

Rancang Sistem Kotak Penerima Paket (KOMPAS) Berbasis IoT ESP32-CAM

Sujono*¹⁾, Joni Alwi²⁾

1. Informatika, Teknologi Informasi, Universitas KH. A. Wahab Hasbullah, Indonesia
2. Informatika, Teknologi Informasi, Universitas KH. A. Wahab Hasbullah, Indonesia

ARTICLE INFO

Kata Kunci: *Internet of Things; ESP32-CAM; kotak penerima paket; Telegram Bot; autentikasi*

Keywords: *Internet of Things; ESP32-CAM; package receiver box; Telegram Bot; authentication*

Article history:

Received: 25 Agustus 2026

Revised: 5 September 2026

Accepted: 16 September 2026

Available online: 1 November 2026

DOI:

[10.48144/suryainformatika.v16i2.2553](https://doi.org/10.48144/suryainformatika.v16i2.2553)

* Corresponding author.

Joni Alwi

E-mail address:

[jonalwi01@gmail.com](mailto:jonialwi01@gmail.com)

ABSTRAK

Proses penerimaan paket menjadi permasalahan ketika penerima tidak berada di lokasi saat kurir datang sehingga paket berpotensi ditiptkan kepada pihak lain atau diletakkan pada tempat yang kurang aman. Penelitian ini bertujuan merancang dan mengevaluasi Kotak Penerima Paket (KOMPAS) berbasis *Internet of Things* menggunakan ESP32 dan ESP32-CAM. ESP32 digunakan sebagai pusat kendali, sedangkan ESP32-CAM digunakan untuk mendokumentasikan kurir dan mengirimkan gambar melalui Telegram Bot. Sistem dilengkapi fingerprint dan keypad 4×4 sebagai mekanisme autentikasi serta servo dan solenoid door lock sebagai mekanisme pengendalian pintu. Evaluasi dilakukan melalui pengujian kuantitatif terhadap latency pengiriman, kekuatan sinyal Wi-Fi (RSSI), keberhasilan penerimaan foto, waktu respons fingerprint, dan respons keypad terhadap beberapa kondisi input. Pengujian latency sebanyak 10 percobaan menghasilkan waktu respons 15–20 detik dengan rata-rata 17,1 detik. Pengujian RSSI menghasilkan nilai -29 sampai -55 dBm dengan rata-rata -36,9 dBm dan seluruh foto berhasil diterima pada 10 percobaan. Pengujian fingerprint menghasilkan keputusan yang sesuai pada seluruh 10 percobaan, sedangkan pengujian keypad menunjukkan PIN benar diterima, PIN salah ditolak, serta sistem tidak mengalami restart pada penekanan tombol secara cepat. Hasil penelitian menunjukkan bahwa prototype KOMPAS mampu menjalankan fungsi dokumentasi, komunikasi, autentikasi, dan pengendalian akses pada skenario pengujian yang dilakukan.

ABSTRACT

The package receiving process becomes a problem when the recipient is not present at the location when the courier arrives, potentially causing the package to be entrusted to other people or placed in an unsecured location. This study aims to design and evaluate a Package Receiver Box (KOMPAS) based on the Internet of Things using ESP32 and ESP32-CAM. The ESP32 functions as the main controller, while ESP32-CAM is used to document the courier and transmit images through Telegram Bot. The system is equipped with fingerprint and 4×4 keypad authentication, as well as a servo and solenoid door lock for door control. Evaluation was conducted through quantitative testing of transmission latency, Wi-Fi signal strength (RSSI), image delivery success, fingerprint response time, and keypad response under several input conditions. Ten latency trials produced response times ranging from 15 to 20 seconds with an average of 17.1 seconds. RSSI testing produced values ranging from -29 to -55 dBm with an average of -36.9 dBm, and all ten images were successfully received. Fingerprint testing produced correct access decisions in all ten trials, while keypad testing showed that valid PINs were accepted, invalid PINs were rejected, and rapid button presses did not cause the system to restart. The results indicate that the KOMPAS prototype can perform documentation, communication, authentication, and access-control functions under the tested scenarios.

1. PENDAHULUAN

Perkembangan layanan perdagangan elektronik meningkatkan kebutuhan terhadap mekanisme penerimaan paket yang tetap dapat digunakan ketika penerima tidak berada di rumah. Ketidakhadiran penerima pada saat kurir datang dapat menyebabkan paket dititipkan kepada pihak lain, diletakkan pada lokasi yang tidak terpantau, atau memerlukan pengiriman ulang. Permasalahan tersebut tidak hanya berkaitan dengan penyimpanan paket, tetapi juga berkaitan dengan kemampuan penerima untuk mengetahui kedatangan kurir, memperoleh dokumentasi proses pengiriman, dan mengendalikan akses terhadap tempat penyimpanan.

Penerapan *Internet of Things* (IoT) telah digunakan untuk mengembangkan sistem penerimaan paket yang dapat berkomunikasi dengan pengguna melalui jaringan internet. Putri et al. mengembangkan Smart Packages Box menggunakan Wemos D1 R32, barcode scanner GM66, solenoid door lock, ESP32-CAM, dan Telegram Bot [1]. Febriyanti et al. mengembangkan box penerima paket menggunakan ESP32, Telegram, servo, sensor inframerah, dan sensor ultrasonik [2]. Jazilah et al. mengembangkan box penerima paket yang menyimpan nomor resi dari Telegram pada ESP32, memverifikasi barcode dengan scanner GM65, kemudian membuka door lock dan memberikan notifikasi ketika paket telah masuk [3]. Handayani et al. mengembangkan package receiver berbasis IoT dengan ESP32-CAM dan mekanisme penguncian [4]. Temuan tersebut menunjukkan bahwa dokumentasi, notifikasi, dan kontrol akses telah menjadi fungsi utama pada penelitian box penerima paket sebelumnya.

