

**SMARTGUARD HELMET: SISTEM HELM PINTAR BERBASIS IOT
UNTUK DETEKSI KECELAKAAN
DAN PELACAKAN OTOMATIS**

SKRIPSI

DISUSUN OLEH

RAFLALRAYYIS

2209020020



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2026

**SMARTGUARD HELMET: SISTEM HELM PINTAR BERBASIS IOT
UNTUK DETEKSI KECELAKAAN
DAN PELACAKAN OTOMATIS**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi, pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

RAFI AL RAYYIS

2209020020

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN


2026

LEMBAR PENGESAHAN

Judul Skripsi : SMARTGUARD HELMET: SISTEM HELM PINTAR
BERBASIS IOT UNTUK DETEKSI KECELAKAAN
DAN PELACAKAN OTOMATIS

Nama Mahasiswa : RAFI AL RAYYIS
NPM : 2209020020
Program Studi : TEKNOLOGI INFORMASI

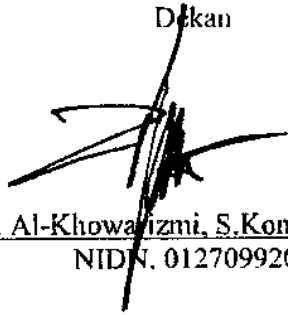
Menyetujui
Komisi Pembimbing


(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

Ketua Program Studi


(Fatma Sari Hutagalung, S.Kom., M.Kom.)
NIDN. 0117019301

Dekan


(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

SMARTGUARD HELMET: SISTEM HELM PINTAR BERBASIS IOT UNTUK
DETEKSI KECELAKAAN DAN PELACAKAN OTOMATIS

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, April 2026

Yang membuat pernyataan



Raf AL Rayyis

NPM. 2209020020

**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK
KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Rafi AL Rayyis
NPM : 2209020020
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**SMARTGUARD HELMET: SISTEM HELM PINTAR BERBASIS IOT
UNTUK DETEKSI KECELAKAAN
DAN PELACAKAN OTOMATIS**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, April 2026

Yang membuat pernyataan

Rafi AL Rayyis



NPM. 2209020020

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Rafi AL Rayyis
Tempat dan Tanggal Lahir : Medan 22 April 2004
Alamat Rumah : Jln Suka Ikhlas No.2
Telepon/Faks/HP : 081263667130
E-mail : rafialrasyid325@gmail.com
Instansi Tempat Kerja :
Alamat Kantor :

DATA PENDIDIKAN

SD : SDS Sarana Tunas Harapan TAMAT: 2015
SMP : Harapan Mandiri Medan TAMAT: 2018
SMK : Multi karya Medan TAMAT:2021

KATA PENGANTAR



Alhamdulillah, segala puji dan Syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan karunianya yang penuh dengan ilmu kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul tentang “SMARTGUARD HELMET: SISTEM HELM PINTAR BERBASIS IOT UNTUK DETEKSI KECELAKAAN DAN PELACAKAN OTOMATIS” untuk memenuhi persyaratan dalam jenjang strata satu dan mencapai gelar sarjana Komputer di jurusan Teknologi Informasi dan Ilmu Komputer, Universitas Muhammadiyah Sumatera Utara. Shalawat serta salam selalu tercurahkan kepada junjungan Nabi besar Muhammad SAW, keluarga dan sahabatnya yang syafaatnya kita nantikan diakhir zaman nanti. Dalam penyusunan Tugas Akhir ini, penulis telah mendapatkan banyak bantuan dan bimbingan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis tentunya berterimakasih kepada pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga berterimakasih kepada:

1. Bapak Prof. Dr. Agussani, MAP, selaku Rektor Universitas Muhammadiyah Sumatera Utara.
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi dan sekaligus dosen pembimbing.
3. Ibu Dr. Firahmi Rizky S.Kom.,M.Kom. selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi.

4. Bapak Mhd Basri, S.Si., M.Kom.selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi.
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. Ketua Program Studi Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
6. Bapak Okvi Nugroho,S.Kom., M.Kom. Sekretaris Program Studi Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
7. Untuk kedua orang tua saya yakni, Ayahanda Dr. Iwan Rudianto serta Ibunda Sri Rahayu yang telah memberikan dukungan serta doa yang tak pernah putus, pengorbanan yang tak terucap dan kasih sayangnya yang tulus dan tak terhingga kepada penulis.
8. Adik saya, Fareza Fatullah yang selalu memberi semangat, canda dan dukungan sederhana yang berarti besar di setiap proses yang penulis jalani.
9. Keluarga besar HIMATIF, Yang telah menjadi ruang belajar, bertumbuh dan menempa karakter. Pengalaman dan kebersamaan yang penulis dapatkan menjadi bekal berharga dalam menyelesaikan setiap proses, termasuk skripsi ini.
10. Penulis juga ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada pasangan penulis, Deby Syafira Samosir, atas segala dukungan, doa, perhatian, serta semangat yang senantiasa diberikan kepada penulis selama proses penyusunan skripsi ini. Kehadiran dan dukungan yang diberikan menjadi salah

satu sumber motivasi bagi penulis untuk terus berjuang hingga skripsi ini dapat terselesaikan dengan baik.

11. For myself, who has felt exhausted and almost given up, who silently carried burdens without saying much, who kept standing even when each step felt heavy. Thank you for not choosing to stop, even when this path often felt lonely and full of pressure. Thank you for continuing to move forward, even when the direction seemed uncertain and the steps were not always strong. This thesis is not merely the result of a research process, but proof of a long journey filled with struggle, failures, doubts, and the effort to keep going. May this achievement serve as a reminder that no effort is ever wasted, and as long as I do not give up, I am capable of overcoming whatever lies ahead.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna dikarenakan keterbatasan pengetahuan serta kemampuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun untuk menyempurnakan penulisan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi penulis khususnya dan bagi semua yang membutuhkan.

Wassalamu 'alaikum warahmatullahi wabarakaat

ABSTRAK

Penelitian ini bertujuan merancang SmartGuard Helmet, yaitu helm pintar berbasis Internet of Things (IoT) untuk mendeteksi kecelakaan dan mengirimkan lokasi secara otomatis. Sistem menggunakan ESP32, sensor accelerometer MPU6050 untuk mendeteksi benturan, serta modul GPS untuk pelacakan lokasi yang dikirim ke Arduino IoT Cloud secara real-time. Metode yang digunakan adalah threshold-based detection, di mana sistem akan mengirim notifikasi dan koordinat lokasi ketika terjadi benturan melebihi ambang batas. Hasil pengujian menunjukkan sistem mampu mendeteksi kecelakaan dan mengirim data lokasi dengan baik, meskipun masih memiliki keterbatasan dalam membedakan benturan biasa dan kecelakaan.

Kata Kunci: IoT Smart Helmet, Deteksi Kecelakaan; GPS, ESP32.

ABSTRACT

This study aims to design the SmartGuard Helmet, an Internet of Things (IoT)-based smart helmet capable of detecting accidents and automatically sending location data. The system utilizes an ESP32 microcontroller, an MPU6050 accelerometer sensor to detect impacts, and a GPS module for location tracking, which is transmitted to the Arduino IoT Cloud in real time. The method used is threshold-based detection, where the system sends notifications and location coordinates when an impact exceeds a predefined threshold. The results show that the system can effectively detect accidents and transmit location data, although it still has limitations in distinguishing between normal impacts and actual accidents.

Keywords: IoT Smart Helmet, Accident Detection, GPS, ESP32.

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERYATAAN ORISINILITAS	ii
PERYATAAN PERSETJUAN PUBLIKASI	iii
RIWAYAT HIDUP	iv
KATA PENGHANTAR	v
ABSTRAK	vii
ABSTRACT	vii
DAFTAR ISI	ii
DAFTAR TABEL	iv
DAFTAR GAMBAR	v
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	5
2.1 Iot (Internet of Things).....	5
2.2 Sensor Accelerometer.....	6
2.3 Mikrokonroller ESP32.....	7
2.4 Mikrokontroler NodeMCU.....	8
2.5 Gps Module.....	9
2.6 Flowchart.....	10
2.7 Arduino Iot Cloud.....	12
2.8 Penelitian Terdahulu.....	15
BAB III METODOLOGI PENELITIAN	17
3.1. Lingkup Penelitian.....	17
3.2. Alur Penelitian.....	18
3.3. Skema Rangkaian Alat.....	21
3.4. Rancangan Dashboard.....	26
BAB IV HASIL DAN PEMBAHASAN	28

4.1.	Hasil Penelitian	28
4.2.	Implementasi Perangkat Keras	30
4.3.	Penjelasan Algoritma Secara Konseptual.....	32
4.4.	Implementasi Arduino Iot Cloud.....	33
4.4.1	Setup Thing (Project Meta Data)	35
4.4.2	Struktur Program (Sketch ESP32).....	36
4.4.3	Konfigurasi Variabel Cloud.....	36
4.4.4	Propety Header.....	37
4.5.	Implementasi Pengujian	38
BAB V	PENUTUP	44
5.1.	Kesimpulan	44
5.2.	Saran.....	45
DAFTAR PUSTAKA	47

DAFTAR TABEL

Tabel 2.1. Simbol Flowchart.....	11
Tabel 2.2. Penelitian Terdahulu	15
Tabel 3.3. Skema Rangkaian Alat.....	22
Tabel 4.1. Pengujian Unit Testing	38

DAFTAR GAMBAR

Gambar 2.1. Bentuk Sistem IOT.....	6
Gambar 2.2. Sensor Accelerometer.....	7
Gambar 2.3. Pin-Pin ESP32.....	8
Gambar 2.4. Pin-Pin NodeMCU.....	9
Gambar 2.5 GPS Module V2.0 with External Antenna	10
Gambar 3.1. Lingkup Penelitian	17
Gambar 3.2. Alur Penelitian	19
Gambar 3.3. Skema Rangkaian Alat	22
Gambar 3.4. Monitoring Dashboard (Real – Time Tracker)	27
Gambar 4.1. Scematic Sistem Internet Of Things.....	30
Gambar 4.2. Tampilan Kondisi Aktif dan History saat Helm di Gunakan.....	35
Gambar 4.3. Dokumentasi Proyek (Setup Thing).....	36
Gambar 4.4. Struktur Kode Program (Sketch/Source Code).....	36
Gambar 3.7. Konfigurasi Variabel Cloud.....	37
Gambar 3.8. Property Header	37

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kecelakaan lalu lintas merupakan salah satu penyebab utama kematian di Indonesia. Berdasarkan data Korlantas Polri dan Badan Pusat Statistik (2023), lebih dari 70% kecelakaan lalu lintas melibatkan pengendara sepeda motor, dan sebagian besar korban meninggal dunia akibat keterlambatan pertolongan medis. Dalam banyak kasus, korban tidak segera ditemukan karena minimnya informasi lokasi kejadian serta tidak adanya sistem otomatis yang dapat memberikan notifikasi cepat kepada pihak keluarga atau layanan darurat.

