

**RANCANG BANGUN ALAT TULIS SALDO DIGITAL  
BERBASIS ESP32 SEBAGAI PENDUKUNG  
ALAT KONTROL BEBAN LISTRIK**

**LAPORAN TUGAS AKHIR**



oleh

**Mohammad Irnanda**  
**NIM E32211535**

**PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI JEMBER  
2024**

**RANCANG BANGUN ALAT TULIS SALDO DIGITAL  
BERBASIS ESP32 SEBAGAI PENDUKUNG ALAT  
KONTROL BEBAN LISTRIK**

**LAPORAN TUGAS AKHIR**



Sebagai salah satu syarat untuk memperoleh gelar Ahli Madya (A.Md)  
di Program Studi Teknik Komputer  
Jurusan Teknologi Informasi

oleh

**Mohammad Irnanda**  
**NIM E32211535**

**PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI JEMBER  
2024**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET DAN TEKNOLOGI  
POLITEKNIK NEGERI JEMBER  
JURUSAN TEKNOLOGI INFORMASI**

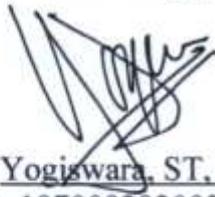
---

**RANCANG BANGUN ALAT TULIS SALDO DIGITAL  
BERBASIS ESP32 SEBAGAI PENDUKUNG ALAT  
KONTROL BEBAN LISTRIK**

**MOHAMMAD IRNANDA (E32211535)**

Telah Diuji pada Tanggal 9 Juli 2024  
Dan Dinyatakan Memenuhi Syarat

Ketua Penguji,



Yogiswara, ST, MT  
NIP. 197009292003121001

Sekretaris Penguji



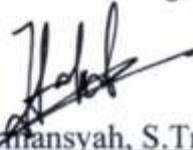
M Hafidh Firmansyah, S.Tr.Kom., M.Sc  
NIP. D19970214202109101

Anggota Penguji



Victor Phoa, S.Si, M.Cs  
NIP. 198510312018031001

Pembimbing,



M Hafidh Firmansyah, S.Tr.Kom., M.Sc  
NIP. D19970214202109101



Mengesahkan  
Ketua Jurusan Teknologi Informasi

Hendra Yulita Ristiawan, S.Kom, M.Cs  
NIP. 19830203006041003

## SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Mohammad Irnanda

NIM : E32211535

Menyatakan dengan sebenar – benarnya bahwa segala pernyataan dalam Laporan Tugas Akhir saya yang berjudul “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik” merupakan gagasan dan hasil karya saya sendiri dengan arahan dosen pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi manapun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan daftar pustaka di bagian akhir Laporan Tugas Akhir ini.

Jember, 9 Juli 2024



Mohammad Irnanda

NIM. E32211535



**PERNYATAAN  
PERSETUJUAN PUBLIKASI  
KARYA ILMIAH UNTUK KEPENTINGAN  
AKADEMIS**

Yang bertanda tangan dibawah ini saya :

Nama : Mohammad Irnanda  
NIM : E32211535  
Program Studi : Teknik Komputer  
Jurusan : Teknologi Informasi

Demi pengembangan Ilmu Pengetahuan, saya menyetujui untuk memberikan kepada UPT. Perpustakaan Politeknik Negeri Jember, Hak Bebas Royalti Non Eksklusif (*Non-Exclusive Royalty Free Right*) atas Karya Ilmiah berupa **Laporan Tugas Akhir** saya yang berjudul :

**RANCANG BANGUN ALAT TULIS SALDO DIGITAL BERBASIS ESP32  
SEBAGAI PENDUKUNG ALAT KONTROL BEBAN LISTRIK**

Dengan Hak Bebas Royalti Non-Eksklusif ini UPT. Perpustakaan Politeknik Negeri Jember berhak menyimpan, mengalih media atau format, mengelola dalam bentuk Pangkala Data (*Database*), mendistribusikan karya dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis atau pencipta.

Saya bersedia untuk menanggung secara pribadi tanpa melibatkan pihak Politeknik Negeri Jember, Segala bentuk tuntutan hukum yang timbul atas Pelanggaran Hak Cipta dalam Karya Ilmiah ini.

Demikian Pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jember  
Pada Tanggal : 9 Juli 2024

Y.



Mohammad Irnanda  
NIM. E32211535

## **MOTTO**

“Aku sesuai persangkaan hamba-Ku. Aku bersamanya ketika ia  
mengingat-Ku”

(Terjemahan HR. Bukhari dan Muslim)

“Maka sesungguhnya bersama kesulitan ada kemudahan. Sesungguhnya  
bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari suatu  
urusan), tetaplah bekerja keras (untuk urusan lain). Dan hanya kepada  
Tuhanmulah engkau berharap”.

(Terjemahan Q.S. Al-Insyirah, 6-8)

“Dan Aku menyerahkan urusanku kepada Allah”

(Terjemahan Q.S. Al-Ghafir, 44)

## **PERSEMBAHAN**

Karya Tulis Ilmiah ini saya persembahkan kepada:

1. Allah S.W.T yang telah memberikan segala kenikmatan dan kelancaran dalam pengerjaan tugas akhir ini.
2. Orang tua saya tercinta Bapak Maskur dan Ibu Sumiati, terima kasih atas semua kasih sayang dan cintanya, dukungan baik moril maupun materil serta doa yang tak henti dan pengorbanan yang tak terhingga dalam menempuh pendidikan selama ini. Terimakasih yang tak terhingga atas semuanya hingga menjadikanku sampai saat ini.
3. Para staf pengajar Politeknik Negeri Jember khususnya Program Studi Teknik Komputer yang telah memberikan banyak ilmu dan pengetahuan serta nasehat yang sangat bermanfaat untuk penulis.
4. Kepada Program Kartu Indonesia Pintar Kuliah yang telah memberikan saya Beasiswa.
5. Almamater tercinta Politeknik Negeri Jember.

## **RINGKASAN**

**RANCANG BANGUN ALAT TULIS SALDO DIGITAL BERBASIS ESP32 SEBAGAI PENDUKUNG ALAT KONTROL BEBAN LISTRIK,**  
Mohammad Irnanda, NIM E32211535, Tahun 2024, Teknik Komputer, Politeknik Negeri Jember, M Hafidh Firmansyah, S.Tr.Kom., M.Sc (Pembimbing I).

Bisnis rumah kost mulai berkembang pesat di lingkungan perkotaan saat ini karena banyak pekerja dan pelajar dari luar daerah yang bekerja di perkotaan. Kondisi rumah yang pada awalnya dibangun hanya untuk menjadi tempat tinggal yang dirubah menjadi rumah indekos tersebut memiliki kendala salah satunya adalah bagaimana cara mengetahui biaya pemakaian tiap ruangnya jika setiap kamar yang ada dihuni oleh para penyewa tersebut dan masalah biaya pemakaian listrik dapat terpantau disetiap kamar secara jelas dan terbukti (Ibrahim et al., 2022).

Melonjaknya tagihan listrik karena faktor tidak terkontrolnya penggunaan listrik merupakan salah satu keluhan bagi pemilik kost. Hal ini dikarenakan pemilik tidak bisa memonitoring penggunaan listrik secara langsung, ditambah lagi dengan adanya penyewa kost yang membawa barang elektronik tambahan seperti ricecooker, laptop secara diam-diam sehingga mengakibatkan melonjaknya tagihan listrik yang harus dibayarkan oleh pemilik kost. Pemilik kost tidak dapat menentukan harga yang akurat untuk alat tambahan yang dibawa oleh penyewa kamar kos. Selain itu, dengan adanya alat kontrol beban listrik, pemilik kost juga membutuhkan perangkat yang reliable untuk top-up saldo alat kontrol beban Listrik yang akan dipasang pada setiap kamar. Untuk menyalakan listrik yang terdapat pada setiap kamar kost nantinya terdapat Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Alat Pengisian saldo untuk alat kontrol beban tersebut.

## PRAKATA

Puji syukur atas ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulisan laporan tugas akhir dengan judul “**Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik**” dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan Diploma tiga (D3) di Program Studi Teknik Komputer Jurusan Teknologi Informasi Politeknik Negeri Jember.

Penulis menyampaikan banyak terima kasih kepada semua pihak yang terlibat dan berkontribusi terhadap proses pengerjaan laporan akhir ini yang meliputi:

1. Saiful Anwar, S.TP, MP selaku Direktur Politeknik Negeri Jember.
2. Bapak Hendra Yufit Riskiawan, S.Kom., M.Cs. selaku Ketua Jurusan Teknologi Informasi.
3. Bapak I Gede Wiryawan, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Komputer.
4. Bapak M Hafidh Firmansyah, S.Tr.Kom., M.Sc. selaku Dosen Pembimbing yang telah membimbing proses pengerjaan Tugas Akhir mulai dari awal sampai selesai.
5. Dosen Penguji Laporan Tugas Akhir bapak Yogiswara S.T., M.T. dan bapak Victor Phoa, S.Si, M.Cs T.
6. Koordinator Tugas Akhir Program Studi Teknik Komputer.
7. Teman - temanku dan semua pihak yang telah ikut membantu dalam pelaksanaan penelitian dan penulisan laporan ini.

Jember, 28 Juni 2023

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN SAMPUL</b> .....	i
<b>HALAMAN JUDUL</b> .....	ii
<b>HALAMAN PENGESAHAN TUGAS AKHIR</b> .....	iii
<b>SURAT PERNYATAAN</b> .....	iv
<b>PERSETUJUAN PUBLIKASI</b> .....	v
<b>MOTTO</b> .....	vi
<b>PERSEMBAHAN</b> .....	vii
<b>RINGKASAN</b> .....	viii
<b>PRAKATA</b> .....	ix
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>DAFTAR GAMBAR</b> .....	xiii
<b>BAB 1. PENDAHULUAN</b> .....	1
<b>1.1 Latar belakang</b> .....	1
<b>1.2 Rumusan masalah</b> .....	2
<b>1.3 Batasan masalah</b> .....	3
<b>1.4 Tujuan</b> .....	3
<b>1.5 Manfaat</b> .....	3
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	4
<b>2.1 Landasan Teori</b> .....	4
<b>2.2 State of The Art</b> .....	13
<b>BAB 3. METODE KEGIATAN</b> .....	16
<b>3.1 Waktu dan Tempat Pelaksanaan</b> .....	16

3.2	Alat dan Bahan .....	16
3.3	Gambaran Umum.....	18
3.4	Tahapan Kegiatan .....	18
3.5	Jadwal Pelaksanaan Kegiatan.....	23
<b>BAB 4.</b>	<b>HASIL DAN PEMBAHASAN.....</b>	<b>24</b>
4.1	Hasil Stuti Literatur .....	24
4.2	Hasil Perancangan Alat .....	24
4.3	Kode Konfigurasi Perangkat.....	26
4.4	Hasil Pembuatan Alat .....	31
4.5	Hasil Pengujian.....	34
<b>BAB 5.</b>	<b>KESIMPULAN DAN SARAN.....</b>	<b>39</b>
5.1	Kesimpulan .....	39
5.2	Saran.....	39
	<b>DAFTAR PUSTAKA .....</b>	<b>40</b>
	<b>LAMPIRAN.....</b>	<b>42</b>

## DAFTAR TABEL

	Halaman
Tabel 2.1 <i>State of The Art</i> .....	15
Tabel 3.1 Kebutuhan <i>Software</i> .....	17
Tabel 3.2 Tabel Kegiatan .....	23
Tabel 4.1 Tabel Penggunaan Pin.....	25
Tabel 4.2 Tabel Library Kode Program .....	26
Tabel 4.3 Percobaan Isi Saldo dan Cek Saldo.....	35
Tabel 4.4 Percobaan Menu Cut Off .....	37

## DAFTAR GAMBAR

	Halaman
Gambar 1.1 Pokok Pembahasan.....	2
Gambar 2.1 <i>Internet of Things</i> .....	5
Gambar 2.2 <i>MQTT Broker Real Protocol</i> .....	5
Gambar 2.3 Node-red.....	6
Gambar 2.4 Aplikasi EasyEDA .....	7
Gambar 2.5 Arduino IDE.....	8
Gambar 2.6 ESP32 .....	9
Gambar 2.7 Modul PN532 .....	9
Gambar 2.8 LCD ILI9255.....	10
Gambar 2.9 I2C PCF8574T .....	11
Gambar 2.10 Keypad Matrix 4x4 .....	11
Gambar 2.11 Stepdown Voltage .....	12
Gambar 2.12 Type-c Power Input.....	13
Gambar 3.1 Alur Tahapan Kegiatan .....	19
Gambar 3.2 Diagram Blok Alat .....	20
Gambar 3.3 Desain Flowchart Sistem.....	21
Gambar 4.1 Skema Fritzing Alat .....	24
Gambar 4.2 Case alat tampak atas .....	32
Gambar 4.3 Bagian dalam alat tampak atas .....	33
Gambar 4.4 Alat setelah disatukan .....	34

# BAB 1. PENDAHULUAN

## 1.1 Latar belakang

Bisnis rumah indekos sudah mulai berkembang pesat di lingkungan perkotaan saat ini karena banyak pekerja pendatang dari luar daerah yang bekerja di perkotaan. Kondisi rumah yang awalnya dibangun hanya untuk menjadi tempat tinggal yang dirubah menjadi rumah indekos tersebut memiliki kendala salah satunya adalah bagaimana cara mengetahui biaya pemakaian tiap ruangnya jika setiap kamar yang ada dihuni oleh para penyewa tersebut dan masalah biaya pemakaian listrik dapat terpantau disetiap kamar secara jelas dan terbukti (Ibrahim et al., 2022). Kemajuan teknologi digital juga mendorong perlunya menjalankan semua pekerjaan dengan kecepatan, efektivitas, dan efisiensi. Contohnya, dalam pembayaran fasilitas publik, kita menyaksikan pergeseran dari transaksi tradisional menuju penggunaan kartu berbasis uang elektronik seperti *e-money*, *e-toll*, dan kartu *flazz*. Penggunaan teknologi ini tidak hanya mempercepat proses pembayaran tetapi juga meningkatkan efisiensi dalam pengelolaan transaksi sehari-hari (Asrar et al., 2013).

