



Embedded System

Rancang Bangun Sistem Keamanan Helm dan Berkendara berbasis Mikrokontroler

Syukurillah Paradiqma Demita¹, Dodon Yendri^{*2}, Rifki Suwandi³

^{1,2,3} Department of Computer Engineering, FTI Universitas Andalas, Limau Manis Limau Manis, Kec. Pauh, Kota Padang, Sumatera Barat, Indonesia

ARTICLE INFORMATION

Received: 5 September 2023
Revised: 19 Desember 2023
Available online: 30 April 2024

KEYWORDS

MQTT Protocol, Anemometer Sensor, ESP32-Cam, Face Detection, Face Recognition

CORRESPONDENCE

E-mail: dodon@it.unand.ac.id

A B S T R A C T

As a two-wheeled vehicle user, a problem that often occurs in the community is the loss of helmets when placed in the parking lot. This can happen due to lack of attention from security personnel at the parking location. Riders who do not use helmets can endanger themselves in the event of an accident. In addition to the use of helmets, speed in driving is one factor that needs to be considered because it has the potential to endanger yourself or others. This study aims to anticipate the occurrence of helmet theft and also provide safety in driving using survey research methods to determine the percentage of ease of use by users. Where the system designed consists of 4 functions, namely, detection of helmet location using the MQTT protocol where if the helmet position with the vehicle has a large distance of 40m then the helmet is detected to have been stolen so that it will issue a warning in the form of sound. The second system is vehicle speed detection using anemometer sensors, if the vehicle is detected to have a large speed of 50km / h then the helmet will issue a sound warning so that the driver can reduce driving speed. The third system is face detection around the helmet using ESP32-Cam, where if the helmet detects an unknown face through the face recognition feature with a distance of 15cm to 60cm from the helmet, the ESP32-Cam will take pictures. The last system is taking pictures periodically to find out the situation around the helmet. Based on research using a questionnaire with 30 samplers, a percentage of 74.95% was obtained, which means that the helmet is easy to use.

INTRODUCTION

Helm merupakan salah satu benda penting yang harus digunakan saat ingin berkendara karena merupakan pelindung diri yang mampu mengurangi risiko saat terjadinya kecelakaan. Sebagai pengguna kendaraan beroda dua, permasalahan yang cukup sering terjadi di masyarakat yaitu terjadinya kehilangan helm pada saat diletakkan di parkiran, misalnya di parkiran pasar, Rumah Ibadah, mall, sekolah, kampus, dan lainnya. Hal ini dapat terjadi karena kurangnya perhatian dari petugas keamanan di lokasi parkir.

Pencurian helm merupakan suatu tindakan yang dapat merugikan orang lain, salah satu contoh kerugian ini dialami oleh pemuda asal Solo, Jawa Tengah, di mana terjadinya kehilangan helm setelah mengikuti pengajian, sehingga dapat dikatakan bahwa pencurian helm ini tidak memandang lokasi[1]. Berdasarkan survei yang telah dilakukan oleh Kabinet Aktualisasi Aksi BEM KM Universitas Andalas yang diisi oleh 117 responden, 26 orang di antaranya pernah mengalami kehilangan barang seperti helm, dompet, motor, *handphone*, dan uang. Dari 26 orang tersebut 22 orang di antaranya mengalami kehilangan helm[2].

Kehilangan helm mengakibatkan pengendara tidak memenuhi aturan tentang Lalu Lintas dan Angkutan Jalan yang telah diatur pada Undang-undang No. 22 Tahun 2009. Selain itu, pengendara

yang tidak menggunakan helm juga dapat membahayakan dirinya sendiri jika terjadinya kecelakaan. Berdasarkan data yang dimiliki oleh Korlantas Polri sejak tanggal 1 Januari hingga 17 Februari 2022 terdapat 15.265 kasus kecelakaan di mana 76% di antaranya terjadi pada pengendara sepeda motor[3]. Kecepatan dalam berkendara merupakan salah satu faktor yang perlu diperhatikan karena berpotensi untuk membahayakan diri sendiri atau orang lain.

Beberapa peneliti sebelumnya telah melakukan penelitian terkait sistem keamanan pada helm, namun sistem ini belum dilengkapi dengan output berupa pemberitahuan langsung ke pemilik saat helm berada di luar jangkauannya [4]. Penelitian lainnya tentang keamanan helm juga telah dilakukan di mana sistem tersebut telah berbasis IoT namun belum dilengkapi dengan kamera yang dapat menangkap kondisi di sekitar helm secara *real-time* sehingga saat terjadinya kehilangan tidak diketahui siapa orang terakhir yang mendekati helm[5]. Sebelumnya juga sudah terdapat penelitian yang berhubungan dengan pengukuran kecepatan pada helm, namun pada penelitian tersebut masih menggunakan SMS sehingga belum dilengkapi dengan fitur terbaru, selain itu juga tidak adanya pemberitahuan secara langsung untuk pengendara saat sedang mengendarai sepeda motor[6]. Berdasarkan penelitian terkait ESP32-Cam wajah yang disimpan di dalam *database* dapat lebih dari 1 dan sensor kamera ESP32-Cam dapat bekerja dengan baik pada jarak 15 cm-40cm[7].

Berdasarkan paparan latar belakang tersebut maka pada penelitian ini dirancang sebuah sistem keamanan yang dapat mengantisipasi terjadinya pencurian helm dan juga memberikan keamanan dalam berkendara dengan adanya pemantauan kecepatan kendaraan yang dikendarai oleh pengendara. Di mana rancangan Tugas Akhir tersebut berjudul “Rancang Bangun Sistem Keamanan Helm dan Berkendara Berbasis Mikrokontroler,” dengan harapan helm ini dapat digunakan sebagai alat yang dapat mengatasi kerisauan pengguna kendaraan beroda dua ketika helm tersebut ditinggalkan di tempat parkir dan juga sebagai monitoring kecepatan dalam berkendara untuk menghindari terjadinya kecelakaan lalu lintas.

Helm

Kewajiban menggunakan helm bagi pengendara diatur melalui Undang-undang No. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan di mana setiap pengendara bermotor wajib dilengkapi dengan helm standar nasional Indonesia[8]. Helm merupakan pelindung diri yang dapat mengurangi risiko saat terjadinya kecelakaan.

Kecepatan Kendaraan

Keselamatan dalam berkendara merupakan salah satu faktor penting yang perlu diperhatikan saat mengendarai kendaraan. Oleh karena itu, kecepatan dalam berkendara harus diatur agar pengendara dapat mematuinya. Menurut Peraturan Menteri Perhubungan Republik Indonesia nomor 111 tahun 2015 tentang cara penetapan batas kecepatan, kecepatan paling tinggi saat berkendara di kawasan perkotaan yaitu 50 km/jam[9].

Face Detection

Face detection merupakan salah satu metode yang digunakan untuk melakukan pengenalan terhadap wajah manusia. Salah satu cara yang dapat dilakukan untuk melakukan face detection dengan menggunakan mikrokontroler ESP32-Cam, di mana dalam mikrokontroler tersebut sudah terdapat built in camera sehingga dapat di program dengan mudah untuk melakukan face detection[10].

Face Recognition

Face recognition merupakan salah satu teknik pengenalan wajah yang mana hasil tangkapan kamera akan dicocokkan dengan foto atau tekstur wajah yang ada dalam database. Untuk melakukan face recognition maka perlu melakukan deteksi wajah yang bertujuan untuk memperoleh akurasi yang lebih baik dan normalisasi citra wajah karena deteksi wajah menyediakan batas lokasi dan skala dari setiap citra wajah yang dapat terdeteksi[11].

ESP32

ESP 32 merupakan mikrokontroler lanjutan dari ESP8266. Pada mikrokontroler ini sudah terdapat modul WIFI di dalam chipnya sehingga sangat cocok digunakan untuk sistem aplikasi Internet of Things. ESP32 memiliki memori yang lebih besar dibandingkan dengan Arduino Uno yaitu sebesar 16 MB, namun tegangan yang dimiliki oleh ESP32 lebih kecil yaitu sebesar 3.3 Volt[12].

ESP32- Cam

ES32-CAM memiliki ukuran yang kecil serta memiliki modul kamera dengan konsumsi daya rendah. Mikrokontroler ini sudah

dilengkapi dengan bluetooth dan WiFi sehingga dapat digunakan untuk merancang sistem berbasis Internet of Things (IoT)[13]. ESP32-CAM memiliki library face detection dan face recognition sehingga sangat cocok digunakan untuk melakukan perancangan sistem.

Message Queue Telemetry Transport

Message Queue Telemetry Transport (MQTT) merupakan serial komunikasi machine to machine yang bekerja pada layer ketujuh. Metode yang digunakan pada protokol ini adalah publish dan subscribe. Di mana publish digunakan sebagai pengiriman data dan subscribe digunakan sebagai penerimaan data, keduanya harus berada di topik yang sama agar pesan MQTT yang dikirim oleh publish dapat diterima oleh broker kemudian diteruskan ke subscriber[14].

