

Sistem Keamanan Pintu Rumah Berbasis Internet of Things via Pesan Telegram

Home Door Security System Based on Internet of Things Through Telegram Message

Jaenal Arifin^{1*}, Jery Frenando², Herryawan³

^{1,2,3} Fakultas Teknik Telekomunikasi dan Elektro Institut Teknologi Telkom Purwokerto
Jl. D.I Panjaitan No.128 Purwokerto
jaetoga@itttelkom-pwt.ac.id^{1*}, 114101060@itttelkom-pwt.ac.id²

Abstrak – Pada era sekarang ini, banyak didapati kasus keamanan rumah yang semakin kompleks. Hal ini yang mendasari penelitian ini dilakukan. Penelitian ini bertujuan membuat sistem keamanan pintu rumah dengan notifikasi pesan via aplikasi telegram yang diharapkan memberikan rasa aman bagi pemilik rumah dengan. Sistem keamanan pintu rumah ini menggunakan Wemos D1 R1 sebagai mikrokontroler. Sistem ini dirancang dengan menggunakan aplikasi telegram messenger sebagai notifikasi. Penelitian ini menggunakan proximity sensor yang berfungsi mendeteksi setiap gerakan di sekitar pintu. Di atas pintu dipasang kamera yang dapat menangkap gambar atau objek saat ada gerakan. Sistem ini memiliki 2 cara kerja, yaitu pengecekan secara otomatis dan manual. Sistem bekerja secara otomatis pada saat sensor menangkap gerakan dan kamera langsung mengambil gambar, selanjutnya mengirimkan gambar tersebut ke aplikasi telegram. Sedangkan sistem bekerja secara manual dengan cara memasukkan perintah lewat BOT telegram untuk membuka pintu rumah. Hasil pengujian nilai rata-rata kinerja alat secara otomatis dari penangkapan gambar atau objek sebesar 16,7 detik, nilai rata-rata waktu kinerja alat secara manual sebesar 9,7 detik. Nilai rata-rata waktu kinerja solenoid doorlock sebesar 7,6 detik. Hasil pengujian kualitas layanan menunjukkan rata-rata delay sebesar 3,244 detik, dan rata-rata throughput sebesar 301.465 byte/s.

Kata Kunci: Sistem keamanan pintu, IoT, Wemos D1, Proximity, Telegram.

Abstract – In today's era, many cases of home security are increasingly complex. This is what underlies this research. This study aims to create a home door security system with message notifications via the telegram application which is expected to provide a sense of security for homeowners. This home door security system uses Wemos D1 R1 as a microcontroller. This system is designed using the telegram messenger application as a notification. This study uses a proximity sensor to detect any movement around the door. Above the door is installed a camera that can capture images or objects when there is movement. This system has 2 ways of working, namely checking automatically and manually. The system works automatically when the sensor captures motion and the camera immediately takes a picture, then sends the image to the telegram application. While the system works manually by entering commands via the telegram BOT to open the door of the house. The results of testing the average value of the tool's performance automatically from capturing images or objects of 16.7 seconds, the average value of manual tool performance takes 9.7 seconds. The average value of the door locked solenoid performance takes 7.6 seconds. The result of testing the quality of data services show an average delay of 3,244 seconds, and an average throughput is 301,465 byte/s.

Keywords: Door security system, IoT, Wemos D1, Proximity, Telegram.

