

**LAPORAN PROYEK AKHIR**  
**SISTEM EMBEDDED BIOMEDIK**

*IoT-Based Smart Haptic Navigational Belt: Integrated Multi-Sensor Spatial Perception and Real-Time Location Tracking System for Enhanced Mobility and Safety of the Visually Impaired*



Disusun oleh: Kelompok 2  
Crystaly (2306202523)  
Farhan Abdurrahman (2306225994)  
Shiva Aqila Finanda (2306267214)

**PROGRAM STUDI SARJANA TEKNIK BIOMEDIK**  
**DEPARTEMEN TEKNIK ELEKTRO**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS INDONESIA**  
**2025**

## ABSTRAK

Kemandirian mobilitas merupakan salah satu aspek penting dalam kualitas hidup penyandang tunanetra, namun alat bantu konvensional seperti tongkat masih memiliki keterbatasan karena hanya mampu mendeteksi rintangan pada area permukaan tanah dan menyisakan blind spot pada bagian tubuh atas. Proyek ini merancang dan merealisasikan sebuah *smart haptic navigational belt* berbasis sistem tertanam yang mengintegrasikan persepsi spasial multi-sensor dan fitur keselamatan berbasis IoT untuk mendukung orientasi ruang serta meningkatkan keamanan pengguna. Sistem dikembangkan dengan memanfaatkan tiga sensor jarak HC-SR04 ultrasonik sebagai deteksi rintangan arah kiri–tengah–kanan, sensor inersia MPU6050 untuk pendekslsian indikasi jatuh, serta modul GPS NEO-6M sebagai pelacak lokasi darurat. Unit pemrosesan ESP32 digunakan sebagai pengendali utama yang mengolah data sensor secara periodik dan menerapkan logika *sensor fusion* sederhana, sementara umpan balik navigasi diberikan melalui motor getar dengan intensitas yang dipetakan secara proporsional terhadap jarak objek. Mekanisme keselamatan dirancang dalam dua mode, yaitu pemicu darurat manual melalui tombol fisik SOS dan notifikasi otomatis ketika terdeteksi perubahan postur ekstrem, di mana keduanya dikirimkan melalui platform Blynk dan diteruskan ke kontak terdekat pengguna melalui Telegram. Prototipe ditenagai menggunakan powerbank USB 5V untuk mendukung portabilitas dan kemudahan penggunaan. Sistem ditargetkan mampu beroperasi pada rentang deteksi 5-200 cm dengan respons getaran berlatensi rendah dan ketahanan operasi yang memadai untuk penggunaan mobilitas harian. Secara keseluruhan, prototipe ini diposisikan sebagai solusi alternatif di antara tongkat konvensional berbiaya rendah dan perangkat navigasi komersial berharga tinggi, dengan potensi meningkatkan aspek keselamatan, rasa percaya diri, dan kemandirian mobilitas pengguna tunanetra.

**Kata kunci:** wearable assistive technology, haptic navigation, Time-of-Flight sensor, sensor fusion, ESP32, visual impairment.

## DAFTAR ISI

<b>ABSTRAK.....</b>	<b>1</b>
<b>DAFTAR ISI.....</b>	<b>2</b>
<b>BAB I</b>	
<b>PENDAHULUAN.....</b>	<b>6</b>
1.1 Latar Belakang Masalah.....	6
1.2 Tujuan Proyek.....	6
1.3 Manfaat Sistem.....	7
<b>BAB II.....</b>	<b>8</b>
<b>DESAIN DAN ARSITEKTUR SISTEM.....</b>	<b>8</b>
2.3 Desain Perangkat Lunak.....	12
2.3.1 Alur Logika Sistem.....	12
2.3.2 Logika Navigasi Adaptif (Haptic Feedback).....	14
2.3.3 Logika Deteksi Jatuh dan Pelacakan Lokasi.....	14
<b>BAB III.....</b>	<b>15</b>
<b>IMPLEMENTASI PERANGKAT KERAS.....</b>	<b>15</b>
3.1 Pemilihan Komponen & Justifikasi.....	15
3.2 Integrasi Hardware.....	16
<b>BAB IV.....</b>	<b>17</b>
<b>IMPLEMENTASI PERANGKAT LUNAK.....</b>	<b>17</b>
4.1 Struktur Kode dan Pustaka.....	17
4.2 Algoritma Akuisisi Data Sensor.....	18
4.3 Algoritma Navigasi Haptic.....	18
4.4 Algoritma Deteksi Jatuh.....	19
4.4.1 Perhitungan Sudut (Trigonometri).....	19
4.4.2 Logika Keputusan dan Safety Startup.....	20
4.5 Implementasi IoT dan GPS Asynchronous.....	20
<b>BAB V.....</b>	<b>23</b>
<b>ANTARMUKA PENGGUNA.....</b>	<b>23</b>
5.1 Antarmuka Fisik.....	23
5.1.1 Desain Interaksi Navigasi (Haptic Feedback).....	23
5.1.2 Mekanisme Kontrol GPS secara Fisik.....	24
5.2 Antarmuka Digital.....	24
5.2.1 Dashboard Monitoring (Blynk).....	24
5.2.2 Sistem Notifikasi Peringatan (Telegram).....	25
<b>BAB VI.....</b>	<b>28</b>
<b>PENGUJIAN DAN ANALISIS.....</b>	<b>28</b>
6.1 Skenario Pengujian.....	28
6.1.1. Tes jarak sensor ultrasonik dan respons indikator arah (T1).....	29
6.1.2 Tes akurasi deteksi jatuh (fall detection) berbasis MPU6050 (T2).....	30
6.1.3 Tes GPS dan latensi notifikasi lokasi via Telegram (T3).....	31
6.2 Hasil Pengujian.....	32
6.3 Analisis Kendala & Perbaikan (Debugging).....	33

<b>BAB VII.....</b>	<b>35</b>
7.1 Kesimpulan.....	35
7.2 Pengembangan Masa Depan.....	35
<b>DAFTAR PUSTAKA.....</b>	<b>37</b>
<b>LAMPIRAN.....</b>	<b>38</b>

## **DAFTAR GAMBAR**

Gambar 2.1. Blok Diagram Arsitektur Sistem Smart Belt.....	8
Gambar 2.2. Skematik Rangkaian Elektronik Smart Belt.....	10
Gambar 2.3. Desain Layout PCB.....	11
Gambar 2.4. Desain Mekanik Smart Belt dan Penempatan Komponen.....	12
Gambar 2.5. Diagram Alur Logika Utama (Main Loop Architecture).....	14
Gambar 4.1. Grafik Hubungan Jarak Halangan terhadap Intensitas PWM Motor.....	19
Kode Program 4.2. Logika Deteksi Jatuh dengan Pengaman.....	20
Kode Program 4.3. Logika Update Status GPS Berkala.....	21
Gambar 4.2. Diagram Sekuensial Pengiriman Pesan GPS dan Lokasi Susulan.....	22
Gambar 5.1. Implementasi Antarmuka Fisik pada Perangkat Smart Belt.....	24
Gambar 5.2. Tampilan Dashboard Monitoring pada Situs Web Blynk.....	25
Gambar 5.3. Format Pesan Peringatan dan Pelacakan Lokasi pada Telegram.....	27
Gambar 6.1 QR-Code video presentasi.....	32
Gambar 6.2 (a) Kondisi ketika sudut berada di bawah 20 derajat (keadaan aman), (b) Kondisi ketika sudut melebihi 20 derajat dan kondisi berbahaya.....	33

## **DAFTAR TABEL**

Tabel 2.1. Konfigurasi Pin Mikrokontroler ESP32.....	9
Tabel 3.1. Spesifikasi Komponen Utama Sistem.....	15
Tabel 4.1. Daftar Libraries dan Fungsinya.....	17
Tabel 5.1. Matriks Logika Interaksi Fisik Pengguna.....	23
Tabel 6.1 Rencana pengujian.....	28
Tabel 6.2 Matriks skenario uji jarak dan ekspektasi respons.....	30
Tabel 6.3 Skenario uji fall detection dan ekspektasi sistem.....	31
Tabel 6.4 Skenario uji SOS dan GPS (berdasarkan alur program).....	32

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang Masalah**

Tunanetra memiliki keterbatasan dalam mengakses informasi visual di lingkungan sekitar, sehingga aktivitas mobilitas sehari-hari sangat bergantung pada indra lain (pendengaran dan peraba). Kondisi ini membuat pengguna lebih rentan mengalami kecelakaan, terutama saat berjalan di area yang memiliki banyak hambatan atau perubahan permukaan lantai.

Beberapa masalah utama yang sering terjadi antara lain:

- Risiko menabrak halangan di depan maupun di sisi kiri/kanan, karena keterbatasan deteksi jarak dan arah hambatan secara cepat.
- Risiko jatuh yang tinggi, misalnya karena tersandung, terpeleset, atau kehilangan keseimbangan, dan kejadian ini sering tidak terdeteksi oleh keluarga/pendamping jika pengguna sedang berada di luar rumah atau sendirian.
- Keterlambatan penanganan saat darurat, karena pengguna kesulitan meminta bantuan secara cepat, sementara keluarga tidak mengetahui kondisi dan lokasi pengguna secara real-time.

Berdasarkan permasalahan tersebut, dibutuhkan sebuah alat bantu yang praktis, wearable, dan mampu memberikan peringatan hambatan serta fitur keselamatan seperti deteksi jatuh dan SOS, termasuk informasi lokasi agar pertolongan dapat diberikan lebih cepat.

#### **1.2 Tujuan Proyek**

Proyek ini bertujuan merancang sebuah *smart belt* berbasis *wearable* sebagai alat bantu mobilitas dan keselamatan bagi penyandang tunanetra. Secara khusus, tujuan yang ingin dicapai adalah sebagai berikut:

- Mengembangkan sistem navigasi jarak dekat menggunakan tiga sensor ultrasonik (kiri, tengah, kanan) untuk mendeteksi hambatan di sekitar pengguna.
- Menerapkan mekanisme peringatan multimodal yang berbeda sesuai arah hambatan, yaitu buzzer untuk sisi kiri, LED untuk arah depan, dan motor getar untuk sisi kanan.