Penelitian yang paling dekat dengan KOMPAS adalah Riezqi et al. yang mengembangkan Smart Box menggunakan Arduino Nano dan ESP32-CAM, dengan barcode sebagai mekanisme pembukaan serta Telegram untuk mengirim foto dan notifikasi [5]. Sistem tersebut berhasil membaca barcode dan mengirim dokumentasi, tetapi evaluasi yang dilaporkan berfokus pada keberhasilan fungsi. Pada sisi autentikasi, Syahputra dan Ananda menggunakan ESP32 dan fingerprint untuk menggantikan kunci konvensional pada smart locker [6]. Permana et al. mengembangkan smart door lock dengan fingerprint dan keypad, Firebase, serta aplikasi Android dan melaporkan keberhasilan black-box 100% pada 20 percobaan [7]. Kurniasyah dan Rakhmawati mengintegrasikan RFID, fingerprint, keypad, BLE, dan MQTT; pengujian mereka melaporkan throughput MQTT QoS 0 sebesar 5,052 kbps, delay 175 ms, serta jitter minimum 19 ms pada QoS 2 [8].

Penelitian lain memperkuat kebutuhan evaluasi komunikasi dan autentikasi. Suandi et al. menerapkan ESP32-CAM dan notifikasi Telegram pada sistem pemantauan [9], sedangkan Munirman et al. secara khusus menganalisis latency dan keandalan notifikasi visual ESP32-CAM melalui Telegram Bot API [10]. Wahyudi dan Edidas menggunakan ESP32-CAM,

beberapa sensor keamanan, dan Telegram untuk notifikasi rumah [11]. Cakrayuda mengembangkan SiCEMOT berbasis ESP32-CAM, PIR, dan Telegram untuk mendeteksi gerakan, menangkap gambar, dan mengirimkan peringatan melalui Telegram [12]. Rahman et al. mengembangkan pengunci pintu berbasis fingerprint dan smartphone serta melaporkan solenoid membutuhkan sekitar 3 detik untuk membuka pintu [13]. Sugiarto dan Herianto menggabungkan fingerprint, pengenalan suara, ESP32, dan Telegram pada smart door lock [14]. Ananda et al. mengembangkan smart locker berbasis ESP32, QR-Code, dan aplikasi Android dengan functional suitability dan compatibility 100%, usability 78,98%, serta rata-rata waktu pembukaan 0,874 detik [15].

Berdasarkan penelitian terdahulu, celah penelitian terletak pada belum ditemukannya evaluasi terpadu yang menggabungkan dokumentasi kurir, komunikasi Telegram, autentikasi *fingerprint* dan *keypad*, serta pengukuran performa komunikasi dalam satu *prototype* penerima paket. Riezqi et al. menitikberatkan *smart box* pada barcode, ESP32-CAM, dan Telegram [5]. Permana et al. memusatkan penelitian pada *smart door lock* dengan *fingerprint* dan *keypad* [7]. Kurniasyah dan Rakhmawati mengevaluasi komunikasi BLE/MQTT pada *smart door lock* [8]. Sementara itu, Wahyudi dan Edidas [11], Cakrayuda [12], Rahman et al. [13], Sugiarto dan Herianto [14], serta Ananda et al. [15] membahas keamanan rumah, pintu, atau locker dengan konfigurasi dan objek pengujian yang berbeda. Celah tersebut menjadi dasar pengembangan KOMPAS sebagai sistem penerimaan paket yang menyatukan dokumentasi visual, komunikasi jarak jauh, autentikasi lokal, dan evaluasi performa *end-to-end*.

Penelitian ini merumuskan tiga pertanyaan: (1) bagaimana kinerja komunikasi dokumentasi KOMPAS berdasarkan latency dan kondisi RSSI; (2) apakah foto dokumentasi dapat diterima melalui Telegram pada kondisi jaringan yang diamati; dan (3) bagaimana respons autentikasi fingerprint dan keypad terhadap input yang sesuai maupun tidak sesuai. Hasil pengujian digunakan untuk menjawab pertanyaan tersebut pada ruang lingkup *prototype* dan tidak dimaksudkan sebagai generalisasi statistik untuk seluruh kondisi penggunaan.

Kontribusi penelitian ini adalah integrasi dan evaluasi empiris pada satu alur penerimaan paket. KOMPAS menghubungkan dokumentasi kurir melalui ESP32-CAM dan Telegram dengan dua mekanisme autentikasi lokal, yaitu *fingerprint* dan *keypad*, kemudian mengukur *latency end-to-end*, RSSI, keberhasilan penerimaan foto, dan waktu respons autentikasi. Berbeda dengan Riezqi et al. yang menggunakan barcode pada *smart box* [5], Permana et al. yang berfokus pada *smart door lock* berbasis *fingerprint* dan *keypad* [7], serta Kurniasyah dan Rakhmawati yang mengevaluasi BLE/MQTT [8], KOMPAS menyajikan data empiris komunikasi visual dan autentikasi dalam konteks khusus penerimaan paket. Dengan cakupan tersebut, kontribusi KOMPAS

terletak pada integrasi alur dan karakterisasi performa sistem, bukan pada kebaruan komponen secara terpisah.

Tujuan penelitian adalah merancang dan mengevaluasi prototype KOMPAS berbasis IoT ESP32-CAM serta menyajikan hasil pengujian komunikasi, dokumentasi, dan autentikasi secara terukur sebagai dasar pengembangan berikutnya.

2. METODE PENELITIAN

2.1 Rancangan Penelitian

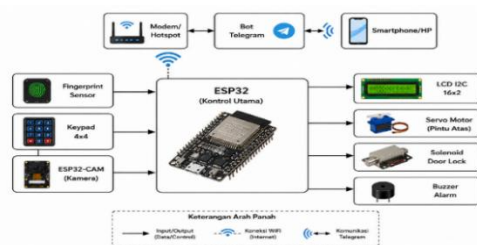
Penelitian menggunakan pendekatan eksperimen terhadap prototype KOMPAS. Tahapan penelitian meliputi perancangan sistem, implementasi perangkat keras dan perangkat lunak, integrasi komponen, kemudian pengujian dengan parameter kuantitatif. Pendekatan eksperimen digunakan agar hasil pengujian tidak hanya berupa status berhasil atau gagal, tetapi menghasilkan data berupa waktu respons, nilai RSSI, keberhasilan pengiriman citra, dan keputusan autentikasi.

