

**RANCANG BANGUN ALAT PENGHANCUR KERTAS
OTOMATIS MENGGUNAKAN MIKROKONTROLER
ARDUINO DENGAN NOTIFIKASI TELEGRAM**

SKRIPSI

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



Oleh

DIMAS PRASETYO WISNU HARYANTO

21101152620061

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

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : DIMAS PRASETYO WISNU HARYANTO

NoBp : 21101152620061

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, Agustus 2025

DIMAS PRASETYO WISNU HARYANTO
21101152620061

LEMBAR PENGESAHAN SKRIPSI

**RANCANG BANGUN ALAT PENGHANCUR KERTAS OTOMATIS
MENGUNAKAN MIKROKONTROLER ARDUINO DENGAN
NOTIFIKASI TELEGRAM**

Yang dipersiapkan dan disusun oleh:

DIMAS PRASETYO WISNU HARYANTO
21101152620061

Telah Memenuhi Persyaratan Untuk Dipertahankan Di Depan Dewan Penguji
Pada Ujian Komprehensif

Padang, Agustus 2025

Pembimbing I

Pembimbing II

(Dr. Guslendra, S.Kom, M.Kom)

NIDN : 1016087201

(Nanda Tommy Wirawan, S.Kom, M.Kom)

NIDN : 1025059301

HALAMAN PENGESAHAN PENGUJI SIDANG SKRIPSI

**RANCANG BANGUN ALAT PENGHANCUR KERTAS OTOMATIS
MENGUNAKAN MIKROKONTROLER ARDUINO DENGAN
NOTIFIKASI TELEGRAM**

OLEH:

DIMAS PRASETYO WISNU HARYANTO
21101152620061

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 : 19 Agustus 2025

TIM PENGUJI

1. **Riska Robianto, S.Kom, M.Kom**
NIDN : 1004077904
2. **Nanda Tommy Wirawan, S.Kom, M.Kom**
NIDN : 1025059301

Padang, 19 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 PENGHANCUR KERTAS OTOMATIS
MENGUNAKAN MIKROKONTROLER ARDUINO DENGAN
NOTIFIKASI TELEGRAM**

Yang dipersiapkan dan disusun oleh

DIMAS PRASETYO WISNU HARYANTO
21101152620061

Telah dipertahankan di depan dewan penguji
Pada tanggal : 19 Agustus 2025
Dan dinyatakan telah lulus memenuhi syarat

Pembimbing I

Pembimbing II

(Dr. Guslendra, S.Kom, M.Kom)

NIDN : 1016087201

(Nanda Tommy Wirawan, S.Kom, M.Kom)

NIDN : 1025059301

Padang, 19 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 CONSTRUCTION OF AN AUTOMATIC PAPER SHREDDER USING AN ARDUINO MICROCONTROLLER WITH TELEGRAM NOTIFICATION**
Student Name : **Dimas Prasetyo Wisnu Haryanto**
Number : **21101152620061**
Study Program : **Computer Engineering**
Degree Granted : **Strata 1 (S1)**
Advisor : **1. Dr. Guslendra, S.Kom, M.Kom**
2. Nanda Tommy Wirawan, S.Kom, M.Kom

Paper waste is an environmental problem that is often overlooked, even though the amount continues to increase along with office, educational, and household activities. Unused paper is usually thrown away or burned, which actually has a negative impact on the environment. This research aims to design an automatic paper shredder based on the Arduino Mega 2560 microcontroller equipped with real-time notifications via the Telegram application. With this system, users can know the status of the device directly from their smartphone, such as when paper is entered, when the protector is full, or if the device is not accessed properly. This device uses an LDR sensor to detect paper, an ultrasonic sensor to measure the capacity of the container, and RFID as a security system for the device. The notification sending process is carried out via the NodeMCU ESP8266 which is connected to the internet. Test results show that this device works automatically, providing accurate information, and can help manage paper waste more efficiently. This device also contributes to maintaining document confidentiality and encourages the use of Internet of Things (IoT)-based technology in everyday life.

Keywords: Automatic paper shredder, Arduino Mega 2560, RFID, LDR sensor, Ultrasonic sensor, Telegram, Internet of Things (IoT)

ABSTRAK

Judul Skripsi : RANCANG BANGUN ALAT PENGHANCUR KERTAS OTOMATIS MENGGUNAKAN MIKROKONTROLER ARDUINO DENGAN NOTIFIKASI TELEGRAM
Nama : Dimas Prasetyo Wisnu Haryanto
No. BP : 21101152620061
Program Studi : Sistem Komputer
Jenjang Pendidikan : Strata 1 (S1)
Pembimbing : 1. Dr. Guslendra, S.Kom, M.Kom
2. Nanda Tommy Wirawan, S.Kom, M.Kom

Limbah kertas menjadi salah satu persoalan lingkungan yang cukup sering diabaikan, padahal jumlahnya terus meningkat seiring dengan aktivitas perkantoran, Pendidikan, dan rumah tangga. Kertas yang tidak terpakai biasanya dibuang atau dibakar, yang justru berdampak negatif pada lingkungan. Penelitian ini bertujuan untuk merancang alat penghancur kertas otomatis berbasis mikrokontroler Arduino Mega 2560 yang dilengkapi dengan notifikasi real-time melalui aplikasi Telegram. Dengan sistem ini, pengguna dapat mengetahui status alat secara langsung dari smartphone, seperti saat kertas masuk, ketika penampungan penuh, atau jika alat tidak diakses dengan benar. Alat ini menggunakan sensor LDR untuk mendeteksi kertas, sensor ultrasonic untuk mengukur kapasitas penampungan, serta RFID sebagai sistem keamanan meangakses alat. Proses pengiriman notifikasi dilakukan melalui NodeMCU ESP8266 yang terhubung ke internet. Hasil pengujian menunjukkan alat ini bekerja secara otomatis memberikan informasi akurat, dan dapat membantu mengelola limbah kertas dengan lebih efisien. Alat ini juga berkontribusi dalam menjaga kerahasiaan dokumen serta mendorong pemanfaatan teknologi berbasis Internet Of Things (IoT) dalam kehidupan sehari-hari.

Kata kunci : Penghancur kertas otomatis, Arduino Mega 2560, RFID, Sensor LDR, Sensor Ultrasonic, Telegram, Internet of Things (IoT)

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 PENGHANCUR KERTAS OTOMATIS MENGGUNAKAN MIKROKONTROLER ARDUINO DENGAN NOTIFIKASI TELEGRAM”**.

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** sebagai pendiri Yayasan Perguruan Tinggi Komputer (YPTK) Padang
2. Ibu **Dr. Hj. Zerni Melmusi, S.E, M.M, Ak, C.A** sebagai Pembina Yayasan Perguruan Tinggi Komputer (YPTK) Padang.

3. Ibu **Sitti Rizki Mulyani, S.Pd., MM** sebagai Ketua Yayasan Perguruan Tinggi Komputer (YPTK) Padang.
4. Bapak **Assoc. Prof. Dr. Muhammad Ridwan, SE., MM** selaku Rektor Universitas Putra Indonesia “YPTK” Padang.
5. Ibu **Dr. Rini Sovia, S.Kom., M.Kom** selaku Dekan Fakultas Ilmu Komputer Universitas Putra Indonesia “YPTK” Padang.
6. Ibu **Dr. Retno Devita, S.Kom., M.Kom** selaku Ketua Jurusan Sistem Komputer Universitas Putra Indonesia “YPTK” Padang.
7. Bapak **Dr. Guslendra, S.Kom, M.Kom** selaku Pembimbing I yang telah membimbing dan sangat banyak memberi masukan kepada penulis.
8. Bapak **Nanda Tommy Wirawan, S.Kom, M.Kom** selaku Pembimbing II yang telah membimbing dan sangat banyak memberi masukan kepada penulis.
9. 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, Agustus 2025

DIMAS PRASETYO WISNU HARYANTO
21101152620061

DAFTAR ISI

LEMBAR PERNYATAAN	ii
LEMBAR 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 TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Hipotesa.....	5
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian	6
BAB II TINJAUAN PUSTAKA.....	8
2.1 Konsep Dasar Sistem	8
2.1.1 Karakteristik Sistem	8
2.1.2 Siklus Pengembangan Sistem	10
2.2 Alat-alat Komunikasi di Tahap Analisis	12
2.2.1 Context Diagram	12

2.2.2	Data Flow Diagram	14
2.2.3	Flowchart Program.....	15
2.3	Sistem Kontrol	16
2.3.1	Sistem Loop Terbuka	17
2.3.2	Sistem Loop Tertutup.....	17
2.4	Komponen Utama	18
2.4.1	Arduino Mega 2560	18
2.4.2	ESP8266.....	19
2.4.3	RFID.....	20
2.4.4	Sensor Ultrasonik.....	21
2.4.5	Sensor LDR (Light Dependent Resistor)	21
2.4.6	Relay	22
2.4.7	Motor DC	23
2.4.8	LED	23
2.4.9	Buzzer	24
2.4.10	LCD IC 16x2.....	24
2.4.11	DFPlayer	25
2.5	Komponen Pendukung.....	26
2.5.1	Kabel Jumper	26
2.5.2	Resistor.....	27
2.6	Bahasa Pemrograman Arduino.....	27
2.6.1	Struktur Program Arduino	27
2.6.2	Tipe Data.....	28
2.7.3	Operator dan ekspresi Pemrograman Arduino	29

2.7	Struktur Kontrol dalam pemograman Arduino	31
2.7.1	If Statement	31
2.7.2	If else Statement	31
2.7.3	if-else if-else statement	32
2.8	Struktur Perulangan (Loops)	32
2.8.1	For Loop	32
2.8.2	While Loop	33
2.8.3	Do While Loop	33
BAB III METODOLOGI PENELITIAN		34
3.1	Kerangka Kerja Penelitian	34
3.2	Uraian Kerangka Kerja Penelitian	35
3.2.1	Identifikasi Masalah	35
3.2.2	Pengumpulan Data	36
3.2.3	Metode Penelitian	36
3.2.3.1	Waktu Penelitian	37
3.2.3.2	Analisa Sistem	38
3.2.3.3	Perancangan Sistem	38
3.2.3.4	Pembuatan Alat	40
3.2.3.5	Pengujian Sistem	41
BAB IV ANALISA DAN HASIL		42
4.1	Desain Sistem Secara Umum	42
4.1.1	<i>Context Diagram</i>	42
4.1.2	<i>Data Flow Diagram (DFD)</i>	46
4.1.3	Blok Diagram	48

4.2	Prinsip Kerja Sistem.....	49
4.3	Rancangan Fisik Alat	50
4.4	Desain Secara Terinci	52
4.4.1	Rangkaian RFID	52
4.4.2	Rangkaian Sensor LDR.....	53
4.4.3	Rangkaian Sensor Ultrasonik.....	53
4.4.4	Rangkaian LCD.....	54
4.4.5	Rangkaian LED 1 & 2.....	55
4.4.6	Rangkaian DFPlayer & Speaker	56
4.4.7	Rangkaian Buzzer	57
4.5	Rancangan Modul Program.....	58
4.5.1	<i>Flowchart</i>	58
4.5.2	<i>Listing Program</i>	62
BAB V PENGUJIAN SISTEM		66
5.1	Pengujian Sistem Permodul	66
5.1.1	Pengujian Rangkaian Sistem Minimum.....	66
5.1.2	Pengujian Rangkaian RFID	69
5.1.3	Pengujian Sensor <i>Ultrasonic</i>	70
5.2	Pengujian sistem keseluruhan	70
BAB VI PENUTUP		77
6.1	Kesimpulan	77
6.2	Saran.....	78
DAFTAR PUSTAKA.....		79
LAMPIRAN.....		80

DAFTAR TABEL

Tabel 2.1 Tahapan-tahapan dalam SDLC	11
Tabel 2.2 Simbol <i>Context Diagram</i>	13
Tabel 2.3 Simbol <i>Data Flow Diagram</i>	15
Tabel 2.4 Simbol <i>Flowchart</i>	16
Tabel 2.5 Data Board Arduino Mega 2560	19
Tabel 2.6 Operator Pemrograman Arduino	30
Tabel 3.1 <i>Hardware dan Software</i>	37
Tabel 3.1 Jadwal Penelitian	38
Tabel 5.1 Pengujian Rangkaian RFID	69
Tabel 5.2 Pengujian Sensor LDR	70
Tabel 5.3 Pengujian sensor Ultrasonik	70

DAFTAR GAMBAR

Gambar 2.1 Tahapan-tahapan Dalam SDLC	12
Gambar 2.2 Sistem Kendali <i>Loop</i> Terbuka.....	17
Gambar 2.3 Sistem Kendali <i>Loop</i> Tertutup	18
Gambar 2.4 Arduino Mega 2560	19
Gambar 2.5 ESP 8266	20
Gambar 2.6 RFID.....	21
Gambar 2.7 Sensor Ultrasonic	21
Gambar 2.8 Sensor LDR.....	22
Gambar 2.9 Driver Motor L298N	22
Gambar 2.10 Motor DC	23
Gambar 2.11 LED	24
Gambar 2.12 Buzzer.....	24
Gambar 2.13 LCD	25
Gambar 2.14 DFPlayer.....	25
Gambar 2.15 Kabel Jumper	26
Gambar 2.16 Resistor.....	27
Gambar 3.1 Kerangka Kerja Penelitian	34
Gambar 4.1 <i>Context Diagram</i>	43
Gambar 4.2 <i>Data Flow Diagram</i>	46
Gambar 4.3 Blok Diagram	49
Gambar 4.4 Rancangan Alat Fisik Tampak Depan.....	51
Gambar 4.5 Rancangan Alat Fisik Tampak Atas.....	51
Gambar 4.6 Rangkaian RFID.....	52

Gambar 4.7 Rangkaian Sensor LDR.....	53
Gambar 4.8 Rangkaian Sensor Ultrasonik.....	54
Gambar 4.9 Rangkaian LCD.....	55
Gambar 4.10 Rangkaian LED 1 & 2.....	56
Gambar 4.11 Rangkaian DFPlayer & Speaker	57
Gambar 4.12 Rangkaian Buzzer	57
Gambar 4.13 <i>Flowchart</i> Arduino Mega 2560.....	61
Gambar 5.1 Tampilan Arduino IDE.....	66
Gambar 5.2 Tampilan <i>new edit program</i>	67
Gambar 5.3 Tampilan <i>save program</i>	67
Gambar 5.4 Tampilan <i>compile</i> pada Arduino IDE	68
Gambar 5.5 Tampilan <i>upload</i> pada Arduino IDE.....	68
Gambar 5.6 Tampilan <i>Upload</i> selesai.....	69
Gambar 5.7 Sistem Alat Tidak Aktif	71
Gambar 5.8 Sistem Alat Aktif.....	71
Gambar 5.9 Mengaktifkan Hotspot Portabel	72
Gambar 5.10 Scan Kartu RFID	72
Gambar 5.11 Scan Kartu Terdaftar	73
Gambar 5.12 Menekan Button	73
Gambar 5.13 Tampilan Alat Sedang Bekerja	74
Gambar 5.14 Hasil Pemotongan Kertas	74
Gambar 5.15 Tampilan Penghancuran Selesai.....	75
Gambar 5.16 Tampilan Kertas Sudah Penuh	75
Gambar 5.17 Tampilan Status Alat Dari Telegram	76

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kertas merupakan salah satu bahan yang memiliki peran penting dalam kehidupan sehari-hari. Kertas digunakan sebagai media penyimpan informasi di berbagai bidang, seperti pendidikan, perkantoran, bisnis, maupun kebutuhan rumah tangga. Fungsinya yang luas membuat penggunaan kertas terus meningkat seiring dengan perkembangan aktivitas manusia. Namun, peningkatan penggunaan kertas ini menimbulkan permasalahan baru, yaitu limbah kertas yang jumlahnya semakin bertambah setiap harinya. Limbah kertas yang tidak dikelola dengan baik dapat berdampak buruk pada lingkungan dan menimbulkan pemborosan sumber daya alam. Indonesia merupakan negara dengan penduduk dan dikenal dengan negara kepulauan yang terletak di daerah tropis dan memiliki kemampuan ekonomi yang besar untuk masa yang akan datang sampah atau limbah kertas merupakan masalah yang sangat serius bagi lingkungan, dikarenakan kertas merupakan bahan yang sulit terurai (Budi Setiawan et al., 2022).

Menurut data Kementerian Lingkungan Hidup (KLHK) 2023), sebanyak 12% atau 34,5 ton sampah kertas dihasilkan di Indonesia. Jumlah tersebut menunjukkan bahwa sampah kertas masih menjadi masalah serius di Indonesia. Masalah ini bisa berdampak buruk pada lingkungan seperti meningkatnya pencemaran lingkungan, jika tidak ada langkah konkret untuk mengatasinya, limbah kertas dapat menyebabkan berbagai dampak negatif, seperti pencemaran udara akibat pembakaran kertas, kerusakan ekosistem karena penumpukan sampah,

dan peningkatan emisi karbon dari proses produksi kertas baru yang membutuhkan bahan baku berupa kayu.

Proses penghancuran kertas yang sudah tidak terpakai seperti dokumen-dokumen kantor ataupun perguruan tinggi, salah cetak dan lain-lain sangat dibutuhkan oleh instalasi-instalasi maupun perusahaan. Selain menghindari teradinya penumpukan-penumpukan kertas yang hanya memenuhi ruangan. Penanganan masalah limbah kertas sudah dilakukan dengan mencacah dengan ukuran tertentu, namun kapasitas produksinya belum maksimal. Pendidikan tinggi sebagai bagian dari pendidikan nasional yang dibina dan dikembangkan untuk memiliki kemampuan akademis dan keterampilan sesuai dengan kebutuhan pembangunan dan pengembangan ilmu pengetahuan dan teknologi diharapkan mampu melakukan tindakan-tindakan pencegahan penyalahgunaan dokumen (Mukhamad Kosim, 2021)

Salah satu cara untuk membantu proses penghancuran kertas tersebut adalah membuat mesin penghancur kertas sederhana, Dengan adanya mesin pencacah kertas ini harapannya adalah untuk mengurangi pembakaran kertas yang dapat menimbulkan polusi udara dan bisa memanfaatkan sampah kertas menjadi barang yang bernilai dan dapat membantu proses daur ulang sampah (Kadek Sastra Guna, 2023)

Penelitian yang dilakukan terhadap alat penghancur kertas yang dilakukan (Tamaji et al., 2021) masih menggunakan komponen mekanis dan elektronik yang lebih sederhana serta langsung berfokus pada control motor untuk menjalankan fungsi penghancur. Perbedaan penelitian ini dengan sebelumnya yaitu penelitian ini lebih berfokus mengembangkan alat penghancur kertas dengan memanfaatkan

mikrokontroler Arduino sebagai otak operasi alat, dan sensor ldr untuk mendeteksi kertas yang masuk ke dalam alat, dan Telegram yang akan memberi informasi status alat secara *real-time*.

Maka dari permasalahan di atas penulis ingin mengangkat judul tentang
**“RANCANG BANGUN ALAT PENGHANCUR KERTAS OTOMATIS
 MENGGUNAKAN MIKROKONTROLER ARDUINO DENGAN
 NOTIFIKASI TELEGRAM”**

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, dalam melakukan penelitian ini dibuat rumusan masalah sebagai berikut :

1. Bagaimana Arduino Mega 2560 sebagai mikrokontroler pada alat dapat berfungsi dengan baik?
2. Bagaimana RFID pada alat dapat mengontrol akses pengaksesan alat sebelum dijalankan sebagai sistem keamanan alat monitoring?
3. Bagaimana sensor LDR pada alat penghancur kertas otomatis dapat mendeteksi kertas?
4. Bagaimana sensor Ultrasonik pada alat penghancur kertas otomatis dapat bekerja sebagai pendeteksi kertas yang sudah penuh di kotak penampungan?
5. Bagaimana Buzzer dan LED pada alat penghancur kertas otomatis bekerja sebagai indikator untuk menunjukkan status alat?
6. Bagaimana LCD dalam menampilkan keadaan status alat?