1. Pendahuluan

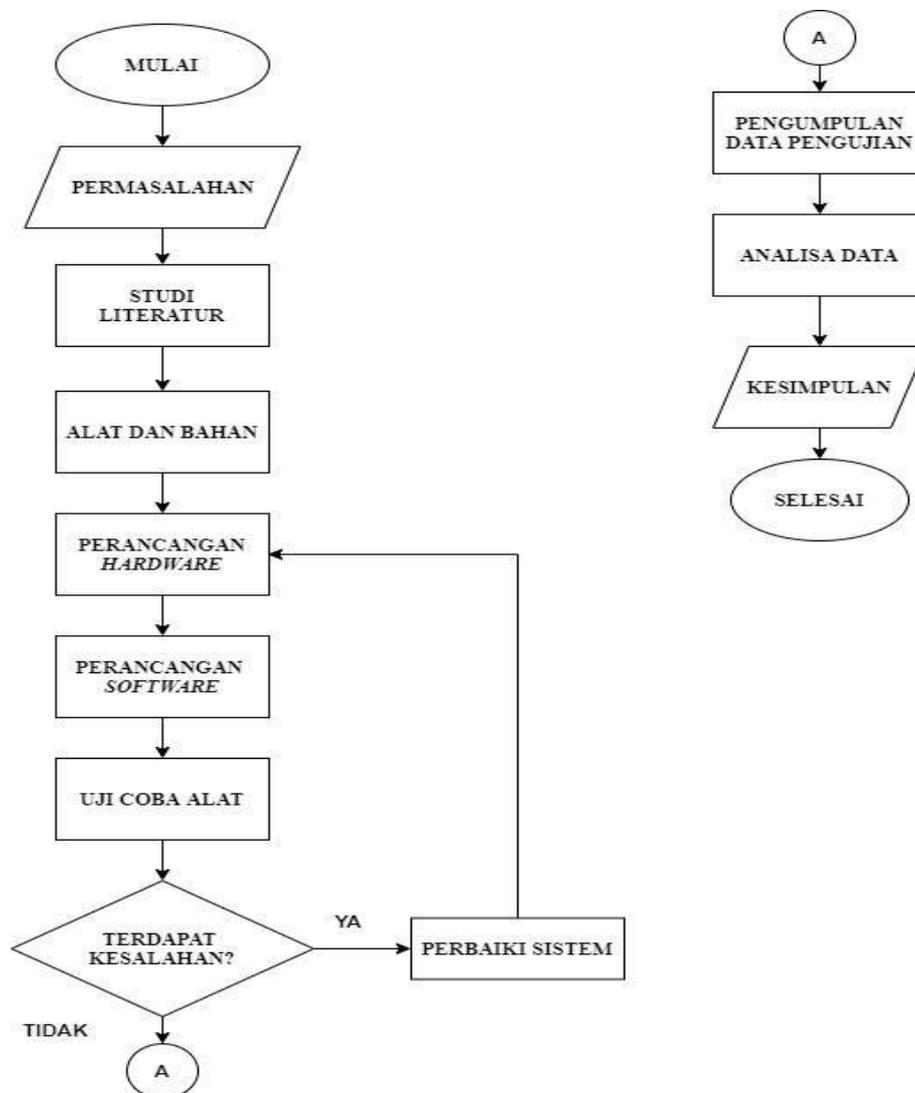
Beberapa penelitian yang membahas terkait sistem keamanan pintu rumah pernah dilakukan, seperti penelitian yang dilakukan oleh Arafat dengan judul “Sistem Pengamanan Pintu Rumah Berbasis Internet of Things dengan ESP8266” [1] dan penelitian yang dilakukan oleh Rajes Khana dengan judul “Rancang Bangun Sistem Keamanan Rumah Berbasis Internet of Things Dengan Platform Android” [2]. Penelitian tersebut mempunyai kelebihan dan kekurangan sesuai dengan sistem keamanan pintu yang dibuat. Misalnya, sering terjadi gagal perintah pada saat eksekusi program dan respon notifikasi yang terlambat dari sistem yang dibuat. Untuk mencari permasalahan dan solusi dari telatnya notifikasi tersebut, maka perlu dilakukan pengujian layanan data. Pengujian layanan data tersebut dapat berupa pengujian data yang dikirim dan data yang diterima. Pengujian rata-rata *delay* dan *throughput* perlu dilakukan untuk mengetahui kualitas layanan data. Penelitian lain yang masih terkait dengan keamanan pintu rumah adalah “Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger” [3]. Pada penelitian tersebut mengutarakan bahwa sistem keamanan rumah yang dibuat menggunakan sensor PIR (*Passive Infra Red*), Raspberry Pi dan aplikasi Telegram Messenger. Pada aplikasi telegram pemilik rumah diberi pilihan mengambil foto atau melakukan video.

Keamanan rumah sangat penting bagi penghuni rumah. Baik keamanan rumah yang menggunakan kode akses berbasis Arduino Mega [4] atau keamanan rumah yang menggunakan perangkat *mobile* [5]. Karena salah satu keamanan rumah terletak pada keamanan pintu, banyak kejadian pencurian yang berawal dari merusak atau melewati pintu rumah. Sudah banyak penelitian terkait keamanan pintu rumah [6] [7]. Dengan sistem yang lebih baik diharapkan dapat menjadi salah satu solusi terkait permasalahan keamanan pintu rumah. Pada penelitian ini penulis menggunakan Wemos D1 mini yang difungsikan sebagai mikrokontroler, Wemos D1 R1 yang merupakan *module development board* yang berbasis Wi-Fi bagian dari ESP8266 dan dapat diprogram menggunakan *software IDE* Arduino [8]. Modul ini memungkinkan mikrokontroler dapat terhubung kedalam jaringan Wi-Fi dan membuat koneksi TCP/IP. Selain itu ESP8266 dapat diterapkan pada desain mikrokontroler sebagai modul Wi-Fi melalui antarmuka SPI/SDIO atau I2C/UART [9]. Arduino IDE (*Integrated Development Environment*) adalah *software* yang digunakan untuk memprogram di arduino. Dengan kata lain, Arduino IDE merupakan suatu *tool* yang berfungsi sebagai media untuk memprogram *board Arduino* [10] [11]. Penelitian terkait sistem pengamanan rumah pernah dilakukan oleh Sriwati dkk [12]. Pada penelitian tersebut memanfaatkan sensor PIR yang ditempatkan di dalam rumah. Sensor dipasang sesuai dengan pemilik rumah. Sensor PIR berfungsi mendeteksi adanya gerakan. Pada penelitian yang dilakukan menggunakan ATmega 8 sebagai mikrokontroler. Pemilihan mikrokontroler ATmega8 dianggap sudah sesuai dengan kebutuhan saat sistem keamanan rumah ini dibuat [4]. Jika ada gerakan keluaran sensor PIR mengirimkan data ke mikrokontroler, selanjutnya mikrokontroler yang terhubung ke *handphone* pengguna akan mengirimkan informasi berupa sms. Isi sms berisi informasi kondisi atau status keadaan rumah.

Tujuan dari penelitian ini adalah membuat perangkat sistem keamanan pintu rumah dengan menambahkan kamera dengan notifikasi pesan via aplikasi telegram. Aplikasi telegram merupakan salah satu aplikasi yang sesuai dengan kebutuhan masyarakat. Aplikasi telegram memungkinkan pengguna untuk dapat mengirimkan pesan yang sifatnya rahasia dan pesan tersebut sudah dapat dienkripsi *end-to-end* sebagai keamanan tambahan. Di dalamnya terdapat fitur bot yang digunakan sebagai pemberi peringatan dan dapat mengecek keamanan pintu rumah melalui sensor yang sudah terhubung ke perangkat. Penelitian ini berfokus pada implementasi sistem keamanan pintu yang dibuat. Dengan penelitian ini diharapkan menjadi salah satu alternatif kaitannya dengan keamanan pintu rumah, sehingga dapat memberikan rasa aman bagi pemilik rumah.

