

**SISTEM PENYIRAMAN TANAMAN HIAS OTOMATIS
DENGAN METODE C4.5 BERBASIS IOT
(Internet of Things)**

Rian Hermawan^{*1}, Gilang Handika Adjie^{#2}

*Program Studi Teknik Informatika, STMIK Subang
Jl. Marsinu No. 5 - Subang, Tlp. 0206-417853 Fax. 0206-411873
E-mail: stmik.rian@yahoo.com^{*1}, handikag61@gmail.com^{#2}*

ABSTRAKSI

Tanaman merupakan tumbuhan yang dibudidayakan agar dapat diambil manfaatnya. Budidaya tanaman sendiri pada dasarnya dapat menjadi peluang usaha yang menjanjikan. Mulai dari budidaya tanaman hias, sayur mayur dan lain sebagainya. Tetapi saat ini produksi tanaman hias masih banyak yang belum menghasilkan hasil yang maksimal, dikarenakan masyarakat masih menggunakan teknologi manual dalam sistem pertanian yang digunakan. Tujuan dilakukan penelitian ini adalah merancang sebuah alat penyiraman tanaman hias otomatis untuk mengatasi masalah dalam penyiraman tanaman hias yang masih dilakukan secara manual dan sebagai bahan pembelajaran. Prototype ini menggunakan Arduino Uno sebagai pengontrol utama, sensor kelembaban tanah digunakan untuk membaca kadar kelembaban tanah dan digunakan sebagai saklar untuk menghidupkan pompa penyiram.

Sistem pengambilan ini mengambil data dengan sensor kelembaban tanah, kelembaban udara, dan suhu, untuk mencari kelembaban tanah, kelembaban udara dan suhu pada tanaman. Kemudian data yang diperoleh di kirim ke jaringan modul ESP8266 dan dikirim ke platform thingspeak, data yang tampil di inputkan ke dalam database untuk diolah menggunakan metode C4.5, hasil perhitungan ke dalam database untuk diolah menggunakan metode C4.5, hasil perhitungan ditampilkan oleh sistem.

Kata Kunci : **Arduino Uno, IoT, C4.5**

1. Pendahuluan

1.1 Latar Belakang

Tanah merupakan salah satu media yang digunakan sebagai media hidup dari berbagai macam tumbuhan. Tanaman memerlukan air untuk dapat tumbuh secara optimal. Untuk itulah kondisi kelembaban tanah harus dijaga pada suatu keadaan tertentu yang sesuai dengan kebutuhan tanaman. Kekurangan kadar air atau kelebihan kadar air dapat mengakibatkan tanaman tidak dapat tumbuh dengan baik. Oleh karena itu, diperlukan suatu sistem yang bekerja secara otomatis dan cerdas untuk melakukan penyiraman tanaman. Sistem ini akan menjaga kadar air pada tanah yang sesuai dengan kebutuhan tanaman. Kondisi kadar air tanah diperoleh dari sensor kelembaban tanah dan dipengaruhi juga oleh suhu udara.

Dalam kasus ini Mikrokontroler dapat membantu tugas manusia dalam melakukan penyiraman dan monitoring pada tanaman, yang biasanya dilakukan dengan cara manual. Cara kerja Mikrokontroler ini cukup mudah hanya dikendalikan lewat website.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas,maka dapat diidentifikasi beberapa permasalahan sebagai berikut:

1. Keterbatasannya indra manusia untuk mengontrol kelembaban tanah pada tanaman.
2. Kekurangan air pada tanaman yang membuat tanaman akan menjadi layu.
3. Kelebihan air pada tanaman membuat permukaan tanah lembab yang memunculkan mikroorganisme jamur yang menyebabkan penyakit bagi tanaman.

1.3 Tujuan

Tujuan yang dipeoleh dari penelitian ini :

1. Membuat alat yang murah dengan tingkat keakuratan tinggi.
2. Menerapkan IoT sebagai sistem pendukung keputusan penyiraman tanaman hias.
3. Membuat alat yang dapat mendeteksi kelembaban tanah, kelembaban udara, dan suhu udara.

1.4 Manfaat

Manfaat yang dapat diperoleh dari hasil penelitian tugas akhir ini adalah :

1. Merancang dan mengimplementasikan sebuah sistem cerdas yang dapat digunakan untuk penyiraman otomatis pada tanaman.
2. Menghemat waktu dalam melakukan penyiraman tanaman.
3. Menambahkan suatu pengalaman dalam mentransformasikan ilmu pengetahuan dan teknologi menjadi suatu media, ataupun barang yang canggih dan efisien.

1.5 Batasan Masalah

Berdasarkan rumusan masalah di atas, maka dalam penyusunan laporan penelitian ini penulis membatasi pembahasan hanya pada :

1. Merancang sistem penyiraman tanaman hias otomatis dengan menggunakan beberapa alat diantaranya: sensor *soil moisture*, sensor suhu DHT 11, kabel jumper, *breadboard*, Arduino Uno dan ESP8266.
2. Sistem pemantauan kelembaban tanah hanya memberikan keterangan proses untuk menyiram tanaman secara otomatis, tidak membahas pertumbuhan tanaman.
3. Sistem yang di bangun memakai bahasa pemrograman arduino uno, dan *platform Thinkspeak*.
4. Metode yang akan di gunakan adalah metode C4.5.

