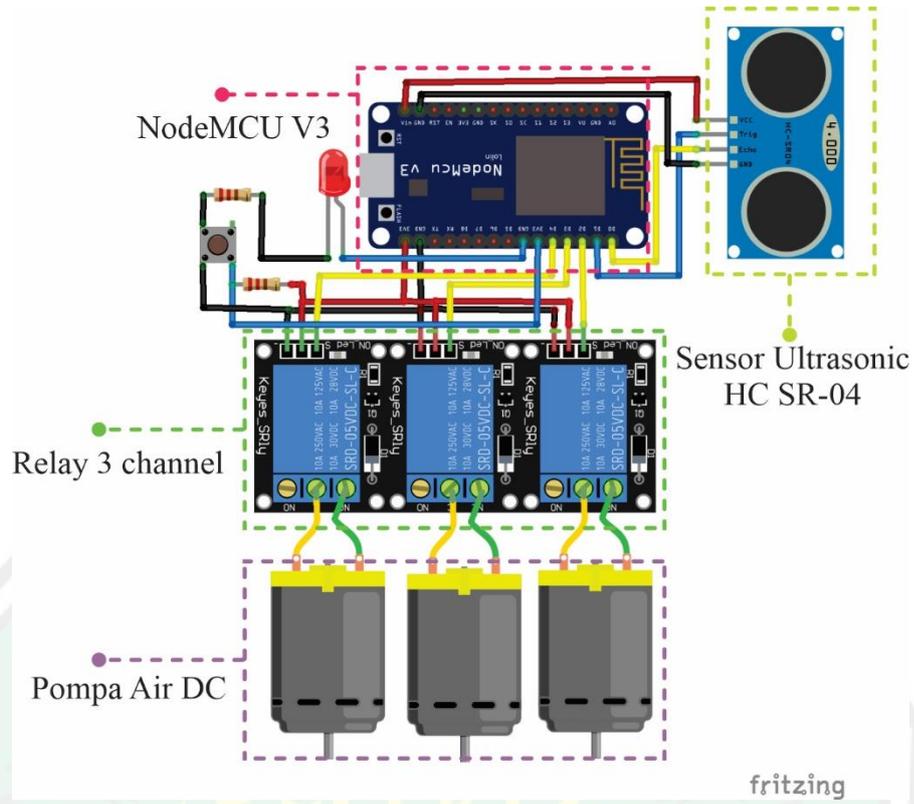
NodeMCU membaca suhu melalui sensor BME280 dan kelembaban tanah menggunakan sensor *soil moisture*. Kemudian data tersebut dikirimkan ke Raspberry Pi yang dalam penelitian ini bekerja sebagai *server* melalui jaringan WiFi. Data sensor tersebut kemudian akan disimpan ke dalam *database*. Data tersebutlah yang kemudian dijadikan parameter untuk melakukan klasifikasi. Klasifikasi menggunakan metode KNN dilakukan di dalam server. Setelah klasifikasi selesai maka server akan mengirimkan data ke NodeMCU untuk melakukan penyiraman sesuai kondisi dari hasil klasifikasi.

3.1.1 Hardware System

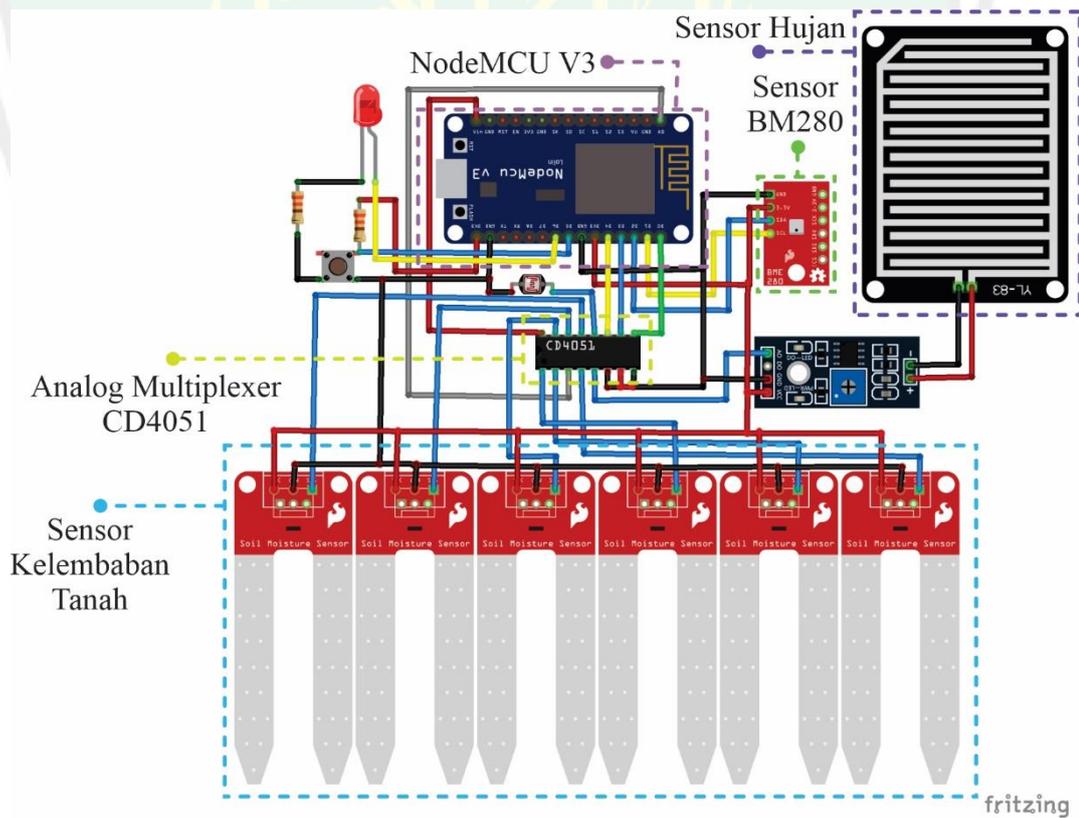
Desain komponen dibutuhkan untuk mengetahui kebutuhan sistem. Desain komponen elektronik ini juga berperan sebagai *blueprint* dalam membangun sistem. Dalam penelitian ini terdapat tiga rangkaian yang akan dibangun, yaitu Raspberry Pi sebagai server, NodeMCU sebagai *sensor node* dan NodeMCU sebagai *watering node*. Gambar 3.2 merupakan *wiring diagram* dari Raspberry Pi sebagai server, sedangkan Gambar 3.3 adalah NodeMCU *watering node* dan Gambar 3.4 adalah NodeMCU *sensor node*. Pada Kabel warna merah pada sistem menunjukkan VCC atau kutub positif, warna hitam *Ground/ GND* atau kutub negatif sedangkan kabel lainnya adalah koneksi data.



Gambar 3.2 *Wiring Diagram* Raspberry Pi sebagai server



Gambar 3.3 Wiring Diagram NodeMCU Watering Node



Gambar 3.4 Wiring Diagram NodeMCU Sensor Node

3.1.1.1 Raspberry Pi

Raspberry Pi 3B+ pada penelitian ini menjadi bagian utama. Pada server inilah metode *K-Nearest Neighbor* dijalankan. Untuk bisa berjalan dilakukan instalasi sistem operasi dan aplikasi pendukung lainnya seperti *Apache web server*, *MySQL server*, *REST server*, *Python*, *web browser* dan *code editor*. Raspberry Pi juga dipasang dengan modul kamera untuk melakukan *streaming* dan modul OLED display untuk menampilkan info sensor tanpa harus menghubungkan ke *external display* seperti monitor.

3.1.1.2 Camera Module

Raspberry Pi Camera Module v2 dihubungkan melalui *camera interface* yang terdapat pada Raspberry Pi 3 B+. Kamera modul ini memiliki resolusi 8 mega pixel dengan sensor Sony IMX219. Untuk bisa terhubung dengan raspberry pi bisa dihubungkan dengan *Camera Interface*. Pada sistem yang dibangun kamera ini digunakan untuk melakukan pengambilan gambar dan *streaming* oleh *user* melalui *interface* halaman web.

3.1.1.3 OLED Display

OLED *display* yang digunakan merupakan salah satu modul *bundling* dari Grove. Modul ini memiliki ukuran layar 0.96 inchi. Modul ini masih monokrom dengan resolusi 128x64dot. Untuk bisa digunakan di raspberry, module ini memerlukan tegangan minimal 5v dan memiliki interface I2C dengan *address* 0x3C. Pada sistem yang dibangun OLED *display* digunakan untuk menampilkan data sensor secara langsung dan juga untuk menampilkan alamat IP.

3.1.1.4 NodeMCU

Pada sistem yang dibangun digunakan dua buah NodeMCU sebagai *sensor node* dan *watering node*. Sensor node merupakan rangkaian yang dibangun untuk membaca data sensor kelembaban

udara, suhu, barometer dan ketinggian (BME280), sensor *soil moisture* (kelembaban tanah), sensor hujan, dan sensor cahaya. Sensor node akan membaca data sensor tersebut dan kemudian mengirim data ke *server* melalui API yang sudah disediakan melalui jaringan WiFi. Pada penelitian ini pengambilan data sensor dilakukan berkala setiap satu menit sekali.

Watering node merupakan rangkaian NodeMCU yang dibangun untuk melakukan penyiraman dan membaca kapasitas air penyiraman atau tandon yang tersisa. Pada rangkaian ini NodeMCU dihubungkan dengan sensor *ultrasonic* dan 3 buah *relay* untuk mengontrol pompa penyiraman. Cara kerja dari rangkaian ini adalah dengan melakukan *request* data penyiraman ke server. Selanjutnya dari *request* tersebut akan diperoleh sebuah respons berupa dokumen JSON. Respons tersebut akan memberitahukan *watering node* tentang waktu penyiraman dan lama penyiraman yang dibutuhkan.

Setiap NodeMCU juga dilengkapi dengan *push button* dan *LED*. *Push button* digunakan untuk melakukan *restart* dan *reset* data konfigurasi. Hal itu tersebut dimaksudkan untuk mempermudah dalam *troubleshooting* jika terjadi kendala sistem. NodeMCU telah diprogram sehingga jika *push button* ditekan satu kali maka akan melakukan *restart* dan lampu LED akan menyala menandakan bahwa berhasil *restart*. Untuk melakukan reset data perlu menekan dan menahan *push button* selama 3 detik dan lampu LED akan berkedip jika berhasil.

3.1.1.5 HC SR-04

Untuk memantau kapasitas tangki penyiraman digunakan sensor *ultrasonic* HC-SR04. Sensor ini mampu mengukur jarak hingga 5 meter. Pada penelitian ini tangki yang digunakan berbentuk tabung. Untuk menentukan ketinggian atau kapasitas air pertamanya dilakukan perhitungan volume tabung kosong dan volume air yang terisi. Adapun perhitungannya digunakan rumus volume tabung sederhana sebagai berikut:

$$V = \text{phi} \times r^2 \times t$$

Keterangan:

V : Volume

Phi : 22/7

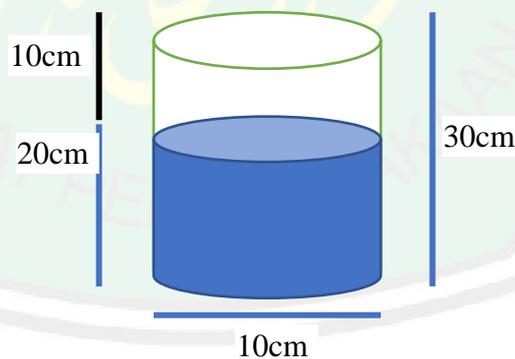
R : jari-jari

T : Tinggi tabung

Kemudian jika volume tabung kosong dan volume air sudah dihitung, selanjutnya adalah menghitung presentasi dari kapasitas air yang tersedia. Adapun persamaan yang digunakan adalah sebagai berikut:

$$\text{Kapasitas air} = \frac{\text{volume air}}{\text{volume tabung}} \times 100$$

Sebagai contoh pada gambar diketahui bahwa ada tangki dengan ketinggian 30cm dan jari-jari 5cm, sedangkan hasil pembacaan sensor *ultrasonic* adalah 10cm. Maka perhitungannya adalah:



Gambar 3.5 Contoh perhitungan kapasitas air

$$v_{air} = \frac{22}{7} \times 5 \times 5 \times (30 - 10)$$

$$v_{air} = \frac{22}{7} \times 25 \times 20$$

$$v_{air} = 1570\text{cm}^3 = 1.57 \text{ liter}$$

$$v_{tabung} = \frac{22}{7} \times 5 \times 5 \times 30$$

$$v_{tabung} = \frac{22}{7} \times 25 \times 30$$

$$v_{tabung} = 2355\text{cm}^3 = 2.36 \text{ liter}$$

$$\text{Kapasitas air} = \frac{1.57}{2.36} \times 100 = 66.5\%$$

Jadi dengan demikian kapasitas air yang tersedia pada contoh kasus di atas adalah 66.5%.

3.1.1.6 Multiplexer

Pada penelitian ini digunakan IC *Analog Multiplexer* CD4051. IC tersebut digunakan untuk memperbanyak analog *input* karena NodeMCU hanya memiliki satu *analog input*. CD4051 merupakan analog mux dengan 8 *channel* keluaran dan memiliki 3 *input switch*. CD4051 dipilih karena pada penelitian ini menggunakan 6 sensor kelembaban tanah, 1 sensor hujan dan 1 sensor cahaya, sehingga diperlukan 8 analog *input*. Adapun tabel kebenaran dari IC CD4051 yang digunakan adalah sebagaimana pada Tabel 3.1. Pada tabel tersebut bisa dilihat jika NodeMCU mengirimkan sinyal LOW pada setiap pin *input switch* maka CD4051 akan dalam *state* 0. Jika dalam *state* 0 maka CD4051 akan membuka jalur *input* pin 0 dan menutup jalur lainnya. Kemudian CD4051 akan menyalurkan pembacaan dari *input* pin 0 tadi ke analog pin NodeMCU, begitu juga *state* selanjutnya.

Tabel 3.1 Tabel kebenaran mux 8 channel

| Input Switch | | | State/ Output | Sensor |
|--------------|---|---|------------------|---------------------------|
| A | B | C | | |
| 0 | 0 | 0 | 0 | Sensor kelembaban tanah 1 |
| 0 | 0 | 1 | 1 | Sensor kelembaban tanah 2 |
| 0 | 1 | 0 | 2 | Sensor kelembaban tanah 3 |
| 0 | 1 | 1 | 3 | Sensor kelembaban tanah 4 |
| 1 | 0 | 0 | 4 | Sensor kelembaban tanah 5 |
| 1 | 0 | 1 | 5 | Sensor kelembaban tanah 6 |
| 1 | 1 | 0 | 6 | Sensor hujan |
| 1 | 1 | 1 | 7 | Sensor cahaya |

3.1.1.7 Relay

Untuk mengaktifkan pompa air digunakan *relay*. *Relay* ini berfungsi layaknya saklar. Pada sistem yang dibangun *relay* dihubungkan dengan NodeMCU *watering node*. Pemasangan *relay* dipasang dengan keadaan *Normally Closed*. Dengan memasang secara *normally closed* pompa air akan terbuka hanya jika ada perintah dari NodeMCU.

3.1.2 Software System

3.1.2.1 Sistem Operasi

Pada penelitian ini digunakan Raspbian OS sebagai sistem operasi pada Raspberry Pi. Sistem operasi ini merupakan sistem operasi resmi dari Raspberry dan merupakan distro dari Linux Debian. Adapun versi Raspbian OS yang digunakan adalah versi Raspbian Buster. Sistem operasi di *install* ke dalam memori *SD Card* berukuran 16GB.

