

BAB I

PENDAHULUAN

1.1. Latar Belakang

Beras merupakan makanan pokok setiap hari bagi sebagian besar masyarakat Indonesia.[1] Selain digunakan sebagai makanan pokok oleh masyarakat Indonesia, beras juga digunakan dalam industri makanan, seperti industri tepung beras atau kue basah.

Beras menjadi salah satu komoditas yang sangat dicari di pasaran.[1] Permintaan beras yang tinggi mempunyai dampak yang positif bagi perekonomian masyarakat di Indonesia, khususnya bagi para pedagang beras sebagaimana dilakukan oleh ibu Emoh yang menjual beras baik secara eceran maupun grosir di Pasar Baru Kuningan, Jawa Barat.

Timbangan menjadi salah satu faktor pendukung pada proses penjualan beras yang dilakukan. Metode penimbangan beras secara manual yang ada sekarang masih sering dilakukan sehingga membutuhkan tenaga yang besar dan tidak efektif. Pedagang beras melayani pembeli dengan mengambil beras dalam karung dan menimbanginya. Kemudian beras dibawa dalam kantong plastik. Jika pembeli ingin membeli beras dengan uang seadanya atau dalam satuan rupiah, maka pedagang harus terlebih dahulu menghitung menggunakan kalkulator berapa gram beras yang diperolehnya dengan uang yang dimiliki pembeli. Selain itu, masih sering terjadi kesalahan pengamatan saat menentukan berat beras.

Penerapan teknologi merupakan salah satu solusi yang dapat dikembangkan untuk mengatasi permasalahan tersebut. Pemanfaatan teknologi dalam bidang perdagangan dan pertanian merupakan langkah maju untuk memudahkan aktivitas masyarakat. Digitalisasi dan otomatisasi berbagai macam perangkat sehari-hari telah banyak dirancang secara komprehensif dan ditransfer ke kehidupan sehari-hari di semua bidang.

Pada Tugas Akhir ini penyusun mendapatkan ide untuk merancang sebuah sistem timbangan digital berupa dispenser penjual beras eceran otomatis. Perancangan timbangan digital telah banyak dilakukan pada beberapa Tugas

Akhir sebelumnya seperti timbangan tepung, gula dan gabah. Input yang diberikan hanya berupa berat. Sedangkan dispenser penjual beras eceran yang dirancang dapat memproses input berupa nominal harga dari beras yang kemudian dikonversi menjadi berat dari beras.

Mengacu pada permasalahan tersebut, maka penyusun merancang dan membuat alat Tugas Akhir dengan judul “**RANCANG BANGUN DISPENSER PENJUAL BERAS ECERAN OTOMATIS BERBASIS ARDUINO UNO R3 MENGGUNAKAN SENSOR LOAD CELL**”. Dengan adanya alat ini diharapkan dapat membantu masyarakat dalam proses jual-beli beras. Kelebihan alat ini adalah konsumen dapat membeli beras dalam satuan rupiah dengan memasukkan nilai rupiah pada *keyboard / keypad*. Terdapat 3 jenis beras dengan harga dan jenis yang berbeda, yaitu beras 1 (Rp13.400), beras 2 (Rp14.000), dan beras 3 (Rp15.000). Penjual juga dapat mengubah harga tanpa harus memprogram ulang dan membuka aplikasi. Sehingga proses transaksi penjualan beras dapat berjalan dengan efektif dan efisien dengan tingkat akurasi pengukuran yang lebih tinggi.

1.2. Perumusan Masalah

Berdasar latar belakang tersebut, maka dapat dirumuskan permasalahan sebagai berikut:

1. Rancang bangun dispenser penjual beras eceran otomatis berbasis Arduino UNO R3 menggunakan sensor *load cell*.
2. Prinsip kerja dari rancang bangun dispenser penjual beras eceran otomatis berbasis Arduino UNO R3 menggunakan sensor *load cell*.
3. Mengukur dan mengontrol aliran beras, agar sesuai dengan harga input yang diminta menggunakan sensor *load cell*.

1.3. Tujuan Tugas Akhir

Tujuan yang ingin dicapai dari Tugas Akhir ini:

1. Membuat rancang bangun sistem mekanik dan kendali pada dispenser penjual beras eceran otomatis berbasis Arduino UNO R3 menggunakan sensor *load cell*.

2. Menguji kinerja dari rancang bangun dispenser penjual beras eceran otomatis berbasis Arduino UNO R3, menggunakan sensor *load cell*.
3. Mengembangkan mekanisme pengendalian dispenser, agar dapat mengeluarkan beras sesuai harga input yang diminta, menggunakan sensor *load cell*.

1.4. Manfaat Tugas Akhir

Adapun manfaat Tugas Akhir, antara lain sebagai berikut:

1. Mampu merancang sebuah sistem dispenser penjual beras eceran otomatis berbasis Arduino UNO R3.
2. Memudahkan penjual dalam proses transaksi penjualan beras, agar dapat berjalan lebih efektif dan efisien.
3. Menjadi bahan informasi dan referensi, bagi mahasiswa Teknologi Rekayasa Otomasi Universitas Diponegoro.

1.5. Pembatasan Masalah

Pada pembuatan Tugas Akhir ini ada beberapa pembatasan yang ditetapkan.

Batasan-batasan tersebut sebagai berikut :

1. Perangkat keras yang digunakan :
 - a. Motor servo M996R sebagai pembuka dan penutup saluran beras.
 - b. LCD 20x4 dengan modul I2C dan *Keypad* 4x4 sebagai *interface* antara perangkat elektronik dengan manusia.
 - c. Sensor *load cell* sebagai pengukur berat beras yang ditimbang.
 - d. Mikrokontroler Arduino Uno R3 sebagai pengendali motor serta pengolah data dari sensor.
 - e. Modul HX711 sebagai penguat sinyal keluaran dari sensor dan konverter analog ke digital untuk sensor *load cell*.
2. Perangkat lunak yang digunakan adalah Arduino IDE.
3. Output beras eceran maksimal 1kg dan minimal pembelian yaitu Rp.2000.
4. Kapasitas penyimpanan beras yaitu 3x5kg, sehingga total penyimpanan beras adalah 15kg.

1.6. Sistematika Tugas Akhir

Penyusunan laporan Tugas Akhir dibuat dengan sistematika sebagai berikut:

HALAMAN JUDUL

HALAMAN PENGESAHAN

BERITA ACARA UJIAN TUGAS AKHIR

HALAMAN PERSEMBAHAN

KATA PENGANTAR

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

ABSTRAK

ABSTRACT

BAB I. PENDAHULUAN

Bagian ini membahas tentang hal-hal yang melatarbelakangi pembuatan Tugas Akhir, perumusan masalah, tujuan Tugas Akhir, manfaat Tugas Akhir, pembatasan masalah, dan sistematika Tugas Akhir.

BAB II. DASAR TEORI

Bagian ini memuat mengenai teori-teori yang berkaitan dengan pembuatan sistem kontrol dispenser penjual beras eceran otomatis.

BAB III. METODE

Bagian ini membahas rancang bangun alat yang terdiri dari blok diagram, gambar 3D, spesifikasi, fitur, dan teknik fabrikasi.

BAB IV. PENGUJIAN DAN ANALISA

Bagian ini berisi hasil pengujian dan analisa dari sistem kontrol dispenser penjual beras eceran otomatis yang telah dirancang sebelumnya.

BAB V. PENUTUP

Bagian ini menjelaskan tentang kesimpulan yang diambil dari hasil yang telah didapat beserta saran untuk penyusunan selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN-LAMPIRAN

BAB II

DASAR TEORI

2.1. Kajian Pustaka

Dalam pembuatan Tugas Akhir ini dilakukan pembelajaran terhadap penelitian terdahulu. Penelitian yang berkaitan dengan ilmu otomasi yang diterapkan pada bidang makanan khususnya beras antara lain yaitu :

1. *“Perancangan dan Pembuatan Alat Penakar Beras Otomatis Berbasis Logika Fuzzy”*. [2] Membahas tentang pembuatan alat penakar beras otomatis berdasarkan berat disertai penjaga kelembaban menggunakan dua buah sensor dan dianalisa menggunakan metode fuzzy.
2. *“Rancang Bangun ATM Beras Raskin Dengan RFID E-KTP Menggunakan Teknik Counter Berbasis Arduino Uno”*. [3] Membahas mengenai teknik counter pada mesin ATM beras raskin yang ditujukan untuk bantuan beras terhadap warga kurang mampu yang sudah terdaftar oleh pemerintah menggunakan satu buah sensor dan RFID untuk deteksi ID card. Hanya terdiri dari satu jenis beras.
3. *“Rancang Bangun Mesin ATM Beras Murah Berkapasitas 100 Liter”*. [4] Membahas mengenai mesin ATM beras untuk kaum dhuafa dan yatim yang disalurkan melalui zakat. Mesin menggunakan satu buah sensor dan RFID. Hanya terdiri dari satu jenis beras.

Berdasarkan referensi yang ada, masih terdapat kekurangan pada sistem yang telah dibangun. Masih perlu penyempurnaan dalam hal mekanik dan kontrol agar alat dapat berfungsi lebih baik, lebih akurat dan efisien. Selain itu, pada penelitian sebelumnya input yang dimasukan hanya berupa berat, belum adanya input nominal harga dari beras, dan beras yang dipakai pada penelitian sebelumnya hanya terdiri dari satu jenis beras. Oleh karena itu, penyusun merancang dan membuat alat Tugas Akhir dengan judul **“RANCANG BANGUN DISPENSER PENJUAL BERAS ECERAN OTOMATIS BERBASIS ARDUINO UNO R3 MENGGUNAKAN SENSOR LOAD CELL”**.

2.2. Sistem Persamaan Linear Dua Variabel

Sistem Persamaan Linear Dua Variabel (SPLDV) merupakan salah satu materi matematika yang menyajikan masalah sesuai situasi yang ada (contextual problem), yaitu permasalahan sederhana yang berkaitan dengan kehidupan sehari-hari. Melalui soal cerita yang mengangkat permasalahan sehari-hari ini, siswa dituntut untuk mengomunikasikan bahasa sehari-hari ke dalam bahasa matematika dan menafsirkan hasil perhitungan yang dilakukan sesuai permasalahan yang diberi untuk memperoleh suatu pemecahan. Materi SPLDV merupakan salah satu pokok bahasan pelajaran matematika yang membahas tentang hubungan variabel satu dengan variabel lainnya. Dalam kehidupan sehari-hari banyak masalah perhitungan yang dapat diselesaikan dengan menerapkan SPLDV, diantaranya masalah uang, bisnis, umur, dan sebagainya.[5]

SPLDV adalah suatu persamaan yang memiliki dua variabel dan pangkat masing-masing variabelnya adalah 1, dengan $a, b \neq 0$. Contohnya $2x + 4y = 5$. Jika dua variabel tersebut adalah x dan y , maka bentuk umum SPLDV adalah :

$$ax + by = c \quad (2.1)$$

Rumus SPLDV dapat digunakan sebagai konversi harga beras ke berat beras, Konversi didasarkan pada rumus sistem linier dua variabel berikut :

$$ax - by = 0 \quad (2.2)$$

Di mana : x = Output Berat Beras

y = Berat beras

a = Harga Beras

b = Input Harga Beli

2.3. Catu Daya (*Power Supply 12V*)

Power Supply 12V adalah perangkat keras yang dapat memasok listrik atau mengarahkan tegangan listrik dari satu sumber tegangan PLN ke tegangan listrik lainnya. Catu daya menerima tegangan input yaitu arus bolak-balik (AC) dan mengubahnya menjadi arus searah (DC). Perangkat Catu Daya berfungsi untuk memasok perangkat elektronik yang membutuhkan arus searah. Komponen umum dicatu daya adalah transformator, penyearah dan filter.[6]



Gambar 2. 1. *Power Supply 12V*

(Sumber : <https://elektrologi.iptek.web.id>)

Switch-Mode Power Supply (SMPS) adalah jenis catu daya yang langsung menyearahkan (*rectify*) dan menyaring (*filter*) tegangan Input AC untuk mendapatkan tegangan DC. Tegangan DC tersebut kemudian di-switch ON dan OFF pada frekuensi tinggi dengan sirkuit frekuensi tinggi sehingga menghasilkan arus AC yang dapat melewati transformator frekuensi tinggi. *Switching Power Supply* atau lebih dikenal dengan sebutan *Switched - Mode Power Supply* (SMPS) merupakan catu elektronik yang terdiri dari regulasi switching yang disediakan sesuai kebutuhan tegangan keluaran. Sebuah SMPS adalah daya pengubah yang meneruskan daya dari sebuah sumber untuk beban yang ideal. Fungsi dari pengubah adalah untuk menyediakan tegangan keluaran pada level yang berbeda dibandingkan tegangan masukan.[6]

2.4. Modul LM2596

Modul LM2596 merupakan modul yang berfungsi untuk menurunkan tegangan DC tinggi menjadi tegangan DC rendah. Modul LM2596 memiliki *IC buck converter* yang bekerja hingga arus 2A dan memiliki nilai efisiensi hingga 92%. Keluaran dari modul ini dapat diatur dengan tegangan berkisar antara 1.5V – 35 VDC dan input masukan memiliki tegangan berkisar antara 4.5V – 40 VDC.[7]



Gambar 2. 2. Modul LM2596

(Sumber : <https://elektrologi.iptek.web.id>)

2.5. Arduino Uno R3

Arduino merupakan platform hardware *open-source* yang ditujukan kepada siapa saja yang ingin membuat rancang bangun peralatan elektronik interaktif berdasarkan hardware dan software yang fleksibel dan mudah digunakan. Mikrokontroler diprogram menggunakan bahasa pemrograman Arduino yang mempunyai kemiripan *syntax* dengan bahasa pemrograman C. Karena sifatnya yang terbuka maka siapa saja dapat mengunduh skema hardware Arduino dan membangunnya.[8]

