

Rancang Bangun Aplikasi Pengontrol Sistem Penyiram Tanaman Berbasis Arduino dan Android

Ayasha Ninda Maharani*, Bana Handaga

Program Studi Teknik Informatika/Fakultas Komunikasi dan Informatia – Universitas Muhammadiyah Surakarta
Surakarta, Indonesia

*ayashanm@gmail.com

Abstract— The existence of plants as living things is very important for human life. Plants survive by utilizing water. The process of watering plants that are not controlled can have a negative impact on their survival. The watering process that is done manually is very inefficient because the owner must be in the same place with his plants. The purpose of this research is to design an application for controlling an automatic plant watering system based on Android and Arduino so that it can be used for mobility. The application is designed using the SDLC waterfall method using Flutter's toolkit with the dart programming language. While the system design uses Arduino IDE software. The results of this study are an android application that can monitor the value of soil moisture in plants in units of time and an on/off button that can turn on and off the water pump for watering plants. Users can see the value of soil moisture in real time through the android application using an internet connection. This usage uses the firebase server as a database by utilizing the firebase server's realtime firebase database feature. The systems and applications that have been designed are tested for usability with the final test of the application using usability testing. The results of the survey on usability testing found that the application was functioning as it should and got the "Very Good" category of 94.44% of the respondents.

Abstrak— Keberadaan tanaman sebagai makhluk hidup sangat penting bagi kehidupan manusia. Tanaman bertahan hidup dengan memanfaatkan air. Proses penyiraman tanaman yang tidak terkontrol dapat berakibat buruk pada kelangsungan hidupnya. Serta proses penyiraman yang dilakukan secara manual sangat tidak efisien karena pemilik harus berada dalam satu tempat dengan tanaman miliknya. Tujuan dari penelitian ini adalah untuk merancang aplikasi pengontrol sistem penyiraman tanaman otomatis yang berbasis android dan arduino supaya dapat digunakan secara mobilitas. Aplikasi dirancang dengan metode SDLC waterfall menggunakan toolkit milik Flutter dengan bahasa perograman dart. Sedangkan perancangan sistem menggunakan software Arduino IDE. Hasil dari penelitian ini berupa aplikasi android yang dapat memonitor nilai kelembapan tanah pada tanaman dalam satuan waktu dan tombol on/off yang dapat menghidupkan dan mematikan pompa air penyiram tanaman. Pengguna dapat melihat nilai kelembapan tanah secara realtime melalui aplikasi android dengan menggunakan koneksi internet. Penggunaan ini menggunakan server firebase sebagai database dengan memanfaatkan fitur realtime firebase database milik server firebase. Sistem dan aplikasi yang telah dirancang diuji kegunaannya dengan pengujian terakhir aplikasi menggunakan usability testing. Hasil dari survey pada usability testing mendapatkan hasil bahwa aplikasi berfungsi sebagaimana semestinya dan mendapat kategori "Sangat Baik" sebesar 94,44% dari responden.

Kata Kunci— *Watering plants; Android; Arduino; Firebase; Flutter*

I. PENDAHULUAN

Manusia bukan merupakan satu-satunya makhluk hidup yang ada di bumi. Manusia hidup di bumi berdampingan dengan makhluk hidup lainnya, salah satunya ialah tanaman. Keberadaan tanaman sebagai makhluk hidup sangat penting bagi kehidupan manusia. Tanaman memiliki banyak manfaat yang berguna untuk manusia.

Naskah diterima 16 Juni 2021, diterima setelah revisi 29 Juni 2021, terbit online 25 Februari 2022. Emitor merupakan jurnal Teknik Elektro Universitas Muhammadiyah Surakarta yang terakreditasi Sinta 4 dengan alamat Gedung H Lantai 2 UMS, Jalan Ahmad Yani Tromol Pos 1 Surakarta Indonesia 57165.

Untuk mendapatkan manfaat tersebut, tanaman harus tetap hidup tumbuh. Faktor penunjang tumbuh dan berkembangnya tanaman ialah pada proses penyiraman [1]. Tanaman bergantung kepada air untuk dapat bertahan hidup [2]. Air dari dalam tanah akan diserap oleh tanaman sesuai jumlah yang dibutuhkan. Apabila kandungan air dalam tanah sangat kering dapat menyebabkan tanaman menjadi layu, sedangkan jika kandungan air dalam tanah sangat berlebihan dapat menyebabkan turunnya kadar oksigen dalam tanah [3]. Kedua hal tersebut dapat berakibat buruk pada kondisi tanaman yang dapat menyebabkan tanaman menjadi

mati. Kurang optimalnya perawatan tanaman tersebut dapat disebabkan oleh faktor waktu dan tenaga bagi pemilik tanaman [3]. Fleksibilitas waktu yang dimiliki pemilik tanaman dan sifat dasar manusia yang mudah lupa berdampak pada kelangsungan hidup tanaman.

