

***SMART DOORBELL* MENGGUNAKAN SISTEM
NOTIFIKASI TELEGRAM**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Muhammad Rafly Fadhel	NIM	0032050
Syahrul Ramadani	NIM	0032059

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2023**

LEMBAR PENGESAHAN

**SMART DOORBELL MENGGUNAKAN SISTEM
NOTIFIKASI TELEGRAM**

Oleh :

Muhammad Rafly Fadhel NIM 0032050

Syahrul Ramadani NIM 0032059

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



(Ocsirendi, M.T.)

Pembimbing 2



(Indra Dwisaputra, M.T.)

Penguji 1



(Surojo, M.T.)

Penguji 2



(Priestiani, M.P.)

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Muhammad Rafly Fadhel

NIM: 0032050

Nama Mahasiswa 2 : Syahrul Ramadani

NIM: 0032059

Dengan Judul : *Smart Doorbell* Menggunakan Sistem Notifikasi Telegram

Menyatakan bahwa laporan akhir ini adalah hasil kerja keras kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 17 Juli 2023

Nama Mahasiswa

Tanda Tangan

1. Muhammad Rafly Fadhel



2. Syahrul Ramadani



ABSTRAK

Dengan kemajuan teknologi, ditemukanlah bel listrik konvensional. Bel listrik konvensional adalah sebuah perangkat yang berguna untuk memberi tahu pemilik rumah tentang kedatangan tamu melalui bunyi yang dihasilkan setelah tamu menekan tombol bel tersebut. Namun, ketika pemilik rumah sedang keluar maka tidak tahu apabila ada tamu yang ingin ke rumah untuk bertamu. Bel ini hanya berlaku saat pemilik rumah di rumah. Jika pemilik rumah sedang keluar maka tidak bisa mendengarkan suara bel tersebut sebagai penanda kalau ada tamu. Dengan menerapkan Smart Doorbell Menggunakan Sistem Notifikasi Telegram, diharapkan pemilik rumah dapat menerima notifikasi dari mikrokontroler ESP32 ke telegram dengan kalimat "Ada Tamu Di Depan Pintu". Pemilik rumah bisa mengambil gambar tamu dengan ESP32-CAM dan gambar hasil capture dikirim ke telegram dengan rata-rata waktu 7,6 detik. Pemilik rumah dapat memberikan informasi kepada tamu melalui pesan yang diketik pada aplikasi Telegram serta dapat ditampilkan pada layar LCD dengan waktu rata-rata kirim 2,8 detik. Begitu juga tamu bisa memberikan pesan ke pemilik rumah dengan memanfaatkan push button serta dikirim ke telegram secara real time. Dengan demikian, tamu dapat mengetahui apakah orang yang ingin ditemui sedang berada di rumah atau tidak, begitu juga pemilik rumah bisa mengetahui siapa tamu yang datang ke rumahnya. Jadi kesimpulannya antara tamu dengan pemilik rumah bisa melakukan komunikasi secara dua arah.

Kata Kunci : Bel, IoT, ESP32 ,ESP32-CAM, Telegram.

ABSTRACT

With the advancement of technology, the conventional electric bell was invented. A conventional electric bell is a device that is useful for informing homeowners about the arrival of guests through the sound produced after the guest presses the bell button. However, when the homeowner is out, they do not know if there are guests who want to come to the house to visit. This bell is only valid when the homeowner is at home. If the homeowner is out, they cannot listen to the sound of the bell as a sign that there are guests. By implementing a Smart Doorbell Using Telegram Notification System, it is expected that homeowners can receive notifications from the ESP32 microcontroller to the telegram with the phrase "There are Guests at the Door". Homeowners can take pictures of guests with ESP32-CAM and the captured images are sent to the telegram with an average time of 7.6 seconds. Homeowners can provide information to guests through messages typed in the Telegram application and can be displayed on the LCD screen with an average sending time of 2.8 seconds. Likewise, guests can give messages to homeowners by utilizing push buttons and sent to telegrams in real time. Thus, guests can find out whether the person they want to meet is at home or not, as well as the homeowner can find out who the guests are who come to his house. So the conclusion is that between guests and homeowners can communicate in two directions.

Keywords: Doorbell, IoT, ESP32, ESP32-CAM, Telegram.

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat, ridha dan hidayah-Nya serta memberikan banyak kesempatan sehingga penulis dapat menyelesaikan laporan proyek akhir yang berjudul “*Smart doorbell* Menggunakan Sistem Notifikasi Telegram” dengan baik.

Laporan proyek akhir ini disusun sebagai salah satu persyaratan akademik dan kewajiban mahasiswa pada semester 6 untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Selesaiannya penyusunan laporan ini tidak terlepas dari dukungan, semangat, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan kali ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang ikut berperan dalam penyelesaian laporan ini, terutama kepada :

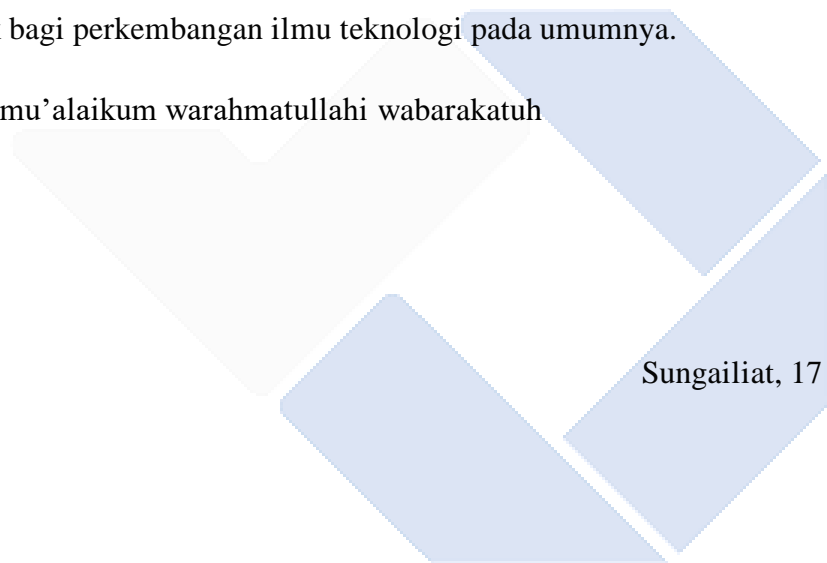
1. Allah SWT yang telah memberikan kemudahan kepada penulis selama pembuatan laporan proyek akhir.
2. Keluarga yang selalu memberikan motivasi, dukungan, serta semangat sehingga penulis dapat menyelesaikan laporan ini sesuai dengan waktu yang telah ditentukan.
3. Bapak I Made Andik Setiawan, M.Eng,Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Zanu Saputra, M.Tr.T. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Ocsirendi, M.T. selaku Kepala Program Studi DIII Teknik Elektronika sekaligus dosen pembimbing 1 dalam proyek akhir ini.
6. Bapak Indra Dwisaputra, M.T, selaku pembimbing 2 dalam proyek akhir ini.
7. Seluruh staf pengajar dan karyawan Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan- rekan mahasiswa yang ikut mendukung dan memberikan bantuan serta masukan dalam menyelesaikan proyek akhir ini.

9. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan proyek akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa penulisan proyek akhir ini masih jauh dari kata sempurna karena kesempurnaan hanya milik Allah SWT. Oleh sebab itu itu, penulis mengharapkan segala petunjuk, kritik dan saran yang bersifat membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya.

Besar harapan penulis semoga laporan tugas akhir dan alat proyek akhir yang dibuat dapat memberikan manfaat bagi pihak yang berkepentingan pada khususnya dan baik bagi perkembangan ilmu teknologi pada umumnya.

Wassalamu'alaikum warahmatullahi wabarakatuh



Sungailiat, 17 Juli 2023

Penulis

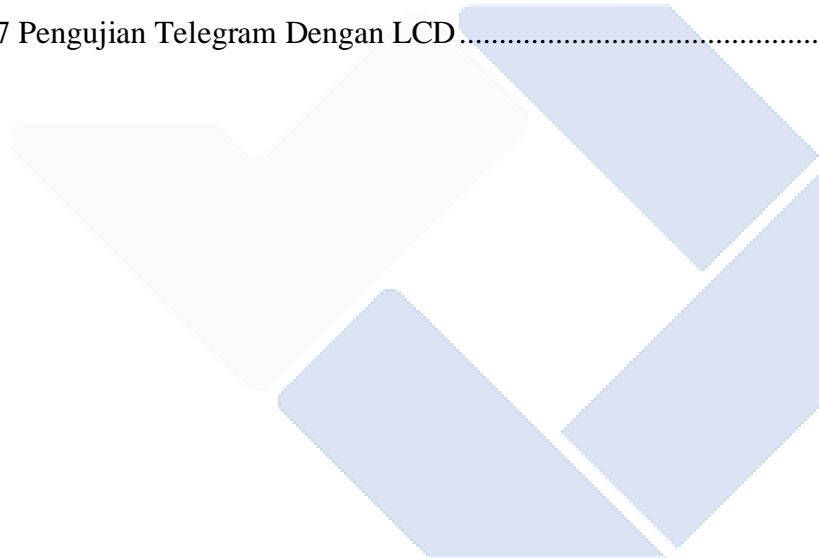
DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Proyek Akhir	3
BAB II DASAR TEORI	4
2.1 Tinjauan Pustaka	4
2.2 <i>Smart Doorbell</i>	6
2.3 Sistem Notifikasi	7
2.4 Aplikasi Telegram	7
2.5 Pengontrolan Sistem	8
2.5.1 Mikrokontroler ESP32	8
2.5.2 Mikrokontroler ESP32-CAM	9
2.6 Komunikasi ESP32 Dengan Wi-Fi	10
2.7 Komunikasi ESP32 Dengan Telegram	10
BAB III METODE PELAKSANAAN	12
3.1 Perancangan Desain	12
3.1.1 Gambaran Umum Sistem	12
3.1.2 Desain Rancangan Alat	13

3.1.3 Model Diagram Blok Sistem	15
3.1.4 Rancangan Wiring Diagram <i>Hardware</i>	16
3.1.5 <i>Flowchart</i> Sistem Kerja Alat	17
3.1.6 Rancangan Tampilan <i>Software</i>	18
3.1.7 Rancangan Pembuatan Bot Telegram.....	19
BAB IV PEMBAHASAN	23
4.1 Pembuatan <i>Hardware</i>	23
4.2 Hasil Pembuatan Bot Telegram.....	23
4.3 Pengujian.....	24
4.3.1 Prosedur Pengujian Mikrokontroler ESP32-CAM.....	24
4.3.2 Prosedur Pengujian <i>Push Button</i>	25
4.3.3 Prosedur Pengujian LCD	26
4.3.4 Hasil Pengujian ESP32-CAM Dengan Telegram.....	26
4.3.5 Hasil Pengujian Push Button Dengan Telegram.....	29
4.3.6 Hasil Pengujian Telegram Dengan LCD	31
BAB V KESIMPULAN DAN SARAN	33
5.1 Kesimpulan.....	33
5.2 Saran.....	33
DAFTAR PUSTAKA	34
DAFTAR LAMPIRAN	36

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	4
Tabel 2.2 Perbandingan Spesifikasi Mikrokontroler	9
Tabel 4.1 Prosedur Pengujian ESP32-CAM.....	24
Tabel 4.2 Prosedur Pengujian <i>Push Button</i>	25
Tabel 4.3 Prosedur Pengujian LCD	26
Tabel 4.4 Pengujian Kamera Dengan Jarak	27
Tabel 4.5 Pengujian ESP32-CAM Dengan Telegram	28
Tabel 4.6 Pengujian <i>Push Button</i> Dengan Telegram	30
Tabel 4.7 Pengujian Telegram Dengan LCD.....	32