Melonjaknya tagihan listrik karena faktor tidak terkontrolnya penggunaan listrik merupakan salah satu keluhan bagi pemilik kost. Hal ini dikarenakan pemilik tidak bisa memonitoring penggunaan listrik secara langsung, ditambah lagi dengan adanya penyewa kost yang membawa barang elektronik tambahan seperti ricecooker, laptop secara diam-diam sehingga mengakibatkan melonjaknya tagihan listrik yang harus dibayarkan oleh pemilik kost. Pemilik kost tidak dapat menentukan harga yang akurat untuk alat tambahan yang dibawa oleh penyewa kamar kos. Selain itu, dengan adanya alat kontrol beban listrik, pemilik kost juga membutuhkan perangkat yang reliable untuk top-up saldo alat kontrol beban Listrik yang akan dipasang pada setiap kamar. Untuk menyalakan listrik yang terdapat pada setiap kamar kost nantinya terdapat alat Tulis Saldo Digital Berbasis Esp32 sebagai alat pengisian saldo untuk alat kontrol beban tersebut. Alat penulisan tersebut nantinya akan digunakan pemilik kost untuk pengisian saldo token digital

pada setiap kamar yang disewakan. Sehingga nantinya token Listrik tersebut digunakan oleh penyewa kamar untuk menyalakan Listrik kamar yang disewanya.



Gambar 1.1 Pokok Pembahasan

Berdasarkan latar belakang tersebut, penulis memutuskan untuk melakukan penelitian dengan judul “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik” Alat ini dirancang untuk Tulis Saldo digital pada kartu rfid yang nantinya akan digunakan pada setiap alat yang ada pada setiap kamar kost supaya meminimalisir kesalahan saat melakukan input token digital. Alat Tulis Saldo Listrik ini digunakan oleh pemilik kost untuk menjual saldo Listrik pada penyewa yang nantinya akan memudahkan pemilik kost dalam membayar tagihan. Alat monitoring ini sudah berbasis mikrokontroller dan Internet of Things.

## 1.2 Rumusan masalah

Dari apa yang dijelaskan latar belakang diatas, dapat diambil rumusan masalah pada sistem sebagai berikut :

- a. Bagaimana cara merancang alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik?
- b. Bagaimana cara membuat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik?
- c. Bagaimana cara kerja Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik?

### **1.3 Batasan masalah**

Berdasarkan rumusan masalah diatas, Adapun batasan masalah pada Alat Tulis Saldo digital ini adalah :

- a. Alat Tulis Saldo digital ini hanya dapat berfungsi apabila terkoneksi dengan internet.
- b. Sistem Tulis Saldo digital hanya mendukung pembelian token listrik pada lingkungan kost.
- c. Alat ini tidak mencakup kontrol dan monitoring penggunaan energi listrik.
- d. Alat ini tidak mencakup alat baca token listrik.

### **1.4 Tujuan**

Dalam penelitian ini terdapat beberapa tujuan yaitu:

- a. Merancang “Alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik”.
- b. Membangun “Alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik”.
- c. Mengetahui cara kerja “Alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik”.

### **1.5 Manfaat**

Adapun Manfaat yang didapatkan dari dibuatnya media pembelajaran berbasis website ini antara lain sebagai berikut :

- a. Memberikan kemudahan bagi pemilik kost dalam menjual token listrik kepada penyewa kost.
- b. Memberikan kemudahan bagi pemilik kost untuk memberikan harga sewa kamar.
- c. Meningkatkan efisiensi penggunaan energi listrik di kost.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Landasan Teori

#### 2.1.1 Internet of Things

*Internet of Things* (IoT) merupakan sebuah sistem yang terdiri dari sensor, aktuator, dan mikrokontroler yang bekerja bersama. Dalam kerangka ini, IoT mengintegrasikan perangkat keras dan perangkat lunak untuk mengumpulkan data dari sensor yang terhubung. IoT dianggap sebagai paradigma baru dalam perolehan data menurut berbagai sumber. Selain itu, IoT memiliki aplikasi yang luas, termasuk dalam rumah pintar dan lokasi lainnya (Kumar et al., 2019). Meskipun memberikan kemudahan dalam pengumpulan dan manajemen data, penggunaan IoT juga membawa risiko, seperti serangan pada berbagai lapisan jaringan, aplikasi, dan lapisan fisik (Kandasamy et al., 2020).

IoT juga dapat didefinisikan sebagai kemampuan berbagai perangkat untuk saling terhubung dan bertukar data melalui jaringan internet. Ini merupakan teknologi yang memungkinkan pengendalian, komunikasi, serta kolaborasi dengan beragam perangkat keras untuk berbagi data melalui internet. Oleh karena itu, IoT merujuk pada konsep menghubungkan "*things*" yang tidak dioperasikan secara langsung oleh manusia ke internet (Priyono et al., 2015). Namun, IoT tidak hanya berfokus pada pengendalian perangkat dari jarak jauh, tetapi juga pada konsep memvirtualisasikan segala hal yang nyata ke dalam bentuk internet, serta elemen-elemen lainnya. Internet menjadi jaringan yang menghubungkan mesin-mesin secara otomatis, di samping adanya pengguna yang bertugas sebagai pengatur dan pengawas langsung atas operasi perangkat tersebut. Manfaat dari penggunaan teknologi IoT termasuk mempercepat, menyederhanakan, dan meningkatkan efisiensi dalam pekerjaan manusia (Hergika et al., 2021).



Gambar 2.1 *Internet of Things*

### 2.1.2 MQTT

*Message Queue Telemetry Transport* (MQTT) adalah salah satu cara koneksi machine to machine (M2M) yang dirancang khusus untuk mengirimkan data dengan ringan menggunakan arsitektur TCP/IP. Protokol ini dirancang untuk memungkinkan perangkat-perangkat yang memiliki sumber daya terbatas, seperti sensor atau perangkat IoT, untuk terhubung dan bertukar data dengan efisien. MQTT memungkinkan perangkat-perangkat tersebut untuk terhubung ke jaringan internet dan bertukar data dengan server atau perangkat lainnya dengan menggunakan protokol yang sederhana dan ringan (Susanto et al., 2018).



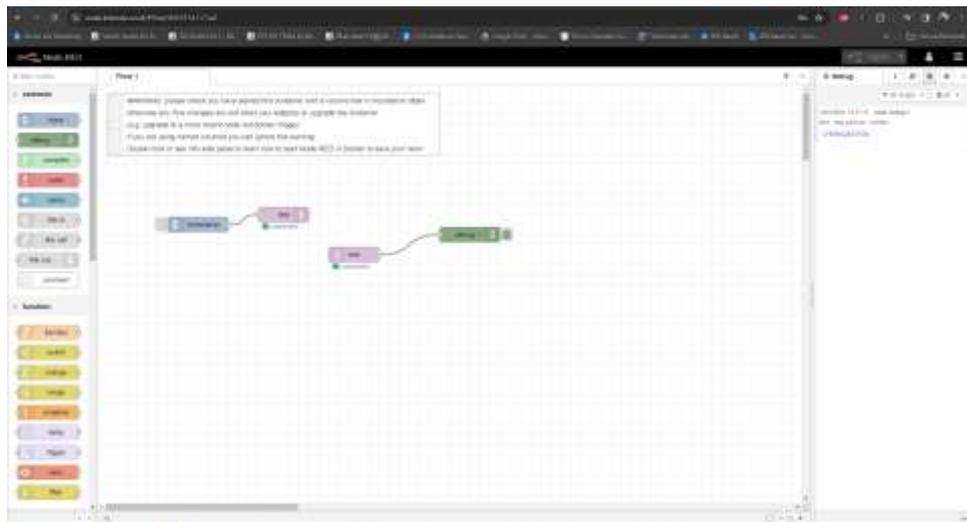
Gambar 2.2 *MQTT Broker Real Protocol*

Pada gambar tersebut memvisualisasikan hubungan antara penerbit, perantara, dan pelanggan dalam konteks sistem publikasi/subskripsi. Penerbit, yang terwakili oleh entitas di sebelah kiri, bertanggung jawab untuk menerbitkan data atau informasi dengan menggunakan perintah `Publish(Topic, Data)`, yang mencakup topik dan data terkait. Di tengah, perantara, atau broker, berfungsi sebagai perantara komunikasi antara penerbit dan pelanggan. Proses dimulai ketika penerbit menerbitkan data, yang kemudian diterima oleh perantara. Pelanggan, yang terwakili oleh entitas di sebelah kanan, berlangganan data dengan perintah `Subscribe(Topic)` yang berisi topik data yang diminati. Perantara kemudian

meneruskan data yang diterimanya dari penerbit kepada pelanggan yang telah berlangganan topik tersebut. Secara keseluruhan, gambar tersebut menggambarkan alur data atau informasi dari penerbit hingga pelanggan melalui perantara dalam suatu sistem publikasi/subskripsi.

### 2.1.3 Node-Red

Node-RED merupakan sebuah alat berbasis browser yang memungkinkan pembuatan aplikasi IoT dengan pendekatan pemrograman visual, memudahkan pengguna dalam membuat aplikasi melalui struktur yang disebut "flow". Flow ini terdiri dari node-node yang saling terhubung, di mana setiap node melaksanakan tugas tertentu. Meskipun dirancang khusus untuk IoT, Node-RED juga dapat diterapkan untuk keperluan umum dan beragam jenis aplikasi. Nama "Node" diberikan karena alat ini diimplementasikan sebagai aplikasi node, meskipun secara konsumen hanya merinci detail implementasi internal (Fajrur, 2020).



Gambar 2.3 Node-red

Pada gambar 2.3 diatas merupakan alur Node-RED yang terdiri dari node pemicu, node fungsi, dan node debug. Meskipun, tanpa konteks lebih lanjut, tujuan pasti dari alur tersebut sulit ditentukan. Node pemicu, dengan label "inject," kemungkinan memulai alur ketika kondisi tertentu terpenuhi, meski detail konfigurasi tersembunyi. Node fungsi, dengan label "function," melakukan tugas berdasarkan kode di dalamnya, namun sayangnya, kode tidak terlihat dalam

gambar. Sementara node debug, dengan label "debug," mengeluarkan informasi eksekusi alur, dalam hal ini menampilkan pesan "test".