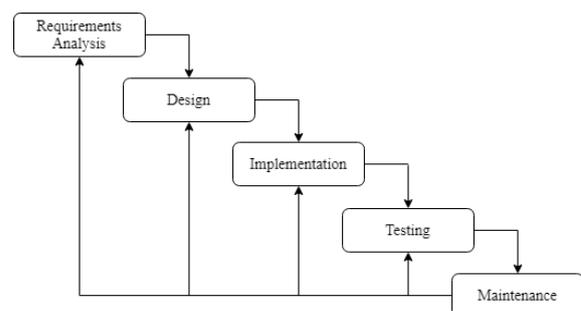
Proses penyiraman tanaman pada saat ini masih dilakukan secara manual dengan menyalakan pompa air secara manual maupun dengan mengambil air dari sumur yang masih membutuhkan waktu dan tenaga pemilik tanaman [4]. Cara tersebut sangat tidak efisien karena pemilik tanaman harus mengontrol secara langsung di taman atau kebun. Era globalisasi menuntut manusia untuk hidup berdampingan dengan teknologi. Seluruh kegiatan manusia hampir tidak pernah luput dari teknologi [5]. Kemajuan dan perkembangan teknologi menciptakan ponsel pintar yang biasa disebut sebagai *smartphone*. *Smartphone* sendiri menggunakan sistem operasi yang berbeda-beda. Namun sistem operasi yang sedang banyak digunakan adalah sistem operasi milik google yaitu Android [6]. Saat ini hampir seluruh dunia menggunakan *smartphone* untuk beraktivitas [4]. Sebelumnya telah dilakukan penelitian oleh [4] dengan membuat Sistem Penyiram Perkebunan Berbasis Android dan Arduino Uno untuk menjalankan sistem. Peneliti menggunakan Modul GSM untuk mengirimkan SMS sebagai perintah menyalakan dan mematikan pompa ketika sedang berada jauh dari tanaman. Selain itu [3] melakukan penelitian dengan membuat Alat Penyiram Tanaman Otomatis menggunakan Sensor Kelembaban Tanah dengan Notifikasi Whastapp. Pengujian yang dilakukan peneliti terdiri dari pengujian sensor *soil moisture*, pengujian lampu led, pengujian lcd, pengujian relay, pengujian pompa air dan pengujian whatsapp. Hasil pengujian didapatkan apabila kelembaban tanah yang didapat sensor *soil moisture* melebihi 6,5 maka pompa air akan menyala secara otomatis. Peneliti akan mendapat notifikasi melalui whatsapp setiap proses penyiraman dan memperoleh informasi kelembaban tanah melalui lcd. Sistem Penyiram Tanaman Pintar Menggunakan *Smartphone* dan Mikrokontroler Arduino Berbasis *Internet of Thing* juga telah dibuat [7]. Sistem tersebut dirancang peneliti menggunakan aplikasi *smartphone* Blynk yang telah terhubung dengan mikrokontroler Arduino uno R3. Dilengkapi dengan pompa air yang telah terhubung dengan Relay 5V. Apabila nilai kandungan tanah di atas nilai 900 maka kondisi tanah membutuhkan air sedangkan jika nilai dibawah nilai 400 maka kandungan tanah sedang lembab. Dengan *smartphone* yang telah terinstall aplikasi Blynk peneliti dapat menggunakan tombol on/off dalam aplikasi tersebut untuk

menyiram tanaman.

Berdasar rumusan masalah dan penelitian sebelumnya, penulis akan merancang Aplikasi Pengontrol Sistem Penyiraman Tanaman Otomatis berbasis Arduino dan Android. Sensor *moisture soil* akan diletakkan di dalam tanah yang terhubung dengan mikrokontroler ESP32 berbasis arduino. Hasil pembacaan sensor kelembaban tanah yang diterima oleh mikrokontroler akan diupdate ke *realtime database* menggunakan platform Firebase. Kemudian data dari pembacaan sensor tersebut akan dipanggil ke dalam aplikasi android yang telah dirancang menggunakan bahasa pemrograman dart berbasis Flutter. Selain memberikan output data dari *realtime database*, aplikasi android tersebut juga dapat memberikan input button ke *realtime database* yang akan dibaca oleh mikrokontroler untuk diteruskan ke relay guna menyalakan dan mematikan pompa air. Perbedaan dengan penelitian yang telah dilakukan sebelumnya yaitu sistem penyiram tanaman ini dapat dilakukan pengontrolan kelembaban tanah dengan menyalakan tombol on/off secara langsung dari aplikasi *mobile* berbasis android, serta dapat memonitor pergerakan kelembaban tanah secara *realtime* yang berubah setiap saat. Sistem ini dapat mengontrol area dengan luas $1m^2$. Dalam area $1m^2$ tersebut terdapat 6 tanaman dalam bentuk pot. Diharapkan sistem ini dapat mempermudah pemilik tanaman untuk mengontrol tanaman dari jarak jauh.

II. METODE PENELITIAN

Metode pengembangan yang digunakan penulis pada sistem ini adalah *SDLC (System Development Life Cycle)* dengan model *waterfall*. Metode SDLC dengan model *waterfall* merupakan model terbaik dimana setiap tahap dimulai dan diakhiri sebelum tahap selanjutnya dikerjakan sehingga setiap tahapan tidak tumpang tindih dalam proses pengerjaannya [8]. Gambaran metode *System Development Life Cycle* [9] dijelaskan pada Gambar 1 sebagai berikut.

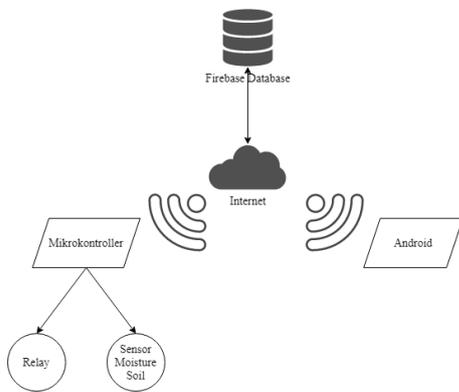


Gambar 1: Tahapan Metode *Waterfall*

i. Requirements Analysis

Kebutuhan sistem perangkat lunak dan perangkat keras dialokasikan pada tahap perancangan sistem dengan membentuk arsitektur sistem secara keseluruhan [10]. Desain arsitektur sistem dan aplikasi ditampilkan dalam bentuk Arsitektur Sistem, Sketsa Lahan, *Use Case Diagram*, dan *Activity Diagram* untuk memudahkan pemahaman *user*.

