Aplikasi Monitoring Alat Pengering Maggot berbasis IoT dengan Database Firebase

1st Fadhillah Padmanaba Fauzi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
padmanaba@student.telkomuniversity.ac.id

2nd Sofia Naning Hertiana
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sofiananing@telkomuniveristy.ac.id

3rd Iman Hedi Santoso
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
imanhedis@telkomuniversity.ac.id

***Abstrak* — Maggot adalah sumber pakan yang sangat bagus bagi hewan ternak karena memiliki protein tinggi. Maggot mentah memiliki kekurangan yaitu masa simpan yang relatif singkat, oleh karena itu biasanya maggot diolah menjadi maggot kering untuk mempertahankan masa simpannya. Saat ini proses pengeringan maggot masih konvensional dengan alat manual yang memliki kekurangan yaitu memerlukan pengawasan konstan selama proses pengeringaan untuk memastikan bahwa maggot sudah kering dan maggot kering yang dihasilkan juga tidak konsisten, sehingga membuat proses pengeringan maggot dengan alat manual menjadi kurang efisien dari segi waktu dan kualitas maggot yang dihasilkan. Oleh karena itu pada penelitian ini penulis membuat alat pengering maggot dengan teknologi *Internet of Things* dan membuat aplikasi monitoring yang dapat mengoperasikan dan memantau proses pengeringan maggot. Pembuatan aplikasi monitoring menggunakan platform Android Studio dan menggunakan Firebase sebagai *realtime database*. Aplikasi *monitoring* yang dibuat memiliki beberapa fitur yaitu *monitoring* suhu *realtime*, pilihan berat maggot yang akan dikeringkan dengan waktu estimasi setiap proses pengeringan dilakukan dan fitur tombol *stop* sebagai tombol *emergency* jika ingin menghentikan proses pengeringan maggot sebelum waktu estimasi yang diberikan telah habis.**

***Kata kunci*— Maggot, *Internet of Things*, Aplikasi Monitoring, Firebase, Realtime Database, Android Studio**

I. PENDAHULUAN

Pada era modern ini perkembangan teknologi sangat pesat dan sangat membantu manusia untuk menyelesaikan permasalahan yang ada. Hal ini menjadi inspirasi dalam mengimplementasikan teknologi dalam budidaya maggot dengan menerapkan teknologi *Internet of Things.* Maggot merupakan larva dari lalat *Black Soldier Fly* (BSF) yang merupakan sumber pakan hewan ternak yang memiliki protein tinggi.

Budidaya maggot saat ini sedang berkembang karena banyak peternak membutuhkan maggot sebagai sumber pakan hewan ternak mereka. Pakan ternak berupa maggot mentah memiliki kekurangan yaitu masa simpan yang singkat karena kadar air yang tinggi dan membuat kualitas maggot menjadi turun. Oleh karena itu maggot akan dikeringkan melalui proses pengeringan untuk memperpanjang masa simpan maggot. Alat pengering maggot yang ada saat ini masih bersifat manual dengan mengandalkan pengawasan selama proses pengeringan yang membuat proses pengeringan menjadi tidak efektif. Proses pengeringan yang dilakukan secara manual memiliki kekurangan yaitu waktu yang terbuang karena harus selalu memantau proses pengeringan dan juga maggot kering yang dihasilkan juga tidak konsisten kualitasnya.

Pada *capstone design* ini , penulis mengembangkan alat pengering maggot dengan teknologi Internet of Things dan aplikasi monitoring sebagai media untuk kontrol dan *monitoring* alat pengering maggot. Aplikasi ini dibuat dengan menggunakan platform Android Studio dan Firebase sebagai *database*. Sistem kerjanya yaitu data dari sensor dan beberapa komponen akan secara realtime mengirimkan data ke *database* yang kemudian data ini akan diolah dan dipakai oleh aplikasi monitoring untuk mengontrol alat pengeringan maggot selama proses pengeringan berlangsung. Melalui aplikasi *monitoring*, peternak maggot dapat mengoperasikan alat pengering maggot tanpa perlu pengawasan secara konstan sehingga lebih efisien dalam segi waktu. Dengan pengembangan ini juga, maggot kering yang dihasilkan memiliki kualitas yang lebih baik dan konsisten.