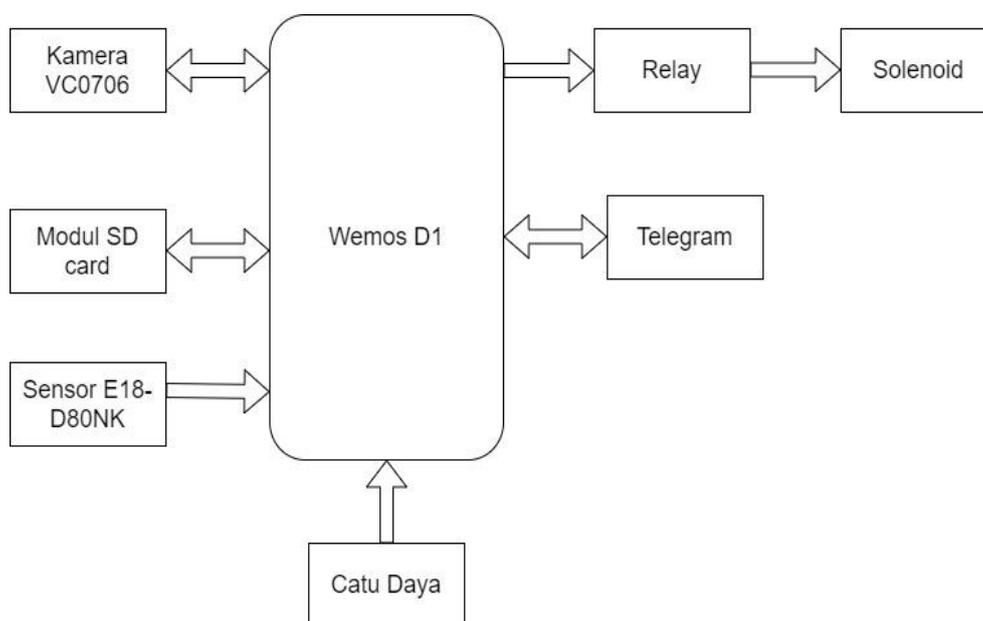
2. Metode Penelitian

Gambar 1 menunjukkan metode yang digunakan pada penelitian ini. Penelitian dilakukan dalam berbagai tahap yang dimulai dari menentukan pokok permasalahan. Selanjutnya yaitu tahap studi literatur yang dapat membantu penulis mencari referensi penelitian-penelitian sebelumnya yang berkaitan dengan topik pembahasan penulis. Studi literatur bertujuan untuk membedakan penelitian sebelumnya baik tentang metode, parameter dan analisis dari rancangan sistem. Setelah itu tahap pengumpulan data yang bertujuan untuk mengetahui informasi yang dibutuhkan untuk mencapai tujuan penelitian dan mendapatkan data yang valid untuk dapat dianalisa dan disimpulkan dari penelitian yang telah dilakukan. Perancangan dan implementasi alat pada tahap ini penulis memulainya dengan merancang perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras (*hardware*) dilakukan secara *step by step* agar mempermudah dalam memantau kinerja alat pada penelitian ini. Setelah perancangan *hardware* dan *software* sudah selesai dilakukan ujicoba alat. Apabila ujicoba alat terdapat kesalahan maka dilakukan perbaikan sistem. Apabila hasil perancangan berfungsi dengan baik maka dilanjutkan dengan pengambilan data dari hasil pembacaan sensor yang telah diproses mikrokontroler melalui program yang sudah dimasukkan.



Gambar1. Alur penelitian.

Gambar 2 merupakan blok diagram sistem dari penelitian yang telah dilakukan.



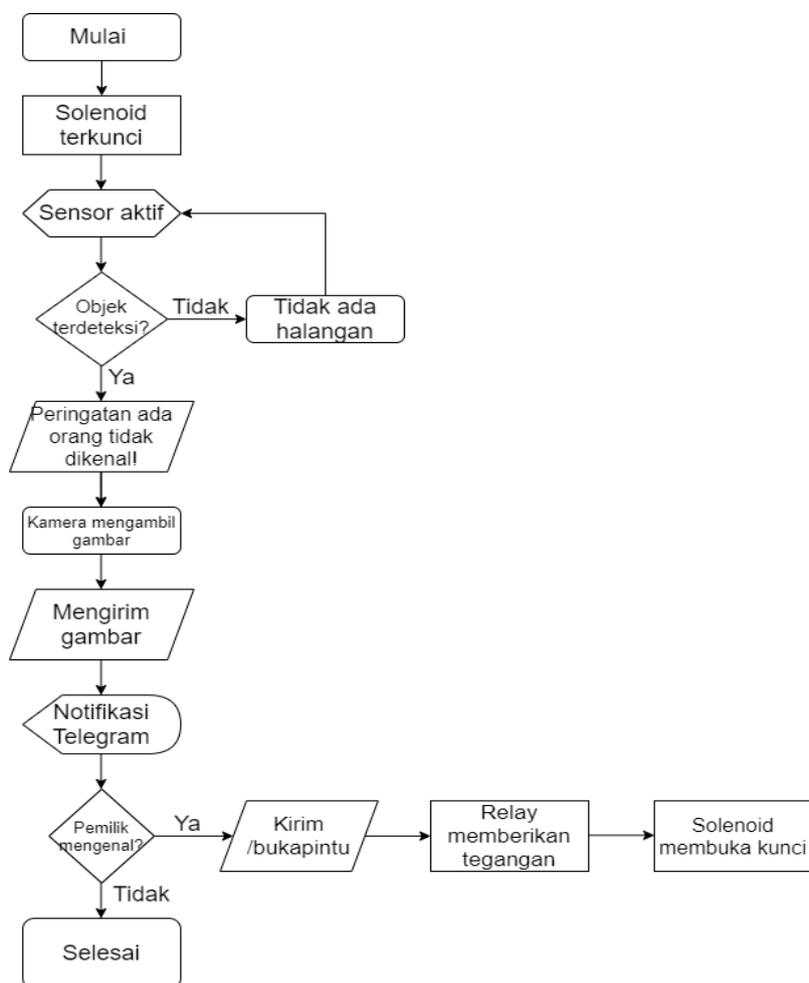
Gambar 2. Diagram blok sistem.

