

**PEMBANGUNAN INTERFACE DI SISI USER BERBASIS ANDROID  
PADA SISTEM *MONITORING* DAN *CONTROLLING* KUALITAS AIR  
PADA KOLAM HIAS IKAN KOI**

***ANDROID BASED INTERFACE DEVELOPMENT ON THE USER SIDE  
OF WATER QUALITY MONITORING AND CONTROLLING SYSTEM IN KOI FISH  
ORNAMENTAL POND***

Iqbal Arbai<sup>1</sup>, Basuki Rahmat<sup>2</sup>, Bagus Aditya<sup>3</sup>

<sup>1,2,3</sup> Universitas Telkom, Bandung

<sup>1</sup>iqbalarbai@student.telkomuniversity.ac.id, <sup>2</sup>basukirahmat@telkomuniversity.ac.id,

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

---

**Abstrak**

Kualitas air dalam suatu kolam ikan koi adalah salah satu faktor yang sangat penting, keterlambatan dalam mengetahui aspek-aspek kualitas air yang ideal akan berdampak negatif bagi kesehatan ikan koi. Tugas Akhir ini merupakan pembangunan sebuah sistem implementasi *Internet of Things* terkait sistem *monitoring* dan *controlling* kualitas air melalui aplikasi android. Dari keseluruhan sistem yang dibangun, penulis hanya berfokus pada perancangan, pembuatan dan pengujian aplikasi android. Kualitas aplikasi diuji menggunakan standar ISO/IEC 25010. Hasil pengujian menunjukkan bahwa aplikasi layak untuk digunakan, dengan persentase sebesar 100% pada aspek fungsional aplikasi dan aspek *compatibility*, 89,27% pada aspek *usability*. Pengujian aspek *performance efficiency* didapatkan hasil rata-rata pada dua aktivitas, yaitu *monitoring* dan *controlling* masing-masing sebesar CPU 4,5% dan 3% , *memory* sebesar 159,2 MB dan 132,2 MB, *frame rates* 4,3 *fps* dan 37,2 *fps*, dan mampu mengeksekusi sebuah thread dalam waktu 0,02 dan 0,03 detik. Kualitas performa jaringan diuji dengan standar ITU-T G.1010 dengan skenario uji antara broker MQTT dan aplikasi android pada dua aktivitas, yaitu *monitoring* dan *controlling*. Hasil rata-rata pada aktivitas *monitoring* dan *controlling* untuk *delay* adalah 155,69 ms dan 265,95 ms, *throughput* adalah 3185 bps dan 1185 bps, *packet loss* sebesar 0,01% dan 0,02%, *availability* dan *reliability* adalah 98,04% dan 98%.

**Kata Kunci:** *Internet of Things*, mikrokontroler, *monitoring*, *controlling*, MQTT, android

---

**Abstract**

Water quality in a koi pond is one of the very important factors, delays in knowing the ideal aspects of water quality will negatively affect the health of koi fish. This final task is the construction of an Internet of Things implementation system related to water quality monitoring and controlling systems through android applications. The quality of the application is tested using the ISO/IEC 25010 standard. Test results showed that the application was fit to use, with a percentage of 100% on the functional aspect and the compatibility aspect, 89.27% on the usability aspect. Performance efficiency aspect testing obtained average results on two activities, namely monitoring and controlling of 4.5% and 3% CPU, 159.2 MB and 132.2 MB, frame rates of 4.3 fps and 37.2 fps, respectively, and able to execute a thread in 0.02 and 0.03 seconds. Network performance quality was tested to the ITU-T G.1010 standard with a test scenario between the MQTT broker and android apps on two activities, namely monitoring and controlling. The average results on monitoring and controlling activities for delay were 155.69 ms and 265.95 ms, throughput was 3185 bps and 1185 bps, packet losses of 0.01% and 0.02%, availability and reliability were 98.04% and 98%.

**Keywords:** Internet of Things, microcontroller, monitoring, controlling, MQTT, android

---

## 1. Pendahuluan

### 1.1 Latar Belakang

Kualitas air dalam suatu kolam ikan hias, khususnya ikan koi adalah salah satu faktor yang sangat penting. Keterlambatan dalam mengetahui aspek-aspek kualitas air yang ideal akan berdampak negatif bagi kesehatan ikan koi bahkan dapat menyebabkan kematian massal. Salah satu cara untuk mengetahui dan mengontrol kualitas air secara cepat adalah dengan membangun sistem *monitoring* dan *controlling* kualitas air pada kolam hias ikan koi. Salah satunya adalah dengan memanfaatkan aplikasi android sebagai *user interface* sistem.

Terdapat penelitian terkait sistem yang akan dibangun pada Tugas Akhir ini yang telah dilakukan oleh beberapa peneliti. Pada jurnal [1], telah dibangun sistem yang memiliki fokus dalam melakukan *monitoring* dan *controlling* terhadap kadar amonia yang nantinya akan diakses oleh Raspberry dan Arduino yang akan mengeksekusi perintah yang diberikan lewat *smart phone*. Kemudian terdapat sebuah penelitian [2] yang berfokus untuk melakukan *monitoring* kadar pH dan suhu berbasis *Wireless Sensor Network* lalu sistem akan menampilkan hasil *monitoring* melalui tampilan web.