II. KAJIAN TEORI

A. Android Studio

Android Studio adalah aplikasi pengembangan yang dapat digunakan untuk membuat aplikasi mobile pada platform android dan dapat melakukan pengujian pada aplikasi secara virtual tanpa perlu perangkat keras asli[1]. Android studio memiliki *library module* yang memungkinkan pengguna untuk meggunakan *library* yang sudah ada untuk digunakan kembali dalam pembuatan proyek yang lain. Android studio merupakan *platform* yang mendukung pembuatan aplikasi android yang memungkinkan pengguna untuk melakukan konfigurasi *source code* dalam proses kompilasi dengan fleksibilitas dapat menggunakan kode yang sama secara berulang-ulang tanpa perlu menulis ulang kode dengan memanfaatkan struktur *gradle* dan tidak mengubah file aslinya.

B. *Internet of Things*

Internet of Things adalah sebuah konsep menghubungkan semua perangkat seperti sensor ke internet dengan tujuan untuk memudahkan pekerjaan manusia karena perangkat dapat melakukan kerja secara otomatis [2]. Cara kerja *Internet of Things* yaitu menghubungkan perangkat keras dan *software* ke jaringan internet. Prosesnya adalah sensor akan mengumpulkan data yang kemudian data tersebut akan dikirimkan ke *database*, *database* akan menyimpan dan mengolah data untuk membuat sebuah keputusan yang cerdas.

C. *Qualitiy of Service*

Qualitiy of Service adalah tolak ukur seberapa bagus kemampuan jaringan dapat memenuhi kebutuhan aplikasi yang dibuat. Berikut adalah *Quality of Service* yang digunakan pada penelitian kali ini:

1. *Delay*

Delay merupakan tambahan waktu yang diperlukan saat proses pengiriman dan penerimaan data dari sumber ke penerima dan merupakan parameter keberhasilan saat proses pengiriman dan penerimaan data[3]. Berikut adalah rumus perhitungan untuk mendapatkan *delay*:

$$Delay = \frac{Total\ Delay}{Jumlah\ Data} \quad (1)$$

Pada perhitungan *delay*, penulis menggunakan aplikasi wireshark untuk *capture* paket data selama proses transmisi data dari ESP8266 ke *database* dan aplikasi.

2. *Throughput*

Throughput adalah parameter keberhasilan suatu sistem yang mengacu pada kemampuan suatu sistem untuk memproses seberapa banyak data dalam waktu yang efektif dengan ukuran tertentu[4]. Berikut adalah rumus perhitungan untuk mendapatkan *throughput*:

$$Throughput = \frac{Jumlah\ Data\ Diterma}{Waktu\ Data\ Terkirim} \quad (2)$$

Pada perhitungan *throughput*, penulis menggunakan aplikasi wireshark untuk *capture* paket data selama proses transmisi data dari ESP8266 ke *database* dan aplikasi.

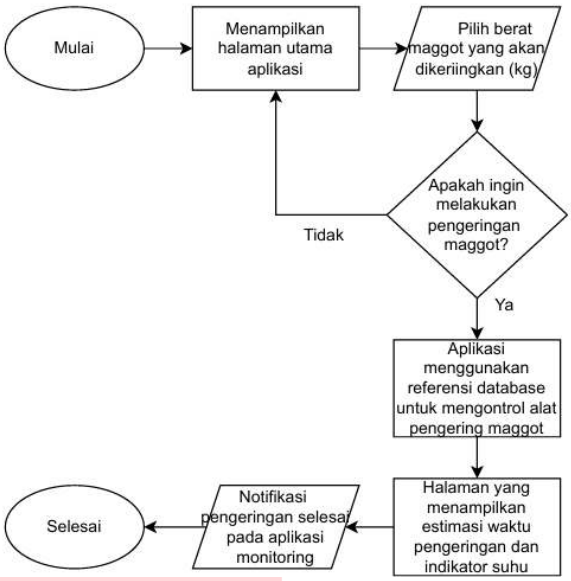
D. Firebase *Database*

Firebase merupakan platform yang menyediakan layanan *database* yang disimpan pada *cloud* dan dapat disinkronisasi secara *realtime* ke sensor atau komponen yang terhubung melalui jaringan internet. Firebase akan melakukan update dan sinkronisasi data secara *realtime* apabila terdapat perubahan data[5]. Firebase memiliki kelebihan yaitu dapat diakses secara realtime dimanapun menggunaka perangkat *mobile* dan web tanpa perlu adanya server application[6].