#### 2.1.4 easyEDA

EasyEDA adalah software yang memungkinkan Anda untuk membuat sirkuit cetak gratis tanpa batasan PCB. Salah satu kelebihan software ini adalah digunakan dari halaman web dan Anda tidak perlu menginstal plugin tambahan, yang membuatnya sangat menarik untuk melakukan sirkuit cepat tanpa peralatan yang sangat kuat. Salah satu manfaat paling menonjol dari perangkat lunak ini adalah memungkinkan Anda membuat sirkuit tercetak atau skema sederhana untuk menguji atau melihat bagaimana sirkuit berperilaku. Dalam hal ini kita dapat melakukan tes sirkuit manapun yang kita inginkan seperti dalam program simulasi. Salah satu fitur paling menonjol dari EasyEDA adalah pengeditan dan penempatan sirkuit yang mudah dalam skema, membuatnya cukup intuitif dan nyaman (Fikri, 2022)



Gambar 2.4 Aplikasi EasyEDA

#### 2.1.5 Arduino IDE

Arduino IDE adalah singkatan dari Integrated Development Environment, yang dapat dijelaskan sebagai lingkungan terintegrasi yang digunakan untuk melakukan pengembangan (Hergika et al., 2021). Istilah "lingkungan" digunakan karena melalui perangkat lunak ini, pemrograman Arduino dapat dilakukan untuk menjalankan fungsi-fungsi yang telah disematkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang mirip dengan bahasa C. Bahasa pemrograman Arduino, yang disebut sebagai "Sketch," telah mengalami perubahan untuk mempermudah pemula dalam memahami pemrograman dari bahasa aslinya. Sebelum dijual di pasar, IC mikrokontroler Arduino telah

diprogram dengan suatu program yang disebut Bootloader, yang berfungsi sebagai perantara antara compiler Arduino dan mikrokontroler. Pentingnya bahasa pemrograman JAVA terlihat dalam fakta bahwa Arduino IDE dibuat dari bahasa pemrograman tersebut. Arduino IDE juga dilengkapi dengan perpustakaan C/C++, yang dikenal sebagai Wiring, yang meningkatkan kemudahan operasi input dan output. Perlu dicatat bahwa Arduino IDE dikembangkan dari perangkat lunak Processing yang dimodifikasi khusus untuk pemrograman dengan Arduino (Hergika et al., 2021).



Gambar 2.5 Arduino IDE

### 2.1.6 ESP32

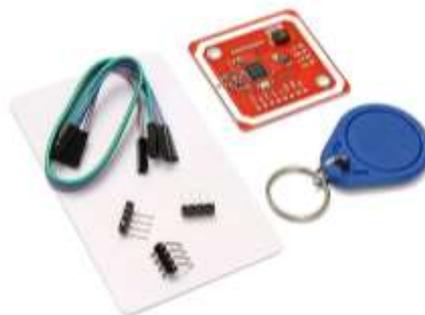
ESP32, dikembangkan oleh Espressif Systems yang berbasis di Shanghai, China, adalah mikrokontroler yang menyediakan solusi jaringan WiFi independen, menghubungkan mikrokontroler ke jaringan WiFi. Mikrokontroler ini menawarkan kombinasi harga yang terjangkau, konsumsi daya rendah, dan fitur konektivitas WiFi dan Bluetooth dual-mode. Generasi ESP32 menggunakan mikroprosesor dual-core Xtensa LX6. Dengan spesifikasi seperti prosesor dual-core dan memori Flash 4MB, ESP32 cocok untuk aplikasi sistem yang ringan dan hemat biaya. Chip ini dapat terhubung langsung ke router melalui Wi-Fi bawaan, menjadikannya pilihan yang optimal untuk berbagai proyek (Kusumah & Pradana, 2019).



Gambar 2.6 ESP32

### 2.1.7 Modul PN532

PN532 adalah salah satu chip NFC yang umumnya terdapat pada smartphone dan perangkat NFC lainnya. Chip ini memiliki kemampuan membaca dan menulis tag serta kartu NFC, dan dapat berkomunikasi melalui modus HSU, I2C, maupun SPI. Dalam penulisan ini, kita akan fokus pada penggunaan modus I2C yang memerlukan empat kabel untuk menghubungkan PN532 dengan mikrokontroler, yaitu pin VCC, GND, SDA, dan SCL. Kemampuannya meliputi pembacaan dan penulisan ke NFC tag dan kartu RFID, berkomunikasi secara nirkabel dengan telepon genggam (misalnya untuk transaksi pembayaran), dan berperan seperti sebuah tag NFC (mensimulasikan 1443 - A atau kartu virtual). Apapun aplikasi yang menggunakan NFC, chip ini dapat memenuhi kebutuhan mulai dari perangkat P2P (Peer - to - peer Communication, berkomunikasi dua arah dengan perangkat NFC lainnya) hingga komunikasi NFC dengan ponsel android (Zurairah et al., 2022).



Gambar 2.7 Modul PN532

### 2.1.8 LCD ILI9225

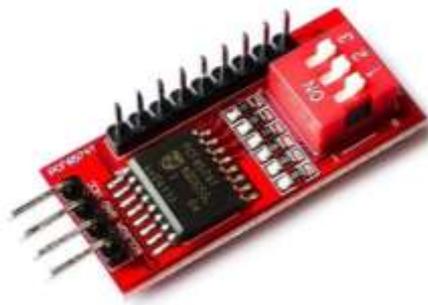
ILI9225 adalah chip tampilan grafis (LCD controller) yang digunakan untuk mengontrol tampilan LCD (Liquid Crystal Display) TFT (Thin Film Transistor). Chip ini memiliki resolusi 176 x 220 piksel dan mendukung berbagai macam mode warna, termasuk RGB (Red, Green, Blue), RGBA (Red, Green, Blue, Alpha), dan YUV (luma, chroma) (Sulistiyowati et al., 2021)



Gambar 2.8 LCD ILI9225

### 2.1.9 I2C

I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Mikrokontroler I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan mikrokontroler I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Berikut adalah rangkaian LCD 16x2 dengan Modul I2C pada rangkaian mikrokontroler arduino (Dita et al., 2021).



Gambar 2.9 I2C PCF8574T

#### 2.1.10 Keypad

Keypad adalah rangkaian tombol yang berfungsi untuk memberi sinyal pada suatu rangkaian dengan menghubungkan jalur-jalur tertentu. Pada sistem pengontrolan ini, digunakan keypad matriks 4 x 4 (16 push button) dengan pin penghubung rangkaian berjumlah 8 pin. Keypad juga merupakan elemen krusial dalam perangkat elektronika yang memerlukan interaksi manusia. Berfungsi sebagai antarmuka antara perangkat elektronik (mesin) dan manusia, keypad dikenal sebagai HMI (Human Machine Interface). Pada artikel ini, Matrix keypad 4x4 dijelaskan sebagai contoh keypad yang memfasilitasi komunikasi antara manusia dan mikrokontroler. Keypad ini memiliki struktur sederhana dan efisien dalam penggunaan port mikrokontroler. Pengaturan keypad dalam bentuk matriks ini bertujuan untuk menghemat penggunaan port mikrokontroler karena sistem tersebut memerlukan banyak tombol pada satu waktu (Sulistiyowat & Febriantoro, 2015).



Gambar 2.10 Keypad Matrix 4x4

### 2.1.11 Stepdown Voltage

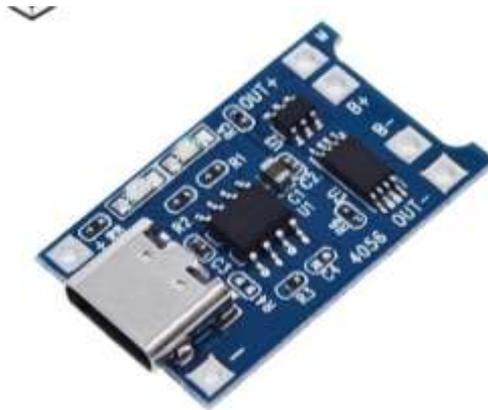
Step down adalah istilah yang digunakan untuk menggambarkan proses menurunkan tegangan listrik dari tegangan yang lebih tinggi ke tegangan yang lebih rendah, seperti pada trafo step-down. Trafo step-down memiliki lilitan primer yang lebih banyak daripada lilitan sekunder. Contoh lainnya adalah pada alat adaptor untuk mengubah tegangan bolak-balik (AC) menjadi tegangan searah (DC). Ciri lain trafo step-down adalah tegangan listrik pada lilitan primer lebih besar dibandingkan tegangan pada lilitan sekunder, serta kuat arus primernya lebih besar dibandingkan kuat arus sekunder (Rumimper et al., 2018).



Gambar 2.11 Stepdown Voltage

### 2.1.12 Type C Power Input

Type C power input adalah sebuah konektor yang dapat digunakan untuk menyalurkan daya listrik ke perangkat elektronik. Konektor ini memiliki beberapa kelebihan dibandingkan dengan konektor lain (Budiyanta et al., 2019), seperti reversible, fast charging, versatile, compact, dan durable. 16 Konektor Type C banyak digunakan pada berbagai macam perangkat elektronik, dan diperkirakan akan menjadi konektor yang paling banyak digunakan di masa depan.



Gambar 2.12 Type-c Power Input

## 2.2 State of The Art

### 2.2.1 Penelitian Terdahulu

Penelitian ini mengambil rujukan dan referensi dari beberapa penelitian sebelumnya yang mempunyai bahasan kurang lebih sama dengan penelitian yang akan dilaksanakan. Dibawah ini akan dijelaskan penelitian yang sebelumnya yang berkaitan dengan Rancang Bangun Alat Input Token Berbasis ESP32 Sebagai Sistem Pendukung Alat Kontrol Beban Listrik.

- a. Model Pengisian Pulsa Listrik KWH Meter Dengan *Smart Card* (Siregar et al., 2018)

Listrik Prabayar adalah salah satu bentuk perkembangan teknologi dari PLN yang digunakan untuk mengatur penggunaan energi listrik rumah tangga melalui meteran listrik yang sudah diintegrasikan langsung dengan database PLN. Tujuan inovasi ini supaya Masyarakat mendapatkan kenyamanan dalam melakukan isi ulang token listrik sehingga masyarakat tidak perlu datang ke konter PLN untuk membayar taguhan listrik tetapi dengan adanya sistem pulsa listrik, Masyarakat dapat mengisi sendiri jumlah energi listrik yang akan digunakan melalui nominal pulsa listrik. Meskipun inovasi ini memberikan bantuan signifikan kepada masyarakat dalam melaksanakan transaksi pembelian energi listrik, beberapa orang masih mengalami kesulitan dalam menggunakan meteran listrik Prabayar. Hal ini disebabkan oleh panjangnya voucher yang mencapai 20 angka yang harus

dimasukkan ke dalam meteran Prabayar. Selain itu, jika terjadi kesalahan dalam memasukkan nomor token, maka seluruh nomor token harus dimasukkan kembali. Dengan adanya model pengisian pulsa Listrik dengan smartcard (kartu pintar) sebagai alternatif pengisian nomor digit ke dalam meteran Listrik Prabayar yang saat ini dilakukan dengan cara manual. Penggunaan smartcard ini digunakan untuk mengurangi Tingkat kesalahan pada saat memasukkan token. Aplikasi ini juga dapat mendeteksi berapa besar daya Listrik yang dipakai, dan menyimpannya di database dan dapat menentukan penggunaan pulsa pada kWh Listrik. Dan membantu pengguna untuk mengetahui penggunaan Listrik termasuk dalam kategori hemat, normal, atau boros. System dirancang dengan program Vb.net dan MySQL. Hasil yang didapat berupa Model pengisian pulsa pada meteran listrik Prabayar berbasis kartu pintar dan aplikasi yang menyimpan riwayat pemakaian energi listrik digunakan untuk menentukan kategori pemakaian listrik.

b. Pengisian Pulsa (Token) Listrik Menggunakan SMS (*Short Messages Services*). (Sudimanto, 2017)

Zaman modern ini banyak sekali alat pendukung kehidupan manusia yang membutuhkan tenaga Listrik sebagai sumber daya untuk mengoperasikannya sama seperti lampu, mesin cuci, pompa air, televisi, radio dan perangkat elektronik lainnya. Listrik merupakan kebutuhan manusia yang sangat penting bagi kehidupan sehari-hari yang membuat manusia sangat bergantung pada Listrik. Untuk memberikan pelayanan yang maksimal kepada Masyarakat PLN mengembangkan sistem pelayanan untuk pembayaran Listrik, sistem layanan listrik Prabayar merupakan sistem yang mengharuskan pelanggan membayar terlebih dahulu dengan membeli token atau pulsa untuk menggunakan daya listrik. Pengisian listrik menggunakan *voucher* token pulsa yang dimasukkan secara manual ke dalam kWh meter. Pemodelan pengisian pulsa listrik ini merupakan pengembangan kWh meter Prabayar yang menggunakan mikrokontroler pada board modul Arduino sebagai pengendali sistem utama. Pengiriman kode kuota dengan 20 digit angka yang dikirimkan oleh pelanggan melalui sms. Mikrokontroler akan menerima dan memasukkan sms berupa kode pulsa dan

keluaran berupa hasil penambahan pulsa kuota sesuai dengan nominal yang dikirimkan kepada pelanggan. Pelanggan akan menerima notifikasi berupa sms apabila kuota listrik akan habis. Mikrokontroler akan memeriksa kode kuota pulsa yang akan diterima sesuai dengan format atau tidak sesuai dengan format, pemeriksaan dilakukan secara bertahap agar tidak terjadi kesalahan dalam pengiriman kode.

