

**IMPLEMENTASI SISTEM KEAMANAN CERDAS UNTUK
PENCEGAHAN PENCURIAN MOTOR DENGAN INTEGRASI
ESP32-CAM DAN SENSOR GETAR**

SKRIPSI

**Diajukan oleh:
Bobli
2011102441069**



**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS
SAINS DAN TEKNOLOGI UNIVERSITAS MUHAMMADIYAH
KALIMANTAN TIMUR
JULI 2024**

**IMPLEMENTASI SISTEM KEAMANAN CERDAS UNTUK
PENCEGAHAN PENCURIAN MOTOR DENGAN INTEGRASI
ESP32-CAM DAN SENSOR GETAR**

SKRIPSI

Diajukan Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer Fakultas Sains Dan
Teknologi Universitas Muhammadiyah Kalimantan Timur

Diajukan oleh:
Bobli
2011102441069



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
JULI 2024**

LEMBAR PERSETUJUAN
IMPLEMENTASI SISTEM KEAMANAN CERDAS UNTUK
PENCEGAHAN PENCURIAN MOTOR DENGAN INTEGRASI ESP32-
CAM DAN SENSOR GETAR

SKRIPSI

Diajukan oleh:

BOBLI
2011102441069

Disetujui untuk diujikan
pada tanggal 30 juni 2024

Pembimbing



Arbansyah, M.TI
NIDN. 1118019203

Mengetahui,
Koordinator Tugas Akhir



Abdul Rahim, S.Kom., M.Cs
NIDN. 0009047901

LEMBAR PENGESAHAN

**IMPLEMENTASI SISTEM KEAMANAN CERDAS UNTUK
PENCEGAHAN PENCURIAN MOTOR DENGAN INTEGRASI
ESP32-CAM DAN SENSOR GETAR**

SKRIPSI


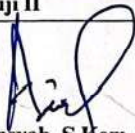
Diajukan oleh:

Bobli

2011102441069

Diseminarkan dan Diujikan

Pada tanggal Juli 2024

Penguji I	Penguji II
 Taghfirul/Azhima Yoga Siswa, S.Kom, M, Kom NIDN. 1118038805	 Arbansyah, S.Kom., M.TI NIDN. 1118019203

**Mengetahui,
Ketua**

Program Studi Teknik Informatika



PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama : Bobli

NIM : 2011102441069

Program Studi : S1 Teknik Informatika

Judul Penelitian : Implementasi Sistem Keamanan Cerdas Untuk Pencegahan Pencurian Motor Dengan Integrasi ESP32-CAM Dan Sensor Getar

menyatakan bahwa **skripsi** yang saya tulis ini benar-benar hasil karya saya sendiri, dan bukan merupakan hasil plagiasi/falsifikasi/fabrikasi baik sebagian atau seluruhnya.

Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam **skripsi** saya ini, atau klaim dari pihak lain terhadap keaslian karya saya ini

Samarinda, 5 Juli 2024

Yang membuat pernyataan

Bobli
2011102441069



ABSTRAK

Meningkatnya kebutuhan akan alat transportasi di Indonesia telah berdampak pada tingginya angka pencurian kendaraan bermotor. Kelemahan dalam sistem keamanan kendaraan sering menjadi celah bagi tindakan kriminal. Penelitian ini bertujuan untuk mengembangkan sistem keamanan sepeda motor yang lebih responsif dan proaktif dengan memanfaatkan teknologi *ESP32-CAM* dan sensor getar *SW-420*. Sistem ini dirancang untuk mendeteksi getaran yang mencurigakan sebagai indikasi aktivitas tidak lazim, sistem akan mengirimkan notifikasi dan gambar melalui aplikasi Telegram kepada pemilik kendaraan, memungkinkan mereka untuk segera mengambil tindakan yang diperlukan. Hasil penelitian menunjukkan bahwa penggunaan *ESP32-CAM* dan sensor getar dalam sistem keamanan sepeda motor dapat meningkatkan efektivitas pencegahan dan penanggulangan tindak pencurian kendaraan.

Kata kunci: Sistem keamanan sepeda motor, *ESP32-CAM*, *SW-420*, Telegram.

ABSTRACT

The increasing demand for transportation in Indonesia has led to a rise in motorcycle theft. Weaknesses in vehicle security systems often provide opportunities for criminal activity. This research aims to develop a more responsive and proactive motorcycle security system utilizing ESP32-CAM technology and the SW-420 vibration sensor. The system is designed to detect suspicious vibrations as an indication of unusual activity, sending notifications and images through the Telegram application to the vehicle owner, allowing them to take immediate action. The research results show that the use of ESP32-CAM and vibration sensors in motorcycle security systems can improve the effectiveness of preventing and addressing vehicle theft.

Keywords: Motorcycle security system, ESP32-CAM, SW-420, Telegram.

PRAKATA

Puji syukur penulis penatkan kehadiran Allah SWT karena atas berkat Rahmat dan hidayah-Nya sehingga penyusunan skripsi dengan judul “Implementasi Sistem Keamanan Cerdas Untuk Pencegahan Pencurian Motor Dengan Integrasi *ESP32-Cam* dan Sensor Getar”. Penyusun proposal skripsi ini untuk memenuhi akademik untuk menyelesaikan Pendidikan pada program studi Teknik Informatika fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur.

Penulis menyadari dalam penyusunan proposal skripsi ini banyak mendapatkan bantuan, bimbingan, serta diberikan kemudahan dari berbagai pihak sehingga proposal ini dapat diselesaikan. Secara khusus penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Bapak Arbansyah, S.Kom, M.TI, selaku pembimbing yang telah membimbing menyediakan waktu, tenaga dan memberikan nasehat serta motivasi untuk menyelesaikan proposal skripsi ini.
2. Kedua Orang Tua beserta keluarga yang senantiasa memberikan dukungan, dan memberikan perhatian maupun semangat yang besar dalam mendukung penulis.
3. Kepada teman-teman penulis yang tidak mungkin penulis sebutkan satu persatu, yang telah memberikan waktu luang dan saran dalam penyusunan skripsi penulis.

Penulis menyadari proposal skripsi ini tidak luput dari berbagai kekurangan. Penulis mengharapkan saran dan kritik demi kesempurnaan dan perbaikan sehingga akhirnya laporan proposal ini dapat memberikan manfaat sekaligus menambah ilmu bagi penulis dan dapat memberikan wawasan bagi pembacanya. Amin Ya Robbal Alamin.

Samarinda, 20 Mei 2024

Penyusun,



Bobli

2011102441069

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
PERNYATAAN KEASLIAN PENELITIAN	v
ABSTRAK.....	vi
ABSTRACT	vii
PRAKATA	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Manfaat penelitian	2
1.5 Batasan Masalah	3
BAB II METODE PENELITIAN	4
2.1 Lokasi Penelitian	4
2.2 Objek Penelitian	4
2.3 Alat dan Bahan	4
2.4 Penggunaan Software	7
2.5 Prosedur Penelitian	8
2.5.1 Studi Literatur.....	8
2.5.2 Analisis Kebutuhan.....	8
2.5.3 Perancangan Perangkat Keras	8
2.5.4 Desain Sistem	10
2.5.5 Perancangan Perangkat Lunak.....	11
2.5.6 Pengujian sistem	21
2.5.7 Implementasi	21
2.5.8 Evaluasi dan peningkatan	21

2.5.9 Analisa Kerja	21
BAB III HASIL ANALISIS DAN PEMBAHASAN.....	22
3.1 Hasil.....	22
3.2 Implementasi Alat.....	29
3.3 Pembahasan	33
BAB IV PENUTUP.....	35
4.1 Simpulan.....	35
4.2 Implikasi	35
4.3 Saran	36
DAFTAR RUJUKAN.....	37
RIWAYAT HIDUP	39
LAMPIRAN	40

DAFTAR TABEL

Tabel	Halaman
Tabel 2.1 Pin-pin I/O ESP32-CAM.....	9
Tabel 3.1 Hasil Pengujian Koneksi Wifi dan Telegram.....	23
Tabel 3.2 Hasil Pengujian Sensor Getar.....	25
Tabel 3.3 Kualitas Kamera ESP32-CAM.....	26
Tabel 3.4 Hasil Pengujian Pengambilan dan Pengiriman Foto	28
Tabel 3.5 Keandalan Deteksi Getaran	30
Tabel 3.6 Respon Sistem	32

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2. 1 ESP32-CAM.....	4
Gambar 2.2 SW-420.....	5
Gambar 2.3 Kabel jumper	5
Gambar 2.4 Powerbank	6
Gambar 2.5 USB Micro.....	7
Gambar 2.6 Telegram.....	7
Gambar 2.7 Arduino IDE	8
Gambar 2.8 Tahapan Penelitian	8
Gambar 2.9 Perancangan perangkat keras.....	9
Gambar 2.10 Desain Sistem Keamanan Sepeda Motor.....	10
Gambar 2.11 Flowchart Sistem	11
Gambar 2.12 Program 1	13
Gambar 2.13 Program 2	14
Gambar 2.14 Program 3	14
Gambar 2.15 Program 4	15
Gambar 2.16 Program 5	15
Gambar 2.17 Program 6	16
Gambar 2.18 Program 7	16
Gambar 2.19 Program 8	17
Gambar 2.20 Program 9	17
Gambar 2.21 Program 10	18
Gambar 2.22 Program 11	18
Gambar 2.23 Program 12	19
Gambar 2.24 Program 13	20
Gambar 2.25 Program 14	20
Gambar 3.1 Pengujian Alat	22
Gambar 3.2 Output Pada Serial Monitor 1	22
Gambar 3.3 Output Pada Serial Monitor 2	23
Gambar 3.4 Output Pada Serial Monitor 3	24
Gambar 3.5 Output Pada Serial Monitor 4.....	24
Gambar 3.6 Output Pada Serial Monitor 5.....	24

Gambar 3.7 Output Pada Serial Monitor 6.....	27
Gambar 3.8 Output Pada Serial Monitor 7.....	27
Gambar 3.9 Notifikasi Telegram.....	29
Gambar 3.10 Alat pada Motor.....	30
Gambar 3.11 Pengambilan Gambar	31
Gambar 3.12 Notifikasi Telegram.....	32

DAFTAR LAMPIRAN

Lampiran

L.1 Surat Izin Penelitian	46
L.2 Kartu Kendali Bimbingan Skripsi.....	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peningkatan jumlah kendaraan bermotor dan kebutuhan hidup yang mendesak telah menyebabkan kasus kriminalitas meningkat di berbagai kota di Indonesia (Andesta & Ferdian, 2018). Salah satunya bentuk kriminalitas yang dominan adalah pencurian kendaraan bermotor, yang sering terjadi karena kurangnya sistem keamanan yang efektif pada sebagian besar kendaraan (Jonas et al., 2022). Sepeda motor menjadi kendaraan paling populer di Indonesia karena harganya yang terjangkau oleh berbagai lapisan masyarakat. Menurut Badan Pusat Statistik, kasus pencurian motor mendominasi statistik kriminalitas pada tahun (Boy et al., 2021).

