

## SISTEM PENGENDALIAN DAN PENGAWASAN PETERNAKAN AYAM BROILER MENGUNAKAN APLIKASI *MOBILE* BERBASIS ANDROID

### *BROILER CHICKEN FARM CONTROL AND MONITORING SYSTEM USING ANDROID BASED MOBILE APPLICATION*

Ghalib Hisyam Ramandita<sup>1</sup>, Ir. Ahmad Tri Hanuranto, M.T.<sup>2</sup>, Ratna Mayasari, S.T., M.T.<sup>3</sup>

Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>ghalibhisyam@student.telkomuniversity.ac.id, <sup>2</sup>athanuranto@telkomuniversity.ac.id,

<sup>3</sup>ratnamayasari@telkomuniversity.ac.id

#### Abstrak

Ayam broiler merupakan salah satu komoditas yang paling diminati diantara hasil peternakan lainnya, karena memiliki kandungan protein yang tinggi. Menurut data dari Badan Pusat Statistik di tahun 2018, jumlah konsumsi daging ayam per kapita meningkat dengan rata – rata 0.5 kg setiap tahunnya. Hal ini membuktikan bahwa kebutuhan masyarakat terhadap daging ayam selalu meningkat sehingga membuat usaha peternakan ayam broiler adalah usaha yang cukup menjanjikan. Namun banyak parameter yang dapat memengaruhi kesehatan dan mutu ayam broiler diantaranya, perubahan suhu dan kelembapan, jumlah konsumsi pakan dan air minum, serta kadar gas amonia yang ditimbulkan oleh kotoran ayam. Pada penelitian ini penulis merancang sebuah aplikasi *mobile* berbasis Android untuk mengontrol dan memonitor peternakan ayam. Aplikasi *mobile* dapat mengirimkan perintah dari smartphone melalui *web server* yang telah terhubung dengan sensor dan alat berbasis *Internet of Things* yang ada di kandang, sebaliknya sensor yang ada di kandang akan mengirimkan data berupa nilai dari parameter yang ada untuk ditampilkan pada aplikasi *mobile*. Data tentang parameter kondisi pada kandang ayam akan ditampilkan secara *realtime* dan diolah menjadi laporan harian pada aplikasi *mobile*. Hasil pengujian fungsionalitas, seluruh fitur yang terdapat di aplikasi *mobile* dapat dijalankan dengan baik. Untuk pengujian *delay* aplikasi *mobile* pada proses *write*, *delay* rata – rata terendah 275.8 ms. Sedangkan pada proses *read*, *delay* rata – rata terendah 625.6 ms, sehingga dapat disimpulkan bahwa *delay* cukup bagus.

Kata kunci : *Smart Poultry Farm*, Ayam Broiler, Android, *Internet of Things*.

#### Abstract

*Broiler chicken is one of the most popular commodities among other livestock products, because it has a high protein content. According to data from Badan Pusat Statistik in 2018, the amount of chicken meat consumption per capita increases by an average of 0.5 kg each year. This proves that the community's need for chicken meat is always increasing so making broiler chicken farming is a promising business. However, many parameters can affect the health and quality of broilers, including changes in temperature and humidity, the amount of consumption of feed and drinking water, as well as ammonia gas levels caused by chicken manure. In this study the author designed an Android-based mobile application to control and monitor chicken farms. The mobile application can send commands from the smartphone via a web server that has been connected with sensors and Internet of Things based devices that are in the coop, otherwise sensors in the enclosure will send data in the form of values from existing parameters to be displayed on the mobile application. Data about the condition parameters in the chicken coop will be displayed in realtime and processed into a daily report on the mobile application. The results of testing the functionality, all features contained in the mobile application can be run properly. For testing the delay of mobile applications in the process of writing, the average delay was 275.8 ms. Whereas in the read process an average delay 625.6 ms, is obtained so that it can be concluded that the delay is quite good.*

Keywords : *Smart Poultry Farm*, Broiler Chicken, Android, *Internet of Things*.