Oleh karena itu, pada penelitian ini dibangun sebuah sistem yang dapat melakukan aktivitas *monitoring* dan *controlling* kualitas air ikan koi dengan menggunakan aplikasi android. Keseluruhan sistem dibangun dengan unit pengatur berupa *microwifi* NodeMCU yang terhubung dengan beberapa sensor. *Device* pengontrol berupa *electronic relay* dan mosfet transistor. Sisi *connectivity* sistem menggunakan tipe komunikasi modem *wifi* yang terhubung dengan *broker* MQTT sebagai penghubung antara sistem mikrokontroler dan aplikasi android. Fokus penulis pada Tugas Akhir ini adalah terkait perancangan, pembuatan, dan pengujian *user interface* berupa aplikasi android.

Ditahap perancangan, penulis membuat *Unified Modeling Language* (UML) berupa *Use Case Diagram* untuk merepresentasikan interaksi antara pengguna dan aplikasi. Kemudian aplikasi akan dirancang menggunakan Android Studio. Pengujian kualitas aplikasi menggunakan standar ISO/IEC 25010 berdasarkan aspek fungsional aplikasi, *performance efficiency*, dan *compatibility*. Pengujian *performance efficiency* dilakukan menggunakan aplikasi desktop Apptim. Pengujian *compatibility* dilakukan menggunakan Firebase Test Lab dengan metode Robo Test. Kemudian pengukuran performansi jaringan berdasarkan pengujian *delay*, *throughput*, *packet loss*, *availability* dan *reliability* akan dilakukan menggunakan Wireshark. Dengan adanya aplikasi ini, aktivitas *monitoring* dan *controlling* diharapkan menjadi lebih efektif dan efisien.

---

## 2. Dasar Teori

### 2.1 Kualitas Air Kolam Ikan Koi

Kualitas air menjadi salah satu faktor yang sangat penting dan mutlak bagi keberlangsungan kegiatan budidaya ikan hias koi. Oleh karena itu, baik kualitas maupun kuantitas air harus benar-benar sesuai dengan kebutuhan hidup ikan. Untuk menunjang kehidupan yang baik untuk ikan dibutuhkan air yang bersih serta tidak mengandung bahan pencemar [3].

### 2.2 Internet of Things

Secara umum konsep *Internet of Things* (IoT) dapat diartikan sebagai kemampuan untuk menghubungkan objek-objek cerdas sehingga dapat saling berinteraksi antar sesama objek maupun berinteraksi dengan lingkungan serta peralatan komputasi cerdas lainnya melalui jaringan internet [4]. Konsep IoT memungkinkan sebuah benda diberi tanda pengenal berupa IP address sehingga dapat saling berkoumunikasi melalui jaringan internet dengan perangkat lain yang juga memiliki tanda pengenal atau IP address [5].

### 2.3 Java

Java adalah bahasa pemrograman yang banyak digunakan pada pembuatan *software*, java mampu membuat sebuah *software* dapat berjalan diberbagai *operation system*, seperti windows, linux hingga *smartphone*. Java secara luas dapat dimanfaatkan dalam pengembangan berbagai jenis *software* aplikasi atau pun aplikasi berbasis web [6].

## 2.4 Android Studio

Android Studio adalah salah satu IDE untuk pengembangan aplikasi android, yang didasarkan pada IntelliJ IDEA. Selain sebagai *code editor*, Android Studio juga menawarkan banyak fitur yang dapat meningkatkan produktivitas seorang developer dalam mengembangkan aplikasi android [7].

## 2.5 Message Queuing Telemetry Transport (MQTT)

*Message Queuing Telemetry Transport* (MQTT) adalah sebuah protokol yang berjalan diatas *stack* TCP/IP. MQTT menggunakan model *publish / subscribe* yang ringan dengan pengiriman pesan dua arah yang andal. MQTT diciptakan oleh Andy Stanford-Clark (IBM) dan Arlen Nipper pada tahun 1999. MQTT didesain untuk mampu menangani volume data yang tinggi di jaringan *bandwith* yang rendah [8].

## 2.6 Unified Modelling Language

*Unified Modelling Language* (UML) adalah sebuah pemodelan yang biasa digunakan untuk pengembangan pemrograman berorientasi objek dengan tujuan guna melakukan pendefinisian kebutuhan, perancangan desain, dan pembuatan arsitektur dari aplikasi [9].

## 2.7 ISO/IEC 25010

ISO/IEC adalah salah satu standar pengujian aplikasi atau *software* yang dibuat oleh *International Organization for Standardization* (ISO) dan *International Electrotechnical Commission* (IEC) [10]. Standar pengujian ISO/IEC 25010 dianggap sebagai salah satu standar yang relevan terhadap perubahan teknologi dan informasi pada saat ini. Model ISO 25010 ini memiliki delapan karakteristik yaitu *functional suitability, performance efficiency, compatibilty, usability, reliability, security, maintainability, dan portability*.

## 2.8 USE Quistionnaire

Salah satu paket kuisioner yang dapat digunakan untuk mengukur *usability* adalah USE Quistionnaire. USE Quistionnaire terdiri atas beberapa variabel, yaitu *usefulness, ease of use, ease of learning, dan satisfication* [11].

