

Sistem *Backend* Pemantauan Makanan Kucing Berbasis *Internet of Things*

1st Tassya Ramadhanti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

tassya@student.telkomuniversity.ac.id

2nd Purba Daru Kusuma
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

purbodaru@telkomuniversity.ac.id

3rd Rifqi Muhammad Fikri
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

rifqmff@telkomuniversity.ac.id

Abstrak — Sistem otomatis yang mengatur pola makan melalui telepon cerdas telah mengubah perawatan hewan peliharaan di era modern yang didorong oleh teknologi yang mana semakin maju. Backend merupakan operasi bisnis yang sangat kompleks dan sangat penting untuk aplikasi *IFT* dimana akan menyimpan semua data dan mengatur jadwal pemberian makanan. Melalui jadwal makan yang teratur dan sehat, aplikasi *IFT* menawarkan solusi inovatif untuk mengotomatiskan proses pemberian makanan dan meningkatkan kualitas hidup hewan seperti kucing. Pengembangan dengan kerangka kerja *React Native* untuk telepon cerdas, *database real-time Firebase*, dan alat bantu seperti *Vysor* dan *Visual Studio Code* memberikan fondasi yang kuat untuk penerapan teknologi ini.

Kata kunci: *Mobile Application, Backend, Firebase.*

I. PENDAHULUAN

Dalam era modern saat ini, di mana teknologi telah berkembang pesat dalam semua aspek kehidupan, kita telah bergantung pada kemajuan teknologi ini dari dahulu, baik secara sadar maupun tidak sadar. Salah satu contoh perkembangan teknologi yang saat ini kita rasakan yaitu perawatan hewan peliharaan dalam mengatur pola makan yang sehat untuk kesejahteraan hewan, yang saat ini otomatis dan dapat diatur melalui telepon cerdas pemilik hewannya.

Selain bergantung pada perangkat keras seperti dispenser pakan, sistem ini sangat bergantung pada backend yang handal. Backend sangat penting untuk menjalankan semua tugas yang rumit secara kompleks di balik layar, seperti jadwal pemberian makanan dan menyimpan serta mengawasi hewan peliharaan. Adanya backend ini memungkinkan pengguna berinteraksi dengan lancar menggunakan aplikasi pada telepon cerdas yang dimiliki.

Salah satu pendekatan inovatif yang sedang dikembangkan oleh penulis adalah aplikasi *IFT (Intelligent Food Timer)*. *IFT* tidak hanya mengotomatiskan proses pemberian makan hewan peliharaan tetapi juga membantu meningkatkan kualitas hidup mereka dengan mengatur jadwal makan yang sehat dan teratur sesuai porsi makan hewan. Dengan cara ini, pemilik hewan peliharaan dapat

merasa lebih tenang dan lebih terhubung dengan kebutuhan harian kucing mereka.

Namun, ada beberapa masalah yang harus diselesaikan, seperti kebutuhan akan integrasi yang lancar antara perangkat keras, telepon cerdas, dan *backend*. Pengguna juga mengharapkan solusi yang dapat diandalkan dan mudah digunakan tanpa mempelajari teknologi yang rumit. Mengingat kebutuhan ini, pengembangan *IFT* tidak hanya berfokus pada aspek teknis, tetapi juga pada kenyamanan dan kepuasan pengguna akhir.

II. KAJIAN TEORI

A. Mobile Application

Aplikasi seluler adalah perangkat lunak yang dibuat untuk perangkat komputasi nirkabel seperti tablet dan telepon cerdas. Aplikasi ini dapat diinstal dan diunduh dari toko aplikasi seperti Google Play atau Apple App Store. Setelah diinstal, aplikasi seluler berkomunikasi dengan sistem operasi perangkat untuk mendapatkan akses ke perangkat [1].

B. Framework Mobile

React native adalah kerangka kerja yang digunakan untuk dikembangkan di berbagai platform, seperti *Android*, *iOS*, dan *website*. Hingga saat ini, *React native* terus berkembang sejak dirilis oleh meta pada tahun 2015. *React native* menggunakan *javascript* untuk menampilkan antarmuka pengguna dengan komponen *view* untuk *layout* dan pemformatan, teks untuk menampilkan teks, dan gambar untuk menampilkan gambar dan banyak komponen lainnya.

Karena mudah, serta *open source* dimana kode sumber yang ada sudah banyak dan pengembang bisa melakukan modifikasi secara lebih luas lagi sehingga *React native* banyak digunakan oleh para pengembang [2].