Pengujian dibagi menjadi empat kelompok utama, yaitu pengujian latency pengiriman dokumentasi, pengujian RSSI dan keberhasilan pengiriman foto, pengujian fingerprint, serta pengujian keypad.

2.2 Arsitektur Sistem

KOMPAS menggunakan ESP32 sebagai mikrokontroler utama. ESP32 mengelola input dari keypad dan sensor fingerprint serta mengendalikan LCD, buzzer, LED, servo, dan solenoid door lock. ESP32-CAM digunakan untuk mengambil gambar kurir dan mendukung pengiriman dokumentasi melalui jaringan Wi-Fi dan Telegram Bot.

Pada proses penerimaan paket, kurir berada di depan kamera untuk didokumentasikan. Setelah proses pemicu diberikan, ESP32-CAM mengambil gambar dan mengirimkan dokumentasi kepada pengguna melalui Telegram. Pengguna dapat memberikan instruksi sesuai mekanisme sistem. Setelah paket dimasukkan, pintu dikembalikan ke kondisi terkunci. Pada proses pengambilan paket, autentikasi dilakukan menggunakan fingerprint atau keypad sebelum sistem memberikan akses.



Gambar 1. Diagram blok/arsitektur KOMPAS

Rancangan ini sesuai dengan implementasi pada tugas akhir, yaitu ESP32 sebagai pusat kendali, ESP32-CAM sebagai kamera, serta fingerprint, keypad, LCD, buzzer, servo, dan solenoid sebagai bagian dari sistem pengendalian dan keamanan.

2.3 Perangkat Keras dan Perangkat Lunak

Perangkat keras utama yang digunakan terdiri atas ESP32, ESP32-CAM, sensor fingerprint, keypad 4×4, LCD 16×2 I2C, servo, solenoid door lock, buzzer, LED, serta komponen pendukung catu daya dan koneksi.

ESP32 digunakan sebagai pengendali utama sistem. ESP32-CAM digunakan untuk pengambilan gambar. Sensor fingerprint dan keypad digunakan sebagai metode autentikasi. Servo digunakan untuk mekanisme pintu bagian atas, sedangkan solenoid digunakan sebagai mekanisme penguncian. LCD, LED, dan buzzer digunakan sebagai indikator status sistem.

Pada sisi perangkat lunak, program dikembangkan menggunakan Arduino IDE dan komunikasi dengan pengguna dilakukan melalui Telegram Bot. Implementasi tugas akhir menunjukkan bahwa ESP32-CAM ditempatkan mengarah ke area kurir sehingga dokumentasi dapat diambil sebelum paket dimasukkan ke dalam kotak.

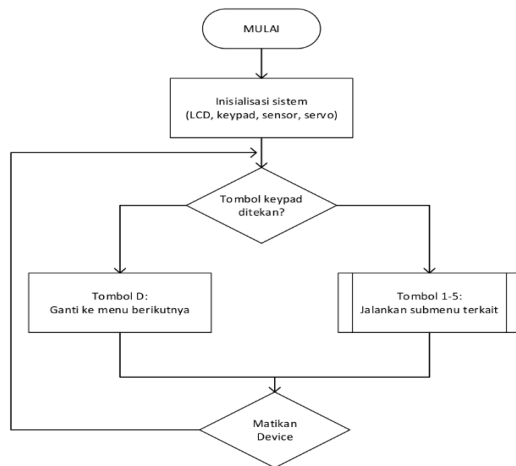
2.4 Variabel dan Parameter Pengujian

Parameter yang digunakan dalam penelitian adalah:

Tabel 1. Variabel dan Parameter Pengujian

Parameter	Satuan	Tujuan
Latency pengiriman	Numerik (detik)	Mengukur waktu <i>end-to-end</i> pengiriman dokumentasi
RSSI	Numerik (dBm)	Menggambarkan kekuatan sinyal Wi-Fi
Penerimaan foto	Kategorikal (Ya/Tidak)	Mencatat keberhasilan pengiriman citra
Waktu respons fingerprint	Numerik (detik)	Mengukur waktu respons autentikasi
Keputusan fingerprint	Kategorikal (Diterima/Ditolak)	Mengevaluasi kesesuaian keputusan akses
Waktu respons keypad	Numerik (detik)	Mengukur waktu respons terhadap input
Respons sistem	Kategorikal	Mencatat respons pada kondisi input yang diuji

2.5 Skenario Pengujian



Gambar 2. Flowchart system KOMPAS

Pengujian latency dilakukan sebanyak 10 kali dan waktu dari proses pemicu sampai dokumentasi diterima dicatat dalam satuan detik.

Pengujian RSSI dilakukan sebanyak 10 kali dengan mencatat nilai RSSI dan keberhasilan penerimaan foto. RSSI digunakan sebagai karakterisasi kondisi jaringan pada saat pengiriman, bukan sebagai variabel terkontrol untuk menguji hubungan sebab-akibat terhadap latency.

Pengujian fingerprint dilakukan dengan lima percobaan menggunakan fingerprint terdaftar dan lima percobaan menggunakan fingerprint tidak terdaftar.

Pengujian *keypad* dilakukan terhadap PIN benar, PIN salah, input tidak lengkap, dan penekanan tombol secara cepat. Selain waktu respons dan keputusan akses, diamati pula respons sistem pada dua percobaan penekanan tombol cepat. Dua percobaan tersebut diperlakukan sebagai uji fungsional kondisi input, bukan *stress test* atau pengujian *reliability*.

2.6 Pengujian dan Analisis Data

Pengujian sistem dilakukan menggunakan pendekatan eksperimen kuantitatif yang dilengkapi pengujian fungsional. Pengujian dilakukan dengan memberikan kondisi atau input tertentu kepada prototype kemudian mencatat respons sistem dalam bentuk data terukur. Pengujian dibagi menjadi empat kelompok, yaitu pengujian latency pengiriman dokumentasi, pengujian RSSI jaringan Wi-Fi dan keberhasilan penerimaan foto, pengujian autentikasi fingerprint, serta pengujian keypad.

