

RANCANG BANGUN SISTEM PERINGATAN DINI RISIKO LONGSOR DAN GEMPA BERBASIS ESP32 DAN FIREBASE REALTIME DATABASE DENGAN KENDALI ALARM BERTINGKAT

Fauzi Bondan Prihananto¹, Erlangga Bayu Yudho Prakoso², Tomy Anugerah Islami³, Pramono⁴

Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta ^{1,2,3,4}

Email: 230103224@gmail.com¹, 230103133@mhs.udb.ac.id², 230103239@mhs.udb.ac.id³, pramono@udb.ac.id⁴

Informasi	Abstract
Volume : 3	<p><i>This article presents the development of an ESP32 and Firebase Realtime Database-based early warning system for landslide and earthquake risks. The main emphasis is placed on a multi-level alarm control mechanism rather than sensor monitoring only. The prototype employs an ADXL345 to capture tilt and vibration data, a Capacitive Soil Moisture Sensor to read soil moisture conditions, an ESP32 as the controller, Firebase as the real-time synchronization medium, a web dashboard as the monitoring interface, and a buzzer as the warning actuator. Sensor data are processed into SAFE, WARNING, and DANGER levels, enabling the system to produce different alarm responses according to the detected risk level. The dashboard provides data visualization, threshold configuration, event recording, device health monitoring, and administrator commands through Firebase. Testing was conducted by changing the ADXL345 position, applying vibration simulations, and comparing soil moisture value variations. The results indicate that the device can acquire sensor data, update Firebase records, display information on the dashboard, and drive the buzzer according to the risk status. The proposed system can be used as a local early warning prototype that combines IoT integration and alarm response control for landslide and earthquake mitigation.</i></p> <p>Keyword: ADXL345, adaptive alarm control, ESP32, Firebase Realtime Database, IoT, landslide, earthquake, soil moisture.</p>
Nomor : 7	
Bulan : Juli	
Tahun : 2026	
E-ISSN : 3062-9624	
Abstrak	<p>Artikel ini menyajikan pengembangan sistem peringatan dini untuk risiko longsor dan gempa dengan basis ESP32 dan Firebase Realtime Database. Fokus utama penelitian diarahkan pada mekanisme kendali alarm bertingkat, bukan hanya pemantauan nilai sensor. Prototipe menggunakan ADXL345 untuk memperoleh data kemiringan dan getaran, Capacitive Soil Moisture Sensor untuk membaca kondisi kelembapan tanah, ESP32 sebagai pengendali, Firebase sebagai media sinkronisasi real-time, dashboard web sebagai antarmuka pemantauan, dan buzzer sebagai aktuator peringatan. Data sensor diolah menjadi level AMAN, WASPADA, dan BAHAYA sehingga sistem dapat memberikan respons alarm yang berbeda sesuai tingkat risiko. Dashboard menyediakan visualisasi data, pengaturan ambang batas, pencatatan kejadian, pemantauan kesehatan perangkat, serta command admin melalui Firebase. Pengujian dilakukan dengan mengubah posisi sensor ADXL345, memberikan simulasi getaran, dan membandingkan variasi nilai kelembapan tanah. Hasil pengujian memperlihatkan bahwa perangkat mampu membaca sensor, memperbarui data ke Firebase, menampilkan informasi pada dashboard, serta menjalankan buzzer sesuai status risiko. Sistem ini dapat menjadi prototipe peringatan dini lokal yang menekankan integrasi IoT dan kendali respons alarm untuk mendukung mitigasi longsor dan gempa.</p>

Kata Kunci: ADXL345, ESP32, Firebase Realtime Database, IoT, kendali alarm, longsor, gempa, soil moisture.

A. PENDAHULUAN

Indonesia berada pada kawasan tektonik aktif sehingga potensi gempa bumi dan dampak ikutannya masih menjadi perhatian penting dalam mitigasi bencana [1], [2]. Selain menimbulkan kerusakan langsung, guncangan tanah dapat berhubungan dengan pergerakan lereng dan kejadian longsor, terutama pada wilayah dengan kondisi tanah labil atau basah [3]. Oleh sebab itu, perangkat peringatan lokal yang mudah dipasang dan mampu membaca perubahan kondisi tanah menjadi relevan untuk dikembangkan.

Sejumlah studi telah memanfaatkan accelerometer dan mikrokontroler untuk mendeteksi getaran maupun mengirimkan peringatan berbasis IoT [4], [5], [6]. Walaupun demikian, sistem yang hanya menampilkan data sensor belum selalu memberikan respons yang mudah dipahami oleh pengguna. Pada kondisi darurat, nilai sensor sebaiknya diterjemahkan menjadi level status dan aksi alarm agar pengguna memperoleh informasi praktis tanpa harus menafsirkan angka mentah.

