

**RANCANG BANGUN ATM BERAS *INTERNET OF THINGS* (IoT)
DENGAN *REAL-TIME DATABASE* BERBASIS ESP32**

(Skripsi)

Oleh

NAHDRUL RIVANDI



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

**RANCANG BANGUN ATM BERAS *INTERNET OF THINGS* (IoT)
DENGAN *REAL-TIME DATABASE* BERBASIS ESP32**

Oleh

NAHDRUL RIVANDI

Skripsi

Sebagai salah satu syarat untuk mendapat gelar

SARJANA TEKNIK

pada

Jurusan Teknik Elektro

Fakultas Teknik

Universitas Lampung



JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2023

ABSTRAK

RANCANG BANGUN ATM BERAS *INTERNET OF THINGS* (IoT) DENGAN *REAL-TIME DATABASE* BERBASIS *ESP32*

Oleh:

Nahdrul Rivandi

Bantuan beras merupakan sebuah program yang biasanya dilakukan oleh masyarakat dan pengurus masjid yang ingin menyedekahkan hartanya dalam bentuk beras bagi masyarakat yang kurang mampu. Biasanya dalam program ini masyarakat masih melakukannya secara manual baik dalam penimbangan maupun dalam pencatatan data-data orang yang sudah mengambil beras sehingga membutuhkan tenaga dan waktu yang lebih. Ditambah lagi masih ada saja masyarakat mampu yang ikut dalam program ini sehingga ada beberapa masyarakat kurang mampu tidak mendapatkan bagiannya dari program bantuan beras ini, dengan kata lain masih belum tepat sasaran. Maka dari itu, perlunya suatu alat yang dapat menjawab permasalahan-permasalahan tersebut, alat yang dimaksud adalah ATM beras. Pada penelitian ini peneliti menggunakan *RFID* (*Radio Frequency Identification*) sebagai alat autentikasi pengguna, *Keypad* sebagai alat untuk memilih opsi dari berat beras, *ESP32* sebagai Mikrokontroler, modul *HX711* untuk pembaca berat pada timbangan (*Load Cell*), modul *A4988* untuk menggerakkan *Motor Stepper Nema17* membuka dan menutup katup wadah penyimpanan beras. Alat ini sudah berbasis *Internet of Things* (*IoT*) yang mana data dari pengguna yang mengambil beras akan disimpan dan ditampilkan pada *Google Sheets*. Pada alat ini berat beras rata-rata yang keluar untuk setiap pengambilan masing-masing target yaitu 1.040 g untuk target 1.000 g, 2.026 g untuk target 2.000 g, dan 3.053 g untuk target 3.000 g, dan *error* rata-rata dari setiap target berturut-turut yaitu 4%, 1,3%, dan 1,9%. Alat ini sudah memiliki sistem keamanan baik keamanan dengan *RFID* maupun dengan sistem keamanan *auto disconnect* saat *WiFi* terputus sehingga proses yang sedang berjalan berhenti.

Kata Kunci: *ATM, RFID, Keypad, Motor Stepper Nema17, Load Cell*

ABSTRACT

DESIGN AND DEVELOPMENT OF INTERNET OF THINGS (IoT) BASED RICE ATM WITH REAL-TIME DATABASE USING ESP32

By:

Nahdrul Rivandi

Rice assistance is a program typically carried out by the community and mosque administrators who wish to donate their wealth in the form of rice to the less fortunate. Usually, in this program, the community still performs the process manually, both in weighing the rice and recording the data of individuals who have taken the rice, requiring more effort and time. Furthermore, there are still affluent individuals who participate in this program, causing some less privileged members of the community to miss out on their share of this rice assistance program; in other words, it is not yet reaching the intended recipients accurately. Therefore, there is a need for a device that can address these issues, and that device is referred to as the Rice ATM. In this research, the researcher employs *RFID (Radio Frequency Identification)* as a user authentication tool, *Keypad* as a means to select rice weight options, *ESP32* as the microcontroller, *HX711* module for reading the weight on the scale (*Load Cell*), *A4988* module to control the *Nema17 Stepper Motor* for opening and closing the rice storage container valve. This device is IoT-based, meaning that data from users who withdraw rice will be stored and displayed on *Google Sheets*. In this device, the average weight of dispensed rice for each respective target is 1,040 g for the 1,000 g target, 2,026 g for the 2,000 g target, and 3,053 g for the 3,000 g target, with average errors for each target being 4%, 1.3%, and 1.9% respectively. The device also features a security system, both with *RFID* security and an auto-disconnect security system when *WiFi* connection is lost, ensuring that the ongoing processes come to a halt.

Keywords: ATM, RFID, Keypad, Nema17 Stepper Motor, Load Cell

Judul Skripsi : **RANCANG BANGUN ATM BERAS
INTERNET OF THINGS (IoT) DENGAN
REAL-TIME DATABASE BERBASIS ESP32**

Nama Mahasiswa : **Nahdrul Rivandi**

Nomor Pokok Mahasiswa : **1815031084**

Jurusan : **Teknik Elektro**

Fakultas : **Teknik**

MENYETUJUI

1. Komisi Pembimbing



Syaiful Alam, S.T., M.T.
NIP 196904161998031004



Herlinawati, S.T., M.T.
NIP 197103141999032001

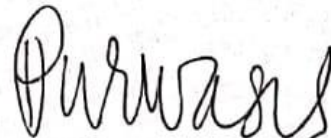
2. Mengetahui

Ketua Jurusan
Teknik Elektro



Herlinawati, S.T., M.T.
NIP. 19710314 199903 2 001

Ketua Program Studi
Teknik Elektro

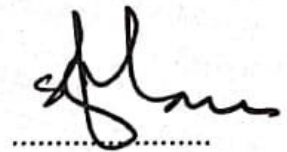


Dr. Eng. Nining Purwasih, S.T., M.T.
NIP. 19740422 200012 2 001

MENGESAHKAN

1. Tim Penguji

Ketua : Syaiful Alam, S.T., M.T.



Sekretaris : Herlinawati, S.T., M.T.



Penguji : Dr. Ir. Sri Ratna Sulistiyanti, S.T., M.T.



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. }
NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : 4 September 2023

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang sepengetahuan saya tidak terdapat atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam Daftar Pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum yang berlaku.

Bandar Lampung, 4 September 2023



Nahdrul Rivandi

NPM 1815031084

RIWAYAT HIDUP

Penulis dilahirkan di Padang pada tanggal 27 Juli 1999. Penulis merupakan anak kedua dari dua bersaudara dari pasangan Bapak Khairul Amri dan Ibu Martina.

Mengenai riwayat pendidikan, Penulis lulus Sekolah Dasar di SD Negeri 1 Tunggal Warga pada tahun 2012, lulus Sekolah Menengah Pertama di SMP Negeri 2 Banjar Agung pada tahun 2015, lulus Sekolah Menengah Atas di SMA Negeri 1 Banjar Agung pada tahun 2018, dan diterima di Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung melalui Jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) pada tahun 2017.

Selama menjadi mahasiswa, Penulis aktif di organisasi Himpunan Mahasiswa Teknik Elektro (HIMATRO) Fakultas Teknik sebagai anggota Departemen Pendidikan pada Divisi Minat dan Bakat pada periode 2019 dan sebagai Kepala Divisi Minat dan Bakat pada periode 2020. Kemudian Penulis juga melakukan kegiatan kerja praktik pada PDAM Way Rilau sebagai tim perawatan, serta Penulis juga mengikuti kegiatan Studi Independen tentang UI/UX Design and Research di Binar Academy.

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Karya ini akan ku persembahkan dengan rasa cinta dan kasih kepada:

Allah SWT. sebagai ucapan rasa syukur kepada-Nya yang telah memberi kemudahan untuk menyusun skripsi ini. Setiap goresan tinta ini adalah wujud dari keagungan dan kasi sayang yang diberikan Allah SWT. kepada hamba-Nya.

Keluarga:

Ayah Khairul Amri, Mamak Martina, Abang Nofri Khairandi sebagai wujud cinta, kasih sayang, dan bakti atas segala yang telah diberikan selama ini.

Serta Dosen Pembimbing, Penguji dan Civitas Akademik Jurusan Teknik Elektro, terimakasih telah memberikan bimbingan, arahan, sara, dan ilmu yang sangat banyak selama perkuliahan serta pengerjaan skripsi ini.

MOTTO

“Cukuplah Allah (menjadi penolong) bagi kami dan Dia sebaik-baik pelindung”

(Q.S. Ali-Imran: 173)

“Mintalah pertolongan Allah dengan sabar dan sholat”

(Q.S. Al-Baqarah: 45)

“Janganlah kamu berputus asa dari rahmat Allah”

(Q.S. Az-Zumar: 53)

“Diantara kebaikan islam seseorang adalah meninggalkan hal yang tidak bermanfaat baginya”

(H.R. Tirmidzi)

SANWACANA

Alhamdulillah puji syukur kehadiran Allah SWT. Atas segala bentuk kenikmatan, pertolongan serta keberkahan yang telah diberikan kepada Penulis sehingga Penulis dapat menyelesaikan Skripsi ini yang berjudul “ **RANCANG BANGUN ATM BERBASIS *INTERNET OF THINGS (IoT)* DENGAN *REAL-TIME DATABASE* BERBASIS *ESP32*” yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.**

Selama perjalanan mengerjakan Skripsi ini, Penulis mendapatkan dukungan moril dan bantuan pemikiran dari berbagai pihak. Oleh karena itu dalam kesempatan kali ini, Penulis ingin mengucapkan terimakasih kepada:

1. Ayah, Mamak, dan Abang yang telah memberikan kasih sayang, perhatian, dukungan, dan doa kepada Penulis sehingga Skripsi dan penyusunan laporan ini dapat terselesaikan dengan baik .
2. Bapak Herry Gusmedi, S.T., M.T. sebagai Dosen Pembimbing Akademik yang telah memberikan nasihat dan semangatnya agar dapat menyelesaikan tanggung jawab yang telah diberikan orang tua.
3. Bapak Syaiful Alam, S.T., MT. sebagai Dosen Pembimbing Utama yang telah memberikan waktu, arahan dan ilmu kepada Penulis.
4. Ibu Herilnawati S.T., M.T selaku Dosen Pembimbing Pendamping dan Ketua Jurusan Teknik Elektro yang telah memberikan koreksi dan masukannya kepada Penulis.

5. Ibu Dr. Ir. Sri Ratna Sulistiyanti. S.T., M.T. selaku Dosen Penguji atas masukan, koreksi dan saran yang telah diberikan kepada Penulis.
6. Ibu Dr. Eng. Nining Purwasih, S.T., M.T. sebagai Ketua Program Studi Teknik Elektro Universitas Lampung yang telah perhatian terhadap Penulis dan angkatan 2018.
7. Bapak Ibu Dosen Jurusan Teknik Elektro yang telah mendidik, membimbing dan memberikan ilmu pengetahuannya, serta Staff Administrasi Jurusan Teknik Elektro Universitas Lampung.
8. Kepada Mertua Mas Mul yang telah membantu dalam pembuatan rangka dari ATM Beras.
9. Angkatan 2018-ku tercinta ELTICS selaku teman seperjuangan, terima kasih sudah menjadi rumah, saudara dan teman dalam segala bentuk perhatian serta kasih sayang yang telah kalian berikan kepada Penulis.
10. Seluruh anggota grup Solidaritas Forever yang telah memberikan masukan, semangat dan bantuannya kepada Penulis.
11. Seluruh anggota grup Akatsuki yang telah memberikan semangat serta menemani hingga Penulis lulus.
12. Seluruh anggota grup Ly-are yang telah mendampingi Penulis sebagai sesama perantauan dari Unit 2.
13. Seluruh keluarga besar HIMATRO yang telah memberikan ilmu dan pengalaman kepada Penulis.
14. Kadek Jata sebagai teman satu kontrakan yang telah memberikan semangat dan nasihatnya agar Penulis bisa menyelesaikan Skripsi ini.
15. Semua pihak yang terlibat dalam menyelesaikan Skripsi ini yang tidak dapat Penulis tuliskan satu persatu.

Semoga Allah SWT. membalas semua bentuk kebaikan yang telah kalian berikan sehingga Skripsi ini dapat terselesaikan. Penulis menyadari bahwa banyak hal yang

kurang dalam penulisan skripsi ini, maka dari itu Penulis terbuka untuk menerima masukan, kritik, dan sara yang dapat membangun Penulis kedepannya. Semoga Penulisan skripsi ini dapat bermanfaat bagi kita semua.

Bandar Lampung, 4 September 2023

Penulis,

A handwritten signature in black ink, appearing to be 'Nahdrul Rivandi', written in a cursive style.

Nahdrul Rivandi

DAFTAR ISI

ABSTRAK.....	iii
SURAT PERNYATAAN.....	vii
RIWAYAT HIDUP.....	viii
SANWACANA.....	xi
DAFTAR ISI.....	xiv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xx
I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
1.6 Hipotesis.....	5
1.7 Sistematika Penulisan.....	5
II. TINJAUAN PUSTAKA.....	7
2.1 Beras.....	7
2.2 <i>Internet of Things (IoT)</i>	8

2.3 ESP32	8
2.4 Radio Frequency Identification (RFID)	11
2.5 Motor stepper	13
2.6 Modul LCD I2C 16x2.....	15
2.7 Load cell	16
2.8 Keypad.....	17
2.9 Adaptor 12V	18
2.10 IC Regulator 5 V	19
III. METODOLOGI PENELITIAN.....	28
3.1 Waktu dan Tempat Penelitian	28
3.2 Alat dan Bahan	28
3.3 Tahap Penelitian	29
3.4 Diagram alir penelitian	30
3.5 Sketsa Desain Perancangan	31
3.6 Diagram Blok Sistem	32
3.7 Perancangan Sistem.....	33
3.8 Langkah-langkah Pengujian	34
3.8.1 Mikrokontroler ESP32.....	35
3.8.2 Koneksi WiFi Oleh ESP32	38
3.8.3 RFID (Radio Frequency Identification)	39
3.8.4 Keypad	42
3.8.5 Motor Stepper Nema17.....	43
3.8.6 Load Cell.....	45
3.8.7 Google Sheets	50
IV. HASIL DAN PEMBAHASAN.....	57

4.1 Implementasi Hasil Perancangan	57
4.2 Prinsip Kerja Alat	60
4.3 Pengujian	61
4.3.1 Pengujian Sub Sistem	61
4.3.2 Pengujian Sistem Perancangan	66
4.3.3 Analisa dan Pembahasan	69
V. KESIMPULAN DAN SARAN.....	77
5.1 Kesimpulan.....	77
5.2 Saran	77
DAFTAR PUSTAKA	78

DAFTAR GAMBAR

Gambar 2. 1 Beras dan gabah	7
Gambar 2. 2 <i>Internet of Things</i>	8
Gambar 2. 3 Modul <i>ESP32 DevKitC</i>	9
Gambar 2. 4 <i>Pin out ESP32</i>	10
Gambar 2. 5 <i>Arduino IDE</i>	11
Gambar 2. 6 <i>RFID Tag</i>	12
Gambar 2. 7 <i>RFID Reader</i>	12
Gambar 2. 8 Motor <i>stepper</i> dan <i>A4988</i>	14
Gambar 2. 9 Modul <i>LCD I2C 16x2</i>	16
Gambar 2. 10 Modul <i>I2C</i>	16
Gambar 2. 11 <i>Load cell</i>	17
Gambar 2. 12 <i>Keypad</i>	18
Gambar 2. 13 <i>Adaptor 12 V</i>	18
Gambar 2. 14 <i>IC Regulator 5 V</i>	19
Gambar 3. 1 Diagram alir penelitian.....	30
Gambar 3. 2 Desain alat	31
Gambar 3. 3 Diagram blok sistem	32
Gambar 3. 4 Diagram alir perancangan sistem	33
Gambar 3. 5 Diagram Pengkabelan Seluruh Komponen	35
Gambar 3. 6 <i>Submenu board</i>	36

Gambar 3. 7 Tampilan jendela <i>editor</i>	36
Gambar 3. 8 <i>Verify code</i>	37
Gambar 3. 9 <i>Upload code</i>	37
Gambar 3. 10 Program <i>ESP32 connect to WiFi</i>	38
Gambar 3. 11 Hasil pengujian <i>ESP32 connect to WiFi</i>	38
Gambar 3. 12 Diagram pengkabelan <i>RFID</i>	39
Gambar 3. 13 Program <i>RFID</i>	40
Gambar 3. 14 Pengukuran jarak kartu <i>RFID</i>	41
Gambar 3. 15 Diagram pengkabelan <i>Keypad</i>	42
Gambar 3. 16 Program <i>Keypad</i>	43
Gambar 3. 17 Diagram pengkabelan Motor <i>Stepper Nema17</i>	44
Gambar 3. 18 Program Motor <i>Stepper Nema17</i>	45
Gambar 3. 19 Diagram pengkabelan <i>Load Cell</i>	46
Gambar 3. 20 Objek dengan berat 209 g.....	46
Gambar 3. 21 Program kalibrasi <i>Load Cell</i>	47
Gambar 3. 22 Menimbang objek.....	47
Gambar 3. 23 Hasil <i>serial monitor</i> kalibrasi <i>Load Cell</i>	48
Gambar 3. 24 Program menimbang <i>Load Cell</i>	49
Gambar 3. 25 Proses penimbangan beras di timbangan digital	49
Gambar 3. 26 Menimbang objek diatas <i>Load Cell</i> (timbangan).....	50
Gambar 3. 27 Tab <i>Create</i>	51
Gambar 3. 28 Menu <i>If This</i>	51
Gambar 3. 29 <i>Service Webhooks</i>	52
Gambar 3. 30 Memilih <i>Trigger</i>	52
Gambar 3. 31 Nama <i>Event</i>	53