Penelitian sebelumnya telah mengusulkan berbagai solusi untuk mengatasi masalah ini (Sujadi et al., 2018) mengembangkan sistem pemantauan keamanan sepeda motor berbasis *Internet of Things* dengan menggunakan sensor *PIR* dan *Ultrasonik* untuk pemantauan real-time yang dapat diakses melalui smartphone. Namun, solusi ini masih menghadapi beberapa keterbatasan, seperti sensitivitas terhadap gangguan cuaca atau lingkungan yang dapat mempengaruhi akurasi deteksi.

Di sisi lain, (Zidane & Rahmadewi, n.d.) mengimplementasikan teknologi *ESP32-CAM* dalam proyek “Implementasi *ESP32-CAM* Pada Alat Sistem Kendaraan Sepeda Motor (MOCLESS)”. Penelitian ini berfokus pada deteksi wajah dan sensor getar untuk mengidentifikasi gangguan atau upaya pencurian. Meskipun memberikan solusi yang lebih spesifik, penggunaan sensor getar dalam konteks ini belum sepenuhnya dieksplorasi untuk mendeteksi getaran yang signifikan sebagai indikasi pencurian.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem keamanan yang lebih responsif dan proaktif menggunakan sensor getar dan mikrokontroler *ESP32-CAM*. Sensor getar dipilih karena kemampuannya merespon langsung terhadap getaran yang mencurigakan pada sepeda motor, memungkinkan sistem untuk memberikan alarm secara cepat dan akurat kepada pemilik kendaraan.

Dalam penelitian ini menggunakan sensor getar *SW-420* untuk mengatasi kelemahan dari penelitian sebelumnya. Keunggulan dari sistem penelitian ini yaitu mengembangkan meliputi fokus deteksi getaran fisik yang signifikan, ketahanan terhadap perubahan suhu atau kondisi cuaca seperti angin dan hujan, dan pengurangan jumlah alarm palsu yang disebabkan oleh gangguan lingkungan. Sistem peringatan modern yang peneliti gunakan mengirim gambar langsung ke pemilik melalui *Telegram*, memungkinkan pemilik menerima notifikasi dan melihat situasi dari jarak jauh, serta meningkatkan responsivitas pemilik bahkan ketika mereka tidak berada di dekat sepeda motor. Selain itu, sistem dapat mengatur ambang batas pada sensor getar untuk mendeteksi hanya getaran yang signifikan, sehingga mengurangi alarm palsu dan meningkatkan keandalan deteksi.

Secara keseluruhan, penelitian ini menawarkan beberapa perbaikan dari penelitian sebelumnya. Sensor getar *SW-420* yang digunakan lebih tahan terhadap gangguan cuaca dan lingkungan, sementara sistem peringatan melalui Telegram memungkinkan pemilik merespon dari jarak jauh. Dengan pengaturan ambang batas yang tepat, sistem penelitian ini mengurangi jumlah alarm palsu dan memberikan notifikasi yang lebih cepat dan akurat, menjadikannya lebih efektif dan andal dalam mendeteksi gangguan.

Tujuan utama penelitian ini meliputi menerapkan teknologi sensor getar dan *ESP3-CAM* dalam pengembangan sistem keamanan cerdas untuk mencegah pencurian motor, mengembangkan sistem notifikasi yang memberitahu pemilik motor secara cepat saat terjadi kejadian mencurigakan, serta menguji keandalan dan keefektifan sistem ini melalui serangkaian uji coba dilingkungan yang terkontrol dan representatif.

Dengan demikian, penelitian ini tidak hanya menghadirkan inovasi baru dalam mengatasi tantangan keamanan kendaraan bermotor, tetapi juga membawa solusi yang lebih efektif dan adaptif dalam menghadapi ancaman pencurian sepeda motor.

1.2 Rumusan Masalah

1. Bagaimana mengintegrasikan sensor getar dengan mikrokontroler *ESP32-CAM* untuk mendeteksi guncangan atau gerakan mencurigakan pada motor?

1.3 Tujuan Penelitian

1. Mengembangkan sistem keamanan sepeda motor yang menggunakan sensor getar *SW-420* dan mikrokontroler *ESP32-CAM* untuk mendeteksi guncangan atau getaran mencurigakan pada motor, serta mengirimkan notifikasi secara real-time melalui *bot Telegram*.

1.4 Manfaat penelitian

1. Sistem yang di implementasikan dapat membantu pemilik kendaraan motor meningkatkan keamanan kendaraan bermotor dengan deteksi dini terhadap potensi pencurian.

2. Pemilik kendaraan motor dapat memantau keadaan kendaraan mereka dari jarak jauh melalui koneksi internet, memberikan rasa aman dan kemudahan dalam mengontrol kendaraan.

3. Sistem notifikasi yang terintegrasi memungkinkan pemilik kendaraan motor untuk menerima pemberitahuan secara real-time melalui berbagai saluran komunikasi, sehingga mereka dapat merespons kejadian mencurigakan dengan cepat.

4. Dengan adopsi teknologi keamanan cerdas seperti yang diusulkan dalam penelitian ini, pemilik kendaraan motor dapat meningkatkan kesadaran terhadap keamanan kendaraan mereka sendiri, sehingga mengurangi risiko pencurian di lingkungan sekitar.

1.5 Batasan Masalah

1. Penelitian ini hanya menggunakan modul *ESP32CAM* sebagai perangkat utama untuk pengambilan gambar dan konektivitas Wi-Fi.
2. Sistem ini difokuskan pada deteksi getaran yang signifikan yang diasumsikan sebagai indikasi upaya pencurian motor. Getaran kecil yang tidak relevan (misalnya, hembusan angin atau getaran kecil dari kendaraan lain) tidak menjadi fokus utama.
3. Pengujian sistem ini dilakukan dalam kondisi lingkungan yang terkontrol dan terbatas pada area tertentu yang representatif untuk uji coba fungsionalitas (misalnya, halaman rumah atau garasi.)

BAB II

METODE PENELITIAN

2.1 Lokasi Penelitian

Penelitian ini dilakukan di Kota Samarinda Kecamatan Samarinda Utara Kabupaten Kalimantan Timur. Penelitian ini akan di lakukan pada bulan Mei 2024.

2.2 Objek Penelitian

Objek penelitian ini adalah sistem keamanan cerdas untuk pencegahan pencurian motor menggunakan *ESP32-CAM* dan sensor getar. Sistem keamanan cerdas ini dirancang untuk diterapkan pada kendaraan roda dua, agar bisa memantau kondisi motor secara real-time, mendeteksi getaran yang mencurigakan, dan memberikan notifikasi kepada pemilik motor.

2.3 Alat dan Bahan

Alat dan bahan yang digunakan untuk penelitian dalam menjelaskan metode ini diantaranya :

1. *ESP32-CAM*

ESP32-CAM adalah mikrokontroler yang dilengkapi dengan fitur tabahan seperti *Bluetooth*, Wifi, kamera, dan slot *microSD*. *ESP32-CAM* biasanya digunakan dalam proyek internet of Things (IoT) yang membutuhkan fitur kamera. Dibandingkan dengan modul *ESP32* sebelumnya seperti *ESP32 Wroom*, *ESP32-CAM* memiliki lebih sedikit pin I/O (Andesta & Ferdian, 2018).

Modul *ESP32-CAM* memiliki dua sisi dalam rangkaiannya. Di bagian atas terdapat modul kamera yang bisa dilepas-pasang, slot *microSD* yang bisa diisi, serta lampu flash yang berfungsi sebagai lampu tambahan untuk kamera jika diperlukan (Andesta & Ferdian, 2018).



Gambar 2. 1 ESP32-CAM

2. Sensor Getar

Sensor getar digunakan untuk mendeteksi getaran pada kondisi tertentu di sepeda motor. Getaran yang dimaksud terjadi ketika seorang pencuri mencoba membobol sepeda motor di area tempat sensor dipasang. Sensor getar ini menggunakan modul *SW-420* tipe NC dengan tegangan kerja antara 3.3V hingga 5V. (Hamdani et al., 2019).



Gambar 2.2 SW-420

3. Kabel Jumper

Kabel jumper adalah kabel elektrik yang digunakan untuk menghubungkan komponen-komponen di *breadboard* tanpa perlu menyolder. Kabel jumper biasanya memiliki konektor atau di kedua ujungnya. Konektor yang digunakan untuk menusuk disebut *male connector*, sedangkan konektor yang digunakan untuk ditusuk disebut *female connector* (Williams, 1930).



Gambar 2.3 Kabel jumper

4. Powerbank

Power bank, juga dikenal sebagai portable charger atau baterai cadangan, adalah perangkat yang digunakan untuk menyimpan energi dan mengisi ulang gadget saat sedang berpergian. Konsep portable external baterai pertama kali dipamerkan pada tahun 2001 di *Las Vegas Internasional Consumer Electronics Show*, dan sejak itu semakin populer dengan munculnya smartphone pada tahun 2009 yang membutuhkan pengisian daya yang terus-menerus untuk tetap terhubung (Lak'apu et al., 2023).



Gambar 2.4 Powerbank

Power bank tersedia dalam berbagai kapasitas, mulai dari 3000 mAh hingga 11000 mAh, yang memungkinkan mereka untuk mengisi ulang gadget dengan kapasitas baterai yang lebih rendah. Sebagai contoh, power bank dengan kapasitas 3000 mAh dapat mengisi ulang smartphone dengan kapasitas baterai 1500 mAh hingga dua kali penuh (Lak'apu et al., 2023).

5. Kabel *USB Micro*

Micro USB memiliki desain yang lebar dan pipih, dan masih umum ditemukan pada smartphone dan perangkat pendukung seperti powerbank. Selain itu, *Micro USB* juga digunakan pada berbagai perangkat elektronik lainnya, misalnya untuk transfer data dan pengisian daya baterai kamera digital (Alfi et al., 2019).



Gambar 2.5 USB Micro

2.4 Penggunaan Software

1. Telegram

Telegram adalah salah satu aplikasi perpesanan yang terkenal karna keunggulanya dalam hal kecepatan, privasi, dan berbagai fitur tambahan yang bermanfaat. Salah satunya adalah *Telegram Bot*, yang memungkinkan pengguna untuk berinteraksi dengan bot melalui pesan teks dan perintah kusus (Ardiansyah et al., 2015)

Penggunaan *Telegram Bot* untuk berkomunikasi dengan perangkat *mikrokontroler* adalah contoh bagaimana teknologi modern seperti aplikasi perpesanan dapat diaplikasikan dalam konteks, mulai dari sistem keamanan kendaraan bermotor dan kontrol perangkat dari jarak jauh.



