Rancang Bangun Sistem Kendali Darurat Berbasis IoT Untuk Pengamanan Perangkat Melalui Akses Jarak Jauh Menggunakan ESP32 dan Firebase

Dimas Adji Yoga Saputra[1*], Ahmad Saifudin[2], Muhammad Alive Muflih[3], Pramono[4]

[1] Fakultas Ilmu Komputer
Universitas Duta Bangsa Surakarta
[1*]230103221@mhs.udb.ac.id

[2] Fakultas Ilmu Komputer
Universitas Duta Bangsa Surakarta
[2]230103214@mhs.udb.ac.id

[3] Fakultas Ilmu Komputer
Universitas Duta Bangsa Surakarta
[3]230103274@mhs.udb.ac.id

[4] Fakultas Ilmu Komputer
Universitas Duta Bangsa Surakarta
[4] pramono@udb.ac.id

Abstrak— **Perkembangan Internet of Things (IoT) memberikan kemudahan dalam melakukan pemantauan dan pengendalian perangkat secara jarak jauh, namun di sisi lain meningkatkan risiko keamanan akibat perangkat yang selalu terhubung ke internet. Sebagian besar sistem keamanan IoT masih berfokus pada proses monitoring dan deteksi ancaman sehingga belum menyediakan mekanisme mitigasi yang memungkinkan pengguna melakukan tindakan darurat secara langsung ketika perangkat mengalami gangguan keamanan. Penelitian ini bertujuan merancang dan mengimplementasikan sistem kendali darurat berbasis IoT yang mengintegrasikan mekanisme *Data Encryption* dan *Device Isolation* menggunakan ESP32 serta Firebase Realtime Database. Penelitian menggunakan metode eksperimental yang meliputi analisis kebutuhan, perancangan sistem, implementasi, serta pengujian terhadap konektivitas, waktu respons, dan tingkat keberhasilan sistem. Hasil pengujian menunjukkan bahwa komunikasi antara *Web Dashboard*, Firebase Realtime Database, ESP32, dan modul relay berjalan dengan baik. Rata-rata waktu respons sistem pada mode *Data Encryption* sebesar 2,149 detik, sedangkan mode *Device Isolation* sebesar 4,650 detik. Selain itu, sistem memperoleh tingkat keberhasilan eksekusi sebesar 100% pada 30 kali percobaan untuk masing-masing mode pengamanan. Hasil tersebut menunjukkan bahwa sistem mampu memberikan respons secara cepat dan andal dalam mendukung pengamanan perangkat serta data secara jarak jauh. Penelitian ini berkontribusi melalui integrasi mekanisme pengamanan digital dan isolasi perangkat fisik dalam satu platform berbasis IoT yang dapat digunakan sebagai solusi mitigasi pada kondisi darurat.**
Kata kunci— Internet of Things, ESP32, Emergency Control, Data Encryption, Device Isolation, Firebase Realtime Database.

***Abstract*— The development of the Internet of Things (IoT) facilitates remote device monitoring and control but simultaneously increases security risks due to constant internet connectivity. Most IoT security systems currently focus on monitoring and threat detection, lacking mitigation mechanisms that allow users to take immediate emergency action when a security breach occurs. This research aims to design and implement an IoT-based emergency control system that integrates data encryption and device isolation mechanisms using the ESP32 microcontroller and Firebase Realtime Database. An experimental method was employed, encompassing requirements analysis, system design, implementation, and testing of connectivity, response time, and system success rate. Test results indicate successful communication among the Web Dashboard, Firebase Realtime Database, ESP32, and relay module. The average system response time was 2.149 seconds for the data encryption mode and 4.650 seconds for the device isolation mode. Furthermore, the system achieved a 100% execution success rate across 30 trials for each security mode. These results demonstrate the system's ability to provide rapid and reliable responses for the remote security of devices and data. This research contributes by integrating digital security and physical device isolation mechanisms into a single IoT-based platform capable of serving as an emergency mitigation solution.**
Keywords— Internet of Things, ESP32, Emergency Control, Data Encryption, Device Isolation, Firebase Realtime Database.

I. PENDAHULUAN

Perkembangan teknologi *Internet of Things* (IoT) telah mendorong implementasi sistem cerdas pada berbagai bidang, mulai dari keamanan rumah dan otomasi berbasis ESP32 [1], *Industrial Internet of Things* (IIoT) untuk meningkatkan efisiensi proses manufaktur [13], pertanian cerdas berbasis IoT [6], hingga sistem pemantauan kesehatan yang memanfaatkan perangkat IoT dan jaringan 5G [8], [11]. Berbagai penelitian tersebut menunjukkan bahwa IoT mampu menyediakan komunikasi data secara *real-time* melalui dukungan teknologi *Low Power Wide Area Networks* (LPWAN) yang memungkinkan transmisi data berdaya rendah dan stabil [3], [9]. Namun, semakin luasnya penerapan IoT juga meningkatkan risiko keamanan karena perangkat yang selalu terhubung ke internet rentan terhadap akses ilegal, pencurian data, pengambilalihan

perangkat, maupun serangan siber lainnya [4], [10]. Kondisi tersebut menunjukkan bahwa selain meningkatkan konektivitas dan efisiensi, sistem IoT juga memerlukan mekanisme keamanan yang mampu memberikan perlindungan secara cepat ketika terjadi ancaman terhadap perangkat maupun data.

