

**ALAT PENGHITUNG KERTAS OTOMATIS  
BERBASIS IOT**

**PROYEK AKHIR**

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



Disusun oleh :

Andreas Lie Alviero      NIM : 1051901

Dimas Setiawan Nugroho      NIM : 1051905

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2023**

**LEMBAR PENGESAHAN**

**JUDUL PROYEK AKHIR**

**ALAT PENGHITUNG KERTAS OTOMATIS BERBASIS IOT**

Oleh :

Andreas Lie Alviero                      NIM : 1051901

Dimas Setiawan Nugroho                NIM : 1051015

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

Menyetujui,

Pembimbing 1



Zanu Saputra, M.Tr.T

Pembimbing 2



Yudhi, M.T

Penguji 1



Surojo, M.T

Penguji 2



Dr. Parulian Silalahi

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Andreas Lie Alviero                      NIRM : 1051901

Nama Mahasiswa 2 : Dimas Setiawan Nugroho              NIRM : 1051905

Dengan Judul : ALAT PENGHITUNG KERTAS OTOMATIS BERBASIS IOT

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

Sungailiat,30 Januari 2023

Nama Mahasiswa

1. Andreas Lie Alviero
2. Dimas Setiawan Nugroho

Tanda Tangan



## **ABSTRAK**

*Perhitungan kertas HVS umumnya masih dilakukan secara manual pada toko percetakan dan Alat tulis kantor. Perhitungan secara manual tentu sangat memakan banyak waktu jika jumlah kertas yang dihitung hingga ratusan. Maka dari itu, dibutuhkan alat yang mampu melakukan perhitungan kertas secara otomatis dan pengoperasian nya dapat dilakukan melalui aplikasi pada smartphone dan juga secara manual. Metode yang dipakai pada penelitian ini adalah dengan melakukan kalibrasi sensor LDR dan arus dengan metode regresi linier dan Polinomial, perhitungan kertas dengan menggunakan sensor LDR, dan menampilkan hasil perhitungan pada aplikasi PCT-1 dan LCD 16x2. Hasil pengujian menunjukan jika pengoperasian alat dengan aplikasi PCT-1 dapat dilakukan hingga jarak 100 meter. Alat yang dibuat mampu melakukan perhitungan kertas dengan kecepatan perhitungan 3 lembar per detik. Bisa disimpulkan jika alat perhitungan kertas otomatis akan efisien jika jumlah lembar kertas yang dihitung lebih dari 50 lembar.*

*Kata kunci: Alat Penghitung Kertas, Aplikasi PCT-1, Regresi Linier*

## **ABSTRACT**

*HVS paper calculations are generally still done manually at printing and stationery shops. Manual calculations are certainly very time-consuming if the number of papers calculated is up to hundreds. Therefore, a tool is needed that is able to do paper calculations automatically and its operation can be done through applications on smartphones and also manually. The method used in this research is to calibrate the LDR and current sensors with linear regression and polynomial methods, paper calculations using LDR sensors, and display the calculation results in the PCT-1 and 16x2 LCD applications. The test results show that the operation of the tool with the PCT-1 application can be carried out up to a distance of 100 meters. The tool made is able to perform paper calculations with a calculation speed of 3 sheets per second. It can be concluded that an automatic paper calculation tool will be efficient if the number of sheets of paper calculated is more than 50 sheets.*

*Keywords: Paper Counter Tools, PCT-1 Applications, Linear Regression*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan makalah proyek akhir ini dengan baik dan sesuai dengan waktu yang telah ditentukan.

Makalah ini dibuat untuk memenuhi salah satu syarat wajib kelulusan Diploma IV di Politeknik Manufaktur Negeri Bangka Belitung. Dengan adanya makalah ini diharapkan pembaca dapat mengetahui gambaran proyek akhir yang dibuat oleh penulis. Makalah proyek akhir ini dibuat dengan melakukan pengembangan dari jurnal-jurnal penelitian yang sudah pernah dilakukan sebelumnya. Penulis mencoba menerapkan ilmu pengetahuan yang didapat selama 4 tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dalam pembuatan proyek akhir ini. Selain itu, penulis juga mendapatkan informasi berupa data-data pendukung dari makalah-makalah proyek akhir mahasiswa Politeknik Manufaktur Negeri Bangka Belitung tahun-tahun sebelumnya.

Selama menyusun makalah proyek akhir ini penulis mendapatkan banyak bantuan dan bimbingan dari berbagai pihak sehingga penulisan makalah ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Orangtua dan keluarga tercinta yang selalu memberikan dukungan baik secara moral maupun materil sehingga penulis dapat memberikan hasil yang terbaik.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Zanu Saputra, M.Tr.T selaku dosen pembimbing 1 dalam proyek akhir ini.
4. Bapak Yudhi, M.T. selaku dosen pembimbing 2 dalam proyek akhir ini.
5. Bapak Muhammad Iqbal Nugraha, M.Eng. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Indra Dwisaputra, M.T. selaku Ketua Program Studi DIV Teknik

Elektronika Politeknik Manufaktur Negeri Bangka Belitung.

7. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah membantu dalam penyelesaian proyek akhir.
9. Seluruh pihak yang ikut terlibat baik secara langsung maupun tidak langsung dalam penyelesaian proyek akhir ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat banyak kekurangan karena terbatasnya pengetahuan dan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan dan pengembangan penulisan makalah ini di masa yang akan datang. Penulis berharap makalah ini dapat bermanfaat bagi pihak yang berkepentingan pada khususnya dan bagi perkembangan ilmu pengetahuan dan teknologi pada umumnya.

Sungailiat, 30 Januari 2023

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN.....	ii
PERNYATAAN BUKAN PLAGIAT.....	<b>Error! Bookmark not defined.</b>
ABSTRAK.....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Proyek Akhir.....	3
BAB II LANDASAN TEORI.....	4
2.1 Kertas.....	4
2.2 Alat Penghitung Kertas Otomatis.....	4
2.3 Regresi Linier.....	4
2.4 Arduino Uno.....	5
2.5 NodeMCU ESP 32.....	6
2.6 Sensor Infrared.....	7
2.7 <i>Light Dependent Resistor (LDR)</i> .....	7
2.8 Sensor Arus ACS712.....	8
2.9 Motor DC.....	9
2.10 Driver Motor L298N.....	10
2.11 <i>AC to DC Power Supply 12 V</i> .....	10
2.12 <i>Thunkable</i> .....	11
2.13 <i>Firebase</i> .....	11
2.14 <i>Internet of Things (IoT)</i> .....	12
BAB III METODE PELAKSANAAN.....	13
3.1 Pengumpulan dan Pengolahan data.....	15

3.2	Perancangan <i>Hardware</i> dan Aplikasi.....	15
3.2.1	Perancangan <i>Hardware</i> .....	15
3.2.2	Perancangan Aplikasi.....	16
3.3	Perakitan <i>hardware</i> dan konstruksi alat.....	16
3.4	Perakitan Aplikasi .....	17
3.5	Pengujian Aplikasi dan <i>Hardware</i> .....	19
3.5.1	Pengujian Aplikasi .....	19
3.5.2	Pengujian <i>Hardware</i> .....	19
3.6	Pengujian Keseluruhan Sistem.....	20
3.7	Analisa Data .....	20
3.8	Pembuatan Laporan Akhir dan Publikasi.....	20
BAB IV HASIL DAN PEMBAHASAN .....		21
4.1	Deskripsi Alat.....	21
4.2	Perancangan dan Pembuatan <i>Hardware</i> .....	22
4.2.1	Perancangan dan Pembuatan Konstruksi Alat.....	22
4.2.2	Perancangan dan Pembuatan Rangkaian Elektrik .....	25
4.3	Pembuatan Aplikasi PCT-1 .....	25
4.3.1	Pembuatan <i>Login Screen</i> .....	26
4.3.2	Pembuatan <i>Logic Block</i> .....	31
4.4	Pengujian Sensor Arus ACS712 .....	37
4.4.1	Perancangan dan pembuatan rangkaian sensor arus ACS712.....	38
4.4.2	Pengkalibrasian sensor arus ACS712.....	38
4.4.3	Pengujian sensor arus ACS712 pada Alat Penghitung Kertas .....	41
4.5	Pengujian sensor modul LDR .....	43
4.5.1	Perancangan dan pembuatan rangkaian sensor modul LDR .....	44
4.5.2	Pengkalibrasian sensor LDR .....	44
4.5.3	Pengujian sensor modul LDR untuk pembacaan kertas.....	48
4.6	Pengujian komunikasi serial Node MCU ESP32 dengan Arduino Uno ....	50
4.7	Pengujian Keseluruhan Alat.....	51
4.7.1	Pengujian Aplikasi PCT-1.....	52
4.7.2	Pengujian waktu perhitungan alat dengan perhitungan manual.....	53
BAB V PENUTUP.....		56

5.1	Kesimpulan.....	56
5.2	Saran.....	56
	DAFTAR PUSTAKA .....	57
	LAMPIRAN 1 Daftar Riwayat Hidup .....	60
	LAMPIRAN 2 Program Arduino.....	63
	LAMPIRAN 3 Program NodeMCU ESP32 .....	69



## DAFTAR TABEL

Tabel 4. 1 Perbandingan hasil konversi tegangan dengan pengukuran amperemeter .....	39
Tabel 4. 2 Hasil Pengujian Sensor Arus ACS712.....	41
Tabel 4. 3 Perbandingan Lux Meter dengan Tegangan Sensor LDR .....	45
Tabel 4. 4 Perbandingan pembacaan sensor LDR (linier) dengan Lux Meter.	47
Tabel 4. 5 Perbandingan pembacaan sensor LDR (Polinomial) dengan Lux Meter.....	47
Tabel 4.6 Hasil pengujian sensor LDR dengan metode regresi linier.....	48
Tabel 4.7 Hasil pengujian sensor LDR dengan metode Polinomial .....	49
Tabel 4. 8 Pengaruh Jarak Terhadap Pengoperasian Alat.....	52
Tabel 4. 9 Perbandingan waktu perhitungan alat dengan perhitungan manual	53

## DAFTAR GAMBAR

Gambar 1. 1	Perhitungan Kertas yang dilakukan Secara Manual.....	2
Gambar 2. 1	Arduino Uno.....	6
Gambar 2. 2	NodeMCU ESP32 .....	6
Gambar 2. 3	Sensor Infrared .....	7
Gambar 2. 4	<i>Light Dependent Resistor</i> .....	8
Gambar 2. 5	Sensor Arus ACS712 .....	9
Gambar 2. 6	Motor DC .....	9
Gambar 2. 7	Driver L298N .....	10
Gambar 2. 8	Catu Daya atau <i>Power Supply</i> .....	11
Gambar 2. 9	Logo <i>Thinkable</i> .....	11
Gambar 2. 10	Logo <i>Firebase</i> .....	12
Gambar 2. 11	<i>Internet of Things</i> .....	12
Gambar 3. 1	<i>FlowChart</i> Metode Pelaksanaan Proyek Akhir.....	14
Gambar 3. 2	<i>FlowChart</i> Program Arduino .....	18
Gambar 4. 1	Blok diagram sistematis alat .....	21
Gambar 4. 2	Desain Konstruksi Alat Penghitung Kertas.....	22
Gambar 4. 3	Desain Alat Tampak Depan .....	23
Gambar 4. 4	Desain Alat Tampak Samping.....	23
Gambar 4. 5	Desain Alat Tampak Atas .....	24
Gambar 4. 6	Konstruksi Alat Penghitung Kertas Otomatis Berbasis IoT.....	24
Gambar 4. 7	Desain Rangkaian Elektrik.....	25
Gambar 4. 8	Tampilan Awal Aplikasi .....	26
Gambar 4. 9	<i>Logic Block</i> Tampilan Awal Aplikasi .....	27
Gambar 4. 10	Tampilan Aplikasi Saat Gagal <i>Login</i> .....	28
Gambar 4. 11	<i>Logic Block</i> Gagal <i>Login</i> .....	28
Gambar 4. 12	Tampilan Aplikasi Saat Berhasil <i>Login</i> .....	29
Gambar 4. 13	<i>Logic Block</i> Berhasil <i>Login</i> .....	30
Gambar 4. 14	Tampilan <i>Logic Block</i> .....	31
Gambar 4. 15	<i>Logic Block</i> Tombol <i>Start</i> .....	32

Gambar 4. 16 <i>Logic Block Value</i> Tombol <i>Start</i> .....	32
Gambar 4. 17 Tampilan <i>Logic Block</i> terhadap <i>Value</i> Tombol <i>Start</i> .....	32
Gambar 4. 18 <i>Logic Block</i> Tombol <i>Stop</i> .....	33
Gambar 4. 19 Tampilan <i>Logic Block</i> Saat <i>Button Stop</i> Ditekan.....	33
Gambar 4. 20 <i>Logic Block</i> Tombol <i>Reset</i> .....	34
Gambar 4. 21 Tampilan <i>Logic Block</i> Saat <i>Button Reset</i> Ditekan .....	34
Gambar 4. 22 <i>Path Data Firebase</i> .....	35
Gambar 4. 23 <i>Logic Block</i> Tampilan <i>countPath</i> .....	35
Gambar 4. 24 <i>Logic Block</i> Tampilan <i>motorPath</i> .....	35
Gambar 4. 25 <i>Logic Block</i> Tampilan <i>detectorPath</i> .....	36
Gambar 4. 26 <i>Logic Block</i> Tampilan <i>warningPath</i> .....	36
Gambar 4. 27 <i>Logic Block</i> Keseluruhan <i>Data Path</i> .....	36
Gambar 4. 28 Desain Rangkaian Pengujian Sensor Arus ACS712 .....	38
Gambar 4. 29 Rangkaian Pengujian Sensor Arus ACS712 .....	38
Gambar 4. 30 Hubungan antara pembacaan sensor arus ACS712 dengan Amperemeter. ....	40
Gambar 4. 31 Kertas Tersangkut pada Alat Penghitung Kertas .....	42
Gambar 4. 32 Nilai Arus yang Terbaca Oleh Sensor Arus ACS712 .....	42
Gambar 4. 33 Rangkaian Skematik Pengujian Sensor Modul LDR .....	44
Gambar 4. 34 Rangkaian Pengujian Sensor Modul LDR .....	44
Gambar 4. 35 Pengambilan Data Sensor LDR dengan alat ukur Lux Meter...	45
Gambar 4. 36 Hubungan antara pembacaan sensor LDR dengan Lux meter ..	46
Gambar 4. 37 Perbandingan Pembacaan Kertas dengan Metode Regresi Linier	48
Gambar 4. 38 Perbandingan Pembacaan Kertas dengan Metode Polinomial..	49
Gambar 4. 39 Pengujian Sensor LDR dengan Objek Kertas .....	49
Gambar 4. 40 Tampilan Serial Monitor pada (a) Arduino Uno dan (b) Node MCU ESP32 .....	51
Gambar 4. 41 Perbandingan perhitungan (a) secara manual dengan (b) perhitungan menggunakan alat penghitung kertas. ....	55

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan ilmu pengetahuan yang semakin berkembang dari waktu ke waktu mendukung terciptanya teknologi yang semakin canggih [1]. Salah satu teknologi nirkabel yang terkenal adalah ‘*Internet of Things*’ atau biasa disebut dengan IoT. IoT sendiri merupakan suatu konsep dimana beberapa benda (*device*) yang masing-masing terhubung di suatu jaringan dan memiliki kemampuan untuk mengirim atau mentransmisikan data ke benda lainnya tanpa membutuhkan bantuan komputer atau manusia (*manual*) [2].

Salah satu pekerjaan yang dapat dipermudah dengan adanya perkembangan teknologi ini adalah menghitung kertas. Dengan adanya perkembangan sistem teknologi nirkabel seperti IoT ini, dapat kita manfaatkan dengan menggabungkan kedua sistem ini sehingga pekerjaan yang dilakukan manusia menjadi lebih efektif dan efisien lagi [2].

Berdasarkan pengamatan yang sudah dilakukan terhadap 5 toko alat tulis dan percetakan di Kota Sungailiat, proses perhitungan kertas ketika pembelian eceran masih dilakukan secara manual. Perhitungan kertas secara *manual* dinilai tidak efektif karena dapat mengakibatkan *human error*. ini terjadi karena apabila kertas yang dihitung telah mencapai ratusan bahkan ribuan, manusia berpeluang salah dalam menghitungnya. Selain itu penghitungan secara *manual* dinilai tidak efektif dari sisi waktu menghitungnya [3].

Maka dari itu dengan memanfaatkan teknologi yang ada dan memperhatikan segala problematika dalam perhitungan kertas di percetakan, dikembangkan suatu alat yang dapat membantu dibidang produksi kertas yakni “Alat Penghitung Kertas Otomatis Berbasis IoT”. Sistem kerja dari alat yang dikembangkan yakni dengan memasukkan sejumlah kertas ke alat penghitung sehingga nantinya kertas satu persatu terdeteksi oleh sensor LDR dan langsung dikirim ke arduino guna di konversi sebagai hitungan. Setelah itu arduino

mengirimkan hasil hitungan ke nodeMCU lalu dikirimkan ke *smartphone* sebagai *User Interface*.



Gambar 1. 1 Perhitungan Kertas yang dilakukan Secara Manual

## 1.2 Rumusan Masalah

Dari latar belakang yang ada, maka rumusan masalah yang akan dikaji adalah sebagai berikut:

1. Bagaimana cara mendesain alat yang mampu melakukan perhitungan kertas secara otomatis?
2. Bagaimana cara membuat alat yang mampu melakukan perhitungan kertas secara otomatis?
3. Bagaimana cara kerja IoT untuk menampilkan perhitungan kertas dan mengontrol alat penghitung kertas?
4. Bagaimana efektivitas dan efisiensi dari alat penghitung kertas otomatis yang dikembangkan?

## 1.3 Batasan Masalah

Batasan-batasan masalah dalam proyek akhir ini adalah sebagai berikut:

1. Jenis kertas yang bisa dihitung adalah jenis kertas A4 dan F4.
2. Jumlah kertas yang dihitung maksimal adalah  $\frac{1}{4}$  rim.

#### **1.4 Tujuan Proyek Akhir**

Tujuan dalam proyek akhir ini adalah sebagai berikut:

1. Mendesain alat yang dapat melakukan perhitungan kertas secara otomatis.
2. Membuat alat yang dapat melakukan perhitungan kertas secara otomatis.
3. Mengontrol dan memonitoring perhitungan kertas melalui *smartphone* dengan memanfaatkan *Internet Of Things*.
4. Mendeskripsikan efektivitas dan efisiensi dari alat penghitung kertas otomatis.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kertas**

Kertas merupakan bahan yang dihasilkan dari proses kompres atau penekanan *pulp* untuk mendapatkan seratnya. Serat dari proses kompresi mengandung selulosa dan hemiselulosa. Kertas umumnya digunakan untuk melukis, mencetak dan juga menulis [4].

#### **2.2 Alat Penghitung Kertas Otomatis**

Penelitian terdahulu terkait Alat Penghitung Kertas Otomatis sudah pernah dilakukan oleh Hanna Alifia dan Putri Riyanto pada tahun 2021 [5]. Peneliti membuat alat penghitung kertas dan sortir kertas A4 dengan sensor LDR dan *Photodiode*. Sensor LDR digunakan untuk mendeteksi kebersihan kertas yang nantinya akan di sortir. *Photodiode* digunakan sebagai penghitung kertas yang sudah disortir. Hasil perhitungan dan penyortiran kertas ditampilkan di aplikasi blynk. Berdasarkan penelitian yang sudah dilakukan, dapat disimpulkan bahwa alat yang sudah dibuat berhasil menghitung dan menyortir kertas A4 berdasarkan pembacaan sensor LDR dan *Photodiode*, namun alat yang dibuat tidak bisa mendeteksi kertas jika tertarik lebih dari satu yang berpotensi mengganggu perhitungan, dan juga tidak ada pendeteksi jika kertas yang tertarik kusut sehingga mengganggu proses mekanisme alat. Pada penelitian ini, peneliti ingin memodifikasi alat yang sudah ada dengan menambahkan sensor arus sebagai pendeteksi jika kertas yang tertarik kusut, dan penggunaan LDR yang dialihfungsikan menjadi pendeteksi kertas jika tertarik lebih dari 1. Proses perhitungan kertas juga akan menggunakan sensor LDR, dan perhitungan akan difokuskan untuk kertas yang masih baru sehingga tidak ada proses penyortiran pada penelitian ini.