Pengujian *latency* dilakukan sebanyak 10 percobaan menggunakan stopwatch pada telepon seluler. Waktu mulai dicatat ketika proses dokumentasi kurir dipicu dan waktu akhir dicatat ketika foto diterima pada aplikasi Telegram pengguna. Metode pengukuran yang sama digunakan pada seluruh percobaan. Pengujian RSSI dilakukan sebanyak 10 percobaan dengan mencatat kekuatan sinyal Wi-Fi dalam dBm dan status penerimaan foto. Pengujian *fingerprint* dilakukan terhadap *fingerprint* terdaftar dan tidak terdaftar dengan

mencatat waktu respons serta keputusan akses. Pengujian *keypad* dilakukan pada PIN benar, PIN salah, input tidak lengkap, dan penekanan tombol secara cepat; dua percobaan tombol cepat hanya digunakan untuk mengamati respons fungsional sistem.

Data dianalisis secara deskriptif menggunakan nilai rata-rata, minimum, maksimum, dan persentase keberhasilan. Nilai rata-rata dihitung menggunakan persamaan:

$$\bar{x} = \sum x / n$$

dengan $\bar{x}$ merupakan nilai rata-rata, $\sum x$ merupakan jumlah seluruh nilai pengamatan, dan n merupakan jumlah percobaan.

Persentase keberhasilan dihitung menggunakan:

$$P = (\text{jumlah percobaan berhasil} / \text{jumlah seluruh percobaan}) \times 100\%$$

Analisis tersebut digunakan untuk menggambarkan kinerja prototype berdasarkan kondisi pengujian yang dilakukan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Sistem

Implementasi menghasilkan prototype KOMPAS yang terdiri atas subsistem pengendalian, dokumentasi, komunikasi, autentikasi, dan penguncian. Berdasarkan pengujian fungsional pada tugas akhir, ESP32 dapat menjalankan fungsi pengendalian, ESP32-CAM dapat mengambil gambar, LCD menampilkan informasi, keypad menerima input, fingerprint melakukan autentikasi, servo dan solenoid menjalankan mekanisme pintu, serta buzzer memberikan indikator kondisi sistem.

Pada sisi komunikasi, sistem dapat merespons perintah Telegram dan menjalankan proses dokumentasi kurir. Implementasi tugas akhir menunjukkan bahwa ESP32-CAM mengambil gambar kurir dan menggunakan koneksi Wi-Fi untuk mengirimkan gambar serta informasi kepada pengguna melalui Telegram Bot.

Hasil tersebut menjadi dasar dilakukannya pengujian kuantitatif yang lebih mendalam untuk mengatasi keterbatasan evaluasi sebelumnya yang hanya menggunakan kategori “berhasil”.



Gambar 3. Implementasi alat KOMPAS pada lokasi penerimaan paket

3.2 Pengujian Latency Pengiriman Dokumentasi

Pengujian latency dilakukan sebanyak 10 kali menggunakan stopwatch pada telepon seluler untuk mengetahui waktu end-to-end sejak proses dokumentasi

kurir dipicu hingga foto diterima pada Telegram. Hasil pengujian ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pengujian Latency

Percobaan	Latency (detik)
1	20
2	15
3	15
4	20
5	15
6	18
7	20
8	15
9	18
10	15
Rata-rata	17,1
Minimum	15
Maksimum	20

Berdasarkan Tabel 2, *latency* pengiriman berada pada rentang 15–20 detik dengan rata-rata 17,1 detik. Nilai ini menunjukkan bahwa jalur dokumentasi KOMPAS masih relatif lambat. Munirman et al. melaporkan *latency end-to-end* ESP32-CAM–Telegram sebesar 6,03–10,45 detik dengan rata-rata 8,09 detik [10]. Dengan definisi pengukuran yang sama-sama dimulai dari pemicu hingga foto tampil pada percakapan penerima, rata-rata KOMPAS sekitar 9,01 detik lebih tinggi. Perbedaan tersebut mengindikasikan adanya ruang optimasi pada *pipeline* dokumentasi KOMPAS.

Secara arsitektural, *latency end-to-end* KOMPAS mencakup beberapa tahap: pemicuan dokumentasi, *capture* dan pengodean citra oleh ESP32-CAM, pembentukan komunikasi aman melalui Wi-Fi, pengunggahan *payload* citra ke Telegram Bot API, pemrosesan server, dan penyampaian foto ke perangkat pengguna. Munirman et al. menunjukkan bahwa ukuran *payload* citra dan secure transport merupakan kontributor penting terhadap *latency* serta merekomendasikan penurunan resolusi dan penggunaan ulang sesi aman [10]. Karena penelitian KOMPAS belum mencatat durasi setiap tahap secara terpisah, penelitian ini tidak menetapkan satu penyebab tunggal. Namun, selisih rata-rata terhadap [10] memperkuat dugaan bahwa konfigurasi citra, proses *capture*, kualitas koneksi, dan pembentukan sesi Telegram perlu diuji secara terpisah pada optimasi berikutnya.

3.3 Pengujian RSSI dan Keberhasilan Penerimaan Foto

Pengujian RSSI dilakukan untuk menggambarkan kondisi kekuatan sinyal Wi-Fi selama proses pengiriman dokumentasi dan mencatat keberhasilan penerimaan foto. RSSI tidak diperlakukan sebagai variabel terkontrol untuk menguji pengaruh terhadap *latency*. Hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3. Hasil Pengujian RSSI dan Pengiriman Foto

Percobaan	RSSI (dBm)	Foto diterima
1	-29	Ya
2	-33	Ya
3	-35	Ya
4	-38	Ya
5	-37	Ya
6	-35	Ya
7	-34	Ya
8	-30	Ya
9	-43	Ya
10	-55	Ya
Rata-rata	-36,9	10/10(100%)

Nilai RSSI yang diperoleh berada pada rentang -29 sampai -55 dBm dengan rata-rata -36,9 dBm. Seluruh 10 percobaan menghasilkan foto yang berhasil diterima sehingga tingkat keberhasilan pengiriman pada skenario pengujian mencapai 100%.

