

# Monitoring Smart Applications for Monitoring and Controlling of IoT-Based Strawberry Hydroponic Plants

Willi Bianyosa Arif Wibiya<sup>a,1</sup>, Aris Nasuha<sup>a,2,\*</sup>

<sup>a</sup> Department of Electrical and Electronic Engineering, Vocational Faculty, UNY

<sup>1</sup> [willibianyosa.2019@student.uny.ac.id](mailto:willibianyosa.2019@student.uny.ac.id); <sup>2</sup> [arisnasuha@uny.ac.id](mailto:arisnasuha@uny.ac.id)

\* Corresponding Author

## ARTICLE INFO

### Article History

Received 30 Aug. 2023

Revised 03 Oct. 2023

Accepted 01 Dec. 2023

### Keywords

Hydroponic,  
Internet of Things,  
Smart Farming,  
IoT Application,  
IoT Dashboard.

## ABSTRACT

Hydroponic farming has become an attractive option in strawberry cultivation as it provides better control over the growing environment of the plants. However, proper management and continuous monitoring are challenges often faced by farmers in hydroponic farming practices. Therefore, in this study, we designed an IoT-based application that aims to monitor and control strawberry hydroponic farming through smartphone devices. The method used in this research involves developing an application that connects with various sensors installed in the hydroponic farm environment. The data obtained from the sensors is transmitted in real-time through the IoT network. In addition to the monitoring function, the application is also equipped with a control system that can be customized according to the needs of farmers. The result of this research is an application that can work from a short distance (local) or a long distance (cloud). The application is equipped with a hazard alarm notification system. The feature allows farmers to take action to maintain the health and quality of strawberry plants. The use of this application is expected to help make it easier for farmers to monitor and control hydroponic farming to be more efficient and optimize crop yields.

Pertanian hidroponik telah menjadi pilihan yang menarik dalam budidaya stroberi karena memberikan kontrol yang lebih baik terhadap lingkungan tumbuh tanaman. Namun, pengelolaan yang tepat dan pemantauan yang berkelanjutan merupakan tantangan yang sering dihadapi oleh petani dalam praktik pertanian hidroponik. Oleh karena itu dalam penelitian ini kami merancang sebuah aplikasi berbasis IoT yang bertujuan untuk memantau dan mengendalikan pertanian hidroponik stroberi melalui perangkat smartphone. Metode yang digunakan dalam penelitian ini melibatkan pengembangan aplikasi yang terhubung dengan berbagai sensor yang terpasang di lingkungan pertanian hidroponik. Data yang diperoleh dari sensor dikirimkan secara real-time melalui jaringan IoT. Selain fungsi pemantauan, aplikasi juga dilengkapi dengan sistem kontrol yang dapat disesuaikan sesuai dengan kebutuhan petani. Hasil dari penelitian ini berupa aplikasi yang dapat bekerja dari jarak dekat (lokal) maupun jarak jauh (cloud). Aplikasi dilengkapi dengan sistem notifikasi alarm bahaya. Dimana fitur ini memungkinkan petani untuk mengambil tindakan guna menjaga kesehatan dan kualitas tanaman stroberi. Dengan penerapan aplikasi ini, diharapkan dapat membantu memudahkan petani dalam memantau dan mengontrol pertanian hidroponik menjadi lebih efisien dan hasil panen yang lebih optimal.

## 1. Pendahuluan

Stroberi adalah salah satu contoh komoditas buah subtropis yang memiliki potensi untuk dikembangkan di Indonesia [1]. Menurut laporan dari Badan Pusat Statistik (BPS), produksi stroberi di Indonesia menunjukkan adanya penurunan [1]. Ada beberapa faktor yang menyebabkan penurunan produksi stroberi di Indonesia, salah satunya adalah kondisi lingkungan yang tidak cocok untuk budidaya stroberi dan berkurangnya kapasitas lahan untuk pertanian [2].

Seiring berkembangnya teknologi, pertanian modern juga berkembang di kota yang biasa disebut dengan urban farming. Urban farming menawarkan solusi untuk memanfaatkan lahan yang tidak terpakai (seperti teras, dinding, dan atap rumah) untuk bercocok tanam. Salah satu teknik yang populer digunakan dalam urban farming adalah hidroponik, yang menjadi solusi inovatif bagi kota-kota yang mengalami kekurangan lahan pertanian [3].

Pertanian hidroponik yang ada di masyarakat umumnya masih menggunakan cara manual. Dimana cara tersebut relatif menguras waktu mulai dari pengkajian sampai pemberian nutrisi yang ada di tandon nutrisi [4]. Cara manual tanpa menggunakan bantuan sensor cenderung sulit untuk mengetahui kadar nutrisi yang tercampur pada larutan secara tepat. Sementara itu cara manual yang hanya menggunakan sensor saja tanpa bantuan sistem, tidak praktis dalam hal analisis data harian dari pembacaan sensor yang diperoleh. Kemudian cara manual juga mengharuskan petani hidroponik datang langsung ke kebun untuk memantau dan mengontrol pertanian hidroponik mereka. Untuk itu peran penggunaan smart farming perlu diimplementasikan guna membantu mengatasi permasalahan tersebut.

Budidaya tanaman hidroponik dengan teknologi smart farming, umumnya menggunakan perangkat berbasis IoT. Perangkat ini dapat mengatur perlengkapan hidroponik secara otomatis termasuk pencampuran pupuk (nutrisi) dan air, sehingga usaha budidaya hidroponik dapat lebih efisien dalam penggunaan tenaga kerja dan biaya produksi [5].