7. Bagaimana Esp8266 dapat bekerja sebagai penghubung antara Telegram dengan alat?
8. Bagaimana Telegram dapat memberikan informasi secara real-time kepada pengguna tentang status alat?

1.3 Batasan Masalah

Agar penelitian ini tidak menyimpang dari tujuan penelitian, maka penelitian ini difokuskan pada masalah yang berkaitan dengan hal-hal sebagai berikut:

1. Penelitian ini hanya difokuskan pada perancangan alat penghancur kertas otomatis menggunakan mikrokontroler Arduino
2. Penelitian ini hanya terbatas pada penggunaan alat penghancur kertas yang dirancang untuk kertas berukuran A4, dan tidak mencakup jenis atau ukuran kertas lainnya
3. Pengujian alat ini dilakukan pada kertas dengan kondisi normal, tidak basah, tidak kusut, dan tidak berbahan khusus
4. Sistem notifikasi alat menggunakan aplikasi Telegram untuk memberi informasi kepada pengguna.
5. Alat ini dirancang untuk penggunaan pribadi atau skala kecil dan tidak diperuntukan untuk industri atau skala besar

1.4 Hipotesa

Berdasarkan pada perumusan masalah diatas, maka dapat diambil beberapa hipotesis yaitu :

1. Diharapkan Mikrokontroler Arduino dapat menjalankan seluruh fungsi alat penghancur kertas otomatis dengan baik.
2. Diharapkan RFID dapat mengontrol pengaksesan alat sebelum dijalankan sebagai sistem keamanan alat.
3. Diharapkan Sensor LDR dapat mendeteksi keberadaan kertas yang masuk ke alat untuk memicu proses penghancuran.
4. Diharapkan Sensor Ultrasonik dapat mendeteksi kapasitas penampungan kertas dan memberikan peringatan apabila tempat penampungan sudah penuh.
5. Diharapkan Buzzer dan LED dapat memberikan informasi berupa peringatan status alat.
6. Diharapkan LCD dapat memberikan tampilan informasi mengenai status alat.
7. Diharapkan ESP8266 dapat digunakan untuk menghubungkan alat dengan jaringan internet sehingga data status alat dapat dikirimkan ke perangkat pengguna.
8. Diharapkan Telegram dapat memberikan notifikasi real-time kepada pengguna mengenai status alat.

1.5 Tujuan Penelitian

Penelitian ini dilakukan untuk mendapatkan tujuan penelitian tentang hal-hal yang dibahas sebagai berikut :

1. Mengembangkan alat penghancur kertas yang berfungsi otomatis menggunakan Mikrokontroler Arduino sebagai pengendali utama.
2. Mengetahui cara kerja alat penghancur kertas otomatis berbasis Arduino Mega 2560
3. Mengetahui keunggulan dari alat penghancur kertas otomatis berbasis Arduino Mega 2560
4. Memanfaatkan LDR sebagai pendeteksi kertas yang masuk
5. Memanfaatkan Telegram sebagai media pemberi informasi kepada pengguna mengenai status penghancuran kertas

1.6 Manfaat Penelitian

Selain memiliki tujuan penelitian, penelitian diharapkan mempunyai manfaat penelitian sebagai berikut:

A. Bagi Penulis

1. Penulis dapat meningkatkan pemahaman tentang perancangan perangkat mikrokontroler dan pemograman Arduino.
2. Penulis dapat memperoleh wawasan lebih mendalam untuk menganalisa suatu permasalahan dan juga mampu mencari solusinya.
3. Sebagai syarat untuk mendapatkan gelar di jenjang Pendidikan Strata 1(S1).

B. Bagi Program Studi

1. Menambah jumlah refrensi penelitian dalam bidang IoT (internet of Things)
2. Dapat dijadikan sebagai pedoman bagi mahasiswa selanjutnya untuk matakuliah yang berhubungan dengan hasil akhir penelitian

3. Dengan penelitian ini diharapkan dapat menambah motivasi bagi mahasiswa sistem komputer untuk berkarya lebih baik lagi dalam menggali ilmu pengetahuan khususnya dalam bidang teknologi komputer

C. Bagi Masyarakat

1. Membantu instansi atau perorangan dalam menghancurkan kertas secara efektif, sehingga mempermudah pengelolaan limbah kertas
2. Membantu menjaga kerahasiaan informasi atau dokumen penting yang sudah tidak terpakai.

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Pada umumnya setiap organisasi mempunyai sistem informasi dalam mengumpulkan, menyimpan, melihat, dan menyalurkan informasi dalam membuat perancangan sistem informasi. Konsep dasar sistem merupakan sekelompok komponen berbasis komputer yang dibuat oleh manusia dalam mengelola data, menyimpan, menghimpun kerangka kerja serta mengkoordinasikan sumber daya manusia dan komputer untuk mengubah sistem masukan menjadi sistem keluaran untuk mencapai tujuan dan sasaran yang telah ditetapkan sebelumnya. (Mei Prabowo, 2020)

2.1.1 Karakteristik Sistem

Beberapa karakteristik yang harus dimiliki sebuah sistem yaitu :

a. Komponen (*Component*)

Sistem terdapat komponen-komponen beberapa diantaranya melakukan interaksi dengan membentuk satu kesatuan dan saling bekerja sama yang terdiri dari berbagai cabang sistem.

b. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar pada sistem (*environment*) merupakan pengaruh operasi sistem oleh lingkungan luar sistem. Lingkungan luar sangat mempengaruhi yang bersifat menguntungkan harus dijaga dan yang bersifat merugikan tetap dijaga namun dikendalikan.

c. Batasan Sistem (*Boundary*)

Lingkup luar sistem yang dibatasi oleh ruang lingkup (*scope*) atau sistem dengan batas sistem lain yang sesuai bundaran daerahnya.

d. Penghubung Sistem (*interface*)

Penghubung sistem merupakan alat bantu yang menghubungkan antara satu subsistem ke subsistem lainnya. Melalui penghubung sumber-sumber daya dimungkinkan mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem ini akan menjadi masukan (*input*) untuk subsistem dengan alat bantu penghubung ini.

e. Masukan Sistem (*input*)

Masukan sistem merupakan sumber daya yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). Maintenance input adalah sumber daya yang dimasukkan agar sistem dapat beroperasi. Signal input adalah sumber daya yang diproses untuk didapatkan keluaran. Sebagai contohnya adalah *software* yang merupakan yang *maintenance* supaya sistem tetap berjalan. Sedangkan signal input adalah data sinyal pada proyektor.

f. Keluaran Sistem (*output*).

Keluaran sistem adalah energi yang dihasilkan setelah pemrosesan inputan keluaran yang dibuang maupun dibutuhkan. Contoh sebuah komputer mengeluarkan suhu panas dikatakan sebagai energi buang dan informasi sebagai keluaran energi yang dipakai.

g. Pengolahan Sistem

Pengolahan sistem merupakan bagian proses yang merubah *input* menjadi *output*. Contohnya sistem akuntansi dengan pengolahan data menjadi laporan-laporan keuangan. Sistem mesin cuci yang merubah baju kotor menjadi bersih.

h. Sasaran Sistem

Sasaran sistem merupakan tujuan (*goal*) atau sasaran (*objective*). (Dedy Rahman Prehanto, 2020)

2.1.2 Siklus Pengembangan Sistem

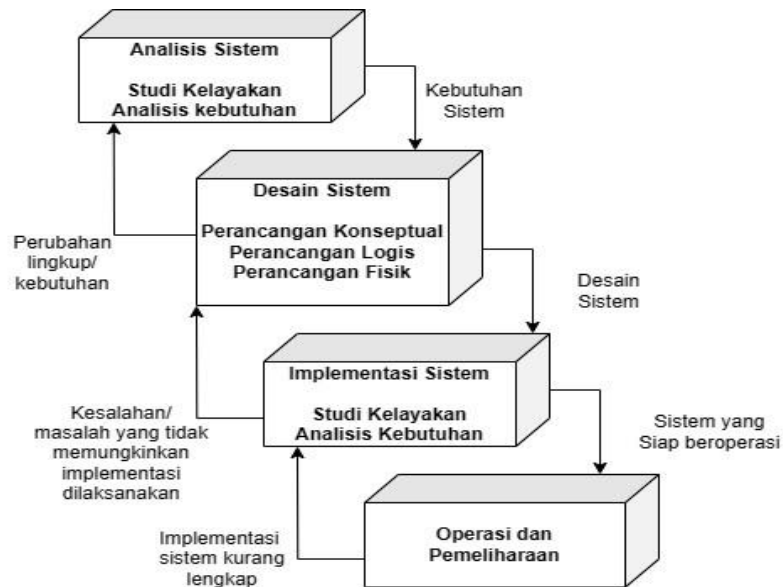
Metodologi klasik atau tradisional yang biasa dipakai untuk mengembangkan sistem informasi yaitu yang dinamakan *System Development Life Cycle* (SDLC). Metodologi ini mencakup kegiatan yang mengawali proyek, menganalisis kebutuhan data, merancang sistem, membuat sistem, dan memelihara sistem (Kadir, 2020) Tabel 2.1 memperlihatkan sejumlah tahapan dalam SDLC menurut sejumlah versi.

Tabel 2.1 Tahapan-Tahapan Dalam SDLC

Sumber	Tahapan-tahapan Dalam SDLC
Alter	Insiasi, pengembangan, implementasi, operasi, dan pemeliharaan.
Fabb dan Schwab	Studi kelayakan, rencana awal, analisis sistem, desain sistem dan implementasi sistem.
Hoffer, George dan Valacich	Identifikasi dan seleksi proyek, inisiasi dan perancangan proyek, analisis, perancangan logis, akusisi atau pengembangan, implementasi, operasi, evaluasi pasca audit dan pemeliharaan.
Turban, Mclean dan Wetherbe	Inisiasi proyek, analisis sistem dan studi kelayakan, analisis dan perancangan logis, akusisi atau pengembangan, implementasi. Operasi, evaluasi pasca audit dan pemeliharaan.
Zwass	Studi kelayakan, analisis kebutuhan, perancangan logis, perancangan fisik, pengodean dan pengujian, konversi dan kajian-kajian pasca implementasi.

Sumber: (Abdul Khadir, Dasar Perancangan dan Implementasi Database Relasional, 2020:20-21).

Meskipun tahapan dalam SDLC bervariasi, pada prinsipnya secara keseluruhan semua kegiatan yang dilakukan sama saja, model air terjun dalam SDLC diperlihatkan pada gambar 2.1 di bawah ini:



(Sumber: Abdul khadir, *Dasar Perancangan dan Implementasi Databse Relasional*, 2020:20-21).

Gambar 2.1 Tahapan-Tahapan Dalam SDLC

Dalam banyak tahapan, *database* dilibatkan. Itulah sebabnya, di depan telah disinggung bahwa proses perencaan *database* sebenarnya merupakan bagian dalam SDLC.

2.2 Alat-alat Komunikasi di Tahap Analisis

Aturan dasar yang harus dilakukan adalah mendefenisikan secara menyeluruh sistem yang akan dirancang. Hal ini mengandung arti bahwa harus ada gambaran yang jelas mengenai ruang lingkup tentang sistem yang dibahas. Media yang digunakan untuk menggambarkan sistem tersebut adalah *Context Diagram* dan *Data Flow Diagram (DFD)* dan *Flowchart*.

2.2.1 Context Diagram

Context Diagram menggambarkan sistem yang dirancang secara keseluruhan, semua external entity harus digambarkan sedemikian rupa, sehingga terlihat data yang mengalir pada input-process-output. *Context Diagram* hanya

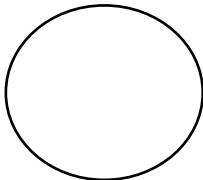
boleh terdiri dari satu proses saja, tidak boleh lebih, dan pada *Context Diagram* tidak digambarkan data store.

Menurut (Sulinta, 2019) Diagram Konteks digunakan untuk mengetahui ruang lingkup dan Batasan-batasan yang ada dalam perangkat lunak yang sedang dikerjakan. Hal ini didapat pada saat Analisa kebutuhan sistem. Batasan yang menjadi dasar dalam Langkah pembuatan diagram konteks adalah:

1. Aktivitas yang dikerjakan oleh perangkat lunak (proses)
2. Siapa yang menggunakan (*entitas*)
3. Apa yang menjadi input dan outputnya (aliran data).

Beberapa simbol yang digunakan dalam pembuatan *Context Diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol Context Diagram

Gambar	Keterangan
	<i>Procces</i> /proses merupakan menggambarkan suatu proses atau sistem yang akan dibangun.
	Proses dapat digambarkan dengan simbol lingkaran atau persegi Panjang dengan sisi-sisi tumpul.
	<i>Entity</i> /entitas/Terminator merupakan entitas atau pengguna dari sistem/aplikasi
	Data <i>flow</i> (aliran data) merupakan aliran data yang masuk dan keluar dari sistem.

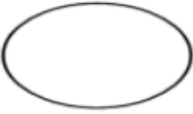
Sumber: (Feri Sulinta, *Strategi Merancang Arsitektur Sistem Informasi Masa Kini*, 2019:129).

2.2.2 Data Flow Diagram

Data *flow* diagram atau dalam bahasa indonesia menjadi Diagram Alir Data adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan (*input*) dan keluaran (*output*).

Menurut (Kusyadi *et al.*, 2021) Data *Flow* Diagram adalah analisis sistem yang dapat Menyusun representasi grafis dari proses data diseluruh organisasi, atau Data *Flow* Diagram dapat disebut dengan alat yang menggambarkan aliran data melalui suatu sistem data dan pekerjaan pada pemrosesan yang dilakukan oleh sistem tersebut. persamaannya meliputi diagram gelembung, grafik transformasi, dan proses model. Alat perencanaan DFD disebut dengan diagram dekomposisi. Definisi kunci dari data *flow* diagram adalah diagram yang terdiri dari proses dan menggambarkan ruang lingkup sistem. Data *flow* diagram merupakan level tertinggi yang mendeskripsikan semua input atau output sistem dalam sistem.. Simbol Data *flow* diagram (DFD) dapat dilihat pada gambar 2.3 berikut:

Tabel 2.3 Simbol *Data Flow Diagram*

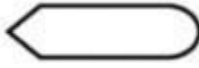
Simbol	Arti
	Entitas (<i>Entity</i>)
	Aliran Data (<i>Data Flow</i>)
	Proses
	Penyimpanan Data (<i>Data Store</i>)

Sumber: (Irpan Kusyadi, et al., *Analisis dan Perancangan Sistem* 2020:77-81).

2.2.3 Flowchart Program

Salah satu diagram yang dikenal luas dan mudah dimengerti adalah diagram alir atau populer dengan sebutan *flowchart*. *Institute Of Distance & Open Learning* memaparkan desain Algoritma, *Flowcharts & Program* terbagi atas *flowchart* sistem, yang memperlihatkan serangkaian prosedur/aktivitas yang terjadi pada sebuah sistem, dan *flowchart* program yang merincikan setiap langkah yang dijalankan komputer untuk menangani proses bisnis yang ingin dicapai oleh sebuah sistem *Flowchart* (Junus Binti, 2020).

Tabel 2.4 Simbol *Flowchart*

Simbol	Nama	Fungsi
	Arus (<i>Flow</i>)	Menyatakan arah sebuah proses. Simbol ini dapat mengarah ke kiri, kanan, atas dan bawah.
	<i>On-Page Connector</i>	Menunjukkan prose lanjutan ke simbol lain yang berada pada halaman yang sama.
	<i>Off-Page Connector</i>	Mengarahkan sambungan proses ke simbol pada halaman yang berbeda.
	<i>Terminal</i>	Menandakan permulaan dan berakhirnya suatu program.
	<i>Input/output data</i>	Mengindikasikan pernyataan membaca masukan dari <i>user</i> dan juga berfungsi sebagai symbol untuk menampilkan hasil dari proses tanpa menspesifikasikan alat.
	Proses	Merepresentasikan tindakan yang dilakukan dalam sistem.
	Decision	Mengisyaratkan proses pilihan terhadap suatu kondisi yang mengarah kepada jawaban ya/benar tidak/salah.
	Display	Menampilkan output ke layar

Sumber: (Fadhla Binti Junus, *Dasar Pemrograman Berbasis Web dengan PHP Native Procedural & MySql*, 2021:37-38).

2.3 Sistem Kontrol

Sistem kontrol merupakan suatu sistem yang dikontrol secara otomatis menggunakan komputer digital dan operasinya di bawah pengawasan manusia. Sesuai dengan fungsi pengontrolan secara menyeluruh, maka komponen sistem pengontrolan dapat dibagi atas 4 bagian yaitu: sensor (*Transducer*), pemroses,

penggerak dan penguat.

Sistem kontrol berdasarkan cara kerjanya dapat dibagi menjadi dua bagian, yaitu sistem kontrol *loop* terbuka dan *loop* tertutup. Adapun pembagiannya sebagai berikut:

2.3.1 Sistem Loop Terbuka

Sistem pengendalian loop terbuka adalah sistem pengendalian yang keluarannya (*process variable*) tidak dapat dibandingkan dengan masukan acuan (*set point*). Sebagai contoh sistem pengendalian loop terbuka adalah mesin cuci. Perendaman, pencucian, dan pembilasan dilakukan atas basis waktu. Gambar diagram blok sistem pengendalian loop terbuka dapat dilihat seperti pada gambar 2.2 berikut :

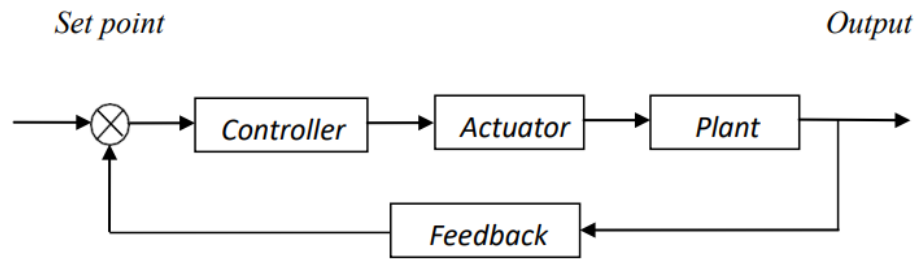


(Sumber: Kiagus Ahmad Roni, Cekmas Cekdin, *Sistem Kendali Proses Industri*, 2020:9-10).

Gambar 2.2 Sistem Kendali Loop Terbuka

2.3.2 Sistem Loop Tertutup

Sistem pengendalian loop tertutup adalah sistem pengendalian yang keluarannya (*process variable*) dapat dibandingkan dengan masukan acuan (*set point*). Sebagai contoh sistem pengendalian *loop* tertutup adalah pengendalian temperature reaktan (*process variable*) secara otomatis pada sebuah *reactor batch* pada gambar 1.2 Gambar blok diagram blok sistem pengendalian *loop* tertutup dapat dilihat seperti pada Gambar 2.3 berikut :



(Sumber: Kiagus Ahmad Roni, Cekmas Cekdin, *Sistem Kendali Proses Industri*, 2020:9-10).