Pada nilai RSSI terendah yang diperoleh, yaitu -55 dBm, foto tetap berhasil diterima. Temuan ini hanya menunjukkan bahwa pada rentang kondisi jaringan yang diamati, sistem masih dapat menjalankan komunikasi dan mengirimkan dokumentasi; hasil tersebut tidak digunakan untuk menyimpulkan hubungan sebab-akibat antara RSSI dan *latency*.

Karena RSSI tidak dikontrol pada tingkat tertentu dan tidak dilakukan pengulangan pada setiap tingkat RSSI, pengujian ini bersifat karakterisasi kondisi jaringan. Pengujian terkontrol dengan variasi jarak, hambatan, dan beban jaringan diperlukan untuk menganalisis pengaruh RSSI terhadap *latency* secara lebih kuat.

3.4 Pengujian Fingerprint

Pengujian fingerprint dilakukan untuk mengetahui kemampuan sistem dalam membedakan pengguna dengan sidik jari yang telah terdaftar dan tidak terdaftar. Pengujian dilakukan sebanyak 10 kali, terdiri atas lima percobaan fingerprint terdaftar dan lima percobaan fingerprint tidak terdaftar.

Tabel 4. Hasil Pengujian Fingerprint

Percobaan	Kondisi Fingerprint	Waktu Respons (detik)	Akses
1	Terdaftar	5	Diterima
2	Terdaftar	8	Diterima
3	Terdaftar	4	Diterima
4	Terdaftar	3	Diterima
5	Terdaftar	5	Diterima
6	Tidak terdaftar	8	Ditolak
7	Tidak terdaftar	10	Ditolak
8	Tidak terdaftar	7	Ditolak

9	Tidak terdaftar	10	Ditolak
10	Tidak terdaftar	9	Ditolak

Fingerprint terdaftar menghasilkan waktu respons 3–8 detik dengan rata-rata 5 detik. Seluruh lima percobaan menghasilkan akses diterima. Sementara itu, fingerprint tidak terdaftar menghasilkan waktu respons 7–10 detik dengan rata-rata 8,8 detik dan seluruh lima percobaan menghasilkan akses ditolak.

Dengan demikian, keputusan autentikasi sesuai dengan kondisi fingerprint pada seluruh 10 percobaan sehingga tingkat keberhasilan keputusan autentikasi pada skenario ini mencapai 100%.

Hasil tersebut menunjukkan bahwa *fingerprint* berfungsi sebagai mekanisme autentikasi lokal pada skenario uji. Namun, keamanan biometrik tidak dapat dinilai hanya dari keputusan 10 percobaan ini. Data sidik jari diproses oleh modul *fingerprint*, sedangkan penelitian belum melakukan audit terhadap mekanisme penyimpanan template, enkripsi, *spoofing*, *False Acceptance Rate* (FAR), dan *False Rejection Rate* (FRR). Oleh karena itu, hasil pengujian ini dibatasi pada fungsi autentikasi, bukan pembuktian keamanan biometrik secara menyeluruh.

3.5 Pengujian Keypad

Pengujian keypad dilakukan untuk mengetahui respons sistem terhadap input normal dan beberapa kondisi input yang berpotensi menyebabkan gangguan pada sistem. Skenario meliputi PIN benar, PIN salah, input tidak lengkap, dan penekanan tombol secara cepat.

Tabel 5. Hasil Pengujian Keypad

Percobaan	Skenario Input	Waktu Respons (detik)	Respons Sistem	Akses
1	PIN benar	5	Tetap merespons	Di terima
2	PIN benar	6	Tetap merespons	Di terima
3	PIN benar	5	Tetap merespons	Di terima
4	Input tidak lengkap	3	Input diterima sistem, akses tidak diberikan	Di tolak
5	Input tidak lengkap	3	Input diterima sistem,	Di tolak

			akses tidak diberikan	
6	Tombol ditekan cepat	8	Tidak restart	Di tolak
7	PIN salah	10	Tidak membuk a pintu tanpa autentikasi	Di tolak
8	Tombol ditekan cepat	7	Tidak restart	Di tolak
9	PIN salah	10	Tidak membuk a pintu tanpa autentikasi	Di tolak
10	PIN salah	10	Tidak membuk a pintu tanpa autentikasi	Di tolak

Pada tiga pengujian menggunakan PIN benar, waktu respons berada pada rentang 5–6 detik dengan rata-rata 5,33 detik dan seluruh input menghasilkan akses diterima. Pada dua pengujian dengan input tidak lengkap, sistem tetap menerima input dari keypad dan memberikan respons dalam waktu 3 detik, tetapi tidak memberikan akses karena PIN belum memenuhi kondisi autentikasi yang dipersyaratkan.

Pada tiga pengujian PIN salah, waktu respons sebesar 10 detik dan seluruh akses ditolak. Pada dua pengujian tombol yang ditekan secara cepat, sistem tetap merespons, tidak membuka akses, dan tidak mengalami restart selama dua percobaan tersebut. Hasil ini hanya menunjukkan perilaku fungsional pada kondisi input yang diuji dan tidak diposisikan sebagai *stress test* maupun bukti *reliability* sistem.

Dengan demikian, dari 10 percobaan *keypad*, hanya input PIN yang benar yang menghasilkan akses diterima, sedangkan tujuh percobaan lainnya tidak memberikan akses. Dari sisi keamanan, pengujian ini belum mencakup *brute-force* berulang, pembatasan jumlah percobaan, *lockout* sementara, maupun *rate limiting*. Implementasi proteksi tersebut diperlukan agar *keypad* tidak hanya benar secara fungsional, tetapi juga lebih tahan terhadap percobaan PIN berulang.

3.6 Rekapitulasi Hasil Pengujian

Rekapitulasi digunakan untuk melihat hasil pengujian sistem berdasarkan seluruh parameter yang telah diuji.

