

# Perancangan Aplikasi Pemantauan Rumah Kaca Pintar Berbasis Android

## Design of Android-based Smart Greenhouse Monitoring Applications

Muhammad Afwan<sup>\*1)</sup>, Sumardi<sup>2)</sup>, Risma Septiana<sup>3)</sup>

<sup>1, 3)</sup>Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro  
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

<sup>2)</sup>Program Studi Teknik Elektro, Fakultas Teknik, Universitas Diponegoro  
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

**How to cite:** M. Afwan, Sumardi, and R. Septiana. "Perancangan Aplikasi Pemantauan Rumah Kaca Pintar Berbasis Android," *Jurnal Teknik Komputer*, vol. 1, no. 1, pp. 21-29, June 2022. doi: <https://doi.org/10.14710/jtk.v1i1.34573> [Online].

**Abstract** - Internet of Things makes it possible to connect with various things, for example we can use smartphones to get information about products in super markets (E-Food Analytics) or monitor heart rate (E-Health), manage transportation (Autonomous smartcars), smarthome, smartfarming and others. Internet of Things technology has been widely applied to monitoring and controlling the agricultural (greenhouse) environment in recent years. In this study, a prototype smart greenhouse has been built with a wireless sensor network based on the Internet of Things that can detect temperature and humidity in real time stored in the adafruit cloud. The system infrastructure that has been created consists of several temperature, humidity and actuator detectors. The purpose of this research is to design an Android-based smart greenhouse monitoring application that displays sensor data optimally and is easy to use. Smart greenhouse monitoring application using RAD development. The results of this study is the application can functionally work well displaying data from the sensor. Then usability testing using the System Usability Scale (SUS) method got a score of 84.5 in the excellent category with a B grade scale or usability acceptable or feasible. Usability testing with another method, namely Single Ease Question (SEQ) this application gets a percentage of assessment with a score of 94% which means this application is very easy to use.

**Keywords** – Internet of Things, Smart Greenhouse, RAD, Adafruit Cloud, Android.

**Abstrak** – Internet of Things memungkinkan untuk terhubung dengan berbagai hal, misalnya kita dapat menggunakan smartphone untuk mendapatkan informasi tentang product pada super market (E-Food Analytics) atau memantau detak jantung (E-Health), mengelola transportasi (Autonomous smart cars), smart home, smart farming dan lain-lain. Teknologi Internet of Things telah banyak diterapkan pada pemantauan dan pengendalian lingkungan pertanian

(rumah kaca) dalam beberapa tahun terakhir. Dalam penelitian ini telah terbangun sebuah prototype smart greenhouse dengan jaringan sensor nirkabel berbasis Internet of Things yang dapat mendeteksi suhu dan kelembaban udara secara real time yang tersimpan pada adafruit cloud. Infrastruktur sistem yang telah dibuat terdiri atas beberapa alat pendeteksi suhu, kelembaban udara dan actuator. Tujuan dari penelitian adalah merancang aplikasi monitoring smart greenhouse berbasis android yang menampilkan data sensor secara optimal dan mudah digunakan. Aplikasi monitoring smart greenhouse menggunakan pengembangan RAD. Hasil dari penelitian ini secara fungsional aplikasi dapat bekerja dengan baik menampilkan data dari sensor. Kemudian pengujian secara usability menggunakan metode System Usability Scale (SUS) mendapatkan skor 84,5 pada kategori excellent dengan grade B atau secara usability dapat diterima atau layak. Pengujian usability dengan metode lain yaitu Single Ease Question (SEQ) aplikasi ini mendapatkan persentase penilaian dengan skor 94% yang artinya aplikasi ini sangat mudah untuk digunakan.

**Kata kunci** – Internet of Things, Smart Greenhouse, RAD, Adafruit Cloud, Android.

### I. PENDAHULUAN

Internet of Things memungkinkan untuk terhubung dengan berbagai hal, misalnya kita dapat menggunakan smartphone untuk mendapatkan informasi tentang product pada super market (E-Food Analytics) atau memantau detak jantung (E-Health), mengelola transportasi (Autonomous smart cars), smart home, smart farming dan lain-lain[1]. Teknologi Internet of Things telah banyak diterapkan pada pemantauan dan pengendalian lingkungan pertanian (rumah kaca) dalam beberapa tahun terakhir[2].

Salah satu penerapan Internet of Things adalah smart agriculture yaitu dengan menggunakan modul sensor kelembaban tanah yang mengirimkan data pembacaan sensor melalui internet dengan

<sup>\*</sup>)Penulis Korespondensi (Muhammad Afwan)  
Email: mafwan@student.ce.undip.ac.id



menggunakan modul *Wi-Fi* ESP-8266 kemudian melakukan pengendalian pompa irigasi (*motor driver-289D*) melalui *Arduino Uno R3*. Data yang didapat dari sensor dan pengendalian pompa di *request* oleh komputer melalui halaman *web* aplikasi *thinkspeak* yang selanjutnya menampilkan nilai sensor secara *real time* dan status pompa dalam bentuk grafik [1].

Dalam penelitian ini telah terbangun sebuah *prototype smart greenhouse* dengan jaringan sensor nirkabel berbasis *Internet of Things* yang dapat mendeteksi suhu dan kelembaban udara secara *real time*. Infrastruktur sistem yang telah dibuat terdiri atas beberapa alat pendeteksi suhu, kelembaban udara dan aktuator. Apabila kondisi di dalam *greenhouse* tidak sesuai dengan yang ditetapkan maka aktuator akan meyala. Data dari sensor dikirimkan melalui *internet* dan tersimpan pada *Adafruit Cloud*. *Smart greenhouse* menggunakan modul sensor DHT-22 untuk mengukur kondisi di dalam rumah kaca.