## 2.9 Skala Likert

Skala Likert merupakan skala yang digunakan untuk mengukur pendapat responden pada suatu penelitian. Skala Likert menunjukkan tingkat setuju dan tidak setuju dengan berbagai pernyataan tentang beberapa sikap, objek, orang atau peristiwa. Peserta dalam survei diminta untuk menyatakan tingkat persetujuan mereka dengan pernyataan yang diberikan dari sangat setuju hingga sangat tidak setuju [12].

## 2.10 Kualitas Jaringan

Standar yang digunakan dalam menentukan baik buruknya kualitas jaringan pada penelitian kali ini adalah Standar ITU-T G.1010 terkait *Quality of Service* (QoS) untuk *end-to-end user*.

**Tabel 2. 1** Spesifikasi QoS Standar ITU-T G.1010 [13].

No	Medium	Application	Key performance parameters and target values		
			Delay	Delay Variation	Information Loss
1	Data	Bulk data transfer/retrieval	Preferred < 15 s Acceptable < 60 s	N.A.	Zero
2	Data	Command/control	< 250 ms	N.A.	Zero

Contoh parameter QoS adalah sebagai berikut:

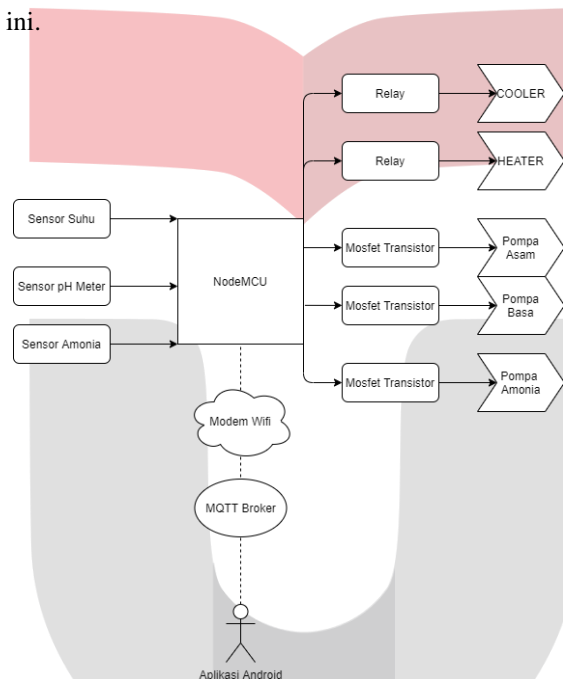
- Waktu yang dibutuhkan oleh data pada suatu jaringan untuk menempuh jarak dari asal ke tujuan disebut dengan *delay*. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama [14]. Satuan dari *delay* adalah *millisecond* (ms).
- Throughput*, yaitu kecepatan (rate) transfer data efektif, yang diukur dalam *bit per second* (bps). *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [14].

- c) *Availability* dan *reliability* diperlukan untuk mengetahui tingkat keberhasilan suatu sistem yang dibangun. *Reliability* merupakan kemungkinan suatu sistem terbebas dari suatu kegagalan pada saat menjalankan fungsinya. *Availability* adalah kemampuan suatu sistem agar selalu tersedia ketika sistem tersebut ingin digunakan [15].
- d) *Packet Loss*, merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena tabrakan (*collision*) dan kemacetan (*congestion*) pada jaringan dan hal ini berpengaruh pada semua aplikasi karena *retransmission* akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi aplikasi tersebut [14].

### 3. Perancangan dan Simulasi Sistem

#### 3.1 Desain Model Sistem

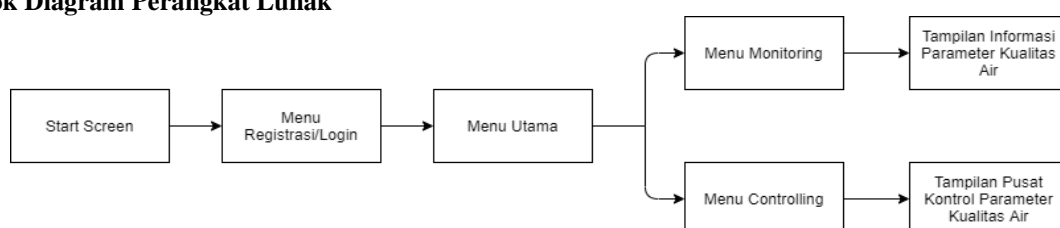
Rancangan desain model sistem *monitoring* dan *controlling* kolam ikan hias koi berbasis konsep IoT yang akan dibangun pada tugas akhir ini.



Gambar 3. 1 Desain Sistem.

Secara umum sistem terdiri atas 3 bagian atau area. Area pertama yaitu area sistem mikrokontroler yang terdiri atas *microwifi* NodeMCU sebagai otak pengatur sistem. *Microwifi* NodeMCU ini terhubung dengan bagian sensor *node*. Sisi lain *microwifi* NodeMCU ini terhubung dengan beberapa relay dan mosfet transistor yang terhubung dengan pompa-pompa untuk penstabilan nilai parameter kualitas air agar selalu berstatus normal. Area kedua berupa konektivitas sistem, di mana *microwifi* NodeMCU akan memanfaatkan sistem komunikasi modem *wifi* untuk mendapatkan koneksi internet. Setelah berhasil terhubung ke internet, data akan diteruskan ke broker MQTT yang bekerja sebagai penghubung antara sistem mikrokontroler dengan *user* interface yang berupa aplikasi android. Area ketiga ialah sisi *user* di mana *user* mampu melakukan *monitoring* dan *controlling* terhadap parameter-parameter yang ada pada sistem melalui *interface* aplikasi android.