- Mengimplementasikan fitur *fall detection* menggunakan sensor MPU6050 guna mengidentifikasi kondisi jatuh atau posisi berbahaya berdasarkan perubahan sudut kemiringan.
- Mengintegrasikan modul *GPS* (Neo GPS) untuk memperoleh koordinat lokasi pada kondisi darurat.
- Menghubungkan sistem dengan Blynk sebagai antarmuka pemantauan jarak hambatan, sudut kemiringan, status kondisi, serta kontrol *SOS*.
- Mengirim notifikasi melalui Telegram saat terjadi kondisi darurat (jatuh atau *SOS*), termasuk status pengguna dan tautan lokasi *Google Maps* yang dapat diakses langsung.

### 1.3 Manfaat Sistem

Sistem *smart belt* ini diharapkan memberikan manfaat baik bagi pengguna maupun pihak pendamping/keluarga. Manfaat utama yang ditargetkan meliputi:

- Membantu pengguna meningkatkan kemandirian mobilitas melalui deteksi hambatan tiga arah dan peringatan yang mudah dipahami.
- Mengurangi risiko kecelakaan saat berjalan, terutama tabrakan dengan objek di sekitar dan kejadian jatuh yang tidak terpantau.
- Mempercepat respons pertolongan karena sistem mampu mengirim peringatan otomatis saat *fall detection* aktif atau saat *SOS* ditekan.
- Memberikan *peace of mind* bagi keluarga melalui *GPS tracking* dan informasi lokasi yang dibagikan dalam bentuk tautan *Google Maps* melalui Telegram.
- Menyediakan pemantauan kondisi pengguna secara ringkas melalui Blynk, sehingga status “normal” atau “*fall*” dapat diketahui tanpa perlu berada di lokasi yang sama.

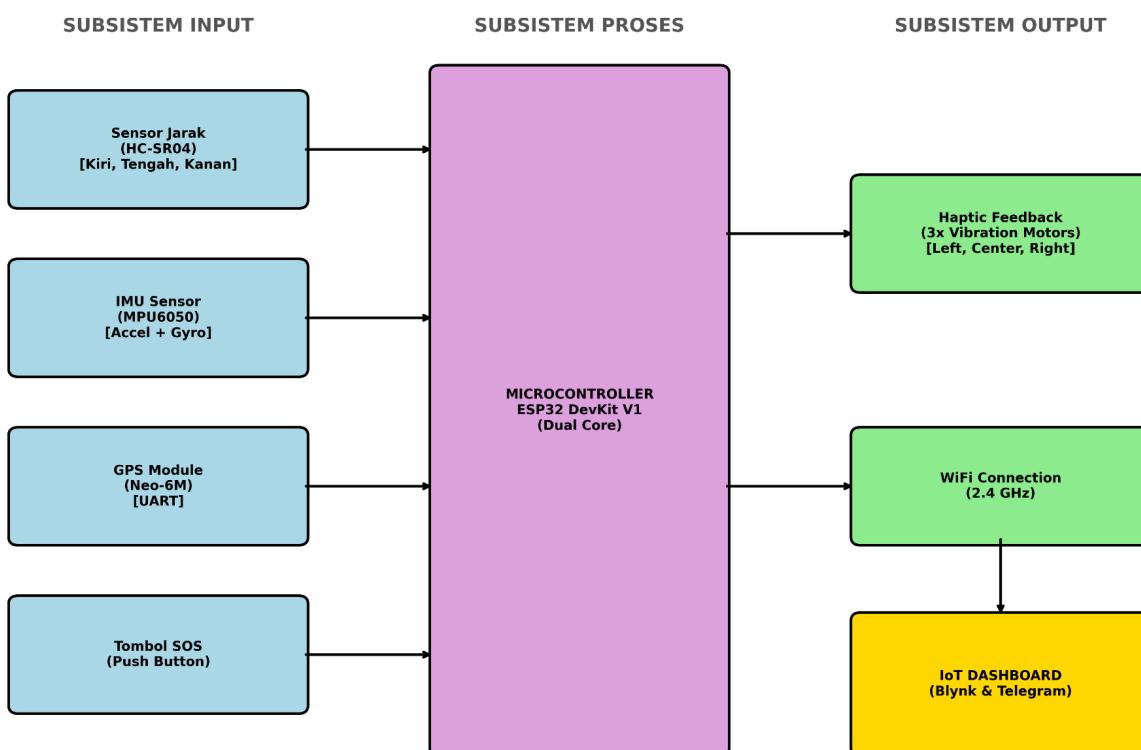
## BAB II

### DESAIN DAN ARSITEKTUR SISTEM

Bab ini membahas perancangan sistem Smart Belt, mencakup arsitektur sistem, perancangan perangkat keras, dan alur logika perangkat lunak. Pendekatan desain difokuskan pada aspek *real-time processing* untuk deteksi jatuh dan responsivitas sensor navigasi.

#### 2.1 Blok Diagram Sistem

Sistem dirancang dengan arsitektur *input-process-output* yang terintegrasi dengan modul komunikasi nirkabel. Gambar 2.1 memperlihatkan bagaimana mikrokontroler ESP32 berfungsi sebagai unit pemrosesan pusat yang mengakuisisi data dari sensor ultrasonik (HC-SR04) dan sensor inersia (MPU6050).



**Gambar 2.1.** Blok Diagram Arsitektur Sistem Smart Belt

Sesuai diagram di atas, sistem dibagi menjadi tiga subsistem:

1. Input: Terdiri dari sensor HC-SR04 untuk mendeteksi halangan dan modul MPU6050 (akselerometer & giroskop) untuk memantau orientasi serta percepatan linear pengguna. Modul GPS ditambahkan untuk akuisisi data geolokasi.

2. Proses: ESP32 dipilih karena memiliki arsitektur *dual-core* yang memungkinkan manajemen tugas antara pembacaan sensor dan komunikasi WiFi secara sekaligus tanpa *blocking* yang signifikan.
3. Output: Umpam balik haptik (*vibration motor*) digunakan sebagai peringatan navigasi kepada pengguna tunanetra, sementara notifikasi berbasis IoT dikirimkan ke platform Blynk dan Telegram untuk pemantauan jarak jauh.

## 2.2 Desain Perangkat Keras

Perancangan perangkat keras dilakukan dalam tiga tahap, yaitu skematik elektronik, tata letak PCB (Printed Circuit Board), dan desain mekanik (*casing*).

### 2.2.1 Skematik Rangkaian

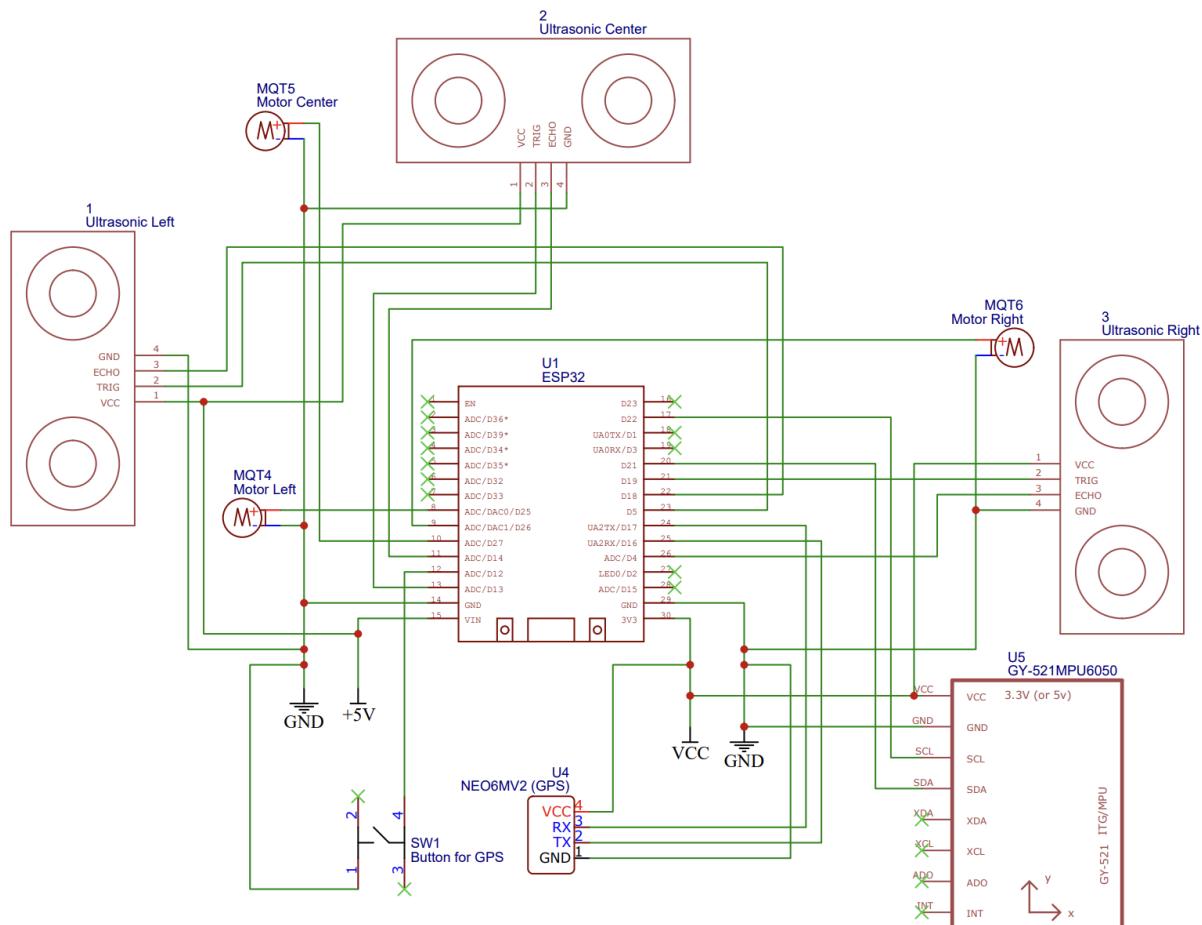
Rangkaian elektronik didesain menggunakan perangkat lunak EasyEDA. Detail koneksi pin mikrokontroler dapat dilihat pada Tabel 2.1 dan visualisasi skematik pada Gambar 2.2.