Mobile WiFi

Mobile WiFi (MIFI) merupakan perangkat yang dapat menghubungkan sistem dengan jaringan internet, MIFI digunakan pada sistem ini karena memiliki ukuran yang kecil dan tidak menggunakan listrik sehingga mudah dibawa ke mana saja. MIFI ini dapat digunakan untuk semua jenis provider sehingga dapat memilih sesuai dengan keinginan[15].

Modul GPS

Modul GPS merupakan modul yang digunakan untuk membantu mengetahui lokasi suatu daerah. GPS UBLOX NEO 6M merupakan salah satu modul GPS yang dapat digunakan untuk mendeteksi koordinat tersebut, di mana seri NEO-6 ini mampu berkinerja tinggi dengan biaya yang hemat. Pada penelitian ini menggunakan modul GPS UBLOX 6M dikarenakan memiliki keakuratan yang lebih baik, berdasarkan penelitian yang telah dilakukan Modul GPS Receiver Neo-M8N memiliki tingkat kecepatan mendapatkan data valid yang tinggi namun tidak terlalu tepat untuk menunjukkan koordinat aktualnya dibandingkan dengan GPS Receiver Neo-6M. Berdasarkan penelitian tersebut, GPS Receiver Neo-M8N menghasilkan pergeseran sejauh 8.7M dan Neo-6M menghasilkan pergeseran sejauh 1.75M[16].

Sensor Anemometer

Sensor anemometer merupakan alat yang digunakan untuk mendeteksi kecepatan angin yang ada di sekitar sensor tersebut, di mana sensor memiliki tiga buah mangkuk yang berfungsi untuk menghambat laju kecepatan angin[17]. Untuk melakukan perhitungan kecepatan angin menggunakan rumus kecepatan putaran berikut:

$$n = \frac{JH}{JC} \times 60 \quad (1)$$

Di mana :
 n = putaran (rpm)
 JH = jumlah hasil celah yang dibaca
 JC = jumlah celah counter

Ukuran baling-baling yang ada pada sensor mempengaruhi kecepatan angin yang dihasilkan sehingga perlu melakukan perhitungan kecepatan sudut dari baling-baling kipas tersebut, menggunakan rumus berikut:

$$\omega = 2 \times \pi \times f \quad (2)$$

Di mana :
 ω = kecepatan sudut (rad/s)
 $\pi = 22/7$ atau 3.14

$$f = \frac{vH}{JC} \quad (rps)$$

Dari hasil kecepatan sudut tersebut maka dapat ditentukan kecepatan angin menggunakan rumus berikut:

$$v = \omega \times r \quad (3)$$

Di mana : v = kecepatan angin (km/jam)
 r = jari-jari (m)

Dari rumus itu saja tidak cukup untuk mendapatkan kecepatan angin yang sesuai dengan kebutuhan, karena sensor anemometer memiliki desain yang berbeda-beda sehingga diperlukan kalibrasi pada sensor tersebut, sehingga untuk mengukur kecepatan angin dapat menggunakan rumus berikut:

$$v = \omega \times r \times \text{kalibrasi} \quad (4)$$

Modul DFPlayer Mini

Modul DFPlayer mini digunakan untuk memutar file suara atau audio yang mendukung format .mp3 dan .wav, di mana pada komponen ini juga dilengkapi dengan celah untuk mini SD Card yang dapat diisi dengan file suara[18]. Pada DFPlayer terdapat inisialisasi awal dan pengaturan level volume sebagai keluaran dari DFPlayer ke *speaker*. Di mana ukuran volume paling tinggi yaitu 30[19].

Speaker

Speaker merupakan komponen yang digunakan untuk mengeluarkan hasil pemrosesan berupa suara [20]. Di mana cara kerjanya dengan mengubah sinyal elektrik menjadi frekuensi suara dengan menggetarkan komponen membran dan udara sehingga menjadi gelombang suara[21].

Arduino IDE

Arduino IDE merupakan perangkat lunak yang digunakan untuk menulis program dan meng-upload program tersebut ke *board* mikrokontroler[22]. Penulisan kode pada Arduino IDE menggunakan bahasa pemrograman C, penulisan kode ini bertujuan untuk menjalankan sistem yang dirancang agar bekerja sesuai dengan program yang telah ditentukan. Jika tidak ada kode program, maka sistem tersebut tidak dapat berfungsi[23].

Kodular

Kodular merupakan salah satu platform open source berbasis web yang dapat digunakan untuk membuat aplikasi berbasis android dengan menggunakan konsep pemrograman berupa blok *programming*. Dengan menggunakan Kodular tersebut setelah aplikasi selesai dirancang dapat langsung dijalankan tanpa harus melakukan ekspor aplikasi terlebih dahulu[24].

Kuesioner Usability

kuesioner usability merupakan suatu teknik pengumpulan data atau informasi yang digunakan untuk meninjau suatu produk tertentu, untuk mengetahui pencapaian target dengan melakukan penyebaran kuesioner ke suatu kelompok orang atau organisasi.

Skala Likert

Metode skala likert merupakan salah satu cara untuk mempermudah melakukan perhitungan hasil akhir dari pengisian kuesioner, di mana masing-masing skala tersebut memiliki nilai tertentu[25]. Ada dua bentuk pertanyaan dalam skala likert, yaitu

pertanyaan positif dengan skor 5, 4, 3, 2, dan 1, serta pertanyaan negatif dengan skor 1, 2, 3, 4, dan 5[26].

Firestore

Firestore merupakan salah satu API dari Google yang dapat digunakan untuk melakukan penyimpanan data serta melakukan penyesuaian informasi pada aplikasi Android, IOS, serta Website. Firestore mendukung fitur *realtime database* yang dapat digunakan untuk perancangan aplikasi *realtime* dan dapat diperbarui secara otomatis jika terjadi perubahan data secara *realtime*, pada Firestore data disimpan sebagai JSON dan akan disinkronkan terus menerus untuk setiap *client*[27].

Google Spreadsheet

Google Spreadsheet merupakan salah satu program buatan Google yang dapat digunakan untuk pengolahan data berupa angka. Google Spreadsheet ini dapat diakses dari mana saja dan menggunakan apa saja, Google Spreadsheet ini juga dapat diakses oleh beberapa orang sekaligus. Google Spreadsheet merupakan aplikasi berbasis *cloud*, sehingga hasilnya nanti akan disimpan di dalam *cloud*[28].

METHOD

Jenis penelitian yang digunakan pada pembuatan tugas akhir ini adalah penelitian survei (*Survey Research*). Pada penelitian ini dilakukan pengambilan sampel berdasarkan suatu populasi dengan mengajukan beberapa pertanyaan survei melalui kuesioner sebagai alat untuk melakukan pengumpulan data. sampel dari penelitian ini ada 30 orang di mana terdiri dari 10 orang mahasiswa, 5 satpam, 5 penjaga parkir, 5 orang tua yang memiliki anak di bawah umur dan sudah diizinkan mengendarai kendaraan, dan 5 orang umum.

Rancangan Umum Sistem



Gambar 1 Rancangan Umum Sistem

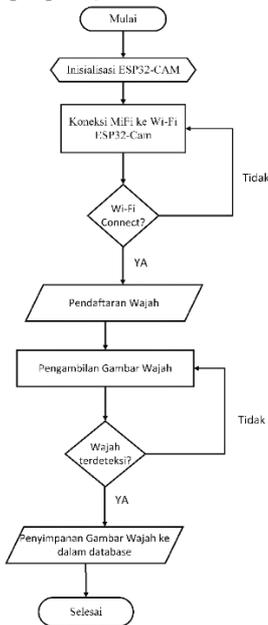
Rancangan umum sistem keamanan helm dapat dilihat pada Gambar 1, ESP 32 dan ESP32-Cam akan digunakan sebagai mikrokontroler untuk melakukan proses input dan mengeluarkan hasil output. Pada sistem ini menggunakan protokol MQTT sebagai serial komunikasi antar ESP32 di mana ESP32 dapat berfungsi untuk menerima pesan dan mengirimkan data. Untuk mendeteksi terjadinya perpindahan yang ada pada helm menggunakan serial Komunikasi MQTT di mana jika ESP32 *subscriber* tidak menerima pesan maka dianggap telah terjadi

perpindahan pada helm. Sensor anemometer pada sistem ini digunakan sebagai alat untuk mengukur kecepatan kendaraan saat dijalankan di jalan raya, semua input ini hasilnya akan di proses oleh ESP32.