#### 3.2 Blok Diagram Perangkat Lunak

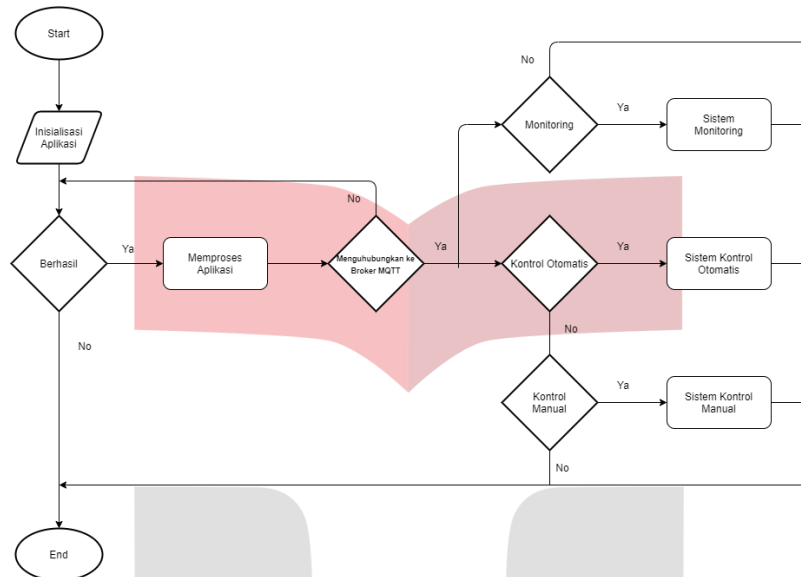


Gambar 3. 2 Blok Diagram Aplikasi.

Aplikasi Android yang dirancang akan menyajikan beberapa menu pada *user interface* (UI). Tampilan pertama berupa UI *start screen*, kemudian setelah itu *user* perlu melakukan registrasi sebelum bisa masuk atau *login* ke dalam aplikasi. Menu utama akan langsung muncul ketika *user* berhasil login, pada menu utama ini, *user* dapat memilih untuk masuk ke menu *monitoring* ataupun menu *controlling*. Baik menu *monitoring* maupun menu *controlling* memiliki tampilan dan fungsinya masing-masing.

### 3.3 Flowchart Sistem Aplikasi

Desain *flowchart* sistem aplikasi yang akan menampilkan proses-proses utama pada penelitian kali ini.

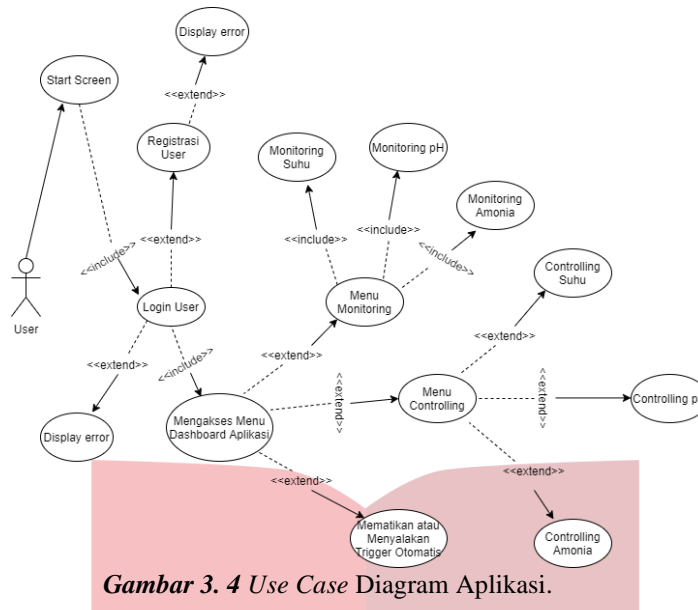


**Gambar 3.3** Flowchart Sistem Aplikasi.

Dimulai dengan inisialisasi aplikasi, ketika proses inisialisasi berhasil, maka aplikasi akan diproses atau terbuka. Setelah pemrosesan aplikasi sudah berjalan, maka sistem akan langsung mencoba melakukan interkoneksi dengan broker MQTT, jika proses interkoneksi ini gagal, maka sistem akan mencoba kembali membangun interkoneksi dengan broker MQTT. Adapun jika proses interkoneksi sudah berhasil, maka sistem menyediakan tiga buah fitur, yaitu *monitoring*, kontrol otomatis, dan kontrol manual.

### 3.4 Pemodelan Use Case Diagram

*Use case* diagram akan menjelaskan mengenai fungsionalitas yang perlu ada dalam sebuah sistem dan menjelaskan hubungan antara sistem dan ekosistem di dalam model program yang dibangun.



Gambar 3. 4 Use Case Diagram Aplikasi.

User akan masuk pada start screen saat pertama kali membuka aplikasi, kemudian setelah memulai start screen, user akan diarahkan pada menu register dan login. Setelah berhasil melakukan proses registrasi dan login, user akan langsung diarahkan dashboard menu utama pada aplikasi. Dashboard utama ini akan menampilkan tiga opsi menu selanjutnya yang dapat dipilih oleh user, opsi pertama yaitu menuju menu monitoring, opsi kedua menuju menu controlling, dan opsi ketiga untuk mematikan trigger otomatis pada sistem. Baik menu monitoring dan controlling memiliki fungsinya masing-masing, dimana menu monitoring akan menyediakan halaman monitoring seluruh parameter kualitas air kolam pada user. Menu controlling akan menyediakan beberapa button kepada user untuk melakukan kontrol sesuai dengan kebutuhan.

