ISSN (p) : 2407-1323 DOI : https://doi.org/10.25124/jett.v9i2.4664
ISSN (e) : 2442-4404 | Vol. 9 | No. 2 | Halaman : 1218 - 1231

JETT

Jurnal Elektro Telekomunikasi Terapan Desember 2022

PERANCANGAN SISTEM PEMBERI MAKAN HEWAN BERBASIS IOT DENGAN SISTEM KONTROL ANDROID MENGGUNAKAN MVVM DAN CLEAN ARCHITECTURE

IOT-BASED ANIMAL FEEDING SYSTEM DESIGN WITH ANDROID CONTROL SYSTEM USING MVVM AND CLEAN ARCHITECTURE

Andrew Sebastian Lehman[1], Joseph Sanjaya[2]

[1]Program Studi Sistem Komputer, Fakultas Teknik, Universitas Kristen Maranatha
[2] Program Studi Magister Ilmu Komputer, Fakultas IT, Universitas Kristen Maranatha
[1] **andrewsebastianl@gmail.com**, [2] **mi1879011@student.it.maranatha.edu**

Abstrak
Pada era modern ini teknologi sudah bergerak dengan sangat cepat. Terlebih dengan adanya IoT (Internet of Things) yang sudah diterapkan dalam berbagai bidang. Tidak hanya dalam bidang primer saja, namun IoT sudah berkembang dalam kegiatan sehari-hari maupun hobi. IoT dapat membantu secara efektif dan efisien dalam kegiatan berat maupun kegiatan sehari-hari. Pemberian makan hewan peliharaan merupakan rutinitas yang dapat digantikan dengan sistem yang memanfaatkan IoT. Makalah ini memberikan perancangan sistem IoT yang terperinci dan penerapannya untuk menghadirkan sistem komunikasi lengkap untuk memantau pemberi makan hewan dengan memeriksa data makanan, camilan, dan wadah air melalui server berbasis cloud (Firebase Platform) menggunakan mikro- NodeMcu (Esp8266) berbasis pengontrol dan aplikasi seluler android.

Kata kunci: Android, MVVM, Internet of Things (IoT), Firebase, Clean Architecture

Abstract
In this modern era, technology has moved very fast. The emergence of IoT has played an important role in every sector of our daily lifestyle. Not only on the primary needs, but IoT also help effective and efficiently on heavy task or just a daily activity. Pet feeding is a rutinity that can be replaced with an automation system that utilize IoT. This paper provides a detailed hands-on IoT and how it can be applied to deliver a complete communication system to monitor dog feeders by checking their food, treats, and water containers data via a cloud-based server (Firebase Platform) using a controller-based micro-NodeMcu (Esp8266) and android mobile application.

Keywords: Android, MVVM, Internet of Things (IoT), Firebase, Clean Architecture

1. PENDAHULUAN

Pesatnya perkembangan teknologi berdampak pada kehidupan manusia yang dibantu dengan berbagai otonomi sehingga dapat menjalankan tugasnya dengan lebih efisien. Terlebih dengan ramainya teknologi IoT (Internet of Things) membuat perusahaan besar mulai mendalami hal tersebut. Dalam beberapa tahun ini IoT sudah banyak diterapkan di beberapa bidang [1]. IoT tidak hanya berkembang dalam bidang primer tetapi sudah dapat diimplementasikan terhadap kegiatan sehari-hari maupun hobi [2].

Sebagai pecinta hewan peliharaan khususnya anjing, banyak orang sering kali dihadapkan pada masalah seperti ketika ada kebutuhan yang mengharuskan pemiliknya untuk meninggalkan peliharaannya dalam jangka waktu yang lama. Masalah yang utama adalah dalam memberi makan anjing dalam waktu yang sesuai.

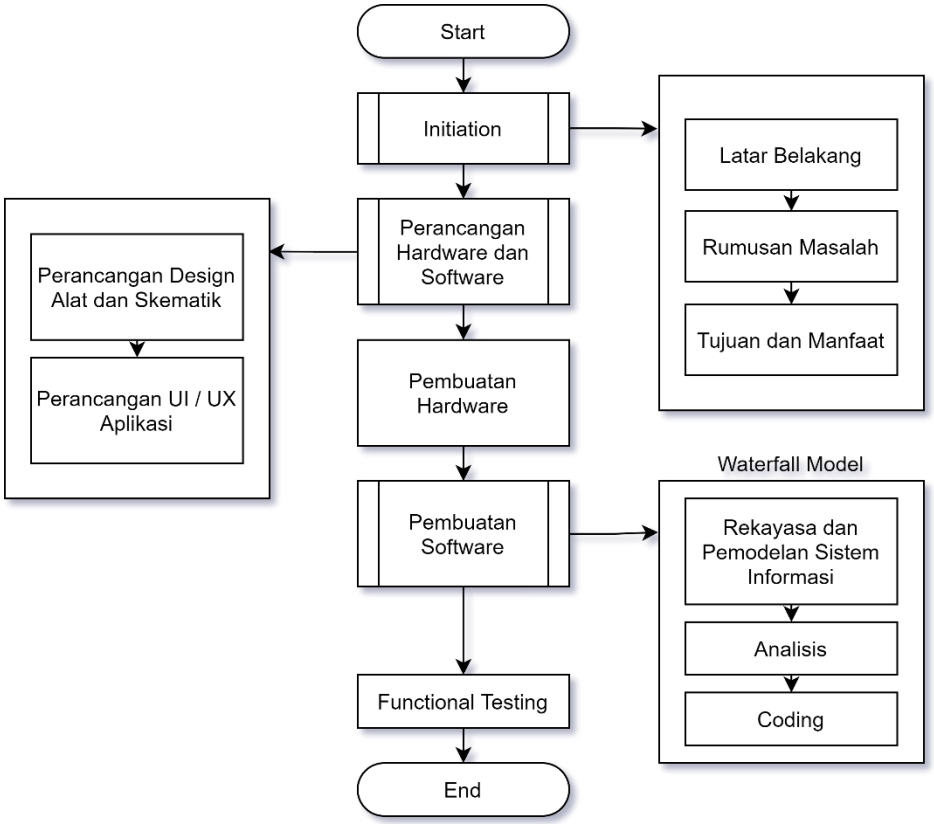
Berdasarkan kenyataan tersebut untuk menyelesaikan masalah dibuatlah perangkat pemberi makan dan minum hewan menggunakan NodeMcu (Esp8266), aplikasi Android, dan Firebase *Platform*. Pengguna dapat melihat kondisi makanan dan minuman untuk anjingnya. Selain itu