Dengan karakteristik *urban agriculture* dimana lahan yang tersedia terbatas, ditengah kesibukan warga perkotaan, maka diperlukan informasi yang akurat dan terupdate agar hasil produksi tanaman dapat tetap maksimal. Keberadaan aplikasi IoT akan membantu memonitor serta mengelola informasi dari pertanian hidroponik secara real-time.

## 2. Pendekatan Pemecahan Masalah

### 2.1. Smart Farming

*Smart farming* adalah konsep pertanian yang menggabungkan teknologi informasi dan komunikasi (TIK) dengan praktik pertanian tradisional untuk meningkatkan produktivitas, efisiensi, dan keberlanjutan pertanian. Tujuannya adalah untuk meningkatkan hasil panen, mengurangi biaya produksi, meningkatkan kualitas produk, dan mengurangi dampak negatif pada lingkungan. Contoh teknologi yang digunakan dalam smart farming adalah sensor tanah, *drone*, robot pertanian, *big data analytics*, *Internet of Things* (IoT), dan sistem informasi geografis (GIS).

### 2.2. Internet of Things

*Internet of Things* (IoT) adalah sebuah konsep yang mengacu pada jaringan perangkat yang saling terhubung dan berkomunikasi dengan satu sama lain melalui internet. Perangkat-perangkat tersebut dapat berupa sensor, mesin, barang elektronik, ataupun peralatan lainnya yang dilengkapi dengan kemampuan untuk mengumpulkan dan mengirimkan data secara otomatis. Teknologi IoT dalam dunia pertanian dapat dimanfaatkan oleh petani untuk membantu mempermudah pekerjaan mereka.

Beberapa contoh penerapan IoT di sektor pertanian adalah sebagai berikut.

- Monitoring lingkungan: Sensor IoT dapat digunakan untuk memonitor suhu, kelembaban udara, dan kadar air di tanah. Data ini dapat digunakan untuk mengoptimalkan penggunaan air dan pupuk, serta untuk memprediksi cuaca yang akan datang.

- Monitoring tanaman: Sensor IoT dapat dipasang pada tanaman guna memonitor pertumbuhan dan kesehatan tanaman. Data ini dapat membantu petani untuk mengambil keputusan tentang waktu yang tepat untuk melakukan penyiraman, pemupukan, dan perlindungan tanaman.
- Pengendalian peralatan: Teknologi IoT dapat digunakan untuk mengontrol peralatan pertanian seperti irigasi, alat penyemprot, dan traktor. Hal ini dapat membantu petani untuk menghemat waktu dan energi dalam pengoperasian peralatan.

Dengan memanfaatkan teknologi IoT petani dapat mengambil keputusan yang lebih baik berdasarkan data yang akurat dan *real-time* dari sensor, sehingga dapat meningkatkan efisiensi dan produktivitas pertanian. Selain itu, IoT juga dapat membantu mengurangi biaya produksi dan meminimalkan dampak negatif pada lingkungan.

### 2.3. Android

Android adalah sistem operasi *mobile* yang dirancang untuk perangkat bergerak dan layar sentuh seperti *smartphone*, *tablet*, *smartwatch*, *smart TV*, dan lain sebagainya. Android pertama kali ditemukan pada tahun 2003 oleh Andy Rubin, Rich Miner, Chris White dan Nick Sears yang dikembangkan menggunakan *kernel* berbasis linux. Pada mulanya, tujuan utama pengembangan Android adalah untuk menciptakan sistem operasi canggih yang dapat digunakan pada kamera digital. Namun, menyadari bahwa peluang pasar untuk perangkat tersebut terbatas, maka pengembangan Android di-alihkan ke pasar *smartphone* untuk bersaing dengan sistem operasi *mobile* yang terlebih dahulu masuk ke pasaran.

Pemrograman Android adalah proses membuat aplikasi *mobile* untuk perangkat yang menjalankan sistem operasi Android. Pemrograman Android dapat dilakukan dengan menggunakan bahasa pemrograman Java, Kotlin, atau Dart (Flutter).

Berkembangnya teknologi di dunia pemrograman mengakibatkan proses pengembangan aplikasi android dapat dilakukan tidak harus dengan cara konvensional saja, melainkan ada teknologi pemrograman baru bernama *block code*. Pemrograman blok kode (*block code*) atau visual programming adalah jenis pemrograman yang memungkinkan pengguna untuk membangun aplikasi dengan cara yang lebih visual, dengan menarik dan menjatuhkan blok kode yang telah disediakan di dalam *platform* pembangun aplikasi. Platform ini biasanya terdiri dari sekumpulan blok kode yang dapat digabungkan dan disusun dalam bentuk alur logika yang mudah dipahami dan diubah oleh pengguna.

Perbedaan utama antara pemrograman android konvensional dan visual adalah dari metode pembuatan yang digunakan. Pada metode konvensional diperlukan pemahaman penulisan kode mengenai suatu bahasa pemrograman tertentu. Sedangkan metode visual tidak demikian, melainkan menggunakan blok kode visual dalam pembuatan aplikasi. Meskipun pemrograman blok kode lebih mudah dipelajari dan digunakan oleh pemula, namun pemrograman konvensional memberikan lebih banyak fleksibilitas dan kontrol atas aplikasi yang dibuat.