Untuk memitigasi risiko tersebut, berbagai penelitian telah mengusulkan beragam pendekatan keamanan pada sistem IoT. Mosenia dan Jha [4] membahas tantangan keamanan IoT secara komprehensif, sedangkan Ferrag et al. [5] mengembangkan skema perlindungan privasi pada jaringan ad hoc untuk meningkatkan keamanan komunikasi antarnode. Pendekatan berbasis *blockchain* juga dikembangkan untuk membangun mekanisme pertukaran data yang lebih terpercaya [7]. Selain itu, Ferrag et al. [12] menunjukkan bahwa algoritma *Machine Learning* mampu mendeteksi anomali dan serangan siber pada lingkungan IoT secara otomatis. Meskipun berbagai pendekatan tersebut memberikan hasil yang menjanjikan, sebagian besar penelitian masih berfokus pada perlindungan di tingkat perangkat lunak (*software*), analisis lalu lintas jaringan, atau deteksi ancaman sehingga belum menyediakan mekanisme mitigasi aktif yang dapat langsung mengamankan perangkat ketika ancaman terjadi.

Penelitian yang mengarah pada penanganan kondisi darurat mulai diperkenalkan oleh Damaševičius et al. [2] melalui konsep *Internet of Emergency Services* (IoES). Penelitian tersebut memanfaatkan jaringan IoT untuk mendukung koordinasi layanan tanggap darurat, penyampaian informasi, dan peringatan dini secara lebih cepat. Namun, implementasi yang diusulkan masih berorientasi pada manajemen layanan kedaruratan dan belum membahas mekanisme pengendalian langsung terhadap perangkat pengguna sebagai respons terhadap ancaman keamanan. Akibatnya, ketika terjadi pencurian perangkat fisik, akses tidak sah, atau indikasi peretasan secara *real-time*, pengguna masih belum memiliki mekanisme untuk melakukan tindakan mitigasi secara langsung dari lokasi yang berbeda. Kondisi tersebut menunjukkan perlunya sistem *Emergency Control* yang tidak hanya mampu mendeteksi ancaman, tetapi juga dapat melakukan tindakan pengamanan secara cepat terhadap perangkat yang terdampak.

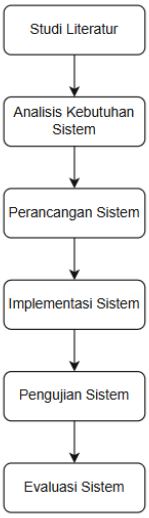
Berdasarkan kajian penelitian terdahulu, masih terdapat *research gap* berupa belum tersedianya sistem yang mengintegrasikan kemampuan pemantauan perangkat secara jarak jauh, pengamanan data melalui mekanisme enkripsi, serta isolasi perangkat secara fisik menggunakan relay dalam satu platform berbasis IoT yang dapat dikendalikan secara *real-time*. Oleh karena itu, penelitian ini mengusulkan rancang bangun sistem *Emergency Control* berbasis IoT menggunakan mikrokontroler ESP32 sebagai pusat eksekusi perangkat keras dan Firebase *Realtime Database* sebagai media komunikasi. Sistem yang dikembangkan mengimplementasikan dua mekanisme mitigasi aktif, yaitu mode *Data Encryption* untuk melindungi data sensitif dan mode *Device Isolation* melalui pemutusan suplai daya menggunakan relay. Integrasi kedua mekanisme tersebut menjadi kontribusi utama penelitian ini karena memungkinkan pengguna tidak hanya memantau kondisi perangkat dari jarak jauh, tetapi juga melakukan tindakan pengamanan secara cepat ketika ancaman keamanan terdeteksi.

II. Metodologi Penelitian

2.1 Metode Penelitian

Penelitian ini menggunakan metode eksperimental (*experimental research*) untuk merancang, mengimplementasikan, dan mengevaluasi sistem kendali darurat (*Emergency Control System*) berbasis *Internet of Things* (IoT). Metode ini dipilih karena penelitian tidak hanya berfokus pada pengembangan sistem, tetapi juga menguji secara langsung kinerja sistem melalui berbagai skenario pengujian. Tahapan penelitian meliputi studi literatur mengenai IoT, keamanan perangkat, komunikasi *cloud*, serta penelitian terdahulu yang relevan [1], [2], [10], [12], dilanjutkan dengan analisis kebutuhan, perancangan arsitektur sistem, implementasi dan integrasi

perangkat keras maupun perangkat lunak, serta pengujian terhadap komunikasi data, fungsi *Emergency Control*, *Data Encryption*, *Device Isolation*, waktu respons (*response time*), dan tingkat keberhasilan eksekusi (*success rate*). Hasil pengujian kemudian dianalisis secara deskriptif untuk mengevaluasi performa sistem dalam mendukung pengamanan perangkat dan data secara *real-time*. Tahapan penelitian secara keseluruhan ditunjukkan pada Gambar 1.1.



Gambar 1.1 Tahapan Penelitian

2.2 State of the Art Penelitian Terdahulu

Tabel 2.1 Perbandingan Sistem dengan Penelitian Terdahulu

No	Peneliti	Tahun	Fokus Penelitian	IoT	Remote Control	Emergency System	Device Isolation	Data Encryption	Firebase
1	Kodali et al.	2018	Smart Security and Home Automation berbasis ESP32	✓	✓	✗	✗	✗	✗
2	Mekki et al.	2019	Studi teknologi komunikasi LPWAN untuk IoT	✓	✗	✗	✗	✗	✗
3	Jo et al.	2019	Emergency Management System pada Smart Building	✓	✓	✓	Sebagian	✗	✗
4	Ahmed et al.	2020	Smart Home Monitoring System berbasis IoT	✓	✓	✗	✗	✗	✗
5	Alsaadi et al.	2021	Real-Time Monitoring and Control berbasis IoT	✓	✓	✗	✗	✗	✗
6	Sharma et al.	2022	Implementasi keamanan data pada perangkat IoT	✓	✗	✗	✗	✓	✗
7	Ashraf et al.	2023	IoT-Based Monitoring System menggunakan ESP32	✓	✓	✗	✗	✗	✓
8	Damaševičius et al.	2023	Internet of Emergency Services (IoES)	✓	✓	✓	✗	✗	✗
9	Hossain et al.	2024	Cloud-Based IoT Security Framework	✓	✓	✗	✗	✓	✗
10	Penelitian Usulan	2025	Sistem Kendali Darurat Berbasis IoT untuk Pengamanan Perangkat	✓	✓	✓	✓	✓	✓