Dalam penelitian ini, kendali tidak diarahkan untuk mengubah fenomena gempa atau longsor karena keduanya merupakan peristiwa alam. Kendali yang dimaksud adalah kendali terhadap respons peringatan. ESP32 menerima umpan balik dari sensor, membandingkan hasil pembacaan dengan ambang, menentukan level risiko, lalu mengatur pola *buzzer*. ADXL345 digunakan karena mampu membaca percepatan pada tiga sumbu [8], sedangkan ESP32 dipilih karena mendukung pemrosesan mikrokontroler dan konektivitas Wi-Fi untuk aplikasi IoT [9].

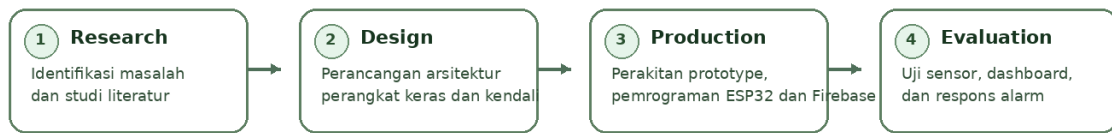
Penelitian ini bertujuan membangun sistem peringatan dini risiko longsor dan gempa berbasis ESP32 dan Firebase Realtime Database dengan kendali alarm bertingkat. Firebase digunakan untuk menyinkronkan data secara *real-time* antara perangkat dan *dashboard* [10]. Kontribusi penelitian meliputi integrasi sensor kelembapan dan accelerometer, klasifikasi level AMAN-WASPADA-BAHAYA, pemetaan pola *buzzer*, pencatatan log kejadian, serta panel admin untuk mengubah ambang dan mengirim *command* tanpa memprogram ulang ESP32.

B. METODE PENELITIAN

Penelitian menggunakan pendekatan *Research and Development* (R&D) karena keluaran yang dihasilkan berupa prototipe sekaligus evaluasi fungsinya. Pendekatan R&D sesuai untuk penelitian yang menghasilkan produk dan menguji kinerja produk tersebut [7]. Alur

penelitian dibagi menjadi tahap *research*, *design*, *production*, dan *evaluation* agar kebutuhan, rancangan, implementasi, dan pengujian dapat ditelusuri secara runtut.

Gambar Tahapan Penelitian

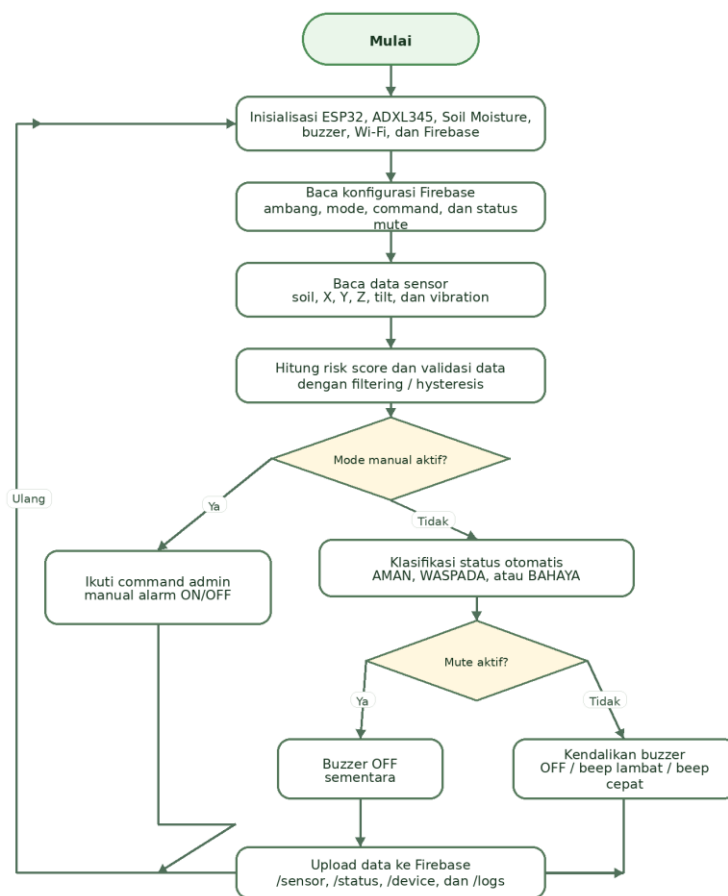


Gambar 1. Tahapan penelitian yang digunakan pada pengembangan sistem

A. Flowchart Sistem

Flowchart sistem menggambarkan alur kerja perangkat mulai dari inisialisasi, pembacaan konfigurasi Firebase, akuisisi data sensor, perhitungan *risk score*, pemilihan mode manual atau otomatis, pemeriksaan mute, pengendalian *buzzer*, hingga pengiriman data kembali ke Firebase. Alur ini menunjukkan bahwa sistem bekerja secara berulang sehingga perubahan parameter tanah dan getaran dapat langsung memengaruhi status serta pola alarm.

