

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Bab ini menjelaskan hasil penelitian yang telah dilakukan, meliputi spesifikasi perangkat lunak yang digunakan, tampilan antarmuka sistem setelah proses deployment, struktur database, serta berbagai pengujian dan analisis yang dilakukan. Setiap bagian akan diuraikan secara terperinci untuk memberikan pemahaman yang menyeluruh mengenai implementasi dan kinerja sistem yang telah dikembangkan.

4.1.1 Spesifikasi *Perangkat lunak* dan *Perangkat keras* yang Digunakan

Dibawah ini merupakan spesifikasi *perangkat lunak* dan *perangkat keras* yang digunakan dalam penelitian ini :

4.1.1.1 Spesifikasi *Perangkat Keras*

Penelitian ini dilakukan menggunakan perangkat keras dengan spesifikasi berikut:

- a. Prosesor: Intel Core I3 1115G4, prosesor hemat daya dengan dua inti dan kecepatan hingga 4.1 GHz, cukup untuk menangani pemrosesan data dari sensor IoT, menjalankan skrip otomatisasi, dan komunikasi jaringan secara efisien.
- b. RAM: 24 GB, kapasitas memori yang cukup untuk menjalankan beberapa layanan IoT secara bersamaan.
- c. Penyimpanan: SSD 466 GB, memberikan kecepatan akses data yang tinggi dan waktu booting yang cukup cepat, mendukung penyimpanan data log dari perangkat IoT serta pengolahan data lokal sebelum dikirim ke cloud atau server pusat.

4.1.1.2 Spesifikasi *Perangkat lunak*

Perangkat lunak yang digunakan dalam penelitian ini meliputi :

- a. *Arduino 1.8.10* : *Arduino* merupakan platform *open-source* yang menggunakan *mikrokontroler* sebagai inti utamanya dan dirancang untuk membangun proyek elektronik yang bersifat interaktif. Sistem ini terdiri dari perangkat keras berupa papan sirkuit (contohnya *Arduino Uno*) dan perangkat lunak bernama *Arduino IDE* untuk pemrogramannya. *Arduino* dikenal mudah digunakan, baik oleh pemula maupun ahli, dalam mengembangkan berbagai sistem otomatisasi seperti robot, sensor, dan perangkat berbasis *Internet of Things (IoT)*.
- b. *Visual Studio Code* : *Visual Studio Code (VS Code)* merupakan editor kode sumber yang tersedia secara gratis dan bersifat *open-source*, dikembangkan oleh Microsoft. Editor ini ringan namun memiliki performa tinggi, mendukung berbagai bahasa pemrograman, dan dilengkapi dengan fitur seperti debugging, IntelliSense (rekomendasi kode otomatis), integrasi Git, serta beragam ekstensi untuk menunjang produktivitas developer.
- c. *Expo* : *Expo* merupakan platform *open-source* yang memungkinkan pengembangan aplikasi mobile menggunakan *React Native*. Melalui *Expo*, developer dapat membangun aplikasi untuk Android dan iOS hanya dengan satu kode dasar berbasis JavaScript, tanpa perlu melakukan konfigurasi lingkungan native secara langsung. Platform ini juga menyediakan berbagai alat pendukung seperti *Expo Go*, koleksi API siap pakai (seperti kamera, lokasi, dan notifikasi), serta fitur hot-reload yang membantu mempercepat proses pembuatan dan pengujian aplikasi.