Output yang digunakan pada sistem ini yaitu modul GPS di mana modul tersebut akan memberikan titik koordinat lokasi jika terjadinya perpindahan posisi pada helm secara real-time, selain itu GPS juga akan memberikan titik koordinat lokasi saat kendaraan yang dijalankan memiliki kecepatan di atas batas normal, di mana batas normal dalam berkendara tersebut adalah 50km/jam, titik lokasi ini nantinya akan dikirimkan ke aplikasi melalui jaringan internet yang difasilitasi oleh MIFI, Pada sistem ini juga terdapat kamera ESP32-Cam yang digunakan sebagai input, kamera pada sistem ini memiliki 2 tugas yaitu untuk melakukan pengambilan gambar ketika terdeteksi wajah yang mendekati helm dan wajah tersebut tidak ada di dalam *database* serta melakukan pengambilan gambar secara berkala sesuai dengan rentang waktu yang telah ditentukan. Output lainnya yang dimiliki oleh sistem ini berupa suara, untuk melakukan pemutaran suara menggunakan DFPlayer mini dan suara tersebut nantinya akan dikeluarkan oleh *speaker*.

Rancangan Perangkat Lunak

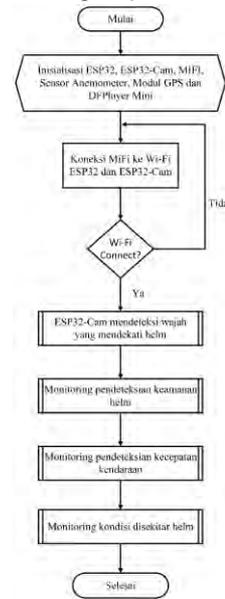
Tahapan pertama yang dilakukan dalam perancangan sistem ini adalah mendaftarkan wajah pengguna ke dalam *Database*, di mana berfungsi untuk melakukan pendeteksian wajah pada sistem. Adapun rancangan proses pendaftaran wajah pada ESP32-CAM terdapat pada *flowchart* berikut:



Gambar 2 *Flowchart* Pendaftaran wajah ke *database*.

Pada Gambar 2 tersebut terlihat alur dimulai dengan inisialisasi terhadap ESP32-Cam, kemudian melakukan koneksi Mobile WIFI, setelah ESP32-CAM terkoneksi dengan jaringan internet dimasukkan alamat IP Server yang berfungsi untuk mengakses browser yang akan digunakan, pada web server tersebut diaktifkan fitur pendaftaran wajah dan sistem akan melakukan pengambilan gambar melalui kamera ESP32-CAM, ketika wajah terdeteksi maka sistem akan melakukan pengambilan gambar dan menyimpannya ke dalam *database*.

Pemrosesan data yang terjadi pada sistem ini diolah oleh mikrokontroler ESP32 dan ESP32-Cam. Adapun rancangan proses umum digambarkan pada *flowchart* berikut:



Gambar 3 *Flowchart* keseluruhan sistem

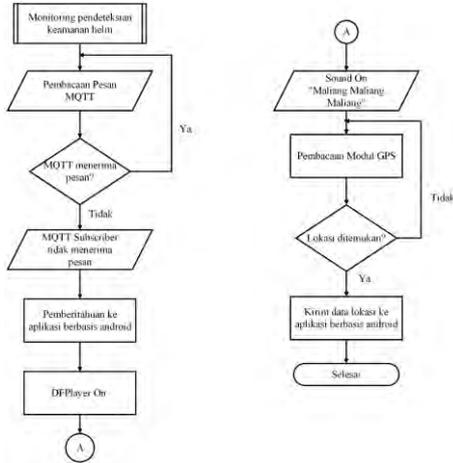
Pada Gambar 3 menjelaskan terkait alur program secara keseluruhan yang dimulai dengan melakukan inisialisasi terkait komponen-komponen yang digunakan yaitu ESP32, ESP32-Cam, Modul GPS, Sensor Anemometer, DFPlayer mini, dan MIFI. Kemudian dilanjutkan dengan proses penghubungan koneksi antara MIFI dengan mikrokontroler agar dapat mengirimkan notifikasi ke pengguna nantinya, jika koneksi belum terhubung maka akan dilakukan terus hingga jaringan internet terhubung. Jika koneksi internet sudah terhubung maka akan masuk ke proses selanjutnya sistem akan bekerja menggunakan ESP32-Cam untuk melakukan pendeteksian wajah yang mendekati helm menggunakan *face detection* dan pengenalan wajah menggunakan *face recognition*, dilanjutkan dengan pengecekan koneksi antar kedua ESP32 dengan menggunakan protokol MQTT untuk mengetahui posisi helm, dan mendeteksi kecepatan kendaraan menggunakan sensor anemometer, selain itu sistem ini juga melakukan monitoring kondisi di sekitar helm melalui ESP32-Cam secara berkala.



Gambar 4 Sub-proses pendeteksian wajah yang mendekati helm

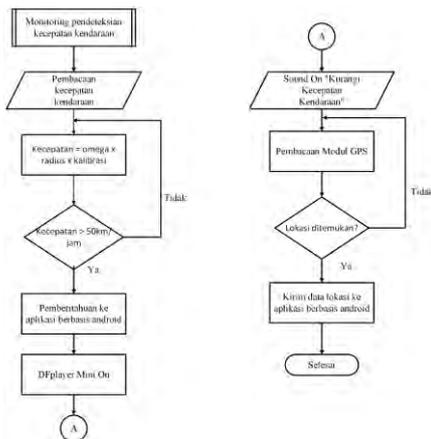
Pada Gambar 4 menjelaskan terkait diagram alur pendeteksian wajah yang mendekati helm, pendeteksian ini menggunakan *face*

detection yang terdapat di ESP32-Cam. Sebelumnya wajah pemilik helm telah didaftarkan ke dalam database. Sehingga jika wajah yang terdeteksi mendekati helm tidak dikenali maka ESP32-Cam akan langsung melakukan pengambilan gambar dan mengirimkan gambar tersebut ke aplikasi berbasis android, gambar tersebut akan disimpan di aplikasi sehingga pemilik helm dapat mengetahui siapa saja orang yang mendekati helm tersebut.



Gambar 5 Monitoring pendeteksian keamanan helm

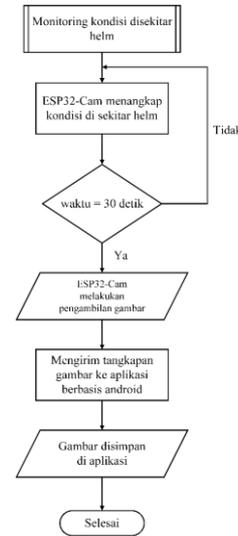
Pada Gambar 5 menjelaskan terkait alur monitoring pendeteksian keamanan helm, pendeteksian protokol MQTT, jika protokol MQTT subscriber tidak menerima pesan dari MQTT publish maka sistem mendeteksi telah terjadinya kehilangan helm sehingga sistem akan memberikan notifikasi ke ponsel pengguna melalui aplikasi berbasis Android, selain itu sistem juga akan menyalakan DFPlayer mini dan mengeluarkan suara “Maliang Maliang Maliang” melalui speaker agar orang sekitar mengetahui bahwa helm dibawa oleh orang lain. Sistem ini juga akan memberikan notifikasi lokasi helm kepada pemilik melalui aplikasi berbasis android agar pemilik dapat menuju ke lokasi helm tersebut.



Gambar 6 Monitoring pendeteksian kecepatan kendaraan

Pada Gambar 6 menjelaskan terkait diagram alur monitoring pendeteksian kecepatan kendaraan, di mana pendeteksian kecepatan kendaraan ini menggunakan sensor anemometer. Batas normal kecepatan kendaraan adalah 50km/jam sehingga jika kendaraan yang dikendarai memiliki kecepatan besar dari 50km/jam maka sistem akan mengirimkan notifikasi titik lokasi ke aplikasi berbasis android dan sistem juga akan menyalakan DFPlayer mini untuk mengeluarkan suara “Kurangi Kecepatan

kendaraan” melalui speaker agar pendengara dapat mengurangi kecepatannya. Sistem ini juga akan memberikan notifikasi titik lokasi di mana kecepatan dikendarai di atas batas normal saat dikendarai oleh pendengara.



Gambar 7 Monitoring kondisi di sekitar helm

Pada Gambar 7 menjelaskan terkait diagram alur monitoring kondisi di sekitar helm, di mana pada sistem ini saat kamera diaktifkan maka ESP32-cam akan menangkap kondisi di sekitar helm, jika sistem telah berjalan selama 30 detik maka akan dilakukan pengambilan gambar oleh kamera ESP32-Cam, di mana gambar tersebut akan dikirim ke aplikasi android dan akan disimpan di dalam aplikasi melalui Google Drive.