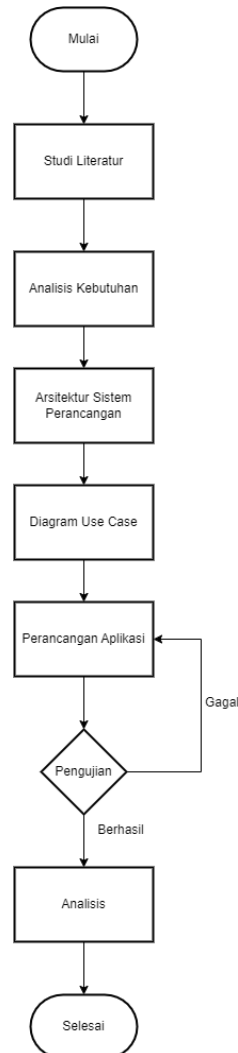
### 2.4. Dashboard Admin

Dashboard Admin adalah antarmuka yang dirancang khusus untuk administrator atau pengelola sistem, yang memungkinkan mereka untuk mengakses dan mengelola aplikasi atau sistem secara efisien. Dalam konteks aplikasi IoT, Dashboard Admin digunakan untuk memantau dan mengelola perangkat IoT, mengatur sensor dan actuator, memantau kinerja jaringan, dan mengelola data sensor.

## 3. Rancangan Penelitian

Metode penelitian yang digunakan adalah metode penelitian dan pengembangan atau dalam Bahasa Inggris disebut *Research and Development* (R&D). Metode ini dipakai dengan tujuan untuk mengembangkan produk baru atau menyempurnakan produk yang sudah ada. Istilah produk disini

dapat diartikan sebagai hasil akhir dari penelitian yang berupa perangkat keras maupun perangkat lunak. Produk yang dirancang dalam penelitian ini berupa aplikasi monitoring dan controlling nutrisi tanaman hidroponik berbasis IoT untuk perangkat bersistem android, serta membuat dashboard untuk manajemen dari aplikasi itu sendiri. Adapun rancangan dari penelitian ini dilihat pada Gambar 1 sebagai berikut.



**Fig. 1.** Rancangan Penelitian

Studi literatur adalah proses mengkaji ulang penelitian terdahulu mengenai topik yang berkaitan dengan topik penelitian yang akan dilakukan. Studi literatur melibatkan pengumpulan data, penilaian, dan sintesis informasi dari berbagai sumber seperti artikel jurnal, buku, artikel, laporan penelitian, dan sumber lain yang relevan. Studi literatur dilakukan sebagai langkah awal dalam merumuskan permasalahan dan mengembangkan penelitian.

### 3.1. Analisis Kebutuhan

Sebelum memasuki proses pembuatan dari produk diperlukan suatu komponen atau peralatan yang digunakan untuk membangun produk secara keseluruhan. Untuk mempermudah dalam klasifikasi kebutuhan, maka analisis kebutuhan dikelompokkan menjadi lima bagian antara lain kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan alat, kebutuhan layanan, serta kebutuhan parameter tanaman. Kebutuhan perangkat keras yang digunakan penelitian ini adalah sebagai berikut:

- Raspberry 3 Model B: difungsikan sebagai server yang akan melayani komunikasi data antar sensor dan pemrosesan data sensor.
- Laptop: laptop berperan penting dalam pembuatan aplikasi, dashboard, maupun untuk me-remote server.

Kebutuhan software yang digunakan penelitian ini adalah sebagai berikut:

- Canva: digunakan sebagai media untuk menangani keperluan desain grafis yang mendukung pembuatan aplikasi.
- Kodular: platform utama untuk membangun aplikasi secara visual programming.
- Kodular Companion: aplikasi mobile bagian dari kodular yang digunakan untuk melakukan debugging atau ujicoba terkait aplikasi yang sedang dikembangkan pada platform kodular.
- Visual Studio Code: merupakan teks editor yang digunakan untuk menulis kode program *dashboard*.
- XAMPP: digunakan sebagai perangkat lunak yang menyediakan paket aplikasi server seperti Apache (*web server*), MySQL (*database server*), dan PHP (*programming language*) yang digunakan untuk menjalankan *dashboard* aplikasi.
- Composer: berperan untuk memasang *library* PHP yang digunakan dalam pembuatan *dashboard*.

Kebutuhan layanan yang digunakan penelitian ini adalah sebagai berikut:

- Firebase: dipakai dalam penelitian ini sebagai media transmisi data sensor yang telah diolah oleh raspi ke *cloud*.
- Web Hosting: digunakan untuk menerapkan program *dashboard* yang telah dibuat ke dalam server yang sebenarnya.

Kebutuhan data parameter tanaman yang digunakan penelitian ini dapat dilihat pada Tabel 1 berikut ini.

**Table 1.** Parameter kondisi ideal untuk tanaman strawberry

Parameter	Bagus	Normal	Bahaya
Suhu	17 - 20°C	21 - 28°C	<17 dan >28°C
Kelembaban Ruangan	85 - 90%	70 - 84%	<70 dan >90%
EC	2.2 - 2.4 mS/cm	1.8 - 2.1 mS/cm	<1.8 dan >2.4 mS/cm
pH	5.5 - 6.4	6.5 - 7.2	<5.5 dan >7.2
Kadar CO2	700 - 800 ppm	600 - 669 ppm	<600 dan >700 ppm
Kelembapan Tanah	70 - 90%	40 - 69%	<40 dan >90%
Suhu Larutan	20 - 27°C	15 - 19°C	<15 dan >27°C
Water Level	71 - 95%	20 - 70 %	<20 dan >95%

Kebutuhan parameter tanaman digunakan untuk memastikan kondisi optimal bagi pertumbuhan dan kesehatan tanaman. Dengan memahami dan memenuhi kebutuhan tersebut, maka dapat menciptakan lingkungan yang mendukung tanaman untuk tumbuh dengan baik. Pada bagian ini kami bekerja sama dengan tim *field assistant* untuk mengetahui kondisi parameter yang cocok untuk tanaman *strawberry*. Data parameter ini nantinya akan digunakan untuk membuat *set point* pada aplikasi. *Set point* ini berperan sebagai klasifikasi dari data sensor. Dimana ada tiga buah kondisi setpoint yang ditentukan, antara lain kondisi bagus (*good*), normal, dan bahaya (*danger*) untuk tanaman.