## 1. Pendahuluan

Peternakan ayam broiler adalah sebuah usaha yang sangat menguntungkan, setiap tahunnya jumlah permintaan untuk konsumsi daging ayam semakin meningkat. Hal ini sebanding dengan manfaat daging ayam yang memiliki kandungan protein yang cukup tinggi [1]. Menurut data dari Badan Pusat Statistik di tahun 2018, jumlah konsumsi daging ayam per kapita meningkat dengan rata – rata 0.5 kg setiap tahunnya [2]. Hal ini membuktikan bahwa kebutuhan masyarakat terhadap daging ayam selalu meningkat. Namun banyak faktor yang harus diperhatikan dalam pengelolaan peternakan ayam, seperti suhu, kelembapan dan konsumsi pakan yang disesuaikan berdasarkan usia ayam agar mendapat hasil panen yang maksimal [3]. Oleh karena itu sistem *smart poultry farm* berbasis *Internet of Things* dapat diimplementasikan sebagai solusi dalam pengelolaan peternakan ayam yang efisien.

Penggunaan sistem berbasis IoT dapat menjadi solusi untuk pengelolaan peternakan ayam. Sistem ini menggunakan mikrokontroler dan sensor pada perangkat dalam kandang, yang bisa mendapatkan data seperti ketersediaan pakan dan minum, suhu dan kelembapan kandang, serta kadar amonia didalam kandang ayam tersebut.

Sistem *smart poultry farm* terbagi menjadi tiga sub segmen yaitu alat, *web server* dan aplikasi *mobile* yang akan saling terintegrasi. Pada penelitian ini akan terfokus pada perancangan aplikasi *mobile*. Dengan aplikasi *mobile* dan sistem yang akan dirancang, peternak dapat melihat dan mengontrol kondisi peternakan dari jarak jauh dan tidak perlu berulang kali ke kandang untuk sekedar mengecek kondisi peternakan. Maka dari itu diharapkan sistem ini dapat menjadi solusi dari segi efisiensi produksi dalam peternakan unggas, khususnya ayam broiler sehingga dapat membantu meningkatkan produktifitas kegiatan peternakan ayam.

## 2. Dasar Teori

### 2.1 Internet of Things

Internet of Things (IoT) merupakan suatu konsep dimana infrastruktur jaringan dapat menghubungkan antar perangkat yang saling terkoneksi melalui internet secara terus menerus. IoT menghubungkan perangkat elektronik sehingga akan memudahkan terhadap proses pengendalian dan kerja sama antara satu perangkat dengan perangkat lainnya [7].

### 2.2 Android

Android adalah sebuah sistem operasi berbasis Linux yang dirancang untuk perangkat *mobile* seperti *smartphone*, komputer dan *tablet*. Sistem operasi android ini bersifat *opensource*, sehingga Android memiliki komunitas pengembang aplikasi yang banyak dan tersebar diseluruh dunia. Pengembang aplikasi Android umumnya melakukan kustomisasi fungsional perangkat dalam bahasa pemrograman Java [8].

### 2.3 Java

Java adalah merupakan bahasa pemrograman dinamis berbasis objek yang dapat dijalankan di berbagai jenis komputer dan *handphone*. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal [9].

### 2.4 White-box Testing

*White box testing* adalah salah satu metode untuk menguji sebuah aplikasi atau *software* dengan menganalisa apakah ada kode yang salah pada program yang telah dibuat. Pengujian ini didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian[10].

### 2.5 Black-box Testing

*Black box testing* adalah suatu metode untuk menguji spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengtesan pada spesifikasi fungsional program. Metode ini merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *white-box testing* [10].

### 2.6 Compatibility Testing

Pengujian *compatibility* adalah sebuah metode pengujian aplikasi yang digunakan untuk mengetahui mampu atau tidaknya sebuah aplikasi untuk berjalan pada *hardware*, sistem operasi, ataupun lingkungan instalasi yang berbeda [11].