RESULTS AND DISCUSSION

Pada bagian ini membahas terkait implementasi dan hasil pengujian di mana pada tahap implementasi dibagi menjadi 3 bagian dan pengujian dibagi menjadi 4 bagian, sebagai berikut:

Implementasi perangkat keras

Perangkat keras yang digunakan pada sistem ini terbagi menjadi 2 bagian, bagian pertama diimplementasikan pada helm di mana berfungsi sebagai pengecekan kecepatan dalam berkendara dan penerimaan data untuk pengecekan posisi helm. Sedangkan bagian kedua diimplementasikan dalam box yang diletakkan di dalam bagasi motor dan berfungsi sebagai pengiriman data untuk pengecekan posisi helm.

Implementasi perangkat keras pada helm

Perangkat keras yang digunakan di sini terdiri dari ESP32, modul GPS, sensor anemometer, DFPlayer mini, speaker, powerbank, dan Mobile Wifi. Untuk implementasinya dapat dilihat pada Gambar 8 berikut:



(a) Tampak Luar Helm



(b) Bagian dalam Helm



(c) Detail Perangkat Keras Pada Helm

Gambar 8 implementasi perangkat keras pada helm

Keterangan Gambar 8:

1. *Power bank*, digunakan sebagai sumber daya sistem.
2. MIFI, digunakan sebagai sumber internet yang berfungsi untuk melakukan pengiriman data ke *database* dan aplikasi android.
3. ESP32, digunakan sebagai mikrokontroler untuk melakukan proses yang terdapat di dalam sistem. Selain itu, ESP32 pada sistem ini juga digunakan sebagai protokol MQTT *subscriber* dan protokol MQTT *publish*.
4. Modul GPS, digunakan untuk menentukan posisi helm saat ini.
5. Speaker, digunakan untuk mengeluarkan suara dari DFplayer mini.
6. DFplayer mini, digunakan sebagai pemutaran suara ketika kecepatan dikendarai dengan kecepatan besar dari 50km/jam atau ketika ESP32 *subscriber* tidak menerima pesan dari ESP32 *publish*.
7. Sensor Anemometer, digunakan untuk menghitung kecepatan kendaraan yang sedang dikendarai.
8. ESP32-Cam, digunakan sebagai pengambilan gambar jika terdeteksi wajah yang tidak dikenal mendekati helm dan pengambilan gambar secara berkala untuk mengetahui kondisi di sekitar helm.

Implementasi perangkat keras pada box

Perangkat keras yang digunakan di sini terdiri dari ESP32 dan baterai. Untuk implementasinya dapat dilihat pada Gambar 9 berikut:



Gambar 9 Implementasi perangkat keras pada box

Keterangan Gambar 9:

1. ESP32, digunakan sebagai protokol MQTT *publish* yang berfungsi untuk melakukan pengiriman pesan kepada Protokol MQTT *subscriber* yang terdapat pada helm.

2. Baterai, digunakan sebagai sumber daya dari sistem yang terdapat di dalam box.
3. Modul Step *Down*, digunakan untuk menurunkan tegangan baterai 12V menjadi 5V.

Implementasi Aplikasi Android

Kodular digunakan untuk membangun aplikasi android monitoring helm di mana pada aplikasi tersebut nantinya akan menampilkan status helm terkini, lokasi helm, hasil tangkapan gambar, serta riwayat. Berikut tampilan awal aplikasi android:



Gambar 10 tampilan awal aplikasi android

Implementasi Sistem

Implementasi dari sistem ini merupakan gabungan dari perangkat keras dan perangkat lunak, di mana hasil implementasi Rancang bangun sistem keamanan helm dan berkendara berbasis mikrokontroler ini dapat dilihat pada Gambar 11 berikut:



(a) Posisi Helm Saat Digunakan

(b) Posisi Helm Saat Tidak Digunakan

Gambar 11 Implementasi Sistem

Pengujian dan Analisa Perangkat Keras

Pengujian dan analisa perangkat keras dilakukan untuk melakukan pengecekan terkait komponen perangkat keras yang digunakan. Pengujian perangkat keras ini terbagi menjadi 4 bagian, yaitu:

Pengujian dan analisa Kebutuhan Daya

Pengujian sistem ini menggunakan 2 buah sumber daya yaitu *power bank* ROBOT RT190S yang mana memiliki kapasitas 10000mAh dengan output DC 5V/2.1A dan 5V/1A. Untuk menentukan ketahanan *power bank* pada sistem yang dirancang ini maka *power bank* tersebut dihubungkan pada kabel yang terdapat pada helm di mana kabel tersebut nantinya akan menyalurkan daya ke 2 buah ESP32, 1 buah ESP32-Cam, 1 buah

Sensor anemometer, 1 buah modul GPS, dan 1 buah DFPlayer mini beserta speaker.

Pengujian ini dilakukan untuk mengetahui ketahanan daya *power bank* jika dihubungkan dengan rangkaian sistem yang dirancang ketika sistem tersebut digunakan. Hasil pengujian ketahanan daya *power bank* dapat dilihat pada Tabel 1 berikut:

Tabel 1 Hasil percobaan ketahanan daya power bank

No.	Tanggal Percobaan	Waktu Sistem Dinyalakan (WIB)	Waktu Sistem Mati (WIB)	Waktu Ketahanan Daya
1.	19 Agustus 2023	08.00	17.23	9 Jam 23 Menit
2.	20 Agustus 2023	08.00	17.34	9 Jam 34 Menit
3.	21 Agustus 2023	08.00	17.11	9 Jam 11 Menit
4.	22 Agustus 2023	10.00	19.11	9 Jam 11 Menit
5.	23 Agustus 2023	11.00	20.27	9 Jam 27 Menit
Rata-rata ketahanan daya power bank				9 Jam 21 Menit

Berdasarkan pengujian yang telah dilakukan sebanyak 5 kali pada tanggal 19 Agustus 2023 sampai dengan tanggal 23 Agustus 2023 di dapatkan rata-rata ketahanan daya *power bank* per hari 9 jam 21 menit. Sehingga dengan mengetahui ketahanan daya ini maka pengguna helm dapat menyesuaikan sumber daya yang akan digunakan jika ingin menggunakan lebih dari rata-rata ketahanan daya tersebut.

Selain menggunakan *power bank* sistem ini juga menggunakan high discharge Lithium Polymer Battery yang diletakkan di dalam *box* pada bagasi motor, baterai Li-Po ini memiliki tegangan 11.1V dengan kapasitas 2200mAh. Untuk menentukan ketahanan daya baterai pada sistem ini baterai dihubungkan dengan modul *step down* untuk mengurangi tegangan yang ada pada baterai dan mengaturnya menjadi 5V kemudian dari modul *step down* tersebut dihubungkan ke ESP32 di mana saat modul *step down* dihubungkan tegangan yang terdapat pada ESP32 tersebut yaitu 4.81 V. Hasil ketahanan daya dari baterai dapat dilihat pada Tabel 2 berikut:

Tabel 2 Hasil percobaan ketahanan daya baterai Li-Po

No.	Tanggal Percobaan	Waktu Sistem Dinyalakan (WIB)	Waktu Sistem Mati (WIB)	Waktu Ketahanan Daya
1.	19 Agustus 2023	08.00	23.17	15 Jam 17 Menit
2.	20 Agustus 2023	08.00	23.10	15 Jam 10 Menit
3.	21 Agustus 2023	08.00	23.41	15 Jam 41 Menit
4.	22 Agustus 2023	10.00	01.13 (Keesokan harinya)	15 Jam 13 Menit

No.	Tanggal Percobaan	Waktu Sistem Dinyalakan (WIB)	Waktu Sistem Mati (WIB)	Waktu Ketahanan Daya
5.	23 Agustus 2023	11.00	02.31 (Keesokan harinya)	15 Jam 31 Menit
Rata-rata ketahanan daya baterai				15 Jam 22 Menit

Berdasarkan pengujian tersebut maka dapat diketahui bahwa komponen yang terdapat di dalam bagasi motor dapat menyala 15 jam 22 menit per harinya, baterai Li-Po dapat bertahan lebih lama dibandingkan dengan *power bank* dikarenakan komponen yang digunakan pada baterai hanya satu yaitu ESP32 yang memiliki tegangan rendah, selain itu perbedaan jenis sumber daya juga dapat mempengaruhi bedanya ketahanan daya pada sistem yang dirancang.

Pengujian dan analisa ESP32-Cam

Pada pengujian ESP32-Cam terbagi menjadi 2 bagian pengujian pertama berfungsi untuk melakukan pengukuran jarak ESP32-Cam agar dapat mendeteksi wajah yang mendekati helm dan pengujian kedua berfungsi untuk melakukan pengecekan ESP32-Cam dalam pengambilan gambar secara berkala sesuai dengan rentang waktu yang telah ditentukan.