III. METODE

A. Rancangan Penelitian

Pada rancangan penelitian, penulis menjabarkan cara kerja dari aplikasi *monitoring* untuk alat pengering maggot melalui sebuah diagram alir. Berikut adalah diagram alir dari cara kerja aplikasi *monitoring*:



GAMBAR 1.
Flowchart Cara Kerja Aplikasi

Dapat dilihat pada gambar 1 untuk cara kerja aplikasi. Pada tahap awal, aplikasi menampilkan halaman utama untuk memilihi berat maggot yang akan dikeringkan, setelah memilih berat maggot yang akan dikeringkan, maka aplikasi akan mengoperasikan alat pengering maggot dengan menggunakan referensi *database* dari firebase untuk mengontrol alat pengering maggot. Selama proses pengeringan berlangsung, aplikasi menampilkan estimasi waktu pengeringan dan suhu *realtime*. Saat proses pengeringan selesai, aplikasi akan memberikan notifikasi dan alat pengering maggot juga akan otomatis berhenti.

B. Pengumpulan Data

Data yang dikumpulkan adalah data dari proses pengiriman data dari sensor dan komponen pada alat yang kemudian akan dikirimkan ke Firebase *database* dan data tersebut akan digunakan aplikasi sebagai referensi untuk mengontrol dan *monitoring* alat pengering maggot. *Capture* data dilakukan menggunakan wireshark untuk mencari nilai *delay* dan *throughput* untuk menentukan *Quality of Service* dari aplikasi yang dibuat. Proses *capture* data menggunakan wireshark dilakukan selama 2 menit 30 detik.

IV. HASIL DAN PEMBAHASAN

A. Integrasi Aplikasi dan Firebase *Databse*

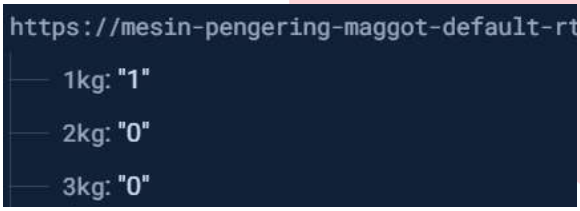
```
databaseReference = FirebaseDatabase.getInstance().reference
ref1kg = databaseReference.child( pathString: "1kg")
ref2kg = databaseReference.child( pathString: "2kg")
ref3kg = databaseReference.child( pathString: "3kg")
refemergency = databaseReference.child( pathString: "emergency")
```

GAMBAR 2.
Database Reference di Android Studio

Pada gambar 2 dibuat sebuah *database reference* pada Android Studio yang akan merujuk ke Firebase *database*.

```
when (selectedBerat) {
    "1kg" -> {
        ref1kg.setValue("1")
        Handler(mainLooper).postDelayed({
            ref1kg.setValue("0")
        }, delayMillis: 7000)
    }
    "2kg" -> {
        ref2kg.setValue("1")
        Handler(mainLooper).postDelayed({
            ref2kg.setValue("0")
        }, delayMillis: 7000)
    }
    "3kg" -> {
        ref3kg.setValue("1")
        Handler(mainLooper).postDelayed({
            ref3kg.setValue("0")
        }, delayMillis: 7000)
    }
```

GAMBAR 3.
Atur *Value* Setiap berat



GAMBAR 4.
Firebase Saat Awal Pengeringan Dimulai

Pada gambar 3 menunjukkan bahwa pada Android Studio diatur *value* berupa "1" pada masing-masing berat yang akan mengubah *value* pada Firebase juga seperti pada gambar 4 ketika sudah memilih berat maggot yang akan dikeringkan pada aplikasi dan berfungsi untuk mengoperasikan alat. Setelah itu diatur sebuah *delay* selama 7 detik untuk mengubah *value* nya kembali menjadi "0" pada Firebase. *Value* diubah menjadi "0" kembali memiliki tujuan agar tidak terjadi *looping* pada proses pengeringan berikutnya saat ditekan tombol stop pada aplikasi yang akan mengubah value emergency pada Firebase menjadi "1" yang berfungsi untuk memberhentikan proses pengeringan di saat proses pengeringan berlangsung.

```
when (selectedBerat) {
    "1kg" -> {
        refemergency.setValue("1")
        Handler(mainLooper).postDelayed({
            refemergency.setValue("0")
        }, delayMillis: 7000)
    }

    "2kg" -> {
        refemergency.setValue("1")
        Handler(mainLooper).postDelayed({
            refemergency.setValue("0")
        }, delayMillis: 7000)
    }

    "3kg" -> {
        refemergency.setValue("1")
        Handler(mainLooper).postDelayed({
            refemergency.setValue("0")
        }, delayMillis: 7000)
    }
```