Arsitektur sistem dari pengontrol penyiraman tanaman ini terdiri dari *software* dan *hardware*. *Software* yang digunakan dalam sistem meliputi *software* yang terdapat di dalam mikrokontroler, *software* aplikasi dalam android, serta *server database* untuk penyimpanan *database*. *Hardware* yang digunakan dalam sistem ini terdiri dari mikrokontroler, sensor moisture soil, relay, dan *smartphone* android. Gambaran arsitektur sistem dijelaskan dalam Gambar 2.

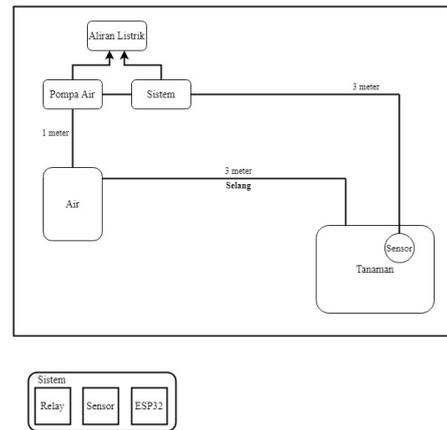


Gambar 2: Arsitektur Sistem

Sistem ini menggunakan sensor moisture soil FC-28 yang terhubung secara langsung dengan mikrokontroler dan bekerja dengan membaca data kelembaban tanah untuk kemudian dikirim ke mikrokontroler. Sensor moisture soil FC-28 memiliki spesifikasi tegangan input 3,3 Volt atau 5 Volt, tegangan output 0 – 4,2 Volt, arus 35 mA dan value range ADC sebesar 1024 bit [11].

Mikrokontroler yang digunakan adalah ESP32 berbasis Arduino dengan spesifikasi RAM 520K, Prosesor Tensilica L108 32 bit, Kecepatan Prosesor Dual 160MHz, tegangan 3,3 Volt, dan dilengkapi WiFi yang tertanam dalam mikrokontroler [12]. Mikrokontroler diprogram dengan bahasa pemrograman C menggunakan *software* Arduino IDE [11] untuk memberi perintah kepada sensor dan relay.

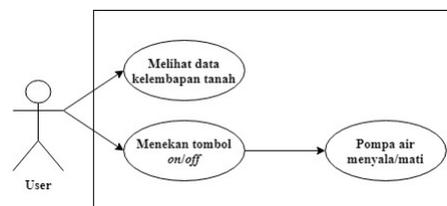
Perintah yang didapat relay dari mikrokontroler adalah untuk menyalakan atau mematikan pompa air. Pompa air terhubung dengan relay dan arus listrik secara langsung. Relay tersebut memiliki spesifikasi te-



Gambar 3: Implementasi Sketsa Lahan

gangan AC 125 V atau 250 V, arus 10 A, dan daya 1250 W atau 2500 W serta tegangan DC 28 V atau 30 V, arus 10 A, dan daya 128 W atau 300 W [13].

Data hasil pembacaan sensor yang diterima oleh mikrokontroler kemudian akan diteruskan ke Server Firebase. Firebase merupakan platform milik Google yang memberikan layanan servis dan alat yang dibutuhkan *web development* maupun *mobile development* untuk mengembangkan aplikasi [14]. Data kelembaban tanah tersebut disimpan dalam platform firebase yang berupa *hosting cloud* atau yang biasa disebut *firebase realtime database* [15].



Gambar 4: Use case diagram

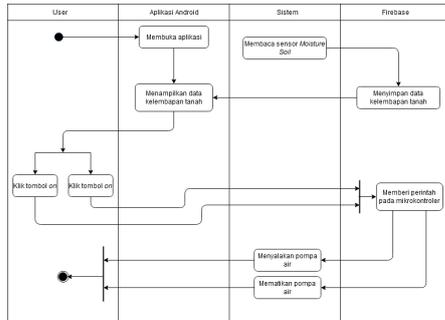
Selain menyimpan data kelembaban tanah, *firebase realtime database* juga menyimpan perintah menyalakan dan mematikan pompa air untuk diteruskan ke mikrokontroler. Data kelembaban tanah dan perintah on/off pompa air ditampilkan dalam aplikasi android yang telah diprogram menggunakan toolkit flutter. Android, server firebase dan mikrokontroler harus terhubung dengan internet untuk dapat menjalankan sistem dengan baik.

Berdasarkan arsitektur sistem pada Gambar 2, dibuatlah sketsa implementasi lahan yang dapat dilihat pada Gambar 3. Terdapat ruangan berisi mikrokontroler yang terhubung dengan arus listrik dan relay di sampingnya. Relay terhubung langsung dengan pompa air untuk mengambil air pada galon di bawahnya pada jarak 1 m, sedangkan sensor moisture soil tertanam

dalam tanaman dengan luas $1m^2$ di luar ruangan.

Interaksi antara *user* dengan sistem dijelaskan dalam *use case diagram*. Interaksi yang dijelaskan dalam *use case* berupa apa yang dilakukan *user* terhadap sistem dan apa yang disampaikan sistem kepada *user* [16]. *Use case diagram* sistem dapat dilihat pada Gambar 4.

Interaksi *user* dengan sistem dijelaskan dalam *activity diagram* yang dapat dilihat pada Gambar 5.