DAFTAR GAMBAR

Gambar 2.1 Logo Telegram	8
Gambar 2.2 Mikrokontroler ESP32	9
Gambar 2.3 ESP32-CAM	10
Gambar 2.4 Komunikasi ESP32 Dengan Wi-Fi.....	10
Gambar 2.5 Komunikasi ESP32 Dengan Telegram.....	11
Gambar 3.1 Gambaran Umum Sistem	12
Gambar 3.2 Desain Kotak Kontrol.....	14
Gambar 3.3 Desain Kamera	14
Gambar 3.4 Desain Speaker	15
Gambar 3.5 Diagram Blok Sistem	15
Gambar 3.6 Wiring Diagram <i>Hardware</i>	16
Gambar 3.7 <i>Flowchart</i> Sistem Kerja Alat.....	18
Gambar 3.8 Tampilan Profil Telegram	19
Gambar 3.9 Pencarian BotFather	20
Gambar 3.10 Mulai Membuat Bot	20
Gambar 3.11 Mengirim Pesan <i>Newbot</i>	21
Gambar 3.12 Pemberian Nama Bot.....	21
Gambar 3.13 Bot Mengirimkan Token API.....	21
Gambar 3.14 Melihat Akun Bot.....	22
Gambar 3.15 Mengaktifkan Bot.....	22
Gambar 4.1 Hasil Pembuatan <i>Hardware</i>	23
Gambar 4.2 Hasil Pembuatan Bot.....	24
Gambar 4.3 Hasil Pengujian Kamera ESP32-CAM.....	27
Gambar 4.4 Hasil Pengujian <i>Push Button</i>	30
Gambar 4.5 Pengujian Kirim Pesan ke LCD	31
Gambar 4.6 Tampilan Pesan di LCD	32

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup.....	36
Lampiran 2 : Program	38
Lampiran 3 : Bukti Bukan Plagiasi	55
Lampiran 4 : Form Revisi	57
Lampiran 5 : Surat Pernyataan.....	60



BAB I

PENDAHULUAN

1.1 Latar Belakang

Ketika ada tamu atau seseorang yang ingin berkunjung ke rumah, biasanya mereka akan mengetuk pintu sebagai tanda kedatangan. Dengan kemajuan teknologi, ditemukanlah bel listrik konvensional. Bel listrik konvensional adalah sebuah perangkat yang berguna untuk memberi tahu pemilik rumah tentang kedatangan tamu melalui bunyi yang dihasilkan setelah tamu menekan tombol bel tersebut[1]. Namun, jika pemilik rumah sedang tidak berada di rumah, mereka tidak akan menyadari kehadiran tamu yang ingin berkunjung. Bel ini hanya berlaku saat pemilik rumah di rumah. Jika pemilik rumah sedang keluar maka tidak bisa mendengarkan suara bel tersebut sebagai penanda kalau ada tamu[2].

Pada beberapa kesempatan saat berada di dalam rumah, pemilik rumah seringkali sibuk menggunakan smartphone, seperti terlalu terfokus dalam bermain game dan mengakses aplikasi chatting yang bisa mengirim pesan dan bertukar informasi, contohnya scroll tiktok, whatsapp, instagram dan telegram. Aplikasi chatting ini menjadi sebuah fenomena sosial yang menyebar di berbagai kalangan, baik kalangan remaja, anak muda, maupun orang dewasa. Dampaknya, mereka lupa akan waktu dan tidak memiliki kesempatan untuk melakukan aktivitas lainnya, karena terlalu terpicat oleh penggunaan aplikasi tersebut, serta tidak menyadari kehadiran tamu yang mengetuk pintu atau menekan bel untuk mengunjungi rumah.

Saat pemilik rumah sedang keluar, tamu yang datang seperti tukang antar paket, penggalang sumbangan, atau orang asing dengan keperluan tidak tahu apakah orang yang mereka cari berada di rumah atau tidak. Ini menciptakan kendala dalam hal keamanan dan identifikasi tamu. Pemilik rumah tidak dapat memastikan apakah tamu tersebut adalah keluarga, teman, atau orang asing yang ingin menemui mereka. Jika tamu tersebut orang yang dikenal mungkin dapat menghubungi pemilik rumah melalui telepon, tetapi tidak semua orang memiliki nomor kontak

pemilik rumah.

Untuk mengatasi masalah yang telah disebutkan sebelumnya, diperlukan suatu perangkat bernama *smart doorbell*. *Smart doorbell* adalah bel pintar yang terkoneksi dengan jaringan dan memberikan pesan notifikasi ke smartphone pemilik rumah saat ada tamu berkunjung. Tujuannya untuk memberi informasi ketika ada tamu yang datang ke rumahnya. Dari penelitian sebelumnya, yang dilakukan oleh Bayu Hendra Nugroho pada tahun 2019 yang memungkinkan tamu dapat memberikan notifikasi kepada pemilik rumah melalui aplikasi telegram dengan menekan *push button*. *Doorcam* dapat berjalan dengan baik untuk sekedar mengirim gambar. Dari penelitian tersebut terdapat kelemahan yaitu hanya pemilik rumah yang mengetahui kalau ada tamu sedangkan tamu tidak mengetahui apakah pemilik rumah sedang di rumah atau sedang keluar. Maka dari itu, pengembangan kali ini menambahkan LCD sebagai alat untuk menampilkan pesan dari pemilik rumah kepada tamu yang diletakkan di depan pintu. Dengan menambahkan LCD, bertujuan supaya terjadi komunikasi dua arah antara tamu dengan pemilik rumah sehingga tamu bisa mengetahui apakah pemilik sedang di rumah atau tidak.

1.2 Rumusan Masalah

Rumusan masalah pada *Smart Doorbell Menggunakan Sistem Notifikasi Telegram* sebagai berikut:

1. Bagaimana cara merancang *smart doorbell* menggunakan sistem notifikasi telegram?
2. Bagaimana cara membuat *smart doorbell* menggunakan sistem notifikasi telegram yang bisa mengambil gambar tamu?
3. Bagaimana hasil pengujian *smart doorbell* menggunakan sistem notifikasi telegram?

1.3 Batasan Masalah

Batasan masalah dalam pembuatan *Smart Doorbell Menggunakan Sistem Notifikasi Telegram* sebagai berikut:

1. Kecepatan komunikasi antara ESP32 dengan aplikasi telegram tergantung pada jaringan internet.

2. Menggunakan kamera OV2640 dari ESP32-CAM.
3. *Smart doorbell* ini hanya bisa diterapkan di rumah yang memiliki Wi-Fi.
4. Menggunakan aplikasi platform yaitu Telegram.

1.4 Tujuan Proyek Akhir

Tujuan dari proyek akhir *Smart Doorbell* Menggunakan Sistem Notifikasi Telegram sebagai berikut:

1. Merancang *smart doorbell* menggunakan sistem notifikasi telegram dan bisa *capture* gambar tamu.
2. Mengetahui cara kerja *smart doorbell* menggunakan sistem notifikasi telegram yang bisa memberikan pesan notifikasi dan *capture* gambar tamu.



BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Tujuan tinjauan pustaka untuk pengumpulan data yang dilakukan penulis untuk mendapatkan referensi yang relevan terhadap topik penelitian yang telah dilakukan sebelumnya. Berikut ini penelitian yang telah dibuat sebelumnya.

Tabel 2.1 Penelitian Terdahulu

No.	Tahun	Judul Penelitian	Hasil Penelitian
1.	2022	<i>Smart Doorbell</i> Berbasis Internet Of Thing	Penelitian dilakukan oleh Elvira Sagita Liansyah mahasiswa dari Universitas Telkom Bandung yaitu menggunakan sensor <i>Passive Infra Red</i> (PIR) sebagai pendeteksi tamu serta mengirim gambar tamu ke telegram pemilik rumah. Hasil penelitiannya pemilik rumah dapat menghemat waktu dengan tidak perlu secara langsung melihat siapa yang di depan pintu dengan melalui smartphone. Mereka juga dapat menerima notifikasi melalui aplikasi Telegram untuk mengetahui berat paket yang diantar oleh kurir yang berada di depan rumah[3].
2.	2019	Rancang Bangun <i>Doorbell Wireless</i> Berbasis IoT	Penelitian yang dilakukan oleh Ade Aprillia mahasiswa Politeknik Negeri Sriwijaya. Penelitian ini pakai sensor <i>Infrared Proximity</i> untuk mendeteksi

			kehadiran tamu dan menggunakan Arduino Mega2560 serta kamera serial VC0706 untuk mengirimkan notifikasi dan gambar tamu ke aplikasi telegram[4].
3.	2021	Rancang Bangun Sistem Pemantauan Kedatangan Tamu Berbasis Internet Of Things (IOT)	Penelitian yang dilakukan Plasida Arri Ape Pane Basabilik mahasiswa dari Universitas Tanjungpura. Penelitian yang dilakukan menggunakan <i>infrared sensor</i> untuk mendeteksi dengan jangkauan 0.5-3 cm dan bekerja sebagai <i>button</i> tanpa sentuh serta ESP32-CAM <i>capture</i> gambar kemudian dikirim ke telegram[5].
4.	2023	Rancang Bangun Sistem Keamanan Rumah Berbasis Telegram menggunakan ESP32-CAM	Penelitian yang dilakukan Ardiansyah dan kawan-kawan mahasiswa Teknik Elektro Universitas Muhammadiyah Makassar. Penelitian mereka dengan menggunakan <i>sensor passive infrared receiver</i> sebagai alarm , sensor PIR setiap mendeteksi gerakan dan ESP32-CAM untuk mengambil gambar tamu serta dikirim ke telegram[6].
5.	2022	Sistem Keamanan Pintu Rumah Berbasis Internet Of Things Via Pesan Telegram	Penelitian yang dilakukan oleh Jaenal Arifin dan kawan-kawan merupakan mahasiswa Institut Teknologi Telkom Purwakarto. Penelitian yang mereka lakukan menggunakan mikrokontroler Wemos D1 R1 dan menggunakan <i>proximity sensor</i> sebagai pendeteksi

			gerakan serta mengirimkan gambar ke telegram pemilik rumah[7].
--	--	--	--

Dengan menyimpulkan hasil dari penelitian sebelumnya, terlihat bahwa prinsip kerja alatnya hampir sama satu dengan yang lainnya, hanya saja komponen yang digunakan berbeda. Dalam konteks ini, penulis berupaya untuk menambahkan LCD sebagai alat untuk menampilkan pesan dari pemilik rumah kepada tamu yang diletakkan di depan pintu. Terlihat dari penelitian sebelumnya, kebanyakan tamu memberikan notifikasi berupa pesan dan gambar tamu yang dikirim ke salah satu platform seperti Telegram. Dengan menambahkan LCD, bertujuan supaya terjadi komunikasi dua arah antara tamu dengan pemilik rumah sehingga tamu bisa mengetahui apakah pemilik sedang di rumah atau tidak.

2.2 *Smart Doorbell*

Bel pintu pintar, juga dikenal dengan *smart doorbell* merupakan salah satu perangkat elektronik yang terhubung ke jaringan nirkabel dan digunakan untuk memberikan informasi ke pemilik rumah ketika ada tamu yang menekan *push button*[8]. *Smart doorbell* biasanya dilengkapi dengan kamera, *push button*, speaker, dan aplikasi pesan lintas platform. Dibawah ini adalah penjelasan lebih rinci tentang komponen dan fitur utama *smart doorbell* :

- Kamera: *Smart doorbell* biasanya menggunakan kamera supaya pemilik rumah dapat melihat gambar ataupun video secara langsung dari pintu depan melalui notifikasi aplikasi di smartphone. Fitur ini memungkinkan pengguna untuk melihat pengunjung ataupun memantau siapa tamu yang datang ke rumah tanpa harus berada di tempat.
- *Push button*: *Smart doorbell* menggunakan *push button* untuk memberikan sinyal aktivitas di depan pintu. Ketika *push button* ditekan, *smart doorbell* akan mengirimkan pemberitahuan langsung kepada pengguna melalui aplikasi. Hal ini berguna untuk memberi tahu pengguna jika ada tamu, kurir, atau orang yang tidak diinginkan di depan pintu masuk.
- Penyimpanan data: *smart doorbell* memiliki penyimpanan data internal

serta mendukung penyimpanan *cloud*. Hal ini memudahkan pengguna untuk mengambil gambar atau video interaksi setelah masuk dan menyimpannya untuk digunakan atau dipantau nanti.