Gambar 2.3 Sistem Kendali 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 Mega 2560

Arduino Mega umumnya dibuat menggunakan jenis mikrokontroler ATmega 2560. Sesuai dengan namanya, Arduino ini dibekali dengan prosesor ATmega2560 yang memiliki 54 pin digital I/O (dimana 15 pin dapat digunakan sebagai output PWM), 16 pin analog Input, 4 pin UART, 2x3 pin ICSP (untuk memprogram Arduino dengan software lain), dan kabel USB komputer yang sekaligus digunakan sebagai sumber tegangan. Spesifikasi dari Arduino mega 2560 secara lebih lengkap seperti diuraikan pada tabel 2.5 sebagai berikut:

Tabel 2.5 Data Board Arduino Mega 2560

Spesifikasi	Uno	Mega	Mega2560
Proseor	ATMega328	ATMega1280	ATMega2560
Memori Program	32KB	128KB	256KB
Memori data	2KB	8KB	8KB
EEPROM	1KB	4KB	4KB
Jumlah Pin	14	54	54
Pin digital I/O	6	16	16
Keluaran PWM	6	14	14
Port Serial	1	4	4

Sumber: (Noor Siti Halimah, Semua Bisa Belajar Arduino, 2021:27-28)



(Sumber: Widodo, Budiharto, Menguasai Pemrograman Arduino dan Robotik, 2020).

Gambar 2.4 Arduino Mega 2560

2.4.2 ESP8266

ESP8266 adalah serangkaian sistem hemat daya pada mikrokontroler chip dengan wi-Fi terintegrasi dan Bluetooth dual-mode. Pada mikrokontroler ini sudah terdapat modul wi-Fi di dalamnya, sehingga sangat berguna untuk membuat sistem aplikasi IoT. Seri ESP8266 menggunakan mikroprosesor Tensilica Xtensa LX6 dalam varian inti ganda dan inti tunggal dan mencakup saklar antenna terintegrasi, RF balun, penguat daya, penguat penerima kebisingan rendah, filter, dan modul manajemen energi makanan. ESP8266 dibuat dan dikembangkan oleh Espressif Systems, sebuah perusahaan Cina yang berbasis di Shanghai dan diproduksi oleh

TSMC menggunakan proses 40nm mereka. Berikut gambar 2.5 memperlihatkan ESP 8266.



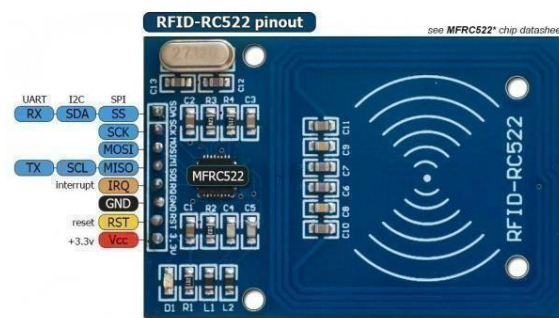
(Sumber :Dodi Yudo Setyawan et al.,*Internet of Things ESP8266 ESP8266 Web Serve*, 2022 : 7).

Gambar 2.5 ESP 8266

2.4.3 RFID

RFID adalah singkatan dari *Radio Frequency Identification* atau dalam bahasa Indonesia berarti identifikasi frekuensi radio. Pengertian lainnya adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut label RFID atau transponder untuk menyimpan dan mengambil data jarak jauh. Label RFID berisi informasi yang disimpan secara elektronik dan dapat dibaca hingga beberapa meter. Sistem pembaca RFID tidak memerlukan kontak langsung seperti sistem pembaca kode batang (*barcode*). Label RFID terdiri atas mikrochip silikon dan antena.

Beberapa ukuran label RFID dapat mendekati ukuran sekecil butir beras. Label yang pasif tidak membutuhkan sumber tenaga, sedangkan label yang aktif membutuhkan sumber tenaga untuk dapat berfungsi. Bentuk dari modul RFID MFRC522 ditunjukkan pada Gambar 2.6.



Sumber : Setiwardhana, SigitWasista, Delima Ayu, 2019

Gambar 2.6 RFID

2.4.4 Sensor Ultrasonik

Sensor Ultrasonik HC-SR04 adalah modul pengukur jarak dengan ultrasonic dengan harga yang cukup murah. Yang didesain khusus untuk teknologi robotika. Dengan ukurannya yang cukup kecil (2,1cm x 4,5cm), sensor HC-SR04 dapat mengukur jarak antara 2 cm sampai 500 cm dengan resolusi 0,3 cm. Berikut gambar 2.7 memperlihatkan Sensor Ultrasonic.



(Sumber : Mada Sanjaya et al., Eksperimen Virtual Mikrokontroler Arduino Menggunakan Tinkercad ,2020 : 127).

Gambar 2.7 Sensor Ultrasonic

2.4.5 Sensor LDR (Light Dependent Resistor)

Sensor Cahaya LDR adalah salah satu jenis resistor yang dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Besarnya nilai hambatan pada sensor cahaya LDR tergantung pada besar kecilnya cahaya yang diterima oleh LDR itu sendiri. LDR sering disebut dengan alat atau sensor yang berupa resistor yang peka terhadap cahaya. Gambar dari sensor ldr dapat dilihat pada 2.8 berikut.



(Sumber : Deni Tri Laksono, *Sensor dan actuator menggunakan Arduino 2*, 2023 : 46-47).

Gambar 2.8 Sensor LDR

2.4.6 Relay

Relay adalah komponen elektronika yang berupa saklar atau switch elektrik yang dioperasikan menggunakan listrik. Komponen relay menggunakan prinsip elektromagnetik sebagai penggerak kontak saklar, sehingga dengan menggunakan arus listrik yang kecil atau *low power*, dapat menghantarkan arus listrik yang memiliki tegangan lebih tinggi. Gambar dari Relay dapat dilihat pada gambar 2.9 berikut.

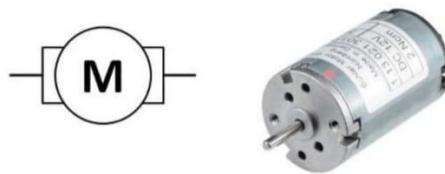


(Sumber : Muhammad Naim *Buku ajar sistem control dan kelistrikan mesin*, 2021 : 29).

Gambar 2.9 Driver Motor L298N

2.4.7 Motor DC

Motor Listrik DC atau DC Motor adalah suatu perangkat yang mengubah listrik menjadi energi kinetic atau Gerakan (*motion*). Motor DC ini juga dapat disebut sebagai Motor Arus Searah. Seperti Namanya, DC Motor memiliki dua terminal dan memerlukan tegangan arus searah atau DC (*Direct Current*) untuk dapat menggerakkannya. Motor Listrik DC ini biasanya digunakan pada perangkat-perangkat Elektronik dan listrik yang menggunakan sumber listrik DC seperti Vibrator Ponsel, Kipas Dc dan Bor Listrik. Berikut gambar 2.10 memperlihatkan Motor DC.



(Sumber : Muhammad Naim, Buku Ajar Sistem Kontrol dan Kelistrikan Mesin, 2021 : 159-160).

Gambar 2.10 Motor DC

2.4.8 LED

Light Emitting Dioda atau LED merupakan salah satu jenis diode yang seringkali digunakan sebagai lampu indicator. LED adalah komponen elektronika yang memancarkan cahaya monokromatik Ketika dibias maju. Warna cahaya LED yang dipancarkan bergantung dengan material semikonduktor yang digunakan. Gambar 2.11 di bawah menunjukan bentuk dan simbol LED pada rangkaian elektronika.



(Sumber : Andy Wahyu et al., *Dasar Elektronika Praktis*, 2023 : 77).

Gambar 2.11 LED

2.4.9 Buzzer

Buzzer adalah komponen yang merubah besaran elektrik menjadi besaran suara. Penggunaan buzzer untuk antarmuka mikrokontroler sama prinsipnya dengan LED. Yang diperlukan hanya menghubungkan salah satu PIN dari mikrokontroler ke kaki positif buzzer, dan kaki satunya ke GND rangkaian. Berikut gambar 2.12 memperlihatkan Buzzer.

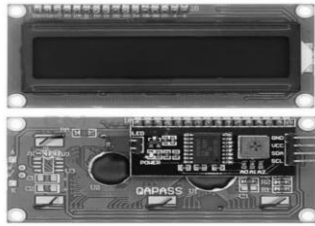


(Sumber : I Gede Suputra Widharma et al., *Mikrokontroler dan Aplikasi*, 2022 : 83 – 84).

Gambar 2.12 Buzzer

2.4.10 LCD IC 16x2

LCD 16x2 adalah display dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks. LCD 16x2 dapat menampilkan sebanyak 32 karakter yang terdiri dari 2 baris dan tiap baris dapat menampilkan 16 karakter. Berikut gambar 2.13 memperlihatkan LCD IC 16x2.



(Sumber : Muhammad Nasir et al., *Internet of Things Aplikasi dan Penerapan Mikrokontroler Arduino*, 2023 : 70-71).

Gambar 2.13 LCD

2.4.11 DFPlayer

Modul MP3 DFPlayer digunakan sebagai pemutar file audio dengan format .mp3 dan menggunakan SD Card sebagai memori penyimpanan. Modul DFPlayer membutuhkan ampliflier atau *speaker* sebagai penguat suara karena modul ini tidak memiliki output. Untuk dapat mengakses file audio harus dilakukan dengan cara membuat folder bernama “mp3” pada SD Card terlebih dahulu, setelah itu audio didalam folder “mp3” haruslah diubah menjadi format angka 4 digit seperti 0001, 0002, dst. Untuk mencegah terjadinya noise pada audio dapat dilakukan dengan memberikan resistor pada kabel RX. Bentuk dari MP3 DFPlayer dapat dilihat pada Gambar 2.14.



Sumber : Rizki Priya Pratama et al, *Implementasi DFPlayer untuk Al-Qur'an Digital berbasis Mikrokontroler ESP32*, 2020

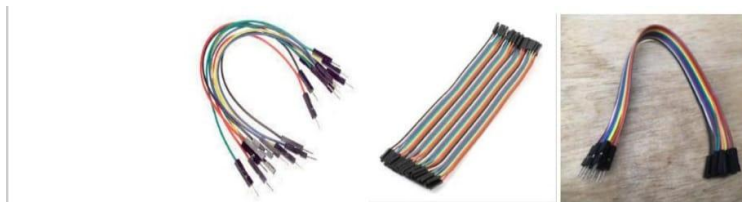
Gambar 2.14 DFPlayer

2.5 Komponen Pendukung

Komponen pendukung digunakan sebagai pelengkap dari sebuah sistem agar dapat berjalan dengan maksimal. Komponen – komponen pendukung tersebut akan dijabarkan sebagai berikut:

2.5.1 Kabel Jumper

Kabel Jumper merupakan kabel yang berfungsi sebagai penghubung (*connector*) pada sebuah sirkuit elektrik yang sering digunakan untuk menghubungkan atau memutus hubungan pada suatu sirkuit. Jumper juga bisa digunakan dalam melakukan setting pada sebuah papan motherboard elektrik seperti *motherboard* computer. Namun belakangan ini proses penyetingan yang sering menggunakan kabel jumper sudah banyak ditinggalkan, karena sudah banyak perangkat sistem menggunakan *auto setting* untuk lebih mempermudah pengguna untuk mengurangi penggunaan kabel *jumper*. Kabel *jumper* biasanya digunakan pada *harddisk*, *motherboard*, dan *optical disk*, dan pada beberapa VGA card tertentu. Kabel Jumper terdiri dari beberapa jenis diantaranya adalah : *male to male*, *female to female*, dan *male to female*. Gambar kabel jumper dapat dilihat pada gambar 2.15.



(Sumber : Dirja Nur Ilham et al., Monitoring dan Stimulasi Detak Jantung dengan Murotal Al-Quran Berbasis Internet of Things (IoT), 2020 : 32-33).

Gambar 2.15 Kabel Jumper

2.5.2 Resistor

Resistor adalah salah satu komponen dasar dalam elektronika yang berfungsi untuk menghambat dan mengendalikan aliran arus listrik dalam sebuah sirkuit. Resistor memiliki nilai resistansi, yang mengukur seberapa besar hambatan yang diberikannya terhadap aliran arus. Dengan menyediakan hambatan tertentu, resistor memungkinkan pengaturan tegangan, arus, atau pembagian daya dalam sirkuit. Resistansi diukur dalam satuan ohm (Ω). Berikut gambar 2.15 memperlihatkan Resistor.



(Sumber : Wawan Kurniawan et al., *Komponen-Komponen Elektronika Pengenalan, Jenis, dan Aplikasi*, 2023 :4-5).

Gambar 2.16 Resistor

2.6 Bahasa Pemrograman Arduino

Bahasa pemrograman Arduino pada dasarnya menggunakan Bahasa pemrograman C. Bahasa C sendiri merupakan Bahasa tingkat tinggi yang sangat populer dan banyak digunakan oleh para programmer. Dengan demikian aturan penulisan dan penggunaan dari Bahasa Arduino akan sama dengan Bahasa C.

2.6.1 Struktur Program Arduino

Program pada Arduino dikenal dengan istilah “sketsa”. Sketsa adalah istilah khusus yang digunakan dalam ekosistem Arduino untuk menyebut program atau kode yang ditulis untuk papan Mikrokontroler Arduino. Memahami struktur dasar

sketsa Arduino sangat penting bagi siapa pun yang ingin menulis dan mengembangkan program dengan efektif. Struktur dasar ini terdiri dari dua fungsi utama, yaitu `setup()` dan `loop()`, yang masing-masing memiliki peran dan fungsi khusus dalam sketsa (Hasan et al., 2024).

```
Void setup() {
    // put your setup code here, to run once;
}

Void loop() {
    // put your main code here, to run repeatedly;
}
```

2.6.2 Tipe Data

Tipe data menentukan jenis informasi apa yang dapat disimpan dalam variabel, mirip dengan bagaimana jenis wadah yang berbeda dapat menyimpan berbagai jenis barang (Hasan et al., 2024).

1. **Int (Integer):** Mirip dengan wadah kecil yang hanya dapat menyimpan bilangan bulat tanpa desimal. Rentang nilai dari int adalah dari -32,768 hingga 32,767. Int
count = 10;
2. **Unsigned int:** Seperti wadah kecil yang hanya bisa menampung bilangan bulat positif, dengan rentang dari 0 hingga 65,535. Unsigned int maxCount = 65535;
3. **Long:** Bayangkan ini sebagai wadah yang lebih besar dibandingkan int, dapat menampung bilangan bulat dengan rentang yang jauh lebih besar, dari -2,147,483,648 hingga 2,147,483,647. Long bigNumber = 1000000;
4. **Unsigned long:** seperti wadah besar hanya untuk bilangan bulat positif, dengan rentang 0 hingga 4,294,967,295. Unsigned long largeNumber = 4000000;

5. **Float** : ini adalah wadah khusus yang dapat menyimpan bilangan desimal, berguna untuk nilai-nilai yang memerlukan presisi lebih tinggi, seperti suhu atau ukuran. `Float pi = 3.14159;`
6. **Double**: sama dengan float, tetapi dengan kapasitas penyimpanan bilangan desimal yang lebih tinggi , meskipun pada banyak papan Arduino, double memiliki presisi yang sama dengan float. `Double preciseValue = 0.123456789;`
7. **Char**: Seperti wadah kecil yang hanya bisa menyimpan satu karakter, seperti huruf atau symbol. `Char letter = 'A';`
8. **Boolean**: ini adalah wadah hanya dapat menyimpan dua nilai, yaitu true (benar) atau false (salah), mirip dengan tombol yang hanya bisa berada dalam dua posisi. `Boolean isOn = true;`

2.7.3 Operator dan ekspresi Pemograman Arduino

Dalam pemograman Arduino, operator dan ekspresi adalah alat dasar yang digunakan untuk melakukan berbagai operasi aritmatika, logika, dan manipulasi data. Berikut adalah tabel yang menjelaskan berbagai operator yang digunakan dalam pemograman Arduino.

Tabel 2.6 Operator Pemrograman Arduino

Kategori Operator	Operator	Deskripsi	Contoh
Operator Aritmatika	+	Penjumlahan	a+b
	-	Pengurangan	a-b
	*	Perkalian	a*b
	/	Pembagian	a/b
	%	Modulus (sisa bagi)	a%b
Operator Penugasan	=	Penugasan nilai	a=5
	+=	Penugasan dengan penjumlahan	A+=3(sama dengan a=a +3)
	-=	Penugasan dengan pengurangan	a-=2(sama dengan a = a -2)
	=	Penugasan dengan perkalian	A=4(sama dengan a=a-4)
	/=	Penugasan dengan pembagian	a/=2(sama dengan a=a/2)
	%=	Penugasan dengan modulus	A%=3(sama dengan a=a%3)
Operator Perbandingan	==	Sama dengan	A==b(benar jika a sama dengan b)
	!=	Tidak sama dengan	A!=b(benar jika a tidak sama dengan b)
	>	Lebih besar dari	a>b
	<	Lebih kecil dari	A=	Lebih besar atau sama dengan	a>=b
	<=	Lebih kecil atau sama dengan	a<=b
Operator Logika	&&	Logika AND	A&&b(true jika a dan b sama-sama benar)
		Logika OR	A b(true jika a atau b benar)
	!	Logika NOT	!a (true jika a salah)

(Sumber : Hasan et al., Buku Ajar Mikrokontroler dan Arduino, 2024 : 23-25).

2.7 Struktur Kontrol dalam pemograman Arduino

Struktur control dalam pemograman adalah mekanisme yang memungkinkan program untuk mengambil keputusan, mengulangi operasi, atau melakukan Tindakan tertentu berdasarkan kondisi yang ada. Dalam pemograman Arduino, struktur control sangat penting untuk menciptakan program yang dinamis dan responsive terhadap berbagai input (Hasan et al., 2024).

2.7.1 If Statement

If adalah pernyataan kondisional yang paling dasar, ini mengeksekusi blok kode hanya jika kondisi yang diberikan bernilai benar (true).

Sintaks :

```
If      (kondisi) {  
    //  
}
```

2.7.2 If else Statement

if else digunakan ketika anda ingin mengeksekusi satu blok kode jika kondisi benar, dan blok kode lain jika kondisi salah(false).

Sintaks :

```
If      (kondisi) {  
    //  
}else  
    //  
}
```

2.7.3 if-else if-else statement

If-else if-else digunakan untuk memeriksa beberapa kondisi. Blok kode pertama yang kondisinya benar akan dijalankan.

Sintaks :

```
if      (kondisi) {
    //
}else if (kondisi2) {
    //
}else {
    //
}
```

2.8 Struktur Perulangan (Loops)

Perulangan memungkinkan program untuk mengeksekusi blok kode berulang kali, yang sangat berguna untuk tugas-tugas yang harus dilakukan secara terus-menerus atau mengulangi operasi tertentu.

2.8.1 For Loop

For Loop digunakan ketika anda tahu berapa kali anda ingin mengulangi blok kode. Biasanya digunakan untuk iterasi melalui array atau melakukan operasi sejumlah tertentu.

Sintaks :

```
For    (inisialisasi; kondisi; iterasi) {
    //
}
```

2.8.2 While Loop

While loop digunakan ketika anda ingin mengulangi blok kode selama kondisi tertentu bernilai benar.

Sintaks :

```
While (kondisi) {  
    //  
}
```

2.8.3 Do While Loop

Do-While loop mirip dengan while loop, tetapi dengan satu perbedaan utama: blok kode akan selalu dijalankan setidaknya sekali, karena kondisi diperiksa setelah eksekusi.