**Tabel 2.1. Konfigurasi Pin Mikrokontroler ESP32**

Komponen	Interface	Pin ESP32	Fungsi
MPU6050	I2C	GPIO 21 (SDA), 22 (SCL)	Deteksi jatuh (sudut kemiringan)
Ultrasonik Kiri	Digital I/O	GPIO 5 (Trig), 18 (Echo)	Sensor jarak sisi kiri
Ultrasonik Tengah	Digital I/O	GPIO 13 (Trig), 14 (Echo)	Sensor jarak depan
Ultrasonik Kanan	Digital I/O	GPIO 19 (Trig), 4 (Echo)	Sensor jarak sisi kanan
GPS Module	UART (Serial2)	GPIO 16 (RX), 17 (TX)	Mengirim data lokasi (NMEA)
Motor Kiri	PWM Output	GPIO 25	Getaran peringatan kiri
Motor Kanan	PWM Output	GPIO 26	Getaran peringatan kanan
Motor Tengah	PWM Output	GPIO 27	Getaran peringatan tengah
Tombol SOS/GPS	Digital Input	GPIO 12	Pemicu sinyal darurat manual dan mengirimkan

		live location kepada user melalui Telegram
--	--	---

Catatan: Karena motor tengah kami rusak, kami menggunakan LED untuk menggantikannya.

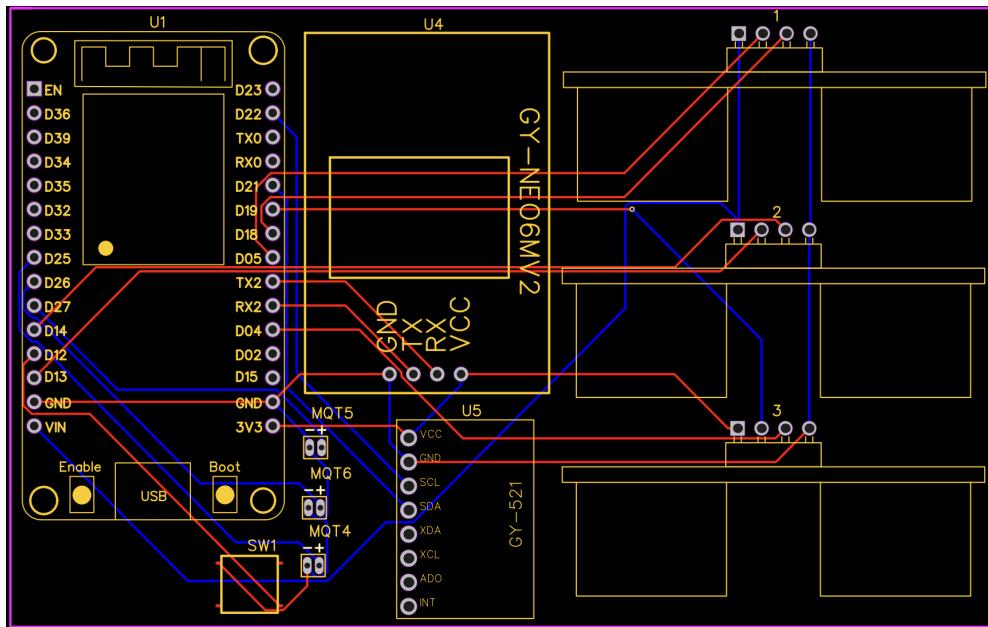


**Gambar 2.2.** Skematik Rangkaian Elektronik Smart Belt

Penggunaan protokol I<sup>2</sup>C (Inter-Integrated Circuit) pada MPU6050 memungkinkan pembacaan data sensor yang cepat dengan penggunaan kabel yang minimal, sementara modul GPS menggunakan komunikasi serial UART (Universal Asynchronous Receiver-Transmitter) [3].

### 2.2.2 Desain PCB (Printed Circuit Board)

Untuk memastikan portabilitas alat saat dikenakan di pinggang, desain PCB dibuat ringkas dengan dimensi 104,5 x 66 mm. Jalur *power* dibuat lebih lebar (20-30 mil) dibanding jalur sinyal untuk menjaga stabilitas arus, terutama saat motor getar aktif. Layout PCB final disajikan pada Gambar 2.3.



**Gambar 2.3.** Desain Layout PCB

### 2.2.3 Desain Mekanik dan Penempatan Sensor

Desain *casing* dibuat menggunakan Autodesk Fusion 360 dengan material PLA melalui proses pencetakan 3D. Gambar 2.4 menunjukkan penempatan sensor ultrasonik yang menghadap ke depan di tengah, kanan, dan kiri pinggang untuk memaksimalkan *field of view* (FoV) deteksi halangan, serta penempatan modul MPU6050 yang diletakkan di pusat massa tubuh pengguna untuk akurasi pembacaan akselerasi. Motor kanan, tengah, maupun kiri diletakkan di bagian dalam *belt* agar getarannya dapat dirasakan secara optimal oleh *user*. Modul GPS dan *button* diletakkan di wadah tengah. *Power supplier* berupa *power bank* diletakkan di *belt* belakang; kabelnya menyusuri *belt* melalui samping ke ESP32 di depan.



**Gambar 2.4.** Desain Mekanik Smart Belt dan Penempatan Komponen

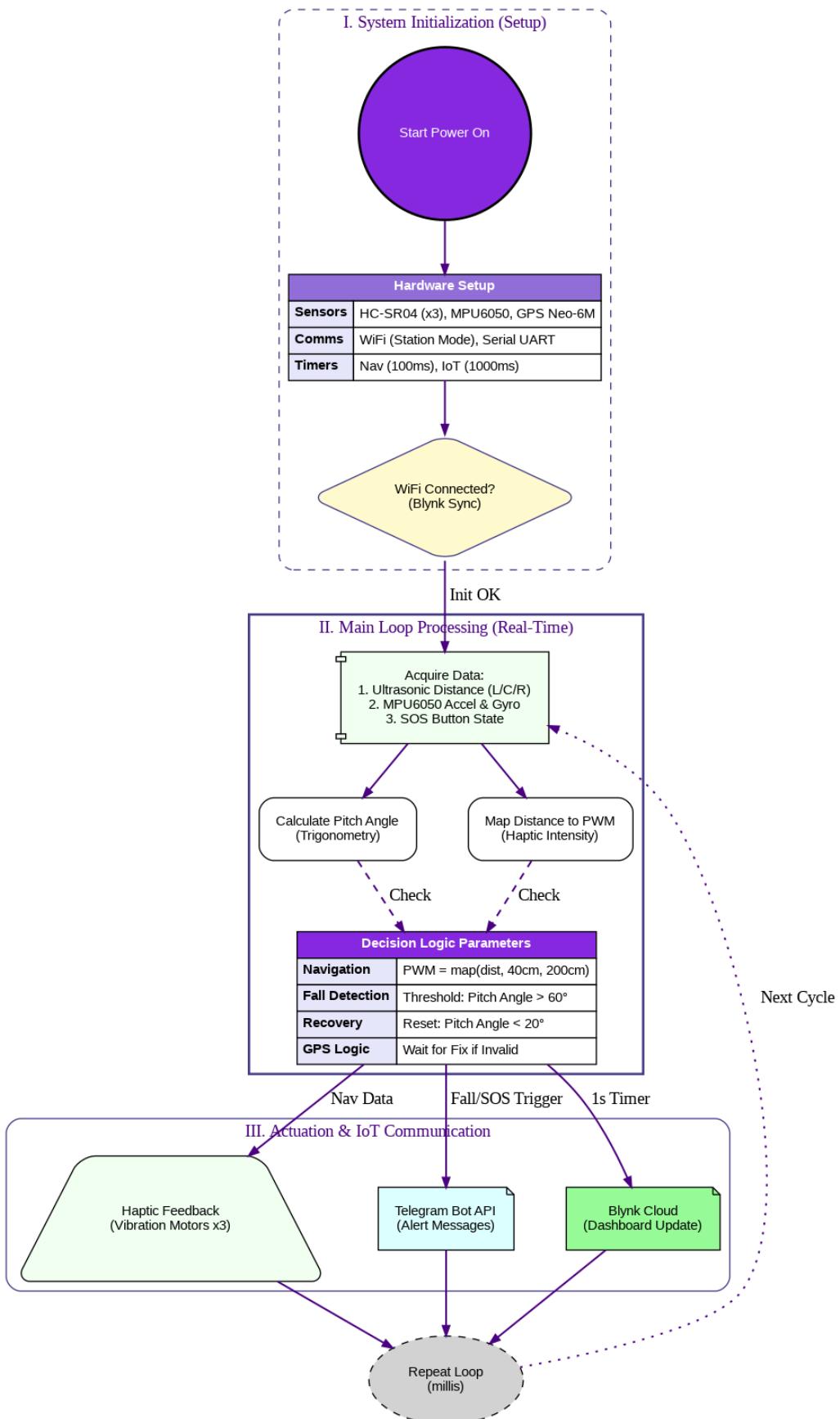
*Catatan: Terdapat LED di bagian depan karena LED menggantikan motor tengah yang rusak. Hanya dipakai 1 LED*

## 2.3 Desain Perangkat Lunak

Perangkat lunak pada mikrokontroler ESP32 dirancang menggunakan arsitektur *time-division multiplexing* sederhana untuk mendorong responsivitas sistem secara *real-time*. Sistem tidak berjalan secara sekuensial linier yang memblokir proses, melainkan dibagi menjadi tiga tugas yang berjalan secara *pseudo-parallel* menggunakan penjadwalan waktu: (1) Akuisisi sensor & navigasi, (2) Pemantauan keselamatan (deteksi jatuh), dan (3) Komunikasi IoT.

### 2.3.1 Alur Logika Sistem

Alur logika utama diilustrasikan pada Gambar 2.5. Sistem dimulai dengan inisialisasi koneksi WiFi dan mempersiapkan sensor. Setelah itu, program memasuki *loop* utama yang diatur oleh *software timer* untuk mengeksekusi fungsi berdasarkan prioritas waktu.



Gambar 2.5. Diagram Alur Logika Utama (Main Loop Architecture)

### **2.3.2 Logika Navigasi Adaptif (Haptic Feedback)**

Sistem menggunakan logika kontrol proportional terbalik untuk fitur navigasi. Intensitas getaran motor umpan balik tidak bersifat biner (nyala/mati), melainkan berbanding terbalik dengan jarak halangan.