#### **2.3 Regresi Linier**

Regresi linier merupakan suatu metode yang digunakan pada ilmu statistika untuk membuat hubungan ataupun model antara variabel terikat yang

bersifat dependen dengan variable bebas yang bersifat independent. Regresi linier sederhana memiliki jumlah variabel bebas yang terdiri dari satu, sedangkan regresi linier berganda memiliki jumlah variabel bebas lebih dari satu. Regresi linier memiliki fungsi untuk melakukan prediksi terhadap variabel terikat [6]. Regresi linier akan digunakan sebagai metode dalam pengkalibrasian sensor dengan menggunakan nilai tegangan keluaran sensor sebagai variabel bebas dan nilai arus pada multimeter sebagai variabel kontrol. Metode ini dipilih karena mampu melakukan prediksi terhadap suatu data dengan nilai yang mendekati nilai aktual [7] [8]. Nilai tegangan keluaran sensor diperoleh dari konversi nilai adc dengan rumus berikut :

$$\text{Tegangan (Vout)} = Vref \times \frac{ADC}{1024} [9] \quad (1)$$

Keterangan:

Vout = Tegangan keluaran sensor (V)

Vref = Tegangan maksimal sensor saat pembacaan maksimum

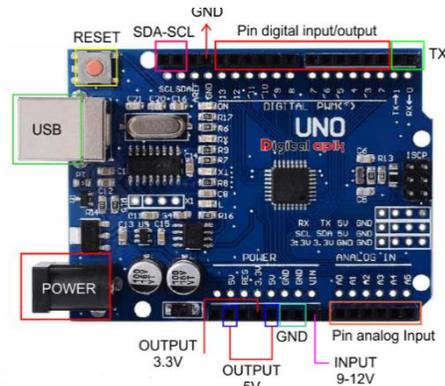
ADC = Pembacaan analog sensor (0 – 1024)

1024 = konversi 10 bit ke bilangan desimal

## 2.4 Arduino Uno

Arduino Uno merupakan papan mikrokontroler yang menggunakan Atmega 328 sebagai basis mikrokontroler nya. Arduino uno terdiri dari 14 pin I/O dengan 6 pin sebagai output PWM, 6 pin sebagai analog input, 2 pin sebagai pin TX dan RX. Arduino digunakan untuk mengatur berbagai komponen elektronika seperti sensor, LED, driver dan lain-lain [10]. Menurut Steven J. Sokop, et.al pada penelitiannya yang berjudul Trainer Periferal Antarmuka Berbasis Mikrokontroler Arduino Uno [11], Arduino uno membutuhkan tegangan *input* optimal 7-12 V, sehingga Arduino uno akan sangat cocok digunakan pada pembuatan alat ini. Jumlah pin I/O dari Arduino uno juga mencukupi dari kebutuhan alat penghitung kertas ini. Arduino uno juga memiliki kemampuan komunikasi serial yang dibutuhkan untuk proses komunikasi serial dengan Node MCU ESP32 sebagai penghubung alat penghitung kertas dengan *Internet of*

*Things*. Dalam penelitian ini Arduino Uno digunakan sebagai pusat kendali dari keseluruhan proses alat, seperti pembacaan sensor, dan mekanisme kerja alat.



Gambar 2. 1 Arduino Uno  
Sumber : (pintarelektro.com)

## 2.5 NodeMCU ESP 32

NodeMCU merupakan *board* elektronik yang dilengkapi dengan chip ESP32 sebagai pusat kendalinya. NodeMCU ESP32 ini memiliki kemampuan seperti mikrokontroler dan juga bisa terhubung dengan internet khususnya *WiFi* [12]. Dalam penelitian ini NodeMCU digunakan untuk media komunikasi serial antara smartphone dengan alat. Node MCU akan terhubung dengan smartphone melalui koneksi internet melalui *Database*. *Smartphone* dapat memberi perintah kepada alat melalui NodeMCU dan NodeMCU akan mengirimkan hasil pembacaan sensor di Arduino ke *smartphone*.



Gambar 2. 2 NodeMCU ESP32  
Sumber : (indobot.co.id)

## 2.6 Sensor Infrared

Sensor Infrared adalah sensor yang mampu memantulkan cahaya infra merah ketika terdapat rintangan atau media pantul didepannya. Sensor infrared ini memiliki 2 bagian utama yaitu LED Infra red dan *Photodiode*. LED Infra red akan memancarkan gelombang atau cahaya infra merah yang nanti nya akan diterima oleh *Photodiode*. Cara kerja dari sensor ini adalah ketika terdapat media pantul, atau rintangan didepan LED Infra red maka gelombang infra red yang dipancarkan akan memantul dan menuju *Photodiode*. Sensor akan mengeluarkan output *LOW* jika *Photodiode* menerima pantulan cahaya infrared, dan juga sebaliknya [13]. Menurut Mulyadi, et.al pada penelitiannya yang berjudul Purwarupa Alat Penghitung Kertas dan Harga Jasa Pemakaian Mesin Fotokopi Berbasis Mikrokontroler AT89S51 [14], sensor infra red digunakan karena sensor ini memiliki *sensitivitas* pembacaan yang tinggi jika objek pantulnya berupa benda yang tidak menyerap cahaya seperti kertas.

Dalam Penelitian ini, sensor infra red LM393 berfungsi sebagai pendeteksi kertas pada bagian *input* alat penghitung kertas. Alat penghitung kertas baru dapat menjalankan proses perhitungan jika tombol *Start* ditekan dan sudah terdapat kertas pada bagian *input* alat penghitung kertas yang ditandai dengan pembacaan logika *LOW* pada sensor infrared.,



Gambar 2. 3 Sensor *Infrared*  
Sumber : (www.nicepng.com)

## 2.7 *Light Dependent Resistor (LDR)*

*Light Dependent Resistor (LDR)* adalah suatu komponen yang mempunyai perubahan resistansi yang besarnya tergantung pada cahaya. Dapat dikatakan bahwa resistansi yang dihasilkan LDR berbanding terbalik dengan jumlah cahaya yang diterima [15]. Pada kondisi gelap, nilai resistansi yang dihasilkan oleh LDR sekitar  $10\text{M}\Omega$ , dan  $1\text{K}\Omega$  pada saat kondisi terang [16]. Menurut Bowo Eko Cahyono, et.al pada penelitiannya yang berjudul

Karakterisasi Sensor LDR dan Aplikasinya pada Alat Ukur Tingkat Kekeruhan Air Berbasis Arduino UNO [17], Sensor LDR memiliki sensitivitas sebesar 0,0082 V per lux atau 2 ADC per lux, dan juga tingkat akurasi pembacaan 87,89 % pada *range* 2 sampai 295 Lux. Berdasarkan pernyataan tersebut, maka sensor LDR dapat digunakan untuk mendeteksi kertas yang tertarik lebih dari satu lembar oleh *roller*.

Dalam penelitian ini, *Light Dependent Resistor* (LDR) digunakan untuk mendeteksi apakah kertas yang lewat tertarik lebih dari 1. Ketika kertas yang melewati LDR lebih dari satu lembar, maka nilai resistansi yang diterima akan berkurang karena kertas akan lebih banyak menyerap cahaya. LDR juga digunakan untuk proses perhitungan kertas. Ketika kertas melewati rangkaian sensor LDR yang disusun secara *vertical* dengan LED, maka cahaya yang diterima LDR akan berkurang, sehingga pembacaan digital dari sensor LDR akan menjadi LOW, begitu juga sebaliknya.

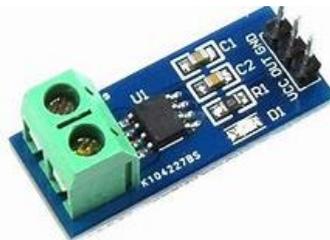


Gambar 2. 4 *Light Dependent Resistor*  
Sumber : (components101.com)

## 2.8 Sensor Arus ACS712

ACS712 adalah *Half Effect current* sensor. ACS712 merupakan sensor yang presisi sebagai sensor arus AC atau DC dalam pembacaan arus. Tingkat akurasi dari pembacaan sensor ini sangat tinggi karena terdapat rangkaian *low-offset linier hall* dengan konstruksi satu lintasan yang terbuat dari bahan tembaga. Cara kerja dari sensor ACS712 adalah dengan memanfaatkan medan magnet yang dihasilkan dari aliran arus yang mengalir di kabel tembaga. Medan magnet tersebut kemudian diproses oleh *integrated Hall IC* dan dikonversi menjadi nilai tegangan proporsional [18]. Menurut Goklas Franz Michael Tambun pada penelitiannya yang berjudul Kalibrasi Tegangan pada Wattmeter dengan sensor Arus Seri ACS712 Berbasis Mikrokontroler ATM 8535 [19], Sensor arus ACS712 yang diuji memiliki persentase *error* 3,34 % yang dibandingkan dengan

perhitungan secara teori. Berdasarkan pernyataan tersebut, maka sensor arus ACS712 dapat digunakan dalam penelitian ini karena nilai *error* pembacaan sensor yang rendah. Dalam penelitian ini sensor arus digunakan untuk mendeteksi adanya kertas yang tersangkut ketika proses penarikan kertas. Ketika kertas yang ditarik oleh *roller* tersangkut, maka beban yang ditanggung oleh motor dc sebagai pemutar *roller* akan meningkat, sehingga arusnya pun akan meningkat dari arus normal.



Gambar 2. 5 Sensor Arus ACS712

Sumber: (<http://kursuselektronikaku.blogspot.com>)

## 2.9 Motor DC

Motor DC merupakan salah satu jenis motor yang memerlukan tegangan yang searah atau tegangan DC yang dialirkan ke kumparan medan motor dan diubah menjadi energi mekanik. Motor DC memiliki 2 kumparan yaitu kumparan jangkar dan kumparan medan. Kumparan jangkar memiliki fungsi sebagai tempat terjadinya gaya gerak listrik (GGL), sedangkan kumparan medan memiliki fungsi sebagai penghasil medan magnet. Cara kerja dari motor DC ini adalah ketika arus pada kumparan jangkar berinteraksi atau bertemu dengan medan magnet, maka akan muncul torsi (T) yang akan memutar motor[20]. Dalam penelitian ini Motor DC digunakan untuk memutar *roller* penarik kertas. Motor DC yang digunakan adalah motor DC printer HP C9508. Tegangan input untuk motor DC ini adalah 6-12 V.

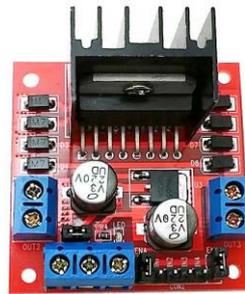


Gambar 2. 6 Motor DC

Sumber : ([printerpoint.co.in](http://printerpoint.co.in))

## 2.10 Driver Motor L298N

Driver Motor L298N adalah modul driver khusus untuk motor DC yang berfungsi untuk mengatur arah putaran dan juga kecepatan motor DC. IC yang digunakan pada driver motor ini adalah IC L298N yang disusun dengan rangkaian H – Bridge. Rangkaian ini mampu mengatur beban induktif kumparan. Driver L298N ini mampu mengubah arah putaran motor DC karena terdapat *transistor transistor Logic* (TTL) dan gerbang NAND [21] [22]. Menurut Arief Pratama Zanofa, et.al pada penelitiannya yang berjudul Pintu Gerbang Otomatis Berbasis Mikrokontroler Arduino Uno R3 [23], selain sebagai pengatur arah putaran motor DC, driver L298N juga berfungsi sebagai penguat arus untuk motor DC. Motor DC umumnya membutuhkan arus 250 mA untuk bekerja, namun beberapa IC ATmega milik Arduino tidak mampu memberikan output arus melebihi kebutuhan motor DC. Dalam penelitian ini driver motor L289N digunakan untuk mengatur arah putaran dari motor DC.



Gambar 2. 7 Driver L298N  
Sumber : ([www.iberobotics.com](http://www.iberobotics.com))

## 2.11 AC to DC Power Supply 12 V

Catu Daya atau *Power Supply* AC to DC merupakan sebuah perangkat yang mampu mengubah tegangan ac menjadi tegangan dc, dan menyuplai daya listrik ke berbagai peralatan elektronika yang membutuhkan. Catu Daya tersusun dari beberapa komponen seperti Transformator, diode, kapasitor, dan IC Regulator[24]. Dalam penelitian ini, Catu daya berfungsi untuk mendistribusikan daya ke beberapa komponen seperti motor DC, Arduino, dan beberapa sensor.



Gambar 2. 8 Catu Daya atau *Power Supply*  
Sumber ([www.circuitspecialists.com](http://www.circuitspecialists.com))

### 2.12 *Thunkable*

*Thunkable* merupakan sebuah tools yang berfungsi untuk membuat aplikasi android. Dengan *Thunkable*, user atau pengguna dapat berkreasi membuat aplikasi sederhana maupun aplikasi yang lebih kompleks. *Thunkable* terdiri dari 2 bagian utama, yaitu bagian desain yang memuat tampilan untuk membuat *interface* atau tampilan antarmuka. Bagian utama yang kedua adalah bagian block code, bagian ini berfungsi untuk media penempatan logika algoritma untuk tampilan *interface* yang sudah dibuat pada bagian desain [25].

Dalam penelitian ini, *Thunkable* digunakan untuk membuat tampilan *interface* dan aplikasi alat penghitung kertas otomatis. Tampilan aplikasi terdiri dari *Button Start*, *Stop*, *Reset*, dan bar perhitungan jumlah kertas.



Gambar 2. 9 Logo *Thunkable*  
Sumber ([www.Thunkable.com](http://www.Thunkable.com))

### 2.13 *Firebase*

*Firebase* merupakan suatu fasilitas atau layanan yang dapat memberikan kemudahan bagi *developer* atau pengembang untuk mengembangkan sebuah aplikasi. Fitur *Firebase* yang digunakan dalam penelitian ini adalah *Firebase*

*Realtime Database. Firebase Realtime Database* adalah salah satu jenis *Database* yang mampu menyimpan data dalam bentuk *JSON* dan langsung disinkronkan secara *realtime* ke setiap pengguna *Database* melalui cloud [26].



Gambar 2. 10 Logo *Firebase*  
Sumber ([www.dicoding.com](http://www.dicoding.com))

#### 2.14 *Internet of Things (IoT)*

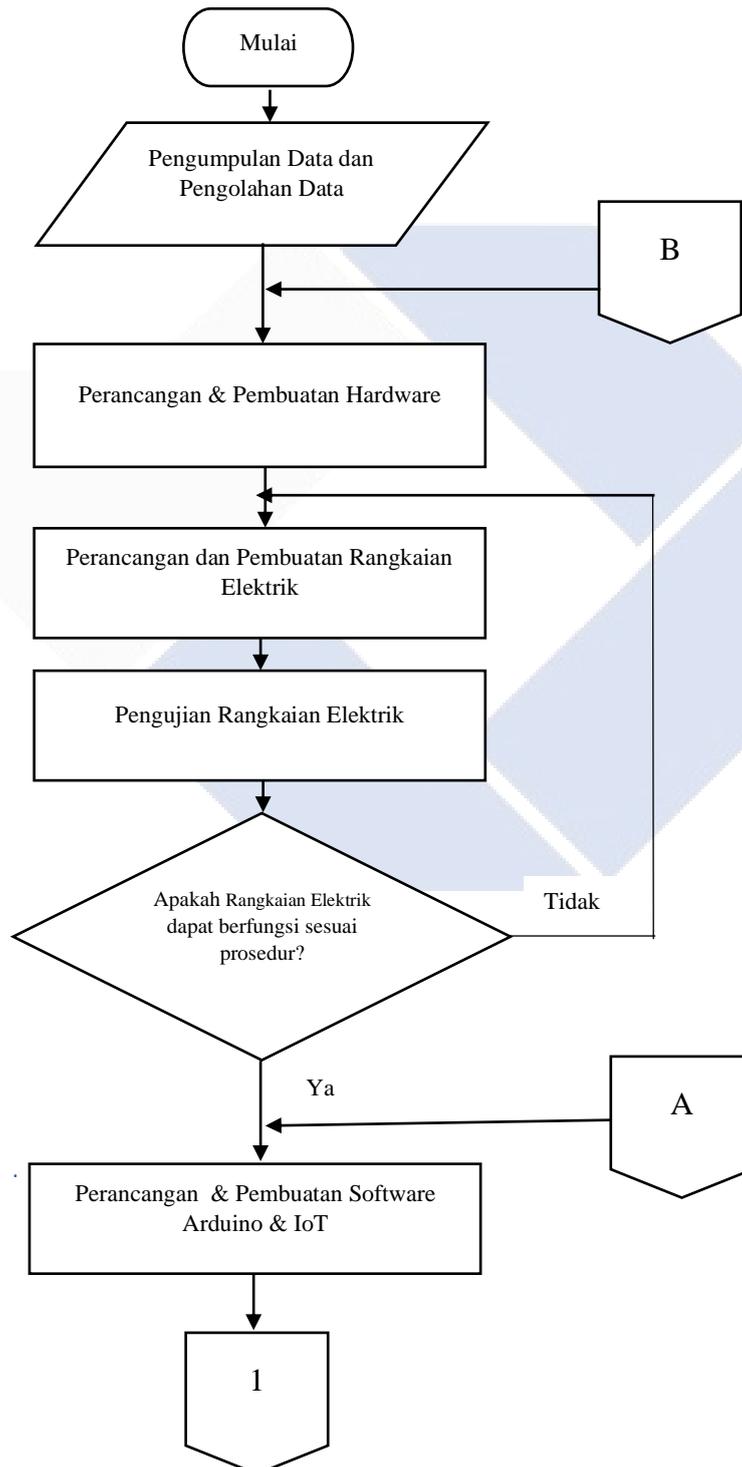
*Internet of Things (IoT)* merupakan sebuah konsep yang dapat menghubungkan setiap objek fisik atau sebuah perangkat di kehidupan sehari-hari dengan jaringan internet [27]. Konsep dasar dari *Internet Of Things* mengacu pada 3 komponen utama, yaitu Barang fisik yang dilengkapi modul IoT, lalu media penghubung internet seperti *Router* atau Modem, dan *Cloud Data Center* sebagai tempat penyimpanan *Database* dan aplikasi. Cara kerja dari *Internet Of Things* adalah setiap perangkat ataupun benda yang terhubung dengan internet khususnya *WiFi* bisa diakses kapanpun dimanapun kita berada [28]. Dalam penelitian ini *Internet Of Things* digunakan untuk penghubung antara alat penghitung kertas otomatis dengan aplikasi pada *Thinkable*.