C. Firebase

Firebase adalah layanan *database* yang didukung *google* sebagai *backend as a services* dan *NoSQL*. *Firebase* memungkinkan pengguna dengan mudah mengelola dan penyimpanan data. *Firebase* memiliki berbagai *tools* seperti *app hosting*, *data connect*, *authentication*, *cloud functions*, *realtime database*, dan *storage*. *Firebase* juga menyediakan *SDK* untuk pengembangan [3].

D. Vysor

Vysor dapat digunakan secara gratis oleh pengguna dan menawarkan fitur pencerminan untuk pengembangan aplikasi sebagai emulator dan model yang bisa diatur layar penuh untuk *Android* dan *iOS*.

Cara penggunaan *vysor*, pertama aktifkan *USB Debugging* di *handphone* pengembang, sambungkan laptop ke *handphone android* ntuk mengetahui apakah sudah tersambung atau belum, gunakan perintah "*adb devices*" di *prompt* perintah atau bisa dilihat langsung pada aplikasi *vysor* [4].

E. Visual Studio Code

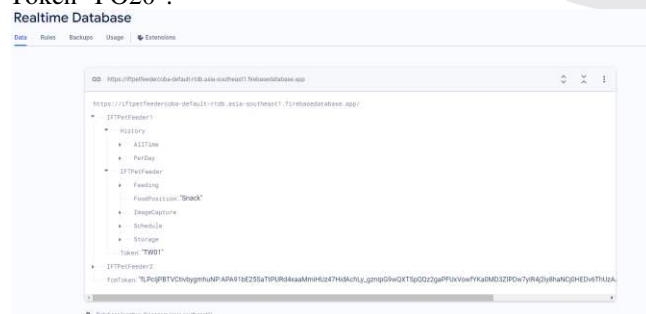
Visual Studio Code adalah editor kode untuk *Windows*, *macOS*, dan *Linux*. *VS Code* memiliki banyak bawaan bahasa pemrograman termasuk *JavaScript*, *TypeScript*, dan *Node.js*, serta ekosistem ekstensi yang luas. Ini memungkinkan pengembang menulis kode dengan lebih cepat dan lebih efektif. *VS Code* sempurna untuk membuat aplikasi *JavaScript* dari penulisan kode hingga tahap *deployment* karena memiliki fitur seperti *IntelliSense* untuk pelengkapan kode yang cerdas, kemampuan *debugging* yang handal, pemformatan otomatis, dan navigasi kode yang mudah digunakan [5].

III. METODE

A. Relatime Database Firebase

Realtime Database kita mempunyai dua properti yaitu *IFTPetFeeder1* dan *IFTPetFeeder2* dibedakan dengan token unik. Pertama, *IFTPetFeeder1* dengan token "TW01" memiliki catatan riwayat semua aktivitas pemberian makan kucing untuk interval *AllTime* dan *PerDay*. Ada juga *FoodPosition* dan *ImageCapture*. Fungsi-fungsi ini memungkinkan pengguna untuk mengetahui posisi makanan yang diberikan, serta menyimpan tangkapan gambar hewan peliharaan mereka saat waktu makan atau sedang berada disekitaran *pet feeder*.

Feeding pada *database* ini akan aktif jika checknya 1, pada *feeding* juga menyimpan takaran gram serta posisi makanan berdasarkan jenis makanannya. *Schedule* pada *database* ini untuk mengelola dan menyimpan *task* yang di input pengguna. Terakhir, *storage* pada *database* juga menyimpan berapa persen makanan yang ada dalam kedua penampungan *snack* maupun *staple*. *IFTPetFeeder2* memiliki tujuan dan fungsi yang serupa, beroperasi dengan Token "PO20".

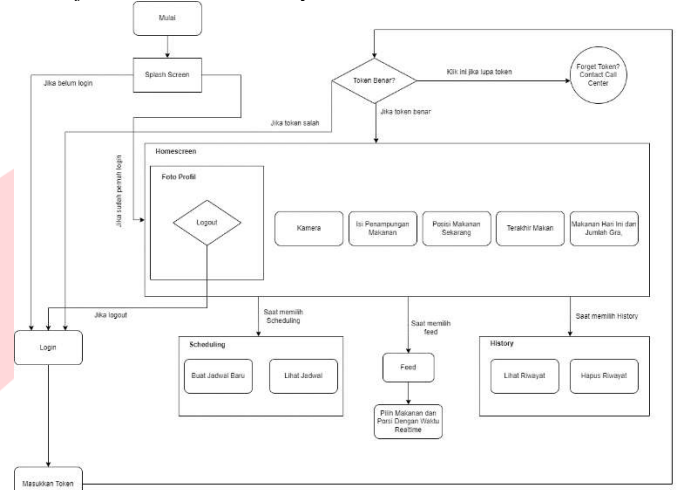


GAMBAR 1. 1
Realtime Database

B. Statechart

Statechart menunjukkan alur navigasi dan transisi dalam aplikasi *IFT Pet Feeder*. Proses ini dimulai dari membuka aplikasi, kemudian ada tampilan *Splash Screen* dan dilanjutkan ke halaman *login*.