### 2.7 User Experience Questionnaire

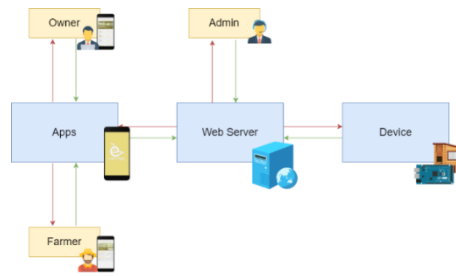
*User Experience Questionnaire* (UEQ) adalah sebuah alat kuesioner yang digunakan untuk mengukur *User Experience* (UX). UEQ dapat digunakan untuk menguji *user experience* sebuah produk dan menentukan area perbaikan. Pengujian ini berisi enam skala penilaian antara lain daya tarik, kejelasan, efisiensi, ketepatan, stimulasi dan kebaruan dari sebuah aplikasi. [12].

### 2.8 Delay

*Delay* adalah waktu yang dibutuhkan sebuah paket dari saat dikirim hingga ke tujuannya *delay* dikatakan buruk apabila nilainya diatas 400 ms. Pada penelitian ini *delay* dihitung saat aplikasi *mobile* mengirimkan perintah hingga data tersimpan ke Firebase, dan sebaliknya aplikasi *mobile* memanggil data dari Firebase ke aplikasi *mobile* [13].

### 3. Model Sistem dan Perancangan

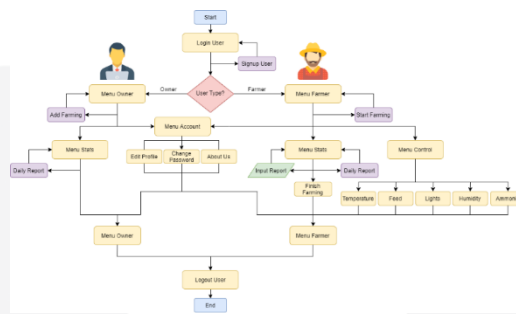
#### 3.1 Desain Sistem



Gambar 3.1 Desain sistem smart poultry farm

Sistem *smart poultry farm* adalah sebuah sistem yang dapat memonitor dan mengontrol peternakan ayam dari jarak jauh melalui aplikasi *mobile*. Sistem ini mengontrol dan memonitor parameter pertumbuhan ayam seperti suhu, kelembapan, intensitas pemberian pakan, kadar gas amonia serta kontrol penerangan kandang. Pada sistem ini akan terdapat beberapa sensor yang akan melakukan optimalisasi beberapa parameter tersebut secara otomatis melalui mikrokontroler serta mengumpulkan data – data tentang parameter pertumbuhan ayam yang ada di kandang untuk diproses dan diunggah ke *websrver*. Pada tahap selanjutnya, data akan diunduh dan ditampilkan dalam *User Interface* (UI) aplikasi *mobile* agar pengguna dapat memonitor kondisi kandang. Selain itu pengguna aplikasi *mobile* dapat melakukan kontrol terhadap kandang dengan proses sebaliknya melalui UI.

#### 3.1.1 Diagram Blok Sistem



Gambar 3.2 Diagram blok aplikasi mobile

Dalam perancangan aplikasi *mobile* ini, terdapat dua tipe pemngguna aplikasi yaitu *farmer* dan *owner*. *Farmer* sebagai pekerja di lapangan yang akan memantau secara *realtime* keadaan kandang dan mengirimkan laporan setiap harinya. *Owner* sebagai pemilik usaha peternakan yang dapat menerima laporan tentang perkembangan peternakannya melalui laporan harian yang telah di-*input* oleh *faremer*. Data tentang laporan parameter kondisi peternakan akan dikirimkan oleh alat melalui *web server* dan ditampilkan pada menu utama aplikasi *mobile*. Setelah mengetahui kondisi peternakan di tampilan menu utama, *farmer* dapat melakukan optimisasi kandang dengan menekan tombol sesuai parameter yang akan disesuaikan. Selain itu *farmer* dapat melakukan laporan harian sehingga keadaan peternakan juga dapat dipantau oleh *owner*.