3.1.2.2 MySQL Server

Pemakaian perangkat lunak selanjutnya yaitu MySQL Server sebagai *database* yang dipakai. Segala penyimpanan yang dilakukan pada web tersimpan pada *database* ini. MySQL merupakan database management system (DBMS) yang querinya memakai *structured query language*. MySQL Server dipilih karena beberapa kelebihanannya dibanding dengan *database* lain, salah satunya karena *software* ini bersifat *open source*.

Data yang berasal dari nilai baca *input* sensor BME280 selanjutnya disimpan ke dalam *database* dengan server MySQL dan diakses melalui PhpMyAdmin. Pada *database* ini yang nantinya akan menyimpan seluruh data yang ada pada sistem. Variabel yang disimpan ke dalam *database* yaitu berupa *date time*, suhu, kelembaban, volume air dan nilai klasifikasi.

3.1.2.3 Web Server

Pada *webservice* kita dimungkinkan untuk menghubungkan ke halaman web dan memproses permintaan oleh browser. Webservice yang paling populer dan digunakan yaitu Apache. Selanjutnya *PHP* (*Hypertext Preprocessor*) digunakan untuk membuat tampilan pada web. Pemakaian PHP pada situs web menyesuaikan dengan keinginan pengguna. Dalam pengaplikasiannya PHP dikombinasikan dengan *HTML* (*Hypertext Markup Language*) untuk mendapat hasil maksimal yang diinginkan. Penggabungan kedua elemen ini biasanya disebut *Scripting Language*. *Website* didesain sedemikian rupa sehingga nyaman untuk dipakai.

3.1.2.4 REST Server

Untuk kemudahan pertukaran data dan pengembangan digunakan REST. REST *Server* menyediakan data *resource* dan REST *Client* akan mengambil dan menampilkan data untuk penggunaan selanjutnya. Pada penelitian ini data *resource*

direpresentasikan dalam format JSON. Untuk melakukan transaksi data *resource* digunakan protokol komunikasi standar HTTP *Request* dan HTTP *Response*. HTTP memiliki beberapa metode umum untuk melakukan transaksi data, diantaranya POST, GET, UPDATE, DELETE, dst.

3.1.3 Pembacaan Sensor

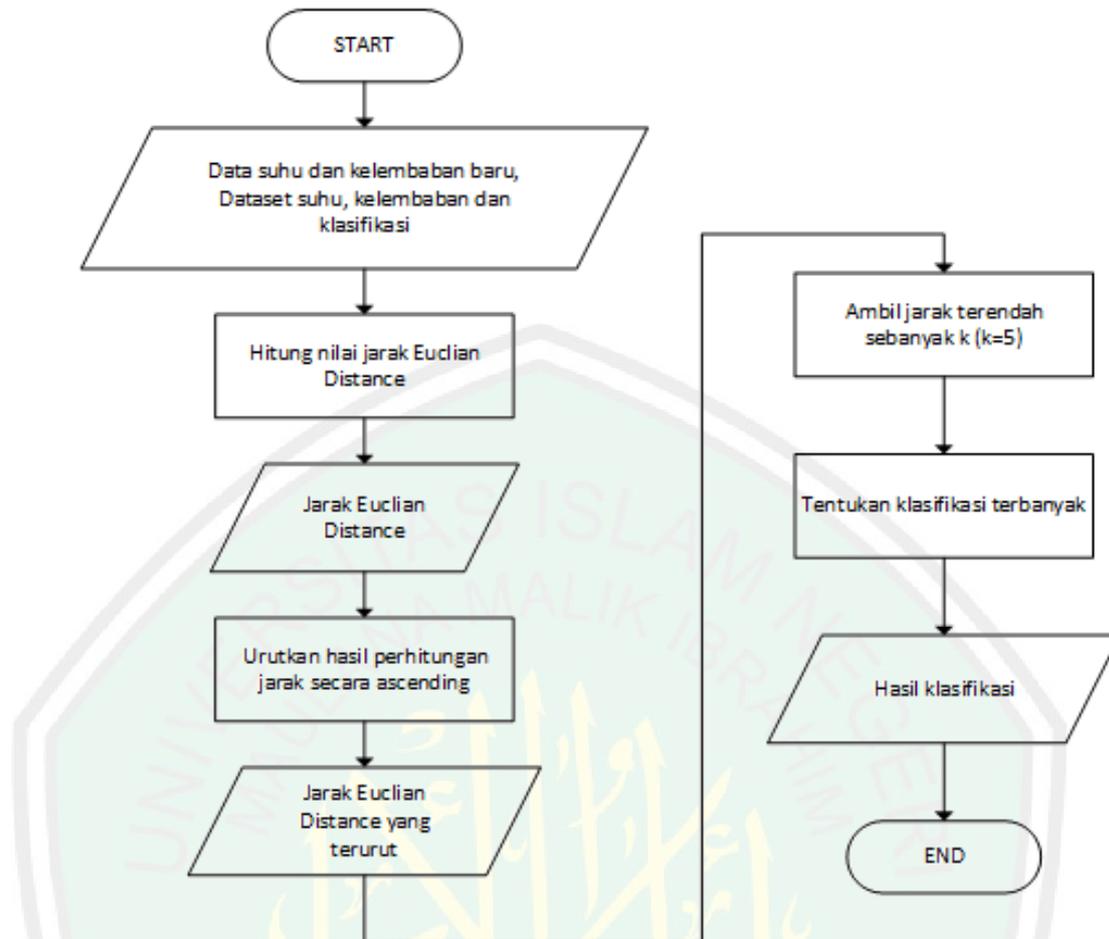
Data sensor diproses oleh NodeMCU yang sudah tersambung dengan masing-masing sensor. Sensor suhu akan mengambil data suhu dalam satuan Celsius, sedangkan kelembaban akan diambil dengan satuan persen. Volume air dalam tangki juga akan diambil dengan sensor *ultrasonic* menggunakan rumus luas volume tabung. Pembacaan data sensor dilakukan secara *real-time*. Kemudian pembacaan data ini dikirimkan oleh NodeMCU ke server. Adapun contoh data yang dikirimkan ke server ditunjukkan pada Tabel 3.2.

Tabel 3.2 Contoh Data Sensor

| No. | Date time | Suhu (C) | Kelembaban (%) | Volume Air |
|-----|------------------------|----------|----------------|------------|
| 1 | 2020-01-20 08:00:00 | 28 | 60 | 27 |
| 2 | 2020-01-20 17:00:00 | 28 | 60 | 26 |
| 3 | 2020-01-21 08:00:00 | 27 | 60 | 26 |

3.1.4 Metode KNN

Proses klasifikasi ini terjadi di Raspberry Pi. Adapun prosesnya ditunjukkan *flowchart* pada Gambar 3.6.



Gambar 3.6 *Flowchart* proses klasifikasi dengan metode KNN

Pertama-tama data suhu dan kelembaban diambil oleh sensor yang terpasang pada NodeMCU. Data kelembaban tanah dan suhu tersebut kemudian dikirim ke *server database* yaitu Raspberry Pi. Setelah berhasil tersimpan, akan diambil *dataset* suhu, kelembaban dan labelnya yang sebelumnya sudah dipersiapkan. Data suhu dan kelembaban yang baru selanjutnya akan dihitung jaraknya menggunakan *Euclidean Distance*.

Setelah perhitungan jarak selesai, selanjutnya hasil perhitungan tersebut akan diurutkan secara *ascending* (mulai dari jarak terdekat ke jarak terjauh). Dari data yang sudah diurutkan tadi, kemudian diambil data pertama sebanyak k (tetangga terdekat). Data yang diperoleh tersebut kemudian dihitung mana label klasifikasi terbanyak. Klasifikasi terbanyak tersebutlah yang nanti akan dijadikan klasifikasi dari data yang baru. Setelah itu data

disimpan ke dalam *database* berupa suhu, kelembaban dan klasifikasi. Data terbaru ini juga nantinya akan menjadi *dataset* baru.

Adapun contoh hasil perhitungan secara manual adalah sebagai berikut. Terdapat *dataset* sebanyak 10 *record* yang sudah ditentukan sebagai data awal sebagaimana pada Tabel 3.3. *Dataset* tersebut terdiri dari nilai kelembaban, suhu, serta label (klasifikasi). Kemudian didapatkan nilai suhu 28°C dan kelembaban 20%. Sebagai contoh digunakan nilai k (tetangga terdekat) = 5.

Tabel 3.3 Contoh *dataset*

| Case | Kelembaban(%) | Suhu(C) | Klasifikasi |
|------|---------------|---------|-------------|
| 1 | 50 | 35 | Kering |
| 2 | 60 | 30 | Agak Kering |
| 3 | 70 | 20 | Agak Basah |
| 4 | 85 | 15 | Basah |
| 5 | 55 | 33 | Kering |
| 6 | 65 | 28 | Agak Kering |
| 7 | 73 | 17 | Agak Basah |
| 8 | 90 | 20 | Basah |
| 9 | 60 | 38 | Kering |
| 10 | 80 | 15 | Basah |

Data yang baru didapat tersebut kemudian dihitung jaraknya terhadap *dataset* menggunakan rumus *Euclidean Distance*. Sebagai contoh data baru akan dihitung jaraknya terhadap *case* 1 dari *dataset* di atas.

$$d_1 = \sqrt{(h_{x1} - h_y)^2 + (t_{x1} - t_y)^2}$$

Keterangan:

d_1 : jarak ke 1

h_{x1} : nilai kelembaban *dataset* ke 1

h_y : nilai kelembaban baru
 t_{x1} : nilai suhu *dataset* ke 1
 t_y : nilai suhu baru

Sehingga:

$$d_1 = \sqrt{(50 - 20)^2 + (35 - 28)^2}$$

$$d_1 = \sqrt{900 + 49}$$

$$d_1 = 30,8058436$$

Adapun hasil dari perhitungan jarak seluruh *dataset* sebagaimana pada Tabel 3.4 berikut.

Tabel 3.4 Tabel hasil perhitungan jarak

| Case | Kelembaban(%) | Suhu(C) | Jarak | Label |
|------|---------------|---------|-------------|----------------|
| 1 | 50 | 35 | 30,8058436 | Kering |
| 2 | 60 | 30 | 40,04996879 | Agak Kering |
| 3 | 70 | 20 | 50,6359556 | Agak Basah |
| 4 | 85 | 15 | 66,28725368 | Basah |
| 5 | 55 | 33 | 35,35533906 | Kering |
| 6 | 65 | 28 | 45 | Agak Kering |
| 7 | 73 | 17 | 54,12947441 | Agak Basah |
| 8 | 90 | 20 | 70,45565982 | Basah |
| 9 | 60 | 38 | 41,23105626 | Kering |
| 10 | 80 | 15 | 61,39218191 | Basah |

Setelah jarak antar nilai diketahui dilanjutkan dengan melakukan *sorting*. *Sorting* dilakukan secara *ascending* mulai jarak terendah ke jarak terjauh. Adapun hasil *sorting* sebagaimana pada Tabel 3.5.

Tabel 3.5 Tabel hasil *sorting*

| Case | Kelembaban(%) | Suhu(C) | Jarak | Label | Urutan |
|------|---------------|---------|-------------|-------------|--------|
| 1 | 50 | 35 | 30,8058436 | Kering | 1 |
| 5 | 55 | 33 | 35,35533906 | Kering | 2 |
| 2 | 60 | 30 | 40,04996879 | Agak Kering | 3 |
| 9 | 60 | 38 | 41,23105626 | Kering | 4 |
| 6 | 65 | 28 | 45 | Agak Kering | 5 |
| 3 | 70 | 20 | 50,6359556 | Agak Basah | 6 |
| 7 | 73 | 17 | 54,12947441 | Agak Basah | 7 |
| 10 | 80 | 15 | 61,39218191 | Basah | 8 |
| 4 | 85 | 15 | 66,28725368 | Basah | 9 |
| 8 | 90 | 20 | 70,45565982 | Basah | 10 |

Langkah selanjutnya adalah mengambil jarak tetangga terdekat sebanyak $k=5$. Sehingga ditemukan hasil sebagaimana pada Tabel 3.6 berikut. Selanjutnya akan ditentukan label klasifikasinya berdasarkan klasifikasi terbanyak pada tetangga terdekat. Dari Tabel 3.6 ditemukan bahwa label terbanyak adalah “Kering”, sehingga data baru tadi (suhu 28°C dan kelembaban 20%) termasuk dalam kondisi tanah kering.

Tabel 3.6 Tetangga terdekat

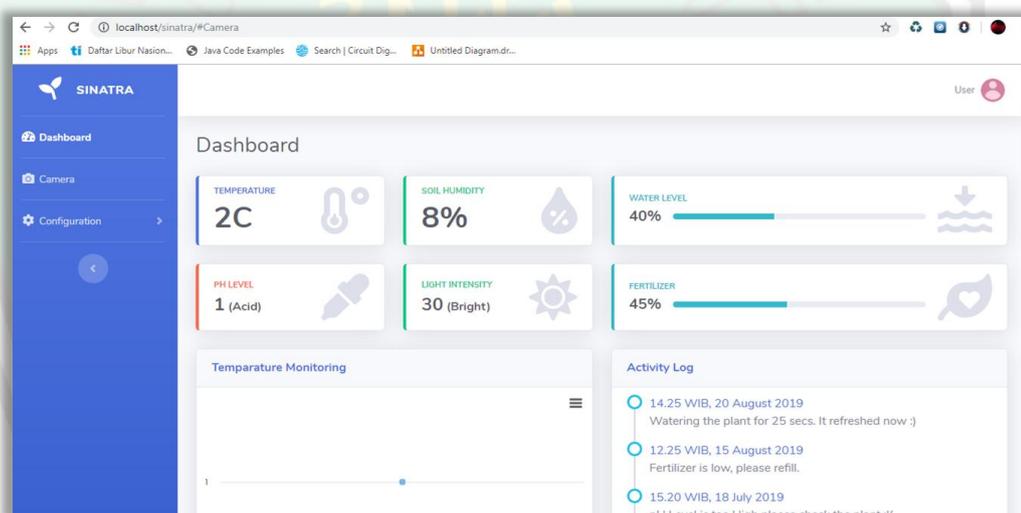
| Case | Kelembaban(%) | Suhu(C) | Jarak | Label | Urutan |
|------|---------------|---------|-------------|---------------|--------|
| 1 | 50 | 35 | 30,8058436 | Kering | 1 |
| 5 | 55 | 33 | 35,35533906 | Kering | 2 |
| 2 | 60 | 30 | 40,04996879 | Agak Kering | 3 |
| 9 | 60 | 38 | 41,23105626 | Kering | 4 |
| 6 | 65 | 28 | 45 | Agak Kering | 5 |

3.1.5 Desain Tampilan Antarmuka

Data yang sudah disimpan kemudian diolah sedemikian rupa yang selanjutnya ditampilkan dalam bentuk diagram. Pembuatan web dilakukan

dengan menggunakan pemrograman *PHP* (*Hypertext Preprocessor*). Pada tampilan data di web pengguna dapat *monitoring* secara langsung (*real-time*). Pada halaman web akan ditampilkan data *log sensor*, data pembacaan sensor dan juga data perhitungan KNN. Pembacaan pada setiap sensor akan muncul pada *dashboard*. Menu selanjutnya yaitu berupa *streaming* secara langsung. Data *streaming* merupakan video tampilan secara langsung yang menampilkan kondisi tanaman.

Halaman web *interface* dibangun menggunakan bahasa pemrograman PHP dengan *framework* CodeIgniter 3. Selain itu juga dikombinasikan dengan HTML dan CSS untuk mendesain tampilan. Javascript juga digunakan untuk menjadikan halaman web yang dibangun lebih interaktif. Penggabungan beberapa elemen tersebut biasa disebut dengan *Scripting Language*. Adapun tampilan dari web *interface* bisa dilihat pada Gambar 3.7.