4.1.2 Tampilan Interface dan Hasil Deploy

Berikut adalah tampilan interface dari hasil deploy

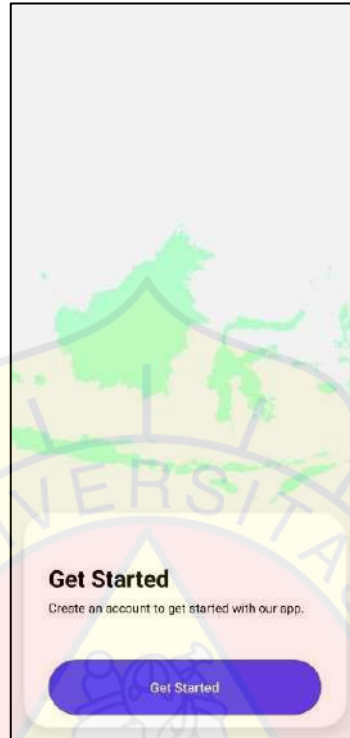
- a. Gambar logo aplikasi



Tabel 4. 1 Logo Aplikasi

Logo pada gambar 4.2 menampilkan singkatan "IFD" yang berasal dari *IoT Fall Detection*, dengan keterangan lengkap terletak di bawahnya. Latar belakang berupa gradasi warna hijau ke biru tua memberi kesan profesional, modern, dan erat kaitannya dengan dunia teknologi serta kesehatan. Aplikasi ini dirancang sebagai sistem otomatis berbasis IoT untuk mendeteksi kejadian jatuh, sangat bermanfaat bagi lansia atau individu dengan risiko jatuh. Secara keseluruhan, desain logo dibuat sederhana, informatif, dan menggambarkan fungsi utama aplikasi secara efektif.

b. Gambar Halaman Pembuka



Tabel 4. 2 Halaman Pembuka

Pada gambar 4.3 terdapat halaman awal pada aplikasi yang ditampilkan menunjukkan menu **“Get Started”** yang berfungsi untuk mengarahkan pengguna membuat akun dan mulai menggunakan layanan aplikasi. Tampilan didesain secara modern dan sederhana, dilengkapi latar peta Indonesia berwarna hijau muda, yang memberi kesan bahwa aplikasi ini menyasar pengguna di Indonesia. Komponen antarmuka seperti tulisan dan tombol ditampilkan secara minimalis, dengan tombol ungu mencolok bertuliskan **“Get Started”** yang mempermudah pengguna dalam memulai proses pendaftaran atau navigasi awal. Secara keseluruhan, tampilan ini menekankan kemudahan akses dan nuansa lokal aplikasi.

c. Gambar halaman Dashboard



Tabel 4. 3 Halaman Dashboard

Antarmuka aplikasi yang ditampilkan pada gambar 4.4 merupakan halaman utama dari *IoT Fall Detector*, yang ditujukan untuk memantau kondisi pengguna, terutama bagi mereka yang menggunakan kursi roda. Bagian atas layar menampilkan sapaan kepada pengguna serta informasi perangkat sensor yang sedang digunakan, seperti nama alat, status deteksi jatuh, dan pesan sistem. Gambar ilustratif pengguna kursi roda menegaskan bahwa aplikasi ini difokuskan pada keselamatan individu dengan keterbatasan mobilitas. Tersedia juga tombol “Location” yang mengindikasikan fitur pelacakan lokasi. Di bagian bawah, terdapat daftar notifikasi yang mencatat riwayat peringatan jatuh secara langsung, memudahkan pengguna maupun pendamping untuk segera mengambil tindakan.

Desain antarmuka ini menggambarkan sistem yang praktis, komunikatif, dan ramah pengguna.

d. Gambar Halaman Map



Tabel 4. 4 Halaman Map

Antarmuka menu aplikasi pada gambar 4.5 menampilkan kemampuan pemantauan posisi lansia secara langsung menggunakan teknologi sensor *GPS NEO-6M*. Penanda merah yang terlihat pada peta Google Maps menunjukkan lokasi aktual pengguna saat ini, yang dalam contoh ini berada di wilayah Jakarta. Fungsi ini memungkinkan anggota keluarga atau perawat untuk memantau keberadaan lansia dengan tepat dan responsif, terutama ketika terjadi kejadian darurat seperti terjatuh. Melalui pemanfaatan sensor *GPS NEO-6M*, sistem mampu menyediakan informasi lokasi yang lebih akurat, sehingga dapat meningkatkan aspek keamanan dan kemudahan dalam mengawasi lansia sepanjang waktu.