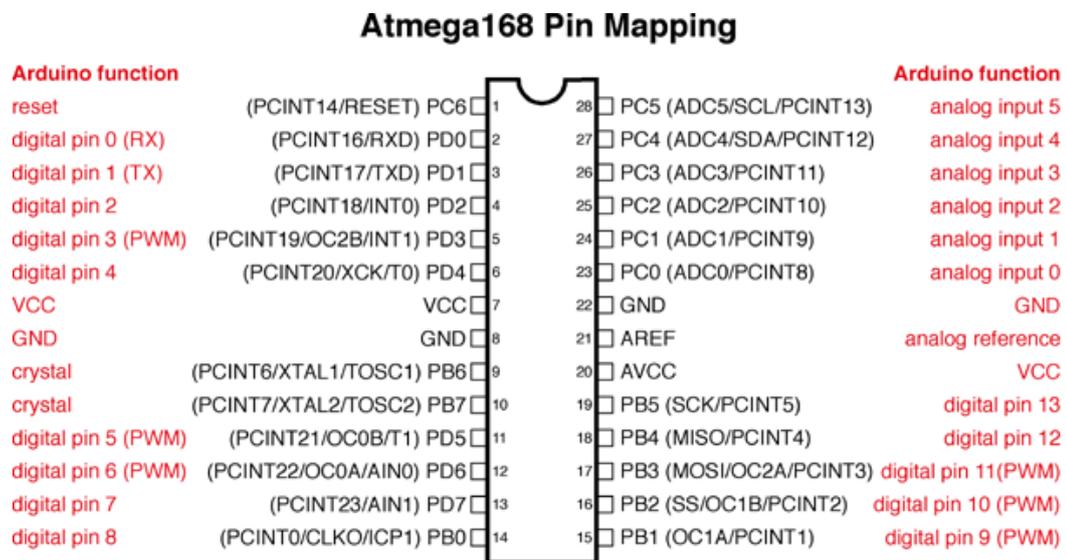
Arduino UNO adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328 (*datasheet*). Arduino UNO memiliki 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.[8]

Setiap 14 pin digital pada Arduino Uno dapat digunakan sebagai input dan output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. [8]

Fungsi -fungsi 14 pin digital Arduino Uno beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima suatu arus maksimum 40 mA dan mempunyai sebuah resistor pull-up (terputus secara default) 20-50 kOhm. [8]

Tabel 2. 1. Spesifikasi Arduino Uno R3

Mikrokontroler	Atmega328
Tegangan Pengoperasian	5V
Tegangan Input yang disarankan	7-12V
Batas Tegangan Input	6-20V
Jumlah pin I/O	14 Pin digital (6 diantaranya PWM)
Jumlah pin input Analog	6 pin
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3,3V	50mA
Memori Flash	32KB (Atmega 328) sekitar 0,5 KB digunakan boot leadet
SRAM	2KB (Atmega 328)
EPROM	1KB (Atmega 328)
Clock Speed	16 MHz



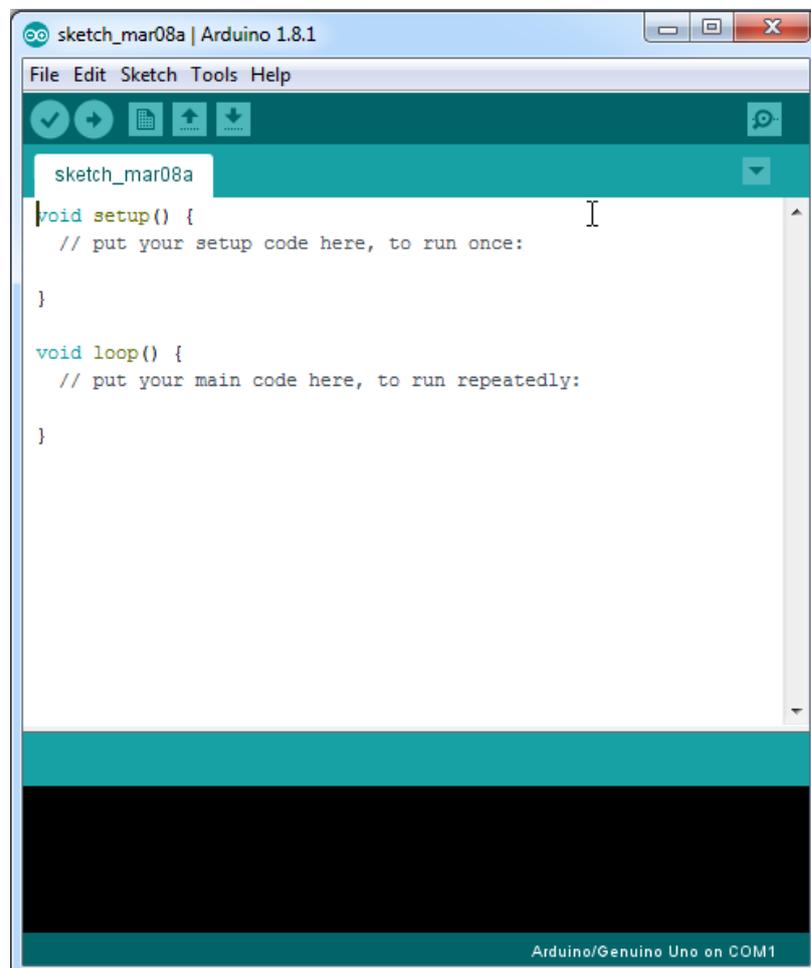
Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Gambar 2. 3. Pinout Atmega168

(Sumber : <https://www.arduino.cc>)

2.6. Software Arduino IDE

IDE (Integrated Development Environment) adalah lingkungan terintegrasi untuk pengembang Arduino menggunakan bahasa pemrogramannya sendiri dan sudah melakukan perubahan pada bahasa pemrograman Arduino atau sketsa. Mikrokontroler IC pada Arduino telah mengimplementasikan suatu program yaitu Boardladder yang berperan sebagai perantara compiler Arduino dan mikrokontroler. Arduino IDE dilengkapi dengan library C/C++ yang disebut dengan Wiring dimana dengan mudah dapat memanipulasi input dan output. Arduino IDE menggunakan software pengolah untuk pengembang dan mengubahnya menjadi Arduino IDE untuk pemrograman Arduino.[9]

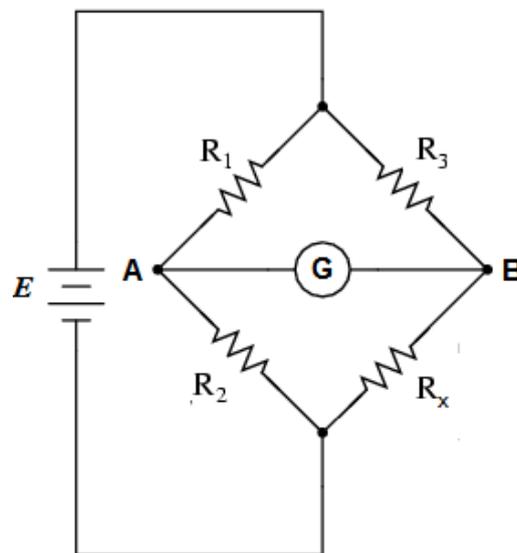


Gambar 2. 4. Tampilan Software Arduino IDE

(Sumber : <https://www.arduino.cc>)

2.7. Sensor Loadcell

Sensor *load cell* adalah sebuah transduser yang digunakan untuk mengubah gaya tekanan menjadi besaran elektrik. Meskipun banyak jenisnya, load cell yang didasarkan *strain gauge* adalah jenis yang paling banyak digunakan. Pada tahun 1843, fisikawan Inggris *Sir Charles Wheatstone* menemukan sebuah rangkaian penghubung yang dapat mengukur tahanan elektrik. Biasanya sensor *load cell* ini terdiri dari empat *strain gauge* dalam jembatan wheatstone, tetapi ada juga yang terdiri dari satu atau dua *strain gauge*. [10]



Gambar 2. 5. Rangkaian Jembatan Wheatstone

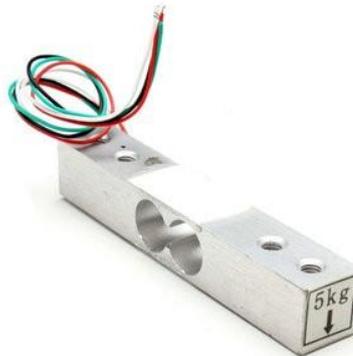
(Sumber : <https://elektronika-dasar.web.id/>)

Rangkaian penghubung wheatstone digunakan untuk mengukur perubahan tahanan yang terjadi pada strain gauge. Pada sensor loadcell dihubungkan dengan driver loadcell. *Load cell strain gauge* memiliki tingkat akurasi dari 0,03% sampai 0,25% skala penuh dan cocok digunakan hampir di semua aplikasi industri. Desain *load cell* dapat dibedakan menurut jenis sinyal keluarannya (pneumatik, hidrolik, listrik) atau sesuai dengan cara mendeteksi beratnya (lentur, geser, ketegangan, dll).[10]

Misal, nilai $R = 350 \text{ K}\Omega$, arus yang mengalir pada R_1 dan $R_3 =$ arus yang mengalir di R_2 dan R_4 , hal ini dikarenakan nilai semua resistor sama dan tidak

ada perbedaan tegangan antara titik 1 dan 2, oleh karena itu rangkaian ini dikatakan seimbang.

Jika rangkaian jembatan Wheatstone diberi beban, maka nilai R_1, R_2, R_3, R_4 pada rangkaian akan berubah. Sehingga membuat sensor *load cell* tidak dalam kondisi yang seimbang dan membuat beda potensial. Beda potensial inilah yang menjadi outputnya.



Gambar 2. 6. Sensor *Load Cell* 5kg

(sumber : <https://robotools.in>)

Secara teori, prinsip kerja *load cell* berdasarkan pada jembatan Wheatstone di mana saat *load cell* diberi beban terjadi perubahan pada nilai resistansi, nilai resistansi R_1 dan R_4 akan naik sedangkan nilai resistansi R_2 dan R_3 akan turun saat diberi beban.[11]

Tabel 2. 2. Spesifikasi Sensor *Load Cell* 5kg

Rated Load	5 Kg
Rated Load	1.0mV/V \pm 0.15mV/V
Zero Output	\pm 0.1mV/V
Creep	0.03%F.S./30min
Input End	Red+(power), Black-(power)
Output End	Green+(signal), White-(signal)

Tabel 2. 3. Lanjutan Spesifikasi Sensor Load Cell 5kg

Recommended operating voltage	3 ~ 12 VDC
Maximum operating voltage	15 VDC
Input Impedance	1115±10%
Output Impedance	1000±10% Û
Protection class	IP65

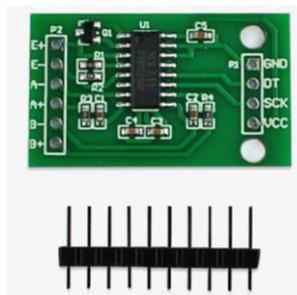
Persentase keberhasilan dan kesalahan pada sensor *load cell* dapat dihitung berdasarkan rumus berikut :

$$\frac{S-X}{S} \times 100\% \quad (2.3)$$

Di mana *S* adalah nilai hasil ukur *load cell* dan *X* adalah nilai range atau batasan kapasitas pengukuran.

2.8. Modul Penguat HX711

HX711 adalah komponen terintegrasi dari “AVIA SEMICONDUCTOR”, HX711 adalah konverter analog ke digital (ADC) 24-bit presisi yang dirancang untuk sensor penimbangan digital dan aplikasi kontrol industri yang terhubung ke jembatan sensor. HX711 adalah modul penimbangan, yang memiliki prinsip kerja mengubah perubahan terukur pada perubahan resistansi dan mengubahnya menjadi besaran tegangan melalui rangkaian yang ada. Modul berkomunikasi dengan komputer/mikrokontroler melalui TTL232.[12]



Gambar 2. 7. Modul Penguat HX711
(sumber : <https://robotools.in>)

Spesifikasinya adalah sebagai dibawah berikut :

1. Differential input voltage: $\pm 40\text{mV}$ (Full-scale differential input voltage $\pm 40\text{mV}$)
2. Data accuracy: 24 bit (24 bit A / D converter chip.)
3. Refresh frequency: 80 Hz
4. Operating Voltage : 5V DC
5. Operating current : $<10\text{ mA}$
6. Size : $38\text{mm} \times 21\text{mm} \times 10\text{mm}$.