Tabel 3 Hasil pengujian pengukuran jarak ESP32-Cam agar dapat mendeteksi wajah

Percobaan Ke-	Waktu Penerimaan Gambar di Aplikasi (WIB)	Jarak (cm)	Kamera ESP32-Cam
1	21:52:48	5	Tidak terdeteksi
2	21:53:33		Tidak terdeteksi
3	21:54:37		Tidak terdeteksi
4	21:55:42	10	Tidak terdeteksi
5	21:56:37		Tidak terdeteksi
6	21:56:55		Tidak terdeteksi
7	21:57:10	15	Terdeteksi
8	22:01:17		Terdeteksi
9	22:01:35		Terdeteksi
10	22:02:09	20	Terdeteksi
11	22:03:29		Terdeteksi
12	22:04:41		Terdeteksi
13	22:05:45	30	Terdeteksi
14	22:06:02		Terdeteksi
15	22:06:18		Terdeteksi
16	22:07:41	40	Terdeteksi
17	22:07:54		Terdeteksi
18	22:09:08		Terdeteksi
19	22:09:46	50	Terdeteksi
20	22:10:01		Terdeteksi
21	22:11:29		Terdeteksi
22	22:11:44	60	Terdeteksi
23	22:11:58		Terdeteksi
24	22:12:27		Terdeteksi
25	22:13:01	75	Tidak terdeteksi

Percobaan Ke-	Waktu Penerimaan Gambar di Aplikasi (WIB)	Jarak (cm)	Kamera ESP32-Cam
26	22:13:57		Tidak terdeteksi
27	22:14:32		Tidak terdeteksi

Berdasarkan Tabel 3 dapat dilihat telah dilakukan pengujian sebanyak 27 kali untuk pendeteksian wajah menggunakan kamera ESP32-Cam di mana jarak antara wajah dengan kamera di mulai dari 5cm-75cm, hasil pengujian menunjukkan bahwa kamera ESP32-Cam dapat mendeteksi wajah pada jarak 15cm hingga 60cm. Pada jarak 5cm hingga 10cm kamera tidak dapat melakukan pendeteksian, dikarenakan pada jarak tersebut hanya sebagian wajah yang terdeteksi oleh kamera sehingga dianggap tidak ada wajah yang terdeteksi. Sedangkan pada jarak besar dari 60cm kamera tidak dapat menangkap dengan jelas wajah yang terdeteksi oleh kamera hal ini disebabkan oleh ukuran wajah yang semakin kecil sehingga jumlah piksel yang mewakili wajah akan berkurang sehingga sulit sistem melakukan pengenalan wajah dengan akurat.

Tabel 4 Pengujian pengambilan gambar oleh ESP32-Cam secara berkala.

No.	Waktu pengambilan gambar (WIB)	Waktu penerimaan gambar di aplikasi (WIB)	Status Kamera ESP32-Cam	Selisih pengambilan gambar (Detik)	Waktu proses (Detik)
1	12:18:36	12:18:41	Berhasil	30	5
2	12:19:14	12:19:19	Berhasil	38	5
3	12:19:45	12:19:51	Berhasil	31	6
4	12:20:18	12:20:30	Berhasil	33	12
5	12:21:38	12:21:43	Berhasil	80	5
6	12:22:10	12:22:14	Berhasil	32	4
7	12:22:49	12:22:54	Berhasil	39	5
8	12:23:20	12:23:26	Berhasil	31	6
9	12:23:52	12:23:57	Berhasil	32	5
10	12:24:26	12:24:29	Berhasil	34	3

Berdasarkan hasil pengujian pengambilan gambar oleh kamera ESP32-Cam secara berkala, di mana pada programnya telah diatur bahwa pengambilan gambar oleh ESP32-Cam dilakukan setelah sistem berjalan selama 30 detik. Dapat dilihat sebelum dilakukan pengujian pada percobaan waktu, pengambilan gambar sebelumnya terjadi pada pukul 12:18:06 kemudian dilakukan pengujian pertama di mana waktu pengambilan gambar terjadi pada pukul 12:18:36 yang mana artinya selisih pengambilan gambar tersebut yaitu selama 30 detik, didapatkan rata-rata pengambilan gambar secara berkala yaitu selama 38 detik. Hal ini dapat terjadi karena jaringan internet yang tidak stabil sehingga proses pengiriman gambar ke aplikasi memakan waktu lebih lama dan mengakibatkan terjadinya delay untuk pengambilan gambar selanjutnya.

Pengujian dan analisa Sensor Anemometer

Pengujian terkait kecepatan kendaraan dilakukan menggunakan sensor anemometer, di mana sensor tersebut diletakkan pada bagian atas helm. Pengujian ini bertujuan untuk melakukan perbandingan antara nilai yang terdapat pada spidometer dengan nilai yang terdapat pada sensor anemometer. Data pengujian dapat dilihat pada tabel berikut:

Tabel 5 Pengujian kecepatan menggunakan sensor anemometer.

No.	Kecepatan Pada Motor (km/Jam)	Kecepatan Sensor Anemometer (km/Jam)	Selisih	Error (%)
1	10	10.342	0.342	3.42
2		10.342	0.342	3.42
3		10.342	0.342	3.42
4	20	20.683	0.683	3.42
5		20.683	0.683	3.42
6		20.683	0.683	3.42
7	30	25.854	4.146	13.82
8		22.407	7.593	25.31
9		24.130	5.870	19.56
10	40	36.196	3.804	9.51
11		39.643	0.357	0.90
12		39.643	0.357	0.90
13	50	48.261	1.739	3.48
14		53.432	3.432	6.86
15		53.432	3.432	6.86
16	60	53.432	6.568	10.95
17		53.432	6.568	10.95
18		53.432	6.568	10.95
Rata-rata Persentase Error				7.80

Pada Tabel 5 di atas, dilakukan 18 kali percobaan menggunakan sensor anemometer terdapat hasil yang berbeda-beda antara pengukuran menggunakan sensor anemometer dengan spidometer kendaraan. Sehingga terdapat nilai *error* pada pengujiannya, untuk melakukan pengukuran nilai *error* menggunakan rumus sebagai berikut:

$$Persentase\ error\ (\%) = \frac{Selisih\ pengukuran}{nilai\ pada\ spidometer} \times 100 \tag{5}$$

Berdasarkan pengujian tersebut juga terlihat kecepatan maksimal yang dapat dikur oleh sensor anemometer adalah 53.432 km/Jam sehingga jika kendaraan dikendarai dengan kecepatan besar dari 53.432 km/Jam maka kecepatan yang terbaca hanya 53.432km/Jam. Nilai yang terdapat pada sensor anemometer dengan yang terdapat pada spidometer kendaraan akan terdapat perbedaan yang tidak terlalu jauh hal ini dikarenakan nilai kecepatan yang terdapat pada spidometer diambil berdasarkan sensor kecepatan kendaraan di mana telah menyesuaikan dengan ukuran diameter roda pada kendaraan, sedangkan sensor anemometer diambil berdasarkan kecepatan angin yang terdeteksi saat kendaraan tersebut dikendarai di mana nantinya dipengaruhi oleh beberapa faktor seperti arah dan kekuatan angin. Sehingga, sensor anemometer tidak dapat mendeteksi tepat dengan nilai yang ditunjukkan oleh spidometer walaupun sensor anemometer tersebut telah dikalibrasi dan disesuaikan dengan spidometer.

Pengujian dan analisa Modul GPS

Pada percobaan ini terdapat 3 lokasi dengan pengumpulan data sebanyak 5 kali untuk masing-masing lokasi. Berdasarkan percobaan yang telah dilakukan data koordinat yang dihasilkan tidak selalu sama walaupun berada pada lokasi yang sama. Untuk itu perlu dilakukan perhitungan nilai *offset* menggunakan persamaan *Euclidean Distance*:

$$Jarak = \sqrt{(lat_{awal} - lat_{GPS})^2 + (long_{awal} - long_{GPS})^2} \tag{6}$$

Titik lokasi ini ditentukan melalui Google Maps Versi 11.81.1502 yang terdapat pada ponsel SAMSUNG A30S. Titik lokasi pertama berada pada nilai -0.9565161, 100.3748428 dengan

kondisi helm berada di dalam ruangan, titik lokasi kedua berada pada nilai -0.956516, 100.374843 dengan kondisi helm berada di luar ruangan dan diletakkan di lapangan, dan titik lokasi terakhir berada pada nilai -0.956542, 100.37504 dengan kondisi helm berada di luar ruangan dan diletakkan di bawah pohon.