### 2.2.2 State of The Art

Berdasarkan penjelasan pada literatur diatas maka project Tugas Akhir (TA) yang berjudul “Rancang Bangun Alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik”.

Tabel 2.1 *State of The Art*

No	Penulis	Judul	Perbedaan
1.	Sudimanto (2017)	Pengisian Pulsa (Token) Listrik Menggunakan SMS (Short Messages Services)	<ul style="list-style-type: none"> <li>- Menggunakan Arduino</li> <li>- Penerapan pada meteran rumah</li> <li>- RFID 125kHz</li> <li>- CT Sensor</li> </ul>
2.	Riki Ruli A. Siregar, Hengki Sikumbang, Riko Jefri Pasaribu (2018)	Model Pengisian Pulsa Listrik KWH Meter dengan Smart Card	<ul style="list-style-type: none"> <li>- Menggunakan Arduino</li> <li>- Penerapan pada meteran rumah</li> </ul>
3.	Mohammad Irnanda (2024)	Rancang Bangun Alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik	<ul style="list-style-type: none"> <li>- Menggunakan ESP32</li> <li>- Penerapan pada lingkungan Kost</li> <li>- Modul PN523</li> <li>- I2C</li> <li>- Keypad</li> <li>- LCD ILI 9255</li> <li>- Stepdown Voltage</li> <li>- Type C Power Input</li> </ul>

## **BAB 3. METODE KEGIATAN**

### **3.1 Waktu dan Tempat Pelaksanaan**

Waktu dan tempat pelaksanaan penyusunan Proposal Tugas Akhir dengan judul “Rancang Bangun Alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik” berlangsung selama 5 (lima) bulan, dari bulan Februari sampai Juli 2024 di Politeknik Negeri Jember.

### **3.2 Alat dan Bahan**

Bahan dan alat merupakan komponen yang harus di perhatikan untuk menyelesaikan Tugas Akhir ini. Adapun bahan dan alat yang digunakan dalam penyusunan penelitian ini yaitu:

#### **3.2.1 Alat**

Alat yang dibutuhkan dalam penelitian ini yaitu :

- a. Perangkat Keras (*hardware*)
  1. Laptop
  2. Obeng
  3. Multimeter
  4. Solder
  5. Tang Potong
  6. Tang Cucut
  7. Jangka Sorong Digital
  8. Cutter
  9. Lem tembak

- b. Perangkat Lunak (*software*)

Perangkat lunak (*software*) yang digunakan dalam pembuatan Tugas Akhir ini adalah sebagai berikut :

Tabel 3.1 Kebutuhan *Software*

<b>Software / tools</b>	<b>Fungsi</b>
<b>Arduino IDE</b>	<i>Software</i> untuk mendukung penulisan code
<b>EasyEDA</b>	<i>Software</i> untuk membuat desain PCB
<b>Microsoft Office</b>	<i>Software</i> untuk membuat Laporan Proposal Tugas Akhir.
<b>Google Chrome</b>	<i>Software</i> mesin pencari online.
<b>Canva</b>	<i>Software</i> untuk mendesain gambar
<b>Mendeley</b>	<i>Software</i> untuk memformat sitasi dan membuat Daftar Pustaka
<b>Publish of Perish</b>	<i>Software</i> untuk melakukan riset jurnal

### 3.2.2 Bahan

Adapun bahan-bahan yang diperlukan dalam pembuatan Tugas Akhir dengan judul “Rancang Bangun Alat Tulis Saldo Digital Berbasis ESP32 Sebagai Pendukung Alat Kontrol Beban Listrik” adalah sebagai berikut :

1. Esp32
2. PN532
3. LCD ILI9255
4. I2C
5. Keypad
6. Stepdown Voltage
7. Type-C Power Input
8. Kabel Fleksibel AWM 20624
9. Socket kabel fleksibel

### 3.3 Gambaran Umum

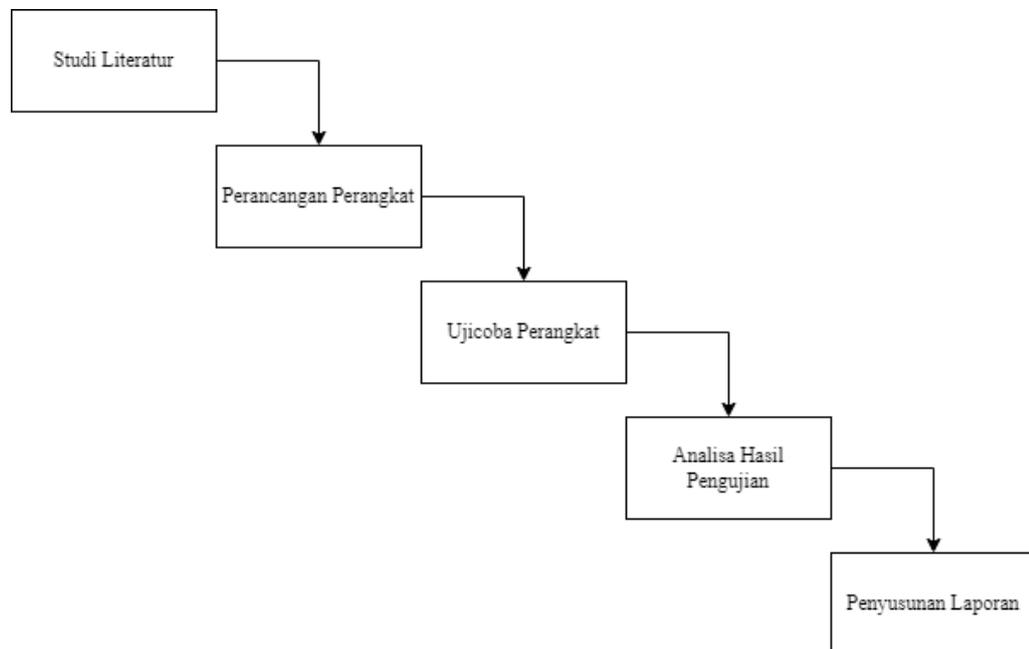
Alat Tulis Saldo Digital yang akan dikembangkan ini nantinya dapat mempermudah pemilik kost untuk menjual saldo listrik untuk kamar yang disewa pemilik kost supaya mempermudah pemilik kost dalam membayar tagihan listrik. Dalam pembuatan alat tulis saldo digital ini dibutuhkan fitur dari sensor RFID NFC yang dapat melakukan penulisan saldo serta melakukan pembacaan saldo yang terdapat didalam kartu. Sistem yang terdapat pada alat ini sangat bergantung pada koneksi internet dan terintegrasi dengan mikrokontroler ESP32 serta teknologi IoT.

Fungsi utama dari alat ini adalah mengisi saldo, di mana pengguna dapat memasukkan nominal saldo melalui keypad. Data yang dimasukkan akan diproses oleh ESP32 dan disimpan pada kartu RFID NFC. Selain itu, alat ini juga memiliki fungsi untuk mengecek saldo, di mana pengguna dapat mengecek saldo yang tersimpan pada kartu RFID dengan menekan menu yang tersedia dan informasi saldo akan ditampilkan pada layar LCD. Alat ini juga memiliki fitur cut-off yang memungkinkan pemilik kost untuk mematikan aliran listrik ke kamar tertentu berdasarkan ID kamar.

Cara kerja alat ini dimulai dengan alat yang harus terhubung dengan internet agar dapat berfungsi. Setelah menyala, alat akan menampilkan menu utama pada layar LCD. Pengguna kemudian memilih menu dan memasukkan nominal saldo melalui keypad. Data nominal saldo dikirim ke server dan diproses oleh sistem backend menggunakan MQTT dan Node-RED. Jika pengisian saldo berhasil, data saldo akan ditulis ke kartu RFID menggunakan modul PN532. Pengguna juga dapat mengecek saldo kapan saja dengan menekan menu cek saldo.

### 3.4 Tahapan Kegiatan

Dalam penyelesaian tugas akhir ini, terdapat beberapa metode yang digunakan untuk memastikan kelancaran dan kesesuaian dengan rencana yang telah dibuat. Pendekatan yang diterapkan melibatkan studi literatur, perancangan perangkat, uji coba perangkat, analisa hasil pengujian, dan penyusunan laporan.



Gambar 3.1 Alur Tahapan Kegiatan

#### 3.4.1 Studi Literatur

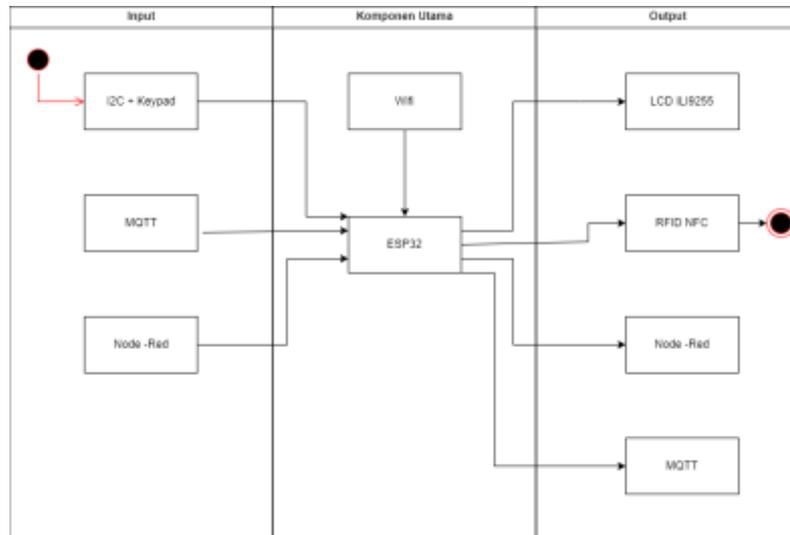
Studi literatur yang dilakukan adalah melakukan pencarian dari berbagai sumber informasi yang relevan untuk dijadikan referensi dalam menghadapi permasalahan yang sedang dikaji. Referensi tersebut didapat dari berbagai jenis sumber, termasuk jurnal ilmiah, buku, laporan penelitian, artikel, dan situs internet yang selaras dengan judul. Studi Literatur ini bertujuan untuk memperkuat pemahaman terhadap permasalahan yang dikaji dan memperkuat dasar teori dalam pengembangan alat, sehingga sesuai dengan rencana yang telah dirancang.