1.6 Metodologi Penelitian

Pada penelitian ini menggunakan metode deskriptif, yaitu metode yang menggambarkan suatu keadaan atau permasalahan yang sedang terjadi berdasarkan fakta dan data-data yang diperoleh dan dikumpulkan pada waktu pelaksanaan penelitian. Berdasarkan hasil yang dilakukan, maka dapat ditetapkan menjadi rumusan masalah, kemudian dibuat batasan-batasan masalah agar pembahasan yang akan dijelaskan tidak keluar dari ruang lingkup penelitian.

1. Studi Literatur
Yaitu pengambilan pustaka yang relevan dengan topik utama yang dikaji, sehingga memperoleh landasan teoritik untuk melakukan rancang bangun.
2. Metode pengumpulan data
Yaitu metode pengumpulan data dan referensi dari media cetak elektronik yang menunjang dalam penyusunan tugas akhir.
3. Metode observasi
Yaitu pengumpulan data dengan cara melakukan pengamatan langsung terhadap alat dan aplikasi yang akan di buat.
4. Pengumpulan alat dan bahan
Dalam metode ini dilakukan pengumpulan alat dan bahan pokok seperti komputer server, mikrokontroler dengan pendukungnya.
5. Melakukan pengujian terhadap tiap komponen dengan alat uji yang sesuai.
Perancangan dan realisasi tiap blok sesuai dengan diagram skema yang telah ditentukan.

2. Tinjauan Pustaka

2.1 Penyiraman Tanaman

Penyiraman merupakan suatu hal yang tidak dapat dilepaskan didalam menjaga serta merawat tanaman agar tanaman tetap tumbuh dengan subur. Kebutuhan air yang cukup sangat mempengaruhi pertumbuhan tanaman. Contohnya seperti tanaman cabai dan tomat yang membutuhkan perhatian khusus karena jika tanaman ini tidak mendapatkan kondisi yang baik maka tanaman tidak dapat tumbuh dengan baik, bahkan akan berdampak fatal bagi tanaman tersebut. [1]

Penyiraman tanaman secara manual dapat mengganggu efisiensi waktu dan tenaga. Penyiraman pada tanaman dengan kelebihan atau kekurangan air dapat pula mengurangi daya tahan maupun menyebabkan kematian pada tanaman itu sendiri. Sehingga berpotensi kerugian pada petani tanaman. [2]

2.2 Mikrokontroler

Mikrokontroler merupakan suatu bagian elektronik yang dalam wujudnya seperti IC (*Integrated Circuit*) yang dapat berguna sebagai komputer tetapi dalam ukuran yang kecil atau minim. Kata Mikrokontroler ini ialah gabungan dari dua kata yaitu Mikro dan Controler. Mikro berarti yang sangat kecil, sedangkan controler berarti sebuah pengendali. [3]

Mikrokontroler terdapat keunggulan tersendiri yaitu tersedia sebuah RAM dan peralatan I/O pendukung yang membuat ukuran board mikrokontroler ini menjadi lebih praktis. Mikrokontroler merupakan suatu chip pada komputer yang memiliki fungsi untuk pengontrol susunan dari alat elektronik dan secara garis besar dapat menyimpan program yang ada. Mikrokontroler terbagi atas Memori, I/O khusus, ROM, Timers (pewaktu), dan unsur pendukung seperti ADC atau (*Analog Digital Converter*) yang sudah terjamin pada isi dalamnya.

2.3 Arduino Uno

Arduino Uno R3 adalah board sistem minimum berbasis mikrokontroller ATmega328P jenis AVR. Arduino Uno R3 memiliki 14 digital input/output (6 diantaranya dapat digunakan untuk PWM output), 6 analog input, 16 MHz osilator kristal, USB connection, power jack, ICSP header dan tombol reset. [4]

2.4 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan. [5]

3. Analisa

3.1 Deskripsi Sistem

Pada Penelitian ini, penulis melakukan rancangan bangun sistem untuk pengujian penyiraman otomatis tanaman hias berbasis IoT pada platform Thingspeak menggunakan metode C4.5. Sistem ini merupakan perpaduan antara perangkat keras dan perangkat lunak yang saling terhubung dan telah deprogram. Perangkat keras yang di pakai yaitu Arduino R3, ESP01, sensor Soil Moisture, Sensor DHT-11, dan komponen lainnya. Sedangkan perangkat lunak yang digunakan yaitu *platform thingspeak* yang meryapakan tempat penyimpanan data.

Sistem ini adalah untuk pengujian sebuah tanaman hias untuk mengetahui kelembaban tanah, kelembaban udara, dan suhu yang ada di sekitar tanaman hias. Untuk menjalankan sistem ini membutuhkan daya listrik dan koneksi jaringan *Wireless* unutk menghubungkan alat ke *platform*. Jaringan yang digunakan oleh laptop harus sama dengan jaringan yang di pakai oleh ESP01.

3.2 Analisis Kebutuhan Perangkat Keras

Berdasarkan studi literature yang telah dilakukan, penulis menentukan beberapa komponen perangkat keras yang dibutuhkan untuk merancang bangun sistem. Perangkat keras yang dimaksud adalah sebagai berikut :

- Arduino Uno R3
- Sensor Soil Moisture
- Sensor DHT-11
- Project Board
- ESP 8266
- Water Pump
- Kabel Jumper

Komputer Server, Komputer Server yang digunakan oleh penulis adalah laptop dengan spesifikasi *processor* Intel Core I5, *RAM* 4GB.