Berdasarkan Tabel 2.1 dapat diketahui bahwa penelitian terdahulu umumnya hanya mengimplementasikan sebagian fungsi, seperti komunikasi IoT, *remote monitoring*, maupun sistem tanggap darurat. Belum terdapat penelitian yang mengintegrasikan seluruh fungsi utama, yaitu *remote control*, *Emergency Control*, *device isolation*, *data encryption*, serta sinkronisasi data secara *real-time* menggunakan Firebase Realtime Database dalam satu platform. Oleh karena itu, penelitian ini menawarkan integrasi seluruh fitur tersebut sebagai upaya meningkatkan kemampuan sistem dalam melakukan pengamanan perangkat dan data secara cepat ketika terjadi kondisi darurat.

2.3 Analisis Kebutuhan dan Spesifikasi Komponen

Analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan fungsional dan nonfungsional sebagai dasar perancangan serta implementasi sistem kendali darurat berbasis *Internet of Things* (IoT). Hasil analisis digunakan sebagai acuan dalam menentukan fungsi sistem, perangkat keras, dan perangkat lunak yang diperlukan agar seluruh proses komunikasi data dan pengendalian perangkat dapat berjalan secara *real-time*. Rincian kebutuhan sistem disajikan pada Tabel 2 dan Tabel 3.

Tabel 2.2 Kebutuhan Fungsional

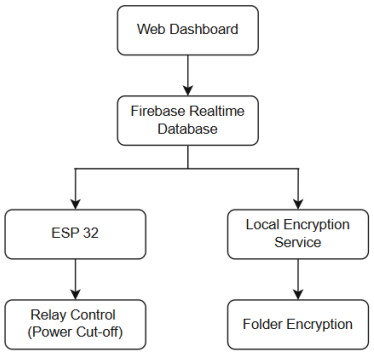
No	Kebutuhan Fungsional
1	Sistem mampu melakukan autentikasi pengguna melalui *Web Dashboard*.
2	Sistem mampu mengirimkan perintah ke Firebase Realtime Database secara *real-time*.
3	ESP32 mampu membaca perubahan data dan mengeksekusi perintah yang diterima.
4	Sistem mampu menjalankan fitur *Data Encryption*.
5	Sistem mampu menjalankan fitur *Device Isolation* melalui modul relay.
6	Sistem mampu menampilkan status perangkat secara *real-time* pada *Web Dashboard*.

Tabel 2.3 Kebutuhan Nonfungsional

Kategori	Komponen	Fungsi
Perangkat Keras	ESP32 DevKit V1	Pengendali utama sistem
	Relay 5V	Memutus dan menghubungkan suplai daya perangkat
	Laptop/Komputer	Pengembangan dan pengujian sistem
	Jaringan Wi-Fi	Media komunikasi antara ESP32 dan Firebase
Perangkat Lunak	Arduino IDE	Pemrograman ESP32
	Visual Studio Code	Pengembangan *Web Dashboard*
	Firebase Realtime Database	Sinkronisasi data *real-time*
	Web Browser	Antarmuka pengguna

2.4 Perancangan Sistem

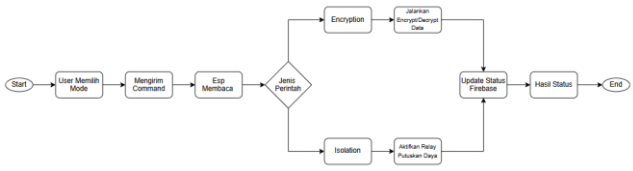
Perancangan sistem dilakukan untuk mengintegrasikan perangkat keras dan perangkat lunak ke dalam satu sistem kendali darurat berbasis *Internet of Things* (IoT). Sistem menggunakan arsitektur *client-server* yang terdiri atas *Web Dashboard* sebagai antarmuka pengguna, Firebase Realtime Database sebagai media komunikasi *real-time*, ESP32 DevKit V1 sebagai pusat pengendali, modul relay sebagai aktuator *Device Isolation*, serta layanan *Data Encryption* sebagai mekanisme pengamanan data. Pengguna mengirimkan perintah melalui *Web Dashboard*, kemudian perintah diteruskan ke Firebase untuk dibaca dan dieksekusi oleh ESP32 sesuai mode yang dipilih, yaitu *Data Encryption* atau *Device Isolation*. Setelah proses eksekusi selesai, status sistem dikirim kembali ke Firebase dan ditampilkan pada *Web Dashboard* sebagai umpan balik kepada pengguna. Diagram blok arsitektur sistem ditunjukkan pada Gambar 2.1



Gambar 2.1 Diagram Blok Sistem

2.5 Alur Kerja Sistem

Alur kerja sistem diawali ketika pengguna mengirimkan perintah melalui *Web Dashboard*, kemudian perintah diteruskan ke Firebase Realtime Database untuk dibaca secara *real-time* oleh ESP32. Berdasarkan jenis perintah yang diterima, ESP32 akan mengeksekusi salah satu mode pengamanan, yaitu *Data Encryption* untuk mengamankan data atau *Device Isolation* dengan mengaktifkan modul relay guna memutus suplai daya perangkat. Setelah proses eksekusi selesai, status sistem dikirim kembali ke Firebase Realtime Database dan ditampilkan pada *Web Dashboard* sebagai umpan balik kepada pengguna sehingga proses pemantauan dan pengendalian dapat berlangsung secara *real-time*. Diagram alur kerja sistem ditunjukkan pada Gambar 3.