- Jika jarak objek > Batas Aman, motor mati.
- Jika jarak objek semakin dekat, motor bergetar semakin kencang (PWM meningkat).
- Jika jarak objek < Batas Bahaya, motor bergetar maksimum.

### **2.3.3 Logika Deteksi Jatuh dan Pelacakan Lokasi**

Untuk deteksi jatuh, sistem tidak hanya mengandalkan percepatan sesaat, tetapi menghitung sudut kemiringan tubuh pengguna secara *real-time*. Jika sudut kemiringan melebihi ambang batas kritis, sistem memicu *state* "Bahaya". Secara bersamaan, modul GPS secara kontinu melakukan *parsing* data satelit NMEA (National Marine Electronics Association) di latar belakang. Saat tombol SOS atau tombol Blynk ditekan, sistem akan mengambil koordinat terakhir yang valid untuk dikirimkan ke server Telegram, memastikan lokasi pengguna dapat dilacak bahkan saat pengguna tidak sadarkan diri.

## BAB III

### IMPLEMENTASI PERANGKAT KERAS

#### 3.1 Pemilihan Komponen & Justifikasi

Perancangan sistem dilakukan dengan pendekatan *wearable embedded system* yang menekankan kemudahan integrasi, konsumsi daya yang stabil, serta keandalan saat digunakan dalam skenario mobilitas pengguna tunanetra. Komponen utama yang digunakan pada purwarupa ditunjukkan pada Tabel berikut.

**Tabel 3.1. Spesifikasi Komponen Utama Sistem**

Komponen	Fungsi Utama	Alasan Pemilihan
ESP32 Dev Board	Unit pemrosesan utama & komunikasi IoT	Mendukung WiFi terintegrasi, pemrosesan real-time, GPIO cukup banyak, konsumsi daya efisien
Ultrasonic Sensor (HC-SR04)	Deteksi jarak dan keberadaan halangan di depan pengguna	Akurat pada jarak dekat–menengah, biaya rendah, stabil untuk lingkungan indoor & outdoor
MPU6050	Deteksi jatuh berbasis akselerasi dan orientasi tubuh	Menggabungkan accelerometer & gyroscope, memungkinkan analisis perubahan percepatan secara dinamis
Tombol Fisik (SOS Button)	<i>Manual emergency trigger</i>	Memberikan kontrol langsung bagi pengguna ketika membutuhkan bantuan
Powerbank 5V	Sumber daya utama perangkat	Mudah diakses, aman, kapasitas besar, dan mendukung penggunaan portabel dalam durasi panjang
Motor Getar (Haptic Feedback)	Penyampaian informasi navigasi secara non-auditori	Meminimalkan distraksi suara dan menjaga privasi pengguna

Berbeda dengan rancangan awal pada proposal, implementasi akhir tidak menggunakan sensor LiDAR. Seluruh modul deteksi jarak direalisasikan menggunakan sensor ultrasonik, karena hasil uji awal menunjukkan bahwa:

1. sensor ultrasonik lebih stabil untuk jarak operasi sabuk navigasi.
2. proses integrasi perangkat keras menjadi lebih efisien.

3. sensor ultrasonik dapat mendeteksi halangan transparan, sedangkan LiDAR tidak.

Sistem tenaga juga disederhanakan menggunakan powerbank sebagai sumber suplai 5V. Pendekatan ini dipilih karena:

- aman untuk penggunaan wearable.
- menyediakan kapasitas operasi yang cukup lama.
- memungkinkan penggantian/pengisian daya secara cepat.
- serta kompatibel langsung dengan tegangan masukan ESP32.

Selain itu, penambahan tombol fisik bertujuan memberikan redundansi fungsi keselamatan. Tombol ini memungkinkan pengguna mengirim sinyal bantuan ketika merasa tidak aman, tersesat, atau membutuhkan pertolongan meskipun tidak terjadi insiden jatuh. Dengan demikian, sistem tidak hanya bersifat *reactive safety*, tetapi juga mendukung *proactive user-initiated safety*.

### 3.2 Integrasi Hardware

Tahap implementasi perangkat keras dilakukan melalui beberapa proses, yaitu perakitan modul sensor pada PCB *custom*, pemasangan tombol fisik SOS, integrasi jalur koneksi ESP32, serta penempatan seluruh komponen pada casing sabuk. Sensor ultrasonik diposisikan pada bagian depan sabuk untuk memperoleh cakupan area yang merepresentasikan bidang pandang pengguna saat berjalan. Motor getar ditempatkan pada beberapa titik sehingga pengguna dapat membedakan arah bahaya melalui pola getaran yang berbeda. Modul ESP32 dan rangkaian PCB dilindungi oleh casing cetak 3D untuk meningkatkan kekokohan serta kenyamanan pemakaian.

MPU6050 dipasang pada posisi yang stabil terhadap orientasi tubuh agar perubahan percepatan saat jatuh dapat terukur dengan baik. Sementara itu, tombol fisik diletakkan pada sisi sabuk yang mudah dijangkau sehingga pengguna dapat menekannya tanpa harus melepas perangkat. Hasil akhir integrasi menghasilkan purwarupa yang kokoh, ergonomis, dan siap digunakan pada tahap pengujian fungsional dan validasi sistem.

## **BAB IV**

### **IMPLEMENTASI PERANGKAT LUNAK**

Implementasi perangkat lunak pada mikrokontroler ESP32 dirancang menggunakan penjadwalan waktu (*time-division multiplexing*) untuk mendorong sistem bekerja secara *real-time*. Mengingat sistem ini adalah alat bantu keselamatan bagi tunanetra, prioritas utama adalah pada responsivitas sensor halangan dan deteksi jatuh, sementara komunikasi IoT dikelola sebagai *background task* agar tidak menghambat fungsi navigasi utama. Pengembangan kode dilakukan menggunakan Arduino IDE.

#### **4.1 Struktur Kode dan Pustaka**

Untuk mengelola fitur (sensor, motor, WiFi, GPS, Telegram, Blynk), kode program dibangun secara modular. Pemilihan pustaka didasarkan pada efisiensi memori dan dukungan terhadap protokol komunikasi aman (SSL/TLS) yang dibutuhkan oleh API Telegram. Rincian pustaka dan fungsinya disajikan pada Tabel 4.1.

**Tabel 4.1. Daftar Libraries dan Fungsinya**

Library	Fungsi	Justifikasi Penggunaan Teknis
BlynkSimpl eEsp32.h	Manajemen IoT	Menangani protokol komunikasi <i>bi-directional</i> dengan server Blynk, termasuk sinkronisasi <i>virtual pin</i> dan fitur <i>keep-alive</i> agar koneksi tidak mudah putus.
WiFiClientS ecure.h	Keamanan HTTPS	Menyediakan lapisan enkripsi SSL/TLS yang wajib digunakan saat mengirim HTTP POST Request ke API Bot Telegram ( <a href="http://api.telegram.org">api.telegram.org</a> ) untuk menjamin keamanan data.
TinyGPS++. h	Parsing NMEA	Menerjemahkan data mentah string NMEA (GPGGA, GPRMC) dari modul GPS menjadi format desimal (Latitude/Longitude) yang mudah diolah oleh logika program.
Adafruit_M PU6050.h	Driver sensor IMU	Mengabstraksi <i>register low-level</i> I2C (alamat 0x68) menjadi fungsi siap pakai untuk membaca data akselerasi ( $m/s^2$ ) dan kecepatan sudut (rad/s).
HTTPClient. h	Request web	Menangani pembuatan <i>header</i> dan <i>payload</i> HTTP untuk mengirim pesan teks ke Telegram.

Selain pustaka di atas, struktur program menggunakan *BlynkTimer* sebagai *task scheduler* pengganti fungsi *delay()*.

- Navigasi (100ms): Interval cepat untuk respon motor yang halus.

- Update IoT (1000ms): Interval sedang untuk sinkronisasi data ke aplikasi Blynk.
- Cek Antrian GPS (2000ms): Interval lambat untuk memantau status penguncian satelit.

Hal ini mencegah *blocking* proses, sehingga pembacaan sensor ultrasonik dapat tetap berjalan mulus meskipun sistem sedang menunggu koneksi WiFi.

## 4.2 Algoritma Akuisisi Data Sensor

Sistem menggunakan tiga unit sensor ultrasonik HC-SR04 untuk memetakan lingkungan sekitar. Tantangan utama pada sensor ini adalah *noise* pembacaan. Oleh karena itu, diimplementasikan filter sederhana pada fungsi *bacaJarak()* dengan membatasi *timeout* pembacaan pulsa (30ms) untuk mencegah sistem menggantung saat tidak ada objek yang memantulkan gelombang.

**Kode Program 4.1.** Fungsi Akuisisi Jarak (Non-Blocking)

```
float bacaJarak(int trigPin, int echoPin) {
    digitalWrite(trigPin, LOW); delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Timeout 30ms (kecepatan suara ~343m/s, 30 ms = ~5 meter) agar loop tidak terhenti lama
    long duration = pulseIn(echoPin, HIGH, 30000);

    if (duration == 0) return 400; // Return nilai max jika timeout
    return duration * 0.034f / 2.0f;
}
```

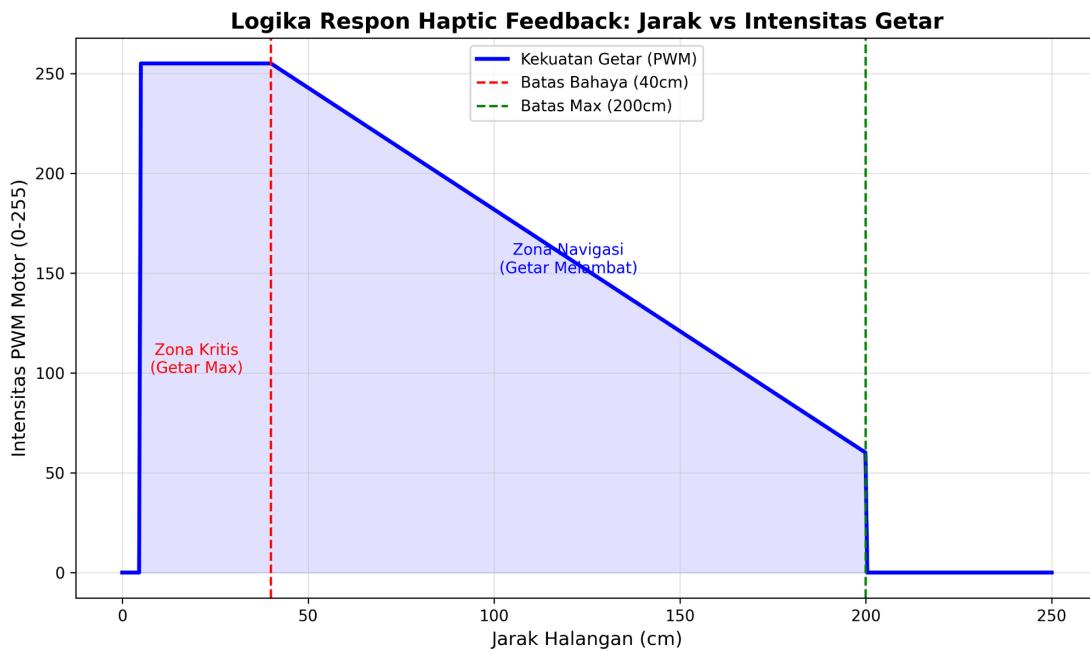
## 4.3 Algoritma Navigasi Haptic

Data jarak yang diperoleh diproses menggunakan logika kontrol proporsional terbalik. Sistem tidak hanya memberikan output biner (On/Off), melainkan memberikan gradasi intensitas getaran berdasarkan kedekatan objek. Fungsi *kontrolOutput* memetakan rentang jarak input (40 cm - 200 cm) menjadi rentang sinyal PWM (255 - 60).