Tingkat ketelitian sensor *load cell* menggunakan HX711 dapat dihitung berdasarkan rumus berikut :

$$\frac{\text{Operating Voltage}}{\text{Data Accuracy}} \times \text{Rated Load Load Cell} \quad (2.4)$$

Di mana : *Operating Voltage* = 5V DC
Data Accuracy = 24 bit
Rated Load Load cell = 5kg (5000gr)

2.9. Motor Servo



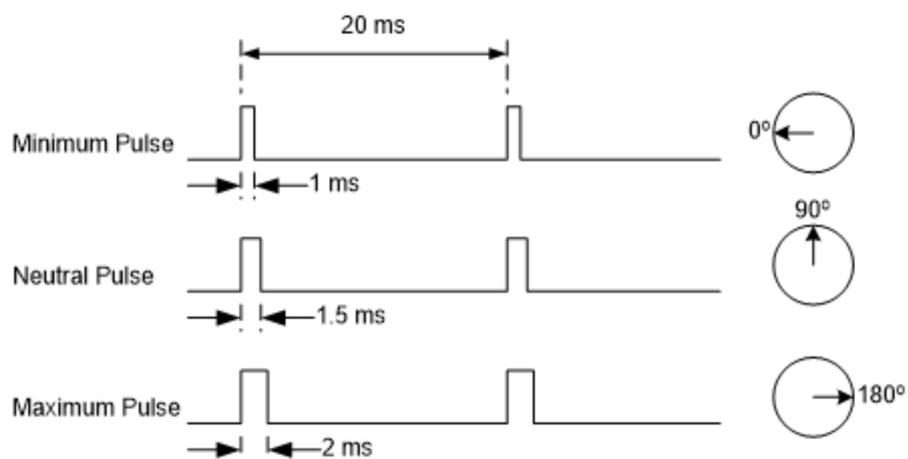
Gambar 2. 8. Motor Servo

(Sumber : <http://zoniaelektro.net>)

Motor servo adalah perangkat atau aktuator (motor) putar yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat diatur atau disesuaikan untuk menentukan dan memastikan posisi sudut poros output

motor. Motor servo adalah perangkat yang terdiri dari motor DC, rangkaian roda gigi, rangkaian kontrol dan potensiometer. Rangkaian roda gigi yang terpasang pada poros motor akan memperlambat putaran poros dan meningkatkan torsi servo, sedangkan potensiometer dengan resistansinya berubah saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo. Penggunaan sistem kontrol loop tertutup pada motor servo berguna untuk mengontrol pergerakan dan posisi akhir poros motor servo. Penjelasan sederhananya begini, posisi poros output akan disensor untuk menentukan apakah posisi poros sudah benar seperti yang diinginkan atau tidak, dan jika tidak, maka kontrol input akan mengirimkan sinyal kontrol agar posisi poros tepat pada posisi yang diinginkan.[13]

Motor servo dikendalikan dengan sinyal PWM dari encoder/potentiometer. Lebar sinyal (pulsa) yang diberikan inilah yang akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar sinyal dengan waktu 1,5 ms (mili second) akan memutar poros motor servo ke posisi sudut 90° . Bila sinyal lebih pendek dari 1,5 ms maka akan berputar ke arah posisi 0° atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila sinyal yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar ke arah posisi 180° atau ke kanan (searah jarum jam).[13]

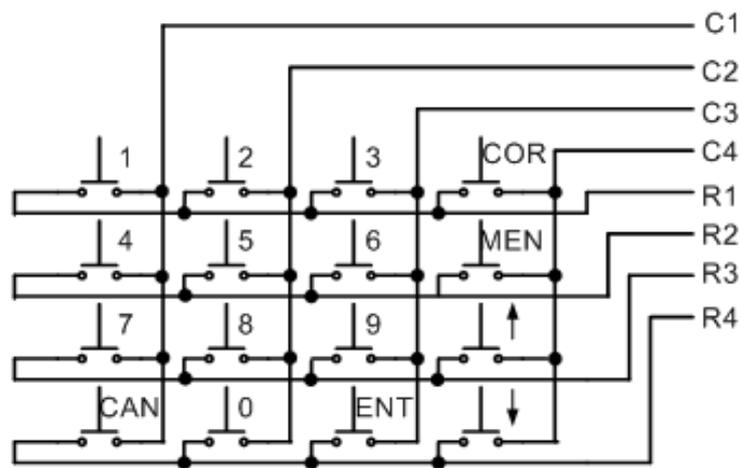


Gambar 2. 9. Pulsa Pengendalian Motor Servo Rotasi Sudut

(Sumber : <https://www.insinyoer.com>)

Ketika sinyal PWM telah diberikan, maka poros motor servo akan bergerak ke posisi yang telah ditargetkan dan berhenti pada posisi tersebut dan akan tetap pada posisi tersebut. Jika ada gaya luar yang mencoba untuk memutar atau mengubah posisi, maka sistem loop tertutup motor servo akan bekerja dengan mencoba menahan atau melawan gaya luar tersebut dengan tenaga dalam motor servo itu sendiri. Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal PWM harus diulang setiap 20 ms (mili second) agar posisi poros motor servo tetap bertahan pada posisinya.[13]

2.10. Keypad 4X4



Gambar 2. 10. Rangkaian Dasar Keypad 4x4

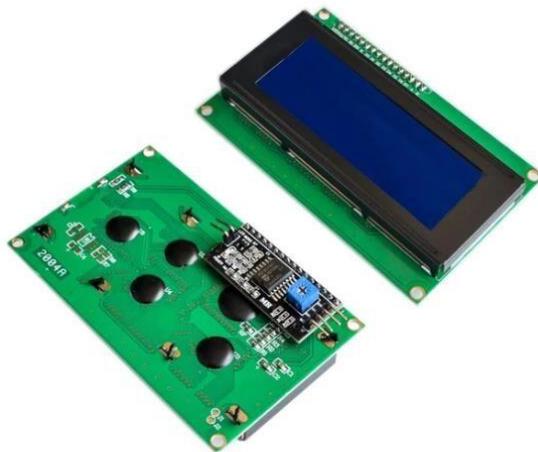
(Sumber : <http://www.mytutorialcafe.com>)

Keypad sering digunakan sebagai input ke beberapa perangkat berbasis mikroprosesor atau mikrokontroler. Keypad sebenarnya terdiri dari sejumlah saklar yang dihubungkan sebagai baris dan kolom dengan susunan seperti pada Gambar 2.10 agar mikrokontroler dapat melakukan pemindaian *Keypad*, port mengeluarkan salah satu dari 4 bit yang terhubung ke kolom dengan logika *low* "0" dan kemudian membaca 4 bit pada baris untuk mengetahui jika ada tombol yang ditekan pada kolom itu. Akibatnya, selama tidak ada tombol yang ditekan, mikrokontroler akan melihat logika *high* "1" pada setiap pin yang terhubung ke baris tersebut.[14]

2.11. LCD 20x4 dengan I2C

Untuk menampilkan menu dari suatu peralatan elektronika seperti catu daya switching misalnya tentang berapa tegangan keluaran yang diinginkan, maka diperlukan suatu tampilan. Yang lebih sering digunakan adalah LCD.

Liquid crystal display (LCD) merupakan komponen elektronika yang digunakan untuk menampilkan suatu karakter baik itu angka, huruf atau karakter tertentu, sehingga tampilan tersebut dapat dilihat secara visual, jenis, dan ukuran



Gambar 2. 11. LCD 20x4 dengan Modul I2C

LCD bermacam-macam, antara lain 16x1, 20x2, 20x4 dan lain-lain.[15]

(Sumber : <https://www.robotistan.com>)

Dalam modul LCD (Liquid Crystal Display) di dalamnya terdapat mikrokontroller yang berfungsi sebagai pengendali tampilan karakter yang ada di dalam LCD. Mikrokontroller pada display ini dilengkapi dengan memori dan register. Memori yang digunakan mikrokontroller internal LCD adalah :

- a. DDRAM (Display Data Random Access Memory) merupakan memori tempat menyimpan dan memproses karakter yang akan ditampilkan.

- b. CGRAM (Character Generator Random Access Memory) merupakan memori untuk menggambarkan pola sebuah karakter yang dibentuk dapat diubah sesuai keinginan.
- c. CGROM (Character Generator Read Only Memory) merupakan memori untuk menggambarkan pola sebuah karakter yang telah dirancang secara permanen oleh pabrikan pembuat LCD, sehingga user hanya tinggal mengambilnya saja sesuai alamat memorinya dan tidak dapat mengedit karakter dasar yang terdapat dalam memori CGROM tersebut.

Ada 2 jenis register yang digunakan pada LCD untuk melakukan tugas kontrolnya sebagai pembentuk karakter diantaranya :

- a. Register perintah yaitu register yang berisi perintah-perintah dari mikrokontroller ke LCD pada saat proses penulisan data.
- b. Register data yaitu register untuk menuliskan atau membaca data menuju DDRAM tentunya dengan alat yang telah diatur sebelumnya. Berikut adalah konfigurasi kaki-kaki LCD karakter 20 x 4 untuk menkoneksikannya ke board Arduino :

Tabel 2. 4. Konfigurasi Kaki-Kaki LCD 20x4

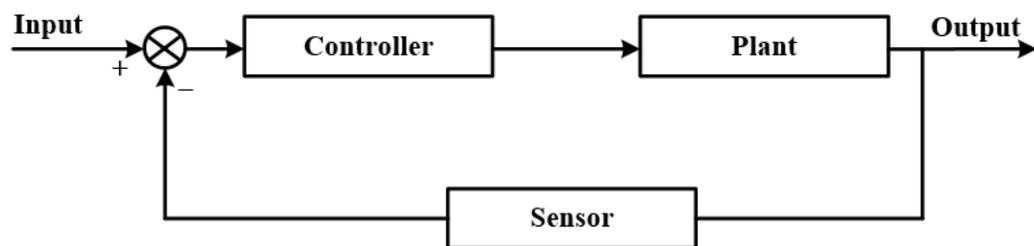
Pin No	<i>Symbol</i>	<i>Details</i>
1	GND	Ground
2	VCC	Supply Voltage +5V
3	Vo	Contrast Adjustment
4	RS	0 > Control Input, 1 > Data Input
5	R/W	Read / Write
6	E	Enable
7-14	D0 to D7	Data
15	VB1	Backlight +5V
16	VB0	Backlight Ground

LCD yang dipakai pada rancang bangun dispenser beras eceran otomatis berukuran 20x4 karakter dengan tambahan modul I2C untuk mempermudah dalam mengakses LCD tersebut. Sebab dengan modul I2C akan lebih

memperhemat penggunaan pin arduino yang akan digunakan, karena hanya diperlukan 4 buah pin arduino, yaitu pin SCL, pin SDA, pin VCC dan pin GND.

2.12. Sistem Kontrol Close Loop

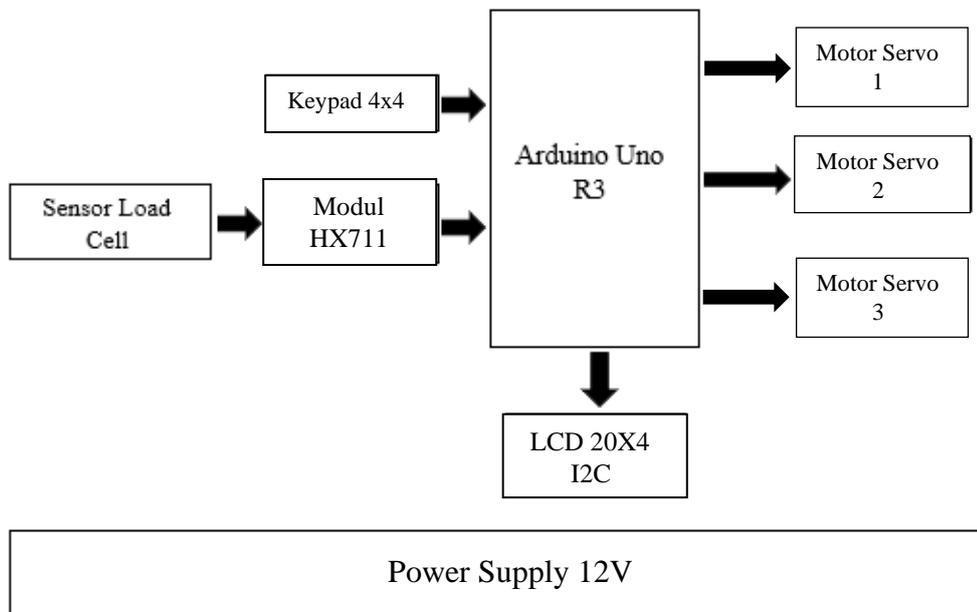
Pada sistem kontrol close loop keluaran yang dihasilkan dapat mempengaruhi masukan dengan menggunakan sensor sehingga kontroler dapat memperkecil kesalahan yang terjadi menggunakan perbandingan antara masukan dan keluaran yang menyebabkan keluaran menjadi sesuai dengan set point yang diberikan.



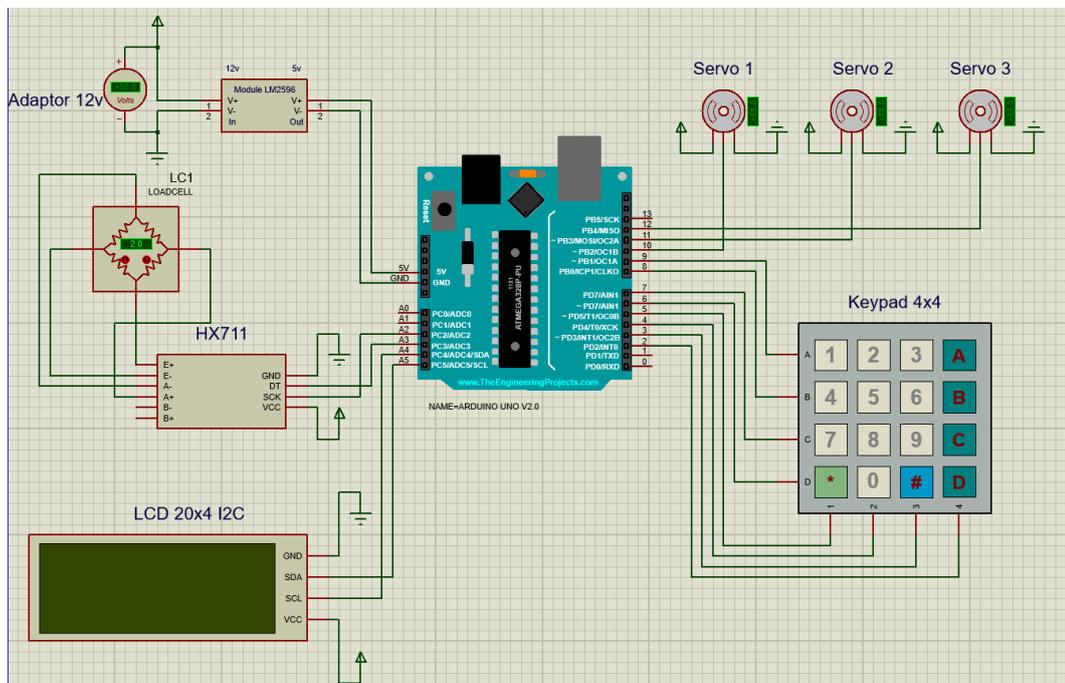
Gambar 2. 12. Sistem Kontrol *Close Loop*

BAB III METODE PENELITIAN

3.1. Blok Diagram



Gambar 3. 1. Blok Diagram Alat

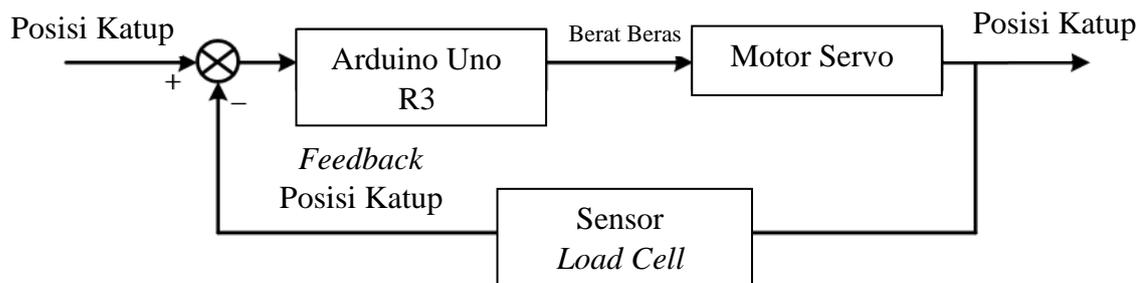


Gambar 3. 2. Rangkaian Alat

Bagian-bagian dari blok diagram tersebut meliputi:

1. Sumber tegangan *power supply* 12 VDC yang berasal dari listrik PLN yang digunakan sebagai catu daya ke Arduino Uno R3 dan motor servo.
2. Arduino Uno R3 bekerja sebagai tempat memproses semua data yang digunakan untuk menggerakkan motor, serta mengolah data dari sensor.
3. *Keypad* berfungsi sebagai *interface* antara perangkat elektronik dengan manusia serta untuk menginput data berupa angka dan huruf serta sebagai tombol navigasi untuk mengakses suatu perangkat.
4. Sensor *load cell* sebagai sensor berat, bekerja dengan cara mengkonversikan regangan pada logam ke tahanan variabel.
5. Motor servo digunakan untuk membuka dan menutup saluran beras.
6. LCD 20x4 I2C berfungsi sebagai *interface* atau penampil kata/data dari sistem.
7. Modul HX711 bekerja mengubah perubahan terukur pada perubahan resistansi dan mengubahnya menjadi besaran tegangan melalui rangkaian yang ada.