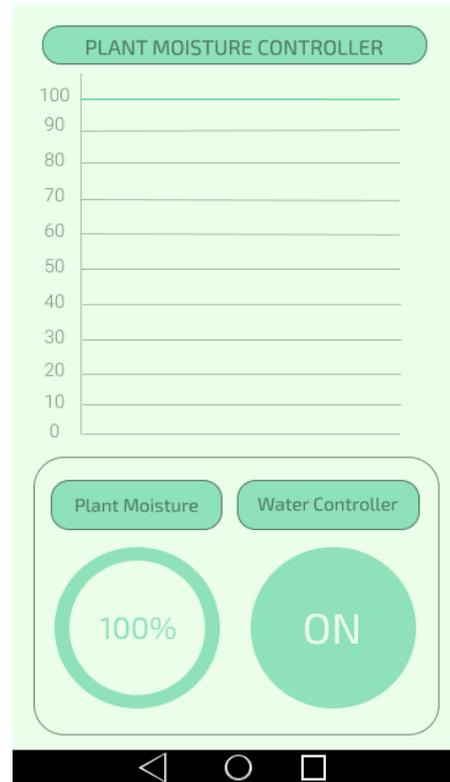
Gambar 5: Activity Diagram

Sistem membaca sensor kelembaban tanah secara berkala tanpa henti ketika mendapat jaringan internet. Data pembacaan sensor akan disimpan dalam realtime firebase database. Saat aplikasi dalam android dijalankan, maka data yang tersimpan dalam database akan ditampilkan. Selain menampilkan data pembacaan sensor, aplikasi juga dapat memberi perintah *on/off* untuk menyalakan atau mematikan pompa air. Ketika tombol *on* dipilih, maka perintah akan diinput ke database guna diteruskan ke sistem untuk menyalakan pompa air. Begitu pula jika tombol *off* dipilih, maka perintah sebaliknya yang akan dikerjakan oleh sistem. Desain *User Interface* dari aplikasi android dapat dilihat pada Gambar 6, yang meliputi grafik data kelembaban tanah dalam satuan waktu, data kelembaban tanah dalam persen, serta *button on/off* untuk menyalakan atau mematikan pompa.

Setelah melewati tahap analisis kebutuhan dan desain, tahap selanjutnya adalah implementasi. Alur kerja dari aplikasi dan sistem secara keseluruhan ditampilkan dalam bentuk *flowchart* seperti Gambar 7.

ii. Implementation

Perancangan sistem ini diawali dengan merancang aplikasi yang akan dibuat menggunakan *toolkit flutter* dan bahasa pemrograman *dart* dengan *software aplikasi Visual Code*. Database yang telah dibuat di *server firebase* memiliki *JSON file* yang nantinya akan digunakan dalam proses pembuatan aplikasi. File *JSON* yang telah didownload diinput ke dalam folder aplikasi yang dibuat. Koneksi dengan *firebase*



Gambar 6: Desain User Interface

kemudian dibuat dengan fungsi `final dbRef = FirebaseDatabase.instance.reference();`. Fungsi tersebut nantinya akan digunakan untuk mengambil data dalam *firebase* dan menginput data boolean ke *firebase*.

Menggunakan fungsi yang telah dibuat, data dalam *firebase database* dipanggil dengan metode *stream* yang dideklarasikan `stream: dbRef.child(\Data").onValue,`. Metode *stream* tidak lepas dari metode *builder*. Pada metode *builder*, library yang digunakan ialah *library snapshot* dengan fungsi untuk mengimplementasikan data dari *JSON* ke *class data*. Jika hasil *snapshot* ditemukan data dan data tersebut tidak *error* serta *value* dari data tidak kosong `if (snapshot.hasData !snapshot.hasError snapshot.data.snapshot.value != null)` maka data akan muncul dengan memanggil metode *return*.

Dengan fungsi yang sama, dibuatlah metode baru untuk memberikan *input* ke *firebase database*. Sebelum membuat metode baru tersebut, digunakan `framework setState() value = !value;` yang berfungsi mengubah objek yang akan diinput antara *true* atau *false*. Selanjutnya dibuat metode yang mendeklarasikan fungsi dan `framework writeData() dbRef.child (\Water").set(\Status": !value);` untuk memberi input ke *firebase* yang mana akan

diterima oleh mikrokontroller dan perintah `\Status" : !value` tersebut dikerjakan oleh relay. Aplikasi tersebut kemudian dihubungkan dengan sistem yang dibuat menggunakan *software* Arduino IDE pada mikrokontroller ESP32.

Sebelumnya mikrokontroller dihubungkan dengan relay dan sensor moisture soil menggunakan kabel *jumper*. Relay terhubung dengan mikrokontroller pada *pin out* 27 menggunakan kabel *jumper* serta terhubung dengan pompa air yang langsung teraliri arus listrik. Sensor moisture soil terhubung dengan mikrokontroller pada *pin out* A0 menggunakan kabel *jumper*. Ketika lahan yang menjadi target memiliki jarak yang cukup jauh, maka kabel *jumper* pada sensor disambung menggunakan kabel LAN.