Tabel 6. Rekapitulasi Hasil Pengujian

Parameter Pengujian	Jumlah Percobaan	Hasil Utama
Latency pengiriman	10	Rata-rata 17,1 detik
RSSI Wi-Fi	10	Rata-rata -36,9 dBm
Penerimaan foto	10	10/10 atau 100%
Fingerprint terdaftar	5	5/5 diterima
Fingerprint tidak terdaftar	5	5/5 ditolak
PIN benar	3	3/3 diterima
PIN salah	3	Tidak restart pada 2 percobaan; uji fungsional
Tombol ditekan cepat	2	Tidak hang pada 2 percobaan; akses tidak diberikan
Input tidak lengkap	2	Tidak hang

Berdasarkan hasil pengujian, prototype KOMPAS dapat menjalankan fungsi komunikasi, dokumentasi, dan autentikasi pada seluruh skenario yang diuji. Hasil kuantitatif menunjukkan bahwa performa komunikasi memiliki variasi latency, sedangkan pengiriman foto berhasil pada seluruh percobaan RSSI. Pada sisi autentikasi, fingerprint dan keypad menghasilkan keputusan sesuai dengan kondisi input yang diberikan.

Hasil tersebut memberikan evaluasi yang lebih terukur dibandingkan pengujian fungsional yang hanya menggunakan kategori berhasil atau gagal. Meskipun demikian, hasil penelitian dibatasi oleh jumlah percobaan dan kondisi pengujian sehingga belum dapat digeneralisasikan untuk seluruh kondisi lingkungan penggunaan.

3.7 Pembahasan dan Perbandingan dengan Penelitian Terdahulu

Riezqi et al. mengembangkan *Smart Box* berbasis Arduino Nano dan ESP32-CAM dengan barcode sebagai mekanisme akses serta Telegram untuk mengirim foto dan notifikasi [5]. KOMPAS memperluas pola tersebut dengan mengganti ketergantungan pada barcode menjadi autentikasi lokal *fingerprint* dan *keypad* serta menambahkan pengukuran *latency end-to-end* dan RSSI. Dengan demikian, nilai tambah KOMPAS terhadap [5] terletak pada integrasi

dua mekanisme autentikasi lokal dan karakterisasi performa komunikasi pada alur penerimaan paket.

Syahputra dan Ananda menggunakan ESP32 dan *fingerprint* untuk menggantikan kunci konvensional pada *smart locker* [6]. Permana et al. menggabungkan *fingerprint* dan *keypad* dengan Firebase serta aplikasi Android dan memperoleh keberhasilan *black-box* 100% pada 20 percobaan [7]. KOMPAS menerapkan *fingerprint* dan *keypad* pada konteks yang berbeda, yaitu penerimaan paket yang juga membutuhkan dokumentasi kurir dan komunikasi Telegram. Data KOMPAS menunjukkan keputusan *fingerprint* sesuai pada 10 percobaan dan PIN benar/salah memberikan keputusan sesuai pada skenario *keypad*. Perbandingan ini menunjukkan bahwa kontribusi KOMPAS bukan sekadar penggunaan sensor autentikasi, tetapi penggabungannya dengan proses dokumentasi dan verifikasi paket.

Kurniasyah dan Rakhmawati mengevaluasi *smart door lock* berbasis BLE/MQTT dan melaporkan throughput MQTT QoS 0 sebesar 5,052 kbps, delay 175 ms, serta jitter minimum 19 ms pada QoS 2 [8]. Pada jalur komunikasi visual yang lebih sepadan, Munirman et al. mengukur ESP32-CAM dan Telegram Bot API serta memperoleh *latency end-to-end* 6,03–10,45 detik dengan rata-rata 8,09 detik [10]. KOMPAS memperoleh rata-rata 17,1 detik, sekitar 9,01 detik lebih tinggi daripada [10]. Temuan ini merupakan kelemahan performa yang terukur sekaligus arah optimasi: *pipeline capture*, ukuran/resolusi citra, pembentukan *secure session*, dan proses *upload* Telegram perlu diprofilkan per tahap agar sumber bottleneck dapat diidentifikasi.

Wahyudi dan Edidas menerapkan ESP32-CAM dan Telegram pada sistem keamanan rumah [11], sedangkan Cakrayuda mengembangkan SiCEMOT menggunakan ESP32-CAM, PIR, dan Telegram [12]. Cakrayuda melaporkan akurasi deteksi hingga 95% pada pencahayaan normal, 90% pada cahaya redup, dan rata-rata waktu respons 2,7 detik [12]. Perbedaan objek pemicu dan definisi waktu respons membuat angka tersebut tidak identik dengan *latency* dokumentasi KOMPAS, tetapi hasilnya tetap menunjukkan bahwa respons sistem berbasis ESP32-CAM dapat dioptimalkan secara signifikan. KOMPAS menambahkan konteks penerimaan paket dan autentikasi lokal, sementara hasil *latency* 17,1 detik menunjukkan bahwa sisi komunikasi visual masih menjadi komponen yang paling membutuhkan peningkatan.

Rahman et al. mengembangkan pengunci pintu berbasis *fingerprint* dan *smartphone* serta melaporkan solenoid membutuhkan sekitar 3 detik untuk membuka pintu [13]. Sugiarto dan Herianto menggabungkan *fingerprint*, pengenalan suara, ESP32, dan Telegram pada *smart door lock* [14]. Ananda et al. mengembangkan *smart locker* berbasis ESP32, QR-Code, dan aplikasi Android dengan *functional suitability* dan *compatibility* 100%, *usability* 78,98%, serta rata-rata waktu pembukaan 0,874 detik [15].

Dibandingkan penelitian tersebut, KOMPAS tidak berfokus pada kecepatan aktuasi pintu, melainkan menyatukan autentikasi, dokumentasi kurir, dan pengiriman citra. Posisi ini memperjelas bahwa kontribusi KOMPAS berada pada integrasi alur penerimaan paket dan evaluasi *end-to-end*, sedangkan kecepatan aktuasi dan *usability* masih dapat menjadi parameter pengembangan berikutnya.

