

**RANCANG BANGUN ALAT METAL DETECTOR
TERINTEGRASI TELEGRAM DALAM UPAYA
PENINGKATAN KEAMANAN DAN EFISIENSI
PROSES PADA INDUSTRI PENGOLAHAN KAYU
BERBASIS *INTERNET OF THINGS* (IOT)**

SKRIPSI

*Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Komputer*



Oleh

AUZI AULIA SAHPUTRA

21101152620054

**JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG
PADANG, 2025**

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : AUZI AULIA SAHPUTRA

NoBP : 21101152620054

Fakultas : ILMU KOMPUTER

Jurusan : SISTEM KOMPUTER

Menyatakan Bahwa :

1. Sesungguhnya skripsi yang saya susun ini merupakan hasil karya tulis saya sendiri. Adapun bahagian-bahagian tertentu dalam skripsi yang saya peroleh dan hasil karya tulis orang lain, telah saya tuliskan sumbernya dengan jelas, sesuai dengan kaidah penulisan ilmiah.
2. Jika dalam pembuatan skripsi baik pembuatan program maupun skripsi secara keseluruhan terbukti dibuatkan oleh orang lain, maka saya bersedia menerima sanksi yang diberikan akademik, berupa pembatalan skripsi dan mengulang penelitian serta mengajukan judul baru.

Demikian surat pernyataan ini saya buat dengan sesungguhnya tanpa ada paksaan dari pihak manapun.

Padang, Juli 2025

(AUZI AULIA SAHPUTRA)
21101152620054

HALAMAN PENGESAHAN SKRIPSI

RANCANG BANGUN ALAT METAL DETECTOR TERINTEGRASI TELEGRAM DALAM UPAYA PENINGKATAN KEAMANAN DAN EFISIENSI PROSES PADA INDUSTRI PENGOLAHAN KAYU BERBASIS *INTERNET OF THINGS (IOT)*

Yang dipersiapkan dan disusun oleh

AUZI AULIA SAHPUTRA
21101152620054

Telah memenuhi persyaratan untuk dipertahankan di depan dewan penguji
pada ujian tahap akhir

Padang, Juli 2025

Pembimbing I

Pembimbing II

(Dr. Mardhiah Masril, S.Kom., M. Kom.)
NIDN : 1012108401

(Dr. Ruri Hartika Zain, S.Kom., M. Kom.)
NIDN : 1016038601

HALAMAN PENGESAHAN PENGUJI SIDANG SKRIPSI

**RANCANG BANGUN ALAT METAL DETECTOR TERINTEGRASI
TELEGRAM DALAM UPAYA PENINGKATAN KEAMANAN DAN
EFISIENSI PROSES PADA INDUSTRI PENGOLAHAN KAYU BERBASIS
*INTERNET OF THINGS (IOT)***

OLEH:

AUZI AULIA SAHPUTRA
21101152620054

PROGRAM STUDI SISTEM KOMPUTER

Skripsi ini telah dinyatakan LULUS oleh Penguji Materi Pada Sidang Program
Studi Strata 1 Ilmu Komputer Program Studi Sistem Komputer
Universitas Putra Indonesia “YPTK” Padang
Pada Hari/Tanggal : Jum’at, 22 Agustus 2025

TIM PENGUJI

1. **Dr. Ruri Hartika Zain, S.Kom., M. Kom.**
NIDN : 1016038601
2. **Riska Robianto, S.Kom., M. Kom.**
NIDN : 1004077904

Padang, 22 Agustus 2025

Mengetahui

**Dekan Fakultas Ilmu Komputer
Universitas Putra Indonesia YPTK Padang**

(Dr. Rini Sovia, S.Kom, M.Kom)
NIDN : 1005047601

HALAMAN PENGESAHAN LULUS SIDANG SKRIPSI

**RANCANG BANGUN ALAT METAL DETECTOR TERINTEGRASI
TELEGRAM DALAM UPAYA PENINGKATAN KEAMANAN DAN
EFISIENSI PROSES PADA INDUSTRI PENGOLAHAN KAYU BERBASIS
*INTERNET OF THINGS (IOT)***

Yang dipersiapkan dan disusun oleh

AUZI AULIA SAHPUTRA
21101152620054

Telah dipertahankan di depan dewan penguji
Pada tanggal : 22 Agustus 2025
Dan dinyatakan telah lulus memenuhi syarat

Pembimbing I

Pembimbing II

(Dr. Mardhiah Masril, S.Kom., M. Kom.)
NIDN : 1012108401

(Dr. Ruri Hartika Zain, S.Kom., M. Kom.)
NIDN : 1020099101

Padang, 22 Agustus 2025

**Dekan Fakultas Ilmu Komputer
Universitas Putra Indonesia YPTK Padang**

(Dr. Rini Sovia, S.Kom, M.Kom)
NIDN : 1005047601

ABSTRACT

Thesis Tittle : **DESIGN AND DEVELOPMENT OF A METAL DETECTOR DEVICE INTEGRATED WITH TELEGRAM FOR ENHANCING SECURITY AND PROCESS EFFICIENCY IN THE WOOD PROCESSING INDUSTRY BASED ON THE INTERNET OF THINGS (IOT)**

Student Name : **AUZI AULIA SAHPUTRA**

Number : **21101152620054**

Study Program : **Computer Engineering**

Degree Granted : **Strata 1 (S1)**

Advisor : **1. Dr. Mardhiah Masril, S.Kom., M. Kom.**
2. Dr. Ruri Hartika Zain, S.Kom., M. Kom.

The increasing need for security and process efficiency in the wood processing industry has encouraged the development of intelligent monitoring systems. This study presents the design and development of a metal detector device integrated with Telegram as a notification and control interface, based on the Internet of Things (IoT). The system is equipped with a fingerprint module to ensure secure access, allowing users to add or delete fingerprint data through Telegram commands. The core function of the device is to detect the presence of metal and tin objects embedded in processed wood materials. Once detected, the system immediately sends real-time notifications to Telegram and stores detection data in a Firebase database for record-keeping and monitoring purposes. The device is controlled by a microcontroller and combines hardware integration, software communication, and cloud-based data storage. The development process includes system requirement analysis, design, hardware integration, software programming, and functional testing. Test results show that the system operates reliably, enhances industrial safety, and improves the efficiency of wood material inspection processes.

Keywords: *Metal Detector, Fingerprint, Telegram, Internet of Things (IoT), Firebase, Wood Processing Industry, Security System, Microcontroller.*

ABSTRAK

Judul Skripsi : RANCANG BANGUN ALAT METAL DETECTOR
TERINTEGRASI TELEGRAM DALAM UPAYA
PENINGKATAN KEAMANAN DAN EFISIENSI
PROSES PADA INDUSTRI PENGOLAHAN KAYU
BERBASIS *INTERNET OF THINGS* (IOT)

Nama : AUZI AULIA SAHPUTRA

No. BP : 21101152620054

Program Studi : Sistem Komputer

Jenjang Pendidikan : Strata 1 (S1)

Pembimbing : 1. Dr. Mardhiah Masril, S.Kom., M. Kom.
2. Dr. Ruri Hartika Zain, S.Kom., M. Kom.

Meningkatnya kebutuhan akan keamanan dan efisiensi proses dalam industri pengolahan kayu mendorong pengembangan sistem pemantauan cerdas. Penelitian ini merancang dan membangun sebuah alat metal detector yang terintegrasi dengan Telegram sebagai antarmuka notifikasi dan kontrol, berbasis Internet of Things (IoT). Sistem ini dilengkapi dengan modul fingerprint untuk menjamin keamanan akses, di mana pengguna dapat menambahkan atau menghapus data sidik jari melalui perintah Telegram. Fungsi utama alat ini adalah mendeteksi keberadaan logam dan timah pada bahan kayu yang diolah. Setelah pendeteksian, sistem secara otomatis mengirimkan notifikasi real-time ke Telegram dan mencatat data tersebut ke dalam database Firebase untuk kebutuhan pencatatan dan pemantauan. Alat ini dikendalikan oleh mikrokontroler dan menggabungkan integrasi perangkat keras, komunikasi perangkat lunak, serta penyimpanan data berbasis cloud. Proses pengembangan meliputi analisis kebutuhan sistem, perancangan, integrasi perangkat keras, pemrograman perangkat lunak, dan pengujian fungsionalitas. Hasil pengujian menunjukkan bahwa sistem bekerja dengan baik, meningkatkan keamanan industri, serta mempercepat dan mempermudah proses pemeriksaan bahan kayu.

Kata Kunci: Metal Detector, Sidik Jari, Telegram, Internet of Things (IoT), Firebase, Industri Pengolahan Kayu, Sistem Keamanan, Mikrokontroler.

KATA PENGANTAR



Puji syukur penulis ucapkan kepada Allah SWT. yang telah memberikan rahmat dan karunianya, sehingga bisa menyelesaikan penulisan skripsi penulis dengan judul **“RANCANG BANGUN ALAT METAL DETECTOR TERINTEGRASI TELEGRAM DALAM UPAYA PENINGKATAN KEAMANAN DAN EFISIENSI PROSES PADA INDUSTRI PENGOLAHAN KAYU BERBASIS *INTERNET OF THINGS* (IOT)”**.

Penulisan laporan ini bertujuan dalam rangka untuk memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer Universitas Putra Indonesia “YPTK” Padang. Penulisan laporan ini diwujudkan dalam bentuk suatu perancangan alat, yang tidak lepas dari kata kekurangan, keterbatasan, dan jauh dari kata sempurna, baik dari segi penulisan, bahasa, dan alatnya. Hal ini dikarenakan oleh ilmu, pengetahuan dan pengalaman penulis miliki masih kurang serta keterbatasan fasilitas yang disediakan. Oleh karena itu penulis berharap adanya kritik saran yang mendukung agar dapat diperbaiki kembali dan sempurna kedepannya.

Selanjutnya di dalam penulisan laporan ini penulis sadar bahwa keberhasilan penulis dalam menyelesaikan skripsi ini berkat dari dorongan dan bimbingan pembimbing, orang tua, teman, dan berbagai pihak, oleh karena itu penulis ingin mengucapkan rasa terima kasih, khususnya kepada:

1. Bapak (Alm) **H. Herman Nawas Rahimahullah** dan Ibu **Dr. Hj. Zerni Melmusi, MM., Akt., CA** sebagai pendiri dan juga pembina Yayasan Perguruan Tinggi Komputer (YPTK) Padang.

2. Ibu **Sitti Rizki Mulyani, S.Pd., MM** sebagai Ketua Yayasan Perguruan Tinggi Komputer (YPTK) Padang.
3. Bapak **Assoc. Prof. Dr. Muhammad Ridwan, SE., MM** selaku Rektor Universitas Putra Indonesia “YPTK” Padang.
4. Ibu **Dr. Rini Sovia, S.Kom., M.Kom** selaku Dekan Fakultas Ilmu Komputer Universitas Putra Indonesia “YPTK” Padang.
5. Ibu **Dr. Retno Devita, S.Kom., M.Kom** selaku Ketua Jurusan Sistem Komputer Universitas Putra Indonesia “YPTK” Padang.
6. Ibu **Dr. Mardhiah Masril, S.Kom., M. Kom** selaku Pembimbing I yang telah membimbing dan sangat banyak memberi masukan kepada penulis
7. Ibu **Dr. Ruri Hartika Zain, S.Kom., M. Kom** selaku Pembimbing II yang telah membimbing dan sangat banyak memberi masukan kepada penulis.
8. Bapak dan Ibu Dosen serta Karyawan dan Karyawati Universitas Putra Indonesia “YPTK” Padang.

Akhir kata, penulis berharap semoga kebaikan dan amal yang mereka berikan kepada penulis mendapatkan balasan yang berlipat ganda hendaknya dari Allah SWT. *Aamiin ya rabbal ‘alamiin.*

Padang, Juli 2025

AUZI AULIA SAHPUTRA
21101152620054

DAFTAR ISI

LEMBAR PERNYATAAN	ii
HALAMAN PENGESAHAN SKRIPSI.....	iii
HALAMAN PENGESAHAN PENGUJI SIDANG SKRIPSI.....	iv
HALAMAN PENGESAHAN LULUS SIDANG SKRIPSI.....	v
ABSTRACT	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	4
1.3 Batasan Masalah	5
1.4 Hipotesa	6
1.5 Tujuan Penelitian	7
1.6 Manfaat Penelitian	8
BAB II TINJAUAN PUSTAKA.....	10
2.1 Konsep Dasar Sistem.....	10
2.1.1 Pengertian Sistem.....	10
2.1.2 Dasar Teori Sistem.....	10
2.1.3 Karakteristik Sistem.....	11
2.1.4 Siklus Pengembangan Sistem	12
2.2 Alat Bantu Perancangan.....	14
2.2.1 <i>Context Diagram</i>	14
2.2.2 <i>Data Flow Diagram (DFD)</i>	15
2.2.3 <i>Flowchart</i>	16
2.3 Sistem Kontrol	17
2.3.1 Sistem <i>Loop</i> Terbuka	18

2.3.2	Sistem <i>Loop</i> Tertutup.....	18
2.4	Komponen Utama	19
2.4.1	Arduino Uno	19
2.4.2	ESP32.....	20
2.4.3	Coil.....	21
2.4.4	<i>Fingerprint</i>	21
2.4.5	Telegram	22
2.4.6	LCD (<i>Liquid Crystal Display</i>)	23
2.4.7	LED.....	24
2.4.8	Buzzer	24
2.5	Komponen Pendukung.....	25
2.5.1	Resistor.....	25
2.5.2	Transistor	28
2.5.3	Kapasitor	28
2.5.4	Dioda.....	32
2.6	Bahasa Pemrograman Arduino	32
2.6.1	Struktur.....	32
2.6.2	Syntax Pemrograman	33
2.6.3	Tipe Data Arduino	34
2.6.4	Operator	34
2.6.5	Struktur Pengaturan.....	36
2.7	<i>Software</i> Arduino IDE	37
BAB III METODOLOGI PENELITIAN		39
3.1	Kerangka Kerja Penelitian	39
3.2	Uraian Kerangka Kerja Penelitian.....	40
3.2.1	Identifikasi Masalah	40
3.2.2	Pengumpulan Data	40
3.2.3	Analisis Sistem.....	42
3.2.4	Perancangan Sistem.....	43
3.2.5	Pembuatan Sistem	44
3.2.6	Pengujian Sistem	44
3.2.7	Implementasi sistem.....	45

BAB IV ANALISA DAN HASIL	46
4.1 Desain Sistem Secara Umum.....	46
4.1.1 Context Diagram	46
4.1.2 Data flow diagram (DFD)	49
4.1.3 Blok Diagram	51
4.2 Prinsip Kerja Sistem	52
4.3 Rancang Fisik Alat.....	52
4.4 Desain Sistem Terperinci.....	54
4.4.1 Desain Rangkaian Arduino Uno	54
4.4.2 Desain Rangkaian ESP32.....	56
4.4.3 Desain Rangkaian <i>Fingerprint</i>	56
4.4.4 Desain Rangkaian <i>Coil</i>	57
4.4.5 Desain Rangkaian LCD 16x2	58
4.4.6 Desain Rangkaian Buzzer	59
4.4.7 Desain Rangkaian LED.....	60
4.5 Rancangan Modul Program	61
4.5.1 <i>Flowchart</i>	61
4.5.2 <i>Listing</i> Program.....	65
BAB V PENGUJIAN SISTEM	71
5.1 Pengujian Sistem Permodul	71
5.1.1 Pengujian Sistem Minimum.....	71
5.1.2 Pengujian <i>Fingerprint</i>	74
5.1.3 Pengujian <i>Coil</i>	74
5.2 Pengujian Sistem Keseluruhan	75
BAB VI PENUTUP	82
6.1 Kesimpulan	82
6.2 Saran	83
DAFTAR PUSTKA.....	85
LAMPIRAN.....	87

DAFTAR GAMBAR

Gambar 2.1 Siklus Pengembangan Sistem.....	14
Gambar 2.2 Sistem Pengendalian Loop Terbuka.....	18
Gambar 2.3 Sistem Pengendalian Loop Tertutup	19
Gambar 2.4 Arduino Uno.....	20
Gambar 2.5 ESP32	20
Gambar 2.6 Coil.....	21
Gambar 2.7 Fingerprint.....	22
Gambar 2.8 Aplikasi Telegram	22
Gambar 2.9 LCD (Liquid Crystal Display).....	23
Gambar 2.10 Light Emitting Diode (LED)	24
Gambar 2.11 Buzzer.....	25
Gambar 2.12 Resistor.....	26
Gambar 2.13 Gambar dan Simbol Transistor	28
Gambar 2.14 Simbol Kapasitor.....	29
Gambar 2.15 Kapasitor Keramik	30
Gambar 2.16 Kapasitor Polyster	31
Gambar 2.17 Kapasitor Elektrolit	31
Gambar 2.18 Dioda	32
Gambar 2.19 Software Arduino IDE	38
Gambar 4.1 <i>Context Diagram</i>	47
Gambar 4.2 <i>Data Flow Diagram</i>	49
Gambar 4.3 Blok Diagram	51
Gambar 4.4 Rancangan Tampilan Atas	53
Gambar 4.5 Rancangan Tampilan Kanan	53
Gambar 4.6 Rancangan Tampilan Kiri	53
Gambar 4.7 Rangkaian Arduino Uno.....	55
Gambar 4.8 Rangkaian ESP32.....	56
Gambar 4.9 Rangkaian Fingerprint.....	57
Gambar 4.10 Rangkaian Coil.....	58
Gambar 4.11 Rangkaian LCD 16x2.....	59

Gambar 4.12 Rangkaian Buzzer	60
Gambar 4.13 Rangkaian LED	61
Gambar 4.14 Flowchart.....	64
Gambar 5.1 Tampilan Ide Arduino	71
Gambar 5.2 Tampilan <i>Save As</i>	72
Gambar 5.3 Tampilan Pemilihan <i>Board</i> Arduino Mega.....	72
Gambar 5.4 Tampilan <i>Compiler</i> pada Arduino IDE.....	73
Gambar 5.5 Tampilan <i>Uploading</i> Program pada Arduino IDE	74
Gambar 5.7 Tampilan Awal LCD 16x2	75
Gambar 5.8 Tampilan LCD 16x2 Untuk Melakukan <i>Scanning</i> Sidik Jari	76
Gambar 5.9 Tampilan LCD 16x2 Saat Sidik Jari Belum Terdaftar	76
Gambar 5.10 Penambahan Sidik Jari Melalui Aplikasi Telegram	76
Gambar 5.11 Proses Pendaftaran Sidik Jari	78
Gambar 5.12 Tampilan LCD 16x2 Apabila Sidik Jari Telah Terdaftar.....	78
Gambar 5.13 LED 2 Menyala Saat Mendeteksi Benda Berjenis Besi.....	79
Gambar 5.14 Tampilan Telegram dan <i>Firebase</i> Apabila Benda Berjenis Besi Terdeteksi.....	80
Gambar 5.15 LED 1 Menyala Saat Mendeteksi Benda Berjenis Timah.....	80
Gambar 5.16 Tampilan Telegram dan <i>Firebase</i> Apabila Benda Berjenis Timah Terdeteksi.....	81
Gambar 5.17 Proses Penghapusan Sidik Jari di Aplikasi Telegram	81

DAFTAR TABEL

Tabel 2.1 Simbol dari Context Diagram	15
Tabel 2.2 Simbol dari Data Flow Diagram	16
Tabel 2.3 Simbol-Simbol Standar dalam Flowchart	17
Tabel 2.4 Kode Warna Resistor	27
Tabel 2.5 Variabel dalam Arduino	34
Tabel 2.6 Operator Aritmatika	35
Tabel 2.7 Operator Kondisi	35
Tabel 2.8 Operator Logika	36
Tabel 5.1 Pengujian <i>Fingerprint</i>	74
Tabel 5.2 Pengujian <i>Coil</i>	75

BAB I

PENDAHULUAN

1.1 Latar Belakang

Metal detector adalah suatu alat keamanan yang memiliki bentuk seperti tongkat yang didalamnya terdapat sensor pendeteksi logam dan cara kerjanya harus diarahkan kebagian yang ingin diperiksa (Mastutie Faizah & Meldawati Artayani, 2023). *Metal detector* menjadi alat penting yang memungkinkan proses pengamanan dan kontrol kualitas, terutama dalam industri pengolahan kayu, di mana benda logam yang terselip dalam kayu dapat merusak alat dan membahayakan operator mesin. Teknologi *Metal detector* bekerja dengan memanfaatkan prinsip elektromagnetik untuk mengidentifikasi adanya logam, sehingga proses inspeksi dapat dilakukan dengan cepat dan tepat. Dengan penerapan yang tepat, *Metal detector* dapat meningkatkan efisiensi dan keamanan proses produksi di sektor industri yang menggunakan bahan mentah seperti kayu.

Produksi kayu olahan terus mengalami kenaikan sepanjang tahun 2021 sampai tahun 2023 dimana produksi kayu olahan pada tahun 2021 berkisar sebanyak 40.264.823,18 m³ kemudian pada tahun 2022 produksi kayu olahan berkisar 50.798.152,94 m³ dan pada tahun 2023 produksi olahan kayu berkisar 50.922.990,7 m³ (Statistik Badan Kehutanan, 2023). Produksi olahan kayu meliputi *chip & partikel*, kayu lapis/*Plywood*, kayu gergajian/*sawn timber*, *veneer*, *barecore*, *wood pallet*, papan partikel/*chipboard*, *moulding/dowel* dan sebagainya. Hal ini membuktikan bahwa industri pengolahan kayu memiliki peluang besar pada zaman

sekarang ini dilihat dari jumlah hasil produksi yang turut dipengaruhi oleh tingginya kebutuhan masyarakat akan bahan baku kayu.