Flowchart Kerja Sistem Peringatan Dini



Gambar 2. Flowchart kerja sistem peringatan dini risiko longsor dan gempa

B. Kebutuhan dan Komponen Sistem

Komponen utama prototipe meliputi ESP32, ADXL345, *Capacitive Soil Moisture Sensor*, *buzzer*, koneksi Wi-Fi, Firebase Realtime Database, dan *dashboard* web. ADXL345 memberikan data percepatan tiga sumbu yang dapat digunakan untuk menganalisis kemiringan dan getaran [8]. ESP32 menjalankan akuisisi sensor, pengolahan status, komunikasi Wi-Fi, dan kendali *buzzer* [9]. Firebase dipakai sebagai lapisan *real-time* database untuk pertukaran data sensor, konfigurasi, *command*, dan status perangkat [10].

Tabel 1. Kebutuhan utama sistem

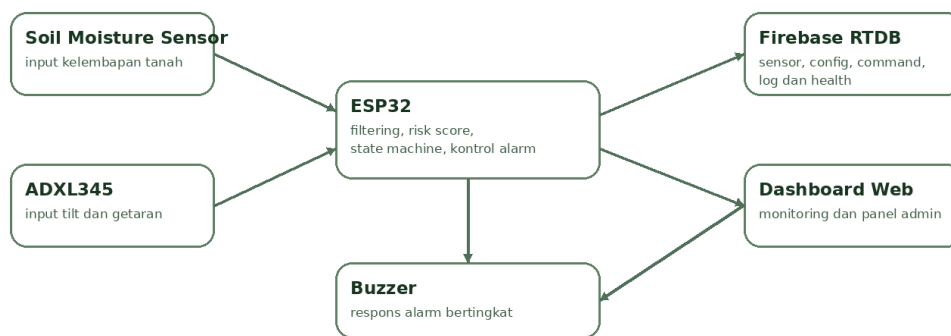
Aspek	Komponen/Kebutuhan	Fungsi
Sensor	ADXL345 dan <i>Capacitive Soil Moisture Sensor</i>	Menyediakan umpan balik kemiringan, getaran, dan kelembapan tanah.
Pengendali	ESP32	Memproses data, menentukan status, mengirim data, dan mengendalikan <i>buzzer</i> .
Aktuator	<i>Buzzer</i>	Menghasilkan alarm bertingkat sesuai level risiko.
Komunikasi	Wi-Fi dan Firebase RTDB	Menghubungkan alat dengan <i>dashboard</i> secara <i>real-time</i> .

Antarmuka	Dashboard web	Pemantauan, log, pemeriksaan kesehatan perangkat, dan panel admin.
-----------	---------------	--

C. Arsitektur Sistem

Arsitektur sistem dibangun dengan memisahkan lapisan input, pemrosesan, komunikasi, antarmuka, dan aktuator. Sensor mengirimkan data ke ESP32, ESP32 memproses status, Firebase menyimpan serta menyebarkan pembaruan data, *dashboard* menampilkan hasil dan mengirim perintah admin, sedangkan *buzzer* menjadi keluaran kendali peringatan. Dengan susunan tersebut, sistem memiliki loop keputusan yang berulang berdasarkan umpan balik sensor.