2.3 Sistem Notifikasi

Notifikasi sistem digunakan untuk memberi tahu pengguna mengenai informasi penting yang terjadi dalam suatu sistem atau aplikasi. Pengguna dapat menerima notifikasi ini sebagai pemberitahuan atau peringatan. Sistem notifikasi seringkali digunakan dalam berbagai jenis aplikasi dan platform, termasuk aplikasi seluler, situs web, media sosial, dan alat komunikasi lainnya. Sebagai contoh, dalam aplikasi pesan instan, notifikasi digunakan untuk memberitahu pengguna tentang pesan baru yang diterima. Pada platform media sosial, notifikasi dapat menginformasikan pengguna mengenai komentar, like, atau pembaruan status dari teman atau hal-hal yang relevan dengan mereka[9].

2.4 Aplikasi Telegram

Aplikasi telegram tersedia di berbagai platform, termasuk perangkat seluler (iOS dan Android), desktop, dan web. Salah satu keunggulan telegram adalah kemampuannya dalam menyediakan API (*Application Programming Interface*) yang dapat digunakan. Salah satu antarmuka pemrograman aplikasi (API) yang dapat digunakan adalah fitur bot. Dalam konteks proyek Internet of Things (IoT), bot telegram digunakan. [10].

Pada proyek akhir ini, menggunakan aplikasi telegram sebagai platform karena telegram memiliki beberapa keunggulan jika dibandingkan dengan aplikasi lainnya. Berikut keunggulan aplikasi telegram:

- Aplikasi telegram sederhana dan sangat mudah untuk digunakan.
- Pengiriman pesan lebih cepat.
- Memiliki keamanan yang baik daripada aplikasi Whatsapp.
- Telegram bisa membuat grub hingga 200.000 anggota.
- Aplikasi telegram gratis tanpa adanya iklan.
- Telegram juga menyediakan fitur channel.
- Kirim file ukuran besar 2 GB sedangkan Whatsapp tidak lebih 64 MB[11].



Gambar 2.1 Logo Telegram[10]

2.5 Pengontrolan Sistem

Pengontrolan sistem *smart doorbell* menggunakan sistem notifikasi telegram menggunakan mikrokontroler ESP32 dan ESP32-CAM . Mikrokontroler ini digunakan untuk mengoneksikan antara *software* dengan *hardware* melalui jaringan internet. Berikut ini sistem pengontrolan *smart doorbell* menggunakan sistem notifikasi telegram.

2.5.1 Mikrokontroler ESP32

Mikrokontroler ESP32 digunakan karena mempunyai beberapa keunggulan jika dibandingkan dengan mikrokontroler lainnya, salah satunya sudah tertanam modul Wi-Fi dalam *chip* dan telah memiliki bluetooth mode ganda di dalamnya. Selain itu, Mikrokontroler ESP32 juga memiliki kelebihan termasuk harganya murah juga berdaya rendah. Kelebihan mikrokontroler ESP32 lainnya yaitu pinout dan pin analog lebih banyak serta memori lebih besar, sehingga untuk mengaplikasikan *internet of things* lebih fleksibel[12]. Pada gambar 2.2 tampilan ESP32.



Gambar 2.2 Mikrokontroler ESP32[12]

Pada proyek akhir ini lebih menggunakan mikrokontroler ESP32 karena memiliki beberapa kelebihan dibandingkan dengan mikrokontroler lainnya seperti Arduino Uno dan Node MCU(ESP8266). Berikut perbedaan ESP32 dengan mikrokontroler lainnya terdapat pada tabel 2.2

Tabel 2.2 Perbandingan Spesifikasi Mikrokontroler

Spesifikasi	Arduino Uno	ESP8266	ESP32
CPU	Atmega328 - 16MHz	Xtensa single core L106 – 60 MHz	Xtensa dual core LX6 - 160 MHz
Tegangan	5 volt	3,3 volt	3,3 volt
Arsitektur	8 bit	32 bit	32 bit
SRAM	2kB	160 kB	512kB
Flash Memory	32kB	16MB	16MB
GPIO PIN (ADC/DAC)	14 (6/-)	17(1/-)	36 (18/2)
SPI/12C/UART	1/1/1	2/1/2	4/2/2
Wi-Fi	Tidak Ada	Ada	Ada
Bluetooth	Tidak Ada	Tidak Ada	Ada

2.5.2 Mikrokontroler ESP32-CAM

Mikrokontroler ESP32-CAM lebih digunakan karena memiliki kemampuan yang kuat karena menggabungkan ESP32, mikrokontroler *dual-core* yang cepat, dengan kemampuan pemrosesan gambar yang dilengkapi kamera OV2640. Modul ini sudah memiliki dukungan Wi-Fi dan Bluetooth, yang bisa komunikasi nirkabel dengan perangkat lain[12].

Dalam konteks *smart doorbell* ini, ESP32-CAM dapat digunakan untuk

mengambil gambar tamu yang berkunjung setelah tamu menekan bel melalui kamera OV2640. Modul ini juga dapat terhubung ke jaringan Wi-Fi untuk mengirimkan notifikasi ke perangkat lain, seperti smartphone pengguna. Pada gambar 2.3 ESP32-CAM.



Gambar 2.3 ESP32-CAM[12]

2.6 Komunikasi Mikrokontroler ESP32 Dengan Wi-Fi

Komunikasi Wi-Fi dengan mikrokontroler ESP32, digunakan sebagai penghubung antara ESP32 dengan Wi-Fi agar data dapat terkirim ke telegram berupa pesan notifikasi. Berikut ini *source code* komunikasi mikrokontroler ESP32 dengan Wi-Fi.

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// Wifi network station credentials
#define WIFI_SSID "Catlover"
#define WIFI_PASSWORD "daktaukubenerla"
```

Gambar 2.4 Komunikasi ESP32 Dengan Wi-Fi

2.7 Komunikasi Mikrokontroler ESP32 Dengan Telegram

Tujuan komunikasi antara mikrokontroler ESP32 dan telegram adalah untuk mengirimkan data yang diperoleh dari ESP32 ke Telegram. Hal ini dicapai dengan mengkonfigurasi mikrokontroler menggunakan token dan ID bot telegram sehingga mikrokontroler dapat berkomunikasi dengan telegram dan mengirimkan data ke

platform tersebut. Berikut ini *source code* komunikasi mikrokontroler ESP32 dengan telegram.

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// Telegram BOT Token (Get from Botfather)
#define BOT_TOKEN "6261729633:AAGSJOIX8kelgVxy4X0VTLpZfEcAPOs94hc"
#define CHAT_ID "1269004513"
```

Gambar 2.5 Komunikasi ESP32 Dengan Telegram



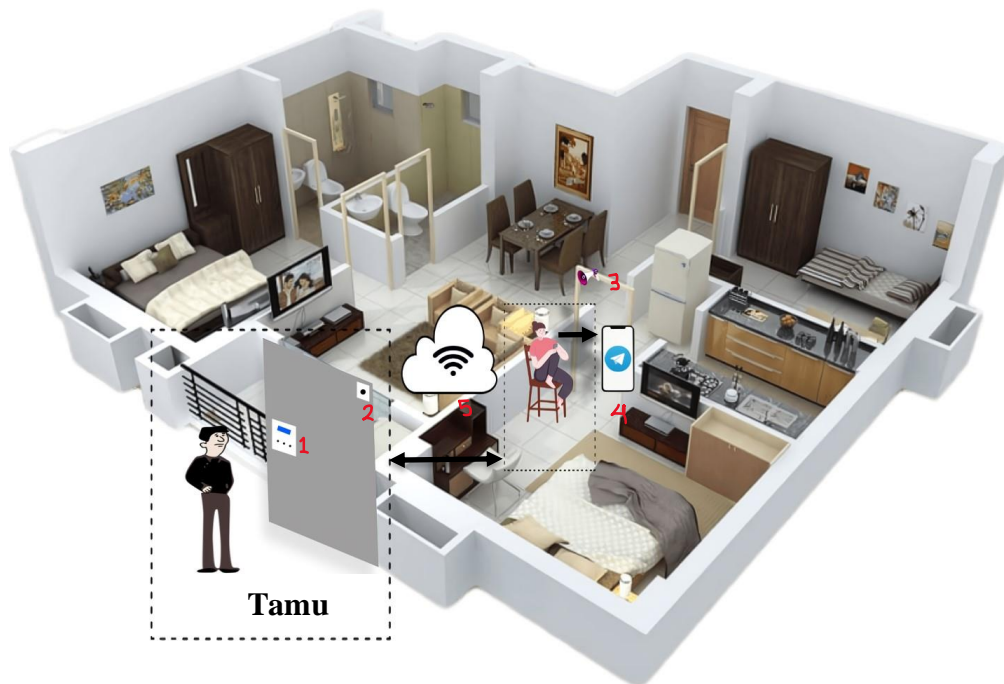
BAB III METODE PELAKSANAAN

3.1 Perancangan Desain

Proses perencanaan desain melibatkan identifikasi, pengembangan, dan penyusunan konsep-konsep awal untuk menciptakan desain yang efektif dan berfungsi. Tujuannya adalah untuk membantu memastikan bahwa desain yang dihasilkan sesuai dengan tujuan yang ditetapkan.

3.1.1 Gambaran Umum Sistem

Perancangan sistem *smart doorbell* menggunakan sistem notifikasi telegram yang dapat mengirimkan pesan notifikasi dan mengambil data berupa gambar tamu. Pada gambar 3. merupakan gambaran umum *smart doorbell* menggunakan sistem notifikasi telegram.



Gambar 3.1 Gambaran Umum Sistem

Dari gambar 3.1 terlihat dua individu, salah satunya merupakan tamu dan yang lainnya adalah pemilik rumah. Ketika tamu menekan *push button*, notifikasi akan muncul pada smartphone pemilik rumah yang menggunakan sistem operasi Android. Pemilik rumah bisa mengambil gambar tamu melalui aplikasi Telegram dan mengirim pesan teks kepada tamu, baik untuk mengizinkan tamu masuk atau meminta tamu untuk menunggu sebentar.

Keterangan gambar :

1. Junction box yang berisi komponen seperti mikrokontroler ESP32, dfplayer mini, *push button*, LED, dan LCD. Mikrokontroler ESP32 digunakan untuk sistem pengontrolan, dfplayer mini digunakan untuk memutar file Mp3 untuk speaker, *push button* digunakan untuk mengaktifkan bel dan memberikan notifikasi, LED sebagai lampu indikator ketika *push button* ditekan, serta LCD digunakan untuk menampilkan pesan teks.
2. Kamera ESP32-CAM tipe OV2640 untuk mengambil gambar tamu.
3. Speaker untuk membunyikan bel.
4. Smartphone untuk menampilkan notifikasi pesan ke telegram.
5. Internet sebagai penghubung antara mikrokontroler dengan smartphone untuk mentransfer data yang berisi pesan teks dan gambar.

3.1.2 Desain Rancangan Alat

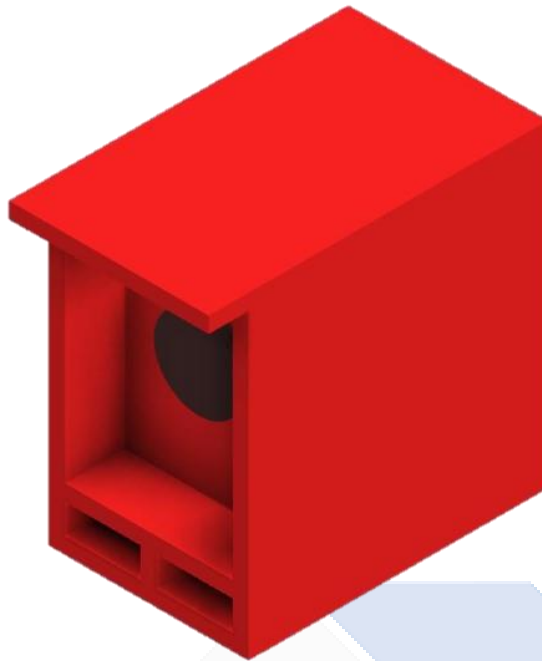
Push button dan LCD diletakkan pada *Junction Box* didepan pintu tujuannya untuk memudahkan tamu menekan *push button*. Ketika ada tamu berkunjung dan menekan *push button* sehingga speaker berbunyi sekaligus memberikan notifikasi pesan ke telegram pemilik rumah. Pemilik rumah bisa mengambil gambar siapa tamu yang berkunjung. Selain itu, pemilik rumah bisa mengirimkan pesan dari telegram untuk ditampilkan dilayar LCD depan pintu sehingga tamu mengetahui. Begitu juga tamu bisa mengirimkan pesan dengan memanfaatkan *push button* yang lainnya untuk memberikan pesan ke telegram pemilik rumah. Hal ini bisa komunikasi dua arah antara tamu dengan pemilik rumah. Berikut ini desain rancangan alat *smart doorbell* menggunakan sistem notifikasi telegram.