Seiring dengan meningkatnya produksi kayu olahan, kebutuhan akan teknologi yang mendukung efisiensi dan keamanan dalam proses produksi menjadi semakin penting. Teknologi *Internet of Things* (IoT) memungkinkan perangkat saling terhubung dan berkomunikasi secara real-time melalui internet dengan integrasi IoT dan aplikasi seperti Telegram, data dari sensor dapat dipantau dan dikontrol secara langsung. Penerapan metal detector berbasis IoT yang terhubung ke Telegram di industri pengolahan kayu memungkinkan notifikasi deteksi logam dikirim segera ke operator, sehingga respons terhadap potensi bahaya dapat dilakukan dengan cepat.

Saat ini, industri pengolahan kayu menghadapi tantangan besar dalam mengidentifikasi benda logam pada kayu yang diproses, seperti paku atau serpihan logam kecil yang tersisa pada kayu bekas. Benda logam yang tidak terdeteksi dalam bahan baku kayu dapat menyebabkan kerusakan pada mesin pemotong dan membahayakan keselamatan kerja. Aspek keselamatan kerja yang terancam akibat sisa paku dalam proses pengolahan kayu meliputi risiko cedera pada tangan pekerja yang terkena paku dan luka pada kaki akibat menginjak paku. Kondisi ini diperburuk oleh kurangnya perhatian atau ketidakpedulian pekerja terhadap penggunaan perlengkapan keselamatan kerja.

Penelitian terdahulu terkait permasalahan ini pernah dilakukan oleh Moch. Andry Nahari dengan judul penelitian “Alat Pendeteksi Logam Pada Kayu Dalam Proses Penggergajian” pada tahun 2016. Penelitian ini menggunakan Arduino Uno

R3 sebagai kendali utama untuk mendeteksi logam pada kayu dengan bantuan sensor *proximity*. Informasi hasil deteksi ditampilkan pada LCD 16x2, sementara tindakan otomatis dilakukan melalui relay yang mengontrol motor DC. Motor DC akan berhenti beroperasi ketika logam terdeteksi, sehingga sistem ini mengintegrasikan sensor dan aktuator untuk mendukung proses kerja yang lebih aman dan efisien. Penelitian ini memiliki beberapa kekurangan dimana penggunaan sensor *proximity* yang memiliki area jangkauan yang dekat dan sensitivitas rendah, kemudian pada penelitian ini tidak memiliki keamanan pada alat sehingga apabila alat dicuri maka alat bisa secara bebas digunakan dan yang terakhir adalah tidak adanya pencacatan mengenai hasil deteksi logam karena sistem tidak dilengkapi dengan *database*, sehingga tidak memungkinkan untuk menyimpan rekam jejak deteksi, melakukan analisis data, memberikan transparansi dalam pelaporan.

Berdasarkan kekurangan dari penelitian sebelumnya penulis ingin mengembangkan sebuah alat *metal detector* yang dapat menutupi kelemahan penelitian sebelumnya seperti penggunaan koil yang mempunyai sensitivitas yang lebih tinggi sekaligus mempunyai jangkauan yang lebih luas. Penggunaan *fingerprint* sebagai fitur keamanan alat dan adanya pencacatan mengenai hasil deteksi logam ke dalam *database*. Integrasi *Metal detector* berbasis IoT yang terhubung dengan Telegram, memungkinkan masalah deteksi logam pada kayu di industri pengolahan dapat diatasi secara lebih efektif dan efisien. Sistem ini memungkinkan notifikasi langsung kepada pihak yang bertanggung jawab ketika terdapat benda logam dalam kayu, sehingga proses produksi dapat dihentikan sementara dan benda logam tersebut dihilangkan sebelum merusak mesin. Alat *Metal detector* IoT ini juga memungkinkan pemantauan jarak jauh sehingga

mengurangi kebutuhan inspeksi manual yang memakan waktu. Dengan demikian, keamanan proses pengolahan kayu dapat ditingkatkan, dan efisiensi produksi pun lebih terjaga.

Secara keseluruhan, penelitian ini bertujuan untuk merancang dan membangun alat *Metal detector* yang terintegrasi dengan Telegram untuk mengoptimalkan proses keamanan dan efisiensi di industri pengolahan kayu. Diharapkan, hasil dari penelitian ini dapat memberikan solusi inovatif yang tidak hanya meningkatkan keamanan, tetapi juga membantu industri menghemat waktu dan biaya dalam proses produksi. Integrasi IoT dalam sistem ini juga menjadi langkah untuk mendorong digitalisasi dalam pengelolaan keamanan dan kontrol kualitas di sektor industri. Oleh karena itu penulis tertarik untuk merancang dan mengembangkan alat *Metal detector* pada industri kayu dalam bentuk judul “RANCANG BANGUN ALAT *METAL DETECTOR* TERINTEGRASI TELEGRAM DALAM UPAYA PENINGKATAN KEAMANAN DAN EFISIENSI PROSES PADA INDUSTRI PENGOLAHAN KAYU BERBASIS INTERNET OF THINGS (IOT)”.

1.2 Perumusan Masalah

Perumusan masalah merupakan langkah penting dalam setiap penelitian, karena masalah yang dirumuskan akan menjadi fokus utama dari studi yang dilakukan. Berdasarkan latar belakang, rumusan masalah dalam penelitian yang dilakukan adalah sebagai berikut:

1. Bagaimana merancang dan mengontrol sistem *Metal detector* menggunakan Arduino Uno dengan baik?

2. Bagaimana sensor *fingerprint* dapat menjadi keamanan pada alat dengan baik?
3. Bagaimana *coil* dapat mendeteksi benda berjenis besi dan timah pada kayu dengan benar?
4. Bagaimana Telegram dapat memberitahukan kepada pengguna terdapat bahan metal pada kayu yang sedang diolah dengan baik?
5. Bagaimana LCD 16x2 dan buzzer dapat memberitahukan kepada pengguna terdapat bahan metal pada kayu yang sedang diolah dengan baik?
6. Bagaimana *database firebase* dapat menyimpan hasil pencatatan mengenai sidik jari pengguna dan pembacaan bahan metal pada kayu dengan baik?

1.3 Batasan Masalah

Batasan masalah digunakan agar pembahasan dalam laporan kerja praktik ini tidak terlalu meluas, maka dari itu perlu adanya pembatasan masalah. Adapun batasan masalah yang dikemukakan adalah sebagai berikut:

1. Alat ini menggunakan telegram untuk menerima notifikasi apakah adanya benda metal pada kayu yang sedang diolah dan notifikasi mengenai pembacaan.
2. Alat ini hanya menggunakan satu sistem keamanan yaitu *fingerprint*.
3. Alat menggunakan *database firebase* dalam melakukan pencatatan seperti benda yang terdeteksi dan pengguna yang menggunakan alat.
4. Alat ini hanya menggunakan indikator berupa buzzer, LED dan LCD 16x2 untuk memberitahukan kepada pengguna adanya benda yang terdeteksi didalam kayu yang sedang diolah.

5. Alat dikontrol oleh mikrokontroler Arduino Uno dan menggunakan *Software* yang digunakan dalam perancangan alat ini adalah arduino IDE.

1.4 Hipotesa

Berdasarkan pada perumusan masalah di atas, maka dapat diambil beberapa hipotesa yaitu:

1. Diharapkan Arduino Uno dapat merancang dan mengontrol sistem *Metal detector* menggunakan dengan baik, sehingga sensor-sensor dapat dibaca secara akurat.
2. Diharapkan *fingerprint* dapat membaca sidik jari pengguna yang telah terdaftar sekaligus dapat menjadi sistem keamanan pada alat, sehingga alat tidak dapat dijalankan apabila terjadi kehilangan atau pencurian pada alat.
3. Diharapkan *coil* dapat mengidentifikasi bahan metal pada kayu yang akan diolah dengan akurat, sehingga meminimalisir kecelakaan pada pekerja dan kerusakan pada alat potong kayu.
4. Diharapkan telegram dapat menambahkan dan menghapus sidik jari dari pengguna dengan baik, sehingga pengguna baru dapat dengan mudah menambahkan sidik jari maupun pengguna yang tidak lagi bekerja pada industri tersebut dapat dihapus.
5. Diharapkan LCD 16x2 dan buzzer dapat memberitahukan kepada pengguna terdapat bahan metal pada kayu yang sedang diolah dengan baik, sehingga pengguna dapat mengetahui dengan cepat bahwa ada bahan metal seperti paku dalam kayu yang akan diolah.

6. Diharapkan *database firebase* dapat menyimpan hasil pencatatan mengenai sidik jari pengguna dan pembacaan bahan metal pada kayu dengan baik, sehingga pihak yang berwenang dapat memantau data secara *real-time*, menganalisis rekam jejak penggunaan alat, dan memastikan sistem berjalan dengan efisien serta aman dari potensi penyalahgunaan.

1.5 Tujuan Penelitian

Tujuan penelitian digunakan untuk menjelaskan secara jelas dan sistematis apa yang ingin dicapai melalui pembuatan alat ini. Adapun tujuan yang dicapai dalam pembuatan alat ini adalah sebagai berikut:

1. Memahami ilmu pengetahuan dibidang komputer dalam pengontrolan alat menggunakan ESP32, *fingerprint*, telegram dan kemudian menjadi salah satu contoh aplikasi pada mata kuliah yang dipelajari.
2. Menganalisis ilmu yang telah penulis peroleh selama pendidikan dan menjalankannya berupa aplikasi dengan cara pembuatan alat yang dapat mempermudah pekerjaan manusia.
3. Merancang suatu program dengan memanfaatkan ESP32 sehingga dapat membaca adanya bahan metal dalam industri pengolahan kayu.
4. Membangun alat dengan inovasi terbaru untuk pembacaan bahan metal pada industri pengolahan kayu.
5. Menguji inovasi sistem pelacakan bahan metal pada industri pengolahan kayu terintegrasi dengan *Internet of Things*.

1.6 Manfaat Penelitian

Adapun manfaat penelitian dalam perancangan sistem ini adalah sebagai berikut:

A. Bagi Penulis

1. Menambah pengetahuan penulis dalam bidang teknologi deteksi logam, sistem kontrol, dan penerapan *Internet of Things* (IoT) dalam industri.
2. Memperluas keterampilan teknis penulis dalam pengembangan perangkat keras, pemrograman, dan integrasi sistem dengan *platform* komunikasi seperti Telegram.
3. Membangun jaringan dengan profesional di bidang industri pengolahan kayu dan teknologi, yang dapat membuka peluang kolaborasi atau pekerjaan di masa depan.

B. Bagi Program Studi

1. Menambah referensi literatur bagi mahasiswa yang tertarik pada penerapan teknologi IoT dalam meningkatkan keamanan dan efisiensi proses industri.
2. Penelitian ini dapat menjadi acuan untuk pengembangan lebih lanjut di bidang teknologi deteksi dan keamanan industri, serta memberikan wawasan baru dalam proses pembelajaran di lingkungan akademis.

C. Bagi Masyarakat dan Industri Pengolahan Kayu

1. Mendukung upaya peningkatan keamanan di industri pengolahan kayu dengan solusi berbasis teknologi untuk mendeteksi logam secara efektif.

2. Menyediakan data deteksi logam yang dapat diakses secara *real-time* oleh pihak terkait, sehingga mempermudah pengawasan dan penanganan potensi bahaya.
3. Meningkatkan kesadaran dan kepatuhan terhadap standar keamanan di industri pengolahan kayu melalui penerapan teknologi yang transparan dan efisien.

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Konsep dasar sistem yaitu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan. Sistem merupakan *entity* yang terdiri dari dua atau lebih komponen yang saling berhubungan untuk mencapai suatu tujuan (Arizal Khoir Rahman, et al, 2022).

2.1.1 Pengertian Sistem

Suatu sistem didefinisikan sebagai seperangkat prosedur yang saling berhubungan dan saling berhubungan untuk secara kolektif menyelesaikan suatu tugas. Secara umum, sebuah sistem informasi terdiri dari tiga komponen utama. Ketiga komponen tersebut meliputi *software*, *hardware* dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain (Myra Andriana & Priyadi, 2023).

2.1.2 Dasar Teori Sistem

Sistem adalah sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan. Sistem mempunyai bagian terkecil yang disebut dengan komponen sistem atau subsistem. Ada beberapa definisi tentang sistem yang ditemukan oleh para ahli yaitu:

1. Menurut Romney dan Steinbart

Sistem adalah suatu rangkaian yang terdiri dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi satu sama lain untuk mencapai tujuan dimana sistem biasanya terbagi dalam subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

2. Menurut Edhi Sutanta

Secara umum, sistem dapat didefinisikan sebagai kumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu-kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan.

3. Menurut Fery Wongso

Sistem adalah kumpulan atau rangkaian komponen-komponen yang saling berhubungan, bekerja sama dan saling berinteraksi untuk mencapai suatu tujuan dengan melalui tiga tahapan *input* (masukan), proses, *output* (keluaran).

2.1.3 Karakteristik Sistem

Sebuah sistem mempunyai beberapa karakteristik, yaitu diantaranya (Imam Muallim, 2021).

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama untuk membentuk satu kesatuan.

2. Batas Sistem (*boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya.

3. Lingkungan Luar Sistem (*environments*)

Lingkungan luar dari suatu sistem adalah apapun di luar batas sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara suatu sistem dengan subsistem lainnya.

5. Masukan (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem.

6. Keluaran (*Output*)

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan berubah menjadi keluaran.

8. Sasaran (*Objectives*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.

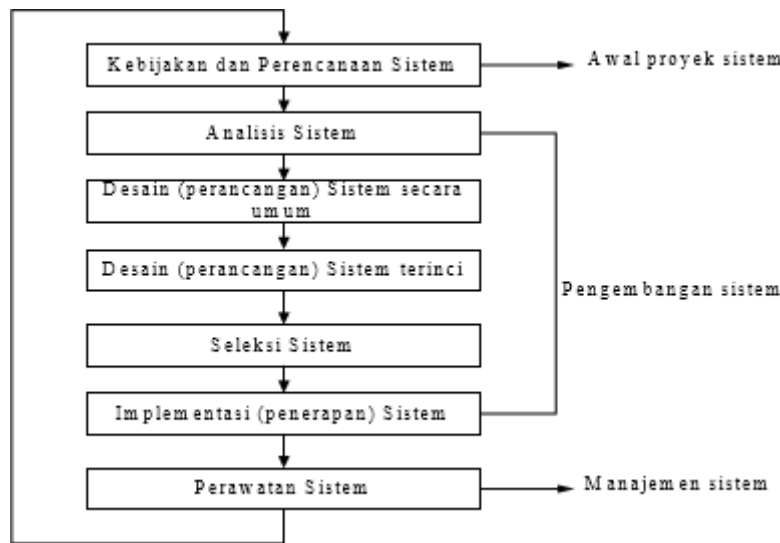
2.1.4 Siklus Pengembangan Sistem

System Development Life Cycle (SDLC) atau siklus hidup pengembangan sistem dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. SDLC juga merupakan pola untuk mengembangkan sistem perangkat lunak yang terdiri dari tahapan perencanaan

(*planning*), analisis (*analyst*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (Aceng Abdul Wahid, 2020). Berikut adalah langkah-langkah utama dari siklus hidup pengembangan sistem:

1. *Requirement* : Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.
2. *Design*: Tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.
3. *Implementation* : Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.
4. *Verification* : Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit *testing* (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).
5. *Maintenance*: Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

Siklus pengembangan sistem dapat dilihat pada Gambar 2.1.



Sumber : Alhamidia, et al., *Implementasi Sistem Informasi Pemesanan Makanan Berbasis Web*, 2020.

Gambar 2.1 Siklus Pengembangan Sistem

2.2 Alat Bantu Perancangan

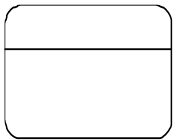
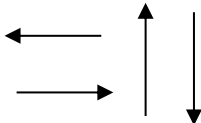
Aturan dasar yang harus dilakukan adalah untuk mendefinisikan sistem yang dirancang secara menyeluruh karena harus memiliki gambaran yang jelas terhadap ruang lingkup sistem yang dibahas. Untuk menggambarkan sistem tersebut beberapa media digunakan yaitu *Context Diagram*, *Data Flow Diagram* dan *Flowchart*.

2.2.1 Context Diagram

Context Diagram adalah gambaran umum suatu sistem yang terdapat di dalam suatu organisasi yang memperlihatkan batasan (*boundary*) sistem, adanya interaksi antara *eksternal entity* dengan suatu sistem dan informasi secara umum mengalir di antara *entity* dan sistem. *Context Diagram* merupakan alat bantu yang digunakan dalam menganalisa sistem yang akan dikembangkan. Simbol-simbol

yang digunakan dalam *Context Diagram* hampir sama dengan simbol-simbol yang ada pada DFD, hanya saja pada *Context Diagram* tidak terdapat simbol *file*. Beberapa simbol yang digunakan dalam *Context Diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol dari *Context Diagram*

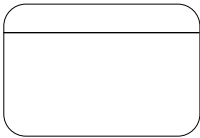
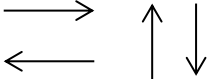
Simbol	Keterangan
	Kesatuan Luar (<i>Eksternal Entity</i>) Merupakan kesatuan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di luar lingkungan yang akan memberikan <i>input</i> atau menerima <i>output</i> sistem.
	Proses (<i>Process</i>) Kegiatan atau kerja yang dilakukan oleh mesin atau komputer dari suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang keluar dari proses.
	Arus Data (<i>Data Flow</i>) Arus data mengalir di antara proses, simpanan data dan kesatuan. Arus data ini menunjukkan arus data dari yang masuk ke dalam proses sistem

Sumber : Khelvin Ovela Putra, et al., *Aplikasi Penjualan Alat Tulis Kantor Pada Alif Photocopy*, 2022.

2.2.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu diagram yang menggambarkan aliran data dari sebuah proses yang disebut dengan sistem informasi (Hastuti, dkk., 2023). *Data flow diagram* juga menyediakan informasi mengenai *input* dan *output* dari tiap entitas dan proses itu sendiri. Simbol-simbol dari *Data Flow Diagram* (DFD) dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol dari *Data Flow Diagram*

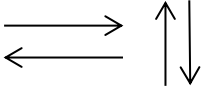
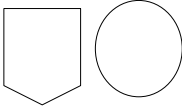
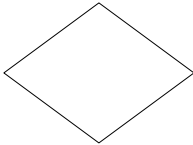
Simbol	Keterangan
	Kesatuan Luar (<i>Eksternal</i>). <i>Entity</i> merupakan sumber atau tujuan data, dapat berupa bagian atau orang yang berada di luar sistem tapi berhubungan dengan sistem tersebut.
	Proses. Simbol ini digunakan untuk melakukan proses pengolahan data, yang menunjukkan suatu kegiatan yang mengubah aliran data yang masuk menjadi keluaran.
	Penyimpanan Data (<i>Data Store</i>) Merupakan tempat penyimpanan dokumen-dokumen atau <i>file-file</i> yang dibutuhkan.
	Aliran Data. Menunjukkan arus data dalam proses.

Sumber : Khelvin Ovela Putra, et al., *Aplikasi Penjualan Alat Tulis Kantor Pada Alif Photocopy*, 2022.

2.2.3 *Flowchart*

Bagan alir (*Flowchart*) adalah bagan yang menunjukkan aliran di dalam sebuah program atau prosedur dari sebuah sistem secara logika. Bagan alir digunakan terutama untuk alat bantu desain proses. Simbol-simbol yang digunakan dalam *Flowchart* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-Simbol Standar dalam *Flowchart*

Simbol	Keterangan
	Simbol <i>Input/Output</i> . Digunakan untuk mewakili data <i>input/output</i> .
	Simbol Proses. Digunakan untuk mewakili suatu proses.
	Simbol Garis Alir. Digunakan untuk menunjukkan arus dari proses.
	Simbol penghubung. Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus. Baik pada halaman yang sama maupun di halaman berikutnya.
	Simbol Keputusan. Digunakan untuk suatu penyelesaian kondisi di dalam program.
	Simbol Proses Terdefinisi. Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
	Simbol Persiapan. Digunakan untuk memberi/menset nilai awal suatu besaran.
	Simbol Titik Terminal. Digunakan untuk menunjukkan awal dan akhir dari suatu proses.

Sumber : Khelvin Ovela Putra, et al., Aplikasi Penjualan Alat Tulis Kantor Pada Alif Photocopy, 2022.

2.3 Sistem Kontrol

Sistem kontrol (*control system*) merupakan suatu kumpulan cara atau metode yang dipelajari dari kebiasaan-kebiasaan manusia dalam bekerja, di mana manusia membutuhkan suatu pengamatan kualitas dari apa yang telah mereka kerjakan

sehingga memiliki karakteristik sesuai dengan yang diharapkan pada mulanya. Sistem kontrol berdasarkan cara kerjanya dapat dibagi menjadi dua bagian, yaitu sistem kontrol *Loop* terbuka dan kontrol *Loop* tertutup.

2.3.1 Sistem *Loop* Terbuka

Sistem pengendali *Loop* terbuka adalah sistem pengendalian yang keluarannya (*process variable*) tidak dapat dibandingkan dengan masukannya (*set point*). Ini sama dengan sistem kerja mesin cuci di mana perendaman, pencucian, dan pembilasan dilakukan atas basis waktu. Diagram mengenai sistem pengendali *Loop* terbuka dapat dilihat pada Gambar 2.2.