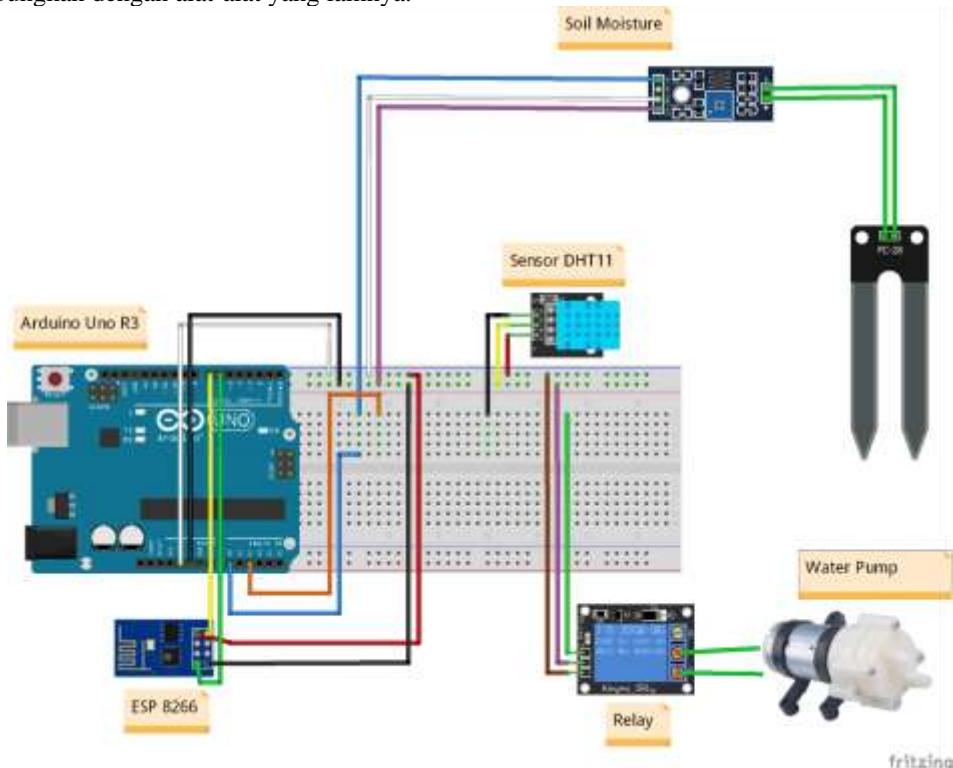
3.3 Analisis Kebutuhan Perangkat Lunak

Penulis telah melakukan analisa terhadap kebutuhan perangkat lunak yang diperlukan untuk melakukan rancang bangun sistem. Perangkat lunak yang dibutuhkan adalah sebagai berikut:

- Arduino IDE, berperan sebagai tempat program Arduino Uno R3, Sensor Soil Moisture, ESP 8266, dan komponen *hardware* lainnya yang dipakai dalam penelitian ini.
- Platform *thingspeak*, yang dibutuhkan sebagai penyimpanan data untuk penelitian ini.
- Sistem Operasi Windows.

3.4 Perancangan Perangkat Keras

Pada gambar ini merupakan rancangan alat atau skematik beserta sensor-sensor yang disambungkan dengan alat-alat yang lainnya.



Gambar 3.1 Perancangan Alat

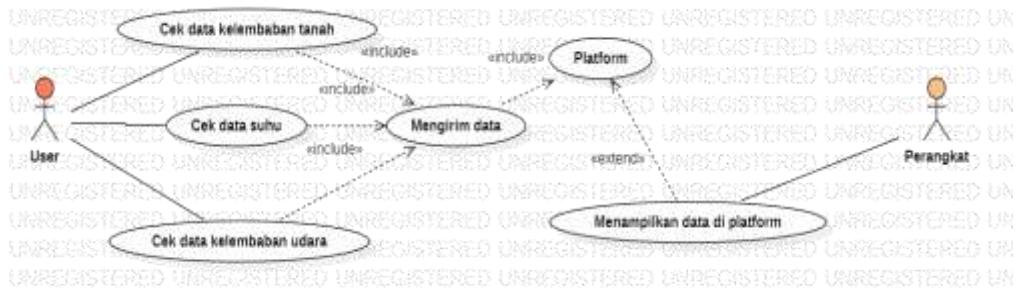
3.5. Software Requirement

Software requirement merupakan suatu statement yang dijelaskan tentang berbagai kebutuhan yang harus dipenuhi oleh suatu *software*. *Software requirement* dibuat atas dasar hasil proses analisa agar mempermudah proses pengembangan perangkat lunak. Berikut *requirement* hasil analisis penulis :

1. Sistem ini memungkinkan sensor untuk membaca data parameter.
2. Sensor *Soil Moisture* dapat membaca data kelembaban tanah.
3. Sensor DHT11 dapat membaca data kelembaban dan suhu.
4. Sistem ini dapat membaca data kemudian ke *Cloud* dengan modul WIFI ESP8266.
5. Data dapat disimpan dan ditampilkan di *Cloud* melalui platform yang digunakan.
6. Data di *Cloud* dapat di *export* ke dalam bentuk Excel.
7. Sistem ini dapat menyimpan data di database yang terkoneksi ke WEB.
8. Sistem ini dapat melakukan perangkingan data dengan metode C4.5.
9. WEB memungkinkan dapat melakukan pengujian data parameter dengan metode C4.5
10. User dapat melakukan edit dan hapus data uji parameter yang telah di inputkan.