Meskipun teknologi keselamatan pada kendaraan terus berkembang, sebagian besar masih berfokus pada mobil dan kendaraan roda empat. Sementara itu, pengendara sepeda motor yang memiliki risiko fatalitas lebih tinggi masih minim perlindungan teknologi yang efektif dan terjangkau. Helm sebagai alat keselamatan utama umumnya hanya berfungsi melindungi kepala dari benturan, tanpa memiliki kemampuan deteksi atau pelaporan otomatis.

Permasalahan ini menuntut adanya inovasi yang mampu mempercepat proses deteksi dan pelaporan kecelakaan secara real-time agar korban dapat segera mendapatkan bantuan. Teknologi Internet of Things (IoT) menjadi solusi potensial karena mampu menghubungkan perangkat ke jaringan internet dan mengirimkan data secara otomatis tanpa campur tangan manusia.

Berdasarkan hal tersebut, penelitian ini mengusulkan pengembangan SmartGuard Helmet, yaitu sistem helm pintar berbasis IoT yang mampu mendeteksi kecelakaan

sepeda motor menggunakan sensor accelerometer tunggal dan secara otomatis mengirimkan data lokasi korban melalui modul GPS dan jaringan internet. Sistem ini dirancang agar efisien, portabel, dan terjangkau, sehingga dapat diterapkan luas di masyarakat Indonesia. Dengan adanya sistem ini, diharapkan waktu tanggap pertolongan darurat dapat dipangkas secara signifikan dan tingkat keselamatan pengendara motor dapat meningkat.

1.2. Rumusan Masalah

Penelitian ini berupaya untuk menjawab bagaimana sistem tersebut dapat mengirimkan data lokasi kejadian secara real-time melalui modul GPS dan jaringan internet agar informasi dapat segera diterima oleh pihak keluarga atau layanan darurat sehingga proses pertolongan dapat dilakukan lebih cepat.

Permasalahan lain yang turut menjadi fokus adalah bagaimana meningkatkan efektivitas dan efisiensi sistem dalam mendeteksi benturan serta mengirimkan notifikasi otomatis, sekaligus menilai keakuratan dan kecepatan respon sistem dalam kondisi simulasi kecelakaan yang terkontrol. Dengan demikian, penelitian ini tidak hanya menitikberatkan pada aspek teknis perancangan perangkat, tetapi juga pada kemampuan sistem untuk memberikan kontribusi nyata terhadap peningkatan keselamatan pengendara sepeda motor di Indonesia, khususnya dalam mempercepat proses penanganan korban kecelakaan.

1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem hanya menggunakan satu sensor accelerometer sebagai alat utama untuk mendeteksi adanya benturan keras yang menjadi indikasi terjadinya kecelakaan.
2. Proses pengiriman data lokasi dilakukan secara otomatis melalui modul IoT seperti ESP32 atau NodeMCU yang terhubung dengan jaringan internet setelah sistem mendeteksi adanya benturan.
3. Data lokasi yang dikirimkan berupa koordinat GPS korban yang diperoleh dari modul GPS yang dipasang langsung pada helm.
4. Penelitian ini difokuskan pada perancangan dan pengujian prototipe sistem helm pintar, bukan pada desain fisik, bahan, atau struktur mekanik helm secara keseluruhan.
5. Sistem ini menggunakan platform Arduino IoT Cloud untuk mendukung komunikasi data secara multi-user dan mengirimkan notifikasi bahaya secara real-time kepada pihak penerima.
6. Pengujian sistem dilakukan dalam skala prototipe dan simulasi kecelakaan terkontrol, bukan dalam kondisi kecelakaan nyata di lapangan, demi menjaga keamanan dan keselamatan selama penelitian.
7. Penelitian ini tidak membahas integrasi sistem dengan kendaraan bermotor secara langsung, melainkan berfokus pada pengembangan sistem helm pintar yang bersifat mandiri dan portabel.

1.4. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Merancang dan membangun sistem helm pintar berbasis IoT yang dapat mendeteksi kecelakaan sepeda motor secara otomatis menggunakan sensor accelerometer tunggal.
2. Mengembangkan mekanisme pengiriman data lokasi korban secara real-time melalui jaringan internet agar pihak keluarga atau layanan darurat dapat segera mengetahui posisi kejadian.
3. Menguji keakuratan dan kecepatan sistem dalam mendeteksi benturan serta mengirimkan informasi lokasi secara otomatis.

1.5. Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut:

1. Menambah referensi dan pengetahuan dalam bidang *Internet of Things* (IoT), khususnya penerapannya pada sistem keselamatan transportasi.
2. Memberikan contoh nyata penerapan sensor accelerometer dalam sistem deteksi kecelakaan yang sederhana namun fungsional.
3. Mendukung perkembangan penelitian di bidang *Intelligent Transportation System* (ITS) di lingkungan akademik, terutama pada konteks negara berkembang seperti Indonesia.

BAB II

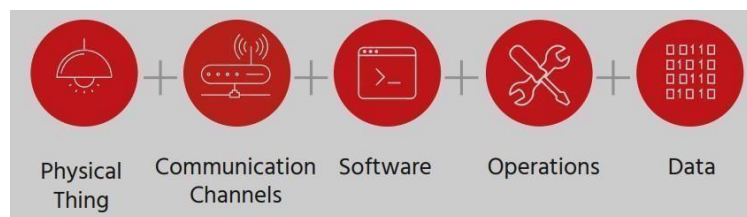
LANDASAN TEORI

2.1 Iot (Internet of Things)

Istilah "*Internet of Things*" terdiri dari dua kata utama, yaitu *Internet* yang berfungsi untuk menghubungkan dan mengatur konektivitas, serta *Things* yang merujuk pada objek atau perangkat. Secara sederhana, konsep ini memungkinkan berbagai perangkat atau objek (*Things*) untuk saling terhubung, mengumpulkan data, dan mengirimkannya ke jaringan internet. Data yang dikumpulkan juga dapat diakses dan digunakan oleh perangkat lain yang terhubung. Sebuah perangkat dalam sistem IoT memiliki kemampuan untuk mengirimkan data melalui jaringan kapan saja dan di mana saja, tanpa memerlukan interaksi langsung antara manusia maupun antara manusia dengan komputer (Selay A *et al.*, 2022).

Internet of Things (IoT) dapat diartikan sebagai kemampuan suatu objek dalam mentransmisikan atau mengirimkan data melalui jaringan tanpa memerlukan tindakan manusia atau perangkat komputer secara langsung. Istilah *Internet of Things (IoT)* pertama kali diperkenalkan oleh Kevin Ashton pada tahun 2002. Perkembangan teknologi internet mulai dimanfaatkan dalam proses produksi di sektor industri, khususnya di negara-negara maju seperti Amerika Serikat dan negara-negara di Eropa Barat. IoT bertujuan untuk membuat komputer dapat memahami lingkungan sekitarnya secara mandiri tanpa ketergantungan pada manusia. Pada awalnya, konsep IoT diterapkan dalam berbagai bidang dengan prinsip komputasi di mana saja dan untuk berbagai keperluan (*ubiquitous computing*). Namun, sejak tahun 2009, Komisi negara-negara Eropa dibentuk

untuk mengkaji ulang definisi IoT. Konsep ini kemudian berkembang menjadi tahapan evolusi baru dalam internet, yang awalnya hanya sekadar jaringan antar komputer yang saling terhubung, menjadi jaringan berbasis objek atau benda yang dapat berkomunikasi satu sama lain (Turyadi IU *et al.*, 2021).



Gambar 2.1. Bentuk sistem IoT (Turyadi IU et al., 2021)

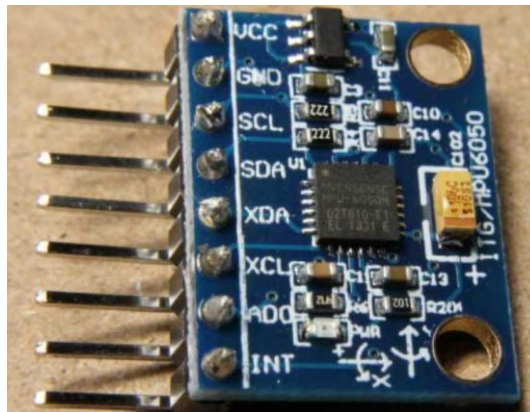
2.2 Sensor Accelerometer

Sensor accelerometer adalah perangkat elektronik yang berfungsi untuk mengukur percepatan (acceleration) suatu objek terhadap waktu. Percepatan yang diukur dapat berupa perubahan kecepatan linier atau arah gerakan suatu benda terhadap sumbu X, Y, dan Z. Sensor ini banyak digunakan untuk mendeteksi gerakan, kemiringan, getaran, atau benturan pada suatu sistem.

Accelerometer bekerja berdasarkan prinsip perubahan gaya inersia yang memengaruhi massa mikro di dalam sensor. Perubahan gaya ini akan menghasilkan sinyal listrik yang sebanding dengan nilai percepatan. Sinyal tersebut kemudian diolah oleh mikrokontroler (misalnya ESP32 atau Arduino) untuk mendeteksi kejadian seperti benturan keras yang mengindikasikan kecelakaan.

Dalam konteks penelitian *SmartGuard Helmet*, sensor accelerometer digunakan untuk mendeteksi benturan ekstrem pada helm pengendara. Ketika

percepatan melebihi ambang batas tertentu (*threshold*), sistem menganggap telah terjadi kecelakaan dan secara otomatis mengirimkan data lokasi melalui jaringan IoT (Turyadi, I. U & Nugraha, D., 2021).