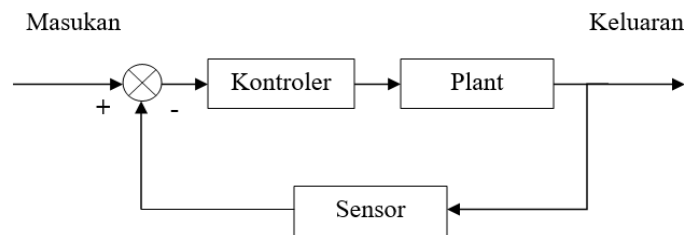


Sumber : Habib Gushardi & Delsina Faiza, Perancangan dan Pembuatan Alat Penghitung Jumlah Uang Otomatis Terintegrasi Internet of Things, 2022.

Gambar 2.2 Sistem Pengendalian *Loop* Terbuka

2.3.2 Sistem *Loop* Tertutup

Sistem pengendali *Loop* tertutup adalah sistem pengendalian yang keluaran dapat dibandingkan dengan masukannya. Sebagai contoh sistem pengendali *Loop* tertutup adalah pengendalian temperatur reaktan (*process variable*) secara otomatis pada sebuah *reaktor batch* yang telah disajikan. Adapun diagram dari sistem pengendali *Loop* tertutup dapat dilihat seperti pada Gambar 2.3.



Sumber : Habib Gushardi & Delsina Faiza, Perancangan dan Pembuatan Alat Penghitung Jumlah Uang Otomatis Terintegrasi Internet of Things, 2022.

Gambar 2.3 Sistem Pengendalian *Loop* Tertutup

2.4 Komponen Utama

Pada bagian ini akan dijelaskan komponen utama yang digunakan pada perancangan sistem ini adalah sebagai berikut :

2.4.1 Arduino Uno

Arduino uno adalah papan mikrokontroler berbasis ATmega328 yang memiliki 14 pin digital *input/output* dimana 6 pin dapat digunakan sebagai *output* PWM (*Pulse Width Modulation*), 6 pin *analog*, *clock speed* 16 MHz, koneksi USB, penghubung daya, header ISCP, dan tombol *reset*. *Board* ini menggunakan daya yang terhubung ke komputer dengan kabel USB atau daya eksternal dengan adaptor AC-DC atau baterai. Bentuk dari mikrokontroler Arduino Uno ditunjukkan oleh Gambar 2.4.



Sumber : Rakhmat Sudrajat & Fahimatu Rofifah, Rancang Bangun Sistem Kendali Kipas Angin dengan Sensor Suhu dan Sensor Ultrasonik Berbasis Arduino Uno, 2023.

Gambar 2.4 Arduino Uno

2.4.2 ESP32

Mikrokontroler NodeMcu ESP32 yang dikenal sebagai *Espressif System* merupakan pengembangan dari mikrokontroler ESP8266. Secara Spesifikasi ESP32 sangat lengkap, sehingga microcontroller ini sangat tepat untuk digunakan terutama untuk aplikasi yang berhubungan dengan *Internet Of Things* (IoT), dikarenakan mikrokontroler ini dapat berkomunikasi menggunakan *Wifi*, BLE dan *Bluetooth*. Bentuk dari mikrokontroler ESP32 ditunjukkan oleh Gambar 2.5.



Sumber : Deni Muhammad Sepudina & Syahid Abdullah, Sistem Keamanan Pintu Rumah Berbasis Internet of Things Berbasis NodeMCU ESP32 dan Telegram, 2022.

Gambar 2.5 ESP32

2.4.3 Coil

Induktor atau *coil* adalah komponen elektronika pasif yang terdiri dari susunan lilitan kawat yang membentuk sebuah kumparan. Induktor pada dasarnya dapat menimbulkan medan magnet jika dialiri oleh arus listrik. Medan magnet yang ditimbulkan tersebut dapat menyimpan energi dalam waktu yang relatif singkat. Bentuk dari *Coil* ditunjukkan oleh Gambar 2.6.



Sumber : Khairina, dkk., Perancangan Alat Pendeteksi Konduktifitas Logam (Emas) Pada Studi Kasus Desa Panton Luas, Sawang, 2021.

Gambar 2.6 Coil

2.4.4 Fingerprint

Fingerprint adalah salah satu bentuk biometrik, yaitu ilmu yang menggunakan karakteristik fisik penduduk untuk mengidentifikasi. Teknologi identifikasi sidik jari berdasarkan fakta bahwa setiap sidik jari adalah unik. Verifikasi sistem menggunakan kontur dan flat image dari jari dan membandingkannya (Ngantung et al., 2014). *Fingerprint* dilengkapi dengan sensor untuk membaca jari. Sensor untuk mendeteksi target dengan merubah besaran non listrik menjadi besaran listrik. Sensor merupakan rangkaian atau komponen paling depan di target (Royhan et al., 2020). Bentuk dari *Fingerprint* ditunjukkan oleh Gambar 2.7.



Sumber : Muhamad Royhan, Fingerprint Untuk mengunci Pintu Terintegrasi Dengan Arduino, 2021.

Gambar 2.7 Fingerprint

2.4.5 Telegram

Telegram adalah Aplikasi pesan chatting yang memungkinkan pengguna untuk mengirimkan pesan chatting rahasia yang dienkripsi *end-to-end* sebagai keamanan tambahan dengan Telegram dimungkinkan berbagi lebih dari sekedar gambar dan video, tapi Telegram juga memungkinkan mentransfer dokumen atau mengirim lokasi (Hergika Gusti, dkk., 2021). Bentuk dari aplikasi telegram dapat dilihat pada Gambar 2.8.



Sumber : Firman Rany Muid, Rancang Bangun Sistem CCTV Dengan Sensor Gerak Menggunakan Arduino Dan Telegram, 2022.

Gambar 2.8 Aplikasi Telegram

2.4.6 LCD (*Liquid Crystal Display*)

Solenoid Dalam modul LCD (*Liquid Crystal Display*) terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan register. Memori yang digunakan mikrokontroler internal LCD adalah:

1. DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan.
2. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter yang bisa diubah-ubah sesuai dengan keinginan.
3. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen sehingga pengguna tinggal mengambilnya sesuai alamat memorinya.

Bentuk fisik dari LCD (*Liquid Crystal Display*) dapat dilihat pada Gambar 2.9.



Sumber : Muhammad Tsatsa Firdausi & Ratnasari Nur Rohmah, Perancangan Sistem Otomatis Pengatur Suhu Dan Kelembaban Pada Kandang Jangkrik Di Daerah Masaran Sragen Berbasis Mikrokontroller Arduino, 2022.

Gambar 2.9 LCD (*Liquid Crystal Display*)

2.4.7 LED

LED adalah salah satu komponen semikonduktor yang termasuk dalam jenis dioda. Sama-sama memiliki kutub positif dan kutub negatif, hanya saja LED memancarkan cahaya ketika diberikan tegangan dari anoda ke katoda. Hal yang perlu diperhatikan adalah cara mengetahui polaritas dari sebuah LED, cara yang digunakan untuk mengetahuinya adalah dengan memperhatikan kedua kaki LED, dimana kaki yang lebih panjang menunjukkan kutub positif (anoda) dan yang pendek adalah kutub negatif (katoda). Beberapa jenis warna LED, diantaranya adalah merah, kuning, hijau, biru, dan putih. Perbedaan warna tersebut terjadi karena perbedaan bahan semikonduktor yang digunakan. Saat ini teknologi LED memiliki banyak kelebihan seperti hemat listrik, tidak menimbulkan panas, lebih tahan lama, dan bentuknya yang kecil memudahkan dalam berbagai penggunaan. Bentuk dari sensor LED dapat dilihat pada Gambar 2.10.



Sumber : Muhamad Abdul Azis, dkk., Spend (Sistem Peringatan Dini Banjir Menggunakan Water Level Sensor Dengan Arduino Uno), 2024.

Gambar 2.10 Light Emitting Diode (LED)

2.4.8 Buzzer

Buzzer merupakan sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer

hampir sama dengan *loud speaker*, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak balik sehingga membuat udara bergetar yang akan menghasilkan suara sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat. Bentuk dari Buzzer ditunjukkan oleh Gambar 2.11.



Sumber : Alfiru Nur Alfa & Viki Ramadhan, Prototype Detektor Gas Dan Monitoring Suhu Berbasis Arduino Uno, 2022

Gambar 2.11 Buzzer

2.5 Komponen Pendukung

Adapun komponen yang dipakai dalam perancangan sistem, adalah sebagai berikut:

2.5.1 Resistor

Resistor adalah komponen elektronika yang berfungsi untuk menghambat atau membatasi aliran listrik yang mengalir dalam suatu rangkaian elektronika. Resistor termasuk komponen pasif pada rangkaian elektronika. Sebagaimana fungsi resistor yang sesuai namanya bersifat resistif dan termasuk salah satu komponen

elektronika dalam kategori komponen pasif. Satuan atau nilai resistansi suatu resistor disebut Ohm dan dilambangkan dengan simbol Omega (Ω). Hukum Ohm menyatakan bahwa resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya. Selain nilai resistansi (Ohm), resistor juga memiliki nilai yang lain seperti nilai toleransi dan kapasitas daya yang mampu dilewatkannya. Resistor adalah komponen pasif dalam rangkaian listrik yang berfungsi untuk menghambat aliran arus listrik, sehingga dapat mengatur distribusi tegangan dan arus dalam sirkuit. Cara kerja resistor melibatkan pengubahan sebagian energi listrik menjadi panas ketika arus melewati material resistif, yang memungkinkan resistor mengontrol jumlah arus yang mengalir ke berbagai komponen lain dalam rangkaian. Dengan cara ini, resistor membantu menjaga kestabilan rangkaian dan melindungi komponen lain dari potensi kerusakan akibat arus berlebih. Gambar 2.12 akan memperlihatkan bentuk fisik dari resistor.



Sumber : Bagus Dwi Cahyono, dkk., Pengembangan Media Pembelajaran Elektronika Dasar Untuk Memahami Nilai Resistor Berdasarkan Kode Warna 3 Gelang Dan 4 Gelang Bagi Siswa SMK Kelas X Jurusan Teknik Otomasi Industri, 2023.

Gambar 2.12 Resistor

Untuk mengetahui nilai-nilai yang terkandung pada setiap lingkaran cincin pada resistor, maka dapat dilihat pada Tabel 2.4.

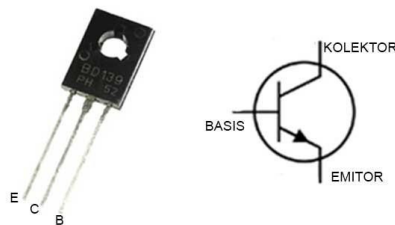
Tabel 2.4 Kode Warna Resistor

Warna	Gelang Ke -				Toleransi
	1	2	3		
			Pengali*	Nilai**	
Hitam	0	0	10 ⁰	0	-
Cokelat	1	1	10 ¹	1	±1%(F)
Merah	2	2	10 ²	2	±2%(F)
Jingga	3	3	10 ³	3	-
Kuning	4	4	10 ⁴	4	-
Hijau	5	5	10 ⁵	5	±0.5%(D)
Biru	6	6	10 ⁶	6	±0.25%(C)
Ungu	7	7	10 ⁷	7	±0.1%(B)
Abu-abu	8	8	10 ⁸	8	±0.05%(A)
Putih	9	9	10 ⁹	9	-
Perak	-	-	0.1	-	±5%(J)
Emas	-	-	0.01	-	±10%(K)
Tanpa warna	-	-	-	-	±20%(M)

Sumber : Bagus Dwi Cahyono, dkk., Pengembangan Media Pembelajaran Elektronika Dasar Untuk Memahami Nilai Resistor Berdasarkan Kode Warna 3 Gelang Dan 4 Gelang Bagi Siswa SMK Kelas X Jurusan Teknik Otomasi Industri, 2023.

2.5.2 Transistor

Nama transistor berasal dari kata *transfer* dan resistor yang artinya adalah merubah bahan dari bahan yang tidak dapat menghantar arus listrik menjadi bahan penghantar atau setengah penghantar (semikonduktor). Transistor adalah komponen aktif. Transistor sendiri diciptakan oleh tiga orang bangsa Amerika yang bernama John Bardeen, Walten Brattain dan William Shockley pada tahun 1948. Sama halnya dengan komponen semikonduktor lainnya, transistor dibuat daribahan Indium, Germanium dan Silikon. Transistor mempunyai berbagai macam jenis, diantaranya transistor *bipolar* dan transistor *unipolar* atau transistor efek medan (FET). Transistor *bipolar* memiliki 3 buah terminal yang membentuk tiga buah elektrode atau kaki, yaitu kaki emiter disingkat dengan huruf E, kaki basis disingkat dengan huruf B dan kakikolektor disingkat dengan huruf C. Bentuk gambar dan simbol dari transistor dapat dilihat pada Gambar 2.13.



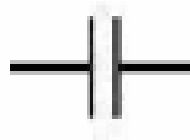
Sumber : Raita Rahmatina, dkk., Implementasi Transistor BD139 dan Rangkaian Relay pada Mesin Air, 2023.

Gambar 2.13 Gambar dan Simbol Transistor

2.5.3 Kapasitor

Kapasitor (*Capacitor*) atau disebut juga dengan Kondensator (*Condensator*) adalah komponen elektronika pasif yang dapat menyimpan muatan listrik dalam waktu sementara dengan satuan kapasitansinya adalah Farad. Kapasitor merupakan

komponen elektronika yang terdiri dari 2 pelat konduktor yang pada umumnya adalah terbuat dari logam dan sebuah isolator diantara pelat tersebut sebagai pemisah. Isolator tersebut disebut juga dengan dielektrika. Kapasitor (Kondensator) yang dalam rangkaian elektronika dilambangkan dengan huruf "C" adalah suatu alat yang dapat menyimpan energi/muatan listrik di dalam medan listrik, dengan cara mengumpulkan ketidakseimbangan internal dari muatan listrik. Bentuk dasar sebuah kapasitor dapat dilihat pada Gambar 2.14.



Sumber : Achmad Sumbaryadi & Dian Gustina, Pensaklaran Otomatis Cooler Pad Menggunakan Sensor Suhu LM 35 Dengan Mikrokontroler ATMEGA8535, 2021.

Gambar 2.14 Simbol Kapasitor

Berikut jenis kapasitor yang paling sering digunakan yang umumnya beredar di pasaran:

1. Kapasitor Keramik

Bentuk kapasitor keramik ada yang bulat tipis, ada yang persegi empat berwarna merah, hijau, coklat dan lain-lain. Ukuran nilai biasanya memiliki kapasitansi kecil yaitu dari 1pF (piko Farad) sampai 0.1 uF (mikro Farad) namun rating tegangannya sangat rendah, seperti yang ditunjukkan pada Gambar 2.15 Kapasitor Keramik.



Sumber : Reza Pebrian, dkk., Analisa Komponen Elektronika Pada Switching Power Supply Tipe Oska Dengan Metode Inspeksi Visual Dan Menggunakan Multimeter Digital, 2024.

Gambar 2.15 Kapasitor Keramik

2. Kapasitor Polyster (*Mylar*)

Bentuk kapasitor polyster persegi empat seperti permen kemudian mempunyai warna merah, hijau, coklat dan sebagainya. Pada jenis ini sesuai namanya pada bahan isolatornya adalah terdiri dari polyester yang kebanyakan toleransinya berkisar 5 – 10 %. Bentuk fisik kapasitor polyester ini adalah kotak dan tidak ada polaritasnya. Range nilainya cukup bervariasi namun umumnya dalam kapasitas yang jelas dan penggunaan tegangan yang rendah. Toleransinya yang cukup besar maka mengakibatkan tidak digunakannya pada rangkaian frekuensi tinggi atau pada rangkaian dengan arus listrik yang besar, tetapi kapasitor polyester ini banyak juga diterapkan pada rangkaian *power supply*. Gambar 2.16 menunjukkan bentuk fisik dari kapasitor Polyster.



Sumber : Nezar Mubarak, Peran Kapasitor Milar Dalam Mengurangi Noise Pada Amplifier, 2021.

Gambar 2.16 Kapasitor Polyster

3. Kapasitor Elektrolit

Kapasitor elektrolit disebut juga dengan kapasitor elko yaitu jenis kapasitor yang banyak digunakan dan umumnya berbentuk tabung. Dalam pemasangannya harus hati-hati karena memiliki polaritas (+) dan polaritas (-). Jika terbalik dalam pemasangannya maka sangat fatal akibatnya karena kapasitor tersebut bisa meledak. Berikut contoh fisik kapasitor elektrolit dapat dilihat pada Gambar 2.17.

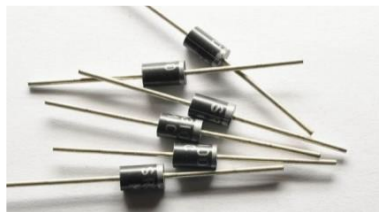


Sumber : Reza Pebrian, dkk., Analisa Komponen Elektronika Pada Switching Power Supply Tipe Oska Dengan Metode Inspeksi Visual Dan Menggunakan Multimeter Digital, 2024.

Gambar 2.17 Kapasitor Elektrolit

2.5.4 Dioda

Dioda salah satu komponen aktif sangat populer digunakan dalam rangkaian elektronika, karena bentuknya sederhana dan penggunaannya sangat luas. Ada beberapa macam rangkaian diode, diantaranya, penyearah setengah gelombang (*Half-Wave Rectifier*), penyearah gelombang penuh (*Full-Wave Rectifier*), rangkaian pemotong (*Clipper*), rangkaian penjepit (*Clamper*) maupun pengganda tegangan (*Voltage Multiplier*). Adapun bentuk fisik jenis dioda dapat dilihat pada Gambar 2.18.



Sumber : Muhammad Khair, Rancang Bangun Media Pembelajaran Praktikum Piranti Elektronika Untuk Memahami Karakteristik Dioda, 2020.

Gambar 2.18 Dioda

2.6 Bahasa Pemrograman Arduino

Bahasa pemrograman *Arduino* yang dipakai adalah bahasa C. Bahasa C adalah bahasa yang lazim digunakan sejak awal komputer diciptakan dan sangat berperan dalam perkembangan *software*. *Arduino* menggunakan bahasa C yang mengizinkan *multi-platfrom*. *Software Arduino* pin bisa dijalankan pada semua mesin operasi yang umum, misalnya windows, Linux dan MacOS.

2.6.1 Struktur

Dasar struktur dari bahasa pemrograman *Arduino* sangatlah mudah dan sederhana. Agar program dapat berjalan dengan baik, maka perlu setidaknya dua bagian atau fungsi, yaitu `setup()` yang dipanggil hanya satu kali, biasanya untuk

inisialisasi program (setting *input* atau setting serial, dan lain-lain) dan *Loop()* tempat untuk mengeksekusi program secara berulang, biasanya untuk membaca *input*, atau men-*trigger output*.

a. *Setup()*

fungsi *setup()* adalah nama fungsi yang telah disediakan oleh Arduino untuk menyatakan fungsi yang akan dijalankan pertama kali. Fungsi ini berisi kode-kode untuk kepentingan inisialisasi.

b. *Loop()*

fungsi *Loop()* adalah fungsi yang secara otomatis dijalankan oleh arduino setelah fungsi *setup()* dieksekusi. Seluruh kode yang ada di fungsi dengan sendirinya akan diulang terus-menerus. Satu-satunya yang bisa menghentikan eksekusi *Loop()* adalah berhentinya pasokan catu daya ke papan Arduino.

2.6.2 Syntax Pemrograman

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan (Aldyrazor, 2020).

a. `//` Komentar satu baris

Pemberian komentar setelah penulisan kode. Apapun yang ditulis dibelakang simbol ini akan diabaikan oleh program.

b. `/* */` Komentar banyak baris

Pemberian komentar pada kode yang ditulis jika memiliki komentar lebih dari satu baris program.

c. `{}` Kurung kurawal

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir.

d. ; Titik koma

Setiap baris kode harus diakhiri dengan titik koma, jika tidak maka program tidak bisa dijalankan.

2.6.3 Tipe Data Arduino

Tipe data yang digunakan di dalam program *sketch* bermacam-macam.

Tipe data pada Arduino dapat dilihat pada Tabel 2.5 (Eko Mardianto, 2022).

Tabel 2.5 Variabel dalam Arduino

Tipe Data	Ukuran (BIT)	JANGKAUAN NILAI
Boolean	1	0, 1 (tipe data bit hanya digunakan untuk variabel global)
Char	8	-128 to 127
<i>Unsigned char</i>	8	0 sampai 255
<i>Byte</i>	8	0 sampai 255
Int	16	32768 sampai 3767
<i>Unsigned int</i>	16	0 sampai 65535
<i>Word</i>	16	0 sampai 65535
<i>Long</i>	32	-2147483648 sampai 2147483647
<i>Unsigned long</i>	32	0 sampai 4294967295
<i>Short</i>	16	-32768 sampai 32767
<i>Float / double</i>	32	$\pm 1,175\text{e-}38$ sampai $\pm 3,402\text{e}38$

Sumber : Eko Mardianto, *Panduan Belajar Mikrokontroller Arduino*, 2022.

2.6.4 Operator

Ada dua jenis operator di Arduino yaitu operator aritmatika yang digunakan untuk operasi matematika dan operator perbandingan yang biasanya digunakan untuk operasi percabangan untuk membandingkan dua nilai/variabel (Wijaya et al., 2021).