4.1.3 Struktur Database

a. Tabel Device

Tabel 4. 5 Tabel Database Device

Field	Tipe Data	Keterangan
id	Varchar (50)	ID unik dari Firebase
device_id	Varchar (100)	ID perangkat sensor
created_at	Timestamp	Waktu data perangkat pertama kali tercatat
last_ping	Timestamp	Waktu terakhir perangkat mengirimkan data (heartbeat)
update_at	Timestamp	Waktu terakhir data diperbarui
is_emergency	Boolean	Menandakan apakah saat itu kondisi darurat (true) atau tidak (false)
location_lat	Float	Koordinat lintang lokasi perangkat
location_long	Float	Koordinat bujur lokasi perangkat
message	Text	Status perangkat (contoh : "Potential fall detected but position returned to normal")
ultrasonic_distance	Float	Jarak objek terdeteksi oleh sensor ultrasonic
ultrasonic_unit	Varchar 3(10)	Satuan jarak ultrasonic, misalnya "cm"
sensor_status	Varchar (50)	Status sensor saat ini (misal :false_alarm, active, disconnected)

Tabel devices memiliki peran penting sebagai penyimpan data utama yang mencatat berbagai informasi terkait kondisi dan status terkini dari perangkat sensor berbasis IoT dalam sistem pemantauan lansia. Setiap baris pada tabel ini merepresentasikan satu perangkat aktif, seperti sensor yang terpasang pada alat bantu jalan atau kursi roda, yang dilengkapi dengan data identitas perangkat (`device_id`), waktu saat data dicatat atau diperbarui (`created_at`, `last_ping`, `updated_at`), serta penanda apakah perangkat sedang dalam kondisi darurat (`is_emergency`).

Tabel ini juga menyimpan informasi lokasi terakhir perangkat (`location_lat`, `location_long`), serta pesan sistem (`message`) yang membantu dalam menilai situasi atau mengidentifikasi gangguan. Data sensor ultrasonik, seperti jarak objek di sekitar lansia (`ultrasonic_distance`) dan satuannya (`ultrasonic_unit`), juga dicatat untuk mencegah potensi tabrakan atau jatuh. Kolom `sensor_status` memberikan gambaran umum tentang kondisi perangkat, apakah normal, alarm palsu, atau tidak aktif. Secara keseluruhan, tabel ini sangat berguna untuk kebutuhan pemantauan secara langsung, analisis perangkat, serta integrasi ke dalam sistem dashboard pemantauan berbasis IoT.

b. Tabel Notifications

Dibawah ini merupakan tabel notifications yang berfungsi sebagai penyimpanan utama notifikasi jatuh dari sensor IoT pada sistem pemantauan lansia.

Tabel 4. 6 Tabel Database Notifications

Field	Tipe Data	Keterangan
createdAt	Timestamp	Waktu saat data pertama kali dicatat atau dibuat di sistem.
device_id	Varchar (100)	ID perangkat sensor, contoh: wheelchair_sensor_01
accelMagnitude	Number	Magnitudo total dari akselerasi (gabungan X, Y, Z)
accel_x	Number	Komponen akselerasi X
accel_y	Number	Komponen akselerasi Y
accel_z	Number	Komponen akselerasi Z
status	String	Status kejadian
titl_angel	Number	Sudut kemiringan tubuh saat kejadian
isRead	Boolean	Menunjukkan apakah data sudah dibaca (true/false)
latitude	Number	Lokasi lintang kejadian
longtitude	Number	Lokasi bujur kejadian
obstacle_distance	Number	Jarak ke objek terdekat
obstacle_unit	String	Satuan jarak (cm)
status	String	Pengiriman data
updateAt	Timestamp	Waktu saat data terakhir kali diperbarui.