### 3.2 Desain dan Perancangan Aplikasi Mobile

#### 3.2.1 Segmentasi Database

Karena keterlibatan lebih dari satu perangkat dan pengguna pada aplikasi *mobile* ini, diperlukan segmentasi *database* untuk manajemen data pada sistem. Data pada aplikasi *mobile* ini tersegmentasi menjadi beberapa bagian antara lain: *Userdata Owner*, *Userdata Farmer*, *Device ID*, *Controlling*, *Monitoring*, *Device Data*, dan *History*.

#### 3.2.2 Menu Utama Farmer



Gambar 3.3 Menu utama farmer

Tampilan menu utama *farmer* memiliki dua kondisi, yaitu saat aktivitas peternakan belum dimulai dan saat aktivitas peternakan sudah berlangsung. Saat aktivitas peternakan sudah berlangsung pengguna sudah terhubung dengan perangkat di kandang sehingga dapat dilakukan pengontrolan dan monitor terhadap kondisi kandang. Pada kondisi tanpa aktivitas peternakan, pengguna belum terhubung pada perangkat di kandang, sehingga akan ditampilkan tombol untuk memulai aktivitas peternakan.

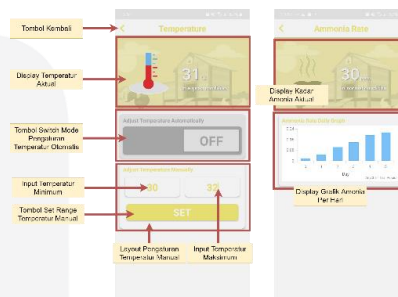
**3.2.3 Menu Statistik *Farmer***



**Gambar 3.4** Menu statistik *farmer*

Menu ini menampilkan data berupa grafik indeks performa peternakan. Selain itu ditampilkan statistik aktual peternakan antara lain: FCR, FI, ABW, deplesi, jumlah ayam mati, jumlah ayam afkir, indeks performa, populasi ayam dan total pakan yang dikeluarkan. Setiap harinya *farmer* melakukan input laporan harian berupa jumlah ayam mati, jumlah ayam afkir dan bobot rata – rata ayam melalui menu input laporan harian. Data aktual per hari akan tersimpan dan ditampilkan pada menu laporan harian. Tombol *finish farming* digunakan untuk mengakhiri kegiatan peternakan.

**3.2.4 Menu Monitor dan Kontrol Perangkat**



**Gambar 3.5** Menu kontrol dan monitor perangkat

Pada menu temperatur, pakan dan kelembapan tombol *switch* berfungsi untuk memilih mode otomatis dan manual. Jika pengguna memilih mode otomatis maka parameter kondisi kandang akan diatur otomatis oleh perangkat di kandang. Sedangkan apabila memilih mode manual maka pengguna dapat menyesuaikan kondisi kandang secara manual. Pengguna dapat menyalakan dan mematikan lampu melalui tombol *switch* pada menu pengaturan lampu. Kadar amonia dimonitor melalui menu amonia, pada menu ini ditampilkan grafik kenaikan amonia per hari sebagai tolok ukur *farmer* untuk melakukan pembersihan kandang.

**4. Analisis dan Hasil Perancangan**

**4.1 Pengujian Non Fungsional**

**4.1.1 White-box Testing**

Pengujian *white-box* dilakukan pada sampel *method* login.

**Tabel 4.1** Uji kasus *method* login

No.	Jalur	Data Masukan	Ekspektasi Hasil	Hasil	Status
1	1-2-3-9-10	Inisialisasi variabel : 1. email : subandi@ingonpitek.com 2. password : subandi	Muncul pesan “ <i>Successfully logged in as Owner</i> ” dan aktivitas berpindah ke OwnerMenuActivity.	Muncul pesan “ <i>Successfully logged in as Owner</i> ” dan aktivitas berpindah ke OwnerMenuActivity.	Valid
2	1-2-4-5-9-10	Inisialisasi variabel : 1. email : daffa@gmail.com 2. password : daffa	Muncul pesan “ <i>Successfully logged in as Farmer</i> ” dan aktivitas berpindah ke MenuActivity.	Muncul pesan “ <i>Successfully logged in as Farmer</i> ” dan aktivitas berpindah ke MenuActivity.	Valid