### 3.2. Arsitektur Sistem Perancangan

Pembangunan produk secara keseluruhan terdiri dari tiga bagian pemfokusan. Tiga bagian itu antara lain bagian *hardware* bertugas merancang, merakit, dan memprogram sensor agar data pem-

bacaannya dapat dikirimkan ke bagian *backend*. Bagian *backend* bertugas memprogram server untuk melayani komunikasi transmisi data antar perangkat, memproses, dan mengirimkan data hasil olahannya ke bagian aplikasi. Bagian aplikasi bertugas memproses data hasil olahan *backend* ke dalam bentuk data yang dapat dimengerti oleh pengguna melalui aplikasi. Arsitektur sistem perancangan dari produk secara keseluruhan yang dibuat dapat dilihat pada Gambar 2 dibawah ini.

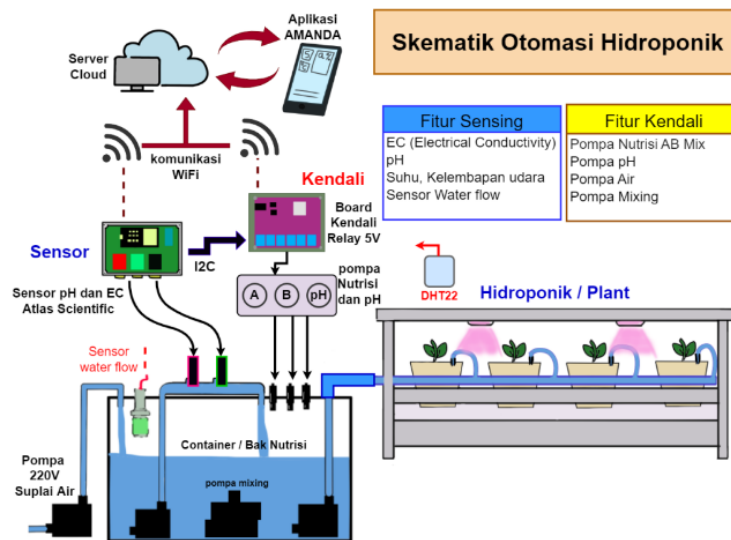


Fig. 2. Arsitektur sistem perancangan produk secara keseluruhan

Dalam arsitektur sistem perancangan produk terdapat sistem transmisi data yang menghubungkan antar bagian untuk dapat saling berkomunikasi. Transmisi data memerlukan suatu protokol komunikasi dalam prosesnya. Dimana dalam sistem IoT, protokol komunikasi data sangat penting untuk memastikan bahwa perangkat dapat berkomunikasi dan bertukar informasi dengan baik. Untuk dapat lebih mudah memahami mengenai arsitektur sistem perancangan dari penelitian ini, akan ditampilkan arsitektur transmisi data pada produk yang ditunjukkan pada Gambar 3.

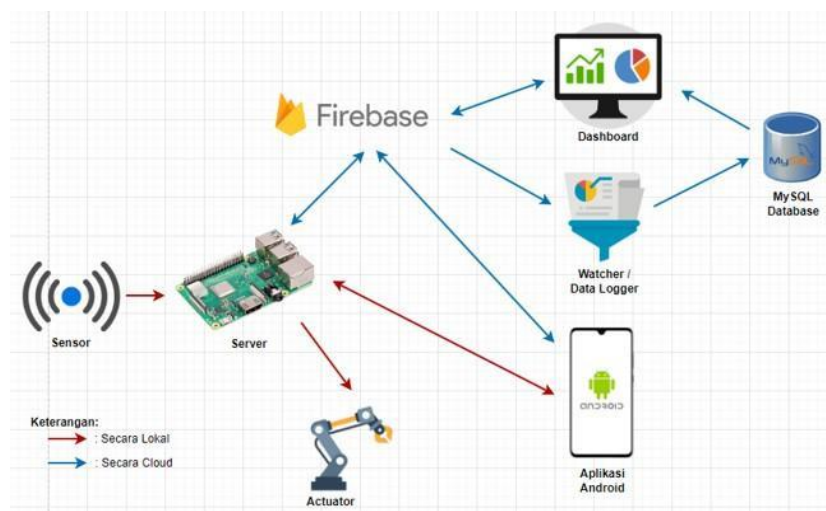


Fig. 3. Arsitektur sistem transmisi data pada produk

Aplikasi bekerja dengan mengambil data dari sensor dan mengendalikan aktuator menggunakan dua metode transmisi data yang berbeda, yakni melalui *cloud* dan lokal. Dimana



transmisi data local menggunakan protokol komunikasi MQTT yang disediakan oleh server, sedangkan cloud menggunakan Realtime Database dari Firebase. Pertama-tama sensor akan membaca data kondisi lingkungan pada pertanian hidroponik. Data yang sudah diperoleh dikirimkan ke server untuk di proses. Hasil olahan data server kemudian dikirimkan ke *cloud* atau dapat langsung diterima oleh aplikasi secara lokal. Data yang sudah masuk di *cloud* akan diambil oleh aplikasi, dashboard, maupun program watcher. Program *watcher* sendiri merupakan *data logger* guna mencatat setiap data yang ditransmisikan via *cloud* untuk disimpan pada database mysql. Data yang sudah dikumpulkan akan diambil kembali oleh dashboard guna keperluan analisis data harian dari sensor.