Sintaks :

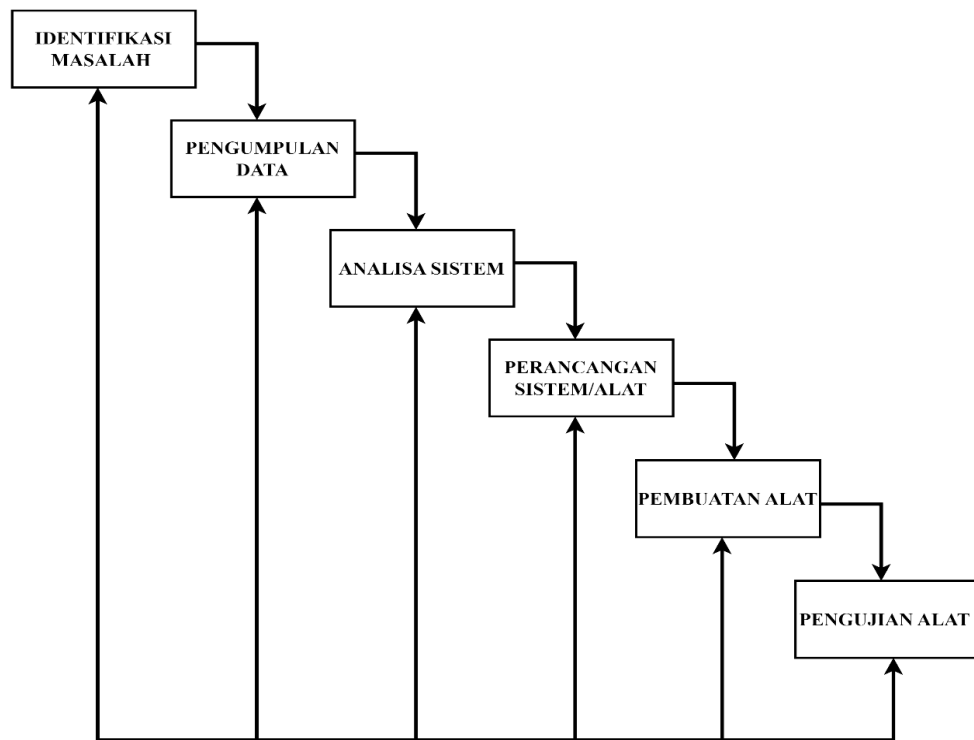
```
do {  
    //  
} while (kondisi)
```

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Kerja Penelitian

Kerangka kerja merupakan langkah-langkah yang akan dilakukan untuk menyelesaikan masalah yang akan dibahas. Adapun kerangka kerja penelitian ini dibuat untuk memberikan struktur atau susunan pembuatan sistem agar lebih jelas. Dalam perancangan ini banyak yang harus ditinjau terlebih dahulu agar dapat mempermudah dan memperjelas langkah-langkah yang digunakan untuk menyelesaikan masalah tersebut. Adapun kerangka kerja dari penelitian ini dapat diilustrasikan pada Gambar 3.1.



Gambar 3.1 Kerangka Kerja Penelitian

3.2 Uraian Kerangka Kerja Penelitian

Berdasarkan kerangka kerja penelitian yang telah diilustrasikan di atas, maka dapat diuraikan tahapan-tahapan dalam penyelesaian penelitian tersebut sebagai berikut:

3.2.1 Identifikasi Masalah

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

alat penghancur kertas otomatis menggunakan mikrokontroler Arduino menghadapi beberapa tantangan yang signifikan. Salah satunya adalah tingkat kesalahan dalam deteksi kertas yang akan dihancurkan, yang menjadi masalah krusial, karena dapat meningkatkan ketidakpastian dalam pengoperasian sistem penghancuran. Selain itu, terdapat kendala pada kerja motor penggerak yang kurang kuat untuk memproses kertas dalam jumlah banyak. Pada faktor kontrol navigasi, mikrokontroler Arduino digunakan untuk mengontrol alat, namun sering kali terjadi gangguan atau kekacauan saat beralih antara mode manual dan otomatis.

Solusi yang dapat diimplementasikan untuk mengatasi masalah tersebut melibatkan penggunaan sensor dan algoritma yang lebih canggih, seperti menggunakan sensor deteksi untuk meningkatkan akurasi dalam mendeteksi kertas yang akan dihancurkan. Peningkatan integritas pemrosesan sinyal dan peningkatan sensitivitas sensor akan membantu mendeteksi objek dengan lebih teliti. Selain itu, sistem akan dilengkapi dengan notifikasi melalui Telegram untuk memberikan informasi secara *real-time* tentang status penghancuran kertas. Langkah-langkah ini

dirancang untuk menghadapi berbagai tantangan yang ada dan meningkatkan performa alat penghancur kertas otomatis.

3.2.2 Pengumpulan Data

Berdasarkan metode yang digunakan dalam melakukan proses penelitian untuk memperoleh data yang akurat serta dalam pembuatan alat agar mencapai hasil yang optimal, maka penulis menggunakan beberapa metode dalam pengumpulan data yang akan digunakan dalam perancangan dan pengembangan sistem tersebut sebagai berikut:

3.2.3 Metode Penelitian

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

1. Penelitian Perpustakaan (*library research*)

Penelitian ini dilakukan untuk melengkapi perbendaharaan konsep, teori dan lain-lain, sehingga menjadi suatu yang menjadi landasan dan keilmuan yang digunakan untuk menambah wawasan dan pemahaman yang mendukung dalam pembuatan sistem ini.

2. Penelitian Lapangan (*field research*)

Penelitian lapangan dilakukan dengan cara riset ke lapangan untuk memperoleh informasi yang diperlukan serta membandingkan dengan peralatan yang dapat mendukung kelancaran pembuatan sistem.

3. Penelitian Laboratorium (*laboratory research*)

Penelitian laboratorium dimaksudkan untuk menguji sistem yang dibuat. Metode ini dilakukan untuk menguji konsep yang ada dengan menggunakan peralatan yang sesuai. Adapun objek yang diuji spesifikasi *hardware* dan *software* yang digunakan dalam penulisan ini adalah sebagai berikut:

Tabel 3.1 *Hardware* dan *Software*

Hardware	Software
Satu Unit Laptop Lenovo Ideapad 3 Memory 8 GB Hard Disk SSD 512 GB Arduino Mega 2560 Solder Multimeter Kabel jumper Kabel Donwloader Arduino	Sistem Operasi Windows 11 64 bit Microsoft Office 2021 Arduino IDE Sketchup Fritzing 7.0 Draw.io Publish or Peris

3.2.3.1 Waktu Penelitian

Penelitian dilakukan dengan memproses data-data yang didapat oleh penulis, pengambilan data yang dilakukan dari bulan Oktober 2024 hingga Agustus 2025, yang dapat dilihat pada tabel 3.2.

Tabel 3.2 Jadwal Penelitian

No	Nama Kegiatan	2024								2025																			
		November				Desember				Januari				April				Mei				Juni				Juli			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Perencanaan Penelitian																												
2	Penelitian Pendahuluan																												
3	Studi Pustaka																												
4	Pengumpulan Data																												
5	Analisis Data																												
6	Perancangan Sistem																												
7	Pembuatan Sistem																												
8	Testing Sistem																												
9	Implementasi																												

3.2.3.2 Analisa Sistem

Berdasarkan identifikasi masalah di atas, peneliti melakukan analisa sistem terlebih dahulu. Hal ini bertujuan agar pemecahan masalah dapat menghasilkan sebuah solusi baru.

3.2.3.3 Perancangan Sistem

Perancangan dalam penelitian membuat media deteksi yang digunakan pada alat Penghancur kertas ini didasari beberapa pertimbangan tertentu. Perancangan alat ini harus memiliki kriteria:

- Alat Penghancur kertas ini diharapkan nantinya dapat membuat pekerjaan menjadi efisien dan hemat tenaga.
- Bahasa Pemrograman Arduino IDE diharapkan dapat digunakan untuk membuat program.
- Komponen standar yang digunakan mudah didapat.
- Komponen yang dirancang mudah dibuat, mudah dalam pemeliharaan serta mudah dalam perawatan

Ada beberapa *tools* yang digunakan dalam merancang sistem yang akan dibuat:

1. *Context Diagram*

Context diagram merupakan tingkat tertinggi dalam diagram aliran data yang hanya memuat satu proses sehingga menunjukkan sistem secara keseluruhan. Menggambarkan rancangan keseluruhan, dimana *entity external* harus digambarkan sedemikian rupa, sehingga terlihat data yang mengalir pada suatu *input - proses - output*.

2. *Data flow diagram (DFD)*

Data flow diagram (DFD) adalah diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sistem ke entitas yang bertujuan untuk memahami sistem secara logika, terstruktur dan jelas.

3. *Blok Diagram*

Blok Diagram dimaksudkan untuk memetakan proses kerja pada suatu alat. Hal ini bertujuan untuk memudahkan seseorang dalam mengenal komponen-komponen elektronika yang dipakai 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* membantu analis dan programmer untuk memecahkan masalah dalam bagian-bagian yang lebih kecil dan membantu dalam menganalisis alternatif lain dalam pengoperasian. *Flowchart* berperan penting dalam menentukan langkah-langkah atau fungsionalitas dari sebuah proyek dalam pembuatan program.

5. *Software* Desain

Desain dari alat yang dibuat berupa gambaran dan bentuk awal dari alat secara keseluruhan. Desain ini dibuat untuk menggambarkan letak komponen-komponen dari sistem yang digunakan agar dapat dilihat dengan jelas. Perangkat lunak yang digunakan untuk perancangan alat adalah *google sketchup*.

3.2.3.4 Pembuatan Alat

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*. Adapun tahap pembuatan sistem yang dilakukan adalah sebagai berikut.

1. Pada tahap awal rencanakan struktur dan desain keseluruhan, termasuk sensor LDR, motor DC, dan mekanisme kontrol manual dan otomatis.
2. Mempersiapkan komponen seperti motor, sensor-sensor, Arduino Mega 2560, dan Perangkat keras lainnya sesuai dengan kebutuhan desain.
3. Selanjutnya Pasang dan hubungkan komponen *hardware* ke papan sirkuit Arduino Mega 2560. Pastikan kabel dan koneksi fisik sesuai dengan desain yang telah direncanakan.
4. Membuat program menggunakan Arduino IDE untuk mengontrol fungsi-fungsi seperti deteksi kertas, pergerakan motor, dan transisi antara mode kontrol manual dan otomatis.
5. Setelah program selesai dibuat maka dilakukan proses *uploading* antara Arduino IDE dengan *board* Arduino. Setelah proses *uploading* selesai tahap selanjutnya adalah pengujian keseluruhan sistem alat.

3.2.3.5 Pengujian Sistem

Pengujian sistem bertujuan untuk melakukan pengujian terhadap sistem yang telah dirancang. Pengujian ini dilakukan agar dapat memahami apakah sistem yang dirancang mampu berjalan dengan baik dan sesuai dengan proses perancangannya. Adapun tahap pengujian sistem adalah sebagai berikut:

1. Pengujian Sensor LDR untuk mendeteksi kertas yang masuk ke dalam alat.
2. Pengujian modul ESP8266 untuk menghubungkan alat pada *SmartPhone* untuk kontrol manual.
3. Pengujian relay untuk menggerakkan motor DC.
4. Pengujian LCD 16x2 untuk menampilkan identitas dan informasi sistem yang bekerja.
5. Pengujian Buzzer dan LED sebagai output indikator pemberitahuan.

BAB IV

ANALISA DAN HASIL

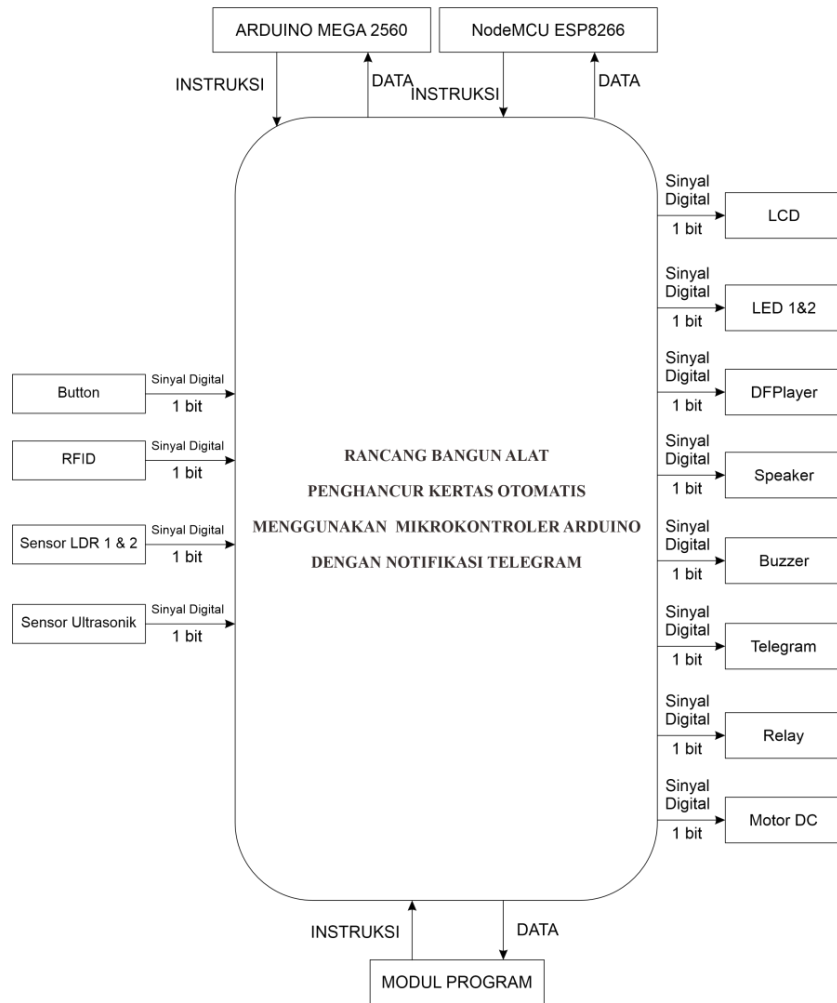
4.1 Desain Sistem Secara Umum

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 akan dapat dilihat dengan jelas.

4.1.1 *Context Diagram*

Sub bab ini merupakan penjabaran setiap *external entity* secara keseluruhan yang digambarkan melalui *context diagram*. *Context diagram* merupakan pendefenisian terhadap sistem yang akan dirancang yang bersifat menyeluruh. *Context diagram* ini digunakan untuk memudahkan dalam proses penganalisaan sistem yang dirancang secara keseluruhan.

Context diagram berfungsi sebagai media, yang terdiri dari suatu proses dan beberapa buah *external entity*. *Context diagram* yang dimaksud dapat dilihat pada Gambar 4.1.



Gambar 4.1 Context Diagram

Sesuai dengan penamaanya proses ini akan mengolah data *input* menjadi *output*. Proses ini akan berinteraksi dengan beberapa *entity*, yaitu :

1. Arduino Mega 2560

Arduino Mega 2560 berfungsi sebagai tempat pusat pengolahan seluruh data dan instruksi.

2. NodeMCU ESP8266

NodeMCU ESP8266 berfungsi untuk mengakses mikrokontroler melalui layanan internet agar dapat memberikan informasi kepada *smartphone* yang terhubung ke internet.

3. *Button*

Button berfungsi sebagai *trigger* untuk memulai mengaktifkan motor DC 1 dan LDR 1.

4. RFID

RFID berfungsi sebagai *inputan* untuk melakukan registrasi penggunaan sistem pada saat akan digunakan.

5. Sensor LDR 1 & 2

Sensor LDR 1 berfungsi sebagai pendeteksi kertas yang akan dimasukkan untuk dihancurkan, jika sensor LDR mendeteksi kertas, maka mesin penghancur kertas akan aktif. Sensor LDR 2 berfungsi sebagai pendeteksi kertas yang telah dihancurkan, jika sensor LDR mendeteksi, maka mesin penghancur akan berhenti.

6. Sensor Ultrasonik

Sensor Ultrasonik berfungsi sebagai pendeteksi isi box penampung kertas yang nantinya akan mengirimkan peringatan jika mendeteksi isi box penampung mulai penuh.

7. LCD

LCD berfungsi sebagai *output* yang akan menampilkan informasi seputar alat atau menampilkan hasil pendeteksian dari sensor.

8. LED 1 & 2

LED 1 & 2 berfungsi sebagai indikator alat dan indikator akses yang dibaca oleh RFID, jika terdata maka LED 2 akan aktif yang berwarna hijau, jika tidak maka LED 1 akan aktif yang berwarna merah.

9. *DFPlayer*

DFPlayer berfungsi sebagai pemroses audio dari *input* yang diterima dan *output* ke *speaker*.

10. *Speaker*

Speaker berfungsi sebagai *output* yang akan menampilkan hasil atau informasi berupa suara.

11. *Buzzer*

Buzzer berfungsi sebagai indikator jika akses yang diinputkan di tolak atau tidak terdaftar pada sistem.

12. Telegram

Telegram berfungsi sebagai sistem monitoring yang akan menerima pesan dari sistem melalui internet yang dikirim ke *smartphone*.

13. Relay

Relay berfungsi sebagai sakelar otomatis yang akan aktif pada saat sensor LDR mendeteksi adanya kertas yang dimasukkan untuk dihancurkan.

14. Motor DC 1 & 2

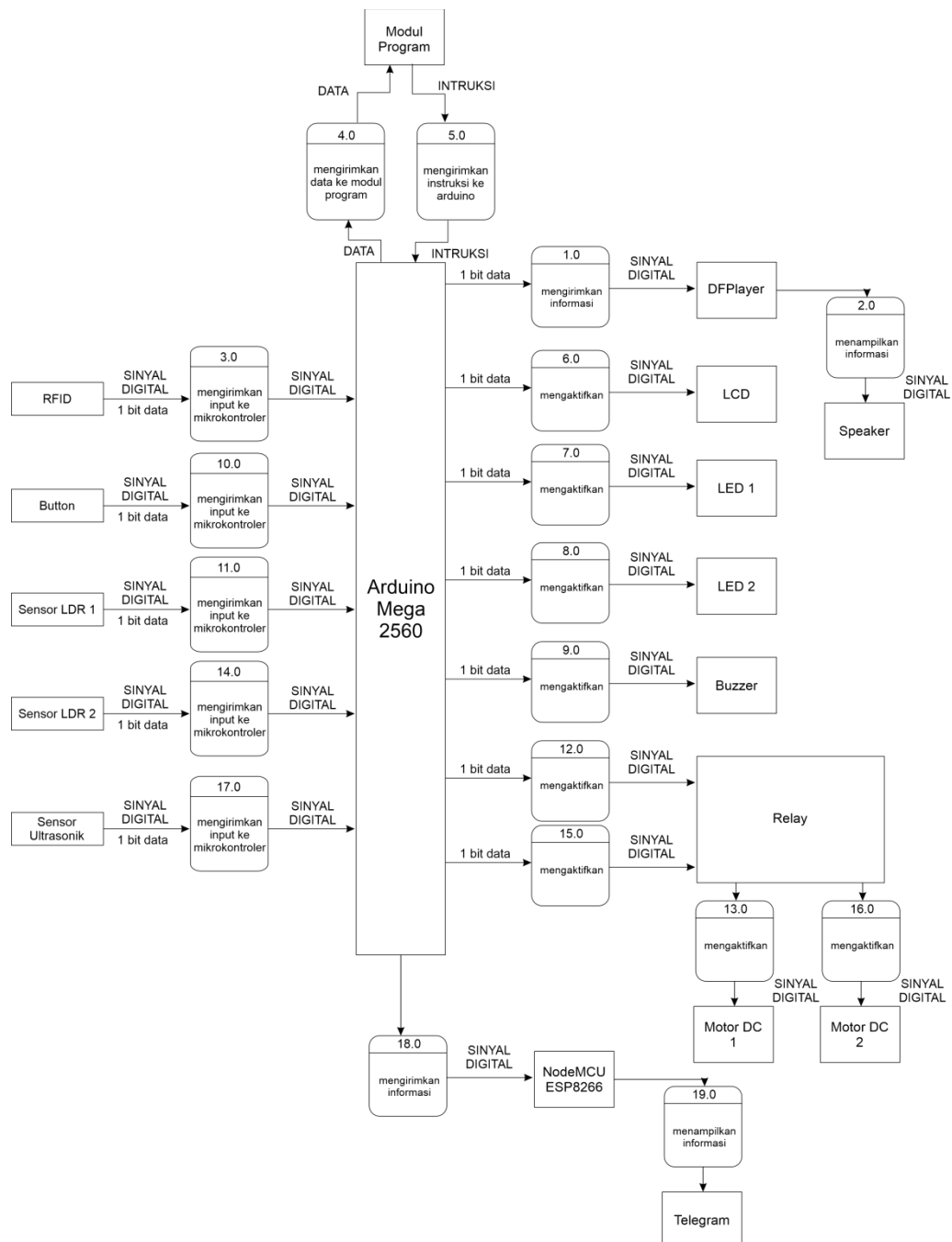
Motor DC 1 berfungsi sebagai penggerak mesin untuk menarik kertas dan motor DC 2 berfungsi sebagai penghancur kertas.