Gambar 2.6 Telegram

2. Arduino IDE

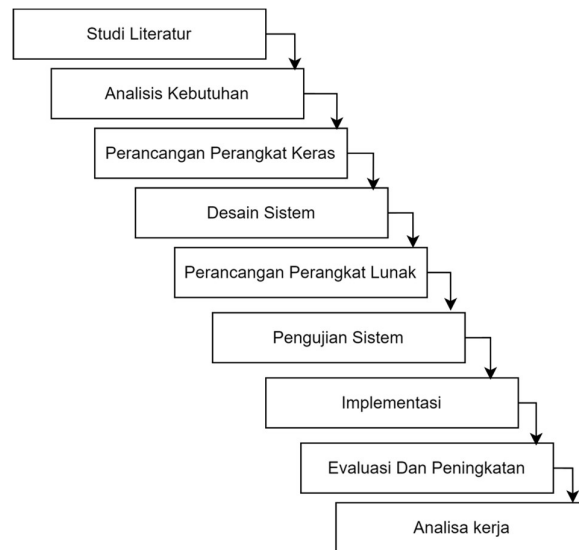
Arduino IDE adalah perangkat lunak yang digunakan untuk membuat dan mengedit kode pemrograman, yang disebut sketch, untuk *board Arduino* yang ingin diprogram. Dengan *Arduino IDE*, Anda dapat membuat, mengedit, mengupload kode ke board Arduino yang spesifik, dan mengembangkan program-program khusus. *Arduino IDE* dikembangkan menggunakan bahasa pemrograman *java* dan dilengkapi dengan berbagai pustaka (library) C/C++ yang disebut *WIRING*, yang mempermudah operasi input dan output pada board Arduino (Mahanin Tyas et al., 2023).



Gambar 2.7 Arduino IDE

2.5 Prosedur Penelitian

Adapun tahap penelitian yang di gunakan dalam penelitian ini adalah sebagai berikut:



Gambar 2.8 Tahapan Penelitian

2.5.1 Studi Literatur

Identifikasi literatur terkait sistem keamanan cerdas, teknologi *ESP32-CAM*, sensor getar, dan pencegahan pencurian motor. Tinjau riset-riset terkait untuk memahami konsep dan teknologi yang relevan.

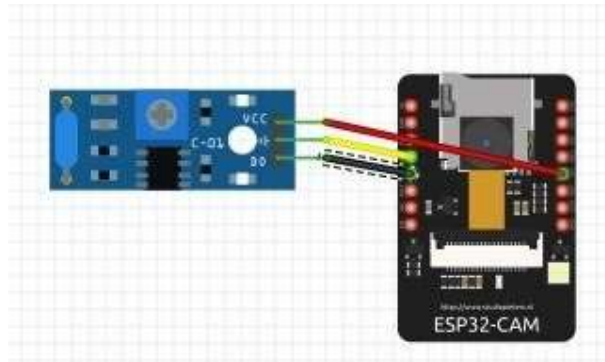
2.5.2 Analisis Kebutuhan

Kebutuhan yang akan digunakan dalam membuat sistem keamanan motor ini menggunakan *ESP32-CAM* dan sensor getar untuk melakukan proses analisis kebutuhan sistem yang bertujuan agar sistem keamanan yang dipakai nantinya sesuai dengan hasil yang akan di buat.

2.5.3 Perancangan Perangkat Keras

Perancangan perangkat keras merupakan tahapan krusal dalam pengembangan sistem keamanan cerdas untuk mencegah pencurian motor. Melalui perancangan yang matang, kita dapat menetapkan spesifikasi yang tepat, mengotimalkan kinerja sistem, mengidentifikasi dan mengatasi tantangan teknis, serta mengelola biaya dengan efektif. Dibawah ini adalah hasil dari rangkaian keseluruhan sistem yang dapat dijelaskan bahwa rangkaian terdiri dari *ESP32-CAM* dan *Sw-420* berfungsi sebagai sistem

keamanan pencurian motor, yaitu mulai dari mendeteksi getaran lalu mengambil gambar dan mengirimkan notifikasi seperti gambar. Berikut rangkaian keseluruhan dapat dilihat pada Gambar 2.6 Perancangan perangkat Keras.



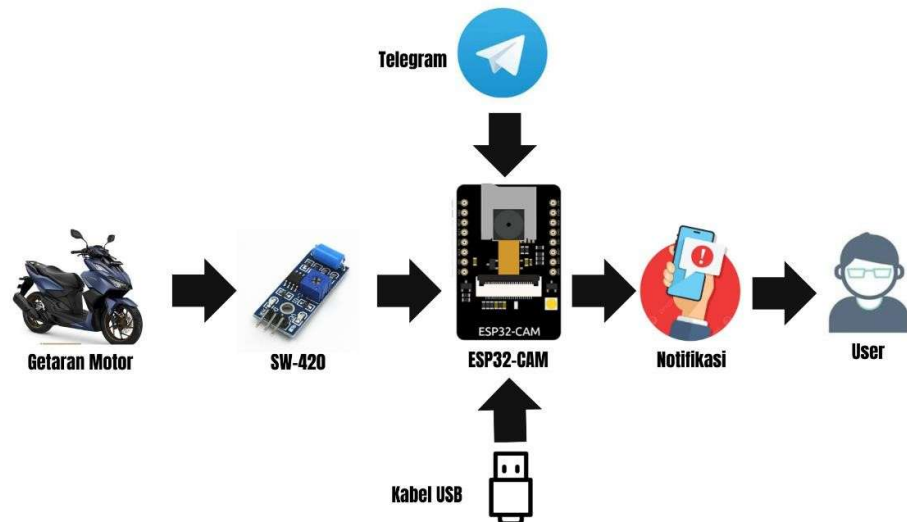
Gambar 2.9 Perancangan perangkat keras

Tabel 2.1 Pin-pin I/O ESP32-CAM

No	Nama Perangkat Keras	Pin pada ESP32-CAM
1	Sensor <i>SW-420</i>	GPIO 13

Tabel 2.1 hanya memiliki 1 sensor yang terhubung ke *ESP32-CAM* yaitu Sensor Getar (*SW-420*) pin yang digunakan adalah pin 13 untuk terhubung ke sensor getar, yang digunakan untuk mendeteksi getaran yang mungkin menandakan aktivitas atau perubahan kondisi pada motor.

2.5.4 Desain Sistem



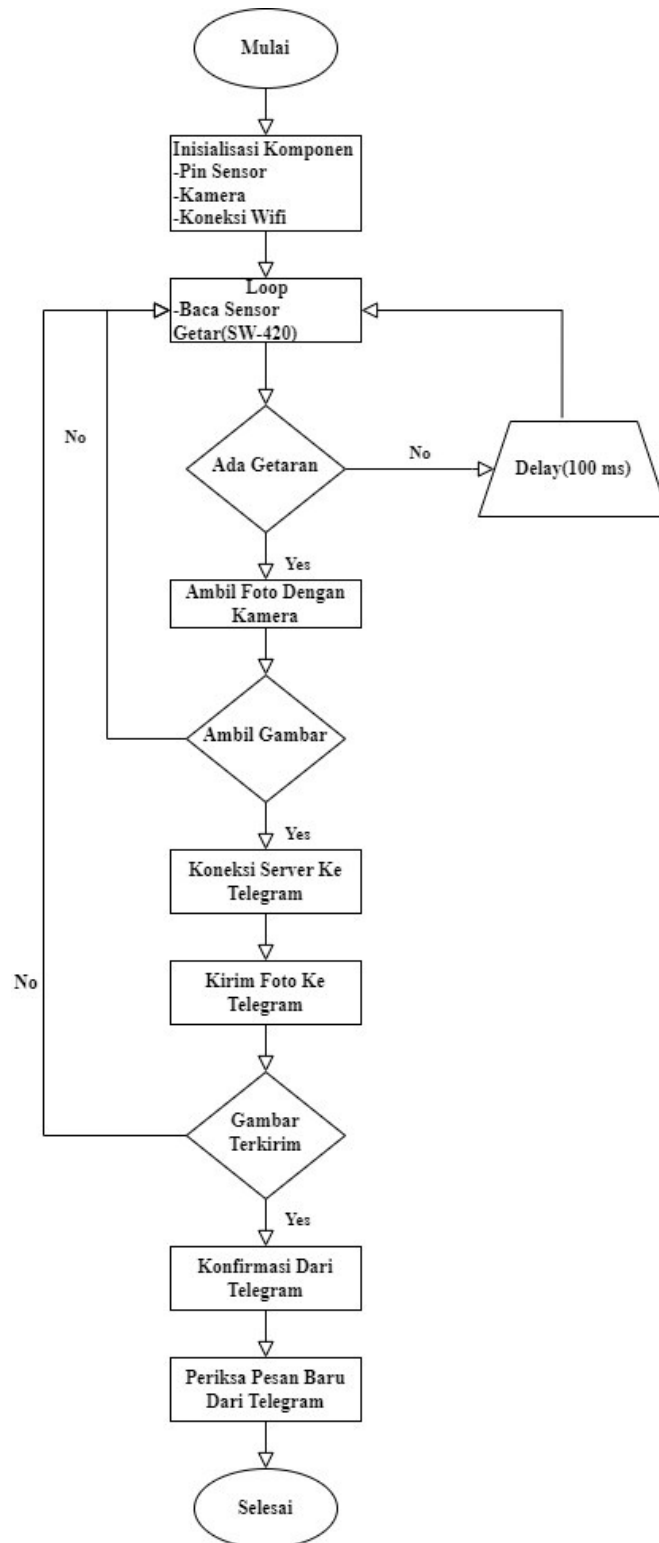
Gambar 2.10 Desain Sistem Keamanan Sepeda Motor

Alur Kerja Sistem pada Gambar 2.10 :

1. Sistem diaktifkan dan menghubungkan ke jaringan Wi-Fi.
2. Sensor mendeteksi getaran pada motor dan mengirimkan sinyal ke *ESP32-CAM*.
3. *ESP32-CAM* mengambil gambar saat menerima sinyal dari sensor getar.
4. Data Gambar dikirim ke server *Telegram* Melalui koneksi Wi-Fi.
5. Server Telegram menerima data gambar.
6. Server mengirim ke pengguna melalui *bot Telegram*.