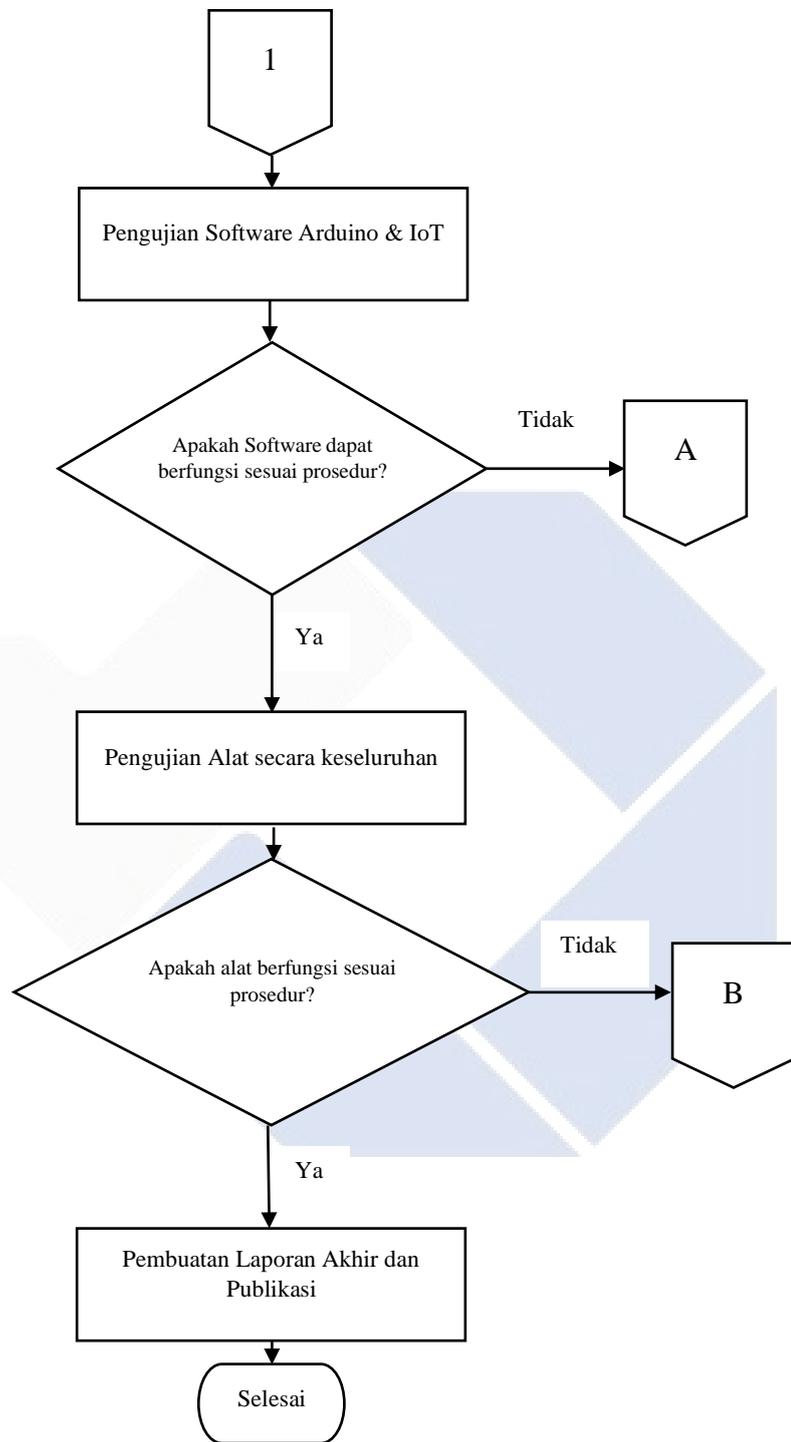


Gambar 2. 11 *Internet of Things*  
Sumber : ([www.hw-group.com](http://www.hw-group.com))

### BAB III METODE PELAKSANAAN

Tahap pelaksanaan proyek akhir yang berjudul Alat Penghitung Kertas Otomatis Berbasis IoT ini dapat dirangkum melalui gambar *FlowChart* sebagai berikut:





Gambar 3. 1 *FlowChart* Metode Pelaksanaan Proyek Akhir

### **3.1 Pengumpulan dan Pengolahan data**

Pada tahap ini, pengumpulan data diperoleh melalui pencarian kajian literatur yang berkaitan dengan judul proyek akhir ini. Metode pengumpulan data yang digunakan pada tahapan ini terdiri dari pengumpulan data secara langsung atau data primer dan pengumpulan data secara tidak langsung atau sekunder. Pengumpulan data secara primer dilakukan dengan melakukan bimbingan dan juga konsultasi mengenai proyek akhir dengan dosen pembimbing. Lalu pengumpulan data secara sekunder dilakukan dengan mencari referensi dari beberapa jurnal baik lingkup nasional maupun internasional yang berkaitan dengan proyek akhir ini. Selain itu dilakukan juga survei ke beberapa tempat percetakan dan juga toko ATK yang berada di kecamatan Sungailiat. Hasil dari survei yang sudah dilakukan adalah semua tempat percetakan juga toko ATK masih menggunakan sistem manual dalam proses perhitungan kertas. Tahap selanjutnya yaitu membuat ringkasan dari keseluruhan data yang sudah diperoleh dan akan digunakan sebagai referensi atau landasan untuk mengembangkan alat ataupun penelitian yang sudah dilakukan sebelumnya.

### **3.2 Perancangan *Hardware* dan Aplikasi**

Pada tahap perancangan *hardware* dilakukan perancangan untuk konstruksi alat penghitung kertas dan juga pengaturan tata letak untuk sensor LDR dan juga LED. Sementara itu, pada tahap perancangan Aplikasi dilakukan desain pengiriman data melalui *Firestore Realtime Database*, pemrograman untuk Arduino Uno dan juga mendesain tampilan *interface* dari aplikasi monitoring penghitung kertas.

#### **3.2.1 Perancangan *Hardware***

Desain *hardware* pada alat penghitung kertas berbasis IoT terdiri dari 2 jenis. Pertama desain konstruksi alat penghitung kertas otomatis dan kedua adalah desain perangkat elektrik. Proses rancang bangun konstruksi alat dilakukan dengan aplikasi *sketch up*. Pada tahapan ini ditentukan jenis bahan yang akan dipilih dan ukuran dari alat penghitung kertas otomatis.

Pada tahapan desain konstruksi alat, didapat bahwa bahan yang digunakan adalah akrilik dan kayu triplek. Bagian bawah alat menggunakan kayu triplek dengan ukuran 43 cm x 45 cm. Lalu bagian samping alat menggunakan akrilik dengan tinggi 15 cm. Berikut rancangan atau desain alat penghitung kertas otomatis pada aplikasi *sketch up*.

Tahap perancangan rangkaian elektrik bertujuan untuk memetakan *wiring* pada rangkaian elektrik yang akan digunakan untuk penentuan tata letak komponen elektrik. Perancangan menggunakan aplikasi *fritzing*.

### **3.2.2 Perancangan Aplikasi**

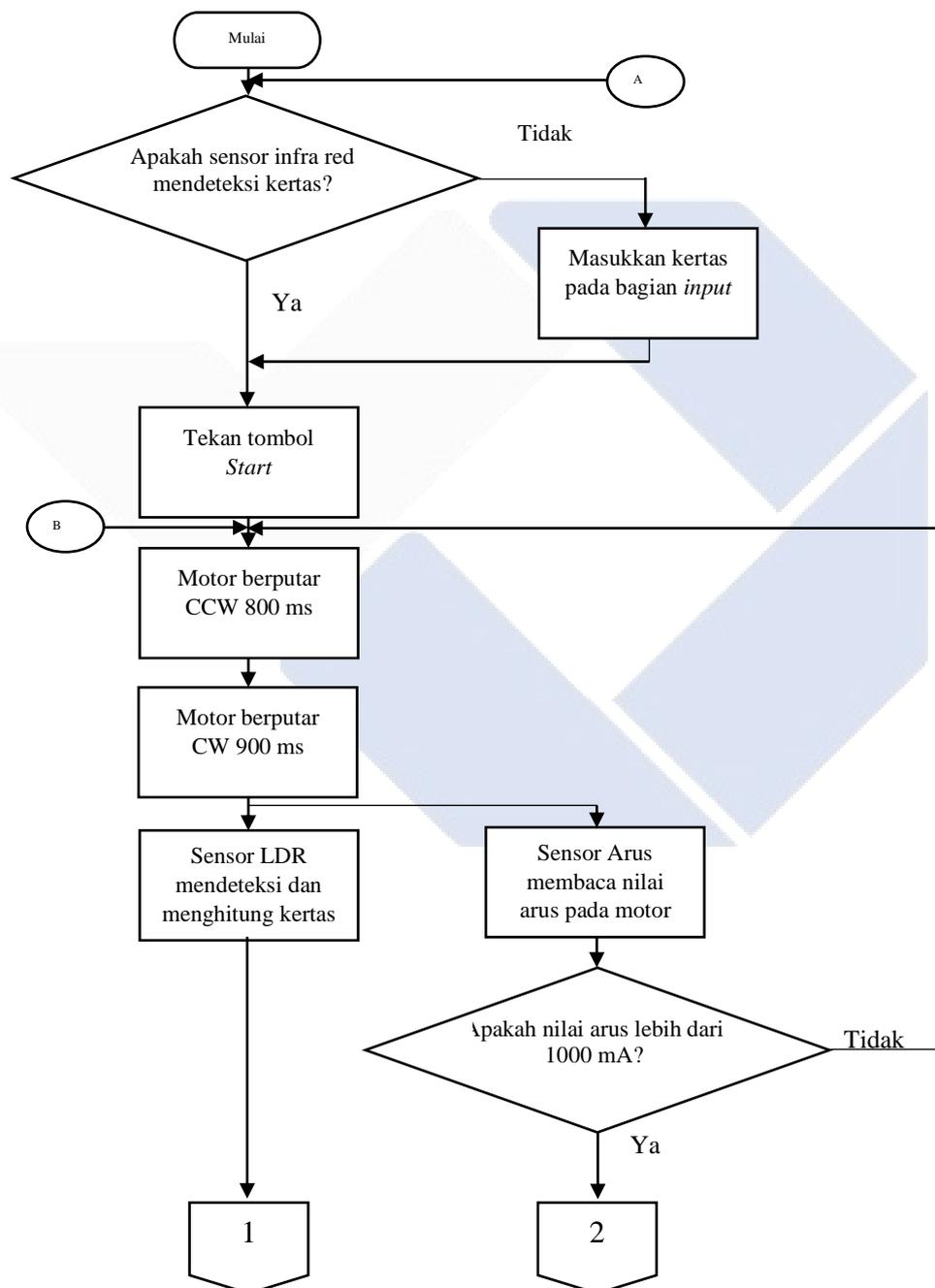
Desain Aplikasi pada alat penghitung kertas otomatis meliputi pemrograman Arduino Uno untuk pengolahan data dari sensor, dan proses transaksi data dengan NodeMCU ESP32 melalui komunikasi serial. Selanjutnya pemrograman NodeMCU ESP32 untuk melakukan pengiriman data pembacaan sensor ke *Firebase Realtime Database*. Tahapan terakhir mendesain aplikasi tampilan *interface* pada smartphone menggunakan *Thunkable* untuk proses *Start* dan *Stop* alat, dan juga proses *Reset* perhitungan dan menampilkan hasil perhitungan kertas.

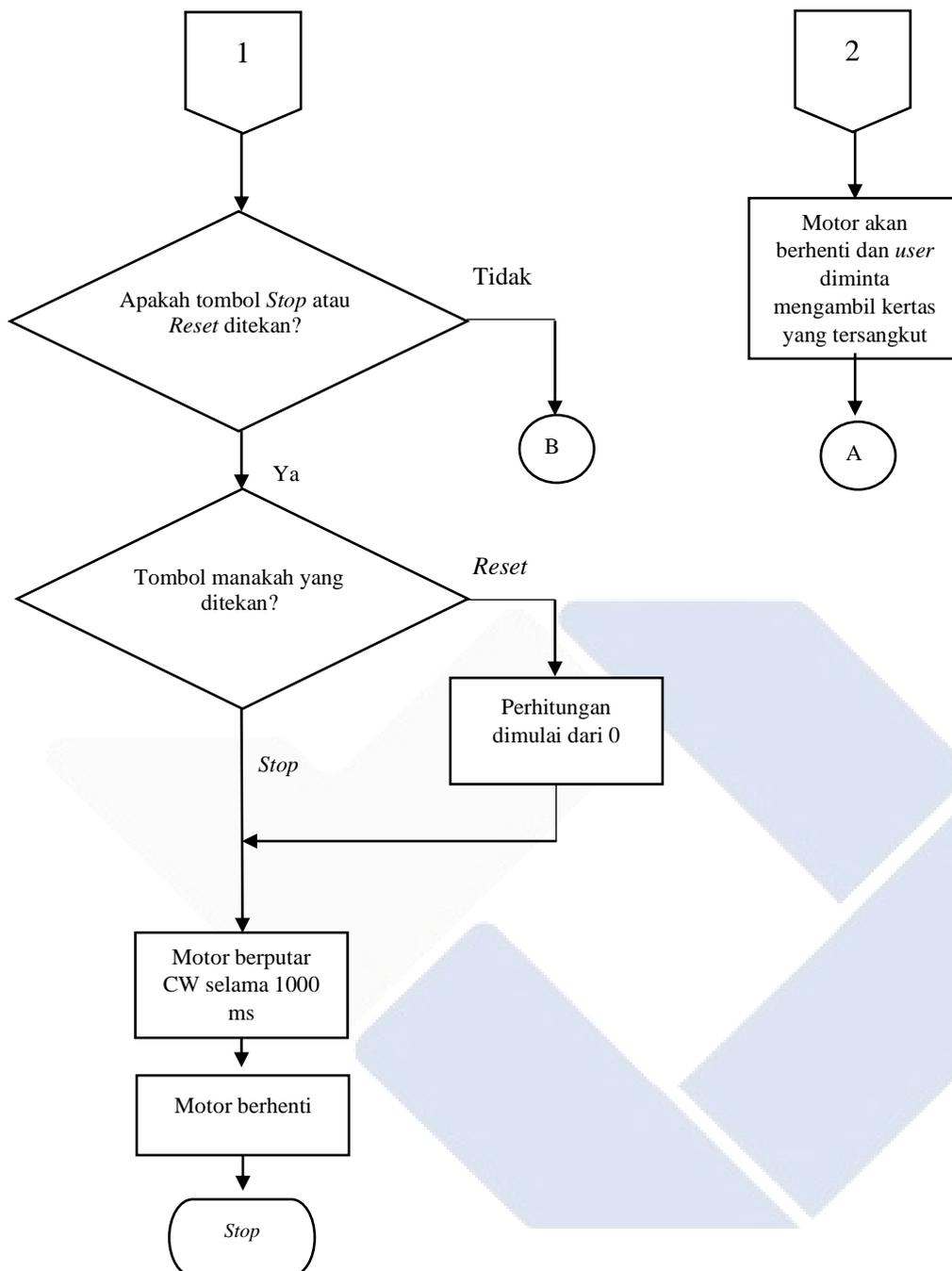
### **3.3 Perakitan hardware dan konstruksi alat**

Pada tahap ini, dilakukan perakitan konstruksi alat penghitung kertas otomatis mengacu pada desain alat yang sudah dibuat. Proses perakitan diawali dengan pemotongan kayu triplek dan akrilik sesuai ukuran yang sudah ditentukan, lalu merekatkan bagian bagian yang sudah dipotong. Dilanjutkan dengan penempatan rangkaian elektrik dan perangkat elektrik pada boks panel, lalu ditempatkan pada bagian belakang alat penghitung kertas. Proses dilanjutkan dengan pemberian warna pada alat penghitung kertas otomatis. Warna yang dipilih adalah hitam. Pemberian warna hitam memiliki tujuan agar cahaya yang berasal dari luar alat (faktor Eksternal) tidak mengganggu pembacaan sensor LDR dalam melakukan perhitungan dan pendeteksian kertas.

### 3.4 Perakitan Aplikasi

Pada tahapan ini diawali dengan pemrograman untuk mikrokontroler. Pemrograman dilakukan untuk membaca data sensor arus ACS712, sensor LDR dan sensor infrared. Lalu pemrograman untuk jalannya siklus motor DC dan LCD 16 x 2. Lalu dilanjutkan dengan pemrograman NodeMCU ESP 32 untuk melakukan transaksi data dengan Arduino Uno dan *Firestore Realtime Database*.





Gambar 3. 2 *FlowChart* Program Arduino

Tahap selanjutnya adalah perakitan aplikasi pada *Thunkable* untuk pembuatan aplikasi penghitung kertas otomatis. Pembuatan aplikasi ditujukan untuk media pengoperasian alat penghitung kertas otomatis. *Content* yang akan ditampilkan pada aplikasi adalah *Button Start*, *Stop* dan *Reset*, lalu jumlah perhitungan kertas, status dari motor DC, status *warning* arus, dan status *detector* dari sensor infra merah.

### **3.5 Pengujian Aplikasi dan *Hardware***

Tahap pengujian ini dilakukan untuk mengetahui apakah *software* yang sudah di rancang dan dibuat dapat menjalankan alat dengan tepat dan benar. Dan juga dilakukan pengujian terhadap *hardware* khususnya perangkat elektrik untuk mengetahui apakah terdapat kesalahan perangkaian ataupun kerusakan dari perangkat elektrik yang digunakan.

#### **3.5.1 Pengujian Aplikasi**

Pada tahap ini dilakukan pengujian terhadap aplikasi yang telah dibuat di *Thinkable*, pengujian dikatakan berhasil apabila aplikasi tersebut dapat mengontrol *Start* dan *Stop* alat penghitung kertas, dan juga dapat menampilkan hasil pembacaan dan perhitungan berdasarkan data yang telah di proses oleh Arduino Uno.

#### **3.5.2 Pengujian *Hardware***

Pada tahap ini dilakukan pengujian terhadap kinerja perangkat elektrik yang digunakan. Tahap ini diawali dengan pengujian terhadap Arduino Uno. Pengujian dikatakan berhasil jika Arduino Uno bisa mengatur kondisi *Start* dan *Stop* alat penghitung kertas otomatis, mengolah data pembacaan sensor LDR, sensor arus ACS712, dan sensor infra red, menampilkan hasil perhitungan ke LCD 16x2, mengirim data pembacaan sensor ke NodeMCU ESP32, dan menerima data dari NodeMCU ESP32 melalui komunikasi serial.

Tahap selanjutnya dilakukan pengujian terhadap setiap komponen sensor. Pengujian dilakukan dengan melakukan kalibrasi hasil pembacaan sensor dengan alat ukur. Metode yang digunakan pada pengujian adalah dengan mengacu ke linieritas antara nilai output ADC sensor dengan hasil pembacaan alat ukur. Persamaan regresi yang didapat dari garis linier akan digunakan sebagai persamaan untuk konversi pembacaan sensor ke nilai output (Arus untuk sensor arus dan intensitas cahaya untuk sensor LDR). Pembacaan sensor hasil dari persamaan akan dibandingkan dengan pembacaan alat ukur, dan didapat nilai *error* dan akurasi yang akan menjadi acuan dari pengujian sensor.

### **3.6 Pengujian Keseluruhan Sistem**

Pada tahap ini, dilakukan pengujian terhadap keseluruhan sistem alat penghitung kertas otomatis. Pengujian diawali dengan pengujian proses IoT, apakah alat penghitung kertas otomatis dapat di monitoring dan di kontrol melalui aplikasi di *smartphone*. Lalu pengujian kinerja dari alat penghitung kertas otomatis, apakah motor dc dan *roller* dapat menarik kertas dan mendorong ke proses perhitungan, pengujian perhitungan oleh sensor LDR dengan hasil perhitungan manual, dan pengujian sensor arus ACS712 sebagai pendeteksi arus pada motor dc. Pada tahap ini juga dilakukan pengujian program pada Arduino ide, apakah program yang dibuat sudah tepat sesuai dengan sistematis dan prosedur alat penghitung kertas yang sudah didesain.

### **3.7 Analisa Data**

Pada tahap ini, perbandingan setiap data dianalisis untuk membuktikan apakah sistem *hardware* dan *software* yang sudah dibuat sudah sesuai dengan rancangan yang sudah dibuat. Jika masih ditemukan kekurangan ataupun masalah pada sistem alat, maka akan dilakukan tahapan perancangan ulang ataupun *Problem Solving* untuk mengatasi permasalahan baik pada *hardware* maupun *software*.