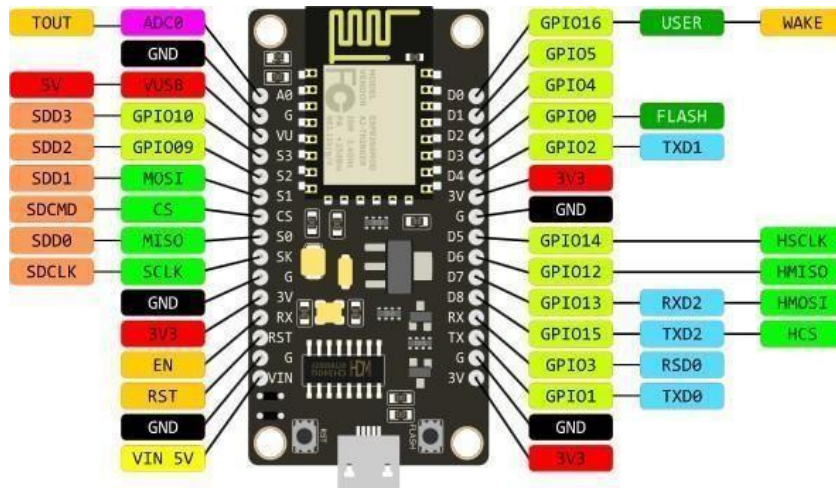
Gambar 2.2. Sensor Accelerometer

Sumber : (domainesia.com)

2.3 Mikrokontroler ESP32

Mikrokontroler ESP32 merupakan mikrokontroler SoC (System on Chip) yang terintegrasi dengan dukungan WiFi 802.11 b/g/n, Bluetooth versi 4.2, dan berbagai periferal lainnya. ESP32 adalah chip yang cukup lengkap karena sudah dilengkapi dengan prosesor, media penyimpanan, serta akses ke GPIO (General Purpose Input Output). Mikrokontroler ini dapat digunakan sebagai alternatif Arduino karena memiliki kemampuan untuk terkoneksi langsung ke jaringan Wi-Fi. Adapun spesifikasi dari ESP32, yakni tersedia dalam dua versi, yaitu 30 GPIO dan 36 GPIO. Kedua versi ini memiliki fungsi yang serupa, namun versi 30 GPIO lebih dipilih karena menyediakan dua pin GND. Semua pin telah dilabeli di bagian atas board, sehingga memudahkan dalam proses identifikasi. Selain itu, board ini

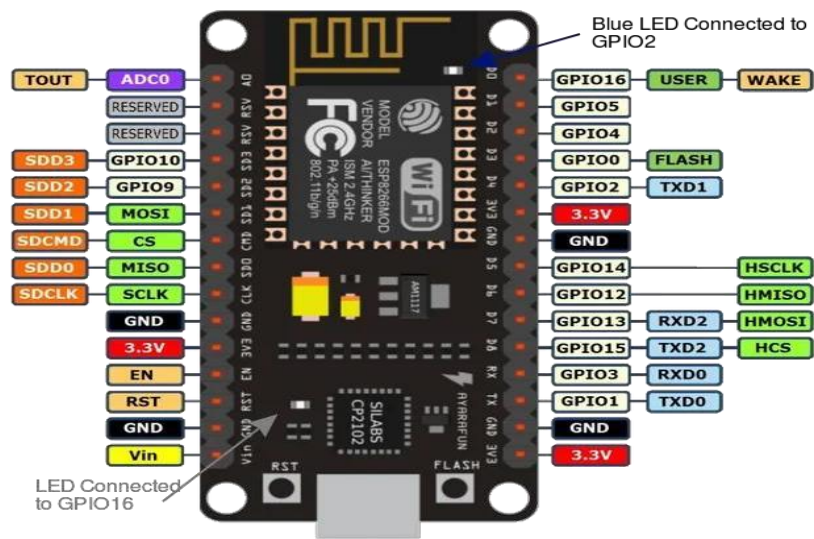
dilengkapi dengan interface USB to UART, yang membuatnya mudah diprogram menggunakan software pengembangan aplikasi seperti Arduino IDE. Untuk sumber daya, ESP32 dapat dioperasikan melalui koneksi micro USB (Nizam M *et al.*, 2022).



Gambar 2.3. Pin-Pin ESP32 (Prafanto A *et al.*, 2021)

2.4 Mikrokontroler NodeMCU

NodeMCU merupakan modul mikrokontroler berbasis ESP8266 yang dilengkapi dengan fitur konektivitas Wi-Fi terintegrasi, sehingga dapat digunakan untuk menghubungkan perangkat sensor dengan sistem berbasis internet secara langsung tanpa memerlukan modul tambahan, NodeMCU digunakan sebagai pengendali utama (main controller) dan penghubung antara sensor dengan server internet melalui jaringan Wi-Fi. Modul ini berfungsi untuk mengolah serta mengirimkan data hasil pembacaan sensor ke sistem cloud atau platform daring yang terhubung dengan aplikasi pemantauan (Turyadi, I. U., & Nugraha, D., 2021).



Gambar 2.4. Pin-Pin NodeMCU (Prafanto A *et al.*, 2021)

2.5 GPS Module

GPS (Global Positioning System) module adalah komponen elektronik yang berfungsi untuk menentukan posisi geografis (koordinat lintang dan bujur) suatu objek di permukaan bumi dengan memanfaatkan sinyal satelit. Modul GPS berisi chip penerima sinyal (receiver), antena, dan sistem pengolah data lokasi, yang memungkinkan perangkat seperti *NodeMCU* atau *ESP32* untuk membaca dan mengirimkan posisi secara real-time (Rahman, F., & Zainal, M., 2021).



Gambar 2.5. GPS Module v2.0 with External Antenna (AT6668)

2.6 Flowchart

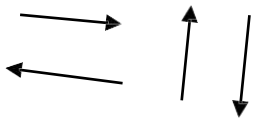

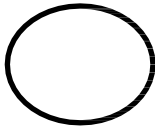
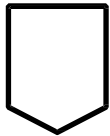

Flowchart merupakan representasi grafis yang menunjukkan langkah-langkah serta urutan prosedur dalam suatu program. Flowchart membantu analis dan programmer dalam memecah permasalahan menjadi bagian-bagian yang lebih kecil serta mengevaluasi berbagai alternatif dalam proses operasional. Diagram ini berperan dalam mempermudah penyelesaian masalah, terutama yang membutuhkan analisis dan evaluasi lebih mendalam. Flowchart berbentuk diagram dengan aliran satu atau dua arah secara berurutan dan digunakan untuk merepresentasikan serta merancang suatu program (Sutanti A *et al.*, 2020).


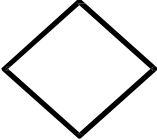
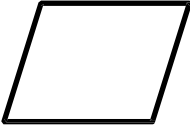

Menurut Wibanto (2017:20) dalam (Everaldo D et al., 2021) "Flowchart adalah sebuah diagram yang menggunakan simbol-simbol khusus untuk menggambarkan urutan proses secara rinci serta keterkaitan antara satu proses (instruksi) dengan proses lainnya dalam suatu program". Diagram alur ini dapat menunjukkan secara jelas bagaimana alur pengendalian suatu algoritma bekerja, yaitu bagaimana menjalankan serangkaian aktivitas secara logis dan sistematis.

Menurut Wibanto (2017:20) dalam (Everaldo D et al., 2021) "Flowchart

adalah sebuah diagram yang menggunakan simbol-simbol khusus untuk menggambarkan urutan proses secara rinci serta keterkaitan antara satu proses (instruksi) dengan proses lainnya dalam suatu program”.

Tabel 2.1. Simbol *Flowchart*

Simbol	Nama	Fungsi
	Flow Direction Symbol	Yaitu simbol yang di gunakan untuk menghubungkan simbol satu dengan simbol yang lain.
	Terminator Symbol	Yaitu simbol untuk permulaan atau akhir dari suatu kegiatan.
	Connector (on-page) Symbol	Yaitu simbol untuk keluar-masuk atau penyambungan proses dalam lembar yang sama.
	Connector (Off-Page) Symbol	Yaitu simbol untuk keluar masuk atau penyambungan proses pada lembar/halaman yang berbeda.
	Processing Symbol	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.

	<p>Symbol Manual Operation</p>	<p>Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.</p>
	<p>Symbol Decision</p>	<p>Yaitu simbol pemilihan proses berdasarkan kondisi yang ada.</p>
	<p>Symbol Input- Output</p>	<p>Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.</p>
	<p>Symbol Preperation</p>	<p>Yaitu simbol yang menyatakan penyediaantempat penyimpanan suatu pengolahan untuk memberikan nilai awal.</p>

2.7 Arduino IOT Cloud

Arduino IoT Cloud adalah platform berbasis awan (cloud) yang dikembangkan oleh Arduino untuk menghubungkan perangkat keras ke internet secara langsung dan memudahkan pengelolaan sistem Internet of Things (IoT). Platform ini berfungsi sebagai penghubung antara perangkat Arduino dengan dashboard digital interaktif yang dapat diakses dari mana saja. Dengan Arduino

IoT Cloud, data dari sensor pada perangkat dapat dikirim, disimpan, dan ditampilkan secara real-time tanpa memerlukan server pribadi atau pengaturan jaringan yang rumit (Yadav, A & Singh, P, 2023)

Dalam konteks judul skripsi “ SMARTGUARD HELMET : Sistem Helm Pintar Berbasis IoT untuk Deteksi Kecelakaan dan Pelacakan Otomatis,” Arduino IoT Cloud berperan sebagai pusat pengelolaan dan komunikasi data utama.

Helm pintar yang dilengkapi sensor accelerometer dan modul GPS akan mengirimkan data ke cloud melalui koneksi internet menggunakan modul Wi-Fi atau GSM. Data tersebut kemudian disimpan dan diproses di server Arduino, lalu ditampilkan secara langsung dalam bentuk informasi visual di dashboard IoT Cloud. Ketika sistem mendeteksi benturan keras atau perubahan posisi ekstrem yang menunjukkan adanya kecelakaan, data tersebut otomatis dikirim ke platform dan dapat memicu pengiriman notifikasi darurat kepada keluarga.

Salah satu keunggulan penting Arduino IoT Cloud yang sangat relevan bagi proyek SmartGuard Helmet adalah dukungan multi-user system. Fitur ini memungkinkan lebih dari satu pengguna untuk mengakses dan memantau data dari satu perangkat secara bersamaan. Misalnya, pengguna utama (pengendara) dan pengguna tambahan seperti anggota keluarga, operator pusat pemantauan, atau pihak keamanan dapat melihat data helm dalam waktu yang sama melalui akun berbeda. Hal ini menjadikan sistem helm pintar tidak hanya bersifat personal, tetapi juga kolaboratif dan terintegrasi dalam jaringan pengawasan bersama. Fitur multi-user ini memperluas cakupan pemantauan keselamatan dan mempercepat respon darurat, karena setiap pengguna yang terotorisasi dapat

langsung melihat data lokasi dan status pengendara tanpa harus menunggu konfirmasi manual (Arduino, 2025).