Secara keseluruhan, KOMPAS menawarkan satu alur penerimaan paket yang menggabungkan dokumentasi visual kurir, Telegram, *fingerprint*, *keypad*, servo, dan solenoid, kemudian mengevaluasi komunikasi serta autentikasi menggunakan data terukur. Kontribusi empiris yang paling jelas adalah tersedianya data *latency end-to-end* 17,1 detik, RSSI rata-rata -36,9 dBm dengan keberhasilan penerimaan foto 100% pada 10 percobaan, serta waktu respons autentikasi lokal. Hasil komparatif juga mengungkap kelemahan utama sistem, yaitu *latency* visual yang lebih tinggi daripada 8,09 detik pada penelitian Munirman et al. [10]. Temuan tersebut membuat kontribusi KOMPAS bersifat evaluatif: sistem menunjukkan integrasi fungsi yang berjalan, sekaligus menghasilkan *baseline* performa yang dapat digunakan untuk optimasi rancangan penerima paket berikutnya.

Analisis keamanan menunjukkan empat area yang harus diperhatikan pada implementasi berikutnya. Pertama, kredensial Telegram Bot tidak boleh dipublikasikan atau disimpan tanpa perlindungan; *source code* yang dibagikan harus menggunakan *placeholder* dan kredensial perlu dipisahkan dari kode aplikasi. Kedua, penyimpanan dan perlindungan template *fingerprint* perlu diaudit sesuai kemampuan modul yang digunakan. Ketiga, *keypad* memerlukan pembatasan percobaan, *lockout* sementara, atau *rate limiting* untuk mengurangi risiko *brute-force*. Keempat, pengujian *spoofing fingerprint* dan pengujian input berulang dalam jumlah besar diperlukan sebelum sistem dapat diberi klaim keamanan atau *reliability* yang lebih kuat. Analisis ini menempatkan autentikasi KOMPAS sebagai fungsi kontrol akses yang telah diuji, tetapi belum sebagai sistem keamanan siber yang tervalidasi secara menyeluruh.

Tabel 7. Perbandingan penelitian

Penelitian	Objek /Fitur utama	Metrik kuantitatif yang dilaporkan	Perbedaan terhadap KOMPAS	Kontribusi terhadap analisis KOMPAS
Riezqi et al. [5]	Smart box; ESP32-CAM; barcode;	Keberhasilan barcode, foto, dan notifikasi	Akses menggunakan barcode	Dasar pembandingan dokumentasi paket dan

	Telegram		Telegram	
Permana et al. [7]	Smart door lock; fingerprint; keypad; Firebase	100% black-box; 20 percobaan	Bukan sistem penerimaan paket	Pembandingan autentikasi fingerprint-keypad
Kurniasyah dan Rakhmawati [8]	Smart door lock; BLE; MQTT; RFID; fingerprint; keypad	5,052 kbps; delay 175 ms; jitter 19 ms	Protokol komunikasi berbeda	Pembandingan karakteristik komunikasi IoT
Munirman et al. [10]	ESP32-CAM; Telegram Bot API; notifikasi visual	Latency 6,03–10,45 detik; rata-rata 8,09 detik	Objek monitoring ruangan	Pembandingan utama latency end-to-end
Cakra yuda [12]	ESP32-CAM; PIR; Telegram	Akurasi 95%/90%; respons rata-rata 2,7 detik	Pemicu dan definisi respons berbeda	Menunjukkan potensi optimasi respons ESP32-CAM
Rahman et al. [13]	Fingerprint; smart phone; solenoid	Solenoid sekitar 3 detik	Fokus keamanan pintu	Pembandingan aktuasi akses
Ananda et al. [15]	Smart locker; QR-Code; Android	0,874 detik; functional/compatibility 100%; usability 78,98%	Tanpa dokumentasi kurir ESP32-CAM/Telegram	Pembandingan smart locker dan evaluasi

4. KESIMPULAN

Penelitian ini menghasilkan *prototype* Kotak Penerima Paket (KOMPAS) berbasis *Internet of Things* menggunakan ESP32 dan ESP32-CAM yang mengintegrasikan dokumentasi kurir, komunikasi Telegram Bot, autentikasi *fingerprint* dan *keypad*, serta pengendalian pintu menggunakan servo dan solenoid door lock. Pengujian *latency* menghasilkan rentang 15–20 detik dengan rata-rata 17,1 detik. Pengujian RSSI menghasilkan -29 sampai -55 dBm dengan rata-rata -36,9 dBm dan seluruh 10 foto berhasil diterima. Pada pengujian autentikasi, *fingerprint* memberikan keputusan sesuai pada 10 percobaan, sedangkan *keypad* menerima PIN benar dan menolak PIN salah pada skenario yang diuji. Dua percobaan tombol cepat hanya digunakan sebagai observasi fungsional dan tidak dijadikan dasar klaim *stress test* atau *reliability*.

Kontribusi KOMPAS terletak pada integrasi dokumentasi visual, komunikasi Telegram, dan autentikasi *fingerprint-keypad* dalam satu alur penerimaan paket serta penyediaan data empiris performa *end-to-end*. Perbandingan dengan penelitian ESP32-CAM–Telegram menunjukkan bahwa rata-rata *latency* KOMPAS 17,1 detik masih lebih tinggi daripada 8,09 detik yang dilaporkan Munirman et al. [10]. Hasil ini menetapkan *baseline* performa sekaligus menunjukkan bahwa optimasi *pipeline capture*, resolusi citra, koneksi Wi-Fi, *secure session*, dan *upload* Telegram merupakan prioritas pengembangan.

Evaluasi keamanan pada penelitian ini dibatasi pada fungsi autentikasi dan keputusan akses. Pengembangan berikutnya perlu memprioritaskan perlindungan kredensial Telegram, audit penyimpanan template *fingerprint*, pengukuran FAR/FRR dan *spoofing*, serta mekanisme *rate limiting* atau *lockout* pada *keypad*. Pengujian terkontrol terhadap kondisi jaringan dan profiling *latency* per tahap juga diperlukan untuk menentukan sumber bottleneck secara kuantitatif. Dengan arah tersebut, hasil penelitian ini dapat digunakan sebagai *baseline* untuk meningkatkan responsivitas dan ketahanan KOMPAS tanpa melebihi klaim dari data yang tersedia.