Gambar 3. 32 Menu <i>Then That</i>	53
Gambar 3. 33 <i>Google Sheets</i>	54
Gambar 3. 34 <i>Add row to spreadhseet</i>	54
Gambar 3. 35 Melengkapi informasi	55
Gambar 3. 36 Opsi <i>Documentation</i>	55
Gambar 3. 37 Konfigurasi alamat <i>web</i>	56
Gambar 3. 38 Hasil tes dengan nilai bebas pada <i>Google Sheets</i>	56
Gambar 3. 39 Program <i>ESP32 to Google Sheets</i>	57
Gambar 4. 1 ATM Beras	57
Gambar 4. 2 Bagian dalam ATM beras	58
Gambar 4. 3 Komponen-komponen di PCB bolong	59
Gambar 4. 4 Motor <i>stepper Nema17</i>	59
Gambar 4. 5 <i>Load Cell</i> (timbangan) dan <i>Adaptor 12 V</i>	60
Gambar 4. 6 Respon <i>ESP32</i>	61
Gambar 4. 7 Hasil pengujian <i>ESP32 connect to WiFi</i>	62
Gambar 4. 8 Hasil pembacaan <i>RFID</i> di <i>serial monitor</i>	62
Gambar 4. 9 Hasil pengujian <i>Keypad</i>	64
Gambar 4. 10 Pengujian Motor <i>Stepper Nema17</i>	65
Gambar 4. 11 Hasil penimbangan	66
Gambar 4. 12 Tampilan data pada <i>Google Sheets</i>	71
Gambar 4. 13 Pemandangan data <i>Google Sheets</i>	72
Gambar 4. 14 Sistem keamanan pertama	73
Gambar 4. 15 Sistem keamanan kedua	74
Gambar 4. 16 Fitur <i>auto disconnect WiFi</i>	75

DAFTAR TABEL

Tabel 1. 1 Penelitian terdahulu.....	2
Tabel 2. 1 Spesifikasi <i>driver</i> motor <i>A4988</i>	14
Tabel 2. 2 Keterangan pin-pin yang ada pada <i>LCD</i> 16 x 2.	15
Tabel 3. 1 Alat dan bahan	28
Tabel 4. 1 Hasil pengukuran <i>RFID</i>	63
Tabel 4. 2 Data hasil pengujian sistem perancangan berat beras 1.000 g.....	67
Tabel 4. 3 Data hasil pengujian sistem perancangan berat beras 2.000 g.....	68
Tabel 4. 4 Data hasil pengujian sistem perancangan berat beras 3.000 g.....	69

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi pada saat ini sudah banyak mempengaruhi kehidupan manusia yang mencakup berbagai bidang baik dari segi informasi, komunikasi, dan juga transportasi. Perkembangan teknologi ini membuat berbagai kegiatan atau pekerjaan manusia menjadi lebih efektif dan efisien. Begitupun pada kegiatan sosial masyarakat contohnya pada program bantuan beras.

Bantuan beras merupakan sebuah program yang biasanya dilakukan oleh masyarakat dan pengurus masjid yang ingin menyedekahkan hartanya dalam bentuk beras bagi masyarakat yang kurang mampu. Biasanya dalam program ini masyarakat masih melakukannya secara manual baik dalam penimbangan maupun dalam pencatatan data-data orang yang sudah mengambil beras sehingga membutuhkan tenaga dan waktu yang lebih. Ditambah lagi masih ada saja masyarakat mampu yang ikut dalam program ini sehingga ada beberapa masyarakat kurang mampu tidak mendapatkan bagiannya dari program bantuan beras ini, dengan kata lain masih belum tepat sasaran. Maka dari itu, perlunya suatu alat yang dapat menjawab permasalahan-permasalahan tersebut, alat yang dimaksud adalah ATM beras.

ATM beras adalah suatu alat yang dapat menampung dan mengeluarkan beras secara otomatis sesuai dengan keinginan tanpa harus menimbang secara manual. Pada alat ini terdapat komponen-komponen tambahan yang digunakan untuk menawarkan fitur-fitur yang dibutuhkan pada program bantuan beras ini seperti *Radio Frequency Identification (RFID)*, *Keypad*, *Load cell* dan juga sudah berbasis *Internet of Things (IOT)*. *RFID* pada alat ini digunakan sebagai proses autentikasi (proses validasi identitas) agar bantuan beras ini sesuai dengan data masyarakat

yang kurang mampu, *Keypad* digunakan untuk memilih berat beras yang diinginkan, *Load cell* digunakan untuk menimbang beras yang sudah keluar dari tabung ATM beras, dan *IoT* digunakan untuk mengirim data dari pengguna yang sudah mengambil beras ke sebuah *software* atau perangkat lunak secara *real time*. Oleh karena itu, diharapkan penelitian ini dapat diimplementasikan sehingga dapat mempermudah program bantuan beras tersebut.

Adapun penelitian terdahulu dan penelitian pendukung pada penelitian ini dapat dilihat pada Tabel 1.1 berikut.

Tabel 1. 1 Penelitian terdahulu

Nama Peneliti	Judul	Hasil
M. Mobtazim Billah, Aji Brahma Nugroho, M. Aan Auliq	Mesin Otomatis Pengambilan Raskin Menggunakan <i>RFID</i> Berbasis Mikrokontroler <i>Arduino</i>	Peneliti berhasil membuat timbangan beras menggunakan <i>Load cell</i> pada ATM beras dengan keluaran 100 dan 200 gram dengan kesalahan 3.4%.
Kholaqul Arief JL	Rancang Bangun Prototype ATM Beras Berbasis Mikrokontroler <i>ATMega328p</i>	Peneliti berhasil membuat ATM beras dengan <i>ATMega328p</i> dengan rata- rata berat beras yang keluar 2.504 gram untuk setiap pengambilan.
Syafrie Syahriel, Adi Prijuna Lubis, Rizky Fauziah	Perancangan ATM Raskin Berbasis <i>RFID</i> dan <i>Internet of Things (IoT)</i> Untuk Masyarakat Tidak Mampu	Peneliti berhasil membuat ATM beras berbasis <i>IoT</i> dengan menggunakan Mikrokontroler <i>NodeMCU</i> <i>esp8266</i> sehingga daftar penerima bantuan dapat dilihat di <i>Web</i> .

Pada penelitian dengan judul “Mesin Otomatis Pengambilan Raskin Menggunakan *RFID* Berbasis Mikrokontroler *Arduino*” yang dilakukan oleh M. Mobtazim Billah dkk. (2018) yang menggunakan *Arduino* Mega 2560 sebagai mikrokontroler dan *RFID* yang berisikan data-data penerima raskin, kelemahan dari penelitian ini adalah beras yang ditampung hanya 1.000 g karena tempat penyimpanan yang kecil dan pengambilan beras hanya 100 sampai 200 gram, belum berbasis *IoT*, dan opsi pengambilan beras terbatas karena setiap *RFID* sudah ditentukan berat beras yang akan diambil [3].

Kemudian penelitian yang dilakukan oleh Khollaqul Arief JL (2022) dengan judul “Rancang Bangun *Prototype* ATM Berbasis Mikrokontroler *ATMega328p*” yang menggunakan *ATMega328p* sebagai mikrokontroler dan *RFID* sebagai masukan serta berisi data-data dari penerima bantuan beras, kelemahan pada penelitian ini adalah opsi pengambilan beras terbatas karena setiap *RFID* sudah ditentukan berat beras yang akan diambil, tidak ada data berat beras sesungguhnya yang keluar, dan belum berbasis *IoT* sehingga tidak terdapat *database* dari penerima bantuan yang sudah mengambil beras [1].

Kemudian penelitian yang dilakukan oleh, Syafire Syahriel dkk. (2021) dengan judul ‘Perancangan ATM Raskin Berbasis *RFID* dan *Internet of Things (IoT)* Untuk Masyarakat Tidak Mampu’ yang menggunakan *NodeMCU ESP8266* sebagai mikrokontroler dan *RFID* sebagai masukan, kelemahan pada penelitian ini adalah opsi pengambilan beras terbatas karena setiap *RFID* sudah ditentukan berat beras yang akan diambil, dan tidak ada data berat beras sesungguhnya yang keluar dari ATM beras karena tidak menggunakan loadcell [4].

Pada penelitian ini akan dibuat sebuah rancang bangun alat yang akan menggunakan komponen utama yang menjadi pembeda dari sebelumnya yaitu menggunakan mikrokontroler *ESP32* dan komponen lainnya seperti *RFID*, *Keypad*, dan *Load cell* yang terdapat dalam alat yang akan dibuat pada penelitian ini serta sudah berbasis *IoT*. Pada penelitian sebelumnya komponen lainnya seperti *RFID*,

Keypad, *Load cell* tidak berada dalam satu alat atau masih belum selengkap dari penelitian ini.

1.2 Perumusan Masalah

Adapun perumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana cara mendapatkan dan mengirimkan data ke *database* dari orang-orang yang sudah mengambil beras dari ATM beras berbasis mikrokontroler *ESP32* secara *real-time*.
2. Bagaimana cara ATM beras menimbang dan mengeluarkan beras yang diinginkan berdasar masukan *Keypad* secara presisi.
3. Bagaimana menambahkan sistem keamanan ATM Beras.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. Autentikasi *user*/pengguna hanya melalui kartu *RFID*.
2. *IoT* pada alat ini hanya digunakan untuk mengirimkan data pengguna ke *database* dari orang-orang yang sudah mengambil beras.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang ATM beras berbasis *IoT* dengan menggunakan mikrokontroler *ESP32* sehingga mendapatkan data dari orang-orang yang sudah mengambil beras secara *real-time*.
2. Mendapatkan berat beras dengan *error rate* sekecil mungkin dari berat yang dipilih.
3. Merancang sistem keamanan menggunakan data *RFID* dan *auto disconnect* ketika tidak tersambung *WiFi*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah untuk mempermudah takmir atau pengurus masjid baik dari segi penimbangan beras dan juga dari segi pendataan masyarakat yang sudah mengambil beras sehingga program bantuan beras ini dapat berjalan dengan efisien.

1.6 Hipotesis

Adapun hipotesis pada penelitian ini adalah alat ini tidak dapat bekerja apabila tidak terhubung dengan *WiFi* yang sudah ditentukan, dapat mengeluarkan beras dengan *error rate* sekecil mungkin berdasarkan masukan dari *Keypad*, dan data dari *user* yang sudah mengambil beras dapat dikirimkan ke *database* secara *real-time*.

1.7 Sistematika Penulisan

Adapun sistematika Penulisan pada penelitian ini adalah sebagai berikut:

I. PENDAHULUAN

Berisikan tentang latar belakang, tujuan penelitian, perumusan masalah, batasan masalah, manfaat penelitian, hipotesis, dan sistematika Penulisan.

II. TINJAUAN PUSTAKA

Berisikan mengenai sejumlah teori pendukung dan referensi materi yang didapat dari berbagai sumber seperti buku, jurnal, *datasheet* dan penelitian ilmiah untuk menyelesaikan laporan tugas akhir ini.

III. METODOLOGI PENELITIAN

Berisikan tentang waktu dan tempat, alat dan bahan, metode penelitian dan pelaksanaan serta pengamatan dalam pengerjaan tugas akhir.

IV. HASIL DAN PEMBAHASAN

Berisikan hasil dan analisis dari perancangan.

V. PENUTUP

Berisikan kesimpulan dari penelitian yang sudah dilaksanakan dan saran agar hasil yang didapatkan lebih baik berdasarkan hasil data yang diperoleh.

II. TINJAUAN PUSTAKA

2.1 Beras

Definisi secara umum beras sesuai Peraturan Menteri Perdagangan RI Nomor 19/M-DAG/PER/3/2014 menjelaskan beras adalah biji-bijian baik berkulit, tidak berkulit, diolah atau tidak diolah yang berasal dari *Oriza Sativa*. Pada definisi ini beras mencakup gabah, beras giling, dan beras pecah kulit.

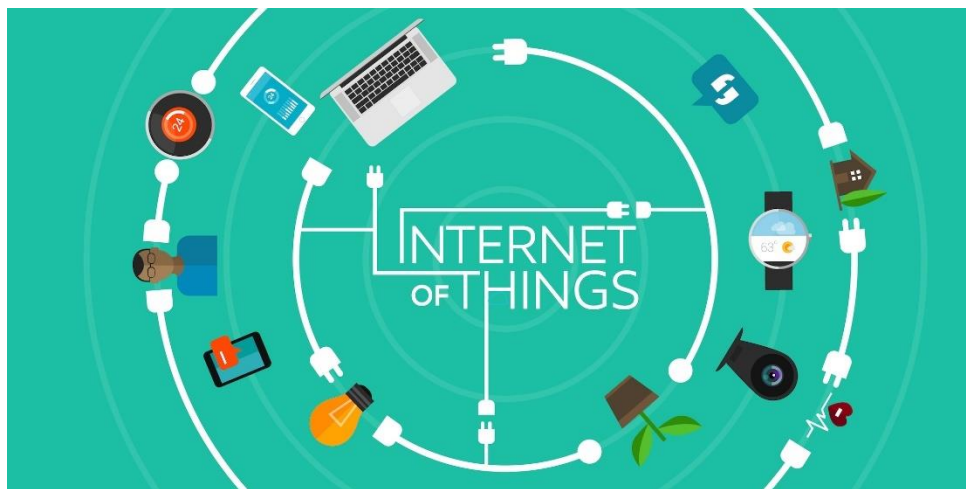
Definisi umum, beras merupakan bagian bulir padi (gabah) yang telah dipisah dari sekam dan dedak atau bekatul (Kementan, 2015). Definisi ini menunjukkan beras sebagai produk akhir dari gabah. Gabah tidak didefinisikan sebagai beras. Bentuk beras dan gabah dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Beras dan gabah

2.2 *Internet of Things (IoT)*

Internet of Things atau biasa disingkat *IoT* adalah suatu konsep saat suatu benda mempunyai teknologi seperti sensor dan *software* yang memiliki tujuan dalam berkomunikasi, menghubungkan, bertukar data menggunakan perangkat lain saat terhubung ke internet. *IoT* merupakan salah satu teknologi yang memiliki hubungan erat terhadap istilah *M2M (Machine to Machine)*. Alat yang digunakan pada *M2M* mampu berkomunikasi sehingga disebut *smart devices* atau perangkat cerdas [9].



Gambar 2. 2 *Internet of Things*

2.3 *ESP32*

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik yang terdiri dari *Central Processing Unit (CPU)*, memori, *input/output*, bahkan sudah dilengkapi dengan *Analog-to-Digital-Converter (ADC)* yang sudah terintegrasi didalamnya. Kelebihan utama dari mikrokontroler adalah tersedianya *Random Access Memory (RAM)* dan peralatan *I/O* pendukung sehingga ukuran *board* mikrokontroler menjadi sangat ringkas. Mikrokontroler pertama kali dikenalkan oleh *Texas Instrument* pada tahun 1974 dengan seri *TMS-1000*. Mikrokontroler pertama ini merupakan mikrokontroler 4 bit. Mikrokontroler ini memiliki sebuah *chip* yang telah dilengkapi dengan *RAM* dan *Read Only Memory (ROM)*. Selanjutnya, pada tahun 1976 intel mengeluarkan mikrokontroler 8 bit dengan nama 8748 yang merupakan mikrokontroler dari keluarga *MMCS-48*.

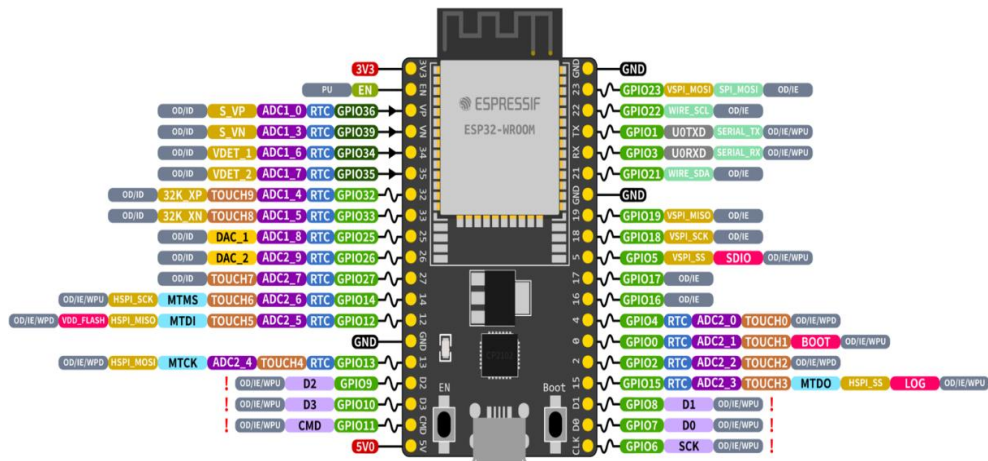
Untuk saat ini telah banyak ditemui tipe-tipe mikrokontroler mulai dari 8 bit sampai dengan 64 bit. Dalam perkembangannya, modul atau minimum sistem dari mikrokontroler dibuat dalam bentuk *chip* yang lebih memudahkan pengguna untuk menggunakannya.