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

pengguna juga bisa menjatuhkan makanan atau mengisi minuman secara manual atau menyesuaikan jadwal. Dengan perangkat tersebut, pengguna mengurangi kekhawatiran dalam meninggalkan hewan peliharaan anjingnya dalam waktu yang ditentukan.

2. METODE PENELITIAN

Desain dari penelitian ini dideskripsikan dengan diagram *flowchart* yang dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.1 Perancangan Hardware dan Software

Perancangan hardware terdiri dari pembentukan skematik dan design perangkat awal sehingga dapat mempermudah bayangan awal dari perangkat yang akan dibuat. Perancangan Software terdiri dari UI dan UX dari aplikasi dan arsitektur dari aplikasi seluler sehingga aplikasi dapat dibentuk berdasarkan prinsip Clean Architecture [3]. Keseluruhan sistem dibuat dengan model proses waterfall. Metode ini mengusulkan pendekatan yang sistematis dan berurutan terhadap masalah perangkat lunak, mulai dari tingkat dan kemajuan sistem hingga analisis, desain, kode, pengujian, dan pemeliharaan [4], dengan tahapan sebagai berikut:

1. Rekayasa perangkat lunak dan perangkat keras
   Implementasi dari hardware, software, dan lainnya menjadi suatu sistem utuh.
2. Pengumpulan data dan analisis
   Pengumpulan *data* pendukung penelitian, penyusunan jadwal kegiatan penelitian dan penguraian alat pendukung yang digunakan.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

3. Pembuatan Program
   Penterjemahan prototipe dan model ke dalam bahasa yang dapat dibaca.

2.2 Penelitian Serupa

IoT adalah topik penting dalam otomatisasi komputer yang telah banyak diteliti. Berbagai kegunaan telah diusulkan oleh beberapa peneliti berdasarkan fungsi yang berbeda. Pada penelitian sebelumnya dirancang sistem pemberi makan ikan otomatis menggunakan teknologi mikrokontroler [5]. Minsung Hong dan Rajendra Akerkar mengusulkan arsitektur Victim Detection Platform (VDP) yang menggabungkan kedua pendekatan tersebut menggunakan berbagai teknologi canggih seperti IoT, drone, dan komputasi edge / cloud untuk menawarkan layanan berkualitas lebih tinggi, yang pada akhirnya menghasilkan respons yang cepat serta banyak nyawa yang diselamatkan [6]. Yixing Chen dan Maher Elshakankiri menerapkan teknologi IoT untuk menerapkan sistem terintegrasi termasuk pengumpan makanan hewan, dispenser air, dan kotak kotoran, yang merupakan tiga elemen paling mendasar yang akan diperhatikan pemilik hewan peliharaan saat mereka sibuk atau jauh dari hewan peliharaannya [7]. Sangvanloy Tannop dan Kingkarn Sookhanaphibarn mengembangkan feeder untuk membantu mengalokasikan makanan kering untuk hewan peliharaan kecil seperti anjing dan kucing dengan memanfaatkan teknologi IoT [8].

2.3 Arduino Mega

Arduino Mega 2560 adalah sebuah *Board* Arduino yang menggunakan *IC* Mikrokontroler ATmega 2560. *Board* ini memiliki *Pin I/O* yang relatif banyak, 54 digital *Input* / *Output*, 15 buah di antaranya dapat digunakan sebagai *output PWM*, 16 buah analog *Input*, 4 UART. Arduino Mega 2560 dilengkapi kristal 16 *MHz.*

Untuk penggunaan yang sederhana tinggal menghubungkan *power* dari *USB* ke *PC* / *Laptop* atau melalui *Jack DC* menggunakan *adaptor* 7-12*VDC* [9]. Spesifikasinya dapat dilihat pada Gambar 2.

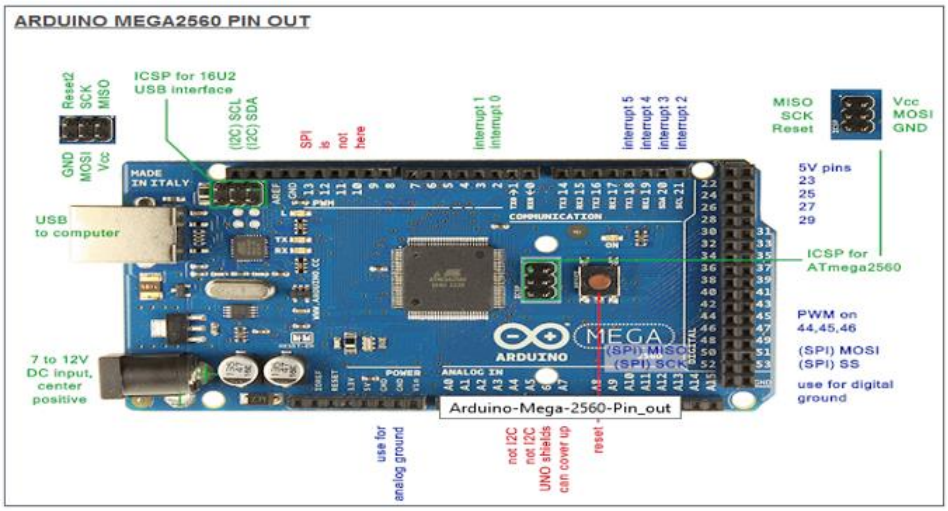
SPESIFIKASI

Mikrokontroler	ATmega2560
Tegangan Operasianal	5V
Tegangan Input (rekomendasi)	7-12V
Tegangan Input (limit)	6-20V
Pin Digital I/O	54 (of which 15 provide PWM output)
Pin Analog Input	16
Arus DC per Pin I/O	20 mA
Arus DC untuk Pin 3.3 V	50 mA
Memori Flash	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	101.52 mm
Lebar	53.3 mm
Berat	37 g