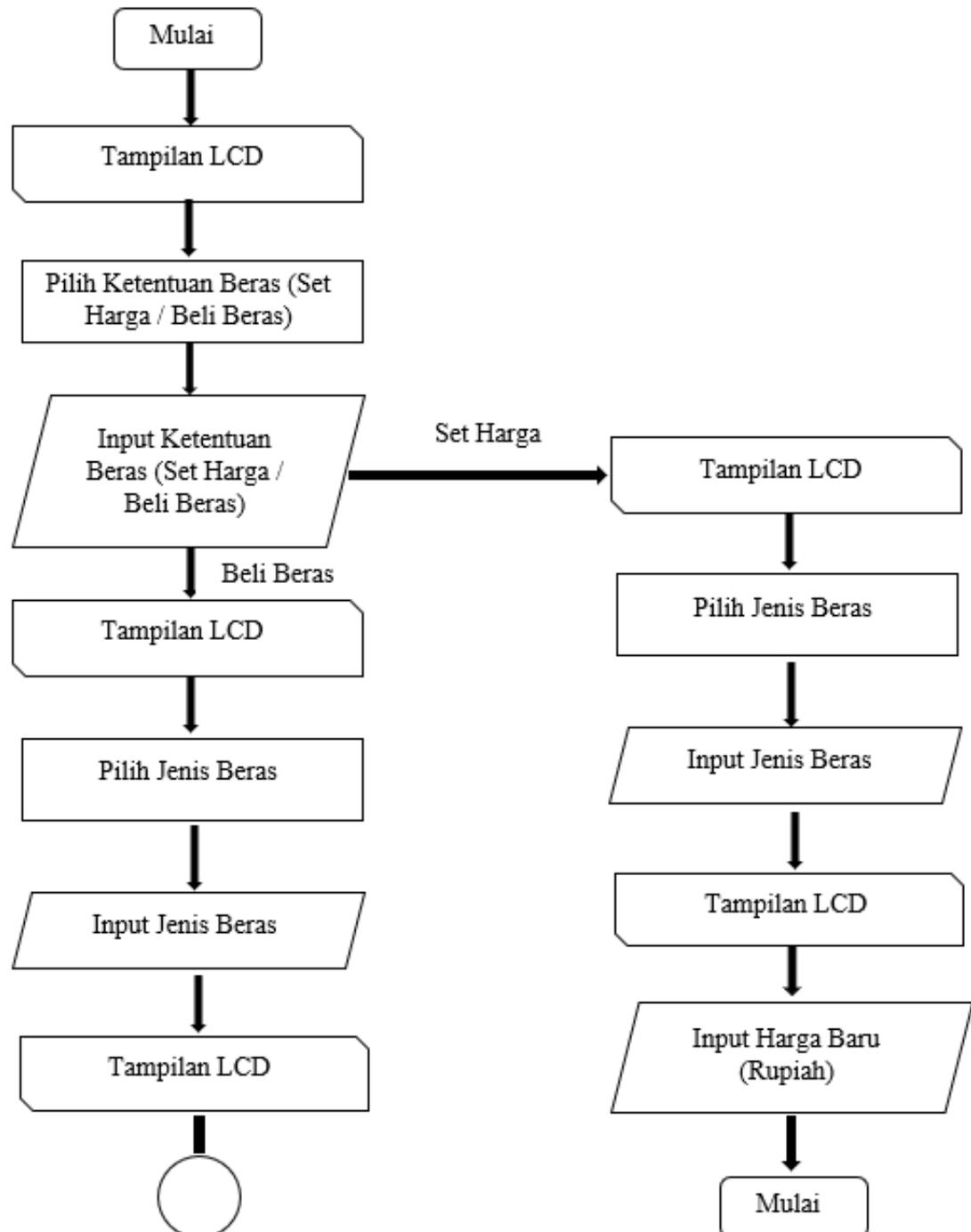
Sistem kontrol yang digunakan pada dispenser penjual beras eceran otomatis ini menggunakan sistem kontrol *close loop*, berikut adalah diagram blok sistem kontrol dari dispenser penjual beras eceran otomatis :

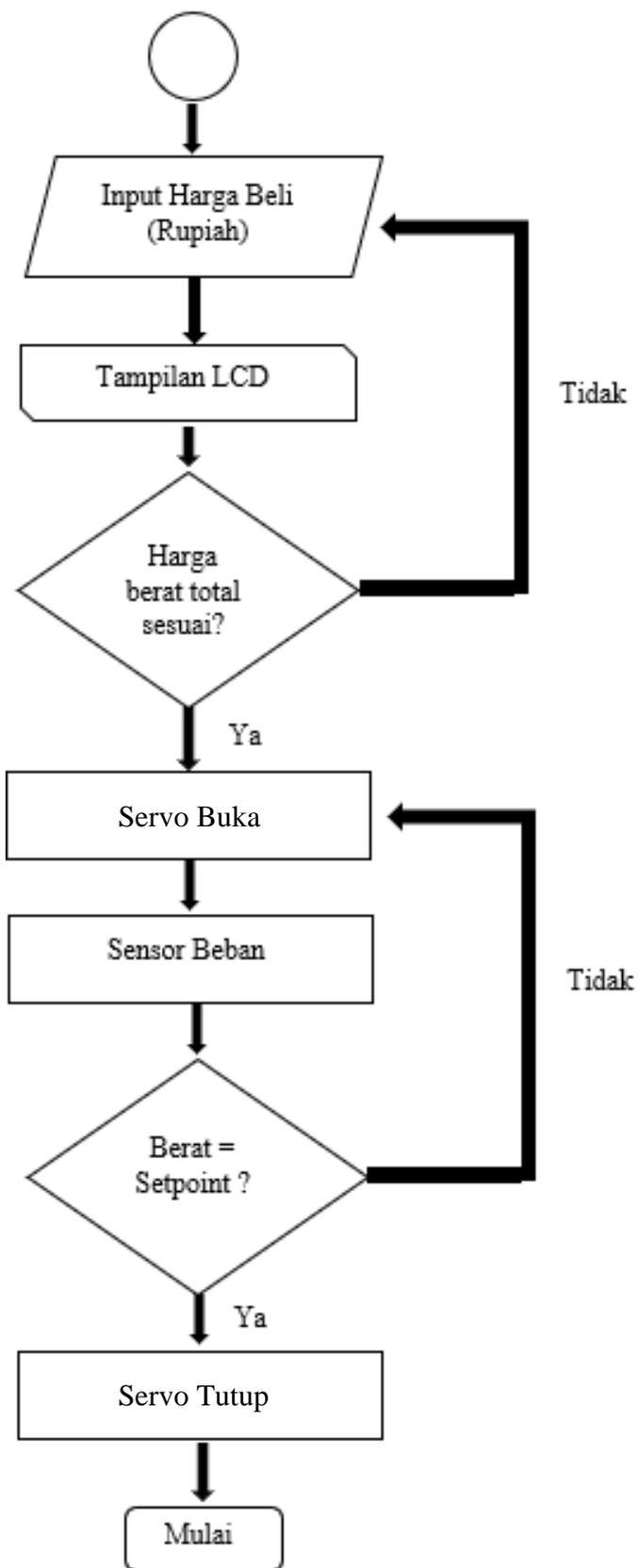


Gambar 3. 3. Diagram Blok Sistem Kontrol Dispenser Penjual Beras Eceran Otomatis

3.2. Flowchart Alat

Pada alat ini terdapat diagram alir untuk menentukan proses kerja alat. Diagram alir dijabarkan sebagai berikut :





Gambar 3. 4. Flowchart Alat

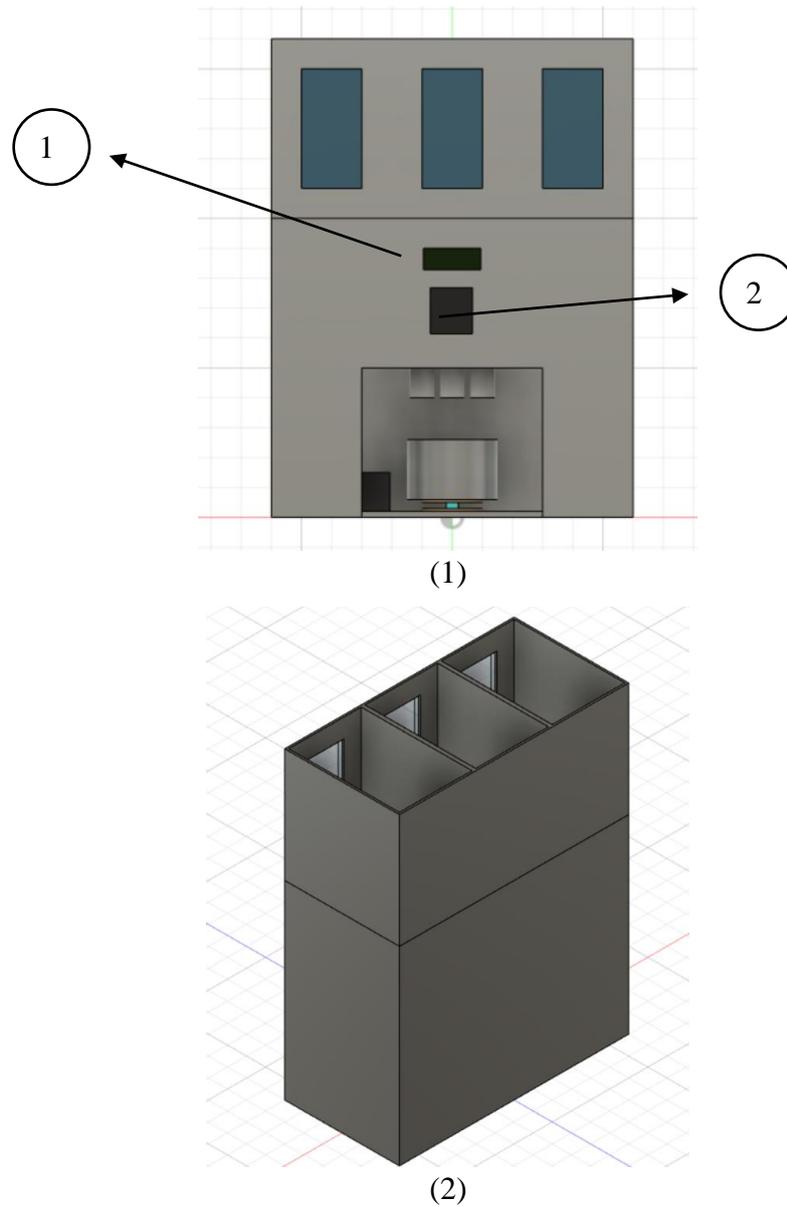
3.3. Cara Kerja

Cara kerja alat dimulai dari tampilan LCD awal berupa pilihan ketentuan beras yaitu set harga beras dan beli beras. Pilihan pertama adalah input set harga beras jika ingin merubah harga beras. Kemudian tampilan LCD akan berubah menjadi pilihan jenis beras. Input jenis beras yang harganya akan dirubah. Setelah itu tampilan LCD berubah, kemudian input harga beras baru dari jenis beras yang telah dipilih. Setelah proses perubahan harga dilakukan, tampilan LCD akan kembali ke awal.

Pilihan kedua adalah input beli beras jika ingin membeli beras. Kemudian tentukan jenis beras dan harga beli beras. Lalu data diolah mikrokontroler dan dieksekusi oleh servo dan sensor *loadcell*. Pada beras 1 dan beras 2, servo akan bergerak dari sudut awal (90°) ke sudut 180° . Pada beras 3, servo akan bergerak dari sudut awal (90°) ke sudut 0° . Hal ini terjadi karena perbedaan pemasangan mekanik pada posisi motor servo. Ketika nilai *loadcell* berada pada rentang $\frac{1}{2}$ dari *setpoint*, maka servo akan bergerak menutup sebanyak 30° ke arah sudut awal servo (90°). Ketika nilai *loadcell* berada pada rentang $\frac{3}{4}$ dari *setpoint*, maka servo akan bergerak menutup sebanyak 60° ke arah sudut awal servo (90°). Ketika *loadcell* sudah sama dengan *setpoint* maka servo memberikan respon berupa penutupan lubang. Jika nilai *loadcell* tidak sama dengan *setpoint* maka servo masih terbuka. Setelah proses penimbangan selesai tampilan LCD akan kembali ke awal.

Ketika terjadi kesalahan dalam input harga dan motor servo sudah dalam posisi terbuka, tekan tombol “*” pada keypad untuk set ulang tampilan kembali ke awal menu dan set ulang sistem, maka otomatis servo akan kembali tertutup.