### 3.3. Diagram Penggunaan (Use Case)

Diagram *use case* menggambarkan berbagai skenario penggunaan atau kasus penggunaan yang terjadi antara aktor dan sistem. *Use case* adalah representasi dari sebuah interaksi antara aktor dan sistem yang menggambarkan aktivitas-aktivitas yang dilakukan oleh aktor tersebut. Dalam penelitian ini diagram *use case* dibagi menjadi dua jenis yaitu untuk aplikasi dan *dashboard*.

#### Diagram Penggunaan Aplikasi

Pengguna menggunakan aplikasi mobile berbasis android untuk dapat mengakses fitur monitoring dan *controlling*. Sebelum dapat mengakses fitur tersebut pengguna harus *login* terlebih dahulu untuk dapat masuk ke sistem atau *register* akun bila ingin membuat akun. Setelah proses *login* berhasil, langkah selanjutnya adalah pemilihan koneksi untuk transmisi data aplikasi. Kemudian setelahnya baru dapat menggunakan fitur monitoring dan *controlling* pada aplikasi. Fitur monitoring pada aplikasi terdiri dari pemantauan pH larutan, EC larutan, suhu larutan, suhu udara, kelembapan udara, *water level*, serta *soil moisture*. Sementara itu untuk fitur *controlling* terdiri dari pengendalian pompa Ph, pompa air, *pompa sample*, pompa tanaman (penyiraman), pompa nutrisi, serta pompa *mixing*. Diagram *use case* pengguna akan ditunjukkan pada Gambar 4 berikut ini.

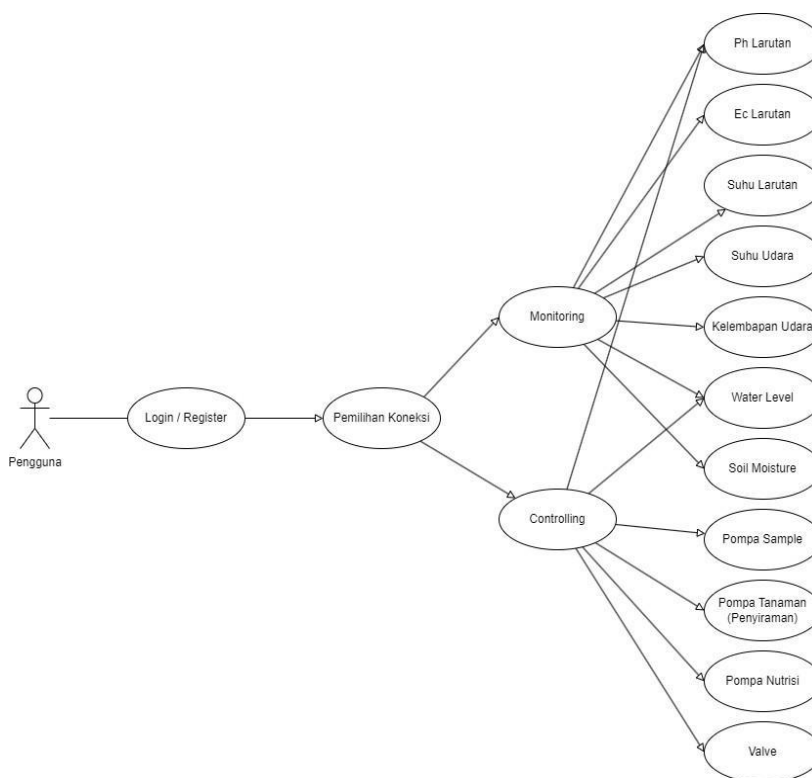


Fig. 4. Diagram *use case* pengguna

### Diagram Penggunaan Admin

Admin menggunakan *dashboard* aplikasi yang berupa web untuk manajemen data pada aplikasi. Untuk dapat menggunakan *dashboard*, admin harus *login* terlebih dahulu. Setelah berhasil masuk, admin akan diarahkan ke menu *dashboard* utama. Selanjutnya admin dapat manajemen aplikasi melalui fitur-fitur yang ada pada dashboard seperti grafik *realtime* sensor, *realtime actuator control*, manajemen *setpoint*, manajemen pengguna, serta analisis data historis sensor. Diagram *use case* admin akan ditunjukkan pada Gambar 5 berikut ini.

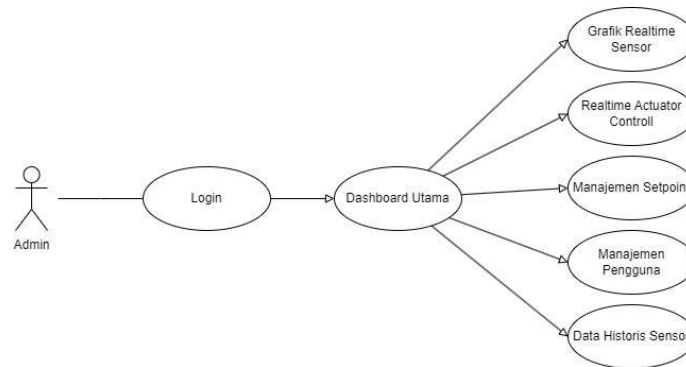


Fig. 5. Diagram *use case* admin

### 3.4. Perancangan Sistem

#### Perancangan Aplikasi

Sebelum masuk ke tahap pembuatan aplikasi, terlebih dahulu dilakukan perancangan terkait sistem aplikasi yang akan dibuat. Perancangan sendiri merupakan suatu proses mencari ide sehingga menghasilkan gambaran, serta merencanakan keseluruhan komponen untuk menghasilkan produk yang akan dibuat. Daftar kegiatan perancangan yang dilakukan untuk menghasilkan aplikasi antara lain mencari desain *mockup* untuk aplikasi yang akan dibuat, membuat rancangan antarmuka aplikasi, membuat flowchart untuk program aplikasi, mendesain gambar grafis untuk keperluan aplikasi, serta mencari aset pendukung untuk aplikasi. Adapun rancangan antarmuka dari aplikasi akan ditunjukkan pada Gambar 6 berikut ini.