2.5.5 Perancangan Perangkat Lunak

Flowchart Sistem Keamanan :



Gambar 2.11 Flowchart Sistem

Pada Gambar 2.11 menjelaskan bagaimana cara kerja sistem keamanan motor berkerja. Berikut penjelasan rinci cara kerja flowchart pada Gambar 2.11 :

1. **Mulai:** Program dimulai dari titik ini.
2. **Inisialisasi Komponen:**
 - a. Inisialisasi pin sensor getar, kamera, dan koneksi WIFI.
 - b. Konfigurasi kamera menggunakan *ESP32-CAM*.
3. **Loop:**
 - a. Program memasuki loop utama untuk terus melakukan pemantauan dan pengambilan tindakan berdasarkan kondisi sensor dan pesan Telegram.
4. **Baca Sensor Getaran:**
 - a. Program membaca sensor getaran (*SW-420*) untuk mendeteksi adanya getaran.
5. **Ada Getaran?:**
 - a. Jika sensor mendeteksi getaran maka akan memasuki langkah untuk mengambil gambar.
6. **Tunggu 100 ms:**
 - a. Setelah tidak ada getaran yang terdeteksi, program menunggu selama 100 ms sebelum membaca sensor lagi. Ini membantu menghindari pembacaan berlebihan.
7. **Ambil Foto dengan Kamera:**
 - a. Program melalui proses untuk mengambil gambar menggunakan kamera *ESP32-CAM*.
 - b. Jika gagal mengambil gambar (misalnya karena kegagalan koneksi atau masalah lain), proses kembali ke langkah “Baca Sensor Getar”.
8. **Gambar Diambil?:**
 - a. Program memeriksa apakah gambar berhasil diambil.
 - b. Jika tidak berhasil, kembali ke langkah “Baca Sensor Getar”.
 - c. Jika berhasil, lanjut ke langkah selanjutnya untuk mengirim gambar ke Telegram.
9. **Koneksi Ke Sever Telegram:**
 - a. Program membuat koneksi ke server Telegram menggunakan protokol HTTPS.
 - b. Mengirim gambar yang telah diambil sebagai pesan dengan format multipart/form-data.
10. **Kirim Foto Ke Telegram:**
 - a. Proses pengiriman gambar ke Telegram dimulai.
 - b. Program menunggu konfirmasi atau respons dari sever Telegram setelah mengirimkan gambar.
11. **Gambar Terkirim?:**
 - a. Program memeriksa apakah gambar berhasil terkirim ke Telegram.
 - b. Jika tidak berhasil, kembali ke langkah “Baca Sensor Getaran” untuk memulai ulang proses.
 - c. Jika berhasil, lanjut ke langkah untuk memeriksa pesan baru dari Telegram.
12. **Periksa Pesan Baru Dari Telegram:**

- a. Program memeriksa apakah ada pesan baru atau respons dari Telegram setelah pengiriman gambar.
- b. Tanggapan dari Telegram bisa berupa konfirmasi pengiriman, pesan dari pengguna, atau instruksi lain.

13. Selesai:

- a. Program berakhir setelah selesai menjalankan tugasnya.

Flowchart di atas menggambarkan Langkah-langkah utama dalam sistem keamanan cerdas untuk pencegahan pencurian motor, ini mencakup inisialisasi perangkat keras dan layanan, deteksi getaran, pengambilan gambar, pengiriman notifikasi ke Telegram. Berikut adalah program yang telah disediakan untuk sistem keamanan yang menggunakan *ESP32-CAM* dan sensor getar, dengan pengiriman notifikasi melalui Telegram:

1. *Inklusi Header dan Definisi Variabel*

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
```

Serial Monitor ×

Gambar 2.12 Program 1

Gambar 2.12 adalah pustaka yang diperlukan untuk koneksi Wifi, kamera, dan komunikasi dengan *bot Telegram*.

2. Konfigurasi Koneksi Wifi dan *Bot Telegram*

```
// WiFi credentials
const char* ssid = "Bobli";
const char* password = "apamserawa";
// Telegram Bot Token and Chat ID
String BOTtoken = "7442583602:AAHuLiRU-hAuJbHII1KN8jmhF-uyYvUzxR0";
String CHAT_ID = "6842233029";
// Telegram and WiFi client initialization
WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);
```

Serial Monitor ×

Gambar 2.13 Program 2

Gambar 2.13 ini berisi kredensial Wifi dan token *bot Telegram* serta menginisialisasi klien Wifi dan *bot Telegram*.

3. Konfigurasi Pin Kamera dan Sensor Getar

```
// Camera pins definition
#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM      5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

// Vibration sensor pin definition
#define VIBRATION_SENSOR_PIN 13
```

Serial Monitor ×

Gambar 2.14 Program 3

Gambar 2.14 adalah deifinisi pin yang digunakan oleh modul kamera dan pin yang digunakan untuk sensor getar *SW-420*.

4. Thershold Getaran

```
const int vibrationThresholdLow = 1;
const int vibrationThresholdMedium = 2;
const int vibrationThresholdHigh = 3;
```

Serial Monitor ×

Gambar 2.15 Program 4

Gambar 2.15 Mendefinisikan nilai ambang untuk mendeteksi level getaran rendah, sedang, dan tinggi.

5. Fungsi 'handleNewMessages()'

```
void handleNewMessages(int numNewMessages) {
  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    String text = bot.messages[i].text;
    String from_name = bot.messages[i].from_name;

    if (text == "/start") {
      String welcome = "Welcome , " + from_name + "\n";
      bot.sendMessage(chat_id, welcome, "");
    } else if (text == "/photo") {
      sendPhoto = true;
      bot.sendMessage(chat_id, "New photo request received.");
    }
  }
}
```

Serial Monitor ×

Gambar 2.16 Program 5

Fungsi pada gambar 2.16 adalah memeriksa pesan baru yang diterima oleh *bot telegram*.

6. Mengirim Foto Ke *Telegram*

```
String sendPhotoTelegram(int vibrationLevel) {
    const char* myDomain = "api.telegram.org";
    String getAll = "";
    String getBody = "";
    digitalWrite(4, HIGH);
    camera_fb_t * fb = esp_camera_fb_get();
    delay(200);
    digitalWrite(4, LOW);

    if (!fb) {
        Serial.println("Camera capture failed");
        delay(1000);
        ESP.restart();
        return "Camera capture failed";
    }
}
```

Serial Monitor ✕

Gambar 2.17 Program 6

```
if (clientTCP.connect(myDomain, 443)) {
    String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat_i\r\n";
    String tail = "\r\n--RandomNerdTutorials--\r\n";

    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;

    clientTCP.println("POST /bot" + BOTtoken + "/sendPhoto HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length: " + String(totalLen));
    clientTCP.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");
    clientTCP.println();
    clientTCP.print(head);
}
```

Serial Monitor ✕

Gambar 2.18 Program 7

```

uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
for (size_t n = 0; n < fbLen; n += 1024) {
    size_t chunkSize = (n + 1024 < fbLen) ? 1024 : fbLen % 1024;
    clientTCP.write(fbBuf, chunkSize);
    fbBuf += chunkSize;
}

clientTCP.print(tail);
esp_camera_fb_return(fb);

int waitTime = 10000;
long startTimer = millis();
boolean state = false;

```

Serial Monitor ×

Gambar 2.19 Program 8

```

while ((startTimer + waitTime) > millis()) {
    delay(100);
    while (clientTCP.available()) {
        char c = clientTCP.read();
        if (state) getBody += String(c);
        if (c == '\n') {
            if (getAll.length() == 0) state = true;
            getAll = "";
        } else if (c != '\r') getAll += String(c);
        startTimer = millis();
    }
    if (getBody.length() > 0) break;
}
clientTCP.stop();
Serial.println(getBody);
} else {
    getBody = "Connection to api.telegram.org failed.";
    Serial.println(getBody);
}
return getBody;
}

```

Serial Monitor ×

Gambar 2.20 Program 9

Pada gambar 2.17, 2.18 dan 2.19 yaitu berfungsi sebagai mengambil foto dan menyimpannya di buffer, menghubungkan ke server *API Telegram* dan mempersiapkan data foto untuk diunggah. Mengirim data foto ke server Telegram dan mengembalikan hasilnya dan jika kamera gagal mengambil foto, *ESP32-CAM* akan restart.

7. Fungsi 'setup()'

```
void setup() {  
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);  
  Serial.begin(115200);  
  pinMode(VIBRATION_SENSOR_PIN, INPUT);  
  pinMode(4, OUTPUT);  
  digitalWrite(4, LOW);  
  
  WiFi.mode(WIFI_STA);  
  WiFi.begin(ssid, password);  
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);  
  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
}
```

Serial Monitor ×

Gambar 2.21 Program 10

```
Serial.println("\nWiFi connected");  
Serial.println("IP address: ");  
Serial.println(WiFi.localIP());  
  
camera_config_t config;  
config.ledc_channel = LEDC_CHANNEL_0;  
config.ledc_timer = LEDC_TIMER_0;  
config.pin_d0 = Y2_GPIO_NUM;  
config.pin_d1 = Y3_GPIO_NUM;  
config.pin_d2 = Y4_GPIO_NUM;  
config.pin_d3 = Y5_GPIO_NUM;  
config.pin_d4 = Y6_GPIO_NUM;  
config.pin_d5 = Y7_GPIO_NUM;  
config.pin_d6 = Y8_GPIO_NUM;  
config.pin_d7 = Y9_GPIO_NUM;  
config.pin_xclk = XCLK_GPIO_NUM;  
config.pin_pclk = PCLK_GPIO_NUM;  
config.pin_vsync = VSYNC_GPIO_NUM;  
config.pin_href = HREF_GPIO_NUM;  
config.pin_sscb_sda = SIOD_GPIO_NUM;  
config.pin_sscb_scl = SIOC_GPIO_NUM;  
config.pin_pwdn = PWDN_GPIO_NUM;  
config.pin_reset = RESET_GPIO_NUM;  
config.xclk_freq_hz = 20000000;  
config.pixel_format = PIXFORMAT_JPEG;
```

Serial Monitor ×

Gambar 2.22 Program 11


```
if (psramFound()) {
  config.frame_size = FRAMESIZE_UXGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
}

if (esp_camera_init(&config) != ESP_OK) {
  Serial.printf("Camera init failed");
  delay(1000);
  ESP.restart();
}

sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF);
}
```

Serial Monitor x

Gambar 2.23 Program 12

Pada gambar 2.21, 2.22 dan 2.23 untuk mencegah masalah saat tegangan rendah, untuk debugging, mengatur pin sensor getar dan *LED*, menghubungkan *ESP32-CAM* ke jaringan Wifi, serta mengatur konfigurasi kamera dan menginisialisasinya.

8. Fungsi 'Loop()'

```
void loop() {  
  int vibrationLevel = analogRead(VIBRATION_SENSOR_PIN);  
  
  if (vibrationLevel >= vibrationThresholdHigh) {  
    Serial.println("High vibration detected!");  
    sendPhoto = true;  
  } else if (vibrationLevel >= vibrationThresholdMedium) {  
    Serial.println("Medium vibration detected!");  
    sendPhoto = true;  
  } else if (vibrationLevel >= vibrationThresholdLow) {  
    Serial.println("Low vibration detected but not triggering photo.");  
  }  
}
```

Serial Monitor ×

Gambar 2.24 Program 13