Gambar 3.2 Desain Kotak Kontrol



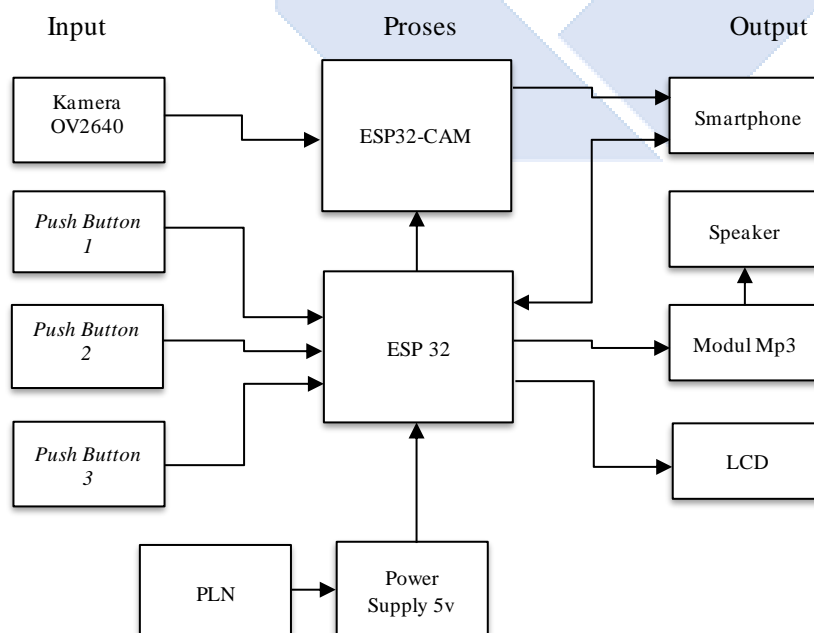
Gambar 3.3 Desain Kamera



Gambar 3.4 Desain Speaker

3.1.3 Model Diagram Blok Sistem

Blok diagram diperlukan untuk menentukan tahapan dari prinsip kerja *smart doorbell* menggunakan sistem notifikasi telegram. Berikut ini blok diagram terdapat pada gambar 3.5.

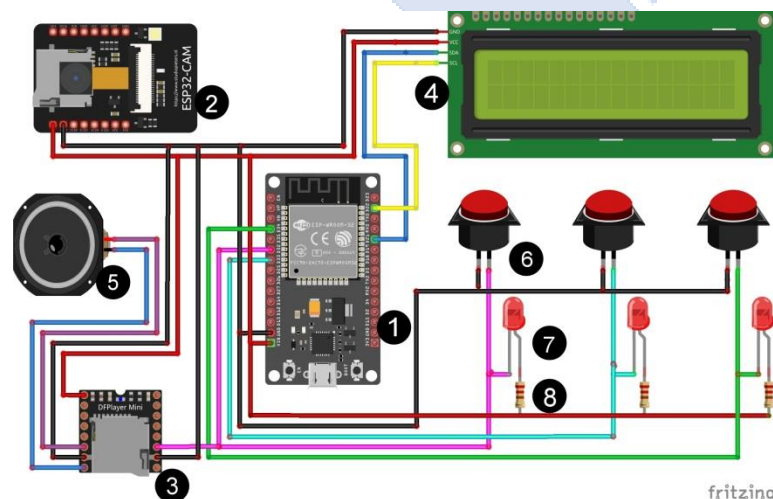


Gambar 3.5 Diagram Blok Sistem

Berdasarkan blok diagram di atas, dapat diketahui bahwa rangkaian keseluruhan dirancang dalam blok diagram perancangan cara kerja. Bagian input terdiri dari tiga *push button* dan kamera OV2640. Sedangkan bagian output terdiri dari dfplayer mini, speaker, LCD 16x2, dan smartphone. Mikrokontroler ESP32 dan ESP32-CAM digunakan untuk proses *input* data. Cara kerja alat ini yaitu ketika *push button* 1 ditekan, mikrokontroler ESP32 kemudian mengirimkan data ke dfplayer mini dan mengaktifkan speaker, serta memberikan notifikasi ke smartphone pemilik rumah dengan tulisan “**Ada Tamu di depan Pintu**”. Pemilik rumah dapat mengambil gambar tamu menggunakan modul kamera ESP32-CAM tipe OV2640. Pemilik rumah juga dapat memberikan pesan dari smartphone kepada tamu yang akan ditampilkan pada layar LCD depan pintu. Selain itu, tamu dapat mengirimkan pesan otomatis melalui dua *push button* lainnya, sehingga pemilik rumah dan tamu dapat berkomunikasi dua arah.

3.1.4 Rancangan Wiring Diagram Hardware

Perancangan pada rangkaian *hardware smart doorbell* menggunakan sistem notifikasi telegram dilakukan dengan cara menyusun rangkaian pada komponen-komponen yang digunakan, seperti ESP32, ESP32-CAM, LCD 16 X 2, Dfplayer mini, Speaker, *Push button*, Resistor, dan LED. Rangkaian ini dibuat dengan menggunakan *software* Fritzing. Berikut ini wiring diagram *hardware*:



Gambar 3.6 Wiring Diagram Hardware

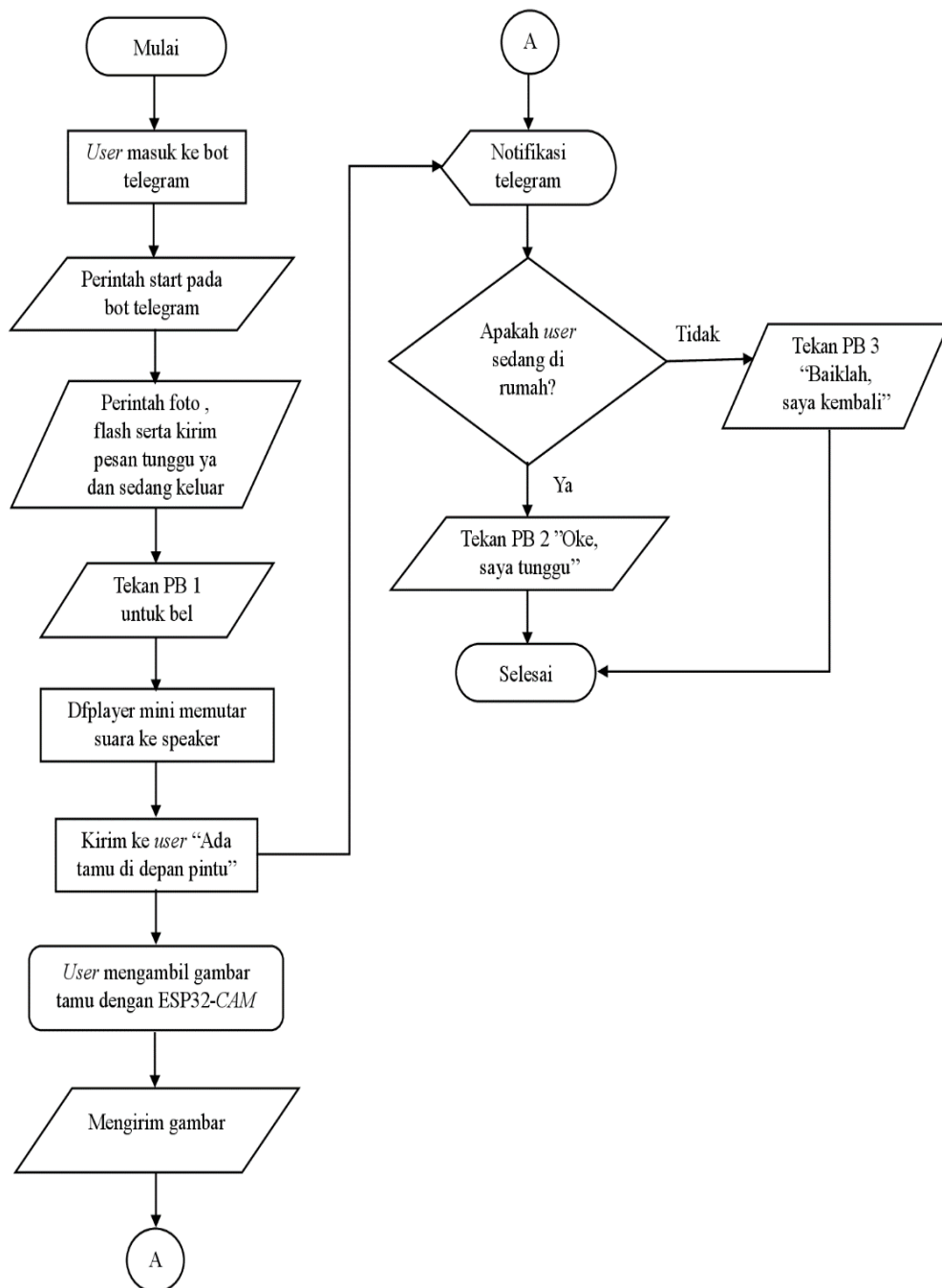
Keterangan:

1. ESP32
2. ESP32-CAM
3. Dfplayer Mini
4. LCD 16 X 2
5. Speaker
6. *Push Button*
7. LED
8. Resistor

Sumber diberikan untuk menyuplai tegangan ke komponen mikrokontroler ESP32 dan ESP32-CAM. Saat *push button* ditekan, lampu indikator akan menyala dan memberikan sinyal data ke mikrokontroler ESP32, serta menyalakan dfplayer mini agar memutar file suara melalui speaker. Pengambilan gambar tamu oleh pemilik rumah dilakukan melalui modul ESP32-CAM setelah tamu menekan *push button* dan mengirimkan pesan untuk tamu yang akan ditampilkan di layar LCD. Dengan demikian, sistem ini dapat beroperasi secara efektif.

3.1.5 *Flowchart* Sistem Kerja Alat

Berikut ini *flowchart* sistem kerja alat *smart doorbell* menggunakan sistem notifikasi telegram.



Gambar 3.7 Flowchart Sistem Kerja Alat

3.1.6 Rancangan Tampilan Software

Pada proyek ini menggunakan platform telegram untuk memberikan notifikasi ke smartphone pemilik rumah. Berikut ini rancangan tampilan aplikasi telegram.