Pada tugas akhir ini Penulis menggunakan mikrokontroler *ESP32 DevKitC*. *ESP32* dibuat oleh *Espressif Systems*, *ESP32* adalah sistem dengan biaya yang rendah, berdaya rendah pada seri *chip* (*SoC*) dengan *Wi-Fi* & kemampuan *Bluetooth* dua mode. Keluarga *ESP32* termasuk chip *ESP32-D0WDQ6* (dan *ESP32-D0WD*), *ESP32-D2WD*, *ESP32-S0WD*, dan sistem dalam paket (*SiP*) *ESP32-PICO-D4*. Pada intinya, ada mikroprosesor *Tensilica Xtensa LX6 dual-core* atau *single-core* dengan *clock rate* hingga 240 MHz. *ESP32* sudah terintegrasi dengan *built-in antenna switches*, *RF balun*, *power amplifier*, *low-noise receive amplifier*, *filters*, and *power management modules*. Didesain untuk perangkat seluler, perangkat elektronik yang dapat dipakai, dan aplikasi *IoT*, *ESP32* juga bekerja dengan konsumsi daya sangat rendah melalui fitur hemat daya termasuk *fine resolution clock gating*, *multiple power modes*, and *dynamic power scaling*. Berikut ini merupakan bentuk fisik dari *ESP32* yang ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Modul *ESP32 DevKitC*

Module *ESP32* merupakan penerus dari module *ESP8266* yang cukup populer untuk Aplikasi *IoT*. Pada *ESP32* terdapat inti *CPU* serta *Wi-Fi* yang lebih cepat, *GPIO* yang lebih, dan mendukung *Bluetooth Low Energy*. Berikut ini merupakan skema pin-pin dari *ESP32 DevKitC* pada Gambar 2.4.



Gambar 2. 4 Pin out *ESP32*

Berikut ini merupakan spesifikasi dari mikrokontroler *ESP32* [10]:

1. Memiliki CPU *Tensilica Xtensa LX6* 32 bit *Dual-Core* di 160/240 MHz.
2. Memiliki SRAM sebesar 520 KB.
3. Memiliki FLASH dengan 2 MB (*max.* 64 MB).
4. Memiliki tegangan kerja 2.2 V sampai 3.6 V.
5. Arus kerja rata-rata 80 mA.
6. Memiliki *Bluetooth 4.2 BR/EDR + BLE*.
7. Memiliki 3 antarmuka *Universal Asynchronous Receiver Transmitter* (UART).
8. Memiliki *General Pin Input Output* (GPIO) sebanyak 32 pin.
9. Memiliki 4 antarmuka SPI.
10. Memiliki 2 antarmuka I2C.
11. Memiliki 8 kanal output PWM.

12. Memiliki 18 (12 bit) kanal ADC .
13. Memiliki 2 (8 bit) kanal DAC.

Mikrokontroler *ESP32* dapat dioperasikan dengan menggunakan program pada *software Arduino IDE*. *Arduino IDE* berfungsi untuk membuat, membuka, dan mengedit program yang akan kita masukkan ke dalam *board ESP32*. Aplikasi *Arduino IDE* dirancang agar memudahkan penggunaanya dalam membuat berbagai aplikasi. *Arduino IDE* memiliki struktur bahasa pemrograman yang sederhana dan fungsi yang lengkap, sehingga mudah untuk dipelajari oleh pemula sekalipun. Dapat dilihat bentuk dari *software Arduino IDE* pada Gambar 2.5.



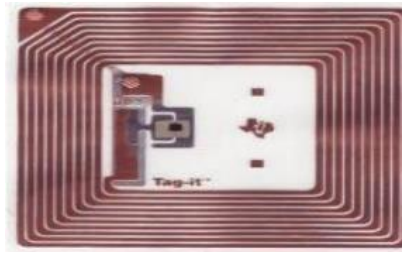
Gambar 2. 5 *Arduino IDE*

2.4 Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) merupakan sebuah teknologi yang menggunakan metode *auto-ID* atau *Automatic Identification*. *Auto-ID* adalah metoda pengambilan data dengan identifikasi objek secara otomatis tanpa ada keterlibatan manusia. *Auto-ID* bekerja secara otomatis sehingga dapat meningkatkan efisiensi dalam mengurangi kesalahan dalam memasukkan data [11].

a. RFID Tag

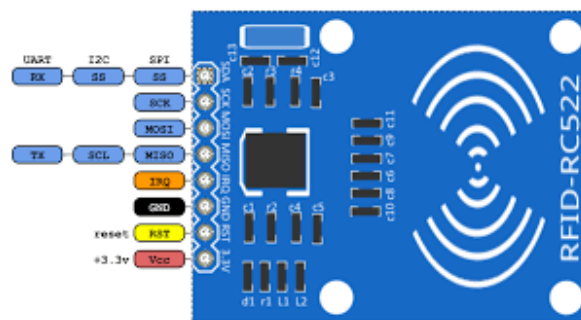
RFID tag adalah sebuah alat yang melekat pada obyek yang akan diidentifikasi oleh *RFID reader*, rangkaian elektronik dari *RFID tag* umumnya memiliki memori yang memungkinkan *RFID tag* mempunyai kemampuan untuk menyimpan data.



Gambar 2. 6 *RFID Tag*

b. *RFID Reader*

RFID reader merupakan penghubung antara *software* aplikasi dengan antena yang akan meradiasikan gelombang radio ke *RFID tag*. Berikut ini merupakan sekema pin-pin dari *RFID reader* yang ditunjukkan pada Gambar 2.7.



Gambar 2. 7 *RFID Reader*

Cara kerja dari *RFID* ketika kartu *tag* ditempelkan atau dapat juga dilekatkan pada objek yang mana di dalamnya terdapat *RFID reader*. *RFID reader* akan memancarkan dan menerima sinyal berupa frekuensi radio, untuk selanjutnya disesuaikan oleh *RFID tag*. Saat *RFID tag* dan *RFID reader* memiliki gelombang frekuensi yang sama maka akan terjadi pembacaan data dan informasi yang ada pada *RFID tag* yang dilakukan oleh *RFID reader*. Hal tersebut menyebabkan kedua komponen (*tag* dan *reader*) dapat berkomunikasi secara *wireless* atau komunikasi tanpa menggunakan kabel. Selain dua komponen ini, *RFID* juga membutuhkan sistem kontroler yang berfungsi sebagai jembatan yang menghubungkan *RFID* dengan seluruh sistem yang ada, agar dapat melakukan penyimpanan dan pemrosesan data yang telah terbaca menuju *database* untuk diteruskan dengan

tugas lainnya yang terdapat pada sistem tersebut. Ada dua jenis *RFID reader* yaitu *reader* pasif dan aktif. *Reader* pasif merupakan sistem pembaca pasif yang hanya dapat menerima sinyal radio dari *tag* aktif (yang dioperasikan dengan baterai). Jangkauan penerima alat ini dapat mencapai sampai dengan jarak 600 meter. Hal ini memungkinkan untuk dijadikan sebagai sistem perlindungan dan pengawasan aset. *Reader* aktif memiliki sistem pembaca aktif yang dapat memancarkan sinyal interogator ke *tag* dan menerima balasan autentikasi dari *tag*. Sinyal interogator ini juga menginduksi *tag* dan akhirnya menjadi sinyal DC sehingga dapat menjadi sumber daya *tag* pasif.

2.5 Motor stepper

Motor *stepper* adalah salah satu jenis motor yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Gerakan motor *stepper* berdasarkan urutan pulsa yang masuk ke motor, oleh karena itu untuk menggerakkan motor *stepper* memerlukan pengendali motor yang membangkitkan pulsa-pulsa periodik [1]. Motor *stepper* memiliki keunggulan sebagai berikut.

1. Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
2. Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
3. Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
4. Memiliki respon yang sangat baik terhadap mulai, berhenti dan berbalik (perputaran).
5. Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.

Prinsip kerja motor *stepper* mirip dengan motor DC, dimana untuk menghasilkan medan magnet pada motor perlu adanya tegangan masuk. Motor DC memiliki magnet tetap pada stator sedangkan motor *stepper* memiliki motor tetap pada rotor. Motor *stepper* bekerja dengan mengkonversi bit-bit masukan menjadi gerakan rotor. Bit-bit tersebut berasal dari terminal-terminal input yang ada pada motor *stepper* yang menjadi kutub-kutub magnet dalam motor.

Pada penelitian ini *driver* yang digunakan adalah *driver* motor *stepper* A4988. *Driver* motor A4988 adalah salah satu *driver* motor *stepper* yang populer dan banyak digunakan. *Driver* ini mendukung kontrol arus dan langkah pada motor *stepper*. A4988 mendukung hingga 1/16 *mikrostepping*. *Mikrostepping* memungkinkan pengendalian yang lebih halus dan presisi pada motor *stepper*, menghasilkan gerakan yang lebih halus dan kurang berisik. Berikut ini spesifikasi dari *driver* motor A4988 seperti pada Tabel 2.1

Tabel 2. 1 Spesifikasi *driver* motor A4988.

Karakteristik	Simbol	Nilai	Unit
Supply tegangan beban	VBB	35	V
Arus keluaran	IOUT	± 2	A
Masukan tegangan logika	VIN	-0.3 to 5.5	V
Supply tegangan logika	VDD	-0.3 to 5.5	V
VBBx to OUTx	-	35	V
Tegangan sense	VSENSE	0.5	V
Tegangan referensi	VREF	5.5	V
Suhu operasi	TA	-20 to 85	°C
Suhu persimpangan maksimum	TJ(max)	150	°C
Suhu penyimpanan	Tstg	-55 to 150	°C

Bentuk dari motor *stepper* dan *driver* A4988 dapat dilihat pada Gambar 2.8.



Gambar 2. 8 Motor *stepper* dan A4988

2.6 Modul LCD I2C 16x2

I2C adalah modul *LCD* yang dikendalikan secara serial sinkron dengan protokol *Inter Integrated Circuit (I2C/IIC)* atau *Two Wire Interface (TWI)*. Normalnya, modul *LCD* dikendalikan secara parallel baik untuk jalur data maupun kontrolnya. Namun, jalur parallel akan memakan banyak pin di sisi kontroler.

Tabel 2. 2 Keterangan pin-pin yang ada pada *LCD* 16 x 2.

No. Pin	Nama Pin	Keterangan
1	VSS	<i>Ground 0 V</i>
2	VDD	<i>Logic Power Supply</i>
3	VO	Data
4	RS	<i>Read/Write</i>
5	R/W	<i>Enable Signal</i>
6	E	Data Bit 0
7	DB0	Data Bit 0
8	DB1	Data Bit 1
9	DB2	Data Bit 2
10	DB3	Data Bit 3
11	DB4	Data Bit 4
12	DB5	Data Bit 5
13	DB6	Data Bit 6
14	DB7	Data Bit 7
15	LED_A	<i>Back Light Anoda (+)</i>
16	LED_K	<i>Back Light Cathoda (-)</i>

Fitur dari modul *LCD* 16x2 adalah sebagai berikut.

1. Tegangan operasi adalah 4.7 V hingga 5.3 V.
2. Konsumsi saat ini adalah 1 mA tanpa lampu latar.
3. Modul tampilan *LCD* alfanumerik, artinya dapat menampilkan huruf dan angka.
4. Terdiri dari dua baris dan setiap baris dapat mencetak 16 karakter.
5. Setiap karakter dibangun dengan kotak 5×8 piksel.

6. Dapat bekerja pada mode 8-bit dan 4-bit.
7. Dapat menampilkan karakter yang dibuat khusus.
8. Tersedia dalam lampu latar hijau dan biru.

Berikut ini adalah gambar dari modul *LCD I2C 16x2*..



Gambar 2. 9 Modul *LCD I2C 16x2*

Agar mempermudah penggunaan dari *LCD* dan mengurangi pin pada mikrokontroler maka digunakan modul *Inter Integrated Circuit (I2C)* adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem *I2C* terdiri dari saluran *Serial Clock (SCL)* dan *Serial Data (SDA)* yang membawa informasi data antara *I2C* dengan pengontrolnya seperti pada Gambar 2.10.



Gambar 2. 10 Modul *I2C*

2.7 *Load cell*

Sensor *Load cell* adalah suatu alat *transducer* yang menghasilkan output yang proporsional dengan beban atau gaya yang diberikan. Sensor *Load cell* dapat

memberikan pengukuran akurat dari gaya dan beban. Sensor *Load cell* mengkonversikan regangan pada logam ke tahanan variabel. Dalam penggunaan, sensor *Load cell* mengkonversi berat menjadi sinyal listrik. Sensor *Load cell* umumnya berisi 4 buah *strain gauge* yang tersusun sebagai rangkaian jembatan *Wheatstone*. Gaya tekan yang dikenakan pada sensor *Load cell* akan membuat keseimbangan 4 buah *strain gauge* tersebut terganggu. Dengan adanya tegangan eksitasi pada sensor *Load cell*, maka ke tidak seimbangan jembatan *Wheatstone* yang disebabkan oleh gaya tekan pada sensor *Load cell* akan diubah menjadi sinyal tegangan [13].



Gambar 2. 11 *Load cell*

2.8 Keypad

Keypad adalah saklar-saklar push *button* yang disusun secara *matriks* yang berfungsi untuk menginput data. *Keypad* berfungsi sebagai *interface* antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah *Human Machine Interface* (HMI). Pada dasarnya *Keypad* adalah sejumlah tombol yang disusun sedemikian rupa sehingga membentuk susunan tombol angka dan beberapa menu lainnya [14]. Pada Gambar 2.12 merupakan bentuk dari *Keypad* yang digunakan.



Gambar 2. 12 Keypad

2.9 Adaptor 12V

Adaptor adalah sebuah rangkaian *konverter* untuk mengubah tegangan AC menjadi DC, tegangan yang dirubah dari AC yang tinggi menjadi tegangan DC yang rendah. Rangkaian *adaptor* ada yang dipasang dalam peralatan elektronika langsung dan ada juga yang terpisah, *adaptor* selalu menyesuaikan kebutuhan peralatan elektronika yang dibutuhkan pada umumnya [15]. *Adaptor* 12 V berfungsi untuk merubah tegangan AC menjadi tegangan DC dengan besar tegangan yaitu 12 V yang nantinya pada alat ini akan digunakan sebagai penyuplai tegangan mikrokontroler dan motor *stepper Nema17*. Gambar 2.13 berikut merupakan bentuk dari *adaptor* yang digunakan.



Gambar 2. 13 Adaptor 12 V

2.10 IC Regulator 5 V

IC *Regulator* atau yang sering disebut sebagai *regulator* tegangan (*voltage regulator*) merupakan suatu komponen elektronik yang melakukan suatu fungsi yang terpenting dan berguna dalam perangkat elektronik baik digital maupun analog. Hal yang dilakukan oleh IC *regulator* ini adalah menstabilkan tegangan yang melewati IC tersebut. Setiap IC *regulator* mempunyai rating tegangannya sendiri-sendiri [16]. IC *Regulator 5 V* merupakan pengatur tegangan untuk mendapatkan tegangan kerja *ESP32* yaitu sebesar 5 V. Pada Gambar 2.14 merupakan bentuk fisik dari IC *Regulator 5 V*.



Gambar 2. 14 IC *Regulator 5 V*

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dimulai pada bulan Januari 2023 sampai dengan Juli 2023. Penelitian ini dilakukan di Perumahan Kota Sepang Jaya Indah, Labuhan Ratu, Kota Bandar Lampung, Lampung dan juga pada Laboratorium Terpadu Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

3.2 Alat dan Bahan

Adapun alat dan bahan pada penelitian ini adalah sebagai berikut:

Tabel 3. 1 Alat dan bahan

No	Alat dan bahan	Justifikasi Penggunaan
1	Laptop <i>Lenovo Ideapad Slim</i> 3 AMD Ryzen 4200U	Berperan dalam media pemrograman <i>software Arduino IDE</i> dan modul <i>RFID</i> , mendesain sketsa dan menyimpan data.
2	<i>Software Arduino IDE</i>	Sebagai media dalam memprogram dan mengupload instruksi-intruksi pada mikrokontroler .
3	<i>ESP32</i>	Sebagai mikrokontroler.
4	Modul <i>RFID RC522 (tag dan reader)</i>	Sebagai media Identifikasi data pengguna.
5	Motor <i>stepper Nema17</i> dan <i>Driver Motor stepper A4988</i>	Sebagai penggerak motor untuk membuka dan menutup katup pada tabung.
6	Modul <i>LCD 12C 16x2</i>	Sebagai <i>feedback</i> visual pada alat.
7	Kabel penghubung	Sebagai penghubung antar komponen.
8	PCB	Sebagai media penghubung seluruh rangkaian komponen.