Tabel 6 Hasil pengujian modul GPS titik lokasi pertama

No.	Koordinat (-0.9565161, 100.3748428)		Offset (m)
	Latitude	Longitude	
1.	-0.956530	100.374890	5,48
2.	-0.956540	100.374860	3,28
3.	-0.956550	100.374910	8,37
4.	-0.956570	100.374870	6,72
5.	-0.956590	100.374890	9,76
Total			33,61
Rata-rata = Total/Jumlah Data			6,72

Tabel 7 Hasil pengujian modul GPS titik lokasi kedua

No.	Koordinat (-0.956516, 100.374843)		Offset (m)
	Latitude	Longitude	
1.	-0.956510	100.374820	2,65
2.	-0.956500	100.374820	3,12
3.	-0.956470	100.374820	5,72
4.	-0.956480	100.374860	4,43
5.	-0.956470	100.374860	5,46
Total			21,38
Rata-rata = Total/Jumlah Data			4,28

Tabel 8 Hasil pengujian modul GPS titik lokasi ketiga

No.	Koordinat (-0.956542, 100.37504)		Offset (m)
	Latitude	Longitude	
1.	-0.956580	100.375040	4,24
2.	-0.956590	100.375030	5,54
3.	-0.956580	100.375010	5,60
4.	-0.956570	100.375020	4,03
5.	-0.956560	100.375010	4,18
Total			23,59
Rata-rata = Total/Jumlah Data			4,72

Berdasarkan percobaan yang telah dilakukan pada 3 kondisi di atas, dapat dilihat bahwa nilai *offset* tertinggi yaitu 9.76 meter berada pada lokasi di dalam rumah. Sedangkan pada titik lokasi di luar ruangan helm memiliki nilai *offset* yang lebih rendah. Sehingga dapat disimpulkan bahwa titik lokasi cenderung lebih akurat jika helm diletakkan pada lokasi yang terbuka hal ini disebabkan oleh ruangan yang terbuka memudahkan modul GPS menangkap titik koordinat dari satelit.

Pengujian dan analisa DFPlayer Mini dan Speaker

Pengujian ini dilakukan untuk mengetahui keakuratan DFPlayer Mini dan *Speaker* pada sistem yang telah di rancang, di mana DFPlayer mini akan melakukan pemutaran suara untuk 2 kondisi, yaitu ketika kecepatan kendaraan terdeteksi besar dari 50km/Jam dan ketika koneksi antara Helm dan Motor terputus. Berikut tabel pengujian DFPlayer mini dan *speaker*:

Tabel 9 Pengujian DFPlayer Mini dan *Speaker* pada kecepatan kendaraan

No.	Kecepatan Pada Motor (km/Jam)	Kecepatan Sensor Anemometer (km/Jam)	DFPlayer Mini	Speaker
1	10	10.342	OFF	OFF
2		10.342	OFF	OFF
3		10.342	OFF	OFF
4	20	20.683	OFF	OFF
5		20.683	OFF	OFF
6		20.683	OFF	OFF
7	30	25.854	OFF	OFF
8		22.407	OFF	OFF
9		24.130	OFF	OFF
10	40	36.196	OFF	OFF
11		39.643	OFF	OFF
12		39.643	OFF	OFF
13	50	48.261	OFF	OFF
14		53.432	ON	ON
15		53.432	ON	ON
16	60	53.432	ON	ON
17		53.432	ON	ON
18		53.432	ON	ON

Tabel 10 Pengujian DFPlayer Mini dan *Speaker* pada koneksi helm

No.	Jarak (m)	Status	DFPlayer Mini	Speaker
1	Helm diletakkan di bawah sadel motor	Terhubung	OFF	OFF
2		Terhubung	OFF	OFF
3		Terhubung	OFF	OFF
4	1	Terhubung	OFF	OFF
5		Terhubung	OFF	OFF
6		Terhubung	OFF	OFF
7	5	Terhubung	OFF	OFF
8		Terhubung	OFF	OFF
9		Terhubung	OFF	OFF
10	10	Terhubung	OFF	OFF
11		Terhubung	OFF	OFF
12		Terhubung	OFF	OFF
13	20	Terhubung	OFF	OFF
14		Terhubung	OFF	OFF
15		Terhubung	OFF	OFF
16	30	Terhubung	OFF	OFF
17		Terhubung	OFF	OFF
18		Terhubung	OFF	OFF
19	35	Terhubung	OFF	OFF
20		Terputus	ON	ON
21		Terhubung	OFF	OFF
22	40	Terhubung	OFF	OFF
23		Terputus	ON	ON
24		Terputus	ON	ON
25	45	Terputus	ON	ON
26		Terputus	ON	ON
27		Terputus	ON	ON
28	50	Terputus	ON	ON
29		Terputus	ON	ON
30		Terputus	ON	ON

kemudian sistem akan menghubungkan ke Google Script untuk melakukan pengambilan gambar dan mengirimkannya ke Google Drive.

Pengujian dan analisa program kecepatan kendaraan

Pada pengujian ini dilakukan pengujian pada program kecepatan, di mana di mulai dari mikrokontroler melakukan pengolahan data yang didapatkan oleh sensor anemometer. Pada Gambar 13 dapat dilihat hasil pengujian pada program kecepatan di mana didapatkan nilai RPS (Rotations Per Second), kemudian nilai RPS tersebut diolah sehingga mendapatkan kecepatan dalam satuan km/Jam.



Gambar 13 Hasil pengujian program kecepatan kendaraan

Pengujian dan analisa program titik lokasi

Pada pengujian ini dilakukan pengujian pada program titik lokasi, di mana untuk mengetahui titik lokasi tersebut menggunakan modul GPS. Pada program ini, titik lokasi yang dihasilkan nantinya akan dikirimkan ke Firebase yang langsung terhubung dengan aplikasi dan Google Spreadsheets yang terhubung dengan bagian riwayat pada aplikasi berbasis android. Hasil pengujian ini dapat dilihat pada Gambar 14 berikut:



Gambar 14 Hasil pengujian program titik lokasi

Pengujian dan Analisa Sistem Keseluruhan

Tahap pengujian dan analisa sistem secara keseluruhan dilakukan untuk mengetahui apakah sistem keamanan helm berbasis mikrokontroler ini dapat berjalan sesuai dengan rancangannya. Di mana pengujian ini serta analisa sistem keseluruhan ini dibagi menjadi 4 bagian, yaitu:

Pengujian dan analisa status koneksi helm

Pengujian ini bertujuan untuk mendeteksi koneksi antar protokol MQTT dan memberikan peringatan melalui speaker ketika koneksi protokol MQTT pada helm terputus dan memberikan titik lokasi ke pemilik melalui aplikasi berbasis android. Hasil pengujian terdapat pada tabel berikut:

Tabel 13 Pengujian status koneksi helm

No.	Jarak (M)	Kondisi	Tampilan pada aplikasi	DF Player Mini	Ind
1	1	Terhubung		OFF	Berhasil
2	12	Terhubung		OFF	Berhasil
3	23	Terhubung		OFF	Berhasil
4	27	Terhubung		OFF	Berhasil
5	30	Terhubung		OFF	Berhasil
6	40	Tidak Terhubung		ON	Berhasil

Pada Tabel 13 dapat dilihat hasil pengujian sistem untuk koneksi protokol MQTT pada helm, di mana pada pengujian ini sampel diambil secara acak. Berdasarkan pengujian yang dilakukan maka dapat diperoleh tingkat keberhasilan dengan menggunakan persamaan:

$$\text{Tingkat keberhasilan} = \frac{\text{Jumlah Total Aktivitas Berhasil}}{\text{Jumlah Banyak Aktivitas}} \times 100\% \quad (7)$$

$$\text{Tingkat keberhasilan} = \frac{6}{6} \times 100\%$$

$$\text{Tingkat keberhasilan} = 100\%$$

Berdasarkan perhitungan dapat diketahui bahwa tingkat keberhasilan untuk sistem koneksi pada helm menggunakan protokol MQTT yaitu 100%

Pengujian dan analisa sistem pengenalan wajah

Tujuan pengujian ini untuk mengenali wajah yang telah tersimpan di dalam database selain itu, juga melakukan pengambilan gambar ketika wajah yang terdeteksi mendekati helm tidak terdapat di dalam database dan mengirimnya ke

aplikasi berbasis android. Hasil pengujian terdapat pada tabel berikut:

Tabel 14 Pengujian sistem pengenalan wajah

No.	Jarak (cm)	Kondisi	Tampilan pada aplikasi	Indikator
1	15	Wajah tidak dikenal		Berhasil
2	30	Wajah tidak dikenal		Berhasil
3	40	Wajah tidak dikenal		Berhasil
4	15	Wajah Dikenal		Berhasil
5	30	Wajah Dikenal		Berhasil
6	40	Wajah Dikenal		Tidak Berhasil

Pada Tabel 14 dapat dilihat hasil pengujian sistem pengenalan wajah, terdapat 2 kondisi yaitu wajah yang dikenali oleh database dan wajah yang tidak dikenali oleh database untuk 6 kali percobaan. berdasarkan hasil yang didapatkan maka dapat diperoleh tingkat keberhasilan dengan menggunakan persamaan:

$$\text{Tingkat keberhasilan} = \frac{\text{Jumlah Total Aktivitas Berhasil}}{\text{Jumlah Banyak Aktivitas}} \times 100\%$$

$$\text{Tingkat keberhasilan} = \frac{5}{6} \times 100\%$$

$$\text{Tingkat keberhasilan} = 83.3\%$$

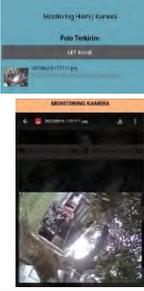
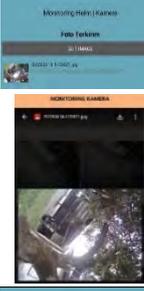
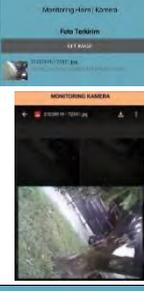
Berdasarkan perhitungan dapat diketahui bahwa tingkat keberhasilan untuk sistem pengenalan wajah ini yaitu 83.3%