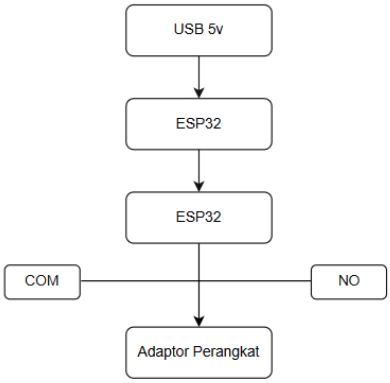


Gambar 2.2 Diagram Alur Kerja

2.6 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan untuk membangun infrastruktur fisik yang mendukung fungsi *Emergency Control*. Sistem menggunakan ESP32 DevKit V1

sebagai pengendali utama yang terhubung ke Firebase Realtime Database melalui jaringan Wi-Fi, serta modul relay 5V sebagai aktuator untuk menjalankan fungsi *Device Isolation* dengan memutus suplai daya perangkat ketika kondisi darurat terjadi. Selain itu, sistem didukung oleh adaptor catu daya dan komputer sebagai media pengembangan serta pengujian sehingga seluruh komponen dapat berkomunikasi dan beroperasi secara stabil. Skema hubungan antara ESP32 dan modul relay ditunjukkan pada Gambar 2.3.



Gambar 2.3 Skema Kelistrikan Pada Perangkat

2.7 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan untuk mendukung komunikasi data, pengendalian perangkat, dan penyajian informasi secara *real-time*. Sistem terdiri atas *Web Dashboard* sebagai antarmuka pengguna, Firebase Realtime Database sebagai media sinkronisasi data, program ESP32 sebagai pengendali utama, serta layanan *Data Encryption* sebagai mekanisme pengamanan digital. Pengguna mengirimkan perintah melalui *Web Dashboard*, kemudian perintah disimpan pada Firebase untuk dibaca dan dieksekusi oleh ESP32 sesuai mode yang dipilih, yaitu *Device Isolation* atau *Data Encryption*. Setelah proses eksekusi selesai, status sistem diperbarui pada Firebase dan ditampilkan kembali pada *Web Dashboard* sehingga pengguna dapat memantau kondisi perangkat secara *real-time*.

2.8 Skenario Pengujian

Pengujian dilakukan setelah proses implementasi selesai untuk memastikan seluruh fungsi sistem kendali darurat berbasis *Internet of Things* (IoT) berjalan sesuai rancangan. Evaluasi meliputi komunikasi data, *Data Encryption*, *Device Isolation*, waktu respons (*response time*), dan tingkat keberhasilan sistem (*success rate*). Setiap skenario pengujian dilakukan melalui *Web Dashboard* dengan mengamati proses sinkronisasi data pada Firebase Realtime Database, eksekusi perintah oleh ESP32, serta umpan balik yang ditampilkan kepada pengguna. Rincian skenario pengujian disajikan pada Tabel 2.4.

Tabel 2.4 Skenario Pengujian Sistem

No	Skenario Pengujian	Parameter yang Diuji
1	Komunikasi Sistem	Keberhasilan komunikasi antara *Web Dashboard*, Firebase, dan ESP32
2	*Data Encryption*	Keberhasilan proses enkripsi dan sinkronisasi status
3	*Device Isolation*	Keberhasilan relay memutus suplai daya perangkat
4	*Response Time*	Waktu eksekusi perintah dari pengguna hingga perangkat merespons
5	*Success Rate*	Persentase keberhasilan seluruh fungsi sistem

2.9 Metode Evaluasi

Evaluasi dilakukan untuk menilai performa sistem berdasarkan hasil pengujian yang telah dilakukan. Pengujian fungsional menggunakan metode *Black Box Testing* untuk memastikan setiap fitur, meliputi autentikasi pengguna, komunikasi antara *Web Dashboard*, Firebase Realtime Database, ESP32, *Data Encryption*, dan

Device Isolation, berjalan sesuai keluaran yang diharapkan. Selain itu, evaluasi dilakukan terhadap waktu respons (*response time*) dengan mengukur selisih waktu sejak perintah dikirim hingga berhasil dieksekusi, serta tingkat keberhasilan (*success rate*) yang dihitung berdasarkan persentase jumlah pengujian berhasil terhadap total pengujian menggunakan Persamaan (1). Seluruh hasil pengujian kemudian dianalisis secara deskriptif untuk mengevaluasi kesesuaian performa sistem dengan tujuan penelitian.

$$Success\ Rate = \frac{Jumlah\ Pengujian\ Berhasil}{Total\ Pengujian} \times 100\% \quad (1)$$

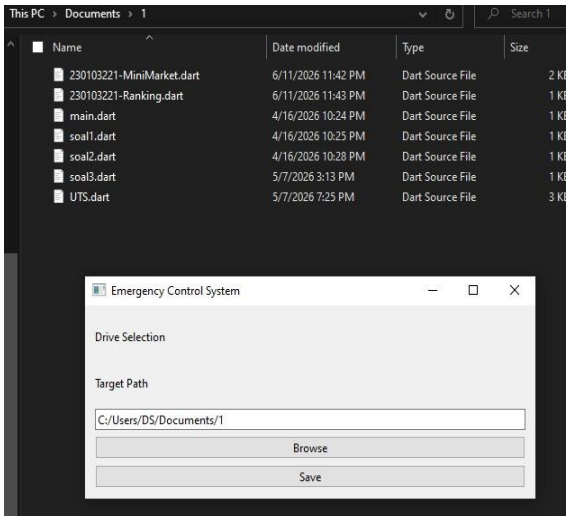
III. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Tahap implementasi dilakukan setelah seluruh proses perancangan sistem selesai. Implementasi bertujuan mengintegrasikan perangkat keras dan perangkat lunak sehingga sistem kendali darurat berbasis *Internet of Things* (IoT) dapat beroperasi sesuai rancangan. Sistem terdiri atas *Web Dashboard* sebagai antarmuka pengguna, Firebase Realtime Database sebagai media komunikasi data secara *real-time*, mikrokontroler ESP32 sebagai pengendali utama, modul relay sebagai aktuator pemutus daya, serta layanan enkripsi data sebagai mekanisme pengamanan digital.