Tabel 3.1 (lanjutan)

No	Alat dan bahan	Justifikasi Penggunaan
9	<i>Load cell</i>	Sebagai alat pengukur berat beras yang keluar dari tabung.
10	<i>Keypad</i>	Sebagai media untuk memilih output berat dari beras yang ingin dikeluarkan dari tabung.
11	<i>Adaptor 12V</i>	Sebagai pemberi tegangan kerja kepada motor <i>stepper</i> .
12	<i>IC Regulator 5 V</i>	Sebagai penurun tegangan menjadi 5 V untuk tegangan kerja <i>ESP32</i> .

3.3 Tahap Penelitian

Dalam penyelesaian tugas akhir ini terdapat beberapa masalah yang dihadapi yaitu bagaimana cara membuat dan perancangan ATM beras berbasis *IoT* dengan mikrokontroler *ESP32* serta bagaimana cara mendapatkan *database* dari data-data para pengguna ATM tersebut.

Sehingga dalam menyelesaikan masalah-masalah tersebut perlu melewati beberapa langkah penyelesaian yaitu sebagai berikut:

1. Studi Literatur

Studi literatur dilakukan oleh Penulis dengan mempelajari dan mengumpulkan literatur agar dapat memahami karakteristik dari alat dan bahan yang akan digunakan pada penelitian ini.

2. Studi Bimbingan

Studi bimbingan dilakukan oleh Penulis dengan berdiskusi kepada Dosen Pembimbing mengenai masalah dalam pembuatan prototype ATM beras berbasis *IoT* dengan mikrokontroler *ESP32*.

3. Pengambilan dan Pengolahan Data

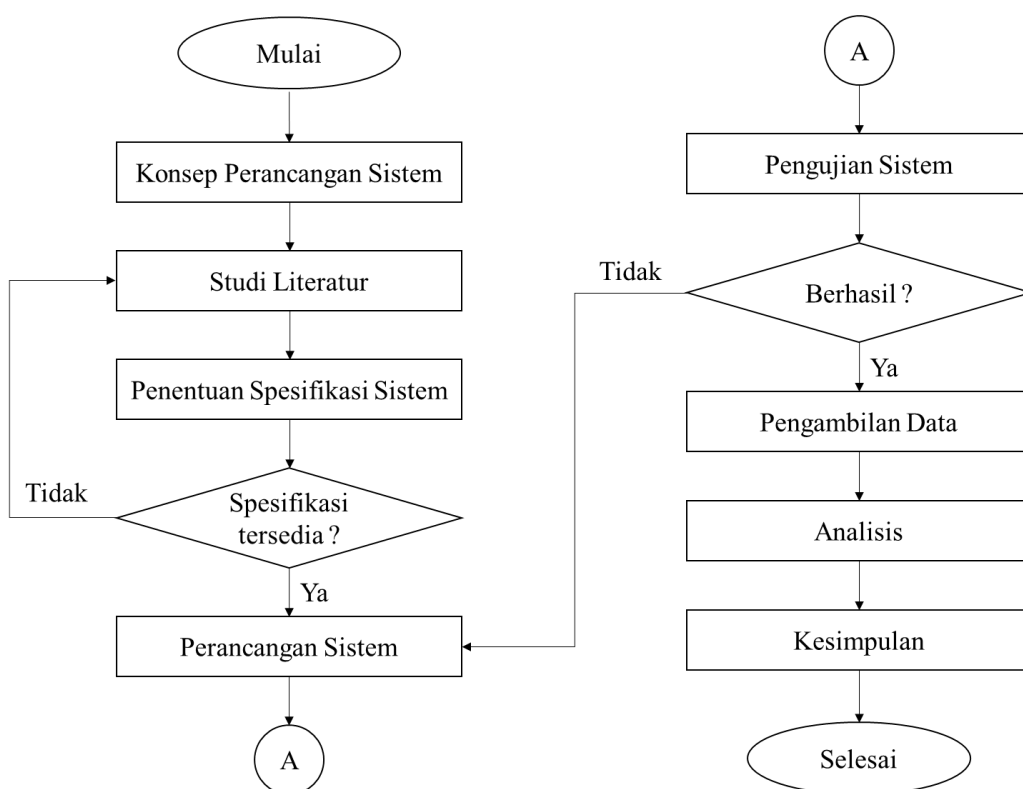
Pengambilan dan pengolahan data dilaksanakan dengan melakukan pengujian pada alat yang sudah dibuat. Data yang telah didapatkan akan dicocokkan dengan tujuan sehingga mendapatkan data yang terukur dan valid.

4. Pembuatan Laporan

Pembuatan laporan ini dilakukan untuk menyajikan hasil dari penelitian yang telah dilakukan dalam bentuk laporan akhir.

3.4 Diagram alir penelitian

Adapun diagram alir dari penelitian ini adalah sebagai berikut.



Gambar 3. 1 Diagram alir penelitian

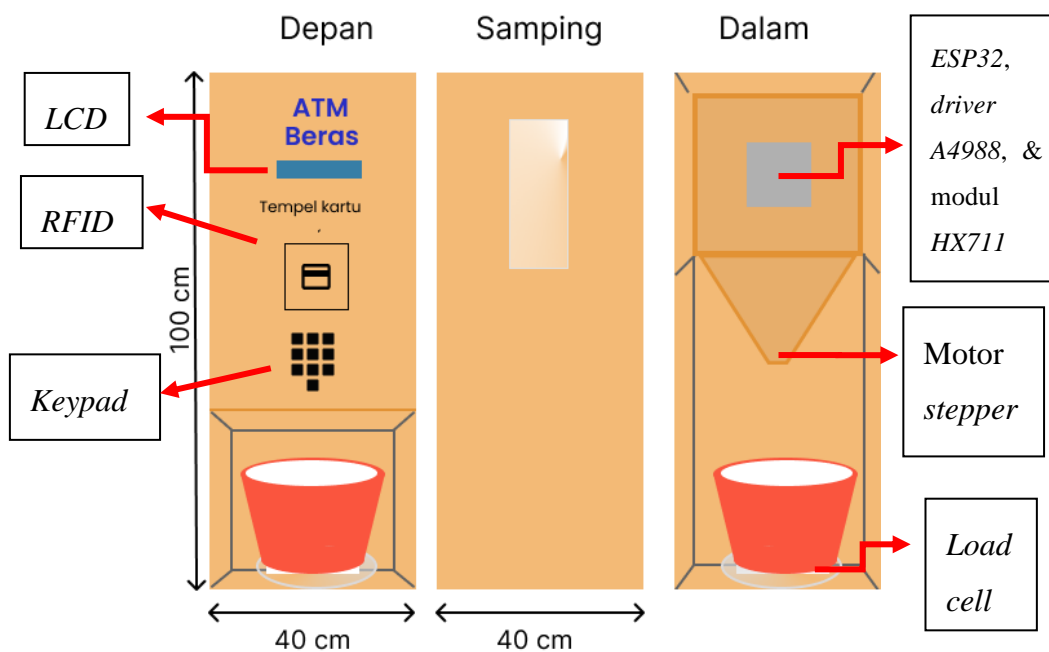
Gambar 3.1 merupakan diagram alir dari penelitian ini yang mana dimulai dari mencari topik atau ide perancangan sistem. Setelah ide perancangan sistem sudah

didapatkan maka akan dilanjutkan dengan mencari dan mempelajari literatur tentang ide perancangan sistem untuk menentukan spesifikasi sistem. Jika spesifikasi tidak tersedia atau kurang efektif, maka pencarian literatur dilakukan kembali agar mendapatkan spesifikasi sistem yang sesuai.

Kemudian penelitian dilanjutkan dengan perancangan sistem, setelah perancangan sistem berhasil, maka dilanjutkan dengan pengujian sistem. Jika pengujian sistem tidak berhasil, maka sistem perlu dirancang kembali agar mendapatkan hasil yang diinginkan. Setelah pengujian sistem berhasil, penelitian dilanjutkan dengan pengambilan data dari sistem yang berhasil diuji untuk nantinya akan dilakukan proses analisis data untuk mendapatkan kesimpulan dari keseluruhan sistem yang sudah berhasil dibuat.

3.5 Sketsa Desain Perancangan

Adapun sketsa desain perancangan dari penelitian ini adalah sebagai berikut.

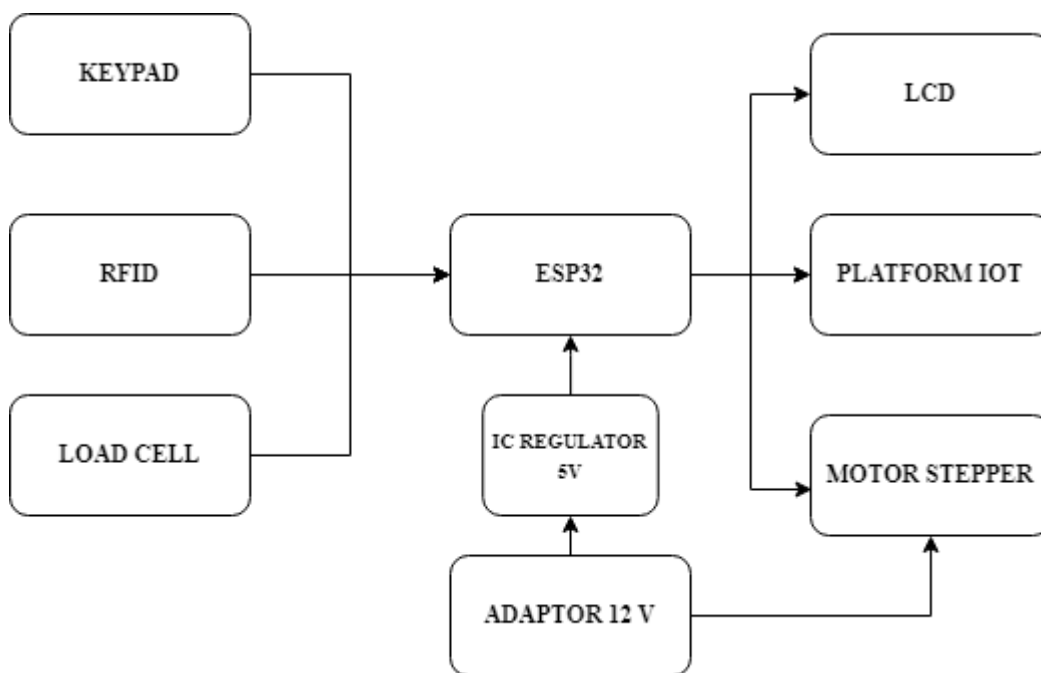


Gambar 3. 2 Desain alat

ATM beras ini berukuran dengan dimensi dengan panjang 40 cm lebar 40 cm dan tinggi 100 cm, bahan yang akan digunakan untuk pembuatan alat ini adalah triplek dan kayu. Bagian depan terdapat komponen-komponen seperti *LCD*, *RFID*, *Load Cell*, dan *Keypad*. Kemudian bagian sampingnya terdapat kaca untuk memantau sisa beras yang ada pada tabung. Kemudian dibagian dalam terdapat *ESP32*, *driver motor A4988*, dan modul *HX711* yang rangkai dalam satu *PCB*, dan terdapat motor *stepper Nema17*.

3.6 Diagram Blok Sistem

Adapun diagram blok sistem pada penelitian ini dapat dilihat sebagai berikut:



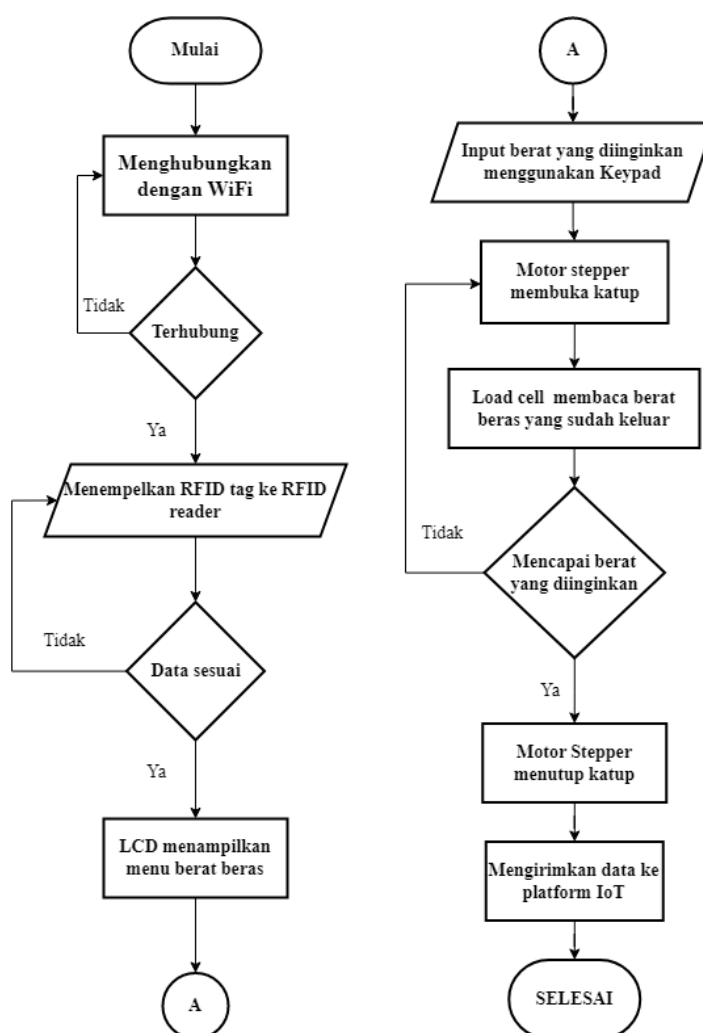
Gambar 3. 3 Diagram blok sistem

Berdasarkan Gambar 3.3 input pada sistem berupa *RFID*, *Keypad* dan *Load cell*. Pada *RFID* di dalamnya berisi data-data penerima bantuan yang akan berperan sebagai alat autentikasi, *Keypad* digunakan untuk memilih output berat dari beras yang diinginkan, dan *Load cell* digunakan untuk menimbang berat dari beras yang keluar dari tabung. Kemudian pemrosesan pada sistem ada *ESP32* sebagai

mikrokontroler. Pada bagian output pada sistem ini terdapat *LCD*, motor *stepper*, dan *platform IoT*. *LCD* pada alat ini digunakan untuk menampilkan informasi atau tanggapan dari sistem yang sedang dijalankan, motor *stepper* berfungsi untuk membuka dan menutup katup pada tabung, dan *platform IoT* yang digunakan untuk menampilkan dan menyimpan data-data dari pengguna yang sudah mengambil beras.

3.7 Perancangan Sistem

Adapun perancangan sistem pada penelitian ini adalah sebagai berikut.