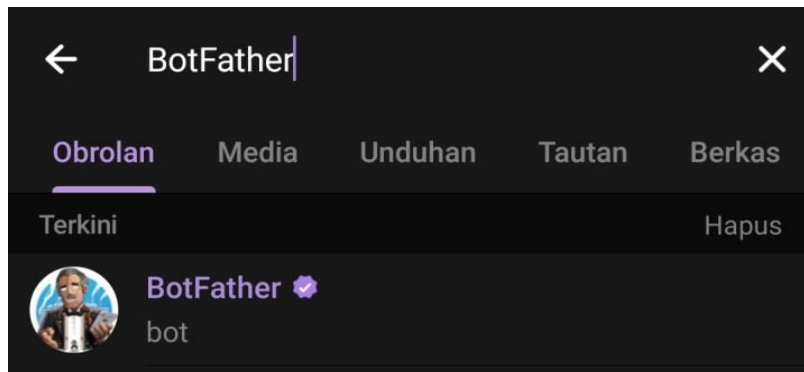


Gambar 3.8 Tampilan Profil Telegram

3.1.7 Rancangan Pembuatan Bot Telegram

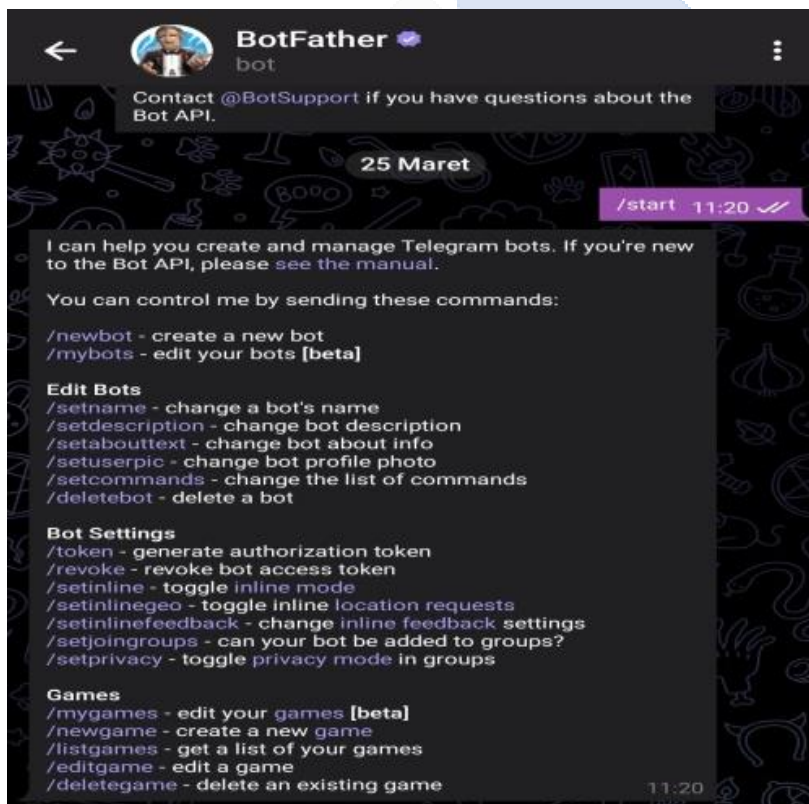
BotFather adalah tokoh utama dalam dunia bot Telegram yang bertanggung jawab atas pengaturan dan manajemen untuk bot pada platform tersebut. Istilah lain, BotFather merupakan sebuah bot khusus yang memberikan pengguna akses ke menu pengaturan untuk mengelola bot yang telah mereka buat. Jadi, jika Anda memiliki bot Telegram, Anda perlu menggunakan BotFather sebagai alat untuk mengatur dan mengelolanya. Berikut ini prosedur dalam membuat bot:

1. Carilah "botfather" pada kotak pencarian pada telegram.



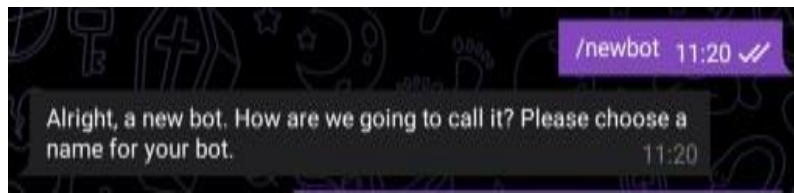
Gambar 3.9 Pencarian BotFather

2. Untuk mulai membuat bot pada telegram harus kirim pesan `/start` terlebih dahulu. BotFather mengirimkan beberapa fitur yang dapat digunakan.



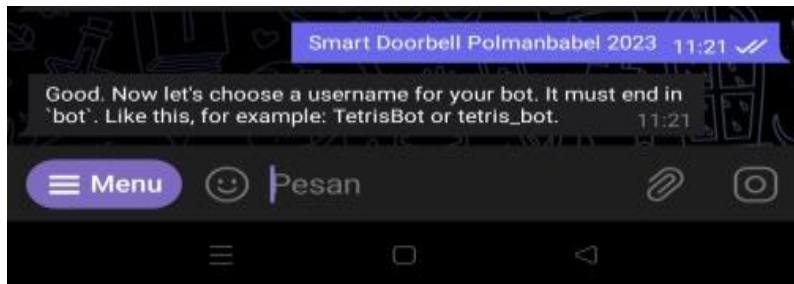
Gambar 3.10 Mulai Membuat Bot

3. Klik pesan `/newbot` untuk membuat bot yang baru. BotFather akan memberikan pesan untuk melihat nama bot yang dibuat.



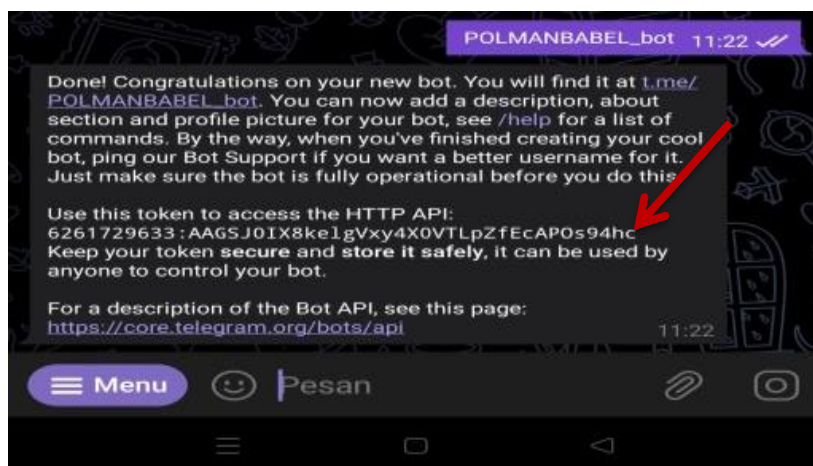
Gambar 3.11 Mengirim Pesan *Newbot*

4. Kemudian buatlah nama bot yang sesuai keinginan.



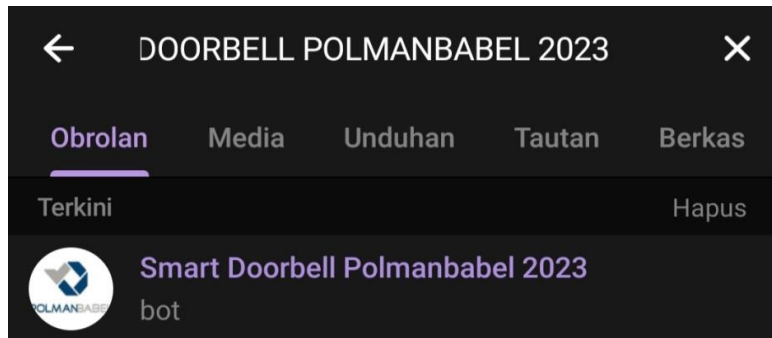
Gambar 3.12 Pemberian Nama Bot

5. Setelah berhasil membuat bot, Anda akan mendapatkan token yang dapat digunakan untuk mengakses API bot pada platform Telegram. API merupakan antarmuka yang memungkinkan interaksi dengan aplikasi tersebut. Umumnya, para programmer menggunakan API ini untuk mengakses fungsi-fungsi program lain yang mereka buat. Setelah itu, dapat melihat token API tersebut dengan merujuk pada Gambar 3.13 yang ditandai dengan centang merah.



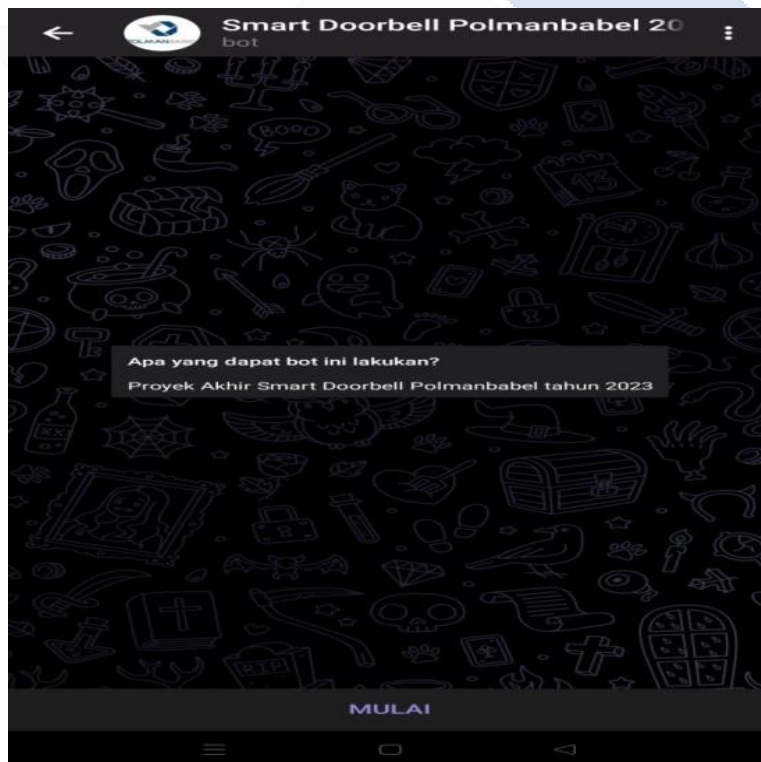
Gambar 3.13 Bot Mengirimkan Token API

- Selanjutnya klik pencarian untuk melihat bot yang sudah dibuat tadi apakah sudah ada atau belum.



Gambar 3.14 Melihat Akun Bot

- Setelah bot sudah bisa digunakan dengan klik “Mulai”.



Gambar 3.15 Mengaktifkan Bot

BAB IV PEMBAHASAN

4.1 Pembuatan *Hardware*

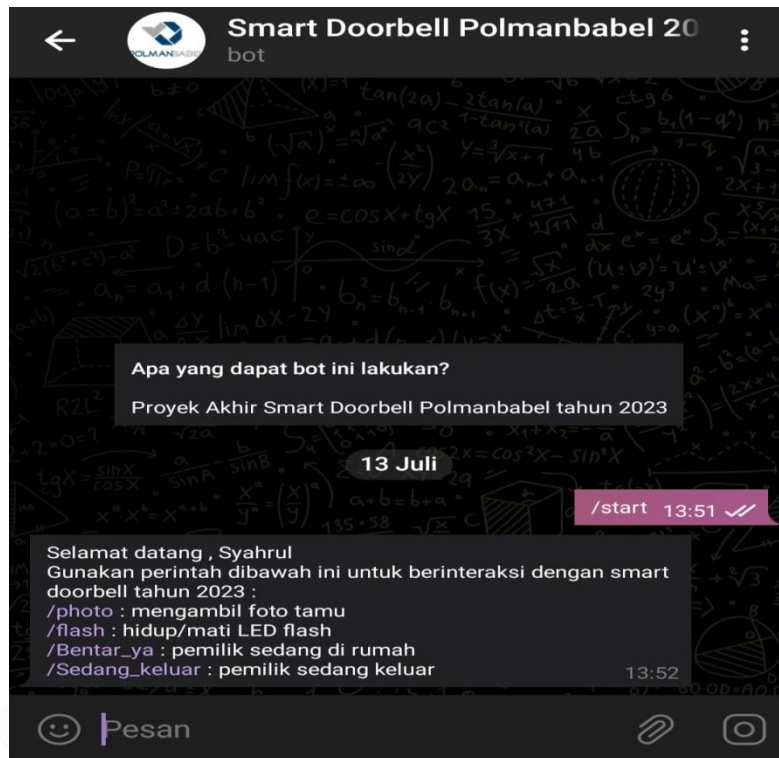
Berikut ini hasil pembuatan *hardware* pada proyek akhir *smart doorbell* menggunakan sistem notifikasi telegram.



Gambar 4.1 Hasil Pembuatan *Hardware*

4.2 Hasil Pembuatan Bot Telegram

Berikut ini hasil pembuatan pada bot telegram terhadap proyek akhir *smart doorbell* menggunakan sistem notifikasi telegram.



Gambar 4.2 Hasil Pembuatan Bot

4.3 Pengujian

Pada tahapan ini, melakukan pengujian alat pada *smart doorbell* yang menggunakan sistem notifikasi telegram berdasarkan rancangan yang telah disusun sebelumnya. Tujuannya untuk membenarkan bahwa proyek akhir *smart doorbell* menggunakan sistem notifikasi telegram berjalan dengan baik sesuai yang diharapkan, dan untuk mengidentifikasi apakah ada masalah seperti kesalahan sistem atau perangkat yang bermasalah. Hasil pengujian ini akan dievaluasi dan dianalisis.