GAMBAR 5.
Atur *Emergency* Setiap Berat

GAMBAR 6.
Emergency di Firebase

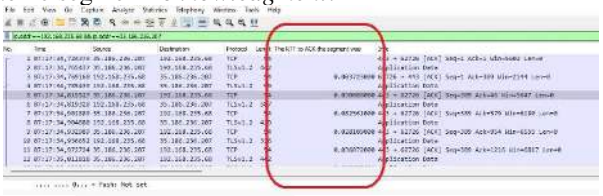
Pada gambar 5 merupakan *source code* pada Android Studio untuk mengatur *value emergency* menjadi "1" pada setiap berat saat tombol *stop* pada aplikasi ditekan. Pada Firebase juga *value* nya akan berubah menjadi "1" seperti terlihat pada gambar 6. Pada *source code* juga dibuat *delay* selama 7 detik yang akan membuat *value* pada Firebase akan kembali menjadi "0" setelah 7 detik.

B. Pengujian *Quality of Service*

Pengujian *Quality of Service* meliputi 2 pengujian yaitu *delay* dan *throughput*. *Delay* dan *throughput* memiliki standar masing-masing untuk menentukan kualitasnya. Nilai *delay* dikategorikan sangat bagus jika nilai *delay* dibawah 150 ms, sedangkan *throughput* dikateogrikan bagus jika memiliki nilai di atas 50 Kbps dan dikategorikan sedang jika memiliki nilai di atas 25 Kbps[7].

1. Pengujian *Delay*

Pada pengujian *delay*, penulis menggunakan wireshark untuk *capture* proses transmisi data dari ESP8266 menuju Firebase dan aplikasi. Data yang digunakan untuk menghitung delay adalah *average time* dari data *Round Trip Time* ke segmen *Acknowledgment*.



GAMBAR 7.
Data pada Wireshark Untuk Menghitung *Delay*

Berdasarkan pada gambar 2, Setelah data diperoleh maka dilakukan perhitungan *delay* dengan rumus sebagai berikut:

$$Delay = \frac{76{,}499}{1683} = 0{,}045\ s = 45\ ms \quad (3)$$

Setelah dilakukan perhitungan, didapatkan *delay* sebesar 45 ms, dengan nilai tersebut maka dapat dikategorikan *delay* rendah dan sangat bagus untuk aplikasi *monitoring* berbasis *Internet of Things*.

2. Pengujian *Throughput*

Pengujian *throughput* juga dilakukan menggunakan wireshark untuk menganalisis *throughput* dari aplikasi yang dibuat.

Statistics

Measurement	Captured	Displayed
Packets	3347	3288 (98.2%)
Time span, s	150.472	150.472
Average pps	22.2	21.9
Average packet size, B	213	216
Bytes	713756	709652 (99.4%)
Average bytes/s	4743	4716
Average bits/s	37 k	37 k

GAMBAR 8.
Data Wireshark Untuk Menghitung *Throughput*

Berdasarkan pada gambar 3, berikut adalah perhitungan *througput*:

$$Throughput = \frac{713756}{150{,}472} \times 8 = 37{,}94\ bps \quad (4)$$

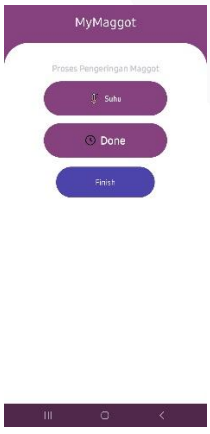
Setelah dilakukan perhitungan, didapatkan *throughput* sebesar 37.94 bps atau 37,94 Kbps, sehingga masuk dalam kategori sedang dan sudah sangat mumpuni untuk aplikasi *monitoring* berbasis *Internet of Things.*

C. Pengujian Pengoperasian Aplikasi

Aplikasi yang dibuat memiliki 3 tampilan halaman. Berikut adalah tampilan dari aplikasi yang sudah dibuat.

GAMBAR 9.
Tampilan Awal

GAMBAR 10.
Tampilan Proses Pengeringan



GAMBAR 11.
Proses Pengeringan Selesai

Pada gambar 9, gambar 10, dan gambar 11 merupakan tampilan pada aplikasi dari setiap tahapan proses pengeringan maggot. Tampilan awal aplikasi terdapat fitur untuk memilih berat maggot yang akan dikeringkan, setelah memilih berat maggot yang akan dikeringkan, tekan tombol *start* untuk memulai proses pengeringan dan aplikasi juga akan menampilkan halaman selanjutnya yaitu halaman saat proses pengeringan berlangsung. Pada halaman ini ditampilkan *monitoring* suhu *realtime* selama proses pengeringan berlangsung dan juga terdapat tombol *stop* yang berfungsi untuk menghentikan proses pengeringan sebelum waktu estimasi pengeringan habis.