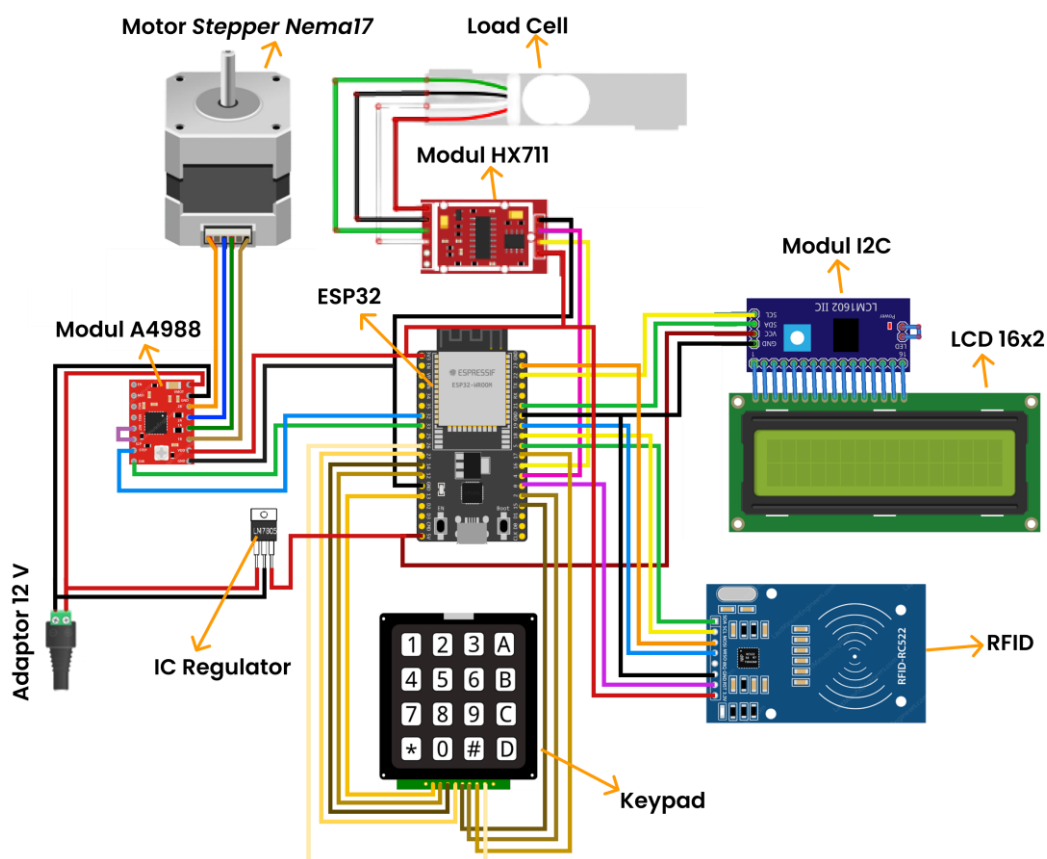


Gambar 3. 4 Diagram alir perancangan sistem

Berdasarkan Gambar 3.4 merupakan diagram alir perancangan sistem yang mana diawali menghubungkan dengan *WiFi*, sistem ini berjalan hanya apabila *WiFi* sudah terhubung. Kemudian pengguna akan menempelkan kartu *tag RFID* yang sudah diberikan sebelumnya untuk proses autentikasi pengguna yang akan diproses oleh *ESP32*, apabila sesuai maka akan dilanjutkan dengan penampilan menu dari berat beras yang disediakan. Setelah menginput berat beras menggunakan *Keypad* maka *ESP32* akan melakukan perintah terhadap motor *stepper* untuk membuka katup yang akan dibaca oleh *Load cell*, apabila berat beras sudah mencapai dari berat yang diinginkan maka motor *stepper* akan menutup katupnya. Selanjutnya data dari pengguna yaitu nama dan berat beras yang sudah diambil akan dikirimkan ke *platform IoT* untuk dijadikan *database*.

3.8 Langkah-langkah Pengujian

Sebelum proses pengujian dilakukan, komponen-komponen yang digunakan perlu dihubungkan dengan mikrokontroler agar dapat bekerja sesuai dengan kebutuhan. Berdasarkan Gambar 3.5 seluruh komponen yang digunakan terhubung dengan mikrokontroler *ESP32* agar masing-masing komponen dapat bekerja sesuai dengan fungsinya. Setiap komponen terhubung menggunakan kabel *jumper* yang memiliki beragam warna. Berikut ini merupakan rangkaian keseluruhan komponen yang digunakan pada penelitian ini.



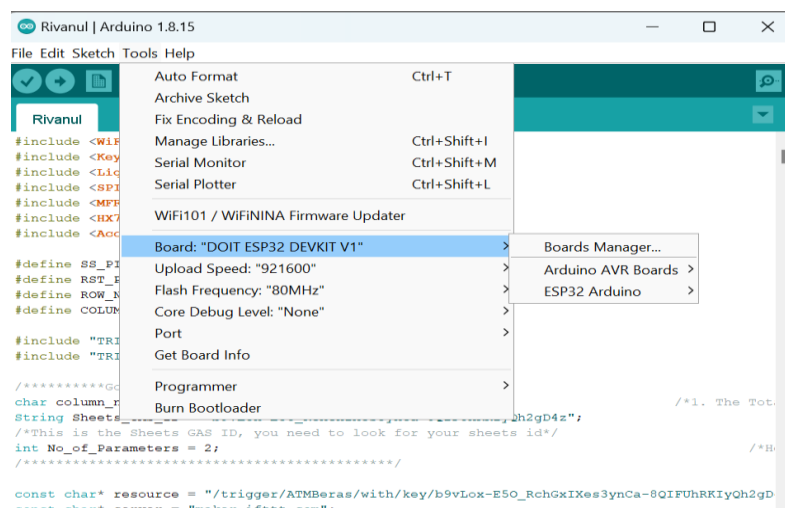
Gambar 3. 5 Diagram Pengkabelan Seluruh Komponen

Berikut ini merupakan langkah-langkah pengujian yang telah dilakukan pada perangkat atau komponen yang digunakan.

3.8.1 Mikrokontroler ESP32

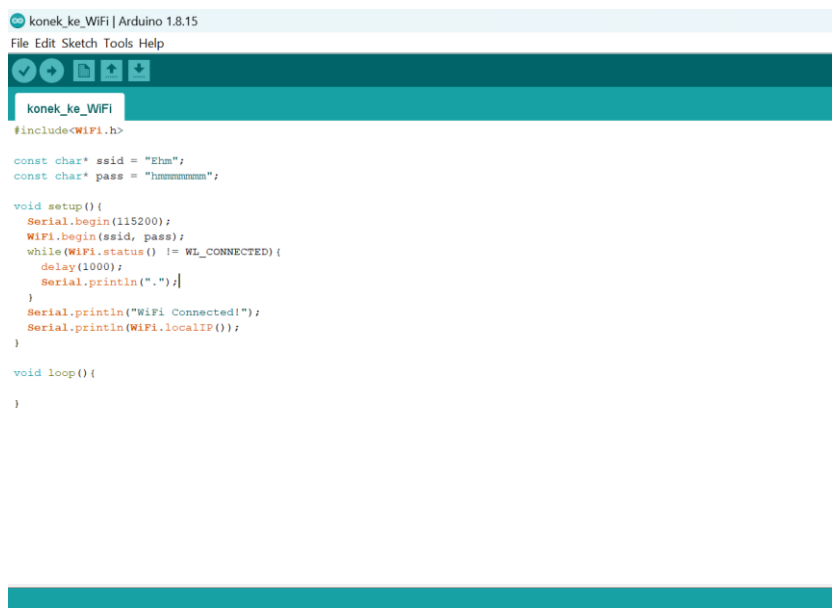
Berikut ini merupakan langkah-langkah pengujian terhadap mikrokontroler ESP32.

1. Meng-*instal software Arduino IDE* versi 1.8.15.
2. Menghubungkan *ESP32* ke laptop dengan kabel USB.
3. Menjalankan *Arduino IDE* versi 1.8.15.
4. Memastikan mikrokontroler *ESP32* apakah terhubung atau tidak dengan cara melihatnya pada menu *tools*, lalu memeriksa *submenu board* seperti pada Gambar 3.6.



Gambar 3. 6 Submenu board

5. Membuat program sederhana pada jendela *editor Arduino IDE*, seperti pada Gambar 3.7.



Gambar 3. 7 Tampilan jendela *editor*

6. Memeriksa program dengan cara memilih *verify* yang berbentuk logo *checklist*, dapat dilihat pada 3.8. Tujuan dari menu *verify* ini adalah untuk memastikan apakah program yang akan di-*upload* ke dalam mikronkontroler *ESP32* terdapat kesalahan atau tidak.



```

konek_ke_WiFi | Arduino 1.8.15
File Edit Sketch Tools Help
konek_ke_WiFi
#include<WiFi.h>

const char* ssid = "Ehm";
const char* pass = "hmmmmmmmm";

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  while(WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }
  Serial.println("WiFi Connected!");
  Serial.println(WiFi.localIP());
}

void loop() {
}

```

Gambar 3. 8 *Verify code*

7. Apabila program tidak terdapat kesalahan, selanjutnya melakukan *upload* program program dengan memilih menu *upload* yang berbentuk logo tanda panah seperti pada Gambar 3.9.



```

konek_ke_WiFi | Arduino 1.8.15
File Edit Sketch Tools Help
konek_ke_WiFi
#include<WiFi.h>

const char* ssid = "Ehm";
const char* pass = "hmmmmmmmm";

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  while(WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }
  Serial.println("WiFi Connected!");
  Serial.println(WiFi.localIP());
}

void loop() {
}

```

Gambar 3. 9 *Upload code*

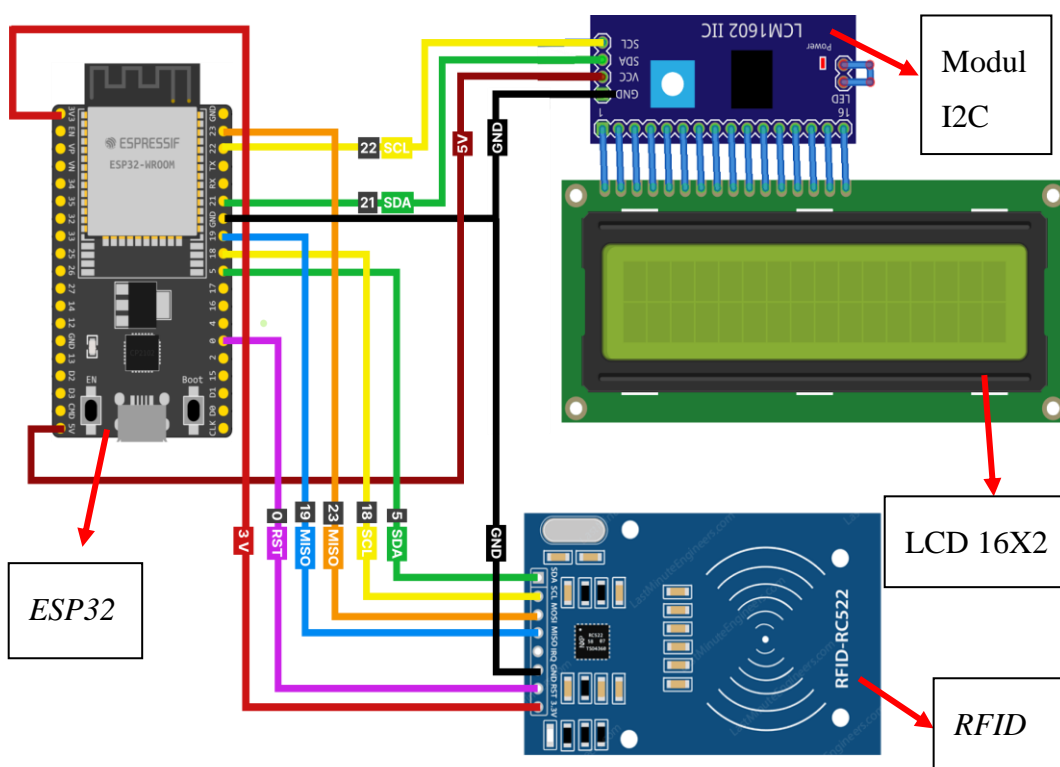
8. Melihat respon dari mikrokontroler *ESP32*, kemudian memastikan respon *ESP32* dalam kondisi *running* dan dapat digunakan sebagai pengendali utama pada sistem.

Dapat dilihat pada Gambar 3.10 bahwa *ESP32* dapat terhubung dengan *WiFi* yang telah ditentukan. Ketika *ESP32* belum terhubung dengan *WiFi* maka berdasarkan program yang diberikan *serial monitor* akan menampilkan titik (.) sampai *ESP32* sudah terhubung dan mendapatkan alamat IP dari *WiFi* tersebut.

3.8.3 RFID (Radio Frequency Identification)

Berikut ini merupakan langkah-langkah dalam pengujian RFID.

- Menghubungkan *RFID* dengan mikrokontroler *ESP32* seperti pada Gambar 3.12.



Gambar 3. 12 Diagram pengkabelan *RFID*

b. Melakukan *upload* program melalui *Arduino IDE*


```

tes_rfid | Arduino 1.8.15
File Edit Sketch Tools Help

tes_rfid

#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 5 // ESP32 pin GIOP5
#define RST_PIN 0 // ESP32 pin GIOP27

MFRC522 rfid(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(115200);
  SPI.begin(); // init SPI bus
  rfid.PCD_Init(); // init MFRC522

  Serial.println("Tap an RFID/NFC tag on the RFID-RC522 reader");
}

void loop() {
  if (rfid.PICC_IsNewCardPresent()) { // new tag is available
    if (rfid.PICC_ReadCardSerial()) { // NUID has been readed
      MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
      Serial.print("RFID/NFC Tag Type: ");
      Serial.println(rfid.PICC_GetTypeName(piccType));

      // print UID in Serial Monitor in the hex format
      Serial.print("UID:");
      for (int i = 0; i < rfid.uid.size; i++) {
        Serial.print(rfid.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(rfid.uid.uidByte[i], HEX);
      }
      Serial.println();

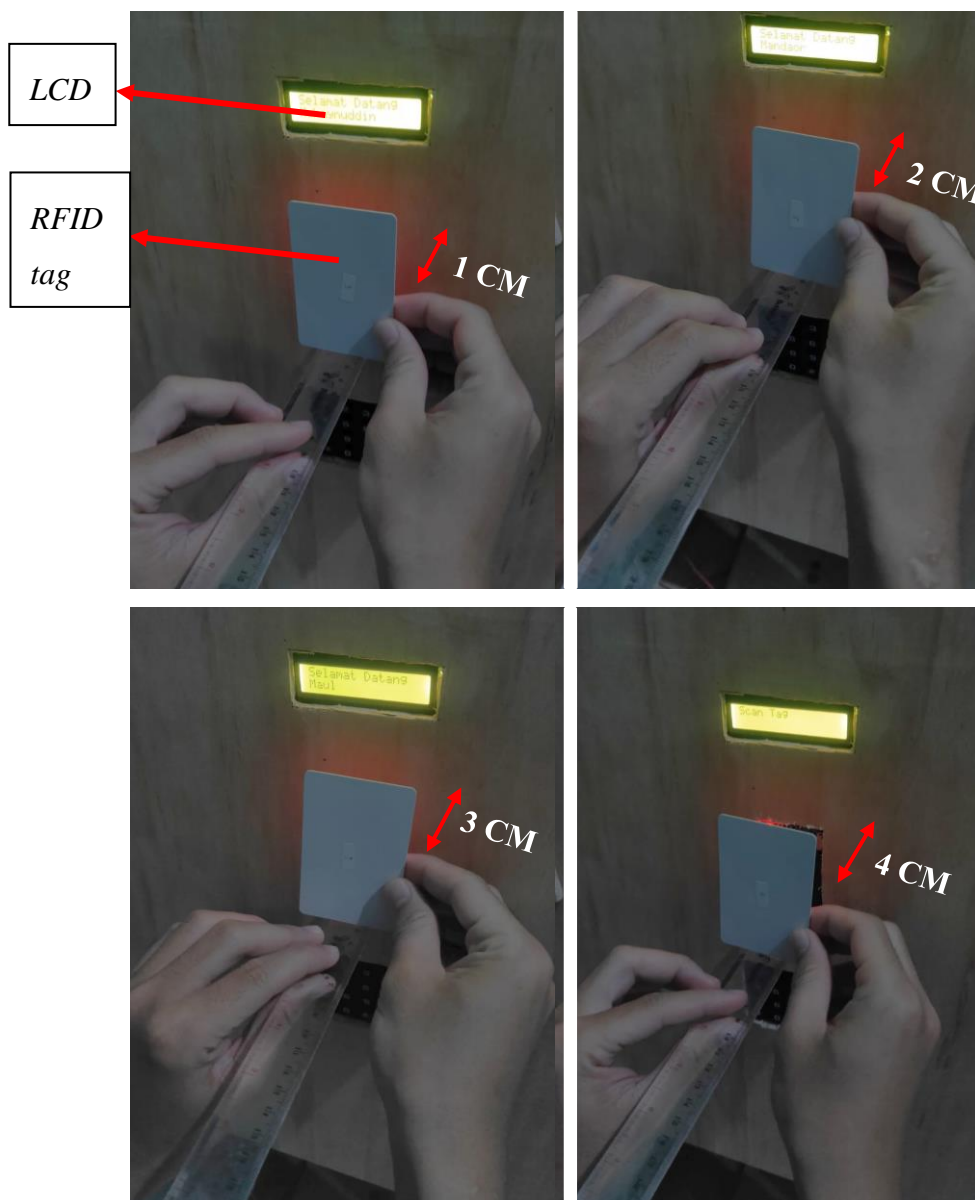
      rfid.PICC_HaltA(); // halt PICC
      rfid.PCD_StopCryptol(); // stop encryption on PCD
    }
  }
}

Leaving...
Hard resetting via RTS pin...
Invalid library found in C:\Program Files (x86)\Arduino\libraries\LiquidCrystal_I2C-1.