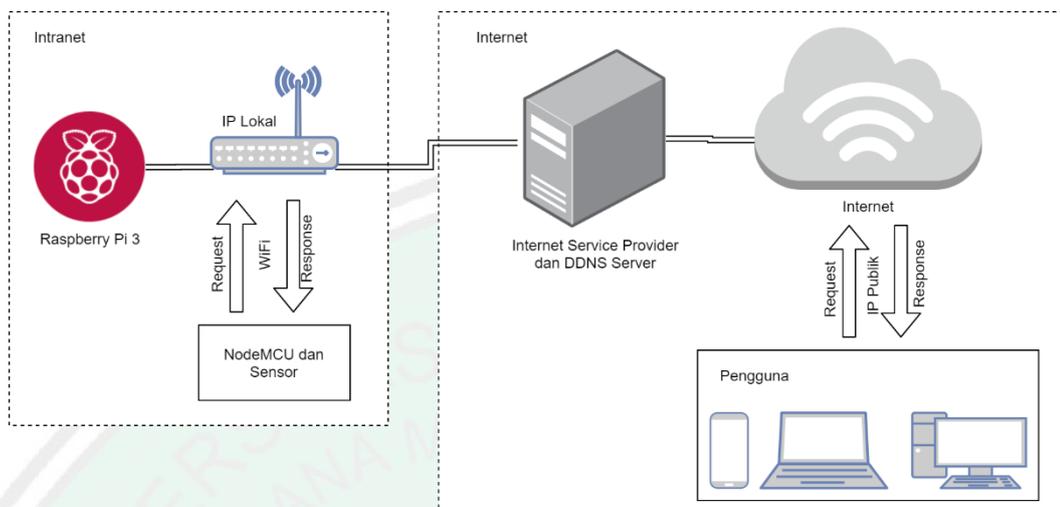


Gambar 3.7 Tampilan *Prototype web interface*

3.2 Desain Jaringan

Adapun desain jaringan pada penelitian ini ditunjukkan pada Gambar 3.8. Raspberry Pi bekerja sebagai server dan dihubungkan dengan *router* dan terkoneksi internet. NodeMCU yang bekerja membaca sensor dan penyiraman terhubung ke server dengan IP Lokal melalui jaringan Intranet. Pengguna bisa mengakses data server melalui internet menggunakan IP

Publik. Sebelumnya *router* sudah dilakukan *port forwarding* sehingga server bisa diakses secara publik.



Gambar 3.8 Desain Jaringan

3.3 Rencana Uji Coba

Pengujian sistem dilakukan untuk mencoba apakah sistem yang dibangun sudah akurat atau belum. Keakuratan sensor akan diuji sebagaimana pada Tabel 3.7. Sensor diuji dengan membandingkan antara nilai pembacaan dari sensor dengan termometer dan higrometer. Dari hasil perbandingan tersebut akan didapatkan selisih dan merupakan *error*. Untuk mendapatkan *error* digunakan persamaan *relative error* (Bakshi & Bakshi, 2008). *Relative error* merupakan persamaan untuk mencari nilai *error* dengan membagi nilai *absolute error* dengan nilai yang sebenarnya. Pada penelitian ini, nilai sebenarnya didapatkan dari hasil pembacaan alat ukur standar baik *thermometer* untuk suhu, *hygrometer* untuk kelembaban maupun mistar untuk jarak. Sedangkan nilai *absolute error* bisa didapatkan dengan mengabsolutkan nilai pengurangan dari pembacaan sensor dengan pembacaan alat ukur.

$$\%error = \frac{|x - y|}{y} \times 100$$

Keterangan:

$\%error$: presentasi *error* dari sensor

x : Nilai baca sensor

y : Nilai baca alat ukur

Tabel 3.7 Tabel Kalibrasi Sensor

| No | Waktu | Suhu | | | Kelembaban | | |
|----|-------|--------|------------|-------|------------|------------|-------|
| | | Sensor | Termometer | Error | Sensor | Higrometer | Error |
| 1 | | | | | | | |
| 2 | | | | | | | |

Untuk mengevaluasi algoritma digunakan *confussion matrix*. Metode evaluasi ini memetakan kinerja algoritma dalam bentuk tabulasi. Matrik ini menunjukkan hubungan antara benar tidaknya sebuah data dikategorikan. (Prakasa & Lhaksamana, 2018) Adapun tabel yang digunakan dalam metode evaluasi *confusion matrix* pada 2 *class* adalah sebagaimana ditunjukkan pada Tabel 3.8.

Tabel 3.8 Tabel *Confusion Matrix 2 Class*

| | Label | |
|---------|----------------------|----------------------|
| | Positif | Negatif |
| Positif | <i>True Positif</i> | <i>False Positif</i> |
| Negatif | <i>False Negatif</i> | <i>True Negatif</i> |

Dari tabel di atas bias dilihat bahwa *confusion matrix* terdiri dari *True positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*. *True positive* merepresentasikan data yang berada pada kelas positif yang diprediksi secara benar oleh algoritma. *False Positive* merepresentasikan data yang seharusnya berada pada kelas positif diprediksi menjadi kelas negatif oleh algoritma. *False Negative* merupakan data yang seharusnya berada di kelas negatif diprediksi menjadi kelas positif oleh algoritma. *True Negative* merupakan data yang berada pada kelas negatif dan

diprediksi secara benar oleh algoritma (Prakasa & Lhaksamana, 2018). Berdasarkan *confusion matrix*, dapat diketahui berbagai parameter pengukuran kinerja algoritma, yaitu presisi, *recall*, f1-measure, dan akurasi.

Presisi merupakan parameter untuk mengukur ketepatan dari suatu algoritma. Misalkan untuk menghitung *precision* algoritma dalam memprediksi data berlabel positif, *precision* dihitung berdasarkan rasio jumlah data berlabel positif yang diprediksi secara benar oleh algoritma dengan jumlah data yang diprediksi memiliki label positif oleh algoritma (Prakasa & Lhaksamana, 2018).

$$precision = \frac{TP}{TP + FP}$$

Recall merupakan parameter untuk mengukur kelengkapan sebuah algoritma. Misalkan untuk menghitung *recall* algoritma dalam memprediksi data berlabel positif, *recall* dihitung berdasarkan rasio jumlah data berlabel positif yang diprediksi secara benar oleh algoritma dengan jumlah semua data yang berlabel positif pada *dataset* (Prakasa & Lhaksamana, 2018).

$$recall = \frac{TP}{TP + FN}$$

F-measure merupakan rata-rata harmonik dari presisi dan *recall*. Nilai tertinggi adalah 1 dan nilai terendah adalah 0 (Prakasa & Lhaksamana, 2018).

$$F - score = \frac{2 \times precision \times recall}{precision + recall}$$

BAB IV HASIL DAN PEMBAHASAN

Bab ini akan membahas tentang hasil uji coba yang telah dilakukan, baik perangkat keras maupun perangkat lunak. Pengujian dilakukan untuk mengetahui apakah sistem yang telah dibangun dapat bekerja dan berfungsi dengan baik atau tidak. Pengujian performa dari metode *K-Nearest Neighbor* yang telah diimplementasikan pada sistem juga akan diuji menggunakan *Confussion Matrix*.

4.1 Hasil dan Uji Coba

4.1.1 Uji coba sensor

Pengujian sensor dilakukan untuk mengetahui rata-rata *error* yang mungkin terjadi. Dari rata-rata *error* tersebut bisa dijadikan acuan untuk melakukan kalibrasi sensor. Pada penelitian ini ada 3 jenis sensor yang diuji akurasi pembacaannya.

Yang pertama adalah sensor *ultrasonic* SR-HC 04. Sensor ini umumnya digunakan untuk mengukur jarak dengan mengandalkan pantulan gelombang yang dikeluarkan. Pada penelitian ini sensor tersebut digunakan untuk mengukur volume air tandon. Untuk menguji akurasi ini digunakan mistar untuk membandingkan hasil pembacaan dengan jarak sesungguhnya. Adapun hasil dari uji coba sensor tersebut sebagaimana pada Tabel 4.1

Tabel 4.1 Tabel uji akurasi sensor *ultrasonic* SR-HC 04

| Pengujian | SR-HC 04 (cm) | Jarak Sebenarnya (cm) | Selisih (cm) | Error (%) |
|-----------|------------------|-----------------------------|--------------|--------------|
| 1 | 3,53 | 3 | 0,53 | 17,67 |
| 2 | 5,3 | 5 | 0,3 | 6,00 |
| 3 | 8,11 | 8 | 0,11 | 1,37 |
| 4 | 10,6 | 10 | 0,6 | 6,00 |
| 5 | 12,7 | 12 | 0,7 | 5,83 |
| 6 | 15,52 | 15 | 0,52 | 3,47 |

| | | | | |
|----------------------------|-------|----|------|-------------|
| 7 | 18,4 | 18 | 0,4 | 2,22 |
| 8 | 20,79 | 20 | 0,79 | 3,95 |
| 9 | 22,14 | 22 | 0,14 | 0,64 |
| 10 | 24,2 | 24 | 0,2 | 0,83 |
| 11 | 26,1 | 26 | 0,1 | 0,38 |
| 12 | 27,8 | 28 | 0,2 | 0,71 |
| 13 | 30,34 | 30 | 0,34 | 1,13 |
| 14 | 32,7 | 32 | 0,7 | 2,19 |
| 15 | 33,37 | 34 | 0,63 | 1,85 |
| Rata-rata Error (%) | | | | 3,62 |

Dari tabel di atas bisa dilihat bahwa sensor SR-HC 04 yang digunakan memiliki rata-rata *error* sebesar 3,62%. Kemudian jika diperhatikan lagi hasil pembacaan sensor selalu lebih tinggi dibandingkan dengan jarak sesungguhnya. Maka untuk melakukan kalibrasi sensor cukup dengan mengurangi hasil pembacaan dengan rata-rata *error* sehingga bisa dituliskan sebagai berikut.

$$\text{kalibrasi} = \text{pembacaan sensor} - (\text{pembacaan sensor} * \text{error})$$

Sensor yang kedua adalah sensor suhu BME-280. Sensor ini merupakan modul sensor yang bisa membaca kelembaban udara, suhu tekanan udara serta ketinggian dari permukaan air laut (*altitude*). Untuk melakukan uji coba akurasi pembacaan suhu digunakan termometer digital. Adapun hasil dari uji coba sensor tersebut sebagaimana pada Tabel 4.2.

Tabel 4.2 Tabel uji akurasi sensor BME-280

| Pengujian | BME-280 (°C) | Thermometer (°C) | Selisih | Error (%) |
|-----------|--------------|------------------|---------|-----------|
| 1 | 26,58 | 28,18 | 1,60 | 5,68 |
| 2 | 26,59 | 27,76 | 1,17 | 4,22 |
| 3 | 26,61 | 27,72 | 1,11 | 4,02 |
| 4 | 26,61 | 27,73 | 1,12 | 4,02 |

| | | | | |
|------------------------|-------|-------|------|-------------|
| 5 | 26,6 | 27,14 | 0,54 | 1,98 |
| 6 | 26,64 | 26,77 | 0,13 | 0,49 |
| 7 | 26,68 | 27,92 | 1,24 | 4,43 |
| 8 | 26,68 | 28,59 | 1,91 | 6,67 |
| 9 | 26,69 | 28,22 | 1,53 | 5,41 |
| 10 | 26,7 | 27,03 | 0,33 | 1,23 |
| 11 | 26,73 | 27,30 | 0,57 | 2,08 |
| 12 | 26,74 | 27,52 | 0,78 | 2,82 |
| 13 | 26,76 | 28,29 | 1,53 | 5,42 |
| 14 | 26,77 | 28,48 | 1,71 | 6,00 |
| 15 | 26,75 | 28,17 | 1,42 | 5,05 |
| Rata-rata Error | | | | 3,97 |

Dari hasil pengujian sensor BME-280 ditemukan rerata error sebesar 3,97%. Untuk mendapatkan hasil yang lebih akurat bisa dilakukan kalibrasi. Kalibrasi sensor bisa dilakukan dengan menjumlahkan rerata error dengan hasil pembacaan sensor karena hasil pembacaan sensor relatif lebih rendah daripada suhu sebenarnya. Adapun persamaannya bisa ditulis sebagaimana:

$$\text{kalibrasi} = \text{pembacaan sensor} + (\text{pembacaan sensor} * \text{error})$$

Sensor ketiga adalah sensor soil moisture. Sensor ini berbentuk probe dan ditancapkan ke tanah. Sensor ini membaca banyaknya kandungan air pada sebuah media tanah. Untuk melakukan ujicoba digunakan digital higrometer. Adapun hasil dari uji coba sensor tersebut sebagaimana pada Tabel 4.3.

Tabel 4.3 Tabel uji akurasi sensor Soil Moisture

| Pengujian | Sensor (%) | Higrometer (%) | Selisih | Error |
|------------------|-------------------|-----------------------|----------------|--------------|
| 1 | 48 | 48,88 | 0,88 | 1,80 |
| 2 | 48,09 | 48,97 | 0,88 | 1,80 |
| 3 | 49,07 | 48,97 | 0,10 | 0,20 |

| | | | | |
|------------------------|-------|-------|------|-------------|
| 4 | 48 | 50,24 | 2,24 | 4,46 |
| 5 | 48,29 | 50,44 | 2,15 | 4,26 |
| 6 | 48,09 | 50,93 | 2,84 | 5,58 |
| 7 | 48,29 | 50,93 | 2,64 | 5,18 |
| 8 | 48,39 | 49,07 | 0,68 | 1,39 |
| 9 | 48,09 | 50,93 | 2,84 | 5,58 |
| 10 | 48,39 | 49,36 | 0,97 | 1,97 |
| 11 | 47,41 | 48,58 | 1,17 | 2,41 |
| 12 | 46,73 | 49,46 | 2,73 | 5,52 |
| 13 | 49,27 | 51,32 | 2,05 | 3,99 |
| 14 | 47,12 | 49,56 | 2,44 | 4,92 |
| 15 | 49,95 | 48,88 | 1,07 | 2,19 |
| Rata-rata Error | | | | 3,42 |

Karena pembacaan sensor relatif lebih rendah, untuk melakukan kalibrasi bisa lakukan menggunakan persamaan berikut:

$$\text{kalibrasi} = \text{pembacaan sensor} - (\text{pembacaan sensor} * \text{error})$$

4.1.2 Pengujian Pompa Air

Pengujian ini dilakukan untuk mengetahui banyaknya air volume air yang disalurkan dalam beberapa detik. Hasil dari pengujian ini nantinya akan digunakan sebagai aturan penentu lama penyiraman untuk setiap klasifikasi. Adapun pengujiannya dilakukan menggunakan *stopwatch* untuk melihat lama waktu yang dibutuhkan untuk menyalurkan 200cc air menggunakan pompa DC dengan tegangan 5V dan arus 1A. Setelah itu akan ditentukan rata-rata laju alirannya. Pada Tabel 4.4 berikut merupakan hasil dari uji coba yang sudah dilakukan.

Tabel 4.4 Tabel laju pompa air 5V

| Percobaan | Volume (cc) | Waktu (s) |
|------------------|-------------|---------------|
| 1 | 200 | 23,30 |
| 2 | 200 | 22,24 |
| 3 | 200 | 21,57 |
| 4 | 200 | 22,43 |
| 5 | 200 | 23,36 |
| 6 | 200 | 22,26 |
| 7 | 200 | 23,43 |
| 8 | 200 | 23,14 |
| 9 | 200 | 22,57 |
| 10 | 200 | 22,34 |
| Rata-rata | | 22,664 |

Dari hasil pengujian pada **tabel 4.4** di atas diketahui bahwa pompa DC yang digunakan memiliki rata-rata 22,664 detik untuk menyalurkan air sebanyak 200cc.