### **3.8 Pembuatan Laporan Akhir dan Publikasi**

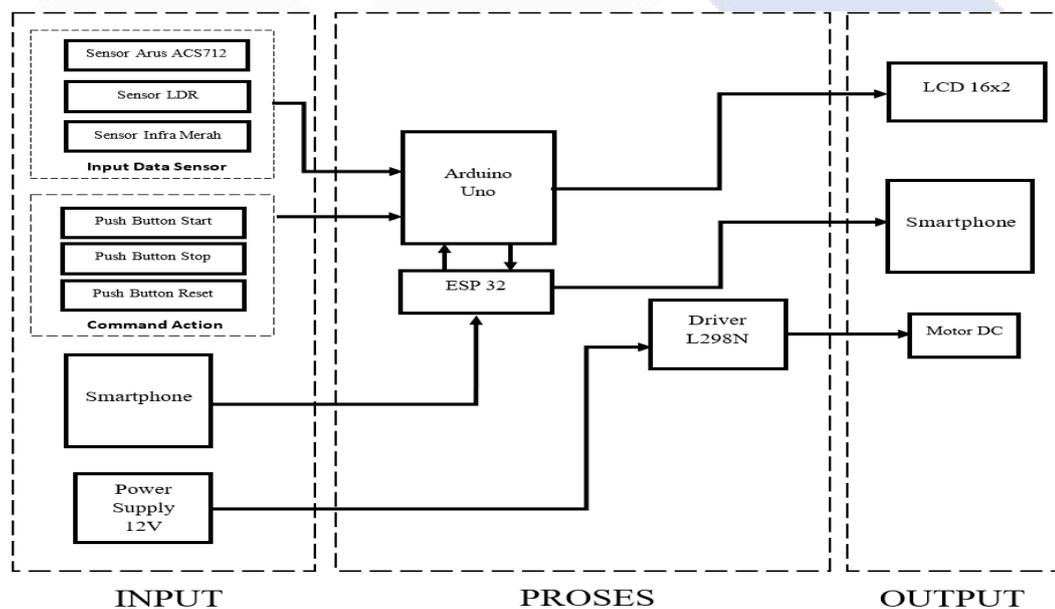
Setelah alat penghitung kertas otomatis dapat bekerja dengan baik dan berhasil dibuat, maka selanjutnya adalah pembuatan laporan akhir dan publikasi dari hasil pembuatan tugas akhir ini. Pembuatan Laporan ditujukan untuk merekap data keseluruhan proyek akhir. Adapun proses publikasi dilakukan dengan media *paper*, jurnal, atau karya tulis lainnya. Hasil karya tulis yang sudah dibuat akan di publikasi ke penerbit jurnal yang memenuhi persyaratan.

## BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai proses pembuatan alat penghitung kertas otomatis sesuai dengan penjabaran pada *FlowChart* atau diagram alir pengerjaan proyek akhir (Gambar 3.1).

### 4.1 Deskripsi Alat

Alat Penghitung Kertas Otomatis berbasis IoT merupakan alat yang mampu melakukan *monitoring* dan juga kontrol *Start* dan *Stop* perhitungan kertas melalui *smartphone* maupun manual menggunakan *push Button* dan LCD 16x2. Berikut ini adalah blok diagram untuk alat penghitung kertas otomatis berbasis IoT.



Gambar 4.1 Blok diagram sistematis alat

Alat penghitung kertas otomatis ini dijalankan dengan beberapa komponen. Sensor LDR digunakan sebagai penghitung kertas melalui pembacaan digital sensor, dan juga mendeteksi jumlah kertas yang tertarik melalui pembacaan analog. Sensor infra merah digunakan sebagai pendeteksi kertas pada bagian *input* alat. Sensor arus ACS712 digunakan untuk proteksi alat ketika ada kertas yang tersangkut pada saat proses penarikan, dengan cara mendeteksi arus pada motor

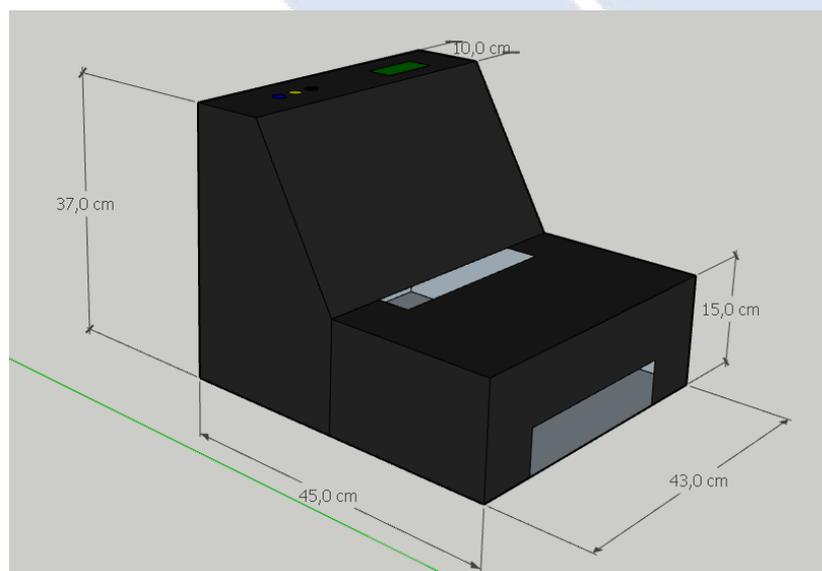
DC. Jika ada arus berlebih yang diakibatkan kertas tersangkut maka sistem akan berhenti. Arduino Uno digunakan untuk mengirim data hasil pembacaan sensor ke *Firebase* melalui komunikasi serial dengan NodeMCU ESP32. Arduino Uno juga digunakan untuk menerima data masukan yang berasal dari NodeMCU ESP32 untuk menjalankan alat dengan perintah yang berasal dari aplikasi pada *smartphone*. User juga bisa memantau hasil perhitungan kertas melalui aplikasi.

## 4.2 Perancangan dan Pembuatan *Hardware*

Pada tahapan perancangan dan pembuatan *Hardware*, dilakukan pembuatan konstruksi alat penghitung kertas dengan menggunakan bahan akrilik dan juga perangkaian rangkaian elektrik untuk pengontrolan motor dc dan pembacaan Arduino.

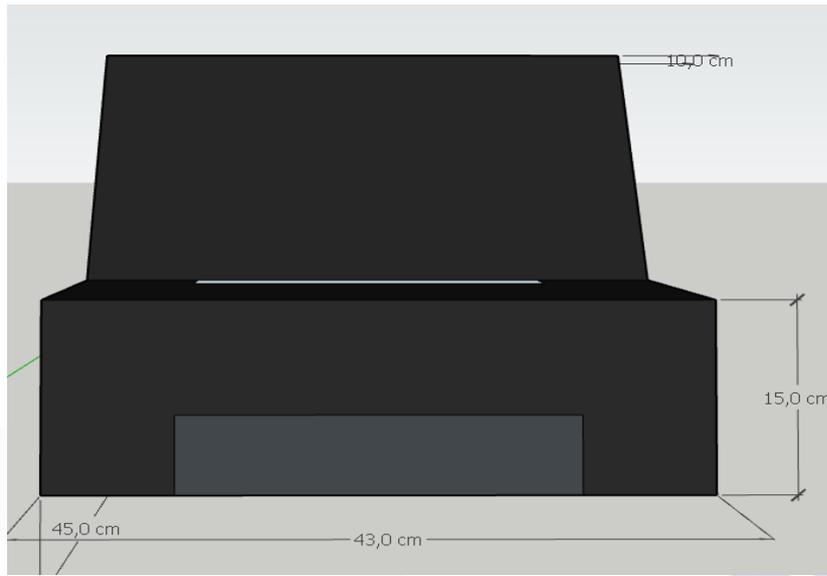
### 4.2.1 Perancangan dan Pembuatan Konstruksi Alat

Pada proses perancangan konstruksi alat, dilakukan desain konstruksi menggunakan aplikasi *Sketch Up*. Ukuran dari alat penghitung kertas ini adalah panjang 43 cm, Lebar 45 cm dan tinggi 37 cm pada bagian belakang dan 15 cm pada bagian depan. Bahan yang digunakan adalah akrilik dengan ketebalan 3 mm dan kayu triplek untuk konstruksi alat penghitung kertas. Berikut adalah desain alat penghitung kertas otomatis pada aplikasi *Sketch Up*.

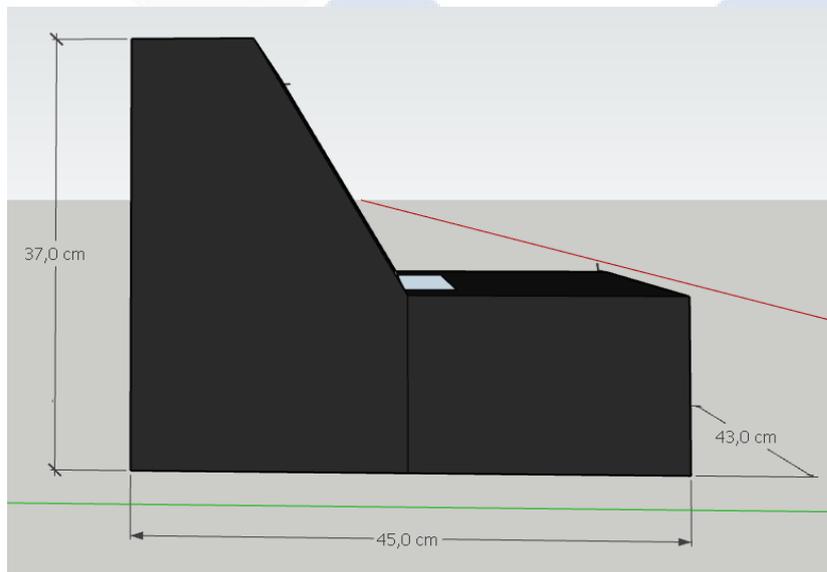


Gambar 4. 2 Desain Konstruksi Alat Penghitung Kertas

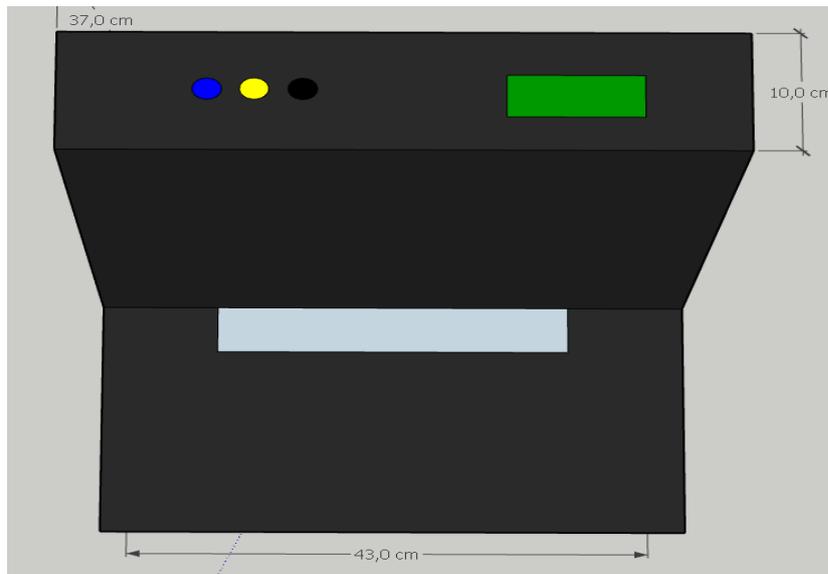
Peletakan komponen LCD 16x2 sebagai *display* perhitungan dan juga push *Button* berada di bagian atas alat. Tujuan dari penempatan ini adalah untuk memudahkan pengoperasian alat penghitung kertas secara manual. Berikut adalah tampak depan, samping dan atas dari desain alat penghitung kertas otomatis.



Gambar 4. 3 Desain Alat Tampak Depan



Gambar 4. 4 Desain Alat Tampak Samping



Gambar 4. 5 Desain Alat Tampak Atas

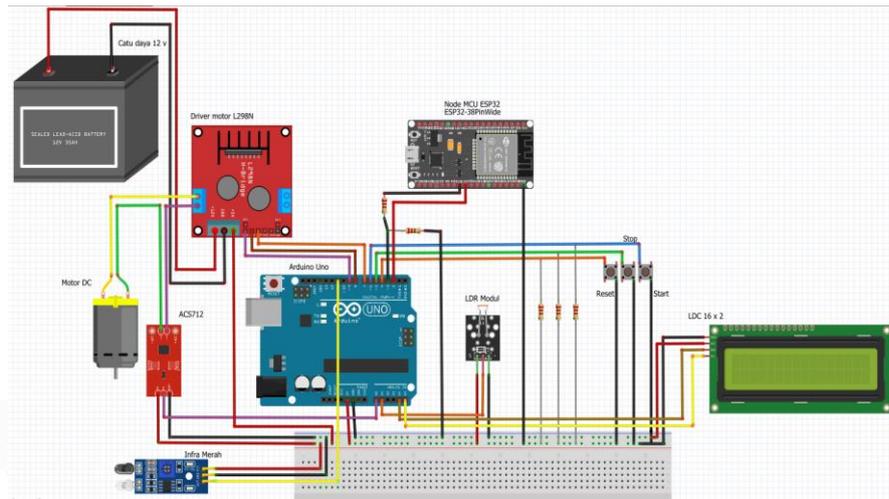
Tahapan pembuatan konstruksi alat dilakukan dengan membuat konstruksi alat penghitung kertas otomatis berdasarkan rancangan yang sudah dibuat sebelumnya. Bahan yang digunakan pada pembuatan konstruksi ini adalah akrilik 3 mm untuk bagian atas, depan, belakang dan samping dari alat penghitung kertas otomatis. Bagian bawah alat menggunakan kayu triplek. Pemilihan warna hitam ditujukan agar cahaya yang berasal dari luar tidak mengganggu pembacaan dari sensor LDR. Berikut adalah hasil pembuatan konstruksi alat penghitung kertas.



Gambar 4. 6 Konstruksi Alat Penghitung Kertas Otomatis Berbasis IoT

#### 4.2.2 Perancangan dan Pembuatan Rangkaian Elektrik

Tahap perancangan rangkaian elektrik bertujuan untuk memetakan *wiring* pada rangkaian elektrik yang akan digunakan untuk penentuan tata letak komponen elektrik. Perancangan menggunakan aplikasi *fritzing*. Berikut desain rancangan rangkaian elektrik.



Gambar 4. 7 Desain Rangkaian Elektrik

Tahap selanjutnya dilakukan dengan membuat rangkaian elektrik yang berdasarkan rancangan yang sudah dibuat. Pada tahap ini dilakukan pengkabelan untuk komponen yang digunakan seperti Arduino Uno, sensor infra merah, sensor LDR, sensor arus ACS712, Node MCU ESP32, *Push Button*, LCD 16x2, Driver L298N, Catu daya 12 V, rangkaian *Pull up push Button*, dan rangkaian pembagi tegangan pin Node MCU ESP 32.

#### 4.3 Pembuatan Aplikasi PCT-1

Pembuatan aplikasi PCT-1 ditujukan untuk *interface* antara pengguna dan juga alat penghitung kertas otomatis dengan media *smartphone* menggunakan teknologi Internet of Things. Aplikasi PCT-1 dapat melakukan pengoperasian alat penghitung kertas dan juga menampilkan hasil perhitungan kertas. Aplikasi PCT-1 dirangkai pada aplikasi *Thunkable*. Berikut adalah tahapan pembuatan aplikasi PCT-1.

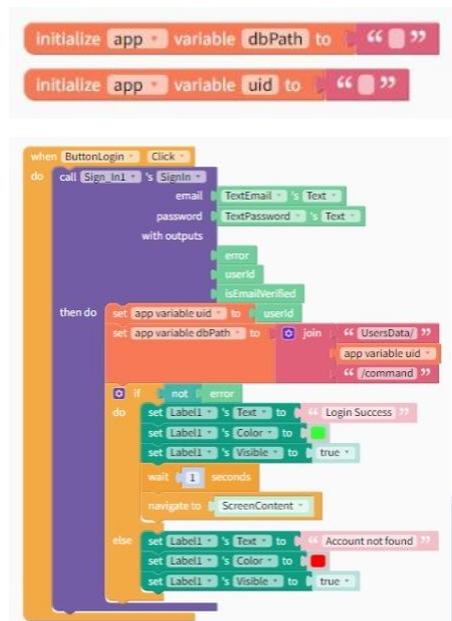
### 4.3.1 Pembuatan *Login Screen*

Hal pertama yang harus diperhatikan dalam pembuatan aplikasi adalah desain yang nantinya akan tampil pada aplikasi tersebut (*User Interface*). Begitupun sama halnya dengan pembuatan *Login screen*. Disini kami mendesain tampilan yang cukup ringkas dan mudah dimengerti oleh pengguna umum. Secara umum, pada *Login screen* ini dibagi menjadi 3 kondisi *screen*, yakni, tampilan *Login*, tampilan gagal *Login*, dan tampilan berhasil *Login*.



Gambar 4. 8 Tampilan Awal Aplikasi

Gambar 4.8 merupakan tampilan paling awal dari aplikasi PCT-1. Elemen yang diperlukan pada *screen* ini adalah satu buah gambar, empat buah label (label *email*, label *text email*, label *password*, dan label *text password*), dan satu buah tombol (*Login Button*).



Gambar 4. 9 Logic Block Tampilan Awal Aplikasi

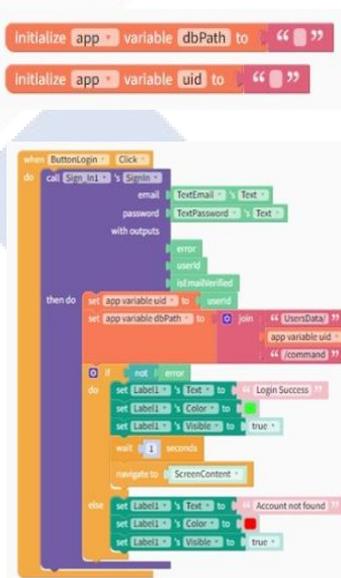
Untuk *Logic Block* tampilan awal aplikasi, yang pertama pengenalan/inisialisasi *variable dbPath* (keseluruhan data yang akan ditampilkan) dan pengenalan/inisialisasi *variable uid* (*user id* yang telah didaftarkan di *Firestore*), kemudian ketika *Button Login* di tekan, lakukan pemanggilan sesi *sign in* pada *Firestore* dan sekaligus konfirmasi alamat *email* dan *password* yang ditulis didalam *text*, dan mempersiapkan pula *output user id* “*error*” / “*apakah email terverifikasi*”. Kemudian memanggil *variable uid* (*userID*) pada *Firestore* yang telah didaftarkan kemudian disinkronkan dengan *userID* yang tertulis pada *text* tampilan. Selain itu memanggil pula *variable dbPath* yang mana merupakan *address variable* tersebut di *Firestore* yakni *UserData/command*. Jika sinkronisasi *uid* tidak *error*, *set text info* menjadi “*Login success*” dan merubah warna *text* menjadi hijau, sekaligus memunculkan tulisan *text info*. Setelah itu sistem diatur untuk menunggu 1 detik dan *screen* pindah ke *screen content*. Jika sinkronisasi *uid error*, *set text info* menjadi “*account not found*” dan merubah warna *text* menjadi merah, sekaligus memunculkan tulisan *text info*.

#### 4.3.1.1 Tampilan *Login* Gagal



Gambar 4. 10 Tampilan Aplikasi Saat Gagal *Login*

Gambar 4.10 merupakan tampilan *Login* gagal dari aplikasi PCT-1. Elemen yang diperlukan pada *screen* ini adalah satu buah gambar, lima buah label (label *email*, label *text email*, label *password*, label *text password*, dan label informasi), dan satu buah tombol (*Login Button*).



Gambar 4. 11 *Logic Block* Gagal *Login*

Sama seperti *tampilan awal pada Login screen*, untuk *Logic Block* pada tampilan *Login* gagal sendiri adalah pengenalan/inisialisasi *variable dbPath* (keseluruhan data yang akan ditampilkan) dan pengenalan/inisialisasi *variable uid* (*user id* yang telah didaftarkan di *Firebase*), kemudian ketika *Button Login* di

tekan, lakukan pemanggilan sesi *sign in* pada *Firestore* dan sekaligus konfirmasi alamat *email* dan *password* yang ditulis didalam *text*, dan mempersiapkan pula *output user id "error" / "apakah email terverifikasi"*. Kemudian memanggil *variable uid (userID)* pada *Firestore* yang telah didaftarkan kemudian disinkronkan dengan *userID* yang tertulis pada *text* tampilan. Selain itu memanggil pula *variable dbPath* yang mana merupakan *address variable* tersebut di *Firestore* yakni *UserData/command*. Jika sinkronisasi *uid* tidak *error*, set *text info* menjadi "*Login success*" dan merubah warna *text* menjadi hijau, sekaligus memunculkan tulisan *text info*. Setelah itu sistem diatur untuk menunggu 1 detik dan *screen* pindah ke *screen content*. Jika sinkronisasi *uid* *error*, set *text info* menjadi "*account not found*" dan merubah warna *text* menjadi merah, sekaligus memunculkan tulisan *text info*.