Berdasarkan permasalahan di atas untuk memudahkan dan efiseinsi waktu dalam pemantauan dari alat tersebut maka akan dibangun sebuah aplikasi *monitoring* berbasis *android* dengan *interface* yang interaktif yang dapat menampilkan data dari dua parameter tersebut sehingga dapat dilakukan pemantuan jarak jauh dengan memanfaatkan jaringan *internet*. Perancangan dari aplikasi *monitoring* ini berjalan pada *system* operasi *android* dan menggunakan *android studio* dengan bahasa pemrograman *java* di mana aplikasi akan mengambil data sensor yang telah disimpan pada *Adafruit Cloud* dengan menggunakan *REST API* dengan *library Retrofit 2* melalui *internet*.

## II. METODE PENELITIAN

Pengembangan ini menggunakan metode *Rapid Application Development* yang mana proses pengembangan ini terdiri dari tiga tahap yaitu rencana kebutuhan, desain *workshop*, dan pengujian & implementasi.

### A. Rencana Kebutuhan

#### 1. Kebutuhan Pengguna

Dalam metode *Rapid Application Development*, cerita pengguna (*user story*) digunakan untuk menjelaskan kebutuhan fungsional pada sebuah aplikasi. Cerita tersebut yang akan menggambarkan bagaimana aplikasi akan berjalan. *User story* dideskripsikan berdasarkan hasil dari *interview* kepada *user*.

Pada pengembangan aplikasi ini terdapat aktor yang dapat mengakses sistem, yaitu *user*. *User* dalam sistem ini adalah juga sebagai pengembang yang sekaligus sebagai orang yang dapat melakukan kontrol pada sistem. *User* dapat melakukan fungsi *monitoring* setiap data sensor.

#### 2. Kebutuhan Perangkat Keras

Dalam aplikasi *monitoring greenhouse* berbasis *android*, perangkat keras digunakan dalam tahap

pengembangan sistem. Perangkat keras tersebut adalah sebagai berikut :

1. *Personal Computer* (PC) atau Laptop  
Perangkat keras yang digunakan dalam pengembangan aplikasi ini adalah sebuah komputer laptop Acer One 14 dengan spesifikasi sebagai berikut.
  - a. *Processor* Intel Core i3-6006U CPU @ 2.0GHz
  - b. Memori RAM 4GB DDR3L.
  - c. *Monitor* LCD 14 *inchi* 1366x768.
  - d. *Harddisk* 1 TB.
  - e. Kartu Grafis Intel HD *Graphics* 5500.
  - f. Sistem Operasi *Windows 10 Education* 64-bit *Build* 16299.

2. Ponsel Cerdas (*Smartphone*)

Ponsel cerdas yang digunakan menggunakan sistem operasi *Android*. Perangkat tersebut akan digunakan saat melakukan pengujian sistem pada aplikasi. Ponsel cerdas yang digunakan dalam pengembangan aplikasi adalah ponsel *Xiaomi Redmi 5A* dengan spesifikasi sebagai berikut.

- a. *Chipset Snapdragon* 425.
- b. *GPU Adreno* 308.
- c. Ukuran layar 5,0 *inchi* dengan resolusi 720 x 1280 *pixels*.
- d. Memori RAM 2GB.
- e. Sistem Operasi *Android v7.1.2 Nougat*.

### 3. Kebutuhan Perangkat Lunak

Selain perangkat keras, dalam tahap pengembangan aplikasi *Monitoring Greenhouse* berbasis *Android* membutuhkan perangkat lunak sebagai berikut:

- a. *Android Studio* 4.2 yang digunakan dalam membangun aplikasi berbasis *Android*.
- b. *Java Development Kit* (JDK) 8 *update* 101 agar dapat mendukung pengembangan aplikasi bahasa pemrograman *Java*.
- c. *Adafruit Cloud* dimana data dari sensor tersimpan

### 4. Kebutuhan Data

Kebutuhan data pada sistem ini telah tersedia pada layanan *cloud Adafruit IO*, yaitu data hasil bacaan sensor-sensor yang ada pada alat yang dikirimkan menggunakan *internet*. Data hasil bacaan sensor dibagi menjadi dua *feeds* yang tersimpan pada layanan *cloud* yaitu *feed* suhu dan *feed* kelembaban udara. Nilai-nilai dari bacaan sensor tersebut yang kemudian akan ditampilkan pada aplikasi *monitoring greenhouse* berbasis *android*.

### B. Desain Workshop

Setelah tahap perencanaan dilakukan, tahap selanjutnya adalah tahap desain atau perancangan dari *workshop* sistem. Pada tahap desain *workshop* terdapat 3 proses yaitu *build* (pengembangan), *demonstrate*, dan *refine*. Pada proses *build* (pengembangan) dilakukan pemodelan sistem aplikasi yang akan dibuat. Pada

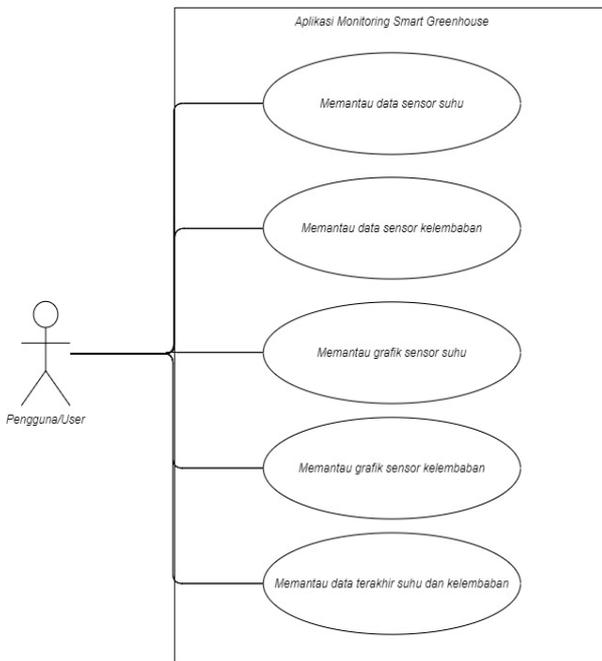


proses pengembangan, perancangan menggunakan UML (*Unified Model Language*) yang merupakan pemodelan dan analisis berorientasi objek. UML yang digunakan adalah Diagram *Use Case* dan Diagram Aktivitas. Pada proses pengembangan juga dilakukan perancangan antarmuka aplikasi.