2.6.4.1 Operator Aritmatika

Operator aritmatika merupakan operator yang digunakan untuk operasi aritmatika seperti penjumlahan, pengurangan, perkalian, atau pembagian. Tabel 2.6 akan menunjukkan contoh operator aritmatika.

Tabel 2.6 Operator Aritmatika

Operator	Keterangan
+	Operasi penjumlahan
-	Operasi pengurangan
*	Operasi perkalian
/	Operasi pembagian
%	Operasi sisa pembagian (modulus)

Sumber : Eko Mardianto, Panduan Belajar Mikrokontroller Arduino, 2022.

2.6.4.2 Operator Kondisi

Operator kondisi biasa digunakan untuk membandingkan dua nilai. Keseluruhan operator relasional dapat dilihat di Tabel 2.7.

Tabel 2.7 Operator Kondisi

Operator	Keterangan
==	Bernilai benar bila kedua data sama dan bernilai salah bila sebaliknya
!=	Tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar dari atau sama dengan
<=	Lebih Kecil atau sama dengan

Sumber : Eko Mardianto, Panduan Belajar Mikrokontroller Arduino, 2022.

2.6.4.3 Operator Logika

Operator yang digunakan untuk membuat kesimpulan logis dari 2 kondisi boolean yaitu *true* atau *false*. Tabel 2.8 menunjukkan beberapa jenis operator

logika.

Tabel 2.8 Operator Logika

Operator	Keterangan
&&	Operator untuk logika AND
	Operator untuk logika OR
!	Operator untuk logika NOT

Sumber: Eko Mardianto, Panduan Belajar Mikrokontroller Arduino, 2022.

2.6.5 Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan (Eko Mardianto, 2022).

2.6.5.1 Kondisi *If*

Pernyataan *if* melakukan pengetesan kondisi jika kondisi tersebut telah terpenuhi, seperti input analog yang diterima telah berada pada kondisi tertentu. Maka akan mengeksekusi semua statement yang berada dalam kurung jika statement tersebut true, dan akan mengeksekusi semua perintah yang berada dalam kurung { ... }. Format penulisannya sebagai berikut (Eko Mardianto, 2022):

```
If(kondisi) { Pernyataan/perintah;
}
```

2.6.5.2 Kondisi *If-else*,

Pernyataan *if ... else* memberikan kejelasan dalam menangani lebih dari satu statement. Misalnya jika input yang terbaca adalah High maka lakukan action A, namun jika Low lakukan action B. Bentuk format penulisannya yaitu (Eko Mardianto, 2022) :

```
If(kondisi){
//Pernyataan_1;
else {Pernyataan_2; }
```

2.6.5.3 For

Pernyataan *for* digunakan untuk melakukan perulangan yang terdapat pada statement di dalam {}. Untuk bisa melakukan perulangan maka terdapat sebuah counter yang akan menaikkan hitungan secara satu per satu, dan memberikan tanda kapan perulangan akan berakhir. Pernyataan *for* sangat berguna untuk operasi yang memerlukan perulangan dan sering digunakan bersama dengan sekelompok array, yang biasanya digunakan untuk menyimpan koleksi data atau pin. Bentuk umum pernyataan *for* seperti berikut (Eko Mardianto, 2022) :

```
for (ungkapan_1; ungkapan_2; ungkapan_3)
Pernyataan; }
```

2.6.5.4 While

Perulangan menggunakan *while* akan terus berlangsung terus menerus sampai ekspresi dalam kurung tutup () bernilai false. Agar perulangan bisa berhenti maka harus ada sesuatu yang bisa mengubah variable, atau perulangan *while* tidak akan pernah berhenti. Sesuatu yang dimaksud bisa berasal dalam kode pemrograman, seperti variable counter menambah satu nilai variable, atau masukan dari luar seperti sensor. Bentuk umum pernyataan *while* seperti berikut ini (Eko Mardianto, 2022) :

```
while (ungkapan_1)
{pernyataan;
}
```

2.7 Software Arduino IDE

Pada *software* arduino IDE terdapat beberapa menu yang memiliki fungsi yang berbeda-beda. Beberapa menu yang terdapat pada *software* Arduino IDE

adalah *File*, *Edit*, *Sketch*, *Tools* dan *help* seperti Gambar 2.19.



Sumber : Retno Devita, et, al., Pengontrolan Pola Dancing Fountain Berirama Music Menggunakan Android Berbasis Mikrokontroler Arduino, 2020.

Gambar 2.19 Software Arduino IDE

Menu *File* terdiri dari beberapa pilihan, seperti misalnya untuk membuat *sketch* baru, menyimpan *sketch*, membuka preferences, pilihan untuk keluar dari program dan yang lainnya. Pada menu *Edit* terdapat pilihan seperti *Copy*, *Paste*, *Cut*, *Select All* untuk menyeleksi semua kode yang sudah ditulis dan yang lainnya. Pada menu *sketch* terdapat pilihan seperti *Verify* yang digunakan untuk memverifikasi *sketch* yang telah dibuat, kemudian pilihan *Upload* yang digunakan untuk mengunggah *sketch* yang telah di buat dan dikompilasi ke Arduino.

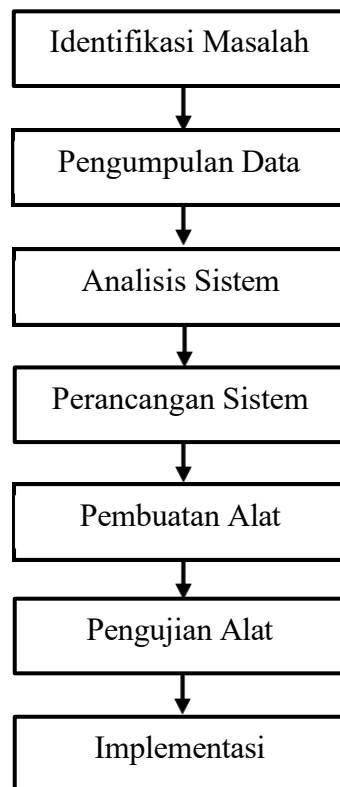
Selanjutnya terdapat pilihan *Include Library* yang didalamnya mencakup pilihan library arduino yang akan digunakan. Pada menu *Tools* terdapat beberapa pilihan submenu. Submenu yang biasadigunakan adalah pilihan untuk memilih jenis *board* Arduino yang digunakan dan pilihan untuk *port* COM dimana Arduino tersebut terhubung dengan komputer. Tombol *Serial Monitor* yang terdapat diujung sebelah kanan dapat digunakan untuk melihat data-data berupa karakter, angka maupun *text* yang dikirimkan dari Arduino ke komputer.

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Kerja Penelitian

Kerangka kerja merupakan langkah-langkah yang akan dilakukan dalam rangka penyelesaian masalah yang akan dibahas. Adapun kerangka kerja dari penelitian ini dapat disajikan pada Gambar 3.1 dibawah ini.



*Sumber : Pati, Mi bnu et al. Sistem Pakar dengan Metode Forward Chaining untuk
Diagnosis Penyakit dan Hama Tanaman Semangka. 2020.*

Gambar 3.1 Kerangka Kerja Penelitian

3.2 Uraian Kerangka Kerja Penelitian

Adapun tahapan-tahapan dalam penyelesaian penelitian ini adalah sebagai berikut:

3.2.1 Identifikasi Masalah

Identifikasi masalah dilakukan dengan melakukan pendekatan terhadap objek penelitian. Tujuan dari tahap ini adalah untuk mengetahui permasalahan yang terjadi secara tepat, sehingga diharapkan penelitian dapat memberikan solusi yang paling optimal terhadap pemecahan permasalahan tersebut.

Dalam penelitian ini, permasalahan yang dijadikan potensi pada sistem ini oleh penulis, yaitu industri pengolahan kayu sering kali susah mengidentifikasi benda logam pada kayu yang akan diproses, seperti paku atau serpihan logam kecil yang tersisa pada kayu bekas. Benda logam yang tidak terdeteksi dalam bahan baku kayu dapat menyebabkan kerusakan pada mesin pemotong dan membahayakan keselamatan kerja selain itu, proses pendeteksian logam secara manual sering kali tidak efisien dan tidak selalu akurat, mengakibatkan potensi kerugian waktu dan biaya. Hal ini mengakibatkan kebutuhan akan sistem otomatis yang lebih efisien dan terintegrasi untuk meminimalkan risiko dan meningkatkan keamanan operasional.

3.2.2 Pengumpulan Data

Data merupakan bagian penting dalam penelitian demi kelancaran penelitian. Pengumpulan data dapat dilakukan dengan melakukan studi-studi dan pengujian terhadap konsep-konsep yang ada dan berkenaan dengan penelitian. Dalam pengumpulan data, dilakukan pengujian terhadap komponen-komponen yang

digunakan pada alat *metal detector* berbasis IoT.

3.2.2.1 Metode Penelitian

Dalam melakukan penelitian agar mendapatkan hasil seperti yang diinginkan, maka sekiranya diperlukan suatu metodologi penelitian yang umum dilakukan yaitu:

1. Penelitian Lapangan (*field research*)

Penelitian yang dilakukan dengan cara riset kelapangan untuk memperoleh informasi yang dibutuhkan serta membandingkan dengan peralatan yang dapat mendukung kelancaran pembuatan *system*.

2. Penelitian Perpustakaan (*library research*)

Penelitian ini dilakukan untuk melengkapi perbendaharaan kaidah, konsep, teori dan lain-lain, sehingga menjadi suatu yang mempunyai landasan dan keilmuan yang mantap, selain itu penelitian ini juga melakukan penelitian pada beberapa material yang sudah ada, baik itu buku-buku, jurnal-jurnal serta yang ada hubungannya dengan penelitian ini maupun catatan yang dilakukan selama perkuliahan. Penelitian ini ditujukan untuk mengumpulkan semua data yang sangat dibutuhkan dalam penelitian ini.

3. Penelitian Laboratorium (*laboratory research*)

Penelitian laboratorium ini dimaksudkan untuk melakukan pengujian terhadap penerapan sistem yang dibuat. Pada penelitian laboratorium ini tidak lepas dari piranti atau perangkat yang digunakan, di mana perangkat ini dapat digunakan untuk membantu penulis untuk melakukan pengujian.

Adapun objek yang diuji spesifikasi *hardware* dan *software* yang digunakan dalam penulisan ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Hardware dan Software

<i>Hardware</i>	<i>Software</i>
Satu Unit Laptop Acer Prosesor Intel(R) Core i3-2330M CPU @ 2.20 GHz (4 CPUs), 2.2 GHz Memory 4 GB RAM HDD 1 TB	Sistem Operasi Windows 11 Professional 64 bit dilengkapi Microsoft Office 2013 Bahasa pemrograman Arduino Google SketchUp Microsoft Visio Fritzing

3.2.2.2 Waktu Penelitian

Penelitian dengan memproses data-data yang dapat oleh penulis, pengambilan data dilakukan dari bulan Maret 2025 hingga Agustus 2025, yang dapat dilihat pada tabel 3.2.

Tabel 3.2 Jadwal Penelitian

Kegiatan	2025																							
	Maret				April				Mei				Juni				Juli				Agustus			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																								
Pengumpulan Data																								
Analisis Sistem																								
Perancangan Sistem																								
Pembuatan Alat																								
Pengujian Sistem																								
Implementasi Sistem																								

3.2.3 Analisis Sistem

Berdasarkan identifikasi masalah penelitian, peneliti melakukan analisa sistem terlebih dahulu. Hal ini bertujuan agar pemecahan masalah dapat menghasilkan sebuah solusi yang baru sebelum melakukan sebuah perancangan alat.

Pada analisa sistem, peneliti ingin menjelaskan *input*, proses dan *output* dari sistem yang akan dibuat. Pada analisa sistem, peneliti ingin menjelaskan *input*, proses dan *output* dari sistem yang akan dibuat. *Input* adalah sebuah masukan yang

akan diproses dan menghasilkan keluaran yaitu *output*. *Input* yang digunakan adalah *Coil*, *Fingerprint*, *Push button*, Telegram dan *DFPlayer*.

Fingerprint digunakan sebagai keamanan pada alat kemudian data dari pengecekan akan dikirimkan ke telegram. ESP32 digunakan sebagai mikrokontroler. *Output* yang dihasilkan adalah Hasil pembacaan seluruh sensor yang dikirim ke telegram.

3.2.4 Perancangan Sistem

Pada tahap ini akan membuat sebuah perancangan sistem yang akan dijalankan. Desain produk ini akan dimulai dari menganalisa program yang sedang berjalan, merancang program yang akan dibangun guna menjalankan sistem yang akan dibuat menggunakan Arduino Uno. Pada tahap ini juga dilakukan pengumpulan fakta-fakta yang mendukung perancangan sistem. Selain dari bentuk fisik sistem, ada beberapa *tools* yang digunakan dalam merancang sistem yang akan dibuat yaitu :

1. *Context Diagram*

Context diagram menggambarkan rancangan keseluruhan, *entity external* harus digambarkan sedemikian rupa, sehingga terlihat data yang mengalir pada input-proses-output.

2. *Data Flow Diagram (DFD)*

Data Flow Diagram (DFD) merupakan suatu diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sistem ke entitas. DFD juga dapat diartikan sebagai teknik grafis yang menggambarkan alir data dan transformasi yang digunakan sebagai perjalanan data dari *input* atau masukan menuju keluaran *output*.

3. *Blok Diagram*

Blok Diagram yang dibuat untuk memetakan proses kerja pada suatu alat, hal ini bertujuan untuk memudahkan seseorang dalam mengenal komponen-komponen elektronika dari alat dan memahami alur kerja didalamnya.

4. *Flowchart*

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah, khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

5. *Design*

Desain dari alat yang dibuat berupa gambaran dan bentuk awal dari alat secara keseluruhan. Dengan adanya desain ini maka prinsip kerja dari alat serta komponen-komponen dari sistem yang digunakan akan dapat dilihat dengan jelas. *Software* yang digunakan untuk perancangan alat adalah google sketchup.

3.2.5 Pembuatan Sistem

Pembuatan sistem didasari oleh beberapa pertimbangan-pertimbangan yang telah disebutkan pada perancangan sistem sebelumnya seperti *context diagram*, *data flow diagram*, *blok diagram*, *flowchart*, dan *design*.

3.2.6 Pengujian Sistem

Pada tahap ini alat yang dirancang akan diuji dan Analisa agar sesuai dengan rancangan sebelum alat diimplementasikan. Adapun tahap pengujian alat adalah

sebagai berikut:

1. Hubungkan alat ke sumber tegangan.
2. Cocokkan sidik jari ke *fingerprint*, hasil dari pengecekan sidik jari ini akan dikirimkan ke telegram. Telegram dapat menambah dan menghapus data sidik jari petugas yang akan menggunakan alat.
3. Buzzer akan berbunyi apabila sidik jari tidak terdaftar.
4. Lakukan pengujian ke kayu yang terdapat bahan berjenis besi atau timah. LED 1 akan hidup apabila alat mendeteksi adanya benda timah yang terdapat pada kayu, sedangkan LED 2 akan aktif apabila alat mendeteksi adanya benda besi yang terdapat pada kayu.
5. Telegram akan menerima notifikasi saat adanya benda besi ataupun timah yang terdeteksi pada kayu yang akan diproses.
6. *Firebase* akan menerima data dari jenis benda yang terdeteksi pada pengolahan kayu dan nama pengguna yang sedang menjalankan alat.

3.2.7 Implementasi sistem

Implementasi sistem adalah sebuah hasil dari sistem yang telah dirancang dan dibangun siap untuk di operasikan. Tujuan dari implementasi ini untuk melihat seberapa tepat alat yang dirancang apakah sistem dapat menyelesaikan sebuah permasalahan yang telah diidentifikasi sebelumnya. Implementasi sistem ini ditujukan untuk membantu sebuah industri pengolahan kayu dalam mengidentifikasi benda logam sebelum bahan kayu diolah. Tujuan tersebut dimaksudkan untuk menaikkan tingkat kualitas suatu industri pengolahan kayu dan mengurangi kerugian produksi apabila saat diolah ternyata masih terdapat bahan metal ataupun timah.

BAB IV

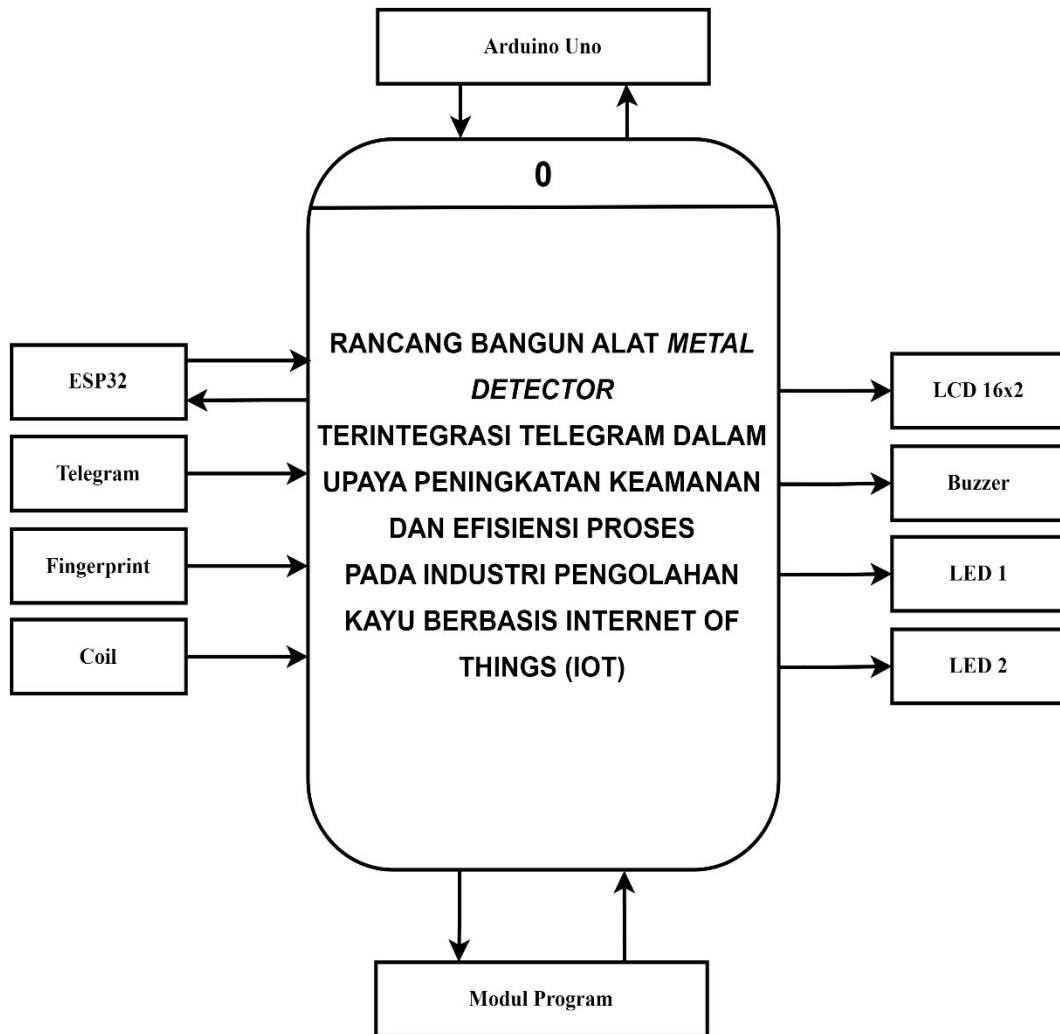
ANALISA DAN HASIL

4.1 Desain Sistem Secara Umum

Aturan dalam proses penganalisaan perlu dilakukan pendefinisian terlebih dahulu terhadap sistem yang akan dirancang secara menyeluruh yaitu gambaran secara jelas mengenai ruang lingkup pembahasan masalah dimana medianya terdiri dari *Context diagram*, *data flow diagram*, blok diagram, dan *flowchart*. Berikut untuk lebih jelasnya dapat dilihat pada uraian berikut :

4.1.1 Context Diagram

Context diagram merupakan pendefinisian rancangan yang bersifat menyeluruh. *Context diagram* berfungsi sebagai proses dan beberapa buah eksternal *entity*. *Context diagram* tersebut dapat dilihat pada Gambar 4.1 berikut :



Gambar 4.1 Context Diagram

Sistem ini berinteraksi dengan beberapa *entity* yaitu Arduino Uno, ESP32, Telegram, *Fingerprint*, *Coil*, LED 1, LED2, Buzzer dan Modul program. Berikut penjelasan dari *entity-entity* tersebut yaitu:

1. Arduino Uno

Arduino Uno berfungsi sebagai pusat pengolahan data dan instruksi.

2. ESP32

Berfungsi untuk menghubungkan alat ke jaringan sehingga alat dapat terkoneksi dengan telegram dan database.

3. Telegram

Digunakan untuk menambah sidik jari dan menghapus sidik jari dari *user* yang nantinya data mengenai sidik jari tersebut dikirim ke database.

4. *Fingerprint*

Berfungsi sebagai keamanan pada alat *metal detector* sehingga sidik jari yang mempunyai akses yang dapat menjalankan alat.

5. *Coil*

Berfungsi untuk membaca apabila terdapat benda seperti logam atau timah pada objek yang diuji.

6. LCD 16x2

Digunakan untuk menampilkan identitas peneliti, benda berjenis timah atau logam yang terdeteksi beserta bar kekuatan frekuensi jarak dari benda yang terdeteksi.

7. Buzzer

Buzzer berfungsi sebagai indikator apabila benda berjenis logam atau timah terdeteksi.