Gambar Arsitektur Sistem

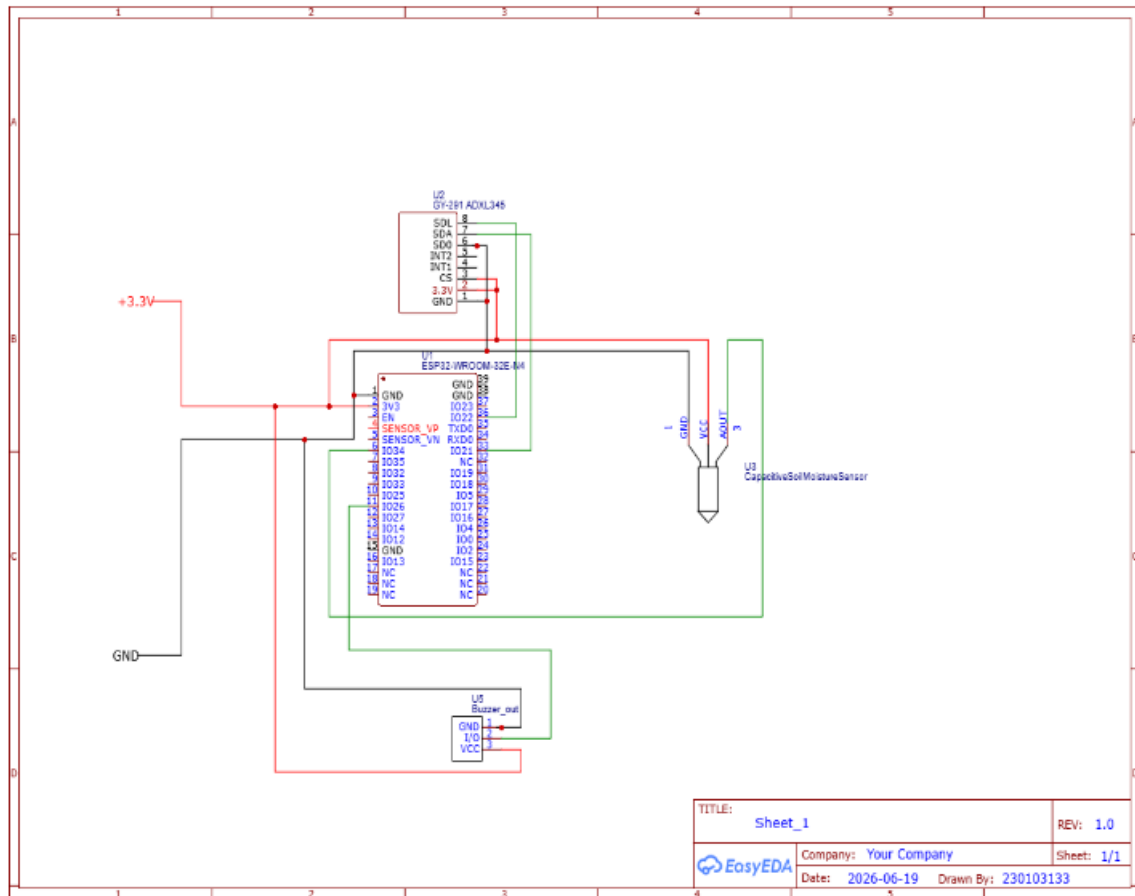


Keluaran terkendali: status dan pola alarm peringatan berdasarkan umpan balik sensor.

Gambar 3. Arsitektur sistem peringatan dini berbasis ESP32 dan Firebase

D. Skema Rangkaian dan Wiring Prototipe

Rangkaian perangkat keras menggunakan komunikasi I2C untuk menghubungkan ADXL345 ke ESP32, sedangkan sensor kelembapan tanah menggunakan masukan analog. Pada implementasi *firmware*, SDA ADXL345 dihubungkan ke GPIO 21 dan SCL ke GPIO 22. Keluaran analog *soil* moisture terhubung ke GPIO 34, sementara sinyal *buzzer* dikendalikan melalui GPIO 26. Pemilihan 3,3 V disesuaikan dengan karakteristik pin ESP32 agar pembacaan analog tetap aman.



Gambar 4. Skema rangkaian perangkat keras sistem

Pin/Komponen	Koneksi ke ESP32
ADXL345 VCC	3V3 ESP32
ADXL345 GND	GND ESP32
ADXL345 SDA	GPIO 21
ADXL345 SCL	GPIO 22
Soil Moisture AO	GPIO 34
Soil Moisture VCC	3V3 ESP32
Buzzer IN	GPIO 26
Buzzer VCC/GND	3V3 dan GND

Tabel 2. Konfigurasi pin utama prototipe

E. Logika Kendali Alarm Bertingkat

Algoritma alarm menerapkan *state machine* dengan status AMAN, WASPADA, BAHAYA, dan MUTE. Parameter *soil*, *tilt*, dan *vibration* dibandingkan dengan ambang yang tersimpan pada Firebase. Ketika nilai risiko meningkat, pola *buzzer* berubah dari mati menjadi beep lambat atau beep cepat. Hysteresis dan validasi beberapa sampel digunakan untuk mengurangi perubahan status yang terlalu sering akibat noise pembacaan sensor.

Gambar State Machine Alarm Bertingkat



Gambar 5. State machine kendali alarm bertingkat

Status	Kondisi Pemicu	Aksi Alarm
AMAN	Semua parameter di bawah ambang waspada	Buzzer mati
WASPADA	Salah satu parameter mendekati atau melewati ambang waspada	Beep lambat/intermiten
BAHAYA	Parameter melewati ambang bahaya atau risk score tinggi	Beep cepat / alarm darurat
MUTE	Perintah admin saat kondisi perlu dibisukan sementara	Buzzer mati sementara, status tetap tersimpan

Tabel 3. Hubungan status sistem dan aksi alarm

F. Struktur Data Firebase

Struktur Firebase disusun menjadi beberapa path utama supaya fungsi pemantauan dan kendali tidak bercampur. Path /sensor menyimpan hasil pembacaan, /config menyimpan ambang, /commands menampung perintah admin, /device mencatat kondisi perangkat, dan /logs menyimpan riwayat kejadian. Pemisahan ini mengikuti karakter Firebase Realtime Database sebagai penyimpanan JSON yang dapat disinkronkan langsung ke klien terhubung [10].