### 1. Build (Pengembangan)

#### 1.1 Perancangan dengan menggunakan Diagram *Use Case*

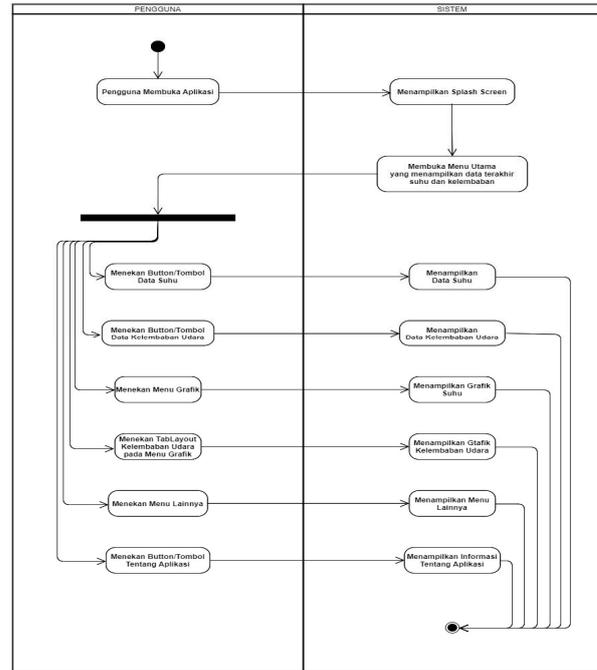
Diagram *Use Case* merupakan sebuah diagram yang menggambarkan kebutuhan fungsional atau layanan disediakan oleh aplikasi kepada pengguna. Pada diagram sistem terdapat satu aktor bernama *user* yang menggunakan sistem. Diagram *Use Case* dari aplikasi ini di tunjukkan pada Gambar 1.



Gambar 1. Diagram *Use Case*

#### 1.2 Perancangan dengan menggunakan Diagram Aktivitas

Diagram aktivitas menggambarkan aliran aktivitas yang dilakukan pada sistem secara berurutan yang mengacu pada skenario dari diagram *use case* dari sistem yang dirancang. Diagram aktivitas dari Aplikasi Monitoring Greenhouse berbasis Android yang ditunjukkan pada gambar 2 menggambarkan bagaimana *user* mengakses aplikasi *monitoring greenhouse*. Terdapat 3 menu yang dapat diakses oleh *user* seperti *menu greenhouse* yang menjadi *menu utama* pada aplikasi ini. Pada *menu greenhouse* aplikasi akan menampilkan data terakhir dari sensor suhu dan sensor kelembaban udara dan terdapat 2 buah *button* yaitu *button* data suhu dan data kelembaban udara yang berfungsi menampilkan semua data dari masing-masing sensor.



Gambar 2. Diagram Aktivitas *User*

Kemudian menu yang kedua ada *menu grafik* di mana pada *menu* ini akan menampilkan *tablayout* suhu dan kelembaban udara, pada *tablayout* suhu, data dari sensor akan ditampilkan dalam bentuk grafik, begitu pula pada *tablayout* kelembaban udara ditampilkan dalam bentuk grafik. Dan yang terakhir yaitu *menu options* atau *menu lainnya* yang dapat diakses oleh *user* yang menampilkan fungsi informasi tentang aplikasi.

#### 1.3 Perancangan Antarmuka Aplikasi (*User Interface Design*)

##### a. *Splash Screen Activity*



Gambar 3. Rancangan Tampilan *Splash Screen*

Ketika *user* membuka aplikasi, beberapa saat sebelum menampilkan halaman utama aplikasi akan menampilkan sebuah *activity* yaitu *splash screen activity*. Pada *activity* ini akan menampilkan identitas dari aplikasi yang dibuat berupa logo dan nama dari aplikasi tersebut. *Splash screen activity* ini hanya akan tampil pada saat *user* pertama kali membuka aplikasi. Rancangan tampilan pada *splash screen activity* dapat dilihat pada gambar 3.

### b. Menu Greenhouse Activity

Ketika *user* membuka aplikasi, maka halaman yang pertama ditampilkan adalah halaman *menu greenhouse*. *Menu greenhouse* pada sistem ini tampil beberapa saat setelah menampilkan *splash screen activity*. Pada *activity* ini data terakhir dari sensor-sensor akan ditampilkan secara *real-time*. Rancangan tampilan pada *menu greenhouse* dapat dilihat sebagai berikut :



Gambar 4. Rancangan Tampilan Menu Greenhouse

### c. Data Suhu Activity

*User* dapat melihat data *sensor* suhu yang tersimpan pada *Adafruit Cloud*. Data tersebut bisa diakses melalui aplikasi *monitoring* ini. Pada *menu utama/greenhouse*, *user* dapat menekan *button* data suhu, yang kemudian sistem akan membawanya ke *activity* lain pada aplikasi. Rancangan tampilan dari data suhu adalah sebagai berikut :