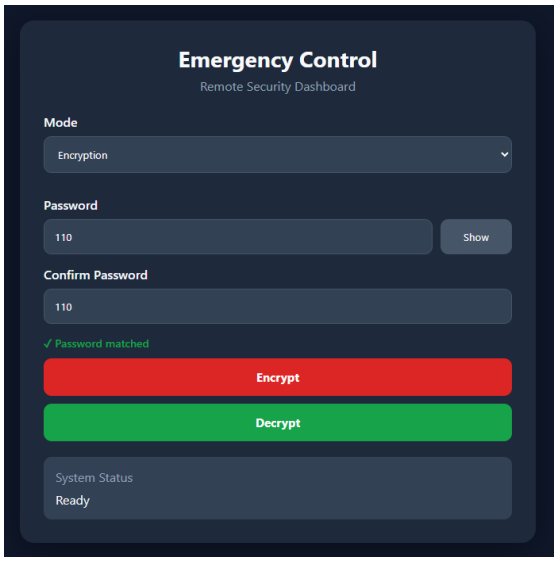
a. Implementasi Web Dashboard

Web Dashboard dikembangkan sebagai media interaksi antara pengguna dan sistem. Melalui antarmuka ini pengguna dapat mengakses fitur *Emergency Control*, memonitor status koneksi perangkat, serta mengirimkan perintah pengamanan secara jarak jauh. Dashboard dirancang menggunakan antarmuka yang responsif sehingga dapat diakses melalui berbagai perangkat, baik komputer maupun perangkat bergerak.



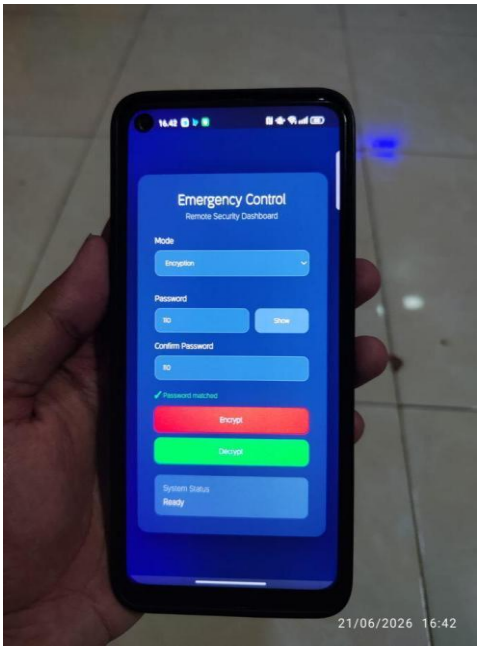
Gambar 3.1 Tampilan Sistem Enkriptor Lokal

Gambar 3.1 menunjukkan tampilan sistem enkriptor yang digunakan untuk mengelola proses pengamanan data



Gambar 3.2 Tampilan Web Dashboard

Gambar 3.2 memperlihatkan tampilan utama *Web Dashboard* dimana pengguna dapat memilih mode, memasukkan password yang diinginkan, lalu memilih aksi yang ingin dilakukan.



Gambar 3.3 Tampilan Web Dashboard di Perangkat Lain

Gambar 3.3 menunjukkan bahwa antarmuka tetap dapat diakses dengan baik pada perangkat lain tanpa mengurangi fungsi utama sistem.

b. Implementasi Sistem Enkripsi

Fitur *Data Encryption* diimplementasikan sebagai mekanisme pengamanan data ketika pengguna mendeteksi adanya potensi ancaman terhadap perangkat. Ketika mode enkripsi diaktifkan melalui *Web Dashboard*, sistem akan mengirimkan status perintah ke Firebase Realtime Database. Selanjutnya layanan enkripsi membaca perubahan status tersebut dan menjalankan proses enkripsi terhadap data yang telah ditentukan.

```
[CMD] encrypt
[23:53:12.018] [ENC] C:/Users/DS/Documents/001 uji
[CMD] decrypt
[23:53:43.403] [DEC] C:/Users/DS/Documents/001 uji
ms)
[CMD] encrypt
[23:54:05.158] [ENC] C:/Users/DS/Documents/001 uji
[CMD] decrypt
[23:54:26.878] [DEC] C:/Users/DS/Documents/001 uji
ms)
```

Gambar 3.4 Tampilan Serial Monitor Ketika ESP32 Menerima Perintah Isolasi

c. Implementasi Integrasi ESP32 dan Firebase

ESP32 dihubungkan dengan jaringan Wi-Fi dan Firebase Realtime Database sehingga mampu melakukan sinkronisasi data secara *real-time*. Mikrokontroler secara terus-menerus memantau perubahan status pada database. Ketika pengguna mengaktifkan mode *Device Isolation*, ESP32 menerima perintah dari Firebase kemudian mengendalikan modul relay untuk memutus suplai daya perangkat target. Setelah proses eksekusi selesai, status sistem diperbarui kembali pada Firebase sehingga informasi keberhasilan tindakan dapat ditampilkan kepada pengguna melalui *Web Dashboard*.

```
23:42:57.168 -> Reading Firebase
23:42:57.579 -> {"approved":false,"command":"cutoff","cre
23:42:57.617 -> Realtime  : 2026-07-10 23.42.56.893
23:42:57.617 -> Command   : cutoff
23:42:57.617 -> Approved  : false
23:42:57.617 -> Executed  : false
23:42:57.617 -> CreatedAt : 1783701775537
23:42:57.617 -> ExpireAt  : 1783701785537
23:42:57.617 -> Password  :
23:42:57.617 -> Status    : pending
23:42:57.617 -> State     : false
23:42:57.617 -> RELAY OFF
23:42:58.127 -> Marked Executed
```