Gambar 2. Spesifikasi Arduino Mega [9]

Arduino Mega 2560 *Pin Out* dapat dilihat pada Gambar 3.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022



Gambar 3. Arduino Mega 2560 Pin Out [9]

2.4 ESP8266 Module

ESP8266 merupakan modul *wifi* yang berfungsi sebagai tambahan mikrokontroler sehingga memungkinkan *system* untuk terhubung dengan *wifi* dan membuat koneksi *TCP/IP*. Daya yang dibutuhkan oleh modul ini adalah sekitar 3.3v dengan tiga mode *wifi* yaitu *Station, Access Point* dan *Both*. Dilengkapi dengan prosesor dan memori. *GPIO* memiliki jumlah pin bergantung dengan jenis *ESP8266* yang digunakan. Modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler karena sudah memiliki perlengkapan seperti mikrokontroler [10].

ESP-01 adalah salah satu dari beberapa modul *ESP8266*. Modul *ESP-01* tergolong *StandAlone* atau *System on Chip* yang tidak selalu membutuhkan mikrokontroler untuk kontrol *input output* karena *ESP-01* dapat bertindak sebagai mini komputer, dengan kondisi jumlah *GPIO* ynag terbatas.

2.5 Sensor Berat (Load Cell)

Sensor berat (*Load Cell*) adalah sensor yang menghasilkan sinyal listrik dan besarnya sinyal tersebut akan sebanding dengan berat yang diukur. Setiap *Load Cell* memiliki karakteristik yang berbeda-beda. Nilai skala dan offset yang baik dapat diperoleh dengan menetapkan nilai beban pengkalibrasi antara 25%-75% dari kapasitas maksimun sebuah *Load Cell*.

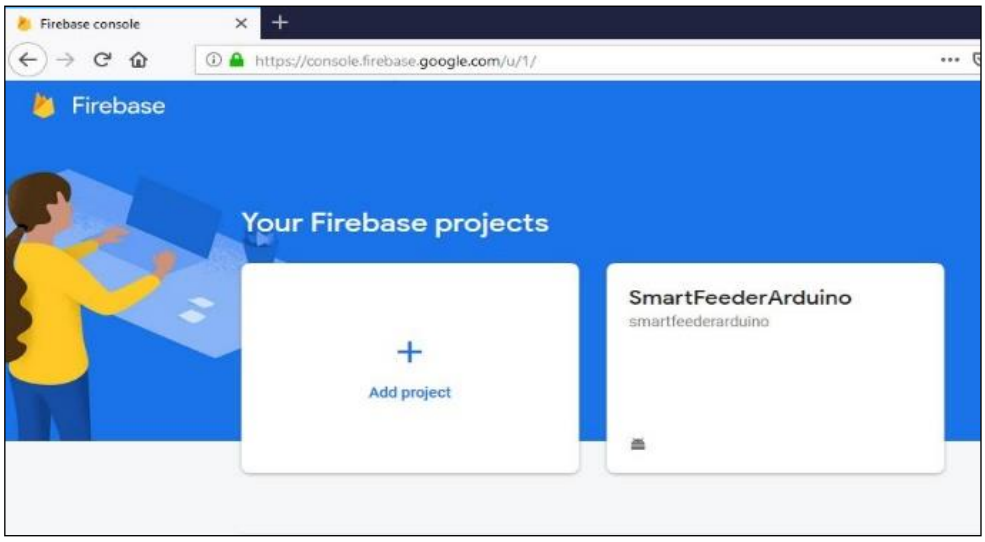
Sensor berat (*Load Cell*) biasanya ditambah dengan *HX711* yaitu komponen (modul) yang berfungsi sebagai *ADC – Analog to Digital converter* 24bit untuk mengkonversi nilai resistansi menjadi nilai yang berbanding lurus dengan berat benda yang ditimbang [11].

2.6 Firebase Realtime Database

Firebase adalah *Cloud Service Provider* dan *Backend as a Service* yang dimiliki oleh Google untuk mempermudah dalam mengembangkan pembuatan aplikasi baik Android, *Web* dan lainnya. Salah satu fitur yang dimiliki *Firebase* adalah Firebase Realtime Database [12].

Firebase Realtime Database adalah *Cloud-Hosted database* yang dapat menyimpan dan melakukan sinkronisasi data secara *realtime* untuk setiap *client* yang terhubung. Saat ada pembaharuan dari dari user, maka secara otomatis data tersebut akan disimpan pada *cloud* dan seluruh *client* yang terhubung akan menrima data yang terbaru. Gambar 4 menunjukan tampilan awal *Firebase Database*.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