#### 3.4.2 Perancangan Perangkat

##### a. Perancangan Perangkat Keras

Dalam tahap perancangan, penulis akan membuat alat berupa rangan dari bahan-bahan yang sudah disediakan, yaitu perangkat keras, bahan bahan dan peralatan yang dibutuhkan. Untuk peralatan dan yang digunakan dalam kegiatan ini antara lain ESP32, PN532, LCD ILI9255, I2C, *Keypad*, *Stepdown Voltage*,

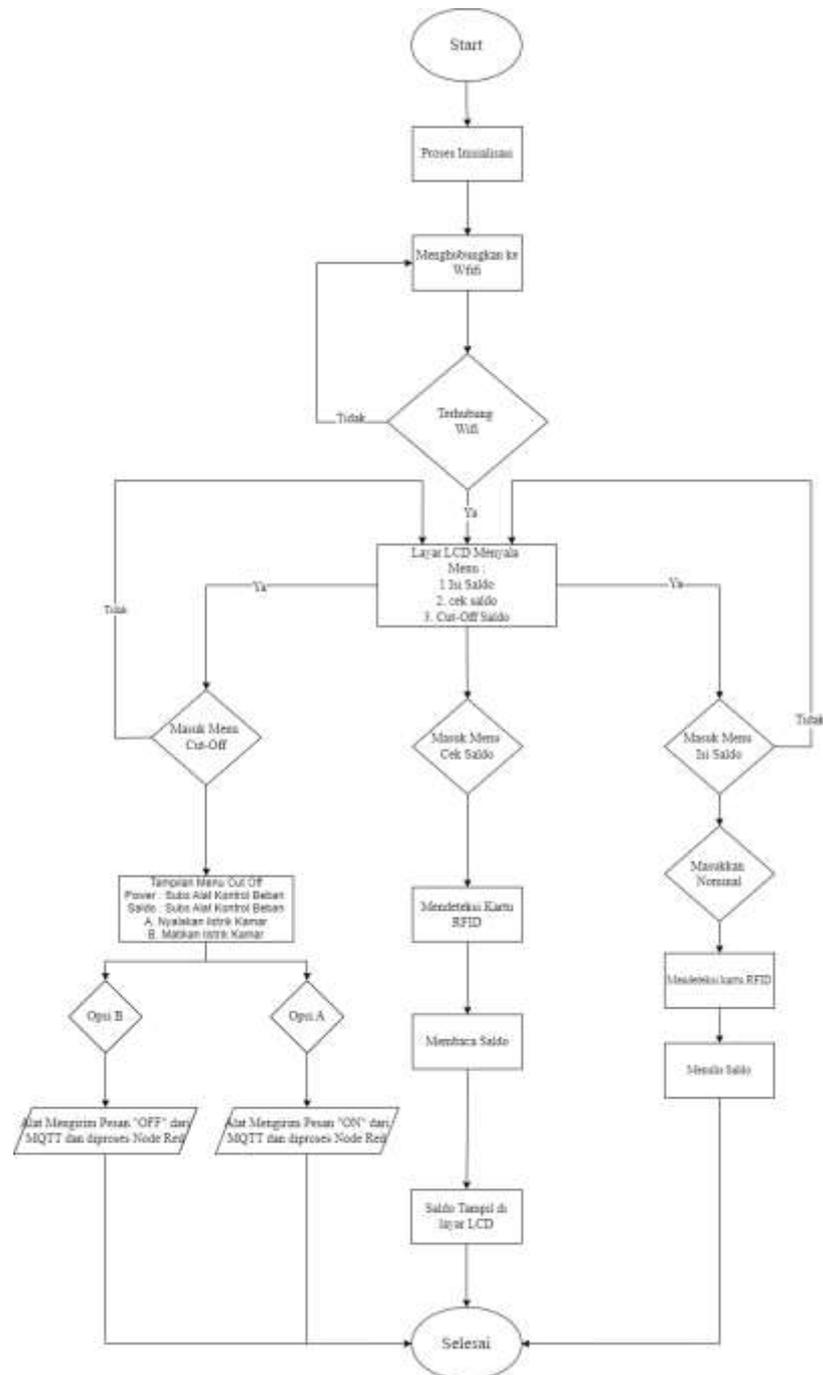
*Type C power input*, kabel fleksibel dan alat pendukung lainnya. Berikut alur dan skema pemasangan komponen dapat dilihat pada gambar dibawah ini :



Gambar 3.2 Diagram Blok Alat

Alat Tulis Saldo digital ini dilengkapi oleh beberapa komponen input dan output. Untuk komponen input sendiri terdiri dari *Type c power input*, *stepdown voltage*, *I2C*, *Keypad*, Dan *WiFi*. Untuk *Type c* sendiri digunakan sebagai sumber daya yang digunakan untuk menyalakan semua komponen yang terdapat pada alat yang kemudian akan disalurkan ke *Stepdown voltage* terlebih dahulu. Karena *power input* pada umumnya memiliki tegangan diatas 5volt maka tegangan listrik yang masuk akan diperkecil terlebih dahulu menggunakan *stepdown voltage* menjadi tegangan yang lebih rendah. *I2C* sendiri digunakan untuk mengontrol *keypad* sehingga apabila tombol *keypad* ditekan maka *i2c* akan mengirimkan informasi yang akan diproses oleh *ESP32* untuk dapat ditampilkan pada *LCD*. Modul *P532* sendiri berfungsi untuk memasukkan token digital pada kartu *rfid*.

b. Flowchart Sistem



Gambar 3.3 Desain Flowchart Sistem.

Desain flowchat yang terdapat pada gambar 3.3 diatas menggambarkan cara kerja alat tulis token digital. Saat alat menyala proses dimulai dengan inisiasi sistem dan dilanjutkan dengan proses menghubungkan perangkat dengan koneksi internet. Setelah itu sistem akan memeriksa apakah berhasil terhubung dengan koneksi internet atau gagal terhubung dengan koneksi internet. Jika perangkat berhasil

terhubung maka sistem akan menampilkan menu pada layar perangkat, jika belum maka sistem akan kembali menghubungkan ke jaringan wifi hingga berhasil.

Setelah terhubung, layar LCD akan menampilkan menu utama yang terdiri dari tiga opsi yaitu isi saldo, cek saldo, dan cut off. Jika pengguna memilih menu isi saldo maka sistem akan meminta pengguna untuk memasukkan nominal. Setelah nominal dimasukkan maka sistem akan mendeteksi kartu RFID dan menulis saldo yang diinputkan ke kartu tersebut. Jika pengguna memilih opsi cek saldo sistem akan membaca saldo yang tersimpan pada kartu dan menampilkan saldo tersebut pada layar perangkat. Untuk opsi Cut Off juga menampilkan informasi mengenai status power dan saldo yang diambil dari alat kontrol beban listrik.

#### 3.4.3 Ujicoba Perangkat.

Pada tahap ini dilakukan serangkaian pengujian alat agar dapat membantu penulis dalam melakukan pengerjaan alat dan dapat mengerti segala sesuatu yang akan dihadapi pada saat pengerjaan alat mengalami gangguan atau terjadi *error* pada perangkat keras dan perangkat lunak yang diharapkan sistem dapat bekerja kembali secara normal kembali sesuai dengan yang diinginkan. Pengujian ini terdiri dari beberapa hal berikut :

1. Pengujian Keypad apakah berfungsi dengan baik.
2. Pengujian modul PN532 apakah dapat membaca dan menulis pada kartu dengan baik.
3. Pengujian LCD apakah dapat menampilkan data dengan baik.
4. Pengujian alat apakah dapat terhubung dengan MQTT serta mengirim data dan menerima data dengan baik.

#### 3.4.4 Analisa Hasil Pengujian

Setelah melakukan pengujian pada alat yang penulis buat dan digabungkan dengan alat yang lainnya, penulis akan menganalisa sekaligus melakukan perbaikan apabila terjadi kesalahan saat perancangan alat. Penulis akan membuat laporan pada setiap pengujian yang dilakukan.

### 3.4.5 Penyusunan Laporan

Setelah melaksanakan seluruh rangkaian kegiatan penulis akan membuat laporan Tugas Akhir . Beserta dokumentasi kegiatan yang penulis lakukan selama pengerjaan tugas akhir ini.

### 3.5 Jadwal Pelaksanaan Kegiatan

Pada tahap pembuatan dibutuhkan waktu 5 (lima) bulan untuk pengerjaan penelitian ini dengan mengacu pada langkah-langkah metode kegiatan yang telah ditentukan. Berikut penjadwalan dalam pembuatan diuraikan pada tabel berikut :

Tabel 3.2 Tabel Kegiatan

No	Metode Kegiatan	Bulan ke				
		1	2	3	4	5
1	Studi Literatur	■	■			
2	Perancangan Perangkat		■			
3	Ujicoba Perangkat		■	■	■	
4	Analisa hasil Pengujian			■	■	■
5	Penyusunan Laporan				■	■

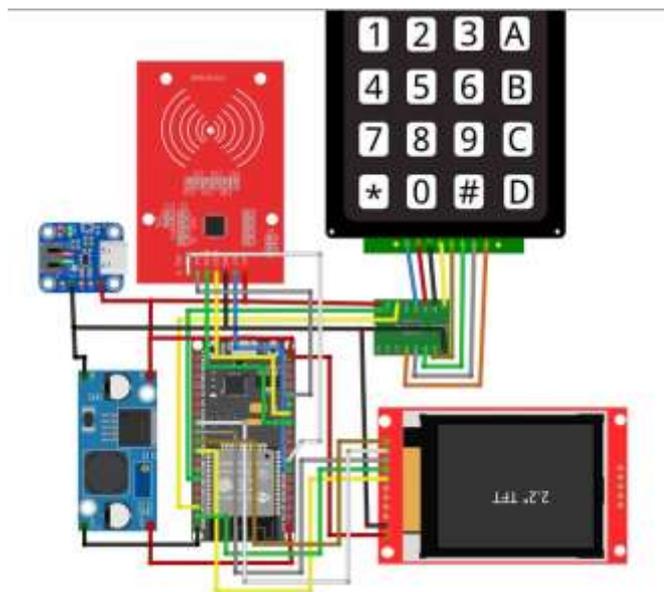
## BAB 4. HASIL DAN PEMBAHASAN

### 4.1 Hasil Stuti Literatur

Hasil studi literatur yang digunakan pada proses penyusunan tugas akhir ini dengan judul “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik” berasal dari berbagai sumber yang relevan sesuai dengan topik yang dibahas. Bahan referensi didapat dari berbagai jurnal, artikel skripsi website dan tugas akhir yang mendukung dalam pembuatan perangkat. Tujuannya adalah untuk memperkuat permasalahan yang dibahas dan dasar teori dalam pembuatan perangkat supaya sesuai dengan apa yang direncanakan.

### 4.2 Hasil Perancangan Alat

Perancangan perangkat keras yang digunakan dalam pembuatan “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik” membutuhkan beberapa komponen seperti keypad,i2c,stepdown voltage,Type C Power Input, LCD ILI9225, RFID NFC.



Gambar 4.1 Skema Fritzing Alat

Gambar diatas merupakan diagram alat yang sudah dibuat menggunakan aplikasi *fritzing*. Pada gambar terlihat banyak sekali warna garis kabel. Terdapat perbedaan warna kabel pada diagram alat menunjukkan bahwa setiap kabel tidak terhubung dan juga perbedaan warna kabel digunakan untuk mempermudah dalam proses perakitan. ESP32 38 pin adalah pusat kendali dari alat yang berperan menyimpan dan memproses kode program yang telah dibuat dan esp32 juga terhubung dengan konektivitas internet supaya alat dapat berjalan dengan sempurna. keypad digunakan untuk mengontrol menu dan memasukkan nominal token listrik yang akan ditulis pada kartu rfid nfc. Stepdown voltage adalah komponen yang digunakan untuk menurunkan tegangan menjadi 5v .Karena tegangan yang berasal dari stopkontak kebanyakan tegangannya diatas 5v maka diperlukan stepdown voltage untuk menurunkan tegangannya supaya alat tidak mengalami kerusakan. Lcd digunakan untuk menampilkan output dari keypad dan juga untuk menampilkan menu yang tersedia di dalam perangkat.RFID NFC digunakan untuk menuliskan token yang berupa saldo kedalam kartu nfc untuk mempermudah proses isi token listrik yang ada pada setiap kamar kost.

Tabel 4.1 Tabel Penggunaan Pin

<b>Komponen</b>	<b>Nama Pin</b>	<b>Pin</b>
RFID NFC	NFC SCK	13
	NFC MISO	12
	NFC MOSI	14
	NFC SS	33
	NFC RST	32
LCD	CS	5
	RST	16
	RS	17
	SDI	23
	CLK	18
KEYPAD I2C	SDA	21
	SCL	22

TYPE C	5V	VIN
	GND	Stepdown
Stepdown	3V	Semua pin 3v
	GND	Semua pin GND

Terdapat beberapa komponen perangkat yang digunakan dalam proses pembuatan “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik” seperti yang tertera pada table diatas diantaranya terdapat ESP32 dengan 38 pin, keypad i2c,lcd dan rfid nfc. Semua komponen tersebut terhubung dengan ESP32 yang akan memproses data yang didapat dari input keypad dan rfid nfc dan kemudian di tampilkan pada layar lcd ili9225.

### 4.3 Kode Konfigurasi Perangkat

Setelah semua komponen telah di rancang menjadi satu kesatuan maka selanjutnya yang perlu dilakukan adalah pembuatan kode program untuk alat yang dibuat supaya dapat bekerja sesuai keinginan, dengan memanfaatkan software Arduino IDE. Penulis membutuhkan beberapa library untuk membuat program berjalan dengan sempurna seperti yang tertera pada tabel (NOMOR TABEL) dibawah ini.