15. Modul Program

Melakukan pembacaan terhadap pin-pin Arduino, baik pembacaan terhadap sinyal-sinyal *input*, memberikan instruksi-instruksi untuk mengaktifkan pin-pin *output*. Modul program mengontrol semua proses yang akan terjadi pada sistem dan program yang digunakan adalah bahasa pemrograman Arduino.

4.1.2 Data Flow Diagram (DFD)

Data flow diagram adalah gambaran yang lebih rinci dari alat yang dirancang. *Data flow diagram* diuraikan berdasarkan *context diagram* yang telah dijabarkan sebelumnya. Gambar *data flow diagram* dapat dilihat pada gambar 4.2 berikut ini :



Gambar 4.2 Data Flow Diagram

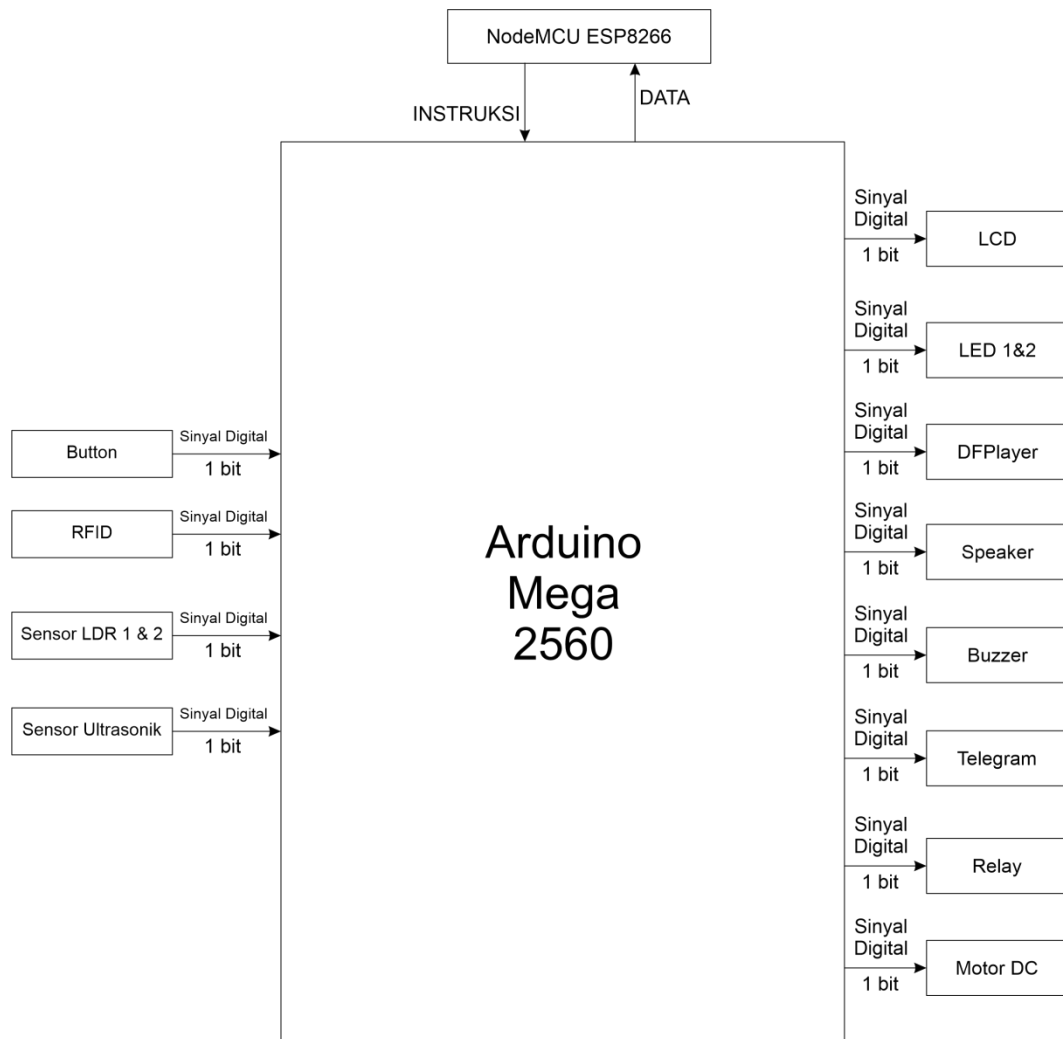
Dari gambar di atas dapat dijelaskan bahwa sistem kendali dari alat memiliki beberapa urutan proses instruksi yaitu :

1. Arduino Mega 2560 mengirimkan informasi ke DFPlayer pada saat sistem di aktifkan (1.0).
2. DFPlayer menampilkan informasi yang dikeluarkan oleh *speaker* melalui suara (2.0).
3. RFID mengirimkan *input* berupa data orang yang mengakses sistem penghancur kertas ke Arduino Mega 2560 (3.0).
4. Arduino Mega 2560 mengirimkan data ke modul program untuk di proses (4.0).
5. Modul program mengirimkan hasil proses ke Arduino Mega 2560 berupa perintah dari *input* menuju *output* (5.0).
6. Arduino Mega 2560 mengirimkan perintah untuk mengaktifkan LCD yang akan menampilkan informasi dan indikator sistem aktif (6.0).
7. Arduino Mega 2560 mengirimkan perintah untuk mengaktifkan LED 1 yang berwarna merah sebagai penanda sistem aktif dan akan tetap aktif jika RFID yang di *input* tidak terdaftar (7.0).
8. Arduino Mega 2560 mengirimkan perintah untuk mengaktifkan LED 2 yang berwarna hijau jika RFID yang di *input* terdaftar (8.0).
9. Arduino Mega 2560 mengirimkan perintah untuk mengaktifkan *buzzer* sebagai indikator jika RFID yang di *input* terdaftar (9.0).
10. *Button* mengirimkan input berupa perintah trigger ke Arduino Mega 2560 untuk mengaktifkan LDR (10.0).

11. Sensor LDR 1 mengirimkan *input* berupa pembacaan objek berupa kertas yang dimasukkan ke dalam mesin (11.0).
12. Arduino Mega 2560 mengirimkan perintah untuk mengaktifkan relay jika sensor LDR mendeteksi ada kertas yang dimasukkan (12.0).
13. Relay mengirimkan perintah untuk mengaktifkan motor DC yang akan menarik kertas (13.0).
14. Sensor LDR 2 mengirimkan *input* berupa pembacaan objek berupa kertas yang telah dimasukkan ke dalam mesin (14.0).
15. Arduino Mega 2560 mengirimkan perintah untuk mengaktifkan relay jika sensor LDR mendeteksi ada kertas yang dimasukkan (15.0).
16. Relay mengirimkan perintah untuk mengaktifkan motor DC yang akan menghancurkan kertas (16.0).
17. Sensor Ultrasonik mengirimkan *input* berupa pembacaan objek berupa kertas yang sudah dihancurkan dan berada didalam mesin penghancur (17.0).
18. Arduino Mega 2560 mengirimkan informasi yang diinput ke NodeMCU ESP8266 (18.0).
19. NodeMCU ESP8266 mengirimkan informasi yang diterima ke Telegram yang berfungsi sebagai software monitoring alat (19.0).

4.1.3 Blok Diagram

Dengan mengacu pada *Data Flow Diagram* di atas, untuk mengetahui komponen-komponen sistem ini dapat dilihat dalam blok diagram pada Gambar 4.3.



Gambar 4.3 Blok Diagram

Dari blok diagram diatas dapat dilihat bahwa sistem terdiri dari beberapa *input*, yaitu *buton*, RFID, sensor LDR dan sensor Ultrasonik. Pemroses atau pengendali dari sistem yang dirancang menggunakan mikrokontroler Arduino Mega 2560 dan NodeMCU ESP8266. Sistem yang dirancang memiliki *output*, yaitu LCD, LED, DFplayer, Speaker, buzzer, telegram, motor DC dan relay.

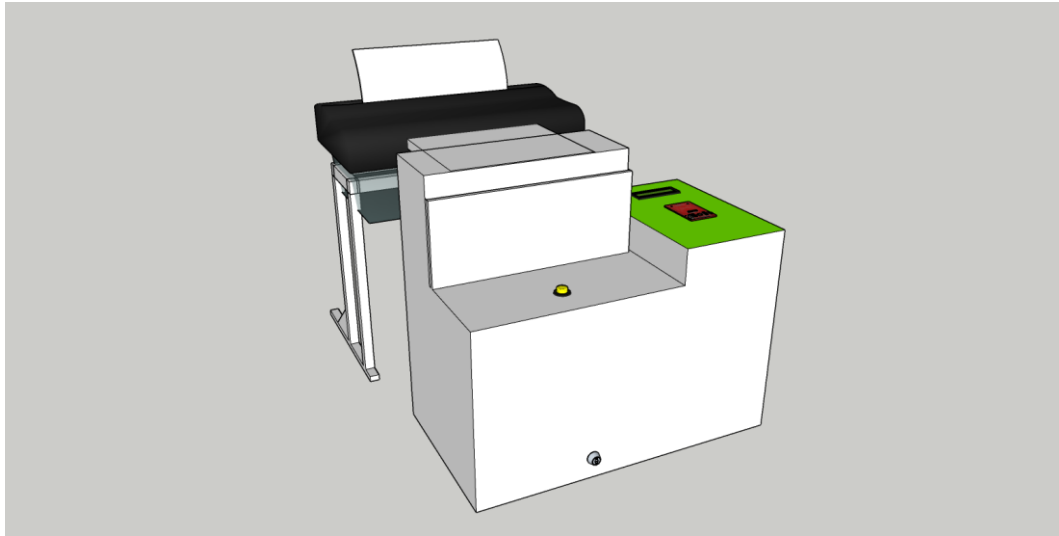
4.2 Prinsip Kerja Sistem

Secara umum bentuk dari alat ini terdiri dari rangkaian elektronika beserta sistem yang akan bekerja secara terstruktur tergantung rancangan yang dibuat. Berikut adalah cara kerja sistem secara terstruktur :

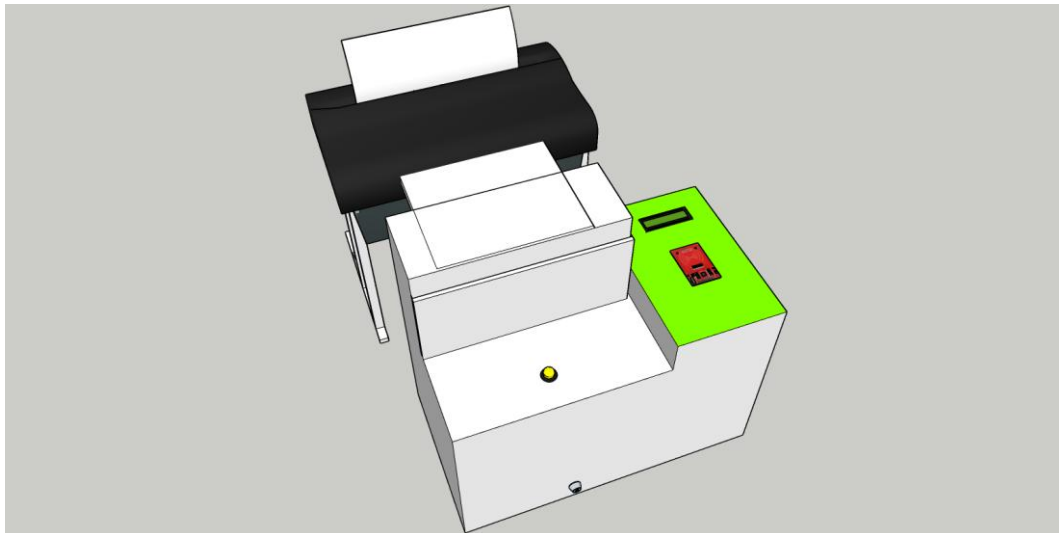
1. Hidupkan alat dengan mencolokan adaptor
2. Lalu led LCD aktif, lalu dfplayer mengeluarkan suara selamat datang di rancang bangun alat penghancur kertas otomatis menggunakan mikrokontroler Arduino
3. Tempel RFID, jika RFID benar maka akan keluar LCD dan suara dfplayer kartu anda benar, jika salah pun sebaliknya
4. Jika RFID benar maka akan keluar suara silahkan tekan button kuning
5. Lalu tekan button kuning, maka LDR 1 akan mendeteksi kertas dan mengaktifkan motor DC 1 untuk menarik kertas
6. Setelah kertas ditarik dan sampai pada ldr 2, maka ldr 2 otomatis mendeteksi kertas dan memberhentikan motor DC 1 , dan mengaktifkan motor DC 2 untuk memotong kertas
7. Jika pemotongan sudah selesai akan muncul tampilan pada LCD dan telegram bahwa pemotongan sudah selesai
8. Dan jika sensor ultrasonik mendeteksi tempat penampungan sudah penuh maka LCD dan telegram menampilkan tempat sudah penuh

4.3 Rancangan Fisik Alat

Perancangan alat ini merupakan tahap awal dari pemasangan dan menganalisa permasalahan yang dihadapi berdasarkan literatur yang menunjang perancangan alat, rancangan fisik dari alat digambarkan menggunakan *software google sketchup*. Rancangan fisik alat yang dibuat dapat dilihat pada gambar 4.4 dan 4.5 berikut.



Gambar 4.4 Rancangan Alat Fisik Tampak Depan



Gambar 4.5 Rancangan Alat Fisik Tampak Atas

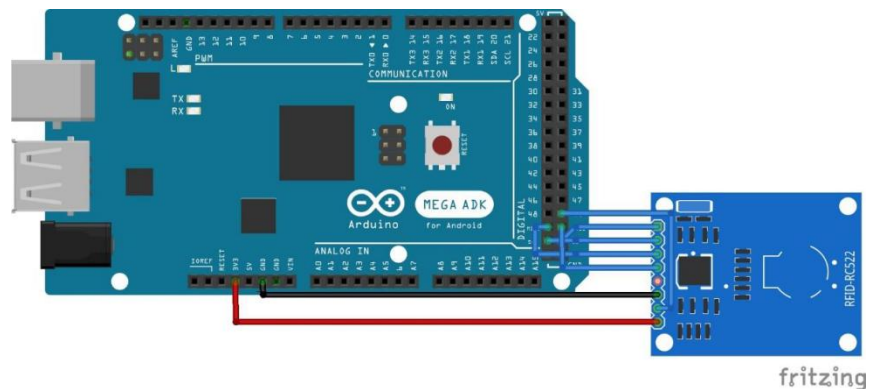
Dari gambaran rancangan diatas, terlihat bahwa sistem yang akan dibuat telah dirancang sedemikian rupa agar sistem dapat bekerja sebagaimana seharusnya sesuai dengan rancangan awal sistem tanpa menyisakan *entity-entity* penyusun sistem, yaitu Arduino Mega 2560 dan NodeMCU ESP8266 merupakan mikrokontroler pengontrol dari sistem yang terlihat pada gambar diatas, button, RFID, sensor LDR dan sensor ultrasonik sebagai *input*. LCD, LED, DFplayer, Speaker, buzzer, telegram, motor DC dan relay sebagai *output*.

4.4 Desain Secara Terinci

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

4.4.1 Rangkaian RFID

RFID berfungsi sebagai *inputan* untuk melakukan registrasi penggunaan sistem pada saat akan digunakan. Rangkaian Barcode QR Scanner dapat dilihat pada gambar 4.6.



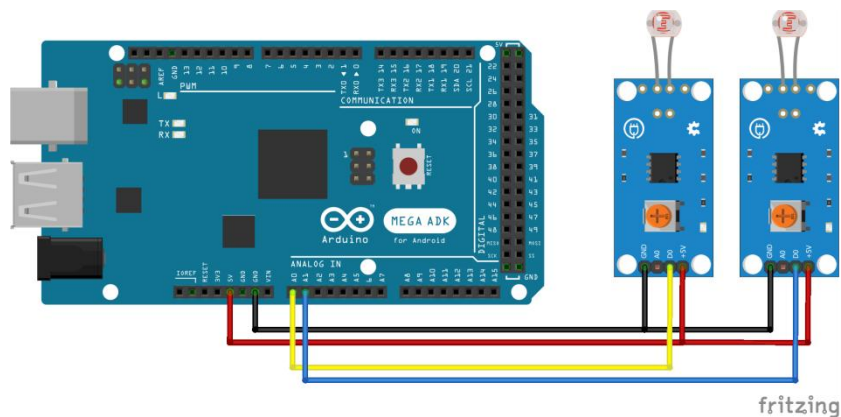
Gambar 4.6 Rangkaian RFID

Penjelasan pin yang digunakan RFID yaitu

1. Pin VCC pada RFID terhubung ke pin 3.3V pada Arduino Mega 2560.
2. Pin GND pada RFID terhubung ke pin GND pada Arduino Mega 2560.
3. Pin MISO pada RFID terhubung ke pin 51 pada Arduino Mega 2560.
4. Pin MOSI pada RFID terhubung ke pin 50 pada Arduino Mega 2560.
5. Pin SCK pada RFID terhubung ke pin 52 pada Arduino Mega 2560.
6. Pin SDA pada RFID terhubung ke pin 53 pada Arduino Mega 2560.
7. Pin RST pada RFID terhubung ke pin 49 pada Arduino Mega 2560.

4.4.2 Rangkaian Sensor LDR

Sensor LDR berfungsi sebagai pendeteksi kertas yang akan dimasukkan untuk dihancurkan, jika sensor LDR mendeteksi kertas, maka mesin penghancur kertas akan aktif. Rangkaian Sensor LDR dapat dilihat pada gambar 4.7.