4.3.1 Prosedur Pengujian Mikrokontroler ESP32-CAM

Tabel 4.1 Prosedur Pengujian ESP32-CAM

Komponen	Prosedur	Kondisi	Hasil
Mikrokontroler ESP32-CAM	Lakukan uji coba ESP32-CAM menggunakan kode bawaan dari	Mikrokontroler ESP32-CAM siap digunakan dan dapat diprogram	Dapat menampilkan gambar secara streaming tetapi

	<i>library</i> ESP32 pada <i>software</i> Arduino IDE	dengan <i>software</i> Arduino IDE.	dengan sedikit delay saat diakses di web menggunakan kode bawaan <i>library</i> ESP32.
--	---	-------------------------------------	--

Tujuan pengujian pada ESP32-CAM adalah untuk memeriksa apakah modul ESP32-CAM beroperasi dengan baik dan apakah dapat digunakan dan diprogram. Jika pengujian berhasil, maka modul ESP32-CAM dapat dianggap siap untuk digunakan dan diprogram dalam proyek *Smart Doorbell* Menggunakan Sistem Notifikasi Telegram.

4.3.2 Prosedur Pengujian *Push Button*

Tabel 4.2 Prosedur Pengujian *Push Button*

Komponen	Prosedur	Kondisi	Hasil
<i>Push Button</i>	Menguji fungsi <i>push button</i> menggunakan contoh program pada <i>software</i> Arduino IDE.	<i>Push button</i> siap digunakan dan dapat diprogram menggunakan mikrokontroler dengan <i>software</i> Arduino IDE.	<i>Source code</i> <i>example</i> pada <i>software</i> Arduino IDE berhasil mendeteksi <i>push button</i> dan menampilkan informasinya melalui serial monitor.

Tujuan pengujian pada *push button* adalah untuk memeriksa apakah modul *push button* beroperasi dengan baik dan apakah dapat digunakan dan diprogram. Jika pengujian berhasil, maka *push button* dapat dianggap siap untuk digunakan dan diprogram dalam proyek *Smart Doorbell* Menggunakan Sistem Notifikasi

Telegram.

4.3.3 Prosedur Pengujian LCD

Tabel 4.3 Prosedur Pengujian LCD

Komponen	Prosedur	Kondisi	Hasil
LCD	pengujian fungsi LCD menggunakan contoh program dengan <i>software</i> Arduino IDE.	LCD dalam kondisi baik dan digunakan serta dapat diprogram menggunakan mikrokontroler dengan <i>software</i> Arduino IDE.	Bisa menampilkan pesan ditampilkan pada layar LCD setelah diprogram dengan <i>software</i> Arduino IDE.

Tujuan pengujian pada LCD adalah untuk memeriksa apakah modul LCD dapat beroperasi dengan baik dan bisa digunakan dan diprogram. Jika pengujian berhasil, maka LCD dapat dianggap siap untuk digunakan dan diprogram dalam proyek *Smart Doorbell* Menggunakan Sistem Notifikasi Telegram.

Di bagian ini, informasi hasil pengujian dari *smart doorbell* menggunakan sistem notifikasi telegram akan dikumpulkan dalam bentuk catatan waktu yang diperoleh dengan menjalankan perintah sesuai keinginan.

4.3.4 Hasil Pengujian ESP32-CAM Dengan Telegram



Hasil pengujian kamera pada gambar dibawah ini menunjukkan efektivitas penggunaan kamera OV2640 pada ESP32-CAM yang memiliki resolusi kamera 2 Megapiksel. Proses pengambilan gambar dimulai setelah tamu menekan push button, yang kemudian mengirim notifikasi pesan. Setelah itu, pemilik rumah dapat mengambil gambar tamu tersebut menggunakan perintah bot telegram.





Gambar 4.3 Hasil Pengujian Kamera ESP32-CAM

Berikut ini hasil pengujian kamera ESP32-CAM dengan jarak pengambilan gambar terdapat pada tabel 4.4.

Tabel 4.4 Pengujian Kamera Dengan Jarak

Percobaan Ke-	Hasil Gambar	Jarak	Lama Respon
1		1 meter	6 detik
2		2 meter	7 detik

3		3 meter	5 detik
4		4 meter	8 detik
5		5 meter	7 detik

Berdasarkan jarak pengambilan gambar di atas dapat disimpulkan bahwa semakin jauh kamera dengan objek maka semakin buram hasil pengambilan gambar.

Berikut ini hasil percobaan waktu mengambil gambar dan waktu kirim gambar ke telegram terdapat pada tabel 4.5.

Tabel 4.5 Pengujian ESP32-CAM Dengan Telegram

Percobaan ke-	Waktu Mengambil Gambar	Waktu Kirim Gambar	Delay
1	12 : 10 : 04	12 : 10 : 10	6 detik
2	12 : 11 : 49	12 : 11 : 54	5 detik
3	12 : 12 : 29	12 : 12 : 34	5 detik
4	12 : 13 : 17	12 : 13 : 23	6 detik
5	12 : 14 : 52	12 : 14 : 59	7 detik
6	12 : 15 : 33	12 : 15 : 39	6 detik
7	12 : 16 : 13	12 : 16 : 21	8 detik
8	12 : 16 : 54	12 : 17 : 03	9 detik

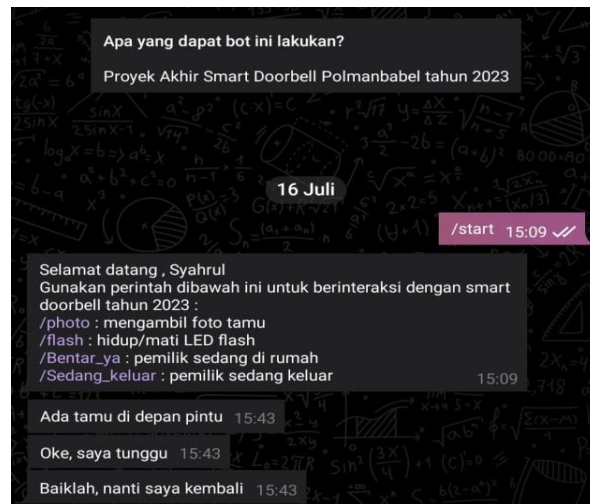
9	12 : 17 : 46	12 : 17 : 54	8 detik
10	12 : 19 : 05	12 : 19 : 13	8 detik
11	12 : 20 : 32	12 : 20 : 38	6 detik
12	12 : 21 : 16	12 : 21 : 22	6 detik
13	12 : 21 : 51	12 : 22 : 01	10 detik
14	12 : 22 : 49	12 : 22 : 56	7 detik
15	12 : 23 : 40	12 : 23 : 54	14 detik
16	12 : 24 : 41	12 : 24 : 50	9 detik
17	12 : 25 : 40	12 : 25 : 47	7 detik
18	12 : 26 : 15	12 : 26 : 22	7 detik
19	12 : 27 : 14	12 : 27 : 22	8 detik
20	12 : 27 : 59	12 : 28 : 09	10 detik

Berdasarkan hasil pengujian kamera ESP32-CAM dalam mengirim gambar ke Telegram, data pada tabel menunjukkan bahwa waktu yang diperlukan untuk mengambil dan mengirim gambar berbeda-beda. Perintah */photo* yang dikirim melalui Telegram diterima oleh Wi-Fi ESP32-CAM mengalami *delay*. Secara rata-rata, waktu *delay* yang dibutuhkan dari pengambilan gambar hingga pengiriman gambar melalui Telegram adalah 7,6 detik.

Salah satu tantangan dalam pengambilan dan pengiriman gambar adalah ketidakstabilan akses kamera akibat gangguan jaringan internet yang terkadang menyebabkan kamera terputus.

4.3.5 Hasil Pengujian *Push Button* Dengan Telegram

Hasil pengujian ketiga *push button* yang dikirimkan melalui telegram telah terbukti efektif. Jika *push button* 1 ditekan, notifikasi yang dikirimkan adalah "Ada tamu di depan pintu". *Push button* 2 digunakan untuk membalas pesan dari pemilik rumah ketika berada di dalam rumah dengan pesan "Oke, saya tunggu". Sedangkan, *push button* 3 digunakan untuk membalas pesan dari pemilik rumah ketika pemilik sedang keluar dengan pesan "Baiklah, nanti saya kembali". Berikut adalah hasil pengujian notifikasi yang muncul saat *push button* ditekan.



Gambar 4.4 Hasil Pengujian *Push Button*

Berikut ini hasil percobaan ketika *push button* dalam kondisi ditekan dan tidak ditekan pada tabel 4.6

Tabel 4.6 Pengujian *Push Button* Dengan Telegram

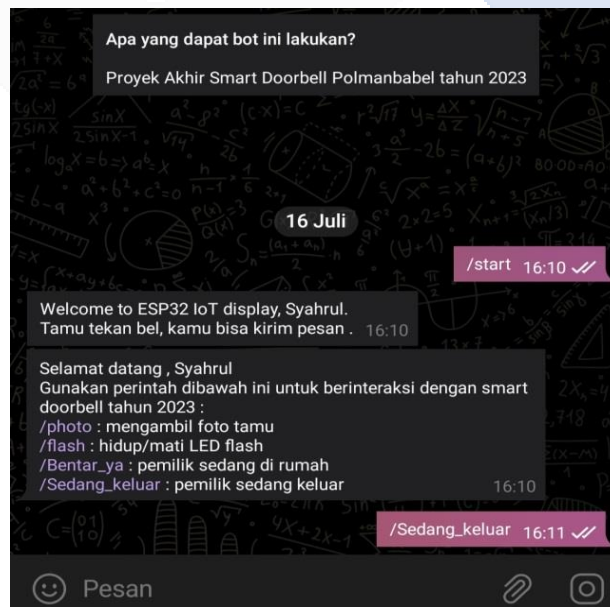
Percobaan ke-	Status	Status <i>Push Button</i>	Hasil	Waktu
1	Ditekan	Sumber terhubung dengan ESP32	Speaker berbunyi dan kirim notifikasi pesan ke telegram	<i>Real Time</i>
2	Tidak ditekan	Sumber terhubung dengan ESP32	Speaker tidak berbunyi dan tidak mengirimkan pesan notifikasi ke telegram	<i>Real Time</i>
3	Ditekan	Sumber terhubung dengan ESP32	Speaker berbunyi dan kirim notifikasi pesan ke telegram	<i>Real Time</i>
4	Tidak ditekan	Sumber terhubung dengan ESP32	Speaker tidak berbunyi dan tidak mengirimkan	<i>Real Time</i>

			pesan notifikasi ke telegram	
5	Ditekan	Sumber terhubung dengan ESP32	Speaker berbunyi dan kirim notifikasi pesan ke telegram	<i>Real Time</i>

Berdasarkan tabel hasil pengujian *push button* pada *smart doorbell* menggunakan sistem notifikasi telegram tidak ditemukan kendala. Saat *push button* ditekan berhasil menghidupkan speaker serta dapat mengirimkan pesan notifikasi ke telegram secara *real time*.

4.3.6 Hasil Pengujian Telegram Dengan LCD

Hasil pengujian kirim pesan dari telegram ke LCD adalah sebagai berikut.



Gambar 4.5 Pengujian Kirim Pesan ke LCD



Gambar 4.6 Tampilan Pesan di LCD

Berikut ini hasil percobaan waktu kirim pesan dan waktu pesan terkirim ke telegram terdapat pada tabel 4.7.