Gambar Struktur Firebase Realtime Database



Gambar 6. Struktur data utama pada Firebase Realtime Database

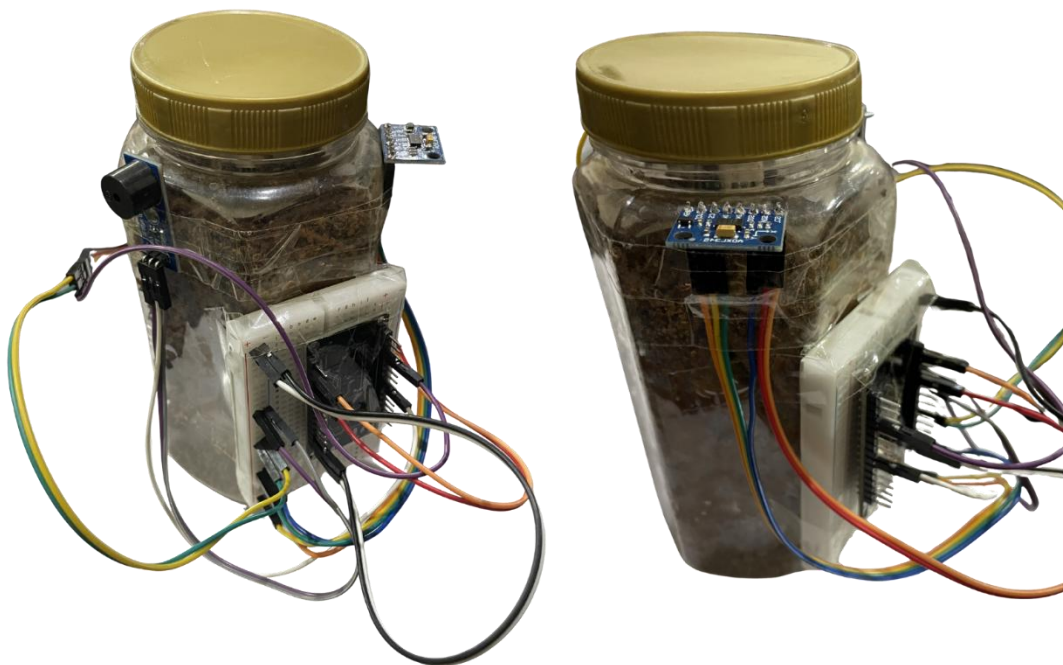
Path	Isi Data Utama	Fungsi
/sensor	soil, tilt, x, y, z, vibration, risk, status	Menyimpan pembacaan sensor dan hasil evaluasi sistem.
/config	soilWarn, soilDanger, tiltWarn, tiltDanger, vibrationWarn, vibrationDanger	Menyimpan setpoint/ambang kendali.
/commands	mute, testAlarm, scanWifi, calibrateADXL, mode	Menyimpan perintah admin dari dashboard.

/device	online, rssi, ip, uptime, <i>firmware</i>	Menyimpan data kesehatan perangkat.
/logs	timestamp, status, <i>soil</i> , <i>tilt</i> , <i>vibration</i> , message	Menyimpan riwayat kejadian sistem.

Tabel 4. Struktur path Firebase yang digunakan

G. Prototipe Alat yang Dikembangkan

Prototipe dirakit menggunakan *breadboard* dan kabel *jumper* dengan komponen ESP32, ADXL345, sensor *soil* moisture, serta *buzzer*. Perakitan tersebut digunakan untuk menguji alur akuisisi data, komunikasi Firebase, *dashboard real-time*, dan kendali alarm. Foto prototipe disertakan agar pembaca dapat melihat bentuk fisik perangkat yang digunakan dalam pengujian.



Gambar 7. Hasil perakitan prototipe alat

C. HASIL DAN PEMBAHASAN

A. Hasil Implementasi Sistem

Implementasi *firmware* menunjukkan bahwa ESP32 dapat melakukan pembacaan sensor, menghitung level risiko, mengirim data ke Firebase, serta menanggapi *command* dari *dashboard*. Antarmuka web menampilkan *soil*, *tilt*, *vibration*, *risk score*, status sistem, timeline kejadian, dan informasi kesehatan perangkat. Fitur administrasi yang diuji meliputi *mute*, *test alarm*, perubahan *threshold*, mode *auto/manual*, kalibrasi ADXL345, dan pemindaian Wi-Fi.