Gambar 5. Rancangan Tampilan Data Suhu

### d. Data Kelembaban Udara Activity

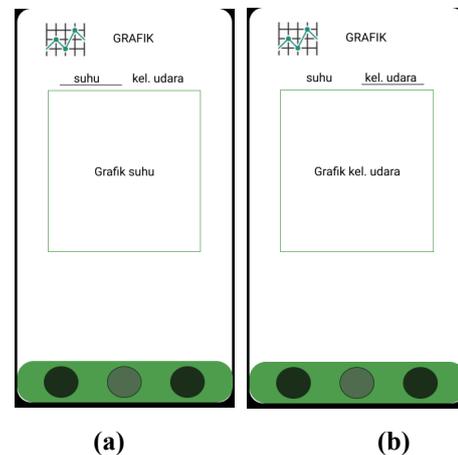
*User* dapat melihat data *sensor* kelembaban udara yang tersimpan pada *Adafruit Cloud*. Data tersebut bisa diakses melalui aplikasi *monitoring* ini. Pada *menu utama/greenhouse*, *user* dapat menekan *button* data kelembaban udara, yang kemudian sistem akan membawanya ke *activity* lain pada aplikasi. Rancangan tampilan dari data suhu adalah sebagai berikut :



Gambar 6 Rancangan Tampilan Data Kelembaban Udara

### e. Menu Grafik

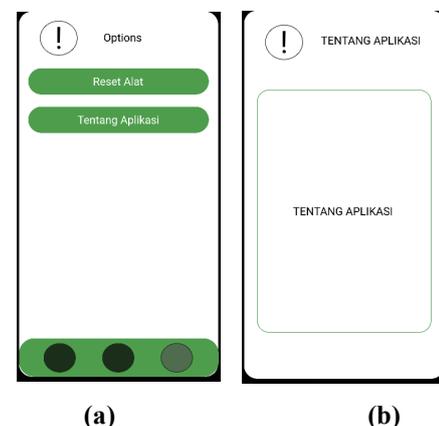
*Menu* grafik adalah sebuah *menu* di mana pengguna atau *user* dapat melihat data dari *sensor* yang direpresentasikan dalam bentuk grafik. Berikut adalah rancangan dari *menu* grafik.



Gambar 7 Rancangan Tampilan Menu Grafik

### f. Menu Options

Pada aplikasi *monitoring greenhouse* terdapat *menu options*, di mana *menu* ini menampilkan *button* untuk melakukan *reset* alat pada *greenhouse* dan juga *button* tentang aplikasi di mana *button* ini akan menuju sebuah *activity* lain yang akan menampilkan informasi tentang aplikasi ini.



Gambar 8 Rancangan Tampilan Menu Options



### III. PENGUJIAN SISTEM

Pada Tahap ini pengujian dilakukan dengan melihat fungsionalitas dari sistem yang telah dibangun yang bertujuan untuk mengetahui apakah aplikasi yang telah dibangun sudah sesuai dengan apa yang telah direncanakan dan dilakukan usability testing yang bertujuan untuk mengetahui apakah user dapat mudah menggunakan aplikasi, seberapa efisien dan efektif sebuah aplikasi dapat membantu user mencapai tujuannya. Pengujian secara fungsional ini dilakukan dengan cara menguji keselarasan dari sistem. Sedangkan usability testing dilakukan dengan 2 metode yakni System Usability Scale (SUS) dan Single Ease Question (SEQ).

#### A. Pengujian Fungsional

##### 1. Pengujian Keselaran Sistem

Pengujian ini dilakukan dengan mengecek apakah aplikasi yang telah dibangun dapat berjalan dengan baik atau tidak, mulai dari menampilkan data terakhir dari sensor, menampilkan seluruh data sensor dan grafik data sensor.

| Created at            | Value | Location |
|-----------------------|-------|----------|
| 2021/09/10 12:40:06AM | 24    |          |
| 2021/09/10 12:40:03AM | 24    |          |
| 2021/09/10 12:39:56AM | 24    |          |
| 2021/09/10 12:39:51AM | 24    |          |
| 2021/09/10 12:39:47AM | 24    |          |
| 2021/09/10 12:39:42AM | 26    |          |
| 2021/09/10 12:39:37AM | 26    |          |
| 2021/09/10 12:39:33AM | 26    |          |
| 2021/09/10 12:39:28AM | 26    |          |
| 2021/09/10 12:39:24AM | 26    |          |

Gambar 9 Data Feeds Suhu

Pada Gambar 9 dan Gambar 10 adalah data yang terdapat pada *adafruit cloud*. Terdapat 2 feeds pada *adafruit cloud* yakni feeds suhu dan feeds kelembaban udara yang berfungsi untuk menyimpan data yang dikirimkan dari prototipe *smart greenhouse*. Kedua data di atas merupakan data mentah atau data *dummy* yang dimasukkan secara manual.

| Created at            | Value | Location |
|-----------------------|-------|----------|
| 2021/09/10 12:42:14AM | 68    |          |
| 2021/09/10 12:42:08AM | 68    |          |
| 2021/09/10 12:42:04AM | 68    |          |
| 2021/09/10 12:41:59AM | 68    |          |
| 2021/09/10 12:41:54AM | 68    |          |
| 2021/09/10 12:41:50AM | 70    |          |
| 2021/09/10 12:41:45AM | 70    |          |
| 2021/09/10 12:41:40AM | 70    |          |
| 2021/09/10 12:41:36AM | 70    |          |
| 2021/09/10 12:41:30AM | 70    |          |

Gambar 10 Data Feeds Kelembaban Udara



Gambar 11 Tampilan Data Terakhir Suhu dan Kelembaban Udara Kondisi OFF

Pada Gambar 11 menunjukkan data terakhir dari suhu dan kelembaban udara yang sudah berhasil terhubung dengan dengan aplikasi android, data tersebut masih berupa data *dummy* yang diambil dari *adafruit cloud* sebelum prototipe *smart greenhouse* dijalankan.