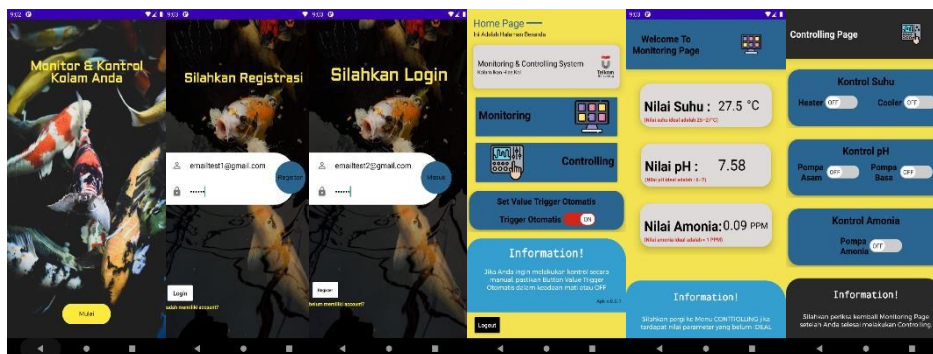
3.5 Konsep Desain Pengolahan Data

Pada sistem atau aplikasi yang akan dirancang pada tugas akhir ini pengolahan data sensor akan dilakukan oleh MQTT broker sebagai pengganti server penghubung antar elemen di dalam sistem. Aplikasi akan mengacu atau melakukan subscribe terhadap broker MQTT untuk mendapatkan data sensor. Proses publish juga dilakukan oleh aplikasi terhadap broker MQTT untuk melakukan command atau perintah controlling.

4. Hasil Pengujian dan Analisis

4.1 Pengujian Fungsional Aplikasi

Pengujian ini dilakukan untuk mengetahui bagaimana performa fungsional aplikasi Koi Fish App. Pengujian ini hanya melihat kesesuaian antara output activity yang dihasilkan aplikasi dengan yang output activity yang diharapkan.



Gambar 4. 1 Tampilan Graphical User Interface Aplikasi.

Seluruh pengujian fungsioanl telah dilakukan dan menghasilkan.

Tabel 4. 1 Konfirmasi Output Activity.

No	Desain GUI	Konfirmasi	
		Sukses	Gagal
1	Start Screen	Sukses	
2	Halaman Registrasi	Sukses	
3	Halaman Login	Sukses	
4	Menu Utama	Sukses	
5	Menu Monitoring	Sukses	
6	Menu Controlling	Sukses	

#### 4.2 Pengujian Performance Efficiency

Aspek pengujian meliputi penggunaan *Central Processing Unit* (CPU), *memory*, *threads counts* dan *Frames per second* (FPS). Pengujian dilakukan dengan menggunakan aplikasi desktop bernama Apptim dan menggunakan *physical device* Android tipe Samsung SM-J530Y dengan API level 28 atau Android *version* 9.

**Tabel 4.2** Hasil Pengujian *Performance Efficiency*.

Skenario	CPU Usage (%)	Memory Usage (MB)	Frame Rates (fps)	Threads Counts (second / thread)
Monitoring	4,5	159,2	4,3	0,02
Controlling	3	132,2	37,2	0,03

Pengujian terdiri atas dua buah skenario pengukuran. Skenario pertama ketika aplikasi melakukan aktivitas *monitoring*, skenario kedua ketika aplikasi melakukan aktivitas *controlling*. Pengujian dilakukan sebanyak 10 kali pada setiap skenario dengan durasi pengamatan sekitar 1 menit pada masing-masing pengujian di kedua skenario.

#### 4.3 Pengujian Usability

Pengujian *usability* menggunakan skala Likert dalam instrumen USE Questionnaire. Kuisisioner tersebut diajukan kepada 11 responden yang seluruhnya terdiri atas mahasiswa, untuk merasakan *experience* dalam menggunakan aplikasi Koi Fish App. Berikut adalah hasil pengujian *usability*.

$$\text{Nilai} = \frac{((120 \times 5) + (83 \times 4) + (16 \times 3) + (1 \times 2) + (0 \times 1))}{(11 \times 20 \times 5)} \times 100\% \quad (4.1)$$

$$\text{Nilai} = \frac{(600 + 332 + 48 + 2 + 0)}{1100} \times 100\%$$

$$\text{Nilai} = \frac{982}{1100} \times 100\% = 89,27\%$$

Persentase nilai pada aspek *usability* adalah 89,27%, terkait kriteria skor pengujian *usability*, dapat disimpulkan bahwa aplikasi android Koi Fish App sangat layak untuk digunakan.

#### 4.4 Pengujian Compatibility

Pengujian *compatibility* memiliki dua jenis pengujian, yaitu *coexistence* dan *interoperability*.

##### a. Coexistence

Pengujian ini dilakukan untuk mengetahui apakah aplikasi Koi Fish App dapat dijalankan bersamaan dengan aplikasi lainnya. Berikut hasil pengujian *coexistence*.