```

Gambar 3. 13 Program *RFID*

c. Melakukan pengukuran dan pengujian *RFID*



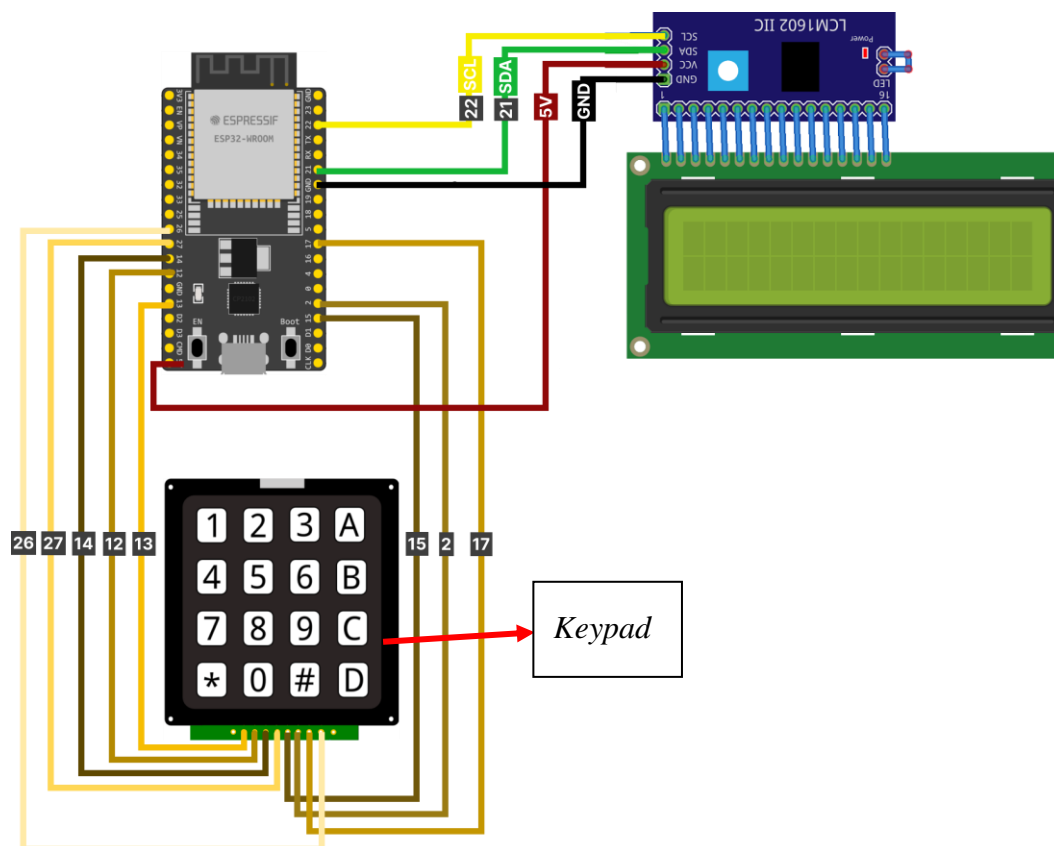
Gambar 3. 14 Pengukuran jarak kartu *RFID*

Gambar 3.14 merupakan proses pengukuran jarak untuk menentukan jarak maksimal *RFID reader* dalam membaca *RFID tag*. Dapat dilihat pada Gambar 3.14 pengukuran dilakukan dalam berbagai jarak yaitu 1 cm, 2 cm, 3 cm, dan 4 cm.

3.8.4 Keypad

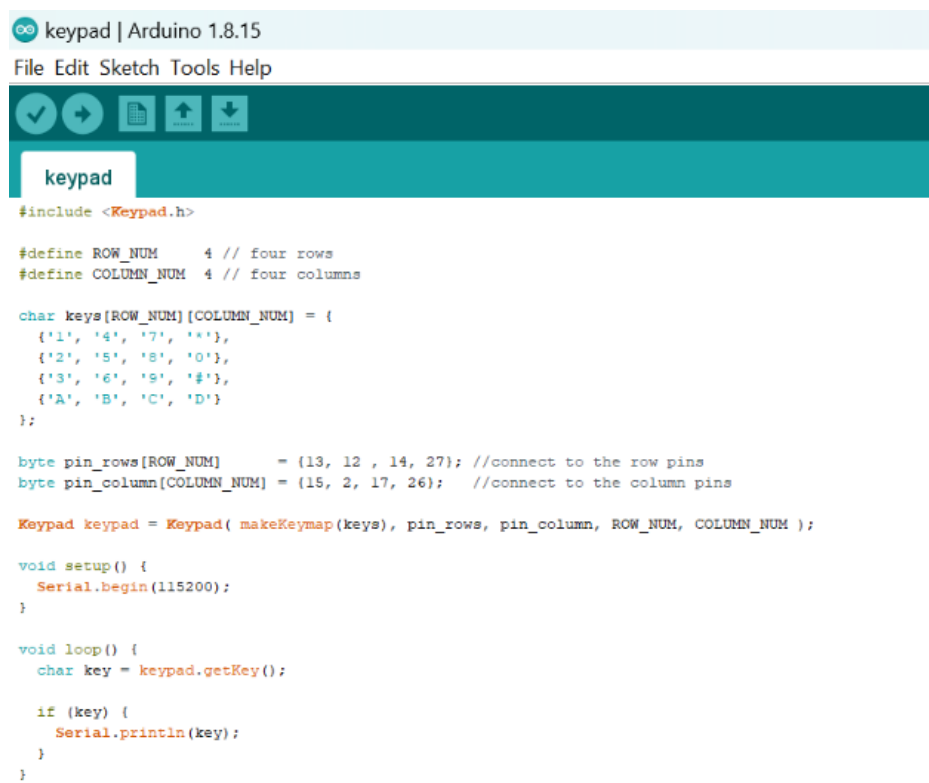
Berikut ini merupakan langkah-langkah dalam pengujian *Keypad*.

1. Menghubungkan *Keypad* dengan *ESP32* seperti pada Gambar 3.15.



Gambar 3. 15 Diagram pengkabelan *Keypad*

2. Melakukan *uploading* program melalui *Arduino IDE*.



```

keypad | Arduino 1.8.15
File Edit Sketch Tools Help

keypad

#include <Keypad.h>

#define ROW_NUM    4 // four rows
#define COLUMN_NUM 4 // four columns

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1', '4', '7', '*'},
  {'2', '5', '8', '0'},
  {'3', '6', '9', '#'},
  {'A', 'B', 'C', 'D'}
};

byte pin_rows[ROW_NUM]    = {13, 12, 14, 27}; //connect to the row pins
byte pin_column[COLUMN_NUM] = {15, 2, 17, 26}; //connect to the column pins

Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM );

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

void loop() {
  char key = keypad.getKey();

  if (key) {
    Serial.println(key);
  }
}

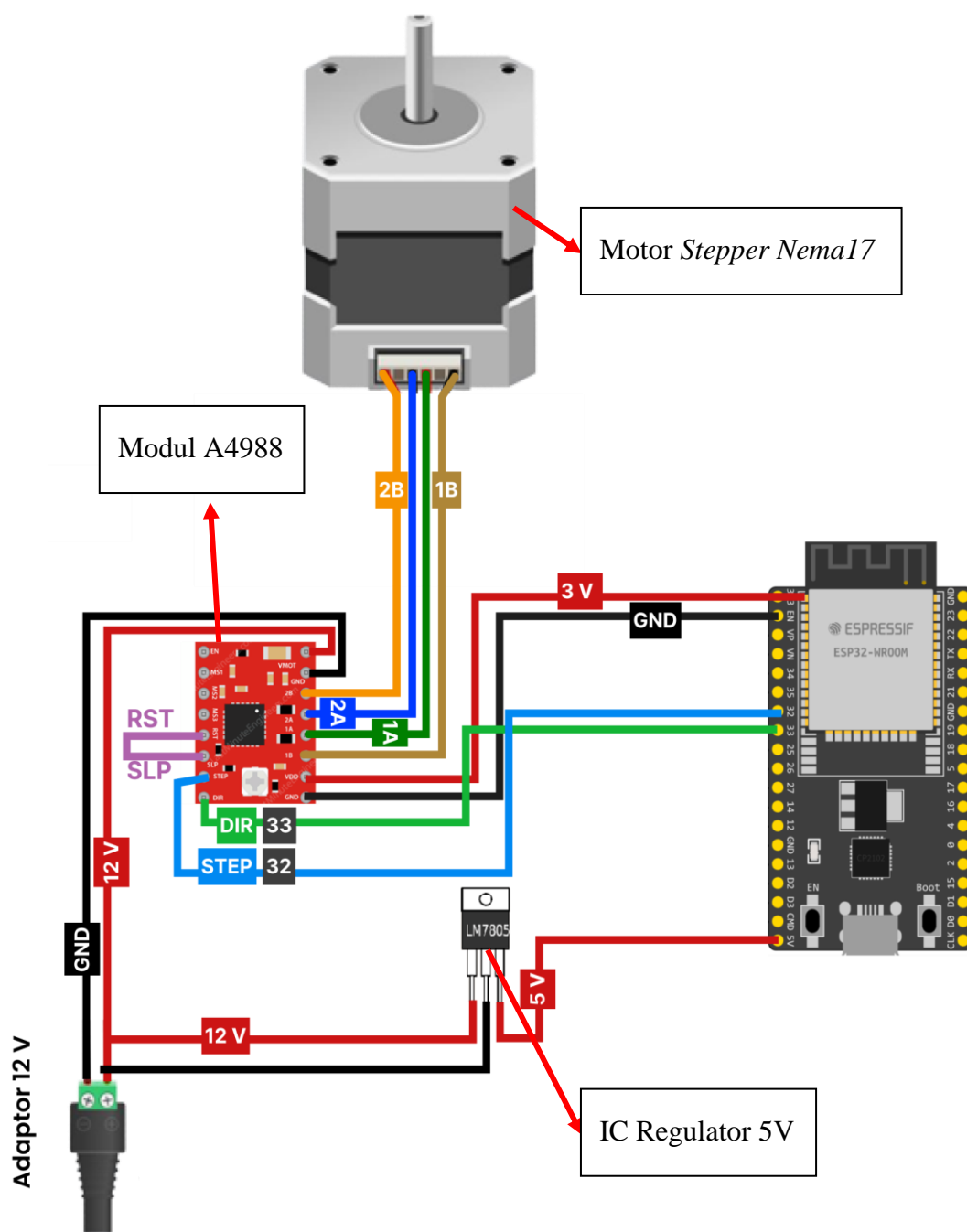
```

Gambar 3. 16 Program *Keypad*

3.8.5 Motor *Stepper Nema17*


Berikut ini merupakan langkah-langkah dalam pengujian Motor *Stepper Nema17*.

1. Menghubungkan *driver A4988* dan Motor *Stepper Nema17* dengan *ESP32* seperti pada Gambar 3.17.



Gambar 3. 17 Diagram pengkabelan Motor Stepper Nema17

2. Melakukan *upload* program melalui *software Arduino IDE*.



```

motor_A4988 | Arduino 1.8.15
File Edit Sketch Tools Help

motor_A4988
#include <AccelStepper.h>

const int DIR = 32;
const int STEP = 33;
const int steps_per_rev = 600;

void setup()
{
  Serial.begin(115200);
  pinMode(STEP, OUTPUT);
  pinMode(DIR, OUTPUT);
}

void loop()
{
  digitalWrite(DIR, HIGH);
  Serial.println("Spinning Clockwise...");

  for(int i = 0; i<steps_per_rev; i++)
  {
    digitalWrite(STEP, HIGH);
    delayMicroseconds(2000);
    digitalWrite(STEP, LOW);
    delayMicroseconds(2000);
  }
  delay(1000);

  digitalWrite(DIR, LOW);
  Serial.println("Spinning Anti-Clockwise...");

  for(int i = 0; i<steps_per_rev; i++)
  {
    digitalWrite(STEP, HIGH);
    delayMicroseconds(1000);
    digitalWrite(STEP, LOW);
    delayMicroseconds(1000);
  }
  delay(1000);
}

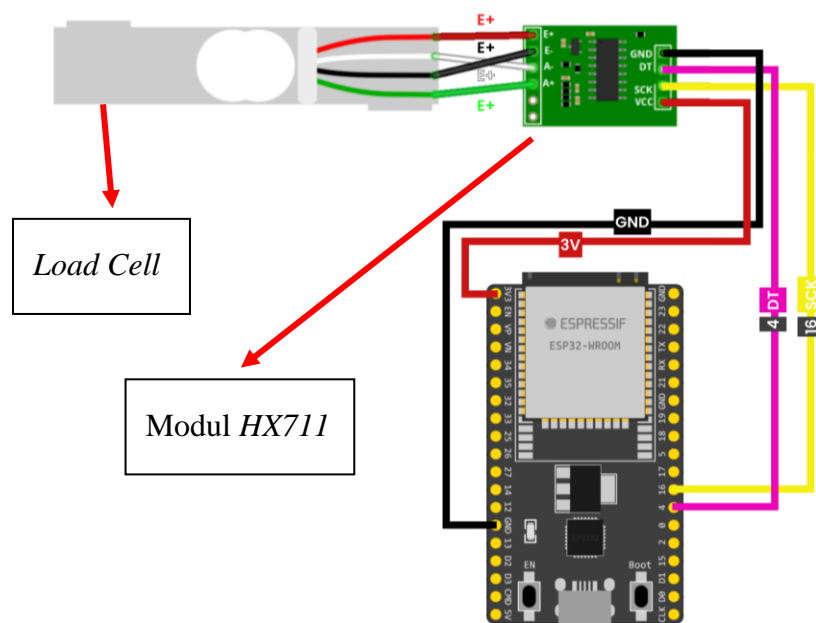
```

Gambar 3. 18 Program Motor *Stepper Nema17*

3.8.6 Load Cell

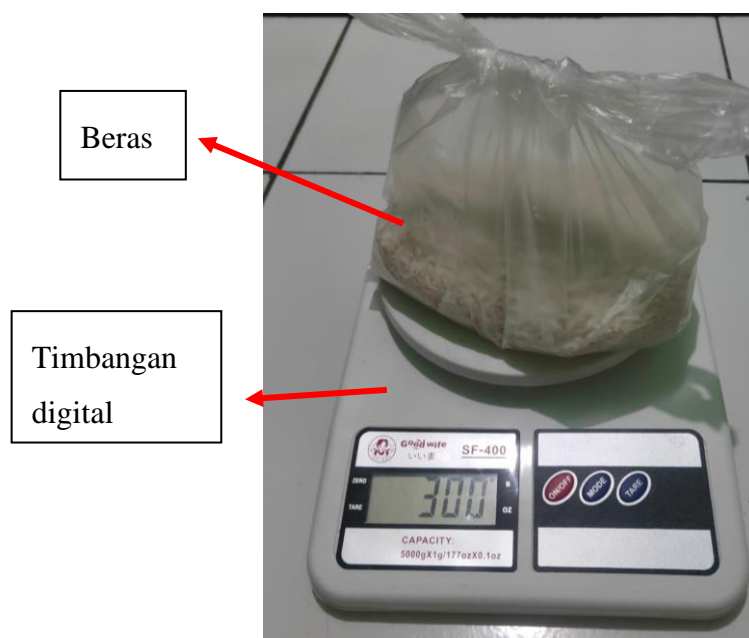
Sebelum melakukan pengujian, *Load Cell* perlu dilakukan kalibrasi agar *Load Cell* bekerja dengan baik dan hasilnya sesuai dengan standar yang sudah ditetapkan. Berikut ini langkah-langkah dalam pengkalibrasian *Load Cell*.

1. Menghubungkan *Load Cell* dengan ESP32 seperti pada Gambar 3.19.



Gambar 3. 19 Diagram pengkabelan *Load Cell*

- Menyiapkan objek yang beratnya sudah diketahui (dalam hal ini objek yang digunakan adalah beras dengan berat 300 g).