Tabel 4.2 Tabel Library Kode Program

No	Library	Keterangan
1	<SPI.h>	Digunakan untuk mengatur komunikasi dengan perangkat SPI dengan layar TFT dan modul NFC PN532
2	<TFT_22_ILI9225.h>	Digunakan untuk mengatur serta mengontrol layar yang terdapat pada alat, mulai dari mengatur text dan orientasi layar
3	<Wire.h>	Digunakan untuk mengatur dan mengontrol keypad melalui komponen i2c

---

4	<Keypad_I2C.h>	Digunakan untuk menginisialisasi dan membaca inputan yang berasal dari keypad
5	<Adafruit_PN532.h>	Digunakan untuk mengontrol dan mengatur modul PN532
6	<WiFi.h>	Digunakan untuk menghubungkan esp32 ke dalam jaringan wifi dengan SSID dan password yang sudah di setel
7	<PubSubClient.h>	Digunakan untuk menghubungkan ESP32 ke broker MQTT

---

Terdapat 3 menu yang terdapat pada alat yang penulis buat terdiri dari menu isi saldo, cek saldo, cut off. Berikut merupakan kode program untuk masing masing menu :

a) Kode program Menu Isi Saldo

Kode program isi saldo ini digunakan untuk pembelian token listrik berupa saldo pada pemilik kost. Saat pemilik kost melakukan pengisian saldo, pemilik kost hanya memasukkan nominal pada alat tulis token digital yang penulis buat. Saldo yang telah dibeli akan dimasukkan kedalam kartu rfid yang kemudian dapat digunakan pada alat kontrol beban listrik untuk menyalakan listrik yang ada pada setiap kamar kost. Berikut kode program 4.1 :

**Kode Program 4.1 Isi Saldo :**

```

1) void isisaldo() {
2) int nominal = readNominalFromKeypad();
3) tft.clear(); // Hapus teks sebelumnya
4) printText(10, 10, "Nominal: " + String(nominal));
5) printText(10, 30, "Tap Kartu");
6) writeNominalToCard(2, 1, nominal);
7) String payload = String(nominal);
8) client.publish(MQTT_TOPIC_SALDO_IN, payload.c_str());
9) delay(4000);
10) tft.clear();
11) display = '0';

```

```
12) }
13) void writeNominalToCard(uint8_t block, uint8_t sector, int
    nominal) {
14) char message[6];
15) snprintf(message, sizeof(message), "%d", nominal);
16) writeMessageToCard(block, sector, message);
17) }
18) void writeMessageToCard(uint8_t block, uint8_t sector, char
    message[6]) {
19) uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };
20) uint8_t uidLength;
21) if (nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
    &uidLength)) {
22) uint8_t keyData[] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
23) if (nfc.mifareclassic_AuthenticateBlock(uid, uidLength, sector,
    0xFF, keyData)) {
24) printText(10, 50, "Autentikasi ");
25) printText(10, 70, "Berhasil");
26) printText(10, 90, "Kembali (0)");
27) uint8_t messageData[6];
28) for (int i = 0; i < 6; i++) {
29) if (i < strlen(message)) {
30) messageData[i] = message[i];
31) } else {
32) messageData[i] = 0x00;
33) }
34) }
35) if(nfc.mifareclassic_WriteDataBlock(block,messageData)) {
36) Serial.print("Writing Message to Block ");
37) Serial.print(block);
38) Serial.println(" successful.");
39) } else {
40) Serial.println("Failed to write data to the card.");
41) }
42) } else {
43) Serial.println("Authentication failed.");
44) }
45) }
```

46) }

b) Kode program menu cek saldo

Kode program ini digunakan untuk melakukan pengecekan saldo yang sudah dimasukkan kedalam kartu rfid. Penulis memasukkan menu ini supaya pemilik kost dapat melakukan pengecekan apakah saldo yang sudah diisi benar benar masuk di dalam kartu dan dapat digunakan pada alat kontrol beban listrik. Menu ini digunakan untuk meminimalisir terjadinya kegagalan saat melakukan pengisian saldo oleh pemilik kost. Berikut kode program 4.2 :

**Kode program 4.2 cek saldo :**

```

1) void readMessageFromCard(uint8_t block, uint8_t sector) {
2)   uint8_t success;
3)   uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };
4)   uint8_t uidLength;
5)   success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
      &uidLength);
6)   if (success) {
7)     idcard = "";
8)     for (byte i = 0; i < uidLength; i++) {
9)       idcard += (uid[i] < 0x10 ? "0" : "") + String(uid[i], HEX);
10)    }
11)    printText(10, 10, "ID CARD : ");
12)    printText(120, 10, idcard);
13)    uint8_t keyData[] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
14)    if (nfc.mifareclassic_AuthenticateBlock(uid, uidLength,
      sector, 0xFF, keyData)) {
15)      Serial.println("Authentication successful.");
16)      uint8_t readBlock[6];
17)      if (nfc.mifareclassic_ReadDataBlock(block, readBlock)) {
18)        Serial.print("Read Data from Block ");
19)        Serial.print(block);
20)        Serial.print(": ");
21)        for (byte i = 0; i < 6; i++) {
22)          Serial.print(readBlock[i], HEX);
23)          Serial.print(" ");
24)        }

```

```

25)         Serial.println();
26)         printText(10, 30, "Saldo : ");
27)         printText(100, 30, (char*)readBlock);
28)         printText(10, 50, "Kembali (0)");
29)     delay(4000);
30)         tft.clear();
31)         display = '0';
32)     } else {
33)         Serial.println("Failed to read data from the card.");
34)     }
35) } else {
36)     Serial.println("Authentication failed.");
37)     delay(4000);
38)     tft.clear();
39)     display = '0';
40) }
41) }
42) }

```

### c) Kode Program Cut Off

Kode program ini digunakan untuk melakukan cut off pada aliran listrik. Cut off disini memanfaatkan relay yang dikontrol melalui mqtt menggunakan pesan “ON dan pesan “OFF” untuk mengontrolnya. Penulis memasukkan menu ini digunakan untuk fitur keamanan apabila saldo di tiap kamar habis namun arus listrik masih berjalan pemilik kost dapat memutus aliran listrik pada kamar melalui alat tulis token ini. Pada menu Cut Off terdapat beberapa tampilan seperti informasi power listrik, saldo kamar, dan informasi tombol untuk melakukan kontrol relay. Informasi power dan saldo ini didapat dari alat kontrol beban listrik yang ada pada setiap kamar kost.

#### **Kode program 4.3 Cut Off :**

```

1) void cutoff() {
2)   if (millis() >= time_1 + interval) {
3)     time_1 = millis();
4)     printText(50, 10, "Cut Off Menu");
5)     printText(10, 30, "Power : " + power + "W");

```

```

6) printText(10, 50, "Saldo : " + Sisa_saldo_strin);
7) printText(10, 70, "A. Mati K1");
8) printText(10, 90, "B. Nyala K1");
9) printText(10, 110, "C. Mati K2");
10)  printText(10, 130, "D. Nyala K2");
11)  }
12)  void sendRelayCommand(const char* topic, const char* command)
      {
13)  if (client.publish(topic, command)) {
14)  tft.clear();
15)  printText(10, 10, "Command Sent:");
16)      printText(10, 30, command);
17)      delay(2000);
18)      tft.clear();
19)      display = '0';
20)  } else {
21)      tft.clear();
22)      printText(10, 10, "Failed to send:");
23)      printText(10, 30, command);
24)      delay(2000);
25)      tft.clear();
26)      display = '0';
27)  }
28)  }

```

#### 4.4 Hasil Pembuatan Alat

Penulis membuat case untuk semua komponen pada alat ini memanfaatkan printer 3D yang ukurannya sudah ditentukan sebelumnya. Bahan yang digunakan adalah filament berwarna biru langit sebanyak 1 roll. Untuk case yang digunakan untuk tempat komponen memiliki spesifikasi lebar 16 cm x 9 cm dengan ketinggian 3,5cm. Case juga dilengkapi lubang untuk layar yang memiliki ukuran 5,5x 4cm dan untuk lubang keypad berukuran 6x6cm. Berikut hasil gambar 4.2 hasil case bagian atas alat dapat dilihat pada gambar dibawah ini.



Gambar 4.2 Case alat tampak atas

Untuk komponen perangkat mulai dari pcb, i2c, type c power input, dan stepdown penulis rekatkan pada bagian dalam case bagian belakang yang dihadapkan keatas. hal ini supaya semua komponen aman dan perangkat menjadi lebih kompak dan tahan terhadap guncangan. Pcb pada alat ini terdapat dua lapisan, pada lapisan pertama atau lapisan bawah berisikan komponen i2c, type c power input, stepdown dan Esp32. Kemudian pada lapisan ke 2 berisikan LCD dan RFID, alasan penggunaan dua lapisan pada pcb untuk alat tulis saldo digital ini supaya bentuk perangkat tidak terlalu besar dan menjadi efisien tanpa mengorbankan fungsionalitasnya .Selain itu alasan penggunaan dua lapisan pcb ini tidak hanya membuat perangkat yang dibuat menjadi lebih kecil tetapi juga nmembuat semua komponen berfungsi sempurna dan mempermudah proses perawatan dan perbaikan jika terjadi kerusakan pada komponen. Berikut hasil dari bagian dalam alat “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik”. Dapat dilihat pada gambar 4.3 dibawah ini :



Gambar 4.3 Bagian dalam alat tampak atas

Hasil keseluruhan alat yang telah dibuat pada alat yang tampak dari atas dapat dilihat pada gambar 4.4 dibawah ini. Pada casing bagian atas terdapat komponen layar dan Keypad 4x4 yang digunakan untuk mengoperasikan alat. Untuk mengoperasikan alat “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik” adalah dengan menghubungkan port USB type C yang terdapat pada bagian samping alat. Setelah perangkat terhubung dengan sumber daya melalui port usb type c yang sudah tersedia pada perangkat, maka perangkat akan menyala dan secara otomatis ke jaringan internet. Kemudian setelah perangkat terhubung dengan jaringan internet layar akan menampilkan tiga menu utama yaitu isi saldo, cek saldo, dan cut off. Menu ini dirancang untuk memberikan kemudahan bagi pengguna untuk mengakses fungsi utama dari perangkat ini. Untuk memilih menu yang tampil di layar perangkat, pengguna dapat menggunakan keypad pada alat dengan menekan nomor yang sesuai dengan pilihan menu pada layar perangkat. Misalnya untuk pengisian saldo, maka pengguna hanya perlu menekan nomor yang sesuai dengan opsi “isi saldo” begitu juga dengan menu cek saldo dan cutoff. Berikut hasil dari tampilan perangkat “Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik”. Dapat dilihat pada gambar 4.4 dibawah ini :



Gambar 4.4 Alat setelah disatukan dan dinyalakan

## 4.5 Hasil Pengujian

Pada tahap pengujian alat Rancang Bangun Alat Tulis Saldo Digital Berbasis Esp32 Sebagai Pendukung Alat Kontrol Beban Listrik akan dilakukan pendataan mengenai keberhasilan dan kegagalan alat dalam proses ujicoba. Terdapat beberapa tahap pengujian yang dilakukan pada alat yang akan dilaksanakan sebagai berikut :

- 1) Pengujian kode program yang terdapat pada mikrokontroler ESP32 yang digunakan pada alat.
- 2) Pengujian Menu Isi saldo dan Cek Saldo.
- 3) Pengujian Menu Cut Off.

### 4.5.1 Pengujian Menu Isi Saldo dan Cek Saldo

Pengujian ini penulis menghubungkan 3 komponen yaitu layar LCD, Keypad dan modul RFID PN532, cara pengujian menu ini dengan cara layar menampilkan input nominal yang dimasukkan melalui keypad kemudian setelah saldo yang

dimasukkan sudah benar maka perangkat akan mengirim instruksi untuk melakukan tap kartu untuk menulis nominal pada kartu RFID. Saldo yang dimasukkan berupa nominal saja tidak termasuk dengan tambahan symbol. Apabila nominal yang dimasukkan dikombinasikan dengan symbol maka symbol tersebut tidak akan terdeteksi oleh alat yang penulis buat. Dan untuk mengetahui saldo yang dimasukkan sudah berhasil atau belum maka pengujian isi saldo dan cek saldo akan dilakukan sekaligus.