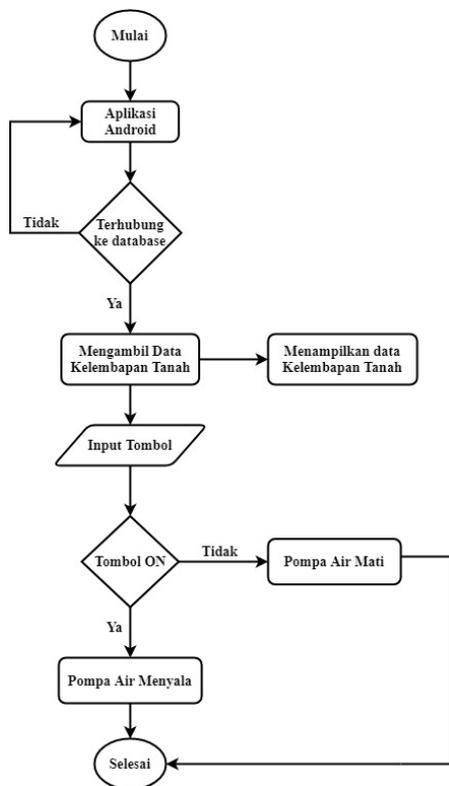
Rangkaian *hardware* yang telah terpasang selanjutnya diinisialisasi menggunakan *software* Arduino IDE pada mikrokontroller ESP32. Mikrokontroller diinisialisasi dengan jaringan WiFi rumah dan *server firebase* serta dihubungkan dengan sensor moisture soil dan relay. Inisialisasi dengan jaringan WiFi rumah dilakukan dengan memasukkan WiFi SSID dan *password* WiFi pada *software* Arduino IDE. Selanjutnya proses inisialisasi terhadap *server firebase* dengan memasukkan *host* dan *auth* yang terdapat dalam *server firebase* pada *project* masing-masing. Setelah terinisialisasi dengan

WiFi dan *firebase*, mikrokontroller melalui *pin out*nya terhubung dengan sensor moisture soil dan relay menggunakan kabel *jumper*. Dengan kabel *jumper* relay terhubung dengan mikrokontroller pada *pin* 27 `const int relay = 27;` dan sensor moisture soil pada *pin* A0 `const int moisturePin = A0;`

Data kelembaban tanah didefinisikan dengan `int moisturePercentage;` dan selanjutnya dipanggil untuk diolah menggunakan perhitungan rumus $moisture\% = (100.00 - ((analogRead(moisturePin)/4095.00) \times 100\%));$. Data hasil perhitungan akan diteruskan ke server *firebase* `Firestore.setDouble(firebaseData, "/Data/Moisture", moisturePercentage);` sesuai dengan nama *database* yang telah dibuat pada *project firebase*. Apabila sensor memberikan data, maka berbanding terbalik dengan relay yang mendapatkan data untuk kemudian dikerjakan. Jika data yang didapat relay berasal dari *firebase* berupa `Firestore.getBool(firebaseData, "Water/Status")` maka akan dibaca relay dengan tipe data Boolean `RelayStatus = firebaseData.boolData();` Relay berada pada kondisi active low, sehingga jika data true maka relay akan menyala `if (RelayStatus == true)digitalWrite(relay,LOW);` sedangkan jika data false maka relay akan mati `else digitalWrite(relay,HIGH);`. Mikrokontroller yang telah selesai diprogram kemudian dihubungkan dengan arus listrik secara langsung untuk selanjutnya sistem dapat bekerja bersamaan dengan aplikasi yang telah dirilis.

iii. Testing

Setelah proses pembuatan sistem pada tahap implementasi dilakukan proses pengujian atau *testing*. Pengujian dilakukan setelah aplikasi selesai dibuat guna menemukan kesalahan dan menilai efisiensi serta keefektifan aplikasi. Selain itu pengujian juga dilakukan terhadap sensor dan relay yang digunakan. Jenis pengujian yang digunakan pada aplikasi android yaitu black box testing. Pengujian black box testing berpusat pada spesifikasi fungsional perangkat lunak [17]. Pengujian pada sensor dan relay dilakukan dengan melihat tingkat keberhasilan dan keakuratan perangkat keras terhadap input yang diberikan. Pengujian pada perangkat keras dilakukan dengan menghubungkan ESP32 ke laptop. Sensor moisture soil yang terhubung dengan ESP32 akan memberikan hasil pembacaan yang akurat dimulai dari angka 0 ketika tidak terkena air, dan akan bertambah perlahan jika terkena air. Ha-



Gambar 7: Flowchart



Gambar 8: Tampilan Splash screen

sil tersebut akan terlihat dalam Serial Monitor pada Arduino IDE. Setelah diuji melalui Arduino IDE, sensor kemudian diuji dengan firebase apakah nilai hasil pembacaan sensor dapat ditampilkan dalam database.

Pengujian pada relay dilakukan melalui firebase database yang telah terhubung dengan ESP32. Dalam firebase diberikan input terhadap relay untuk kemudian dikerjakan. Jika input yang diberikan TRUE maka lampu pada relay akan menyala dan Serial Monitor menampilkan nilai 1. Jika input yang diberikan FALSE maka lampu pada relay akan mati dan Serial Monitor menampilkan nilai 0. Selanjutnya relay diuji dalam kondisi telah terhubung dengan pompa air. Pengujian ini dilakukan untuk mengetahui apakah input yang diberikan ke relay berpengaruh terhadap pompa air. Jika input yang diberikan melalui firebase berupa TRUE maka pompa air akan bekerja, dan jika input berupa FALSE maka pompa air akan berhenti bekerja.