```
  if (sendPhoto) {  
    bot.sendMessage(CHAT_ID, "Sending photo...");  
    sendPhotoTelegram(vibrationLevel);  
    sendPhoto = false;  
  }  
  
  if (millis() > lastTimeBotRan + botRequestDelay) {  
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);  
    while (numNewMessages) {  
      handleNewMessages(numNewMessages);  
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);  
    }  
    lastTimeBotRan = millis();  
  }  
  delay(50);  
}
```

Serial Monitor ×

Gambar 2.25 Program 14

Pada gambar 2.24 dan 2.25 menjelaskan program tentang membaca nilai dari sensor getaran. Misalnya jika getaran tinggi dan sedang terdeteksi maka akan mengambil foto dan mengirimkannya ke telegram. Jika getaran rendah terdeteksi, hanya mencetak pesan ke serial monitor

2.5.6 Pengujian sistem

Setelah perancangan hardware dan software, langkah berikutnya adalah menjalankan program dan menguji setiap untuk memastikan semua yang berfungsi sesuai yang diinginkan. Pengujian ini meliputi beberapa aspek penting, seperti mengukur respon sistem, mengevaluasi cakupan sistem, dan memastikan keseluruhan rangkaian bekerja dengan baik. Proses pengujian melibatkan pemeriksaan konektivitas Wifi, respon *bot Telegram*, dan deteksi getaran oleh sensor *SW-420*. Semua fungsi sistem, termasuk pengambilan dan pengiriman gambar, harus diuji untuk memastikan sistem beroperasi dengan baik dan sesuai harapan.

2.5.7 Implementasi

Setelah melakukan pengujian selanjutnya akan dilakukan implementasi pada alat yang telah dibuat. Pada tahap ini hasil yang sudah dirancang akan diimplementasikan untuk menjadi sebuah system yang nyata. Dalam konteks implementasi system keamanan cerdas untuk mencegah pencurian motor dengan integrasi *ESP32-CAM* dan sensor getar, berikut adalah beberapa Langkah implementasi yang dilakukan:

- A. Perancangan Fisik: Memasang komponen-komponen yang sudah di siapkan.
- B. Instalasi perangkat keras: Pasang perangkat keras ke motor sesuai dengan desain yang telah ditentukan.
- C. Konfigurasi perangkat lunak: Muat program yang telah dibuat kedalam *ESP32-CAM* menggunakan Arduino ID
- D. Dengan pengujian awal: Uji setiap komponen perangkat keras untuk memastikan fungsionalitasnya. Pastikan *ESP32-CAM* dapat mengambil gambar, sensor getar dapat mendeteksi getaran.
- E. Uji coba lapangan: Uji system secara menyeluruh dilapangan untuk memastikan semua system dapat mendeteksi dan memberikan respon yang tepat.

2.5.8 Evaluasi dan peningkatan

Melakukan Evaluasi sistem setelah digunakan dalam jangka waktu tertentu. Identifikasi area-area yang perlu ditingkatkan atau diperbaiki, dan melakukan perbaikan atau peningkatan sesuai kebutuhan .

2.5.9 Analisa Kerja

Analisa Kerja sistem keamanan akan melibatkan pemahaman tentang bagaimana setiap komponen berintraksi satu sama lain dan bagaimana sistem secara keseluruhan bekerja untuk mencapai tujuan perlindungan terhadap pencurian.

BAB III

HASIL ANALISIS DAN PEMBAHASAN

3.1 Hasil

Setelah merancang hardware dan software, langkah berikutnya adalah menjalankan program dan menguji setiap rangkaian untuk memastikan semuanya berfungsi sesuai yang diinginkan. Pengujian ini meliputi beberapa aspek penting, seperti mengukur respons sistem, mengevaluasi cakupan sistem, dan memastikan keseluruhan rangkaian bekerja dengan baik. Proses pengujian melibatkan pemeriksaan konektivitas WiFi, respons *bot Telegram*, dan deteksi getaran oleh sensor *SW-420*. Semua fungsi sistem, termasuk pengambilan dan pengiriman foto, harus diuji untuk memastikan sistem beroperasi dengan baik dan sesuai harapan.



Gambar 3.1 Pengujian Alat

Dalam pengujian ini mencakup evaluasi *ESP32-CAM* dan sensor getar (*SW-420*). Pengujian ini dilakukan untuk memungkinkan peneliti memahami keunggulan dan kelemahan sistem yang telah dikembangkan. Berikut adalah hasil pengujian yang tercatat:

1. Pengujian Koneksi Wifi dan *Telegram*

a. Pengujian Koneksi Wifi

Pengujian ini memeriksa apakah *ESP32-CAM* dapat terhubung ke jaringan WIFI dengan *SSID* dan password yang benar.

Hasil:

```
01:00:08.631 -> WiFi connected
01:00:08.631 -> IP address:
01:00:08.631 -> 172.20.10.3
```

Gambar 3.2 Output Pada Serial Monitor 1

Pada gambar 3.2 *ESP32-CAM* berhasil terhubung ke jaringan WIFI dan mendapatkan Alamat IP.

b. Pengujian Koneksi Telegram

Pada pengujian ini adalah memeriksa apakah *ESP32-CAM* dapat terhubung ke server dan mengirim pesan atau foto.

Hasil:

```
01:00:35.149 -> Connected to Telegram API
01:00:45.376 ->
```

Gambar 3.3 Output Pada Serial Monitor 2

Pada gambar 3.3 menjelaskan Ketika *ESP32-CAM* berhasil terhubung ke server Telegram dan mengirim pesan serta foto.

c. Tabel Hasil Pengujian Koneksi Wifi dan Telegram

Tabel 3.1 Hasil Pengujian Koneksi Wifi dan Telegram

No	Jenis Pengujian	Deskripsi Pengujian	Output pada Serial Monitor	Status	Observasi
1	Koneksi Wifi	Menghubungkan <i>ESP32-CAM</i> ke jaringan Wifi dengan <i>SSID</i> dan password yang benar	Wifi connected IP address: 192.168.1.2	Berhasil	<i>ESP32-CAM</i> berhasil terhubung ke WIFI dan mendapatkan Alamat IP
2	Koneksi Telegram(Pesan)	Mengirim pesan ke bot <i>Telegraam</i>	<i>Connected to Telegram API</i>	Berhasil	<i>ESP32-CAM</i> berhasil terhubung ke server <i>Telegram</i> dan mengirimkan pesan
3	Koneksi Telegram(Foto)	Mengirim foto ke bot <i>Telegram</i>	<i>Sending photo... Connected to Telegram API Photo sent! Vibration level: 2</i>	Berhasil	<i>ESP32-CAM</i> berhasil terhubung ke server <i>Telegram</i> dan mengirimkan foto

Pada tabel 3.1 yaitu memastikan bahwa sistem berhasil terhubung ke jaringan Wifi dan dapat berkomunikasi dengan server Telegram untuk mengirim pesan dan foto.

2. Pengujian Sensor Getar (*SW-420*)

a. Getaran Rendah

Pengujian pada gambar 3.4 yaitu dengan melakukan ketukan ringan pada sensor.

Hasil:

```
15:46:45.428 -> Low vibration detected but not triggering photo!
```

Gambar 3.4 Output Pada Serial Monitor 3

Pada gambar 3.4 menjelaskan sistem mendeteksi getaran namun tidak mencapai ambang batas sedang atau tinggi, sehingga tidak mengirim foto.

b. Getaran Sedang

Pengujian pada gambar 3.5 yaitu dengan mengetuk sensor dengan sedang

Hasil:

```
01:00:13.692 -> Medium vibration detected!  
01:00:35.149 -> Connected to Telegram API  
01:00:45.376 ->  
01:00:45.376 -> Time taken to send photo: 23022 ms
```

Gambar 3.5 Output Pada Serial Monitor 4

Pada gambar 3.5 menjelaskan sistem mendeteksi getaran sedang dan mengirimkan foto ke Telegram.

c. Getaran Tinggi

Pengujian pada gambar 3.6 yaitu dengan menjatuhkan benda berat dekat sensor.

Hasil:

```
01:01:26.884 -> High vibration detected!  
01:01:45.612 -> Connected to Telegram API  
01:01:47.491 -> {"ok":true,"result":{"message_id":2  
01:01:47.568 -> Time taken to send photo: 14411 ms
```

Gambar 3.6 Output Pada Serial Monitor 5

Pada gambar 3.6 sistem mendeteksi getaran tinggi dan mengirimkan foto ke *Telegram*.

d. Tabel Hasil Pengujian Sensor Getar

Tabel 3.2 Hasil Pengujian Sensor Getar

No	Tingkat Getaran	Deskripsi Pengujian	Output pada Serial Monitor	Pengiriman Foto ke Telegram	Observasi
1	Rendah	Ketukan ringan pada sensor	<i>Low Vibratio detected but not triggering photo</i>	Tidak	Sistem mendeteksi getaran rendah
2	Sedang	Getaran sedang pada sensor(mengetuk sensor dengan ketukan sedang)	<i>Medium vibration detected! Sending photo... Connected to Telegram API Photo sent! Vibration level: 2</i>	Ya	Sistem mendeteksi getaran sedang dan mengirimkan foto
3	Tinggi	Getaran kuat pada sensor (menjatuhkan benda berat dekat sensor)	<i>High vibration detected! Sanding photo.. Connected to Telegram API Photo sent! Vibration level: 3</i>	Ya	Sistem mendeteksi getaran tinggi dan mengirimkan foto

Tabel 3.2 diatas menunjukkan hasil pengujian sensor getaran engan berbagai tingkat getaran dan bagaimana sistem merespons masing-masing tingkat getaran dengan output yang sesuai serta status pengiriman foto ke Telegram.

3. Pengujian Kualitas Kamera *ESP32-CAM*

Pengujian kualitas kamera *ESP32-CAM* melibatkan pengujian berbagai parameter seperti resolusi gambar, waktu pengambilan gambar, waktu pengiriman gambar, dan kualitas hasil foto.

Hasil Pengujian :

Tabel 3.3 Kualitas Kamera ESP32-CAM

Pengaturan Kamera	Resolusi	Waktu Pengambilan Foto(ms)	Waktu Pengiriman Foto(ms)	Kualitas Hasil Foto
<i>UXGA</i>	1600x1200	1500	5000	Sangat Baik
<i>AVGA</i>	800x600	1000	3000	Baik
<i>CIF</i>	352x288	700	2000	Cukup
<i>QVGA</i>	320x240	500	1500	Rendah

a. *UXGA* (1600X1200)

1. Waktu Pengambilan Foto: 1500 ms. Waktu yang diperlukan untuk mengambil foto dengan resolusi tinggi lebih lama dibandingkan dengan resolusi lebih rendah.
2. Waktu Pengiriman Foto: 500 ms. Waktu pengiriman lebih lama karena ukuran file foto yang besar.
3. Kualitas Hasil Foto: Sangat baik, memberikan detail yang tinggi dan warna yang akurat.

b. *SVGA* (800x600)

1. Waktu Pengambilan Foto: 1000 ms. Waktu pengambilan lebih cepat dari pada *UXGA*.
2. Waktu Pengiriman Foto: 3000 ms. Waktu pengiriman lebih cepat dari pada *UXGA*
3. Kualitas Hasil Foto: Baik, memberikan detail yang baik dengan ukuran file yang lebih kecil daripada *UXGA*.

c. *CIF* (352x288)

1. Waktu Pengambilan Foto: 700 ms. Waktu pengambilan lebih cepat dari pada *SVGA*.
2. Waktu Pengiriman Foto: 2000 ms. Waktu pengiriman lebih cepat dari pada *SVGA*.
3. Kualitas Hasil Foto: Cukup, detail foto berkurang tetapi ukuran file lebih kecil.

d. *QVGA* (320x240)

1. Waktu Pengambilan Foto: 500 ms. Waktu pengambilan paling cepat.
2. Waktu Pengiriman Foto: 1500 ms. Waktu pengiriman paling cepat.
3. Kualitas Hasil Foto: Rendah, detail foto paling sedikit dan ukuran file terkecil.

Pada pengujian kamera ini kita dapat simpulkan bahwa semakin tinggi resolusi kualitas foto, semakin baik kualitas foto dengan yang lebih banyak dan warna yang lebih akurat. Waktu pengambilan dan pengiriman foto meningkat seiring dengan peningkatan resolusi, yang dapat mempengaruhi kinerja jika waktu adalah faktor penting. Pilih pengaturan kualitas berdasarkan kebutuhan aplikasi. Untuk aplikasi kebutuhan detail tinggi, gunakan resolusi tinggi. Untuk aplikasi yang memerlukan detail tinggi, gunakan resolusi tinggi. Untuk aplikasi yang memerlukan kecepatan dan efisiensi bandwidth, gunakan resolusi rendah.

Pada pengujian ini peneliti menggunakan pengawasan detail cukup yaitu menggunakan resolusi cukup seperti *CIF*.

4. Pengujian Pengambilan dan Pengiriman Foto

a. Pengujian Pengambilan Foto (Gagal)

Pada pengujian ini peneliti memeriksa apakah *ESP32-CAM* dapat mengambil foto dengan menggunakan perintah “/photo”.

Hasil:

```
[E][camera.c:1085] esp_camera_init(): Camera probe failed with error 0x20001  
Camera init failed with error 0x20001
```

Gambar 3.7 Output Pada Serial Monitor 6

Pada gambar 3.7 *ESP32-CAM* gagal mengambil foto. Kemungkinan masalah dengan inisialisasi kamera atau koneksi hardware.

b. Pengujian Pengambilan Foto (Berhasil)

Pada pengujian ini memeriksa apakah *ESP32-CAM* dapat mengambil dan mengirimkan foto otomatis saat mendeteksi getaran.

```
01:01:26.884 -> High vibration detected!  
01:01:45.612 -> Connected to Telegram API  
01:01:47.491 -> {"ok":true,"result":{"message_id":2  
01:01:47.568 -> Time taken to send photo: 14411 ms
```

Gambar 3.8 Output Pada Serial Monitor 7

Pada gambar 3.8 *ESP32-CAM* berhasil mengambil dan mengirimkan foto ke *bot Telegram*.

c. Tabel Hasil Pengujian Pengambilan dan Pengiriman Foto

Tabel 3.4 Hasil Pengujian Pengambilan dan Pengiriman Foto

No	Jenis Pengujian	Deskripsi Pengujian	Output pada Serial Monitor	Status	Observasi
1	Pengambilan Foto(Gagal)	Mengambil foto dengan perintah “/photo”	Sending photo.... Camera capture failed	Gagal	Kamera gagal mengambil foto ada masalah pada hardware atau inisialisasi
2	Pengambilan Foto(Berhasil)	Mengambil dan mengirim foto otomatis saat getaran sedang atau tinggi	High vibration detected! Sending photo Connected to Telegram API Photo sent! Vibration level: High	Berhasil	<i>ESP32-CAM</i> berhasil mendeteksi getaran mengambil foto dan mengirimkannya ke <i>bot Telegram</i>

Pada tabel 3.4 di atas peneliti dapat memastikan bahwa system berhasil mengambil dan mengirimkan foto ke bot Telegram berdasarkan perintah deteksi getaran.

Pengiriman foto dengan ambang getaran sedang, dan abang getaran tinggi bisa dilihat melalui *bot Telegram* pada gambar 3.9 di bawah.



Gambar 3.9 Notifikasi Telegram

Pada gambar 3.9 menunjukkan hasil notifikasi yang diperoleh dari sensor getar dan menunjukkan hasil ambang getaran berada di sedang dan tinggi.

3.2 Implementasi Alat

Setelah melakukan pengujian selanjutnya akan dilakukan implementasi pada alat yang telah dibuat. Pada tahap ini hasil yang sudah dirancang akan diimplementasikan untuk menjadi sebuah sistem yang nyata. Dalam konteks implementasi sistem keamanan cerdas untuk mencegah pencurian motor dengan integrasi *ESP32-CAM* dan sensor getar, berikut adalah beberapa langkah implementasi yang akan dilakukan:

1. Pemasangan Alat

Pada pemasangan alat ini dipasang di motor vario 160, alat diletakan dalam kotak untuk melindungi dari elemen cuaca. Untuk lokasi pemasangan alat, alat diletakan pada holder dekat kaca supion supaya bisa melakukan pemotretan wajah pencuri jika terjadinya getaran. Tahap ini merupakan bagian penting dari penelitian, di mana alat yang telah melalui proses pengujian diimplementasikan langsung pada sepeda motor, yang ditempatkan di halaman rumah. Implementasi ini bertujuan untuk memastikan bahwa alat dapat berfungsi dengan baik dalam kondisi nyata dan memberikan respons yang diharapkan ketika mendeteksi getaran. Berikut adalah gambar 3.10 pemasangan alat.



Gambar 3.10 Alat pada Motor

2. Keandalan Deteksi Getaran

Pengujian deteksi getaran bertujuan untuk mengevaluasi kinerja sensor *SW-420* dalam mendeteksi getaran pada sepeda motor dan memastikan bahwa sistem dapat merespons dengan tepat. Berikut adalah penjelasan proses pengujian serta tabel yang menggambarkan cara kerja alat secara rinci.

Tabel 3.5 Keandalan Deteksi Getaran

No	Skenario	Kondisi	Respon Sistem	Notifikasi Telegram
1	Tanpa Getaran	Sepeda Motor Diam	Tidak ada deteksi	Tidak ada notifikasi
2	Getaran Ringan	Ketukan ringan pada bodi	Getaran rendah terdeteksi tetapi tidak memicu foto	Tidak ada notifikasi
3	Getaran Sedang	Goyang stang atau bodi	Deteksi getaran sedang, ambil foto	Kirim foto dengan pesan “Getaran terdeteksi!”. Ambang Getaran Sedang
4	Getaran Kuat	Tendangan pada ban atau bodi	Deteksi getaran kuat, ambil foto	Kirim foto dengan pesan “Getaran terdeteksi!”. Ambang Getaran Tinggi
5	<i>Multipe</i>	Beberapa Getaran berturut - turut	Deteksi setiap getaran, ambil foto untuk setiap getaran	Kirim foto dengan pesan “Getaran terdeteksi!”. Ambang Getaran Tinggi

Dari hasil tabel 3.5 menunjukkan bahwa sistem mampu mendeteksi berbagai tingkat getaran dan merespon dengan mengirimkan notifikasi ke pengguna melalui Telegram. Sensitivitas sensor *SW-420* cukup baik dalam mendeteksi getaran dan memberikan notifikasi secara real-time, memastikan pengguna selalu mendapat informasi tentang potensi ancaman terhadap sepeda motor.



Gambar 3.11 Pengambilan Gambar

Gambar 3.11 menunjukkan bagaimana ketika saat alat mendeteksi getaran sedang dan tinggi maka alat akan memotret dan mengirimkan gambar melalui *bot Telegram*.

3. Kecepatan Respon Sistem

Respon sistem adalah waktu yang diperlukan untuk mendeteksi getaran, mengambil gambar, dan mengirimkan gambar melalui *bot Telegram*. Mengukur respons sistem adalah penting untuk memastikan bahwa notifikasi dapat dikirimkan secepat mungkin setelah terdeteksinya getaran, sehingga meningkatkan efektivitas sistem keamanan.

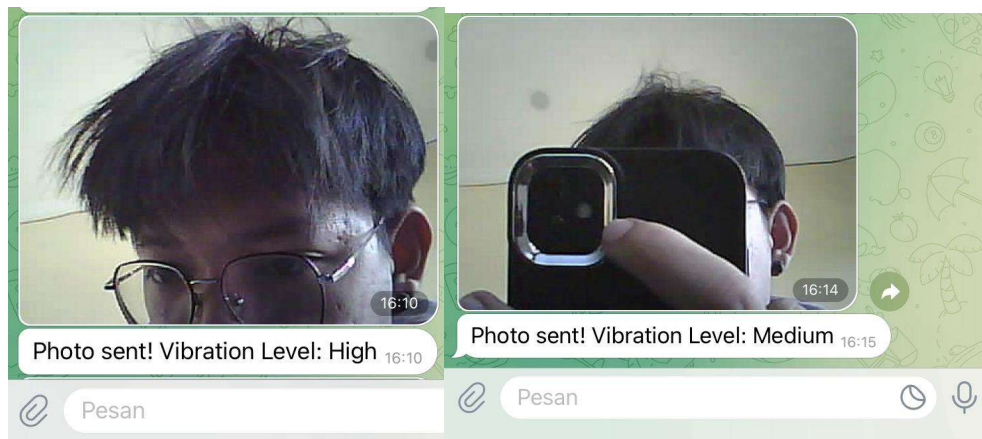
Pengujian respons sistem dilakukan dengan memberikan empat jenis getaran pada sepeda motor dan mengukur waktu yang dibutuhkan untuk setiap tahap proses, yaitu deteksi getaran, pengambilan gambar, dan pengiriman gambar.

Berikut adalah tabel yang menunjukkan waktu yang dibutuhkan untuk mendeteksi getaran dan mengirimkan gambar melalui *bot Telegram* pada berbagai skenario getaran:

Tabel 3.6 Respon Sistem

No	Tingkat Getaran	Jenis Getaran	Waktu Deteksi Getaran (ms)	Waktu Pengambilan Gambar (ms)	Waktu Pengiriman Gambar	Toal Waktu (ms)
1	Sedang	Goyang pada stang atau bodi	90	300	2000	2390
2	Kuat	Tendang pada ban atau bodi	80	300	2000	2380
3	Berturut - turut	Beberapa getaran berturut - turut	85	300	2000	2385

Tabel 3.6 mengelompokan getaran menjadi sedang, kuat, dan berturut-turut untuk menunjukkan variasi instensi getaran yang diuji. Waktu pengiriman gambar melalui *bot Telegram* relatif konsisten di semua tingkat getaran, berkisar sekitar 2000 ms. Tingkat getaran yang lebih tinggi (sedang dan kuat) cenderung terdeteksi lebih cepat oleh sensor *SW-420* dibandingkan dengan getaran ringan. Total waktu dari deteksi getaran hingga pengiriman gambar bervariasi antara 2385 hingga 2390 ms, menunjukkan bahwa sistem mampu merespon dengan cepat terhadap berbagai tingkat getaran.



Gambar 3.12 Notifikasi Telegram

Gambar 3.12 adalah notifikasi dari pengujian yang dilakukan pada sepeda motor, saat terjadinya getaran yang sedang dan kuat maka sistem akan otomatis mengirimkan gambar, jika getaran ringan dan tidak ada getaran sama sekali maka tidak ada notifikasi.

Pengujian menunjukkan bahwa sistem ini dapat mendeteksi getaran dan mengirimkan gambar dalam waktu yang relatif singkat, memberikan notifikasi real-time yang dapat diandalkan kepada pengguna. Kecepatan pengiriman yang konsisten memastikan bahwa pengguna dapat segera menerima informasi visual tentang kondisi sepeda motor setelah terdeteksi adanya getaran.

3.3 Pembahasan

Pada penelitian ini, peneliti mengembangkan sistem yang mengintegrasikan sensor getar *SW-420* dengan mikrokontroler *ESP32-CAM* untuk mendeteksi guncangan atau gerakan mencurigakan pada kendaraan motor. Pemilihan sensor getar *SW-420* didasarkan pada keandalannya dalam mendeteksi getaran dan biaya yang relatif rendah. Sensor ini mampu mendeteksi berbagai tingkat getaran yang dapat dikalibrasi untuk mengidentifikasi getaran mencurigakan. *ESP32-CAM* dipilih karena multifungsinya, yaitu dapat mengambil gambar dan mengirimkannya melalui jaringan Wifi. Kalibrasi sensor getar sangat penting untuk memastikan bahwa sensor hanya merespon getaran yang signifikan. Ambang batas getaran ditentukan melalui serangkaian pengujian, menunjukkan bahwa sensor dapat konsisten mendeteksi getaran yang disebabkan oleh aktivitas mencurigakan seperti upaya pencurian.

ESP32-CAM memiliki kemampuan untuk mengambil gambar dengan resolusi cukup baik. Pengambilan gambar dimulai setelah sensor getar mendeteksi aktivitas mencurigakan. Pengujian menunjukkan bahwa kualitas gambar sangat dipengaruhi oleh pencahayaan pengaturan kamera. Kamera diatur pada resolusi *CIF* (352x288) untuk memastikan ukuran file gambar kecil, sehingga dapat dikirim cepat melalui Wifi. Pengiriman foto menggunakan Telegram API merupakan komponen kunci dari sistem ini. Setelah gambar diambil, *ESP32-CAM* mengirimkannya ke *bot Telegram* yang kemudian meneruskan notifikasi ke pemilik kendaraan. Pengujian koneksi menunjukkan bahwa stabilitas jaringan Wifi sangat penting untuk memastikan pengiriman gambar berjalan lancar.

Stabilitas koneksi Wifi diuji dengan menghubungkan *ESP32-CAM* ke berbagai jaringan dengan tingkat kekuatan sinyal berbeda. Hasil menunjukkan bahwa koneksi Wifi yang stabil memungkinkan pengiriman notifikasi dan gambar dalam waktu relatif singkat. Namun, pada kondisi sinyal lemah, waktu pengiriman meningkat dan kadang terjadi kegagalan pengiriman. Sistem ini dirancang untuk memberikan respon cepat terhadap potensi ancaman. Dari pengujian yang dilakukan, waktu dari deteksi getaran hingga pengiriman notifikasi ke Telegram berkisar antara 2 hingga 3 detik tergantung pada kualitas sinyal Wifi. Waktu ini dianggap cepat untuk memberikan peringatan dini kepada pemilik kendaraan sehingga dapat segera mengambil tindakan.