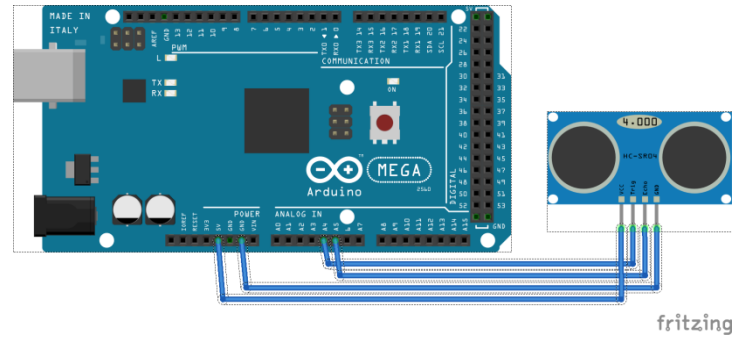
Gambar 4.7 Rangkaian Sensor LDR

Gambar 4.7 Penjelasan pin yang digunakan LDR yaitu:

1. Pin GND pada LDR 1 & 2 terhubung dengan pin GND pada Arduino Mega 2560.
2. Pin DO pada LDR 1 terhubung dengan pin D0 pada Arduino Mega 2560..
3. Pin DO pada LDR 2 terhubung dengan pin D1 pada Arduino Mega 2560.
4. Pin 5V pada LDR 1 & 2 terhubung dengan pin 5V pada Arduino Mega 2560.

4.4.3 Rangkaian Sensor Ultrasonik

Sensor Ultrasonik berfungsi sebagai pendeteksi isi box penampung kertas yang nantinya akan mengirimkan peringatan jika mendeteksi isi box penampung mulai penuh. Rangkaian sensor ultrasonik dapat dilihat pada gambar 4.8.



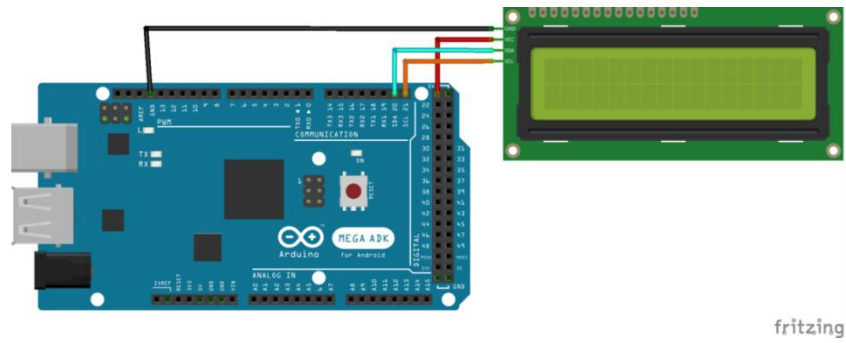
Gambar 4.8 Rangkaian Sensor Ultrasonik

Gambar 4.8 Penjelasan pin yang digunakan Sensor Ultrasonik yaitu:

1. Pin GND pada sensor ultrasonik terhubung dengan pin GND pada Arduino Mega 2560.
2. Pin VCC pada sensor ultrasonik terhubung dengan pin 5V pada Arduino Mega 2560.
3. Pin Echo pada sensor ultrasonik terhubung dengan pin A4 pada Arduino Mega 2560.
4. Pin Trigger pada sensor ultrasonik terhubung dengan pin A5 pada Arduino Mega 2560.

4.4.4 Rangkaian LCD

LCD berfungsi sebagai *output* yang akan menampilkan informasi seputar alat atau menampilkan hasil pendeteksian dari sensor. Rangkaian LCD dapat dilihat pada gambar 4.9.



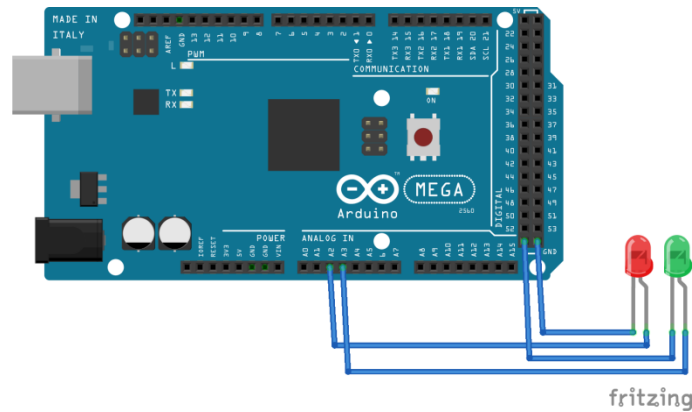
Gambar 4.9 Rangkaian LCD

Pada rangkaian di atas dapat dilihat susunan pin pada rangkaian LCD adalah sebagai berikut :

1. Pin GND pada LCD terhubung dengan pin GND pada Arduino Mega 2560.
2. Pin VCC pada LCD terhubung dengan pin 5V pada Arduino Mega 2560.
3. Pin SDA pada LCD terhubung dengan pin SDA pada Arduino Mega 2560
4. Pin SCL pada LCD terhubung dengan pin SCL pada Arduino Mega 2560.

4.4.5 Rangkaian LED 1 & 2

LED 1 & 2 berfungsi sebagai indikator alat dan indikator akses yang dibaca oleh RFID, jika terdata maka LED 2 akan aktif yang berwarna hijau, jika tidak maka LED 1 akan aktif yang berwarna merah. Rangkaian LED 1 & 2 dapat dilihat pada gambar 4.10.



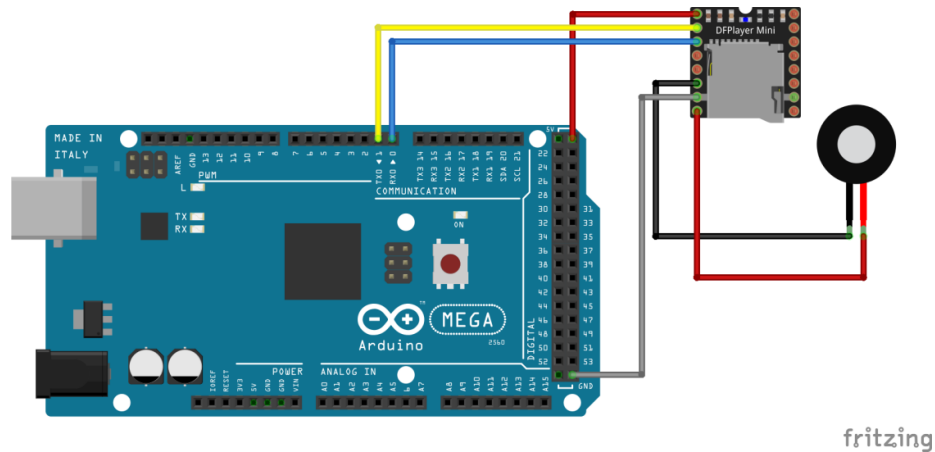
Gambar 4.10 Rangkaian LED 1 & 2

Pada rangkaian di atas dapat dilihat susunan pin pada rangkaian LED 1 & 2 adalah sebagai berikut :

1. Pin anoda pada LED 1 terhubung dengan pin GND pada arduino mega.
2. Pin katoda pada LED 1 terhubung dengan pin A2 pada arduino mega.
3. Pin anoda pada LED 2 terhubung dengan pin GND pada arduino mega.
4. Pin katoda pada LED 2 terhubung dengan pin A3 pada arduino mega.

4.4.6 Rangkaian DFPlayer & Speaker

DFPlayer berfungsi sebagai pemroses audio dari *input* yang diterima dan *output* ke *speaker*. *Speaker* berfungsi sebagai *output* yang akan menampilkan hasil atau informasi berupa suara. Rangkaian DFPlayer & Speaker dapat dilihat pada gambar 4.11.



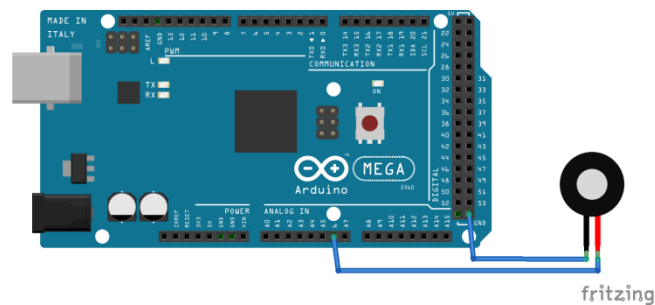
Gambar 4.11 Rangkaian DFPlayer & Speaker

Pada rangkaian diatas dapat dilihat susunan pin pada rangkaian DFPlayer & Speaker adalah sebagai berikut :

1. Pin VCC pada sensor *DFPlayer* terhubung dengan pin 5v pada arduino mega.
2. Pin RX pada *DFPlayer* terhubung dengan pin TX pada arduino mega.
3. Pin TX pada *DFPlayer* terhubung dengan pin RX pada arduino mega.
4. Pin SPK1 pada *DFPlayer* terhubung dengan pin 1 pada *speaker*.
5. Pin GND pada *DFPlayer* terhubung dengan pin GND pada arduino mega.
6. Pin SPK2 pada *DFPlayer* terhubung dengan pin 2 pada *speaker*.

4.4.7 Rangkaian Buzzer

Buzzer berfungsi sebagai indikator jika akses yang diinputkan di tolak atau tidak terdaftar pada sistem. Rangkaian Buzzer dapat dilihat pada gambar 4.12.



Gambar 4.12 Rangkaian Buzzer

Pada rangkaian diatas dapat dilihat susunan pin pada rangkaian Buzzer adalah sebagai berikut :

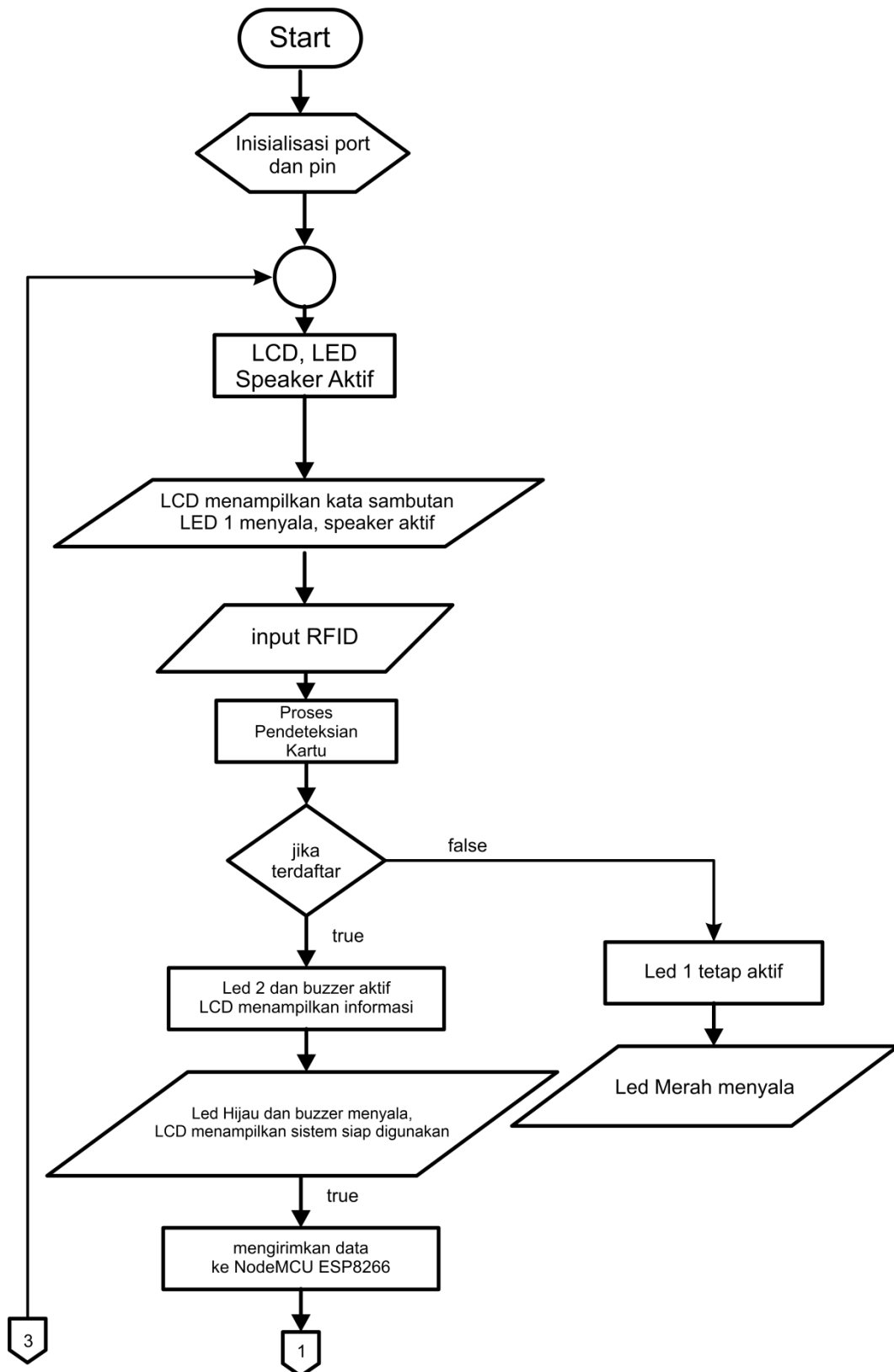
1. Pin 1 pada *buzzer* terhubung dengan pin GND pada Arduino Mega 2560.
2. Pin 2 pada *buzzer* terhubung dengan pin A6 pada Arduino Mega 2560.

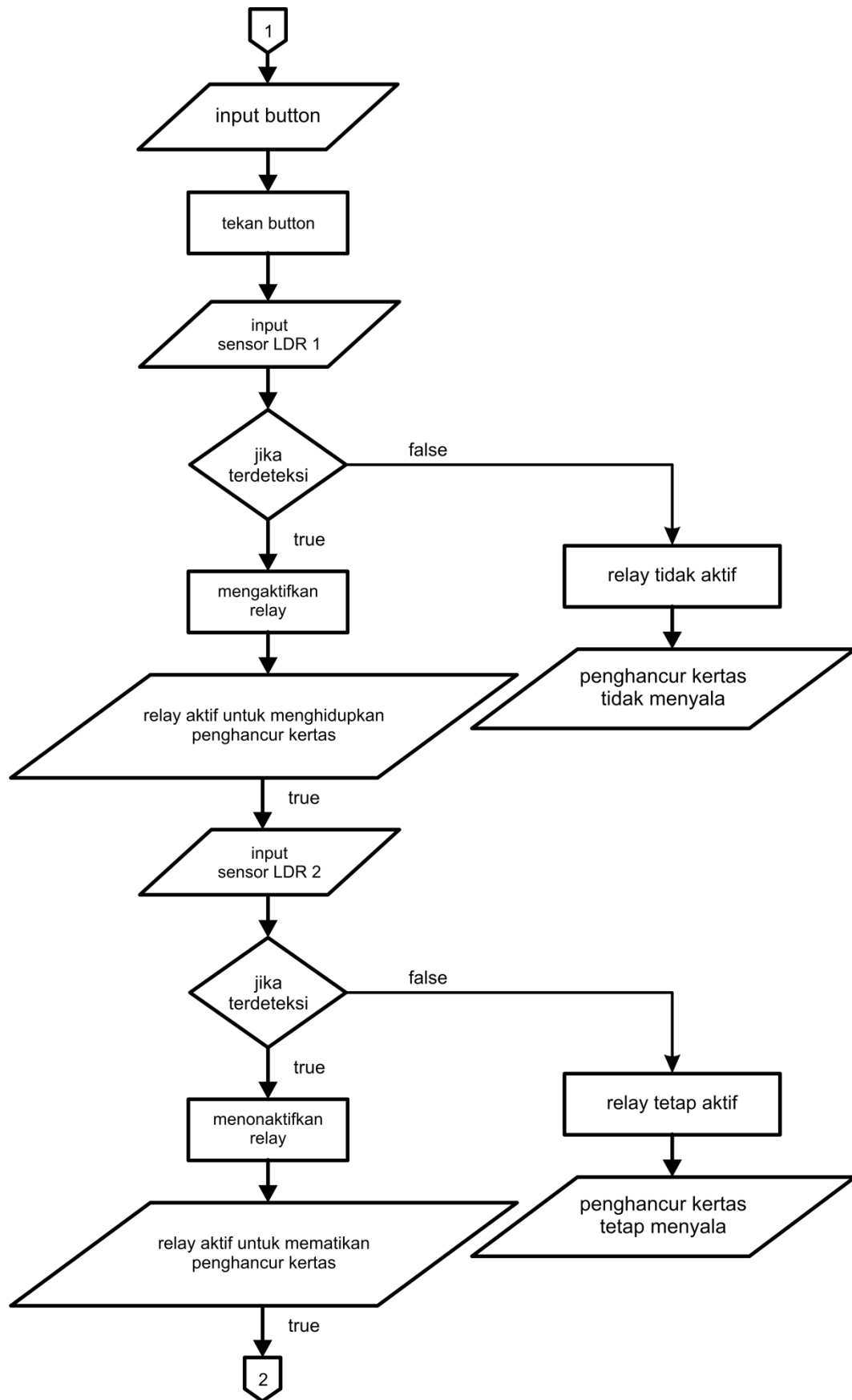
4.5 Rancangan Modul Program

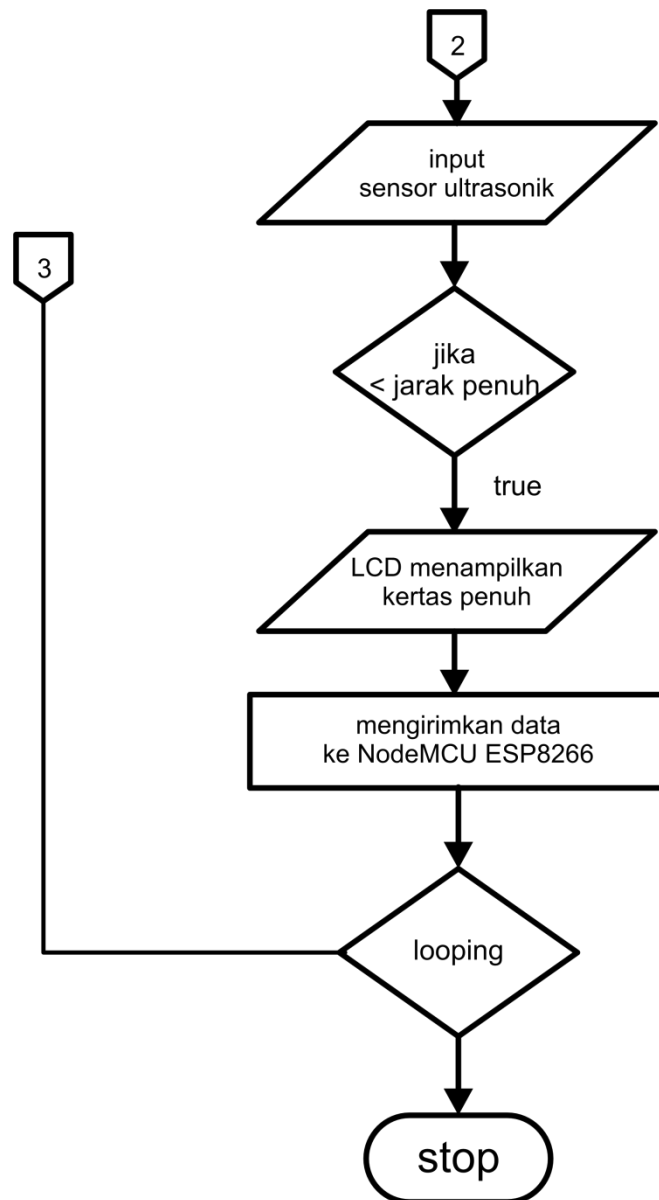
Sub bab ini menjelaskan tentang Modul Program yang digunakan untuk mengontrol kerja dari sistem yang dirancang. Untuk lebih mudah dimengerti rancangan modul dapat menjadi dua bagian yaitu *flowchart* dan *listing Program*.