4.1.3 Metode *K-Nearest Neighbor*

Metode *K-Nearest Neighbor* pada penelitian ini digunakan untuk mengklasifikasikan kondisi tanah berdasarkan data kelembaban tanah dan suhu. Data suhu dan kelembaban tanah diambil menggunakan sensor mulai tanggal 9 April – 15 April 2020. Pengambilan data sensor dilakukan setiap satu jam sekali mulai pukul 05.00 – 20.00 WIB, sehingga dalam satu hari mendapatkan 16 data sensor untuk diklasifikasikan, sehingga dalam satu Minggu (7 hari) akan didapatkan 112 data klasifikasi.

Adapun tahapan awal yang dilakukan dalam melakukan proses klasifikasi dengan metode *K-Nearest Neighbor* ini adalah dengan menentukan label kelas klasifikasi. Pada penelitian ini digunakan 4 kelas label dalam mengklasifikasikan keadaan tanah. Pada kelas label ini juga akan

ditentukan durasi waktu penyiraman yang dibutuhkan. Adapun kelas label yang ditentukan sebagaimana pada Tabel 4.5

Tabel 4.5 Tabel label kelas klasifikasi

| Id | Label | Keterangan | Waktu (s) |
|----|-------|-------------|-----------|
| 1 | K | Kering | 100 |
| 2 | AK | Agak kering | 50 |
| 3 | B | Basah | 10 |
| 4 | AB | Agak basah | 20 |

Kelas label pada Tabel 4.5 di atas ditentukan dengan melihat kondisi kelembaban tanah di tempat penelitian yaitu di Dusun Karangbesuki - Kecamatan Sukun - Kota Malang. Dari pengamatan ditemukan bahwa kelembaban tanah terendah adalah 36% dan kelembaban tertinggi 61% dengan 2 kali penyiraman pada jam 07.00 dan 17.00 WIB. Sedangkan indeks rata-rata keseragaman irigasi yang dibutuhkan untuk *strawberry* sendiri adalah sekitar 49% (El-Farhan & Pritts, 1997). Dari data tersebut kemudian ditentukan label klasifikasi dengan *range* kelembaban masing-masing dengan batas tengahnya adalah 50%. Adapun pembagian labelnya adalah, label K mempresentasikan kondisi tanah dengan kelembaban tanah <30%. Label AK mempresentasikan kelembaban tanah >30% dan <50%. Sedangkan Label AB mempresentasikan kondisi tanah dengan kelembaban >50% dan <60%. Label B mempresentasikan kondisi tanah dengan kelembaban >60%.

Sedangkan untuk menentukan lamanya waktu penyiraman dilihat berdasarkan data uji coba pompa air. Dimana untuk menaikkan kelembaban tanah sebanyak 5-7% dibutuhkan air 200cc. Sedangkan pompa air yang digunakan mampu menyalurkan air sebanyak 200cc setiap 22,64 detik. Dengan demikian jika kita ingin meningkatkan tanah dengan kelembaban 30% menjadi $\pm 50\%$ maka dibutuhkan sekitar 20%. Untuk mendapatkan kenaikan kelembaban 20% maka dibutuhkan air sekitar 800-1000cc. Untuk mengalirkan air sebanyak 800-1000cc tersebut dibutuhkan waktu 91-113 detik. Untuk mempermudah penghitungan digunakan nilai bulat antara *range* 91-113 yaitu 100 detik.

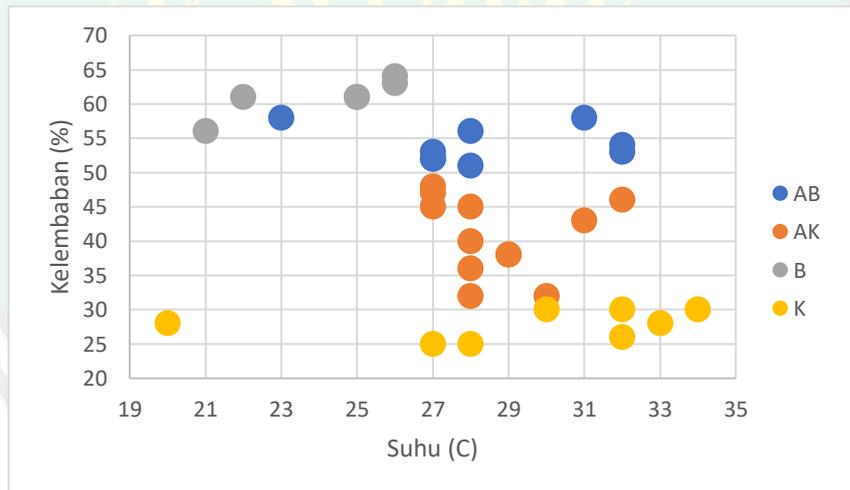
Tahapan selanjutnya yang dilakukan dalam implementasi metode ini adalah dengan menentukan *dataset* yang memiliki kondisi tanah sebenarnya atau data faktual sebenarnya. *Dataset* ini akan dijadikan acuan dalam klasifikasi data-data baru. Pada penelitian ini digunakan sebanyak 35 *dataset* awal sebagaimana pada Tabel 4.6. *Dataset* tersebut diperoleh dengan melihat kondisi tanah sebenarnya.

Tabel 4.6 Tabel *Dataset*

| No. | Suhu (%) | Temperatur (°C) | Klasifikasi |
|-----|----------|-----------------|-------------|
| 1 | 30 | 30 | K |
| 2 | 30 | 34 | K |
| 3 | 25 | 27 | K |
| 4 | 25 | 28 | K |
| 5 | 30 | 32 | K |
| 6 | 28 | 20 | K |
| 7 | 32 | 28 | AK |
| 8 | 40 | 28 | AK |
| 9 | 38 | 29 | AK |
| 10 | 36 | 28 | AK |
| 11 | 47 | 27 | AK |
| 12 | 45 | 28 | AK |
| 13 | 38 | 29 | AK |
| 14 | 32 | 30 | AK |
| 15 | 48 | 27 | AK |
| 16 | 61 | 25 | B |
| 17 | 61 | 25 | B |
| 18 | 61 | 22 | B |
| 19 | 63 | 26 | B |
| 20 | 64 | 26 | B |
| 21 | 52 | 27 | AB |
| 22 | 56 | 28 | AB |

| | | | |
|----|----|----|----|
| 23 | 51 | 28 | AB |
| 24 | 56 | 21 | B |
| 25 | 58 | 23 | AB |
| 26 | 53 | 27 | AB |
| 27 | 52 | 27 | AB |
| 28 | 45 | 27 | AK |
| 29 | 26 | 32 | K |
| 30 | 28 | 33 | K |
| 31 | 54 | 32 | AB |
| 32 | 58 | 31 | AB |
| 33 | 53 | 32 | AB |
| 34 | 46 | 32 | AK |
| 35 | 43 | 31 | AK |

Adapun persebaran dari *dataset* sebagaimana pada Gambar 4.1. Pada gambar tersebut menggambarkan bagaimana persebaran dari *dataset*.



Gambar 4.1 Scatter Plot persebaran *dataset*

Tahap selanjutnya adalah perhitungan menggunakan metode *K-Nearest Neighbor*. Perhitungan dilakukan dengan menghitung jarak terdekat dari data suhu dan kelembaban terhadap masing-masing anggota dari *dataset* menggunakan metode *Euclidean Distance*. Selanjutnya dari jarak terdekat akan diurutkan secara *ascending*. Kemudian dari urutan jarak tersebut akan

diambil 5 teratas ($k=5$), dari 5 urutan tersebut akan dipilih klasifikasi terbanyak. Dari klasifikasi terbanyak tersebut akan dijadikan klasifikasi dari data yang baru dan data tersebut akan ditambahkan ke dalam *dataset*.

Sebagai contoh, diketahui pembacaan sensor suhu dan kelembaban adalah $22,34^{\circ}\text{C}$ dan $43,67\%$ dengan $k=5$. Maka pertama-tama hitung jarak masing-masing anggota *dataset* dengan data baru sebagaimana model persamaan *Euclidean Distance* berikut:

$$d_i = \sqrt{(h_{\text{new}} - h_i)^2 + (t_{\text{new}} - t_i)^2}$$

Keterangan:

d_i = Jarak ke- i

h_{new} = data kelembaban baru

h_i = kelembaban i pada *dataset*

t_{new} = data suhu baru

t_i = suhu i pada *dataset*

Sehingga:

$$\begin{aligned} \text{Jarak ke-1.....} \quad d_1 &= \sqrt{(43,67 - 30)^2 + (22,34 - 30)^2} \\ d_1 &= 15,6698596 \end{aligned}$$

$$\begin{aligned} \text{Jarak ke-2.....} \quad d_2 &= \sqrt{(43,67 - 30)^2 + (22,34 - 34)^2} \\ d_2 &= 17,96731755 \end{aligned}$$

$$\begin{aligned} \text{Jarak ke-3.....} \quad d_3 &= \sqrt{(43,67 - 25)^2 + (22,34 - 27)^2} \\ d_3 &= 19,24277787 \end{aligned}$$

$$\begin{aligned} \text{Jarak ke-35.....} \quad d_{35} &= \sqrt{(43,67 - 43)^2 + (22,34 - 31)^2} \\ d_{35} &= 8,685879345 \end{aligned}$$

Pada Tabel 4.7 berikut merupakan hasil perhitungan jarak yang sudah diurutkan.

Tabel 4.7 Tabel Sort *Euclidean Distance*

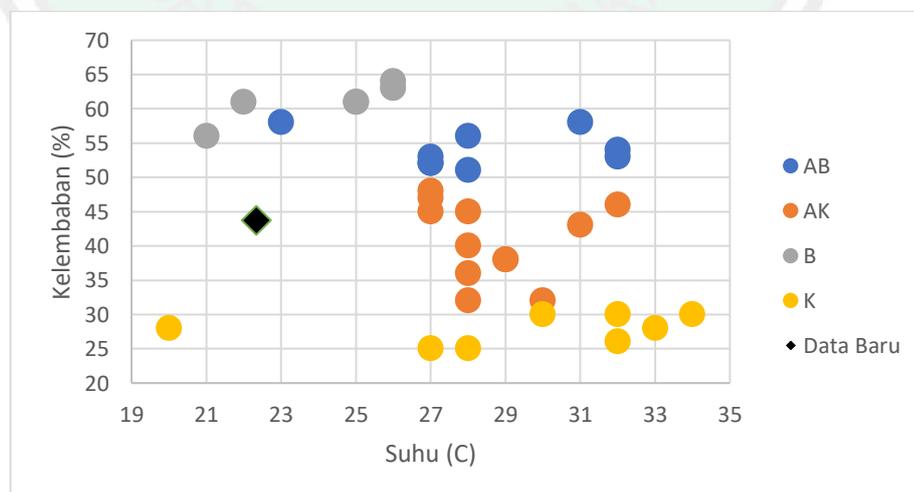
| Kelembaban (%) | Suhu (°C) | Klasifikasi | <i>Euclidean distance</i> |
|-----------------------|------------------|--------------------|----------------------------------|
| 45 | 27 | AK | 4,846081 |
| 47 | 27 | AK | 5,727521 |
| 45 | 28 | AK | 5,814164 |
| 48 | 27 | AK | 6,361171 |
| 40 | 28 | AK | 6,745702 |
| 43 | 31 | AK | 8,685879 |
| 38 | 29 | AK | 8,746685 |
| 38 | 29 | AK | 8,746685 |
| 51 | 28 | AB | 9,260912 |
| 36 | 28 | AK | 9,532287 |
| 52 | 27 | AB | 9,544868 |
| 52 | 27 | AB | 9,544868 |
| 46 | 32 | AK | 9,937027 |
| 53 | 27 | AB | 10,42902 |
| 56 | 21 | B | 12,4026 |
| 32 | 28 | AK | 12,97014 |
| 53 | 32 | AB | 13,42999 |
| 56 | 28 | AB | 13,56704 |
| 32 | 30 | AK | 13,95939 |
| 54 | 32 | AB | 14,143 |
| 58 | 23 | AB | 14,34519 |
| 30 | 30 | K | 15,66986 |
| 28 | 20 | K | 15,84375 |
| 30 | 32 | K | 16,73871 |
| 58 | 31 | AB | 16,74349 |
| 61 | 22 | B | 17,33333 |
| 61 | 25 | B | 17,53295 |
| 61 | 25 | B | 17,53295 |

| | | | |
|----|----|---|----------|
| 30 | 34 | K | 17,96732 |
| 28 | 33 | K | 18,95216 |
| 25 | 27 | K | 19,24278 |
| 25 | 28 | K | 19,50909 |
| 63 | 26 | B | 19,67345 |
| 26 | 32 | K | 20,13814 |
| 64 | 26 | B | 20,65683 |

Setelah setiap jarak dihitung dan diurutkan maka langkah selanjutnya adalah melihat tetangga terdekat sebanyak k . Pada penelitian ini digunakan $k=5$, sehingga hasilnya sebagaimana pada Tabel 4.8. Dari tabel tersebut bisa dilihat klasifikasi terbanyak adalah AK dengan tetangga sebanyak 5. Sehingga hasil klasifikasi dari data suhu dan kelembaban $22,34^{\circ}\text{C}$ dan $43,67\%$ adalah AK.

Tabel 4.8 Tabel Tetangga Terdekat

| Kelembaban (%) | Suhu ($^{\circ}\text{C}$) | Klasifikasi | Euclidean distance |
|----------------|-----------------------------|-------------|--------------------|
| 45 | 27 | AK | 4,846081 |
| 47 | 27 | AK | 5,727521 |
| 45 | 28 | AK | 5,814164 |
| 48 | 27 | AK | 6,361171 |
| 40 | 28 | AK | 6,745702 |



Gambar 4.2 Scatter Plot persebaran baru

Pada Gambar 4.2 di atas menggambarkan persebaran data baru terhadap *dataset*. Jika dilihat data baru tersebut cenderung dekat dengan klasifikasi AK.

4.1.4 Pengujian metode *K-Nearest Neighbor*

Pengujian performa dari model yang dibangun dilakukan dengan menggunakan *confussion matrix* dengan 4 kelas. Data yang digunakan adalah data hasil klasifikasi selama 7 hari mulai tanggal 9 April – 15 April 2020 sebanyak 145 data klasifikasi. Data klasifikasi tersebut dibagi secara acak sebanyak 60% : 40% yang berturut-turut digunakan sebagai data latih dan data test. Sehingga didapat 87 data latih dan 58 data tes. Adapun data latih dan data *test* masing-masing disajikan pada Tabel 4.9 dan Tabel 4.10.