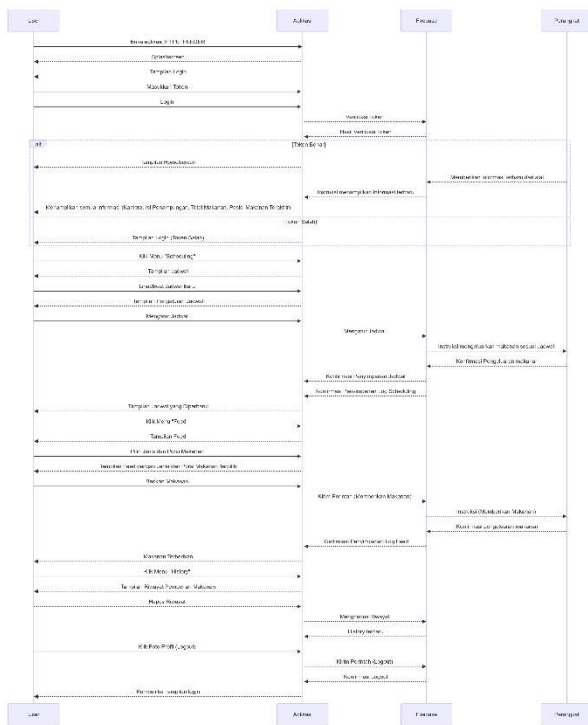
Pada halaman *login* pengguna diminta untuk memasukkan token yang mereka miliki. Setelah *login* atau pengguna sudah pernah *login* nantinya akan langsung diarahkan ke menu utama, disana pengguna dapat mengakses berbagai fitur pada *menu homescreen*, *menu Scheduling*, *menu feed* dan *menu history*.



GAMBAR 1. 2
Statechart

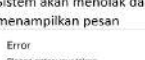
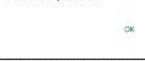
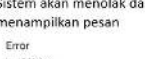
C. Sequence Diagram

Sequence diagram ini menggambarkan alur interaksi antara pengguna, aplikasi, *firebase*, dan perangkat *IFT Pet Feeder*. Pengguna memulai dengan membuka aplikasi, *login* dengan memasukkan token yang diverifikasi oleh *Firebase*. Pengguna dapat mengakses berbagai fitur seperti membuat jadwal, memberikan makanan, dan melihat riwayat pemberian makan, dengan setiap tindakan dikonfirmasi oleh *firebase* atau perangkat. Setelah selesai, pengguna dapat keluar dari aplikasi maupun juga memilih tetap aktif pada aplikasi.



GAMBAR 1.3
Sequence Diagram

TABEL 1. 1
Pengujian *Login*

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Token tidak diisi kemudian klik tombol Login	Token: ()		Sesuai Harapan	Valid
2	Mengetikkan token tidak sesuai, kemudian klik tombol Login	Token: (ty01)		Sesui Harapan	Valid
3	Mengetikkan token sesuai, kemudian klik tombol Login	Token: (1W01)		Sesuai Harapan	Valid

2. Pengujian *Scheduling*

Tujuan dari pengujian ini untuk memvalidasi fungsi scheduling sistem dan memastikan bahwa sistem dapat menangani berbagai kombinasi input pengguna dimana diharapkan sistem dapat menolak permintaan yang tidak lengkap atau tidak sesuai.

Berikut Tampilan *Scheduling* yang akan diuji pada beberapa skenario seperti tabel dibawah ini:

IV. HASIL DAN PEMBAHASAN

A. Blacbox Testing

Metode Blackbox menguji aplikasi untuk memenuhi kebutuhan pengguna. Ini adalah jenis pengujian yang tidak membutuhkan pemahaman tentang struktur internal kode program dan berfokus pada respons aplikasi atau program terhadap tindakan pengguna. Dilakukan secara bertahap menurut skenario yang telah ditentukan dan terdokumentasi [6].