- Zona Noise ( $0 \leq Jarak < 5 \text{ cm}$ ): PWM 0 (motor mati).
- Zona Bahaya ( $5 \leq Jarak \leq 40 \text{ cm}$ ): PWM maksimum (255).
- Zona Transisi ( $40 < Jarak \leq 200 \text{ cm}$ ): PWM berskala linear.
- Zona Aman ( $> 200 \text{ cm}$ ): PWM 0 (motor mati).

*Catatan: Pada LED, maka gradien mengatur terangnya*

Ilustrasi logika pemetaan sinyal ini dapat dilihat pada Gambar 4.1.



**Gambar 4.1.** Grafik Hubungan Jarak Halangan terhadap Intensitas PWM Motor

#### 4.4 Algoritma Deteksi Jatuh

Salah satu fitur keselamatan adalah deteksi jatuh. Algoritma ini dirancang dengan mekanisme *safety startup* dan *hysteresis* untuk menghindari *false alarm*.

##### 4.4.1 Perhitungan Sudut (Trigonometri)

Sistem menghitung sudut kemiringan tubuh pengguna menggunakan data akselerometer pada sumbu X, Y, dan Z. Rumus yang diimplementasikan adalah:

$$Pitch = \left| \arctan \left( \frac{\sqrt{A_x^2 + A_y^2}}{A_z} \right) \times \frac{180}{\pi} \right|$$

Rumus ini menghitung orientasi gravitasi terhadap sumbu vertikal sensor ( $A_z$ ).

- $\sqrt{A_x^2 + A_y^2}$ : Menghitung besaran vektor gaya gabungan pada bidang horizontal (sumbu X dan Y).
- Pembagian dengan  $A_z$ : Membandingkan seberapa besar gaya horizontal dibandingkan gaya vertikal.
  - Saat berdiri tegak:  $A_z$  besar (1g),  $A_x/A_y$  kecil (0g). Hasil pembagian mendekati 0. Sudut ~0 derajat.

- Saat jatuh datar (tertidur):  $Az$  mendekati 0,  $Ax/Ay$  besar (1g). Hasil pembagian sangat besar. Sudut mendekati 90 derajat.
- Fungsi *atan* (Arc Tangent): Mengubah perbandingan nilai tersebut menjadi satuan sudut dalam radian.
- $180/\pi$ : Mengonversi radian ke derajat agar mudah dibaca.

#### 4.4.2 Logika Keputusan dan Safety Startup

Ambang batas bahaya ditetapkan pada sudut  $> 60$  derajat. Pada sudut  $< 20$  derajat, artinya tegak (status Aman), sedangkan pada sudut  $20 - 60$  derajat, artinya miring atau bungkuk (status Waspada). Untuk mencegah alarm palsu saat alat baru dinyalakan (posisi sensor belum stabil), diterapkan logika pengaman waktu (*boot time check*).

1. Safety Startup: Alarm deteksi jatuh dinonaktifkan selama 5 detik pertama setelah *booting*.
2. Debouncing: Status jatuh dikunci dan tidak akan mengirim notifikasi berulang-ulang karena memakai variabel *isFalling* dan timer *FALL\_COOLDOWN* (15 detik).

#### Kode Program 4.2. Logika Deteksi Jatuh dengan Pengaman

```
// Menghitung kemiringan
float tilt = atan2(sqrt(ax*ax + ay*ay), az) * 180.0 / PI;
pitchAngle = abs(tilt);

if (pitchAngle > ANGLE_DANGER) { // Jika > 60 derajat
    // Fitur Pengaman: Mengabaikan jika baru nyala < 5 detik
    if (millis() - bootTime > 5000) {
        if (!isFalling) {
            isFalling = true;
            triggerTelegram("⚠ PERINGATAN DARURAT!", false); // Kirim notif
        }
    }
}
```

#### 4.5 Implementasi IoT dan GPS Asynchronous

Modul GPS membutuhkan waktu untuk mengunci sinyal satelit, yang dapat memakan waktu beberapa menit. Jika sistem dipaksa menunggu GPS terkunci sebelum mengirim pesan darurat, respon sistem akan menjadi lambat. Oleh karena itu, diterapkan logika *asynchronous messaging*.

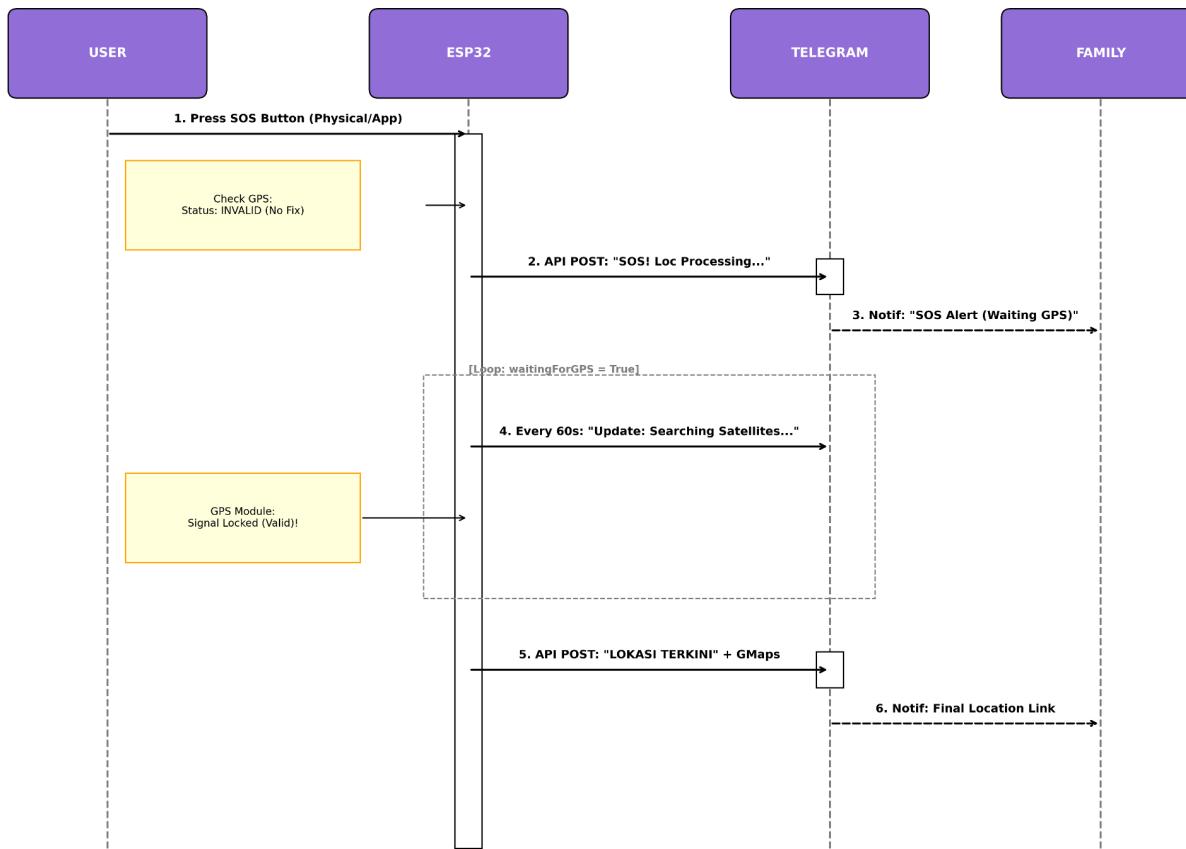
1. Saat tombol SOS atau tombol Blynk ditekan, sistem segera mengirim pesan peringatan ke Telegram tanpa menunggu GPS.
2. Jika GPS belum valid, sistem mengaktifkan mode *waitForGPS*.
3. Fungsi *cekAntrianGPS()* akan berjalan setiap 2 detik. Jika GPS masih belum terkunci, sistem akan mengirimkan laporan status berkala ke Telegram setiap 60 detik (misal: "Update Proses GPS: Masih mencari sinyal... Jumlah Satelit: 2"). Ini memberikan kepastian kepada keluarga bahwa sistem sedang bekerja.
4. Segera setelah sinyal satelit terkunci, sistem otomatis mengirimkan tautan Google Maps.

Alur komunikasi data ini divisualisasikan dalam diagram sekuensial pada Gambar 4.2.