Tabel 4.9 Tabel Data Latih

| Id | Kelembaban (%) | Suhu (°C) | Klasifikasi |
|-----------|-----------------------|------------------|--------------------|
| 1 | 30 | 30 | 1 |
| 2 | 25 | 28 | 1 |
| 3 | 28 | 20 | 1 |
| 4 | 32 | 28 | 2 |
| 5 | 40 | 28 | 2 |
| 6 | 36 | 28 | 2 |
| 7 | 47 | 27 | 2 |
| 8 | 38 | 29 | 2 |
| 9 | 32 | 30 | 2 |
| 10 | 61 | 25 | 3 |
| 11 | 63 | 26 | 3 |
| 12 | 52 | 27 | 4 |
| 13 | 56 | 28 | 4 |
| 14 | 56 | 21 | 3 |
| 15 | 58 | 23 | 4 |
| 16 | 52 | 27 | 4 |

| | | | |
|----|-------|-------|---|
| 17 | 45 | 27 | 2 |
| 18 | 54 | 32 | 4 |
| 19 | 53 | 32 | 4 |
| 20 | 46 | 32 | 2 |
| 21 | 43 | 31 | 2 |
| 22 | 36,5 | 28,48 | 2 |
| 23 | 37,11 | 29,4 | 2 |
| 24 | 37,35 | 28,15 | 2 |
| 25 | 37,86 | 29,11 | 2 |
| 26 | 37,88 | 26,61 | 2 |
| 27 | 38,11 | 26,33 | 2 |
| 28 | 38,16 | 26,27 | 2 |
| 29 | 38,6 | 27,67 | 2 |
| 30 | 38,66 | 25,74 | 2 |
| 31 | 38,81 | 30,67 | 2 |
| 32 | 38,91 | 25,16 | 2 |
| 33 | 39,01 | 25,37 | 2 |
| 34 | 39,25 | 30,5 | 2 |
| 35 | 39,3 | 25,06 | 2 |
| 36 | 39,49 | 27,33 | 2 |
| 37 | 39,55 | 24,8 | 2 |
| 38 | 39,59 | 27,29 | 2 |
| 39 | 39,69 | 31,1 | 2 |
| 40 | 39,78 | 24,55 | 2 |
| 41 | 39,91 | 24,41 | 2 |
| 42 | 39,96 | 29,84 | 2 |
| 43 | 39,97 | 30,24 | 2 |
| 44 | 40,09 | 24,22 | 2 |
| 45 | 40,37 | 27,76 | 2 |
| 46 | 40,45 | 27,73 | 2 |
| 47 | 40,5 | 23,78 | 2 |

| | | | |
|----|-------|-------|---|
| 48 | 40,7 | 29,94 | 2 |
| 49 | 40,95 | 27,92 | 2 |
| 50 | 41,33 | 29,7 | 2 |
| 51 | 41,55 | 30 | 2 |
| 52 | 41,87 | 30,84 | 2 |
| 53 | 42,07 | 30,57 | 2 |
| 54 | 42,56 | 28,07 | 2 |
| 55 | 42,86 | 27,57 | 2 |
| 56 | 43,38 | 30,45 | 2 |
| 57 | 43,74 | 29,54 | 2 |
| 58 | 44,33 | 28,16 | 2 |
| 59 | 44,38 | 28,14 | 2 |
| 60 | 44,49 | 27,33 | 2 |
| 61 | 45,23 | 26,31 | 2 |
| 62 | 45,78 | 30,19 | 2 |
| 63 | 45,84 | 28,35 | 2 |
| 64 | 46,24 | 26,85 | 2 |
| 65 | 46,73 | 28,39 | 2 |
| 66 | 47,31 | 26,89 | 2 |
| 67 | 47,87 | 27,18 | 2 |
| 68 | 47,95 | 27,92 | 2 |
| 69 | 47,96 | 29,07 | 2 |
| 70 | 48,52 | 27,7 | 2 |
| 71 | 48,71 | 27,63 | 2 |
| 72 | 49,41 | 27,36 | 4 |
| 73 | 49,79 | 26,61 | 2 |
| 74 | 50,29 | 27,02 | 4 |
| 75 | 50,67 | 27,36 | 4 |
| 76 | 51,02 | 29 | 4 |
| 77 | 51,2 | 26,67 | 4 |
| 78 | 51,7 | 27,1 | 4 |

| | | | |
|----|-------|-------|---|
| 79 | 51,82 | 26,43 | 4 |
| 80 | 52,19 | 26,29 | 4 |
| 81 | 53,61 | 26,51 | 4 |
| 82 | 54,46 | 27,37 | 4 |
| 83 | 54,92 | 27,22 | 4 |
| 84 | 55,64 | 25,99 | 4 |
| 85 | 55,72 | 27,11 | 4 |
| 86 | 56,6 | 27,8 | 4 |
| 87 | 57,05 | 27,54 | 4 |

Tabel 4.10 Tabel Data *Test*

| Id | Kelembaban (%) | Suhu (°C) | Klasifikasi/ Aktual |
|-----------|-----------------------|------------------|--------------------------------|
| 1 | 30 | 32 | 1 |
| 2 | 61 | 22 | 3 |
| 3 | 36,61 | 28,48 | 2 |
| 4 | 38,34 | 29,85 | 2 |
| 5 | 39,65 | 27,65 | 2 |
| 6 | 42,86 | 28,34 | 2 |
| 7 | 38,99 | 25,39 | 2 |
| 8 | 45 | 28 | 2 |
| 9 | 51 | 28 | 4 |
| 10 | 30 | 34 | 1 |
| 11 | 53 | 27 | 4 |
| 12 | 64 | 26 | 3 |
| 13 | 58 | 31 | 4 |
| 14 | 37,85 | 29,59 | 2 |
| 15 | 39,96 | 24,36 | 2 |
| 16 | 40,71 | 27,63 | 2 |
| 17 | 40,42 | 30,82 | 2 |
| 18 | 38 | 29 | 2 |

| | | | |
|----|-------|-------|---|
| 19 | 44,06 | 25,14 | 2 |
| 20 | 45,56 | 28,84 | 2 |
| 21 | 51,66 | 27,08 | 4 |
| 22 | 48,23 | 30,18 | 2 |
| 23 | 54,29 | 25,48 | 4 |
| 24 | 47,29 | 26,88 | 2 |
| 25 | 49,98 | 27,14 | 4 |
| 26 | 55,7 | 26,56 | 4 |
| 27 | 39,39 | 24,97 | 2 |
| 28 | 38,67 | 25,73 | 2 |
| 29 | 42,05 | 30,19 | 2 |
| 30 | 46,11 | 26,66 | 2 |
| 31 | 48 | 27 | 2 |
| 32 | 39,86 | 29,11 | 2 |
| 33 | 45,02 | 28,28 | 2 |
| 34 | 52,55 | 26,15 | 4 |
| 35 | 37,99 | 27,52 | 2 |
| 36 | 42,34 | 30,08 | 2 |
| 37 | 48,86 | 25,57 | 2 |
| 38 | 41,34 | 30,37 | 2 |
| 39 | 50,93 | 27,76 | 4 |
| 40 | 42,89 | 30,34 | 2 |
| 41 | 37,21 | 28,36 | 2 |
| 42 | 38,85 | 25,54 | 2 |
| 43 | 40,32 | 30,28 | 2 |
| 44 | 61 | 25 | 3 |
| 45 | 38,96 | 25,42 | 2 |
| 46 | 40,55 | 23,73 | 2 |
| 47 | 50,07 | 26,72 | 4 |
| 48 | 53,05 | 27,29 | 4 |
| 49 | 45,19 | 27,83 | 2 |

| | | | |
|----|-------|-------|---|
| 50 | 26 | 32 | 1 |
| 51 | 28 | 33 | 1 |
| 52 | 51,54 | 26,54 | 4 |
| 53 | 25 | 27 | 1 |
| 54 | 39,64 | 24,7 | 2 |
| 55 | 37,37 | 30,07 | 2 |
| 56 | 38,45 | 29,96 | 2 |
| 57 | 42,33 | 31,24 | 2 |
| 58 | 47,94 | 26,69 | 2 |

Pada Tabel 4.9 dan Tabel 4.10 tersebut label klasifikasi disimbolkan dengan id dari label klasifikasi sebagaimana Tabel 4.5 di atas. Klasifikasi pada *test* akan digunakan sebagai klasifikasi aktual atau yang sebenarnya. Kemudian data *test* akan membandingkan hasil prediksi yang diperoleh dari perhitungan dengan data latih dengan nilai aktual tersebut. Sehingga diperoleh perbandingan sebagaimana pada Tabel 4.11. Pada tabel tersebut hasil prediksi yang berbeda dengan klasifikasi sebenarnya/ aktual ditandai dengan *shading* berwarna merah.

Tabel 4.11 Tabel Perbandingan Hasil Prediksi dan Klasifikasi Aktual

| Id | Kelembaban (%) | Suhu (°C) | Hasil Prediksi | Klasifikasi/ Aktual | Keterangan |
|-----------|-----------------------|------------------|-----------------------|----------------------------|-------------------|
| 1 | 30 | 32 | 2 | 1 | BEDA |
| 2 | 28 | 33 | 2 | 1 | BEDA |
| 3 | 26 | 32 | 2 | 1 | BEDA |
| 4 | 36.61 | 28.48 | 2 | 2 | SAMA |
| 5 | 25 | 27 | 1 | 1 | SAMA |
| 6 | 30 | 34 | 2 | 1 | BEDA |
| 7 | 38.34 | 29.85 | 2 | 2 | SAMA |
| 8 | 39.65 | 27.65 | 2 | 2 | SAMA |
| 9 | 42.86 | 28.34 | 2 | 2 | SAMA |

| | | | | | |
|----|-------|-------|---|---|------|
| 10 | 38.99 | 25.39 | 2 | 2 | SAMA |
| 11 | 37.85 | 29.59 | 2 | 2 | SAMA |
| 12 | 45 | 28 | 2 | 2 | SAMA |
| 13 | 39.96 | 24.36 | 2 | 2 | SAMA |
| 14 | 40.71 | 27.63 | 2 | 2 | SAMA |
| 15 | 40.42 | 30.82 | 2 | 2 | SAMA |
| 16 | 38 | 29 | 2 | 2 | SAMA |
| 17 | 44.06 | 25.14 | 2 | 2 | SAMA |
| 18 | 45.56 | 28.84 | 2 | 2 | SAMA |
| 19 | 48.23 | 30.18 | 2 | 2 | SAMA |
| 20 | 47.29 | 26.88 | 2 | 2 | SAMA |
| 21 | 39.39 | 24.97 | 2 | 2 | SAMA |
| 22 | 38.67 | 25.73 | 2 | 2 | SAMA |
| 23 | 42.05 | 30.19 | 2 | 2 | SAMA |
| 24 | 46.11 | 26.66 | 2 | 2 | SAMA |
| 25 | 48 | 27 | 2 | 2 | SAMA |
| 26 | 39.86 | 29.11 | 2 | 2 | SAMA |
| 27 | 45.02 | 28.28 | 2 | 2 | SAMA |
| 28 | 37.99 | 27.52 | 2 | 2 | SAMA |
| 29 | 42.34 | 30.08 | 2 | 2 | SAMA |
| 30 | 48.86 | 25.57 | 2 | 2 | SAMA |
| 31 | 41.34 | 30.37 | 2 | 2 | SAMA |
| 32 | 42.89 | 30.34 | 2 | 2 | SAMA |
| 33 | 37.21 | 28.36 | 2 | 2 | SAMA |
| 34 | 38.85 | 25.54 | 2 | 2 | SAMA |
| 35 | 40.32 | 30.28 | 2 | 2 | SAMA |
| 36 | 40.55 | 23.73 | 2 | 2 | SAMA |
| 37 | 38.96 | 25.42 | 2 | 2 | SAMA |
| 38 | 45.19 | 27.83 | 2 | 2 | SAMA |
| 39 | 39.64 | 24.7 | 2 | 2 | SAMA |
| 40 | 37.37 | 30.07 | 2 | 2 | SAMA |

| | | | | | |
|----|-------|-------|---|---|------|
| 41 | 38.45 | 29.96 | 2 | 2 | SAMA |
| 42 | 47.94 | 26.69 | 2 | 2 | SAMA |
| 43 | 42.33 | 31.24 | 2 | 2 | SAMA |
| 44 | 61 | 22 | 3 | 3 | SAMA |
| 45 | 64 | 26 | 4 | 3 | BEDA |
| 46 | 61 | 25 | 4 | 3 | BEDA |
| 47 | 51 | 28 | 4 | 4 | SAMA |
| 48 | 53 | 27 | 4 | 4 | SAMA |
| 49 | 58 | 31 | 4 | 4 | SAMA |
| 50 | 51.66 | 27.08 | 4 | 4 | SAMA |
| 51 | 54.29 | 25.48 | 4 | 4 | SAMA |
| 52 | 49.98 | 27.14 | 4 | 4 | SAMA |
| 53 | 55.7 | 26.56 | 4 | 4 | SAMA |
| 54 | 52.55 | 26.15 | 4 | 4 | SAMA |
| 55 | 50.93 | 27.76 | 4 | 4 | SAMA |
| 56 | 50.07 | 26.72 | 4 | 4 | SAMA |
| 57 | 53.05 | 27.29 | 4 | 4 | SAMA |
| 58 | 51.54 | 26.54 | 4 | 4 | SAMA |

Dari data uji coba tersebut bisa dilakukan pengujian menggunakan *confussion matrix*. Adapun tabel *confussion matrix* pada penelitian ini sebagaimana pada Tabel 4.12.

Tabel 4.12 Tabel Confussion Matrix Data Uji coba

| | | Expected | | | |
|-----------|----|----------|----|---|----|
| | | K | AK | B | AB |
| Predicted | K | 1 | 0 | 0 | 0 |
| | AK | 4 | 38 | 0 | 0 |
| | B | 0 | 0 | 1 | 0 |
| | AB | 0 | 0 | 2 | 12 |
| | | | | | |

Dari tabel *confussion matrix* tersebut bisa ditentukan *preccission*, *recall* dan *f1-score* untuk setiap kelas. Pada *confussion matrix multi class True Positive* adalah *cell* dengan nilai *expected* dan *predicted* yang sama. Sedangkan untuk *False Negative* adalah kolom dari *cell True Positive*. *False Positif* adalah baris dari *cell True Positive*, sedangkan *True negative* adalah bagian yang bukan *True Positive*, *False Postive* dan *false Negative*. Untuk menghitung nilai dari *preccission* setiap kelas adalah dengan menggunakan persamaan:

$$preccission_{class} = \frac{TP_{class}}{\Sigma FN_{class}} * 100$$

$$preccission_K = \frac{1}{(1 + 0 + 0 + 0)} * 100 = 100$$

$$preccission_{AK} = \frac{38}{(4 + 38 + 0 + 0)} * 100 = 90,48$$

$$preccission_B = \frac{1}{(1 + 0 + 0 + 0)} * 100 = 100$$

$$preccission_{AB} = \frac{12}{(1 + 0 + 2 + 12)} * 100 = 85,71$$

Untuk menghitung nilai dari *recall* setiap kelas adalah dengan menggunakan persamaan:

$$recall_{class} = \frac{TP_{class}}{\Sigma FP_{class}} * 100$$

$$recall_K = \frac{1}{(1 + 4 + 0 + 0)} * 100 = 20$$

$$recall_{AK} = \frac{38}{(0 + 38 + 0 + 0)} * 100 = 100$$

$$recall_B = \frac{1}{(0 + 0 + 1 + 2)} * 100 = 33,33$$

$$recall_{AB} = \frac{12}{(0 + 0 + 0 + 12)} * 100 = 100$$

Untuk menghitung nilai dari *f1-score* setiap kelas adalah dengan menggunakan persamaan:

$$f1 - score_{class} = 2 * \frac{(precision_{class} * precision_{class})}{(precision_{class} + precision_{class})}$$

$$f1 - score_K = 2 * \frac{(100 * 20)}{(100 + 20)} = 33,33$$

$$f1 - score_{AK} = 2 * \frac{(90,48 * 100)}{(90,48 + 100)} = 95$$

$$f1 - score_B = 2 * \frac{(100 * 33,33)}{(100 + 33,33)} = 50$$

$$f1 - score_{AB} = 2 * \frac{(85,71 * 33,33)}{(85,71 + 33,33)} = 92,31$$

Selanjutnya adalah menghitung nilai makro *average* dari *precision*, *recall* dan *f1* dari setiap kelas, adapun persamaan yang digunakan adalah sebagai berikut:

$$Makro = \frac{(n_K + n_{AK} + n_B + n_{AB})}{4}$$

$$Macro\ precision = \frac{(100 + 90,48 + 100 + 85,71)}{4} = 94,05\%$$

$$Macro\ recall = \frac{(20 + 100 + 33,33 + 100)}{4} = 63,3\%$$

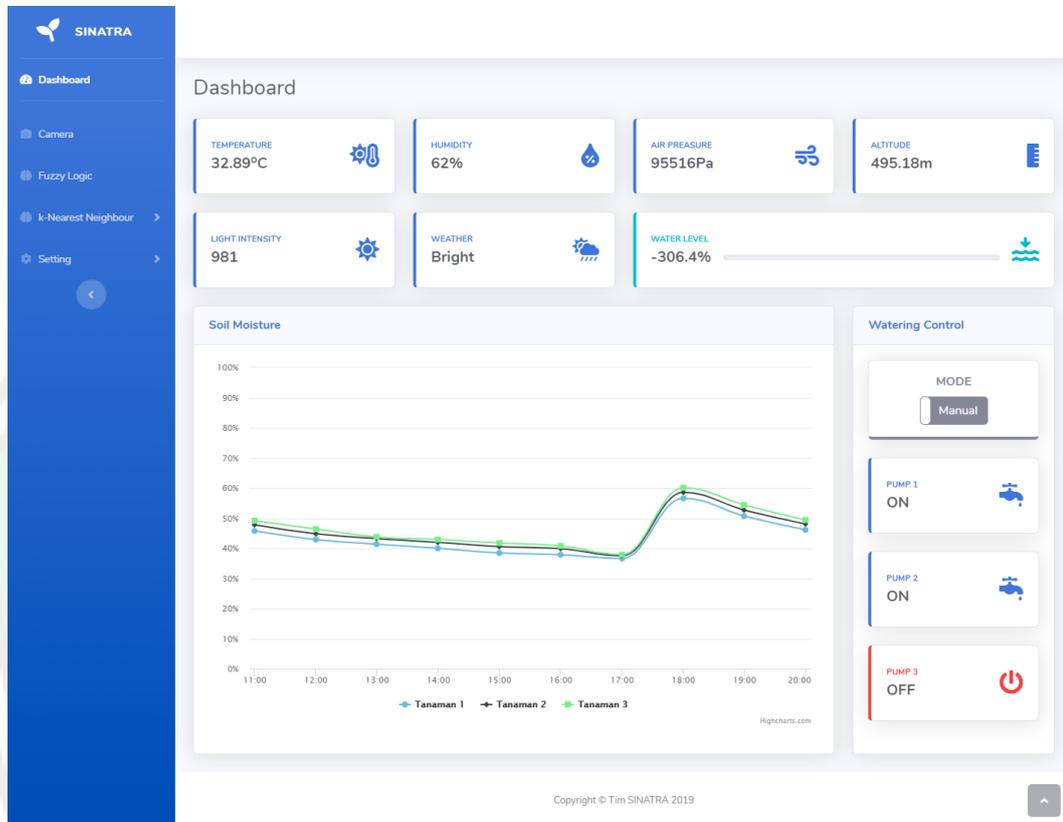
$$Macro\ f1 = \frac{(33,33 + 95 + 50 + 92,31)}{4} = 67,7\%$$

4.2 Implementasi Antarmuka

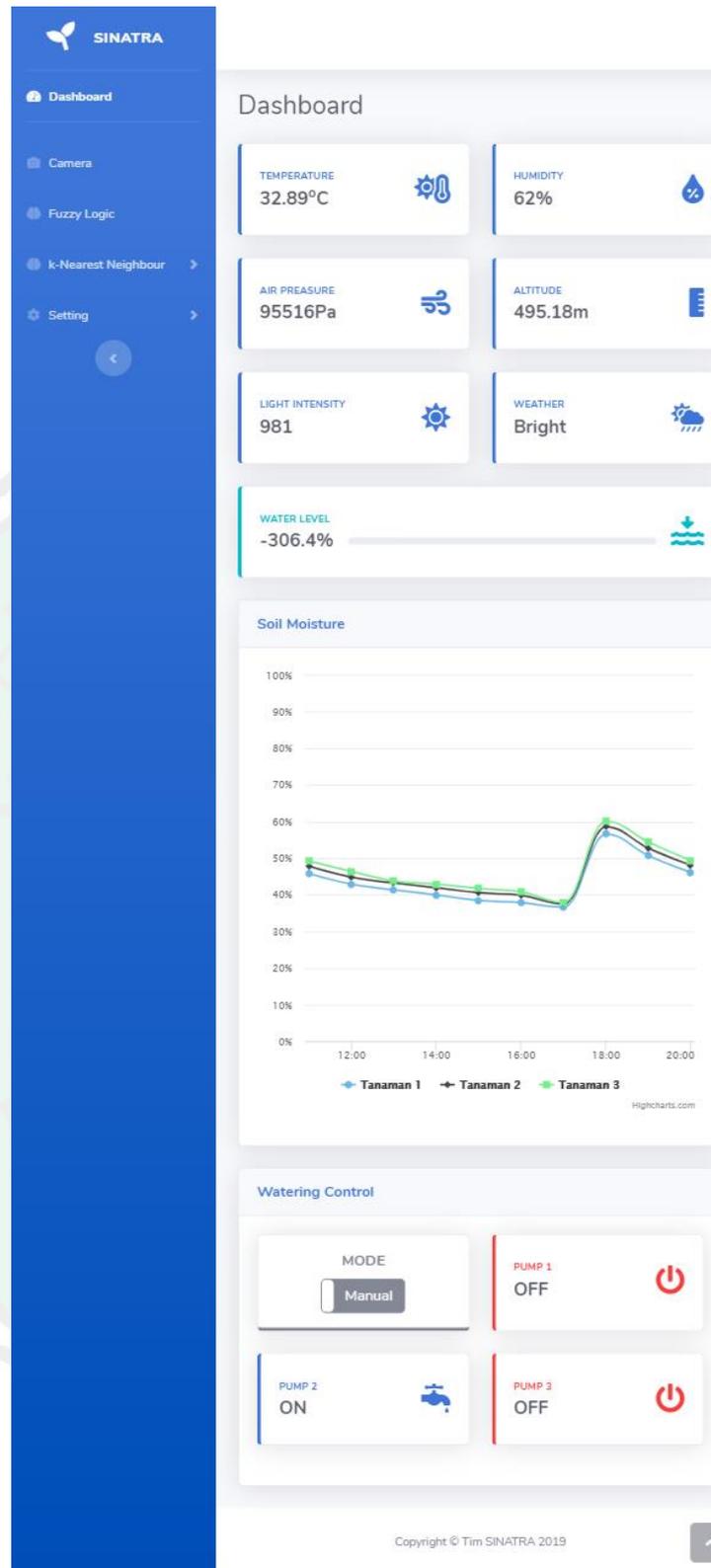
Antarmuka berhasil dibangun menggunakan PHP *framework* Code Igniter dan CSS *framework* Bootstrap. *System interface* yang telah dibangun sudah *responsive* sehingga dapat digunakan di berbagai *device* dengan resolusi berbeda. Sedangkan untuk pengambilan data di *interface* ini dilakukan dengan akses REST API yang sudah dibangun.

4.2.1 Halaman Dashboard

Halaman *dashboard* adalah halaman utama dari *system interface*. Pada halaman ini menyajikan data dari setiap sensor secara *real-time*. Terdapat grafik kelembaban tanah yang diambil setiap satu jam. Sedangkan pada sisi kanan terdapat tombol kontrol pompa air yang bisa diubah mode aktivasinya.



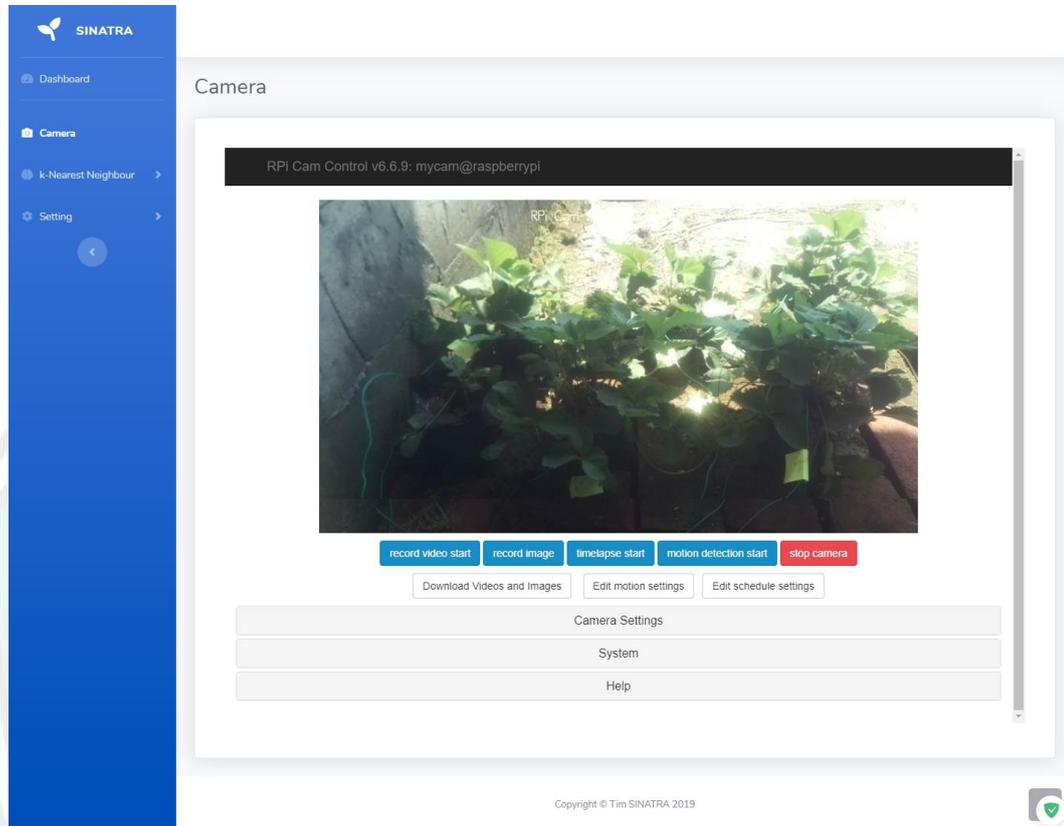
Gambar 4.3 Halaman *Dashboard* tampilan pada komputer



Gambar 4.4 Halaman *Dashboard* tampilan pada tablet dan *mobile*

4.2.2 Halaman *Camera*

Halaman ini merupakan halaman untuk mengakses kamera yang terpasang di server. Pada halaman ini bisa dilakukan *setting* pada kamera, perekaman video dan mengambil gambar.



Gambar 4.5 Halaman *camera*

4.2.3 Halaman *K-Nearest Neighbor*

Pada halaman menampilkan *dataset*, data klasifikasi, data label klasifikasi, dan halaman untuk melakukan perhitungan KNN secara manual.

Calculate

Temperature (°C) Soil Moisture (%)

Result

| Temperature (°C) | Soil Moisture % | Klasifikasi | Neighbor(s) |
|------------------|-----------------|-------------|-------------|
| 22 | 54 | AB | 4 |

Neighbor

Show entries Search:

| No | Temperature (°C) | Soil Moisture % | Klasifikasi | Distance |
|----|------------------|-----------------|-------------------------------------|--------------------|
| 1 | 21 | 56 | <input type="button" value="✖ B"/> | 2.23606797749979 |
| 2 | 23 | 58 | <input type="button" value="✔ AB"/> | 4.123105625617661 |
| 3 | 27 | 53 | <input type="button" value="✔ AB"/> | 5.0990195135927845 |
| 4 | 27 | 52 | <input type="button" value="✔ AB"/> | 5.385164807134504 |
| 5 | 27 | 52 | <input type="button" value="✔ AB"/> | 5.385164807134504 |

Showing 1 to 5 of 5 entries

Copyright © Tim SINATRA 2019

Gambar 4.6 Halaman *Calculate* KNN

Dataset

Show entries Search:

| No | Temperature (°C) | Soil Moisture (%) | Classification | Action |
|----|------------------|-------------------|----------------|---|
| 1 | 30 | 30 | K | <input type="checkbox"/> <input type="checkbox"/> |
| 2 | 34 | 30 | K | <input type="checkbox"/> <input type="checkbox"/> |
| 3 | 27 | 25 | K | <input type="checkbox"/> <input type="checkbox"/> |
| 4 | 28 | 25 | K | <input type="checkbox"/> <input type="checkbox"/> |
| 5 | 32 | 30 | K | <input type="checkbox"/> <input type="checkbox"/> |
| 6 | 20 | 28 | K | <input type="checkbox"/> <input type="checkbox"/> |
| 7 | 28 | 32 | AK | <input type="checkbox"/> <input type="checkbox"/> |
| 8 | 28 | 40 | AK | <input type="checkbox"/> <input type="checkbox"/> |
| 9 | 29 | 38 | AK | <input type="checkbox"/> <input type="checkbox"/> |
| 10 | 28 | 36 | AK | <input type="checkbox"/> <input type="checkbox"/> |

Showing 1 to 10 of 35 entries

Copyright © Tim SINATRA 2019

Gambar 4.7 Halaman *Dataset*

Copyright © Tim SINATRA 2019

Gambar 4. 8 Halaman Label

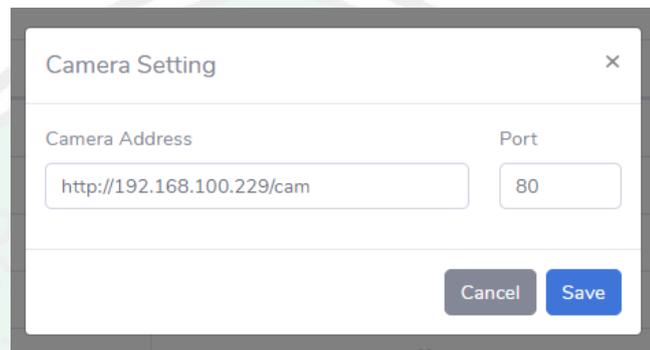
Pada halaman label, *dataset* dan data klasifikasi bisa dilakukan CRUD setiap data.

Gambar 4.9 Modal Tambah Data

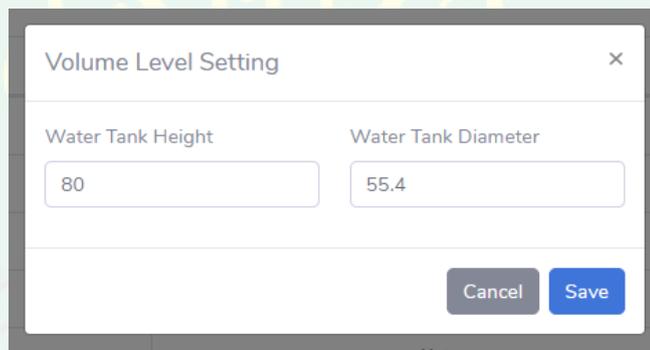
Gambar 4.10 Modal Edit Data

4.2.4 Menu *Setting*

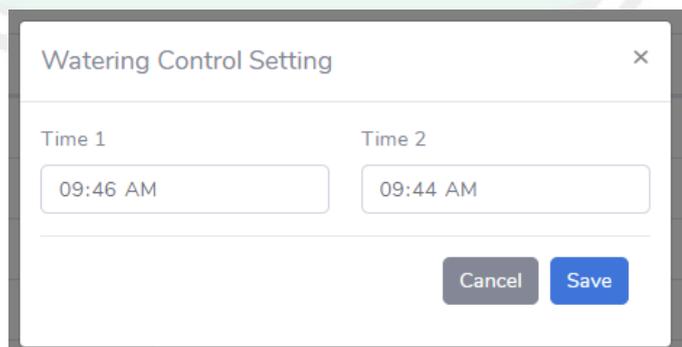
Pada sistem yang dibangun terdapat 3 macam *setting*. *Setting Camera* berfungsi untuk mengubah *hostname* dan PORT kamera. *Water volume setting* digunakan untuk mengubah ukuran dari penampungan/ tandon air penyiraman. *Watering control setting* digunakan untuk menentukan waktu penyiraman. *Setting* ditampilkan menggunakan modal.