REFERENSI

- [1] S. A. N. H. Putri, O. B. Kharisma, H. Simaremare, dan A. Abdillah, "Smart Packgaes Box Berbasis *Internet of Things* Menggunakan Telegram Bot," *J. MEDIA Inform. BUDIDARMA*, vol. 7, no. 1, hal. 342–350, Jan 2023, doi: 10.30865/mib.v7i1.5517.
- [2] E. D. Febriyanti, 'Alimuddin 'Alimuddin, dan H. Mandala Putra, "Rancang Bangun Penerimaan Box Paket Berbasis *Internet of Things* (IoT)," *J. Print. J. Pengemb. Rekayasa Inform. dan Komput.*, vol. 2, no. 1, hal. 37–45, Jun 2024, doi: 10.29408/jprinter.v2i1.23825.
- [3] A. Jazilah, I. Siradjuddin, dan Gillang Al Azhar, "Implementasi Keamanan Box Penerima Paket Berbasis IoT," *J. Elektron. dan Otomasi Ind.*, vol. 11, no. 1, hal. 148–158, Mei 2024, doi: 10.33795/elkolind.v11i1.4038.
- [4] R. D. Handayani, Z. Jamal, M. Alkahfiansyah, L. Rosmalia, N. H. Sudibyo, dan R. Herwanto, "Intelligent and Secure Package Receiver System Utilizing *Internet of Things* (IoT) Technology," *Vokasi Unesa Bull. Eng. Technol. Appl. Sci.*, vol. 2, no. 3, hal. 581–592, Sep 2025, doi: 10.26740/vubeta.v2i3.38254.
- [5] A. D. F. Riezqi, H. Saputra, dan A. Sutrisman, "Rancang Bangun Smart Box Untuk Penyimpanan Paket Menggunakan Arduino Nano Dan Esp32-Cam Berbasis *Internet of Things*," *Politek. Negeri Sriwij.*, vol. 17, no. 3, hal. 941–952, 2025, doi: 10.5281/zenodo.17118468.
- [6] I. Syahputra dan Yussa Ananda, "Rancang Bangun Smart Locker Penitipan Barang Berbasis Fingerprint," *Journals Telecommun. Electr. Sci.*, vol. 01, no. 01, hal. 19–25, 2024, doi: <https://doi.org/10.24010/jtels.v1i02.943>.
- [7] K. A. K. Permana, I. N. Piarsa, dan A. A. K. A. C. Wiranatha, "IoT-Based Smart Door Lock System with Fingerprint and Keypad Access," *J. Inf. Syst. Informatics*, vol. 6, no. 3, hal. 2086–2098, Okt 2024, doi: 10.51519/journalisi.v6i3.844.
- [8] I. Kurniasyah dan L. Rakhmawati, "Smart Door Lock Innovation Using Integration of Bluetooth Low Energy and MQTT IoT Protocol," *Ina. Indones. J. Electr. Electronics Eng.*, vol. 8, no. 1, hal. 41–46, Jan 2025, doi: 10.26740/inajee.v8n1.p41-46.
- [9] S. D. Suandi, T. Andriani, L. Suryadi, M. Husyairi, dan Dahman, "Implementasi ESP32-CAM pada Pemantauan Penetasan Telur Ayam Berbasis Notifikasi Telegram," *J. Ris. Rekayasa Elektro*, vol. 5, no. 2, hal. 155–164, Des 2023, doi: 10.30595/jrre.v5i2.19734.
- [10] Sufianti Munirman, Riesa Krisna Astuti Sakir, Nurfitri, Airin Dewi Utami Thamrin, dan Nining Satriani, "Analisis Latensi dan Keandalan Notifikasi Visual Berbasis ESP32-CAM dan Telegram Bot API untuk Monitoring Keamanan Ruangan," *Jupiter Publ. Ilmu Keteknikan Ind. Tek. Elektro dan Inform.*, vol. 4, no. 4, hal. 156–171, Jul 2026, doi: 10.61132/jupiter.v4i4.1600.
- [11] Wahyudi, Rio, dan Edidas, "Perancang dan Pembuatan Sistem Keamanan Rumah Berbasis *Internet of Things* Menggunakan ESP32-CAM," *J. Pendidik. Tambusai*, vol. 6, no. 1, hal. 1135–1141, Feb 2022, doi: 10.31004/JPTAM.V6i1.3045.
- [12] L. Cakrayuda, M. R. Arhieadhie, dan A. S. Putra, "SICEMOT: Sistem keamanan cerdas berbasis Esp32-Cam, Sensor Gerak, Dam Notifikasi Telegram," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 2, Apr 2025, doi: 10.23960/jitet.v13i2.6420.
- [13] E. Suhardi Rahman, S. Gunawan Zain, dan A. Hidayat Adam, "Pengembangan Sistem Keamanan Pintu Menggunakan Fingerprint Dengan Sistem Notifikasi Berbasis *Internet of Things*," *J. Media Elektr.*, vol. 20, no. 1, hal. 51–61, Des 2022, doi: 10.59562/metrik.v20i1.5484.
- [14] B. I. Sugiarto dan Herianto, "Kombinasi smart door lock, Suara dan fingerprint pada sistem iot untuk keamanan ruangan," *J. Sains Teknol. Fak. Tek. Univ. Darma Persada*, vol. 14, no. 1, hal. 30–39, Agu 2024, doi: 10.70746/jstunsada.v14i1.489.
- [15] F. E. Ananda, D. Widiyanto, S. Haurameuthia, dan M. I. Tejasumirat, "Perancangan Smart Locker dengan Implementasi Sistem IoT dan Aplikasi Mobile Android," *Electr. J. Rekayasa dan Teknol. Elektro*, vol. 17, no. 1, hal. 92–99, Jan 2023, doi: 10.23960/elc.v17n1.2421.