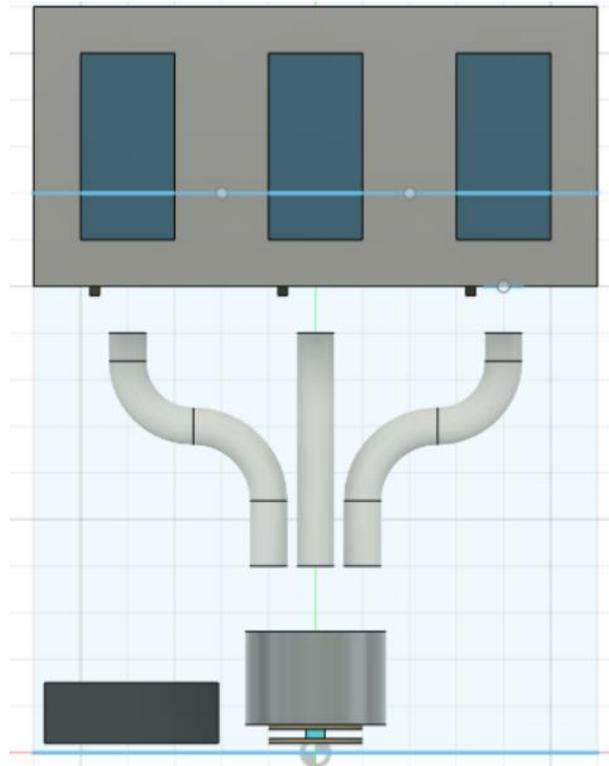
3.4. Gambar 3D



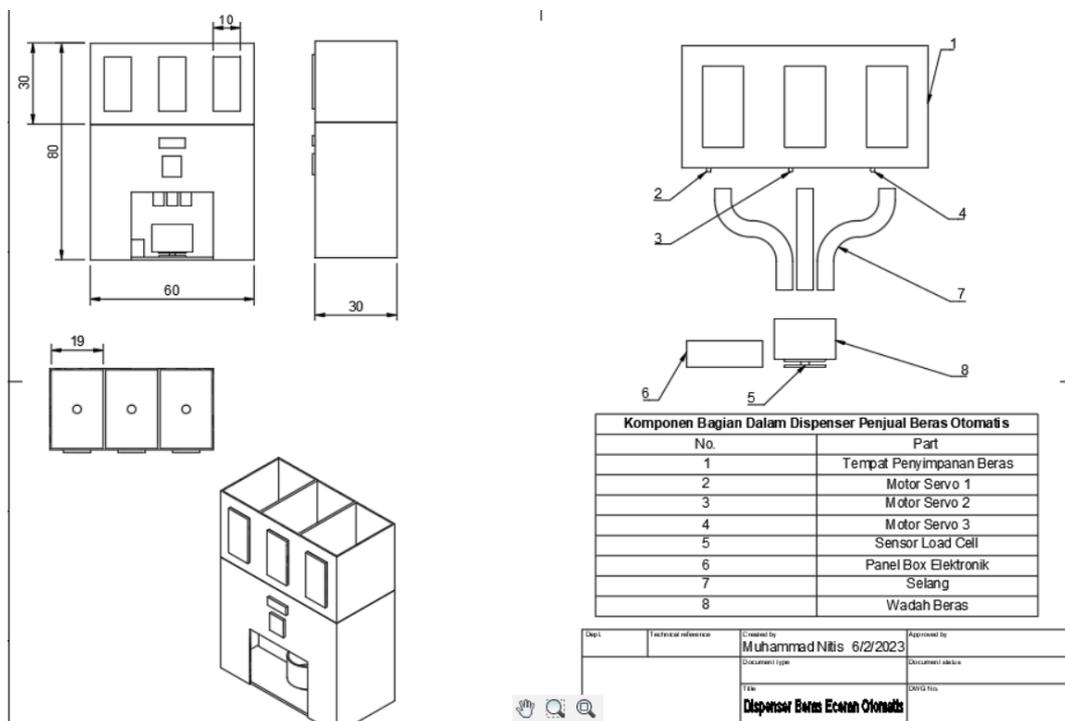
Gambar 3. 5. (1) Tampak Depan Bagian Luar (2) Tampak Belakang Bagian Luar

Keterangan Dispenser Bagian Luar :

1. LCD 20x4 dengan I2C
2. Keypad 4x4



Gambar 3. 6. Tampak Depan Bagian Dalam



Gambar 3. 7. Detil Gambar

3.5. Spesifikasi dan Fitur

Spesifikasi dispenser penjual beras eceran otomatis ini adalah sebagai berikut:

1. Dimensi dispenser 80cm x 60cm x 30cm.
2. Bahan kerangka dispenser terbuat dari besi hollow ukuran 4cm x 4cm dengan ketebalan 3mm.
3. Bahan cover dispenser menggunakan plat ACP.
4. Menggunakan tegangan 220VAC yang diturunkan menjadi 12VDC 10A menggunakan *power supply* sebagai suplai daya pada komponen elektrik seperti Arduino.
5. Menggunakan tegangan 5VDC untuk sensor *load cell* dan motor servo.
6. Beban maksimal *load cell* yaitu 5kg.
7. Jumlah tandon / penyimpanan beras yaitu 3 tandon dengan kapasitas penyimpanan beras 5kg.

Tabel 3. 1. Spesifikasi dan Fitur Dispenser Beras Eceran Otomatis

Spesifikasi dan Fitur Dispenser Penjual Beras Eceran Otomatis		
No.	Fitur	Keterangan
1.	Dimensi	80x60x30 cm
2.	Bahan Kerangka dispenser	Besi Hollow 4x4 3mm
3.	Bahan Dispenser	ACP
4.	Jumlah Tandon/Penyimpanan Beras	3
5.	Penggerak	Motor Servo
6.	Daya	12 VDC

3.6. Cara Pembuatan/Teknik Fabrikasi

Dalam proses pembuatan dispenser penjual beras eceran otomatis dibutuhkan konsep pembuatan dalam pengerjaannya. Konsep tersebut terbagi menjadi beberapa bagian, antara lain :

3.6.1. Penyusunan Perangkat Keras

a. Dispenser

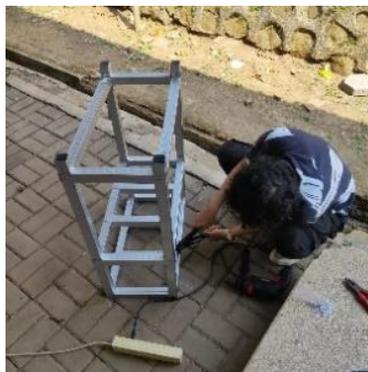
Tahap pertama dalam pembuatan dispenser adalah pembuatan kerangka dispenser dengan memotong besi hollow sesuai dengan ukuran yang telah ditentukan pada saat proses desain dengan menggunakan gerinda, kemudian dilanjutkan dengan pengeboran besi hollow tersebut dengan tujuan untuk melubangi besi sehingga dapat dirangkai menggunakan paku rivet agar potongan-potongan besi hollow tersebut dapat menjadi kerangka dispenser.



(a)



(b)



(c)

Gambar 3. 8. (a) Pemotongan Besi, (b) Pengeboran Besi, (c) Pemasangan Paku Rivet

Tahap kedua yaitu pembuatan cover dispenser yaitu dengan menutup kerangka dispenser memakai plat berbahan ACP menggunakan bor dan paku rivet.



Gambar 3. 9. Pemasangan Cover ACP Menggunakan Paku Rivet

Tahap ketiga yaitu pemasangan *bracket* pada motor servo sesuai desain 3D yang telah dibuat. Pemasangan ini bertujuan agar motor servo dan selang dapat menempel dengan kuat pada rangka dispenser.



Gambar 3. 10. Pemasangan Bracket Motor Servo

Tahap keempat yaitu pemasangan selang pada dispenser sebagai jalur jatuhnya beras ke wadah beras.



Gambar 3. 11. Pemasangan Selang

b. Elektronika Dispenser



Gambar 3. 12. Elektronika Dispenser

Tahap pertama yang harus dilakukan adalah memasang adaptor. Pemasangan adaptor bertujuan untuk mengubah tegangan 220VAC yang diturunkan oleh *power supply* 12VDC kemudian dihubungkan ke modul step down LM2596 untuk menurunkan tegangan menjadi 5V yang digunakan sebagai tegangan mikrokontroler Arduino UNO R3. Pin input modul step-down LM2596 terdiri dari pin + dan -. Pin + dihubungkan ke kabel tegangan 12V dan pin - dihubungkan ke kabel ground. Sedangkan pin keluaran modul LM2596 antara lain adalah pin + yang dihubungkan dengan pin 5V dan pin - yang dihubungkan dengan pin GND.



Gambar 3. 13. Pemasangan *Power Supply* 12VDC

Tahap kedua yaitu pemasangan komponen lain seperti LCD 20x4 dengan modul I2C, *keypad*, motor servo, dan sensor *load cell* pada Arduino Uno R3. Masing-masing komponen dihubungkan kepada pin yang telah ditentukan skematik rangkaian alat sesuai gambar 3.2.

LCD 20x4 dengan modul I2C memiliki 4 pin, pin SDA dihubungkan dengan pin A5, pin SCL dihubungkan dengan pin A4, pin VCC dihubungkan dengan pin 5V, dan pin GND dihubungkan dengan pin GND.



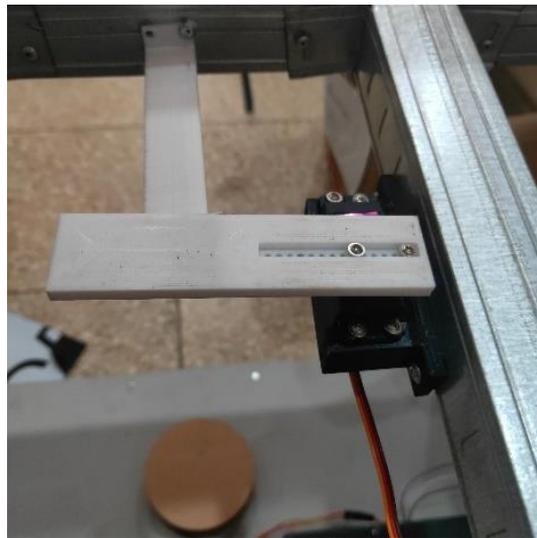
Gambar 3. 14. Pemasangan LCD 16x4 I2C

Keypad 4x4 memiliki 8 pin, pin pada keypad dihubungkan dengan pin 2,3,4,5,6,7,8,9. Keypad berfungsi sebagai input pada LCD.



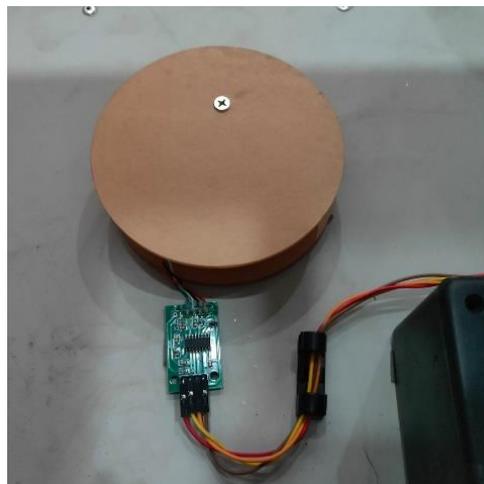
Gambar 3. 15. Pemasangan Keypad 4x4

Motor Servo yang digunakan berjumlah 3. Masing-masing motor servo mempunyai 3 pin. Pin VCC dihubungkan dengan pin 5V. pin GND dihubungkan dengan pin GND. Pin signal pada motor servo 1 dihubungkan dengan pin GND. Pin signal pada motor servo 2 dihubungkan dengan pin 10. Pin signal pada motor servo 3 dihubungkan dengan pin 12.



Gambar 3. 16. Pemasangan Motor Servo

Sensor *load cell* membutuhkan modul HX711 yang berfungsi untuk memudahkan pembacaan nilai input dalam pengukuran berat. Modul HX711 memiliki empat pin yaitu pin DT masuk pada pin A3. Pin SCK masuk pada pin A2. Pin GND masuk pada pin GND. Dan pin VCC masuk pada pin 5V.



Gambar 3. 17. Pemasangan Sensor *Load Cell* dan Modul HX711

c. Peletakan Toples

Peletakan toples yaitu penyimpanan toples diatas sensor *load cell*. Toples berkapasitas maksimal 1kg dengan ukuran diameter 14cm dan tinggi 13 cm.