Gambar 4.11 Modal *Camera Setting*



Gambar 4.12 Modal *Water Level Setting*



Gambar 4.13 Modal *Watering Control Setting*

4.3 Implementasi REST Service

Untuk mempermudah pengembangan di masa depan dan untuk mengatasi keamanan agar *client* tidak bisa akses langsung ke *database*, maka dibangun REST service pada sistem ini. Semua data request akan ditangani oleh REST server sesuai dengan *method* yang sudah dibuat. Kemudian REST server akan mengirimkan *response* berupa data JSON. Adapun daftar *method* dan parameternya adalah sebagaimana **tabel 4.13**.

Tabel 4.13 Tabel Daftar REST *method*

| Method | Parameter | Keterangan |
|--------------------------|---|--|
| /api/sensor | | |
| GET | - | Menampilkan seluruh data sensor |
| POST | dt: datetime (opsional) nama: nama sensor moisture: sensor kelembaban temperature: sensor suhu | Input data sensor baru |
| api/dataset | | |
| GET | - | Menampilkan seluruh dataset |
| POST | moisture: sensor kelembaban temperature: sensor suhu ide_label: id label klasifikasi | Input dataset baru |
| DELETE | id: id dataset | Menghapus dataset dengan ID tertentu |
| PUT | ide: id dataset moisture: sensor kelembaban temperature: sensor suhu ide_label: id label klasifikasi | Mengubah dataset dengan id tertentu |
| api/dataset/label | | |
| GET | - | Menampilkan seluruh data label klasifikasi |

| | | |
|--------------------------------|---|--|
| POST | label: nama label keterangan: keterangan label (optional) time: waktu penyiraman | Input data label klasifikasi |
| DELETE | id: id label | Menghapus dataset dengan ID tertentu |
| PUT | id: id label label: nama label keterangan: keterangan label (optional) time: waktu penyiraman | Mengubah data label klasifikasi dengan id tertentu |
| api/dataset/klasifikasi | | |
| GET | - | Menampilkan seluruh dataset |
| POST | moisture: sensor kelembaban temperature: sensor suhu id_label: id label klasifikasi | Input data klasifikasi baru |
| DELETE | id: id dataset | Menghapus data klasifikasi dengan ID tertentu |
| PUT | id: id dataset moisture: sensor kelembaban temperature: sensor suhu id_label: id label klasifikasi | Mengubah data klasifikasi dengan id tertentu |
| api/sens | | |
| GET | - | Menampilkan semua data sensor dashboard |
| POST | dt: datetime (optional) water_level: ketinggian air air_pressure: tekanan udara rain: sensor hujan light: sensor cahaya temperature: sensor suhu | Insert data baru |

| | | |
|---------------------------|--|---|
| | humidity: sensor kelembaban altitude: ketinggian | |
| PUT | id: id data sensor dashboard dt: datetime (optional) water_level: ketinggian air air_pressure: tekanan udara rain: sensor hujan light: sensor cahaya temperature: sensor suhu humidity: sensor kelembaban altitude: ketinggian | Mengubah data dengan id tertentu |
| api/pump | | |
| GET | - | Menampilkan semua data pompa air |
| PUT | id: ide pompa air status: status aktivasi pompa air | Mengubah data pompa air dengan ide tertentu |
| api/setting/camera | | |
| GET | - | Menampilkan setting kamera |
| PUT | id: id setting port: port camera host: lokasi hostname | Mengubah setting camera |
| api/setting/volume | | |
| GET | - | Menampilkan setting volume level |
| PUT | id: id setting tinggi: tinggi penampung diameter: diameter penampung | Mengubah setting water_level |
| api/setting/pump | | |
| GET | - | Menampilkan setting volume pompa air |
| PUT | id: id setting time1: waktu penyiraman 1 | Mengubah setting penyiraman |

| | | |
|----------------|---|---|
| | time: waktu penyiraman 1 | |
| api/knn | | |
| GET | moisture: data kelembaban temperature: data suhu k: tetangga terdekat (default 5) | Melakukan perhitungan KNN tanpa menyimpan hasil klasifikasi dan menampilkan nya |
| POST | moisture: data kelembaban temperature: data suhu k: tetangga terdekat (default 5) | Melakukan perhitungan KNN dan menyimpan hasil klasifikasi ke database |

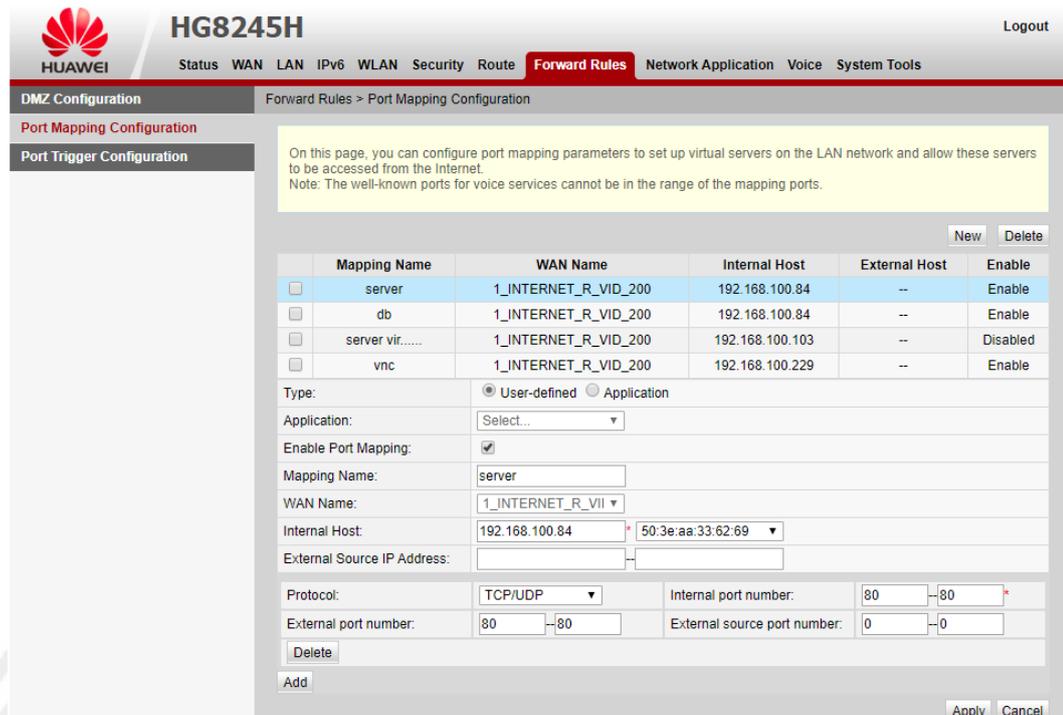
4.4 Implementasi Hardware

Pada penelitian ini telah berhasil dibangun sebuah perangkat *internet of things* dengan mengandalkan Raspberry Pi 3 B+ sebagai *base server*. Untuk pengambilan sensor dan penyiraman digunakan 2 buah NodeMCU ESP8266 sebagai node. Masing-masing NodeMCU ESP8266 dan Raspberry Pi 3 B+ tersebut dihubungkan melalui sebuah jaringan internet untuk saling berkomunikasi.

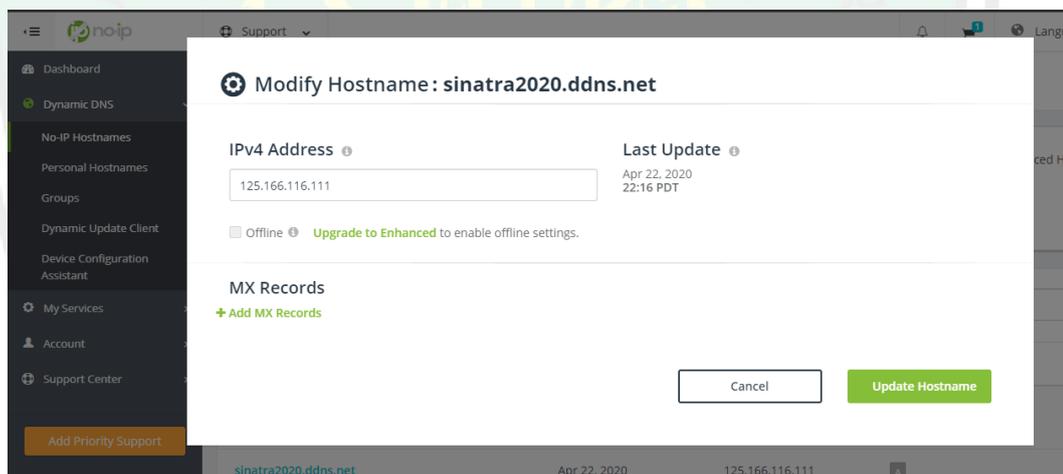
4.4.1 Server (Raspberry Pi 3 B+)

Raspberry Pi 3 B+ telah berhasil dijadikan web server beserta *database server* dan REST server. Server inilah yang bertugas untuk memberikan *response* terhadap semua *request method* yang diterima. Server dikoneksikan dengan internet yang memiliki IP Publik sehingga bisa diakses. Dari luar jaringan lokal.

Pada penelitian ini sudah dilakukan *PORT forwarding* dari server sehingga bisa diakses dari luar. Untuk mempermudah ingatan dan penulisan dari IP Publik maka perlu dilakukan *pointing* dengan *hostname*. Untuk mendapatkan *hostname* dipergunakan layanan dari no-ip.com. No.-ip.com merupakan penyedia layanan *Dynamic Domain Name Server (DDNS)* gratis ini yang bisa digunakan untuk memberikan *A Record* dari IP Publik server menjadi sebuah *hostname*. Adapun pada penelitian ini digunakan *hostname* sinatra2020.ddns.net sebagai *hostname* dari IP Publik server.



Gambar 4.14 konfigurasi *Port Forwarding* pada router

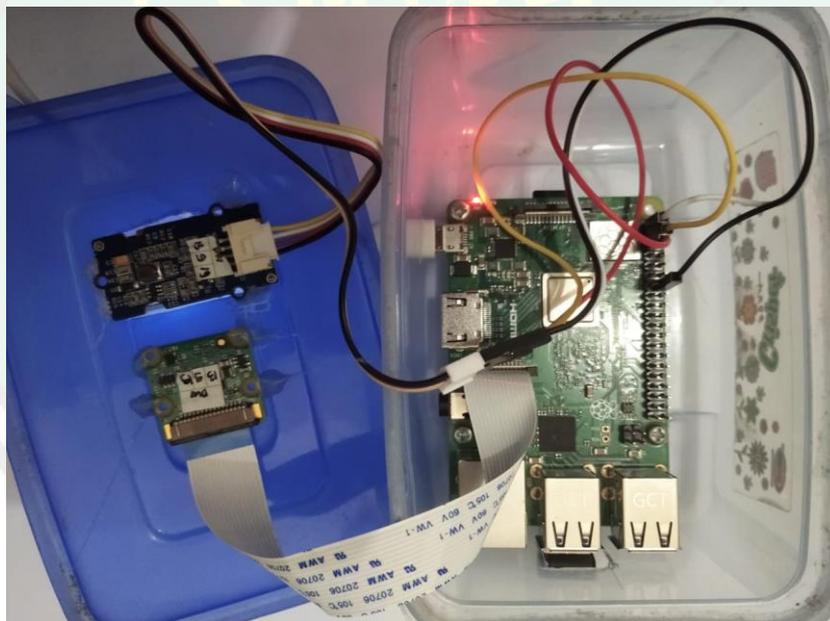


Gambar 4.15 Konfigurasi *Hostname* di halaman no-ip.com

Pada server ini juga dilengkapi kamera dan OLED *display*. Kamera digunakan untuk melakukan *streaming*, pengambilan gambar dan video. Sedangkan OLED *display* digunakan untuk memantau kondisi server, melihat IP dan menampilkan kondisi sensor.



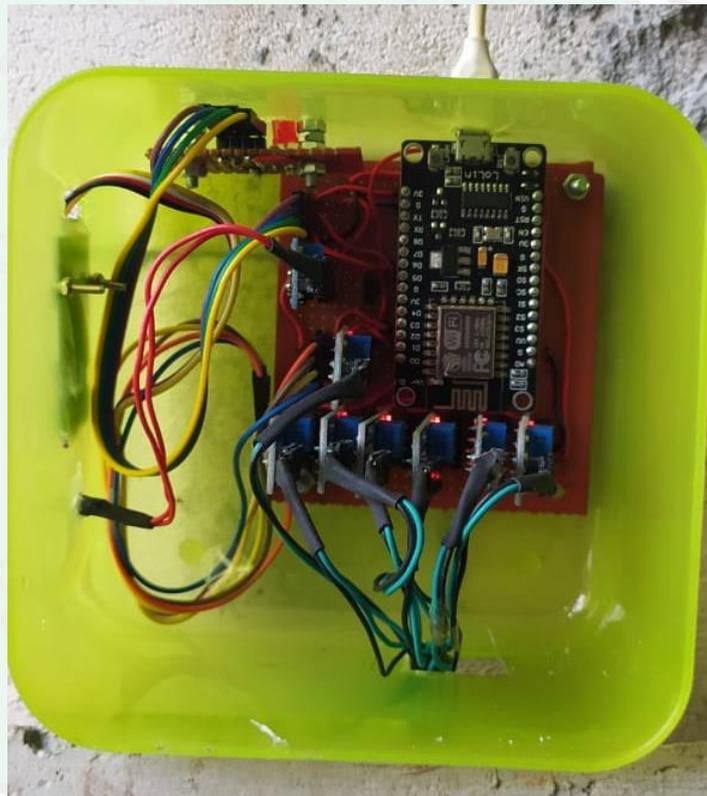
Gambar 4.16 Raspberry Pi tampak depan



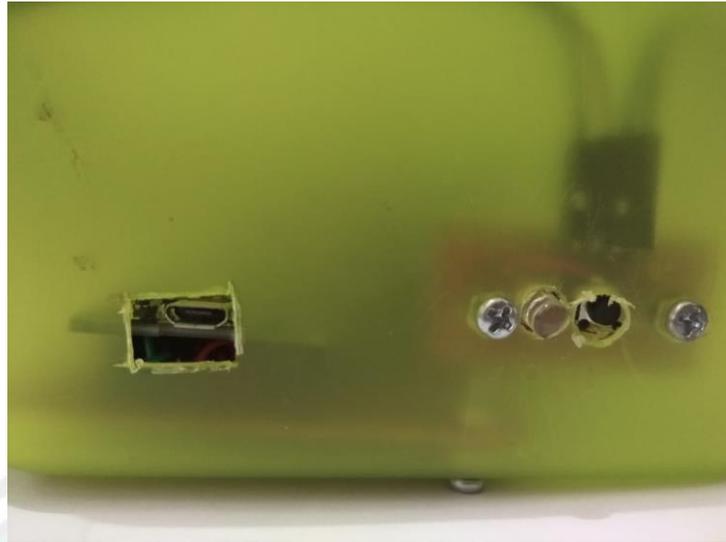
Gambar 4.17 Raspberry Pi tampak dalam

4.4.2 Sensor dan *Watering Node*

Pada penelitian ini data sensor diambil menggunakan 2 buah NodeMCU ESP8266. Pada Node Sensor dipasang *module* sensor berupa sensor cahaya, hujan, kelembaban, kecepatan angin, suhu, ketinggian. Sedangkan pada Node *watering control* dilengkapi dengan sensor ultrasonik dan *relay* untuk mengontrol penyiraman. Setiap node dioperasikan dengan daya dari USB dengan tegangan 5V DC. Kedua node juga dilengkapi tombol untuk melakukan restart dan reset data untuk me-reset data *setting* ke awal tanpa harus dikoneksikan ke komputer.