Tabel 4.3 Percobaan Isi Saldo dan Cek Saldo

No	Nominal saldo	Simbol	Digit	Status	Cek Saldo
1	100	*	3	Berhasil	<b>100</b>
2	200	D	3	Berhasil	<b>200</b>
3	300	B	3	Berhasil	<b>300</b>
4	400	C	3	Berhasil	<b>400</b>
5	5.000		4	Berhasil	<b>5000</b>
6	65.000		5	Berhasil	<b>65000</b>
7	75.000		5	Berhasil	<b>75000</b>
8	80.000		5	Berhasil	<b>80000</b>
9	95.000		5	Berhasil	<b>95000</b>
10	100.000		6	Gagal	<b>10000</b>

Berdasarkan tabel 4.3 diatas , dilakukan percobaan dan pengujian menu isi saldo dan cek saldo sekaligus supaya dapat mengetahui nominal yang masukkan pada menu isi saldo berhasil ditulis pada kartu rfid melalui sensor PN532. Berikut Hasil percobaan yang telah dilakukan :

Pada percobaan 1 hingga 4 dilakukan pengisian saldo dengan nominal 100 hingga 400 rupiah disertai dengan adanya symbol di akhir nominal. Setelah itu kartu RFID ditempelkan pada layar lcd, dikarenakan sensor RFID PN532 berada tepat dibawah layar LCD. Setelah pengisian saldo berhasil maka dilakukan pengecekan yang ada di menu 2 yaitu menu Cek Saldo. Setelah memilih menu Opsi 2 maka layar akan menampilkan instruksi untuk menempelkan kartu pada layar yang kemudian kartu akan dibaca oleh sensor rfid nfc. Dan hasil yang didapat dari kartu

yang dibaca memiliki nilai saldo yang sesuai dengan yang dimasukkan sesuai dengan percobaan pertama hingga ke keempat.

Kemudian pada percobaan ke lima penulis melakukan pengisian saldo dengan nominal 5.000 rupiah pada alat. setelah melakukan pengisian penulis melakukan pengecekan saldo untuk percobaan ke lima. Hasil yang didapatkan sesuai dengan nominal yang di input oleh penulis sebesar 5.000.

Dan untuk percobaan yang terakhir penulis melakukan uji coba pengisian saldo sebesar 100.000 rupiah yang merupakan saldo memiliki 6 digit nominal, maka dilakukan pengecekan apakah saldo yang dimasukkan tetap masuk atau tidak. Hasil yang didapatkan saldo berhasil dimasukkan namun nominal yang dimasukkan menjadi 10.000 karena alat diatur untuk menerima message hanya 5 digit saja ditambah *null terminator*.

#### 4.5.2 Pengujian Menu Cutt Off

Pada pengujian ini penulis menghubungkan alat Alat Tulis Saldo Digital Berbasis Esp32 dengan Alat kontrol beban listrik yang terdapa pada setiap kamar kost. Cara pengujian menu ini adalah alat tulis token mengirimkan pesan berupa “ON” atau “OFF” melalui mqtt menuju node red, kemudian pesan tersebut akan dikelola oleh node red dan kemudian diterukan menuju alat kontrol beban listrik untuk memproses perintah yang sudah dikelola oleh node-red. Berikut merupakan hasil pengujian yang sudah dilakukan terdapat pada tabel 4.4.

Tabel 4.4 Percobaan Menu Cut Off

<b>Percobaan</b>	<b>Nomor kamar</b>	<b>Perintah</b>	<b>Status relay</b>	<b>Hasil</b>
1.	Kamar	On	On	<b>Berhasil</b>
2.	Kamar	Off	Off	<b>Berhasil</b>
3.	Kamar	On	On	<b>Berhasil</b>
4.	Kamar	Off	Off	<b>Berhasil</b>
5.	Kamar	On	On	<b>Berhasil</b>
6.	Kamar	On	On	<b>Berhasil</b>
7.	Kamar	Off	Off	<b>Berhasil</b>
8.	Kamar	On	On	<b>Berhasil</b>
9.	Kamar	Off	Off	<b>Berhasil</b>
10.	Kamar	On	On	<b>Berhasil</b>

Pada tabel 4.4 penulis telah melakukan pengujian menu cut off. Tabel uji coba di atas memberikan rincian pengujian yang dilakukan untuk mengontrol relay pada kamar. Setiap percobaan melibatkan pemberian perintah untuk menyalakan ("On") atau mematikan ("Off") relay, kemudian mencatat status aktual relay setelah perintah diberikan. Kolom pertama menunjukkan urutan percobaan dari 1 hingga 10, sementara kolom kedua menunjukkan kamar yang diuji dalam setiap percobaan.

Pada percobaan 1 hingga 5. Hasil perintah untuk menyalakan relay pada percobaan 1 dan 3 berhasil, serta perintah untuk mematikan relay pada percobaan 2 dan 4 juga berhasil. Pada percobaan 5, relay menerima pesan "ON" dan hasilnya menunjukkan keberhasilan. Hal ini menunjukkan bahwa relay pada kamar berfungsi dengan baik sesuai dengan perintah yang diberikan.

Pada percobaan 6 hingga 10, penulis melakukan pengujian pada relay Kamar di hari yang berbeda. Hasilnya menunjukkan bahwa perintah untuk menyalakan relay pada percobaan 6, 8, dan 10 berhasil, sementara perintah untuk mematikan relay pada percobaan 7 dan 9 juga berhasil. Ini menunjukkan bahwa relay di Kamar berfungsi dengan baik sesuai dengan perintah yang diberikan.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem berhasil mengontrol relay untuk kedua kamar sesuai dengan perintah yang diberikan. Setiap

perintah "On" dan "Off" untuk relay di Kamar 1 dan Kamar 2 berhasil dijalankan dengan baik, dan status relay berubah sesuai dengan perintah yang diberikan. Semua percobaan berakhir dengan hasil "Berhasil", menandakan bahwa sistem kontrol relay berfungsi dengan baik dan dapat diandalkan untuk mengontrol relay di kedua kamar tersebut. Ini menunjukkan keandalan dan konsistensi dari sistem kontrol yang digunakan dalam pengujian ini.

## **BAB 5. KESIMPULAN DAN SARAN**

### **5.1 Kesimpulan**

Berdasarkan hasil pengujian yang dilakukan, alat Tulis Saldo digital berbasis ESP32 berhasil berfungsi sesuai dengan yang diharapkan. Semua percobaan menunjukkan bahwa perintah untuk menyalakan dan mematikan relay pada kamar berjalan dengan lancar, menunjukkan keberhasilan dalam implementasi sistem kontrol beban listrik. Alat ini mampu memberikan solusi efektif untuk manajemen penggunaan listrik di lingkungan kost, memudahkan pemilik dalam penjualan dan pemantauan penggunaan listrik.

### **5.2 Saran**

Dari hasil rancang bangun alat Tulis Saldo digital berbasis esp32 sebagai pendukung alat kontrol beban listrik masih terdapat kekurangan yang memungkinkan untuk terjadinya pengembangan lebih lanjut. Maka saran yang dapat penulis sampaikan adalah sebagai berikut ;

1. Menggunakan Sensor RFID NFC yang lebih baik agar dapat membaca dan menulis token pada kartu secara maksimal.
2. Selain itu dapat dikembangkan dengan adanya baterai sebagai Cadangan daya apabila terjadi suatu gangguan pada daya alat.

## DAFTAR PUSTAKA

- Asrar, L. D., Studi, P., & Elektro, T. (2013). *RANCANG BANGUN PROTOTIPE kWh METER PRABAYAR DENGAN SISTEM PENGISIAN MENGGUNAKAN KARTU RADIO FREQUENCY IDENTIFICATION ( RFID )*. 40–43.
- Budiyanta, N. E., Wishnu, M. C., W, D. R., & Lukas, L. (2019). Perancangan Fidget Device Berbasis Internet Of Things. *TESLA: Jurnal Teknik Elektro*, 21(1), 1. <https://doi.org/10.24912/tesla.v21i1.3241>
- Dita, P. E. S., Fahrezi, A. Al, Prasetyawan, P., & Amarudin, A. (2021). Sistem Keamanan Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino UNO R3. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 121–135. <https://doi.org/10.33365/jtikom.v2i1.111>
- Fajrur, A. (2020). Pengenalan Node-Red. *Wiki.Rdd-Tech.Com*. <https://wiki.rdd-tech.com/index.php/knowledge-base/pengenalan-node-red/>
- Fikri, A. (2022). *Pengujian unjuk kerja transmisi data lora 433 mhz dan 915 mhz non line of sight (nlos)*. 1–67.
- Hergika, G., Siswanto, & S, S. (2021). Perancangan Internet of Things (Iot) Sebagai Kontrol Infrastruktur Dan Peralatan Toll Pada Pt. Astra Infratoll Road. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 8(2), 86–98. <https://doi.org/10.30656/prosisko.v8i2.3862>
- Ibrahim, Ridyandhika Riza , Bekti Yulianti, S. M. (2022). RANCANG BANGUN MONITORING PEMAKAIAN ARUS LISTRIK PLN BERBASIS IoT. *Jurnal Teknologi Industri*, 11(1), 43–51.
- Kandasamy, K., Srinivas, S., Achuthan, K., & Rangan, V. P. (2020). IoT cyber risk: a holistic analysis of cyber risk assessment frameworks, risk vectors, and risk ranking process. *Eurasip Journal on Information Security*, 2020(1). <https://doi.org/10.1186/s13635-020-00111-0>
- Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0268-2>
- Kusumah, H., & Pradana, R. A. (2019). Penerapan Trainer Interfacing Mikrokontroler Dan Internet of Things Berbasis Esp32 Pada Mata Kuliah Interfacing. *Journal CERITA*, 5(2), 120–134. <https://doi.org/10.33050/cerita.v5i2.237>
- Priyono, M., Sulistyanto, T., Nugraha, D. A., Sari, N., Karima, N., & Asrori, W. (2015). Implementasi IoT (Internet of Things) dalam pembelajaran di Universitas Kanjuruhan Malang Muhammad Priyono Tri Sulistyanto1, Danang Aditya Nugraha2, Nurfatika Sari3, Novita Karima, Wahid Asrori. *SMARTICS Journal*, 1(1), 20–23.
- Rumimper, R., Sompie, S. R. U. A., & Mamahit, D. J. (2018). Rancang Bangun Alat Pengontrol Lampu Dengan Bluetooth Berbasis Android. *E-Journal*

*Teknik Elektro Dan Komputer*, 5(3), 24–33.

- Siregar, R. R. A., Sikumbang, H., & Pasaribu, R. J. (2018). Model Pengisian Pulsa Listrik Kwh Meter Dengan Smart Card. *Jetri : Jurnal Ilmiah Teknik Elektro*, 16(1), 39–54. <https://doi.org/10.25105/jetri.v16i1.2914>
- Sudimanto. (2017). Pengisian Pulsa (Token) Listrik menggunakan SMS (Short Messages Services). *Sekolah Tinggi Manajemen Informatika Dan Komputer LIKMI, Bandung*, 16(2), 20–24.
- Sulistyowat, R., & Febriantoro, D. D. (2015). Perancangan Prototype Sistem Kontrol Dan Monitoring Pembatas Daya Listrik Berbasis Mikrokontroler. *Jurnal Iptek*, 16, 10–21. <http://jurnal.itats.ac.id/wp-content/uploads/2013/06/4.-RINY-FINAL-hal-24-32.pdf>
- Sulistyowati, D., Dewanta, F., & Ph, D. P. (2021). *PERANCANGAN DAN IMPLEMENTASI SMART WEIGHT SCALE MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD ( AES ) DALAM SISTEM TELEMEDICINE DESIGN AND IMPLEMENTATION OF SMART WEIGHT SCALE*. 8(2), 1560–1569.
- Susanto, B. M., Atmadji, E. S. J., & Brenkman, W. L. (2018). Implementasi Mqtt Protocol Pada Smart Home Security Berbasis Web. *Jurnal Informatika Polinema*, 4(3), 201. <https://doi.org/10.33795/jip.v4i3.207>
- Zurairah, M., Adam, M., Harahap, P., & Zaharuddin, Z. (2022). Sistem Keamanan Brankas Berbasis Mikrokontroler Atmega 328 Dengan Munggunakan Kode One Time Password (OTP). *Jurnal MESIL (Mesin Elektro Sipil)*, 3(1), 1–6. <https://doi.org/10.53695/jm.v3i1.681>