4.5.1 Flowchart

Modul program dirancang memiliki struktur dengan kualitas yang baik dan mudah dimengerti, maka sebelum pembuatan listing program perlu diawali dengan penentuan logika program.







Gambar 4.13 *Flowchart* Arduino Mega 2560

4.5.2 Listing Program

Pada *subbab* ini diuraikan mengenai modul program untuk menunjang kemampuan sistem dengan menggunakan bahasa pemrograman C.

Arduino Mega

```
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>
#include <DFPlayer_Mini_Mp3.h>
#include <SoftwareSerial.h>
#include <Buzzer.h>

#define ldr      A0
#define ldr2     A7
#define relay1   A1
#define relay2    8
#define led1     A2
#define led2     A3
#define trig_pin A4
#define echo_pin A5
#define BUZZER_PIN A6

Buzzer buzzer(BUZZER_PIN,13); // create ezBuzzer object that attach to a pin;

#define RST_PIN    5      // Configurable, see typical pin layout above
#define SS_PIN     53     // Configurable, see typical pin layout above

SoftwareSerial mySerial(11, 10); // RX, TX
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
LiquidCrystal_I2C lcd(0x27, 16, 2);

String strUID;
String UidSet = "1952472320";
bool izin_masuk = false;
unsigned long prevTime = millis();
int jarak_penuh = 8;
int tampil = 0;
bool tampilkan_1x = false;
bool stt_alat_bekerja = false;
bool detek_kertas = false;
bool potong_berakhir = false;

void setup() {
  Serial.begin(9600);
  Serial2.begin(9600);
  mySerial.begin (9600);
  pinMode(ldr,INPUT);
```

```

pinMode(ldr2,INPUT_PULLUP);
pinMode(led1,OUTPUT);
pinMode(led2,OUTPUT);
pinMode(relay1,OUTPUT);
pinMode(relay2,OUTPUT);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,HIGH);
pinMode(trig_pin, OUTPUT);
pinMode(echo_pin, INPUT);

while (!Serial);          // Do nothing if no serial port is opened (added for
                          // Arduinos based on ATMEGA32U4)
mp3_set_serial (mySerial); //set softwareSerial for DFPlayer-mini mp3 module
mp3_set_volume (30);

lcd.begin();
lcd.backlight();
SPI.begin();              // Init SPI bus
mfrc522.PCD_Init();        // Init MFRC522
delay(4);                  // Optional delay. Some board do need
                          // more time after init to be ready, see Readme
mfrc522.PCD_DumpVersionToSerial(); // Show details of PCD -
MFRC522 Card Reader details
digitalWrite(led1,HIGH);
digitalWrite(led2,LOW);
lcd2("DIMAS PRASETYO","2110115260061");
mp3_play(1);
delay(7000);
mp3_play(2);
lcd2("Tempel Kartu","untuk memulai");
}

void loop() {

int ultra = ultra_sonic();
if (ultra < jarak_penuh ){
  lcd2("Kertas Sudah","Penuh");
  tampil=0;
  return;
} else {
  if (izin_masuk) {
    if (tampil++==1) lcd2("ALAT SIAP","DIGUNAKAN");
  }
  else {
    if (tampil++==1) lcd2("Tempel Kartu","untuk memulai");
  }
}
}

```

```

if (millis() - prevTime > 5000) {
  prevTime = millis();
  if (izin_masuk) {
    if (stt_alat_bekerja && !detek_kertas) {
      delay(2000);
      digitalWrite(relay1,HIGH);
      potong_berakhir=true;
    }
  }
}

```

```

if (izin_masuk) {
  if (ldr2_value()){
    while (!ldr1_value()) {
      digitalWrite(relay2,LOW);
    }
  } else {
    if (stt_alat_bekerja){
      Serial2.println("end");
      lcd2("PENGHANCURAN","SELESAI");
      stt_alat_bekerja = false;
    }
  }
  delay(700);
  digitalWrite(relay2,HIGH);
  if (ldr1_value()){
    if (!tampilkan_1x){
      tampilkan_1x = true;
      detek_kertas = true;
      stt_alat_bekerja = true;
      Serial2.println("kerja");
      lcd2("ALAT SEDANG","BEKERJA");
      digitalWrite(relay1,LOW);
      digitalWrite(relay2,HIGH);
    }
  } else {
    potong_berakhir=false;
    detek_kertas = false;
    tampilkan_1x = false;
  }
}

```

```

delay(100);
if ( ! mfrc522.PICC_IsNewCardPresent()) return;
if ( ! mfrc522.PICC_ReadCardSerial()) return;
MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
strUID = String(mfrc522.uid.uidByte[0]) + String(mfrc522.uid.uidByte[1]) +

```

```

        String(mfrc522.uid.uidByte[2]) + String(mfrc522.uid.uidByte[3]);

        Serial.println(strUID);

        buzzer_rfid();
        if (strUID == UidSet) {
            Serial2.println("ok");
            mp3_play(3);
            lcd2("ALAT SIAP", "DIGUNAKAN");
            izin_masuk = true;
            digitalWrite(led1, LOW);
            digitalWrite(led2, HIGH);
            Serial.println(strUID);
        } else {
            Serial2.println("er");
            mp3_play(4);
            izin_masuk = false;
            digitalWrite(led1, HIGH);
            digitalWrite(led2, LOW);
        }
    }

    int ultra_sonic(){
        long echotime;
        float distance;
        digitalWrite(trig_pin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trig_pin, LOW);
        echotime= pulseIn(echo_pin, HIGH);
        distance= 0.0001*((float)echotime*340.0)/2.0;
        return distance;
    }

    void buzzer_rfid(){
        buzzer.begin(100);
        buzzer.sound(NOTE_E7, 80);
        buzzer.sound(NOTE_C7, 80);
        buzzer.sound(NOTE_B6, 80);
        buzzer.end(100);
    }

    void lcd2(String text1, String text2) {
        lcd.clear();
        lcd.setCursor((16-text1.length())/2,0);
        lcd.print(text1);
        lcd.setCursor((16-text2.length())/2,1);
        lcd.print(text2);
    }

```

```

bool ldr1_value(){
  if (analogRead(ldr)>30) return true;else return false;
}

bool ldr2_value(){
  bool ret_ = false;
  if (analogRead(ldr2)>35 && analogRead(ldr2)<800 ) ret_=true;
  return (ret_);
}

```

NodeMCU ESP8266

```

#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>

const char* ssid = "qwerty";
const char* password = "12345678";

#define BOTtoken "8175096019:AAGc71JFZrGwzeKbShjdTufPY2g5ZSpg4Uk"
// your Bot Token (Get from Botfather)
#define CHAT_ID "6323903621"

#ifdef 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;
String msg_last="";

void setup() {
  Serial.begin(9600);

#ifdef ESP8266
  configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP

```

```

    client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
#endif

// 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.sendMessage(CHAT_ID, "Bot Started", "");
}

void loop() {

    if (Serial.available()){
        String msg = "";
        while(Serial.available()){
            msg+=char(Serial.read());
            delay(50);
        }
        msg.trim();
        if (msg=="ok") {
            bot.sendMessage(CHAT_ID, "KARTU ANDA COCOK", "");
        } else
        if (msg=="er") {
            bot.sendMessage(CHAT_ID, "KARTU ANDA TIDAK COCOK", "");
        } else
        if (msg=="kerja") {
            bot.sendMessage(CHAT_ID, "MESIN SEDANG BEKERJA", "");
        } else
        if (msg=="end") {
            bot.sendMessage(CHAT_ID, "PEMOTONGAN SELESAI", "");
        }
    }
}
}

```

BAB V

PENGUJIAN SISTEM

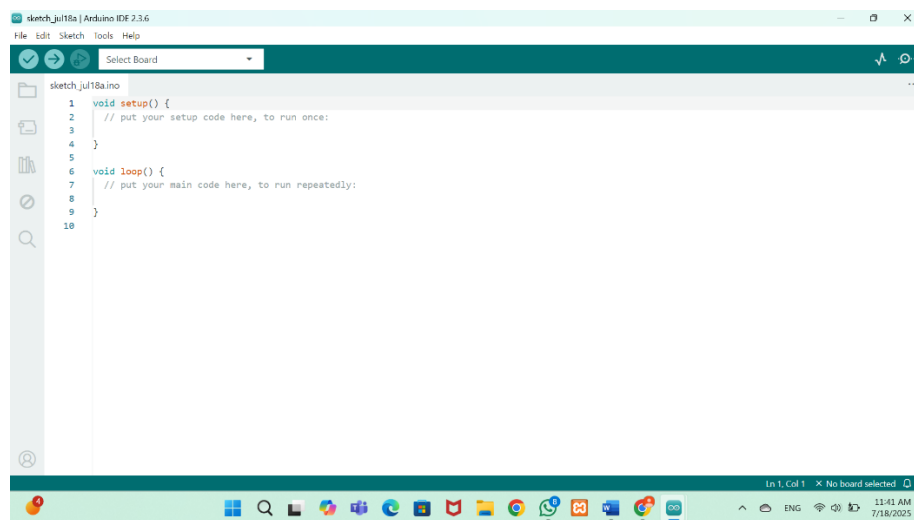
5.1 Pengujian Sistem Permodul

Adapun langkah-langkah yang dapat dilakukan dalam menjalankan sistem alat permodul adalah sebagai berikut:

5.1.1 Pengujian Rangkaian Sistem Minimum

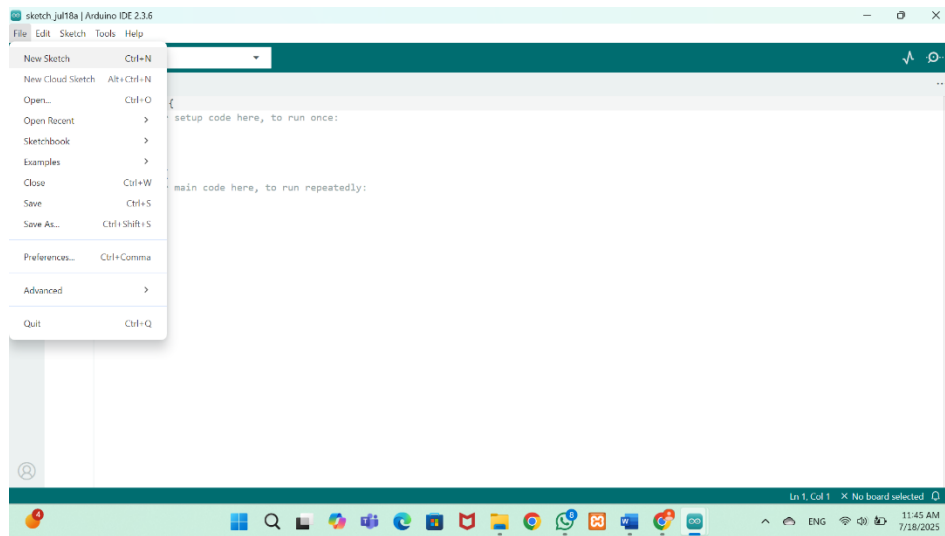
Pengujian sistem minimum dilakukan dengan menguji rangkaian sistem minimum setelah proses uploading telah dilakukan oleh pemrograman Arduino IDE. Terdapat langkah-langkah yang harus dilalui dalam pemrograman arduino adalah sebagai berikut :

1. Jalankan Aplikasi Arduino dengan *double click* pada *icon* Arduino, sehingga muncul tampilan pada gambar 5.1



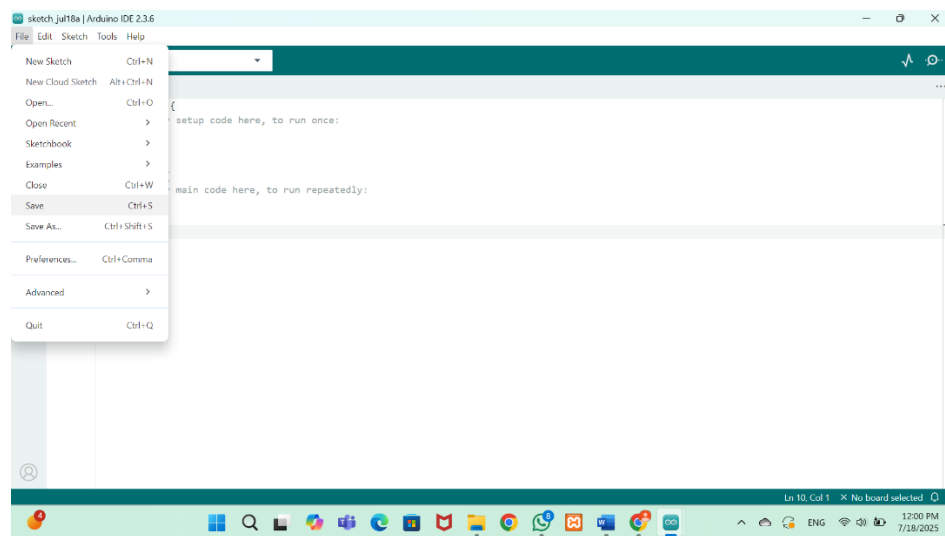
Gambar 5.1 Tampilan Arduino IDE

2. Kemudian buat Program baru dengan memilih menu *file > new* sehingga muncul tampilan seperti gambar 5.2 dan dapat membuat program sesuai dengan yang diperlukan



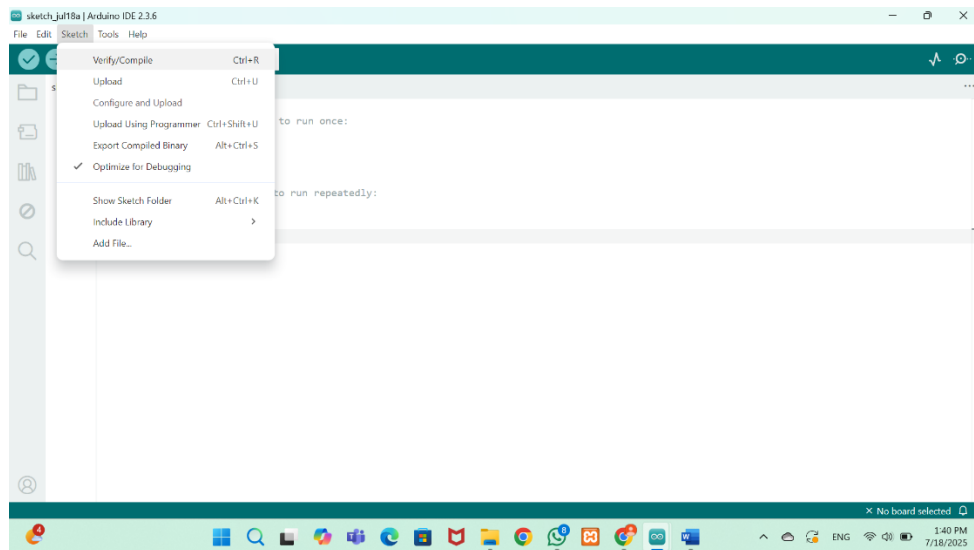
Gambar 5.2 Tampilan new edit program

3. Ketik kan program, kemudian simpan pada folder yang diinginkan dengan memilih menu *file > save* seperti pada gambar 5.3.



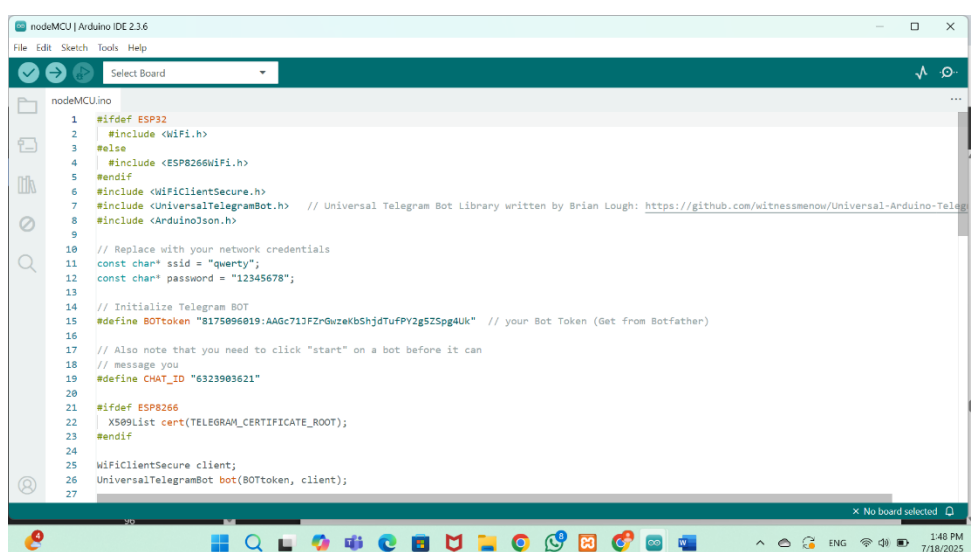
Gambar 5.3 Tampilan save program

4. Setelah itu compile program yang telah dibuat dengan memilih menu *sketch > verify / compile* seperti gambar 5.4



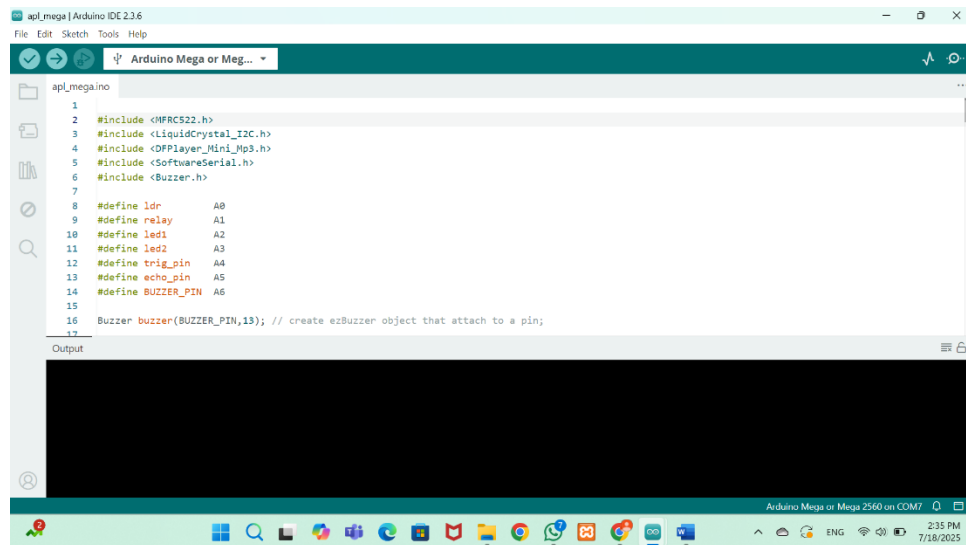
Gambar 5.4 Tampilan *compile* pada Arduino IDE

5. Langkah selanjutnya meng-*upload file* program yang telah dibuat dan di kompilasi tersebut ke dalam *board* Arduino digunakan sebagai control dari sistem yang dipakai.
6. Pertama pasanglah kabel USB printer ke Arduino dan ke komputer atau laptop yang digunakan.
7. Kemudian pilih menu *sketch > upload* (CTRL + U untuk cara lebih cepat) seperti gambar 5.5.