#### Kode Program 4.3. Logika Update Status GPS Berkala

```
// Variabel
bool waitForGPS = false;      // Flag status: apakah sedang menunggu sinyal?
unsigned long lastGpsReportTime = 0; // Timer untuk laporan berkala

// Logika Utama: Cek antrian GPS (background process)
// Fungsi ini berjalan setiap 2 detik tanpa mengganggu sensor lain
void cekAntrianGPS() {
    if (waitForGPS) {
        // SKENARIO 1: Sinyal akhirnya terkunci
        if (gps.location.isValid()) {
            String link = "http://maps.google.com/?q=" + String(gps.location.lat(), 6) + "," +
String(gps.location.lng(), 6);
            sendTelegramMessage("📍 LOKASI TERKINI (SUDAH LOCK):\n" + link);
            waitForGPS = false; // Matikan mode tunggu
        }
        // SKENARIO 2: Masih mencari sinyal
        else {
            // Kirim update status ke Telegram setiap 60 detik
            if (millis() - lastGpsReportTime > 60000) {
                int sat = gps.satellites.value();
                sendTelegramMessage("🕒 UPDATE STATUS: Masih mencari sinyal
satelit...\nJumlah Satelit: " + String(sat));
                lastGpsReportTime = millis();
            }
        }
    }
}
```



**Gambar 4.2.** Diagram Sekuensial Pengiriman Pesan GPS dan Lokasi Susulan

Penerapan logika ini memastikan bahwa keluarga pengguna segera mengetahui adanya keadaan darurat, meskipun lokasi spesifik menyusul beberapa saat kemudian.

## BAB V

### ANTARMUKA PENGGUNA

Antarmuka pengguna pada sistem Smart Belt terdiri dari antarmuka fisik berbasis umpan balik haptik untuk pengguna tunanetra, serta antarmuka digital berbasis IoT untuk pemantauan jarak jauh oleh keluarga atau pengasuh. Desain ini bertujuan meminimalkan beban kognitif pengguna saat bernavigasi sekaligus memaksimalkan kecepatan penyampaian informasi saat darurat.

#### 5.1 Antarmuka Fisik

Mengingat target pengguna adalah penyandang tunanetra, interaksi alat menggunakan indra peraba. Sistem menerapkan mekanisme *haptic feedback* yang berbanding lurus dengan jarak objek, memungkinkan pengguna mengetahui jarak benda di lingkungan sekitarnya tanpa melihat.

##### 5.1.1 Desain Interaksi Navigasi (Haptic Feedback)

Sistem menggunakan tiga unit aktuator getar (*vibration motor*) yang ditempatkan pada posisi kiri, tengah, dan kanan. Getaran motor berbanding lurus dengan jarak objek; semakin dekat halangan, semakin kuat getaran yang dirasakan pada sisi tersebut. Hal ini memungkinkan pengguna membangun mengetahui jarak halangan di lingkungan sekitar tanpa melihat. Rincian interaksi fisik disajikan pada Tabel 5.1.

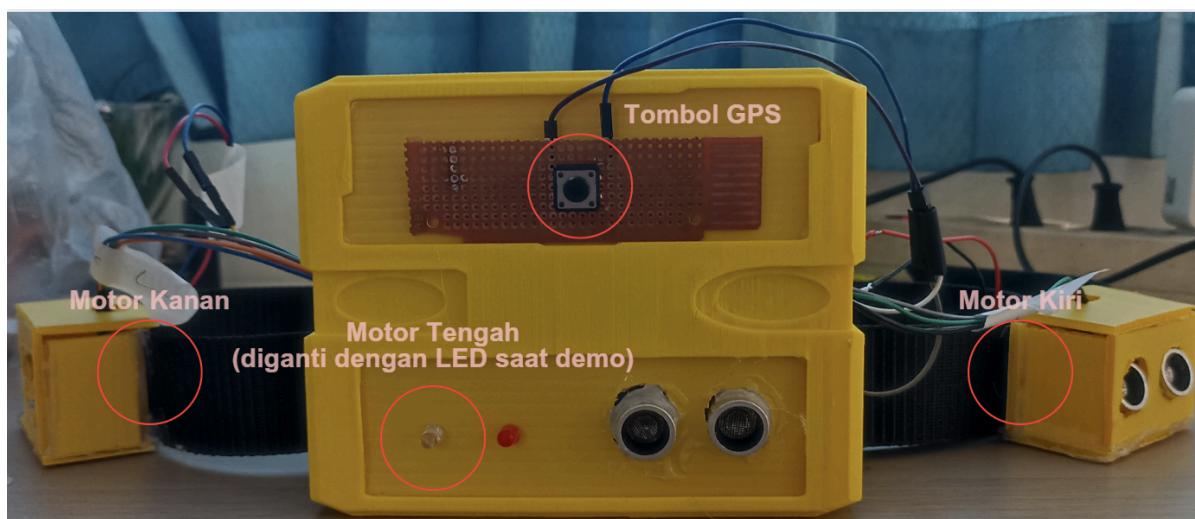
**Tabel 5.1.** Matriks Logika Interaksi Fisik Pengguna

Kondisi Lingkungan	Jarak Terdeteksi (d)	Respon Output (Motor)	Arti bagi Pengguna
Aman (bebas hambatan)	$d > 200 \text{ cm}$	Mati (diam)	Jalan aman, silakan maju.
Peringatan Awal	$40 < d \leq 200 \text{ cm}$	Gradasi lemah ke kuat	Ada objek di depan, waspada.
Bahaya (berhenti)	$5 \leq d \leq 40 \text{ cm}$	Maksimum (getaran penuh)	Objek sangat dekat, segera berhenti.
Noise	$d < 5 \text{ cm}$	Mati (filter software)	Pembacaan tidak valid, abaikan.

*Catatan Implementasi:* Pada perangkat prototype yang didemonstrasikan, motor bagian tengah digantikan sementara oleh LED. Hal ini dilakukan karena motor tidak berfungsi saat perakitan akhir. Namun, secara logika program dan skematik desain, sistem tetap dirancang untuk menggerakkan tiga motor getar secara simultan.

### 5.1.2 Mekanisme Kontrol GPS secara Fisik

Selain output, terdapat input fisik berupa tombol *tactile* yang berfungsi sebagai pemicu manual sinyal GPS. Tombol ini dirancang dengan fitur *debouncing* dan *safety delay* 2 detik untuk mencegah aktivasi tidak sengaja akibat gerakan tubuh pengguna.



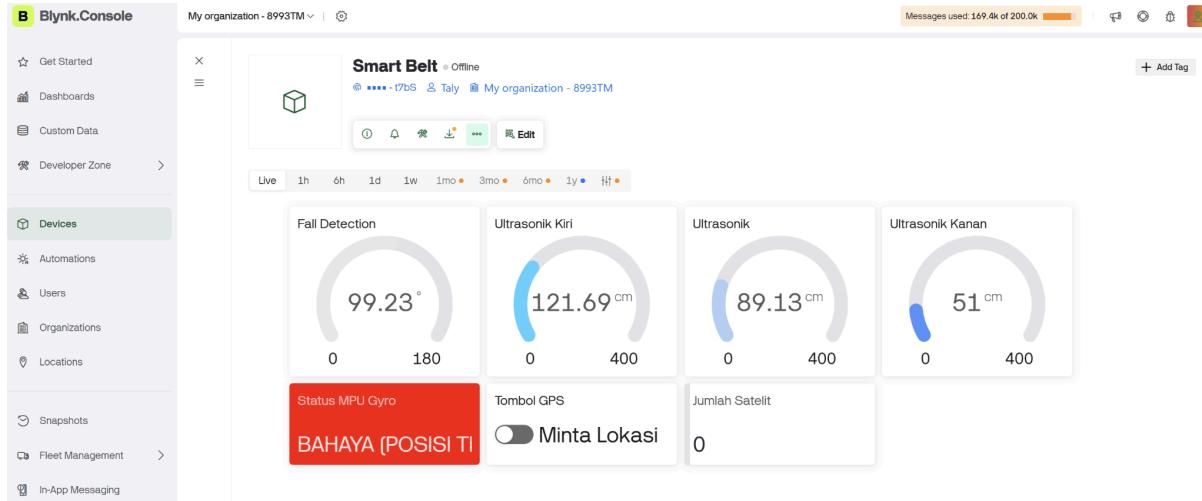
**Gambar 5.1.** Implementasi Antarmuka Fisik pada Perangkat Smart Belt

## 5.2 Antarmuka Digital

Antarmuka digital dikembangkan menggunakan platform Blynk untuk *real-time monitoring* dan Telegram untuk notifikasi darurat. Kedua platform ini menjadi jembatan informasi antara pengguna tunanetra dengan pengasuh atau keluarga.

### 5.2.1 Dashboard Monitoring (Blynk)

Aplikasi Blynk berfungsi sebagai panel utama yang memvisualisasikan data sensor secara *live* dengan latensi ~1 detik. Tata letak antarmuka dirancang agar semua informasi dapat dibaca dalam sekali pandang. Tampilan antarmuka aplikasi dapat dilihat pada Gambar 5.2.



**Gambar 5.2.** Tampilan Dashboard Monitoring pada Situs Web Blynk

Berdasarkan Gambar 5.2, dashboard menampilkan elemen-elemen berikut:

1. Tombol GPS Virtual (V0): Memungkinkan pengasuh mengetahui lokasi pasien dari jarak jauh tanpa pasien menekan tombol GPS fisik
2. Indikator Jarak (V1, V2, V3): Menampilkan jarak terdeteksi dari ketiga sensor ultrasonik
3. Monitor Kemiringan (V4): Menampilkan nilai sudut kemiringan tubuh pengguna secara *real-time* untuk memantau apakah pengguna dalam posisi tegak/miring/jatuh.
4. Status Keselamatan (V5): Menampilkan teks status ("AMAN" berwarna hijau, "WASPADA" berwarna kuning, atau "BAHAYA" berwarna merah) hasil olahan sensor MPU6050.
5. Monitor Satelit (V6): Menampilkan jumlah satelit GPS yang terkunci sebagai indikator validitas data lokasi.