| Temperatur | Waktu               |
|------------|---------------------|
| 24°C       | 10/09/2021 00:40:06 |
| 24°C       | 10/09/2021 00:40:03 |
| 24°C       | 10/09/2021 00:39:56 |
| 24°C       | 10/09/2021 00:39:51 |
| 24°C       | 10/09/2021 00:39:47 |
| 26°C       | 10/09/2021 00:39:42 |
| 26°C       | 10/09/2021 00:39:37 |
| 26°C       | 10/09/2021 00:39:33 |
| 26°C       | 10/09/2021 00:39:28 |
| 26°C       | 10/09/2021 00:39:24 |

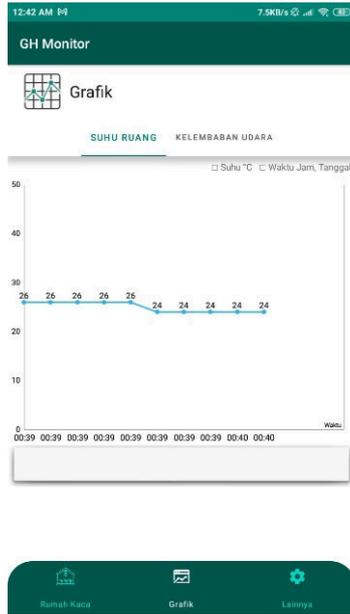
| Kelembaban | Waktu               |
|------------|---------------------|
| 68%        | 10/09/2021 00:42:14 |
| 68%        | 10/09/2021 00:42:08 |
| 68%        | 10/09/2021 00:42:04 |
| 68%        | 10/09/2021 00:41:59 |
| 68%        | 10/09/2021 00:41:54 |
| 70%        | 10/09/2021 00:41:50 |
| 70%        | 10/09/2021 00:41:45 |
| 70%        | 10/09/2021 00:41:40 |
| 70%        | 10/09/2021 00:41:36 |
| 70%        | 10/09/2021 00:41:30 |

(a)

(b)

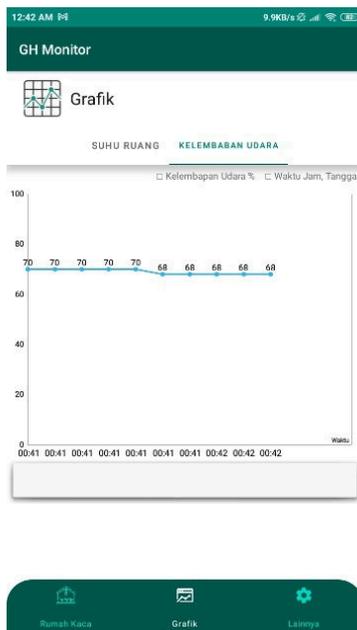
Gambar 12 Tampilan Data Suhu dan Kelembaban Udara Kondisi OFF

Pada Gambar 12(a) menunjukkan seluruh data sensor suhu begitu pula pada Gambar 12(b) data sensor kelembaban udara yang sudah berhasil terhubung dengan aplikasi android, data tersebut masih berupa data *dummy* yang diambil dari *adafruit cloud* sebelum prototipe *smart greenhouse* dijalankan.



Gambar 13 Tampilan Grafik Suhu Kondisi OFF

Gambar 13 menggambarkan tampilan dari grafik sistem pada sensor suhu, di mana sumbu Y menunjukkan nilai dari sensor, sedangkan pada sumbu X menunjukkan waktu dari data sensor didapatkan.



Gambar 14 Tampilan Grafik Kelembaban Udara Kondisi OFF

Gambar 14 menggambarkan tampilan dari grafik sistem pada sensor kelembaban udara, di mana sumbu Y menunjukan nilai dari sensor, sedangkan pada sumbu X menunjukan waktu dari data sensor didapatkan. Pengujian dilakukan dengan menyalakan alat yang berlokasi di Kos Afwan. Pengujian ini bertujuan untuk mengetahui keselarasan antara data yang dikirimkan oleh alat melalui internet ke *adafruit cloud* dan ditampilkan pada aplikasi *android* yang telah dibangun.

| Created at           | Value | Location  |
|----------------------|-------|-----------|
| 2021/09/10 1:47:42AM | 25    | 0.0.0.0.0 |
| 2021/09/10 1:47:37AM | 25    | 0.0.0.0.0 |
| 2021/09/10 1:47:31AM | 25    | 0.0.0.0.0 |
| 2021/09/10 1:47:25AM | 25    | 0.0.0.0.0 |
| 2021/09/10 1:47:19AM | 25    | 0.0.0.0.0 |
| 2021/09/10 1:47:13AM | 26    | 0.0.0.0.0 |
| 2021/09/10 1:47:08AM | 26    | 0.0.0.0.0 |
| 2021/09/10 1:47:02AM | 26    | 0.0.0.0.0 |
| 2021/09/10 1:46:56AM | 26    | 0.0.0.0.0 |
| 2021/09/10 1:46:50AM | 26    | 0.0.0.0.0 |
| 2021/09/10 1:46:45AM | 26    | 0.0.0.0.0 |
| 2021/09/10 1:46:39AM | 26    | 0.0.0.0.0 |

Gambar 15 Data Feeds Suhu saat Alat ON

| Created at           | Value | Location  |
|----------------------|-------|-----------|
| 2021/09/10 1:47:43AM | 61    | 0.0.0.0.0 |
| 2021/09/10 1:47:37AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:47:31AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:47:26AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:47:19AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:47:14AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:47:08AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:47:02AM | 58    | 0.0.0.0.0 |
| 2021/09/10 1:46:56AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:46:51AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:46:45AM | 60    | 0.0.0.0.0 |
| 2021/09/10 1:46:39AM | 60    | 0.0.0.0.0 |