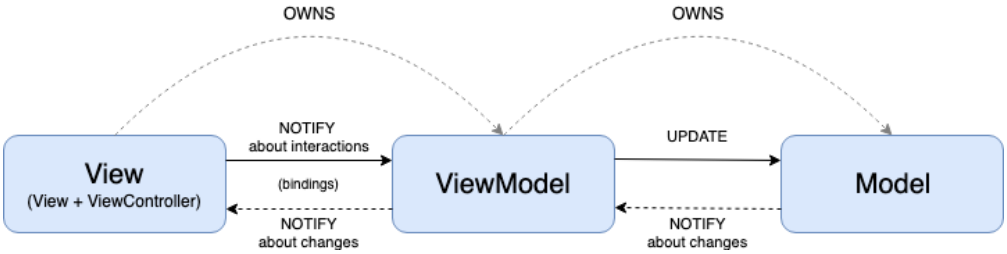


Gambar 4. Tampilan awal Firebase Database [12]

2.7 MVVM Software Architecture

Model-View-ViewModel (MVVM) adalah pola desain struktural yang memisahkan objek menjadi tiga kelompok berbeda:

1. Model menyimpan *data* aplikasi. Model biasanya *struct* atau kelas sederhana.
2. *Views* menampilkan elemen visual dan kontrol di layar. *Views* biasanya subkelas dari *UIView*.
3. *View Model* mengubah informasi *model* menjadi nilai yang dapat ditampilkan pada tampilan. *View Model* biasanya kelas, sehingga dapat diedarkan sebagai referensi



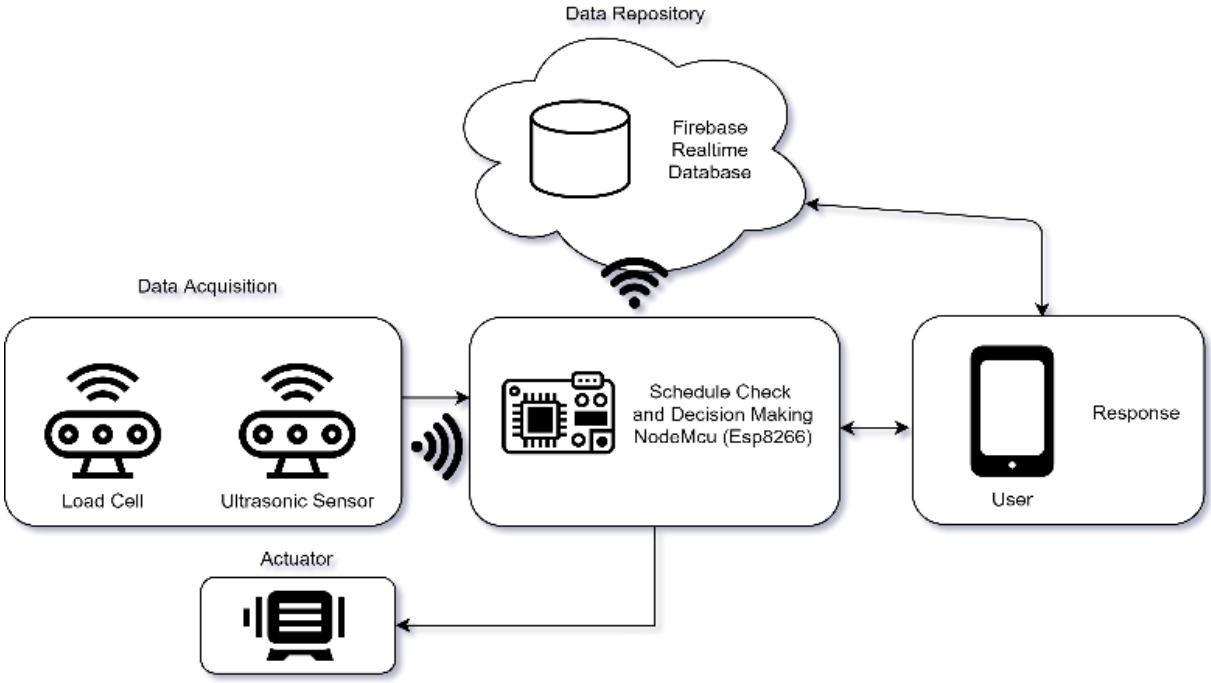
Gambar 5. Cara Kerja MVVM Design

Cara kerja dari *design MVVM* pada *software* dapat dilihat pada Gambar 5. Gambar 5 menjelaskan cara kerja *design MVVM* yaitu:

1. *Activity* bertanggung jawab sebagai *View*
2. *View* akan melakukan observasi terhadap data yang disimpan di *ViewModel*. Jika terdeteksi ada perubahan pada data di *ViewModel*, maka *View* bertanggung jawab untuk melakukan *update* pada antarmuka sesuai dengan data.
3. *ViewModel* menyimpan data berupa *LiveData* agar *View* dapat melakukan observasi.
4. *ViewModel* berkomunikasi dengan *Repository* (*Model*) untuk mendapatkan data atau perubahan data dan melakukan *update* terhadap data yang dimiliki.
5. *Repository* bertanggung jawab untuk mengatur sumber data yang dibutuhkan. Data bisa didapatkan baik dari *server* maupun dari *local database* menggunakan *SQLite*.

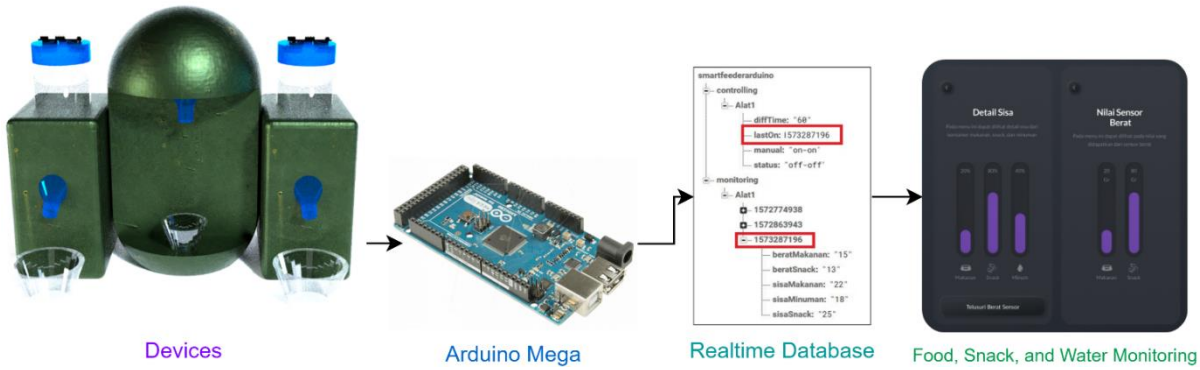
ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

2.8 Perancancangan Struktur Komunikasi



Gambar 6. Struktur Komunikasi

Pada subbab ini akan dijelaskan struktur komunikasi pada implementasi feeder IoT ini. Struktur keseluruhan dapat dilihat pada Gambar 6 dan Gambar 7.