8. LED 1

Berfungsi sebagai indikator apabila adanya benda berjenis timah pada objek yang diuji.

9. LED 2

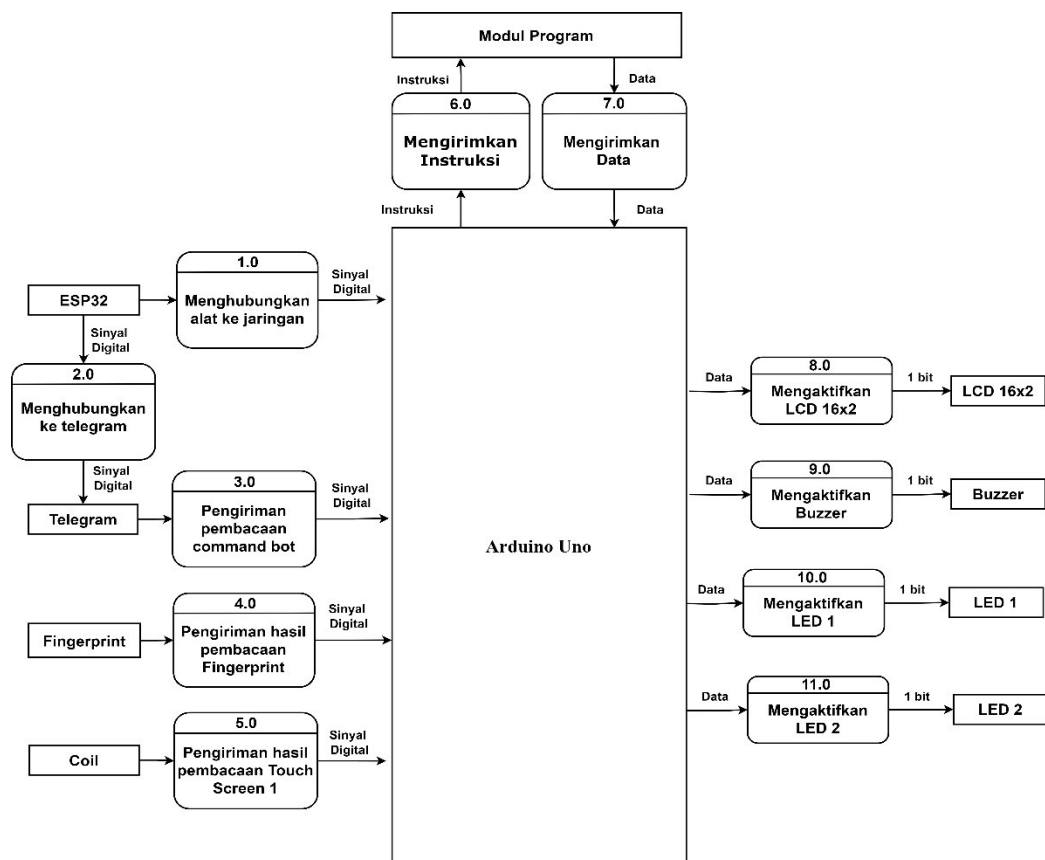
Berfungsi sebagai indikator apabila adanya benda berjenis besi pada objek yang diuji.

10. Modul Program

Modul Program digunakan untuk melakukan pembacaan terhadap pin-pin mikrokontroler, baik pembacaan terhadap sinyal-sinyal *input*, memberikan instruksi-instruksi untuk mengaktifkan pin-pin *output*. Modul program mengontrol semua proses yang terjadi di arduino menggunakan bahasa pemrograman arduino.

4.1.2 Data flow diagram (DFD)

Data flow diagram merupakan suatu cara atau metode yang digunakan untuk membuat rancangan sebuah sistem yang berorientasi pada alur data yang bergerak pada sebuah sistem. DFD menggambarkan arus data didalam sistem dengan terstruktur dan jelas. Berikut DFD dapat dilihat pada Gambar 4.2.



Gambar 4.2 Data Flow Diagram

Dari gambar di atas dapat dijelaskan bahwa sistem tempat penitipan sepatu memiliki beberapa proses instruksi sebagai berikut :

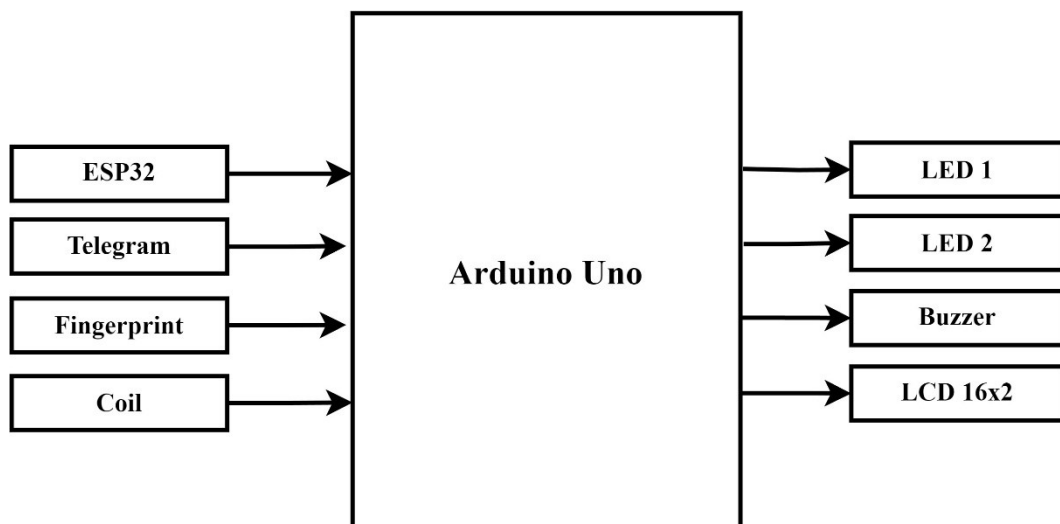
1. Sistem dimulai dengan ESP32 menghubungkan alat ke jaringan sehingga data mengenai pembacaan sensor dapat dikirim melalui komunikasi serial ke Arduino Uno. Pada konfigurasi ini, Pin GPIO 9 (RX1) pada ESP32 menerima data yang dikirim dari Arduino Uno melalui Pin D11 (TX). Data sensor yang diterima oleh ESP32 kemudian diproses (1.0).
2. Setelah jaringan terhubung ESP32 akan menghubungkan alat ke aplikasi telegram sehingga telegram dapat membaca data dari Arduino Uno dimana menggunakan komunikasi serial yang terhubung menggunakan komunikasi UART (RX dan TX) pada pin D10 dan D11 (2.0).
3. Telegram akan mengirimkan data command yang dipilih seperti penambahan sidik jari atau penghapusan sidik jari ke Arduino Uno (3.0).
4. Fingerprint mengirimkan data mengenai hasil pembacaan sidik jari melalui pin RX (GPIO 17) dan TX (GPIO 16) setelah *command* pada telegram dipilih (penambahan atau penghapusan sidik jari) ke Arduino Uno (4.0).
5. Coil akan mengirimkan data pembacaan benda yang berjenis timah atau logam melalui pin D5 yang terhubung dengan beberapa rangkaian elektronika seperti kapasitor 100nF, dioda dan resistor 220ohm ke Arduino Uno (5.0).
6. Arduino Uno mengirimkan instruksi ke modul program untuk diproses (6.0).
7. Hasil dari proses modul program dikirimkan kembali ke Arduino Uno, setelah itu akan dihasilkan data yang telah diolah (7.0).
8. Arduino Uno mengirimkan perintah aktif pada LCD 16x2 melalui pin A4 (SDA) dan pin A5 (SCL) untuk menampilkan identitas peneliti, hasil

pembacaan dari coil dan bar kekuatan frekuensi dari benda yang terdeteksi kemudian LCD 16x2 pun aktif (8.0).

9. Arduino Uno mengirimkan perintah aktif ke buzzer melalui pin D8 apabila benda berjenis timah atau besi terdeteksi kemudian buzzer pun aktif (9.0).
10. Arduino Uno mengirimkan perintah aktif ke LED 1 melalui pin D12 apabila terdeteksi benda berjenis timah kemudian LED 1 pun hidup (10.0).
11. Arduino Uno mengirimkan perintah aktif ke LED 2 melalui pin D13 apabila terdeteksi benda berjenis besi kemudian LED 2 pun hidup (11.0).

4.1.3 Blok Diagram

Blok Diagram digunakan untuk lebih menyederhanakan pemahaman dengan menjelaskan prinsip kerja alat secara keseluruhan yaitu berupa gabungan prinsip elektronika dan mekanika yang dikontrol oleh pemrograman. Lebih jelasnya dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 Blok Diagram

Pada perancangan sistem ini, dapat dilihat *entity* yang digunakan terdiri dari *input* berupa ESP32, Telegram, *Fingerprint* dan *Coil*. Pemroses yang digunakan

pada sistem berupa Arduino Uno. *Output* yang digunakan berupa LED 1, LED 2, Buzzer, dan LCD 16x2.

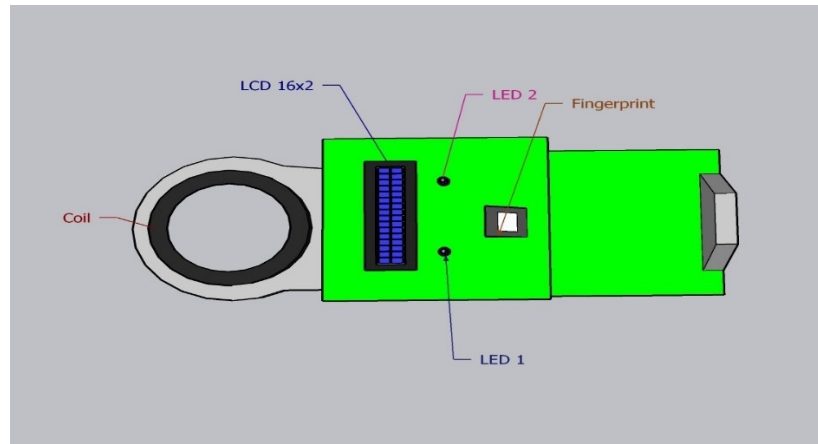
4.2 Prinsip Kerja Sistem

Secara umum bentuk dari alat ini terdiri dari rangkaian elektronika dan sistemnya akan bekerja secara terstruktur tergantung rancangan yang dibuat. Berikut adalah cara kerja sistem secara terstruktur. Pertama, ESP32 akan menghubungkan alat ke jaringan sehingga alat dapat berkomunikasi dengan telegram dan database. Tambahkan sidik jari pada *command* yang tersedia di telegram kemudian *scan* sidik jari pada sensor *fingerprint*.

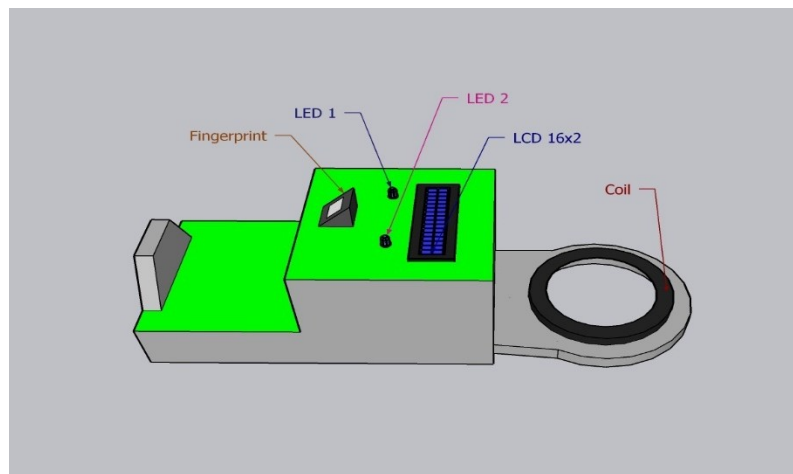
Coil akan mendeteksi apabila terdapat benda yang berjenis timah atau besi. LCD 16x2 akan menampilkan jenis benda yang terdeteksi beserta bar frekuensi kekuatan dari jarak antara benda tersebut dengan alat. Buzzer juga akan berbunyi menandakan adanya benda yang terdeteksi. LED 1 akan hidup apabila benda yang terdeteksi yaitu benda berjenis timah dan LED 2 akan hidup apabila benda yang terdeteksi berjenis besi.

4.3 Rancang Fisik Alat

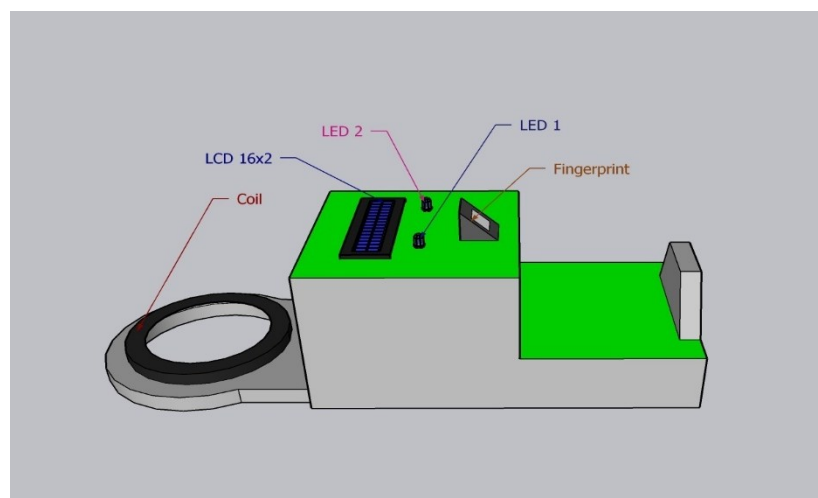
Perancangan alat ini merupakan tahap awal dari pemasangan dan menganalisa permasalahan yang dihadapi berdasarkan literatur yang menunjang perancangan alat. Rancangan fisik alat yang dibuat dapat dilihat sebagai berikut.



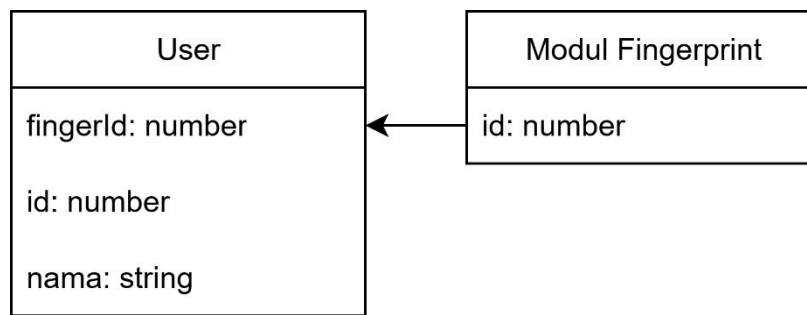
Gambar 4.4 Rancangan Tampilan Atas



Gambar 4.5 Rancangan Tampilan Kanan



Gambar 4.6 Rancangan Tampilan Kiri



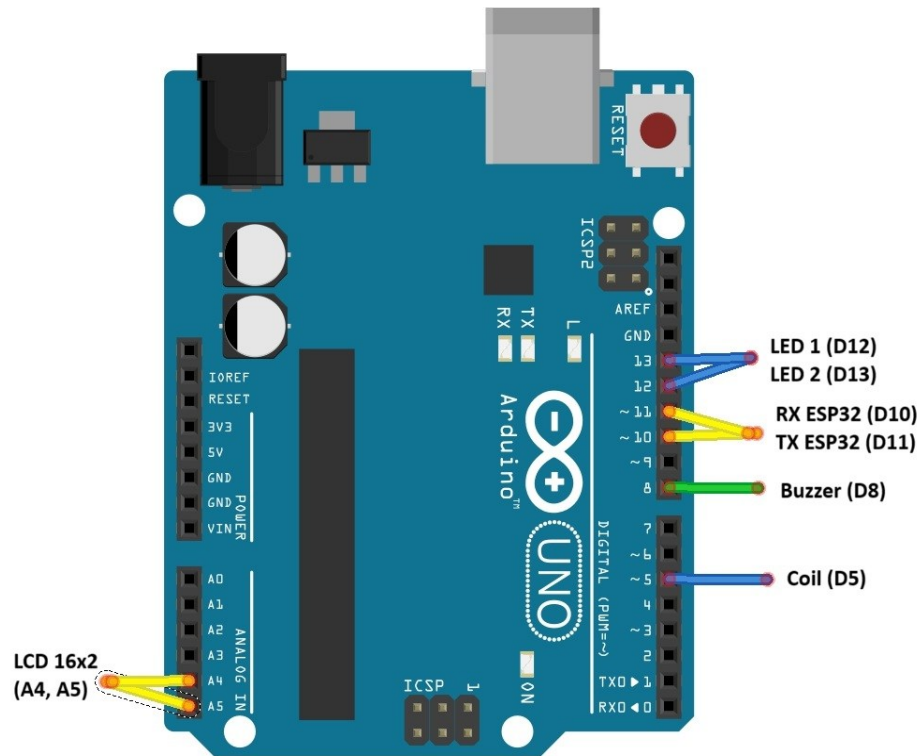
Gambar 4.7 Rancangan Database

4.4 Desain Sistem Terperinci

Desain dari sistem yang dibuat merupakan gambaran dari sistem secara keseluruhan. Dengan adanya desain ini maka prinsip kerja dari sistem serta komponen-komponen dari sistem yang digunakan dapat dilihat dengan jelas.

4.4.1 Desain Rangkaian Arduino Uno

Arduino Uno berfungsi untuk memproses dan mengontrol seluruh sistem agar sistem dapat bekerja seperti yang diinginkan dengan cara menghubungkan seluruh *entity* ke pin-pin Arduino Uno seperti yang dijelaskan pada Gambar 4.8.



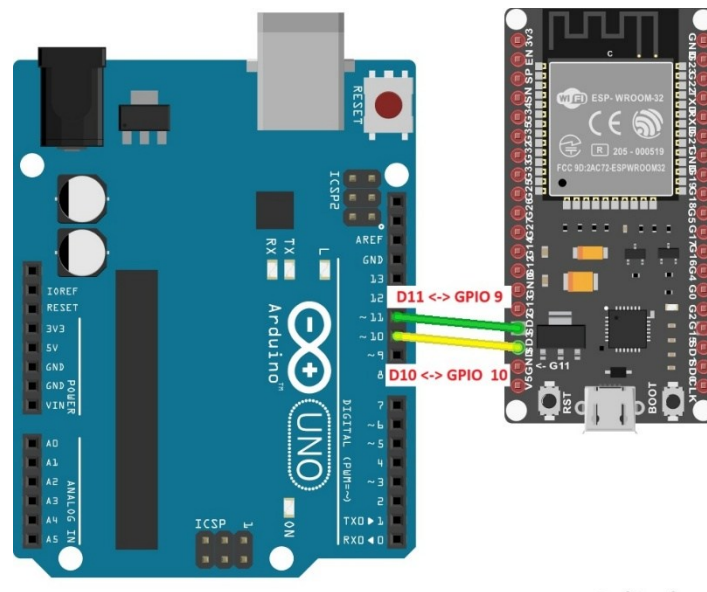
Gambar 4.8 Rangkaian Arduino Uno

Penjelasan pin-pin yang terhubung ke Arduino Uno sebagai berikut :

1. Pin D5 pada Arduino Uno terhubung ke pin positif coil.
2. Pin D8 pada Arduino Uno terhubung ke positif buzzer.
3. Pin D10 pada Arduino Uno terhubung ke pin TX pada ESP32.
4. Pin D11 pada Arduino Uno terhubung ke pin RX pada ESP32.
5. in D12 pada Arduino Uno terhubung ke pin anoda LED 1.
6. Pin D13 pada Arduino Uno terhubung ke pin anoda LED 2.
7. Pin A4 pada Arduino Uno terhubung ke pin SDA LCD 16x2.
8. Pin A5 pada Arduino Uno terhubung ke pin SCL LCD 16x2.
9. Pin 5V pada Arduino Uno terhubung ke pin VCC seluruh komponen.
10. Pin GND pada Arduino Uno terhubung ke pin GND seluruh komponen.

4.4.2 Desain Rangkaian ESP32

Pada perancangan alat ini menggunakan ESP32 yang digunakan untuk menghubungkan alat ke jaringan sehingga data hasil pembacaan sensor dapat dikirim ke telegram dan database. Rangkaianya dapat dilihat pada Gambar 4.9.



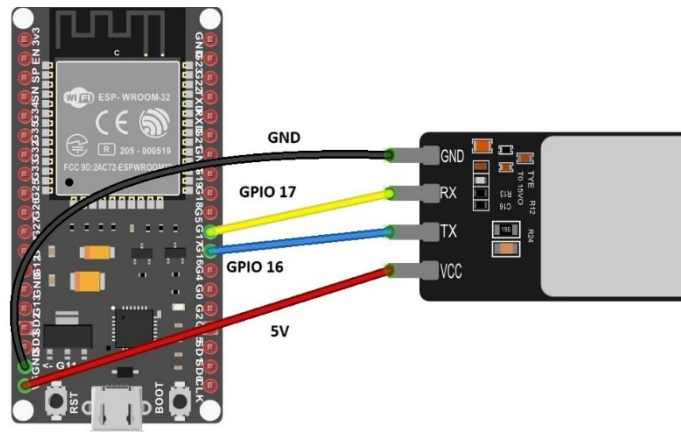
Gambar 4.9 Rangkaian ESP32

Penjelasan pin pada ESP32 yang terhubung ke Arduino Uno adalah sebagai berikut :

1. Pin GPIO 10 (TX 1) terhubung pada pin D10 Arduino Uno.
2. Pin GPIO 9 (RX 1) terhubung pada pin D11 Arduino Uno.

4.4.3 Desain Rangkaian *Fingerprint*

Pada perancangan alat ini menggunakan *fingerprint* yang digunakan sebagai sistem keamanan sehingga *user* yang memiliki akses yang dapat menjalankan alat. Rangkaianya dapat dilihat pada Gambar 4.10.