Gambar 3.5 Tampilan Serial Monitor Ketika ESP32 Menerima Perintah Isolasi

3.2 Hasil Pengujian Sistem

Pengujian dilakukan berdasarkan skenario yang telah dijelaskan pada Bab II untuk mengevaluasi kinerja sistem dari aspek komunikasi data, waktu respons, serta tingkat keberhasilan pelaksanaan setiap fungsi utama. Seluruh pengujian dilaksanakan setelah proses implementasi selesai sehingga hasil yang diperoleh menggambarkan performa sistem secara keseluruhan.

a. Pengujian Komunikasi Sistem

Pengujian komunikasi bertujuan memastikan bahwa pertukaran data antara

Web Dashboard, Firebase Realtime Database, ESP32, dan modul relay berlangsung dengan baik. Pengujian dilakukan dengan memberikan beberapa perintah melalui *Web Dashboard* kemudian mengamati apakah setiap komponen mampu menerima dan mengeksekusi perintah tersebut.

Tabel 3.1 Pengujian Komunikasi dengan Blackbox

Komponen	Status
Web Dashboard – Firebase	Berhasil
Firebase – ESP32	Berhasil
ESP32 – Relay	Berhasil
Sistem Keseluruhan	Berhasil

Tabel 3.1 menunjukkan bahwa seluruh jalur komunikasi berhasil beroperasi dengan baik. Komunikasi antara *Web Dashboard* dan Firebase berjalan tanpa kendala, sinkronisasi data dari Firebase menuju ESP32 berhasil dilakukan secara *real-time*, sedangkan ESP32 mampu mengendalikan modul relay sesuai perintah yang diterima. Berdasarkan hasil tersebut dapat disimpulkan bahwa integrasi seluruh komponen sistem berhasil diimplementasikan tanpa ditemukan kegagalan komunikasi selama proses pengujian.

b. Pengujian Mode Data Encryption

Pengujian mode *Data Encryption* dilakukan sebanyak sepuluh kali percobaan untuk mengetahui waktu respons sistem ketika menjalankan proses enkripsi data. Hasil pengujian ditunjukkan menggunakan objek uji coba berukuran 10 MB pada Tabel 3.2

Tabel 3.2 Pengujian Delay Mode Enkripsi

Percobaan	Delay (detik)
1	3,502
2	2,093
3	1,796
4	2,038
5	2,029
6	1,988
7	2,021
8	2,044
9	2,036
10	1,951

Berdasarkan hasil pengujian terhadap file berukuran 10 MB menggunakan algoritma enkripsi Fernet atau menggunakan AES-128 dalam mode CBC dengan HMAC-SHA256 dan SHA-256 digunakan untuk membangkitkan kunci enkripsi, diperoleh total waktu eksekusi sebesar 21,498 detik dengan rata-rata waktu respons sebesar 2,149 detik. Hasil tersebut menunjukkan bahwa sistem mampu menjalankan proses enkripsi secara cepat dan stabil, sehingga pengguna dapat segera mengamankan data ketika terjadi ancaman terhadap perangkat.

c. Pengujian Mode Device Isolation

Pengujian *Device Isolation* dilakukan sebanyak sepuluh kali percobaan dengan cara mengaktifkan mode isolasi melalui *Web Dashboard*. ESP32 menerima perubahan status dari Firebase kemudian mengendalikan modul relay untuk memutus suplai daya perangkat target.

Tabel 3.3 Pengujian Delay Mode Isolasi Perangkat

Percobaan	Delay (detik)
1	4,624
2	4,579
3	4,748
4	4,691
5	4,619
6	4,651
7	4,703
8	4,586
9	4,632
10	4,673

Hasil pengujian pada Tabel 3.3 menunjukkan total waktu eksekusi sebesar 46,506 detik dengan rata-rata waktu respons sebesar 4,650 detik. Nilai tersebut sedikit lebih besar dibandingkan mode enkripsi karena proses isolasi melibatkan perpindahan kontak mekanis pada modul relay sebelum arus listrik berhasil diputus.

d. Pengujian Waktu Respons Sistem

Perbandingan hasil pengujian menunjukkan bahwa rata-rata waktu respons mode enkripsi adalah 2,149 detik, sedangkan mode isolasi perangkat sebesar 4,650 detik. Selisih waktu sebesar 2,501 detik menunjukkan bahwa tambahan waktu pada mode isolasi berasal dari proses aktivasi relay sebagai aktuator mekanis.

Meskipun demikian, kedua mode masih memiliki waktu respons di bawah 5 detik sehingga sistem dapat dikategorikan mampu memberikan respons yang cepat dalam menjalankan fungsi *Emergency Control* secara *real-time*.

e. Pengujian Tingkat Keberhasilan Sistem

Pengujian reliabilitas dilakukan dengan memberikan tiga puluh kali perintah secara berturut-turut untuk masing-masing mode pengamanan. Hasil pengujian disajikan pada Tabel 3.4.