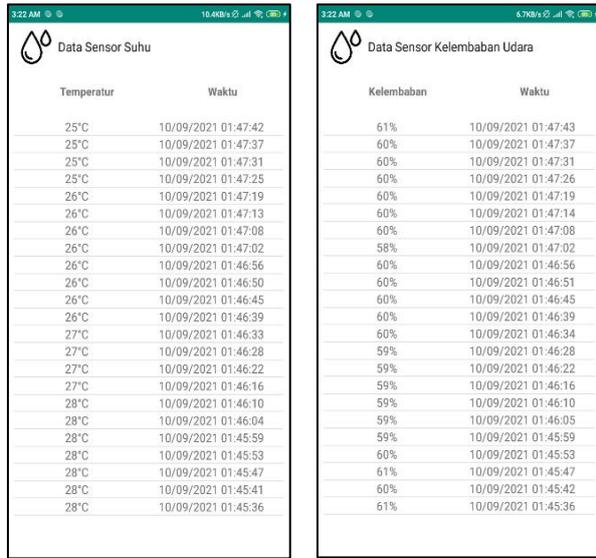
Gambar 16 Data Feeds Kelembaban Udara saat Alat ON

Pada Gambar 15 dan Gambar 16 merupakan tampilan dari data suhu dan kelembaban udara pada *adafruit cloud*. Pada saat alat dinyalakan, alat berhasil mengirim data sensor melalui internet dan disimpan pada *adafruit cloud*. Data tersebut yang akan ditampilkan pada aplikasi *android* yang telah dibangun.



Gambar 17 Data Terakhir Suhu dan Kelembaban Udara saat Alat ON

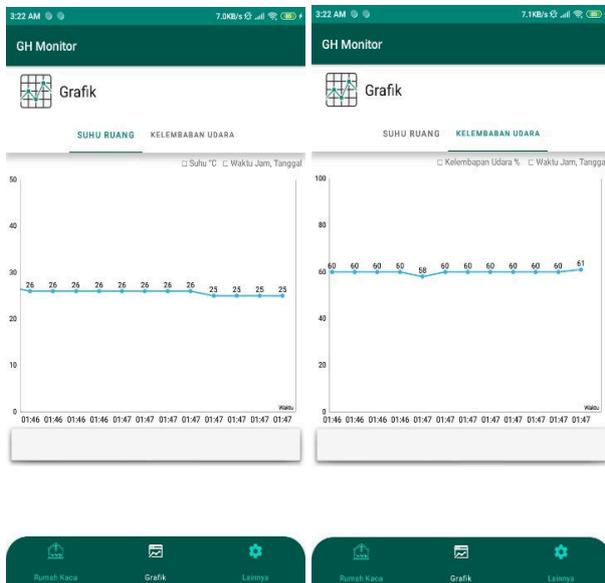
Gambar 17 menggambarkan tampilan halaman *Greenhouse* pada aplikasi *android* dan menampilkan data terakhir dari sensor suhu dan sensor kelembaban udara. Di sini dapat dilihat keberhasilan aplikasi mengambil data dari *adafruit cloud* dan menampilkan data terakhir. Data tersebut diambil ketika alat dalam kondisi nyala atau ON.



(a) (b)

Gambar 18 Data Suhu dan Kelembaban Udara saat Alat ON

Gambar 18(a) menggambarkan tampilan halaman data suhu dan pada Gambar 18(b) menggambarkan tampilan data kelembaban udara. Di sini dapat dilihat keberhasilan aplikasi mengambil data dari *adafruit cloud* dan menampilkan data suhu dan kelembaban udara yang tersimpan pada *adafruit cloud*. Data tersebut diambil ketika alat dalam kondisi nyala atau ON.



(a) (b)

Gambar 19 Grafik Suhu (a) dan Kelembaban Udara (b) saat Alat ON

Gambar 19 a menggambarkan tampilan halaman grafik suhu. Di sini dapat dilihat keberhasilan aplikasi mengambil data dari *adafruit cloud* dan menampilkan data suhu dalam bentuk grafik. Data tersebut diambil ketika alat dalam kondisi nyala atau ON.

Di sisi lain, Gambar 19 b menggambarkan tampilan halaman grafik kelembaban udara. Di sini dapat dilihat keberhasilan aplikasi mengambil data dari *adafruit cloud* dan menampilkan data kelembaban udara dalam bentuk grafik. Data tersebut diambil ketika alat dalam kondisi nyala atau ON.

## B. Usability Testing

### 1. System Usability Scale (SUS)

Pengujian *system usability scale* (SUS) ini merupakan metode pengujian *usability* suatu sistem secara sederhana dengan sepuluh pertanyaan yang memberikan pandangan secara menyeluruh dari evaluasi tujuan kegunaan sistem. *System usability scale* ini berupa skala *likert* yang sederhana dengan responden diharuskan menjawab tingkat kesetujuan dan ketidaksetujuan dalam skala 5 atau 7 poin.

Tabel 1 Penilaian Responden

| R  | Q 1 | Q 2 | Q 3 | Q 4 | Q 5 | Q 6 | Q 7 | Q 8 | Q 9 | Q 10 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 1  | 4   | 1   | 5   | 1   | 4   | 2   | 5   | 1   | 5   | 1    |
| 2  | 5   | 1   | 1   | 1   | 5   | 1   | 5   | 1   | 5   | 2    |
| 3  | 3   | 2   | 5   | 2   | 3   | 4   | 5   | 2   | 3   | 3    |
| 4  | 5   | 2   | 5   | 2   | 5   | 2   | 5   | 2   | 3   | 2    |
| 5  | 5   | 1   | 5   | 1   | 5   | 2   | 5   | 1   | 5   | 2    |
| 6  | 4   | 1   | 4   | 1   | 4   | 1   | 4   | 1   | 4   | 1    |
| 7  | 5   | 2   | 5   | 2   | 5   | 1   | 5   | 2   | 5   | 4    |
| 8  | 5   | 1   | 5   | 1   | 5   | 1   | 5   | 1   | 5   | 1    |
| 9  | 4   | 1   | 5   | 3   | 5   | 1   | 5   | 1   | 5   | 1    |
| 10 | 5   | 1   | 5   | 5   | 5   | 1   | 5   | 1   | 5   | 5    |
| 11 | 5   | 2   | 5   | 4   | 4   | 5   | 5   | 1   | 5   | 4    |
| 12 | 3   | 1   | 5   | 1   | 5   | 3   | 4   | 2   | 5   | 2    |
| 13 | 3   | 1   | 5   | 1   | 5   | 4   | 4   | 1   | 5   | 1    |
| 14 | 3   | 2   | 4   | 3   | 5   | 2   | 4   | 2   | 4   | 3    |
| 15 | 4   | 1   | 5   | 1   | 5   | 2   | 4   | 1   | 5   | 1    |