Komponen Uji	Deskripsi Hasil	Status
--------------	-----------------	--------

Pemantauan <i>real-time</i>	Dashboard menampilkan <i>soil, tilt, vibration, dan risk score</i>	Berhasil
Kendali alarm	Buzzer menyesuaikan status AMAN/WASPADA/BAHAYA	Berhasil
Panel admin	Command melalui Firebase diterapkan pada ESP32	Berhasil
Health check	Status online, RSSI, IP, uptime, <i>firmware</i> tampil	Berhasil
Log kejadian	Data status dan penyebab tersimpan pada log	Berhasil

Tabel 5. Ringkasan hasil implementasi sistem

B. Hasil Pengujian Respons Sensor dan Alarm

Pengujian dilakukan dengan membuat beberapa kondisi input untuk mengetahui hubungan antara nilai sensor, level risiko, dan respons *buzzer*. Skenario uji mewakili kondisi normal, peningkatan awal, bahaya, dan mute admin. Hasil pada Tabel 6 menunjukkan bahwa status sistem berubah sesuai kenaikan parameter dan *buzzer* mengikuti aturan alarm bertingkat.

No	Soil (%)	Tilt (°)	Vibration	Risk (%)	Status	Respons Buzzer
1	55	2.24	0.12	14	AMAN	OFF
2	62	4.10	0.26	31	AMAN	OFF
3	67	6.16	0.58	48	WASPADA	Beep lambat
4	74	9.40	1.20	73	BAHAYA	Beep cepat
5	80	12.87	2.35	91	BAHAYA	Beep cepat
6	80	12.87	2.35	91	MUTE (bahaya tersimpan)	OFF sementara

Tabel 6. Pengujian logika status dan respons alarm

C. Data Pengamatan Naik-Turun Parameter

Data pengamatan aktual disajikan kembali dalam bentuk urutan perubahan agar pola naik-turun sensor lebih mudah dianalisis. Penyajian ini juga membantu menunjukkan bahwa transisi status tidak berdiri sendiri, melainkan mengikuti perubahan nilai *tilt*/getaran dan *soil* dari waktu ke waktu. Ringkasan pengamatan ditampilkan pada Tabel 7.

No	Waktu	Tilt/Getaran	Soil	Status	Buzzer	Interpretasi Perubahan
1	11:13:59	2.24	3197	Aman	Mati	Baseline awal, kondisi stabil
2	11:13:59	2.24	3197	Aman	Mati	Nilai tetap, tidak ada perubahan
3	11:14:06	2.24	3204	Aman	Mati	Nilai <i>soil</i> naik, <i>tilt</i> tetap rendah

4	11:14:08	6.16	3216	Waspada	Mati	Tilt naik ke ambang waspada
5	11:14:08	12.87	3216	Bahaya	Hidup	Tilt naik tajam, alarm aktif
6	11:14:08	12.87	3216	Bahaya	Hidup	Kondisi bahaya bertahan
7	11:14:06	12.87	3204	Bahaya	Hidup	Status bahaya tetap saat parameter tinggi
8	11:14:06	2.24	3204	Aman	Mati	Tilt turun kembali, alarm mati

Tabel 7. Data pengamatan naik-turun parameter berdasarkan hasil uji sistem

D. Pengujian Dashboard dan Command Admin

Pengujian *command* admin dilakukan dengan mengirim perintah dari *dashboard* ke Firebase, kemudian diamati respons ESP32 pada perangkat. Hasilnya menunjukkan bahwa *command* dapat digunakan untuk membisukan alarm sementara, menguji *buzzer*, mengubah mode operasi, melakukan kalibrasi, memperbarui threshold, serta memicu pemindaian jaringan Wi-Fi. Fitur tersebut memperlihatkan peran Firebase sebagai jalur supervisory control.

Fitur	Path Firebase	Respons Sistem	Status
Mute sementara	<i>/commands/muteUntil</i>	<i>Buzzer</i> diam sesuai durasi	Berhasil
Test alarm	<i>/commands/testAlarm</i>	<i>Buzzer</i> berbunyi singkat	Berhasil
Mode auto/manual	<i>/mode</i>	Mode operasi berubah	Berhasil
Kalibrasi ADXL	<i>/commands/calibrateADXL</i>	Offset sensor diperbarui	Berhasil
Scan WiFi	<i>/commands/scanWifi</i>	Daftar SSID dikirim ke <i>dashboard</i>	Berhasil
Update threshold	<i>/config</i>	Setpoint diterapkan pada loop ESP32	Berhasil

Tabel 8. Hasil pengujian dashboard dan command admin

E. Estimasi Respons Real-time

Berdasarkan konfigurasi *firmware*, pembacaan sensor dilakukan sekitar setiap dua detik, heartbeat perangkat sekitar lima detik, dan pembacaan konfigurasi sekitar sepuluh detik. Karena *dashboard* menggunakan listener *real-time*, perubahan data dapat tampil tanpa refresh manual. Delay aktual dipengaruhi oleh kualitas Wi-Fi, waktu eksekusi loop ESP32, dan respons layanan Firebase.