3.6. Use Case Diagram

Perancangan *use case* Diagram mendeskripsikan proses operasi sistem penyiraman tanaman hias.



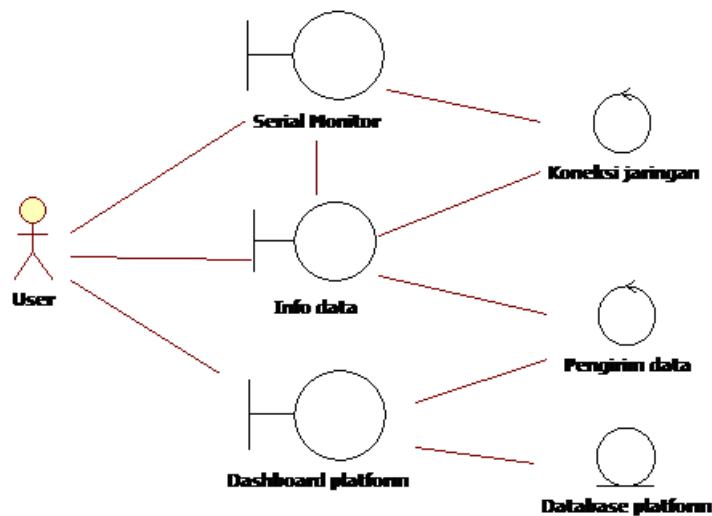
Gambar 3.2 Use Case Diagram

Penjelasan *use case* diagram :

1. Sistem mengirim data ke *Thingspeak*.
2. Sistem menampilkan data objek yang diterima dari *Thingspeak*.
3. User melihat data objek yang ditampilkan oleh sistem.
4. User menyimpan data objek pada database *Thingspeak*.

3.7 Robustness Diagram

Penulis membuat *robustness* diagram untuk menampilkan gambaran objek untuk setiap *use case* yang berinteraksi dengan sistem.

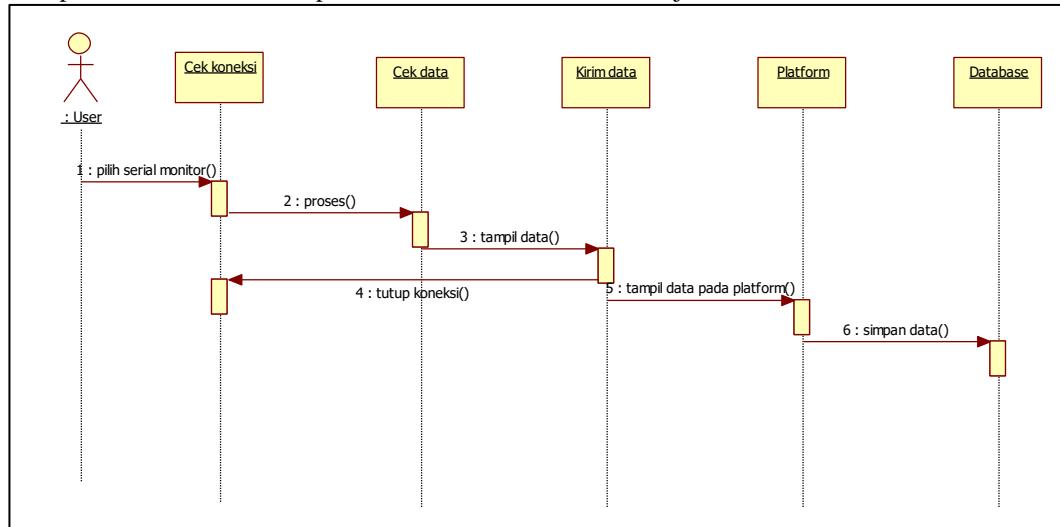


Gambar 3.3 Robustness Diagram

1. User melihat serial monitor untuk mengetahui sistem itu berjalan.
2. Sistem menyambungkan koneksi internet.
3. User melihat data sensor melalui serial monitor.
4. Sistem mengirim data ke platform.
5. User melihat hasil data sensor melalui platform.
6. Data sensor masuk database platform.

3.8 Sequence Diagram

Penulis membuat *sequence diagram* untuk menggambarkan suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek didalam sistem.

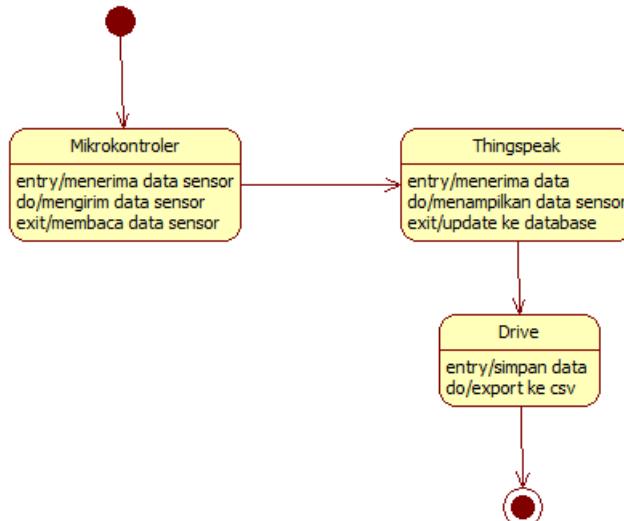


Gambar 3.4 Sequence Diagram

1. User membuka serial monitor.
2. Menyambungkan jaringan internet.
3. Menampilkan data serial monitor.
4. Jika koneksi terputus kemudian ke serial monitor dan coba menyambungkan ulang jaringan internet.