Tabel 1 merupakan penilaian yang diberikan oleh responden terhadap aplikasi yang telah dibangun. Terdapat 15 responden yang memberikan penilaian dimana masing-masing responden menjawab 10 pertanyaan dan memberikan nilai antara 1-5 dengan skala *likert* pada masing-masing pertanyaan. Dalam aturan menghitung SUS untuk mendapatkan skor SUS, setiap pertanyaan bernomor ganjil yaitu nomor 1, 3, 5, 7, dan 9, skor dari setiap pertanyaan yang didapat dari penilaian responden akan dikurangi 1. Sementara untuk setiap pertanyaan bernomor genap yaitu nomor 2, 4, 6, 8, dan 10, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari penilaian responden. Kemudian skor SUS yang didapat dari hasil penjumlahan skor setiap pertanyaan dikali dengan 2,5. Aturan perhitungan skor ini berlaku untuk 1 responden.



Tabel 2 adalah hasil perhitungan skor SUS dari setiap responden.

Tabel 2 Hasil Skor SUS Setiap Responden

| Responden | Jumlah | Nilai (Jumlah x 2,5) |
|-----------|--------|----------------------|
| 1         | 37     | 92,5                 |
| 2         | 35     | 87,5                 |
| 3         | 26     | 65                   |
| 4         | 33     | 82,5                 |
| 5         | 38     | 95                   |
| 6         | 35     | 87,5                 |
| 7         | 34     | 85                   |
| 8         | 40     | 100                  |
| 9         | 37     | 92,5                 |
| 10        | 32     | 80                   |
| 11        | 28     | 70                   |
| 12        | 33     | 82,5                 |
| 13        | 34     | 85                   |
| 14        | 28     | 70                   |
| 15        | 37     | 92,5                 |
| TOTAL     |        | 1267,5               |

Pada Tabel 2 sudah didapatkan skor SUS dari setiap responden. Untuk perhitungan selanjutnya, skor SUS dari masing-masing responden dicari skor rata-rata dari SUS dengan cara menjumlahkan seluruh skor SUS dan dibagi dengan jumlah responden. Berikut rumus menghitung skor SUS :

$$\bar{x} = \frac{\sum x}{n} \quad (1)$$

Keterangan :

$\bar{X}$  = skor rata-rata

$\sum x$  = jumlah/total skor SUS

$n$  = jumlah responden

Dari hasil perhitungan skor SUS pada table 4.3 didapatkan jumlah atau total skor SUS 1267,5 dari total 15 jumlah responden. Sehingga berdasarkan rumus di atas dapat dihitung skor rata-rata SUS adalah 84,5. Dari perhitungan tersebut dapat disimpulkan aplikasi yang dibangun masuk pada kategori *excellent* dengan *grade scale* B. Artinya secara *usability* berdasarkan data tersebut mendapatkan penilaian dapat diterima atau layak.

## 2. Single Ease Question (SUQ)

Pengujian *single ease question* (SEQ) merupakan metode pengujian *usability* suatu sistem dengan menggunakan kuesioner *post-scenario* atau *task* untuk mengukur kemudahan yang dirasakan *user* atau pengguna setelah menyelesaikan semua skenario atau *task* yang berikan dengan menggunakan skala *likert* 7 poin, di mana angka 1 diartikan sangat sulit dan angka 7 diartikan sangat mudah. Pada pengujian *single ease question* (SEQ), terdapat 15 responden yang memberikan penilaian. Berikut Tabel 4.6 hasil penilaian dari responden.

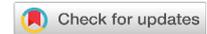
Tabel 3 Hasil Penilaian Responden

| R     | T1 | T2 | T3 | T4 | T5 | T6 | Jml Nilai 6 | Jml Nilai 7 |
|-------|----|----|----|----|----|----|-------------|-------------|
| 1     | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 2     | 7  | 7  | 7  | 7  | 6  | 7  | 1           | 5           |
| 3     | 7  | 7  | 6  | 7  | 6  | 7  | 2           | 4           |
| 4     | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 5     | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 6     | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 7     | 7  | 6  | 6  | 7  | 7  | 7  | 2           | 4           |
| 8     | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 9     | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 10    | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 11    | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 12    | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 13    | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 14    | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| 15    | 7  | 7  | 7  | 7  | 7  | 7  |             | 6           |
| TOTAL |    |    |    |    |    |    | 5           | 85          |

Pada Tabel 3 merupakan hasil penilaian dari responden setelah mengerjakan *post-scenario* atau *task* yang diberikan. Dari total 15 responden jumlah suara pada angka 7 yang diberikan oleh responen sebanyak 85 suara, dan jumlah suara pada angka 6 sebanyak 5 suara. Sedangkan untuk nilai 1 sampai 5 tidak ada. Dari data tersebut dapat dihitung persentase SEQ dengan cara total suara pada angka 7 dibagi dengan total suara kemudian dikalikan 100%.

$$\left(\frac{85}{90}\right) 100\% = 94\% \quad (2)$$

Sehingga dapat disimpulkan bahwa aplikasi ini sangat mudah untuk digunakan.