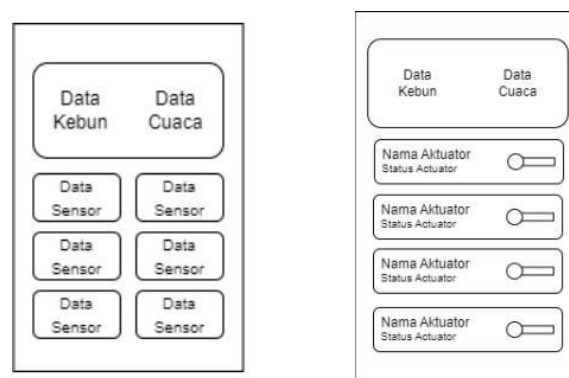


Fig. 6. Rancangan antarmuka aplikasi

#### Perancangan Dashboard

Perancangan *dashboard* dilakukan setelah selesai melakukan perancangan aplikasi. Dimana ini bertujuan untuk merencanakan pembuatan *dashboard* yang akan digunakan oleh aplikasi. Pembuatan program *watcher* juga direncanakan pada tahap ini karena berkaitan dengan *dashboard*. Adapun rancangan antarmuka dari *dashboard* akan ditunjukkan pada Gambar 7 dibawah ini.



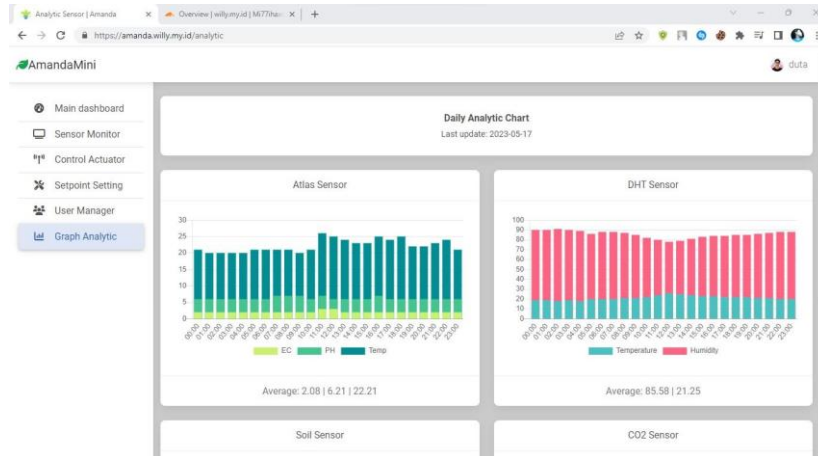


Fig. 7. Rancangan antarmuka *dashboard*

## 4. Hasil dan Pembahasan

### 4.1. Implementasi Pengembangan Produk Secara Keseluruhan

Produk yang dibuat memiliki cara kerja dimana sensor membaca nutrisi pada bak nutrisi melalui proses *sampling*. Proses *sampling* adalah proses pengambilan data sensor secara berkala dalam periode waktu tertentu. Proses *sampling* ini diperlukan karena sensor yang digunakan tidak tercelup ke bak nutrisi secara langsung. Dimana ketika proses *sampling* berlangsung, air nutrisi yang ada pada bak nutrisi akan dialirkan ke sensor menggunakan pompa dan kembali lagi ke bak nutrisi.

Perangkat sensor dan aktuator keseluruhannya dikendalikan oleh mikrokontroler yang terhubung ke sebuah jaringan WiFi. Selain mikrokontroler, terdapat juga server yang terhubung dengan jaringan yang sama. Pada jaringan WiFi tersebut mikrokontroler terhubung ke server untuk proses komunikasi data. Proses komunikasi data antara mikrokontroler dengan server menggunakan layanan broker MQTT yang disediakan oleh server. Server bertugas melayani seluruh transmisi data antar perangkat dan memproses data tersebut.



Fig. 8. Implementasi produk secara keseluruhan

Data yang dikirimkan oleh mikrokontroler adalah data pembacaan sensor dan data kondisi

aktuator. Dimana data-data tersebut akan diolah oleh server menjadi tipe data JSON. Data yang sudah diolah akan dikirimkan menuju basis data firebase atau dapat langsung diterima oleh aplikasi dengan menggunakan protokol komunikasi dan jaringan yang sama dengan server.

Penerimaan data secara langsung (*local*) oleh aplikasi bertujuan sebagai *backup* apabila koneksi internet terkendala. Penerimaan data dengan metode ini dikirimkan melalui protokol MQTT dan harus terhubung ke jaringan yang sama dengan server. Cakupan penerimaan data dengan menggunakan metode ini terbatas, dikarenakan bergantung pada jangkauan jaringan WiFi yang digunakan. Sementara itu untuk yang menggunakan *firebase* cakupannya tidak terbatas, karena data dikirimkan ke *cloud* (internet).

Basis data firebase menerima data dalam bentuk JSON, oleh karena itu data dari sensor perlu diolah dahulu pada server. Data yang sudah masuk pada firebase akan diambil oleh aplikasi dan juga dashboard guna keperluan penampilan data.

Aplikasi akan mengambil data sensor dan membandingkan data tersebut dengan data *setpoint*. Data *setpoint* berperan sebagai batas yang mengklasifikasikan data sensor ke dalam suatu kondisi yang telah ditentukan. Dimana kondisi-kondisi tersebut antara lain *normal*, *good*, dan *danger*. Kondisi normal berarti merupakan kondisi umum yang diperlukan oleh tanaman untuk tumbuh, kondisi *good* merupakan kondisi ideal untuk tanaman, sedangkan kondisi *danger* adalah kondisi tidak cocok untuk tanaman. Ketika dalam kondisi *danger*, aplikasi akan bereaksi dengan menggetarkan perangkat dan membunyikan alarm peringatan kepada pengguna.