Proses	Estimasi Delay	Keterangan
Sensor → Firebase → <i>Dashboard</i>	1-4 detik	Tergantung interval sampling dan jaringan
Heartbeat perangkat	±5 detik	Memperbarui status online dan health
Perintah mute / <i>test alarm</i>	1-3 detik	Respon cepat untuk fungsi admin
Perubahan threshold	hingga ±10 detik	Diterapkan sesuai interval sinkronisasi config
Scan WiFi	5-10 detik	Bergantung jumlah SSID yang terdeteksi

Tabel 9. Estimasi respons real-time sistem

F. Pembahasan

Hasil pengujian memperlihatkan bahwa kombinasi ESP32 dan Firebase Realtime Database mampu membentuk sistem peringatan dini yang tidak terbatas pada tampilan data. Adanya *state machine*, *risk score*, konfigurasi threshold, dan pemetaan alarm membuat sistem memiliki mekanisme kendali respons yang jelas. Hal ini menjadi pembeda utama dari sistem pemantauan yang hanya mengirim nilai sensor.

Firestore memberikan keuntungan karena dapat menjadi penghubung antara perangkat dan *dashboard* tanpa server tambahan. Data sensor, status perangkat, *command* admin, dan konfigurasi dapat dipisahkan dalam *path* yang berbeda sehingga pengelolaan sistem lebih mudah. Dari sisi pengguna, *dashboard* dapat dipakai untuk memantau kondisi sekaligus mengatur respons sistem dari jarak jauh.

Parameter *vibration* perlu dicatat secara eksplisit karena topik penelitian mencakup gempa. Jika log hanya berisi kelembapan dan *tilt*, maka kontribusi deteksi getaran menjadi kurang terlihat. Dengan memasukkan *vibration* ke *dashboard*, log, export data, dan tabel pengujian, hasil penelitian dapat menunjukkan hubungan antara getaran dan perubahan status alarm secara lebih lengkap.

Keterbatasan prototipe terletak pada skala uji yang masih menggunakan simulasi sederhana. Ambang *soil*, *tilt*, dan *vibration* perlu dikalibrasi berdasarkan jenis tanah, posisi pemasangan sensor, serta kondisi lokasi. Selain itu, panel admin yang masih berbasis PIN perlu diperkuat menggunakan Firebase Authentication dan rules database apabila sistem digunakan pada jaringan publik.

G. Analisis Elemen Sistem Kendali

Untuk memperjelas aspek sistem kendali, elemen-elemen kendali pada prototipe dipetakan pada Tabel 10. Pemetaan ini menunjukkan bahwa proses kendali terjadi pada respons alarm, sedangkan nilai kelembapan, kemiringan, dan getaran berfungsi sebagai

umpan balik. Dengan model tersebut, Firebase dapat dipandang sebagai lapisan supervisory control karena setpoint dan *command* dapat diperbarui dari *dashboard*, sementara ESP32 tetap menjadi controller lokal yang menjalankan keputusan secara periodik [9], [10].

Elemen	Implementasi pada Prototipe	Peran dalam Sistem
Umpan balik	Soil moisture, <i>tilt</i> , dan <i>vibration</i>	Menyediakan informasi kondisi lereng dan getaran untuk proses keputusan.
Controller	ESP32	Membaca sensor, menghitung status risiko, membaca <i>command</i> , dan mengatur <i>buzzer</i> .
Setpoint	Ambang <i>soil</i> , <i>tilt</i> , dan <i>vibration</i> pada Firebase	Menentukan batas peralihan status AMAN, WASPADA, dan BAHAYA.
Aktuator	<i>Buzzer</i>	Mewujudkan respons kendali berupa OFF, beep lambat, beep cepat, atau mute.
Keluaran terkendali	Pola alarm dan status peringatan	Menjadi respons sistem yang dapat dikendalikan secara otomatis maupun manual.
Gangguan	Noise sensor, perubahan kelembapan, dan simulasi getaran	Diredam dengan filtering, hysteresis, dan validasi status.

Tabel 10. Pemetaan elemen sistem kendali pada prototipe

Dari pemetaan tersebut, sistem memenuhi karakteristik dasar kendali berbasis umpan balik: terdapat sensor sebagai input, controller sebagai pengambil keputusan, setpoint sebagai nilai acuan, aktuator sebagai keluaran, serta mekanisme pembaruan status secara berulang. Walaupun tidak ada aktuator mekanis untuk mengubah kondisi tanah, pengendalian alarm tetap sah sebagai kendali respons peringatan karena sistem mengatur keluaran berdasarkan perubahan parameter yang dipantau.