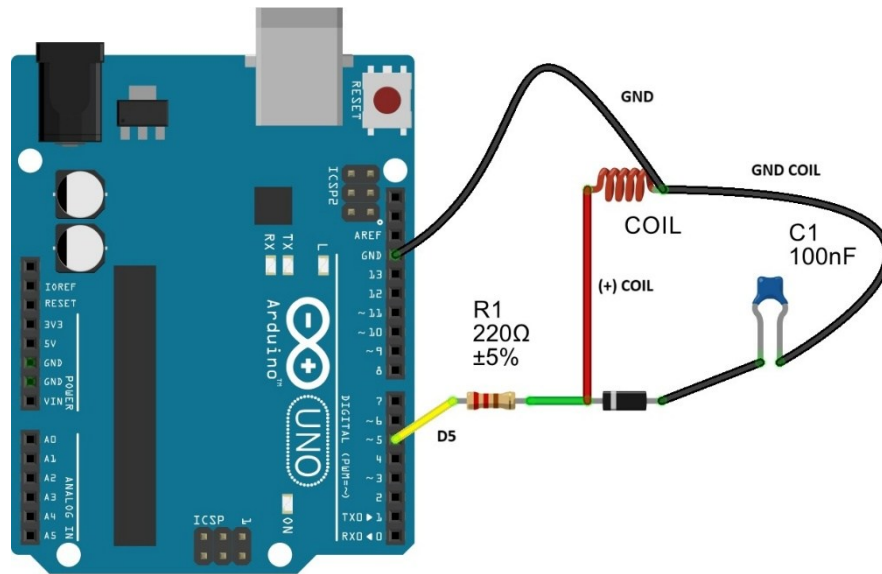
Gambar 4.10 Rangkaian Fingerprint

Penjelasan pin pada *Fingerprint* yang terhubung ke ESP32 adalah sebagai berikut :

1. Pin RX *Fingerprint* dihubungkan ke pin GPIO 17 ESP32.
2. Pin TX *Fingerprint* dihubungkan ke pin GPIO 16 ESP32.
3. Pin GND *Fingerprint* dihubungkan ke pin GND ESP32.
4. Pin VCC *Fingerprint* dihubungkan ke pin 5V ESP32.

4.4.4 Desain Rangkaian *Coil*

Pada perancangan alat menggunakan *coil* yang digunakan untuk mendeteksi benda berjenis timah dan besi sehingga pembacaan tersebut dapat dikirim ke database. Rangkaianannya dapat dilihat pada Gambar 4.11.



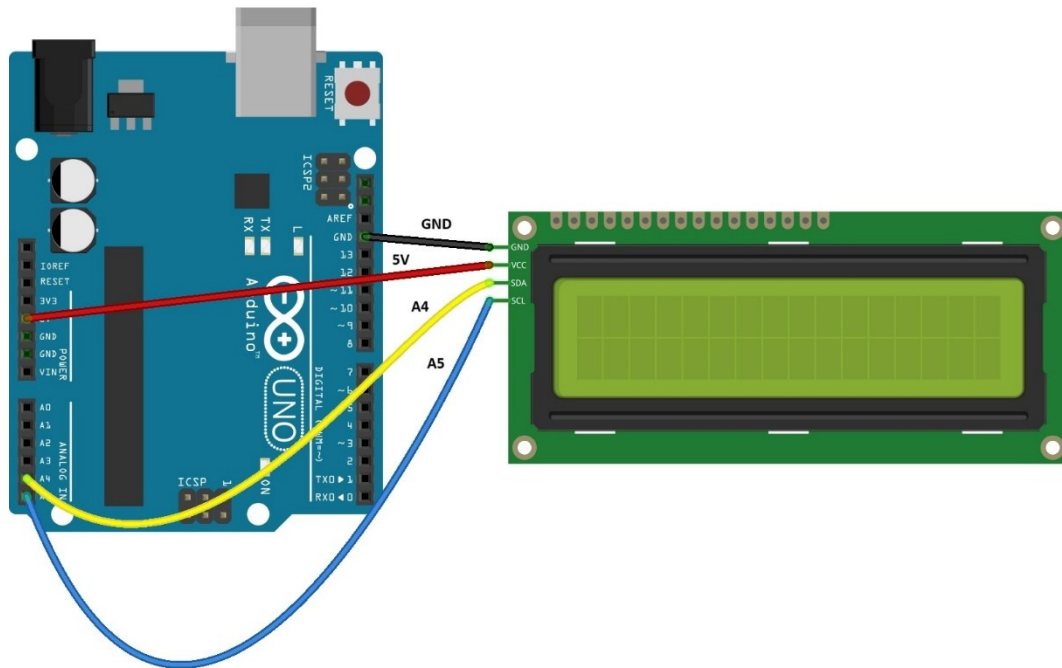
Gambar 4.11 Rangkaian Coil

Penjelasan pin pada *coil* yang terhubung ke Arduino Uno adalah sebagai berikut :

1. Kaki negatif *coil* terhubung pada kapasitor *polyester* 100nF.
2. Kaki kapasitor yang lain dihubungkan dengan kaki katoda dioda.
3. Kaki anoda dioda dihubungkan dengan kaki resistor 220 ohm.
4. Kaki resistor 220 ohm yang lain dihubungkan dengan pin D5 Arduino Uno.
5. Kaki negatif *coil* dihubungkan pada pin GND Arduino Uno.
6. Kaki positif *coil* dihubungkan ke kaki anoda dioda.

4.4.5 Desain Rangkaian LCD 16x2

Pada perancangan alat menggunakan LCD 16x2 yang digunakan untuk menampilkan jenis benda yang terdeteksi dan juga menampilkan bar frekuensi dari kekuatan jarak benda yang terdeteksi. Rangkaianya dapat dilihat pada Gambar 4.12.



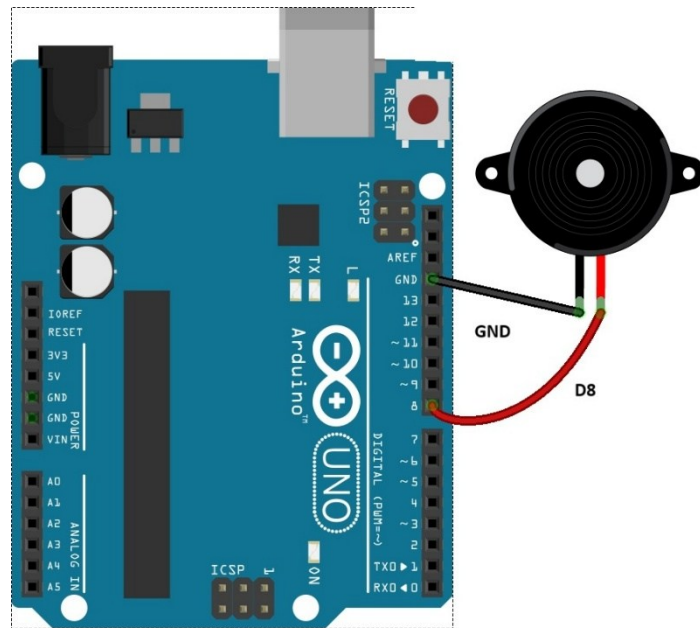
Gambar 4.12 Rangkaian LCD 16x2

Penjelasan pin pada LCD 16x2 yang terhubung ke Arduino Uno adalah sebagai berikut:

1. Pin SDA pada LCD 16x2 terhubung ke pin A4 Arduino Uno.
2. Pin SDL pada LCD 16x2 terhubung ke pin A5 Arduino Uno.
3. Pin VCC pada LCD 16x2 terhubung ke pin 5V Arduino Uno.
4. Pin GND pada LCD 16x2 terhubung ke pin GND Arduino Uno.

4.4.6 Desain Rangkaian Buzzer

Alat ini menggunakan buzzer yang digunakan untuk memberitahukan bahwa adanya benda yang terdeteksi. Rangkaiannya dapat dilihat pada Gambar 4.13.



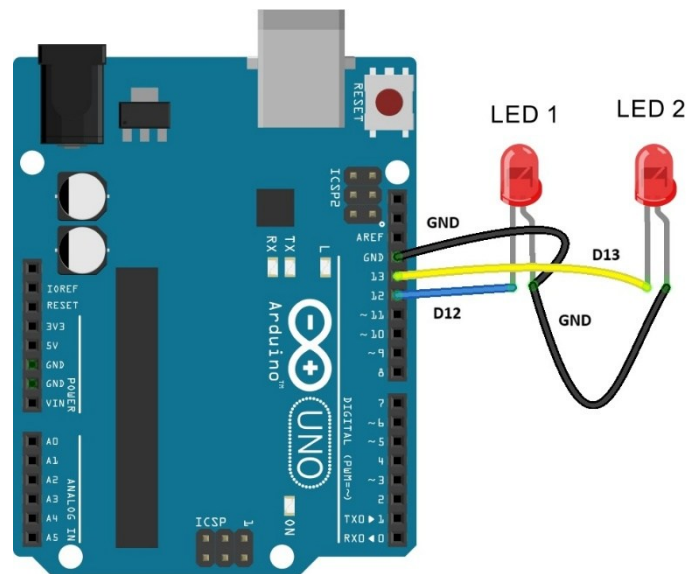
Gambar 4.13 Rangkaian Buzzer

Penjelasan pin pada Buzzer yang terhubung ke Arduino Uno adalah sebagai berikut:

1. Pin VCC pada buzzer terhubung ke pin D8 Arduino Uno.
2. Pin GND pada buzzer terhubung ke pin GND Arduino Uno.

4.4.7 Desain Rangkaian LED

Terdapat 2 buah LED pada alat ini. LED 1 digunakan sebagai indikator apabila timah terdeteksi dan LED 2 digunakan sebagai indikator apabila besi terdeteksi. Rangkaianannya dapat dilihat pada Gambar 4.14.



Gambar 4.14 Rangkaian LED

Penjelasan pada pin LED yang terhubung dengan Arduino Uno:

1. Pin positif pada LED 1 terhubung ke pin D12 Arduino Uno.
2. Pin positif pada LED 2 terhubung ke pin D13 Arduino Uno.
3. Pin GND pada LED dan LED 2 diparalelkan kemudian terhubung ke pin GND Arduino Uno.

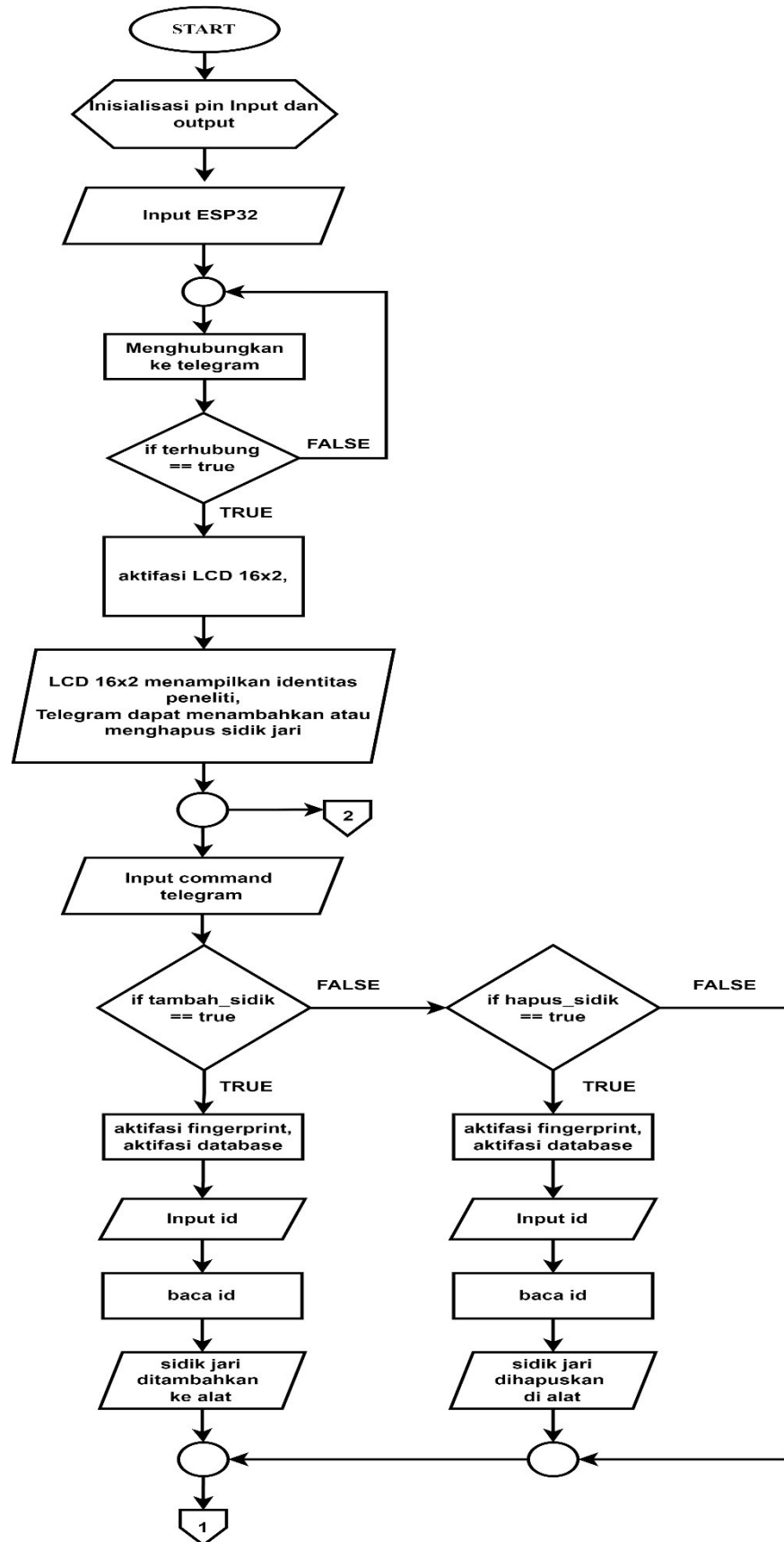
4.5 Rancangan Modul Program

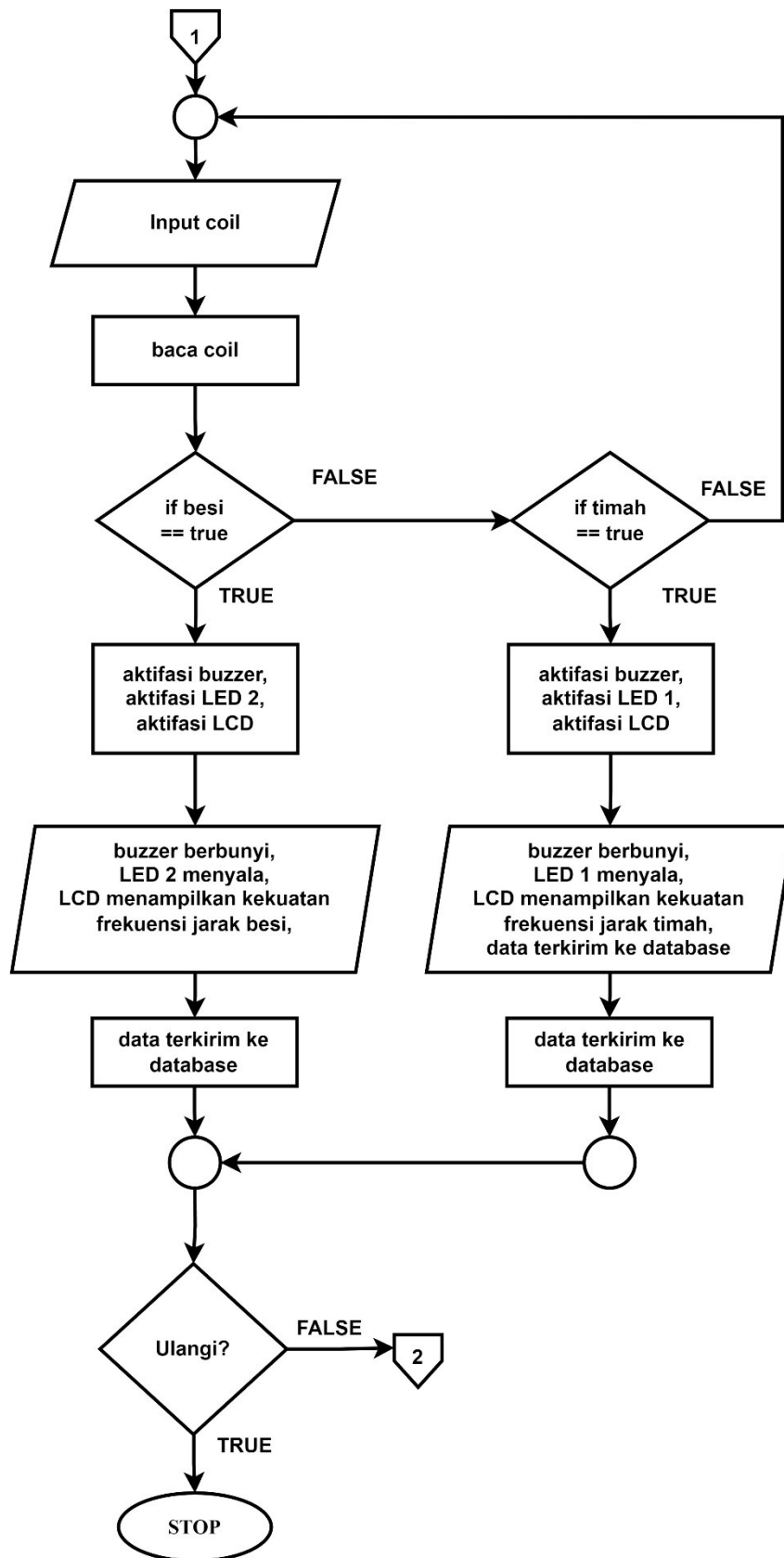
Sub bab ini menjelaskan tentang modul program yang digunakan untuk mengontrol kerja dari sistem yang dirancang. Untuk lebih mudah dimengerti rancangan modul dibagi menjadi dua bagian yaitu *Flowchart* dan *Listing Program*.

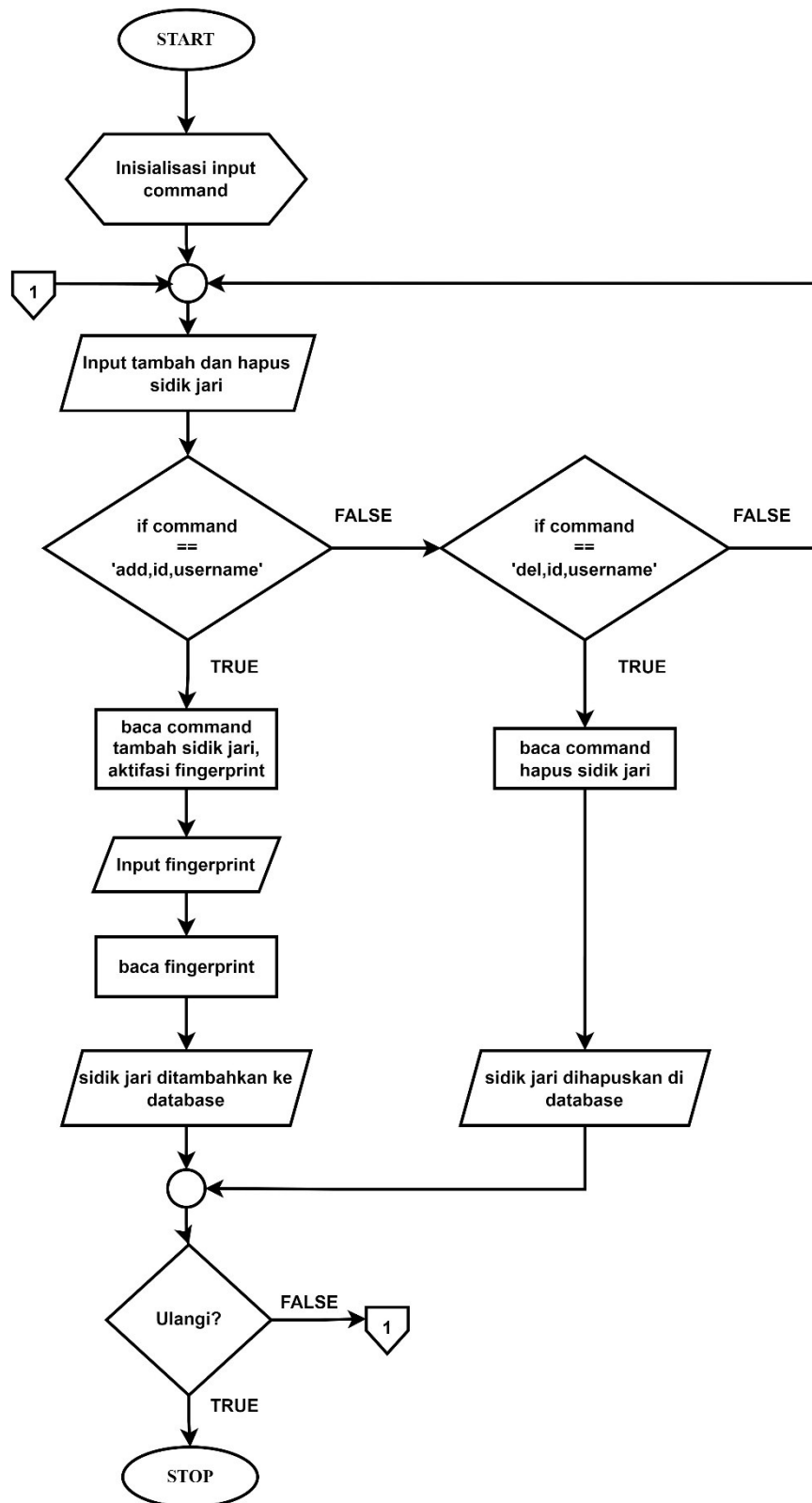
4.5.1 *Flowchart*

Sistem ini dikendalikan sepenuhnya oleh mikrokontroler dengan bantuan sebuah aplikasi yang dirancang dengan logika bahasa pemrograman Arduino IDE *compiler*. Untuk lebih jelasnya logika program dapat dilihat pada Gambar 4.15.