Evaluasi keseluruhan menunjukkan bahwa integrasi sensor getar dengan *ESP32-CAM* ini efektif dalam mendeteksi dan melaporkan guncangan atau gerakan mencurigakan pada kendaraan motor. Sistem ini memberikan notifikasi real-time yang memungkinkan pemilik kendaraan untuk segera mengetahui potensi ancaman. Dalam perbandingan dengan penelitian sebelumnya yang menggunakan sensor getar dengan mikrokontroler lain atau sistem keamanan tanpa kemampuan pengambilan

gambar, sistem ini menawarkan keunggulan signifikan dalam fungsional dan respon waktu. Penggunaan *Telegram API* sebagai platform pengiriman notifikasi dan gambar menawarkan fleksibilitas dan kemudahan penggunaan yang lebih baik dibandingkan metode notifikasi tradisional.

Penelitian sebelumnya (Sujadi et al., 2018) mengembangkan sistem pemantauan keamanan sepeda motor berbasis *IoT* dengan *PIR* dan *Ultrasonik*, namun menghadapi keterbatasan sensitivitas terhadap gangguan cuaca dan lingkungan. (Zidane, 2022) menggunakan *ESP32-CAM* untuk mendeteksi wajah dan getaran *MOCLESS*, namun penggunaan sensor getar belum sepenuhnya dieksplorasi. Penelitian ini memperkuat teori bahwa sensor getar yang dikalibrasi dengan baik dapat memberikan deteksi yang lebih akurat dan respon cepat terhadap upaya pencurian, memperbaiki kelemahan dari solusi sebelumnya.

Beberapa rekomendasi yang diusulkan untuk peneliti selanjutnya termasuk penggunaan algoritma kompresi gambar lebih efisien, penambahan modul *GPS*, implementasi sistem backup dengan kartu *SIM* untuk mengirim notifikasi melalui jaringan seluler saat koneksi Wifi gagal, penambahan modul kecerdasan buatan untuk analisis gambar otomatis, pengujian dalam berbagai kondisi lingkungan, dan integrasi dengan sistem keamanan lainnya. Dengan demikian, penelitian ini tidak hanya menunjukkan efektivitas integrasi sensor getar dengan *ESP32-CAM*, tetapi juga memberikan dasar bagi pengembangan sistem keamanan kendaraan yang lebih canggih dan andal di masa depan.

BAB IV

PENUTUP

4.1 Simpulan

Penelitian ini bertujuan untuk mengembangkan sistem keamanan sepeda motor berbasis *ESP32-CAM* dan sensor getaran *SW-420*, dengan kemampuan mengirimkan notifikasi melalui *bot Telegram*. Sensor getaran *SW-420* telah terbukti mampu mendeteksi berbagai tingkat getaran dengan akurasi yang baik, memungkinkan sistem untuk merespons hanya terhadap getaran yang signifikan sebagai indikasi potensial upaya pencurian motor. Integrasi modul *ESP32-CAM* dengan *bot Telegram* sukses dalam mengirimkan notifikasi real-time saat terdeteksi getaran, dilengkapi dengan informasi visual berupa foto kondisi sekitar sepeda motor. Hal ini memberikan kemudahan bagi pengguna untuk memantau keadaan sepeda motor secara jarak jauh dan merespons kejadian mencurigakan dengan cepat.

4.2 Implikasi

Penelitian ini menghasilkan beberapa implikasi yang signifikan untuk berbagai bidang dan aplikasi, terutama dalam konteks keamanan kendaraan dan teknologi *Internet of Things (IoT)*:

1. Peningkatan Keamanan Kendaraan:

Sistem yang dikembangkan dapat diterapkan secara luas untuk meningkatkan keamanan sepeda motor dan kendaraan lainnya. Dengan mendeteksi getaran dan mengirim notifikasi real-time kepada pemilik, sistem ini dapat mengurangi risiko pencurian kendaraan dan meningkatkan respons terhadap potensi ancaman.

2. Pemanfaatan Teknologi *IoT*:

Penelitian ini menunjukkan potensi besar teknologi *IoT* dalam mengembangkan sistem keamanan yang terhubung dan cerdas. Penggunaan modul *ESP32-CAM* dan sensor *SW-420* sebagai bagian dari jaringan *IoT* memungkinkan pengawasan dan kontrol yang lebih baik terhadap aset berharga.

3. Adaptasi dan Skalabilitas:

Sistem ini dapat diadaptasi untuk berbagai aplikasi lain yang memerlukan deteksi getaran dan notifikasi real-time, seperti pemantauan keamanan bangunan, mesin industri, dan perangkat berharga lainnya. Skalabilitas sistem memungkinkan penggunaannya dalam berbagai skala dan lingkungan.

4. Meningkatkan Kesadaran dan Respons:

Sistem ini membantu meningkatkan kesadaran dan respons pemilik terhadap potensi ancaman. Dengan notifikasi langsung yang diterima di perangkat seluler, pemilik dapat mengambil tindakan cepat untuk melindungi kendaraannya.

4.3 Saran

Berdasarkan hasil penelitian ini, beberapa saran dapat diajukan untuk pengembangan lebih lanjut dan implementasi yang lebih efektif dari sistem keamanan sepeda motor berbasis *ESP32-CAM* dan sensor getaran *SW-420*:

1. Pengembangan Aplikasi Mobile:

Buat aplikasi mobile khusus yang lebih terintegrasi untuk pengguna. Aplikasi ini bisa memberikan notifikasi, menampilkan gambar yang diambil, dan memungkinkan pengguna mengatur sensitivitas sensor serta status perangkat eksternal.

2. Penambahan Fitur GPS:

Integrasikan modul GPS untuk memberikan lokasi real-time dari sepeda motor. Ini akan sangat membantu dalam pelacakan kendaraan yang hilang atau dicuri.

3. Pengujian Lapangan yang Lebih Luas:

Lakukan pengujian lapangan dalam berbagai kondisi lingkungan dan cuaca untuk memastikan keandalan dan robusta sistem. Pengujian dalam situasi nyata akan memberikan data lebih akurat tentang kinerja sistem.

4. Penggunaan Modul Tambahan:

Tambahkan modul seperti *SIM800L* sebagai cadangan untuk komunikasi data ketika jaringan Wifi tidak tersedia atau sinyal Wifi lemah.

DAFTAR RUJUKAN

- Alfi, R. N., Hijjayanti, K., Saptoaji, N., & Rizal, A. (2019). Analisis Perbandingan Kecepatan Transfer Data Dengan Kabel USB Tipe A Dan USB Tipe C. *NJCA (Nusantara Journal of Computers and Its Applications)*, 4(2), 144. <https://doi.org/10.36564/njca.v4i2.156>
- Andesta, D., & Ferdian, R. (2018). Sistem Keamanan Sepeda Motor Berbasis Mikrokontroler dan Modul GSM. *Journal of Information Technology and Computer Engineering*, 2(02), 51–63. <https://doi.org/10.25077/jitce.2.02.51-63.2018>
- Ardiansyah, Beni Irawan, & Tedy Rismawan. (2015). Rancang Bangun Sistem Keamanan Kendaraan Bermotor Dengan Sms Gateway Berbasis Mikrokontroler Dan Android. *Jurnal Coding, Sistem Komputer Untan*, 03(1), 11–19.
- Boy, A., Manullang, P., Saragih, Y., Hidayat, R., Elektro, S. T., Karawang, U. S., & Karawang, K. (2021). Implementasi nodemcu esp8266 dalam rancang bangun sistem keamanan sepeda motor berbasis iot. 4(2), 163–170.
- Hamdani, R., Puspita, H., & Wildan, D. R. (2019). Pembuatan Sistem Pengamanan Kendaraan Bermotor Berbasis Radio Frequency Identification (Rfid). *Indept*, 8(2), 56–63.
- Jonas, D., Supriyono, I. A., & Junianto, H. (2022). Perancangan Sistem Pencegahan Pencurian Kendaraan Bermotor Berbasis ESP32 pada PT. Suwarna Dwipa Maju. *Technomedia Journal*, 7(2), 216–230. <https://doi.org/10.33050/tmj.v7i2.1748>
- Lak'apu, D. A. ., Fahmi, I., & Tamal, C. P. (2023). Rancang Bangun Power Bank Untuk Modem Indihome ZTE F609. *Jurnal Spektro (Ejurnal.Undana.Ac.Id)*, Vol.6(No.2), 14–22.
- Mahanin Tyas, U., Apri Buckhari, A., Studi Pendidikan Teknologi Informasi, P., & Studi Pendidikan Teknologi dan Kejuruan, P. (2023). Implementasi Aplikasi Arduino Ide Pada Mata Kuliah Sistem Digital. *TEKNOS: Jurnal Pendidikan Teknologi Informasi*, 1(1), 1–9.
- Sujadi, H., Prasetyo, T. F., & Paisal, P. (2018). Pengembangan Sistem Monitoring Keamanan Sepeda Motor Berbasis Internet of Things. *J-Ensitec*, 5(01), 226–231. <https://doi.org/10.31949/j-ensitec.v5i01.1209>
- Williams, A. (1930). "Jumper." *Notes and Queries*, 158(15), 259–260. <https://doi.org/10.1093/nq/158.15.259-j>

Zidane, C. Z. (2022). Implementasi Esp 32 Cam Pada Alat Sistem Kendaraan Sepeda Motor (Mocless). *Power Elektronik: Jurnal Orang Elektro*, 11(2), 262. <https://doi.org/10.30591/polektro.v12i1.3818>

Zidane, C. Z., & Rahmadewi, R. (n.d.). *IMPLEMENTASI ESP 32 CAM PADA ALAT SISTEM KENDARAAN*. 262–265.

CODE PROGRAM

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

// WiFi credentials
const char* ssid = "Bobli";
const char* password = "apamserawa";

// Telegram Bot Token and Chat ID
String BOTtoken = "7442583602:AAHuLiRU-hAuJbHIIIKN8jmhF-uyYvUzxR0";
String CHAT_ID = "6842233029";

// Telegram and WiFi client initialization
WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

// Global variables
bool sendPhoto = false;
int botRequestDelay = 1000;
unsigned long lastTimeBotRan = 0;
unsigned long lastPhotoTime = 0;
const unsigned long photoInterval = 10000; // Minimum time between photos

// Camera pins definition
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
```

```

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

// Vibration sensor pin definition
#define VIBRATION_SENSOR_PIN 13

// Vibration thresholds
const int vibrationThresholdLow = 1;
const int vibrationThresholdMedium = 2;
const int vibrationThresholdHigh = 3;

void handleNewMessages(int numNewMessages) {
  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    String text = bot.messages[i].text;
    String from_name = bot.messages[i].from_name;

    if (text == "/start") {
      String welcome = "Welcome , " + from_name + "\n";
      bot.sendMessage(chat_id, welcome, "");
    } else if (text == "/photo") {
      sendPhoto = true;
      bot.sendMessage(chat_id, "New photo request received.");
    }
  }
}