Selain berperan pada sisi alarm, rancangan ini juga memudahkan proses audit kejadian karena setiap perubahan status dapat dikaitkan dengan parameter penyebabnya. Misalnya, status BAHAYA dapat ditelusuri apakah dipicu oleh peningkatan getaran, kenaikan kelembapan, perubahan *tilt*, atau kombinasi beberapa parameter. Informasi penyebab tersebut berguna untuk evaluasi setelah pengujian, karena tim pengembang dapat membedakan kondisi yang benar-benar dipengaruhi sensor dari kondisi yang muncul akibat gangguan koneksi atau pembacaan yang belum stabil. Dengan demikian, *dashboard* tidak hanya menjadi tampilan angka, tetapi juga menjadi alat bantu interpretasi keputusan sistem.

Pada skenario penggunaan lapangan, keluaran sistem perlu dipahami sebagai indikator kewaspadaan awal. Keputusan evakuasi tetap memerlukan pertimbangan manusia dan informasi resmi, tetapi alarm lokal dapat mempercepat perhatian pengguna terhadap

perubahan parameter di sekitar titik pemasangan alat. Karena itu, penyajian status, alasan status, dan riwayat kejadian pada *dashboard* menjadi bagian penting dari desain sistem.

D. KESIMPULAN

Sistem peringatan dini risiko longsor dan gempa berbasis ESP32 dan Firebase Realtime Database dengan kendali alarm bertingkat telah berhasil dibangun. Prototipe mampu membaca ADXL345 dan *Capacitive Soil Moisture Sensor*, mengolah data menjadi level risiko, mengirimkan data secara *real-time* ke Firebase, menampilkan hasil pada *dashboard*, serta mengendalikan *buzzer* pada status AMAN, WASPADA, BAHAYA, dan MUTE.

Pengujian menunjukkan bahwa Firebase tidak hanya berperan sebagai penyimpanan data, tetapi juga sebagai jalur pengaturan ambang dan *command* admin. Tabel pengamatan naik-turun parameter memperlihatkan bahwa transisi status mengikuti perubahan nilai sensor, sedangkan pengujian *command* memperlihatkan bahwa *dashboard* dapat mengendalikan respons sistem tanpa pemrograman ulang perangkat.

Prototipe ini dapat digunakan sebagai media peringatan dini lokal sekaligus dasar pengembangan sistem kendali alarm adaptif berbasis IoT. Pengembangan berikutnya dapat diarahkan pada pengujian lapangan, validasi ambang pada jenis tanah berbeda, penguatan keamanan akses admin, dan penambahan notifikasi eksternal seperti aplikasi mobile atau pesan instan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing, rekan satu tim, serta Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta atas arahan, dukungan, dan fasilitas yang diberikan selama proses pengembangan prototipe dan penyusunan artikel ini.

E. DAFTAR PUSTAKA

- [1] Y. Syafitri, B. Bahtiar, and L. A. Didik, "Analisis pergeseran lempeng bumi yang meningkatkan potensi terjadinya gempa bumi di Pulau Lombok," *Konstan-Jurnal Fisika dan Pendidikan Fisika*, vol. 4, no. 2, pp. 139-146, 2019.
- [2] R. Akbar, R. Darman, J. N. Marizka, and N. Ardewati, "Implementasi Business Intelligence Menentukan Daerah Rawan Gempa Bumi di Indonesia dengan Fitur Geolokasi," *JEPIN: Jurnal Edukasi dan Penelitian Informatika*, vol. 4, no. 1, pp. 30-35, 2018.
- [3] B. Arham and A. Wibowo, "Analisis Bencana Longsor akibat Gempa Bumi Cianjur 2022 dengan DInSAR," *Journal of Regional and Rural Development Planning*, vol. 8, no. 2,

2024.

- [4] L. Lenni and F. Ardiansyah, "Prototype Alat Pendeteksi Gempa Menggunakan Sensor Accelerometer ADXL335 Berbasis Internet of Things," *Jurnal Teknik Elektro*, vol. 8, no. 1, pp. 30-36, 2024.
- [5] J. Iriani and B. V. Sundawa, "Design of earthquake warning alarm using accelerometer sensor based on Internet of Things," *International Journal of Research in Vocational Studies (IJRVOCAS)*, vol. 3, no. 1, pp. 32-35, 2023.
- [6] N. Siswanto and S. F. Saputra, "Prototype Sistem Peringatan Dini Bencana Gempa Bumi dan Tsunami Berbasis Internet of Things," *Jurnal PROSISKO*, vol. 9, no. 1, pp. 60-66, 2022.
- [7] D. Sugiyono, *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta, 2013.
- [8] Analog Devices, "ADXL345: 3-Axis, ± 2 g/ ± 4 g/ ± 8 g/ ± 16 g Digital Accelerometer Data Sheet," Rev. G, 2022.
- [9] Espressif Systems, "ESP32 Series Datasheet," Version 5.2, 2025.
- [10] Google Firebase, "Firebase Realtime Database Documentation." [Online]. Available: <https://firebase.google.com/docs/database>.