Tabel notifications berfungsi sebagai media utama penyimpanan data insiden jatuh yang terdeteksi oleh perangkat sensor IoT seperti MPU-6050 dan HC-SR04, yang umumnya dipasang pada sabuk sensor atau kursi roda lansia. Setiap entri mencatat informasi penting seperti ID perangkat, waktu kejadian, data percepatan (accel_x, accel_y, accel_z, accel_magnitude), sudut kemiringan (tilt_angle), status kejadian, lokasi jatuh (latitude, longitude), jarak ke objek terdekat (obstacle_distance, obstacle_unit), dan status notifikasi (is_read). Tabel ini mendukung pemantauan real-time, analisis kejadian, serta membantu keluarga atau pengasuh merespons cepat, dan dapat diintegrasikan ke dalam dashboard untuk visualisasi, pelaporan medis, dan pengembangan sistem deteksi jatuh yang lebih cerdas.

4.2 Analisis Hasil

4.2.1 Percobaan Input – Output

Percobaan dilakukan dengan memberikan input berupa melakukan uji coba alat secara langsung. Dari hasil uji coba tersebut hasil menunjukkan bahwa :

- a. Terdapat delay rata – rata sekitar 10,7 detik kemudian alarm peringatan berbunyi baik di alat dan di smartphone pengguna setelah dilakukan uji coba sebanyak 20 kali percobaan.
- b. Sensor ultrasonik akan memicu bunyi *buzzer* dan lampu merah akan menyala apabila terdapat benda didepannya berjarak 30 cm.

4.2.2 Testing Hasil

Pada bagian ini merupakan testing hasil yang dilakukan pada penelitian ini dengan menguji beberapa sensor yang terdapat pada alat.

4.2.2.1 Pengujian Sensor MPU6050

- a. Sistem dimiringkan secara bertahap sampai melewati 40 derajat.
- b. Diuji apakah LED dan *buzzer* aktif.
- c. Dicek apakah data terkirim ke API.

4.2.2.2 Pengujian Sensor Ultrasonik

- a. Objek didekatkan ke depan sensor pada jarak < 30 cm.
- b. Dicek apakah *buzzer* berbunyi dan LED merah menyala.

4.2.2.3 Pengujian GPS

- a. Sistem diletakkan di luar ruangan.
- b. Dicek apakah latitude dan longitude muncul dan dikirim.

4.2.2.4 Pengujian WiFi dan API

- a. Dicek apakah data JSON terkirim ke server.
- b. Digunakan aplikasi seperti Postman atau Firebase Console untuk memverifikasi data masuk.

Tabel 4. 7 Hasil Pengujian

No	Skema Pengujian	Kondisi Input	Hasil Diharapkan	Hasil Aktual
1	Sensor jatuh (kemiringan)	Sudut > 40°	LED merah dan <i>buzzer</i> ON, kirim data	Sesuai
2	Sensor tidak jatuh	Sudut < 20°	Tidak ada alarm	Sesuai
3	Sensor ultrasonik aktif	Objek di < 30 cm	<i>Buzzer</i> bunyi pendek	Sesuai
4	GPS aktif di luar ruangan	Modul NEO-6M dapat sinyal	Data lokasi tampil di Firebase	Sesuai
5	Koneksi WiFi	WiFi aktif	Terhubung dan kirim data	Sesuai

4.2.3 Modifikasi atau Optimalisasi dari Sistem Terdahulu

Modifikasi dan Optimalisasi yang dilakukan meliputi :

- a. Penambahan fitur berupa deteksi benda di depan lansia dengan menambahkan sensor ultrasonik HC-SR04.
- b. Penambahan fitur berupa deteksi alat dengan menggunakan GPS Neo-6M untuk mendeteksi lokasi lansia.