3	1-2-4-6-9-10	Inisialisasi variabel : 1. email : bilal@gmail.com 2. password : bilal	Muncul pesan "Successfully logged in as Farmer" dan aktivitas berpindah ke MenuNoActivity.	Muncul pesan "Successfully logged in as Farmer" dan aktivitas berpindah ke MenuNoActivity.	Valid
4	1-2-7-9-10	Inisialisasi variabel : 1. email : null 2. password : null	Tombol login tidak dapat ditekan.	Tombol login tidak dapat ditekan.	Valid
5	1-2-8-9-10	Inisialisasi variabel : 1. email : ghalib@gmail.com 2. password : ghalib	Muncul pesan "Email or password incorrect."	Muncul pesan "Email or password incorrect."	Valid

#### 4.1.2 Black-box Testing

Sama seperti *white-box testing*, pengujian ini mengambil sampel menu login.

**Tabel 4.2** Uji *black-box* menu login

Deskripsi	Test Step	Test Data	Expected Result	Actual Result
Melakukan login pada sistem.	1. User memasukkan email dan password. 2. User menekan tombol login.	email: subandi@ingonpitek.com password: subandi	Berhasil masuk kedalam sistem dan berpindah ke OwnerMenuActivity.	Verified
		email: daffa@gmail.com password: daffa	Berhasil masuk kedalam sistem dan berpindah ke MenuActivity.	Verified
		email: bilal@gmail.com password: bilal	Berhasil masuk kedalam sistem dan berpindah ke MenuNoActivity.	Verified
		email: robert@gmail.com password: robert	Gagal masuk kedalam sistem dan ditampilkan Toast: "Email or password incorrect."	Verified
		email dan/atau password kosong.	Tombol login tidak bisa di-klik.	Verified

## 4.2 Pengujian Non Fungsional

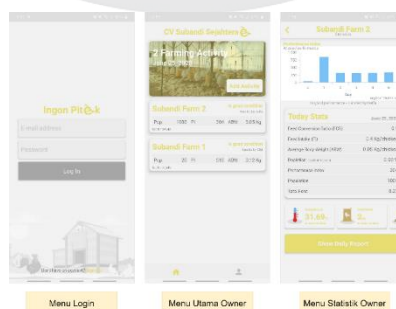
### 4.2.1 Compatibility Testing

Pada pengujian ini dilakukan pengujian compatibility pada beberapa perangkat *smartphone* berbasis Android antara lain Samsung SM-A507FN dan Haier Andromax Q.

**Tabel 4.3** Prosedur pengujian *compatibility*

<b>Prosedur</b>	1. Aplikasi <i>mobile</i> dijalankan. 2. Melakukan <i>login</i> . 3. Berpindah ke menu utama <i>owner</i> . 4. Membuka menu statistik <i>owner</i> .
<b>Hasil Yang Diharapkan</b>	Menu statistik <i>owner</i> ditampilkan.
<b>Hasil</b>	Menu statistik <i>owner</i> ditampilkan.
<b>Status</b>	Valid.

#### 1. Pengujian *Compatibility* Pada *Smartphone* Samsung SM-A507FN



**Gambar 4.1** Pengujian *compatibility* pada *Samsung SM-A507FN*

Pada Gambar 4.2 didapatkan hasil berupa menu *login*, menu utama *owner* dan menu statistik *owner* yang berhasil

di buka pada *smartphone* Samsung SM-A507FN. Hasil yang ditampilkan menunjukkan aplikasi dapat dibuka tanpa adanya kendala dan sesuai dengan hasil yang diharapkan.