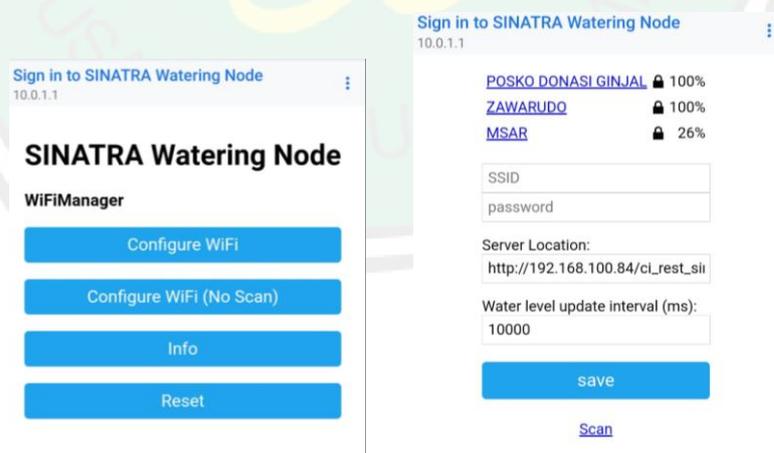


Gambar 4.18 rangkaian NodeMCU tampak dalam



Gambar 4.19 rangkaian NodeMCU tampak samping

NodeMCU ESP8266 ini terkoneksi dan berkomunikasi melalui jaringan WiFi yang memiliki akses internet. Untuk mempermudah konektivitas dan *setting* disediakan wifi portal. Dengan wifi portal yang sudah terprogram di NodeMCU ESP8266 dapat mempermudah pemilihan jaringan WiFi yang tersedia dan konfigurasi *hostname* server tanpa harus *upload* ulang kode program. Untuk bisa masuk ke portal WiFi NodeMCU ESP8266 pertama-tama harus terkoneksi ke akses *point* dari NodeMCU kemudian akan diarahkan ke portal.



Gambar 4.20 WiFi Portal NodeMCU ESP8266

4.5 Integrasi Islam

Bercocok tanam merupakan salah satu pekerjaan yang sudah dilakukan manusia sejak jaman dahulu. Di Indonesia sendiri bercocok tanam merupakan salah satu mata pencaharian pokok. Namun dalam bercocok tanam masyarakat Indonesia masih menggunakan cara yang terkesan manual dalam melakukan penyiraman. Alhasil dengan penyiraman manual tersebut kadang pemberian air dapat berlebihan maupun kekurangan. Padahal air merupakan salah satu unsur yang diperlukan tanaman untuk hidup. Allah berfirman dalam Al-Quran surah Az-Zumar ayat 21 sebagai berikut:

أَلَمْ تَرَ أَنَّ اللَّهَ أَنْزَلَ مِنَ السَّمَاءِ مَاءً فَسَلَكَهُ يَنْبِيعٌ فِي الْأَرْضِ ثُمَّ يُخْرِجُ بِهِ زُرْعًا
مُخْتَلِفًا أَلْوَانُهُ ثُمَّ يَهَيِّجُ فَتْرَهُ مُصْفَرًّا ثُمَّ يَجْعَلُهُ حُطَامًا إِنَّ فِي ذَلِكَ لَذِكْرًا لِأُولِي
الْأَلْبَابِ ع - ٢١

Artinya: “Apakah engkau tidak memperhatikan, bahwa Allah menurunkan air dari langit, lalu diaturnya menjadi sumber-sumber air di bumi, kemudian dengan air itu ditumbuhkan-Nya tanam-tanaman yang bermacam-macam warnanya, kemudian menjadi kering, lalu engkau melihatnya kekuning-kuningan, kemudian dijadikan-Nya hancur berderai-derai. Sungguh, pada yang demikian itu terdapat pelajaran bagi orang-orang yang mempunyai akal sehat..” [Az-Zumar: 21].

Dalam tafsir Aljalalain (AL-MAHALLI & AS-SUYUTHI, 2018) ditafsirkan sebagai berikut:

(Apakah kamu tidak memperhatikan) maksudnya tidak mengetahui (bahwa sesungguhnya Allah menurunkan air dari langit, maka diatur-Nya menjadi sumber-sumber) yakni, dia memasukkan air itu ke tempat-tempat yang dapat menjadi sumber air (di bumi kemudian ditumbuhkan-Nya dengan air itu tanam-tanaman yang bermacam-macam warnanya, lalu ia menjadi kering)

menjadi layu dan kering (lalu kamu melihatnya) sesudah hijau menjadi (kekuning-kuningan kemudian dijadikan-Nya hancur berderai) yakni rontok (Sesungguhnya pada yang demikian itu benar-benar terdapat pelajaran) peringatan (bagi orang-orang yang mempunyai akal) bagi orang-orang yang mau mengambil pelajaran darinya untuk menyimpulkan keesaan dan kekuasaan Allah swt.

Dari tafsir di atas ada dua kesimpulan yang bisa diambil. Pertama, air merupakan bagian terpenting yang tidak bisa dipisahkan dalam pertumbuhan tanaman. Sudah jelas bahwa untuk mendapatkan kualitas yang baik maka kita harus menjaga suplai kadar air agar sesuai dengan tumbuhan. Kedua, kita diharuskan untuk melakukan pengamatan/ pembelajaran terhadap tumbuhan mulai dari benih hingga kekuningan/ mati. Bagian ini juga menjelaskan perlunya *monitoring*/ pengamatan tumbuhan selain untuk menjaga tumbuhan tersebut, juga dibutuhkan untuk mengenali karakteristik dari tumbuhan tersebut.

BAB V PENUTUP

5.1 Kesimpulan

Pada penelitian ini sistem penyiraman otomatis berbasis *internet of things* berhasil dibangun. Sistem penyiraman ini menggunakan Raspberry Pi 3 B+ sebagai web server, database server, REST server dan pemroses metode *K-Nearest Neighbor*. Untuk pengambilan data dan penyiraman digunakan NodeMCU ESP8266. Antara NodeMCU dan Raspberry Pi saling berkomunikasi menggunakan HTTP melalui jaringan *wireless*. NodeMCU yang bekerja sebagai *client* mengirimkan data sensor maupun penyiraman dengan melakukan *request* REST API. Dari *request* tersebut REST server akan mengirimkan hasil *response* ke NodeMCU. *Response* yang dikirimkan berupa pesan sukses dan data klasifikasi.

Pada sistem yang dibangun juga telah diimplementasikan dengan metode *K-Nearest Neighbor*. Dengan implementasi tersebut sistem mampu melakukan penyiraman otomatis berdasarkan kondisi tanah. Kondisi tanah tersebut merupakan hasil dari klasifikasi *K-Nearest Neighbor* dengan menggunakan parameter suhu dan kelembaban tanah. Dari hasil klasifikasi tersebut sistem akan menentukan banyaknya volume air yang harus disiramkan terhadap tanaman.

Pada pengujian dari implementasi metode *K-Nearest Neighbor* diperoleh nilai *macro precision* sebesar 94,05%, *macro recall* 63,33% dan *macro f1* 67,7%. Nilai dari *precision* tersebut menunjukkan seberapa besar klasifikasi yang benar dari keseluruhan data prediksi yang benar. Sedangkan nilai *recall* menunjukkan berapa besar klasifikasi yang diprediksi benar dibandingkan keseluruhan klasifikasi yang benar. Adapun nilai *f1* menunjukkan tingkat akurasi dari uji coba yang sudah dilakukan.

Pengguna dapat mengakses sistem *interface* menggunakan IP lokal Raspberry Pi dengan menghubungkan perangkat ke jaringan WiFi yang sama. Pengguna juga dapat mengakses dari mana saja melalui jaringan internet dengan menggunakan IP publik yang sudah di *routing*. Pada sistem *interface*

yang dibangun dapat dilakukan *monitoring* dari sensor secara *real-time*. Adapun sensor yang di-*monitoring* berupa sensor kelembaban tanah, suhu, kelembaban udara, ketinggian, kecepatan angin, cahaya, dan ketinggian air tandon. Pada sistem *interface* juga disediakan kontrol penyiraman untuk mengubah mode dari otomatis ke manual. Pada halaman kamera, sistem mampu melakukan *streaming* video maupun mengambil gambar dari tanaman secara *real-time*.

5.2 Saran

Setelah mengembangkan Tanaman Otomatis Dan *Monitoring* Berbasis *Internet Of Things* Menggunakan *K-Nearest Neighbour*, ada beberapa saran yang perlu diterapkan guna pengembangan kedepannya.

1. Untuk *interface* aplikasi bisa dikembangkan untuk aplikasi *mobile native*. Hal ini dikarenakan di masa depan perangkat *mobile* akan mendominasi penggunaan perangkat komputasi. Penggunaan aplikasi berbasis web memang multi platform, namun dalam beberapa kasus pada perangkat *mobile* tidak bisa berjalan baik sebagaimana ketika diakses di komputer.
2. Untuk penjadwalan penyiraman perlu dibuatkan sistem *schedule automation* yang lebih kompleks. *Schedule automation* seharusnya bisa melakukan penjadwalan berdasarkan pemilihan hari dan bisa melakukan beberapa kali penyiraman dalam waktu yang sama.
3. Untuk pengembangan selanjutnya sistem dan perangkat ini bisa dipasarkan ke masyarakat khususnya petani atau penggemar tanaman. Pemasaran bisa dalam bentuk layanan server, DIY kit maupun perangkat siap pakai. Untuk itu perlu diperbaiki dalam sisi desain perangkat dan sistem.

DAFTAR PUSTAKA

- A.Neelima, Sravanthi, M. S., M.L.Ramyasri, & K.Yuvasri. (2018). Sensor Based Automated Moisture Monitoring Irrigation System with IOT. *International Journal of Innovative Research in Computer and Communication Engineering* , 6(1), 115-121.
- AL-MAHALLI, J., & AS-SUYUTHI, J. (2018). *TAFSIR JALALAIN*. UMMUL QURO .
- Bachrudin, Z., Widodo, C. E., & Adi, K. (2017). Simulator input-output sistem kontrol menggunakan Raspberry Pi. *Youngster Physics Journal*, 6(3), 272-279.
- Bakshi, U. A., & Bakshi, A. V. (2008). *Electronic Measurements and Instrumentation*. Pune: Technical Publication Pune.
- Bosch Sensortec. (2018, September). *BME280 - Datasheet*.
- components101.com. (2020, January 20). *HC-SR04 Ultrasonic Sensor*. Diambil kembali dari Components 101: <https://components101.com/ultrasonic-sensor-working-pinout-datasheet>
- Efendi, Y. (2018). INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU. *Jurnal Ilmiah Ilmu Komputer*, Vol. 4, No. 1, April 2018, 19-26.
- El-Farhan, A. H., & Pritts, M. (1997). Water Requirements And Water Stress In Strawberry. *Advances in strawberry research Vol. 16*, 5-12.
- Eliyen, K., Tolle, H., & Muslim, M. A. (2017). K-NEAREST NEIGHBOR UNTUK KLASIFIKASI PENILAIAN PADA VIRTUAL PATIENT CASE. *Jurnal Arus Elektro Indonesia (JAEI) Vol. 3 No. 1*, 13-17.
- Junaidi, A. (2015). INTERNET OF THINGS, SEJARAH, TEKNOLOGI DAN PENERAPANNYA : REVIEW. *Jurnal Ilmiah Teknologi Informasi Terapan (JITTER) Volume I, No 3, 10 Agustus 2015* , 62-66.
- Kementerian Riset dan Teknologi / Badan Riset dan Inovasi Nasional. (2000). *TTG BUDIDAYA PERTANIAN: STROBERI (Fragaria chiloensis L. / F. vesca L.)*. Jakarta: ristek.go.id.

- Krisandi, N., & Helmi, B. P. (2013). ALGORITMA k-NEAREST NEIGHBOR DALAM KLASIFIKASI DATA HASIL. *Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster)*, 2(1), 33-38.
- Naik, P., Kumbi, A., Katti, K., & Telkar, N. (2018). AUTOMATION OF IRRIGATION SYSTEM USING IoT. *International Journal of Engineering and Manufacturing Science Volume 8, Number 1 (2018)*, 77-88.
- Nishom, M. (2019). Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, Vol.04, No.01, Januari 2019, 20-24.
- Panduardi, F., & Haq, E. S. (2016). WIRELESS SMART HOME SYSTEM MENGGUNAKAN RASPBERRY PI BERBASIS ANDROID. *Jurnal Teknologi Informasi dan Terapan*, Vol. 03, No. 01, Juli-Desember 2016, 320-325.
- Prakasa, O. S., & Lhaksamana, K. M. (2018). KLASIFIKASI TEKS DENGAN MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR PADA KASUS KINERJA PEMERINTAH DI TWITTER. *e-Proceeding of Engineering Vol.5 No.3*, 8237-8248.
- Pulver, T. (2019). *Hands-On Internet of Things with MQTT*. Birmingham: Packt Publishing.
- Putri, R. E., Suparti, & Rahmawati, R. (2014). Perbandingan Metode Klasifikasi Naive Bayes dan kNearest Neighbor pada Analisis Data Status Kerja di Kabupaten Demak. *Jurnal Gaussian Vol. 3 No. 4*, 831-838.
- Raju, B. V. (2017). An Automatic Form Monitoring System Using Arduino and Wireless Sensor Networks. *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, 87-96.
- Raspberrypi.org. (2019). Raspberry Pi 3 Model B+ Product Brief.
- Raspberrypi.org. (2020, January 20). *Camera Module*. Diambil kembali dari Raspberry Pi: <https://www.raspberrypi.org/documentation/hardware/camera/>
- Rawal, S. (2017). IOT based Smart Irrigation System. *International Journal of Computer Applications Volume 159 – No 8, February 2017*, 7-11.

- SEED Studio. (2020, January 20). *SEED Studio*. Diambil kembali dari Grove - OLED Display 0.96 inch: http://wiki.seeedstudio.com/Grove-OLED_Display_0.96inch/
- Shekhar, Y., Dagur, E., & Mishra, S. (2017). Intelligent IoT Based Automated Irrigation System. *International Journal of Applied Engineering Research*, 12(18), 7306-7320.
- Sujarwata. (2012). *Belajar Mikrokontroler BS2SX Teori, Penerapan dan Contoh pemrograman PBasic*. Yogyakarta: Deepublish.
- Texas Instruments Incorporated. (2003). *CD4051B, CD4052B, CD4053B*.
- Turang, D. A. (2015). Pengembangan Sistem Relay Pengendalian Dan Penghematan Pemakaian Lampu Berbasis Mobile. *Seminar Nasional Informatika 2015 (semnasIF 2015)*, 75-85.
- Widjanarka, W. (2006). *Teknik Digital*. Ciracas: Penerbit Erlangga.
- Yahwe, C. P., Isnawaty, & Aksara, L. M. (2016). Rancang Bangun Prototype System Monitoring Kelembaban Tanah Melalui SMS Berdasarkan Hasil Penyiraman Tanaman "Studi Kasus Tanaman Cabai dan Tomat". *semantIK*, 97-110.