Gambar 2 merupakan blok diagram sistem yang dibuat secara keseluruhan. Catu daya yang digunakan memakai adaptor 9V. Diagram blok sistem tersebut terdiri dari *input* atau masukan, proses dan keluaran atau *output*. Pada posisi *input* atau masukan terdiri dari sensor E18-D80NK [13] yang berfungsi mendeteksi gerakan objek di sekitar pintu dan kamera VC0706 [14] yang berfungsi menangkap gambar di sekitar pintu. Pada sisi proses terdiri dari mikrokontroler Wemos D1 [15] dan modul SD card [16]. Mikrokontroler ini akan memerintahkan kamera untuk mengambil gambar jika sensor terhalang oleh objek. Sedangkan hasil dari pengambilan gambar kamera akan disimpan pada modul SD card sebagai perangkat tambahan. Hasil gambar yang tersimpan pada modul SD card dapat dikirimkan ke aplikasi telegram yang dipegang oleh pengguna. Pesan yang terkirim pada aplikasi telegram berupa notifikasi yang berisi peringatan (ada objek bergerak) hasil dari tangkapan kamera. Proses tersebut merupakan proses dimana sistem keamanan pintu yang dapat dipantau secara *online* oleh pengguna atau pemilik rumah. Pada posisi keluaran atau *output* solenoid akan selalu pada posisi *standby* pada keadaan mengunci pintu rumah. Solenoid *door lock* berfungsi sebagai pengunci pintu secara elektronik.

Solenoid akan bekerja dengan diberikan tegangan sesuai dengan kapasitasnya, yaitu 5 volt. Solenoid *door lock* yang sudah mengunci akan membuka secara otomatis setelah digerakkan oleh *relay* yang sudah diberi perintah. Perintah untuk membuka *relay* dapat dilakukan dengan menuliskan *chat bot* pada aplikasi *telegram*. Perintah tersebut akan diteruskan ke komponen *relay*. Posisi *relay* pada kondisi buka/*on* atau tutup/*off* sesuai dengan apa yang diketikkan pada *chat bot* pada aplikasi *telegram*. Artinya, selain aplikasi *telegram* dapat difungsikan sebagai output atau keluaran, pada kondisi tertentu aplikasi *telegram* dapat difungsikan sebagai input atau masukan. Dengan aplikasi *telegram* yang dapat difungsikan sebagai input atau output pada perangkat ini harapannya dapat memberikan kenyamanan pada pengguna atau pemilik rumah. Aplikasi *telegram* ini [17] mempunyai peran yang sangat penting pada perangkat ini karena dapat memberi perintah atau masukan/input dan penerima hasil yaitu keluaran/output dan sebagai notifikasi peringatan kepada pengguna setelah mendapat peringatan dari sensor bahwa ada objek yang tertangkap dan dinyatakan bergerak.

2.1. Flowchart Perangkat Bekerja Secara Otomatis

Gambar 3 menunjukkan *flowchart* ketika perangkat bekerja secara otomatis.



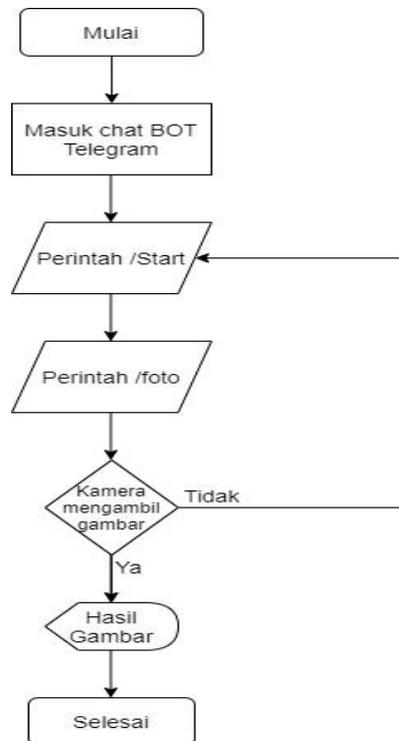
Gambar 3. *Flowchart* perangkat ketika bekerja secara otomatis.