#### IV. Kesimpulan

Hasil pengujian keselarasan sistem sesuai dengan yang diharapkan. Pada pengujian ini aplikasi dapat bekerja dengan baik menampilkan data terakhir dari sensor suhu dan sensor kelembaban, menampilkan seluruh data sensor suhu dan data sensor kelembaban, dan menampilkan data sensor suhu dan kelembaban dalam bentuk grafik. Dari usability testing dengan metode *system usability scale* (SUS) diketahui bahwa aplikasi mendapatkan skor 84,5 masuk pada kategori excellent dengan grade scale B dan secara usability dapat diterima atau layak. Sementara itu, dari usability testing dengan metode *single ease question* (SEQ) diketahui bahwa aplikasi mendapatkan persentase penilaian dengan skor 94% yang artinya aplikasi ini sangat mudah digunakan.

Saran untuk penelitian selanjutnya adalah penambahan fungsi kontrol pada aplikasi seperti melakukan kontrol pada keran, pendingin ruangan, atau pemanas guna mendapatkan hasil yang maksimal pada budidaya tanaman. Penambahan fitur pemberitahuan pada aplikasi apabila ada perubahan yang tidak sesuai dengan parameter yang ditentukan dan penambahan fitur webcam pada sistem maupun aplikasi sehingga pengguna dapat melihat secara jelas kondisi di dalam greenhouse.

#### DAFTAR PUSTAKA

- [1] K. Nanda Kumar, A. Vijayan Pillai, and M. K. Badri Narayanan, "Smart agriculture using IoT," *Mater. Today Proc.*, no. xxxx, 2021, doi: 10.1016/j.matpr.2021.02.474.
- [2] J. Wang, M. Chen, J. Zhou, and P. Li, "Data communication mechanism for greenhouse environment monitoring and control: An agent-based IoT system," *Inf. Process. Agric.*, vol. 7, no. 3, pp. 444–455, 2020, doi: 10.1016/j.inpa.2019.11.002.
- [3] A. R. Firdhausi, A. Budiyanto, and I. Nurcahyani, "Rancang Bangun Smart Greenhouse untuk Budidaya Tanaman Cabai ( *Capsicum Annum L.* ) dengan OSAndroid," vol. 2018, no. November, pp. 16–22, 2018.
- [4] N. Wicaksana, F. Hadary, and A. Hartoyo, "Rancang Bangun Sistem Monitoring Smart Greenhouse Berbasis Android Dengan Aplikasi Sensor Suhu , Kelembaban Udara Dan Tanah Untuk Budidaya Jamur Merang," *J. Tek. Elektro Univ. Tanjungpura*, vol. 2, no. 1, pp. 1–5, 2018.
- [5] L. C. Wijaya, "Analisis Usabilitas pada Sistem Monitoring dan Otomasi Greenhouse untuk Budidaya Tanaman Cabai Berbasis Android," *Edu Komputika J.*, vol. 6, no. 2, pp. 60–67, 2019, doi: 10.15294/edukomputika.v6i2.34412.
- [6] E. Lumba, "Pertukaran Data Pada Android Menggunakan Java Script Object Notation (JSON) Dan REST API Dengan Retrofit 2," *Pros. Has. Nas. Apl. Sains Teknol.*, pp. 118–127, 2021.
- [7] E. V. Nusantara, I. Ardiansah, and N. Bafdal, "Desain Sistem Otomatisasi Pengendalian Suhu Rumah Kaca Berbasis Web Pada Budidaya Tanaman Tomat," *J. Keteknikan Pertan. Trop. dan Biosist.*, vol. 9, no. 1, pp. 34–42, 2021, doi: 10.21776/ub.jkptb.2021.009.01.05.
- [8] A. Herliana and P. M. Rasyid, "Sistem Informasi Monitoring Pengembangan Software Pada Tahap," *J. Inform.*, no. 1, pp. 41–50, 2016.
- [9] J. Friesen, *Learn Java for Android Development*, Third Edit. Apress, 2014.
- [10] M. Natsir, "Pengembangan Prototype Sistem Kriptografi Untuk Enkripsi Dan Dekripsi Data Office," *Jurnal*, vol. 6, pp. 2089–5615, 2016.
- [11] A. Juansyah, "Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS ) Dengan Platform Android," *J. Ilm. Komput. dan Inform.*, vol. 1, no. 1, pp. 1–8, 2015.
- [12] S. Mulyati and Wardono, "Kreativitas Matematis Siswa Pada Pembelajaran Discovery Learning Dengan Media Berbasis Android Studio," *Prism. Pros. Semin. Nas. Mat.*, vol. 2, pp. 788–797, 2019.
- [13] M. Ali and M. K. Paracha, "an Iot Based Approach for Monitoring Solar Power Consumption With Adafruit Cloud," *Int. J. Eng. Appl. Sci. Technol.*, vol. 04, no. 09, pp. 335–341, 2020, doi: 10.33564/ijeast.2020.v04i09.042.
- [14] A. Krispriansa, N. Kumaladewi, and E. Rahajeng, "Sistem Informasi Computer Assisted Test (Cat) Kementerian Agama Republik Indonesia," *Stud. Inform. J. Sist. Inf.*, vol. 9, no. 2, pp. 203–211, 2018.
- [15] M. P. Puteri and H. Effendi, "Implementasi Metode RAD Pada Website Service Guide 'Tour Waterfall South Sumatera,'" *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 7, no. 2, p. 130, 2018, doi: 10.32736/sisfokom.v7i2.570.
- [16] D. Hariyanto et al., "Implementasi Metode Rapid Application Development Pada Sistem Informasi Perpustakaan," *J. JUPITER*, vol. 13, no. 1, pp. 110–117, 2021.