5. Berhasil mengirim data ke *platform* dan menampilkan data sensor.
6. Lalu menyimpan data sensor di *database platform*.

3.9 Statechart Diagram

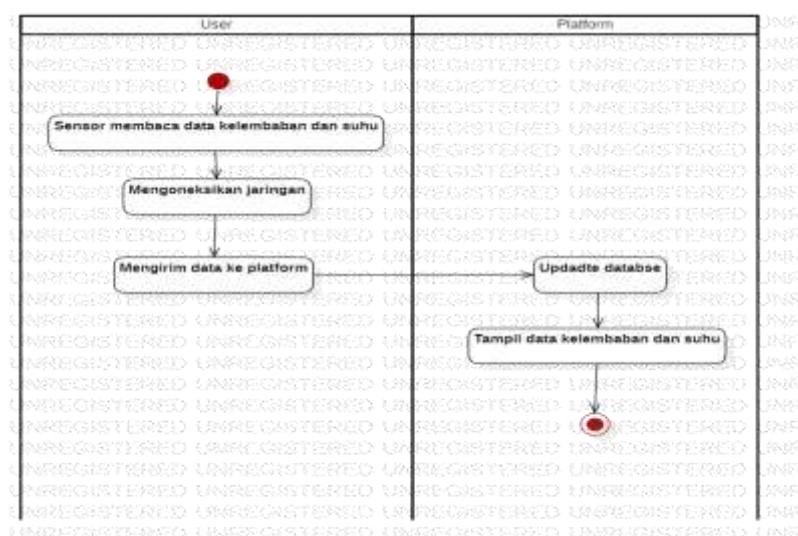


Gambar 3.5 Statechart Diagram

Sistem menghubungkan jaringan untuk mengirim data sensor, jika jaringan terputus maka sistem tidak akan mengirim data ke platform dan akan mengecek ulang jaringan. Kemudian jika jaringan terhubung dengan baik maka data sensor di *platform* dan tersimpan otomatis oleh *database platform*.

3.10 Activity Diagram

Penulis membuat *activity diagram* untuk menampilkan aliran aktivitas dalam sebuah sistem yang dirancang.

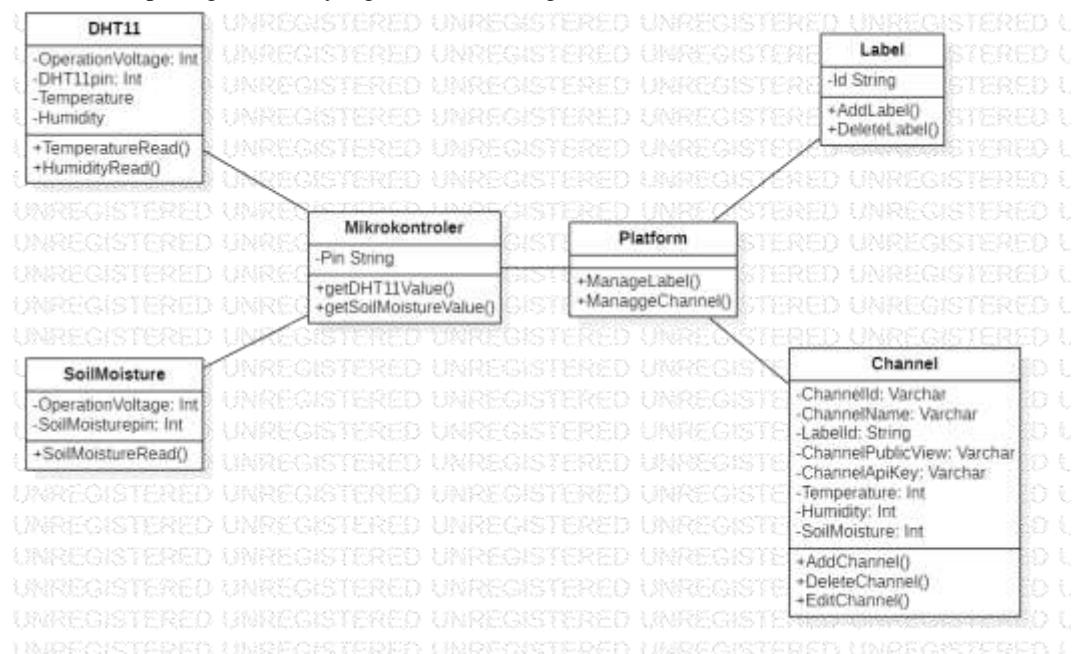


Gambar 3.6 Activity Diagram

Alat dihidupkan, kemudian alat terhubung dengan haringan, jika jaringan tidak terhubung (tidak ditemukan) maka alat akan menghubungkan kembali ke jaringan samoa terhubung. Setelah alat terhubung ke jaringan maka alat akan menghubungkan ke *Thingspeak*, setelah terhubung ke *Thingspeak* maka data dari sensor di kirimi ke *Thingspeak* untuk ditampilkan di *dashboard Thingspeak*. *Databse* akan mengupdate sendiri ketika data sensor itu terkirim.