Tabel 4.7 Pengujian Telegram Dengan LCD

Percobaan ke-	Waktu Kirim Pesan	Waktu Pesan Terkirim	Delay
1	11 : 01 : 48	11 : 01 : 50	2 detik
2	11 : 05 : 56	11 : 05 : 58	2 detik
3	11 : 06 : 48	11 : 06 : 51	3 detik
4	11 : 07 : 35	11 : 07 : 38	3 detik
5	11 : 08 : 33	11 : 08 : 35	2 detik
6	11 : 15 : 01	11 : 15 : 05	4 detik
7	11 : 15 : 33	11 : 15 : 36	3 detik
8	11 : 16 : 20	11 : 16 : 23	3 detik
9	11 : 16 : 51	11 : 16 : 53	2 detik
10	11 : 17 : 39	11 : 17 : 43	4 detik
11	11 : 18 : 17	11 : 18 : 19	2 detik
12	11 : 19 : 02	11 : 19 : 06	4 detik
13	11 : 20 : 18	11 : 20 : 20	2 detik
14	11 : 20 : 56	11 : 20 : 59	3 detik
15	11 : 21 : 38	11 : 21 : 41	3 detik

Berdasarkan hasil pengujian telegram ke LCD dalam mengirim pesan teks ke Telegram, data pada tabel menunjukkan bahwa waktu yang diperlukan untuk mengirim dan pesan terkirim berbeda-beda. Secara rata-rata, waktu *delay* yang dibutuhkan dari pengiriman pesan hingga pesan terkirim melalui Telegram adalah 2,8 detik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan dan pengujian alat proyek akhir *Smart Doorbell* Menggunakan Sistem Notifikasi Telegram, dapat diambil kesimpulan sebagai berikut:

1. Sistem yang dibuat mampu mengambil gambar wajah tamu menggunakan ESP32-CAM tipe OV2640 yang diaktifkan oleh pemilik rumah setelah tamu menekan *push button* serta mengirimkan notifikasi ke telegram pemilik rumah. Waktu yang dibutuhkan dari mengambil gambar hingga terkirim ke telegram dengan rata-rata 7,6 detik
2. *Smart doorbell* ini mampu mengirimkan dan menampilkan notifikasi kehadiran tamu pada aplikasi telegram. Saat tamu menekan *push button* langsung mengirimkan notifikasi ke telegram secara *real time*.
3. Sistem yang dibangun menampilkan informasi dari pemilik rumah berupa pesan teks yang dikirim dari aplikasi telegram ke LCD. Waktu yang dibutuhkan untuk mengirim pesan dari telegram ke LCD dengan waktu rata-rata 2,8 detik.

5.2 Saran

Dari hasil pembuatan alat proyek akhir ini, apabila dikembangkan lebih lanjut, penulis akan memberikan beberapa saran yaitu:

1. Perlu ditingkatkan penggabungan program agar tidak menjalankan fungsi program lainnya.
2. Perlu jaringan internet yang kuat karena Wi-Fi sering terputus dan tidak terkoneksi.
3. Perlu menggantikan kamera dengan kualitas yang lebih baik.

DAFTAR PUSTAKA

- [1] A. Fajardi, A. G. Putrada, and M. Abdurohman, "Pengujian Smart Doorbell Menggunakan Kamera dan Metode Haar-Cascade," vol. 6, no. 2, pp. 9336–9346, 2019.
- [2] Sujono and E. Aqilah, "Bel Rumah Otomatis berbasis IoT menggunakan Sensor Pir dan NodeMCU," *Epic (Exact Pap. Compil.*, vol. 2, no. 1, pp. 211–218, 2020.
- [3] E. Sagita, "SMART DOORBELL BERBASIS IOT." Universitas Telkom, S1 Teknik Elektro, Bandung. [Online]. Available: <http://bitly.ws/L9sy>
- [4] A. Aprillia, "RANCANG BANGUN DOORBELL WIRELESS BERBASIS IoT," 2019, [Online]. Available: <http://eprints.polsri.ac.id/id/eprint/8111>
- [5] S. Pandelaki, L. Sitanayah, and M. Liem, "RANCANG BANGUN SISTEM PEMANTAU KEDATANGAN TAMU BERBASIS INTERNET OF THINGS (IOT)," *JEECOM J. Electr. Eng. Comput.*, vol. 5, no. 1, pp. 4–10, 2021, doi: 10.33650/jeecom.v5i1.5802.
- [6] A. M, A. Febryan, Andriani, and Rahmania, "Rancang Bangun Sistem Keamanan Rumah Berbasis Telegram Menggunakan Esp 32 Cam," *VERTEX ELEKTRO-Jurnal Tek. Elektro UNIMUH*, vol. 15, no. 1, pp. 64–71, 2023, [Online]. Available: <https://journal.unismuh.ac.id/index.php/vertex/article/view/10246/5624>
- [7] J. Arifin and J. Frenando, "Sistem Keamanan Pintu Rumah Berbasis Internet of Things via Pesan Telegram Home Door Security System Based on Internet of Things Through Telegram Message," *Telka*, vol. 8, no. 1, pp. 49–59, 2022.
- [8] D. Wijanarko and A. Hariyanto, "Rancang Bangun Bel Pintu Tanpa Sentuh Menggunakan Microcontroller dan Sensor Infra Merah Berbasis Internet of Things," vol. 03, no. 01, pp. 29–32, 2022.
- [9] R. Andri, N. A. O. Saputri, and M. Akbar, "Sistem Notifikasi Tugas Akhir Universitas Bina Darma Berbasis Mobile," *Sistemasi*, vol. 9, no. 1, p. 155, 2020, doi: 10.32520/stmsi.v9i1.630.

- [10] P. Studi, T. Informatika, F. Teknik, D. A. N. Komputer, and U. P. Batam, “Rancangan bangun sistem keamanan rumah berbasis iot,” 2022.
- [11] Y. Fauzan, “KOTAK PENERIMA PAKET BERBASIS IoT MENGGUNAKAN MODUL ESP32-CAM,” *Institutional Repos. UIN Syarif Hidayatullah Jakarta*, pp. 1–66, 2020, [Online]. Available: <https://repository.uinjkt.ac.id/dspace/handle/123456789/56069>
- [12] S. A. Arrahma and R. Mukhaiyar, “Pengujian Esp32-Cam Berbasis Mikrokontroler ESP32,” *Jtein*, vol. 4, no. 1, pp. 60–66, 2023.



DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Muhammad Rafly Fadhel
Tempat Tanggal Lahir : Pangkalpinang, 13 Mei 2002
Alamat Rumah : Jalan Nyatoh Bukit Lama No.273
Jenis Kelamin : Laki-laki
Agama : Islam
Telp : -
Email : fadhelrafly@gmail.com



2. Riwayat Pendidikan

SD Negeri 26 Kelapa	2008 - 2014
SMP Negeri 2 Kelapa	2014 - 2017
SMA Negeri 1 Kelapa	2017 - 2020
Politeknik Manufaktur Negeri Bangka Belitung	2020 - 2023

3. Pengalaman Kerja

Praktik Kerja Lapangan di PT. Global Sahabat Otomasi Tahun 2022.

Sungailiat, 17 Juli 2023

(Muhammad Rafly Fadhel)

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Syahrul Ramadani
Tempat Tanggal Lahir : Bikang, 31 Desember 2000
Alamat Rumah : Jalan Raya Desa Bikang,
Toboali, Bangka Selatan
Jenis Kelamin : Laki-laki
Agama : Islam
Telp : -
Email : syahrulramadani060@gmail.com



2. Riwayat Pendidikan

SD Negeri 17 Toboali	2008 - 2014
SMP Negeri 1 Toboali	2014 - 2017
SMA Negeri 1 Toboali	2017 - 2020
Politeknik Manufaktur Negeri Bangka Belitung	2020 - 2023

3. Pengalaman Kerja

Praktik Kerja Lapangan di PT. Global Sahabat Otomasi Tahun 2022.

Sungailiat, 17 Juli 2023

(Syahrul Ramadani)

Lampiran 2 : Program

1. Program ESP32 Untuk Bel

```
#include <Arduino.h> // Library utama untuk Arduino
#include <WiFi.h> // Library untuk koneksi WiFi
#include <HTTPClient.h> // Library untuk membuat permintaan HTTP

// Konfigurasi SSID dan password WiFi
const char *ssid = "Catlover"; // nama jaringan wifi
const char *password = "daktaukubenerla"; // kata sandi wifi

// Konfigurasi token dan ID obrolan Telegram
String botToken = "6261729633:AAGSJ0IX8kelgVxy4X0VTLpZfEcAPOs94hc";
String chatId = "1269004513";

// Pin untuk tombol
const int BUTTON1 = 32;
const int BUTTON2 = 33;
const int BUTTON3 = 34;

// Status tombol
int button1State = LOW;
int button2State = LOW;
int button3State = LOW;
HTTPClient http;

void setup()
{
  Serial.begin(115200); // Inisialisasi komunikasi serial
  pinMode(BUTTON1, INPUT); // Mengatur pin BUTTON1 sebagai input
  pinMode(BUTTON2, INPUT); // Mengatur pin BUTTON2 sebagai input
```

```

pinMode(BUTTON3, INPUT); // Mengatur pin BUTTON3 sebagai input
pinMode(LED_BUILTIN, OUTPUT); // Mengatur pin LED sebagai output

WiFi.begin(ssid, password); // Menghubungkan ke jaringan WiFi
while (WiFi.status() != WL_CONNECTED) // Menunggu koneksi WiFi
{
    delay(1000);
Serial.println("Connecting to WiFi..."); // Pesan sedang terhubung ke WiFi
}

Serial.println(WiFi.macAddress()); // Menampilkan alamat MAC WiFi
Serial.println("Connected to the WiFi network"); // Pesan terhubung ke WiFi
}

// Fungsi untuk memeriksa perubahan status tombol
bool isChange(int state, int &buttonState)
{
    if (state != buttonState)
    {
        buttonState = state;
        return true;
    }
    return false;
}

void sendTelegramMessage(String message)
{
    http.begin("https://api.telegram.org/bot" + botToken + "/sendMessage?chat_id="
        + chatId + "&text=" + message); // Menginisialisasi koneksi HTTP dengan
        URL API Telegram
    int httpCode = http.GET(); // Mengirim permintaan HTTP GET untuk

```

```

    mengirimkan pesan ke bot Telegram
    Serial.println(httpCode); // Menampilkan kode status HTTP
    http.end();// Melepaskan sumber daya yang digunakan
}

void loop()
{
    // Memeriksa status tombol dan mengirim pesan ke Telegram jika ditekan
    if (isChange(digitalRead(BUTTON1), button1State))
    {
        // Memeriksa apakah tombol 1 ditekan
        if (button1State == LOW)
        {
            // onButton1Down
            Serial.println("Button 1 Pressed"); //Menampilkan pesan di serial monitor
            ketika tombol 1 ditekan
            digitalWrite(LED_BUILTIN, HIGH); // Menghidupkan LED_BUILTIN

            String message = "Ada tamu di depan pintu";
            sendTelegramMessage(message); // Mengirim pesan ke Telegram
            digitalWrite(LED_BUILTIN, LOW); // Mematikan LED_BUILTIN setelah
            pesan terkirim
        }
    }

    if (isChange(digitalRead(BUTTON2), button2State))
    {
        // Memeriksa apakah tombol 2 ditekan
        if (button2State == LOW)
        {
            // onButton2Down

```

```

Serial.println("Button 2 Pressed"); //Menampilkan pesan di serial monitor
ketika tombol 2 ditekan
digitalWrite(LED_BUILTIN, HIGH); // Menghidupkan LED_BUILTIN

String message = "Oke, saya tunggu";
sendTelegramMessage(message); // Mengirim pesan ke Telegram
digitalWrite(LED_BUILTIN, LOW); // Mematikan LED_BUILTIN setelah
pesan terkirim
}
}

if (isChange(digitalRead(BUTTON3), button3State))
{
// Memeriksa apakah tombol 3 ditekan
if (button3State == LOW)
{
// onButton3Down
Serial.println("Button 3 Pressed"); //Menampilkan pesan di serial monitor
ketika tombol 3 ditekan
digitalWrite(LED_BUILTIN, HIGH); // Menghidupkan LED_BUILTIN

String message = "Baiklah, nanti saya kembali";
sendTelegramMessage(message); // Mengirim pesan ke Telegram
digitalWrite(LED_BUILTIN, LOW); // Mematikan LED_BUILTIN setelah
pesan terkirim

}
}
}