Gambar 7. Garis besar komunikasi perangkat kepada aplikasi seluler

Struktur komunikasi yang diusulkan pada Gambar 7 menekankan pada pengaturan platform database yang dihosting pada cloud dan menjelaskan peran Arduino dan aplikasi seluler pengguna untuk visualisasi data. Gambar 7 menunjukkan seluruh skema komunikasi dari perangkat, mikrokontroler, database Firebase Realtime yang dihosting pada cloud hingga aplikasi seluler.

2.9 Prototype Hardware

Desain alat berbentuk persegi panjang, dan terdapat 2 toples plastik sebagai wadah untuk memasukan makanan dan *snack* anjing. Di bagian atas terdapat 2 sensor ultrasonik untuk mengetahui kondisi makanan dan *snack* apakah sudah habis atau belum. Di bagian bawah terdapat 2 wadah yang masing-masing bertujuan untuk menampung sereal dan *snack*. Selain itu ada 2 sensor berat untuk

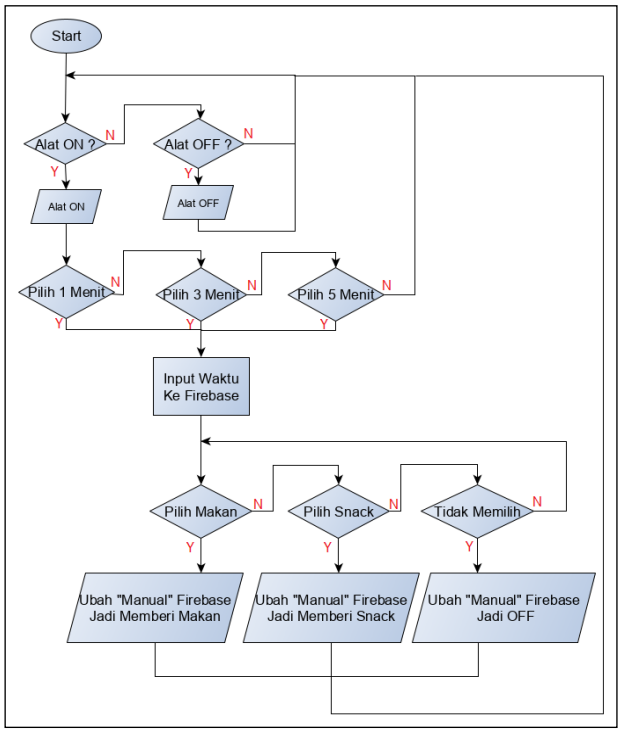
ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

melihat berat di masing-masing wadah. Gambar 8 merupakan desain dari alat pemberi makan dan minum.

Gambar 8. Prototype Hardware

2.10 Flowchart Perangkat

Pada Gambar 9 menunjukan *flowchart* aplikasi dari alat pemberi makan dan minum menggunakan aplikasi seluler. *User* memilih untuk mematikan atau menghidupkan alat. Jika *user* memilih menghidupkan alat, *user* memilih waktu yang akan menjadi *input* dari Firebase. Setelah memilih waktu, *user* bisa menekan tombol di aplikasi untuk memberi makan dan *snack*. User juga dapat untuk tidak menekan tombol makan dan *snack*. Setelah menekan salah satu tombol maka Firebase akan di-*update*.