## LAMPIRAN

### Lampiran 1 Kode program alat

```
1) #include <SPI.h>
2) #include <TFT_22_ILI9225.h>
3) #include <Wire.h>
4) #include <Keypad_I2C.h>
5) #include <Adafruit_PN532.h>
6) #include <WiFi.h>
7) #include <PubSubClient.h>
8)
9) #define PN532_SCK (13)
10)    #define PN532_MISO (12)
11)    #define PN532_MOSI (14)
12)    #define PN532_SS (33)
13)
14)    #define TFT_CS 5
15)    #define TFT_RST 16
16)    #define TFT_RS 17
17)    #define TFT_SDI 23
18)    #define TFT_CLK 18
19)    #define TFT_LED 4
20)
21)    #define WIFI_SSID "KOSAN KAMAR DEPAN"
22)    #define WIFI_PASSWORD "ngooooook"
23)    #define MQTT_SERVER "192.168.100.77"
24)    #define MQTT_PORT 1883
25)    #define MQTT_TOPIC ""
26)    #define MQTT_TOPIC_PZEMIN3 "pzemin3/power"
27)    #define MQTT_TOPIC_SISA_SALDO "home/Sisa_saldo_strin"
28)    #define MQTT_TOPIC_RELAY1 "home/relay1"
29)    #define MQTT_TOPIC_RELAY2 "home/relay2"
30)
31)    TFT_22_ILI9225 tft = TFT_22_ILI9225(TFT_RST, TFT_RS, TFT_CS,
    TFT_SDI, TFT_CLK, TFT_LED);
32)    Adafruit_PN532 nfc(PN532_SCK, PN532_MISO, PN532_MOSI,
    PN532_SS);
33)    String idcard, power, Sisa_saldo_strin;
```

```
34)
35)     const byte ROWS = 4; // Four rows
36)     const byte COLS = 4; // Four columns
37)     char keys[ROWS][COLS] = {
38)         { 'D', '#', '0', '*' },
39)         { 'C', '9', '8', '7' },
40)         { 'B', '6', '5', '4' },
41)         { 'A', '3', '2', '1' }
42)     };
43)
44)     byte rowPins[ROWS] = { 0, 1, 2, 3 }; //connect to the
        row pinouts of the keypad
45)     byte colPins[COLS] = { 4, 5, 6, 7 }; //connect to the
        column pinouts of the keypad
46)
47)     Keypad_I2C kpd(makeKeymap(keys), rowPins, colPins,
        ROWS, COLS, 0x20, PCF8574);
48)
49)     WiFiClient wifiClient;
50)     PubSubClient client(wifiClient);
51)     long time_1 = 0;
52)     int interval = 1000;
53)     char display = '0';
54)
55)     void callback(char* topic, byte* payload, unsigned int
        length) {
56)         String message = "";
57)         for (int i = 0; i < length; i++) {
58)             message += (char)payload[i];
59)         }
60)
61)         Serial.println(message);
62)         Serial.print("Message arrived [");
63)         Serial.print(topic);
64)         Serial.print("] ");
65)         if (strcmp(topic, MQTT_TOPIC_PZEMIN3) == 0) {
66)             power = message;
67)         }
```

```
68)         if (strcmp(topic, MQTT_TOPIC_SISA_SALDO) == 0) {
69)             Sisa_saldo_strin = message;
70)         }
71)     }
72)
73)     void setup() {
74)         tft.begin();
75)         tft.setOrientation(3); // Adjust screen orientation
as needed
76)         tft.setFont(Terminal12x16);
77)         tft.setBackgroundColor(COLOR_BLACK); // Set
background color to black
78)         tft.clear();
79)         nfc.begin();
80)
81)         uint32_t versiondata = nfc.getFirmwareVersion();
82)         if (!versiondata) {
83)             Serial.print("Didn't find PN53x board");
84)             while (1)
85)                 ;
86)         }
87)
88)         nfc.SAMConfig();
89)         Serial.println("Waiting for an NFC card...");
90)
91)         Wire.begin();
92)         kpd.begin(makeKeymap(keys));
93)         Serial.begin(115200);
94)         Serial.println("start");
95)
96)         // Connect to Wi-Fi
97)         WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
98)         while (WiFi.status() != WL_CONNECTED) {
99)             delay(1000);
100)            Serial.println("Connecting to WiFi...");
101)        }
102)        Serial.println("Connected to WiFi");
103)
```

```
104) // Connect to MQTT broker
105) client.setServer(MQTT_SERVER, MQTT_PORT);
106) client.setCallback(callback);
107) reconnect();
108) }
109)
110) void loop() {
111)
112) //baca keypad
113) char key = kpd.getKey();
114)
115) //ganti halaman jika input berbeda dengan halaman
    sebelumnya
116) if (key != NO_KEY) {
117)     display = key;
118)     Serial.println(key);
119)     tft.clear();
120) }
121)
122) //buat ganti halaman
123) switch (display) {
124)     case '1':
125)         printText(10, 10, "Masukkan Nominal:");
126)         isisaldo();
127)         break;
128)     case '2':
129)         printText(10, 10, "Silahkan Tap ");
130)         printText(10, 30, "Kartu");
131)         readMessageFromCard(2, 1);
132)         break;
133)     case '3':
134)         cutoff();
135)         break;
136)     case '0':
137)         displayMenu();
138)         break;
139)     case 'A':
140)         sendRelayCommand(MQTT_TOPIC_RELAY1, "ON");
```

```
141)         break;
142)     case 'B':
143)         sendRelayCommand(MQTT_TOPIC_RELAY1, "OFF");
144)         break;
145)     case 'C':
146)         sendRelayCommand(MQTT_TOPIC_RELAY2, "ON");
147)         break;
148)     case 'D':
149)         sendRelayCommand(MQTT_TOPIC_RELAY2, "OFF");
150)         break;
151)     default:
152)         break;
153) }
154)
155) // Handle MQTT client
156) if (!client.connected()) {
157)     reconnect();
158) }
159) client.loop();
160) }
161)
162) void isisaldo() {
163)     int nominal = readNominalFromKeypad();
164)     tft.clear(); // Hapus teks sebelumnya
165)     printText(10, 10, "Nominal: " + String(nominal));
166)     printText(10, 30, "Tap Kartu");
167)     writeNominalToCard(2, 1, nominal);
168)     String payload = String(nominal);
169)     client.publish(MQTT_TOPIC, payload.c_str());
170)     delay(4000);
171)     tft.clear();
172)     display = '0';
173) }
174) void reconnect() {
175)     // Loop until we're reconnected
176)     while (!client.connected()) {
177)         printText(10, 10, "MQTT CONNECTING...");
178)         // Attempt to connect
```

```
179)         if (client.connect("ESP323_Client")) {
180)             printText(10, 30, "connected");
181)             tft.clear();
182)             client.subscribe(MQTT_TOPIC_PZEMIN3);
183)             client.subscribe(MQTT_TOPIC_SISA_SALDO);
184)             break;
185)         } else {
186)             Serial.print("failed, rc=");
187)             Serial.print(client.state());
188)             Serial.println(" try again in 5 seconds");
189)             // Wait 5 seconds before retrying
190)             delay(5000);
191)         }
192)     }
193) }
194)
195) void readMessageFromCard(uint8_t block, uint8_t sector)
196) {
197)     uint8_t success;
198)     uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0, 0 };
199)     uint8_t uidLength;
200)     success =
nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
    &uidLength);
201)
202)     if (success) {
203)         idcard = "";
204)         for (byte i = 0; i < uidLength; i++) {
205)             idcard += (uid[i] < 0x10 ? "0" : "") +
String(uid[i], HEX);
206)         }
207)
208)         printText(10, 10, "ID CARD : ");
209)         printText(120, 10, idcard);
210)
211)         uint8_t keyData[] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF };
```

```
212)
213)         if          (nfc.mifareclassic_AuthenticateBlock(uid,
                uidLength, sector, 0xFF, keyData)) {
214)             Serial.println("Authentication successful.");
215)
216)             uint8_t readBlock[6];
217)             if          (nfc.mifareclassic_ReadDataBlock(block,
                readBlock)) {
218)                 Serial.print("Read Data from Block ");
219)                 Serial.print(block);
220)                 Serial.print(": ");
221)                 for (byte i = 0; i < 6; i++) {
222)                     Serial.print(readBlock[i], HEX);
223)                     Serial.print(" ");
224)                 }
225)                 Serial.println();
226)                 printText(10, 30, "Saldo : ");
227)                 printText(100, 30, (char*)readBlock);
228)                 printText(10, 50, "Kembali (0)");
229)
230)                 delay(4000);
231)                 tft.clear();
232)                 display = '0';
233)             } else {
234)                 Serial.println("Failed to read data from the
                card.");
235)             }
236)         } else {
237)             Serial.println("Authentication failed.");
238)             delay(4000);
239)             tft.clear();
240)             display = '0';
241)         }
242)     }
243) }
244)
245) void cutoff() {
246)     if (millis() >= time_1 + interval) {
```

```
247)         time_1 = millis();
248)         printText(50, 10, "Cut Off Menu");
249)         printText(10, 30, "Power : " + power + "W");
250)         printText(10, 50, "Saldo : " + Sisa_saldo_strin);
251)         printText(10, 70, "A. Nyala K1");
252)         printText(10, 90, "B. Mati K1");
253)         printText(10, 110, "C. Nyala K2");
254)         printText(10, 130, "D. Mati K2");
255)     }
256) }
257)
258) void printText(int x, int y, String text) {
259)     int lineHeight = 16; // Default font height
260)     int charWidth = 12; // Default font width
261)
262)     int cursorX = x;
263)     int cursorY = y;
264)
265)     for (int i = 0; i < text.length(); i++) {
266)         char character = text[i];
267)
268)         if (character == '\n') {
269)             cursorY += lineHeight;
270)             cursorX = x;
271)         } else {
272)             tft.drawChar(cursorX, cursorY, character);
273)             cursorX += charWidth;
274)         }
275)     }
276) }
277)
278) int readNominalFromKeypad() {
279)     String input = ""; // String untuk menyimpan input
    dari keypad
280)     while (true) {
281)         char key = kpd.getKey();
282)         if (key != NO_KEY) {
```

```
283)         if (key == '#') { // Jika '#' ditekan, proses
            input selesai
284)             break;
285)         } else {
286)             input += key; // Menambahkan karakter ke string
            input
287)         }
288)     }
289) }
290)     return input.toInt();
291) }
292)
293)     void writeNominalToCard(uint8_t block, uint8_t sector,
        int nominal) {
294)         char message[6];
295)         snprintf(message, sizeof(message), "%d", nominal);
296)         writeMessageToCard(block, sector, message);
297)     }
298)
299)     void writeMessageToCard(uint8_t block, uint8_t sector,
        char message[6]) {
300)         uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0, 0 };
301)         uint8_t uidLength;
302)
303)         if (nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A,
            uid, &uidLength)) {
304)             uint8_t keyData[] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
                0xFF };
305)
306)             if (nfc.mifareclassic_AuthenticateBlock(uid,
                uidLength, sector, 0xFF, keyData)) {
307)                 printText(10, 50, "Autentikasi ");
308)                 printText(10, 70, "Berhasil");
309)                 printText(10, 90, "Kembali (0)");
310)
311)                 uint8_t messageData[6];
312)                 for (int i = 0; i < 6; i++) {
313)                     if (i < strlen(message)) {
```

```
314)         messageData[i] = message[i];
315)         } else {
316)         messageData[i] = 0x00; // Fill the remaining
        bytes with 0x00
317)         }
318)     }
319)
320)     if      (nfc.mifareclassic_WriteDataBlock(block,
        messageData)) {
321)         Serial.print("Writing Message to Block ");
322)         Serial.print(block);
323)         Serial.println(" successful.");
324)     } else {
325)         Serial.println("Failed to write data to the
        card.");
326)     }
327) } else {
328)     Serial.println("Authentication failed.");
329) }
330) }
331) }
332)
333) void displayMenu() {
334)     printText(10, 10, "MENU");
335)     printText(10, 30, "1. ISI SALDO");
336)     printText(10, 50, "2. CEK SALDO");
337)     printText(10, 70, "3. CUT OFF");
338) }
339)
340) void sendRelayCommand(const char* topic, const char*
        command) {
341)     if (client.publish(topic, command)) {
342)         tft.clear();
343)         printText(10, 10, "Command Sent:");
344)         printText(10, 30, command);
345)         delay(2000);
346)         tft.clear();
347)         display = '0';
```

```
348)         } else {
349)             tft.clear();
350)             printText(10, 10, "Failed to send:");
351)             printText(10, 30, command);
352)             delay(2000);
353)             tft.clear();
354)             display = '0';
355)         }
356)     }
```