Pengujian pengoperasian aplikasi dilakukan dengan melakukan pengujian langsung bersama penanggung jawab produksi maggot *Green House* Universitas Telkom untuk menguji aplikasi yang sudah dibuat. Berdasarkan pengujian langsung yang dilakukan maka didapat beberapa kesimpulan berikut:

1. Aplikasi memiliki tampilan yang *user friendly* sehingga sangat mudah dioperasikan.
2. Fitur yang tersedia pada aplikasi sangat bermanfaat bagi peternak maggot seperti estimasi waktu proses pengeringan dan dapat *monitoring* suhu pengeringan.
3. Aplikasi memiliki kinerja yang responsif degan kemampuan pengoperasian secara *realtime* tanpa ada kendala.

V. KESIMPULAN

Dari hasil penelitian menunjukkan bahwa aplikasi *monitoring* yang dibuat untuk mengontrol alat pengering maggot mampu beroperasi dengan baik tanpa ada kendala. Aplikasi *monitoring* dibuat dengan *platform* Android Studio dan menggunakan Firebase *database* sebagai tempat menyimpan data dari ESP8266 dan menggunakan data tersebut sebagai referensi. Proses diawali dengan ESP8266 mengumpulkan data dari sensor dan komponen yang terpasang pada alat pengering maggot dan kemudian data tersebut secara *realtime* dikirimkan ke Firebase *database*. Setelah itu pada Android Studio dibuat sebuah *database reference* yang mengambil referensi dari Firebase yang selanjutnya referensi tersebut digunakan oleh aplikasi *monitoring* untuk mengoperasikan alat pengering maggot.

Pada pengujian aplikasi dilakukan 2 tahapan pengujian yaitu pengujian *Quality of Service* yang meliputi pengujian *delay* dan *throughput*, serta selanjutnya yaitu pengujian pengoperasian aplikasi. Pada pengujian *Quality of Service* didapatkan nilai untuk *delay* yaitu 45 ms yang masuk dalam kategori sangat bagus untuk aplikasi *monitoring* berbasis *Internet of Things* dan untuk nilai *throughput* didapatkan nilai sebesar 37,974 Kbps yang masuk dalam kategori sedang, tetapi sudah sangat cukup untuk aplikasi *monitoring* berbasis *Internet of Things*. Pada pengujian pengoperasian dilakukan dengan pengujian langsung bersama pengurus produksi maggot *Green House* Universitas Telkom. Setelah dilakukan pengujian dapat disimpulkan bahwa aplikasi mudah untuk dioperasikan dan memiliki fitur yang bermanfaat bagi peternak maggot karena dapat *monitoring* suhu dan waktu estimasi pengeringan, selain itu aplikasi juga menunjukkan performa yang responsif dengan dapat dioperasikan secara *realtime* tanpa ada masalah.

REFERENSI

[1] S. Sibuea, M. Ikhsan Saputro, A. Annan, and Y. Bowo Widodo, "APLIKASI MOBILE COLLECTION BERBASIS ANDROID PADA PT. SUZUKI FINANCE INDONESIA," vol. 2, no. 1, 2022.

[2] K. K. Patel and S. M. Patel, "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges," *International Journal of Engineering Science and Computing*, 2016, doi: 10.4010/2016.1482.

[3] F. H. Danufane and M. Di Renzo, "Analysis of the Delay Distribution in Cellular Networks by Using Stochastic Geometry," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 1728–1744, 2023, doi: 10.1109/OJCOMS.2023.3294791.

[4] J. Xie and T. Murase, "Effective collaboration to maximize throughput based on multiuser cooperative mobility in social-physical Ad Hoc Networks," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 818–835, 2021, doi: 10.1109/OJCOMS.2021.3071853.

[5] I. Firman Maulana, "Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android," vol. 1, no. 3, pp. 854–863, 2017.

[6] E. A. W. Sanad, "Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire," *Jurnal Penelitian Enjiniring*, vol. 22, no. 1, pp. 20–26, May 2019, doi: 10.25042/jpe.052018.04.

[7] A. B. Aldiansyah, M. Hakimah, and D. T. Tukadi, "Sistem Monitoring dan Kontrol Rumah Berbasis Internet Of Things (IoT)," 2022.