Gambar 3. 18. Peletakan Toples

3.6.2 Pembuatan Perangkat Lunak

Perangkat lunak yang digunakan pada rancang bangun dispenser penjual beras eceran otomatis ialah platform Arduino IDE. Berikut langkah-langkah pembuatan perangkat lunak sebagai berikut:

1. Penetapan pin dan variabel program.

Penetapan pin dan variabel program ini bertujuan untuk memudahkan dalam pemasangan pin di Arduino UNO R3 dan membuat variabel dengan tipe data tertentu sekaligus mengisi variabel dengan nilai awal pada saat mikrokontroler diaktifkan.

```
#define pin_DT      (A3)
#define pin_SCK     (A2)
#define kBerat      (493)

#define pin_ServoA (10)
```

```

#define pin_ServoB (11)
#define pin_ServoC (12)

#define _sudut      (90)
#define _sudut1    (0)
#define _sudut2    (30)
#define _sudut3    (60)
#define _sudut1ab  (180)
#define _sudut2ab  (150)
#define _sudut3ab  (120)

//*****//

#define I2C_ADDR    (0x27)
#define LCD_COLUMNS (20)
#define LCD_LINES   (4)
//*****//

const uint8_t Bar = 4;
const uint8_t Col = 4;
uint8_t colPin[Col] = { 9, 8, 7, 6 };
uint8_t barPin[Bar] = { 5, 4, 3, 2 };
//*****//

#define ENABLE_USER_DEBUG (1)

#define time_task_user (20)
#define time_task_debug (1000)
#include "Scale_Beras.h"

#include <EEPROM.h>
#define alamatBerasA (10)
#define alamatBerasB (20)
#define alamatBerasC (30)

```

```

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLUMNS,
LCD_LINES);

#include "LCD_DATA.h"
String array_beras[] = {"", S1_2a, S1_3a, S1_4a};
#include <Servo.h>

Servo ServoA;
Servo ServoB;
Servo ServoC;

#define buka_ServoA(a) (ServoA.write(a))
#define buka_ServoB(a) (ServoB.write(a))
#define buka_ServoC(a) (ServoC.write(a))
//*****
*****//

#include <Keypad.h>

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

Keypad keypad = Keypad(makeKeymap(keys), barPin,
colPin, Bar, Col);
char key;
long dKey = 0, buf_dKey = 0, oldKey = 0;
String sdKey = "";
#include "HX711.h"

```

```
HX711 timbangan;
//SATUAN

#define Satuan (" gr ")

#define Rupiah ("Rp ")

//LOADING SCREEN

#define Header1 (" DISPENSER BERAS ")
#define Header2 (" UNDIP ")
#define Header3 (" SELAMAT DATANG ! ")
#define Header4 (">> TEKAN APA SAJA <<")

//SCREEN 1

#define S1_1 (" ")
#define S1_2 ("1. SET HARGA ")
#define S1_3 ("2. BELI BERAS ")
#define S1_4 (" ")

//SCREEN SET A

#define S1_1a (" HARGA BERAS ")
#define S1_2a ("BERAS 1 ")
#define S1_3a ("BERAS 2 ")
#define S1_4a ("BERAS 3 ")

//SCREEN SET B

#define S1_1b (" ")
#define S1_2b ("HARGA : ")
```

```

#define S1_3b ("HARGA BARU : ")
#define S1_4b ("TEKAN 'D' UNTUK SAVE")

//SCREEN 2
#define S2_1 (" PILIH JENIS BERAS ")

//SCREEN 3 //max 4 digit
#define S3_1 ("MAU BELI BERAPA? ")

//SCREEN 4
#define S4_1 ("OKE, TUNGGU YAA... ")

//SCREEN 5
#define S5_1 ("BERAT IN = ")
#define S5_2 ("BERAT OUT = ")

```

2. Pembuatan program void setup

```

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

    Dispenser_init();
    // simpan_EEPROM(alamatBerasA, 10000);
    // simpan_EEPROM(alamatBerasB, 12000);
    // simpan_EEPROM(alamatBerasC, 13000);
}

```

```

uint32_t CntTime_TaskDebug = 0;
uint32_t CntTime_TaskUser = 0;

```

Program void setup terdapat program yang mendefinisikan fungsi setup yang akan dieksekusi sekali saat Arduino pertama kali dinyalakan, dimulai

dengan menjalankan protokol komunikasi serial sebagai komunikasi antara mikrokontroler dengan perangkat lunak Arduino IDE yang berada di komputer dengan kecepatan 115200 bps (bits per second). Kemudian dilakukan inisialisasi pada sistem dispenser yang bertujuan untuk menyimpan data pada alamat EEPROM tertentu dengan nilai yang ditentukan.

Program ini merupakan implementasi dasar dari task scheduling menggunakan waktu dalam program Arduino. Dalam loop utama, fungsi `task_user()` dan `task_debug()` akan dieksekusi dengan interval waktu tertentu, seperti yang ditentukan oleh `time_task_user` dan `time_task_debug`.

3. Pembuatan program void loop

Program void loop merupakan program utama yang dijalankan Arduino setelah program pada void setup selesai. Program void loop akan terus dijalankan berulang kali sampai Arduino dimatikan.

```
void loop() {
    uint32_t tempTime = millis();

    tempTime = millis();
    if (tempTime - CntTime_TaskUser > time_task_user) {
        CntTime_TaskUser = tempTime;
        task_user();
    }

    #if (ENABLE_USER_DEBUG)
        tempTime = millis();
        if (tempTime - CntTime_TaskDebug > time_task_debug)
        {
            CntTime_TaskDebug = tempTime;
            task_debug();
        }
    #endif
}
```

4. Pembuatan program void task_user dan void task_debug

Program ini bertujuan untuk mendefinisikan fungsi ‘task_user’ dan ‘task_debug’ yang akan digunakan dalam loop utama program.

```
void task_user(void) {
    baca_Keypad();
    Dispenser_control();
}

void task_debug(void) {
    Serial.print("key : ");
    Serial.print(key);
    Serial.print("\tberat input : ");
    Serial.print(berat_input);
    Serial.print("\tberat output : ");
    Serial.println(berat);
}
```

Program ‘task_user’ bertujuan untuk menangani tugas yang berkaitan dengan pengguna atau input dari pengguna. Dalam program ini berisi perintah untuk membaca input dari keypad atau tombol yang dikontrol oleh pengguna dan mengendalikan sistem dispenser oleh motor.

Program ‘task_debug’ bertujuan untuk menangani tugas *debugging* atau pencatatan informasi.

5. Pembuatan program untuk pembacaan dan penyimpanan EEPROM

EEPROM adalah memori yang tidak terhapus meski tanpa dialiri listrik. EEPROM memiliki alamat dan jumlah memori.

```
void simpan_EEPROM(int address, unsigned int number)
{
    EEPROM.write(address, number >> 8);
    EEPROM.write(address + 1, number & 0xFF);
}

unsigned int baca_EEPROM(int address)
{
```

```

    return (EEPROM.read(address) << 8) +
EEPROM.read(address + 1);
}

```

Fungsi ‘`simpan_EEPROM()`’ digunakan untuk menyimpan nilai number ke alamat *address* dalam EEPROM, sedangkan ‘`baca_EEPROM`’ digunakan untuk membaca nilai dari alamat *address* dan *address + 1* dalam EEPROM dan menggabungkannya menjadi satu nilai ‘`unsigned int`’.

6. Pembuatan program untuk pembacaan keypad pada Arduino dan LCD

Program ini bertujuan agar input yang diketikan lewat keypad dengan format yang telah ditentukan dapat terbaca oleh Arduino dan dapat ditampilkan oleh LCD.

```

void baca_Keypad() {
    key = keypad.getKey();
    if (key != NULL) {
        if (key == '*' && Menu != 0) {
awal:
            buka_ServoA(_sudut);
            buka_ServoB(_sudut);
            buka_ServoC(_sudut);
            Menu = 0;
            dMenu = 0;
            berat = 0;
            LCD_Out(Header1, Header2, Header3, Header4);
            return;
        }

        if (Menu == 0 && dMenu == 0) {
            Menu = 1;
            berat = 0;
        }
        else {
            if (key != NULL && key != 'A' && key != 'B' &&
key != 'C' && key != 'D') {
                if (key == '0')

```

```

        key = 0;
    if (dKey <= 99999) {
        if (Menu == 4 && dKey >= 999)
            return;
        dKey = 10 * dKey + key;
        return dKey;
    }
}
else if (key == 'C') { //Fungsi hapus nilai
    if (dKey <= 999999)
        dKey = dKey / 10;
    return dKey;
}
else if (key == 'D') {
    if (dMenu == 2 && setMenu != 0 && dKey != 0) {
        simpan_EEPROM(array_alamat[setMenu], dKey);
        dKey = 0;
        setMenu = dKey;
        dMenu = dKey;
        Menu = dKey;
        goto awal;
    }
    if (dMenu == 4 && setMenu == 0 && dKey != 0) {
        // berat_input = dKey;
        berat_input = (float)(dKey * 1000) /
        baca_EEPROM(array_alamat[oldKey]);
        LCD_Out();
        LCD_Out(S1_1, S4_1, S1_1, S1_1);
        dKey = 0;
        Menu = 5;
    }
    return key;
}
}
return key;

```

```

    }
}

```

7. Pembuatan program output/tampilan teks pada layar LCD

```

void LCD_Out(String a = "           ",
              String b = "           ",
              String c = "           ",
              String d = "           ") {
    lcd.setCursor(0, 0);
    lcd.print(a);
    lcd.setCursor(0, 1);
    lcd.print(b);
    lcd.setCursor(0, 2);
    lcd.print(c);
    lcd.setCursor(0, 3);
    lcd.print(d);
}

```

Program void LCD_Out berfungsi untuk menampilkan teks pada layar LCD dengan format yang ditentukan. Dengan menggunakan fungsi LCD_Out() dapat mempermudah dalam menampilkan teks pada layar LCD dengan mengatur nilai string a, b, c, dan d sesuai kebutuhan. Nilai string tersebut akan ditampilkan pada baris-baris yang sesuai pada layar LCD. Jika tidak ada nilai yang diberikan untuk parameter-parameter tersebut, string kosong dengan panjang 20 karakter akan ditampilkan pada baris yang bersangkutan.

8. Pembuatan program untuk inisialisasi sensor *load cell*

```

void init_timbangan(void) {
    timbangan.begin(pin_DT, pin_SCK);

    timbangan.read();

    timbangan.read_average(20);

    timbangan.get_units(5);
}

```

```

timbangan.set_scale(kBerat);

timbangan.tare();

timbangan.read();

timbangan.read_average(20);

timbangan.get_units(5);
}

```

Program void `init_timbangan` berfungsi untuk menginisialisasi agar timbangan dapat memulai komunikasi dengan Arduino, melakukan konfigurasi awal pada timbangan, seperti membaca data awal, melakukan kalibrasi, mengatur faktor skala dan mendapatkan pembacaan bobot yang lebih stabil dan akurat

9. Pembuatan program untuk mengatur gerak motor servo dengan membaca berat beras

Program void `Baca_Berat` berfungsi untuk membaca berat pada sensor *load cell* dan membandingkannya dengan nilai yang ditentukan ('A')

```

void baca_Berat(int16_t A = 0) {
  uint8_t sudut;
  uint8_t sudut2;

  while (berat < A) {
    if (key == '*')
      break;
    berat = timbangan.get_units();
    if (berat <= 0)
      berat = 0;

    if ((A - berat) <= float(A / 4))
      sudut = _sudut3;
    else if ((A - berat) <= float(A / 2))

```

```
        sudut = _sudut2;
    else
        sudut = _sudut1;

    if ((A - berat) <= float(A / 4))
        sudut2 = _sudut3ab;
    else if ((A - berat) <= float(A / 2))
        sudut2 = _sudut2ab;
    else
        sudut2 = _sudut1ab;

    if (oldKey == 1)
        buka_ServoA(sudut2);
    if (oldKey == 2)
        buka_ServoB(sudut2);
    if (oldKey == 3)
        buka_ServoC(sudut);

    lcd.setCursor(12, 2);
    lcd.print(berat, 0);
    lcd.print(Satuan);
    baca_Keypad();
#ifdef (ENABLE_USER_DEBUG)
    task_debug();
#endif
}

    sudut = _sudut;
    buka_ServoA(_sudut);
    buka_ServoB(_sudut);
    buka_ServoC(_sudut);

    delay(3000);
    berat = 0;
```

```

berat_input = 0;
Menu = 0;
dMenu = 0;
LCD_Out(Header1, Header2, Header3, Header4);
}

```

Di akhir while loop, bobot pada timbangan mencapai atau melebihi nilai 'A'. Kemudian sudut servo diatur ke nilai '_sudut' (90). Buka servo 1, 2 dan 3 ke sudut yang diinginkan. Kemudian gunakan fungsi delay(3000) untuk menundanya selama 3000 milidetik (3 detik). Setelah itu variabel berat, berat_input, Menu, dan dMenu direset ke nilai awalnya atau nol. Terakhir, panggil fungsi 'LCD_Out()' dengan argumen 'Header1', 'Header2', 'Header3', 'Header4' untuk menampilkan teks di lokasi yang ditentukan pada layar LCD.

10. Pembuatan program inisialisasi pada sistem dispenser beras

Program void Dispenser_init berfungsi untuk melakukan inisialisasi dan konfigurasi awal pada sistem dispenser.

```

void Dispenser_init(void) {
    lcd.init();
    lcd.backlight();

    ServoA.attach(pin_ServoA);
    ServoB.attach(pin_ServoB);
    ServoC.attach(pin_ServoC);

    buka_ServoA(_sudut);
    buka_ServoB(_sudut);
    buka_ServoC(_sudut);

    init_timbangan();

    LCD_Out(Header1, Header2, Header3, Header4);
}

```

Fungsi ini melakukan konfigurasi awal pada sistem dispenser, seperti menginisialisasi LCD, mengaktifkan backlight, menghubungkan servo-servo

dengan pin yang ditentukan, membuka servo-servo pada posisi awal ('_sudut'), melakukan inisialisasi dan konfigurasi awal pada timbangan, serta menampilkan teks pada layar LCD.