4.2.4 Proses Deploy Sistem Aplikasi

Dibawah ini merupakan beberapa proses yang dilakukan untuk deploy aplikasi sistem

4.2.4.1 Persiapan Perangkat Keras (Hardware)

Komponen yang terdapat pada alat penelitian ini meliputi :

- a. NodeMCU ESP32
- b. Sensor MPU-6050 (*akselerometer* dan *gyroscope*)
- c. Sensor *Ultrasonic* HC-SR04
- d. GPS Neo-6m
- e. Buzzer
- f. 3 LED indicator
- g. Baterai Li-Po 6800mAh dan modul charger TP5100

Dibawah ini merupakan langkah – langkah yang dilakukan dalam perakitan alat dalam penelitian ini adalah :

- a. Hubungkan semua sensor ke pin ESP32 sesuai dengan desain rangkaian yang telah dibuat.
- b. Rakit seluruh komponen di atas pada casing wearable.
- c. Pastikan daya dari baterai dan charger modul bekerja stabil.

4.2.4.2 Pemrograman Mikrokontroler (Firmware)

Tools yang digunakan dalam penelitian ini adalah Arduino IDE, langkah - langkah pemrograman yaitu :

- a. Gunakan Arduino IDE 1.8.10 untuk menulis program.
- b. Integrasikan library seperti Wire.h, MPU6050.h, dan WiFi.h.
- c. Program fungsi:
 - 1) Deteksi jatuh berdasarkan nilai X/Y/Z dari sensor MPU-6050.
 - 2) Deteksi benda di sekitar dengan HC-SR04.
 - 3) Pembacaan koordinat dari GPS Neo-6m.
 - 4) Notifikasi dan pengiriman data ke cloud/Firebase.
- d. Upload program ke NodeMCU ESP32 melalui kabel USB.

4.2.4.3 Pengembangan Aplikasi Mobile

Tools yang digunakan dalam project ini untuk pengembangan aplikasi mobile adalah Expo dan React Native. Langkah – langkah yang dilakukan adalah :

- a. Kembangkan UI/UX notifikasi seperti: “Peringatan! Lansia Terjatuh”, Lokasi koordinat pada peta (Google Maps)
- b. Gunakan Expo untuk mem-build aplikasi ke Android/iOS.
- c. Integrasikan aplikasi dengan Firebase untuk mendapatkan data real-time dari perangkat.

4.2.4.4 Setup Database dan backend

Tools yang digunakan dalam project ini untuk pengembangan aplikasi mobile adalah firebase firestore. Langkah – langkah yang dilakukan adalah :

- a. Buat dua koleksi utama:
 - 1) devices: menyimpan status sensor, lokasi, status darurat, dan jarak ultrasonic.
 - 2) notifications: menyimpan log insiden jatuh, nilai percepatan, kemiringan, lokasi.
- b. Gunakan Firebase SDK untuk membaca dan menampilkan data secara real-time di aplikasi.

4.2.4.5 Deploy ke Pengguna Akhir

Langkah – langkah yang perlu dilakukan pada tahap deploy ke pengguna akhir adalah :


- a. Untuk *perangkat keras*: tempatkan alat pada sabuk pinggang atau pakaian lansia.
- b. Untuk aplikasi: install melalui APK.
- c. Pastikan Wi-Fi atau koneksi GSM tersedia agar ESP32 dapat mengirimkan data.




4.2.4.6 Uji Coba dan Validasi Sistem

Tahap uji coba dan validasi sistem merupakan langkah akhir dalam proses pengembangan alat deteksi jatuh berbasis IoT untuk lansia. Pada tahap ini, sistem diuji dengan melakukan simulasi kejadian jatuh untuk memastikan apakah alat mampu mendeteksi insiden tersebut secara tepat dan akurat. Pengujian dilakukan dengan mensimulasikan skenario nyata, termasuk posisi jatuh pengguna serta keberadaan benda di sekitar, untuk melihat respon dari sensor seperti MPU-6050, HC-SR04, dan modul GPS.