Pengujian dan analisa pengambilan gambar secara berkala

Pengujian terakhir yaitu melakukan pengambilan gambar secara berkala sesuai dengan rentang waktu yang telah ditentukan dan

mengirimnya ke aplikasi berbasis android. Hasil pengujian terdapat pada tabel berikut:

Tabel 15 Pengujian pengambilan gambar secara berkala

Percobaan ke-	Waktu penerimaan foto oleh aplikasi	Tampilan pada aplikasi	Indikator
1	17:27:11		Berhasil
2	17:27:41		Berhasil
3	17:28:21		Berhasil
4	17:28:51		Berhasil
5	17:29:34		Berhasil
6	17:29:54		Berhasil

Percobaan ke-	Waktu penerimaan foto oleh aplikasi	Tampilan pada aplikasi	Indikator
			

Pengujian pengambilan gambar secara berkala dilakukan dengan menggunakan ESP32-Cam di mana ESP32-Cam telah diprogram sehingga dapat melakukan pengambilan gambar secara berkala sesuai dengan waktu yang telah dilakukan. Berdasarkan pengujian yang dilakukan maka dapat diperoleh tingkat keberhasilan dengan menggunakan persamaan:

$$\text{Tingkat keberhasilan} = \frac{\text{Jumlah Total Aktivitas Berhasil}}{\text{Jumlah Banyak Aktivitas}} \times 100\%$$

$$\text{Tingkat keberhasilan} = \frac{6}{6} \times 100\%$$

$$\text{Tingkat keberhasilan} = 100\%$$

Berdasarkan perhitungan dapat diketahui bahwa tingkat keberhasilan untuk sistem pengambilan gambar secara berkala yaitu 100%.

Pengujian dan analisa sistem kecepatan kendaraan

Pengujian kecepatan kendaraan memiliki tujuan untuk mendeteksi kecepatan pengendara dalam mengendarai motor dan memberikan notifikasi berupa suara melalui *speaker* ketika kecepatan yang dikendarai di atas batas normal dan memberikan notifikasi titik lokasi ke pengguna melalui aplikasi berbasis android. Hasil pengujian terdapat pada tabel berikut:

Tabel 16 Pengujian Kecepatan kendaraan.

No.	Kec. pada motor (km/h)	Kec. Pada aplikasi (km/h)	Tampilan pada aplikasi	DF Player Mini	Ind.
1	10	10.34		OFF	Berhasil
2	20	22.413		OFF	Berhasil
3	30	34.477		OFF	Berhasil

No.	Kec. pada motor (km/h)	Kec. Pada aplikasi (km/h)	Tampilan pada aplikasi	DF Player Mini	Ind.
4	40	39.641		OFF	Berhasil
5	50	53.430		ON	Berhasil
6	60	53.430		ON	Berhasil

Pengujian kecepatan kendaraan dihitung menggunakan sensor anemometer. Pada Tabel 16 dapat dilihat hasil kecepatan kendaraan yang dihitung oleh sensor, di mana berdasarkan pengujian tersebut DFPlayer mini akan memberikan peringatan ketika sensor anemometer menghasilkan kecepatan besar dari 50km/Jam. Berdasarkan pengujian yang dilakukan maka dapat diperoleh tingkat keberhasilan dengan menggunakan persamaan:

$$\text{Tingkat keberhasilan} = \frac{\text{Jumlah Total Aktivitas Berhasil}}{\text{Jumlah Banyak Aktivitas}} \times 100\%$$

$$\text{Tingkat keberhasilan} = \frac{6}{6} \times 100\%$$

$$\text{Tingkat keberhasilan} = 100\%$$

Berdasarkan perhitungan dapat diketahui bahwa tingkat keberhasilan untuk sistem kecepatan kendaraan yaitu 100%.

Pengujian Kuesioner Usability

Kuesioner penelitian ini diberikan ke 30 orang sampel, di mana responden mengisi kuesioner setelah mendengarkan penjelasan terkait alat yang dibuat dan mencobakan secara langsung atau melihat melalui video demonstrasinya. Jawaban dari kuesioner ini berupa skala *likert* yang memiliki bobot 1-5.

Tabel 17 Hasil pengujian kuesioner *usability*

No.	Pertanyaan	Hasil perhitungan kuesioner															
		SS			S			CS			KS			TS			
		J	B	N	J	B	N	J	B	N	J	B	N	J	B	N	
1.	L	L.1	7		35	15		60	7		21	1		2	0		0
		L.2	10		50	15		60	5		15	0		0	0		0
		L.3	18		90	10		40	2		6	0		0	0		0
2.	E	E.1	2		10	13		52	13		39	2		4	0		0
		M.1	8		40	16		64	5		15	1		2	0		0
3.	M	M.2	7		35	17		68	6		18	0		0	0		0
		Er	4		20	3		12	8		24	14		28	1		1
5.	S	S.1	2	5	10	3	4	12	12	3	36	12	2	24	1	1	1
		S.2	17		85	10		40	3		9	0		0	0		0
		S.3	3		15	3		12	6		18	18		36	0		0
		S.4	3		15	11		44	13		39	3		6	0		0
		S.5	16		80	12		48	2		6	0		0	0		0
		S.6	13		65	15		60	2		6	0		0	0		0
		S.7	3		15	7		28	13		39	7		14	0		0
			113		565	150		600	97		291	58		116	2		2

Untuk menentukan indeks keberhasilan perlu dilakukan perhitungan skor maksimum menggunakan rumus sebagai berikut:

$$\text{Skor Maksimum} = \text{Nilai tertinggi skala likert} \times \text{Jumlah Responden} \times \text{total pertanyaan kuesioner} \quad (8)$$

$$\text{Skor Maksimum} = 5 \times 30 \times 14$$

$$\text{Skor Maksimum} = 5 \times 30 \times 14$$

$$\text{Skor Maksimum} = 2100$$

Berdasarkan nilai skor maksimum tersebut, maka didapatkan indeks keberhasilan dengan menggunakan rumus berikut:

$$\text{Indeks (\%)} = \frac{\text{Total Skor}}{\text{Skor Maksimum}} \times 100 \quad (10)$$

$$\text{Indeks (\%)} = \frac{\text{Bobot}_{SS} + \text{Bobot}_{S} + \text{Bobot}_{CS} + \text{Bobot}_{KS} + \text{Bobot}_{RS}}{\text{Skor Maksimum}} \times 100$$

$$\text{Indeks (\%)} = \frac{565+600+291+116+2}{2100} \times 100$$

$$\text{Indeks (\%)} = \frac{1574}{2100} \times 100$$

$$\text{Indeks (\%)} = 74.95$$

Berdasarkan perhitungan tersebut dapat disimpulkan bahwa persentase keberhasilan Sistem Keamanan Helm Berbasis Mikrokontroler menggunakan kuesioner *usability* dengan total *sample* 30 orang adalah 74.95% dengan keterangan helm ini mudah digunakan.

CONCLUSIONS

Berdasarkan pengujian yang telah dilakukan maka dapat disimpulkan:

1. Sistem keamanan pada helm berbasis mikrokontroler berhasil di rancang di mana sistem ini dapat mendeteksi kehilangan helm melalui protokol MQTT dengan jarak antara helm dan motor 35 meter saat berada di lokasi yang memiliki sedikit hambatan dan 15 meter saat berada di lokasi yang memiliki banyak hambatan. Selain itu, helm ini juga dapat melindungi pengendara dengan mendeteksi kecepatan kendaraan sehingga pengendara lebih berhati-hati ketika kecepatan besar dari 50km/jam dengan persentase *error* sensor anemometer yaitu 7.23%.
2. Sistem berhasil mengantisipasi kehilangan helm memberikan notifikasi peringatan suara dan titik lokasi ke ponsel ketika koneksi antara *publish* dan *subscriber* pada protokol MQTT terputus dengan tingkat keberhasilan 100%.
3. Sistem dapat mendeteksi wajah manusia yang mendekati helm dan mengambil gambar ketika wajah tersebut tidak dikenali oleh *database* dengan tingkat keberhasilan 83.3% dengan jarak antara wajah dan kamera yaitu 15cm hingga 60cm.
4. Sistem dapat menangkap kondisi sekitar helm secara berkala sesuai dengan rentang waktu yang telah ditentukan dengan tingkat keberhasilan 100% dan rata-rata rentang waktu pengambilan gambar adalah 38 detik dan rata-rata proses pengiriman gambar ke aplikasi selama 5.6 detik.
5. Sistem mampu mendeteksi kecepatan kendaraan menggunakan sensor anemometer dengan persentase eror 7.23% dan memberikan notifikasi peringatan suara serta titik lokasi ke ponsel ketika kecepatan di atas batas normal dengan tingkat keberhasilan 100%.