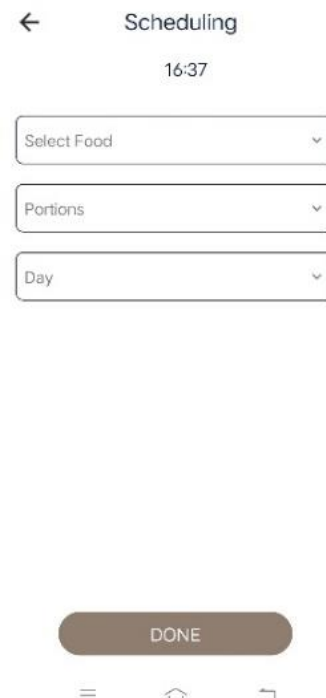
1. Pengujian *Login*

Tujuan dari pengujian ini adalah untuk memastikan respons sistem dalam berbagai situasi ketika pengguna mencoba masuk menggunakan token.

Berikut Tampilan login yang akan diuji pada beberapa skenario seperti tabel dibawah ini:



GAMBAR 1.4
Tampilan *Login*



GAMBAR 1. 5
Tampilan *Scheduling*

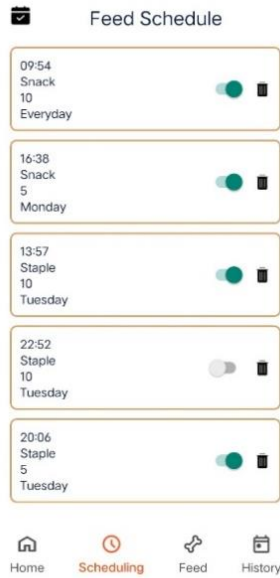
TABEL 1. 2
Pengujian *Scheduling*

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memilih jenis makanan tanpa memilih porsi, hari, dan tidak memilih waktu spesifik (melainkan memilih menggunakan waktu tersedia) lalu mengklik tombol "Done"	Sistem akan menolak dan menampilkan pesan <i>Please select food, portion, and day!</i>	Sesuai Harapan	Valid
2	Tidak memilih jenis makanan dan tidak memilih waktu spesifik (melainkan memilih menggunakan waktu tersedia) tetapi memilih porsi dan hari lalu mengklik tombol "Done"	Sistem akan menolak dan menampilkan pesan <i>Please select food, portion, and day!</i>	Sesuai Harapan	Valid
3	Memilih jenis makanan dan porsi, namun tidak memilih hari dan tidak memilih waktu spesifik (melainkan memilih menggunakan waktu tersedia) lalu mengklik tombol "Done"	Sistem akan menolak dan menampilkan pesan <i>Please select food, portion, and day!</i>	Sesuai Harapan	Valid
4	Memilih jenis makanan, memilih porsi, memilih hari, memilih waktu dan kemudian mengklik tombol "Done"	Sistem menerima akses penjadwalan dan kemudian menampilkan halaman hasil penjadwalan	Sesuai Harapan	Valid
5	Memilih jenis makanan, memilih porsi, memilih hari, memilih waktu dan kemudian mengklik tombol "Done" sebanyak 2 kali	Sistem menerima akses penjadwalan dan kemudian menampilkan halaman hasil penjadwalan yang terinput hanya satu	Sesuai Harapan	Valid

3. Pengujian Hasil *Scheduling*

Tujuan dari pengujian ini sistem mampu menerima akses dan mengubah status jadwal sesuai dengan permintaan pengguna.

Berikut Tampilan Hasil *Scheduling* yang akan diuji pada beberapa skenario seperti tabel dibawah ini:



GAMBAR 1. 6
Tampilan Hasil *Scheduling*

TABEL 1. 3
Pengujian Hasil *Scheduling*

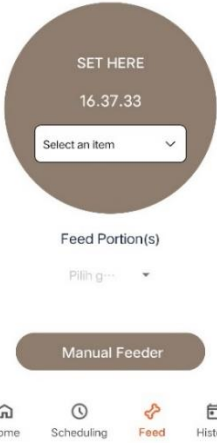
No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Pengguna memilih jadwal yang sudah ada dalam daftar jadwal mereka, lalu mengklik ikon sampah atau tombol "Hapus" untuk menghapus jadwal.	Sistem menerima akses dan menghapus jadwal	Sesuai Harapan	Valid
2	Pengguna memilih jadwal yang sudah ada dalam daftar jadwal mereka, lalu mengklik ikon switch untuk mengubah statusnya (aktif/tidak aktif).	Sistem menerima akses dan mengubah status jadwal sesuai permintaan pengguna.	Sesuai Harapan	Valid

4. Pengujian *Manual Feeder*

Tujuan pengujian ini adalah untuk memastikan bahwa sistem dapat mengendalikan dan menanggapi permintaan pengguna dengan benar saat pengguna menggunakan fitur pengumpan manual.