```

2. Program ESP32 Untuk LCD

```
#include <LiquidCrystal_I2C.h> // untuk mengendalikan layar LCD menggunakan
    protokol I2C.
```

```
// atur nomor LCD kolom dan baris
```

```
int lcdColumns = 16;
```

```
int lcdRows = 2;
```

```
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
```

```
// Jika tidak tahu alamat I2C layar LCD yang digunakan, dapat menjalankan sketsa
    I2C scanner.
```

```
// atur alamat LCD, jumlah kolom dan baris
```

```
#include <WiFi.h> // untuk menghubungkan ESP32 ke jaringan WiFi.
```

```
#include <WiFiClientSecure.h> // untuk membuat koneksi WiFi yang aman.
```

```
#include <UniversalTelegramBot.h> // menghubungkan ESP32 dengan API
    Telegram
```

```
// Konfigurasi jaringan wifi
```

```
#define WIFI_SSID "Catlover" // nama jaringan wifi
```

```
#define WIFI_PASSWORD "daktaukubenerla" // kata sandi wifi
```

```
// Konfigurasi token dan id bot telegram
```

```
#define BOT_TOKEN BOT_TOKEN
```

```
"6261729633:AAGSJ0IX8keIgvxy4X0VTLpZfEcAPOs94hc"
```

```
#define CHAT_ID "1269004513"
```

```
const unsigned long BOT_MTBS = 1000; // rata-rata waktu antara pesan
    pemindaian
```

```
WiFiClientSecure secured_client; // digunakan untuk membentuk koneksi aman
```

ke server Telegram

```
UniversalTelegramBot bot(BOT_TOKEN, secured_client); // digunakan untuk  
    membuat dan mengirim permintaan ke server Telegram  
unsigned long bot_lasttime; // pemindaian pesan terakhir kali dilakukan
```

```
void handleNewMessages(int numNewMessages)  
{  
    Serial.print("handleNewMessages ");  
    Serial.println(numNewMessages);  
  
    for (int i = 0; i < numNewMessages; i++)  
    {  
        String chat_id = String(bot.messages[i].chat_id);  
        if (chat_id != CHAT_ID )  
        {  
            bot.sendMessage(chat_id, "Unauthorized user", "");  
        }  
        else  
        {  
            String text = bot.messages[i].text;  
  
            String from_name = bot.messages[i].from_name;  
            if (from_name == "")  
                from_name = "Guest";  
  
            if (text == "/start")  
            {  
                String welcome = "Welcome to ESP32 IoT display, " + from_name + ".\n";  
                welcome += "Tamu tekan bel, kamu bisa kirim pesan .\n\n";  
                bot.sendMessage(chat_id, welcome, "Markdown");  
            }  
        }  
    }  
}
```

```

    else
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(text);
    }
}
}
}

void setup() {
    // untuk menginisialisasi layar LCD.
    lcd.begin();
    // untuk menyalakan backlight pada layar LCD.
    lcd.backlight();

    Serial.begin(115200); // untuk menginisialisasi koneksi Serial dengan kecepatan
        115200 bps.
    Serial.println();

    // Menampilkan status koneksi WiFi pada Serial Monitor.
    Serial.print("Connecting to Wifi SSID ");
    Serial.print(WIFI_SSID);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Tambahkan
        sertifikat root untuk api.telegram.org
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(500);
    }
}

```



```

Serial.print("\nWiFi connected. IP address: ");
Serial.println(WiFi.localIP());

}

void loop() {

if (millis() - bot_lasttime > BOT_MTBS)
{
int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

while (numNewMessages)
{
Serial.println("got response");
handleNewMessages(numNewMessages);
numNewMessages = bot.getUpdates(bot.last_message_received + 1);
}

bot_lasttime = millis();
}
}

```

3. Program ESP32-CAM

```

#include <Arduino.h> // untuk pengembangan pada platform Arduino.
#include <WiFi.h> // untuk mengelola koneksi WiFi pada perangkat ESP32.
#include <WiFiClientSecure.h> // untuk mengelola koneksi WiFi yang aman
#include "soc/soc.h" // untuk kontrol atas System-on-Chip (SoC) pada ESP32.
#include "soc/rtc_cntl_reg.h" // untuk kontrol atas Real-Time Clock pada ESP32.
#include "esp_camera.h" // untuk mengakses fitur kamera pada perangkat ESP32.
#include <UniversalTelegramBot.h> // menghubungkan dengan API Telegram

```

```

#include <ArduinoJson.h> // untuk memanipulasi data JSON pada Arduino.

// Konfigurasi SSID dan password WiFi
const char* ssid = "Catlover"; // nama jaringan wifi
const char* password = "daktaukubenerla"; // kata sandi wifi

// Konfigurasi token dan Id bot telegram
String BOTtoken =
  "6261729633:AAGSJ0IX8kelgVxy4X0VTLpZfEcAPOs94hc"; // your Bot
  Token (Get from Botfather)
String CHAT_ID = "1269004513";

bool sendPhoto = false; // digunakan untuk mengirimkan foto ,apakah suatu foto
  harus dikirimkan atau tidak.

WiFiClientSecure clientTCP; // digunakan untuk menginisialisasi koneksi aman
  melalui jaringan Wi-Fi menggunakan protokol HTTPS.
UniversalTelegramBot bot(BOTtoken, clientTCP); // digunakan untuk berinteraksi
  dengan Telegram

#define FLASH_LED_PIN 4 // digunakan untuk mendefinisikan konstanta
  FLASH_LED_PIN dengan nilai 4.
bool flashState = LOW; // deklarasi variabel flashState dengan tipe data bool
  (boolean) dan nilai awal LOW.

//Memeriksa pesan baru setiap 1 detik.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

// Konfigurasi kamera model AI thinker
#define PWDN_GPIO_NUM 32

```

```
#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
```

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

```
void configInitCamera(){
// konfigurasi atribut sesuai dengan kebutuhan untuk mengatur kamera
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
```

```

config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

//init spesifikasi tinggi untuk melakukan pra-alokasi buffer yang lebih besar
if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10; //0-63 angka yang lebih rendah berarti kualitas yang
    lebih tinggi
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12; //0-63 angka yang lebih rendah berarti kualitas yang
    lebih tinggi
    config.fb_count = 1;
}

// init kamera
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
}

```

```

// Ukuran bingkai tarik-turun untuk rasio bingkai awal yang lebih tinggi
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s,          FRAMESIZE_CIF);          //
    UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
}

void handleNewMessages(int numNewMessages) {
    Serial.print("Handle New Messages: ");
    Serial.println(numNewMessages);

    for (int i = 0; i < numNewMessages; i++) {
        String chat_id = String(bot.messages[i].chat_id);
        if (chat_id != CHAT_ID){
            bot.sendMessage(chat_id, "Unauthorized user", "");
            continue;
        }

        // Cetak pesan yang diterima
        String text = bot.messages[i].text;
        Serial.println(text);

        String from_name = bot.messages[i].from_name;
        if (text == "/start") {
            String welcome = "Selamat datang , " + from_name + "\n";
            welcome += "Gunakan perintah dibawah ini untuk berinteraksi dengan smart
            doorbell tahun 2023 :\n";
            welcome += "/photo : mengambil foto tamu\n";
            welcome += "/flash : hidup/mati LED flash \n";
            welcome += "/Bentar_ya : pemilik sedang di rumah \n";
            welcome += "/Sedang_keluar : pemilik sedang keluar \n";
        }
    }
}

```

```

    bot.sendMessage(CHAT_ID, welcome, "");
}
if (text == "/flash") {
    flashState = !flashState;
    digitalWrite(FLASH_LED_PIN, flashState);
    Serial.println("Change flash LED state");
}
if (text == "/photo") {
    sendPhoto = true;
    Serial.println("New photo request");
}
}
}
}

```

```

String sendPhotoTelegram() {
    const char* myDomain = "api.telegram.org";
    String getAll = "";
    String getBody = "";

```

```

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

```

```

Serial.println("Connect to " + String(myDomain));

```

```

if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connection successful");

    String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data;
    name=\"chat_id\"; \r\n\r\n" + CHAT_ID + "\r\n--
    RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"photo\";
    filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
    String tail = "\r\n--RandomNerdTutorials--\r\n";

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

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

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

```

```

    }
}

clientTCP.print(tail);

esp_camera_fb_return(fb);

int waitTime = 10000; // batas waktu 10 detik
long startTimer = millis();
boolean state = false;

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

```



```

    getBody="Connected to api.telegram.org failed.";
    Serial.println("Connected to api.telegram.org failed.");
  }
  return getBody;
}

void setup(){
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  // Init Serial Monitor
  Serial.begin(115200);

  // Tetapkan Flash LED sebagai keluaran
  pinMode(FLASH_LED_PIN, OUTPUT);
  digitalWrite(FLASH_LED_PIN, flashState);

  // Konfigurasi dan init kamera
  configInitCamera();

  // Sambungkan ke Wi-Fi
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Tambahkan
    sertifikat root untuk api.telegram.org
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
}

```

```
Serial.print("ESP32-CAM IP Address: ");
Serial.println(WiFi.localIP());
}

void loop() {
  if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }
  if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }
}
```

Lampiran 3 : Bukti Bukan Plagiasi

smart door bel			
ORIGINALITY REPORT			
5 %	5 %	1 %	1 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	repository.ub.ac.id Internet Source		1 %
2	jurnal.uisu.ac.id Internet Source		1 %
3	www.scribd.com Internet Source		<1 %
4	docplayer.info Internet Source		<1 %
5	repo.darmajaya.ac.id Internet Source		<1 %
6	jurnal.untan.ac.id Internet Source		<1 %
7	Submitted to Sriwijaya University Student Paper		<1 %
8	adoc.pub Internet Source		<1 %
9	www.researchgate.net Internet Source		<1 %


10	digilib.its.ac.id Internet Source	<1%
11	journal.uin-alauddin.ac.id Internet Source	<1%
12	123dok.com Internet Source	<1%
13	text-id.123dok.com Internet Source	<1%
14	jurusan.tik.pnj.ac.id Internet Source	<1%
15	www.coursehero.com Internet Source	<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

FORM-PPR-3- 8: Form Revisi Laporan Akhir



FORM REVISI LAPORAN AKHIR
TAHUN AKADEMIK
...../...../.....

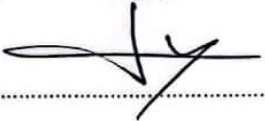
JUDUL : Suat donkbel menggunakan
ste aplikasi telegram

Nama Mahasiswa :

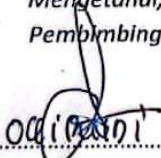
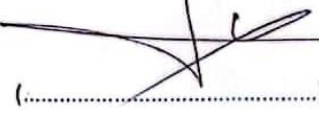
1.	Mr. Sap Jadel	NIRM:	
2.	syahrul hamada	NIRM:	
3.		NIRM:	
4.		NIRM:	
5.		NIRM:	

Bagian yang direvisi	Halaman
- Batasan paralel untuk alat uji	✓
- data hasil Cemerah dan jarak	✓

Sunggailiat, ...27...2023
Penguji


(.....)

Menyatakan telah menyetujui revisi laporan akhir yang telah dilakukan oleh mahasiswa

<p style="text-align: center;">Mengetahui, Pembimbing</p> <p style="text-align: center;"> (.....)</p>	<p style="text-align: right;">Sunggailiat, ...28...2023 Penguji</p> <p style="text-align: center;"> (.....)</p>
--	---

Lampiran 5 : Surat Pernyataan

Lampiran Nomor : 020/PROYEKAKHIR/DIII/2023

SURAT PERNYATAAN

Saya/Kami yang bertandatangan dibawah ini telah menyelesaikan Proyek Akhir yang berjudul:

SMART DOORBELL MENGGUNAKAN SISTEM NOTIFIKASI TELEGRAM



Oleh :

1. Muhammad Rafly Fadhel /NPM 0032050
2. Syahrul Ramadani /NPM 0032059

Dengan ini menyatakan bahwa isi laporan akhir proyek akhir sama dengan *hardcopy*.

Demikian surat pernyataan ini dibuat dengan sebenar-benarnya.

Sungailiat, 10 Agustus 2023

1. Muhammad Rafly Fadhel ()
2. Syahrul Ramadani ()

Mengetahui,

Pembimbing 1



(Ocsirendi, M.T)

Pembimbing 2



(Indra Dwisaputra, M.T)