Selain itu, Arduino IoT Cloud menyediakan dashboard visual interaktif yang dapat dikustomisasi untuk setiap pengguna. Data lokasi GPS, status sensor benturan, atau kondisi perangkat dapat divisualisasikan dalam bentuk peta, indikator digital, dan grafik real-time. Semua informasi tersebut dapat diakses dari berbagai perangkat baik komputer, tablet, maupun ponsel melalui aplikasi Arduino IoT Remote. Dengan dukungan keamanan berbasis enkripsi TLS/SSL, setiap pertukaran data antar pengguna dan server tetap terlindungi.

Secara keseluruhan, Arduino IoT Cloud memungkinkan sistem SmartGuard Helmet beroperasi secara efisien, aman, dan fleksibel. Dukungan terhadap arsitektur multi-user membuat sistem ini cocok tidak hanya untuk penggunaan individu, tetapi juga untuk penerapan skala besar seperti pengendara, layanan ojek, hingga perusahaan logistik yang memantau banyak helm sekaligus. Dengan kemampuan menghubungkan banyak perangkat dan pengguna dalam satu jaringan cloud yang aman, Arduino IoT Cloud menjadi fondasi yang kuat untuk mewujudkan sistem helm pintar berbasis IoT yang modern, responsif, dan berorientasi pada keselamatan.

2.8. Penelitian Terdahulu

Adapun 5 penelitian terdahulu yang dapat saya ambil, sebagai berikut :

Tabel 2.2. Penelitian Terdahulu

No	Judul Penelitian	Pengarang (Tahun)	Kesimpulan / Relevansi dengan Penelitian Ini
1	Simplified IoT Accident Alert System Using Single Sensor	Hafiz, M., Rahman, A., & Singh, K. (2022)	Penelitian ini menggunakan satu sensor accelerometer untuk mendeteksi benturan dan mengirimkan lokasi kejadian melalui NodeMCU. Kelebihannya adalah rancangan yang sederhana dan murah, namun belum diuji di kondisi nyata dan belum terintegrasi dengan sistem darurat. Penelitian SmartGuard Helmet mengembangkan konsep ini dengan penempatan sensor di helm, sistem ambang batas dinamis, dan notifikasi langsung ke pihak keluarga.
2	IoT-Based Motorcycle Accident Detection and Notification System	Turyadi, I. U., & Nugraha, D. (2021)	Sistem menggunakan kombinasi accelerometer dan GPS untuk mendeteksi kecelakaan sepeda motor. Namun, kompleksitas sensor membuat biaya meningkat. Penelitian SmartGuard Helmet memperbaikinya dengan hanya menggunakan satu sensor accelerometer dan tetap menjaga akurasi deteksi.
3	Development of Smart Helmet Using IoT for Accident Detection	Selay, A., & Prasetyo, M. (2022)	Fokus pada helm pintar dengan modul IoT dan GPS untuk mengirimkan sinyal lokasi ketika terjadi benturan. Kelemahan penelitian ini terletak pada ambang batas deteksi yang masih statis. Penelitian ini dikembangkan lebih lanjut melalui sistem ambang batas dinamis untuk meningkatkan ketepatan deteksi.
4	Design of Low-Cost Accident Detection System for Two-Wheelers	Kumar, P., & Devi, S. (2020)	Penelitian ini menekankan aspek biaya rendah dalam pembuatan sistem deteksi kecelakaan dengan mikrokontroler Arduino dan sensor GY-61. Namun, sistem hanya berfungsi secara lokal tanpa koneksi IoT. SmartGuard Helmet melengkapi kekurangan ini dengan integrasi IoT untuk pelacakan otomatis berbasis internet.

No	Judul Penelitian	Pengarang (Tahun)	Kesimpulan / Relevansi dengan Penelitian Ini
5	IoT-Enabled Emergency Response System for Road Accidents	Rahman, T., & Alam, S. (2023)	Sistem IoT ini mampu mengirimkan data lokasi dan kondisi kecelakaan ke rumah sakit terdekat secara otomatis. Namun, penelitian ini lebih difokuskan pada kendaraan roda empat. SmartGuard Helmet mengadaptasi pendekatan serupa untuk sepeda motor, menyesuaikan dengan kondisi sosial dan infrastruktur Indonesia.

BAB III

METODOLOGI PENELITIAN

3.1. Lingkup Penelitian

Gambar di bawah menunjukkan rancangan fisik dari sistem SmartGuard Helmet, yaitu helm pintar berbasis Internet of Things (IoT) yang dirancang untuk mendeteksi kecelakaan dan melakukan pelacakan otomatis secara real-time. Desain helm menggunakan model full-face karena memberikan perlindungan maksimal terhadap kepala serta menyediakan ruang yang memadai untuk pemasangan komponen elektronik tanpa mengganggu kenyamanan pengguna.

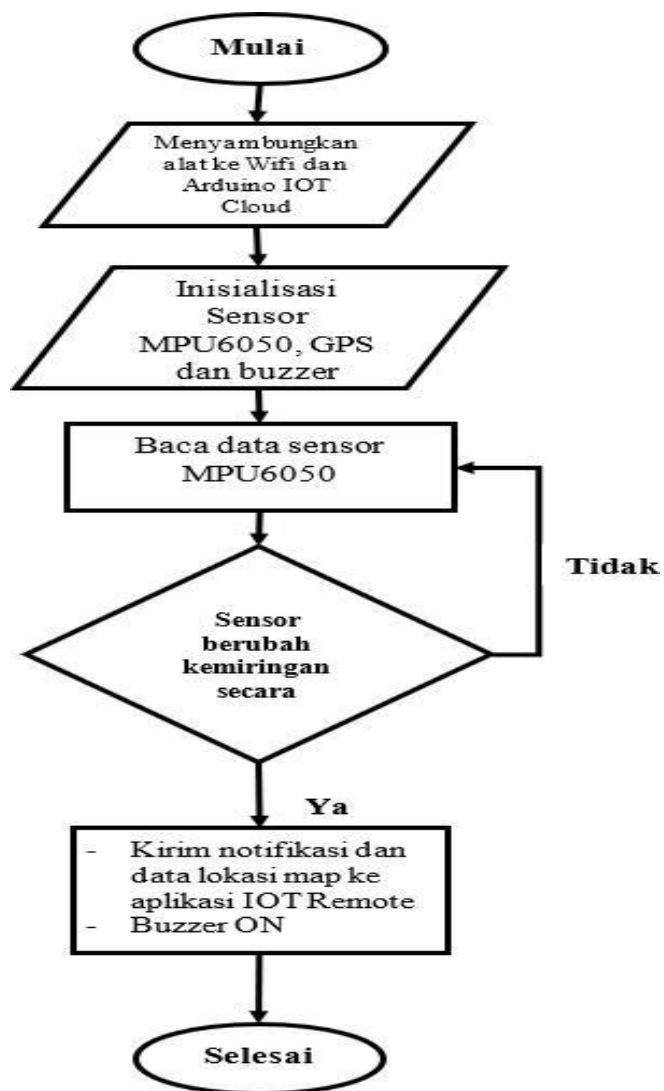


Gambar 3.1. Lingkup Penelitian

Komponen utama sistem ditempatkan pada bagian kanan helm, menyerupai posisi perangkat intercom, agar mudah dijangkau untuk proses pemeliharaan maupun pengaturan ulang perangkat. Penempatan ini juga mempertimbangkan keseimbangan bobot serta kenyamanan pengguna saat berkendara.

3.2. Alur Penelitian

Berikut ini adalah alur penelitian yang menggunakan, Arduino IOT Cloud, Sensor MPU 6050, Gps serta buzzer untuk membaca benturan pada helm jika terjadi kecelakaan ataupun helm itu jatuh. Dan berikut ini adalah gambar alur penelitian yang dapat kita lihat pada gambar 3.2.



Gambar 3.2. Alur Penelitian

Alur penelitian pada sistem SmartGuard Helmet diatas menggambarkan tahapan kerja alat yang dirancang untuk mendeteksi kecelakaan sepeda motor secara otomatis dan mengirimkan notifikasi lokasi kejadian melalui jaringan Internet of Things (IoT). Proses dimulai ketika sistem diaktifkan, di mana helm akan melakukan koneksi otomatis ke jaringan Wi-Fi yang telah dikonfigurasi sebelumnya dan tersambung dengan Arduino IoT Cloud sebagai media utama pertukaran data antara perangkat keras di helm dan aplikasi pemantauan berbasis IoT Remote. Setelah koneksi berhasil, sistem melakukan inisialisasi terhadap komponen utama, yaitu sensor MPU6050, modul GPS, dan buzzer, untuk memastikan semua perangkat siap bekerja sesuai fungsi masing-masing.

Ketika tahap inisialisasi selesai, sistem mulai membaca data dari sensor MPU6050 secara berkelanjutan untuk mendeteksi perubahan kemiringan, percepatan, atau guncangan ekstrem yang mungkin terjadi pada helm. Data dari sensor tersebut terus dipantau dan diolah oleh mikrokontroler ESP32. Jika sensor tidak mendeteksi adanya perubahan kemiringan yang signifikan, sistem akan tetap berada pada kondisi siaga dan melanjutkan pembacaan data secara terus-menerus. Namun, ketika sensor mendeteksi adanya perubahan kemiringan ekstrem yang melampaui ambang batas (threshold) yang telah ditentukan, sistem mengidentifikasi kondisi tersebut sebagai indikasi terjadinya kecelakaan.