Pengujian selanjutnya dilakukan pada aplikasi android dan sistem secara bersamaan. Hal ini dilakukan untuk mengetahui apakah aplikasi dan sistem dapat berjalan sesuai yang diharapkan. Aplikasi akan menerima hasil pembacaan sensor dari firebase. Nilai akan akurat jika aplikasi menampilkan data *realtime* yang selalu berubah sesuai dengan yang terdapat dalam firebase. Dalam aplikasi juga

terdapat button yang jika ditekan ON mengartikan pompa air akan bekerja dan jika ditekan OFF pompa air akan berhenti bekerja. Pengujian terakhir yaitu *usability testing* dengan menggunakan instrumen penelitian SUPR-Q (Standardized Universal Percentilr Rank Questionnarie) sebagai tolak ukur bagaimana sebuah perangkat lunak dikatakan mudah saat digunakan [18].

iv. Maintenance

Tahap akhir dari metode *waterfall* adalah *maintenance* atau pemeliharaan. Pemeliharaan dilakukan pada sistem yang sudah jadi untuk mengatasi munculnya *bug* atau *error*. Pengecekan terhadap sistem dilakukan secara berkala pada pemeliharaan guna menjaga kualitas aplikasi tetap baik.

III. HASIL PENELITIAN DAN DISKUSI

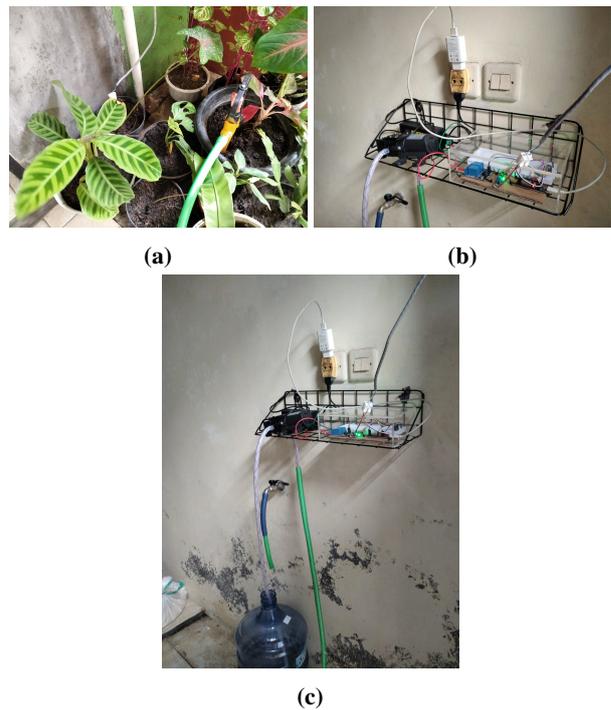
Bagian ini membahas tentang implemmentasi aplikasi dan sistem serta pengujiannya. Saat aplikasi dibuka pertama kali akan muncul splash screen seperti pada Gambar 8. Splash screen ini berfungsi sebagai loading sebelum masuk ke halaman utama pada aplikasi. Tampilan dalam splash screen tersebut berupa logo pohon dengan tulisan “Plant” di atasnya. Logo tersebut melambangkan tumbuhan yang berkaitan dengan aplikasi ini. Setelah melewati splash screen muncul halaman utama aplikasi seperti pada Gambar 9. Dalam halaman utama aplikasi terdapat data kelembaban tanah berbentuk grafik dan dalam bentuk progress control dengan label “Plant Moisture”. Terdapat 3 warna indikator dalam progress control berupa warna merah, oranye, dan hijau sesuai dengan data yang direspon. Tombol dalam halaman utama aplikasi juga akan berubah ketika ditekan. Jika tombol dalam keadaan OFF ditekan, maka tombol tersebut akan berubah tampilan menjadi tombol ON dan begitu juga sebaliknya. Implementasi dari sistem adalah terhubungnya seluruh perangkat keras seperti relay dan sensor moisture soil ke ESP32, relay dengan pompa air serta sensor moisture soil dengan tanaman seperti pada Gambar 10.

Pengujian dilakukan sebanyak 6 tahap. Pada aplikasi pengujian dilakukan menggunakan black box testing. Pengujian terhadap sensor dan relay masing-masing 2 tahap. Tahap pengujian terakhir yaitu pada keseluruhan sistem dan aplikasi. Black box testing digunakan sebagai pengujian pada aplikasi. Berikut hasil dari black box testing pada Tabel 1. Pengujian tahap 1 dilakukan terhadap sensor moisture soil.

Tabel 1: Black box testing

Kelas Uji	Pengujian	Input	Output	Hasil
Mulai aplikasi	User membuka aplikasi	Menekan logo aplikasi	Menampilkan splash screen	Valid
Tampilan awal aplikasi	Loading	Loading	Menampilkan halaman utama	Valid
Halaman utama aplikasi	Melihat data kelembapan tanah	Tidak ada input dari user	Menampilkan data kelembapan tanah secara berkala dalam bentuk grafik dan indikator persen	Valid
Halaman utama aplikasi	User menekan tombol pada water controller	Menekan tombol ON	Tombol men-switch menjadi tombol OFF	Valid
Halaman utama aplikasi	User menekan tombol pada water controller	Menekan tombol OFF	Tombol men-switch menjadi tombol ON	Valid
Keluar dari aplikasi	User menutup aplikasi	Menekan tombol keluar pada perangkat mobile	Menutup aplikasi	Valid

Pengujian dilakukan untuk mengetahui apakah sensor berfungsi dengan baik dalam pembacaan kelembapan tanah. Pada tahap ini pengujian sensor dimonitor menggunakan *software* Arduino IDE. Pengujian dilakukan pada 04 Juni 2021 sebanyak 60 kali percobaan dengan pembagian waktu yang berbeda. Hasil dari pengujian sensor menampilkan nilai 40% - 45% pada pukul 07.24 - 07.25 WIB dengan kondisi tanah yang lembab, lalu nilai 55% - 54% pada pukul 12.09 - 12.10 WIB dengan kondisi tanah masih lembab, kemudian nilai 53% - 51% pada pukul 16.02 - 16.03 WIB dengan kondisi tanah yang masih tetap lembab, dan nilai 65% - 63% pada pukul 20.15 - 20.16 WIB dengan kondisi tanah