A. Flowchart Alat





B. Flowchart Telegram**Gambar 4.15 Flowchart**

4.5.2 Listing Program

a. Syntax Program Fingerprint

```
#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
// Gunakan UART2 di ESP32 (RX = GPIO17, TX = GPIO16)
HardwareSerial mySerial(2);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(115200);
  while (!Serial);
  // Inisialisasi Serial2 di GPIO 16 (TX2) dan GPIO 17 (RX2)
  mySerial.begin(57600, SERIAL_8N1, 17, 16);
  Serial.println("Cek sensor fingerprint...");
  if (finger.begin()) {
    Serial.println("Sensor fingerprint terdeteksi!");
  } else {
    Serial.println("Sensor fingerprint TIDAK ditemukan :(");
    while (1);
  }
  finger.getTemplateCount();
  Serial.print("Jumlah sidik jari yang tersimpan: ");
  Serial.println(finger.templateCount);
}

void loop()
{
  getFingerprintID();
  delay(2000); // Tunggu 2 detik sebelum pembacaan ulang
}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Gambar sidik jari berhasil diambil.");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("Tidak ada jari di sensor.");
      return p;
    case FINGERPRINT_PACKETRECEIVEERR:
      Serial.println("Kesalahan komunikasi.");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Gagal mengambil gambar sidik jari.");
```

```

        return p;
    default:
        Serial.println("Kesalahan tak dikenal.");
        return p;
    }

    p = finger.image2Tz();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Gambar dikonversi.");
            break;
        case FINGERPRINT_IMAGEMESS:
            Serial.println("Gambar terlalu berisik.");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Kesalahan komunikasi.");
            return p;
        case FINGERPRINT_FEATUREFAIL:
            Serial.println("Fitur tidak ditemukan.");
            return p;
        case FINGERPRINT_INVALIDIMAGE:
            Serial.println("Gambar tidak valid.");
            return p;
        default:
            Serial.println("Kesalahan tak dikenal.");
            return p;
    }
}

```

b. Syntax Program *Coil*

```

#include <FreqCount.h>

int buff=0;
int frq=0;
int resta=0;
int sens=2; // sensitivity
int autoBalance=0;

void setup() {
    FreqCount.begin(200);
    // Balance
    buff = frq;
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 10; j++) {
            digitalWrite(12, HIGH);
            delay(2);
            digitalWrite(12, LOW);

```

```

        delay(2);
    }
    delay(20);
    frq = FreqCount.read();
    if (frq != buff) {
        i = 0;
        buff = frq;
    }
}
for (int i = 0; i < 20; i++) {
    digitalWrite(12, HIGH);
    delay(1);
    digitalWrite(12, LOW);
    delay(1);
}
}

void loop() {
    frq = FreqCount.read();
    String msg="";
    resta = buff - frq;
    // Serial.println(frq); // max 4594

    if (resta > sens) {
        for (int i = 0; i < 10; i++) {
            digitalWrite(12, HIGH);
            delay(2);
            digitalWrite(12, LOW);
            delay(2);
        }
        delay(40 - (constrain(resta * 5, 10, 40)));

        lcd.setCursor(0,0);
        msg = "Besi "+String(resta)+"      ";
        msg = msg.substring(0,16);
        lcd.print(msg);

        bar = map(resta,3,111,1,16);
        // Serial.println(resta); // 112
        AB = false;
        tampil_nama=true;
        // High sound LED on
    }
    else if (resta < -sens) {
        resta = -resta;
        for (int i = 0; i < 20; i++) {
            digitalWrite(12, HIGH);
            delay(1);

```

```

        digitalWrite(12, LOW);
        delay(1);
    }
}

```

c. Syntax Program LED

```

void setup() {
    pinMode(12, OUTPUT); // LED indikator
    pinMode(13, OUTPUT); // LED Auto Balance
    Serial.begin(9600);
}

void loop() {
    // Simulasi pembacaan frekuensi (di proyek asli: FreqCount.read())
    frq = analogRead(A0); // Ganti ini sesuai kebutuhan
    resta = buff - frq;
    AB = true;

    if (resta > sens) { // Deteksi logam BESI
        for (int i = 0; i < 10; i++) {
            digitalWrite(12, HIGH);
            delay(2);
            digitalWrite(12, LOW);
            delay(2);
        }
        delay(40 - (constrain(resta * 5, 10, 40)));
        AB = false;
    }
    else if (resta < -sens) { // Deteksi logam TIMAH/EMAS
        resta = -resta;
        for (int i = 0; i < 20; i++) {
            digitalWrite(12, HIGH);
            delay(1);
            digitalWrite(12, LOW);
            delay(1);
        }
        delay(40 - (constrain(resta * 5, 10, 40)));
        AB = false;
    }
    else {
        // Tidak ada perubahan signifikan → LED dimatikan
        digitalWrite(12, LOW);
    }

    // Kontrol LED pin 13 untuk Auto Balance
    if (AB && resta != 0) {

```

```

    if (autoBalance > 1000) {
        autoBalance = 0;
        buff = frq; // Update buffer (frekuensi acuan)
    }
    autoBalance++;
    digitalWrite(13, HIGH); // Auto Balance aktif
} else {
    autoBalance = 0;
    digitalWrite(13, LOW); // Auto Balance tidak aktif
}

Serial.println(resta);
delay(100);
}

```

d. Syntax Program Buzzer

```

void setup() {
    pinMode(buzzerPin, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    // Simulasi pembacaan frekuensi (ganti dengan FreqCount.read() jika
    tersedia)
    frq = analogRead(A0);
    resta = buff - frq;
    AB = true;

    if (resta > sens) {
        for (int i = 0; i < 10; i++) {
            digitalWrite(buzzerPin, HIGH);
            delay(2);
            digitalWrite(buzzerPin, LOW);
            delay(2);
        }
        delay(40 - constrain(resta * 5, 10, 40));
        AB = false;
    }
    else if (resta < -sens) {
        resta = -resta;
        for (int i = 0; i < 20; i++) {
            digitalWrite(buzzerPin, HIGH);
            delay(1);
            digitalWrite(buzzerPin, LOW);
            delay(1);
        }
    }
}

```

```
    delay(40 - constrain(resta * 5, 10, 40));  
    AB = false;  
  }  
  else {  
    digitalWrite(buzzerPin, LOW); // Tidak bunyi jika tidak ada deteksi  
  }  
  
  // Auto balance (jika perlu tetap dipertahankan)  
  if (AB && resta != 0) {  
    if (autoBalance > 1000) {  
      autoBalance = 0;  
      buff = frq;  
    }  
    autoBalance++;  
  } else {  
    autoBalance = 0;  
  }
```

BAB V

PENGUJIAN SISTEM

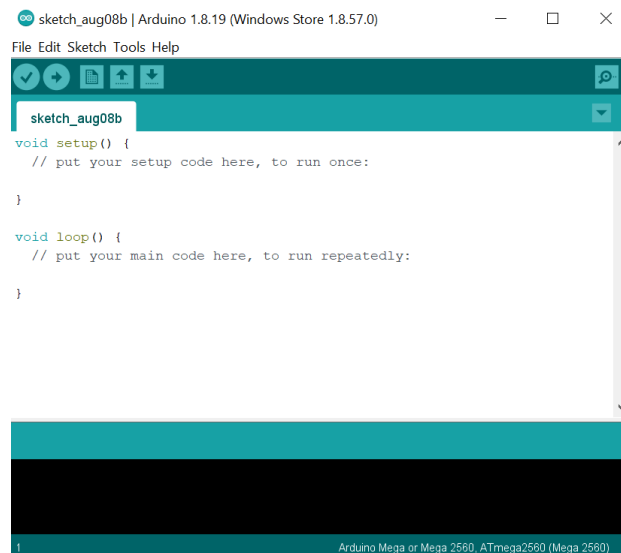
5.1 Pengujian Sistem Permodul

Pada tahap pengujian sistem ini merupakan salah satu tahap dalam daur hidup pengembangan sistem. Adapun langkah-langkah yang dapat dilakukan dalam menjalankan sistem alat per modul adalah sebagai berikut.

5.1.1 Pengujian Sistem Minimum

Pengujian sistem minimum dilakukan dengan menguji rangkaian sistem minimum setelah proses *uploading* dilakukan oleh pemrograman Arduino IDE. Terdapat langkah-langkah yang harus dilalui dalam pemrograman arduino adalah sebagai berikut :

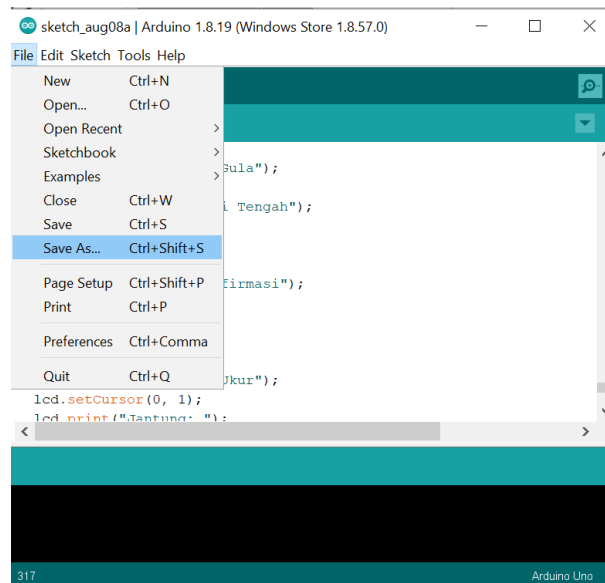
1. Jalankan *software* Ide Arduino dengan mengklik ganda *icon* aplikasi Arduino.exe, sehingga muncul tampilan, seperti Gambar 5.1.



Gambar 5.1 Tampilan Ide Arduino

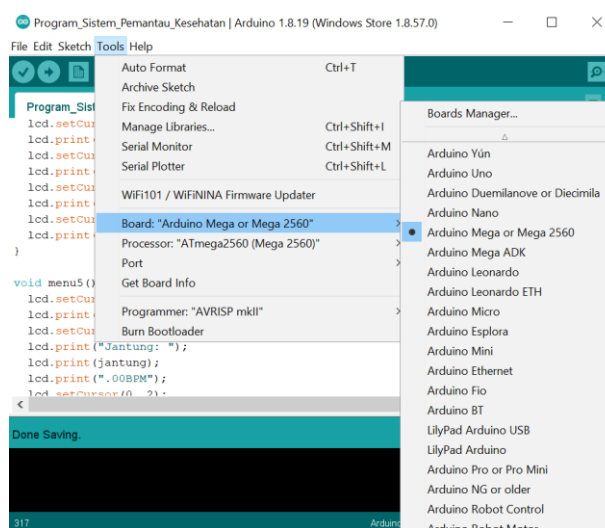
2. Lakukan pengetikan program, kemudian simpan file dengan memilih menu

File > Save as, seperti Gambar 5.2.



Gambar 5.2 Tampilan *Save As*

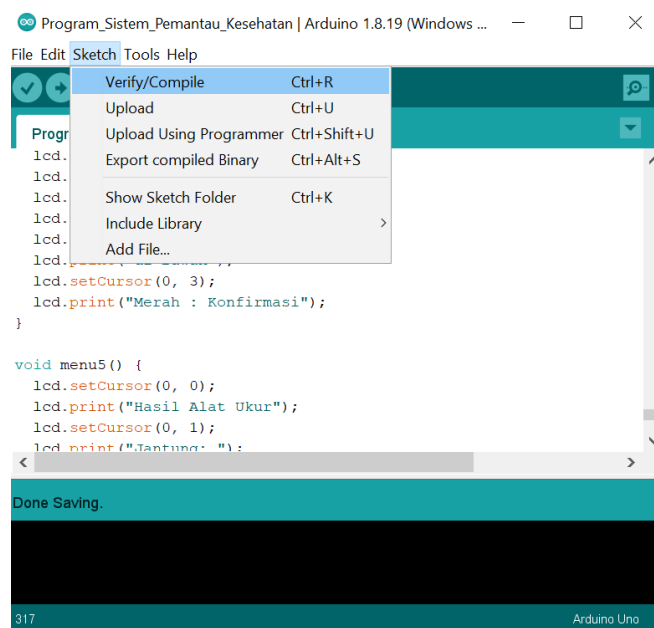
3. Langkah selanjutnya yaitu pilih *board* Arduino Mega yang digunakan sebagai mikrokontroler dari alat, seperti Gambar 5.3.



Gambar 5.3 Tampilan Pemilihan *Board* Arduino Mega

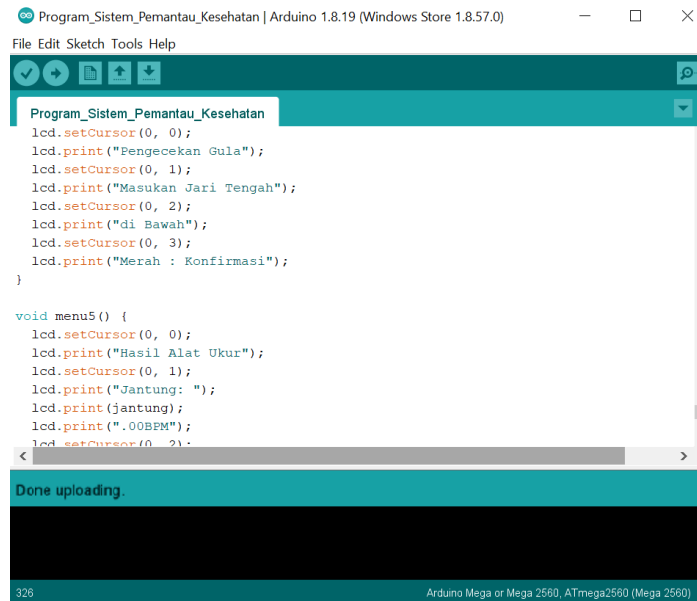
4. Setelah itu kompilasi program yang telah dibuat tadi dengan memilih menu

Sketch > Verify/Compile, seperti Gambar 5.4.



Gambar 5.4 Tampilan *Compiler* pada Arduino IDE

5. Jika program sukses di *upload* ke dalam arduino, maka akan muncul tulisan “*done uploading*”. Pasang kabel USB Arduino Mega *port type* A ke PC dan *port type* B ke Arduino Mega.
6. Pilih menu *Sketch > Upload* untuk mengupload program, *Upload* program yang telah dibuat ke Arduino Mega agar dapat mengontrol sistem supaya berfungsi dengan baik, seperti Gambar 5.5 berikut.



Gambar 5.5 Tampilan *Uploading* Program pada Arduino IDE

5.1.2 Pengujian *Fingerprint*

Pada alat *metal detector*, *fingerprint* digunakan sebagai sistem keamanan pada alat sehingga alat dapat dijalankan oleh pengguna yang sidik jarinya telah terdaftar kemudian status mengenai sidik jari pengguna akan ditampilkan pada layar LCD 16x2. Adapun hasil pengujian *fingerprint* dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian *Fingerprint*

No.	<i>Fingerprint</i>	LCD 16x2
1.	Jika <i>fingerprint</i> telah terdaftar	Layar LCD 16x2 menampilkan tulisan “Alat Ready untuk digunakan”
2.	Jika <i>fingerprint</i> belum terdaftar	Layar LCD 16x2 menampilkan tulisan “Sidik Jari Anda Belum Terdaftar”

5.1.3 Pengujian *Coil*

Pada alat *metal detector*, *coil* digunakan untuk mendeteksi benda berjenis besi atau timah. Adapun hasil pengujian *coil* dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pengujian *Coil*

No.	<i>Coil</i>	LCD 16x2	LED	Firestore
1.	<i>Coil</i> mendeteksi besi	LCD 16x2 menampilkan tulisan besi dan bar kekuatan frekuensi dari besi yang ditangkap.	LED 1 Mati LED 2 Hidup	Firestore menerima pembacaan besi
2.	<i>Coil</i> mendeteksi timah	LCD 16x2 menampilkan tulisan timah dan bar kekuatan frekuensi dari timah yang ditangkap.	LED 1 Hidup LED 2 Mati	Firestore menerima pembacaan timah

5.2 Pengujian Sistem Keseluruhan

Setelah melakukan pengujian untuk masing-masing komponen, seperti *fingerprint* dan *coil*, selanjutnya akan dilakukan pengujian untuk sistem secara keseluruhan. Langkah-langkah yang harus dilakukan untuk pengujian sistem secara keseluruhan yaitu sebagai berikut :

1. Hidupkan alat dengan cara menekan *switch* ke arah *on* kemudian alat akan aktif dan LCD 16x2 akan menampilkan identitas peneliti, seperti Gambar 5.7 berikut.

**Gambar 5.6 Tampilan Awal LCD 16x2**

2. Setelah itu LCD 16x2 akan menampilkan informasi untuk melakukan *scan* sidik jari, seperti Gambar 5.8 berikut.



Gambar 5.7 Tampilan LCD 16x2 Untuk Melakukan *Scanning* Sidik Jari

3. Setelah itu scan sidik jari pada sensor *fingerprint* yang terdapat pada bagian atas alat apabila sidik jari belum terdaftar LCD 16x2 akan menampilkan tulisan “Sidik Jari Anda Belum Terdaftar”, seperti Gambar 5.9 berikut.



Gambar 5.8 Tampilan LCD 16x2 Saat Sidik Jari Belum Terdaftar

4. Tambahakn sidik jari melalui aplikasi telegram dengan command “Add,(nomor ID),(Nama pengguna yang ingin ditambahkan), seperti Gambar 5.10 berikut.



Gambar 5.9 Penambahan Sidik Jari Melalui Aplikasi Telegram

5. Setelah itu tempelkan jari pada *fingerprint* kemdian LCD 16x2 akan menampilkan bar proses *scanning* sidik jari. Proses scanning sidik jari

dilakukan 2 kali sampai sidik jari terdaftar, seperti Gambar 5.11 berikut.



(a) Proses Pertama Saat Melakukan *Scanning* Sidik Jari



(b) LCD 16x2 Menampilkan Informasi Untuk Melakukan Proses *Scanning*

Ulang



(c) LCD Menampilkan Informasi Saat Proses Pendaftaran Sidik Jari

Berhasil

Gambar 5.10 Proses Pendaftaran Sidik Jari

6. Setelah itu scan kembali sidik jari yang telah didaftarkan kemudian LCD 16x2 akan menampilkan informasi alat telah siap digunakan, seperti Gambar 5.12 berikut.



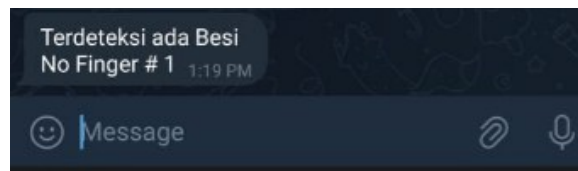
Gambar 5.11 Tampilan LCD 16x2 Apabila Sidik Jari Telah Terdaftar

7. LED 2 akan menyala apabila saat proses pemindaian pada kayu yang diuji ternyata memiliki benda berjenis besi, seperti yang ditunjukkan oleh Gambar 5.13.



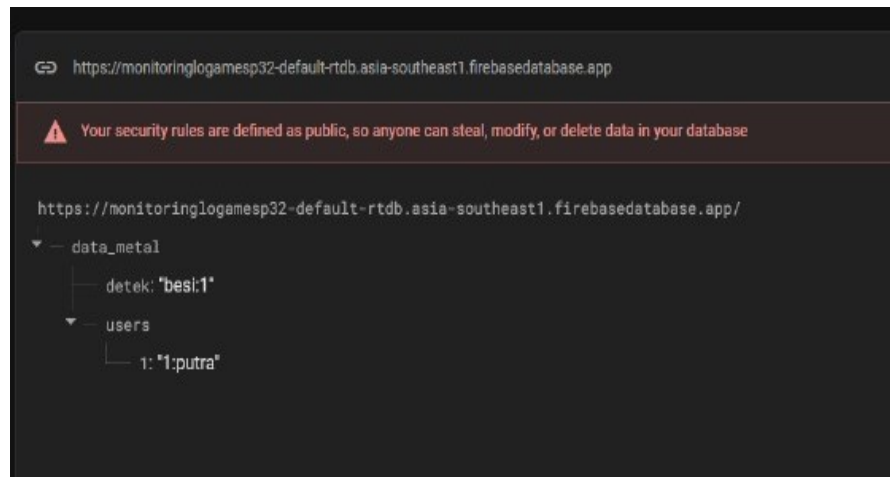
Gambar 5.12 LED 2 Menyala Saat Mendeteksi Benda Berjenis Besi

8. Setelah telegram dan database akan menerima notifikasi benda berjenis besi telah terdeteksi beserta sidik jari user yang sedang mengoperasikan alat, seperti Gambar 5.14 berikut.