2. Pengujian *Compatibility* Pada *Smartphone* Haier Andromax Q

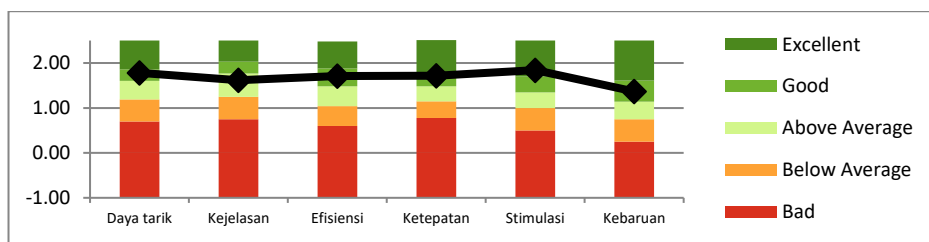
Pada pengujian ini, dilakukan dengan membuka mengakses aplikasi melalui perangkat *smartphone* Haier Andromax Q dengan versi android 5.0 dan dimensi layar 854 x 480 px.



Gambar 4.2 Pengujian *compatibility* pada Haier Andromax Q

Pada Gambar 4.3 didapatkan hasil berupa menu *login*, menu utama *owner* dan menu statistik *owner* yang berhasil di buka pada *smartphone* Haier Andromax Q. Hasil yang ditampilkan menunjukkan aplikasi dapat dibuka tanpa adanya kendala dan sesuai dengan hasil yang diharapkan.

4.2.2 *User Experience Questionnaire*



Gambar 4.3 Grafik hasil pengujian *user experience* dengan UEQ

Dari hasil pengujian didapatkan hasil daya tarik sebesar 1.78, kejelasan 1.62, efisiensi 1.71, ketepatan 1.72, stimulasi 1.84 dan kebaruan 1.32. Sehingga dapat disimpulkan bahwa *user experience* dari aplikasi *mobile* ini sudah cukup baik.

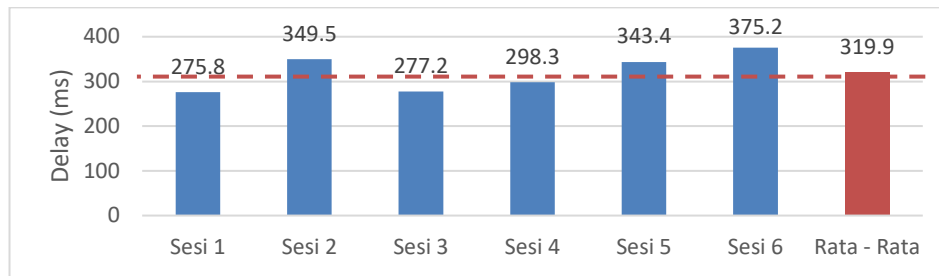
Tabel 4.4 Hasil pengujian *User Experience* dengan UEQ

Scale	Bad	Below Average	Above Average	Good	Excellent	Mean
Daya tarik	0.7	0.49	0.41	0.26	0.64	1.78
Kejelasan	0.75	0.5	0.52	0.26	0.47	1.62
Efisiensi	0.6	0.44	0.44	0.4	0.6	1.71
Ketepatan	0.78	0.37	0.33	0.23	0.8	1.72
Stimulasi	0.5	0.5	0.35	0.35	0.8	1.84
Kebaruan	0.25	0.5	0.39	0.47	0.89	1.37

4.3 Pengujian *Delay Aplikasi Mobile*

Pengujian *delay* aplikasi *mobile* dilakukan dengan mengambil dua sampel aktivitas yaitu aktivitas *write*/unggah dan *read*/unduh data dari *Firestore*. Pengujian *write* dilakukan pada menu pengaturan lampu, sedangkan pengujian *read* dilakukan pada menu statistik *owner*.