11. Pembuatan program yang mengatur *flow* dari kinerja dispenser beras

```
void Dispenser_control(void) {
    switch (Menu) {
        case 1:
            if (dMenu == 0) {
                LCD_Out(S1_1, S1_2, S1_3, S1_4);
                dMenu = 1;
            }

            if (dKey == 1)
                Menu = 2;
            else if (dKey == 2)
                Menu = 3;
            dKey = 0;
            break;

        case 2:
            if (dMenu == 1) {
                LCD_Out(S1_1a, array_beras[1], array_beras[2],
array_beras[3]);
                for (int x = 1; x <= 3; x++) {
                    lcd.setCursor(8, x);
                    lcd.print(Rupiah);
                    lcd.print(baca_EEPROM(array_alamat[x]));
                }
                dMenu = Menu;
            }
            if (dMenu == 2 && setMenu == 0) {
                if (dKey == 1 || dKey == 2 || dKey == 3) {
                    setMenu = dKey;
                    lcd.clear();
                }
            }
        }
    }
}
```

```

S1_4b);
    LCD_Out(array_beras[dKey], S1_2b, S1_3b,
    lcd.setCursor(8, 1);
    lcd.print(Rupiah);
    lcd.print(baca_EEPROM(array_alamat[dKey]));
    dKey = 0;
    buf_dKey = dKey;
    }
}
if (dMenu == 2 && setMenu != 0) {
    if (dKey != buf_dKey) {
        lcd.setCursor(13, 2);
        lcd.print(dKey);
        lcd.print(" ");
        buf_dKey = dKey;
    }
}
break;

case 3:
    if (dMenu == 1) {
        LCD_Out(S2_1, array_beras[1], array_beras[2],
array_beras[3]);
        for (int x = 1; x <= 3; x++) {
            lcd.setCursor(8, x);
            lcd.print("(");
            lcd.print(Rupiah);
            lcd.print(baca_EEPROM(array_alamat[x]));
            lcd.print(")");
        }
        dMenu = Menu;
    }
    if (dMenu == 3) {
        if (dKey == 1 || dKey == 2 || dKey == 3) {
            Menu = 4;

```

```

    }
}
break;

case 4:
    if (dMenu == 3) {
        lcd.clear();
        LCD_Out(array_beras[dKey], S3_1);
        oldKey = dKey;
        dKey = 0;
        lcd.setCursor(0, 2);
        lcd.print(Rupiah);
        lcd.print(dKey);
        // lcd.print(Satuan);
        dMenu = Menu;
    }
    if (dKey != buf_dKey) {
        if (dKey >= 0 && dKey <= 99999) {
            lcd.setCursor(3, 2);
            lcd.print(dKey);
            // lcd.print(Satuan);
            buf_dKey = dKey;
        }
    }
    break;

case 5:
    if (dMenu == 4) {
        init_timbangan();
        lcd.clear();
        LCD_Out(array_beras[oldKey], S5_1, S5_2);
        dKey = 0;
        lcd.setCursor(12, 1);
        lcd.print(berat_input, 0);
    }
}

```

```
        lcd.print(Satuan);  
        lcd.setCursor(12, 2);  
        lcd.print(berat, 0);  
        lcd.print(Satuan);  
        dMenu = Menu;  
    }  
    baca_Berat(berat_input);  
    break;  
}  
}
```

Program void `Dispenser_control` digunakan untuk mengontrol sistem dispenser berdasarkan nilai variabel 'Menu' yang ditentukan. Variabel 'Menu' yang dipakai sesuai dengan flowchart yang telah dibuat seperti pada gambar 3.3.

BAB IV

PENGUJIAN DAN ANALISA

Setelah seluruh proses pembuatan dispenser beras eceran otomatis dilakukan, tahapan selanjutnya adalah pengujian dan analisa. Pengujian dilakukan untuk mendapatkan data spesifikasi, agar jika terdapat sebuah kerusakan atau kesalahan dapat dilakukan pelacakan kesalahan. Untuk mendapatkan data yang sesuai dan akurat diperlukan beberapa kali percobaan.

4.1. Peralatan Yang Digunakan

Alat dan bahan yang digunakan pada pengukuran dan pengujian perangkat keras ini adalah multimeter digital, timbangan digital, rangkaian Catu Daya, *Step-Down Module* LM2596, Arduino Uno R3, sensor load cell, LCD 20x4 dengan I2C, *keypad* 4x4, motor servo.

4.2. Prosedur Pengukuran dan Pengujian

Langkah – langkah yang dilakukan pada saat proses pengujian sistem dispenser beras eceran otomatis, agar bisa mendapatkan data yang benar dan sesuai diperlukan prosedur – prosedur yang harus disiapkan, yaitu :

1. Mempersiapkan dan memastikan semua peralatan yang akan digunakan untuk pengujian dalam kondisi baik.
2. Menghubungkan mikrokontroler dengan laptop untuk dilakukan monitoring pada serial monitor Arduino IDE.
3. Melakukan pengukuran tegangan AC dan DC pada tiap rangkaian dengan menggunakan multimeter digital.
4. Melakukan pencatatan dari hasil pengukuran menggunakan multimeter digital.
5. Melakukan pengujian pada sensor *load cell* untuk mengetahui nilai berat yang ditampilkan oleh LCD.
6. Melakukan Pengujian terhadap keakuratan sudut 3 Motor servo saat membuka dan menutup katup beras.
7. Mencatat dan mendokumentasikan hasil dari pengujian yang telah dilakukan.

8. Menganalisa dari hasil pengujian dengan membuat tabel.

4.3. Pengujian Komponen

Pengujian komponen bertujuan agar fungsi komponen dapat berjalan dengan optimal, menguji adanya kesalahan pada sistem dan mengetahui besarnya tegangan masukan maupun tegangan keluaran pada rangkaian tersebut. Berikut hasil dari pengukuran :

4.3.1. Catu Daya

Sumber daya yang digunakan pada dispenser beras eceran otomatis adalah *power supply* 12VDC yang memberikan suplai daya ke semua komponen elektronika yang terdapat pada alat. Pengukuran yang dilakukan pada catu daya berupa pengukuran tegangan masukan dan tegangan keluaran. Pengukuran ini bertujuan agar tegangan masukan untuk rangkaian alat tidak mengalami *over voltage*, yang menyebabkan rusaknya komponen dan gagalnya operasi sistem. Multimeter digunakan untuk mengukur dan membandingkan nilai tegangan pada *voltmeter* VAC dan *voltmeter* VDC, sehingga menghasilkan selisih. Berikut adalah besar nilai pengukuran dan pengujian dalam rangkaian :

Tabel 4. 1. Hasil Pengujian Tegangan Catu Daya

Pengujian	Tegangan Sumber (VAC)	Tegangan Tercantum (Volt)	Tegangan Terukur (Volt)	Selisih (Volt)	Error (%)
1	212	12	11.6	0.4	3.3%
2	212	12	11.6	0.4	3.3%
3	212	12	11.6	0.4	3.3%



Gambar 4. 1. Tegangan Keluaran

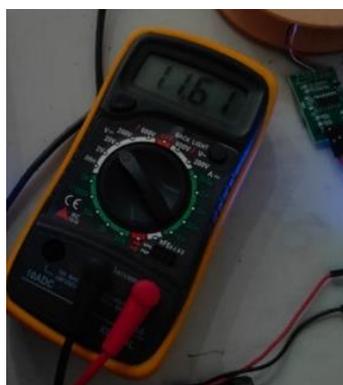
Berdasar tabel di atas, *power supply* merespon tegangan kerja 12V. Terdapat perbedaan tegangan dari *output* yang terbaca dengan tegangan tecantum, sehingga memiliki *error* sebesar 3.3%. Penyebab terjadinya *error* yaitu seringnya penggunaan *power supply*, sehingga menyebabkan penurunan kemampuan. Dari hasil yang didapatkan, pengukuran *power supply* dengan *voltmeter* sesuai dan bekerja dengan baik.

4.3.2. Stepdown LM2596

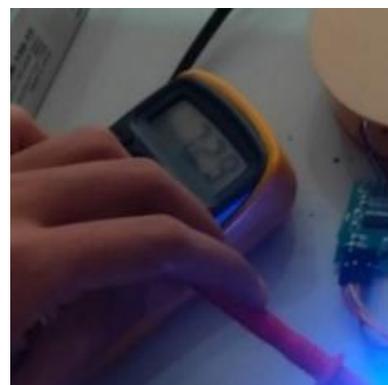
Pengujian *Stepdown* ini bertujuan untuk mengetahui tegangan yang nantinya digunakan sebagai sumber catu daya utama pada Arduino dan komponen lainnya. Tegangan masukan dari *power supply* 12VDC akan diturunkan menjadi sekitar 7VDC sebagai sumber tegangan Arduino dan komponen lainnya.

Tabel 4. 2. Hasil Pengujian Tegangan *Stepdown* LM2596

Pengujian	Tegangan Catu Daya (volt)	Tegangan Pada Alat (volt)	Tegangan Terukur (volt)	Selisih (volt)	Error (%)
1	11.6	7.2	7.2	0	0
2	11.6	7.2	7.2	0	0
3	11.6	7.2	7.2	0	0



(a)



(b)

Gambar 4. 2. (a) Tegangan Masukan *Stepdown* (b) Tegangan Keluaran *Stepdown*

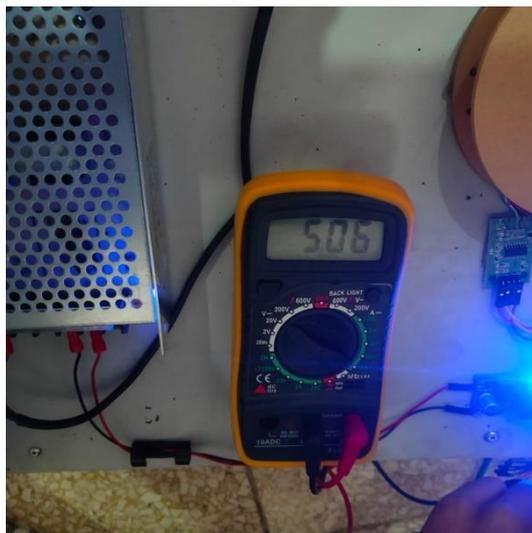
Berdasar tabel di atas pengukuran tegangan masukan dan keluaran pada Stepdown LM2596 mendapatkan hasil yang optimal dan sesuai dengan kebutuhan komponen elektronika dispenser beras eceran otomatis. Tidak ada perbedaan selisih antara tegangan pada alat dan tegangan terukur, sehingga Stepdown LM2596 bekerja dengan baik.

4.3.3. Arduino Uno R3

Pengujian ini bertujuan untuk mengetahui output tegangan Arduino Uno. Pengukuran Arduino dilakukan dengan cara melakukan pengukuran tegangan output pada pin 5V menggunakan voltmeter.

Tabel 4. 3. Hasil Pengujian Tegangan Arduino Uno R3

Beban	Pengujian	V-Out Berdasarkan Spesifikasi (volt)	Hasil Pengukuran (volt)	Error (%)
Tanpa Beban	1	5	5.0	0%
	2	5	5.0	0%
	3	5	5.0	0%
Dengan Beban	1	5	5.0	0%
	2	5	5.0	0%
	3	5	5.0	0%



Gambar 4. 3. Tegangan Keluaran Arduino Uno R3 dengan beban

Berdasarkan tabel di atas, hasil pengujian tegangan Arduino tanpa beban dan dengan beban memiliki *error 0%*, sehingga Arduino bekerja dengan optimal dan berjalan dengan baik karena sudah sesuai dengan kebutuhan perangkat sensor dan komponen lainnya.

4.3.4. LCD 20x4 I2C

Pengujian LCD 20x4 I2C bertujuan untuk mengetahui apakah LCD dapat menampilkan data dan terkoneksi dengan mikrokontroler sesuai dengan program yang telah dibuat.

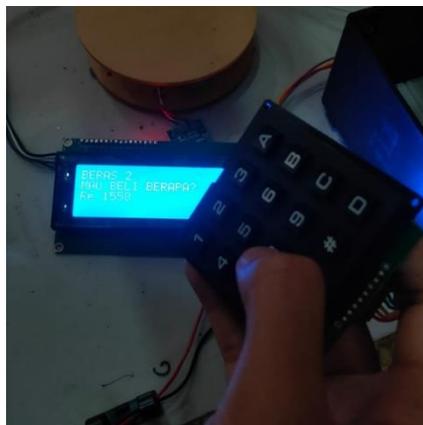


Gambar 4. 4. Hasil Pengujian LCD 20x4 I2C

Berdasarkan gambar 4.4., LCD dapat menampilkan data dan terkoneksi dengan Arduino sesuai program yang telah dibuat.

4.3.5. Keypad 4x4

Pengujian *keypad* 4x4 bertujuan untuk mengetahui apakah *keypad* bekerja dengan baik dan dapat menghasilkan input yang sesuai saat tombol ditekan.



Gambar 4. 5. Hasil Pengujian Keypad 4x4

4.3.6. Motor Servo

Pengujian motor servo bertujuan untuk mengetahui akurasi sudut dari motor servo. Pengukuran akurasi sudut dilakukan dengan bantuan busur derajat guna mengetahui besar pergeseran dari motor servo. Sudut acuan motor servo yang dipakai pada motor servo 1, 2 dan 3 adalah 90° yang menandakan bahwa katup beras tertutup. Motor servo 1 dan motor servo 2 bergerak dari sudut awal 90° ke sudut 180° , sedangkan motor servo 3 bergerak dari sudut awal 90° ke sudut 0° . Hal ini karena perbedaan pemasangan lokasi *bracket* motor servo. Pengujian dilakukan dengan membandingkan sudut motor servo dengan sudut pada busur derajat.

a. Motor Servo 1 ($90\text{-}180^\circ$) (Beras 1)

Tabel 4. 4. Hasil Pengujian Motor Servo 1

Sudut Tercantum (derajat)	Pengujian	Sudut Terukur (derajat)	Selisih (derajat)	<i>Error</i>
90	1	90	0	0%
	2	90	0	0%
	3	90	0	0%
120	1	119	1	3.3%
	2	119	1	3.3%
	3	119	1	3.3%
150	1	147	3	5%
	2	147	3	5%
	3	147	3	5%
180	1	175	5	5.5%
	2	175	5	5.5%
	3	175	5	5.5%
Rata-Rata			2.25	3.4%

b. Motor Servo 2 (90-180)° (Beras 2)

Tabel 4. 5. Hasil Pengujian Motor Servo 2

Sudut Tercantum (derajat)	Pengujian	Sudut Terukur (derajat)	Selisih (derajat)	Error
90	1	90	0	0%
	2	90	0	0%
	3	90	0	0%
120	1	120	0	0%
	2	120	0	0%
	3	120	0	0%
150	1	149	1	1.6%
	2	149	1	1.6%
	3	149	1	1.6%
180	1	175	2	2.2%
	2	175	2	2.2%
	3	175	2	2.2%
Rata-Rata			0.75	0.9%

c. Motor Servo 3 (0-90)° (Beras 3)

Tabel 4. 6. Hasil Pengujian Motor Servo 3

Sudut Tercantum (derajat)	Pengujian	Sudut Terukur (derajat)	Selisih (derajat)	Error
90	1	90	0	0%
	2	90	0	0%
	3	90	0	0%

Tabel 4. 7. Lanjutan Hasil Pengujian Motor Servo 3

Sudut Tercantum (derajat)	Pengujian	Sudut Terukur (derajat)	Selisih (derajat)	Error
60	1	59	1	3.3%
	2	59	1	3.3%
	3	59	1	3.3%
30	1	28	2	3.3%
	2	28	2	3.3%
	3	28	2	3.3%
0	1	4	4	4.4%
	2	4	4	4.4%
	3	4	4	4.4%
Rata-Rata			1.75	2.75%



Gambar 4. 6. Pengukuran Sudut Motor Servo Menggunakan Busur Derajat

Berdasarkan tabel di atas, semakin besar interval sudut dari sudut awal, maka semakin besar juga error yang dihasilkan, didapatkan rata-rata *error* pada motor 1 = 3.4%, motor 2 = 0.9% dan motor 3 = 2.75%. Hal ini disebabkan karena penurunan performa dari motor servo yang dapat dipengaruhi dari faktor lingkungan.