Berikut Tampilan *Manual Feeder* yang akan diuji pada beberapa skenario seperti tabel dibawah ini:

Manual Feeder



GAMBAR 1. 7
Tampilan *Manual Feeder*

TABEL 1. 4
Pengujian *Manual Feeder*

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Pengguna memilih jenis makanan dan porsi dan klik manual feeder.	Sistem menerima akses dan mengeluarkan makanan.	Sesuai Harapan	Valid
2	Pengguna tidak memilih jenis makanan, tetapi memilih porsi dan klik manual feeder.	Sistem akan menolak dan menampilkan pesan <i>Pilih jenis makanan terlebih dahulu!</i>	Sesuai Harapan	Valid
3	Pengguna memilih jenis makanan, tetapi tidak memilih porsi dan klik manual feeder.	Sistem akan menolak dan menampilkan pesan <i>Pilih jumlah porsi terlebih dahulu!</i>	Sesuai Harapan	Valid
4	Pengguna tidak memilih jenis makanan dan juga tidak memilih porsi dan klik manual feeder.	Sistem akan menolak dan menampilkan pesan <i>Pilih jenis makanan terlebih dahulu!</i>	Sesuai Harapan	Valid

5. Pengujian *History*

Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem dapat mengelola penghapusan riwayat dengan benar.

Berikut Tampilan *History* yang akan diuji pada beberapa skenario seperti tabel dibawah ini:



GAMBAR 1. 8
Tampilan *History*

TABEL 1. 5
Pengujian *History*

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Pengguna secara tidak sengaja mengklik ikon sampah atau tombol "Hapus Semua Riwayat".	Sistem menampilkan pesan konfirmasi untuk memastikan bahwa pengguna benar-benar bermaksud menghapus semua riwayat. Konfirmasi Yakin ingin menghapus semua riwayat? BATAL YA	Sesuai Harapan	Valid
2	Pengguna mengklik ikon sampah atau tombol "Hapus Semua Riwayat" di dalam fitur history dan memilih menghapus.	Sistem menerima akses dan menghapus semua riwayat.	Sesuai Harapan	Valid

6. Pengujian *Logout*

Tujuan pengujian ini adalah untuk memastikan bahwa sistem aplikasi dapat mengelola proses logout secara akurat terhadap tindakan logout yang tidak disengaja maupun disengaja.

TABEL 1. 6
Pengujian *Logout*

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Pengguna secara tidak sengaja mengklik ikon profil untuk "Logout".	Sistem menampilkan pesan konfirmasi sebelum melakukan logout. Logout Are you sure you want to logout? Logout Cancel	Sesuai Harapan	Valid
2	Pengguna mengklik ikon sampah atau ikon profil untuk "Logout" dan memilih logout.	sistem menghapus sesi login dan mengarahkan pengguna ke halaman login.	Sesuai Harapan	Valid

V. KESIMPULAN

Pengembangan aplikasi *IFT (Intelligent Food Timer)* menunjukkan betapa pentingnya penggunaan teknologi dalam perawatan hewan peliharaan. Aplikasi ini menawarkan pengalaman manajemen jadwal makan hewan peliharaan yang responsif dan dapat diandalkan dengan backend yang solid dan antarmuka pengguna yang mudah digunakan. *IFT* telah mewakili kemajuan besar dalam hal kesejahteraan hewan peliharaan melalui pemanfaatan teknologi canggih, meskipun masih ada beberapa masalah teknis, seperti tidak adanya informasi tentang jadwal makan atau jumlah makanan yang telah habis berupa notifikasi secara *realtime*.

REFERENSI

[1] K.T. Hanna, "Mobile App". Internet: <https://www.techtarget.com/whatis/definition/mobile-app>, [Accessed: Jul. 7, 2024].

[2] React Native. "Learn once, write anywhere". Internet: <https://reactnative.dev/>, [July 7, 2024].

[3] Firebase. "Google's Mobile and Web App Development Platform". Internet: <https://firebase.google.com/>, [July 7, 2024].

[4] Vysor. "Vysor". Internet: <https://www.vysor.io/>, [July 7, 2024].

[5] "Documentation for Visual Studio Code". Internet: <https://code.visualstudio.com/docs>, [July 7, 2024].

[6] Telkom University. "User Acceptance Test (UAT)". Internet: <https://fit.labs.telkomuniversity.ac.id/user-acceptance-test-uat/>, [Accessed: Jul. 7, 2024]