Berdasarkan *flowchart* penelitian pada Gambar 3, posisi awal solenoid pada kunci pintu dalam keadaan terkunci. Kondisi pintu rumah dalam keadaan terkunci dan aman. Pada saat sensor mendeteksi adanya halangan atau objek maka secara otomatis kamera akan mengambil gambar dari objek yang menghalangi. Diasumsikan bahwa kamera mengambil gambar atau objek manusia tidak dikenal oleh pemilik rumah. Objek gambar tersebut selanjutnya akan disimpan pada *micro SD card*. Objek gambar dikirim ke bot telegram sebagai informasi adanya notifikasi di *handpone* pemilik rumah. Jika pemilik rumah mengenali objek gambar (diasumsikan orang terdekat dari pemilik rumah) yang tertangkap kamera, maka pemilik rumah dapat mengirim perintah buka pintu pada aplikasi telegram (fitur *chat* pada aplikasi telegram). Selanjutnya solenoid pada pintu akan membuka slot pengunci pintu, sehingga orang yang dimaksud dapat masuk rumah. Pengamanan pintu rumah seperti ini dianggap efektif, karena pemilik rumah tidak harus berada didalam atau sekitar rumah. Pemilik rumah berada diluar rumah pun tetap dapat mengetahui kondisi rumahnya dengan cara memasang kamera di depan pintu rumah. Kondisi seperti ini mewajibkan pemilik rumah untuk selalu membawa *handpone*-nya jika berada di luar rumah. Pada *handpone* juga tentunya sudah terpasang aplikasi telegram. Pada pengaturan notifikasi aplikasi telegram harus selalu aktif. Jadi pada saat ada pesan/notifikasi masuk pemilik rumah mengetahui adanya notifikasi/pesan yang harus dibaca. Jika pesan atau notifikasi tersebut

membutuhkan tindakan atau langkah nyata maka dapat memerintahkan pada chat bot telegram untuk dapat membuka solenoid pada pintu rumah.

2.2. Flowchart perangkat bekerja secara manual

Gambar 4 menunjukkan *flowchart* ketika perangkat bekerja secara manual.



Gambar 4. *Flowchart* perangkat ketikan bekerja secara manual.

Pada pengecekan secara manual ini pengguna perlu memasukkan perintah */check* pada *chat bot* telegram. Selanjutnya wemos sebagai mikrokontroler mengirim perintah cek keadaan sekitar. Kamera akan menangkap gambar sekitar dan dikirimkan balik ke *chat bot* telegram sebagai notifikasi. Pengiriman gambar hasil tangkapan kamera dengan sudut 60° atau sesuai pengaturan yang disesuaikan dengan kebutuhan pengguna. Pada saat penulis mengambil gambar dari kamera ukuran file berupa 640×480 pixel dengan format JPEG, pemilik rumah dapat melihat keadaan sekitar dengan jelas tanpa harus bergantung pada sensor. Pengecekan secara manual ini bertujuan sebagai alternatif solusi jika sensor tidak berfungsi atau rusak di kemudian hari.

3. Hasil dan Pembahasan

Pengujian dan pembahasan perangkat dilakukan setelah perancangan dan pembuatan perangkat. Pengujian dilakukan untuk mengetahui kinerja perangkat dan untuk memenuhi parameter kelayakan perangkat. Dari pengujian yang dilakukan diharapkan mendapatkan hasil pengukuran data secara akurat dan mendapatkan keluaran yang sesuai dengan kebutuhan dari tujuan awal pembuatan perangkat.

3.1. Pengujian Kinerja Perangkat

Pada pengujian ini dilakukan 30 kali pengambilan data secara otomatis dengan cara memasang sensor PIR agar dapat menangkap objek yang menghalangi. Kamera berfungsi mengambil gambar dan menyimpannya di *SD card*. Selanjutnya mengirimkan gambar tersebut ke aplikasi telegram sebagai notifikasi. Pengujian dilakukan untuk mengetahui respon dari sensor saat mendeteksi objek. Berikut pengujian yang didapatkan melalui serial monitor program IDE ditunjukkan pada Gambar 5.

```

21:34:19.543 -> Object terdeteksi
21:34:25.074 -> Storing Picture...!
21:34:26.105 -> 12632
21:34:33.324 -> Siap kirim file : img.jpg
21:34:33.417 -> 0
21:34:33.417 -> 1
21:34:33.417 -> 2
21:34:33.417 -> 3
21:34:33.417 -> 3
21:34:33.417 -> 4

```

Gambar 5. Tampilan hasil data kinerja perangkat dari serial monitor program IDE.

Pengujian dilakukan sebanyak 30 kali dan didapatkan rata-rata rentang waktu selama pengujian sebesar 70558 ms. Pengujian ini berfokus pada durasi waktu sensor saat mendeteksi objek dan durasi kamera saat pengimanan dimulai. Pengujian ini bertujuan untuk mengetahui rentang waktu sensor pada saat sensor mendeteksi keberadaan objek yang menghalangi atau terdeteksi. Selanjutnya objek yang terdeteksi tersebut ditangkap oleh kamera. Hasil gambar dapat disimpan pada SD card. Pada tabel pengujian rentang waktu objek terdeteksi oleh sensor sampai pada objek atau gambar terkirim ke aplikasi telegram. Pada pengujian tersebut didapatkan selisih waktu antara objek terdeteksi dengan pada saat gambar terkirim ke aplikasi telegram. Hasil pengujian rentang waktu objek terdeteksi dan menyimpan gambar ditunjukkan pada Tabel 1.

Tabel 1. Hasil data rentang waktu objek terdeteksi dan menyimpan gambar.