Gambar 10: Implementasi perangkat keras (a) Hubungan sensor moisture soil dengan tanaman (b) Hubungan relay dan sensor moisture soil ke ESP32 (c) Hubungan relay dengan pompa air

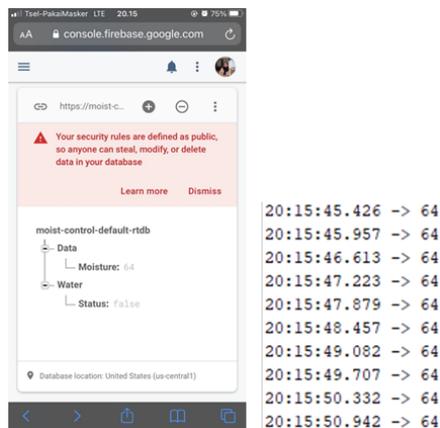


Gambar 9: Halaman utama aplikasi

yang basah. Hasil pembacaan sensor tersebut sesuai dengan kondisi tanah dimana jika nilai yang ditampilkan kurang dari 20% maka kondisi tanah kering dan membutuhkan air, lalu jika nilai yang ditampilkan antara 20% dan 60% maka kondisi tanah lembab, sedangkan jika nilai yang ditampilkan lebih dari 60% maka kondisi tanah masih basah.

Setelah dilakukan pengujian menggunakan *software* Arduino IDE, sensor diuji dengan membandingkan antara nilai yang muncul pada serial monitor Arduino IDE dan nilai yang muncul pada firebase. Pengujian dilakukan secara bersamaan dengan pengujian sensor sebelumnya terhadap tanggal, waktu, dan banyak-

nya percobaan. Nilai yang muncul pada firebase selalu sama dengan nilai yang muncul pada serial monitor seperti pada Gambar 11.



Gambar 11: Perbandingan nilai pada firebase dan serial monitor

Selanjutnya pengujian terhadap relay. Pengujian dilakukan dengan memberi input perintah berupa *data boolean* dari firebase database. Pengujian relay yang pertama dilihat dari respon indikator LED yang terdapat pada relay. Jika input yang diberikan TRUE maka indikator LED akan menyala, dan jika FALSE maka indikator LED akan mati. Pengujian dilakukan pada tanggal 01 Juni 2021 dan 03 Juni 2021 sebanyak 31 kali percobaan dengan rentang waktu yang berbeda. Hasil dari pengujian tersebut menyatakan bahwa ketika input TRUE diberikan maka LED akan menyala dan ketika FALSE maka LED akan mati dengan delay 0 detik atau tanpa delay.

Jika pengujian relay terhadap indikator LED telah selesai, selanjutnya dilakukan pengujian relay terhadap pompa air. Seperti dengan pengujian relay sebelumnya, input diberikan dari firebase berupa kondisi TRUE dan FALSE. Apabila input TRUE maka pompa air akan menyala dan bekerja. Apabila input FALSE maka pompa air akan mati dan berhenti bekerja. Pengujian dilakukan secara bersamaan dengan pengujian relay sebelumnya terhadap tanggal, waktu, dan banyaknya percobaan. Hasil pengujian menyatakan bahwa ketika diberi input TRUE maka pompa air menyala dan bekerja, sedangkan jika diberi input FALSE maka pompa air akan mati dan berhenti bekerja dengan delay 0 detik atau tanpa delay.

Tahap selanjutnya adalah menguji sistem dan aplikasi android secara keseluruhan. Hasil pengujian ini menentukan apakah sistem dan aplikasi android telah bekerja sesuai dengan yang diharapkan. Pengujian dilakukan pada tanggal 04 Juni 2021 sebanyak 60 kali

percobaan dengan pembagian waktu yang berbeda. Hasil dari percobaan tersebut menyatakan bahwa nilai yang ditampilkan pada aplikasi android bernilai sama dengan nilai yang ditampilkan dalam firebase. Ketika tombol ON ditombol maka relay akan menyala dan menyiram, sedangkan jika tombol OFF ditombol maka relay akan mati dan berhenti menyiram dengan delay 0 detik atau tanpa delay.

Setelah dilakukan beberapa pengujian sistem, selanjutnya dilakukan usability testing dengan menyebarkan kuisioner terhadap 30 responden dari minimal 20 responden [19]. Dilakukan pada orang-orang di lingkungan tempat pemasangan *hardware* karena harus menyaksikan *hardware* dan lahan untuk menilai apakah aplikasi berjalan dengan semestinya. Pengujian aplikasi dilakukan dalam rentang waktu 2 – 3 menit setiap sekali uji pada pengguna android usia 18 – 58 tahun. Dari survey tersebut didapat perolehan hasil perhitungan pada kategori “Baik” sebesar 10% dan kategori “Sangat Baik 90%” dengan proses perhitungan berdasarkan SUPR-Q. Rata-rata presentase yang diperoleh dari perhitungan berkeseluruhan ialah 94,44% dengan kategori “Sangat Baik”.