Gambar 3. 20 Objek dengan berat 209 g

3. Melakukan *upload* program untuk kalibrasi. Berikut ini program kalibrasi *Load Cell* yang ditunjukkan pada Gambar 3.21.

```

LoadCell_Calibration | Arduino 1.8.15
File Edit Sketch Tools Help

LoadCell_Calibration

#include "acc/rsc.h"
#include <HX711.h>

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 4;
const int LOADCELL_SCK_PIN = 16;

HX711 scale;

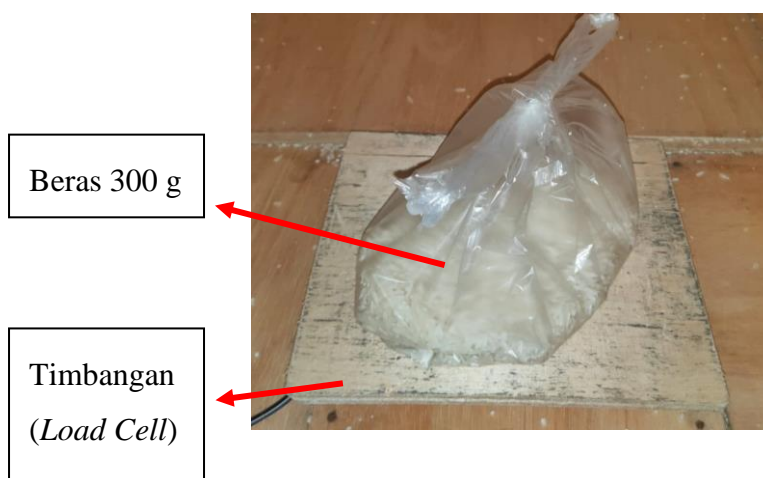
void setup() {
  Serial.begin(115200);
  rsc_clk_cpu_freq_set(RTC_CPU_FREQ_80M);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
}

void loop() {
  if (scale.is_ready()) {
    scale.set_scale();
    Serial.println("Tare... remove any weights from the scale.");
    delay(5000);
    scale.tare();
    Serial.println("Tare done...");
    Serial.print("Place a known weight on the scale...");
    delay(5000);
    long reading = scale.get_units(10);
    Serial.print("Result: ");
    Serial.println(reading);
  }
  else {
    Serial.println("HX711 not found.");
  }
  delay(1000);
}

```

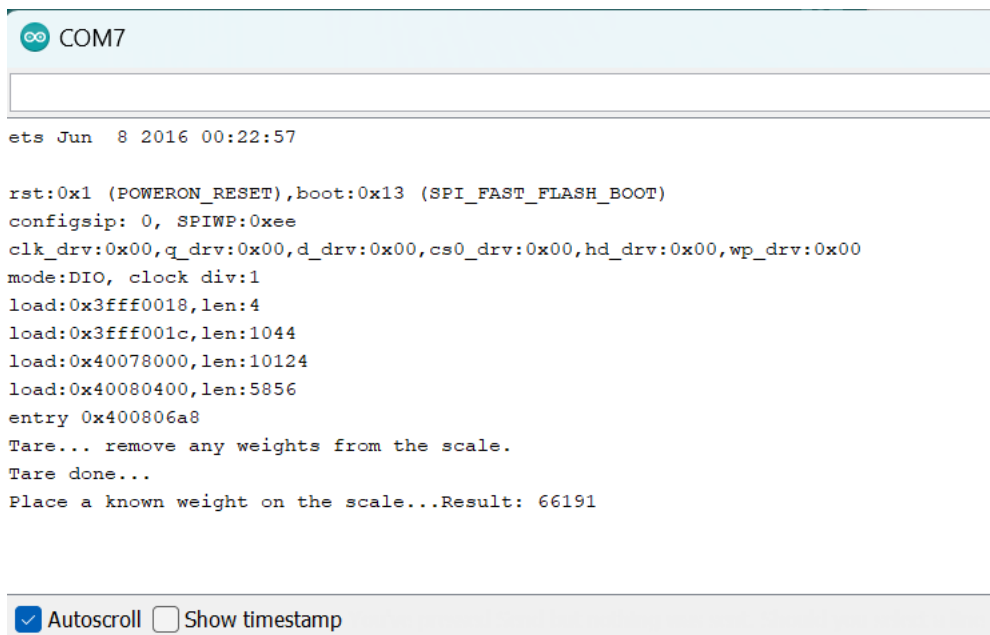
Gambar 3. 21 Program kalibrasi *Load Cell*

Mengikuti perintah dari *serial monitor*: “*remove any weights form the scale*” (menghilangkan semua beban dari timbangan). “*Then, place an object with a known weight on the scale*” (kemudian, meletakkan objek yang sudah diketahui beratnya).



Gambar 3. 22 Menimbang objek

4. Melihat hasil pembacaan nilai dari proses yang sudah dijalankan.



```

ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8
Tare... remove any weights from the scale.
Tare done...
Place a known weight on the scale...Result: 66191

```

Autoscroll Show timestamp

Gambar 3. 23 Hasil *serial monitor* kalibrasi *Load Cell*

5. Menghitung kalibrasi dengan rumus:

$$\text{Kalibrasi} = (\text{pembacaan nilai}) / (\text{berat yang diketahui})$$

Dapat dilihat sebelumnya pada *serial monitor*, nilai yang didapat yaitu 66191. Kemudian karena berat objek yang diketahui adalah 300 g, maka hasil kalibrasi yang didapatkan adalah $66191/300 = 220.63$

Setelah proses kalibrasi selesai, maka selanjutnya dilakukan proses pengujian terhadap *Load Cell*. Berikut ini merupakan langkah-langkah yang dilakukan dalam pengujian.

a. Melakukan *upload* program untuk menimbang melalui *Arduino IDE*.

```
LoadCell_Weighing
1 #include <HX711.h>
2
3 // HX711 circuit wiring
4 const int LOADCELL_DOUT_PIN = 4;
5 const int LOADCELL_SCK_PIN = 16;
6
7 HX711 scale;
8
9 void setup() {
10   Serial.begin(115200);
11   Serial.println("HX711 Demo");
12   Serial.println("Initializing the scale");
13   scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
14   scale.set_scale(220.63);
15   scale.tare();
16 }
17
18 void loop() {
19   Serial.print("Berat:\t");
20   int berat = scale.get_units();
21   Serial.println(berat);
22   delay(3000);
23 }
```

Gambar 3. 24 Program menimbang *Load Cell*

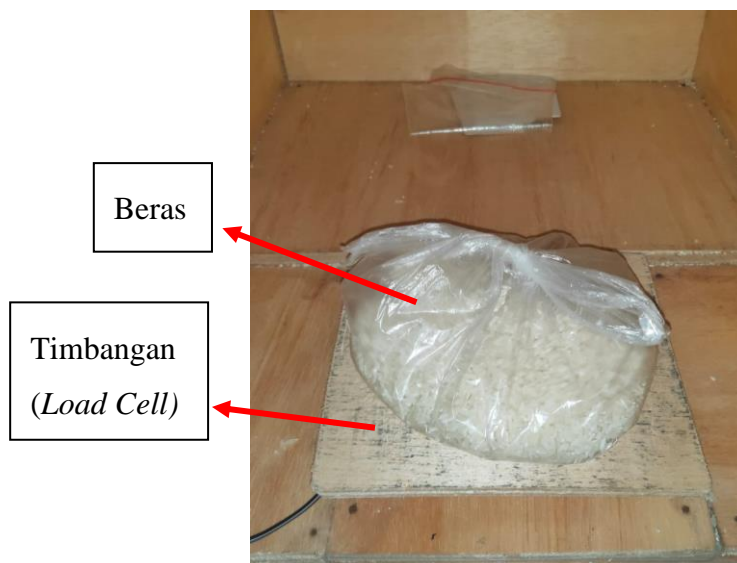
b. Melakukan penimbangan objek di timbangan digital.



Gambar 3. 25 Proses penimbangan beras di timbangan digital

Dapat dilihat pada Gambar 3.25 hasil penimbangan yang dilakukan timbangan digital terhadap beras yaitu 500 g.

c. Meletakkan beras di timbangan (*Load Cell*).

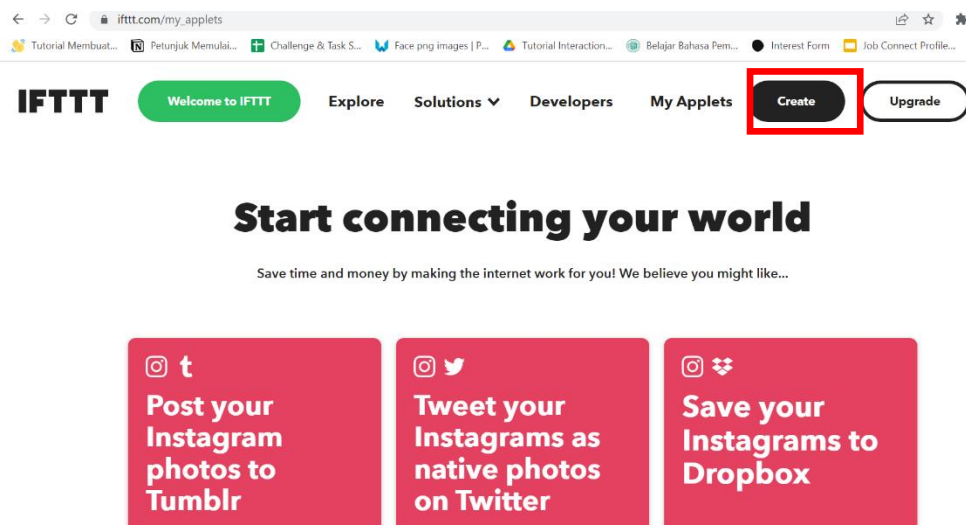


Gambar 3. 26 Menimbang objek diatas *Load Cell* (timbangan)

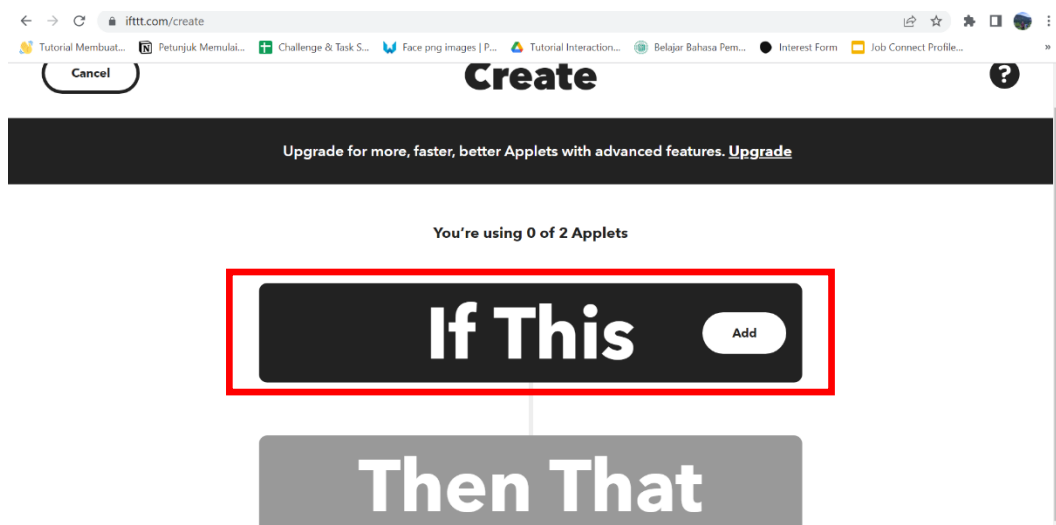
3.8.7 Google Sheets

Pengujian ini dilakukan untuk mengetahui apakah mikrokontroler *ESP32* dapat mengirimkan data yang dibutuhkan ke *Google Sheets* sebagai *Platform IoT* yang digunakan pada penelitian ini. Dalam prosesnya kita perlu menggunakan layanan *third-party*, dalam hal ini yang digunakan adalah *IFTTT Webhooks*. Berikut ini merupakan langkah-langkah agar *ESP32* dapat mengirimkan data ke *Google Sheets*.

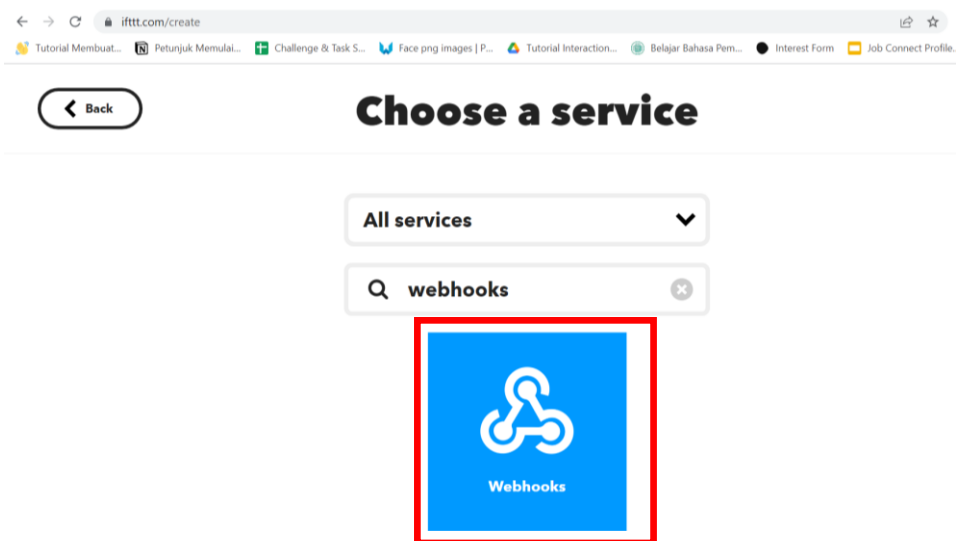
1. Mendaftarkan akun terlebih dahulu di ifttt.com.
2. Melakukan pembuatan *Applet* baru dengan membuka tab "Create".

Gambar 3. 27 Tab *Create*

3. Memilih tulisan “*If This*” untuk menambahkan pelayanan yang diinginkan sebagai kondisi awal untuk men-*trigger* suatu aksi lainnya.