Tabel 3.4 Jumlah Keberhasilan Uji

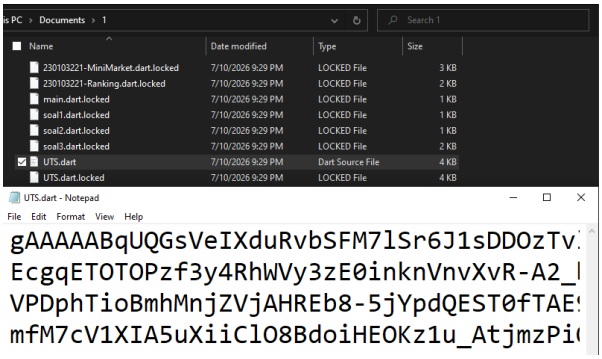
Mode	Jumlah Uji	Berhasil	Gagal	Success Rate
Enkripsi	30	30	0	100%
Isolasi	30	30	0	100%

Berdasarkan hasil pengujian diperoleh bahwa seluruh percobaan berhasil dijalankan tanpa mengalami kegagalan sehingga tingkat keberhasilan sistem mencapai 100% baik pada mode *Data Encryption* maupun *Device Isolation*. Hasil tersebut menunjukkan bahwa mekanisme komunikasi antara *Web Dashboard*, Firebase Realtime Database, ESP32, dan modul relay berjalan secara konsisten selama proses pengujian.

f. Pengujian Penanganan Gangguan Perangkat dan Data

Pengujian ini dilakukan untuk mengevaluasi kemampuan sistem dalam merespons kondisi gangguan yang disimulasikan pada perangkat maupun data. Simulasi dilakukan dengan memberikan beberapa skenario yang mewakili kondisi darurat sehingga dapat diketahui apakah mekanisme *Emergency Control* mampu menjalankan fungsi pengamanan sesuai dengan tujuan penelitian.

Pada pengujian gangguan data, skenario yang digunakan adalah adanya indikasi risiko kebocoran data pada perangkat. Pengguna kemudian mengaktifkan mode *Data Encryption* melalui *Web Dashboard*. Sistem mengirimkan perintah ke Firebase Realtime Database, selanjutnya layanan enkripsi melakukan proses enkripsi terhadap folder yang telah ditentukan. Hasil pengujian menunjukkan bahwa data berhasil diubah menjadi bentuk terenkripsi sehingga tidak dapat diakses sebelum dilakukan proses dekripsi.



Gambar 3.6 Tampilan File Ketika dibuka Secara Paksa

Gambar 3.6 menunjukkan bahwa file dapat diakses, akan tetapi isinya tidak akan sama atau rusak jika tanpa akses dekripsi secara sah.

Pada pengujian gangguan perangkat, disimulasikan kondisi perangkat berada pada situasi yang memerlukan tindakan pengamanan segera, seperti akses berlebih. Pengguna mengaktifkan mode *Device Isolation* melalui *Web Dashboard*, kemudian ESP32 menerima perintah dari Firebase Realtime Database dan mengaktifkan modul relay untuk memutus suplai daya perangkat. Hasil

pengujian menunjukkan bahwa relay berhasil memutus aliran listrik sehingga perangkat tidak dapat digunakan selama mode isolasi masih aktif.

Hasil keseluruhan pengujian disajikan pada Tabel 3.5.

Tabel 3.5 Hasil Pengujian Gangguan

Skenario	Kondisi	Aksi Sistem	Hasil
Normal	Tidak ada ancaman	Standby	Berhasil
Gangguan Data	Risiko pencurian data saat perangkat dipinjam	Encrypt Folder	Berhasil
Gangguan Perangkat	Terdeteksi penggunaan berlebihan atau tidak wajar	Relay memutus daya	Berhasil
Gangguan selesai	Perintah dinonaktifkan	Sistem kembali normal	Berhasil

3.3 Pembahasan

Hasil implementasi menunjukkan bahwa sistem kendali darurat berbasis *Internet of Things* (IoT) berhasil mengintegrasikan Web Dashboard, Firebase Realtime Database, ESP32, layanan *Data Encryption*, dan modul relay ke dalam satu platform yang mampu berkomunikasi secara *real-time*. Berdasarkan hasil pengujian komunikasi menggunakan metode *Black Box Testing*, seluruh fungsi utama sistem berjalan sesuai dengan rancangan tanpa ditemukan kegagalan komunikasi antar komponen. Hasil tersebut menunjukkan bahwa Firebase Realtime Database mampu berperan sebagai media sinkronisasi data yang stabil sehingga setiap perubahan status yang dikirimkan melalui Web Dashboard dapat diterima dan dieksekusi oleh ESP32 secara tepat. Temuan ini sejalan dengan penelitian Ashraf et al. [8] yang memanfaatkan teknologi IoT berbasis cloud untuk komunikasi data secara *real-time*.

Pengujian waktu respons menunjukkan bahwa mode *Data Encryption* memiliki rata-rata waktu eksekusi sebesar 2,149 detik, sedangkan mode *Device Isolation* memerlukan rata-rata waktu 4,650 detik. Perbedaan waktu tersebut disebabkan oleh karakteristik proses yang berbeda, di mana mode *Data Encryption* hanya melibatkan proses digital pada perangkat lunak, sedangkan mode *Device Isolation* memerlukan aktivasi modul relay sebagai aktuator elektromekanis. Meskipun demikian, kedua mode masih memiliki waktu respons kurang dari lima detik sehingga sistem mampu memberikan tindakan pengamanan secara cepat ketika pengguna mengaktifkan fitur *Emergency Control*. Hasil tersebut menunjukkan bahwa integrasi ESP32, Firebase Realtime Database, dan jaringan internet mampu mendukung komunikasi dengan latensi yang rendah sesuai karakteristik sistem IoT modern [3], [9].

Selain memiliki waktu respons yang cepat, sistem juga menunjukkan tingkat reliabilitas yang tinggi dengan keberhasilan eksekusi mencapai 100% pada seluruh percobaan untuk fitur *Data Encryption* maupun *Device Isolation*. Keberhasilan tersebut menunjukkan bahwa mekanisme sinkronisasi data antara Web Dashboard, Firebase Realtime Database, dan ESP32 berlangsung secara konsisten selama proses pengujian. Pengujian skenario gangguan juga membuktikan bahwa sistem mampu melakukan mitigasi terhadap dua jenis ancaman, yaitu gangguan pada data melalui proses *Data Encryption* dan gangguan pada perangkat melalui mekanisme *Device Isolation* menggunakan modul relay. Dengan demikian, sistem tidak hanya berfungsi sebagai media pemantauan

(*monitoring*), tetapi juga mampu melakukan tindakan pengamanan secara langsung terhadap perangkat maupun data.