**Fig. 9.** Tampilan aplikasi yang dikembangkan

Penambahan nutrisi, air, pH, pencampuran, dan pengairan nutrisi untuk tanaman dilakukan melalui aplikasi dengan mengontrol aktuator. Dalam pengontrolan aktuator menggunakan aplikasi, aplikasi akan mengirimkan perintah untuk menyalakan atau mematikan aktuator yang dituju menggunakan metode transmisi data yang dipilih. Apabila menggunakan opsi *cloud* maka aplikasi akan mengirimkan data ke *cloud* terlebih dahulu, server mengambil data tersebut, kemudian diteruskan ke mikrokontroler menggunakan protokol MQTT. Namun apabila memilih opsi *local*, aplikasi akan mengirimkan data melalui protokol MQTT kepada mikrokontroler menggunakan layanan broker MQTT yang disediakan oleh server. Mikrokontroler akan mematikan atau menyalakan aktuator berdasarkan perintah yang dikirimkan oleh aplikasi.

Sebelum nutrisi dialirkan ke tanaman, baik itu nutrisi, larutan pH, dan air perlu dipompa

terlebih dahulu dari wadahnya untuk dipindahkan ke bak penampung nutrisi. Setelah masuk ke bak penampung nutrisi, dilakukan pengadukan larutan menggunakan pompa *mixing*. Pada tahap ini proses *sampling* juga dilakukan guna mendapatkan takaran nutrisi yang tepat untuk tanaman. Setelah mendapat nutrisi yang tepat, larutan siap untuk dialirkan ke tanaman menggunakan pompa tanaman.



Fig. 10. Implementasi penggunaan aplikasi terhadap produk

Pada implementasi pengujian produk dalam kondisi yang nyata, dilakukan dengan menjalankan alat selama 3 jam dengan pengambilan data setiap 10 menit selama dua jam. Pengujian ini melibatkan penggunaan 2 sensor, yaitu sensor yang membaca larutan nutrisi (Atlas) dan sensor untuk mengukur suhu dan kelembaban ruangan (DHT22)

Skenario pengujian yang dilakukan terdiri dari pengujian monitoring dan *controlling*. Untuk monitoring, dilakukan pemantauan melalui aplikasi terhadap data sensor yang dikirimkan oleh mikrokon- troller ke aplikasi, apakah terdapat nilai error, lalu dihitung data error dan persentase errornya, serta selang waktu selama proses pengiriman data. Untuk *controlling*, dilakukan uji coba untuk mengendalikan actuator melalui aplikasi dengan dua kondisi yakni on atau off, kemudian melihat respons dari target actuator yang dikendalikan, apakah actuator tersebut berjalan sesuai perintah atau tidak, jika sesuai maka dihitung juga selang waktu antara proses pemberian perintah dengan respons aksinya.

#### 4.2. Analisis Data

Berdasarkan data rata-rata pengujian dari implementasi produk yang telah dilakukan, dapat dibuat grafik perbandingan dari kedua metode transmisi. Grafik dari perbandingan persentase error dan delay dari implementasi *monitoring* akan ditunjukkan pada Gambar 11 berikut ini.

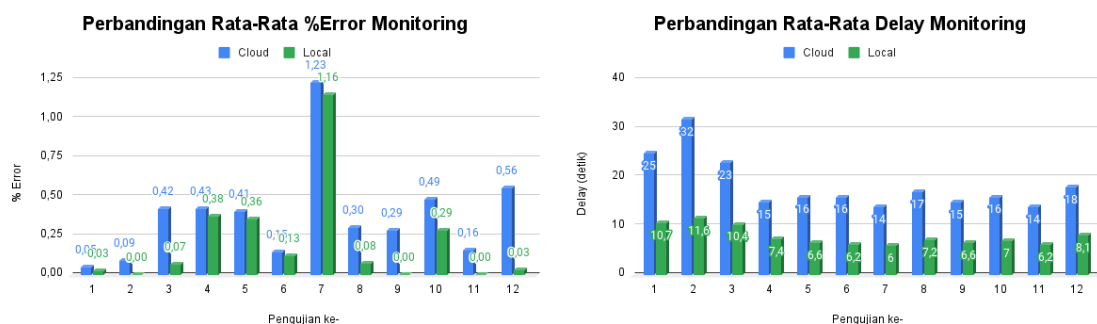
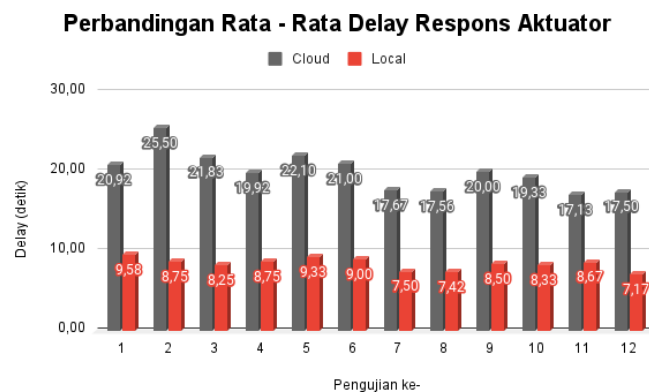


Fig. 11. Grafik perbandingan persentase *error* dari monitoring

Grafik perbandingan *delay* dari implementasi *controlling* akan ditunjukkan pada Gambar 12 berikut ini.