#### 4.3.1.2 Tampilan *Login* Berhasil



Gambar 4. 12 Tampilan Aplikasi Saat Berhasil *Login*

Gambar 4.12 merupakan tampilan *Login* berhasil dari aplikasi PCT-1. Elemen yang diperlukan pada *screen* ini adalah satu buah gambar (kami mencantumkan desain gambar alat kami), lima buah label (label *email*, label *text email*, label *password*, label *text password*, dan label informasi), dan satu buah tombol (*Login Button*).



Gambar 4. 13 *Logic Block Berhasil Login*

Sama seperti tampilan awal pada Login screen, untuk *Logic Block* pada tampilan Login berhasil sendiri adalah pengenalan/inisialisasi *variable dbPath* (keseluruhan data yang akan ditampilkan) dan pengenalan/inisialisasi *variable uid* (*user id* yang telah didaftarkan di *Firebase*), kemudian ketika *Button Login* di tekan, lakukan pemanggilan sesi *sign in* pada *Firebase* dan sekaligus konfirmasi alamat *email* dan *password* yang ditulis didalam *text*, dan mempersiapkan pula *output user id* “*error*” / “*apakah email terverifikasi*”. Kemudian memanggil *variable uid* (*userID*) pada *Firebase* yang telah didaftarkan kemudian disinkronkan dengan *userID* yang tertulis pada *text* tampilan. Selain itu memanggil pula *variable dbPath* yang mana merupakan *address variable* tersebut di *Firebase* yakni *UserData/command*. Jika sinkronisasi *uid* tidak *error*, set *text info* menjadi “*Login success*” dan merubah warna *text* menjadi hijau, sekaligus memunculkan tulisan *text info*. Setelah itu sistem diatur untuk menunggu 1 detik dan *screen* pindah ke *screen content*. Jika sinkronisasi *uid* *error*, set *text info* menjadi “*account not found*” dan merubah warna *text* menjadi merah, sekaligus memunculkan tulisan *text info*.

### 4.3.2 Pembuatan *Logic Block*

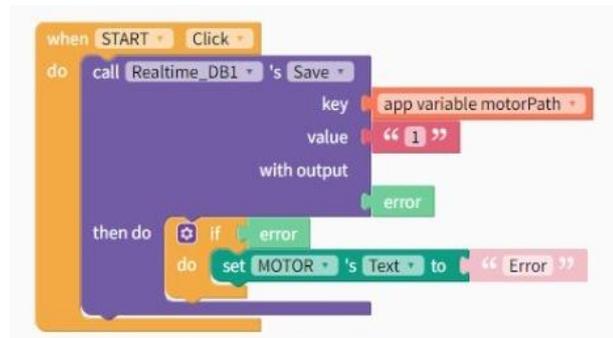
*Logic Block* merupakan tampilan yang berisi tampilan data alat penghitung kertas sekaligus kontrol yang bisa langsung pengguna pakai saat menggunakan aplikasi. Untuk data yang ditampilkan yakni, data hitung kertas, status motor, status *detector*, *warning* (peringatan arus berlebih pada motor). Untuk sistem kontrol yang dapat dikendalikan yakni, *Start*, *Stop*, dan *Reset*. Pada aplikasi yang sudah dirancang, apabila *detector* mendeteksi nilai 0 (*low*), maka tombol *Start* pada alat tidak bisa digunakan (*disable*). Ini dilakukan karena kertas yang dijadikan sebagai objek hitung harus berada pada alat hitung terlebih dahulu, maka dari itu kertas yang nantinya akan *trigger detector* apabila sudah diletakkan pada alat hitung.



Gambar 4. 14 Tampilan *Logic Block*

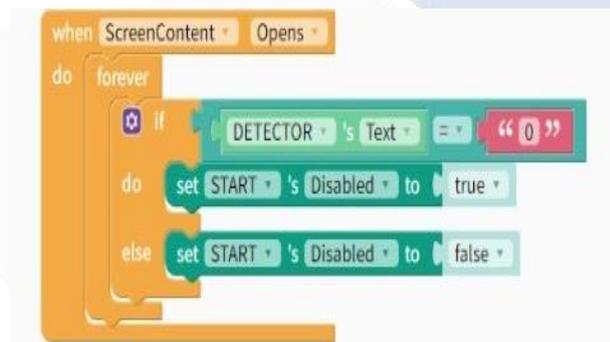
Komponen yang digunakan dalam mendesain tampilan *Logic Block* ini adalah 12 label (label judul, label perhitungan kertas, label kolom kertas, label status motor, label kolom motor, label status *detector*, label kolom *detector*, label status *warning*, label kolom *warning*, dan tiga buah label tanda panah) dan 3 *Buttons* (*Start*, *Stop*, dan *Reset*).

#### 4.3.2.1 Pembuatan *Logic* Ketika Tombol *Start* Ditekan



Gambar 4. 15 *Logic Block* Tombol *Start*

Pada saat *Start* ditekan, panggil *realtime* Database lalu simpan *variable motorPath* dengan *Value* 1 (pada *coding* di *arduino*, *Value* 1 pada *motor* terbaca *on*). Lalu jika *error*, *set* kolom *motor* pada tampilan menjadi *error*.



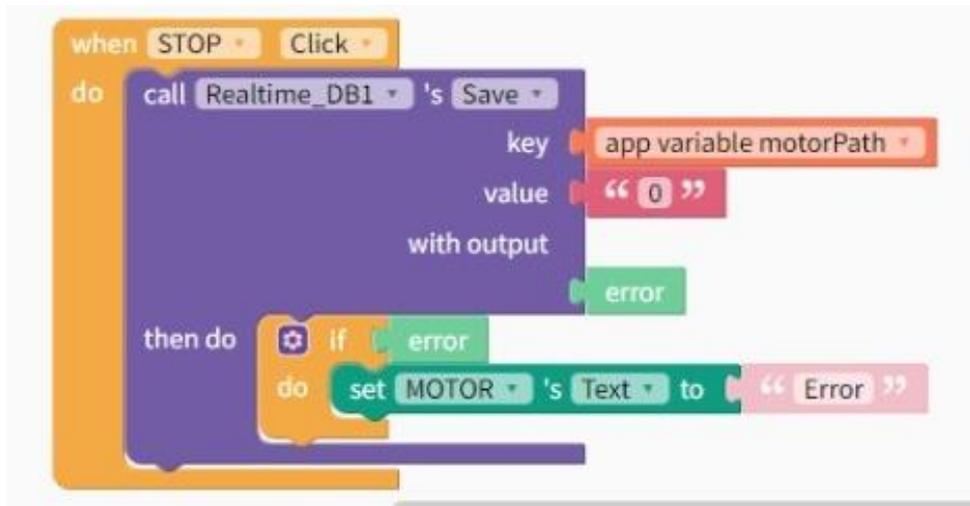
Gambar 4. 16 *Logic Block Value* Tombol *Start*

Pada saat *screencontent* dibuka, jika *detector* mendeteksi *Value* *motor* 0, maka tombol *Start* akan terbuka. Jika *Value* *motor* tidak sama dengan 0, maka tombol *Start* akan tertutup.



Gambar 4. 17 Tampilan *Logic Block* terhadap *Value* Tombol *Start*

#### 4.3.2.2 Pembuatan *Logic* Ketika Ditekan Tombol *Stop*



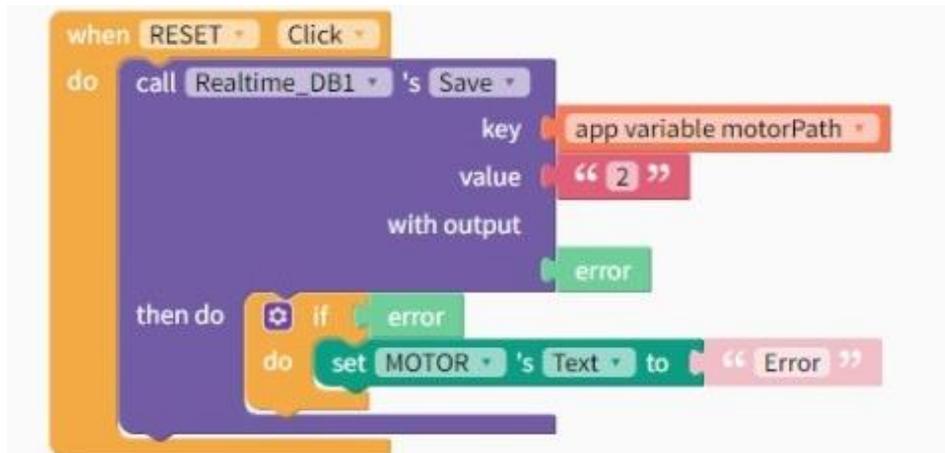
Gambar 4. 18 *Logic Block* Tombol *Stop*

Ketika *Stop* ditekan, panggil *realtime Database* lalu simpan *variable motorPath* dengan *Value 0* (pada *coding* di *arduino*, *Value 0* pada *motor* terbaca *off*). Lalu jika *error*, *set* kolom *motor* pada tampilan menjadi *error*.



Gambar 4. 19 Tampilan *Logic Block* Saat *Button Stop* Ditekan

### 4.3.2.3 Pembuatan *Logic* Ketika Ditekan Tombol *Reset*



Gambar 4. 20 *Logic Block* Tombol *Reset*

Ketika *Reset* ditekan, panggil *realtime* Database lalu simpan *variable* *motorPath* dengan *Value* 2 (pada *coding* di *arduino*, *Value* 2 pada *motor* terbaca *off* sekaligus mengatur ulang perhitungan). Lalu jika *error*, *set* kolom *motor* pada tampilan menjadi *error*.



Gambar 4. 21 Tampilan *Logic Block* Saat *Button Reset* Ditekan

#### 4.3.2.4 Pembuatan Data- Data di Aplikasi



Gambar 4. 22 Path Data Firebase

Hal pertama yang dilakukan adalah membuat inisialisasi variable dari *Firestore* (*countPath*, *motorPath*, *detectorPath*, *warningPath*) untuk diambil datanya ke *Thunkable*. Sedangkan Data *Changedlist* pada *block* berfungsi menginisialisasi perubahan data pada keempat *Path* yang ada.



Gambar 4. 23 Logic Block Tampilan *countPath*

Pada saat *screenContent* dibuka, *Thunkable* menyiapkan *variable countPath* dengan *address /count*. Dan juga memanggil *realtime Database* dengan *key variable countPath*.



Gambar 4. 24 Logic Block Tampilan *motorPath*

Pada saat *screenContent* dibuka, *Thunkable* menyiapkan *variable motorPath* dengan *address /motor*. Dan juga memanggil *realtime Database* dengan *key variable motorPath*.



Gambar 4. 25 Logic Block Tampilan *detectorPath*

Pada saat *screenContent* dibuka, *Thunkable* menyiapkan *variable detectorPath* dengan *address /detector*. Dan juga memanggil *realtime Database* dengan *key variable detectorPath*.



Gambar 4. 26 Logic Block Tampilan *warningPath*

Pada saat *screenContent* dibuka, *Thunkable* menyiapkan *variable warningPath* dengan *address /warning*. Dan juga memanggil *realtime Database* dengan *key variable warningPath*.



Gambar 4. 27 Logic Block Keseluruhan *Data Path*

Block ini berguna untuk *update* data yang masuk dari *Firestore* sehingga bisa ditampilkan secara *realtime* pada *screen content*. Ketika *realtime Database* membaca perubahan data, maka :

1. Jika *key Path* memiliki nilai berupa *address count*, maka kolom jumlah kertas pada tampilan akan berubah sesuai dengan nilai *address count* yang masuk.
2. Jika *key Path* memiliki nilai berupa *address detector*, maka kolom *detector* pada tampilan akan berubah sesuai dengan nilai *address detector* yang masuk.
3. Jika *key Path* memiliki nilai berupa *address motor*, maka kolom motor pada tampilan akan berubah sesuai dengan nilai *address motor* yang masuk.
4. Jika *key Path* memiliki nilai berupa *address warning*, maka kolom warning pada tampilan akan berubah sesuai dengan nilai *address count* yang masuk

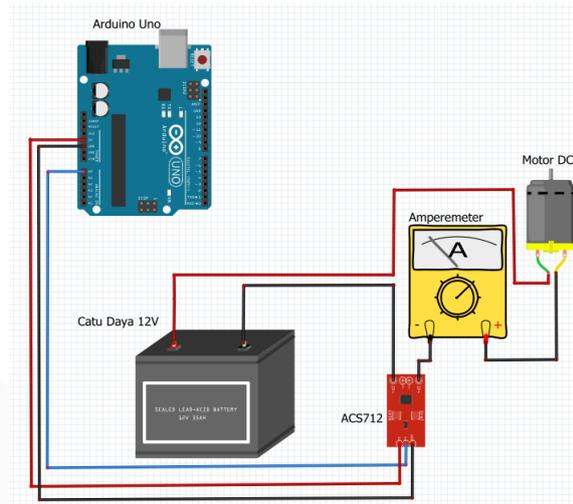
#### **4.4 Pengujian Sensor Arus ACS712**

Untuk proyek akhir ini, sensor arus ACS712 digunakan untuk membaca nilai arus yang terdapat pada motor DC. Sensor arus akan digunakan sebagai sistem proteksi ketika terdapat kertas yang tersangkut pada saat sistem penarikan kertas oleh *roller*. Pada saat terdapat kertas yang tersangkut di proses penarikan kertas, maka beban motor DC akan bertambah dan arus motor DC pun akan meningkat. Ketika nilai arus yang terdeteksi oleh sensor lebih dari nilai normal atau nilai yang sudah ditentukan, maka sistem akan langsung berhenti dan meminta *user* atau pengguna alat untuk menarik atau memperbaiki kertas yang tersangkut.

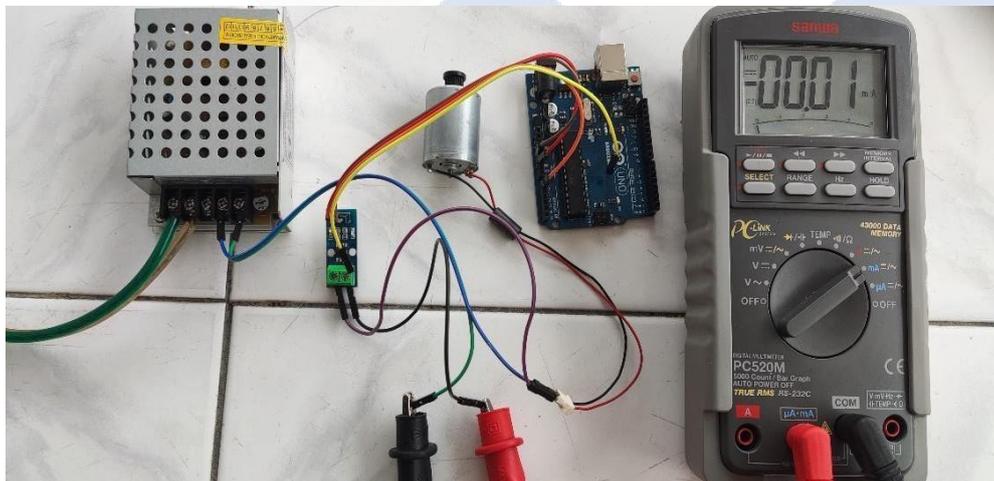
Sensor arus yang digunakan memiliki range pengukuran dari – 5 A sampai 5 A. Pengujian sensor dilakukan untuk mengetahui kemampuan pembacaan sensor terhadap arus DC. Pengujian sensor arus ACS712 dilakukan dengan menggunakan motor DC sebagai beban pengukuran. Tahapan pengujian sensor arus ACS712 adalah sebagai berikut.

#### 4.4.1 Perancangan dan pembuatan rangkaian sensor arus ACS712

Pada proses ini, dilakukan perancangan rangkaian elektrik untuk sensor arus ACS712 menggunakan aplikasi *fritzing*. Tujuan dari perancangan adalah untuk mendesain rancangan pengkabelan ataupun *wiring* masing-masing komponen. Berikut adalah hasil desain perancangan pada aplikasi *fritzing*.



Gambar 4.28 Desain Rangkaian Pengujian Sensor Arus ACS712



Gambar 4.29 Rangkaian Pengujian Sensor Arus ACS712

#### 4.4.2 Pengkalibrasian sensor arus ACS712

Pada proses ini, metode yang digunakan untuk pengkalibrasian sensor adalah metode regresi linier dengan menguji kelinieran dari pembacaan sensor terhadap hasil pembacaan pada amperemeter. Pembacaan sensor arus ACS712

yang berupa nilai ADC akan dikonversi menjadi nilai tegangan proporsional dengan rumus berikut:

$$\text{Tegangan (Vout)} = V_{ref} \times \frac{ADC}{1024} \quad [9] \quad (1)$$

Keterangan:

Vout = Tegangan keluaran sensor (V)

Vref = Tegangan maksimal sensor saat pembacaan maksimum

ADC = Pembacaan analog sensor (0 – 1024)

1024 = konversi 10 bit ke bilangan desimal

Berikut adalah program yang digunakan pada aplikasi Arduino IDE untuk mengolah hasil pembacaan sensor arus ACS712.

```
#define pinSensor A0

double tegangan;
double vpb = 0.00488758553;
double adc;

void setup() {
  Serial.begin (9600);
  pinMode(pinSensor, INPUT);
}

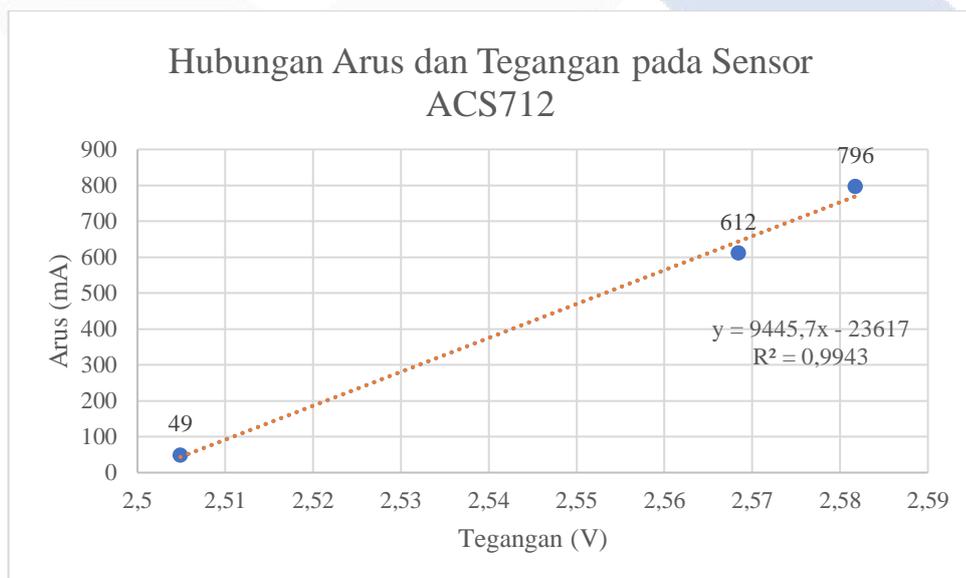
void loop() {
  adc = analogRead(pinSensor);
  tegangan = adc*vpb;
  Serial.print("Tegangan: ");
  Serial.println(tegangan,5);
  delay(500);
}
```

Berikut adalah hasil perbandingan antara pembacaan amperemeter dengan pembacaan sensor.

Tabel 4. 1 Perbandingan hasil konversi tegangan dengan pengukuran amperemeter

No	Arus (Amperemeter)	Tegangan (Sensor)
1	49 mA	2,50489 V
2	612 mA	2,56843 V
3	796 mA	2,58178 V

Berdasarkan tabel 4.1 dibuatlah grafik hubungan antara nilai arus pada amperemeter dengan nilai tegangan pada sensor arus ACS712. Tujuannya adalah untuk mendapatkan persamaan regresi linier yang akan digunakan untuk pengujian sensor arus ACS712. Selain itu grafik digunakan untuk mengetahui seberapa linier pembacaan sensor arus ACS712 terhadap pembacaan amperemeter yang ditunjukkan dengan nilai koefisien determinasi ( $R^2$ ). Nilai koefisien determinasi berkisar dari 0 sampai 1. Semakin mendekati 1, maka semakin dekat hubungan antara data dependen (Amperemeter) dengan data independent (Tegangan Sensor) [29], [30]. Pada proses kalibrasi sensor arus ACS712 ini, nilai  $R^2$  yang ditunjukkan adalah 0,993 atau 99,4 %, sehingga pembacaan sensor arus memiliki linieritas yang tinggi dengan pembacaan Amperemeter. Berikut adalah grafik hubungan antara pembacaan sensor arus ACS712 dengan Amperemeter.