Pada saat kejadian tersebut terdeteksi, ESP32 akan memproses data dan mengirimkan notifikasi darurat beserta koordinat lokasi GPS ke Arduino IoT Cloud secara real-time. Data ini kemudian diteruskan ke aplikasi IoT Remote

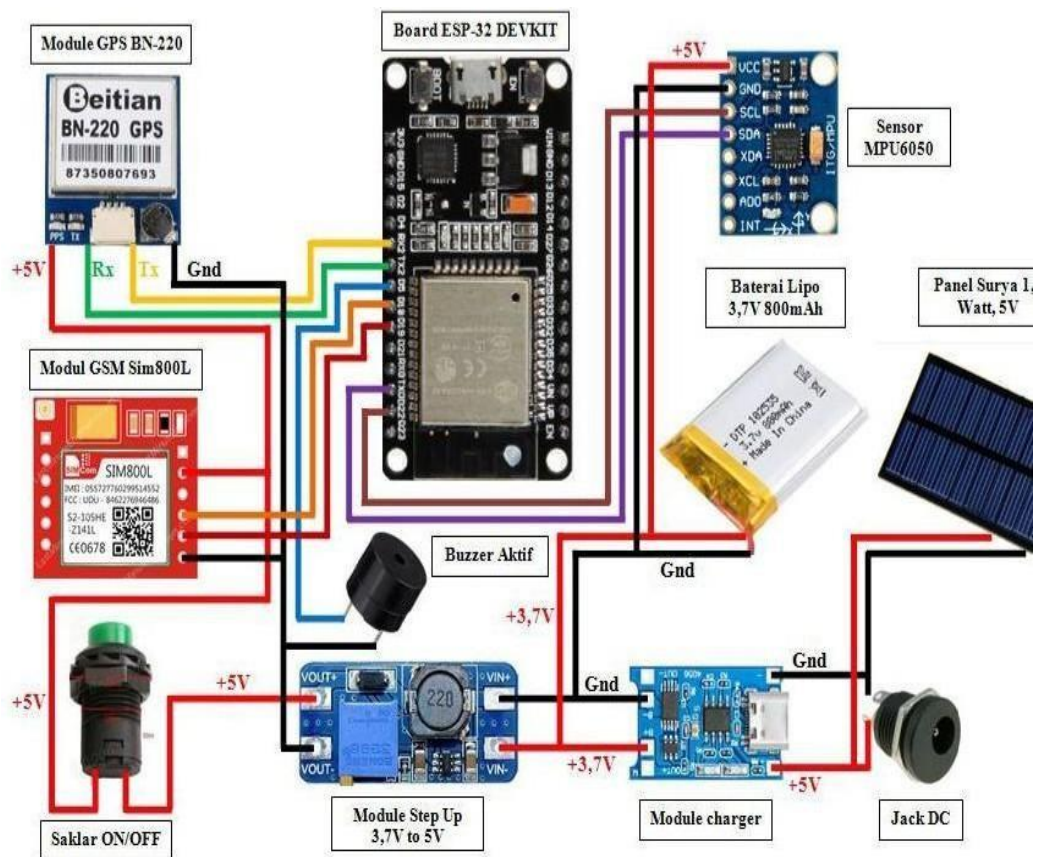
yang terhubung dengan akun keluarga atau pihak darurat, sehingga mereka dapat segera mengetahui lokasi korban dan mengambil tindakan cepat untuk memberikan pertolongan. Bersamaan dengan pengiriman data ke cloud, sistem juga mengaktifkan buzzer yang berfungsi sebagai alarm lokal di lokasi kejadian untuk menarik perhatian masyarakat sekitar agar segera memberikan bantuan.

Setelah proses pengiriman notifikasi dan aktivasi buzzer selesai, sistem masuk ke tahap akhir dari siklus kerja. Dalam tahap ini, alat dapat kembali ke mode siaga untuk mendeteksi kejadian berikutnya tanpa perlu intervensi manual, atau dapat dimatikan secara langsung oleh pengguna jika sudah tidak diperlukan. Dengan alur kerja ini, sistem SmartGuard Helmet mampu beroperasi secara mandiri, cepat, dan efisien dalam mendeteksi kecelakaan serta mengirimkan informasi lokasi secara otomatis.

Secara keseluruhan, alur penelitian ini menunjukkan bagaimana integrasi antara sensor accelerometer MPU6050, modul GPS, mikrokontroler ESP32, dan Arduino IoT Cloud dapat membentuk sistem yang cerdas dan responsif terhadap kondisi darurat. Rancangan ini diharapkan mampu menjadi solusi inovatif dalam meningkatkan keselamatan pengendara sepeda motor melalui penerapan teknologi IoT yang efektif, efisien, dan terjangkau bagi masyarakat luas.

3.3. Skema Rangkaian Alat

Skema rangkaian alat merupakan gambaran dari model alat yang akan digunakan. Skema ini mengintegrasikan mikrokontroler ESP32 sebagai otak pemrosesan data dari sensor accelerometer dan modul GPS guna mengimplementasikan sistem deteksi kecelakaan serta pelacakan lokasi secara *real-time* pada proyek SmartGuard Helmet berbasis IoT.



Gambar 3.3. Rangkaian Alat

Berikut ini tabel penjelasan dari skema rangkaian alat pada tabel 3.3. :

Tabel 3.3 Skema Rangkaian Alat

No.	Nama Komponen	Gambar/Label di Skema	Fungsi dan Penjelasan
1	Board ESP32 DEVKIT	Tengah (utama)	Berfungsi sebagai mikrokontroler utama yang mengontrol seluruh sistem, memproses data dari sensor, serta mengirimkan data ke Arduino IoT Cloud melalui koneksi Wi-Fi. ESP32 memiliki fitur Wi-Fi dan Bluetooth terintegrasi yang mendukung komunikasi IoT secara real-time.
2	Sensor MPU6050	Kanan atas	Sensor accelerometer dan gyroscope yang mendeteksi percepatan dan kemiringan helm. Data dari sensor ini digunakan untuk menentukan apakah terjadi benturan ekstrem yang mengindikasikan kecelakaan.
3	Module GPS BN-220	Kiri atas	Komponen yang berfungsi untuk menentukan posisi geografis (latitude dan longitude) secara akurat. Modul ini mengirimkan data lokasi korban ke sistem

No.	Nama Komponen	Gambar/Label di Skema	Fungsi dan Penjelasan
			IoT melalui ESP32 ketika terjadi kecelakaan.
4	Buzzer Aktif	Kiri tengah	Bekerja sebagai alarm atau indikator suara ketika sistem mendeteksi kecelakaan. Buzzer akan berbunyi otomatis untuk menarik perhatian orang di sekitar lokasi kejadian.
5	Saklar ON/OFF	Kiri bawah	Digunakan untuk menghidupkan dan mematikan sistem secara manual. Saklar ini menghubungkan atau memutus aliran daya utama dari baterai ke seluruh rangkaian alat.
6	Baterai Li-Po 3.7V 800mAh	Kanan tengah	Menjadi sumber daya utama bagi seluruh sistem. Baterai ini memberikan suplai tegangan 3.7V yang kemudian dinaikkan menjadi 5V melalui modul step-up agar sesuai dengan kebutuhan komponen seperti GPS dan buzzer.
7	Module Step-Up	Bawah tengah	Modul ini berfungsi untuk menaikkan tegangan output dari baterai Li-Po dari

No.	Nama Komponen	Gambar/Label di Skema	Fungsi dan Penjelasan
	(3.7V ke 5V)		3.7V menjadi 5V agar dapat digunakan oleh komponen lain yang memerlukan tegangan lebih tinggi, seperti ESP32 dan GPS.
8	Module Charger TP4056	Bawah kanan	Komponen pengisi daya baterai Li-Po yang mengatur arus pengisian agar baterai tidak overcharge. Modul ini terhubung dengan jack DC untuk proses pengisian daya eksternal.
9	Jack DC	Paling kanan bawah	Digunakan sebagai port pengisian daya eksternal untuk menghubungkan adaptor atau sumber daya listrik lain ke modul charger TP4056.
10	Kabel Jalur +5V, +3.7V, GND, dan Data (TX/RX, SCL/SDA)	Garis merah, hitam, dan kuning pada skema	Menunjukkan hubungan antar komponen dalam sistem. Jalur +5V (merah) mengalirkan daya ke komponen utama, GND (hitam) menjadi jalur ground, sedangkan jalur data (kuning/hijau) digunakan untuk komunikasi antar modul seperti GPS dan sensor.

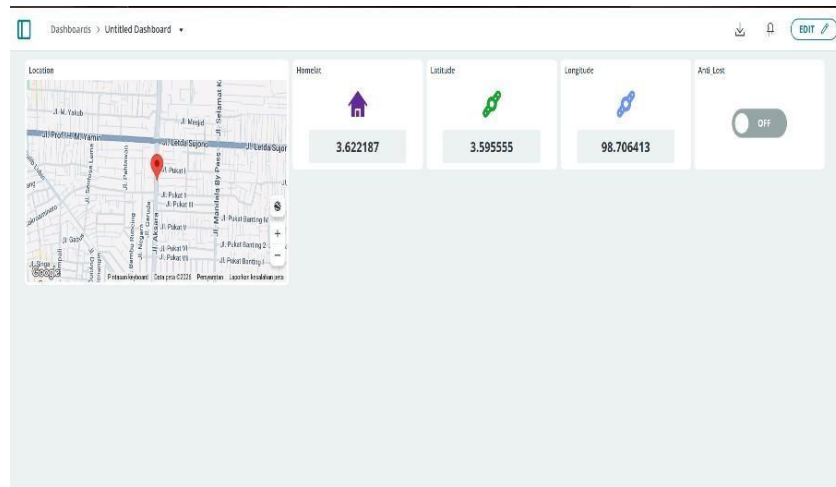
No.	Nama Komponen	Gambar/Label di Skema	Fungsi dan Penjelasan
11	Modul GSM Sim 800L	Kiri tengah	<p>Modul GSM SIM800L pada rangkaian tersebut berfungsi sebagai perangkat komunikasi berbasis jaringan seluler yang memungkinkan ESP32 untuk mengirim dan menerima informasi jarak jauh seperti SMS, panggilan, maupun data internet (GPRS), dengan komunikasi dilakukan melalui antarmuka serial UART (pin TX dan RX), di mana modul ini membutuhkan suplai tegangan stabil sekitar 3,7–4,2V dari baterai LiPo karena memiliki kebutuhan arus yang cukup besar saat proses transmisi, serta memanfaatkan SIM card aktif dan sinyal GSM agar dapat terhubung ke jaringan operator dan menjalankan perintah AT Command yang dikirim oleh ESP32 untuk mengontrol seluruh fungsi komunikasinya.</p>

3.4 Rancangan Dashboard

Fitur utama yang nantinya akan direncanakan yaitu:

1. Pelacakan Lokasi Real-Time: Menampilkan titik koordinat presisi pengendara di peta digital (Maps) yang datanya dikirim langsung oleh modul GPS ke dashboard Arduino IoT Cloud.
2. Deteksi Benturan Otomatis: Mengidentifikasi jika terjadi kecelakaan atau jatuh secara instan berdasarkan lonjakan nilai G-force yang dibaca oleh sensor Accelerometer.
3. Notifikasi Bahaya (SOS): Memberikan peringatan otomatis melalui cloud jika sistem mendeteksi adanya benturan keras, sehingga posisi kecelakaan bisa segera diketahui tanpa perlu bantuan manual dari korban.
4. Monitoring Status Perangkat: Menunjukkan kondisi operasional helm secara *real-time*, mulai dari status konektivitas ESP32 ke jaringan internet hingga kesiapan sensor dalam memantau perjalanan.
5. Log Riwayat Perjalanan: Menyimpan data pergerakan dan aktivitas sensor yang bisa diakses kembali untuk menganalisis kronologi kejadian jika sewaktu-waktu terjadi insiden di jalan.