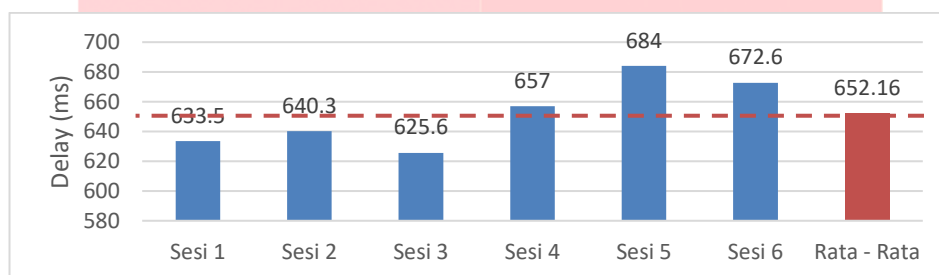
### 4.3.1 Pengujian *Delay* Pada Proses *Write Data* Dari Firebase



**Gambar 4.4** Grafik pengujian *delay* pada proses *write data*

Dapat dilihat pada Gambar 4.11, pengukuran *delay* pada proses *write data* ke Firebase memiliki nilai rata - rata terendah pada sesi 1 dengan nilai 275.8 ms, sedangkan untuk nilai rata-rata tertinggi berada pada sesi 6 yang didapatkan sebesar 375.2 ms.

### 4.3.2 Pengujian *Delay* Pada Proses *Read Data* Dari Firebase



**Gambar 4.5** Grafik pengujian *delay* pada proses *read data*

Dapat dilihat pada Gambar 4.12, pengukuran *delay* pada proses *read data* dari Firebase memiliki nilai rata-rata terendah pada sesi 3 dengan nilai 625.6 ms, sedangkan untuk nilai rata-rata tertinggi berada pada sesi 6 yang didapatkan sebesar 684 ms.

## 5. Penutup

### 5.1 Kesimpulan

Dari hasil perancangan sistem, pengujian, dan analisis pada penelitian Tugas Akhir ini, maka penulis mendapatkan beberapa kesimpulan yaitu:

1. Aplikasi *mobile* untuk pengendalian dan pengawasan peternakan ayam broiler telah berhasil berfungsi sebagaimana mestinya. Aplikasi *mobile* telah berhasil terintegrasi dengan *database*, *web server* dan perangkat di kandang sehingga dapat menampilkan parameter – parameter kondisi kandang dan membantu dalam sistem pengendalian dan pengawasan.
2. Aplikasi *mobile* dapat digunakan untuk menghubungkan lebih dari satu pengguna dan perangkat. Segmentasi *database* pada Firebase cukup berhasil.
3. Dari pengujian fungsional aplikasi *mobile* dengan *white-box testing* didapatkan hasil bahwa aplikasi *mobile* dapat berjalan sesuai dengan algoritma dan *pseudocode* yang dirancang. Begitu pula pada pengujian *black-box* aplikasi *mobile* dapat berjalan sesuai dengan interface yang dirancang.
4. Pada pengujian non fungsional didapatkan hasil dari *compatibility testing*, aplikasi *mobile* dapat berjalan dengan baik di beberapa *smartphone* Android yakni Samsung SM-A507FN dan Haier Andromax Q dengan *range* versi antara Android 5.0 hingga Android 10. Hasil dari *user experience questionnaire* didapatkan hasil daya tarik sebesar 1.78, kejelasan 1.62, efisiensi 1.71, ketepatan 1.72, stimulasi 1.84 dan kebaruan 1.32. Sehingga dapat disimpulkan bahwa *user experience* dari aplikasi *mobile* ini sudah cukup baik.
5. Hasil pengujian *delay* pada proses *write data* ke Firebase menunjukan nilai rata - rata terendah adalah 275.8 ms, sedangkan untuk nilai rata-rata tertinggi adalah 375.2 ms. Sedangkan hasil pengukuran *delay* pada proses *read data* dari Firebase memiliki nilai rata-rata 625.6 ms, sedangkan untuk nilai rata-rata tertinggi adalah 684 ms.