Gambar 4. 30 Hubungan antara pembacaan sensor arus ACS712 dengan Amperemeter.

Berdasarkan grafik diatas, didapatkan persamaan regresi linier sebagai berikut.

$$Arus = (9445,7x \text{ V sensor}) - 23617 \quad (2)$$

Persamaan (2) akan digunakan untuk mengonversi pembacaan sensor arus yang berupa tegangan menjadi nilai arus dengan satuan mA. Setelah itu pembacaan

sensor akan diuji nilai *error* nya dengan pembacaan Amperemeter. Nilai error diperoleh dari rumus berikut:

$$Error = \frac{|nilai\ sensor - nilai\ alat\ ukur|}{nilai\ alat\ ukur} \times 100\ \% \quad (3)$$

Berikut merupakan tabel hasil pengujian sensor arus ACS712

Tabel 4. 2 Hasil Pengujian Sensor Arus ACS712

No	Arus (Amperemeter)	Arus (Sensor)	Error (%)
1	49 mA	43,43 mA	11,3 %
2	612 mA	643,61 mA	5,16 %
3	796 mA	769,71 mA	3,3 %
Rata-rata Error			6,58 %

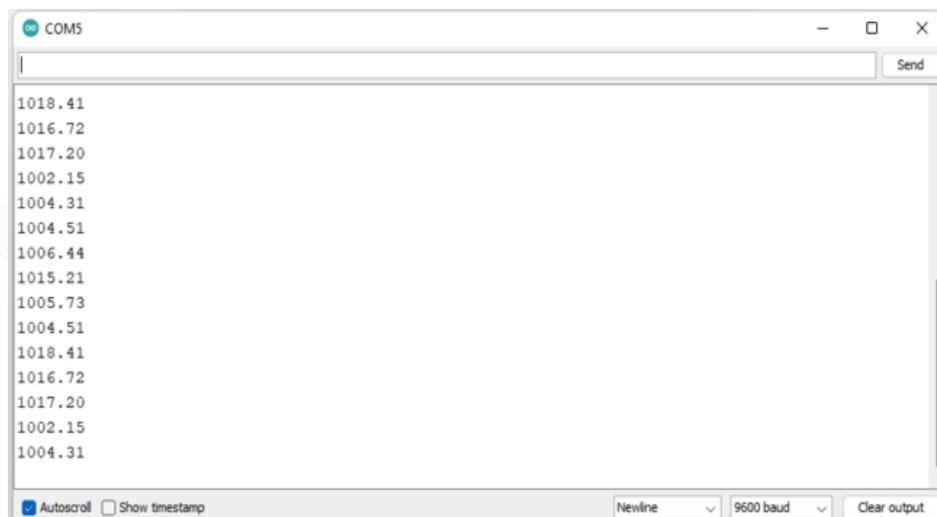
Berdasarkan pengujian sensor arus ACS712 yang telah dilakukan, dapat disimpulkan bahwa sensor arus ACS712 ini memiliki ketepatan pembacaan arus yang akurat dikarenakan memiliki nilai *error* rata-rata sebesar 6,58%. Maka sensor arus ACS712 ini dapat digunakan pada sistem alat penghitung kertas otomatis berbasis IoT.

#### 4.4.3 Pengujian sensor arus ACS712 pada Alat Penghitung Kertas

Pengujian dilakukan untuk mengetahui apakah sensor arus ACS712 mampu mendeteksi peningkatan arus yang terjadi akibat adanya beban berlebih pada motor DC yang diakibatkan oleh kertas yang tersangkut. Simulasi kertas tersangkut dilakukan dengan memberikan objek pada jalur kertas untuk menghalangi proses pendorongan kertas. Kertas yang tertahan akan menahan laju dari *roller* sehingga arus pada motor DC akan meningkat. Nilai peningkatan arus akan dijadikan sebagai syarat pada algoritma Arduino untuk membuat program



Gambar 4. 31 Kertas Tersangkut pada Alat Penghitung Kertas



Gambar 4. 32 Nilai Arus yang Terbaca Oleh Sensor Arus ACS712

Berdasarkan pengujian yang telah dilakukan, pada saat proses penarikan kertas dilakukan, nilai yang terbaca di sensor adalah 684 mA pada saat perputaran searah jarum jam (CW) dan 796 mA pada saat perputaran berlawanan arah jarum jam (CCW). Pada saat motor berputar berlawanan arah jarum jam, *roller* akan menarik kertas dari bagian *input* alat, sehingga nilai arus pada motor DC akan lebih besar dibandingkan dengan proses perputaran searah jarum jam.

Ketika objek penghalang ditempatkan pada alat, maka kertas yang akan didorong keluar tertahan sehingga beban motor DC akan meningkat. Nilai arus yang terdeteksi pada proses ini berada di kisaran 1 A. Berdasarkan pengujian tersebut, dapat disimpulkan jika sensor arus ACS712 mampu mendeteksi peningkatan arus akibat beban motor DC yang berlebih sehingga sensor arus ACS712 ini dapat digunakan pada alat penghitung kertas otomatis sebagai sistem

proteksi. Ketika sensor mendeteksi arus diatas 800 mA, maka alat penghitung kertas akan berhenti secara otomatis dan memberi notifikasi kepada *user* melalui Aplikasi ataupun melalui LCD bahwa alat sudah berhenti. *User* akan diminta untuk menarik kertas yang tersangkut secara manual.

#### **4.5 Pengujian sensor modul LDR**

Untuk proyek akhir ini, sensor modul LDR digunakan untuk melakukan perhitungan pada kertas melalui pembacaan digital dan juga mendeteksi jumlah kertas yang tertarik melalui pembacaan analog. Semakin banyak cahaya yang diterima oleh sensor LDR, maka semakin kecil resistansi yang dihasilkan sehingga tegangan *output* sensor pun akan semakin kecil. Begitu juga untuk sebaliknya, semakin kecil cahaya yang diterima sensor LDR, maka semakin besar resistansi yang dihasilkan sehingga tegangan *output* sensor akan semakin besar [31].

Pada saat tidak ada kertas yang terdeteksi, maka sensor LDR akan menerima cahaya yang berasal dari LED dengan intensitas yang tinggi. Pembacaan digital akan memberikan logika *LOW*. Lalu pada saat kertas yang ditarik oleh *roller* melewati sensor, maka intensitas cahaya yang diterima sensor akan berkurang secara drastis karena cahaya diserap oleh kertas, sehingga pembacaan sensor LDR menjadi logika *HIGH*. Setelah 2 siklus tersebut, maka sensor LDR melakukan proses perhitungan.

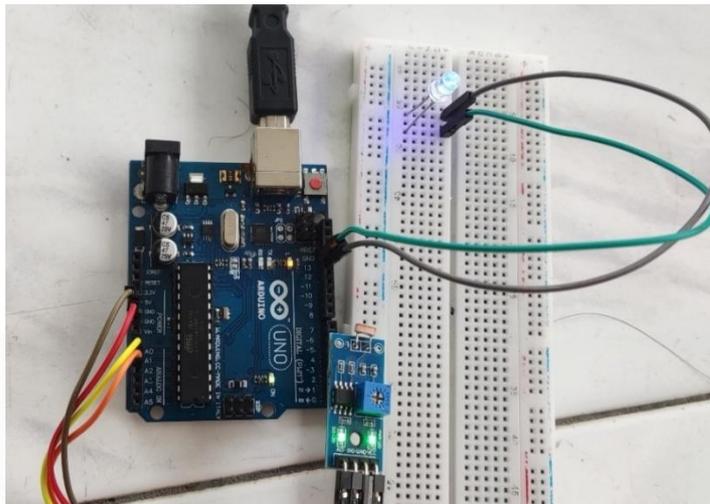
Kertas yang tertarik 1 lembar akan memiliki nilai intensitas cahaya yang berbeda dengan kertas yang tertarik lebih dari 1 lembar. Nilai pembacaan jumlah kertas yang tertarik akan diuji pada pengujian sensor LDR baik untuk kertas 70 gram dan juga 80 gram. Pengujian akan dilakukan dengan membaca nilai keluaran analog sensor LDR.

Pengujian sensor dilakukan dengan menggunakan intensitas cahaya sebagai variabel dengan lux meter sebagai acuan dalam pengkalibrasian sensor. Tujuan dari pengujian sensor ini adalah untuk mengetahui kinerja dari sensor modul LDR ini, dan memastikan bahwa sensor LDR ini dapat digunakan pada alat penghitung kertas otomatis.

#### 4.5.1 Perancangan dan pembuatan rangkaian sensor modul LDR

Berikut adalah desain untuk rangkaian skematik pengujian sensor modul LDR.

Gambar 4.33 Rangkaian Skematik Pengujian Sensor Modul LDR



Gambar 4.34 Rangkaian Pengujian Sensor Modul LDR

#### 4.5.2 Pengkalibrasian sensor LDR

Pada tahapan ini, sensor modul LDR akan dikalibrasi dengan membandingkan pembacaan analog sensor berupa tegangan proporsional yang diperoleh dari konversi persamaan (1) dengan pembacaan cahaya menggunakan *lux meter*. Metode yang dipakai adalah dengan membandingkan metode regresi linier dan juga metode regresi polinomial. Persamaan garis yang didapat dari kedua metode akan diuji pada tahapan pengujian sensor. Berikut adalah program pada Arduino IDE yang digunakan untuk proses kalibrasi ini

```
int pinLDR = A0;  
int L1 = 13;  
double vpb = 0.00488758553;  
double teganganLDR;  
int ldr;
```

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

```

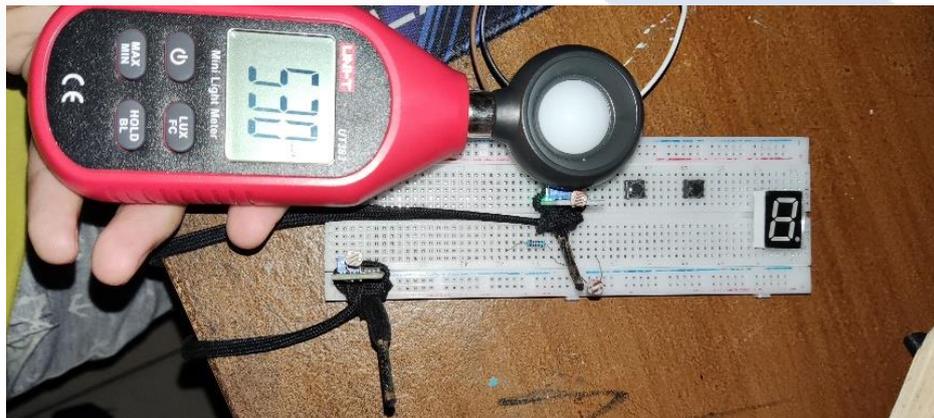
pinMode(13,OUTPUT);
}

void loop() {

ldr = analogRead(pinLDR);
teganganLDR = ldr * vpb;
Serial.print("Nilai Tegangan LDR: ");
Serial.println(teganganLDR,5);
delay(500);
}

```

Pengambilan data dilakukan dalam kondisi ruangan gelap dengan kondisi cahaya 0 lux. Kondisi cahaya yang masuk ke sensor LDR diatur intensitas nya dengan mengatur jarak LED dengan sensor LDR. Berikut data hasil perbandingan alat ukur Lux Meter dengan tegangan proporsional dari sensor modul LDR.



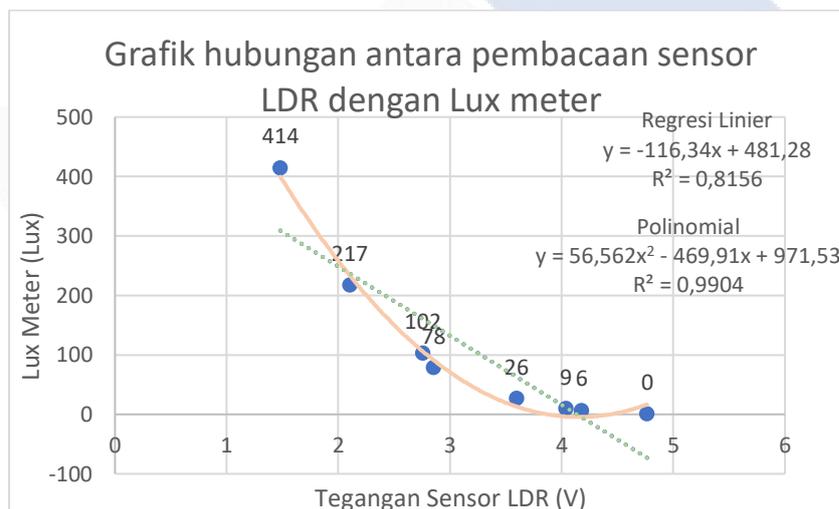
Gambar 4. 35 Pengambilan Data Sensor LDR dengan alat ukur Lux Meter

Tabel 4. 3 Perbandingan Lux Meter dengan Tegangan Sensor LDR

No	Intensitas Cahaya (Lux Meter)	Tegangan (Sensor LDR)
1	0	4,765 V
2	6	4,17889 V
3	9	4,03715 V
4	26	3,59726 V
5	78	2,85435 V

<b>6</b>	102	2,7566 V
<b>7</b>	217	2,10166 V
<b>8</b>	414	1,48094 V

Berdasarkan tabel 4.3, dibuatlah grafik hubungan antara pembacaan sensor LDR dan pengukuran intensitas cahaya oleh lux meter. Tujuan dibuat nya grafik adalah untuk menampilkan linieritas dari pembacaan sensor LDR terhadap alat ukur yaitu lux meter. Selain itu persamaan garis baik regresi linier maupun persamaan regresi polinomial diperoleh dari grafik tersebut. Berikut adalah grafik hubungan antara pembacaan sensor LDR dengan Lux meter.



Gambar 4. 36 Hubungan antara pembacaan sensor LDR dengan Lux meter

Berdasarkan grafik diatas, didapatkan persamaan garis untuk regresi linier dan polinomial sebagai berikut:

1. Regresi Linier:

$$Lux\ sensor = (-116,34 \times V\ sensor) + 481,28 \quad (3)$$

2. Regresi Polinomial :

$$Lux\ Sensor = (56,562 * (V\ sensor^2)) - (469,91 * Vsensor) + 971,53 \quad (4)$$

Persamaan (3) dan (4) akan digunakan untuk konversi nilai *output* sensor berupa tegangan menjadi satuan intensitas cahaya (lux). Tiap persamaan akan

diuji nilai pembacaan nya dan dibandingkan dengan pembacaan alat ukur (lux meter). Berikut adalah hasil perbandingan antara pembacaan sensor LDR yang sudah dikonversi dengan pembacaan alat ukur lux meter.

Tabel 4. 4 Perbandingan pembacaan sensor LDR (linier) dengan Lux Meter

No	lux meter	Lux sensor (linier)	<i>Error</i>
1	6	1,93	67,83 %
2	9	11,03	22,5 %
3	26	63,91	145 %
4	78	147,5	89,1 %
5	102	160	56,86 %
6	217	237,34	9,3 %
7	414	298,75	27,83 %
Rata – rata <i>error</i>			59,77 %

Tabel 4. 5 Perbandingan pembacaan sensor LDR (Polinomial) dengan Lux Meter.

No	lux meter	Lux sensor (Polinomial)	<i>Error</i>
1	6	-4,41	173,5 %
2	9	-2,81	131,1%
3	26	12,46	52 %
4	78	86	10,2 %
5	102	117,88	15,4 %
6	217	240,63	10,5 %
7	414	423,67	2,17
Rata – rata <i>error</i>			56,32 %

Berdasarkan tabel 4.4 dan 4.5, dapat disimpulkan jika kalibrasi sensor LDR dengan polinomial memiliki rata – rata *error* pembacaan yang lebih kecil. Namun selisih nilai *error* dari kedua metode tidak lah jauh. Maka kedua metode akan diuji lagi pada pembacaan kertas untuk alat penghitung kertas otomatis.

### 4.5.3 Pengujian sensor modul LDR untuk pembacaan kertas

Pada tahapan ini, sensor LDR akan diuji dengan menggunakan kedua metode yang sudah digunakan, yaitu regresi linier dan polinomial. Tujuan dari pengujian ini adalah untuk mengetahui berapa nilai yang terbaca oleh sensor LDR pada saat jumlah kertas yang tertarik 1 lembar, ataupun lebih dari 1 lembar. Pengujian juga dilakukan untuk mengetahui metode mana yang lebih baik digunakan pada alat penghitung kertas. Metode yang memiliki nilai pembacaan yang sesuai dengan konsep pembacaan kertas akan diimplementasikan pada alat penghitung kertas otomatis. Pengujian dilakukan dengan mengambil data pembacaan kertas 1 lembar sampai 3 lembar. Jenis kertas yang digunakan adalah kertas 70 gram dan 80 gram. Berikut adalah hasil pengujian pembacaan sensor LDR.

Tabel 4.6 Hasil pengujian sensor LDR dengan metode regresi linier

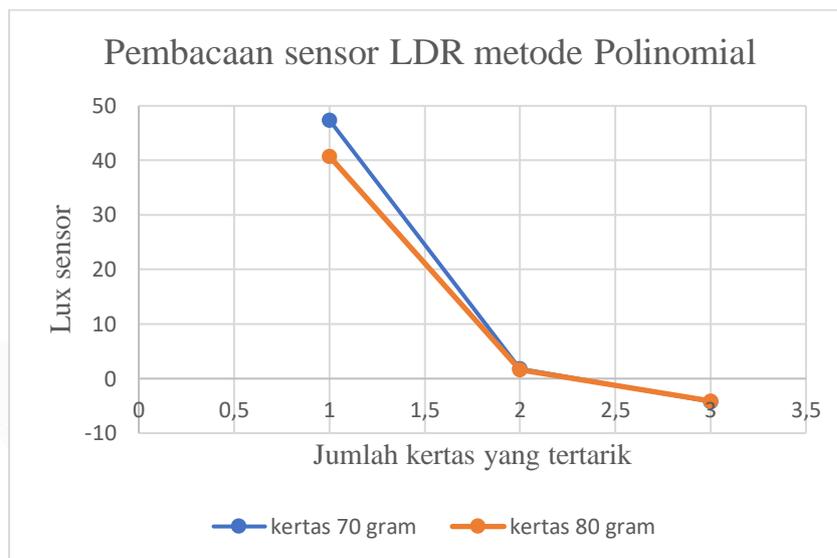
No	Jumlah Kertas	Kertas 70 gram	Kertas 80 gram
1	1	109,97	102,21
2	2	36,63	36,05
3	3	-9,44	-12,6



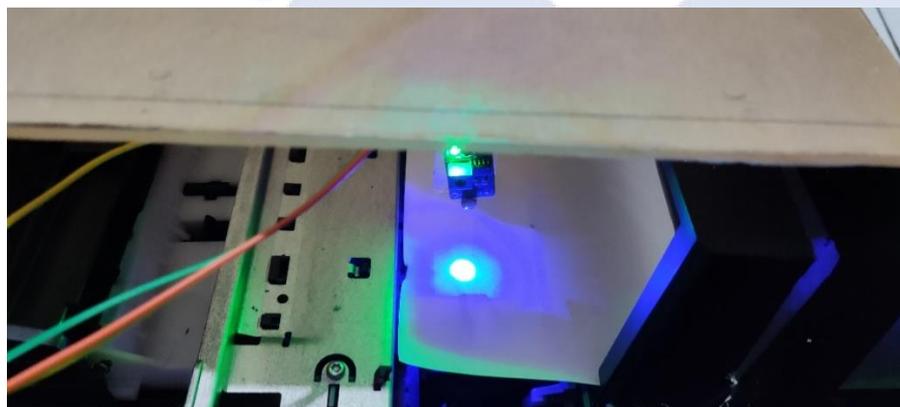
Gambar 4. 37 Perbandingan Pembacaan Kertas dengan Metode Regresi Linier

Tabel 4.7 Hasil pengujian sensor LDR dengan metode Polinomial

\`	Jumlah Kertas	Kertas 70 gram	Kertas 80 gram
1	1	47,39	40,74
2	2	1,77	1,6
3	3	-4,23	-4,1



Gambar 4.38 Perbandingan Pembacaan Kertas dengan Metode Polinomial



Gambar 4. 39 Pengujian Sensor LDR dengan Objek Kertas

Kertas dengan berat 80 gram tentunya memiliki tingkat ketebalan yang lebih besar dibandingkan dengan kertas 70 gram. Maka cahaya yang menembus kertas 80 gram akan lebih sedikit dibandingkan dengan kertas 70 gram. Pembacaan sensor LDR pun akan mengalami hal yang serupa, nilai pembacaan

sensor (dengan satuan lux) pada saat kertas 80 gram harus lebih kecil dibandingkan dengan kertas 70 gram.