Adapun tampilan yang direncanakan yaitu :



Gambar 3.4. Monitoring Dashboard (Real-time Tracker)

Gambar 3.4. menunjukkan dashboard utama berbasis cloud (kemungkinan besar Arduino IoT Cloud sesuai dokumen). Di sini terlihat widget untuk memantau koordinat GPS secara *real-time*. Fungsi: Menampilkan lokasi pengguna helm di peta digital. Data ini didapat dari GPS Module yang terpasang pada helm dan dikirim via ESP32/NodeMCU ke cloud.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Penelitian ini bertujuan untuk memantau keamanan pengendara sepeda motor menggunakan sistem deteksi kecelakaan dan pelacakan berbasis Internet of Things (IoT) yang dikembangkan dengan menggunakan sensor Accelerometer dan GPS Module. Sistem ini diharapkan dapat memberikan notifikasi *real-time* kepada keluarga atau pihak terkait, memungkinkan respon cepat terhadap kondisi darurat di jalan raya. Sistem yang dikembangkan terdiri dari beberapa komponen utama, yaitu:

1. Sensor Accelerometer

Sensor Accelerometer digunakan untuk mendeteksi guncangan atau benturan ekstrem yang dialami oleh pengendara. Sensor ini bekerja dengan mengukur perubahan percepatan pada sumbu X, Y, dan Z. Ketika terdeteksi lonjakan G-Force yang melewati ambang batas (*threshold*), sistem akan mengklasifikasikan kondisi tersebut sebagai kecelakaan atau jatuh.

2. GPS Module

Modul GPS berfungsi untuk mendapatkan titik koordinat presisi pengendara secara *real-time*. Data yang dihasilkan berupa Latitude (Lintang) dan Longitude (Bujur). Ketika sistem mendeteksi kecelakaan, data lokasi ini menjadi kunci utama untuk mengetahui di mana posisi korban berada secara akurat.

3. Platform Monitoring Berbasis Arduino IoT Cloud

Platform ini merupakan antarmuka pengguna (*user interface*) yang memungkinkan keluarga atau pihak berwenang untuk memantau kondisi pengendara secara *real-time*. Dashboard ini menampilkan informasi dari variabel cloud, di mana status sensor menunjukkan apakah kondisi dalam keadaan "Aman" atau "Bahaya". Berdasarkan data dari sensor GPS, sistem akan menampilkan posisi pengendara langsung pada widget peta digital. Platform ini berfungsi untuk menyajikan data sensor secara visual dan dapat diakses melalui perangkat komputer atau smartphone yang terhubung dengan internet.

4. Pemberian Notifikasi (Emergency Alert)

Sistem deteksi ini dilengkapi dengan fitur pemberitahuan otomatis yang akan aktif ketika terdeteksi kondisi yang membahayakan (benturan keras). Apabila sensor accelerometer mengirimkan sinyal bahaya, sistem akan otomatis mengirimkan notifikasi *real-time* beserta link lokasi koordinat ke dashboard pemantau. Fungsi utama dari fitur ini adalah untuk memberikan peringatan instan sehingga tindakan penyelamatan dapat segera dilakukan. Dengan adanya fitur notifikasi ini, sistem mampu meningkatkan responsivitas keselamatan bagi pengendara sepeda motor.

Hasil Implementasi: Hasil implementasi menunjukkan bahwa sistem berhasil membaca dan mengirimkan data sensor secara berkala ke server Arduino IoT Cloud, serta menampilkan informasi tersebut dalam bentuk grafik data sensor dan visualisasi peta. Sistem juga mampu mengirimkan peringatan jika terjadi kecelakaan. Penerapan sistem ini terbukti

meningkatkan rasa aman bagi pengendara serta memudahkan pihak keluarga dalam melakukan pemantauan lokasi secara berkelanjutan.

4.2. Implementasi Perangkat Keras

Rangkaian pada gambar 4.1 merupakan skema rangkaian dan fungsi dari bagian-bagian yang terdapat pada skema sistem Smart Guard Helmet : Sistem Helm Pintar Berbasis IOT Untuk Deteksi Kecelakaan Dan Pelacakan Otomatis yang telah dibuat:



Gambar 4. 1 Scematic Sistem Internet of Things

Integrasi arsitektur SmartGuard Helmet merepresentasikan sebuah ekosistem *Embedded IoT* yang kompleks, di mana sinkronisasi antara perangkat keras dan lapisan perangkat lunak bekerja secara simultan untuk menciptakan solusi keselamatan aktif. Pada level fisik, sistem ini mengandalkan mikrokontroler ESP32 sebagai unit pemrosesan utama yang mengeksekusi instruksi dalam struktur *multithreading*. Penggunaan ESP32 sangat strategis bagi mahasiswa Teknologi Informasi karena kemampuannya dalam menangani interupsi *real-time* dari sensor MPU6050 melalui bus komunikasi I2C. Sensor ini secara kontinu memantau enam derajat kebebasan (6-DOF) yang mencakup akselerasi linear dan kecepatan sudut. Data mentah tersebut kemudian diolah menggunakan algoritma filter sederhana di tingkat *edge* untuk memvalidasi apakah sebuah guncangan bersifat mekanis biasa atau merupakan impak benturan keras yang dikategorikan sebagai kecelakaan berdasarkan besaran *G-force* yang terdeteksi.

Keunggulan sistem ini terletak pada mekanisme *failover* komunikasinya yang sangat adaptif terhadap kondisi lapangan. Ketika modul GPS dengan antenna Ciocomm berhasil mengunci sinyal satelit (melalui proses *GPS fix*), data koordinat berupa *latitude* dan *longitude* akan terus diperbarui dalam memori volatili mikrokontroler. Jika logika sistem mendeteksi adanya kecelakaan, ESP32 akan segera menginisiasi transmisi data darurat. Dalam skenario di mana pengendara berada di area *blank spot* Wi-Fi, modul SIM800L akan mengambil alih peran konektivitas dengan memanfaatkan jaringan seluler GSM/GPRS. Proses ini melibatkan pengiriman paket data terenkripsi menuju broker MQTT pada platform Arduino IoT Cloud, sehingga informasi lokasi

korban dapat diakses secara instan oleh pihak terkait melalui *dashboard* pemantauan yang tersinkronisasi secara otomatis.

Dari sisi manajemen daya, SmartGuard Helmet mengadopsi prinsip *autonomous power management* dengan memanfaatkan panel surya fotovoltaik yang terintegrasi pada bagian *shell* atas helm. Energi surya yang dikonversi menjadi arus searah (DC) dialirkan menuju modul *charger* TP4056 untuk melakukan pengisian daya pada baterai Lithium secara teregulasi, mencegah terjadinya *overcharge* yang dapat membahayakan pengguna. Secara visual, implementasi kabel *jumper* dan pengaturan modul di dalam *enclosure* hitam yang terlihat pada gambar menunjukkan kerapian *prototyping* yang memperhatikan aspek distribusi beban dan proteksi terhadap interferensi elektromagnetik. Seluruh komponen ini bekerja dalam satu kesatuan logika *Sense-Think-Act*, di mana setiap data yang masuk disaring dan diproses dengan presisi tinggi guna memberikan respon keselamatan yang efisien, transformatif, dan berbasis data geospasial yang akurat.

4.3. Penjelasan Algoritma Secara Konseptual

Sistem membaca data percepatan pada sumbu X, Y, dan Z dari sensor MPU6050. Nilai tersebut kemudian diolah untuk menentukan apakah terjadi perubahan signifikan (benturan).

Jika nilai percepatan melebihi batas yang telah ditentukan, maka sistem menganggap terjadi benturan dan akan:

1. Mengaktifkan buzzer (alarm)
2. Mengambil koordinat GPS
3. Mengirim data melalui SIM800L

Namun jika tidak melebihi *threshold*, sistem tetap dalam kondisi monitoring.

Di sisi lain Adapun keterbatasan yang didapat dalam algoritma tersebut yaitu:

1. Tidak bisa membedakan benturan kecelakaan vs benturan biasa
2. Tidak mempertimbangkan kecepatan kendaraan
3. Sensitif terhadap perlakuan seperti helm dibanting
4. Bergantung pada nilai *threshold* (harus dikalibrasi)

4.4. Implementasi Arduino Iot Cloud

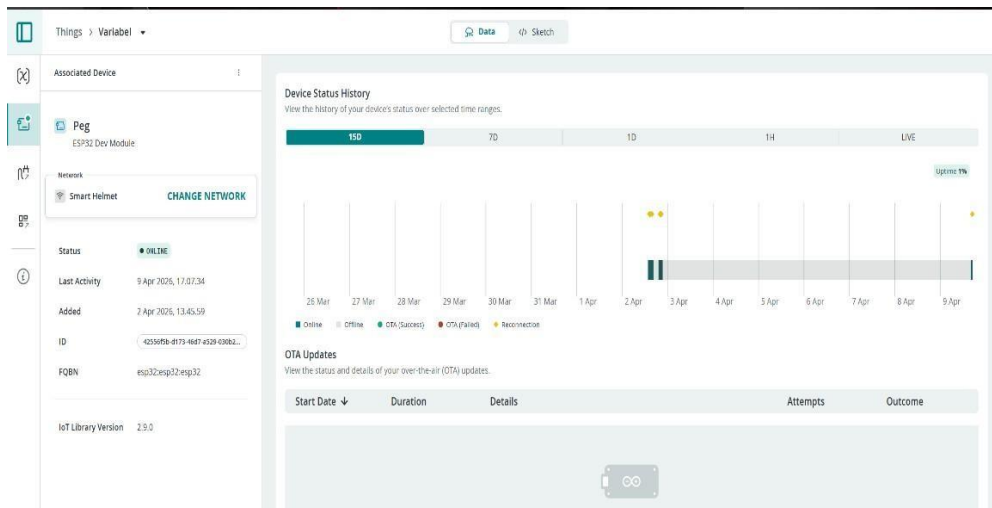
Implementasi sistem ini dimulai dengan mengintegrasikan *hardware layer* yang terdiri dari ESP32, sensor MPU6050, dan modul GPS ke dalam ekosistem Arduino IoT Cloud. Di sini, kita melakukan *provisioning* perangkat agar mikrokontroler bisa terhubung ke jaringan internet dan melakukan sinkronisasi variabel secara dua arah. Data dari sensor accelerometer diproses untuk mendeteksi *threshold* guncangan, sementara modul GPS secara kontinu mengekstraksi titik koordinat Latitude (Lintang) dan Longitude (Bujur). Semua data mentah ini kemudian ditransmisikan ke cloud melalui protokol komunikasi yang efisien agar bisa divisualisasikan langsung pada dashboard website.