Dibandingkan penelitian terdahulu yang umumnya hanya mengimplementasikan sebagian fungsi, seperti *remote monitoring*, komunikasi IoT, atau keamanan data [1], [2], [10], penelitian ini menawarkan integrasi beberapa mekanisme keamanan dalam satu platform, yaitu *remote control*, *Data Encryption*, *Device Isolation*, dan komunikasi *real-time* menggunakan Firebase Realtime Database. Integrasi tersebut menjadi kontribusi utama penelitian karena memungkinkan pengguna melakukan tindakan mitigasi secara cepat ketika terjadi ancaman terhadap perangkat maupun data, sehingga sistem memiliki kemampuan yang lebih komprehensif dibandingkan pendekatan yang hanya berfokus pada pemantauan atau perlindungan data.

.

IV. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem kendali darurat berbasis *Internet of Things* (IoT) yang mengintegrasikan Web Dashboard, Firebase Realtime Database, ESP32, layanan *Data Encryption*, dan modul relay dalam satu platform. Integrasi tersebut memungkinkan pengguna melakukan pengamanan perangkat dan data secara jarak jauh melalui mekanisme *Data Encryption* dan *Device Isolation* yang dapat dijalankan secara *real-time*. Berdasarkan hasil pengujian menggunakan metode *Black Box Testing*, pengujian waktu respons, pengujian tingkat keberhasilan, serta skenario gangguan perangkat dan data, seluruh fungsi sistem berjalan sesuai dengan rancangan dan mampu memberikan respons yang cepat serta konsisten dalam mengeksekusi setiap perintah.

Kontribusi utama penelitian ini terletak pada integrasi mekanisme pengamanan digital dan fisik dalam satu sistem sehingga pengguna tidak hanya memperoleh informasi kondisi perangkat, tetapi juga dapat melakukan tindakan mitigasi secara langsung ketika terjadi ancaman keamanan. Untuk pengembangan selanjutnya, sistem dapat ditingkatkan dengan penambahan mekanisme autentikasi yang lebih kuat, notifikasi otomatis kepada pengguna, serta pengujian pada lingkungan jaringan yang lebih kompleks agar implementasinya semakin optimal.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah memberikan dukungan dan kontribusi dalam pelaksanaan penelitian ini, khususnya kepada dosen pembimbing, Program Studi Sistem Kendali, Fakultas Ilmu Komputer, serta rekan-rekan yang telah membantu dalam proses perancangan, implementasi, dan pengujian sistem. Penulis juga menyampaikan apresiasi kepada institusi yang telah menyediakan fasilitas penelitian sehingga penelitian ini dapat diselesaikan dengan baik. Semoga hasil penelitian ini dapat memberikan manfaat bagi pengembangan teknologi Internet of Things, khususnya pada bidang keamanan perangkat dan sistem kendali darurat berbasis jaringan.

REFERENSI

[1] R. K. Kodali, V. Jain, S. Bose, and L. Boppana, "IoT Based Smart Security and Home Automation System," in *Proc. International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2018, pp. 1286–1289, doi: 10.1109/CCAA.2018.8777672.

[2] R. Damaševičius, N. Bacanin, and S. Misra, "From Sensors to Safety: Internet of Emergency Services (IoES) for Emergency Response and Disaster Management," *Journal of Sensor and Actuator Networks*, vol. 12, no. 3, p. 41, 2023, doi: 10.3390/jsan12030041.

[3] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019, doi: 10.1016/j.icte.2017.12.005.

[4] A. Mosenia and N. K. Jha, "A Comprehensive Study of Security of Internet-of-Things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, 2017, doi: 10.1109/TETC.2016.2606384.

[5] M. A. Ferrag, L. Maglaras, A. Ahmim, and M. Derdour, "Privacy-Preserving Schemes for Ad Hoc Social Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 3015–3045, 2017, doi: 10.1109/COMST.2017.2749519.

[6] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges," *IEEE Access*, vol. 8, pp. 32031–32053, 2020, doi: 10.1109/ACCESS.2020.2973178.

[7] M. H. ur Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Trust in Blockchain Cryptocurrency Ecosystem," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1196–1212, 2020, doi: 10.1109/TEM.2019.2948861.

[8] S. Ashraf, S. P. Khattak, and M. T. Iqbal, "Design and Implementation of an Open-Source and Internet-of-Things-Based Health Monitoring System," *Journal of Low Power Electronics and Applications*, vol. 13, no. 4, p. 57, 2023, doi: 10.3390/jlpea13040057.

[9] A. A. A. Ari, B. K. Yenke, C. Titouna, A. Gueroui, and Z. Aliouat, "Transmitting IoT Data over Low Power Wide Area Networks: Challenges and Approaches," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2052–2069, 2019, doi: 10.1109/JIOT.2018.2868036.

[10] M. A. Khan and K. Salah, "IoT Security: Review, Blockchain Solutions, and Open Challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018, doi: 10.1016/j.future.2017.11.022.

[11] A. Ahad, M. Tahir, and K.-L. A. Yau, "5G-Based Smart Healthcare Network: Architecture, Taxonomy, Challenges and Future Research Directions," *IEEE Access*, vol. 7, pp. 100747–100762, 2019, doi: 10.1109/ACCESS.2019.2930628.

[12] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Machine Learning Techniques for Cyber Security in the Internet of Things: A Survey," *Journal of Network and Computer Applications*, vol. 149, p. 102463, 2020, doi: 10.1016/j.jnca.2019.102463.

[13] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The Industrial Internet of Things (IIoT): An Analysis Framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018, doi: 10.1016/j.compind.2018.04.015.