Gambar 5.5 Tampilan *upload* pada Arduino IDE

8. Jika program sukses di upload ke dalam Arduino, maka akan muncul tulisan “*Done Uploading*”, seperti gambar 5.6.



Gambar 5.6 Tampilan Upload selesai

5.1.2 Pengujian Rangkaian RFID

Pengujian RFID berfungsi sebagai pendeteksi kartu yang telah dibuat dan didaftarkan peneliti, data hasil pembacaan sensor dikirim ke mikrokontroler kemudian mikrokontroler akan menggunakan data tersebut untuk mengaktifkan atau tidaknya alat, dari pengujian sensor RFID dapat dilihat pada tabel 5.1

Tabel 5.1 Pengujian Rangkaian RFID

No	Sensor RFID	LCD	Speaker	LED	Buzzer
1	Jika sensor RFID membaca kartu terdaftar, maka	Maka LCD aktif menampilkan “Alat Siap Digunakan”	Aktif	Hijau Aktif	Aktif
2	Jika sensor RFID membaca kartu tidak terdaftar	Maka LCD aktif	Aktif	Merah Aktif	Aktif

5.1.3 Pengujian sensor LDR

Sensor LDR berfungsi sebagai pendeteksi kertas yang masuk, apabila sensor mendeteksi kertas yang masuk maka motor akan aktif dan mulai memotong kertas dan LCD akan menampilkan bahwa alat sedang bekerja. Sebaliknya jika sensor LDR tidak mendeteksi kertas maka motor tidak akan aktif untuk memotong kertas dan lcd hanya menampilkan alat siap digunakan.

Tabel 5.2 Pengujian Sensor LDR

No	Sensor LDR	LCD
1	Jika sensor LDR mendeteksi kertas yang masuk	Maka LCD menampilkan “Alat Sedang Bekerja”
2	Jika sensor LDR tidak mendeteksi kertas yang masuk	Maka LCD hanya menampilkan “Alat Siap Digunakan”

5.1.3 Pengujian Sensor Ultrasonik

Sensor Ultrasonik berfungsi sebagai pendeteksi tempat penampungan kertas yang telah dipotong. LCD mengeluarkan output berupa pemberitahuan kertas yang berada didalam kotak penampungan penuh.

Tabel 5.3 Pengujian sensor Ultrasonik

No	Sensor Ultrasonik	LCD
1.	Jika sensor Ultrasonik telah mendeteksi tempat penampungan sudah penuh	Maka lcd menampilkan kertas sudah penuh
2.	Jika sensor Ultrasonik tidak mendeteksi tempat penampungan penuh	Maka lcd hanya akan menampilkan tentang status alat

5.2 Pengujian sistem keseluruhan

Pengujian alat secara keseluruhan dapat dilakukan setelah pengujian sistem permodul diselesaikan. Dengan demikian, pengujian sistem keseluruhan dapat dijalankan melalui langkah- langkah berikut ini :

1. Sistem dalam kondisi tidak aktif, jika kabel *power* tidak dihubungkan ke sumber tegangan arus listrik, dapat dilihat pada gambar 5.7.



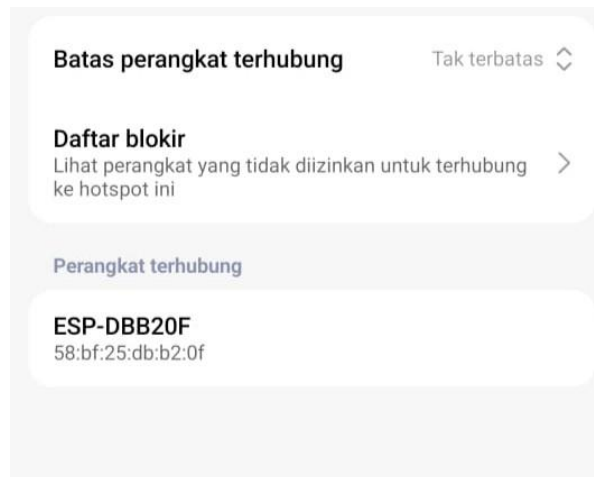
Gambar 5.7 Sistem Alat Tidak Aktif

2. Sistem aktif setelah kabel *power* dihubungkan ke sumber tegangan arus listrik dan LCD akan menampilkan nama dan nomor bp dari pembuat alat. Dapat dilihat pada gambar 5.8.



Gambar 5.8 Sistem Alat Aktif

3. Aktifkan hotspot *portable* pada *smartphone* android untuk menghubungkan NodeMCU ESP8266 ke koneksi jaringan internet dapat dilihat pada gambar 5.9.



Gambar 5.9 Mengaktifkan Hotspot Portabel

4. Jika sensor RFID melakukan scan kartu yang salah maka LED merah akan aktif dan buzzer akan berbunyi, dapat dilihat pada gambar 5.10.



Gambar 5.10 Scan Kartu RFID

5. Jika sensor RFID melakukan scan kartu yang benar maka LCD akan menampilkan alat siap digunakan dan LED hijau menyala dan buzzer berbunyi, dapat dilihat pada gambar 5.11.



Gambar 5.11 Scan Kartu Terdaftar

6. Jika alat sudah aktif dan kartu RFID sudah benar, maka akan muncul perintah suara untuk menekan button untuk memulai sistem untuk menghancurkan kertas, dapat dilihat pada gambar 5.12.



Gambar 5.12 Menekan *Button*

7. Ketika *button* sudah ditekan maka semua sistem aktif dan mulai memotong kertas, mulai dari mengaktifkan motor DC 1 dan LDR 1 untuk menarik kertas dan motor DC 2 dan LDR 2 untuk memotong kertas, alat sedang bekerja dapat dilihat pada gambar 5.13.



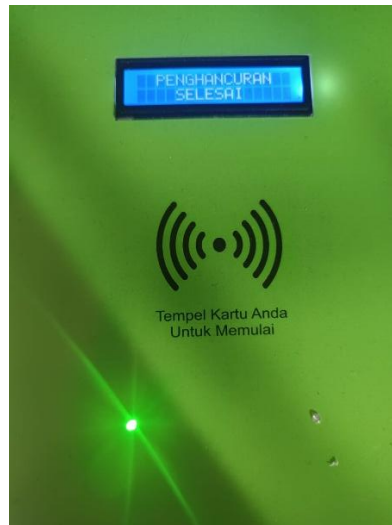
Gambar 5.13 Tampilan Alat Sedang Bekerja

8. Setelah motor menyala dan mesin pemotong kertas bekerja maka hasil pemotongan akan keluar pada tempat penampungan kertas, dapat dilihat pada gambar 5.14.



Gambar 5.14 Hasil Pemotongan Kertas

9. Jika pemotongan kertas sudah selesai maka LCD akan menampilkan bahwa pemotongan kertas telah selesai, dapat dilihat pada gambar 5.15.



Gambar 5.15 Tampilan Penghancuran Selesai

10. Jika tempat penampungan kertas sudah penuh, maka sensor ultrasonik akan mendeteksi kertas yang telah penuh tersebut, dapat dilihat pada gambar 5.16.



Gambar 5.16 Tampilan Kertas Sudah Penuh

11. Hasil dari kerja alat yang terhubung ke NodeMCU akan menampilkan status alat secara *real-time* menggunakan aplikasi Telegram, dapat dilihat pada gambar 5.17.



Gambar 5.17 Tampilan Status Alat Dari Telegram

BAB VI

PENUTUP

Berdasarkan hasil perancangan sistem dan pembuatan sistem ini, dapat diambil kesimpulan serta saran-saran yang nantinya dapat bermanfaat bagi pengembangan sistem berikutnya.

6.1 Kesimpulan

Berdasarkan penjelasan yang tertera pada bab-bab sebelumnya, maka dapat diambil kesimpulan sebagai berikut :

1. RFID dapat bekerja dengan baik sebagai keamanan sebelum memulai mengaktifkan alat.
2. Sensor LDR dapat bekerja dengan baik untuk mendeteksi kertas yang masuk ke dalam mesin pemotong kertas.
3. Sensor Ultrasonik dapat bekerja dengan baik untuk mendeteksi tempat penampungan kertas yang telah terpotong apabila tempat penampungan sudah penuh.
4. Speaker dapat mengeluarkan suara notifikasi dan status dengan baik.
5. LCD dapat menampilkan notifikasi RFID dan hasil pendeteksian dari sensor LDR dan sensor Ultrasonik dengan baik.
6. LED dan buzzer dapat bekerja dengan baik sebagai indikator pembacaan RFID.
7. Telegram dapat bekerja dengan baik sebagai indikator status alat secara *real-time*.

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. Sistem keamanan alat menggunakan sensor RFID yang kartunya mudah hilang, maka disarankan pengembang berikutnya menggunakan sistem *face id*.
2. Perlu dilakukan peningkatan pada kekuatan dan ketahanan mekanik pisau penghancur, agar alat dapat digunakan untuk jenis media selain kertas.
3. Penambahan sensor pendukung, seperti sensor suhu atau sensor beban, disarankan untuk mendeteksi kondisi seperti motor *overheat* atau macet, yang kemudian dapat dikirimkan melalui Telegram.
4. Pengembangan sistem *monitoring* berbasis *cloud* atau integrasi dengan *database*, untuk menyimpan riwayat penggunaan alat dan notifikasi secara lebih sistematis.

Berdasarkan kesimpulan di atas, diharapkan tugas akhir ini bisa dipahami dan sistem yang dikembangkan yang dibuat dapat dikembangkan menjadi lebih sempurna dari penelitian sebelumnya dan dapat menjadi sumber informasi yang cukup untuk menyusun tugas akhir maupun penelitian berikutnya yang lebih baik.

DAFTAR PUSTAKA

- Kosim, Mukhammad. "Desain Kontruksi Mesin Pencacah Kertas Dengan Pisau Pemotong Zig-Zag Dan Tegak Lurus." (2021).
- Setiawan, Budi, and Fatkur Rhohman. "Rancang Bangun Alat Pencacah Sampah Kertas Dengan Ketebalan 5 mm." *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*. Vol. 6. No. 1. 2022.
- Quadratullah, Fahmi, et al. "RANCANG BANGUN ALAT PENCACAH KERTAS (Dengan Kapasitas 4 Kilo Gram/Jam)." *TECHNOMA 2.2* (2023): 81-89.
- Guna, Kadek Sastra. *Rancang Bangun Mesin Pencacah Kertas Kapasitas 15 Lembar Kertas Hvs*. Diss. Politeknik Negeri Bali, 2024.
- Setiawan, Irfan, and Bangkit Ayatullah Datupalinge. "INOVASI PENGURANGAN SAMPAH PERKOTAAN DALAM MEWUJUDKAN SUSTAINABLE DEVELOPMENT GOALS DI INDONESIA." *Jurnal Ilmiah Administrasi Pemerintahan Daerah* 15.2 (2023): 165-180.
- Mei Prabowo (2020). "Metodologi Pengembangan Sistem Informasi." Lembaga Penelitian dan Pengabdian kepada Masyarakat (LP2M) IAIN Salatiga. 1-2.
- Dedi Rahman Prehanto (2020). "Buku Ajar Konsep Sistem Informasi". Scopindo Media Pustaka JL. Kebon Sari No. 03, Surabaya. 5-6.

- Fadhla Binti Junus (2020).”Dasar Pemograman Berbasis Web Dengan Php NativeProcedural & MySql”. CV Budi Utama JL. Rajawali, G Elang 6 No, 3 Sleman : 37-38.
- Sanjaya dan Anggraeni (2020).”Eksperimen Virtual Mikrokontroler Arduino Menggunakan Tinkercard”. CV Bolabot Komp. Permata Biru Blok i No.76, Cinunuk, Cileunyi, Bandung : 127.
- Hermanto dan Chairandy (2023).”Dasar Elektronika Praktis”. Politeknik Ilmu Pelayaran Semarang. JL. Singosari 2A, Wonodri, Semarang Selatan, Kota Semarang : 77.
- Kiagus Ahmad Roni dan Cekmas Cekdin (2020), “Sistem Kendali Proses Industri”. CV ANDI OFFSET. JL.Beo 38-40, Yogyakarta : 9-10.
- Halimah dan Oktawati (2021), “Semua Bisa Belajar Arduino”. Penerbit Lindan Bestari. JL. Raya Leuwiliang, Ds Cibeber II, Leuwiliang, Bogor : 27-28.
- Miftachul dan Saputro (2023), “Sensor dan Actuator menggunakan Arduino”. Media Nusa Creative. Bukit Cemara Tidar H5 N0. 34, Malang : 46-47.
- Widharma dan Wiranata (2022), “ Mikrokontroler dan Aplikasi”. Wawasan Ilmu. Desa Kaliwedi, Banyumas, Jawa Tengah : 83-84.
- Aswandi dan Safriadi (2023), “*Internet of Things* dan aplikasi penerapan Mikrokontroler Arduino”. JL. Beo 38-40 , Yogyakarta : 70-71.
- Ilham, Hardisal, Candra (2020), “Monitoring dan Stimulasi Detak Jantung dengan Murottal Al-quran berbasis Internet of Things”. Jl. Bojong Genteng No. 18, Jawa Barat : 32-33.

LAMPIRAN

```
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>
#include <DFPlayer_Mini_Mp3.h>
#include <SoftwareSerial.h>
#include <Buzzer.h>
```

```
#define ldr      A0
#define ldr2     A7
#define relay1   A1
#define relay2   8
#define led1     A2
#define led2     A3
#define trig_pin A4
#define echo_pin A5
#define BUZZER_PIN A6
```

```
Buzzer buzzer(BUZZER_PIN,13); // create ezBuzzer object that attach to a pin;
```

```
#define RST_PIN    5      // Configurable, see typical pin layout above
#define SS_PIN     53     // Configurable, see typical pin layout above
```

```
SoftwareSerial mySerial(11, 10); // RX, TX
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
String strUID;
String UidSet = "1952472320";
bool izin_masuk = false;
unsigned long prevTime = millis();
int jarak_penuh = 8;
int tampil = 0;
bool tampilkan_1x = false;
bool stt_alat_bekerja = false;
bool detek_kertas = false;
bool potong_berakhir = false;
```

```
void setup() {
  Serial.begin(9600);
  Serial2.begin(9600);
  mySerial.begin (9600);
  pinMode(ldr,INPUT);
  pinMode(ldr2,INPUT_PULLUP);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
```

```

digitalWrite(relay1,HIGH);
digitalWrite(relay2,HIGH);
pinMode(trig_pin, OUTPUT);
pinMode(echo_pin, INPUT);

while (!Serial); // Do nothing if no serial port is opened (added for Arduinos
based on ATMEGA32U4)
mp3_set_serial (mySerial); //set softwareSerial for DFPlayer-mini mp3 module
mp3_set_volume (30);

lcd.begin();
lcd.backlight();
SPI.begin(); // Init SPI bus
mfrc522.PCD_Init(); // Init MFRC522
delay(4); // Optional delay. Some board do need more time after init to be
ready, see Readme
mfrc522.PCD_DumpVersionToSerial(); // Show details of PCD - MFRC522
Card Reader details
digitalWrite(led1,HIGH);
digitalWrite(led2,LOW);
lcd2("DIMAS PRASETYO","2110115260061");
mp3_play(1);
delay(7000);
mp3_play(2);
lcd2("Tempel Kartu","untuk memulai");
}

void loop() {

int ultra = ultra_sonic();
if (ultra < jarak_penuh){
  lcd2("Kertas Sudah","Penuh");
  tampil=0;
  return;
} else {
  if (izin_masuk) {
    if (tampil++==1) lcd2("ALAT SIAP","DIGUNAKAN");
  }
  else {
    if (tampil++==1) lcd2("Tempel Kartu","untuk memulai");
  }
}

if (millis() - prevTime > 5000) {
  prevTime = millis();
  if (izin_masuk) {

```

```

        if (stt_alat_bekerja && !detek_kertas) {
            delay(2000);

digitalWrite(relay1,HIGH);
    potong_berakhir=true;
    }
    }
    }

    if (izin_masuk) {
        if (ldr2_value()){
            while (!ldr1_value()) {
                digitalWrite(relay2,LOW);
            }
        } else {
            if (stt_alat_bekerja){
                Serial2.println("end");
                lcd2("PENGHANCURAN","SELESAI");
                stt_alat_bekerja = false;
            }
        }
        delay(700);
        digitalWrite(relay2,HIGH);
        if (ldr1_value()){
            if (!tampilkan_1x){
                tampilkan_1x = true;
                detek_kertas = true;
                stt_alat_bekerja = true;
                Serial2.println("kerja");
                lcd2("ALAT SEDANG","BEKERJA");
                digitalWrite(relay1,LOW);
                digitalWrite(relay2,HIGH);
            }
        } else {
            potong_berakhir=false;
            detek_kertas = false;
            tampilkan_1x = false;
        }
    }

    delay(100);
    if ( ! mfrc522.PICC_IsNewCardPresent()) return;
    if ( ! mfrc522.PICC_ReadCardSerial()) return;
    MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
    strUID = String(mfrc522.uid.uidByte[0]) + String(mfrc522.uid.uidByte[1]) +
        String(mfrc522.uid.uidByte[2]) + String(mfrc522.uid.uidByte[3]);

```

```

        Serial.println(strUID);

    buzzer_rfid();

    if (strUID == UidSet) {
        Serial2.println("ok");
        mp3_play(3);
        lcd2("ALAT SIAP","DIGUNAKAN");
        izin_masuk = true;
        digitalWrite(led1,LOW);
        digitalWrite(led2,HIGH);
        Serial.println(strUID);
    } else {
        Serial2.println("er");
        mp3_play(4);
        izin_masuk = false;
        digitalWrite(led1,HIGH);
        digitalWrite(led2,LOW);
    }
}

int ultra_sonic(){
    long echotime;
    float distance;
    digitalWrite(trig_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_pin, LOW);
    echotime= pulseIn(echo_pin, HIGH);
    distance= 0.0001*((float)echotime*340.0)/2.0;
    return distance;
}

void buzzer_rfid(){
    buzzer.begin(100);
    buzzer.sound(NOTE_E7, 80);
    buzzer.sound(NOTE_C7, 80);
    buzzer.sound(NOTE_B6, 80);
    buzzer.end(100);
}

void lcd2(String text1, String text2) {
    lcd.clear();
    lcd.setCursor((16-text1.length())/2,0);
    lcd.print(text1);
    lcd.setCursor((16-text2.length())/2,1);
    lcd.print(text2);
}

```

```

bool ldr1_value(){
  if (analogRead(ldr)>30) return true;else return false;
}

bool ldr2_value(){
  bool ret_ = false;
  if (analogRead(ldr2)>35 && analogRead(ldr2)<800 ) ret_=true;
  return (ret_);
}
#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>

const char* ssid = "qwerty";
const char* password = "12345678";

#define BOTtoken
"8175096019:AAGc71JFZrGwzeKbShjdTufPY2g5ZSpg4Uk" // your Bot Token
(Get from Botfather)
#define CHAT_ID "6323903621"

#ifdef 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;
String msg_last="";

void setup() {
  Serial.begin(9600);

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

```

```

    // 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.sendMessage(CHAT_ID, "Bot Started", "");
}

void loop() {

    if (Serial.available()){
        String msg = "";
        while(Serial.available()){
            msg+=char(Serial.read());
            delay(50);
        }
        msg.trim();
        if (msg=="ok") {
            bot.sendMessage(CHAT_ID, "KARTU ANDA COCOK", "");
        } else
        if (msg=="er") {
            bot.sendMessage(CHAT_ID, "KARTU ANDA TIDAK COCOK", "");
        } else
        if (msg=="kerja") {
            bot.sendMessage(CHAT_ID, "MESIN SEDANG BEKERJA", "");
        } else
        if (msg=="end") {
            bot.sendMessage(CHAT_ID, "PEMOTONGAN SELESAI", "");
        }
    }
}

```