**Tabel 4.3** Hasil Pengujian *Coexistence*.

No	Test Case	Berhasil	Gagal
1	Membuka aplikasi Line	1	0
2	Membuka aplikasi Twitter	1	0

3	Membuka aplikasi Instagram	1	0
4	Membuka aplikasi YouTube	1	0
5	Membuka aplikasi Shopee	1	0
6	Membuka aplikasi Telegram	1	0
7	Membuka aplikasi Gojek	1	0
8	Membuka aplikasi WhatsApp	1	0
9	Membuka aplikasi Tokopedia	1	0
Total		9	0

Presentase kelayakan dari pengujian *coexistence* dapat dilihat pada perhitungan berikut.

$$\text{Presentase Kelayakan} = \frac{9}{9} \times 100\% = 100\% \quad (4.2)$$

#### b. *Interoperability*

Pengujian *interoperability* dilakukan dengan tujuan untuk mengetahui apakah aplikasi dapat digunakan oleh berbagai tipe perangkat Android yang berbeda. Pengujian dilakukan dengan memanfaatkan salah satu fitur yang ada pada *firebase* yaitu *Firebase Test Lab* berbasis *Robo Test*.

**Tabel 4. 4** Hasil Pengujian *Interoperability*.

No	Nama Device	Tipe Device	API	Hasil
1	Pixel 5e	<i>Physical</i>	30	Berhasil
2	Pixel 4	<i>Physical</i>	30	Berhasil
3	SM-A105FN	<i>Physical</i>	29	Berhasil
4	Pixel 3	<i>Physical</i>	28	Berhasil
5	Redmi Note 6 Pro	<i>Physical</i>	28	Berhasil
6	Pixel 2	Virtual	30	Berhasil
7	Pixel 2	Virtual	29	Berhasil
8	Pixel 2	Virtual	28	Berhasil
9	Low-resolution MDPI phone	Virtual	30	Berhasil
10	Low-resolution MDPI phone	Virtual	29	Berhasil
11	Low-resolution MDPI phone	Virtual	28	Berhasil

Presentase kelayakan dari pengujian *interoperability* dapat dilihat pada perhitungan berikut.

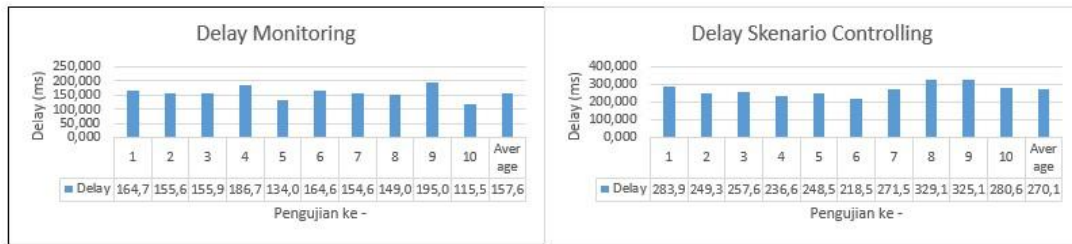
$$\text{Presentase Kelayakan} = \frac{11}{11} \times 100\% = 100\% \quad (4.3)$$

Berdasarkan hasil perhitungan presentase kelayakan dapat dikatakan aplikasi telah memenuhi standar pada kedua sub pengujian *compatibility*.

#### 4.5 Pengujian *Delay*

Pengamatan yang dilakukan pada pengujian *delay* terdiri atas dua buah skenario pengamatan. Skenario pertama ketika aplikasi melakukan proses atau aktivitas *monitoring*, skenario kedua ketika aplikasi melakukan aktivitas *contolling*. Waktu pengamatan dilakukan pada pukul 15.00 – 15.30 sebanyak 10 kali pengamatan pada

masing-masing scenario. Proses pengamatan memanfaatkan jaringan *wifi Indihome* dengan menggunakan *device* virtual android atau simulator android.

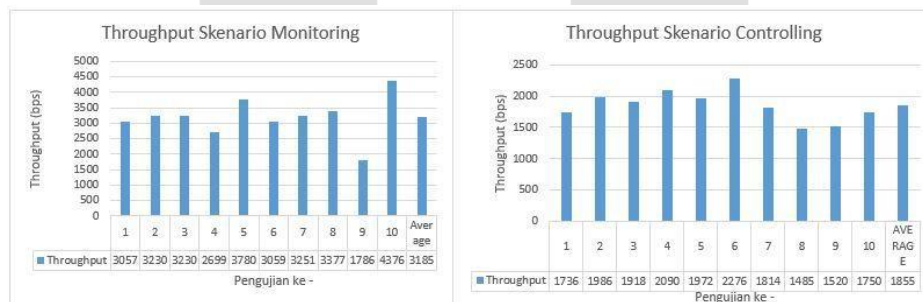


**Gambar 4. 2** Hasil Pengujian *Delay*.

Skenario *monitoring* memiliki nilai rata-rata *delay* sebesar 157,61 ms, nilai tersebut jika mengacu pada Standar ITU-T G.1010 mengenai *bulk data transfer / retrieval* merupakan hasil yang *preferred* atau dianggap sangat memuaskan. Skenario *controlling* memiliki nilai rata-rata *delay* sebesar 270,13 ms, nilai tersebut jika mengacu pada Standar ITU-T G.1010 mengenai *application command/control* merupakan hasil yang dianggap belum mencapai target *value delay* yang diharapkan, karena target *value delay* pada kategori tersebut adalah kurang dari 250 ms.