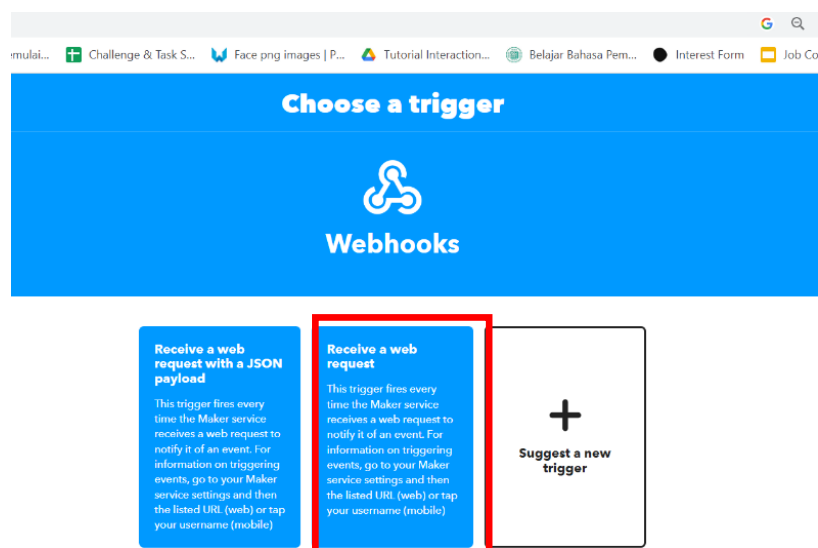
Gambar 3. 28 Menu *If This*

4. Memilih *service Webhooks* untuk menyimpan data dari *ESP32* dan meneruskannya ke layanan *Google Sheets*.



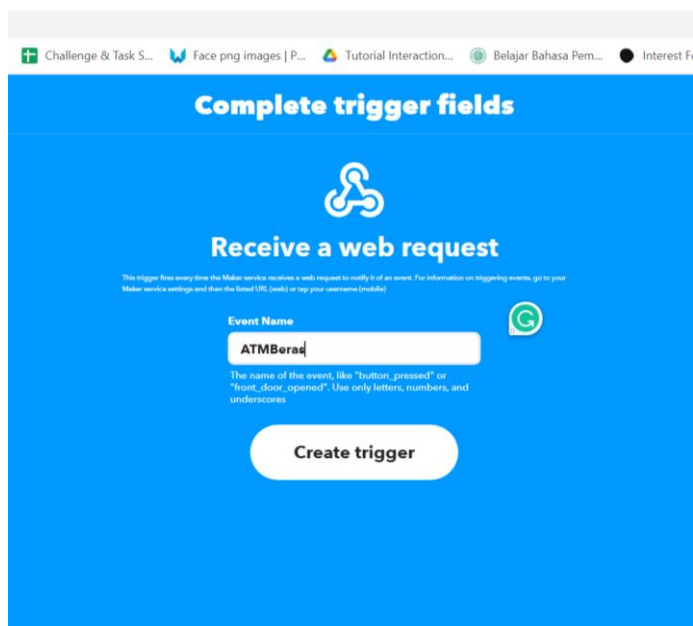
Gambar 3. 29 Service Webhooks

5. Memilih *trigger* “Receive a web request”.



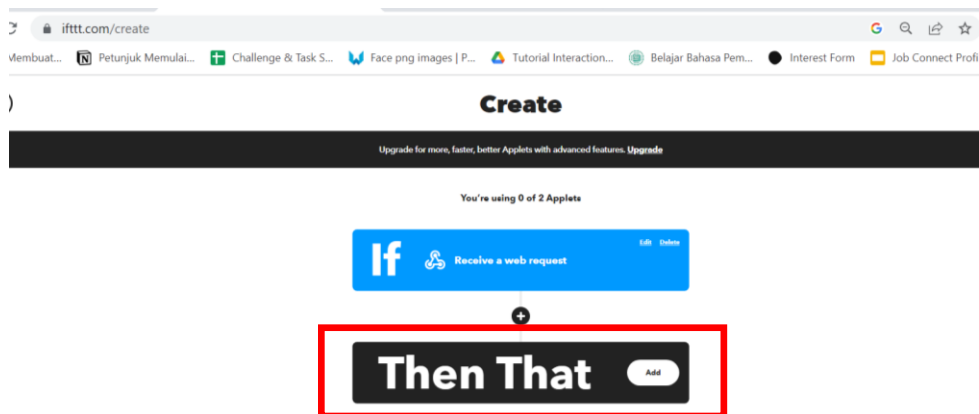
Gambar 3. 30 Memilih *Trigger*

6. Memberi nama *event* yang akan dibuat.



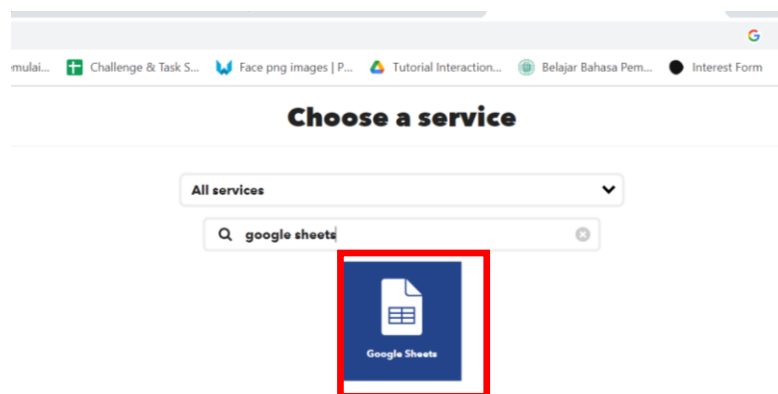
Gambar 3. 31 Nama *Event*

7. Memilih kalimat “*Then That*” untuk dijadikan eksekutor karena kondisi “*If This*” sudah terpenuhi.



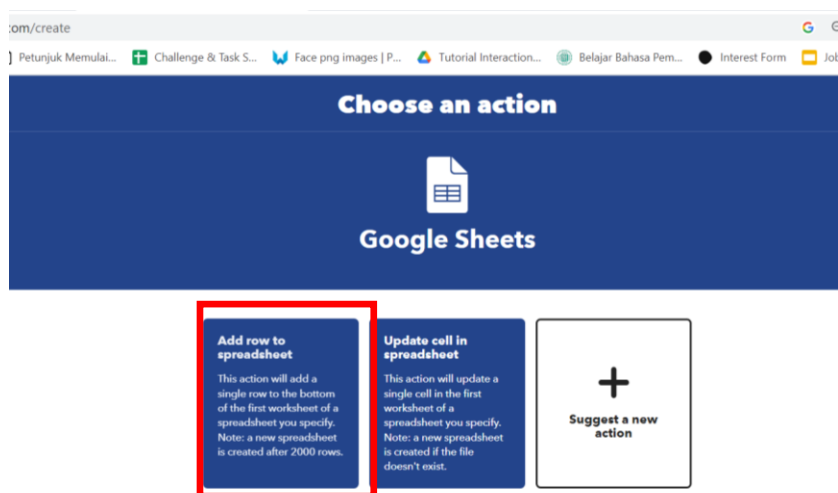
Gambar 3. 32 Menu *Then That*

8. Memilih *Google Sheets* sebagai eksekutor.



Gambar 3. 33 *Google Sheets*

9. Memilih aksi “*Add row to spreadsheet*”



Gambar 3. 34 *Add row to spreadhseet*

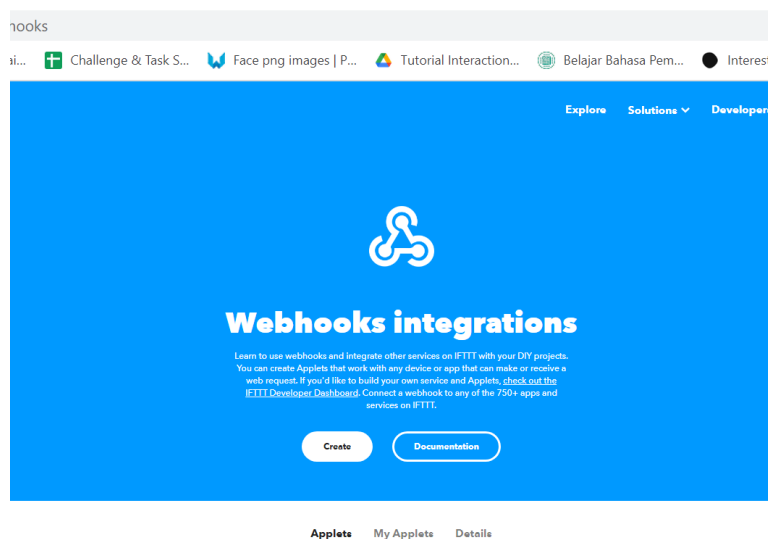
10. Melengkapi informasi lainnya seperti judul *file Spreadsheet* yang ingin dibuat, format baris, serta *path file Spreadsheet* di *Google Drive*.

The screenshot shows a web form titled "Add row to spreadsheet" on a dark blue background. At the top, there's a Google Sheets icon and the title. Below the title, a small text explains the action: "This action will add a single row to the bottom of the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created after 2000 rows." The form consists of several sections, each with an "Add ingredient" button:

- Google Sheets account:** A dropdown menu showing "nahdrulrivandio7@gmail.com" and an "Add new account" button.
- Spreadsheet name:** A text input field containing "IFTTT_Maker_Webhooks_Events". Below it, a note says "Will create a new spreadsheet if one with this title doesn't exist".
- Formatted row:** A text input field containing "OccurredAt ||| EventName ||| Value1 ||| Value2". Below it, a note says "Use '|||' to separate cells".
- Drive folder path:** A text input field containing "IFTTT/MakerWebhooks/EventName". Below it, a note says "Format: some/folder/path (defaults)".

Gambar 3. 35 Melengkapi informasi

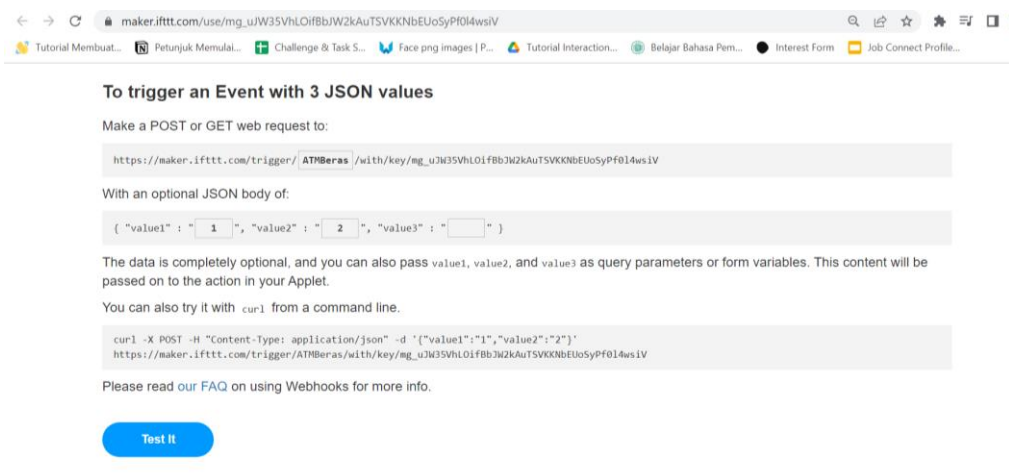
11. Membuka halaman layanan *Webhook* kemudian mengklik "*Documentation*".



Gambar 3. 36 Opsi *Documentation*

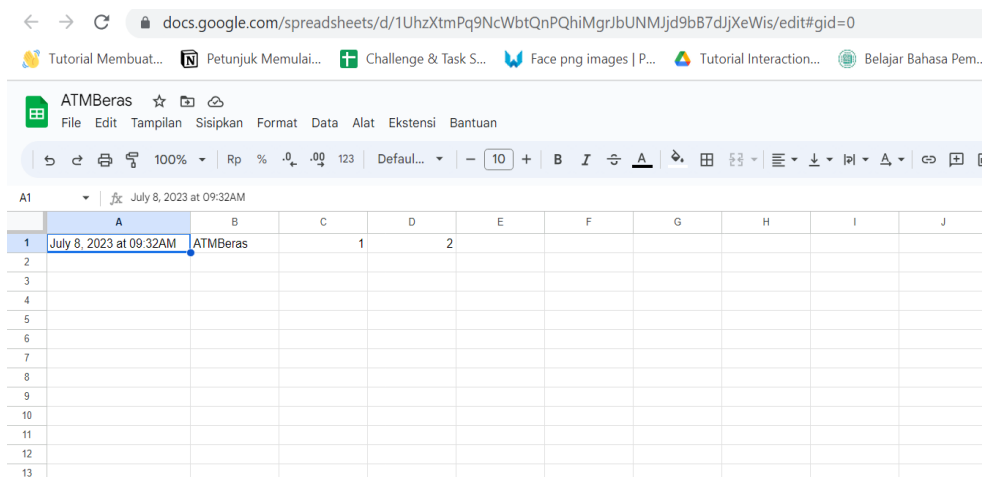
12. Mengkonfigurasi alamat *web request* dengan memasukan nama *triggering event* yang sebelumnya sudah dibuat pada URL dibawah kalimat

“*Make a Post or GET web request to:*”. Kemudian mengisi nilai bebas untuk menguji *Applet* yang sudah dibuat.



Gambar 3. 37 Konfigurasi alamat web

- File *Google Sheets* yang sudah dibuat terletak pada *Google Drive* dari email yang didaftarkan, yaitu pada folder IFTTT > *Maker Webhooks* > Nama *Event*. Selanjutnya, untuk mengetahui apakah *Google Sheets* berjalan dengan baik maka dapat dilihat apakah nilai bebas yang sebelumnya diberikan sudah tersimpan pada *Google Sheets*.



Gambar 3. 38 Hasil tes dengan nilai bebas pada *Google Sheets*

14. Melakukan *upload* program melalui *software Arduino IDE*.

The screenshot shows the Arduino IDE interface for the project 'ESP32_to_Sheets' on an Arduino 1.8.15 board. The code is as follows:

```

#include <WiFi.h>

#include "TRIGGER_WIFI_ESP32.h"
#include "TRIGGER_GOOGLESHEETS_ESP32.h"

char column_name_in_sheets[ ][20] = {"value1", "value2"};          /*1. The Total no of column depends on how ma
String Sheets_GAS_ID = "b9vLox-E5O_RchGxIXes3ynCa-8QIFUhrRKiYQh2gD4z";

const char* resource = "/trigger/ATMBeras/with/key/b9vLox-E5O_RchGxIXes3ynCa-8QIFUhrRKiYQh2gD4z";
const char* server = "maker.ifttt.com";

char* ssid = "Ehm";
char* password = "hmmmmmmmm";

char *name = "";
char *berat;

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}

void loop() {
  Serial.print("Connecting to ");
  Serial.print(server);

  WiFiClient client;
  int retries = 5;
  while(!client.connect(server, 80) && (retries-- > 0)) {
    Serial.print(".");
  }
  Serial.println();
  if(!client.connected()) {
    Serial.println("Failed to connect ");
  }
}

```

Below the code, the IDE shows the compilation status:

```

Done compiling.

Sketch uses 703234 bytes (53%) of program storage space. Maximum is 1310720 bytes.
Global variables use 38336 bytes (11%) of dynamic memory, leaving 289344 bytes for local variables. Maximum is 327680 bytes.

```

Gambar 3. 39 Program *ESP32 to Google Sheets*

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan dari penelitian adalah sebagai berikut.

1. Telah terealisasi sistem ATM Beras berbasis *IoT* dengan mikrokontroler *ESP32* yang dapat menyimpan *database* secara *real-time*
2. Berdasarkan pengujian, berat beras rata-rata yang keluar untuk setiap pengambilan masing-masing target yaitu 1.040 g untuk target 1.000 g, 2.026 g untuk target 2.000 g, dan 3.053 g untuk target 3.000 g, dan *error* rata-rata dari setiap target berturut-turut yaitu 4%, 1,3%, dan 1,9%.
3. Telah terealisasi sistem keamanan ATM Beras baik keamanan dengan *RFID* maupun dengan sistem *auto disconnect* saat *WiFi* terputus sehingga proses yang sedang berjalan berhenti.

5.2 Saran

Adapun saran dari penelitian ini adalah sebagai berikut.

1. Memperbaiki konstruksi dari ATM Beras baik dari wadah penyimpanan maupun dari lubang keluarnya beras, sehingga selisih antara *error* dengan target berat tidak terlalu jauh.
2. Menambahkan sensor pengukuran agar dapat mengetahui batas minimal beras yang berada dalam wadah penyimpanan.
3. Menambahkan sistem keamanan tambahan dari ATM Beras agar ketika terjadi listrik mati maka proses yang sedang dijalankan akan tetap terus berjalan.
4. Menambahkan mekanisme penambahan kartu *tag* pengguna/*user* tanpa harus melakukannya dengan pemrograman.

DAFTAR PUSTAKA

- [1] Arief JL, Khollaqul. 2022. “Rancang Bangun *Prototype* Atm Beras Berbasis Mikrokontroler *ATMega328p*”. Lampung : Universitas Lampung, Teknik Elektro.
- [2] Mirfan. 2016. “Mesin Penyaji Beras Secara Digital, ” dalam *Jurnal Ilmiah ILKOM Volume 8 No. 2* (halm. 126-131). Makasar: STMIK Handayani
- [3] Billah, M.Mobtazim dkk. 2018. “Mesin Otomatis Pengambilan Raskin Menggunakan *RFID* Berbasis Mikrokontroler *Arduino*”. Jember: Universitas Muhammadiyah Jember, Teknik Elektro.
- [4] Syahriel, Syafrie dkk. 2021. “Perancangan ATM Raskin Berbasis FRID dan *Internet of Things (IoT)* untuk Masyarakat Tidak Mampu” dalam *J-Com (Journal of Computer) Volume 1 No. 1* (hlm 153-158). Sumatera Utara: STMIK Royal, Prodi Sistem Komputer dan Sistem Informasi.
- [5] Putri, Safira Angelia dkk. 2022. “Rancang Bangun ATM Beras Raskin Dengan *RFID* E-KTP Menggunakan Teknik Counter Berbasis *Arduino Uno*” dalam *Jurnal Sistem Komputer TGD Volume 1 No. 6* (halm. 214-223). Sumatera Utara: STMIK Triguna Dharma, Sistem Komputer, Sistem Informasi dan Teknik Komputer.
- [6] Simbolon, Abiko dkk. 2020. “Rancang Bangun Alat ATM Beras Dengan Menggunakan *Keypad* Berbasis Mikrokontroler *Arduino Uno*” dalam e-ISSN: 2407 - 7267. Samarinda: Universitas Mulawarman, Teknik Elektro.
- [7] Rahmawati, Diana dkk. 2021. “Rancang Bangun Mesin Penjual Beras Berbasis Mikrokontroler *ATMega16*” dalam *Cyclotron Volume 4 No. 2* (hlm. 46-51). Madura: Universitas Trunojoyo, Teknik Elektro.
- [8] Y. Efendi. 2018. “*Internet Of Things (IoT)* Sistem Pengendalian Lampu Menggunakan *Raspberry Pi* Berbasis *Mobile*,” dalam *J. Ilm vol. 4, no. 2*, (hlm. 21–27). Ilmu Komput. Fak. Ilmu Komput. Univ. Al Asyariah Mandar.
- [9] Binar Academy. 8 Juni 2022. “*IoT* adalah *Internet of Things*, Simak Penjelasan Lengkapnya”. Diakses pada 20 Januari 2023, dari <https://www.binaracademy.com/blog/internet-of-things-dan-penjelasan-lengkapnya>.

- [10] Ardutech. 5 Maret 2020. “Mengenal *ESP32 Development Kit* untuk *IoT(Internet of Things)*”. Diakses pada tanggal 20 Januari 2023 dari <https://www.ardutech.com/mengenal-ESP32-development-kit-untuk-IoT-internet-of-things/>.
- [11] Lestari, Hesty. 2010. “Perancangan Sistem Absensi dengan *RFID* Menggunakan Custom *RFID Reader*”. Bandung: Perpustakaan UNIKOM.
- [12] Mahir Elektro. 8 Januari 2023. Tutorial Lengkap Menggunakan *Driver L298N* dengan *Arduino*. Diakses pada tanggal 21 Januari 2023 dari <https://www.mahirelektro.com/2020/02/tutorial-menggunakan-driver-motor-l298n-pada-Arduino.html>.
- [13] Medila, Kusriyanto. 2016. “Rancang Bangun Timbangan Digital Terintegrasi Informasi Bmi Dengan Keluaran Suara Berbasis *Arduino Mega 2560*”. *Jurnal Teknoin*, Volume 22 Nomor 4. Universitas Islam Indonesia, Fakultas Teknologi Industri, Jurusan Teknik Elektro.
- [14] Agus, Purnama. 2018. “*Matrix Keypad* 4x4 Untuk Mikrokontroler – Elektronika Dasar”. Diakses pada tanggal Januari 2023 dari <https://elektronika-dasar.web.id/matrix-Keypad-4x4-untuk-mikrokontroler/>.
- [15] Juniardi, Lalu Satria. 2018. “Pengaman Arus Lebih Pada Kabel Listrik Dalam Instalasi Rumah Tangga Berbasis Web”. Universitas Muhammadiyah Malang.
- [16] Sari, Nur Indah. 2015. “Alat Pengering Tangan Otomatis Menggunakan Sensor Pir Berbasis Mikrokontroler Atmega 8535”. Politeknik Negeri Sriwijaya