(a) Tampilan Notifikasi Telegram Apabila Benda Berjenis Besi

Terdeteksi



(b) Tampilan *Firebase* Saat Benda Berjenis Besi Terdeteksi

Gambar 5.13 Tampilan Telegram dan *Firebase* Apabila Benda Berjenis Besi Terdeteksi

9. LED 1 akan menyala apabila saat proses pemindaian pada kayu yang diuji ternyata memiliki benda berjenis timah, seperti Gambar 5.15 berikut.



Gambar 5.14 LED 1 Menyala Saat Mendeteksi Benda Berjenis Timah

10. Setelah telegram dan database akan menerima notifikasi benda berjenis timah telah terdeteksi beserta sidik jari user yang sedang mengoperasikan alat, seperti

Gambar 5.16 berikut.



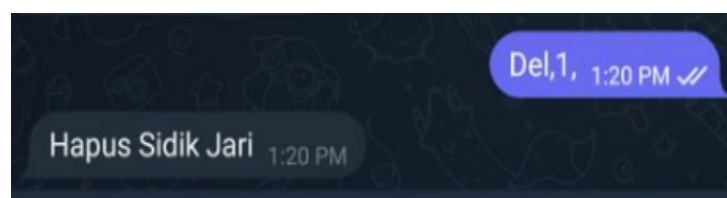
(a) Tampilan Notifikasi Telegram Apabila Benda Berjenis Timah Terdeteksi



(b) Tampilan *Firebase* Saat Benda Berjenis Timah Terdeteksi

Gambar 5.15 Tampilan Telegram dan *Firebase* Apabila Benda Berjenis Timah Terdeteksi

11. Sidik jari juga dapat dihapus menggunakan aplikasi telegram dengan *command* "Del, (nomor ID yang ingin dihapus)," seperti yang ditunjukkan oleh Gambar 5.17 berikut.



Gambar 5.16 Proses Penghapusan Sidik Jari di Aplikasi Telegram

BAB VI

PENUTUP

Berdasarkan hasil perancangan sistem dan pembuatan alat ini, maka dapat diambil kesimpulan dan saran yang nantinya bisa bermanfaat untuk pengembangan sistem berikutnya.

6.1 Kesimpulan

Berdasarkan pembahasan yang telah dijelaskan pada bab-bab sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Perancangan sistem *metal detector* pada industri pengolahan kayu menggunakan Arduino Uno sebagai kontrol utama pada sistem dapat berjalan dengan baik.
2. Penggunaan *fingerprint* dapat membaca sidik jari dari pengguna dengan akurat dan menjadi sistem keamanan pada alat dengan baik. Hal ini membuat alat hanya dapat dijalankan oleh pengguna yang sidik jarinya telah terdaftar dan tersimpan di database.
3. Pemanfaatan *coil* dapat mendeteksi adanya benda yang terdapat pada kayu yang akan diolah. *Coil* juga dapat membedakan benda yang terdeteksi didalam kayu seperti jenis besi dan timah.
4. Telegram dapat memberitahukan notifikasi ke pada pengguna bahwa adanya benda berjenis besi ataupun timah yang terdetekdsinya. Pemanfaatan telegram juga dapat menambahkan sidik jari pengguna baru dan menghapus sidik jari pengguna.

5. Pemanfaatan LED dan LCD 16x2 dapat memudahkan pengguna dalam membedakan jenis benda yang terdeteksi. LED 1 hidup jika benda berjenis timah terdeteksi dan benda LED 2 hidup jika benda berjenis besi terdeteksi. LCD 16x2 menampilkan jenis benda yang terdeteksi beserta bar frekuensi dari benda yang terdeteksi.

6.2 Saran

Berdasarkan pengalaman yang didapatkan dari perancangan dan pembuatan alat, ada beberapa saran yang ingin disampaikan agar berguna untuk pengembangan alat selanjutnya adalah sebagai berikut:

1. Alat ini hanya menggunakan RFID sebagai sistem keamanan, diharapkan pada pengembangan selanjutnya untuk menambahkan metode verifikasi ganda (*two-factor authentication*), seperti kombinasi sidik jari dan PIN atau kartu RFID, terutama untuk akses fitur penghapusan dan penambahan sidik jari pengguna.
2. Alat ini dirancang menggunakan bahan berjenis akrilik, diharapkan pada pengembangan selanjutnya sistem sebaiknya dibuat dalam *casing* tahan debu dan benturan (*dustproof dan shockproof*), mengingat lingkungan industri pengolahan kayu yang cukup keras dan penuh partikel.
3. Pada pengembangan berikutnya, diharapkan untuk menggunakan LCD 20x4 atau layar OLED/TFT kecil agar bisa menampilkan lebih banyak informasi sekaligus, seperti ID pengguna aktif, status sensor, dan riwayat deteksi singkat.
4. Disarankan untuk menambahkan indikator level baterai lithium pada system sehingga pengguna dapat memantau kondisi daya secara *real-time* dan melakukan pengisian ulang sebelum baterai benar-benar habis. Hal ini

bertujuan mencegah kegagalan fungsi alat di saat kritis, terutama saat proses deteksi sedang berlangsung di industri pengolahan kayu.

5. Disarankan untuk mengembangkan desain sensor dengan penambahan lilitan coil pada rangkaian detektor guna memperluas jarak deteksi melebihi 5 cm. Peningkatan jarak ini akan memperluas cakupan aplikasi alat, sehingga tidak hanya dapat diimplementasikan pada sektor pengolahan kayu, tetapi juga pada bidang lain seperti industri daur ulang, pengolahan makanan, dan keamanan, yang membutuhkan deteksi logam pada jarak yang lebih jauh dan area yang lebih luas.

Berdasarkan kesimpulan dan saran di atas, diharapkan bisa dipahami dengan sebaik mungkin supaya alat ini dapat dikembangkan menjadi lebih baik lagi dan bisa menjadi sumber referensi untuk penelitian berikutnya.

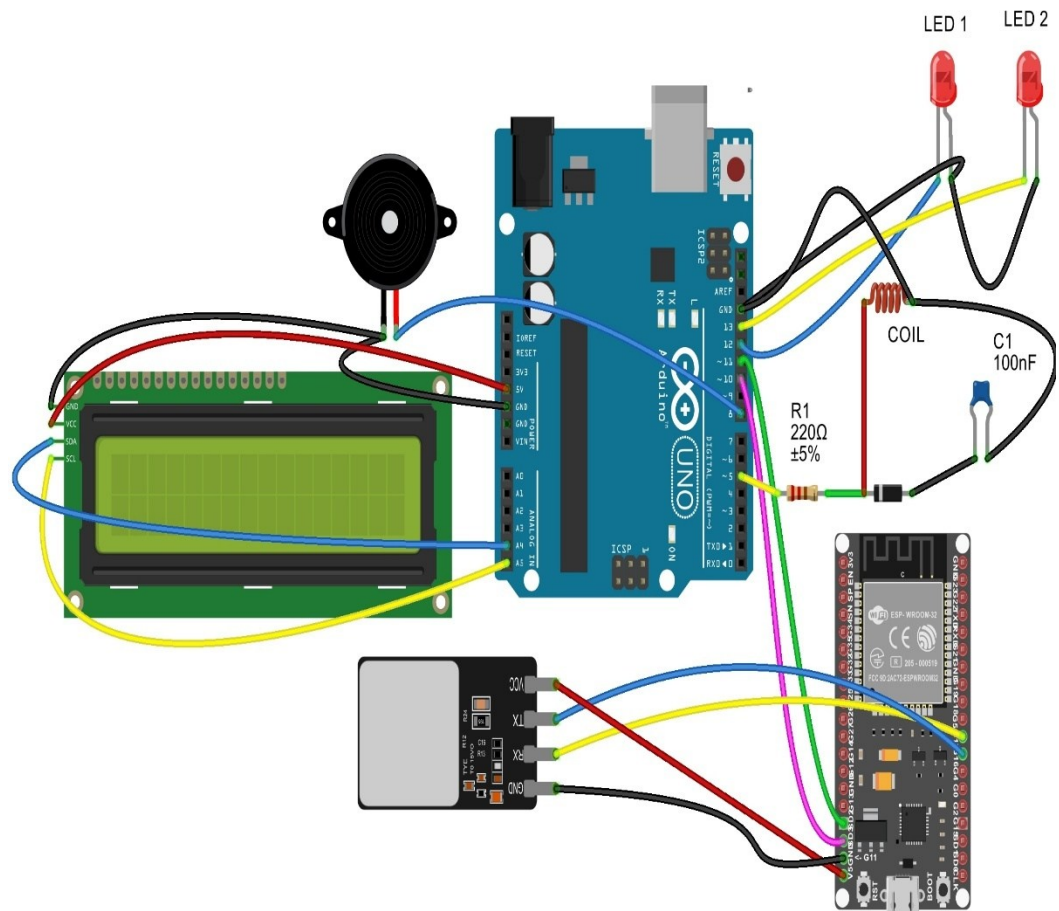
DAFTAR PUSTKA

- Alfan, A. N., & Ramadhan, V. (2022). Prototype Detektor Gas Dan Monitoring Suhu Berbasis Arduino Uno. *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput*, 9(2), 61-69.
- Azzis, N. F., Anggoro, D. A., & Kom, S. (2023). *Sistem Informasi Penjualan Makanan Berbasis Web Pada Rumah Makan Jagosore* (Doctoral dissertation, Universitas Muhammadiyah Surakarta).
- Cahyono, B. D., Nugraha, M. I., Tirtayasa, U. S. A., NoKm, J. R. P., Jaya, K. C., & Serang, K. (2023). Pengembangan Media Pembelajaran Elektronika Dasar Untuk Memahami Nilai Resistor Berdasarkan Kode Warna 3 Gelang Dan 4 Gelang Bagi Siswa SMK Kelas X Jurusan Teknik Otomasi Industri. *J. Educ*, 5(04), 11547-11557.
- Gushardi, H., & Faiza, D. (2022). Perancangan dan Pembuatan Alat Penghitung Jumlah Uang Otomatis Terintegrasi Internet of Things. *vol*, 6, 2996-3005.
- Gusti, W. R., Zakarijah, M., & Rochayati, U. (2022). Perancangan embedded system untuk kendali rumah burung walet berbasis atmega8. *JEPIN (Jurnal Edukasi dan Penelit. Inform., vol. 8, no. 3, pp. 500–507.*
- Listiaji, P., Dewi, N. R., Amalia, A. V., Amelia, R. N., Widianingrum, R. A., Ardhani, N. E., & Rahman, N. F. A. (2024). Development of Digital Calorimeter Experiment Kit to Train the Designing Experiments Ability of Pre-Service Science Teacher. *JIPF (Jurnal Ilmu Pendidikan Fisika)*, 9(1), 125-134.
- Putra, K. O., Yubarda, E., & Jannah, M. (2022). Aplikasi Penjualan Alat Tulis Kantor Pada Alif Photocopy. *JSR: Jaringan Sistem Informasi Robotik*, 6(1), 114-121.
- Savitri, C. E., & Paramytha, N. (2022). Sistem monitoring parkir mobil berbasis mikrokontroller ESP32. *Jurnal Ampere*, 7(2), 135-144.
- Sepudin, D. M., & Abdullah, S. (2022). Sistem keamanan pintu rumah berbasis internet of things berbasis nodemcu esp32 dan telegram. *Jurnal RESTIKOM: Riset Teknik Informatika Dan Komputer*, 4(3), 93-99.

- Sudrajat, R., & Rofifah, F. (2023). Rancang Bangun Sistem Kendali Kipas Angin dengan Sensor Suhu dan Sensor Ultrasonik Berbasis Arduino Uno. *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, 7(1), 555-564.
- Sumbaryadi, A., & Gustina, D. (2021). Pensaklaran Otomatis Cooler Pad Menggunakan Sensor Suhu LM 35 Dengan Mikrokontroler ATMEGA8535. *Jurnal Sistem Informasi*, 8(2014), 57-82.

LAMPIRAN

A. Rangkaian Keseluruhan



B. *Listing Program*

a. *Arduino Uno*

```
#include <FreqCount.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

SoftwareSerial fpSerial(3, 2);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fpSerial);

int buff=0;
int frq=0;
int resta=0;
int sens=2; // sensitivity
int autoBalance=0;
bool AB=true;
bool tampil_nama=true;
int nView=0;
int detek = 0;
int barLoading=0;

int bar=0;
#define led1 8
#define led2 9
#define buzzer 12

String status="";

uint8_t id;

bool alat_ready = false;

void setup()
{

    Serial.begin(9600);
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode (buzzer, OUTPUT);
    digitalWrite(led1,LOW);
    digitalWrite(led2,LOW);

    // set the data rate for the sensor serial port
```

```

finger.begin(57600);

if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
} else {
  Serial.println("Did not find fingerprint sensor :(");
  while (1) { delay(1); }
}

lcd.begin();
lcd.backlight();
lcd_center("AUZI AULIA S","21101152620054");
FreqCount.begin(200);
delay(100);

frq = FreqCount.read();
// Balance
buff = frq;
for (int i = 0; i < 5; i++) {
  for (int j = 0; j < 10; j++) {
    digitalWrite(12, HIGH);
    delay(2);
    digitalWrite(12, LOW);
    delay(1);
  }
  delay(10);
  frq = FreqCount.read();
  if (frq != buff) {
    i = 0;
    buff = frq;
  }
}

}

void loop()          // run over and over again
{

  if (Serial.available()){
    String msg="";
    while(Serial.available()) {
      msg+=char(Serial.read());
      delay(10);
    }
    msg.trim();
    String myArr[4];
    int index=0;
    for(int i=0;i< msg.length();i++){

```

```

        if (msg[i]!=';') myArr[index]+=msg[i]; else index++;
    }
    if (myArr[0]=="Add"){
        id = atoi(myArr[1].c_str());
        while (! getFingerprintEnroll() );
        delay(1000);
    }

    if (myArr[0]=="Del"){
        id = atoi(myArr[1].c_str());
        deleteFingerprint(id);
        alat_ready=false;
        delay(1000);
    }
}

```

```

if (!alat_ready) {
    while(!getFingerprintID());
    if (nView++==1) {
        lcd_center("Silakan Scan","Jari Anda");
    }
    return;
}

```

```

nView=0;

```

```

// Ready aplikasi

```

```

AB = true;
bar = 0;
frq = FreqCount.read();
String msg="";
resta = buff - frq;

if (resta > sens) {
    for (int i = 0; i < 10; i++) {
        digitalWrite(12, HIGH);
        delay(2);
        digitalWrite(12, LOW);
        delay(2);
    }
    delay(40 - (constrain(resta * 5, 10, 40)));

    lcd.setCursor(0,0);
    digitalWrite(led1,LOW);
    digitalWrite(led2,HIGH);
    msg = "Besi "+String(resta)+" ";
}

```

```

msg = msg.substring(0,16);
status="besi";
lcd.print(msg);

bar = map(resta,3,32,1,16);
if (bar>16) bar=16;
AB = false;
tampil_nama=true;
// High sound LED on
}
else if (resta < -sens) {
  resta = -resta;
  for (int i = 0; i < 20; i++) {
    digitalWrite(12, HIGH);
    delay(1);
    digitalWrite(12, LOW);
    delay(1);
  }
  delay(40 - (constrain(resta * 5, 10, 40)));
  lcd.setCursor(0,0);
  digitalWrite(led1,HIGH);
  digitalWrite(led2,LOW);
  status="timah";
  msg = "Timah/Emas "+String(resta)+" ";
  msg = msg.substring(0,16);
  lcd.print(msg);
  bar = map(resta,3,32,1,16);
  if (bar>16) bar=16;
  AB = false;
  tampil_nama=true;
}
else {

  digitalWrite(led1,LOW);
  digitalWrite(led2,LOW);

  // Turn off both LEDs if the sound is in the range

}

if (AB) {
  if (tampil_nama) {
    lcd_center("Alat Ready","untuk digunakan");
    Serial.println(status+", "+id);
    detek = 0;

  }
  tampil_nama=false;
}

```

```

    } else
    {
        lcd.setCursor(0,1);
        lcd.print(chr_bar(bar));
    }

    resta = 0;

    if (true) { // auto Balance
        if (AB && resta != 0) {
            if (autoBalance > 1000) {
                autoBalance = 0;
                buff = frq;
            }
            autoBalance++;
        } else {
            autoBalance = 0;
        }
        delay(1);
    }

}

```

b. ESP32

```

#ifdef ESP32
#include <WiFi.h>
#else
    #include <ESP8266WiFi.h>
#endif
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h> // Universal Telegram Bot Library
written by Brian Lough: https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot
#include <ArduinoJson.h>
#include <Firebase_ESP_Client.h>
#include <Wire.h>

// Provide the token generation process info.
#include "addons/TokenHelper.h"
// Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

// Replace with your network credentials
const char* ssid = "Zyy";
const char* password = "putra1234";

```

```
// Initialize Telegram BOT
#define BOTtoken
"7562027140:AAFsTZ3Vto9V4geXaNW0oF0cbTHFBOLDPww" // your
Bot Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
#define CHAT_ID "5906870995"

#ifndef ESP8266
  X509List cert(TELEGRAM_CERTIFICATE_ROOT);
#endif

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

// Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

const int ledPin = 2;
bool ledState = LOW;

// firebase
#define API_KEY "AIzaSyCpuPVws7D1n_anWGmecyVdgaG8zwVWiNc"
#define DATABASE_URL "https://monitoringlogamesp32-default-
rtddb.asia-southeast1.firebaseio.com/"

// Insert Authorized Email and Corresponding Password
#define USER_EMAIL "auziauliasputra@gmail.com"
#define USER_PASSWORD "Putra123"

FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
String stt_detek = "";
String no_finger = "";

// Handle what happens when you receive new messages
void handleNewMessages(int numNewMessages)
{
  String pesan="";
  for (int i = 0; i < numNewMessages; i++)
  {
    // Handle each message
  }
}
```

```

telegramMessage &msg = bot.messages[i];
if (msg.text=="/start"){
    pesan = "Silakan ketik perintah\nuntuk finger print sbb:\n";
    pesan += "1. Tambah ID finger => Add,IdFinger,Nama\n : contoh
add,1,Budi \n";
    pesan += "2. Hapus ID finger => Del,IdFinger,Nama\n : contoh
del,1,Budi \n";
    bot.sendMessage(CHAT_ID, pesan);
}

String myArr[4];
String txt = msg.text;
int index=0;
for(int i=0;i< txt.length();i++){
    if (txt[i]!=';') myArr[index]+=txt[i]; else index++;
}

Serial2.println(txt);

if (myArr[0]=="Add") {
    Serial.println("Tambah sidik jari");
    bot.sendMessage(CHAT_ID, "Tambah Sidik Jari\n");
    pesan = myArr[1]+":"+myArr[2];
    String folder="data_metal/users/"+myArr[1];
    Firebase.RTDB.setString(&fbdo, folder, pesan);
}

if (myArr[0]=="Del") {
    Serial.println("Hapus sidik jari");
    bot.sendMessage(CHAT_ID, "Hapus Sidik Jari\n");
}
}

}

void bot_setup()
{
    const String commands = F("[
        \"{\"command\": \"help\", \"description\": \"\"}, \"
        \"{\"command\": \"start\", \"description\": \"\"}, \"
        \"{\"command\": \"status\", \"description\": \"Answer device
current status\"}\"
    ]");
    bot.setMyCommands(commands);
}

void setup() {
    Serial.begin(115200);

```

```

Serial2.begin(9600,SERIAL_8N1,16,17);

#ifdef ESP8266
    configTime(0, 0, "pool.ntp.org");    // get UTC time via NTP
    client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
#endif

pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, ledState);

// Connect to Wi-Fi
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
#ifdef ESP32
    client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root
certificate for api.telegram.org
#endif
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
}
// Print ESP32 Local IP Address
Serial.println(WiFi.localIP());
bot_setup();
bot.sendMessage(CHAT_ID, "Bot Started", "");

config.api_key = API_KEY;
// Assign the user sign in credentials
auth.user.email = USER_EMAIL;
auth.user.password = USER_PASSWORD;

/* Assign the RTDB URL (required) */
config.database_url = DATABASE_URL;

/* Sign up */
if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("ok");
}
else {
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
}

/* Assign the callback function for the long running token generation task
*/
config.token_status_callback = tokenStatusCallback; //see
addons/TokenHelper.h

Firebase.begin(&config, &auth);

```

```

    Firebase.reconnectWiFi(true);

}

void loop() {

    if (Serial2.available()){
        String msg = "";
        while (Serial2.available()){
            msg+=char(Serial2.read());
            delay(50);
        }
        msg.trim();
        String myArr[4];
        int index=0;
        for(int i=0;i< msg.length();i++){
            if (msg[i]!=';') myArr[index]+=msg[i]; else index++;
        }

        String pesan="";

        if (myArr[0]=="besi"){
            stt_detek = "besi";
            no_finger = myArr[1];
            pesan="Terdeteksi ada Besi\nNo Finger # "+myArr[1];
            bot.sendMessage(CHAT_ID, pesan);

        }
        if (myArr[0]=="timah"){
            stt_detek = "timah/emas";
            no_finger = myArr[1];
            pesan="Terdeteksi ada Timah/Emas\nNo Finger # "+myArr[1];
            bot.sendMessage(CHAT_ID, pesan);
        }

    }

}

```