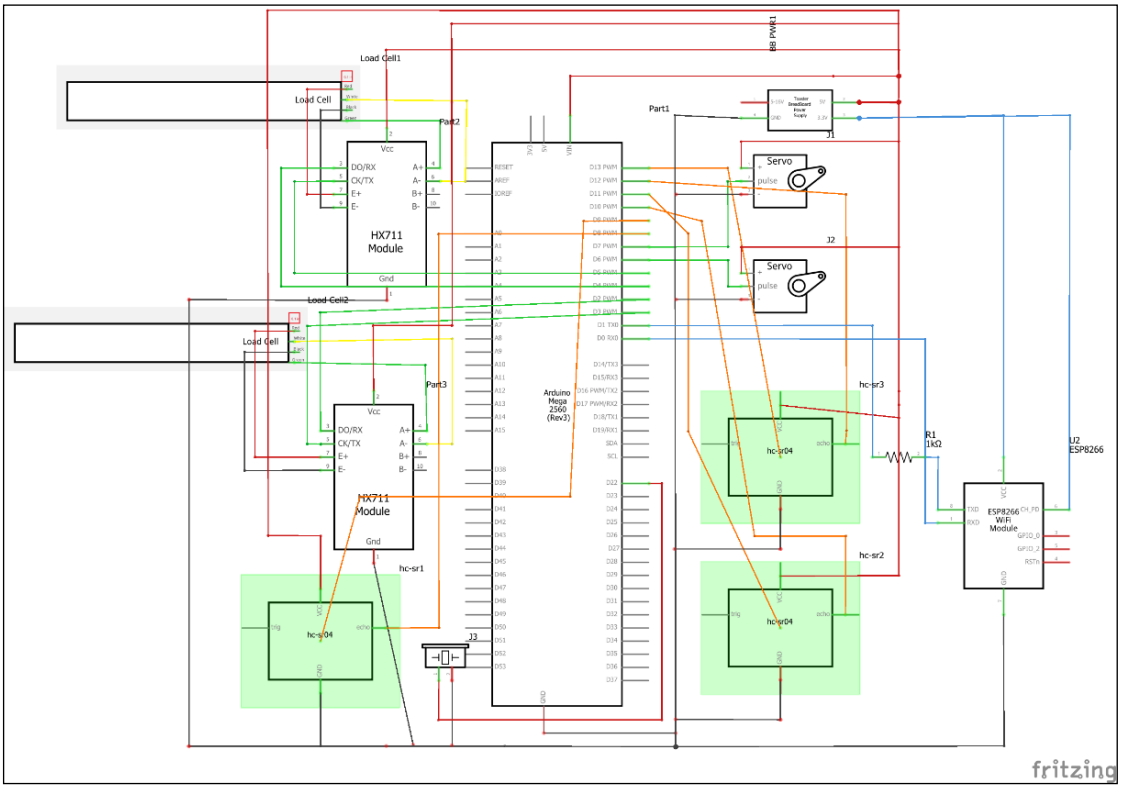


Gambar 9. Flowchart aplikasi dari alat pemberi makan dan minum.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

2.11 Perancangan Skematik Hardware

Gambar 10 menunjukan skematik rangkaian pada Arduino Mega dan pada Tabel I adalah konfigurasi port yang terhubung dengan masing-masing komponen input dan output pada Arduino Mega.



Gambar 10. Skematik rangkaian pada Arduino Mega

Dalam rangkaian Arduino Mega menggunakan sensor ultrasonik sebagai *input* sedangkan motor *servo* dan *buzzer* sebagai *output*. Untuk lebih jelas konfigurasi ada pada Tabel 1.

Tabel 1. Konfigurasi sistem minimum Arduino Mega

Jenis Port	Nomor Port	Terhubung Dengan
Digital	22	*Buzzer*
Digital	13	*Trig* Makanan
Digital	12	*Echo* Makanan
Digital	11	*Trig* Snack
Digital	10	*Echo* Snack
Digital	9	*Trig* Minuman
Digital	8	*Echo* Minuman
Digital	7	Servo Makanan
Digital	6	Servo Snack
Digital	5	CLK Makan
Jenis Port	Nomor Port	Terhubung Dengan
Digital	4	DOUT Makan
Digital	3	CLK Snack
Digital	2	DOUT Snack
TX	1	*ESP8266*
RX	0	ESP8266

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

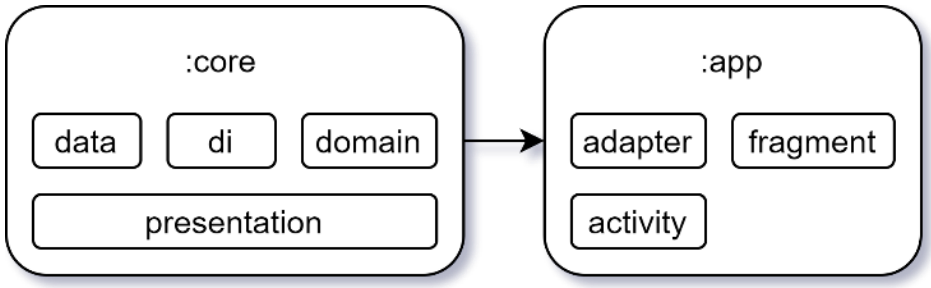
2.12 Pembuatan Tampilan Aplikasi

Pada Gambar 11 menunjukan tampilan untuk pembuatan halaman *monitoring* pada aplikasi. Pada Gambar 11 *user* dapat melihat sisa makanan, sisa *snack*, sisa minuman, berat makanan dan berat *snack* yang akan di *update* secara terus menerus.

Gambar 11. Tampilan halaman monitoring pada aplikasi seluler

2.13 Perancangan Modul Aplikasi

Aplikasi seluler dirancang menggunakan prinsip multi-module sehingga modul presentasi dan data dapat dipisahkan sesuai dengan Clean Architecture [3]. Perancangan multi-module aplikasi dapat dilihat Gambar 12.



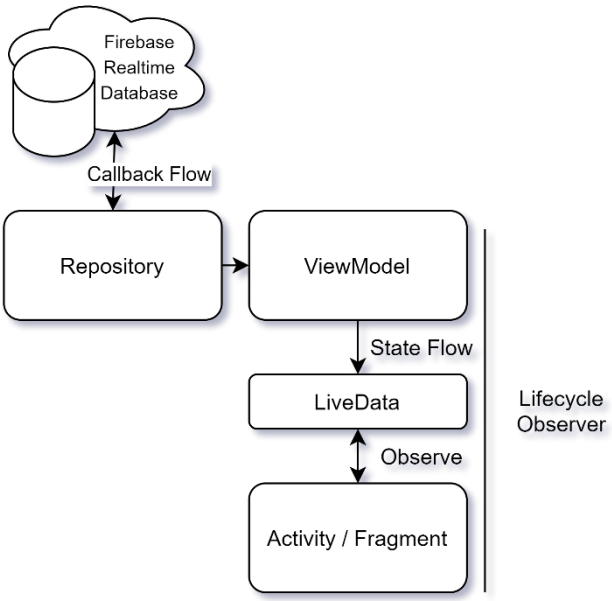
Gambar 12. Multi-module design aplikasi

Pada Gambar 12 terlihat pada module core terdapat beberapa lapisan yaitu data, di, domain, dan presentation. Pada lapisan data terdiri dari kelas repository dan network interfaces yang berguna sebagai lapisan pengirim dan penyedia data. Lapisan di berguna sebagai penyedia dependencies yang diperlukan, menggunakan teknologi dependencies injection. Lapisan domain berisi kelas data yang

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

berguna sebagai serializer data yang didapat dari Firebase Realtime Database. Lapisan presentation berisi kelas ViewModel dan Observer yang berfungsi sebagai komunikasi antara adapter, fragment, dan activity dengan lapisan data.