#### 4.6 Pengujian *Throughput*

Skenario pengamatan yang dilakukan pada pengujian *throughput* sama seperti skenario pada pengujian *delay*.

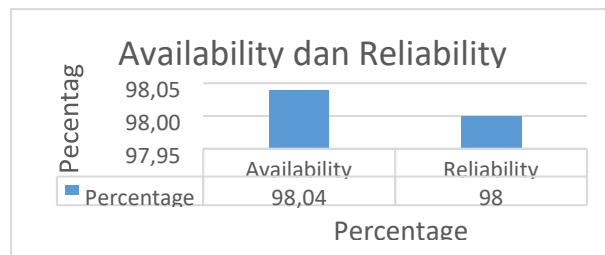


**Gambar 4. 3** Hasil Pengujian *Throughput*.

*Throughput* rata-rata yang dihasilkan melalui 10 kali pengamatan pada skenario *monitoring* dan *controlling* adalah 3185 bps dan 1855 bps. Jumlah *throughput* yang dihasilkan pada proses *monitoring* lebih besar dibandingkan yang dihasilkan oleh aktivitas *controlling*. Hal itu disebabkan karena aktivitas *monitoring* dilakukan secara terus menerus sehingga paket atau *traffic* yang mengalir selama aktivitas *monitoring* tentunya lebih banyak.

#### 4.7 Pengujian *Availability* dan *Reliability*

Skenario pengamatan terhadap sistem aplikasi dengan menggunakan *virtual device* android atau android simulator, durasi pengamatan dilakukan selama 5 jam yang dibagi kepada 2 aktivitas, yaitu aktivitas *monitoring* selama 2 jam 30 menit dan aktivitas *controlling* selama 2 jam 30 menit. *Downtime* yang terjadi selama pengamatan adalah kurang lebih 6 menit karena pada saat proses pengamatan pada aktivitas *monitoring* aplikasi sedikit mengalami *crash* yang memerlukan aplikasi di *running* ulang melalui Android Studio.

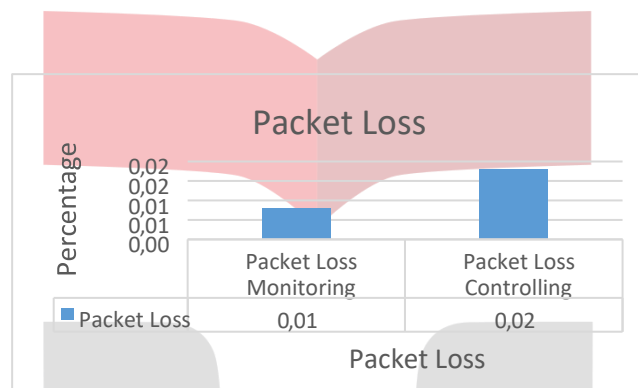


**Gambar 4. 4** Hasil Pengujian *Availability* dan *Reliability*.

Menurut hasil tersebut sistem aplikasi yang dibangun dikatakan telah layak karena sudah mendekati 100%.

#### 4.8 Pengujian *Packet Loss*

Skenario pengamatan yang dilakukan pada pengujian *packet loss* sama seperti skenario pada pengujian *availability* dan *reliability*.



**Gambar 4. 5** Hasil Pengujian *Packet Loss*.

Selama durasi pengamatan, aktivitas *monitoring* berhasil mengirim paket data sejumlah 37758 paket dan mengalami *packet loss* sejumlah 3 paket, adapun aktivitas *controlling* menerima paket data sejumlah 5518 paket dan mengalami *packet loss* sebanyak 1 paket

## 5. Simpulan dan Saran

### 5.1 Simpulan

Berdasarkan perancangan dan pengujian yang telah dilakukan dalam pengembangan aplikasi Android monitoring Trans Metro Bandung dapat disimpulkan bahwa:

1. Pada Tugas Akhir ini berhasil dibangun sebuah aplikasi berbasis android yang dapat melakukan aktivitas *monitoring* dan *controlling* kualitas air kolam ikan koi dari mana saja dan kapan saja secara *realtime*. Konektivitas komunikasi aplikasi dilakukan terhadap broker MQTT. Aplikasi mampu melakukan kontrol terhadap sensor melalui proses *publish* ke broker MQTT, serta aplikasi mampu menerima data sensor melalui proses *subscribe* ke broker MQTT.
2. Hasil pengujian kualitas aplikasi android berdasarkan standar ISO/IEC 25010 mengindikasikan bahwa aplikasi layak untuk digunakan, dengan presentase aspek fungsional aplikasi sebesar 100%, aspek *performance efficiency* pada masing-masing aktivitas *monitoring* dan *controlling* didapatkan rata-rata konsumsi CPU sebesar 4,5% dan 3%, konsumsi *memory* sebesar 159,2 MB dan 132,2 MB, *frame rates* sebesar 4,3 *fps* dan 37,2 *fps*, dan mampu mengeksekusi sebuah thread dalam waktu 0,02 detik dan 0,03 detik. Aspek *usability* menghasilkan presentase 89,27%. Aspek *comptibility* menghasilkan presentase masing-masing 100% pada setiap sub aspek nya.
3. Pengujian peformansi jaringan dilakukan berdasarkan standar ITU-T G.1010 dengan skenario uji antara broker MQTT dengan aplikasi android pada dua aktivitas yaitu *monitoring* dan *controlling*, pengukuran menggunakan aplikasi *wireshark*. Hasil rata-rata pada aktivitas *monitoring* dan *controlling* untuk *delay* adalah 157,61 ms dan 270,13 ms, *throughput* adalah 3185 bps dan 1185 bps, *packet loss* sebesar 0,01% dan 0,02%. Adapun *Availability* dan *reliability* adalah 98,04% dan 98% dengan durasi pengamatan selama 5 jam.