Selama pengujian, dilakukan pula integrasi antara sensor dan layanan cloud Firebase guna memastikan data dapat dikirim dan diterima secara real-time. Validasi dilakukan dengan mencocokkan output sistem, seperti aktivasi buzzer, LED indikator, serta pengiriman koordinat lokasi ke aplikasi mobile, dengan kondisi input yang telah dirancang. Hasilnya menunjukkan bahwa sistem bekerja secara konsisten sesuai dengan yang diharapkan, baik dari sisi perangkat keras maupun perangkat lunak. Uji coba ini membuktikan bahwa sistem tidak hanya mampu mengenali kejadian jatuh secara akurat, tetapi juga mampu mengirimkan notifikasi secara cepat kepada pihak yang berkepentingan, sehingga dapat meningkatkan keselamatan dan kualitas hidup lansia secara signifikan. Gambar ini bawah ini merupakan hasil uji coba yang dilakukan terhadap 9 lansia di lingkungan RW. 09 :

Tabel 4. 8 Uji Coba Pada Lansia

No	RT	Nama Lansia	Hasil Ujicoba	Respon Sensor	Bukti Foto
1	RT 01	Ibu Ramot Hutapea	Berfungsi dengan baik	Sensor Terdeteksi	

2	RT 02	Ibu Parni	Berfungsi dengan baik	Sensor Terdeteksi	
3	RT 03	Bapak Masturi	Berfungsi dengan baik	Sensor Terdeteksi	
4	RT 04	Bapak Rachman	Berfungsi dengan baik	Sensor Terdeteksi	

5	RT 05	Ibu Sarmiyah	Berfungsi dengan baik	Sensor Terdeteksi	
6	RT 06	Bapak Sidik	Berfungsi dengan baik	Sensor Terdeteksi	
7	RT 07	Ibu Atih	Berfungsi dengan baik	Sensor Terdeteksi	

8	RT 08	Ibu Saking	Berfungsi dengan baik	Sensor Terdeteksi	
9	RT 09	Bapak Wakino	Berfungsi dengan baik	Sensor Terdeteksi	

Tabel 4. 9 Hasil Pengujian Akurasi sistem

Skenario	Jumlah Percobaan	Deteksi Akurat	Deteksi Tidak Akurat	Akurasi
Jatuh ke Depan	10	8	2	80%
Jatuh ke Belakang	10	9	1	90%
Jatuh ke Kanan	10	7	3	70%
Jatuh ke Kiri	10	8	2	80%

Pengujian akurasi sistem deteksi jatuh dilakukan pada empat skenario arah jatuh, yaitu jatuh ke depan, jatuh ke belakang, jatuh ke kanan, dan jatuh ke kiri, masing-masing dengan sepuluh kali percobaan. Berdasarkan hasil pengujian yang ditampilkan pada Tabel 4.5, sistem menunjukkan variasi tingkat akurasi untuk setiap arah jatuh.

Pada skenario jatuh ke belakang, sistem mampu mendeteksi dengan akurasi tertinggi, yaitu 90% (9 deteksi benar dari 10 percobaan). Jatuh ke kiri dan jatuh ke depan memiliki akurasi sama sebesar 80%, masing-masing dengan 8 deteksi benar dan 2 deteksi salah. Sementara itu, skenario jatuh ke kanan menunjukkan akurasi terendah, yaitu 70% (7 deteksi benar dari 10 percobaan).

Perbedaan akurasi ini diperkirakan disebabkan oleh variasi sudut kemiringan tubuh pada saat jatuh. Arah jatuh ke kanan cenderung menghasilkan sudut yang mendekati batas ambang deteksi sensor accelerometer, sehingga beberapa kejadian tidak teridentifikasi sebagai jatuh. Hal ini menunjukkan bahwa posisi dan orientasi sensor berpengaruh terhadap kemampuan sistem dalam mengenali kejadian jatuh secara konsisten.