3.11 Class Diagram

Class Diagram adalah visual dari jenis yang dibentuk. *Class diagram* merupakan jalur jalannya *database* pada seluruh sistem yang digunakan untuk menampilkan beberapa kelas yang ada dalam sistem perangkat lunak yang akan dikembangkan.



Gambar 3.7 *Class Diagram*

3.12 Rancangan Antarmuka

Rancangan antarmuka dibuat sesederhana mungkin agar user *friendly* dan tidak membantakan server, mengingat sistem yang penulis buat ini harus memuat data yang banyak. Rancangan antarmuka dibuat menggunakan visio 2013.

1. Antarmuka Login



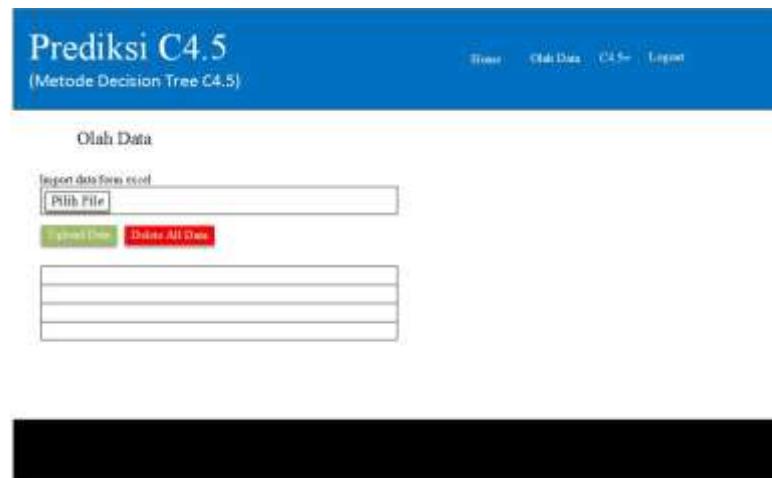
Gambar 3.8 Antarmuka *Login*

2. Antarmuka Home



Gambar 3.9 Antarmuka Home

3. Antarmuka Olah Data



Gambar 3.10 Antarmuka Olah Data

4. Antarmuka Data Mining



Gambar 3.11 Antarmuka Data Mining

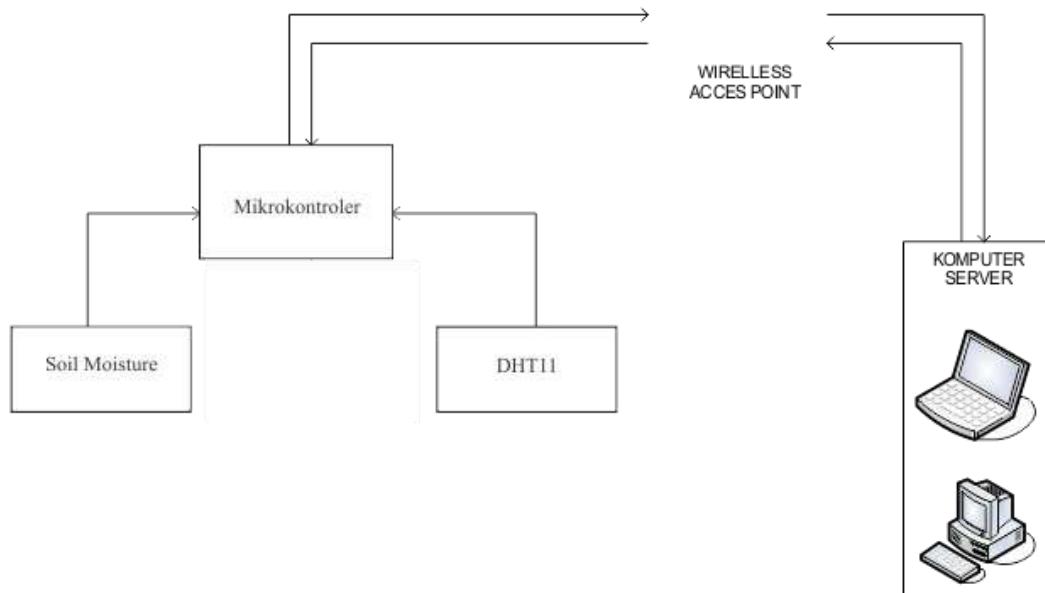
5. Antarmuka Pohon Keputusan



Gambar 3.12 Antarmuka Pohon Keputusan

3.13 Arsitektur Jaringan

Sistem penyiraman otomatis ini menggunakan arsitektur jaringan komputer LAN dan menggunakan *access point* sebagai alat yang digunakan.