### 5.2.2 Sistem Notifikasi Peringatan (Telegram)

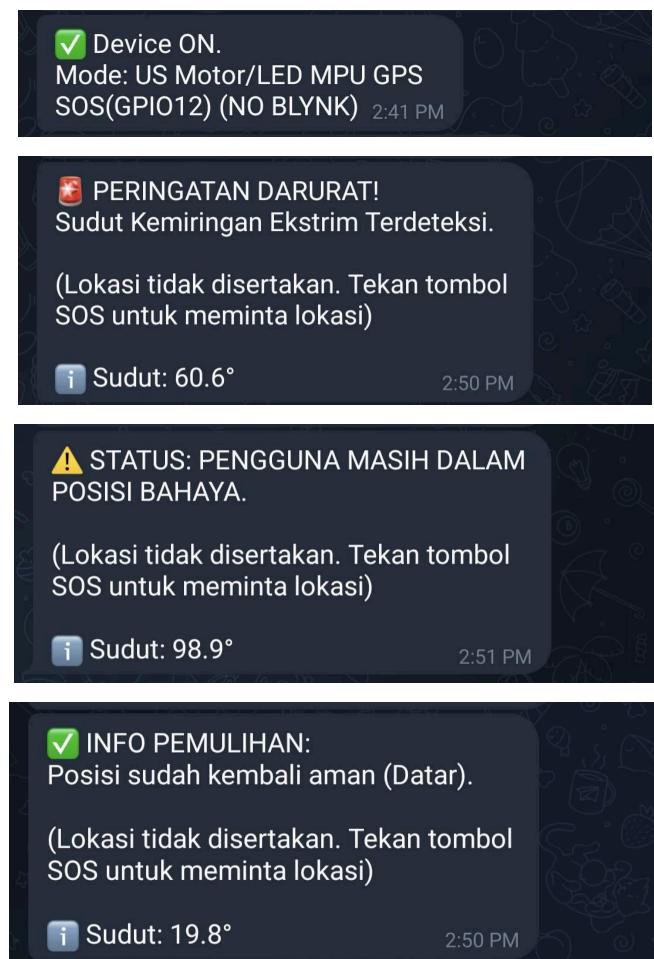
Telegram Bot digunakan untuk mengirimkan notifikasi *push* yang bersifat mendesak. Berbeda dengan Blynk yang harus dibuka aplikasinya, Telegram bisa memberikan peringatan aktif yang muncul di *lock screen* pengasuh.

Sistem notifikasi dirancang dengan pesan:

1. Peringatan Bahaya: Menginformasikan jatuh segera setelah kejadian.

2. Informasi Pulih: Menginformasikan jika pasien sudah kembali berdiri setelah jatuh
3. Update Status GPS: Memberikan informasi progres pencarian sinyal GPS secara berkala jika satelit belum terkunci.
4. Lokasi Final: Mengirimkan tautan Google Maps yang presisi segera setelah koordinat didapatkan.

Contoh riwayat percakapan sistem ditunjukkan pada Gambar 5.3.





**Gambar 5.3.** Format Pesan Peringatan dan Pelacakan Lokasi pada Telegram

Penggunaan API Telegram memungkinkan pengiriman *hyperlink*, sehingga pengasuh dapat langsung menuju lokasi pengguna jika pengguna mengalami kecelakaan.

## BAB VI

### PENGUJIAN DAN ANALISIS

#### 6.1 Skenario Pengujian

Pengujian pada *smart belt* dilakukan untuk memastikan seluruh fungsi utama berjalan sesuai implementasi pada program, meliputi: pembacaan jarak tiga sensor ultrasonik, respons keluaran (buzzer kiri, LED tengah, motor getar kanan), deteksi jatuh (*fall detection*) berbasis MPU6050, integrasi monitoring melalui Blynk, serta pengiriman notifikasi Telegram saat kondisi darurat (*fall* atau *SOS*). Seluruh pengujian dilakukan dengan mengamati keluaran fisik perangkat, *serial monitor*, serta data yang tampil pada Blynk (V1–V6).

Berikut adalah rencana skenario pengujian pada subbab ini (pengujian latensi *GPS* akan ditambahkan pada tahap berikutnya).

No	Jenis pengujian	Tujuan	Variabel yang diamati	Kriteria keberhasilan
1	Tes jarak sensor (ultrasonik)	Memvalidasi akurasi baca jarak dan respons indikator per arah	Nilai jarak (cm) di <i>serial monitor</i> dan Blynk (V1–V3), respons buzzer/LED/motor	Nilai jarak stabil dan indikator aktif sesuai ambang batas serta arah sensor
2	Tes akurasi jatuh ( <i>fall detection</i> )	Memastikan deteksi kemiringan ekstrem memicu status bahaya dan notifikasi	Sudut kemiringan (V4), status (V5), notifikasi Telegram	Status “bahaya” muncul pada sudut ekstrem dan Telegram terkirim sesuai logika program
3	Tes latensi notifikasi <i>GPS</i>	Mengukur jeda waktu hingga pesan lokasi terkirim dan memastikan mekanisme “menunggu <i>GPS lock</i> ” berjalan	kirim lokasi jika koordinat valid; jika belum valid: kirim status proses dan lokasi susulan; pembaruan progres tiap ±60 s saat menunggu.	Telegram berisi tautan Google Maps saat valid; satelit pada Blynk V6; perhitungan latensi t0 menuju t1 dan t0 menuju t2

**Tabel 6.1 Rencana pengujian**

### 6.1.1. Tes jarak sensor ultrasonik dan respons indikator arah (T1)

Pengujian ini memvalidasi tiga sensor ultrasonik (kiri, tengah, kanan) dan keluaran indikator per arah: buzzer kiri, LED tengah, motor getar kanan. Logika respons mengikuti program: jarak sangat dekat ( $< 5$  cm) dianggap *noise* sehingga output tidak aktif; jarak bahaya ( $\leq 40$  cm) memicu output maksimum; jarak 41–200 cm memicu output aktif dengan intensitas menurun (*PWM mapping*); jarak  $> 200$  cm output tidak aktif.

Langkah uji:

1. Tempatkan objek datar (misalnya karton) di depan sensor kiri pada jarak tertentu (contoh: 3 cm, 20 cm, 40 cm, 100 cm,  $>200$  cm).
2. Amati respons buzzer kiri (aktif/tidak dan intensitas relatif) serta catat nilai jarak di *serial monitor* dan Blynk (V2 untuk kiri).
3. Ulangi langkah yang sama untuk sensor tengah (LED, Blynk V1) dan sensor kanan (motor, Blynk V3).
4. Lakukan minimal 3 pengulangan untuk tiap jarak agar terlihat kestabilan pembacaan.

Kondisi jarak (cm)	Ekspektasi pembacaan	Ekspektasi output kiri (buzzer)	Ekspektasi output tengah (LED)	Ekspektasi output kanan (motor)
$< 5$	Diabaikan sebagai <i>noise</i>	Tidak aktif	Tidak aktif	Tidak aktif
5 – 40	Bahaya	Aktif kuat (maksimum)	Aktif kuat (maksimum)	Aktif kuat (maksimum)
41 – 200	Hambatan terdeteksi	Aktif menurun bertahap (semakin jauh semakin lemah)	Aktif menurun bertahap	Aktif menurun bertahap
$> 200$	Dianggap aman/tidak ada objek	Tidak aktif	Tidak aktif	Tidak aktif

**Tabel 6.2 Matriks skenario uji jarak dan ekspektasi respons**

### 6.1.2 Tes akurasi deteksi jatuh (*fall detection*) berbasis MPU6050 (T2)

Pengujian ini memverifikasi perhitungan sudut kemiringan (*pitchAngle*) dari MPU6050 dan perubahan status yang dihasilkan. Program mengelompokkan kondisi menjadi:

- aman (*reset*) bila sudut  $< 20^\circ$
- waspada bila sudut berada di antara kondisi aman dan bahaya
- bahaya bila sudut  $> 60^\circ$  dan memicu pesan peringatan

Program juga menerapkan *safety startup* selama 5 detik setelah perangkat menyala untuk mencegah notifikasi jatuh palsu, serta *cooldown* 15 detik untuk pengiriman pengingat berulang ketika kondisi bahaya berlanjut.

Langkah uji:

1. Nyalakan perangkat dan pastikan selama 5 detik pertama tidak ada notifikasi jatuh (verifikasi *safety startup*).
2. Posisikan perangkat pada kondisi normal (sudut kecil) dan amati status di Blynk V5 serta sudut di V4.
3. Miringkan perangkat hingga mendekati kondisi bahaya (sekitar  $>30^\circ$ ) untuk memicu status waspada, lalu miringkan lebih ekstrem hingga  $>60^\circ$  untuk memicu status bahaya.
4. Pertahankan kondisi bahaya dan amati apakah sistem mengirim pengingat secara berkala (mengikuti *cooldown*).
5. Kembalikan perangkat ke kondisi  $<20^\circ$  dan verifikasi sistem mengirim pesan pemulihan serta status kembali aman.

Kondisi uji	Rentang sudut ( $^\circ$ )	Status pada Blynk (V5)	Ekspektasi Telegram
Normal/aman	$< 20$	“AMAN (DATAR)”	Jika sebelumnya bahaya, kirim pesan pemulihan; jika tidak, tidak kirim
Miring (transisi)	20 – 60	“WASPADA (MIRING)”	Tidak mengirim peringatan jatuh

Bahaya/jatuh	> 60	“BAHAYA (POSISI TEGAK/JATUH?)”	Kirim peringatan bahaya (sekali), lalu pengingat mengikuti <i>cooldown</i> bila tetap bahaya
Proteksi awal nyala	0–5 detik pertama	Status tetap berjalan	Tidak boleh ada peringatan jatuh walau perangkat belum stabil

**Tabel 6.3** Skenario uji *fall detection* dan ekspektasi sistem

#### 6.1.3 Tes *GPS* dan latensi notifikasi lokasi via Telegram (T3)

Pengujian ini memastikan *SOS* (tombol fisik dan tombol Blynk V0) memicu pengiriman pesan Telegram dan, bila memungkinkan, menyertakan lokasi dari *GPS*. Program memiliki dua perilaku utama: (1) bila koordinat valid (`gps.location.isValid()`), pesan *SOS* langsung menyertakan tautan *Google Maps* serta jumlah satelit; (2) bila koordinat belum valid, sistem mengirim pesan bahwa lokasi sedang diproses, mengaktifkan `waitForGPS`, lalu mengirim lokasi susulan ketika *GPS lock* diperoleh. Selama menunggu, sistem dapat mengirim pembaruan progres sekitar setiap 60 detik yang menyertakan jumlah satelit.