Pengujian ke	Objek terdeteksi (WIB)	Menyimpan gambar (WIB)	Selisih waktu
1	Pukul 21:34:19.543	Pukul 21:34:25.074	5531 ms
2	Pukul 21:38:09.465	Pukul 21:38:14.434	4969 ms
3	Pukul 21:41:18.963	Pukul 21:41:23.931	4968 ms
4	Pukul 21:44:45.110	Pukul 21:44:53.406	8296 ms
5	Pukul 21:51:11.582	Pukul 21:51:17.254	5627 ms
6	Pukul 21:54:35.169	Pukul 21:54:40.372	5203 ms
7	Pukul 21:58:36.506	Pukul 21:58:42.084	5578 ms
8	Pukul 22:00:17.815	Pukul 22:00:23.627	5812 ms
9	Pukul 22:04:04.589	Pukul 22:04:10.354	5765 ms
10	Pukul 22:07:08.234	Pukul 22:07:14.161	5931 ms
11	Pukul 14:56:13.448	Pukul 14:56:23.419	5971 ms
12	Pukul 14:58:36.738	Pukul 14:58:43.978	7240 ms
13	Pukul 15:02:52.354	Pukul 15:03:01.762	9408 ms
14	Pukul 15:04:26.004	Pukul 15:04:35.325	9321 ms
15	Pukul 15:06:09.917	Pukul 15:06:18.917	9000 ms
16	Pukul 15:09:00.691	Pukul 15:09:07.111	6420 ms
17	Pukul 15:14:45.860	Pukul 15:14:52.882	7022 ms
18	Pukul 22:07:08.234	Pukul 15:17:29.453	8111 ms
19	Pukul 15:19:50.122	Pukul 15:19:58.558	8446 ms
20	Pukul 15:21:37.734	Pukul 15:21:44.750	7016 ms
21	Pukul 15:23:03.454	Pukul 15:23:10.539	7085 ms
22	Pukul 15:24:50.876	Pukul 15:24:57.910	7034 ms
23	Pukul 15:27:00.108	Pukul 15:27:09.093	8385 ms
24	Pukul 15:29:45.832	Pukul 15:29:52.951	7119 ms
25	Pukul 15:30:58.328	Pukul 15:31:05.207	7031 ms
26	Pukul 15:32:46.770	Pukul 15:32:54.714	7944 ms
27	Pukul 15:33:56.772	Pukul 15:34:06.284	9867 ms
28	Pukul 15:35:04.244	Pukul 15:35:11.555	8678 ms
29	Pukul 15:35:49.423	Pukul 15:35:58.694	8587 ms
30	Pukul 15:37:56.761	Pukul 15:38:05.174	9587 ms
Rata-rata	rentang waktu pengujian		7058 ms

Tabel 2 dan Tabel 3 menunjukkan rentang waktu sensor mendeteksi sampai ke aplikasi telegram secara otomatis dan secara manual. Pengujian ini penting dilakukan untuk menguji deteksi sensor dan kamera yang telah terpasang pada sistem keamanan pintu rumah yang telah dibuat.

Pada Tabel 2 dapat dilihat bahwa rata-rata rentang waktu pengujian perangkat secara keseluruhan sebesar 16754 ms atau 16,7 *second*. Pengujian ini bertujuan untuk mengetahui rentang waktu sensor pada saat sensor mendeteksi keberadaan objek yang menghalangi sampai pada gambar terkirim ke aplikasi telegram. Gambar yang terkirim pada aplikasi telegram sebagai notifikasi bagi pemilik rumah. Pada pengujian ini perangkat berkerja dengan baik. Sensor mampu mendeteksi objek yang menghalangi dan memberikan notifikasi hasil data gambar pada pengujian kinerja perangkat secara otomatis.

Rata-rata rentang waktu pengujian sebesar 9751 ms atau 9,7 *second*. Pada pengujian ini alat mampu berkerja dengan baik dan mampu memberikan notifikasi telegram dengan memberikan keadaan gambar sekitar yang tertangkap kamera dengan jelas. Jika dibandingkan dengan rata-rata rentang waktu pengujian secara manual dan otomatis kinerja perangkat secara manual lebih cepat. Ini disebabkan proses secara manual membutuhkan langkah yang lebih sedikit dibandingkan secara otomatis. Pada pengujian secara manual kamera hanya mengambil gambar pada pantauan sekitar pintu dan mengirimkan objek atau gambar melalui aplikasi telegram.

Tabel 2. Rentang waktu proses objek terdeteksi sampai ke telegram secara otomatis.

Pengujian ke	Objek terdeteksi (WIB)	Gambar terkirim (WIB)	Selisih waktu
1	Pukul 21:34:19.543	Pukul 21:34:34.120	14577 ms
2	Pukul 21:38:09.465	Pukul 21:38:28.199	18734 ms
3	Pukul 21:41:18.963	Pukul 21:41:33.822	14859 ms
4	Pukul 21:44:45.110	Pukul 21:45:01.703	16703 ms
5	Pukul 21:51:11.582	Pukul 21:51:30.566	18984 ms
6	Pukul 21:54:35.169	Pukul 21:54:48.903	13734 ms
7	Pukul 21:58:36.506	Pukul 21:58:50.849	14343 ms
8	Pukul 22:00:17.815	Pukul 22:00:37.267	19452 ms
9	Pukul 22:04:04.589	Pukul 22:04:24.932	20343 ms
10	Pukul 22:07:08.234	Pukul 22:07:29.591	21357 ms
11	Pukul 14:56:13.448	Pukul 14:56:30.963	17515 ms
12	Pukul 14:58:36.738	Pukul 14:58:50.456	13718 ms
13	Pukul 15:02:52.354	Pukul 15:03:10.591	18541 ms
14	Pukul 15:04:26.004	Pukul 15:04:44.685	18681 ms
15	Pukul 15:06:09.917	Pukul 15:06:29.174	19203 ms
16	Pukul 15:09:00.691	Pukul 15:09:18.791	18100 ms
17	Pukul 15:14:45.860	Pukul 15:15:03.683	18211 ms
18	Pukul 22:07:08.234	Pukul 15:17:39.278	17936 ms
19	Pukul 15:19:50.122	Pukul 15:20:08.296	18245 ms
20	Pukul 15:21:37.734	Pukul 15:21:55.264	17530 ms
21	Pukul 15:23:03.454	Pukul 15:23:21.976	18513 ms
22	Pukul 15:24:50.876	Pukul 15:25:07.001	16896 ms
23	Pukul 15:27:00.108	Pukul 15:27:18.317	18205 ms
24	Pukul 15:29:45.832	Pukul 15:30:02.229	16468 ms
25	Pukul 15:30:58.328	Pukul 15:31:14.441	16278 ms
26	Pukul 15:32:46.770	Pukul 15:33:03.783	17013 ms
27	Pukul 15:33:56.772	Pukul 15:34:13.100	16420 ms
28	Pukul 15:35:04.244	Pukul 15:35:20.967	16723 ms
29	Pukul 15:35:49.423	Pukul 15:36:09.011	19607 ms
30	Pukul 15:37:56.761	Pukul 15:38:14.457	17696 ms
Rata-rata	rentang waktu pengujian		16754 ms