2.14 Perancangan Komunikasi antar Lapisan Aplikasi



Gambar 13. Diagram komunikasi antar lapisan aplikasi

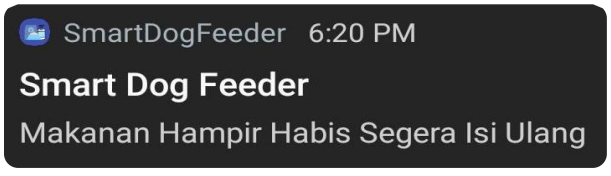
Pada Gambar 13 komunikasi antara Firebase dengan aplikasi terjadi pada lapisan data di kelas repository. Komunikasi dilakukan dengan menggunakan Firebase KTX, state dari request data dibaca menggunakan fasilitas Kotlin Coroutines yaitu Callback Flow, sehingga data dapat dibaca diluar dari Coroutines Context. Data akan disimpan pada ViewModel dan dipantau sebagai LiveData yang nantinya akan ditampilkan pada Activity atau Fragment. Semua komunikasi ini dilakukan di dalalm Lifecycle Observer sehingga tidak terjadi memory leaks.

3. PEMBAHASAN

Pada subbab ini akan dilakukan percobaan dan evaluasi berdasarkan Functional Testing dari setiap komponen dari perangkat dan aplikasi yang dibuat.

3.1 Pengujian Notifikasi

Pengujian notifikasi dilakukan untuk mengetahui kondisi makanan, *snack* dan minum hampir habis. Notifikasi makanan akan muncul di *Smartphone* saat ultrasonik makanan membaca jarak lebih dari 15. Gambar 14 menunjukan notifikasi yang keluar di layar *Smartphone.*



Gambar 14. Notifikasi makanan di layar Smartphone

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

Data pengujian notifikasi makanan dapat dilihat pada Tabel 2.

Tabel 2. Pengujian pada notifikasi makanan

Percobaan	Notif Makanan
1	Berhasil (Muncul)
2	Berhasil (Muncul)
3	Berhasil (Muncul)
4	Berhasil (Muncul)
5	Berhasil (Muncul)
Persentase Keberhasilan	100%

Notifikasi *snack* akan muncul di *Smartphone* saat ultrasonik *snack* membaca jarak lebih dari 15. Gambar 15 menunjukan notifikasi yang keluar di layar *Smartphone.*

Gambar 15. Notifikasi snack di layar Smartphone

Data pengujian notifikasi *snack* dapat dilihat pada Tabel 3.

Tabel 3. Pengujian pada notifikasi *snack*

Percobaan	Notif Snack
1	Berhasil (Muncul)
2	Berhasil (Muncul)
3	Berhasil (Muncul)
4	Berhasil (Muncul)
5	Berhasil (Muncul)
Persentase Keberhasilan	100%

Notifikasi minum akan muncul di *Smartphone* saat ultrasonik minum membaca jarak lebih dari 15. Gambar 16 menunjukan notifikasi yang keluar di layar *Smartphone.*

Gambar 16. Notifikasi minum di layar Smartphone

Data pengujian notifikasi minum dapat dilihat pada Tabel 4.

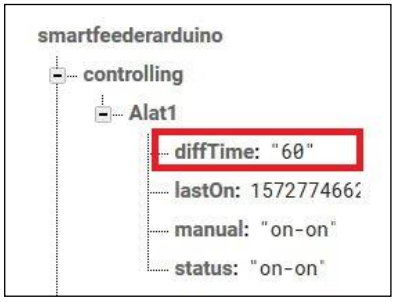
ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

Tabel 4. Pengujian pada notifikasi minum

Percobaan	Notif Minum
1	Berhasil (Muncul)
2	Berhasil (Muncul)
3	Berhasil (Muncul)
4	Berhasil (Muncul)
5	Berhasil (Muncul)
Persentase Keberhasilan	100%

3.2 Pengujian Waktu Pemberian Makan

Pada pengujian ini dilakukan pada 3 waktu yang berbeda. Tampilan "*diffTime*" pada Firebase akan berubah sesuai waktu yang dipilih saat di menu *controlling*. Gambar 17 menunjukan perubahan waktu di *Firebase*.



Gambar 17. Perubahan waktu 1 menit (60 detik) pada Firebase

Data pengujian waktu 1 menit, 3 menit, dan 5 menit dapat dilihat pada Tabel 5.

Tabel 5. Pengujian "*diffTime*"

Percobaan	1 Menit	3 Menit	5 Menit
1	Berhasil	Berhasil	Berhasil
2	Berhasil	Berhasil	Berhasil
3	Berhasil	Berhasil	Berhasil
4	Berhasil	Berhasil	Berhasil
5	Berhasil	Berhasil	Berhasil
Persentase Keberhasilan			100%

3.3 Pengujian Berat

Pengujian ini dilakukan untuk melihat rata-rata berat makanan dan *snack* yang jatuh ke wadah. Pengujian dilakukan secara 5 kali dan melihat berat makanan yang jatuh kedalam wadah saat *servo* makanan aktif. Tabel pengujian berat makanan dapat dlihat pada Tabel 6.