Berdasarkan hasil pengujian sensor LDR, nilai pembacaan sensor dengan metode linier menunjukkan jika pembacaan pada jenis kertas 70 gram nilai intensitasnya selalu lebih besar dari pembacaan kertas 80 gram. Sedangkan pada metode polinomial, pembacaan intensitas cahaya pada kertas 70 gram lebih besar hanya pada saat pembacaan 1 – 2 lembar saja. Pada pembacaan 3 lembar, nilai intensitas cahayanya lebih kecil dibandingkan dengan kertas 80 gram. Hal ini dikarenakan pembacaan dengan metode polinomial hanya linier di kisaran pembacaan 414 lux sampai 26 lux (Gambar 4. 11). Di bawah 26 lux, pembacaan sensor akan menjadi berbentuk kurva dan tidak linier. Hal ini yang membuat ketika pembacaan di 3 lembar, nilai intensitas cahaya yang terbaca pada saat kertas 70 gram lebih kecil dari kertas 80 gram.

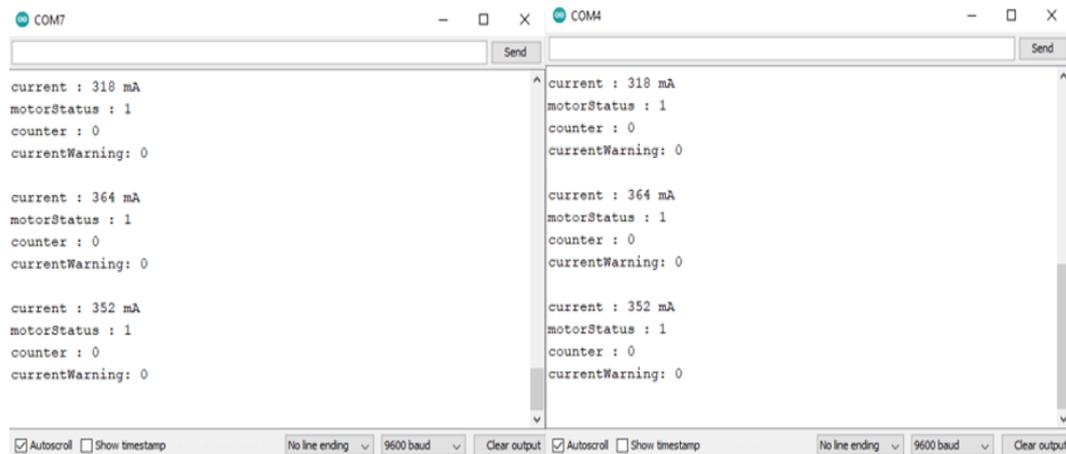
Berdasarkan analisa diatas, dapat disimpulkan jika metode yang akan digunakan pada sensor LDR sebagai sensor penghitung dan pendeteksi jumlah kertas yang tertarik adalah metode regresi linier. Metode regresi linier memiliki pembacaan yang linier terhadap jumlah kertas dibandingkan dengan metode polinomial.

#### **4.6 Pengujian komunikasi serial Node MCU ESP32 dengan Arduino Uno**

Pada tahap ini, pengujian dilakukan untuk mengetahui kemampuan komunikasi baik arduino uno maupun Node MCU ESP32. Pengujian dilakukan dengan melakukan pengiriman data sensor dari arduino ke Node MCU ESP 32. Data yang akan dikirim adalah data pembacaan sensor arus ACS712, data status motor, data perhitungan kertas, dan data peringatan arus berlebih. Rangkaian skematik yang digunakan pada pengujian ini adalah rangkaian keseluruhan alat (Gambar 4.7).

Pin 2 Arduino Uno dihubungkan dengan pin 17 Node MCU ESP32 sebagai *receiver* arduino. Fungsi pin *receiver* adalah sebagai penerima perintah atau informasi dari Node MCU ESP32. Lalu pin 3 Arduino uno digunakan sebagai pin *Transmitter*, dihubungkan dengan pin 16 Node MCU ESP32. Fungsi dari pin *Transmitter* Arduino adalah sebagai pengirim data ke Node MCU ESP32.

Pada pin *Transmitter* diberikan rangkaian pembagi tegangan. Tujuan dari penggunaan rangkaian ini adalah mengurangi tegangan keluaran pin Arduino uno yang bernilai 5 V menjadi 3,3 V, dikarenakan Node MCU ESP32 hanya bisa membaca maksimal tegangan masukkan pada pin sebesar 3,3 V. Berikut adalah hasil pengujian serial komunikasi Arduino Uno dengan Node MCU ESP32.



(a)

(b)

Gambar 4. 40 Tampilan Serial Monitor pada (a) Arduino Uno dan (b) Node MCU ESP32

Berdasarkan hasil pengujian, dapat disimpulkan jika komunikasi serial antara Arduino uno dan Node MCU ESP32 dapat dilakukan. Pembacaan sensor arus, status motor DC, perhitungan kertas dan notifikasi arus berlebih yang diolah oleh Arduino uno (Gambar (a)) berhasil dikirim ke Node MCU ESP32 (Gambar(b)).

#### 4.7 Pengujian Keseluruhan Alat

Pengujian Alat secara keseluruhan dilakukan untuk mengetahui apakah sistem yang dibuat sudah sesuai, baik dari *hardware* maupun *software*. Pengujian aplikasi dilakukan dengan melakukan kontrol *Start* dan *Stop* alat melalui aplikasi PCT-1 dan menampilkan hasil perhitungan. Pengujian alat dilakukan dengan membandingkan waktu perhitungan alat dengan perhitungan secara manual. Berikut adalah hasil pengujian alat secara keseluruhan.

#### 4.7.1 Pengujian Aplikasi PCT-1

Pada tahap ini, pengujian dilakukan untuk mengetahui apakah aplikasi PCT-1 dapat melakukan pengoperasian alat penghitung kertas otomatis. Proses *Start* dan *Stop* alat akan dilakukan melalui aplikasi, begitu juga hasil perhitungan akan ditampilkan melalui aplikasi. Pada tahap ini juga akan dilakukan pengujian pengaruh jarak perangkat terhadap alat penghitung kertas otomatis. Berikut adalah hasil pengujian pengoperasian alat penghitung kertas melalui aplikasi.

Tabel 4. 8 Pengaruh Jarak Terhadap Pengoperasian Alat

Jarak (Meter)	Fungsi alat		
	<i>Start</i>	<i>Stop</i>	<i>Reset</i>
5	✓	✓	✓
10	✓	✓	✓
15	✓	✓	✓
20	✓	✓	✓
25	✓	✓	✓
30	✓	✓	✓
35	✓	✓	✓
40	✓	✓	✓
45	✓	✓	✓
50	✓	✓	✓
55	✓	✓	✓
60	✓	✓	✓
65	✓	✓	✓
70	✓	✓	✓
75	✓	✓	✓
80	✓	✓	✓
85	✓	✓	✓
90	✓	✓	✓
95	✓	✓	✓
100	✓	✓	✓

Keterangan:

✓ : Percobaan Berhasil

✗ : Percobaan Gagal

Berdasarkan hasil pengujian, dapat disimpulkan jika pengoperasian alat penghitung kertas otomatis dapat dilakukan sampai dengan jarak 100 Meter menggunakan aplikasi PCT-1. Namun kelemahan dari sistem IoT ini adalah sangat bergantung terhadap jaringan koneksi internet. Jika koneksi internet yang tersedia tidak memadai, maka akan sulit untuk menghubungkan alat penghitung kertas dan *smartphone* milik pengguna. Maka alternatif yang dibuat adalah dengan membuat pengoperasian alat penghitung kertas secara manual.

#### 4.7.2 Pengujian waktu perhitungan alat dengan perhitungan manual

Pada pengujian ini, dilakukan perbandingan waktu perhitungan alat yang ditampilkan pada aplikasi PCT-1 dengan hasil perhitungan manual. Tujuan dari pengujian ini adalah untuk mengetahui seberapa efektif dan efisien alat penghitung kertas yang sudah dibuat. Perhitungan kertas akan dilakukan pada kertas ukuran A4 dengan berat 70 gram. Pengujian akan dilakukan sebanyak 10 kali dengan selisih 10 lembar. Perhitungan manual akan dilakukan dengan menggunakan 5 orang sebagai sampel dan mengambil nilai rata-rata pada setiap perhitungan berdasarkan jumlah lembar yang dihitung. Perbandingan dari hasil perhitungan alat dan perhitungan manual akan dijadikan indikasi efisiensi dan efektivitas dari alat yang sudah dibuat. Berikut adalah hasil pengujian perhitungan kertas alat penghitung kertas dengan perhitungan secara manual.

Tabel 4. 9 Perbandingan waktu perhitungan alat dengan perhitungan manual

No	Jumlah kertas	Waktu Perhitungan (detik)	
		Alat penghitung	Manual
1	10	29,78	11,13
2	20	60,17	21,71
3	30	90,97	40,01
4	40	119,87	49,72

5	50	151,12	70,09
6	60	180,52	87,25
7	70	209,98	102,19
8	80	240,24	122,29
9	90	270,16	134,51
10	100	301,98	143,91

- Rata- rata waktu perhitungan alat

$$\frac{\text{total waktu}}{\text{total lembar}} = \frac{1654,79 \text{ detik}}{550 \text{ lembar}} = 3 \text{ detik/lembar}$$

- Rata-rata perhitungan manual

$$\frac{\text{total waktu}}{\text{total lembar}} = \frac{783,08 \text{ detik}}{550 \text{ lembar}} = 1,42 \text{ detik/lembar}$$

Berdasarkan rata-rata kecepatan perhitungan alat penghitung kertas dan perhitungan manual didapatkan jika kecepatan perhitungan untuk alat yang sudah dibuat adalah 3 detik/lembar, lalu untuk kecepatan perhitungan manual adalah 1,4 detik/lembar. Terdapat selisih 1,6 detik untuk perhitungan alat yang sudah dibuat dan perhitungan secara manual. Kecepatan pembacaan alat ini dipengaruhi oleh daya pada motor DC. Daya yang terukur pada motor DC adalah 4,8 Watt (12 V x 0,4 A). Sumber daya motor DC diambil dari *power supply* 12 V yang juga digunakan untuk *supply* tegangan Arduino Uno. Solusi untuk dapat meningkatkan RPM dari motor DC adalah dengan cara menaikkan tegangan *input* dan arus untuk motor DC. Bisa dengan menggunakan *Transistor* sebagai penguat arus, dan juga menggunakan *power supply* tersendiri untuk motor DC agar daya motor DC tidak terganggu oleh komponen elektronik yang lain.



(a)



(b)

Gambar 4. 41 Perbandingan perhitungan (a) secara manual dengan (b) perhitungan menggunakan alat penghitung kertas.

Alat penghitung kertas otomatis ini akan efisien jika perhitungan kertas lebih dari 50 lembar. Perhitungan kertas lebih dari 50 lembar secara manual membutuhkan waktu lebih dari 1 menit. Waktu 1 menit dapat digunakan pelayan toko untuk melayani pembeli ataupun melakukan pekerjaan yang lain, sehingga alat penghitung kertas otomatis ini akan membantu efisiensi pelayanan jika jumlah perhitungan kertas lebih dari 50 lembar. Meskipun kecepatan perhitungan alat yang sudah dibuat tidak secepat perhitungan manual, namun alat penghitung kertas ini mampu membantu pekerjaan pelayan toko alat tulis dan kantor.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian yang sudah dilakukan pada Alat Penghitung Kertas Otomatis Berbasis IoT, maka didapatkan kesimpulan sebagai berikut:

1. Alat penghitung kertas otomatis yang didesain memiliki ukuran panjang 43 cm, Lebar 45 cm dan tinggi 37 cm pada bagian belakang dan 15 cm pada bagian depan.
2. Alat Penghitung Kertas Otomatis dapat menjalankan fungsi utama yaitu menghitung kertas dengan pengoperasian manual dan juga pengoperasian otomatis. Sistematis alat dapat berfungsi dengan baik, namun masih terdapat beberapa malfungsi seperti proses penarikan yang tersendat diakibatkan roller tidak mampu menarik kertas.
3. Berdasarkan hasil pengujian aplikasi PCT-1, aplikasi dapat mengontrol dan memonitoring alat penghitung kertas otomatis dari jarak 100 M.
4. Alat penghitung kertas yang sudah dibuat memiliki kecepatan perhitungan 3 detik per lembar. Alat penghitung kertas akan efisien jika perhitungan lembar kertas lebih dari 50 lembar.

#### **5.2 Saran**

Pada proyek akhir ini, terdapat kekurangan yaitu di kecepatan pembacaan kertas. Saran untuk penelitian selanjutnya, kecepatan pembacaan kertas untuk alat penghitung kertas dapat ditingkatkan dengan menggunakan catu daya independen untuk supply ke motor DC, sehingga daya pada motor DC tidak terganggu oleh komponen lain. Selain itu dengan cara mengatur lama waktu atau interval waktu motor DC pada siklus *Clockwise* untuk pendorongan kertas dan siklus *Counter Clockwise* untuk proses penarikan kertas.

## DAFTAR PUSTAKA

- [1] S. Lestari, "Peran teknologi dalam pendidikan di era globalisasi," *EDURELIGIA: Jurnal Pendidikan Agama Islam*, vol. 2, no. 2, pp. 94–100, 2018.
- [2] E. D. Meutia, "Internet of things–Keamanan dan Privasi," in *Seminar Nasional dan Expo Teknik Elektro*, 2015, vol. 1, no. 1, pp. 85–89.
- [3] A. Sutanto, A. Adimulyono, and F. Widyani, "Alat Penghitung Jumlah Lembar Kertas Berbasis Internet Of Things Menggunakan Infra Red Pada PT. Indah Kiat," *Semnasteknomedia Online*, vol. 6, no. 1, pp. 1–9, 2018.
- [4] S. Bahri, "Pembuatan pulp dari batang pisang," *Jurnal Teknologi Kimia Unimal*, vol. 4, no. 2, pp. 36–50, 2017.
- [5] H. A. P. Riyanto, "Rancang Bangun Sortir dan Hitung Lembar Kertas A4 Otomatis Menggunakan Sensor LDR dan Aplikasi Blynk," *Journal of Energy and Electrical Engineering (JEEE)*, vol. 2, no. 2, 2021.
- [6] D. Kurniawan, "Regresi linier," *R-Foundation for Statistical Computing. Vienna, Austria*, vol. 17, 2008.
- [7] M. Syafruddin, L. Hakim, and D. Despa, "Metode Regresi Linier Untuk Prediksi Kebutuhan Energi Listrik Jangka Panjang (Studi Kasus Provinsi Lampung)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 2, no. 2, 2014.
- [8] U. M. Rifanti, H. Pujiharsono, A. Setiawan, and J. Hendry, "Implementasi Moving Average Filter untuk Koreksi Kesalahan Sensor Pengukur Kedalaman Air," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, no. 2, p. 432, 2020.
- [9] F. R. Usman, W. Ridwan, and I. Z. Nasibu, "Sistem peringatan dini bencana banjir berbasis mikrokontroler Arduino," *Jambura Journal of Electrical and Electronics Engineering*, vol. 1, no. 1, pp. 1–6, 2019.
- [10] Arga, "Pengertian Arduino Uno dan Spesifikasinya," <https://pintarelektro.com/pengertian-arduino-uno/>, Jul. 20, 2020.
- [11] S. J. Sokop, D. J. Mamahit, and S. R. U. A. Sompie, "Trainer perifer al antarmuka berbasis mikrokontroler arduino uno," *Jurnal Teknik Elektro Dan Komputer*, vol. 5, no. 3, pp. 13–23, 2016.
- [12] I. Suharjo, "Prototype Alat Kendali Otomatis Penjemur Pakaian Menggunakan NodeMCU ESP32 Dan Telegram Bot Berbasis Internet of

- Things (IOT),” *Journal Of Information System And Artificial Intelligence*, vol. 1, no. 1, pp. 17–24, 2020.
- [13] T. Suryana, “Sistem Pendeteksi Objek Untuk Keamanan Rumah Dengan Menggunakan Sensor Infra Red,” 2021.
- [14] M. Mulyadi, T. Sutikno, and I. Iswanjono, “Purwarupa Alat Penghitung Kertas dan Harga Jasa Pemakaian Mesin Fotokopi Berbasis Mikrokontroler AT89S51,” *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 11, no. 2, pp. 60–64, 2009.
- [15] D. Siswanto, “Jemuran pakaian otomatis menggunakan sensor hujan dan sensor ldr berbasis Arduino Uno,” *e-NARODROID*, vol. 1, no. 2, 2015.
- [16] S. Supatmi, “Pengaruh sensor LDR terhadap pengontrolan lampu,” *Majalah Ilmiah UNIKOM*, 2011.
- [17] B. E. Cahyono, I. D. Utami, N. P. Lestari, and N. S. Oktaviany, “Karakterisasi Sensor LDR dan Aplikasinya pada Alat Ukur Tingkat Kekeruhan Air Berbasis Arduino UNO,” *Jurnal Teori dan Aplikasi Fisika*, vol. 7, no. 2, pp. 179–186, 2019.
- [18] Yonanda and Yoga Bagus, “Monitoring Arus Beban yang Tersalurkan Pada Gardu Induk PLTU Gresik Dengan *Android* Menggunakan *Bluetooth* HC-05 Berbasis Mikrokontroler ARM,” *UMG Repository*, Jul. 2017.
- [19] Goklas Franz Michael Tambun, “Kalibrasi Tegangan pada Wattmeter dengan Sensor Arus Seri ACS 712 Berbasis Mikrokontroler ATM 8535,” *Repositori Institusi Universitas Sumatera Utara*, 2017.
- [20] N. Nugroho and S. Agustina, “Analisa Motor DC (direct current) Sebagai Penggerak Mobil Listrik,” *UNSRI E-Journal*, vol. 2, no. 1, pp. 28–34, Jan. 2015.
- [21] A. Alfstudio, “L298N Motor Driver,” <https://www.teknikelektro.com/2021/08/l298n-motor-driver.html>, Aug. 23, 2021.
- [22] T. I. Nasution and P. F. A. Azis, “MPU-6050 Wheeled Robot Controlled Hand Gesture Using L298N Driver Based on Arduino,” in *Journal of Physics: Conference Series*, 2023, vol. 2421, no. 1, p. 012022.
- [23] A. P. Zanofa, R. Arrahman, M. Bakri, and A. Budiman, “Pintu Gerbang Otomatis Berbasis Mikrokontroler Arduino UNO R3,” *Jurnal Teknik Dan Sistem Komputer*, vol. 1, no. 1, pp. 22–27, 2020.
- [24] R. Abadi, “Catu Daya: Pengertian, Fungsi, Prinsip Kerja, Jenis, Komponen,” <https://thecityfoundry.com/catu-daya/>, Nov. 16, 2022.