## 5.2 Saran

Dari pengembangan dan pengujian yang telah dilakukan, penulis memiliki saran agar penelitian ini dapat berkembang menjadi lebih baik lagi. Berikut merupakan beberapa saran yaitu:

1. Meningkatkan kualitas aplikasi dari sisi *frame rates* sehingga aplikasi dapat digunakan dengan lebih *smooth* dan nyaman.
2. Melakukan pengujian performa jaringan pada kondisi jaringan yang lebih stabil agar memperoleh nilai parameter kualitas jaringan yang lebih memuaskan.
3. Menambahakan fitur *history* sehingga aplikasi mampu menampilkan riwayat data sensor.
4. Menambahakan fitur *trigger* notifikasi untuk memberikan peringatan lebih dini kepada *user* ketika sedang tidak membuka aplikasi.
5. Mengembangkan aplikasi agar dapat menjangkau platform iOS.

---

## Referensi

- [1] M. A. Nugroho and M. Rivai, "Sistem Kontrol dan Monitoring Kadar Amonia untuk Budidaya Ikan yang Diimplementasi pada Raspberry Pi 3B," *J. Tek. ITS*, vol. 7, no. 2, pp. A374–A379, 2019.
- [2] A. Fadillah, A. T. Hanuranto, and N. Bogi, "Implementasi Sistem Monitoring Kualitas Air Kolam Ikan Lele Berbasis Wireless Sensor Network Implementation of Monitoring System in Catfish Pond Based on Wireless Sensor Network," in *e-Proceeding of Engineering*, 2019, vol. 6, no. 2, pp. 4084–4090.
- [3] D. Satyani and B. Priono, "PENGUNAAN BERBAGAI WADAH UNTUK PEMBUDIDAYAAN IKAN HIAS AIR TAWAR," *Media Akuakultur*, 2012, doi: 10.15578/ma.7.1.2012.14-19.
- [4] E. D. Meutia, "Internet of Things – Keamanan dan Privasi," *Semin. Nas. dan Expo Tek. Elektro 2015*, pp. 85–89, 2015.
- [5] R. S. Utama, R. Mayasari, and M. A. Murti, "Implementasi Sistem Pemantauan Dan Pengontrolan Perangkat Listrik Otomatis Menggunakan Aplikasi Android Berbasis Internet of Things," in *e-Proceeding of Engineering*, 2018, vol. 5, no. 3, pp. 5012–2019, [Online]. Available: <https://openlibrary.telkomuniversity.ac.id/pustaka/144903/implementasi-sistempemantauan-dan-pengontrolan-perangkat-listrik-otomatis-menggunakan-aplikasi-androidberbasis-internet-of-things.html>.
- [6] A. Fatoni *et al.*, *Java Fundamental Dengan Netbeans 8.02*. 2018.
- [7] A. Developers, "Meet Android Studio," *Android Developers*. <https://developer.android.com/studio/intro> (accessed Nov. 24, 2020).
- [8] F. Quintero, "MQTT – IoT Messaging Protocol Francisco Quintero Lead Firmware Engineer."
- [9] and M. S. Rosa A.S, *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Obyek)*, vol. 53, no. 1. Bandung, 2011.
- [10] S. Wagner, *Software product quality control*. Springer Berlin Heidelberg, 2013.
- [11] G. I. Marthasari and N. Hayatin, "Analisis Usability Terhadap Sistem Lective Gegulang Berbasis USE Qestionnaire," *J. Semin. Nas. Teknol. dan Rekayasa*, pp. 1–8, 2017, [Online]. Available: <http://research-report.umm.ac.id/index.php/sentra/article/view/1458>.
- [12] H. Taherdoost, "What Is the Best Response Scale for Survey and Questionnaire Design; Review of Different Lengths of Rating Scale / Attitude Scale / Likert Scale," *Int. J. Acad. Res. Manag.*, vol. 8, no. 1, pp. 1–10, 2019, [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02557308>
- [13] International Telecommunication Union, "ITU-T Recommendation G. 1010: End-user multimedia QoS categories (Quality of service and performance)," *Int. Telecommun. Union*, vol. 1010, 2001.
- [14] Y. A. Pranata, I. Fibriani, and S. B. Utomo, "ANALISIS OPTIMASI KINERJA QUALITY OF SERVICE PADA LAYANAN KOMUNIKASI DATA MENGGUNAKAN NS-2 DI PT. PLN (PERSERO) JEMBER," *SINERGI*, 2016, doi: 10.22441/sinergi.2016.2.009.
- [15] D. J. Trujillo and C. J. B. Scharmer, "Reliability , Availability , and Maintainability Considerations in the Design and Evaluation of Physical Security Systems," pp. 1–12, 2012.