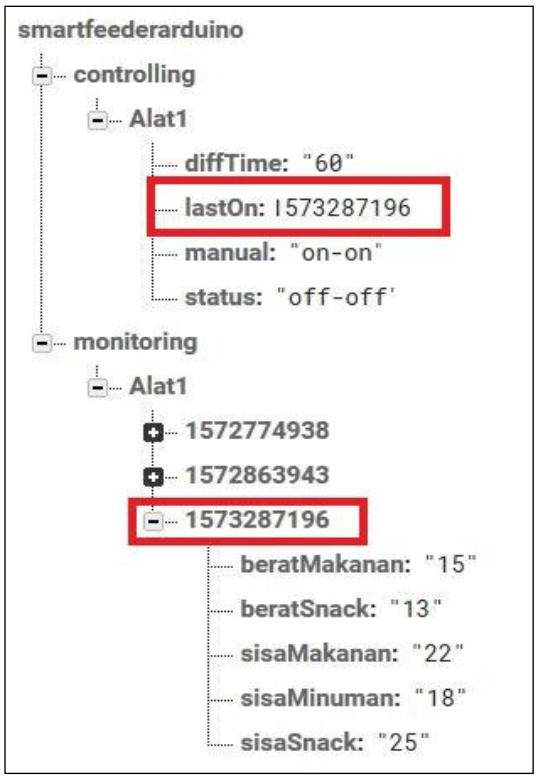
Tabel 6. Data Pengujian berat

Percobaan	Makanan	Snack
1	15 dag	13 dag
2	13 dag	10 dag
3	15 dag	12 dag
4	15 dag	12 dag
5	12 dag	13 dag
Rata-Rata	14 dag	12 dag

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

3.4 Pengujian Firebase Database

Pengujian dilakukan untuk melihat apakah data pada *monitoring* sesuai dengan waktu yang dipilih *user* di menu *controlling*. Gambar 18 menunjukan "*lastOn*" dan *monitoring* terakhir sesuai.



Gambar 18. Kesamaan "lastOn" dengan data terakhir monitoring

Data pengujian pada Gambar 18 dapat dilihat pada Tabel 7.

Tabel 7. Data pengamatan "*lastOn*" dengan *monitoring* terakhir

Percobaan	"*lastOn*"
1	Berhasil (Muncul)
2	Berhasil (Muncul)
3	Berhasil (Muncul)
4	Berhasil (Muncul)
5	Berhasil (Muncul)
Persentase Keberhasilan	100%

3.5 Pengujian Keseluruhan

Pengujian keseluruhan dilakukan dengan menyabungkan motor servo, *buzzer*, sensor ultrasonik, dan ESP 8266 disambungkan dengan Arduino Mega. Setelah semua terpasang alat dinyalakan dengan mencolokkan adaptor 12*V* yang dihibungkan ke modul *power supply*.

4. KESIMPULAN

Pada bab ini akan membahas kesimpulan dalam pembuatan Sistem Pemberi Makan Hewan Berbasis IoT, dari penelitian ini dapat diambil kesimpulan yaitu, Perancangan Sistem Pemberi Makan Hewan Berbasis IoT telah berhasil terealisasi. Pemberian makan dapat dijadwalkan sesuai dengan pemberian waktu dari *user*. Tingkat keberhasilan pada percobaan waktu 1 menit adalah 70%. Tingkat keberhasilan pada percobaan waktu 3 menit, 5 menit, dan 6 hari dengan waktu 6 jam adalah 100%.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.4664
| Vol. 9 | No. 2 | Halaman : 1218 - 1231
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

DAFTAR PUSTAKA

[1] L. Tan and N. Wang, "Future internet: The Internet of Things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE 2010)*, 2010, Chengdu, China: IEEE, pp. V5-376-V5-380, doi: 10.1109/ICACTE.2010.5579543.

[2] A. I. A. Ahmed *et al.*, "Service Management for IoT: Requirements, Taxonomy, Recent Advances and Open Research Challenges," *IEEE Access*, vol. 7, pp. 155472-155488, 2019, doi: 10.1109/ACCESS.2019.2948027.

[3] R. C. Martin and R. C. Martin, *Clean architecture: a craftsman's guide to software structure and design* (Robert C. Martin series). London, England: Prentice Hall, 2018, p. 404.

[4] C. A. Crespo-Santiago and S. d. l. C. D. Cosme, "Waterfall method: a necessary tool for implementing library projects," *HETS Online Journal*, vol. 1, no. 2, pp. 86-99, 2011.

[5] A. S. Lehman and J. Sanjaya, "Automatic Fish Feeder Using Microcontroller," in *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 9*, Pekanbaru, 2017: Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, pp. 345-351.

[6] M. Hong and R. Akerkar, "Victim detection platform in IoT paradigm," (in en), *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, 2021, doi: 10.1002/cpe.5254.

[7] Y. Chen and M. Elshakankiri, "Implementation of an IoT based Pet Care System," in *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2020, Paris, France: IEEE, pp. 256-262, doi: 10.1109/FMEC49853.2020.9144910.

[8] T. Sangvanloy and K. Sookhanaphibarn, "Automatic Pet Food Dispenser by using Internet of Things (IoT)," in *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*, 2020, Kyoto, Japan: IEEE, pp. 132-135, doi: 10.1109/LifeTech48969.2020.1570620257.

[9] S. Siswanto, M. Anif, D. N. Hayati, and Y. Yuhefizar, "Pengamanan Pintu Ruangan Menggunakan Arduino Mega 2560, MQ-2, DHT-11 Berbasis Android," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 1, pp. 66-72, 2019, doi: 10.29207/resti.v3i1.797.

[10] C. Batrinu, *ESP8266 home automation projects: leverage the power of this tiny WiFi chip to build exciting smart home projects*. Birmingham, UK Mumbai: Packt (in eng), 2017, p. 182.

[11] Q. Zhuang, "Weighing System Design Based on Single Chip Microcomputer," 2015, vol. 1070: Trans Tech Publ, pp. 1572-1575.

[12] L. Moroney, "The firebase realtime database," in *The Definitive Guide to Firebase*. New York, NY: Springer Science, 2017, pp. 51-71.