4.3.7. Sensor *Load Cell*

Pengujian sensor *load cell* bertujuan untuk mengetahui apakah sensor dapat berfungsi dengan baik dan menghasilkan pembacaan yang akurat saat diberikan beban. Pengujian dilakukan dengan mengukur tingkat ketelitian sensor *load cell* menggunakan modul HX711 sebagai penguat dan tanpa modul HX711 serta menguji keberhasilan pembacaan sensor *load cell* sesuai dengan tingkat ketelitian yang diperoleh. Tingkat ketelitian sensor *load cell* memakai modul HX711 dapat diukur berdasar persamaan rumus 2.4 berikut:

$$\frac{\text{Operating Voltage}}{\text{Data Accuracy}} \times \text{Rated Load Load Cell}$$

$$\frac{5V}{24 \text{ bit}} \times 5kg = \frac{5V}{2^{24}} \times 5000g$$

$$\frac{5V}{16777216} \times 5000g = 0.0014 g$$

Dari rumus tersebut didapatkan hasil tingkat ketelitian sensor *load cell* menggunakan HX711 adalah 0.0014 gram.

Jika sensor *load cell* tidak memakai modul HX711, maka sensor *load cell* akan beroperasi menggunakan ADC yang tersedia pada Arduino UNO R3 dengan *data accuracy* sebesar 10 bit. Dengan menggunakan persamaan rumus 2.4, maka akan didapatkan hasil berikut:

$$\frac{\text{Operating Voltage}}{\text{Data Accuracy}} \times \text{Rated Load Load Cell}$$

$$\frac{5V}{10 \text{ bit}} \times 5kg = \frac{5V}{2^{10}} \times 5000g$$

$$\frac{5V}{1024} \times 5000g = 24.4g$$

Dari rumus tersebut didapatkan hasil tingkat ketelitian sensor *load cell* tanpa menggunakan HX711 adalah 24.4 gram.

Oleh karena itu, penggunaan modul HX711 dapat meningkatkan tingkat ketelitian pada sensor *load cell* yang menyebabkan pembacaan nilai berat menjadi lebih akurat.

Pengujian berikutnya yaitu menghitung persentase keberhasilan dan kesalahan pada sensor *load cell*, dihitung berdasarkan persamaan rumus 2.3.

Tabel 4. 8. Hasil Pengujian Sensor *Load cell* Berat 47 gram

Pengujian	Pengukuran Pada Timbangan Digital (gram)	Penguukuran Pada Load Cell (gram)	Selisih (gram)	Error
1	47	47	0	0%
2	47	47	0	0%
3	47	47	0	0%

Tabel 4. 9. Hasil Pengujian Sensor *Load cell* Berat 340 gram

Pengujian	Pengukuran Pada Timbangan Digital (gram)	Penguukuran Pada Load Cell (gram)	Selisih (gram)	Error
1	340	340	0	0%
2	340	341	1	0.29%
3	340	340	0	0%

Tabel 4. 10. Hasil Pengujian Sensor *Load cell* Berat 721 gram

Pengujian	Pengukuran Pada Timbangan Digital (gram)	Penguukuran Pada Load Cell (gram)	Selisih (gram)	Error
1	721	721	0	0%
2	721	721	0	0%
3	721	722	1	0.13%

Tabel 4. 11. Rata-Rata Hasil Pengujian Sensor *Load cell*

Pengujian	Berat (gram)	Selisih Rata-Rata (gram)	Rata-Rata Error
1	47	0	0%
2	340	0.3	0.09%
3	721	0.3	0.04%
Rata-Rata		0.2	0.04%

**Gambar 4. 7.** Pengukuran Berat Menggunakan Timbangan Konvensional dan Sensor *Load Cell*

Berdasarkan pengukuran tingkat ketelitian dan tabel di atas didapatkan rata-rata *error* sebesar 0.04%. Sensor *load cell* bekerja dengan baik dan akurat sesuai dengan tingkat ketelitian sensor yaitu 0.0014 gram dan dapat membaca sesuai dengan berat dari beban. Sensor dapat membaca beban dengan kelipatan 1 gram. Berbeda jika tanpa menggunakan modul HX711, maka sensor hanya bisa membaca beban dengan kelipatan 24.4 gram, karena tingkat ketelitian sensor *load cell* tanpa modul HX711 adalah 24.4 gram. Dengan demikian kalibrasi sensor *load cell* pada program Arduino sudah tepat dan sensor *load cell* berfungsi dengan optimal.

4.4. Pengujian Alat Keseluruhan

Pengujian ini dilakukan dalam dua tahap, pertama dengan cara melakukan perubahan harga pada beras yang akan disimpan dalam EEPROM pada Arduino, kedua dengan melakukan pembelian pada masing-masing jenis beras dengan

input tiga harga berbeda. Input harga tersebut akan dikonversi secara otomatis berdasarkan program yang telah dibuat menjadi berat (gram). Konversi didasarkan pada rumus konversi harga beras berdasar rumus sistem linier dua variabel seperti yang tercantum pada persamaan rumus 2.1 dan 2.2.

Implementasi pada program :

```
if (dMenu == 4 && setMenu == 0 && dKey != 0) {
    // berat_input = dKey;
    berat_input = (float)(dKey * 1000)/
    baca_EEPROM(array_alamat[oldKey]);
    LCD_Out();
    LCD_Out(S1_1, S4_1, S1_1, S1_1);
    dKey = 0;
    Menu = 5;
}
```

kemudian dilakukan pengujian ulang sebanyak tiga kali, sehingga dapat diketahui presentase respon dan keakuratan dispenser beras.

4.4.1. Pengujian Perubahan Harga Beras

Pengujian ini dilakukan dengan cara merubah harga beras pada pilihan menu “Set Harga Beras” di tampilan awal layar LCD 20x4. Kemudian pilih jenis beras yang harganya akan dirubah.

Tabel 4. 12. Hasil Pengujian Perubahan Harga Beras

Jenis Beras	Pengujian	Harga Awal (Rupiah)	Harga Akhir (Rupiah)	Hasil Percobaan
Beras 1	1	Rp13.400	Rp12.000	Berhasil
	2	Rp12.000	Rp11.000	Berhasil
	3	Rp11.000	Rp13.400	Berhasil
Beras 2	1	Rp14.000	Rp13.000	Berhasil
	2	Rp13.000	Rp13.400	Berhasil
	3	Rp13.400	Rp14.000	Berhasil
Beras 3	1	Rp15.000	Rp14.700	Berhasil
	2	Rp14.700	Rp13.000	Berhasil
	3	Rp13.000	Rp15.000	Berhasil

Data yang diperoleh dari tabel diatas kembali diuji pada masing-masing beras dengan menggunakan input harga Rp.2.000, agar diketahui apakah harga yang telah berhasil dirubah dapat terkonversi secara otomatis menjadi berat beras sesuai rumus sistem linear dua variabel yang digunakan.

a. Beras 1 (Rp13.400/kg)

Tabel 4. 13. Hasil Pengujian Konversi Harga Beras 1

Input Harga Beras (Rupiah)	Pengujian	Harga Awal (Rupiah)	Harga Akhir (Rupiah)	Konversi Harga ke Berat Awal (Gram)	Konversi Harga ke Berat Akhir (Gram)
Rp2.000	1	Rp13.400	Rp12.000	149	166
	2	Rp12.000	Rp11.000	166	181
	3	Rp11.000	Rp13.400	181	149

b. Beras 2 (Rp14.000/kg)

Tabel 4. 14. Hasil Pengujian Konversi Harga Beras 2

Input Harga Beras (Rupiah)	Pengujian	Harga Awal (Rupiah)	Harga Akhir (Rupiah)	Konversi Harga ke Berat Awal (Gram)	Konversi Harga ke Berat Akhir (Gram)
Rp2.000	1	Rp14.000	Rp13.000	142	153
	2	Rp13.000	Rp13.400	153	149
	3	Rp13.400	Rp14.000	149	142

c. Beras 3 (Rp15.000/kg)

Tabel 4. 15. Hasil Pengujian Konversi Harga Beras 3

Input Harga Beras (Rupiah)	Pengujian	Harga Awal (Rupiah)	Harga Akhir (Rupiah)	Konversi Harga ke Berat Awal (Gram)	Konversi Harga ke Berat Akhir (Gram)
Rp2.000	1	Rp15.000	Rp14.700	133	136
	2	Rp14.700	Rp13.000	136	153
	3	Rp13.000	Rp15.000	153	133

Berdasarkan tabel diatas diperoleh rata-rata keberhasilan sebesar 100%. Dengan demikian sistem perubahan harga beras pada dispenser penjual beras eceran otomatis berfungsi dan bekerja dengan sangat baik. Ini membuktikan jika harga yang dirubah dapat otomatis berjalan mengikuti rumus sistem linear dua variabel yang digunakan.

4.4.2. Pengujian Pembelian Beras

Pengujian ini dilakukan dengan cara mengisi input harga beras pada pilihan menu “Beli Beras” di tampilan awal layar LCD 20x4. Kemudian pilih jenis beras yang akan dibeli, lalu input harga beli beras. Harga beli beras akan otomatis terkonversi menjadi berat (gram) beras.

a. Beras 1 (Rp13.400/kg)

Tabel 4. 16. Hasil Pengujian Pembelian Beras 1 (Rp13.400/kg)

Harga	Pengujian	Berat Konversi (gram)	Berat Output (gram)	Selisih (gram)	Error
Rp2.000	1	149	151	2	1.3%
	2	149	151	2	1.3%
	3	149	150	1	0.6%

Tabel 4. 17. Lanjutan Hasil Pengujian Pembelian Beras 1 (Rp13.400/kg)

Harga	Pengujian	Berat Konversi (gram)	Berat Output (gram)	Selisih (gram)	Error
Rp7.500	1	560	563	3	0.5%
	2	560	562	2	0.3%
	3	560	562	2	0.3%
Rp15.500	1	1157	1161	4	0.3%
	2	1157	1160	3	0.2%
	3	1157	1161	4	0.3%
Rata-Rata				1.9	0.3%

b. Beras 2 (Rp14.000/kg)

Tabel 4. 18. Hasil Pengujian Pembelian Beras 2 (Rp14.000/kg)

Harga	Pengujian	Berat Konversi (gram)	Berat Output (gram)	Selisih (gram)	Error
Rp2.000	1	143	146	3	2.0%
	2	143	145	2	1.3%
	3	143	145	2	1.3%
Rp7.500	1	536	540	4	1.3%
	2	536	538	2	0.3%
	3	536	538	2	0.3%
Rp15.500	1	1107	1110	3	0.2%
	2	1107	1109	2	0.1%
	3	1107	1110	3	0.2%
Rata-Rata				2.2	0.5%

c. Beras 3 (Rp15.000/kg)

Tabel 4. 19. Hasil Pengujian Pembelian Beras 3 (Rp15.000/kg)

Harga	Pengujian	Berat Konversi (gram)	Berat Output (gram)	Selisih (gram)	Error
Rp2.000	1	133	146	2	1.5%
	2	133	147	0	0%
	3	133	147	1	0.7%
Rp7.500	1	500	540	3	0.6%
	2	500	538	2	0.4%
	3	500	538	2	0.4%
Rp15.500	1	1033	1110	1	0.1%
	2	1033	1109	0	0%
	3	1033	1110	1	0.1%
Rata-Rata				1	0.3%

Berdasarkan tiga tabel di atas, didapatkan selisih pada masing-masing pembelian beras sebesar 0-4 gram dan rata-rata *error* pada pembelian beras 1 = 0.3%, pembelian beras 2 = 0.5%, dan pembelian beras 3 = 0.3%. Dengan demikian, sistem pembelian pada dispenser beras berjalan baik dan akurat, ini membuktikan jika keseluruhan sistem bekerja dengan baik.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah dilakukan perancangan dan pengujian dengan judul Tugas Akhir “Rancang Bangun Dispenser Penjual Beras Eceran Otomatis Berbasis Arduino Uno R3 Menggunakan Sensor *Load Cell*” diperoleh kesimpulan sebagai berikut :

1. Alat dispenser penjual beras eceran otomatis ini dapat bekerja sesuai dengan tahapan proses dan perancangan alat.
2. Sensor *load cell* pada sistem dispenser penjual beras eceran otomatis bekerja dengan baik dan akurat, dengan selisih 0-4 gram dan rata-rata error sebesar 0.4%.
3. Perubahan harga beras pada dispenser penjual beras eceran otomatis ini dapat dilakukan secara langsung pada tampilan menu awal LCD 20x4, tanpa memprogram ulang atau membuka aplikasi terlebih dahulu. Sistem perubahan harga pada beras 1 (Rp13.400), beras 2 (Rp14.000), dan beras 3 (Rp15.000) berfungsi dengan baik dengan tingkat keberhasilan 100%.
4. Pembelian beras eceran pada dispenser penjual beras eceran otomatis ini dapat dibeli dengan harga pembelian minimal Rp2.000 yang akan dikonversi menjadi muatan berat beras.
5. Alat ini dapat mengukur beras dengan maksimal 1kg, menyesuaikan ukuran wadah/toples beras dengan diameter 14cm dan tinggi 13cm.

5.2. Saran

Dari hasil pengujian Alat Dispenser Penjual Beras Eceran Otomatis Berbasis Arduino Uno R3 Menggunakan Sensor *Load Cell*, ditemukan beberapa hal untuk saran penelitian selanjutnya, antara lain sebagai berikut :

1. Dispenser harus dibuat lebih tinggi, agar wadah/toples beras yang dipakai dapat lebih besar, sehingga dapat meningkatkan jumlah output beras dan meningkatkan penjualan beras.
2. Tempat penyimpanan beras harus dibuat sangat rapat, agar beras tidak ada yang jatuh dan keluar.