**Fig. 12.** Grafik perbandingan delay dari *controlling*

Berdasarkan grafik perbandingan rata-rata pengujian dari kedua metode transmisi data yang digunakan pada penelitian ini, dapat dianalisis bahwa protokol komunikasi MQTT lebih unggul dari pada menggunakan firebase. Hal ini ditunjukkan dengan *delay* transmisi MQTT lebih cepat serta persentase *error* yang lebih kecil, baik itu ketika proses *monitoring* maupun *controlling*.

Karena berbasis lokal maka protokol MQTT yang digunakan dalam penelitian ini tidak bergantung pada internet, melainkan bergantung pada server yang menjadi broker untuk proses bertukar data antar perangkat. Spesifikasi atau kemampuan server berpengaruh terhadap performa layanan yang disediakan. Banyaknya data yang dikirim secara terus menerus dalam waktu bersamaan, menyebabkan terdapatnya antrian data yang diproses oleh server. Sehingga hal itulah yang menyebabkan *delay* pada proses transmisi data. Selain itu penyebab lain juga dapat berasal dari waktu komputasi program pada mikrokontroler dalam hal pemrosesan pengiriman maupun penerimaan data.

Sementara itu transmisi data yang menggunakan firebase bergantung pada koneksi internet yang digunakan. Jika koneksi internet bermasalah, maka data tidak dapat tersinkron ke firebase. Dalam implementasi alat, *delay* transmisi data firebase lebih lama dari pada protokol MQTT, hal itu dikarenakan setiap proses transmisi data yang dibuat pada proyek akhir ini melalui protokol komunikasi MQTT terlebih dahulu. Meskipun demikian, jika dihitung selisih proses transmisi data diantara keduanya, masih terdapat selisih waktu delay yang menunjukkan MQTT lebih cepat dari firebase.

Disisi lain ketika kedua protokol diuji dalam tahap pengembangan dimana pengujian dilakukan tanpa melibatkan proses pengolahan data dari server, tanpa mikrokontroler, serta keduanya menggunakan protokol berbasis *cloud*, kedua protokol ini menunjukkan kecepatan transmisi data yang sama cepat tanpa adanya nilai *error*. Namun, ketika dilakukan pengujian pengiriman data secara bersamaan dan kontinu, protokol komunikasi MQTT lebih unggul karena hampir tidak ada *delay* jika dibandingkan dengan firebase yang mengalami *delay* yang singkat.

Meskipun keduanya memiliki keunggulan yang hampir sama, dalam konteks Internet of Things (IoT), protokol MQTT lebih cocok digunakan untuk pengiriman data yang cepat dan kontinu. Jika data yang dikirimkan tidak terlalu besar dan kontinu, firebase dapat menjadi opsi pengganti dalam mentransmisikan data antar perangkat IoT.

## 5. Kesimpulan

Dari implementasi alat yang telah dilakukan, baik aplikasi maupun dashboard, semuanya berjalan sesuai fungsionalitas yang diharapkan. Produk yang dibuat berhasil digunakan untuk memonitor dan mengontrol pertanian hidroponik stroberi agar tetap berada dalam kondisi ideal. Dengan kondisi nutrisi yang terjaga dengan baik, pertumbuhan tanaman dapat lebih cepat dan

terkontrol. Aplikasi ini memungkinkan para petani untuk memantau dan *mengontrol* pertanian hidroponik mereka melalui *smartphone*, serta memberikan notifikasi alarm jika kondisi nutrisi tidak sesuai dengan *setpoint* yang telah ditetapkan.

Selain itu, para petani juga dapat melihat analisis data rata-rata harian dari sensor pertanian hidroponik mereka melalui *dashboard*. Keberadaan alat ini dapat membantu petani dalam mengetahui kondisi nutrisi serta mengontrol pertanian hidroponik mereka dari jarak dekat maupun jarak jauh hanya dengan menggunakan *smartphone*.

## References

- [1] Sadya S., "Produksi Stroberi Indonesia Sebanyak 9.860 Ton pada 2021," Data Indonesia, Sep. 21, 2022. [Online]. Available: <https://dataindonesia.id/agribisnis-kehutanan/detail/produksi-stroberi-indonesia-sebanyak-9860-ton-pada-2021>. [Accessed: 06 Feb., 2023].
- [2] Herlindawati, A., Trimio, L., & Noor, T. I., "Analisis tekanan penduduk terhadap petani padi sawah" *Jurnal Pemikiran Masyarakat Ilmiah Berwawasan Agribisnis.*, vol. 4, pp. 12-24, Jan 2018.
- [3] Crisnapati, P. N., Wardana, I. N. K., Aryanto, I. K. A. A., Hermawan, A., "Hommons: Hydroponic management and monitoring system for an IOT based NFT farm using web technology" in *Proc. of the 2017 5th Int. Conf. on Cyber and IT Service Management (CITSM), International Conference, 08-10 Aug 2017, Bali, Indonesia* [Online]. Available: IEEE Xplore, <http://www.ieee.org>. [Accessed: 2 Feb. 2023].
- [4] Putra, E. R. R. A., "Sistem kontrol dan monitoring pemberian pupuk nutrisi hidroponik model nutrient film technique (nft) pada tanaman pakcoy berbasis internet of things (IoT)", Reading, Surabaya: ITATS, 2019. [Online] Available: PERPUSTAKAAN ITATS.
- [5] Mulyati, E., Hamidin, D., Fauzan, M. N., "Kelayakan Teknologi IoT Untuk Kebun Hidroponik Hortikultura Menggunakan Hydropo 4.0 Di Perkebunan Alam Pasundan, Jawa Barat" *J@ti Undip: Jurnal Teknik Industri.*, vol. 16, pp. 109-115, May 2021.