### 5.2 Saran

Dari hasil penelitian Tugas Akhir ini, terdapat beberapa saran yang dapat dilakukan kedepannya terkait sistem yang dirancang yaitu:

1. Menambahkan fitur notifikasi saat terjadi perubahan kondisi parameter pada kandang.
2. Menambahkan fitur *live stream* CCTV untuk melengkapi sistem pemantauan sehingga dapat mengontrol hal yang tidak dapat dibaca oleh sensor pada perangkat di kandang.
3. Menambahkan fitur *multi-language* pada aplikasi *mobile* sehingga aplikasi *mobile* dapat dipahami banyak kalangan.
4. Menambahkan fitur *tutorial* pada awal pengguna mendaftar ke sistem aplikasi *mobile*. Sehingga memudahkan pengguna untuk mempelajari cara penggunaan aplikasi *mobile*.

### Daftar Pustaka

- [1] E. J. Christopher, dan H. Bagus, 28 Hari Panen Ayam Broiler. Jakarta: Agro Media Pustaka, 2011.
- [2] N. Maman, R. Aslila, dan Nasrullah, Statistik Peternakan Dan Kesehatan Hewan 2018. Jakarta: Direktorat Jenderal Peternakan dan Kesehatan Hewan, Kementerian Pertanian RI, 2018.
- [3] Fadilah, Roni, Beternak Ayam Broiler. Jakarta: Agro Media Pustaka, 2013.
- [4] R. Teuku, M. Syaryadhi, dan D. Rahmad, "Penerapan *Wireless Sensor Network* Berbasis *Internet of Things* Pada Kandang Ayam untuk Memantau dan Mengendalikan Operasional Peternakan Ayam," Seminar Nasional dan Expo Teknik Elektro 2017, hal. 81-88, Universitas Syiah Kuala, Banda Aceh, 2017.
- [5] N. M. Maharatih, I W. Sukanata, dan I. P. Astawa, "Analisis Performance Usaha Ternak Ayam Broiler Pada Model Kemitraan Dengan Sistem Open House," *Journal of Tropical Animal Science*, Universitas Udayana, Denpasar, 2017.
- [6] Direktorat Jendral Peternakan dan Kesehatan Hewan "Mengintip Performa Usaha Ayam Ras Pedaging," 25 Februari 2019 [Online]. Tersedia di: <http://pphna.ditjenpkh.pertanian.go.id/berita/pemasaran-hasil-peternakan/pphnaadmin/mengintip-performa-usaha-ayam-ras-pedaging/6716/>. [Diakses pada 23 Februari 2020].
- [7] G. Pradyumna, B. Omkhar, and B. Sagar, "Introduction to IoT," *International Advanced Research Journal in Science, Engineering and Technology*, Vol. 5, Issue 1, 2018.
- [8] Tutlane, "Android Introduction," 2018 [Online]. Tersedia di: <https://www.tutlane.com/tutorial/android/android-introduction>. [Diakses pada 27 Oktober 2019].
- [9] Natsir, Mohamad, "Pengembangan Prototype Sistem Kriptografi Untuk Enkripsi Dan Dekripsi Data Office Menggunakan Metode Blowfish Dengan Bahasa Pemrograman Java," *Jurnal Format* Vol. 6 no. 2, Universitas Mercu Buana, Jakarta, 2016.
- [10] H. Reshi, F. Taufik dan A. Ramzi, "Pengembangan Sistem Pemantauan Proses Backup Data Terdistribusi Menggunakan Owncloud Berbasis Web," *Jurnal Online Teknik Elektro*, Vol.3 No.3 2018: 11-19, 2018.
- [11] W. Martantio, P. K. Agi dan F. Lutfi, "Pengembangan Aplikasi Perangkat Bergerak Berbasis Android Untuk Survei Kepuasan Masyarakat Dengan Metode Mobile-D," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 3, No. 3, hlm. 2312-2319, 2019.
- [12] M. Schrepp, A. Hinderks and J. Thomaschewski, "Construction of a Benchmark for the User Experience Questionnaire (UEQ)," *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 4, N°4, University of Applied Sciences Emden/Leer, Germany, 2017.
- [13] R. Wulandari, "Analisis Quality of Service (QoS) Pada Jaringan Internet Studi Kasus : UPT Lokauji Teknik Penambangan Jampang Kulonprogo - LIPI," vol. 2, pp. 162-172, 2016.