Berdasarkan implementasi dan pengujian yang telah dilakukan untuk membangun sistem keamanan helm berbasis mikrokontroler, terdapat beberapa saran pengembangan agar dapat meningkatkan kinerja sistem, yaitu:

1. Pengembangan aplikasi menggunakan Android Studio, sehingga nantinya dapat ditambahkan fitur *push notification* agar saat terjadinya perpindahan helm atau kecepatan kendaraan dikendarai di atas batas maksimal langsung mendapatkan pemberitahuan ke ponsel.
2. Menggunakan komponen dengan ukuran yang lebih kecil, sehingga bobot pada helm lebih ringan.
3. Melakukan perancangan infrastruktur MQTT agar broker protokol MQTT yang digunakan tidak sering mengalami masalah dan sistem dapat berjalan dengan stabil.

REFERENCES

- [1] Prathama, Aditya. 2022. *Viral, Curhat Kehilangan Helm, Pemuda Ini Dapat Gantinya Langsung dari Gibran Rakabuming*. Tersedia : <https://www.motorplus-online.com/read/253453009/viral-curhat-kehilangan-helm-pemuda-ini-dapat-gantinya-langsung-dari-gibran-rakabuming> Diakses: 18 September 2022
- [2] Unand on Data. 2022. *Survei keamanan kampus*. Tersedia : <https://bem.unand.ac.id/unandondata/pusatdata/survei-keamanan-kampus/> Diakses: 18 September 2022
- [3] Adisti, Naomi. 2022. *Berapa Angka Kecelakaan Lalu Lintas Di Indonesia Setiap Tahun?*. Tersedia : <https://goodstats.id/article/berapa-angka-kecelakaan-lalu-lintas-di-indonesia-tiap-tahun-YYx0u> Diakses: 27 September 2022
- [4] Hanafie, Ahmad, et al. 2022. "Perancangan alat keamanan helm berbasis alarm dalam mengatasi pencurian helm di parkir." *Ash-Shahabah: Jurnal Pengabdian Masyarakat* Vol. 1, No.1: 25-33.
- [5] Santoso, P.W, I.N. Piarsa, dan N.M.I.M Mandenni. 2021. "Sistem keamanan helm berbasis internet of things dengan fitur pelacakan menggunakan android." *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)* Vol. 5 No.5: 967-976.
- [6] Nurliana, S., dan A. Supani. 2018. "Rancang Bangun Alat Pemberi Isyarat Kecepatan Maksimum Melalui SMS Gateway Berbasis Mikrokontroler Pada Helm." *TEKNIKA* Vol. 12, No.2: 77-84.
- [7] Yanto, B., Basorudin, S. Anwar, A. Lubis, dan Karmi. 2020. "Smart home monitoring pintu rumah dengan identifikasi wajah menerapkan camera ESP32 berbasis IoT." *Jurnal Sistem Informasi dan Komputer* Vol 11, No. 1: 53-59.
- [8] Indonesia. *Undang-Undang Nomor 22 Tahun 2009 tentang lalu lintas dan angkutan jalan*. Lembaran Negara Republik Indonesia Tahun 2009 Nomor 96. Tambahan Lembaran Negara Republik Indonesia Nomor 5025. Sekretariat Negara. Jakarta.
- [9] Peraturan Menteri Perhubungan Republik Indonesia. *Nomor PM 111 tahun 2015 tentang Tata Cara Penetapan Batas Kecepatan*. Jakarta: Menteri Perhubungan Republik Indonesia.

- [10] Murwadi, R., dan R.R. Adisaputro. 2021. "Design sistem keamanan pintu menggunakan *face detection*." *Jurnal Teknologi Elektro*, Vol 12. No.3: 120-128.
- [11] Aryani, D., M.N. Ihsan, dan P. Septiyani. 2017. "Prototype sistem absensi dengan metode face recognition berbasis arduino pada SMK negeri 5 Kabupaten Tangerang." *Seminar Nasional Teknologi Informasi dan Multimedia, STMIK AMIKOM Yogyakarta*, Vol.1, No. 3: 37-42.
- [12] Muliadi, A. Imran, dan M. Rasul. 2020. "Pengembangan tempat sampah pintar menggunakan ESP32." *Jurnal Media Elektrik*. Vol. 17, No. 2: 73-79.
- [13] Dfrobot. Tanpa Tahun. "ESP32-CAM development board." *DFROBOT*. 1-5.
- [14] Abilovani, Z. B, W. Yahya, dan F. A. Bakhtiar. 2018. "Implementasi protokol MQTT untuk sistem monitoring perangkat IoT." *Jurnal Pembangunan Teknologi Informasi dan Ilmu Komputer*, Vol. 2, No. 12: 7521-7527.
- [15] Matimpas, J.O., Dr. Eng. A. Soetedjo, ST, MT., dan Ir. K.A. Widodo, MT. 2022. "Rancang bangun sistem pendeteksi kebisingan kenalpot pada kendaraan bermotor sesuai aturan yang berlaku untuk membantu kinerja kepolisian berbasis *internet of things*." *Seminar Hasil Elektro S1 ITN Malang*. 1-7.
- [16] Firdaus, dan Ismail. 2020. "Komparasi akurasi global position system receiver U-Blox Neo-6 M dan U-blox Neo-8MN pada navigasi quadcopter." *Elektron Jurnal Ilmiah*, Vol. 12, No.1: 12-15.
- [17] Isra, R.L., dan R. Mukhaiyar. 2022. "Monitoring kecepatan angin berbasis mikrokontroler dan IoT." *Jurnal Teknik Elektro Indonesia*. Vol. 3, No. 2: 437-447.
- [18] Linarta, A., dan Nurhadi. 2018. "Aplikasi bel sekolah otomatis berbasis arduino dilengkapi dengan output suara." *Jurnal Indormatika, manajemen, dan komputer*. Vol.10, No.2: 1-7.
- [19] Rama, dan W. Sundari. 2022. "Alat bantu pendeteksi uang kertas untuk penyandang tunanetra." *Proyek akhir*. 1-34.
- [20] Afdali, M., M. Daud, dan R. Putri. 2017. "Perancangan alat ukur digital untuk tinggi dan berat badan dengan output suara berbasis Arduino Uno." *Jurnal ELKOMIKA*. Vol. 5, No. 1: 106-118.
- [21] Saputra, J., dan F. Eliza. 2022. "Perancangan pintu masuk gedung otomatis berdasarkan suhu tubuh manusia dengan informasi display dan suara." *Jurnal Teknik Elektro Indonesia*. Vol.3, No. 2: 448-457.
- [22] Wahyuni, R.D., S. Utomo, dan I.S. Sutjiningtyas. 2022. "Rancang bangun prototype alat penghitung jumlah orang dalam antrian berbasis ESP8266." *Prosiding SNASIKOM*. Vol. 2, No. 1: 91-99.
- [23] Samsugi, S., Z. Mardiyansyah, dan A. Nurkholis. 2020. "Sistem pengontrol irigasi otomatis menggunakan mikrokontroler Arduino Uno." *JTST*. Vol. 01, No. 01: 17-22.
- [24] Herlianus, dan G. Gunadi. 2022. "Pengembangan media pembelajaran organ gerak hewan dan manusia berbasis android menggunakan kodular." *Jurnal Informatik*. Edisi ke-18, Nomor 1: 88-96.
- [25] Cahyo, K.N., Martini, dan E. Riana. 2019. "Perancangan sistem informasi pengelolaan kuesioner pelatihan pada PT Brainmatics Cipta Informatika." *Jurnal of Information System Research (JOSH)*, Vol. 1, No. 1: 45-53.
- [26] Pranatawijaya, V.H., Widiatry, R. Priskila, dan P.B.A.A. Putra. 2019. "Pengembangan aplikasi kuesioner survey berbasis web menggunakan skala linkert dan guttman." *Jurnal Sains dan Informatika*, Vol. 5, No.2 : 128-137.
- [27] Shonta, A.A, dan kawan-kawan. 2022. "Penerapan *firebase real time database* pada aplikasi media informasi dan pendaftaran training IT berbasis android." *Jurnal Media Informatika Budidarma*, Vol. 6, Bo. 3: 1517-1525.
- [28] Mabruuri, M. I. 2022. "Simulasi rasio dan proposisi menggunakan Google Spreadsheet." *Jurnal Didaktika Pendidikan Dasar*, Vol. 6, No. 1: 347-368.