Skenario	Kondisi <i>GPS</i>	Pemicu	Ekspektasi pesan Telegram	Data pendukung di Blynk
S1	koordinat valid	V0 (app) = 1	pesan “SOS MANUAL (APP)!” + link <i>Google Maps</i> + satelit	V0 berubah; V6 menampilkan satelit
S2	koordinat belum valid	V0 (app) = 1	pesan “lokasi sedang diproses” lalu lokasi susulan saat valid; ada update progres bila lama	V6 satelit bertambah saat mencari
S3	koordinat valid	tombol fisik (pin 12)	pesan “SOS MANUAL (TOMBOL FISIK)!” + link <i>Google Maps</i>	V0 ikut ditoggle sementara (1 lalu 0), V6 satelit

S4	koordinat belum valid	tombol fisik (pin 12)	pesan “lokasi sedang diproses” + lokasi susulan saat valid	V6 satelit, pesan susulan dari cekAntrianGPS()
S5	proteksi awal	< 5 s setelah boot (Blynk) / < 2 s (fisik)	tidak boleh mengirim SOS palsu akibat sync	V0 dipaksa kembali 0 saat <i>startup</i>

**Tabel 6.4** Skenario uji *SOS* dan *GPS* (berdasarkan alur program)

## 6.2 Hasil Pengujian



**Gambar 6.1** *QR-Code video presentasi*

Link video demonstrasi alat : [Video Presentasi Finpro Sisben\\_Kelompok 2](#)

Untuk navigasi, saat objek didekatkan ke sensor kiri, buzzer di sisi kiri langsung aktif. Saat objek dipindah ke depan (sensor tengah), LED menyala. Lalu ketika objek didekatkan ke sisi kanan, motor getar kanan aktif. Ketika objek dijauhkan, ketiga indikator kembali mati. Nilai jarak dari ketiga sensor juga terlihat ter-update di Blynk secara real-time.



(a)

(b)

**Gambar 6.2** (a) Kondisi ketika sudut berada di bawah 20 derajat (keadaan aman), (b) Kondisi ketika sudut melebihi 20 derajat dan kondisi berbahaya.

Untuk *fall detection*, perangkat disimulasikan dalam posisi miring ekstrem sampai masuk kondisi bahaya. Setelah itu status di Blynk berubah dan Telegram mengirim peringatan. Saat perangkat dikembalikan ke posisi aman, status kembali normal dan sistem mengirim pesan pemulihan. Selama awal perangkat baru dinyalakan, tidak terjadi notifikasi palsu, sehingga fitur proteksi awal (*safety startup*) berjalan.

Untuk *SOS*, baik tombol fisik maupun tombol di Blynk sama-sama berhasil mengirim notifikasi ke Telegram. Saat data *GPS* sudah valid, pesan *SOS* berisi link *Google Maps*. Jika *GPS* belum terkunci, Telegram tetap mengirim info “lokasi sedang diproses” dan lokasi dikirim menyusul ketika sudah terbaca (Lihat Gambar 5.3).

Untuk memudahkan verifikasi, pengujian perangkat direkam dalam video demo dan disimpan di Google Drive. Video tersebut memperlihatkan respons indikator (buzzer, LED, dan motor getar), perubahan status di Blynk, serta notifikasi Telegram. Sebagai catatan, LED menggantikan motor tengah, dan *buzzer* menggantikan motor kiri. Hal ini dilakukan karena motor tengah dan kiri rusak sesaat sebelum demonstrasi final.

### 6.3 Analisis Kendala & Perbaikan (Debugging)

Selama pengembangan sampai proses pengambilan video demo, kendala yang paling terasa ada di sisi hardware. Motor getar DC yang awalnya berfungsi sempat menjadi tidak stabil dan akhirnya mengalami kondisi seperti *short*. Dari hasil pengecekan, masalah ini paling mungkin dipicu oleh suplai daya yang kurang stabil saat motor menarik arus, ditambah kondisi rangkaian yang sudah padat sehingga mudah terjadi kontak yang tidak diinginkan. Di tahap ini, perangkat masih bisa menyala, tetapi

output motor sering tidak konsisten dan kadang membuat sistem berperilaku tidak normal saat motor aktif.

Kendala lain yang cukup sering muncul adalah kualitas sambungan pada PCB bolong. Karena ini proyek perakitan awal, beberapa titik solder sempat mengalami sambungan kurang kuat, ada yang terlihat kusam, dan ada yang berpotensi menyentuh jalur lain. Efeknya terasa seperti masalah “acak”, misalnya sensor terbaca tapi indikator tidak respons, atau indikator menyala putus nyambung. Perbaikan dilakukan dengan inspeksi ulang semua jalur penting, terutama jalur daya, ground, dan jalur yang menuju output (buzzer, LED, motor). Titik yang meragukan disolder ulang dan diuji ulang dengan multimeter untuk memastikan tidak ada hubungan pendek dan koneksi benar benar solid.

Setelah perakitan lebih rapi, kendala berpindah ke tahap integrasi sistem, terutama saat perangkat baru menyala. Pada beberapa percobaan awal, notifikasi *SOS* dari Blynk bisa terkirim tanpa tombol ditekan karena proses sinkronisasi nilai widget saat perangkat baru *connect*. Ini diperbaiki dengan menambahkan proteksi awal (*safety startup*) sehingga sinyal tombol dari aplikasi diabaikan selama beberapa detik pertama setelah boot, dan tombol dipaksa kembali ke kondisi OFF saat koneksi. Selain itu, saat GPS belum langsung mendapatkan *lock*, sistem dibuat tetap mengirim pesan bahwa lokasi sedang diproses dan mengirim lokasi susulan ketika koordinat sudah valid. Setelah rangkaian perbaikan ini diterapkan, perangkat berjalan stabil dan seluruh fitur pada video demo dapat berfungsi sesuai rencana.

## **BAB VII**

### **PENUTUP**

#### **7.1 Kesimpulan**

Proyek ini berhasil merealisasikan sebuah *wearable smart navigational belt* berbasis ESP32 dengan integrasi sensor ultrasonik, modul MPU6050, aktuator *haptic*, tombol fisik GPS, serta konektivitas IoT melalui Blynk dan Telegram. Sistem dirancang untuk memberikan dukungan navigasi pada pengguna tunanetra sekaligus menyediakan mekanisme keselamatan berbasis notifikasi jarak jauh.

Fitur keselamatan bekerja dalam dua mekanisme utama, yaitu:

1. notifikasi otomatis yang dikirim ketika MPU6050 mendeteksi pola gerak yang mengindikasikan potensi jatuh, dan
2. notifikasi manual melalui tombol fisik GPS ketika pengguna secara sadar meminta bantuan.

Notifikasi dikirim ke kontak terdekat melalui Telegram, sehingga keluarga atau pendamping dapat segera mengetahui kondisi pengguna. Dari sisi navigasi, umpan balik *haptic* memungkinkan pengguna menerima informasi keberadaan halangan tanpa mengganggu persepsi auditori. Secara keseluruhan, hasil implementasi menunjukkan bahwa sistem ini telah memenuhi tujuan perancangan sebagai alat bantu navigasi sederhana yang portabel, praktis, dan dilengkapi fitur keamanan mandiri.

#### **7.2 Pengembangan Masa Depan**

Untuk kedepannya, sistem ini masih memiliki ruang pengembangan yang cukup luas baik dari sisi teknis maupun aspek kenyamanan penggunaan. Salah satu arah pengembangan yang potensial adalah penyempurnaan algoritma deteksi jatuh dengan pendekatan *multi-stage threshold* atau pemodelan berbasis *machine learning*, sehingga sistem mampu membedakan kejadian jatuh yang nyata dari gerakan tubuh normal dan dengan demikian menurunkan kemungkinan *false positive*. Selain itu, integrasi *data logging* dan histori lokasi berbasis *cloud platform* dapat menambah fungsi pemantauan jangka panjang, memungkinkan keluarga atau pendamping untuk meninjau rekam jejak aktivitas dan lokasi pengguna secara lebih komprehensif. Pada sisi perangkat keras, penggunaan modul daya yang lebih terintegrasi dan berukuran ringkas dapat

menggantikan powerbank eksternal agar perangkat semakin nyaman dipakai sebagai *wearable device*. Desain mekanik juga dapat terus dioptimalkan melalui reduksi bobot, peningkatan kontur ergonomis, serta pemilihan material yang lebih fleksibel dan tahan lama. Di masa mendatang, sistem ini juga berpotensi dikembangkan menuju integrasi sensor berbasis visi atau kamera dengan pemrosesan sederhana sebagai fitur opsional, sehingga deteksi rintangan dapat diperluas tanpa menghilangkan karakter utama alat sebagai solusi navigasi *haptic* yang ringan, praktis, dan mudah digunakan dalam aktivitas sehari-hari.

## **DAFTAR PUSTAKA**

- [1] R. Velázquez, "Wearable Assistive Devices for the Blind," in Wearable and Autonomous Biomedical Devices and Systems, A. Lay-Ekuakille and S. C. Mukhopadhyay, Eds. Berlin, Germany: Springer, 2010, pp. 331–350.
- [2] Espressif Systems, "ESP32 Series Datasheet," ver. 4.0, Shanghai, China, 2023. [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
- [3] S. Bhatlawande et al., "Electronic mobility aids for visually impaired: A review," in Proc. IEEE Int. Conf. Pervasive Comput. (ICPC), Pune, India, 2015, pp. 1–6.
- [4] InvenSense, "MPU-6000 and MPU-6050 Product Specification," ver. 3.4, Sunnyvale, CA, USA, 2013.
- [5] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," Neurocomputing, vol. 100, pp. 144–152, Jan. 2013.
- [6] u-blox, "NEO-6 u-blox 6 GPS Modules Data Sheet," ver. 1.6, Thalwil, Switzerland, 2011.
- [7] Telegram, "Bot API Documentation," 2024. [Online]. Available: <https://core.telegram.org/bots/api>.
- [8] Blynk Inc., "Blynk IoT Platform Documentation," 2024. [Online]. Available: <https://docs.blynk.io/en/>. (Referensi untuk implementasi Dashboard monitoring dan Virtual Pins).
- [9] Analog Devices, "Tilt Sensing Using a Three-Axis Accelerometer," Application Note AN-1057, Norwood, MA, USA, 2010.

## LAMPIRAN

### Lampiran 1: Kode Arduino Sketch

[https://drive.google.com/file/d/1x6\\_7T0r4NJI7Rqrf-s43UM8xFJ4aOdXP/view?usp=sharing](https://drive.google.com/file/d/1x6_7T0r4NJI7Rqrf-s43UM8xFJ4aOdXP/view?usp=sharing)