- [25] D. A. Lestari, "Pengertian Thinkable," <https://thinkable.dwitari.my.id/2018/08/pengertian-thinkable.html>, Aug. 12, 2018.
- [26] D. Intern, "Apa itu *Firebase*? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya," <https://www.dicoding.com/blog/apa-itu-Firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>, Nov. 25, 2020.
- [27] A. N. Baharsyah, "Internet of Things (IoT) : Pengertian, Manfaat, Contoh, Cara Belajar," <https://www.jagoanhosting.com/blog/pengertian-internet-of-things-iot/>, Aug. 26, 2019.
- [28] C. N. Tanuwijaya, "Apakah itu IoT (Internet of Things)?," <https://sis.binus.ac.id/2018/03/08/apakah-itu-iot-internet-of-things/>, Mar. 08, 2018.
- [29] I. Ghozali, "Aplikasi analisis multivariate dengan program IBM SPSS 25," 2018.
- [30] J. F. Hair, "Multivariate data analysis," 2009.
- [31] R. Abadi, "Sensor LDR : Pengertian, Simbol, Fungsi, Cara Kerja, Karakteristik," <https://thecityfoundry.com/sensor-ldr/>, Dec. 26, 2022.



**LAMPIRAN 1**  
**Daftar Riwayat Hidup**

## Daftar Riwayat Hidup

### 1. Data Pribadi

Nama Lengkap : Andreas Lie Alviero  
Tempat, Tanggal Lahir : Bandung, 15 Mei 2000  
Alamat Rumah : Jl. Kenanga permai atas  
No. HP : 081379689489  
Email : andreasliealviero@gmail.com  
Jenis Kelamin : Laki- Laki  
Agama : Katholik



### 2. Riwayat Pendidikan

1. SD IGN. Slamet Riyadi 2 Lulus 2012
2. SMP YPK Air Kenanga Lulus 2015
3. SMA N 1 Sungailiat Lulus 2018
4. Politeknik Manufaktur Negeri Bangka Belitung 2019-sekarang

### 3. Pendidikan Non Formal

-

Sungailiat, 30 Januari 2023

Andreas Lie Alviero

## Daftar Riwayat Hidup

### 1. Data Pribadi

Nama Lengkap : Dimas Setiawan Nugroho  
Tempat, Tanggal Lahir : Jakarta, 11 Mei 2001  
Alamat Rumah : Jalan Muhidin no 139  
No. HP : 082278296875  
Email : dtuhuh1@gmail.com  
Jenis Kelamin : Laki - Laki  
Agama : Islam



### 2. Riwayat Pendidikan

1. SD N Pengadilan 5 Lulus 2013
2. SMP N 2 Sungailiat Lulus 2016
3. SMA N 1 Sungailiat Lulus 2019
4. Politeknik Manufaktur Negeri Bangka Belitung 2019-sekarang

### 3. Pendidikan Non Formal

-

Sungailiat, 30 Januari 2023

Dimas Setiawan Nugroho



**LAMPIRAN 2**  
**Program Arduino**

```

#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

#define SAMPLE_TIME 1500
//#define MOTOR_TIME 2000

SoftwareSerial mySerial(2, 3); //RX,TX
LiquidCrystal_I2C lcd(0x27, 16, 2); //Address I2C, Baris, Kolom

const int enA = 9;           // PIN Enable A di modul L298D
const int in1 = 8;          // PIN IN1 di modul L298D
const int in2 = 7;          // PIN IN2 di modul L298D
const int buttonStart = 6;  // output tombol start
const int buttonStop = 5;   // output tombol start
const int buttonReset = 4;  // output tombol reset
const int currentSensor = A0; // output current sensor
const int ldr = A1;         // ouput LDR
const int dldr = 10;        // ouput digital LDR
const int detector = 11;    // Input IR Sensor
const int L1 = 13;
int counter = 0;
bool pressed = false;
bool btnPressed = false;
uint8_t motorStatus = 0;
uint8_t stopStatus = 0;
char bufferData[40];
int detectorStatus = 0;
int currentWarning = 0;
int motorRotation = 0;

unsigned int motorTime = 800;
unsigned long previousMillis = 0;
unsigned long currentMillis = 0;
unsigned long previousMillisMotor = 0;
unsigned long currentMillisMotor = 0;
unsigned long previousMillisStop = 0;
unsigned long currentMillisStop = 0;

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

  // Set pin for motor
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(buttonStart, INPUT);
  pinMode(buttonStop, INPUT);
  pinMode(buttonReset, INPUT);
  pinMode(dldr, INPUT);
  pinMode(detector, INPUT);
  pinMode(L1, OUTPUT);

  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);

```

```

    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("Alat Penghitung");
    lcd.setCursor(0, 1);
    lcd.print("Kertas Otomatis");
    delay(2000);
    lcd.clear();
}

void checkButton() {
    detectorStatus = !digitalRead(detector);
    if ((digitalRead(buttonStop) == LOW) && (stopStatus == 0)) {
//Turn Off Motor
        currentMillisStop = millis();
        previousMillisStop = millis();
        stopStatus = 1;

        digitalWrite(enA, HIGH);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        motorTime = 10000;
        motorStatus = 0;
    }
    else if (detectorStatus == 1) {
        if ((digitalRead(buttonStart) == LOW) && (motorStatus == 0)) {
//Turn On Motor
            digitalWrite(enA, HIGH);
            digitalWrite(in1, HIGH);
            digitalWrite(in2, LOW);
            motorStatus = 1;
            motorRotation = 1;
            previousMillisMotor = millis();
            currentMillisMotor = millis();
        }
    }

    currentMillisMotor = millis();

    if (digitalRead(buttonReset) == LOW) {
        currentMillisStop = millis();
        previousMillisStop = millis();
        stopStatus = 1;

        digitalWrite(enA, HIGH);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        motorTime = 10000;
        motorStatus = 0;
        counter = 0;
    }
}

float readCurrent() {

```

```

unsigned int x=0;
float AcsValue=0.0,Samples=0.0,AvgAcs=0.0,AcsValueF=0.0;

for (int x = 0; x < 150; x++){
  AcsValue = analogRead(currentSensor);
  Samples = Samples + AcsValue;
  delay (3);
}
AvgAcs=Samples/150.0;
AcsValueF = (9445.7 *(AvgAcs * 5 / 1024)) - 23617;

delay(50);
return AcsValueF;
}

void checkLDR() {
  if (digitalRead(dldr) == HIGH && pressed == false) {
    int readPaper = analogRead(ldr);
    double voltage = readPaper * 0.0048875855;
    double lux = (-116.34 * voltage) + 481.28;
    if ((89 < lux) && (lux <= 423)) {
      counter = counter + 1;
    }
    else if ((34 < lux) && (lux <= 89)) {
      counter = counter + 2;
    }
    else if (( 7 < lux) && (lux <= 34)) {
      counter = counter + 3;
    }
    else if ((-13 < lux) && (lux <= 7)) {
      counter = counter + 4;
    }
    pressed = true;
  }
  else if (digitalRead(dldr) == LOW && pressed == true) {
    pressed = false;
  }
}

void stopMotor() {
  digitalWrite(enA, LOW);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  motorStatus = 0;
  motorTime = 2000;
  stopStatus = 0;
}

void loop() {
  checkButton();
  digitalWrite(L1,HIGH);
  if (stopStatus == 1) {
    currentMillisStop = millis();
    if ((currentMillisStop-previousMillisStop) > 2000) {
      stopMotor();
    }
  }
}

```

```

}
checkLDR();
float current = readCurrent();
if (current > 1.0) {
    currentWarning = 1;
    digitalWrite(enA, LOW);
    motorStatus = 0;
    stopStatus = 0;
}
else {
    currentWarning = 0;
}
sprintf(bufferData,
"%s,%d,%d,%d,%d",String(current).c_str(),motorStatus,counter,dete
ctorStatus,currentWarning);

while (mySerial.available()){
    char data = (char)mySerial.read();
    if (data == '1') {
        digitalWrite(enA, HIGH);
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        previousMillisMotor = millis();
        currentMillisMotor = millis();
        //Serial.println(data);
        motorStatus = 1;
        motorRotation = 1;
    }
    else if (data == '0') {
        currentMillisStop = millis();
        previousMillisStop = millis();
        motorTime = 10000;
        stopStatus = 1;
        digitalWrite(enA, HIGH);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        motorStatus = 0;
    }

    else if (data == '2') {
        currentMillisStop = millis();
        previousMillisStop = millis();
        motorTime = 10000;
        stopStatus = 1;
        digitalWrite(enA, HIGH);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        motorStatus = 0;
        counter = 0;
    }
}

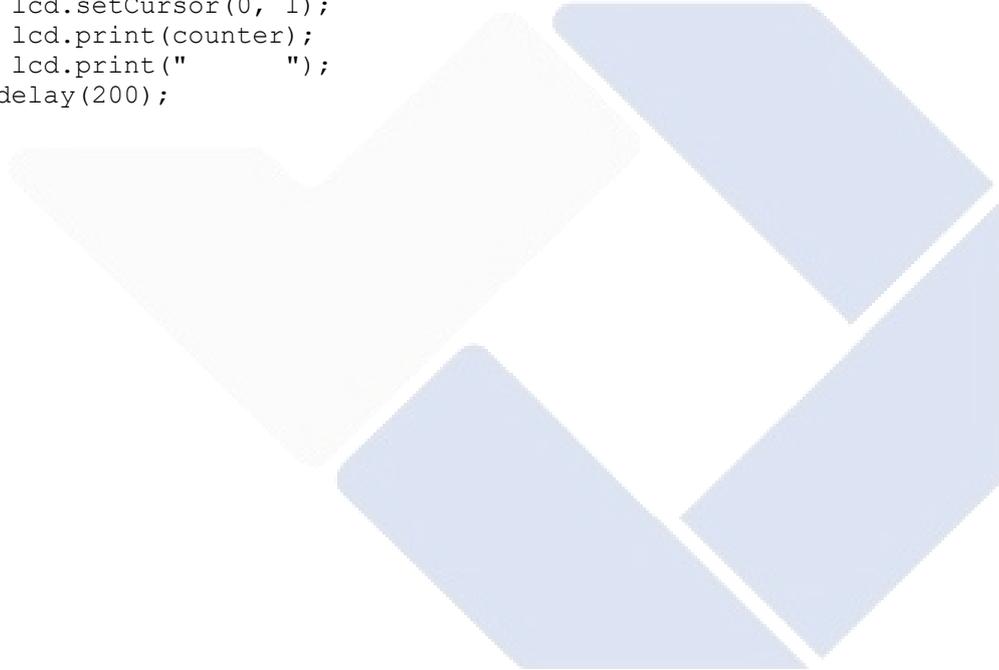
currentMillis = millis();
if ((currentMillis-previousMillis) > SAMPLE_TIME) {
    previousMillis = currentMillis;
    Serial.println(bufferData);
}

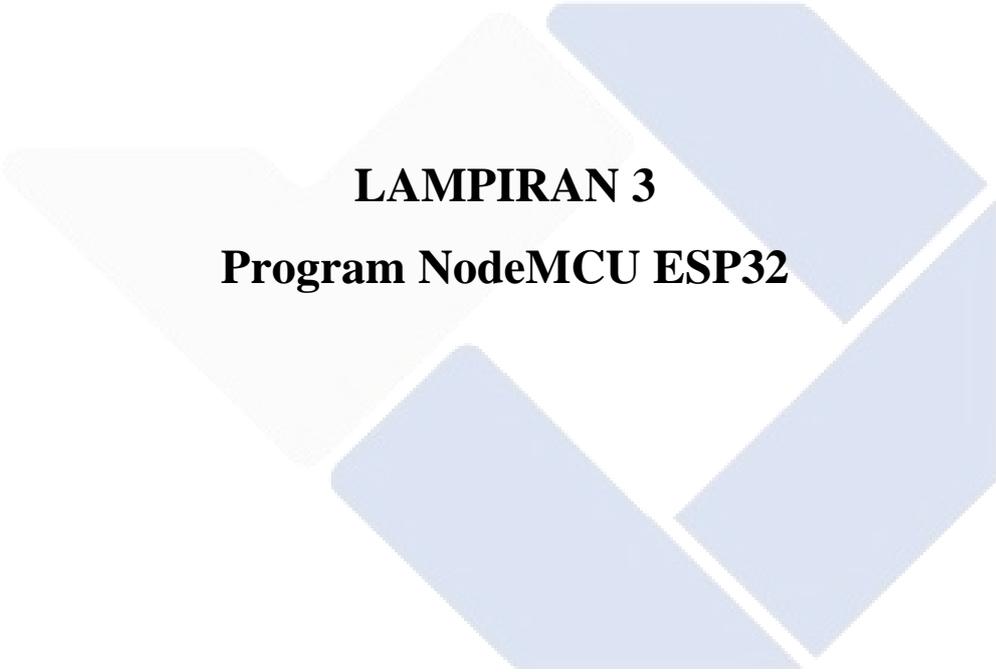
```

```

    mySerial.println(bufferData);
}
if ((currentMillisMotor-previousMillisMotor) > motorTime) {
    if (motorRotation == 1) {
        motorTime = 850; //DORONG
        motorRotation = 0;
    }
    else {
        motorTime = 760; //NARIK
        motorRotation = 1;
    }
    previousMillisMotor = currentMillisMotor;
    PORTB ^= (1 << PB0); // Toggle PIN in1 (D8)
    PORTD ^= (1 << PD7); // Toggle PIN in2 (D7)
}
lcd.setCursor(0, 0);
lcd.print("Jumlah Kertas: ");
lcd.setCursor(0, 1);
lcd.print(counter);
lcd.print("      ");
delay(200);
}

```





**LAMPIRAN 3**  
**Program NodeMCU ESP32**

```

#include <Arduino.h>
#include <WiFi.h>
#include <Firebase_ESP_Client.h>
#include "time.h"

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

// Insert your network credentials
#define WIFI_SSID "uratnadi"
#define WIFI_PASSWORD "anakmama"

#define RXPIN2 16 //Connect to PIN D3 Arduino
#define TXPIN2 17 //Connect to PIN D2 Arduino

// Insert Firebase project API Key
#define API_KEY "AIzaSyBulX0w7J9FDy8EDxKGE0pU1LHhxO_LANY"

// Insert Authorized Email and Corresponding Password
#define USER_EMAIL "dtuhuh1@gmail.com"
#define USER_PASSWORD "Dimas110501"

// Insert RTDB URL
#define DATABASE_URL "https://parsing-data-45735-default-
rtdb.firebaseio.com/"

// Define Firebase objects
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// Variable to save USER UID
String uid;

// Database main path (to be updated in setup with the user UID)
String databasePath;
String readPath;
String updatePath;
// Database child nodes
String currentPath = "/current";
String motorPath = "/motor";
String countPath = "/count";
String timePath = "/timestamp";

// Parent Node (to be updated in every loop)
String parentPath;

int timestamp;
FirebaseJson json;
String dataReceived = "";
String dataParsing[5];
String current = "0";
String motor = "0";

```

```

String count = "0";
String detector = "0";
String currentWarning = "0";

int lastMotorStatus = 9;
int lastPaperCount = 0;
int lastOnlineButton = 0;
int lastDetectorStatus = 0;
int lastCurrentWarningStatus = 0;

const char* ntpServer = "pool.ntp.org";

// Timer variables (send new readings every three minutes)
unsigned long sendDataPrevMillis = 0;
unsigned long timerDelay = 60000;

unsigned long readDataPrevMillis = 0;

// Initialize WiFi
void initWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  Serial.println(WiFi.localIP());
  Serial.println();
}

// Function that gets current epoch time
unsigned long getTime() {
  time_t now;
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    //Serial.println("Failed to obtain time");
    return(0);
  }
  time(&now);
  return now;
}

void setup(){
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, RXPIN2, TXPIN2);

  initWiFi();
  configTime(0, 0, ntpServer);

  // Assign the api key (required)
  config.api_key = API_KEY;

  // Assign the user sign in credentials
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
}

```

```

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

Firebase.reconnectWiFi(true);
fbdo.setResponseSize(4096);

// Assign the callback function for the long running token
generation task */
config.token_status_callback = tokenStatusCallback; //see
addons/TokenHelper.h
// Assign the maximum retry of token generation
config.max_token_generation_retry = 5;
// Initialize the library with the Firebase authen and config
Firebase.begin(&config, &auth);
// Getting the user UID might take a few seconds
Serial.println("Getting User UID");
while ((auth.token.uid) == "") {
  Serial.print('.');
  delay(1000);
}
// Print user UID
uid = auth.token.uid.c_str();
Serial.print("User UID: ");
Serial.println(uid);

// Update database path
databasePath = "/UsersData/" + uid + "/readings";
readPath = "/UsersData/" + uid + "/command/motor";
updatePath = "/UsersData/" + uid + "/command";
}

void loop(){

while (Serial2.available()) {
  char tempData = (char)Serial2.read();
  if(tempData != '\n') {
    dataReceived += tempData;
  }
  else {
    dataParsing [0] = "";
    dataParsing [1] = "";
    dataParsing [2] = "";
    dataParsing [3] = "";
    dataParsing [4] = "";
    if (dataReceived[0] == '#') {
      int index = 0;
      for (int i=1; i < (dataReceived.length()-1); i++) {
        if (dataReceived[i] != ',') {
          dataParsing[index] += dataReceived[i];
        }
        else {
          index ++;
        }
      }
    }
    current = dataParsing[0];
  }
}

```

```

motor = dataParsing[1];
count = dataParsing[2];
detector = dataParsing[3];
currentWarning = dataParsing[4];

//Serial.println(detector);
dataReceived = "";
index = 0;
int motorStatus = motor.toInt();
int paperCount = count.toInt();
int detectorStatus = detector.toInt();
int currentWarningStatus = currentWarning.toInt();
if (motorStatus != lastMotorStatus) {
    lastMotorStatus = motorStatus;
    compareLogic2();
}
if (paperCount != lastPaperCount) {
    FirebaseJson updateData;
    updateData.add("count", count.c_str());
    while(!Firebase.ready());
    Firebase.RTDB.updateNode(&fbdo, updatePath.c_str(),
&updateData);
    lastPaperCount = paperCount;
}
if (detectorStatus != lastDetectorStatus) {
    //Serial.println("masuk");
    FirebaseJson updateData;
    updateData.add("detector", detector.c_str());
    while(!Firebase.ready());
    Firebase.RTDB.updateNode(&fbdo, updatePath.c_str(),
&updateData);
    lastDetectorStatus = detectorStatus;
}
if (currentWarningStatus != lastCurrentWarningStatus) {
    //Serial.println("masuk");
    FirebaseJson updateData;
    updateData.add("warning", currentWarning.c_str());
    while(!Firebase.ready());
    Firebase.RTDB.updateNode(&fbdo, updatePath.c_str(),
&updateData);
    lastCurrentWarningStatus = currentWarningStatus;
}
}
else {
    dataReceived = "";
}
}

// Send new readings to database
if (Firebase.ready() && (millis() - sendDataPrevMillis >
timerDelay || sendDataPrevMillis == 0)){
    sendDataPrevMillis = millis();

    //Get current timestamp
    timestamp = getTime();

```

```

Serial.print ("time: ");
Serial.println (timestamp);

parentPath= databasePath + "/" + String(timestamp);

Serial.println(current);
Serial.println(motor);
Serial.println(count);

json.set(currentPath, current);
json.set(motorPath, motor);
json.set(countPath, count);
json.set(timePath, String(timestamp));
Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo,
parentPath.c_str(), &json) ? "ok" : fbdo.errorReason().c_str());
}

if ((millis()-readDataPrevMillis) > 1000) {
    readDataPrevMillis = millis();
    if (Firebase.RTDB.getString(&fbdo, readPath.c_str())) {
        //if (fbdo.dataTypeEnum() == fb_esp_rtdb_data_type_string) {
            String fbdataString = fbdo.to<String>();
            int fbdata = fbdataString.toInt();
            if (lastOnlineButton != fbdata) {
                lastOnlineButton = fbdata;
                compareLogic();
                //Serial2.print(fbdo.to<int>());
                //Serial.println(fbdo.to<int>());
            }
        //}
    }
    else {
        Serial.println(fbdo.errorReason());
    }
}

void compareLogic() {
    Serial.println(lastMotorStatus);
    Serial.println(lastOnlineButton);
    //Serial2.println(lastOnlineButton);
    if (lastMotorStatus ^ lastOnlineButton) {
        //Serial.println(lastOnlineButton);
        Serial2.print(lastOnlineButton);
    }
}

void compareLogic2() {
    if (lastMotorStatus ^ lastOnlineButton) {
        FirebaseJson updateData;
        updateData.add("motor", String(lastMotorStatus).c_str());
        while(!Firebase.ready());
        Firebase.RTDB.updateNode(&fbdo, updatePath.c_str(),
&updateData);
    }
}

```