```

```

String sendPhotoTelegram() {
    const char* myDomain = "api.telegram.org";
    String getAll = "";
    String getBody = "";
    digitalWrite(4, HIGH);
    camera_fb_t *fb = esp_camera_fb_get();
    delay(200);
    digitalWrite(4, LOW);

    if (!fb) {
        Serial.println("Camera capture failed");
        delay(1000);
        ESP.restart();
        return "Camera capture failed";
    }

    if (clientTCP.connect(myDomain, 443)) {
        String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat_id\";\r\n\r\n" +
        CHAT_ID + "\r\n--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
        String tail = "\r\n--RandomNerdTutorials--\r\n";

        uint16_t imageLen = fb->len;
        uint16_t extraLen = head.length() + tail.length();
        uint16_t totalLen = imageLen + extraLen;

        clientTCP.println("POST /bot" + BOTtoken + "/sendPhoto HTTP/1.1");
        clientTCP.println("Host: " + String(myDomain));
        clientTCP.println("Content-Length: " + String(totalLen));
        clientTCP.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");
        clientTCP.println();
        clientTCP.print(head);

        uint8_t *fbBuf = fb->buf;

```

```

    size_t fbLen = fb->len;
    for (size_t n = 0; n < fbLen; n += 1024) {
        size_t chunkSize = (n + 1024 < fbLen) ? 1024 : fbLen % 1024;
        clientTCP.write(fbBuf, chunkSize);
        fbBuf += chunkSize;
    }

    clientTCP.print(tail);
    esp_camera_fb_return(fb);

    int waitTime = 10000;
    long startTimer = millis();
    boolean state = false;

    while ((startTimer + waitTime) > millis()) {
        delay(100);
        while (clientTCP.available()) {
            char c = clientTCP.read();
            if (state) getBody += String(c);
            if (c == '\n') {
                if (getAll.length() == 0) state = true;
                getAll = "";
            } else if (c != '\r') getAll += String(c);
            startTimer = millis();
        }
        if (getBody.length() > 0) break;
    }
    clientTCP.stop();
    Serial.println(getBody);
} else {
    getBody = "Connection to api.telegram.org failed.";
    Serial.println(getBody);
}
return getBody;
}

```



```

void setup() {
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
    Serial.begin(115200);
    pinMode(VIBRATION_SENSOR_PIN, INPUT);
    pinMode(4, OUTPUT);
    digitalWrite(4, LOW);

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;

```

```

config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

if (esp_camera_init(&config) != ESP_OK) {
    Serial.printf("Camera init failed");
    delay(1000);
    ESP.restart();
}

sensor_t *s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF);
}

void loop() {
    int vibrationLevel = analogRead(VIBRATION_SENSOR_PIN);

    if (vibrationLevel >= vibrationThresholdHigh && millis() - lastPhotoTime > photoInterval) {
        Serial.println("High vibration detected!");
        sendPhoto = true;
        lastPhotoTime = millis();
    }
}

```

```

    } else if (vibrationLevel >= vibrationThresholdMedium && millis() - lastPhotoTime >
photoInterval) {
        Serial.println("Medium vibration detected!");
        sendPhoto = true;
        lastPhotoTime = millis();
    } else if (vibrationLevel >= vibrationThresholdLow) {
        Serial.println("Low vibration detected but not triggering photo.");
    }
}

if (sendPhoto) {
    bot.sendMessage(CHAT_ID, "Sending photo...");
    sendPhotoTelegram();
    sendPhoto = false;
}

if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
        handleNewMessages(numNewMessages);
        numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
}
delay(50);
}

```



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 056-001/KET/FST.1/A/2024
Lampiran : -
Perihal : Keterangan Melakukan Penelitian

Assalamu 'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Amin.

Dengan surat ini, kami menerangkan bahwa mahasiswa berikut:

No	Nama	NIM
1	Muhammad Fahri Alfianur	1911102441174
2	Rifat Fakhriy Naufal	2011102441041
3	Nur Anjeni Lestari	2011102441024
4	Ridha Anisa Soldzu Pamga	2011102441241
5	Bobli	2011102441069

Melakukan penelitian dengan membuat sebuah alat IoT di Laboratorium Hardware & Networking.

Demikian hal ini disampaikan, atas kerjasamanya kami ucapkan terima kasih.

Wassalamu 'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 18 Dzulhijjah 1445 H
25 Juni 2024 M

Direktur Program Studi S1 Teknik Informatika



Arbansyah, S.Kom., M.TI
IDN. 1118019203

L.1 Surat Izin Penelitian

KARTU KENDALI BIMBINGAN SKRIPSI

Nama Mahasiswa : Bobli
 NIM : 2011102441069
 Nama Dosen Pembimbing : Arbansyah, S.Kom, M.T.I.
 Judul Penelitian : Implementasi Sistem Keamanan Cerdas Untuk Pencegahan Pencurian Motor Dengan Integritasi ESP32-CAM Dan Sensor Getar

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	25 Januari 2024	Pengajuan judul skripsi	
2	29 Januari	Format Canva	
3	25 Februari 2024	Format pembuatan bab 1 & latar belakang	
4	20 Maret 2024	pembahasan 1 bab 2	
5	15 April 2024	Membuat rangkaian flowchart & penulisan	
6	20 April 2024	Membuat skematik rangkaian alat	
7	25 April 2024	membuat desain Visual	
8	26 April 2024	Uji tes alat	
10	25 Juni 2024	Membuat Program	
11	26 Jun. 2024	Uji tes Kegunaan alat	
12			
13			
14			

Dosen Pembimbing

 Arbansyah, S.Kom, M.Ti
 NIDN. 1118019203

Mengetahui,



 Program Studi

 Arbansyah, S.Kom, M.Ti
 NIDN. 1118019203

RIWAYAT HIDUP



Nama saya Bobli, lahir pada tanggal 27 juni 2002 di Desa Senyuir, saya memulai pendidikan dasar di SDN 001 Senyuir dari tahun 2008 sampai 2014, kemudian melanjutkan pendidikan menengah pertama di SMPN 004 Senyuir dari tahun 2014 sampai 2017. Setelah itu, saya menempuh pendidikan menengah atas di SMAN 002 Senyuir dari tahun 2017 sampai dengan 2020. Saat ini, saya sedang menyelesaikan studi di Universitas Muhammadiyah Kalimantan Timur . Selama masa kuliah, saya pernah melakukan magang di Universitas Muhamadiyah Kalimantan Timur selama 3 bulan. Pengalaman magang tersebut memberikan saya banyak pengetahuan praktis yang mendukung teori yang saya pelajari di bangku kuliah. Penulis sangat mengharapkan kritik dan saran yang membangun dari para pembaca mengenai skripsi ini. Semoga karya ini dapat memberikan manfaat dan kontribusi positif bagi perkembangan ilmu pengetahuan dan teknologi. Terima kasih.