Cara kerja website monitoringnya sendiri berfungsi sebagai antarmuka atau *Control Center* bagi pengguna atau pihak keluarga. Website ini secara *real-time* menarik data koordinat dari cloud dan memplotnya ke dalam widget peta digital, sehingga posisi presisi pengendara bisa dipantau berdasarkan titik lintang dan bujurnya. Selain menampilkan lokasi, website juga menyajikan grafik tren aktivitas yang merepresentasikan data dari sensor accelerometer untuk memantau stabilitas pengendara. Jika sistem mendeteksi lonjakan guncangan yang melewati ambang batas kecelakaan, variabel status di cloud akan berubah

secara otomatis, memicu munculnya notifikasi bahaya pada dashboard, dan memberikan informasi lokasi kejadian secara instan untuk mempercepat proses evakuasi.

Nantinya, fitur utama yang direncanakan mencakup pelacakan posisi lintang dan bujur yang akurat melalui peta digital serta deteksi benturan otomatis menggunakan analisis data sensor accelerometer. Sistem ini juga akan memberikan informasi mengenai status konektivitas perangkat untuk memastikan helm tetap dalam kondisi *online* saat digunakan. Selain itu, dashboard akan menyediakan log riwayat perjalanan dan data aktivitas sensor yang tersimpan secara terstruktur, serta sistem peringatan darurat *real-time* yang aktif seketika saat parameter keamanan mendeteksi adanya potensi kecelakaan di jalan raya.

Berikut ini ditampilkan hasil dari implementasi Arduino IOT Cloud yang berfungsi sebagai antarmuka pengguna untuk memantau lokasi secara real-time. Website ini menampilkan status sensor, grafik pergerakan, serta notifikasi peringatan secara langsung dan dirancang agar mudah digunakan oleh pengguna. Dokumentasi hasil tampilan antarmuka website disajikan dalam bentuk tangkapan layar berikut ini sebagai bukti dari proses implementasi yang telah dilakukan.



Gambar 4.2 Tampilan Kondisi Aktif dan History Saat Helm di Gunakan

Papan pemantauan IoT yang ditampilkan pada gambar ini dirancang khusus untuk memantau aktivitas dan status helm aktif pada saat berkendara ataupun helm tersebut terjatuh, memberikan gambaran real-time serta riwayat singkat perjalanan yang di tempuh.

4.4.1. Setup Thing (Project Metadata)

Berikut gambar 4.3. merupakan Dokumentasi Proyek (Project Metadata) Ini adalah langkah paling awal di mana mendefinisikan identitas proyek di platform cloud. Fungsinya, Sebagai identitas unik perangkat. Tanpa dokumentasi/registrasi ini, server Arduino Cloud nggak akan tahu perangkat mana yang lagi ngirim data GPS atau deteksi kecelakaan.

```

1  :Author: smartquart_helmet
2  :Email:
3  :Date: 02/04/2026
4  :Revision: version#
5  :License: Public Domain
6
7  = Project: {Project}
8
9  Describe your project
10
11 == Step 1: Installation
12 Please describe the steps to install this project.
13
14 For example:
15
16 1. Open this file
17 2. Edit as you like
18 3. Release to the World!
19
20 == Step 2: Assemble the circuit
21
22 Assemble the circuit following the diagram layout.png attached to the sketch.
23
24 == Step 3: Load the code
25
26 Upload the code contained in this sketch on to your board

```

Gambar 4.3. Dokumentasi Proyek (Setup Thing)

4.4.2. Struktur Program (Sketch ESP32)

Setelah itu di gambar 4.4. merupakan Struktur Kode Program yang berfungsi untuk Mengolah input dari sensor menjadi informasi. Fungsinya buat memerintahkan ESP32 kapan harus ngirim sinyal bahaya ke cloud dan kapan harus ngirim koordinat lokasi terbaru.

```

// --- BAGIAN IMPORT LIBRARY ---
#include <TinyGPS++.h> // Library untuk mengolah data dari modul GPS
#include "thingProperties.h" // Konfigurasi koneksi ke Arduino IoT Cloud
#include <HardwareSerial.h> // Library untuk komunikasi serial hardware (khususnya ESP32)
#include <Adafruit MPU6050.h> // Library sensor MPU6050 (akselerasi & gyro)
#include <Adafruit_Sensor.h> // Library dasar sensor Adafruit
#include <Wire.h> // Library untuk komunikasi I2C (untuk MPU6050)

// --- DEFINISI PIN & OBJEK ---
#define SIM800L_RX 18 // Pin RX untuk modul SIM800L
#define SIM800L_TX 19 // Pin TX untuk modul SIM800L
HardwareSerial SerialAT(1); // Menginisialisasi komunikasi Serial1 untuk SIM800L

TinyGPSPlus gps; // Objek untuk memproses data GPS
Adafruit MPU6050 mpu; // Objek untuk sensor MPU6050
int buzzer = 4; // Pin buzzer dihubungkan ke Pin 4
int alarms = 0; // Variabel status alarm (0=mati, 1=bunyi)
double homeLat, homeLon; // Variabel penyimpan koordinat awal (titik aman)
int isArmed = 0; // Variabel status pengaman (1=aktif, 0=nonaktif)
float radiusBatas = 5.0; // Batas toleransi pergerakan dalam meter

```

Gambar 4.4. Struktur Kode Program (Sketch/Source Code)

4.4.3 Konfigurasi Variabel Cloud

Gambar 4.5. merupakan Konfigurasi Variabel Cloud yang berfungsi sebagai perantara data. Fungsinya Sebagai jalur sinkronisasi data posisi. Fungsinya agar setiap perubahan titik koordinat lintang dan bujur di jalan bisa terdeteksi dan dikirim secara otomatis ke dashboard tanpa jeda (real-time).

Name ↑	Declaration	Last Value	Last Update
Location	Location	{lat:"3.598",lon:"98.706"}	2 Apr 2026, 20:47:11
homeLat	homeLatitud	3.622	2 Apr 2026, 20:47:11
latitud	latitud	3.596	2 Apr 2026, 20:47:11
longitud	longitud	98.706	2 Apr 2026, 20:47:11
switching	switching	false	7 Apr 2026, 20:35:00

Gambar 4.5. Konfigurasi Variabel Cloud

4.4.4. Property Header (Thing Properties)

Pada gambar 4.6. merupakan Header Properti Sebagai jembatan penghubung (*bridge*). Header ini yang memastikan kodingan di mikrokontroler "nyambung" dengan variabel lintang, bujur, dan sensor yang ada di cloud.

```

1 // Code generated by Arduino IoT Cloud, DO NOT EDIT.
2
3 #include <ArduinoIoTCloud.h>
4 #include <Arduino_ConnectionHandler.h>
5
6 const char DEVICE_LOGIN_NAME[] = "42556f5b-d175-46e7-e529-093b2773e402";
7
8 const char SSID[] = SECRET_SSID; // Network SSID (name)
9 const char PASS[] = SECRET_OPTIONAL_PASS; // Network password (use for WPA, or use as key for WEP)
10 const char DEVICE_KEY[] = SECRET_DEVICE_KEY; // Secret device password
11
12 void onHomeLatitudChange();
13 void onLatitudChange();
14 void onLongitudChange();
15 void onSwitchingChange();
16 void onLocationChange();
17
18 float homeLatitud;
19 float latitud;
20 float longitud;
21 CloudSwitch switching;
22 CloudLocation Location;
  
```

Gambar 4.6. Property Header

4.5. Implementasi Pengujian

Pengujian dilakukan untuk memastikan bahwa sistem monitoring IoT smart guard helmet berfungsi sesuai dengan perencanaan dan dapat mendeteksi kondisi penting, termasuk pemantauan lokasi. Tahapan ini sangat krusial untuk mengevaluasi kesesuaian antara rancangan awal dan hasil implementasi, sehingga dapat meminimalkan kemungkinan kesalahan atau penyimpangan yang tidak diinginkan dalam pendeteksian dan pelaporan. Berikut tabel uji testing:

Tabel 4.1 Pengujian Unit Testing

Komponen Sistem	Parameter Pengujian	Metodologi & Skenario Uji	Hasil yang Diharapkan (Success Metric)
ESP32 Core Processor	Network Latency & Handshaking	Melakukan <i>booting</i> perangkat dan mengukur durasi waktu yang dibutuhkan SoC untuk melakukan autentikasi pada <i>Access Point</i> (Hotspot).	ESP32 berhasil terkoneksi ke Wi-Fi/GPRS dalam waktu < 10 detik dengan status LED indikator internal menyala stabil.
MPU6050 (Inertial Unit)	Threshold Validation & G-Force Impact	Melakukan simulasi jatuh bebas (<i>free fall</i>) dan kemiringan ekstrem >	Nilai akselerasi pada sumbu X, Y, Z berubah drastis

Komponen Sistem	Parameter Pengujian	Metodologi & Skenario Uji	Hasil yang Diharapkan (Success Metric)
		60 [°] untuk melihat perubahan nilai <i>raw data</i> pada <i>Serial Monitor</i> .	dan sistem berhasil mengklasifikasikan data sebagai "Accident Detected" saat melampaui ambang batas.
GPS Neo-6M/Cirocomm	Cold Start & Geolocation Locking	Pengujian di area terbuka (<i>outdoor</i>) untuk menghitung durasi waktu akuisisi sinyal satelit hingga mendapatkan data NMEA.	Sistem mendapatkan <i>Lock Signal</i> (LED biru berkedip) dan menampilkan koordinat <i>Latitude/Longitude</i> dengan tingkat akurasi presisi < 5 meter.

Komponen Sistem	Parameter Pengujian	Metodologi & Skenario Uji	Hasil yang Diharapkan (Success Metric)
SIM800L Module	Cellular Data Transmission	Mematikan koneksi Wi-Fi utama untuk memicu protokol <i>failover</i> menggunakan jaringan seluler (GSM/GPRS).	Modul berhasil melakukan registrasi ke jaringan operator (Telkomsel) dan mengirimkan data paket ke <i>cloud</i> tanpa terjadi <i>data loss</i> .
Arduino IoT Cloud Sync	Real-time Dashboard Telemetry	Memantau sinkronisasi variabel antara <i>hardware</i> dan <i>web dashboard</i> secara bersamaan saat sensor dipicu.	Variabel pada <i>cloud dashboard</i> terupdate secara <i>real-time</i> (latensi < 2 detik) mencerminkan kondisi fisik sensor di lapangan.