Gambar 3.13 Arsitektur Jaringan

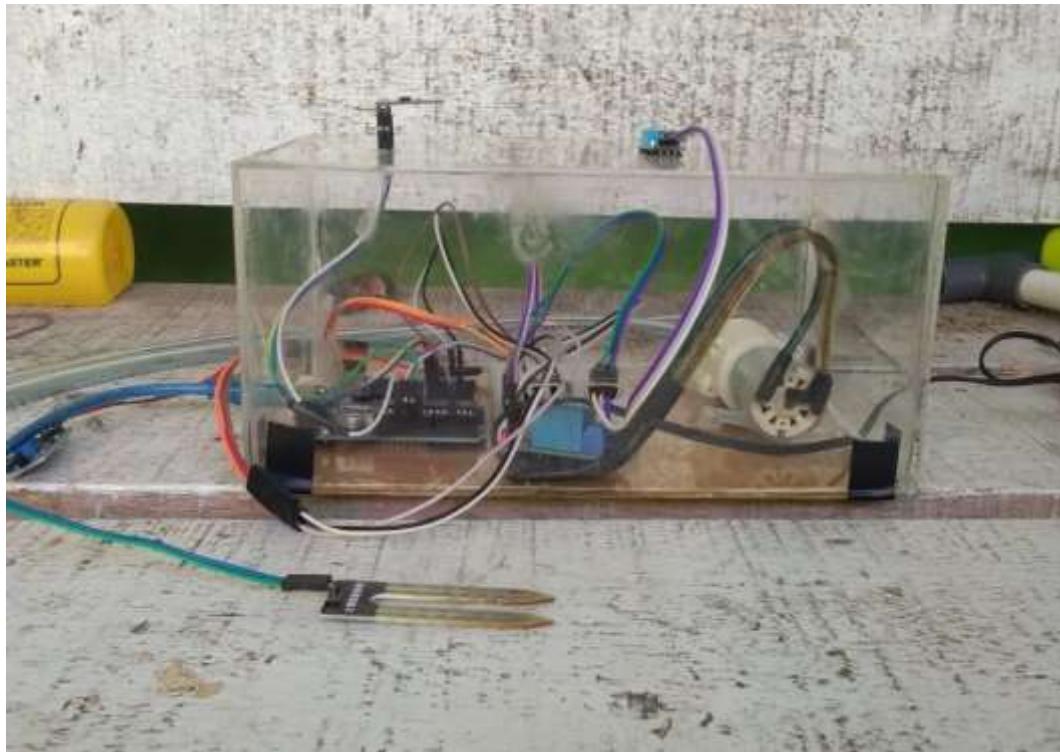
4 Hasil dan Pembahasan

4.1 Implementasi Mikrokontroler

Pada penelitian ini penulis menggunakan Arduino R3 sebagai *board controller*, sensor *soil moisture* untuk membaca parameter kelembaban tanah, sensosr DHT11 unutk membaca parameter kelembaban udara dan suhu modul wifi ESP8266 sebagai alat untuk mengirimkan data parameter dari sensor-sensor, serta perangkat power bank sebagai sumber tegangan agar Arduino tetap berfungsi.

Agar Arduino dan ESP8266 saling terhubung dan sensor dapat berfungsi sesuai dengan yang diinginkan, perlu adanya pemrograman pada *board Arduino* dan ESP8266, untuk proses kerja alat yang dibuat yaitu Arduino menerima data yang diperoleh dari sensor, kemudian dikirimkan oleh

Arduino ke ESP8266 yang dikoneksikan dengan jaringan kemudian data dikirim oleh ESP8266 ke komputer server melalui jaringan.



Gambar 4.1 Implementasi Mikrokontroler

Penulis melakukan pemrograman mikrokontroler menggunakan aplikasi Arduino IDE. Proses pemrograman Arduino Uno R3 meliputi proses pemrograman sensor *soil moisture* dan DHT11. Serta dengan pemrograman untuk proses konfigurasi Arduino dengan ESP8266.

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <Adafruit_DHT.h>
#include <SoftwareSerial.h>

// DHT pins
#define DHTPIN 2
#define DHTTYPE DHT11

// BME280 pins
#define BME280_SDA 14
#define BME280_SCL 13
#define BME280_CS 12

// WiFi settings
const char* ssid = "your_ssid";
const char* password = "your_password";
const int port = 80;
const String host = "http://192.168.4.1:80/test.html";

// WiFi connection
WiFiClient client;

// BME280 sensor
Adafruit_BME280 bme;

void setup() {
    // Set pins as digital
    pinMode(DHTPIN, INPUT);
    // Initialize WiFi
    WiFi.begin(ssid, password);
    // Initialize BME280
    if (!bme.begin(BME280_SDA, BME280_SCL)) {
        Serial.println("Failed to initialize BME280!");
        while (true);
    }
}

void loop() {
    // Read DHT11 sensor
    float temperature = Adafruit_DHT.readTemperature(DHTTYPE);
    float humidity = Adafruit_DHT.readHumidity(DHTTYPE);

    // Read BME280 sensor
    float pressure = bme.readPressure();
    float temperatureBME = bme.readTemperature();
    float humidityBME = bme.readHumidity();

    // WiFi connection
    if (!client.connect(host, port)) {
        Serial.println("Connection failed");
        while (true);
    }

    // Send data to server
    client.print("GET / HTTP/1.1\r\n");
    client.print("Host: " + host + "\r\n");
    client.print("Port: " + String(port) + "\r\n");
    client.print("User-Agent: Arduino\r\n");
    client.print("Content-Type: application/x-www-form-urlencoded\r\n");
    client.print("Content-Length: 0\r\n\r\n");