IV. KESIMPULAN

Aplikasi pengontrol sistem penyiraman tanaman ini telah selesai dirancang. Dengan dikombinasikan bersama perangkat keras ESP32 yang saling terhubung dengan sensor moisture soil dan relay. Aplikasi ini memiliki fitur untuk melihat data kelembaban tanaman secara *realtime* dari sensor dan tombol untuk mengontrol aktifitas pompa air yang terhubung ke relay. Berdasarkan hasil pengujian black box fitur dalam aplikasi telah berjalan dengan semestinya. Dari usability testing dihasilkan rata-rata presentase sebesar 94,4% dengan kategori “Sangat Baik” yang artinya para responden setuju aplikasi ini membantu pekerjaan mereka.

DAFTAR PUSTAKA

- [1] N. Azzaky dan A. Widiantoro, “Alat penyiram tanaman otomatis berbasis arduino menggunakan internet of things (iot),” *J-Eltrik*, vol. 2, no. 2, pp. 86–91, 2020.
- [2] K. Sreeram, R. S. Kumar, S. V. Bhagavath, K. Muthumeenakshi, dan S. Radha, “Smart farming—a prototype for field monitoring and automation in agriculture,” in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 2189–2193.
- [3] Y. F. Hidayat, A. H. Hendrawan, dan R. Ritzkal, “Purwarupa alat penyiram tanaman otomatis menggunakan sensor kelembaban tanah dengan notifikasi whatsapp,” *Prosiding Semnastek*, 2019.

- [4] D. Satria, "Desain prototype penyiraman perkebunan berbasis android," *Jaringan Sistem Informasi Robotik-JSR*, vol. 3, no. 2, pp. 250–256, 2019.
- [5] R. Ridarmin dan Z. P. Pertiwi, "Prototype penyiram tanaman hias dengan soil moisture sensor berbasis arduino," *Informatika*, vol. 10, no. 1, pp. 7–11, 2018.
- [6] I. Warangkiran, S. T. Kaunang, A. S. Lumenta, dan A. M. Rumagit, "Perancangan kendali lampu berbasis android," *Jurnal Teknik Elektro Dan Komputer*, vol. 3, no. 1, pp. 65–72, 2014.
- [7] A. S. Pambudi, S. Andryana, dan A. Gunaryati, "Rancang bangun penyiraman tanaman pintar menggunakan smartphone dan mikrokontroler arduino berbasis internet of thing," *Jurnal Media Informatika Budidarma*, vol. 4, no. 2, pp. 250–256, 2020.
- [8] A. Alshamrani dan A. Bahattab, "A comparison between three sdlc models waterfall model, spiral model, and incremental/iterative model," *International Journal of Computer Science Issues (IJCSI)*, vol. 12, no. 1, p. 106, 2015.
- [9] W. Hardyanto, A. Purwinarko, F. Sujito, D. Alighiri *et al.*, "Applying an mvc framework for the system development life cycle with waterfall model extended," in *Journal of Physics: Conference Series*, vol. 824, no. 1. IOP Publishing, 2017, p. 012007.
- [10] G. W. Sasmito, "Penerapan metode waterfall pada desain sistem informasi geografis industri kabupaten tegal," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 2, no. 1, pp. 6–12, 2017.
- [11] H. Husdi, "monitoring kelembaban tanah pertanian menggunakan soil moisture sensor fc-28 dan arduino uno," *ILKOM Jurnal Ilmiah*, vol. 10, no. 2, pp. 237–243, 2018.
- [12] H. Kusumah dan R. A. Pradana, "Penerapan trainer interfacing mikrokontroler dan internet of things berbasis esp32 pada mata kuliah interfacing," *Journal Cerita*, vol. 5, no. 2, pp. 120–134, 2019.
- [13] M. R. Nurkamiden, M. E. Najoan, dan M. D. Putro, "Rancang bangun sistem pengendalian perangkat listrik berbasis web server menggunakan mini pc raspberry pi studi kasus gedung fakultas teknik universitas sam ratulangi," *Jurnal Teknik Informatika*, vol. 11, no. 1, 2017.
- [14] A. Rajappa, A. Upadhyay, A. S. Sabitha, A. Bansal, B. White, dan L. Cottrell, "Implementation of pinger on android mobile devices using firebase," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 698–703.
- [15] A. Al-Kababji, L. Shidqi, I. Boukhenoufa, A. Amira, F. Bensaali, M. S. Gastli, A. Jarouf, W. Aboueata, dan A. Abdalla, "Iot-based fall and ecg monitoring system: wireless communication system based firebase realtime database," in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI)*. IEEE, 2019, pp. 1480–1485.
- [16] M. Muslihudin, W. Renvillia, T. Taufiq, A. Andoyo, dan F. Susanto, "Implementasi aplikasi rumah pintar berbasis android dengan arduino microcontroller," *Jurnal Keteknik dan Sains (JUTEKS)*, vol. 1, no. 1, pp. 23–31, 2018.
- [17] T. Hidayat dan M. Muttaqin, "Pengujian sistem informasi pendaftaran dan pembayaran wisuda online menggunakan black box testing dengan metode equivalence partitioning dan boundary value analysis," *Jutis (Jurnal Teknik Informatika)*, vol. 6, no. 1, pp. 25–29, 2020.
- [18] B. Pramitasari dan N. Nurgiyatna, "Sistem informasi unit kegiatan mahasiswa marching band universitas muhammadiyah surakarta berbasis web," *Emitor: Jurnal Teknik Elektro*, vol. 19, no. 2, pp. 59–65, 2019.
- [19] T. Kallio, A. Kaikkonen *et al.*, "Usability testing of mobile applications: A comparison between laboratory and field testing," *Journal of Usability studies*, vol. 1, no. 4-16, pp. 23–28, 2005.