Tabel 3. Rentang waktu menyimpan gambar sampai terkirim ke telegram secara manual.

Pengujian ke	Gambar tersimpan (WIB)	Gambar terkirim (WIB)	Selisih waktu
1	Pukul 22:12:56.790	Pukul 22:13:05.462	8672 ms
2	Pukul 22:16:11.834	Pukul 22:16:20.412	8578 ms
3	Pukul 22:19:19.634	Pukul 22:19:28.259	8625 ms
4	Pukul 22:22:17.519	Pukul 22:22:29.988	12469 ms
5	Pukul 22:39:01.901	Pukul 22:39:10.244	8343 ms
6	Pukul 22:45:37.712	Pukul 22:45:46.383	8617 ms
7	Pukul 22:47:19.537	Pukul 22:47:28.161	8624 ms
8	Pukul 22:50:59.007	Pukul 22:51:07.632	8625 ms
9	Pukul 22:52:53.879	Pukul 22:53:02.503	8624 ms
10	Pukul 22:54:49.703	Pukul 22:54:58.328	8625 ms
11	Pukul 15:38:59.496	Pukul 15:39:07.798	8320 ms
12	Pukul 15:40:25.820	Pukul 15:40:34.885	9065 ms
13	Pukul 15:46:03.516	Pukul 15:46:15.468	11952 ms
14	Pukul 15:48:55.458	Pukul 15:49:05.870	11206 ms
15	Pukul 15:52:31.310	Pukul 15:52:41.913	10603 ms
16	Pukul 15:52:50.408	Pukul 15:53:00.583	10825 ms
17	Pukul 15:55:30.753	Pukul 15:55:39.978	9225 ms
18	Pukul 15:57:51.747	Pukul 15:58:00.691	9253 ms
19	Pukul 16:07:20.219	Pukul 16:07:31.852	11633 ms
20	Pukul 16:09:15.370	Pukul 16:09:26.624	11254 ms
21	Pukul 16:11:49.180	Pukul 16:11:59.584	10404 ms
22	Pukul 16:12:42.190	Pukul 16:12:53.633	11443 ms
23	Pukul 16:13:01.062	Pukul 16:13:11.389	10327 ms
24	Pukul 16:15:01.220	Pukul 16:15:10.972	9752 ms
25	Pukul 16:15:27.069	Pukul 16:15:39.832	12763 ms
26	Pukul 16:16:12.332	Pukul 16:16:20.239	7904 ms
27	Pukul 16:16:55.524	Pukul 16:17:06.049	10125 ms
28	Pukul 16:17:55.051	Pukul 16:18:06.384	10333 ms
29	Pukul 16:18:43.221	Pukul 16:18:51.239	8018 ms
30	Pukul 16:20:24.357	Pukul 16:20:32.722	8365 ms
Rata-rata	rentang waktu pengujian		9751 ms

3.2. Pengujian Delay

Pengujian ini dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan untuk melakukan pengiriman data pada perangkat. Dengan menghubungkan laptop yang ter-*install software wireshark* pada jaringan yang digunakan menggunakan Wemos D1.

Tabel 4. Hasil Pengujian delay.

Pengujian ke	Pengambilan data ke	Delay (Byte/s)
1	Pengambilan data pertama	0,3868
2	Pengambilan data kedua	0,306
3	Pengambilan data ketiga	0,626
4	Pengambilan data keempat	0,474
5	Pengambilan data kelima	0,254
6	Pengambilan data keenam	0,423
7	Pengambilan data ketujuh	0,164
8	Pengambilan data kedelapan	0,245
9	Pengambilan data kesembilan	0,199
10	Pengambilan data kesepuluh	0,167
Rata-rata	Delay	0,32448

Pengujian dilakukan sebanyak 10 kali dengan waktu 5 menit pada setiap pengujian. Rata-rata delay yang diperoleh sebesar 0,32448 detik. Dari rata-rata delay yang diperoleh dapat diinformasikan bahwa pengujian perangkat mendapatkan parameter kurang bagus, karena nilai *delay* yang diperoleh berada diatas 300 s/d 450 ms.

3.3. Pengujian *Packet Loss*

Pengujian paket *loss* dilakukan untuk mengetahui berapa persen paket data yang dikirimkan dengan yang diterima. Mikrokontroler wemos D1 diposisikan sebagai pengirim dan laptop yang ter-install *wireshark* sebagai penerima. Masing-masing pengujian dilakukan selama 5 menit.

Tabel 5. Hasil Pengujian Packet Loss.