    // Read response
    String response = client.readStringUntil('\r');
    Serial.println(response);
}
```

Gambar 4.2 Pemrograman Mikrokontroler

4.2 Pengujian Alat

Tabel hasil pengujian alat merupakan *record* sistem untuk menguji apakah alat yang penulis buat berjalan dengan baik tanpa error atau tidak.

Tabel 4.1 Pengujian Alat

No.	Nama Proses	Hasil Pengujian
1	Koneksi sensor-sensor ke arduino	Ya
2	Koneksi <i>ESP 8266 WiFi</i> ke access point	Ya
3	Koneksi mikrokontroler ke laptop	Ya
2	Pengambilan data suhu dan kelembaban udara	Ya
3	Pengambilan data kelembaban tanah	Ya
4	Pengambilan data volume air sawah	Ya
5	Monitoring data dari sensor melalui aplikasi berbasis android	Ya
7	Koneksi ke Database	Ya

4.3 Implementasi Sistem

Antarmuka dibuat sesuai sistem dengan rancangan antarmuka yang telah dijelaskan di bab sebelumnya.

1. Halaman *Login*

Halaman login yang telah penulis buat sesuai dengan rancangan antarmuka sebelumnya. Untuk masuk ke dalam sistem kita harus melakukan login terlebih dahulu dengan memasukan *username* dan *password*.



Gambar 4.3 Halaman *Login*

2. Halaman *Home*

Setelah melakukan *login* kita akan diarahkan ke halaman utama atau *dashboard* dari sistem yang berisi informasi tentang sistem.



Gambar 4.4 Halaman *Home*

3. Halaman Olah Data

Halaman ini digunakan untuk upload data latih yang nantinya akan dihitung. Data yang di *upload* sudah berupa data kategorikal yang sebelumnya telah diubah dari data *numeric*.

No.	Kebutuhan Air	Suhu	Aktivitas Tanah	Suhu-Air
1	0	Normal	Normal	0
2	0	Lebih	Normal	0
3	0	Normal	Normal	0
4	0	Normal	Normal	0
5	0	Normal	Normal	0
6	0	Normal	Normal	0
7	0	Normal	Normal	0
8	0	Normal	Normal	0
9	0	Normal	Normal	0
10	1	Normal	Normal	0

Gambar 4.5 Halaman Olah Data

4. Halaman Proses Data Mining

Data yang telah di upload sebelumnya akan tampil pada halaman proses *data mining*. Pada halaman ini nantinya data akan dihitung dan menampilkan hasil entropy dan gain.

No.	Kondisi Air Tanah	Suhu	Kelembaban Tanah	Akar
1	Tinggi	Normal	Normal	Mati
2	Tinggi	Normal	Normal	Mati
3	Tinggi	Normal	Normal	Mati
4	Tinggi	Normal	Normal	Mati
5	Tinggi	Normal	Normal	Mati
6	Tinggi	Normal	Normal	Mati
7	Tinggi	Normal	Normal	Mati
8	Tinggi	Normal	Normal	Mati
9	Tinggi	Normal	Normal	Mati
10	Tinggi	Normal	Normal	Mati
11	Panas	Normal	Normal	Mati

Gambar 4.6 Halaman Proses Data Mining

Dengan menekan tombol **Proses Mining** maka data akan dihitung secara otomatis. Perhitungan ini nantinya menghasilkan beberapa *node* dan setiap *node* menunjukkan *parent* dan akar yang nantinya berfungsi sebagai pohon keputusan.

4.4 Hasil Pengujian Sistem

Tabel ini merupakan *record* sistem untuk menguji apakah sistem yang penulis buat berjalan dengan baik tanpa *error* atau tidak.

Tabel 4.2 Pengujian Sistem

No.	Nama Halaman dan Proses	Hasil Pengujian
1	Halaman Utama	Ya
2	Halaman Olah Data	Ya
3	Halaman Poses Data Mining	Ya
4	Halaman Pohon Keputusan	Ya

5. Kesimpulan

Dalam penelitian ini penulis mendapatkan beberapa kesimpulan, diantaranya :

1. Setelah melakukan penelitian ini penulis mendapatkan bahwa dengan adanya alat ini dapat mengurangi terjadinya tanaman hias yang layu/mati, dan juga dapat mempermudah untuk mengontrol tanaman.
2. Penelitian ini dilakukan dengan menggunakan metode C4.5 yang mana metode ini digunakan untuk melakukan klasifikasi data dengan teknik pohon keputusan

Pustaka

- [1] Yawhe, C. P., Isnawati, & Aksara, L. F. (2016). Rangcangan Bangun *Prototype System Monitoring Kelembaban Tanah Melalui Sms Berdasarkan Hasil Penyiraman Tanaman "Studi Kasus Tanaman Cabai dan Tomat"*. *teknik informatika*.
- [2] Candra, J. E., & Maulana, A. (2019). *Penerapan Soil moisture sensor untuk Desain System Penyiraman Tanaman Otomatis*. SNISTEK.
- [3] Putra, H. A. (2018). “*Pengertian dan Dasar-dasar dari Mikrokontroler*”. . <https://garudacyber.co.id/artikel/694-pengertian-dan-dasar-dasar-dari-mikrokontroler>.

- [4] Shafiqudin, S., Rohma, F. J., Prayetya, A. E., & Firmansyah, R. (2016). Pemantauan Ruang Inkubator Penetasan Telur Ayam Dengan Berbasis Telemetri Menggunakan Arduino Uno R3. *Jurnal Nasional Teknik Elektro*, 5(1).
- [5] Dharwiyanti, S. (2003). *Pengantar Unified Modeling Language (UML)*. Diambil kembali dari RomiSatriaWahono.net: <http://romisatriawahono.net>.