Komponen Sistem	Parameter Pengujian	Metodologi & Skenario Uji	Hasil yang Diharapkan (Success Metric)
Power Management (Solar)	Charging Efficiency	Meletakkan helm di bawah paparan sinar matahari langsung dan mengukur tegangan input pada modul TP4056.	Multimeter menunjukkan adanya aliran arus masuk ke baterai ($V > 3.7V$) yang mengonfirmasi fitur <i>energy harvesting</i> berfungsi.
Buzzer & LED (Output)	Emergency Alert Response	Memicu sensor kecelakaan dan mengukur respon komponen <i>output</i> sebagai peringatan bagi pengguna/sekitar.	Buzzer mengeluarkan frekuensi audio tinggi (alarm) dan LED merah berkedip seketika saat status bahaya terkonfirmasi.

1. Uji Unit (Unit Testing): Validasi Hardware Interface

Di level paling dasar, kita fokus ke komunikasi antar modul. Masalah klasik di proyek IoT kayak gini biasanya ada di *wiring* yang berantakan atau *noise* listrik.

- 1) Fokusnya: Kita harus mastiin bus I2C (buat sensor MPU6050) dan UART (buat GPS/SIM800L) nggak bentrok.
- 2) Realitanya: Seperti yang kelihatan di gambar box sirkuitmu, kabelnya lumayan padat. Di sini kita uji apakah *common ground*-nya sudah benar-benar stabil. Kalau sirkuitnya "kotor", data dari akselerometer bisa ngaco (*jitter*), yang ujung-ujungnya bikin *false alarm* seolah-olah helm jatuh padahal cuma kena getaran mesin.

2. Uji Integrasi (Integration Testing): Validasi Logika Firmware

Setelah semua komponen "ngobrol", sekarang kita uji otaknya (ESP32). Pakai skenario *Edge Case*, kondisi di mana sistem dipaksa berpikir dalam situasi sulit.

- 1) Skenarionya: "Gimana kalau sensor mendeteksi benturan hebat, tapi GPS-nya lagi *blind spot* (misal di bawah terowongan)?"
- 2) Logika IT-nya: Sistem nggak boleh *stuck*. Di sini kita uji fungsi *Last Known Location*. Jadi, meskipun koordinat saat itu belum dapet, sistem tetap harus bisa kirim titik terakhir yang tersimpan di memori sebelum sinyal GPS hilang. Ini yang membedakan proyek asal jadi sama sistem yang *reliable*.

3. Uji Reliabilitas Konektivitas: Mekanisme Failover

Karena helm ini perangkat *mobile*, tantangan terbesarnya adalah internet. Kita nggak bisa ngandelin Wi-Fi terus.

- 1) Fokusnya: Kita uji proses Handover. Pas kita keluar rumah dan sinyal Wi-Fi putus, seberapa cepat ESP32 bisa "pindah haluan" ke koneksi GPRS via SIM800L?
- 2) Targetnya: Kita mau proses transisi ini *seamless*. Jangan sampai ada paket data yang *lost* atau tertahan gara-gara proses autentikasi jaringan selulernya kelamaan. Kecepatan kirim data ke Arduino IoT Cloud di sini jadi harga mati, karena tiap detik itu berharga buat evakuasi.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mulai dari tahap perancangan, implementasi, hingga pengujian sistem, dapat disimpulkan bahwa sistem SmartGuard Helmet berbasis Internet of Things (IoT) telah berhasil dirancang dan diimplementasikan sesuai dengan tujuan penelitian. Sistem ini menggunakan mikrokontroler ESP32 sebagai pusat kendali yang terintegrasi dengan sensor MPU6050 untuk mendeteksi benturan, modul GPS untuk menentukan koordinat lokasi, serta modul GSM SIM800L sebagai media komunikasi dalam pengiriman data secara real-time.

Proses kerja sistem dimulai dari inisialisasi perangkat, kemudian pembacaan data sensor MPU6050 secara terus-menerus untuk mendeteksi perubahan percepatan atau kemiringan ekstrem. Metode yang digunakan dalam sistem ini adalah algoritma Threshold-Based Detection, di mana sistem akan membandingkan nilai percepatan dengan ambang batas tertentu. Ketika nilai tersebut terlampaui, sistem akan mengidentifikasi adanya benturan, mengaktifkan buzzer sebagai alarm, serta mengirimkan data lokasi ke platform IoT yang kemudian ditampilkan melalui aplikasi IoT Remote.

Berdasarkan hasil pengujian yang dilakukan dalam skenario simulasi, sistem mampu menjalankan fungsi utamanya, yaitu mendeteksi benturan dan mengirimkan informasi lokasi secara otomatis. Hal ini menunjukkan bahwa sistem telah berhasil menjawab rumusan masalah terkait bagaimana

mendeteksi kecelakaan dan mengirimkan lokasi secara real-time kepada pihak terkait.

Namun demikian, sistem yang dikembangkan masih memiliki beberapa keterbatasan, di antaranya belum mampu membedakan antara benturan akibat kecelakaan dengan benturan biasa, seperti ketika helm dibanting, serta belum dapat mendeteksi atau mengukur kecepatan kendaraan. Dengan demikian, sistem masih berfokus pada deteksi benturan tanpa mempertimbangkan parameter lain yang dapat meningkatkan akurasi deteksi kecelakaan.

Secara keseluruhan, penelitian ini telah berhasil mencapai tujuan yang diharapkan, yaitu membangun sistem helm pintar berbasis IoT yang mampu memberikan notifikasi dini dan informasi lokasi secara otomatis, meskipun masih diperlukan pengembangan lebih lanjut agar sistem menjadi lebih akurat, adaptif, dan dapat digunakan secara optimal dalam kondisi nyata.

5.2. Saran

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan terhadap sistem SmartGuard Helmet, masih terdapat beberapa aspek yang dapat dikembangkan untuk meningkatkan kinerja dan keakuratan sistem di masa yang akan datang. Salah satu hal yang perlu diperhatikan adalah pengembangan metode deteksi yang lebih cerdas, sehingga sistem tidak hanya bergantung pada nilai ambang batas (threshold), tetapi juga mampu membedakan antara benturan akibat kecelakaan dan benturan biasa seperti saat helm terjatuh atau terbentur secara tidak sengaja.

Selain itu, sistem yang dikembangkan saat ini belum mempertimbangkan parameter kecepatan kendaraan, sehingga disarankan untuk menambahkan

fitur pengukuran kecepatan, baik melalui integrasi data GPS maupun sensor tambahan lainnya. Dengan adanya parameter kecepatan, sistem diharapkan dapat memberikan analisis yang lebih akurat dalam menentukan kondisi kecelakaan.

Dari sisi implementasi, pengujian sistem sebaiknya dilakukan dalam kondisi yang lebih mendekati situasi nyata di lapangan agar dapat diketahui tingkat keandalan sistem secara lebih akurat. Hal ini penting mengingat pengujian yang dilakukan dalam penelitian ini masih terbatas pada simulasi terkontrol.

Pengembangan juga dapat dilakukan pada aspek perangkat lunak, khususnya dalam pembuatan aplikasi monitoring yang lebih optimal dibandingkan penggunaan IoT Remote, sehingga notifikasi yang diterima dapat lebih cepat, informatif, dan mudah dipahami oleh pengguna. Di sisi lain, desain perangkat keras juga perlu disempurnakan agar lebih ringkas, stabil, serta nyaman digunakan tanpa mengganggu aktivitas pengendara.

Dengan adanya pengembangan lebih lanjut pada berbagai aspek tersebut, diharapkan sistem SmartGuard Helmet dapat menjadi solusi yang lebih efektif, akurat, dan siap untuk diimplementasikan secara luas dalam meningkatkan keselamatan pengendara sepeda motor.

DAFTAR PUSTAKA

- Selay, A., & Prasetyo, M. (2022). PENGEMBANGAN HELM PINTAR MENGGUNAKAN INTERNET OF THINGS (IoT) UNTUK DETEKSI KECELAKAAN. *International Journal of Innovative Science and Research Technology*. Volume 7 Nomor 6. Halaman 115–121.
- Turyadi, I. U., & Nugraha, D. (2021). SISTEM DETEKSI DAN NOTIFIKASI KECELAKAAN SEPEDA MOTOR BERBASIS IoT. *Journal of Robotics and Control (JRC)*. Volume 2 Nomor 5. Halaman 487–494.
- Turyadi, I. U., Nugraha, D., & Widodo, A. (2021). INTERNET OF THINGS: ARSITEKTUR DAN APLIKASINYA. *Journal of Emerging Technologies and Innovative Research*. Volume 8 Nomor 8. Halaman 210–216.
- Sutanti, A., Rahayu, D., & Wijayanti, E. (2020). PERANCANGAN FLOWCHART DALAM PENGEMBANGAN SISTEM. *Journal of Information System Development*. Volume 5 Nomor 2. Halaman 45–52.
- Everaldo, D., Pratama, H., & Naufal, R. (2021). MODELLING FLOWCHART VISUAL UNTUK SISTEM TERTANAM. *Journal of Applied Informatics and Computing*. Volume 5 Nomor 4. Halaman 312–320.
- Nizam, M., Hasan, R., & Karim, M. (2022). IMPLEMENTASI MIKROKONTROLER ESP32 DALAM SISTEM PEMANTAUAN BERBASIS IoT. *Sensors*. Volume 22 Nomor 19. Halaman 7379.
- Prafanto, A., Hidayat, D., & Rahman, F. (2021). KONFIGURASI PIN DAN PENERAPAN NODEMCU SERTA ESP32 DALAM PROYEK IoT. *Journal of Electrical and Electronic Systems Research*. Volume 19 Nomor 1. Halaman 34–40.

- Rahman, F., & Zainal, M. (2021). *PEMANFAATAN MODUL GPS UNTUK PELACAKAN WAKTU NYATA DALAM APLIKASI IoT*. International Journal of Advanced Computer Science and Applications (IJACSA). Volume 12 Nomor 10. Halaman 560–566.
- Patel, S., & Choudhary, R. (2022). *INTEGRASI TELEGRAM BOT API DALAM SISTEM DARURAT BERBASIS IoT*. International Journal of Emerging Technology and Advanced Engineering. Volume 12 Nomor 4. Halaman 58–64.
- Yadav, A., & Singh, P. (2023). Arsitektur IoT Terintegrasi Cloud untuk Sistem Mobilitas Cerdas. *International Journal of Smart Devices and Applications*, 8(2), 44–57.
- Arduino. (2025). *Gambaran Umum Arduino IoT Cloud*. Arduino.cc.