Pengujian ke	Besaran Data (Byte)	Paket yang ditangkap	Paket yang ditampilkan	Paket Loss (%)
1	124993	2424	2424	0%
2	96426	678	678	0%
3	190254	913	913	0%
4	143272	956	956	0%
5	85754	813	813	0%
6	12660	850	850	0%
7	53149	694	694	0%
8	71308	655	655	0%
9	59355	611	611	0%
10	67421	872	872	0%

Dari hasil data pengujian sesuai dengan Tabel 5 terlihat bahwa hasil pengujian paket *loss* perangkat didapatkan persentase paket *loss*-nya sebesar 0%. Ini berarti untuk data paket yang dikirimkan berhasil/sukses semua atau dalam arti lain tidak ada paket yang gagal.

3.4. Pengujian *Throughput*

Pengujian parameter *throughput* bertujuan untuk mengetahui berapa banyak data yang diterima dalam kurun waktu tertentu. Dalam pengujian ini wemos D1 diposisikan sebagai pengirim dan *wireshark* sebagai penerima. *Wireshark* sudah ter-install pada laptop atau pc pada saat melakukan pengujian.

Tabel 6. Hasil pengujian *throughput*.

Pengujian ke	Delay (s)	Throughput (Byte/s)
1	0,3868	323,134
2	0,306	314,813
3	0,626	303,622
4	0,474	303,622
5	0,254	336,849
6	0,423	299,138
7	0,164	323,580
8	0,245	290,718
9	0,199	297,602
10	0,167	403,573
Rata-rata	Throughput	301,465

Dari hasil data pengujian menurut Tabel 6 didapat nilai *throughput* terkecil adalah 290.718 byte/s. Sedangkan nilai rata-rata *throughput* sebesar 301.465 byte/s.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa nilai rata-rata pengujian kinerja alat secara otomatis dari objek terdeteksi sampai menyimpan gambar mendapatkan hasil data rentang waktu dari 30 kali pengujian sebesar 7055 ms. Nilai rata-rata pengujian kinerja *relay* pada solenoid sebesar 7659 ms atau 7,6 detik pada 30 kali pengujian. Hasil pengujian kualitas layanan data yaitu rata-rata *delay* sebesar 3,2448 detik dan rata-rata *throughput* sebesar 301.465 byte/s.

Referensi

- [1] Arafat, "Sistem Pengamanan Pintu Rumah Berbasis Internet of Things dengan ESP8266," *Technologia*, vol. 7, no. 3, pp. 262-267, 2013.
- [2] R. Khana and U. Usnul, "Rancang Bangun Sistem Keamanan Rumah Berbasis Internet of Things Dengan Platform Android," *Jurnal Kajian Teknik Elektro*, vol. 3, no. 1, pp. 18-31, 2018.
- [3] M. I. Kurniawan, U. Sunarya, and R. Tulloh, "Internet of Things : Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger," *ELKOMIKA Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi & Teknik Elektronika*, vol. 6, no. 1, p. 1, 2018.
- [4] D. Nataliana, S. Anwari, and M. S. Akbar, "Implementasi Prototype Sistem Home security dengan Pemanfaatan Kode Akses berbasis Arduino Mega," *ELKOMIKA Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi & Teknik Elektronika*, vol. 5, no. 2, p. 119, 2018.
- [5] Z. Abidin, S.I. Lesatariningati, "Sistem keamanan dan monitoring rumah pintar secara online menggunakan perangkat mobile," *Jurnal Teknik Komputer*, vol. 3, no. 2, pp. 13-17, 2014.
- [6] I. Alfannizar and Y. Rahayu, "Perancangan Dan Pembuatan Alat Home Electricity Based Home Appliance Controller Berbasis Internet of Things," *Jom FTEKNIK*, vol. 5, no. 1, pp. 1-6, 2018.
- [7] S. Siswanto, T. Nurhadiyan, and M. Junaedi, "Prototype Smart Home Dengan Konsep Iot (Internet of Thing) Berbasis Nodemcu Dan Telegram," *Jurnal Sistem Informasi dan Informatika*, vol. 3, no. 1, pp. 85-93, 2020.
- [8] Eeeboxbd.com, "WeMos D1 R1 WiFi Development Board." [Online]. Available: <https://eeeboxbd.com/product/wemos-d1-r1-wifi-development-board/>.
- [9] E. Systems, "ESP8266EX Datasheet," in *Espressif Systems*, 2020, p. 31.
- [10] H. Sullinger, *Getting Started With The Arduino: Arduino Integrated Development Environment: Install The Arduino Software (Ide) On Windows Pcs*. Independently Published, 2021.
- [11] J. Ozer and H. Blemings, *Practical Arduino: Cool Projects for Open Source Hardware*. Apress, 2011.
- [12] S. Baco, L. Belakang, and R. Masalah, "Pengembangan Sistem Pengaman Rumah dengan Security Password Menggunakan Sensor Gerak berbasis Mikrokontroler ATmega8," *Seminar Nasional Inovasi dan Aplikasi Teknologi Di Industri*, pp. 1-7, 2017.
- [13] Tinkbox, "Proximity Sensor/Switch E18-D80NK," 2014, pp. 1-5.
- [14] A. Industries, "Overview TTL Serial Camera." [Online]. Available: <https://learn.adafruit.com/ttl-serial-camera>. [Accessed: 30-Sep-2021].
- [15] E. A. Core, "ESP8266 Arduino," *GitHub*. [Online]. Available: <https://arduino-esp8266.readthedocs.io/en/latest/>. [Accessed: 29-Sep-2021].
- [16] M. C. Adapter and M. Sd, "Micro SD Card Module for Arduino," 2014, pp. 1-21.
- [17] Telegram.org, "Telegram Applications." [Online]. Available: <https://telegram.org/apps>.